FinanceVis: A Customer-Centric Visualization Tool for Stock Investments

Submitted April 2024, in partial fulfilment of

the conditions for the award of the degree **BSc (Hons) Computer Science.**

Supervised by Professor Robert S. Laramee

20318199

School of Computer Science

University of Nottingham

I hereby declare that this dissertation is all my own work,

except as indicated in the text:

Signature: *Kaiyi Chen*

Date: 19 / 04 / 2024

Abstract

The evolution of financial markets has necessitated the development of more advanced analytical tools to decipher complex investment data. This thesis introduces a customer-centric financial visualization tool designed specifically for analysing and comparing stock investment performance of a specific user. The tool integrates sophisticated data visualization techniques and user interaction capabilities to assist investors in making informed decisions. This tool is built on a robust platform that utilizes Python libraries and web technologies to present a user-friendly interface comprising multiple interactive views. Key features include Parallel Coordinate Charts for multidimensional financial data exploration, Multiple Line Charts for tracking stock performance over time, and Stacked Bar Charts for detailed transaction and dividend analysis. Each component is designed to provide comprehensive insights into market trends, investment returns, and potential financial risks. A series of case studies were conducted to demonstrate the effectiveness of this tool. These include detecting anomalies in transaction data, comparing capital gains and dividends across mREIT stocks, and assessing the quality of investment decisions based on historical data trends. The tool's capabilities are further enhanced by interactive elements such as linked chart brushing, which synchronizes views between different data representations, and dynamic filtering that allows users to customize data displays according to their specific needs. The tool sets itself apart by offering a depth of analysis typically reserved for more complex financial systems within a more accessible framework. This thesis not only illustrates the functionality of the developed system but also its potential impact on personal investment strategies. It highlights the importance of visual analytics in financial decision-making and paves the way for future advancements in financial visualization tools.

Keywords: Financial Visualization; Finance Visual Analysis; Transaction Data; Stock Investment Visualization; Stock Markets; Investment Comparison

Acknowledgments

First and foremost, I want to express my profound gratitude to Professor Robert Laramee for his exceptional mentorship and expertise during this project. The well-structured meetings and valuable advice have been crucial in shaping this project. His enthusiasm for visualization and dedication to this project have not only inspired me but have also played a significant role in my professional development. Thank you to the whole BSc visualization project group and the FinVis group. The discussions and weekly project meetings have been incredibly inspiring.

Additionally, I would like to extend my heartfelt thanks to my family for their unwavering support and understanding throughout my academic pursuits. Finally, I would like to thank Ian Tan for proofreading the document before submission.

Table of Contents

1 Introduction

The domain of financial technology is undergoing a rapid transformation, significantly driven by the integration and advancement of visual analysis tools (Abad-Segura, E. et al., 2020). These tools play a crucial role in decoding complex behavioural patterns inherent in financial investment transactions, making the intricate and dynamic sphere of financial markets more accessible. Moving away from traditional data management methods that focused primarily on handling vast amounts of data, the modern approach strives to convert raw data into actionable insights (Blyakhman, A., 2022). This shift has given rise to a variety of financial visualization methodologies, enabling stakeholders to discern patterns, trends, and insights with enhanced clarity and precision.

The emergence of tools that can handle such complexities is not just a technical development; it represents a significant shift in how financial information is processed and understood (Shao, C. *et al.,* 2022). These tools can not only simplify complex data but also promote broader access to financial information. This evolution in access and interpretation practices has made financial information more engaging and insightful for a diverse range of stakeholders (Roberts, R. and Laramee, R., 2018). Consequently, the growing field of financial visualization facilitates the decoding of complex datasets and represents a pivotal shift in how financial information is accessed and leveraged.

Continuous innovation in the realm of financial data visualization tool is crucial to align with the rapidly changing market dynamics (Yi et al., 2018). However, the pursuit of more insightful visual analysis is challenged by privacy and security concerns associated with real investment transaction data from customers, which restricts its accessibility for thorough research and analysis. This project benefits from the first open investment transaction data set from an anonymized investor, which resolves these concerns and provides a valuable resource for developing and testing new visualization techniques.

This thesis presents the research, design, implementation and evaluation of a Customer-Centric Visualization Tool for Stock Investments. This tool employs a novel approach to visualize the investment data of a specific investor. Visual analysis techniques that provide insights into the individual datasets and facilitate more informed decisions are implemented, potentially contributing to future improvements in related applications.

1.1 Motivation

In the rapidly evolving landscape of financial technology, the pivotal role of visual analysis tools in elucidating complex market dynamics and uncovering investment opportunities has grown increasingly evident. However, despite the availability of platforms designed to provide overarching views of the stock markets, such as Yahoo Finance, there remains a noticeable gap in the development of customer-centric financial visualization tools. This gap underscores a pressing need for innovative solutions that customise financial data analysis to align with the distinct investment patterns and preferences of individual users, providing tailored insights. This demand highlights the importance of developing tools that not only compile real-time market data but also synthesis this with the unique data profiles of users. By leveraging these tools, investors are equipped with the ability to assess their investment performance, thereby enhancing their capability to evaluate and optimise their investment decisions. Consequently, these tools empower them to make more informed choices in the ever-changing landscape of stock investments with greater confidence.

1.2 Aims and Objectives

The primary goal of this project is to conduct research on visual analysis tools and develop a new customer-centric and interactive tool tailored to personalized financial investment data analysis, which allows the comparison on various aspects such as the evaluation of the good and bad decisions made by the investor, making complex financial information more accessible. This focus extends to employing case studies to assess the efficacy of the tool in facilitating informed investment decisions.

The key objectives of the project are:

● Explore and apply existing tools on the Investment Transaction Dataset, gaining insights for the new tool through analysing their strengths and limitations.

● Develop an interactive visualization platform that enables dynamic visualization of financial data, providing an overview and comparison of the investment performance.

● Integrate real-time stock data from financial APIs, which is tailored to the ones that exist in the Investment Transaction Dataset.

● Implement functionalities that allow for the direct comparison of investment decisions, showing the impact of different market conditions on investment portfolios.

● Conduct testing using several case studies to evaluate the developed tool and provide insights into future developments.

1.3 Challenges

This section delves into the key challenges in developing a Customer-Centric Finance Visualization Tool, including two field challenges and two project challenges.

1.3.1 Field Challenges

1) Access to Individual Investment Data: Collecting personal investment data for a customer-centric financial visualization tool faces obstacles due to privacy concerns and regulatory restrictions. The sensitive trading data is safeguarded by strict privacy measures. Consequently, obtaining these data requires navigating complex legal and ethical frameworks to maintain user confidentiality while still acquiring data that is detailed enough to provide valuable insights. The breadth and depth of data available for analysis will impact the ability of tools to offer personalized insights into stock investment performance. However, with the first authentic, publicly available dataset dedicated to individual stock investment transactions for analysis and research purposes, this challenge is mitigated in this project.

2) Facilitating Comparison for Customized Investment Analysis: The challenge of developing a system that allows effective comparison between decisions from specific customer-centric investment histories as well as providing comprehensive comparisons across an entire investment portfolio is considerable. The complexity arises from the varied nature of transaction data, different investment strategies, and unique stock performance metrics. Aligning these for meaningful comparisons necessitates advanced data integration and normalization. Additionally, designing an intuitive user interface for seamless navigation between detailed transaction analysis and broader portfolio insights adds to the development challenge.

1.3.2 Project Challenges

1) **Integration of Customer and Real-Time Stock Data:** Beyond the obstacles of data access, effectively merging detailed customer investment profiles with real-time stock market data presents complex technical challenges, including data streaming, processing, and analysis. This is because stock trading data is inherently complex, involving various elements like daily trades, dividends, and capital gains. Moreover, the variability in data formats across different financial institutions and platforms complicates data aggregation and standardization processes. The tool must adeptly match correlating real-time market trends with individual user data, leveraging sophisticated analytical frameworks to identify and extract insights tailored to each user.

2) **Ensuring Intuitive User Interaction and Clarity:** The challenge of creating a user-friendly and intuitive interface involves making the complex interplay of real-time market data and individual investment profiles accessible and understandable. Users need to be able to interact with and understand their financial information without feeling overwhelmed, despite the underlying complexity. Achieving this balance necessitates a strategic approach to both processing and visualizing vast, fluctuating market data with precision, ensuring that the tool delivers clear, actionable insights without compromising on the richness of detail.

1.4 Thesis Structure

The thesis is structured based on the suggestion from Bob’s Project Guideline (Laramee, 2021). Following the introduction section, Section 2 is related to the background of this project, including the literature review and investigation of existing systems. This section also introduces the data characteristics. Section 3 details the project specification, including the feature specification and the technology choices. Section 4 is on project plan and timeline. Section 5 introduces the design diagrams of the project. The detailed implementations of each component within the project are described in Section 6, including both basic implementations and enhancements. Section 7 presents three case studies that show the result of evaluation and demonstrates the ability of the system, especially regarding comparison. Section 8 summaries the conclusions of the thesis. Finally, Section 9 discusses the possible extensions to the presented work.

2 Background

The background section comprises three subsections. Section 2.1 introduces the related work and literature within the scope of information and financial visualization. Section 2.2 discusses previous systems already exist in the industry. By conducting a thorough review of the literatures and analysing widely used visualization platforms, a more profound comprehension of how to visualize and analyse customer-centric investment transaction data is developed, laying a solid foundation for future research initiatives. These two sections seek to examine current advancements in financial visualization and ensure the novelty of the tool developed in this project. Section 2.3 delves into the data characteristics of the datasets utilised the project.

2.1 Related Work

This section delves into the existing body of research that forms the foundation of this project, aiming to provide a comprehensive understanding of the current field. The significant contributions and theories are reviewed, and literature gaps are identified. Section 2.1.1 introduces the scope of the literature review. Section 2.1.2 and 2.1.3 provide detailed literature reviews on information visualization surveys and financial visualization research, respectively.

2.1.1 Literature Scope

The scope of the literature to be reviewed was carefully selected under the guidance of the project supervisor, Professor Robert Laramee, an expert in information visualization. According to Laramee (2010), choosing the scope of a literature review is inherently subjective, thus leveraging the supervisor's extensive expertise ensures a well-informed and robust selection. The chosen literature covers a wide range of topics within the domains of financial and information visualization, highlighting recent advancements and methodologies. The literatures are organized into two main sections: the first focuses on visualization surveys, providing a comprehensive overview of the current state of information visualization; the second delves specifically into financial visualization, exploring in-depth studies and applications in this field.

2.1.2 Visualisation Surveys

The "Survey of Surveys" (SoS) for information visualization by McNabb and Laramee (2017) presents a novel and significant advancement in literature review methodology. Their pivotal contribution effectively addresses the challenges posed by the extensive volume of literature and offers a crucial guide for researchers to understand the current state of information visualization. They systematically reviewed over 80 survey papers, highlighting the evolving trends and themes within information visualisation and categorizing a vast array of papers into thematic clusters. These efforts are visually supported by a histogram (Figure 2.1.2.1) that displays the frequency of survey papers published each year, segmented by classification categories like Data Enhancement & Transformation, Visual Mapping & Structure, and others. These thematic categories are adaptations of the stages in the Information Visualization pipeline defined by Card et al. (1999) in "Readings in Information Visualization". The meta-survey also presents a structured overview of well-established areas and those requiring further investigation, serving as a valuable guide for newcomers and a resource for experienced researchers. The Figure 2.1.2.2 demonstrates this overview by focusing on open-research keywords in a word cloud format, representing the frequency of topics addressed in recent surveys. This visualization highlights essential aspects of the research landscape, such as "evaluation", "scalability", and "interaction techniques", underlying the critical need for continuous innovation in visualization techniques and tools and that can adapt to evolving market dynamics and user requirements. Specifically, the study underscores the importance of advancing Interaction Techniques and Tools, pinpointing these areas as essential for future research investigations. This emphasis aligns closely with one of the primary objectives of this project.

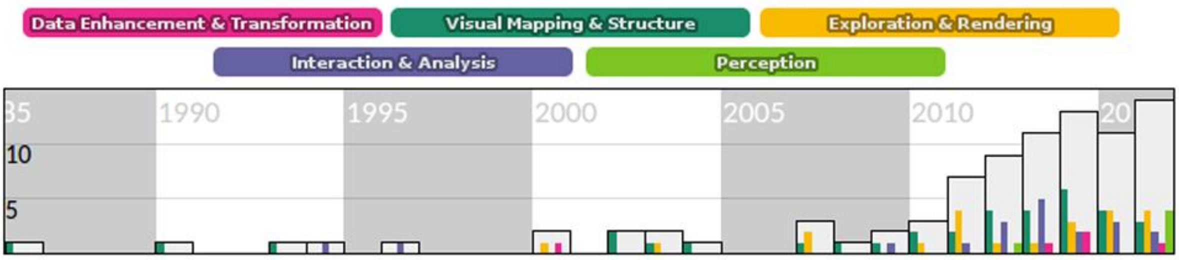


Figure 2.1.2.1 *A histogram providing the annual distribution of survey papers, with the x-axis marking the years and the y-axis indicating the number of surveys published each year. The coloured bars represent a further breakdown of the survey papers based on their given classification dimension. Figure from McNabb and Laramee (2017).*

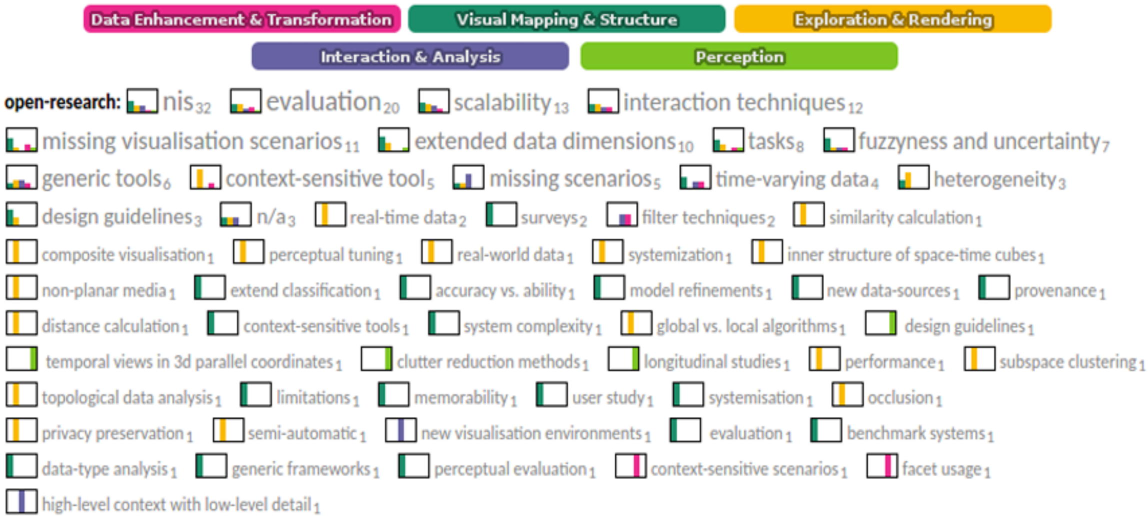


Figure 2.1.2.2: *Open research keywords collected across all recent, reviewed survey papers (2010-2017). Figure from McNabb and Laramee (2017).*

In the work of survey for information visualization books by Rees and Laramee (2019), the authors expand upon the analytical foundation established by the earlier "Survey of Surveys" to delve into the domain of textual resources, presenting the first exhaustive review of books on information visualization and visual analytics. This landmark study fills a crucial gap in scholarly resources by cataloguing and evaluating the wide array of published books in the field, guiding researchers, educators, and practitioners towards the most influential and informative texts. Rees and Laramee (2019) introduced a novel two-level classification system based on book and chapter topics, enabling readers to quickly locate resources relevant to specific areas of interest, whether they are delving into the depths of data enhancement and transformation, visual mapping and structure, or any of the key stages in the information visualization pipeline. Central to their findings, Figure 2.1.2.3 provides a comparative analysis of topics across the surveyed books, indicating the number of pages devoted to each subject within the focus books of this survey. This figure highlights a notable emphasis on visual design and identifies areas with less coverage, such as glyph-based visualization, presentation, and animation. This unique survey not only streamlines the exploration of the extensive literature but also illuminates the special challenges associated with the collection of books, such as searching, browsing, and cost, providing an invaluable resource for navigating the rich body of information visualization books.

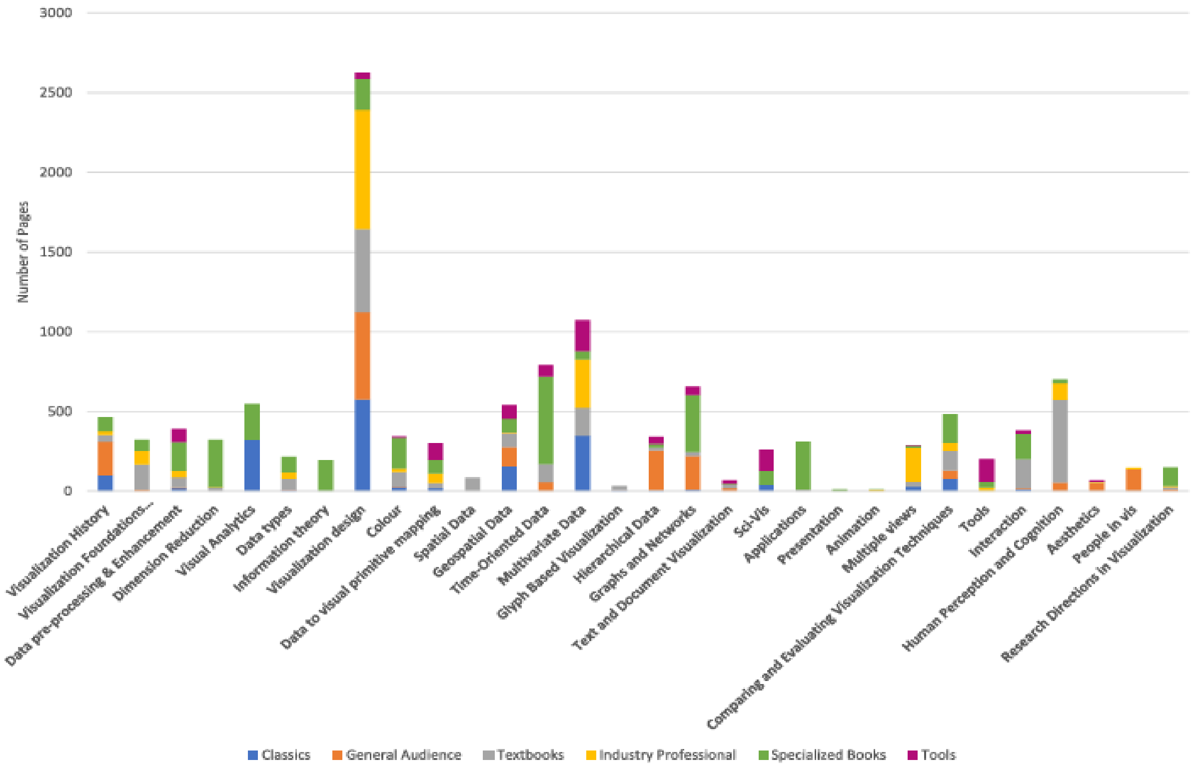


Figure 2.1.2.3: *Total number of pages, from surveyed books, dedicated to each topic, coloured by book classification. Figure from Rees and Laramee (2019).*

Liu et al. (2014) provided a comprehensive exploration into the realm of information visualization in their work of survey on information visualization's recent advances and challenges, which maps out recent progress and identifies ongoing challenges that shape the current landscape. This investigation offers an expansive survey of how visualization techniques have evolved to meet the needs of interpreting complex datasets across various sectors, including finance and business. Through an intricate taxonomy, they categorized the research into areas such as empirical studies, interaction techniques, visualization frameworks, and real-world applications. This methodology affords a structured perspective on the discipline's scope and depth. This taxonomy not only elucidates the current state of research but also emphasizes the crucial role of visual analytics in an era overwhelmed by large volumes of data. Liu et al. (2014) underscore the importance of visual analytics in fields that require in-depth analysis and strategic decision-making. They delved into the field's potential and sketched a roadmap for future innovations in visualization techniques. Their survey emerges as an essential resource, guiding researchers in navigating the complexities of information visualization and its application in analysing intricate data patterns.

The synthesis of insights from these surveys underscores a collective concern towards refining the visualisation tools, which are designed to overcome the existing limitations and better meet the needs of customers seeking to understand their specific financial transactions. According to the findings of Liu et al. (2023), the visualisation tools are categorized based on their focus areas, such as literature, web interfaces, developer orientation, and specialized topics, offering numerous opportunities for investigation and analysis. Based on different kinds of visualisation resources available to researchers, it is a novel approach to analyse the strengths and limitations of the visualization tools based on an investment transaction dataset from a specific customer. This approach is not merely about evaluating existing visualization methods; it is about leveraging the unique properties of the dataset to uncover new possibilities and limitations within the realm of financial visualization. Parallelly, Liu et al. (2023) pointed out the significance of colour in data visualization, with resources like ColorBrewer and Chroma.js Colour Palette Helper offering foundational tools for effective colour mapping. These tools, highlighted within specialized discussions on colour application in visualization, provide essential guidelines for choosing colour schemes that enhance the readability and interpretability of visual data. Especially in financial visualization, where clarity and distinction are paramount, the judicious application of colour mapping resources enables the creation of intuitive and insightful visual representations, facilitating a deeper understanding of complex financial data.

Roberts and Laramee (2018) provided a comprehensive survey on visualising business data, pinpointing the essential role of visualization in interpreting the overwhelming amounts of data generated by modern businesses. Through their analysis, Roberts and Laramee (2018) advocate for the democratization of data interpretation, enabling a broader audience within organizations to leverage visual analytics for decision-making, trend identification, and problem-solving. This approach underscores the shift towards inclusive data analysis practices, broadening the accessibility of complex data insights across different organizational levels. By offering a novel classification, Roberts and Laramee (2018) then organized these findings into three overarching categories: Business Intelligence, Business Ecosystem, and Customer Centric, capturing the predominant themes within the domain. Figure 2.1.2.4 visually summarizes this classification, offering readers an immediate understanding of the landscape of business data visualization literature. By detailing the scope and focus of each category, this study serves as a pivotal guide for navigating the expanding landscape of business visualization literature. This work not only aids in the effective management of voluminous business data but also highlights the importance of visual analytics in facilitating an understanding of complex business environments, underscoring the necessity for continual innovation in visualization tools and techniques that respond to changing business dynamics and user needs.

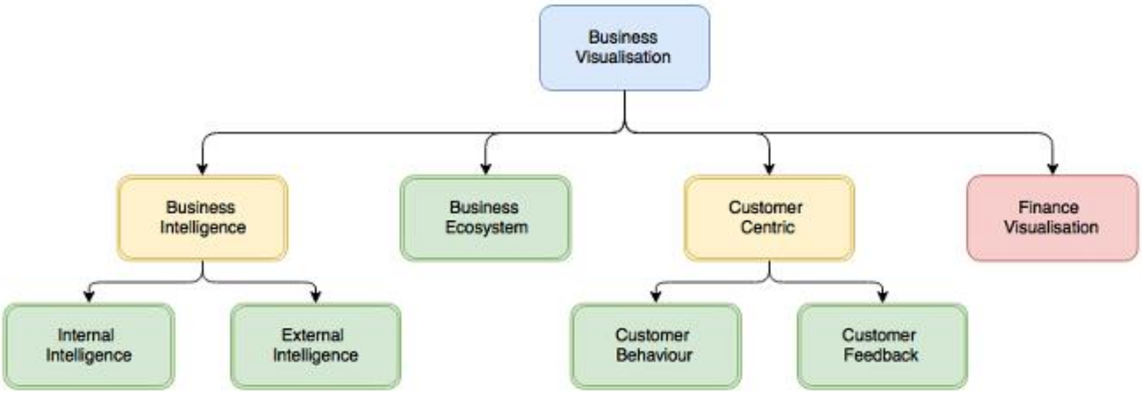


Figure 2.1.2.4: *The top-level hierarchical classification of the literature survey. Green classifications represent leaf nodes, while yellow represents umbrella classifications. Figure from Roberts and Laramee (2018).*

Similar to the survey for the visualisation of business data, the survey on visual analysis approaches for financial data by Ko et al. (2016) delves into the intricate world of financial data through the lens of visual analysis. Much like the challenges faced in visualizing business data, financial data presents its own set of complexities due to its vast volume, diverse nature, and the critical need for precise interpretation. This survey methodically categorizes and assesses financial systems, drawing parallels to existing frameworks while incorporating fresh insights from domain experts. It emphasizes the evolution of visualization and analytics within the finance sector, advocating for a collaborative approach between financial professionals and visualization researchers to develop solutions that tailored to the specific demands of the industry. Highlighting the crucial role of visual analytics in financial data examination, Ko et al. (2016) provided a comprehensive overview of the current landscape and identifies potential research directions. This contribution not only bridges the understanding between financial experts and visual analytics researchers but also establishes a foundation for developing advanced tools that enhance strategic planning in finance, reflecting the ongoing efforts to tackle similar challenges in the field of data visualization.

2.1.3 Finance Visualisation

The "MoneyVis" project by Firat et al. (2023) represented a breakthrough in field of finance visualisation by using the first publicly available bank transaction dataset from an anonymized retail customer to analyse and visually represent financial spending. This pioneering effort not only opens up a new frontier of financial visualization but also established a novel benchmark for the utilization of open, customer-centric datasets in the field. This move towards leveraging anonymized datasets is indicative of a broader shift in the visualization technology in the financial sector. The emergence of datasets like the Money Transaction Dataset underscores the growing need for tools that can not only handle complex and varied financial data but also present it in a way that is both accessible and meaningful to a wide range of users. Through preliminary analyses and visualizations, such as the stacked bar chart (Figure 2.1.3.1) and the sunburst chart (Figure 2.1.3.2) by Firat et al. (2023), "MoneyVis" demonstrates the effectiveness of visual analytics in enhancing the comprehension of specific consumer financial behaviours. These visual representations also highlight the transformative potential of such customer-centric datasets in advancing financial data analysis and visualization, laying the groundwork for future innovation in the sector.

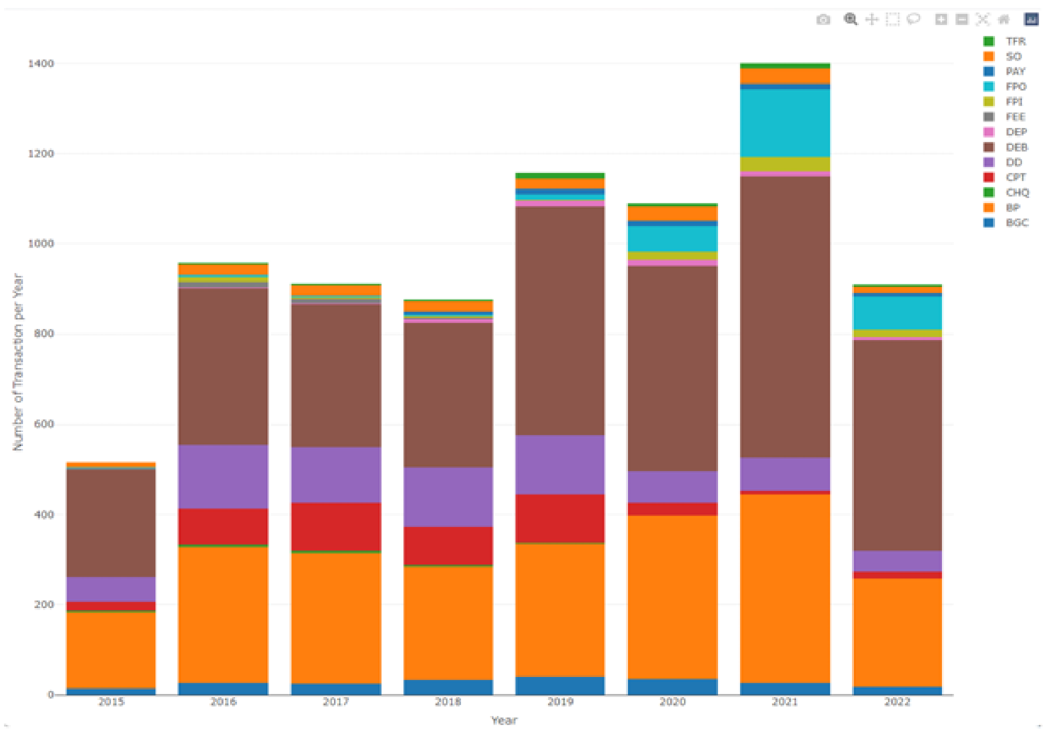


Figure 2.1.3.1: *This chart shows the number of transactions and bank transaction categories in each year from 2015-2022 in the Dataset. Figure from Firat et al. (2023).*

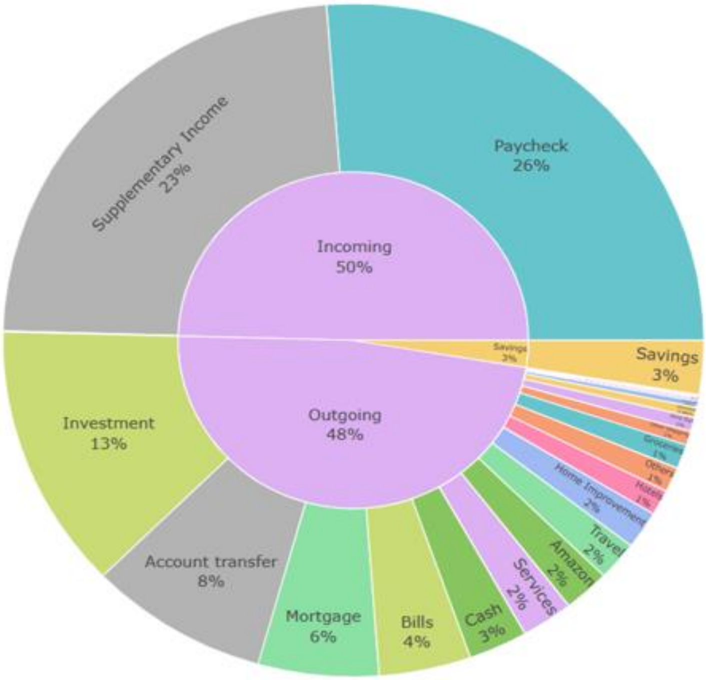


Figure 2.1.3.2: *This chart displays the total incoming and outgoing transactions, based on the total amount not the number of transactions. Figure from Firat et al. (2023).*

"Sabrina" (Figure 2.1.3.3), developed by Arleo et al. (2019), is a visual analytics tool designed to transform the analysis of financial data by creating detailed financial transaction networks. It combines data from individual firms with broader economic indicators and domain knowledge to offer insights into the financial ecosystem over time. This approach enables financial analysts to explore complex economic relationships and trends at multiple levels, from specific companies to entire sectors, showcasing the tool's potential to revolutionize economic data analysis.

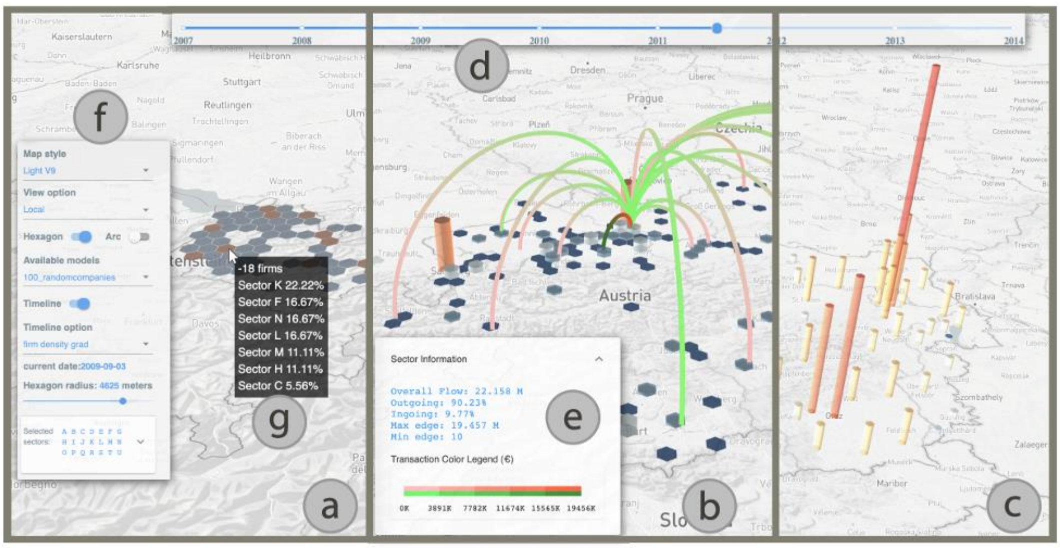


Figure 2.1.3.3: *A screenshot of "Sabrina" from Arleo et al. (2019). The three views (a, b, c) display the firm density and transaction data of an Austrian economy dataset.*

Building on the foundation laid by Sabrina, Arleo et al. (2023) introduced Sabrina 2.0 (Figure 2.1.3.4), an enhanced version of the visual analytics tool designed for a more nuanced exploration of financial data. It extends the capabilities of its predecessor by incorporating a refined pipeline for the generation of firm-to-firm financial transaction networks. This innovation allows for the integration of sector-to-sector transaction data and macroeconomic indicators, offering users the ability to create and compare multiple transaction models under different economic scenarios. The collaboration with domain experts from finance and economics has been deepened, ensuring that the tool's development aligns closely with the needs of professionals in these fields. A qualitative insight-based evaluation with seven domain experts showcases Sabrina 2.0's capacity to facilitate the generation of insights into the dynamics of national economies.

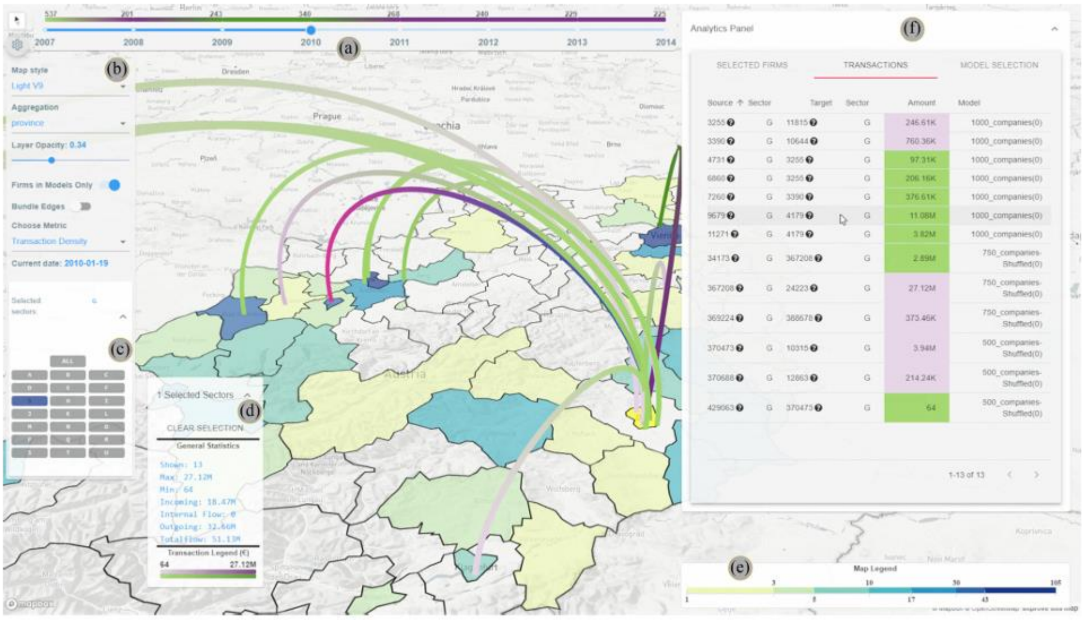


Figure 2.1.3.4: *A screenshot of "Sabrina2.0" from Arleo et al. (2023), showing Timeline, Configuration panel, Selection panel, Detailed box, Colour map and Transaction Table.*

Leite et al. (2018) presented a novel visual analytics tool "EVA", shown in Figure 2.1.3.5, which is designed to enhance fraud detection mechanisms within financial institutions. By integrating VA techniques with traditional data mining and customer profile analysis, EVA provides a multifaceted approach to identifying fraudulent activities. Central to its design is the emphasis on temporal analysis and a customer profile-based scoring system, which together facilitate the detailed examination of transactional data for anomalies indicative of fraud. This tool enables the construction of intricate financial transaction networks, allowing for a granular exploration of financial data across multiple scales.

EVA's utility and efficiency in reducing false positives have been validated through qualitative evaluations with finance and economics experts, highlighting its significant contributions to the field. Moreover, its user-friendly interface exemplifies the tool's comprehensive analytical capabilities, from conducting temporal analyses to generating transaction scores, thus providing analysts with comprehensive tools to delve into and investigate suspicious financial activities effectively.

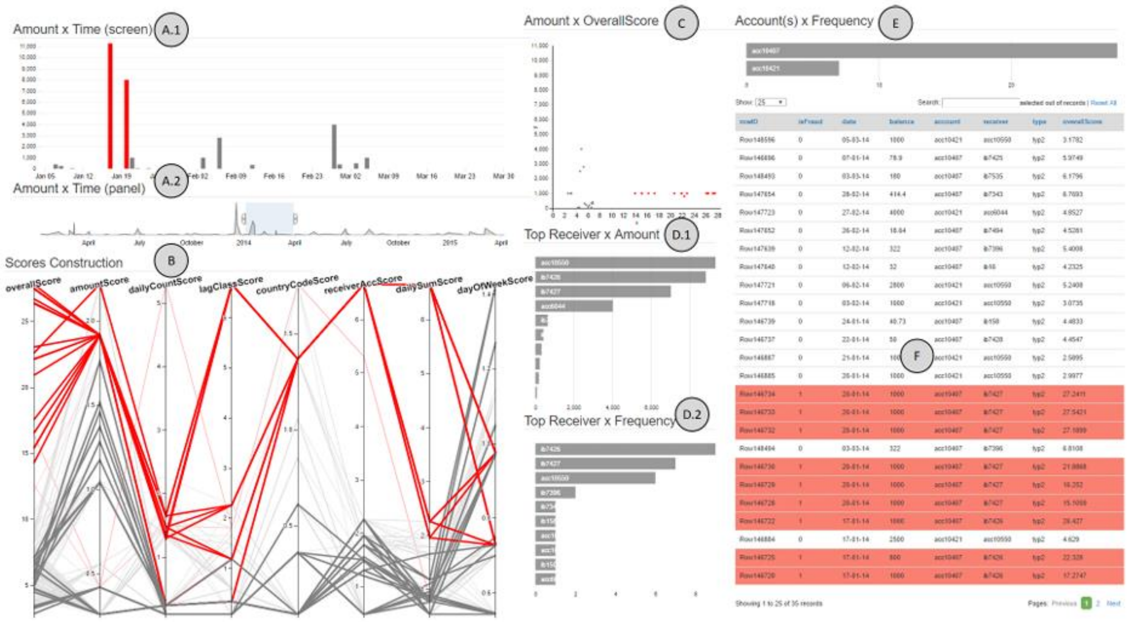


Figure 2.1.3.5: *A screenshot of "EVA" from Leite et al. (2018). (A.1, A.2) Temporal views. (B) Parallel coordinates plot showing score construction. (C) Amount vs Overall Score Scatterplot. (D) Sorted bar chart of the selected account amount. (E) Bar chart as accounts selector showing each account's number of transactions. (F) Table view of raw transaction data. Elements that represent suspicious data are highlighted in red.*

Extending the capabilities established by EVA, Leite et al. (2020) introduced NEVA, which shifts the focus towards the analysis of fraudulent networks, offering a more granular exploration of financial fraud. NEVA capitalizes on the strengths of visual analytics by incorporating a network analysis dimension, enabling the examination of intricate relationships and dependencies within transaction data. This tool features a novel guidance-enriched component for network pattern generation, detection, and filtering, which facilitates investigations into the complex web of transactions that characterize fraudulent networks. While EVA (Figure 2.1.3.5) enhances fraud detection through individual transaction analysis, NEVA (Figure 2.1.3.6) broadens the scope by unravelling the interconnected networks of fraudulent activity, thereby providing a more comprehensive toolkit for financial institutions to detect and combat fraud effectively.

Figure 2.1.3.6: *A screenshot of "NEVA" from Leite et al. (2020). (A) Temporal views. (B) Node-Link view showing the network of the accounts to analyse. (C) Pattern search panel with guidance. (D) History of the clicked nodes. Suspicious data is highlighted.*

Sun et al. (2023) developed "BitAnalysis", an innovative visual analytics system tailored for the intricate examination of Bitcoin wallet transactions. This tool excels in aiding financial regulators and law enforcement officers, enabling them to dive deep into the transaction dynamics. It also emphasises wallet correlation analysis and the tracking of transaction flows. Unique in its approach, BitAnalysis addresses the challenges posed by the pseudonymous nature of Bitcoin. It facilitates a detailed investigation into suspicious activities and interactions between wallets. A comprehensive user study involving cryptocurrency experts validated its effectiveness, highlighting its potential to significantly enhance investigative processes in the financial sector, making it an asset for both monitoring and analysing financial transactions.

Rodriguez and Kaczmarek (2016) conducted a comprehensive exploration of innovative techniques for presenting financial information in a visually engaging and easily interpretable format. The authors delve into the psychology behind effective data visualization, presenting a plethora of real-world applications that guide financial professionals on how to transform complex data sets into intuitive visual narratives. Their work stands out by bridging the gap between traditional financial analysis and the emerging field of data visualization, providing readers with practical strategies to enhance decision-making processes. Through a combination of theoretical insights and hands-on examples, Rodriguez and Kaczmarek (2016) equip readers with the skills needed to leverage the power of visual analytics in the financial sector. Their work is particularly valuable for analysts, investors, and financial managers looking to convey financial concepts and analyses more compellingly. With its clear explanations and focus on actionable knowledge, their work marks a significant contribution to the literature, encouraging a shift towards more accessible and impactful financial communication.

Apart from systems focused on visualizing financial transactions, there are also platforms dedicated to stock visualization. Yue et al. (2019) presented "sPortfolio", a sophisticated visual analytics system tailored for the nuanced analysis of stock portfolios. The system is notably advanced in its ability to provide stratified visual insights across three levels of financial analysis: Risk-Factor, Multiple-Portfolio, and Single-Portfolio levels. Noteworthy is its utilization of four distinct visual components as depicted in Figure 2.1.3.7: the portfolio cluster view, the factor correlation view, the comparison view, and the individual portfolio view. Each view offers a unique perspective on the data, from high-level market trends to detailed individual asset strategies. This granular approach is pivotal in aiding investors to discern and devise robust investment strategies based on multifactorial data insights. The effectiveness of sPortfolio is underscored by a series of case studies involving domain experts, highlighting its practical application and potential for broader adoption in the financial industry. This system represents a significant leap forward in the visual analysis of stock investment strategies, combining detailed data inspection capabilities with user-centric design to support informed decision-making in complex market conditions.

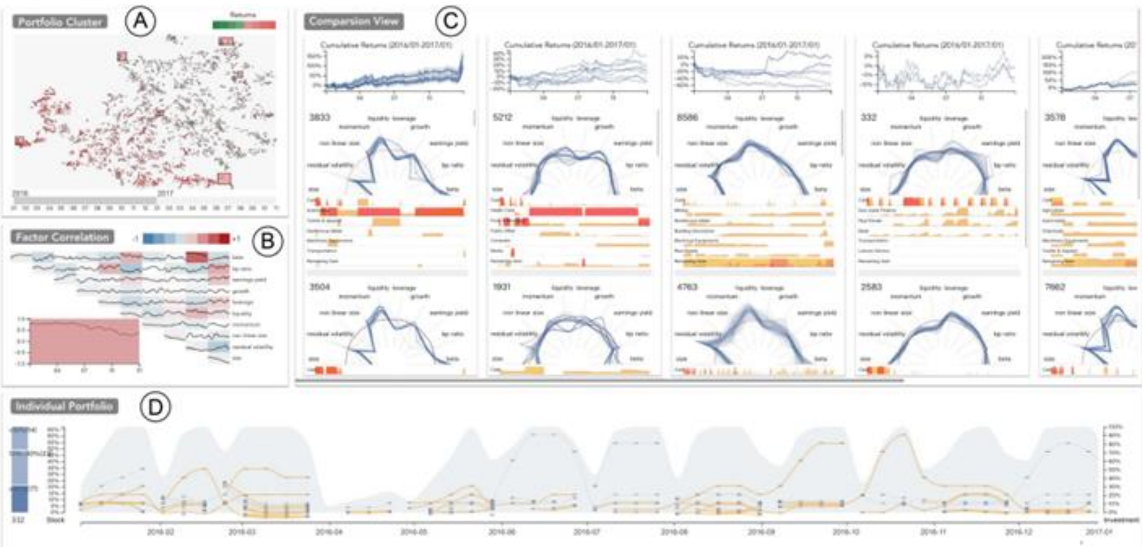


Figure 2.1.3.7: *A screenshot of "sPortfolio" from Yue et al. (2019). (A) Portfolio cluster view. (B) Factor correlation view. (C) Comparison view. (D) Individual portfolio view.*

Lei and Zhang (2010) delved into the complex realm of financial time-series data analysis by advocating a visual analytics approach to decipher sophisticated dynamics of the stock market. Their system is designed to empower individual investors by providing a comprehensive view of market data through several innovative perspectives, including the Market View, Underperforming Stock View, Asset View, and Pattern View. These perspectives enable a more nuanced analysis and informed decision-making process. By identifying the limitations of traditional visual charts that typically focus on single indicators over time, the authors introduce a novel visualization technique using ring charts, as demonstrated in Figure 2.1.3.8. This technique facilitates the observation of financial time-series data movement, allowing for a multi-dimensional exploration of market trends and asset performance. Furthermore, the system integrates interactive visual methods with knowledge discovery techniques to uncover predictable patterns and support prompt investment decisions. This system represents a significant advancement in financial data analysis, offering a holistic and interactive tool for navigating the complexities of the stock market.

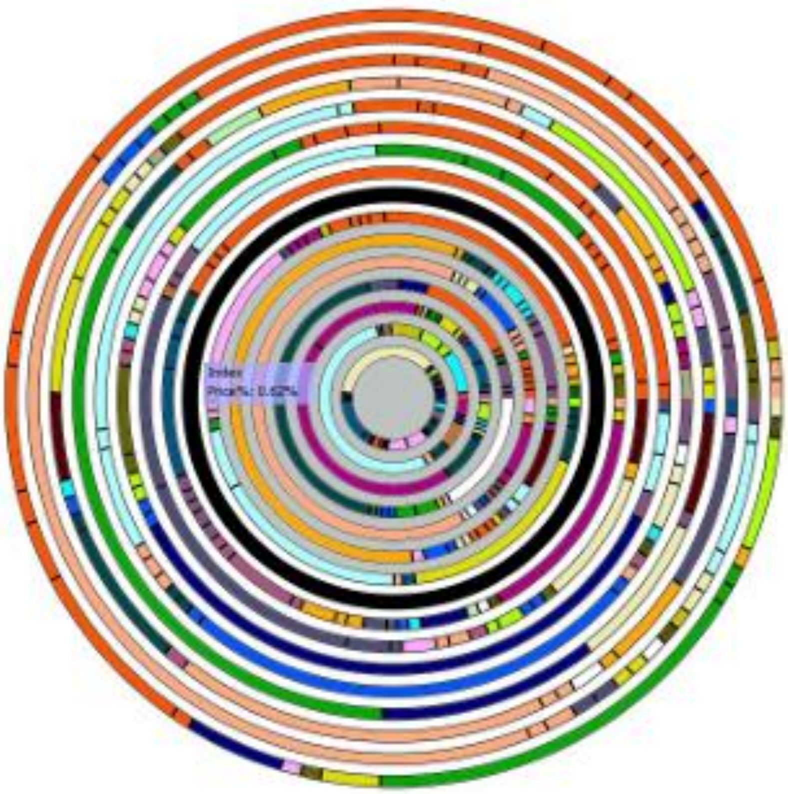


Figure 2.1.3.8: *A screenshot of the ring chart for the stock market from Lei and Zhang (2010). The arc length indicates the size of the capital. Different colours signify different stock sectors. Grey rings indicate stocks that experienced a decline in price, while the outer orange rings denote stocks that performed better than the Hang Seng market index, which is represented by the black ring.*

Guo et al. (2022) introduced “RankFIRST”, a comprehensive visual analytics system designed to enhance factor investment strategies by effectively ranking and visualizing stock time series. The system is particularly noteworthy for its integration of a visual interface that supports the end-to-end factor investing process, including factor analysis, stock selection, and portfolio management. As shown in Figure 2.1.3.9, RankFIRST incorporates several innovative views: a configuration panel for setting investment parameters, a factor view displaying time series for selection, a Treemap that shows the returns of aggregated factors, and a ranked list of stocks. It also includes individual views for detailed stock and portfolio analyses, which allow users to see changes in stock rankings over time and compare the returns of the stock index with the portfolio. These views enable users to interactively explore and manipulate complex financial data to identify optimal investment strategies. The evaluation of RankFIRST, based on full-scale case studies with the Chinese stock market, demonstrates its superior performance in enhancing the decision-making process in factor investing compared to existing tools, marking a significant advancement in the application of visual analytics to financial data. The system’s ability to demonstrate complex financial datasets through intuitive visual representations and interactive elements allows users to make more informed decisions, aligning with the needs of both seasoned investors and those new to investing.

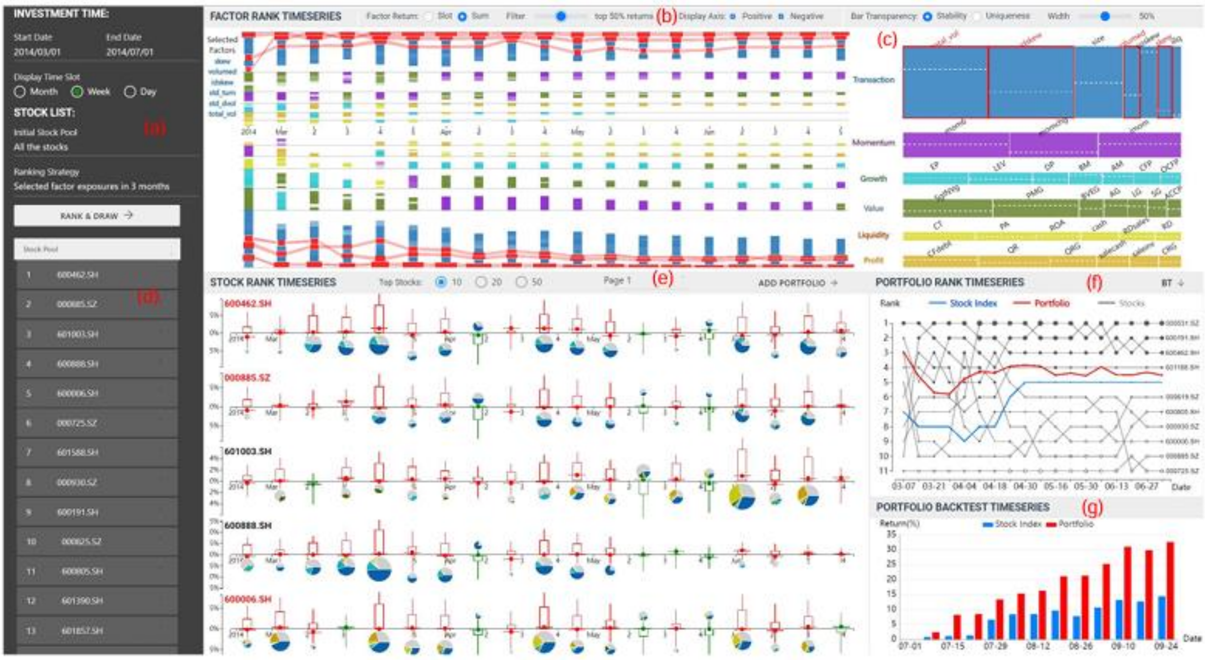


Figure 2.1.3.9: *A screenshot of “RankFIRST” from Guo et al. (2022). (a) Control panel. (b) Factor view. (c) Factor tree map. (d) List of ranked stock. (e) Stock timeseries view. (f) Portfolio rank timeseries. (g) portfolio back test timeseries.*

2.2 Previous systems

This section investigates established existing software tools that are specialized in stock visualization and comparative analysis, offering insights into their system architectures and functionalities. Section 2.2.1, 2.2.2 and 2.2.3 will delve into Yahoo Finance, Google Finance, and Morningstar, respectively. Information such as the URL of the system, the screenshot depicting its interface, the range of platforms supporting the application, and the length of the trial license are provided. This section also assesses the intended user base of each system and examines the advantages and limitations of them, offering a comprehensive understanding of the investment tools currently available to consumers and the visual strategies they employ to present complex financial data effectively.

2.2.1 Yahoo Finance

Yahoo Finance (https://finance.yahoo.com/) is a comprehensive free platform designed to offer users detailed insights into the stock market and individual stock performance. Available through web browsers and as mobile applications for both iOS and Android, Yahoo Finance combines real-time stock quotes, statistical overviews, and interactive charts to facilitate users with informed investment strategies. As shown in Figure 2.2.1.1, users can delve into historical data and key performance metrics, such as volume and price fluctuations, enhancing their decision-making process. The platform also enables users to customize charts with various indicators, compare multiple stocks, and stay updated with the latest financial news and market events.

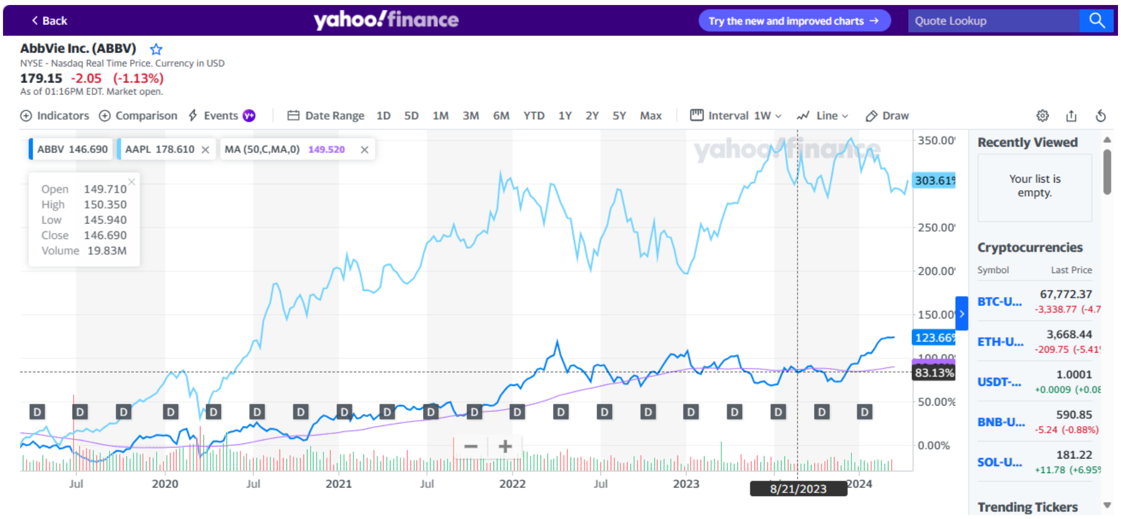


Figure 2.2.1.1: *A screenshot of the user interface of "Yahoo Finance" (Yahoo, 2024). The Moving Average and Comparison lines are demonstrated.*

With a focus on delivering a rich array of data points and a user-friendly experience, Yahoo Finance has become an indispensable resource for investors seeking to analyse the complexities of the financial markets. However, its real-time data is limited, which may be a consideration for users requiring up-to-the-minute market information.

2.2.2 Google Finance

Google Finance (https://www.google.com/finance) is a free, user-centric financial data service provided by Google, designed to offer instant access to market data and financial news. Easily navigable on devices that support Google search, including Android, iOS, and web platforms, this service facilitates a quick overview of financial markets, making it particularly appealing to novice traders and individuals seeking immediate market insights. For instance, searching for the stock "AAPL" on Google Finance, as shown in Figure 2.2.2.1, presents the latest stock price, after-hours movement, and an interactive chart reflecting the intraday price changes. Key metrics such as market cap and volume, alongside a comparison feature for related stocks are also displayed. Additionally, the real-time updates and alerts keep users informed about significant market movements. While Yahoo Finance offers a wide array of analytical tools for in-depth market analysis, Google Finance's simplicity leads to a more limited depth of data exploration. However, it excels in providing quick access to market data and clear, concise summaries, making it ideal for users who prioritize speed and simplicity in their stock assessments.

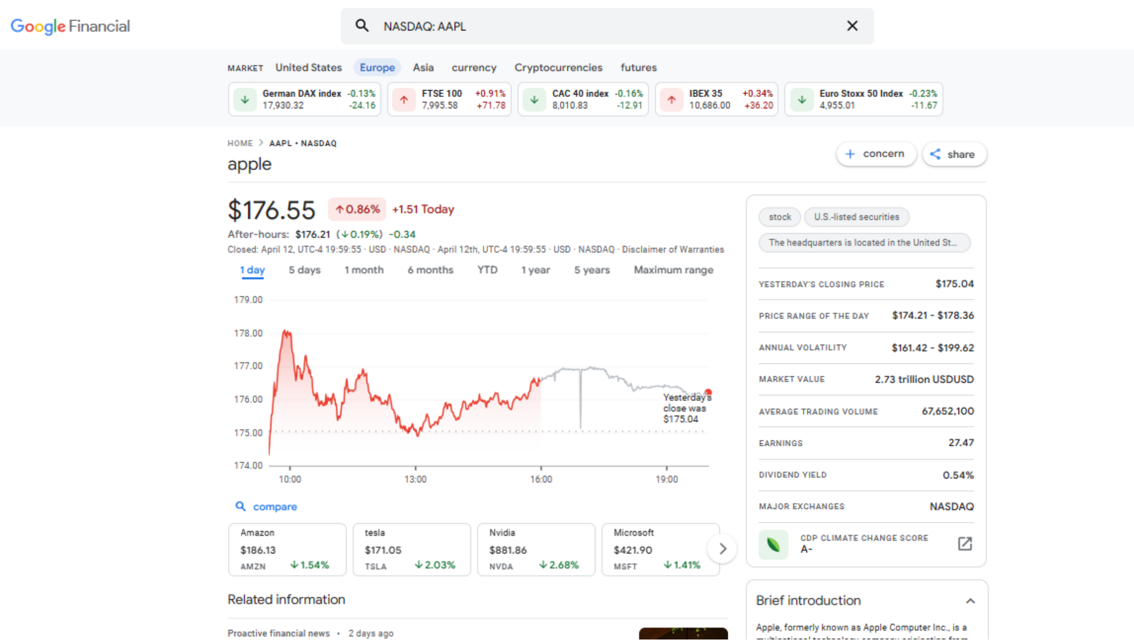


Figure 2.2.2.1: *A screenshot of the user interface of "Google Finance" (Google,2024).*

2.2.3 Morningstar

Morningstar (https://www.morningstar.com) stands out as a leading investment web platform, renowned for its comprehensive and research-intensive approach to evaluating stock performance and financial trends. Illustrated in Figure 2.2.3.1, the interface offers users a comprehensive overview with an emphasis on long-term investment strategies. It showcases extensive charts that trace historical price data, allowing investors to follow stock trajectories comprehensively over extended periods. Additionally, Morningstar offers a rich suite of financial metrics, including trading volume, P/E ratios, and market capitalization, all of which are enhanced by detailed company profiles. The platform's strength lies in its combination of quantitative data and qualitative insights, featuring expert commentary and ratings that guide users in making informed assessments of stock value and potential. Morningstar offers basic access for free, while its premium service requires a paid subscription, with a 14-day trial included. The robust blend of data and expert analysis ensures that Morningstar remains as an essential resource for investors dedicated to conducting deep market analysis and thorough investment research.

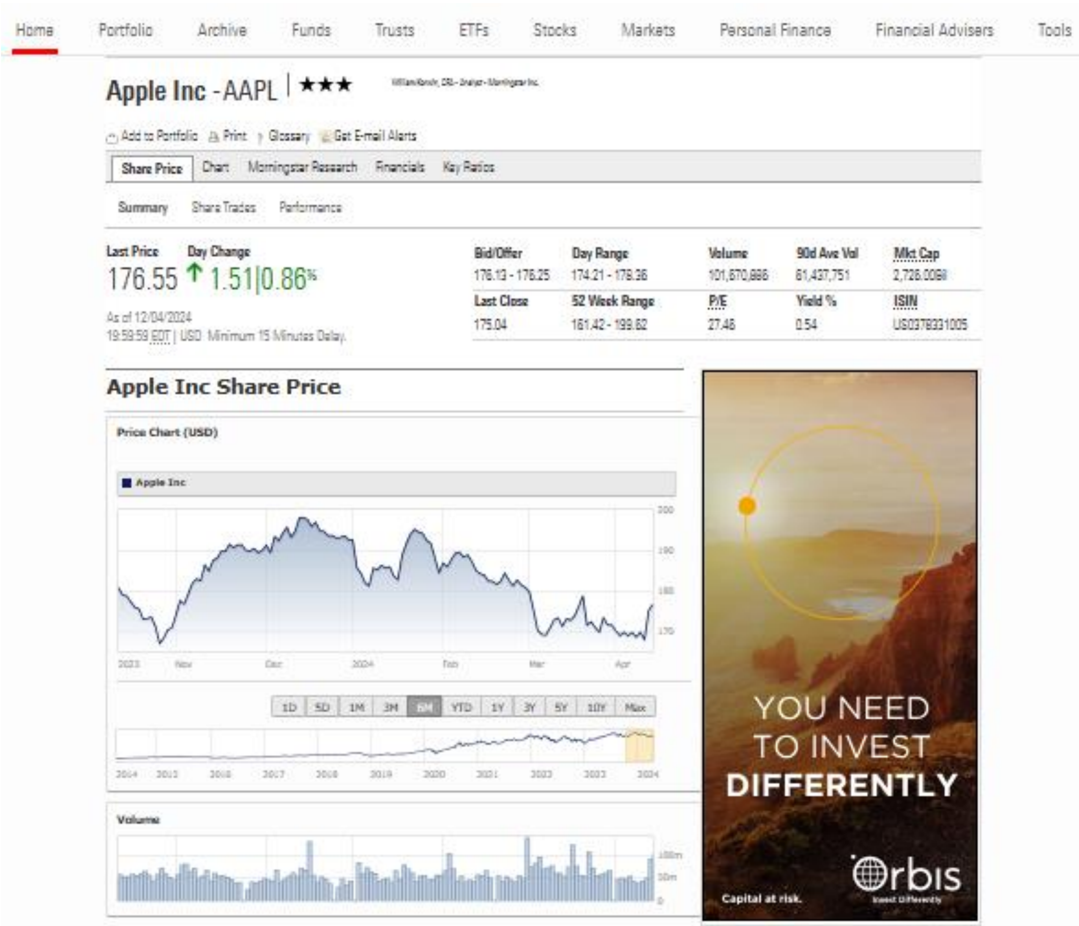


Figure 2.2.3.1: *A screenshot of the user interface of "Morningstar" (Morningstar, 2024).*

2.3 Data Characteristics

This project utilizes three datasets related to stock investments. The core dataset used is the "Investment Transaction Dataset"; the other two datasets are the "Company-Centred Metadata" and "Stock API Data". The introduction of the very first open Investment Transaction Dataset from a specific investor offers a unique opportunity for researchers to explore detailed, customer-focused financial transactions. Unlike generic financial datasets, this dataset provides a window into the individual patterns and behaviours that influence financial decision-making processes at the consumer level, which presents a significant opportunity to align research and development with the exploration, analysis, and enhancement of financial visualization tools.

Section 2.3.1 provides the origins and context of the datasets employed in this project. Section 2.3.2 provides a technical description of the datasets, covering aspects such as the file size, number of samples, data range, characteristics, and formats. This provides a comprehensive overview of how the data is structured and stored.

2.3.1 Data Sources

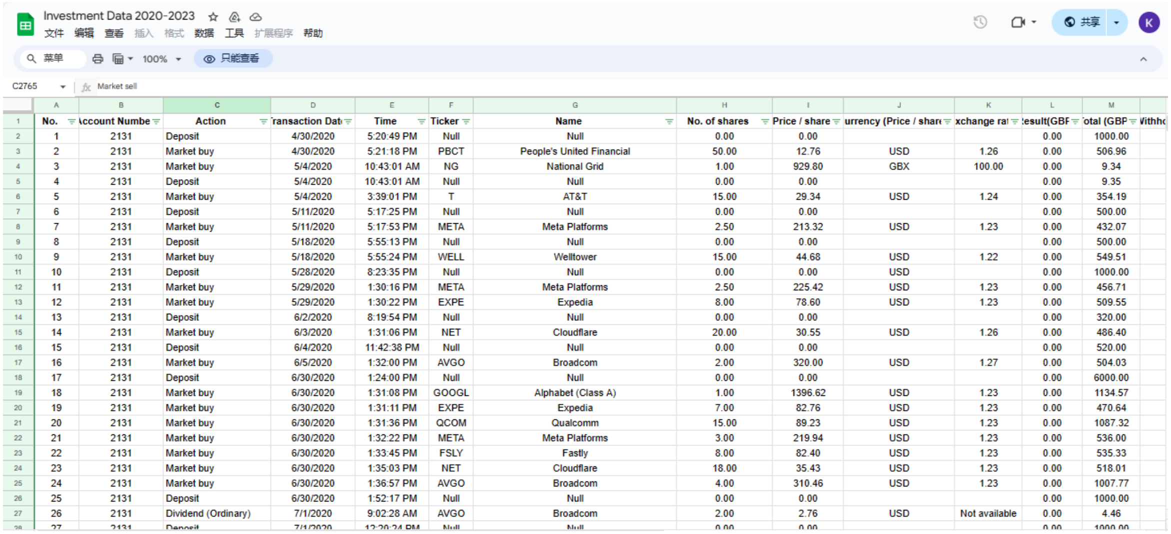
The primary dataset utilized for this project is the "Investment Transaction Dataset", an innovative and anonymized dataset provided by a stock investor, covering the period from April 30, 2020, to December 12, 2023. This core dataset is complemented by the "Company-Centred Metadata" dataset, which is calculated and derived from the original investment transaction data. Both datasets can be accessed via the following link: https://docs.google.com/spreadsheets/d/1BLUATC6DDbnfDTqoXzAXDSOh12Vc8Nw8aEUqjEJWWVU/edit?usp=sharing. The project is further enriched by incorporating "Stock API Data" datasets from "Google Finance", which includes detailed metrics such as Open, High, Low, Close, Volume, Dividends, and Stock Splits for each stock. These datasets span from April 30, 2020, to the present day. Following in the footsteps of MoneyVis project (Firat et al., 2023), which utilized the first money transaction dataset from a specific customer, this dataset facilitates a deep dive into individual investment behaviours. Utilizing these datasets, this project aims at providing a new financial visualization tool, thereby revealing nuances in financial decision-making processes of specific customers that are often obscured in more aggregated data.

2.3.2 Data Descriptions

● Investment Transaction Dataset

This dataset covers a comprehensive range of trading activities from April 30, 2020, to December 12, 2023. Stored in CSV format and with a size of 310KB, it includes 2,772 entries, encapsulating a wealth of transaction data across two different types of accounts. The data is classified as abstract data and time-dependent, which is crucial for analysing trends over time. The description of the 23 attributes of the dataset are as follows:

- **No.** ***(Int)***: A simple numeric identifier for each record.
- **Account number** ***(Int)***: The ISA is tax-free and the INV account may be taxed.
- **Action** ***(String)***: Describes the transaction type, including Market Buy, Market Sell, various Dividend types, Interest on cash, Deposit, and Withdrawal.
- **Transaction Date** ***(Date)***: The date on which the transaction occurred.
- **Time** ***(Time)***: The exact time of the transaction.
- **Ticker** ***(String)***: Stock ticker symbol used to identify a specific stock.
- **Name** ***(String)***: Name of the stock, usually the company name.
- **No. of shares** ***(Float)***: - The number of shares involved in the transaction.
- **Price / share** ***(Float)***: Transaction price per share.
- **Currency (Price / share)** ***(String)***: Currency used for the transaction price.
- **Exchange rate** ***(Float)***: Rate used to convert the transaction currency to GBP.
- **Result (GBP)** ***(Float)***: Profit or loss from selling stocks, in British pounds.
- **Total (GBP)** ***(Float)***: Total transaction amount in British pounds.
- **Withholding tax** ***(Float)***: Tax withheld at the time of dividend distribution.
- **Currency (Withholding tax)** ***(String)***: Currency of the withholding tax.
- **Charge amount (GBP)** ***(Float)***: Additional charges with the transaction.
- **Deposit fee (GBP)** ***(Float)***: Fee charged for deposits.
- **Transaction fee (GBP)** ***(Float)***: Fee that may apply to 'Market Sell' actions.
- **Finra fee (GBP)** ***(Float)***: Regulatory fees charged by financial authorities.
- **Currency conversion fee (GBP)** ***(Float)***: Fees for converting currencies.
- **Stamp duty reserve tax (GBP)** ***(Float)***: A government tax on the transaction.
- **ID** ***(Int)***: Unique identification number of the transaction.
- **Notes** ***(String)***: Any additional information about the transaction.



No.	Account Numbe	Action	Transaction Date	Time	Ticker	Name	No. of shares	Price / share	urrency (Price / share	xchange rat	esult(GBF	otal (GBP	Withho
1	2131	Deposit	4/30/2020	5:20:49 PM	Null	Null	0.00	0.00			0.00	1000.00	
2	2131	Market buy	4/30/2020	5:21:18 PM	PBCT	People's United Financial	50.00	12.76	USD	1.26	0.00	506.96	
3	2131	Market buy	5/4/2020	10:43:01 AM	NG	National Grid	1.00	929.80	GBX	100.00	0.00	9.34	
4	2131	Deposit	5/4/2020	10:43:01 AM	Null	Null	0.00	0.00			0.00	9.35	
5	2131	Market buy	5/4/2020	3:39:01 PM	T	AT&T	15.00	29.34	USD	1.24	0.00	354.19	
6	2131	Deposit	5/11/2020	5:17:25 PM	Null	Null	0.00	0.00			0.00	500.00	
7	2131	Market buy	5/11/2020	5:17:53 PM	META	Meta Platforms	2.50	213.32	USD	1.23	0.00	432.07	
8	2131	Deposit	5/18/2020	5:55:13 PM	Null	Null	0.00	0.00			0.00	500.00	
9	2131	Market buy	5/18/2020	5:55:24 PM	WELL	Welltower	15.00	44.68	USD	1.22	0.00	549.51	
10	2131	Deposit	5/28/2020	8:23:35 PM	Null	Null	0.00	0.00	USD		0.00	1000.00	
11	2131	Market buy	5/29/2020	1:30:16 PM	META	Meta Platforms	2.50	225.42	USD	1.23	0.00	456.71	
12	2131	Market buy	5/29/2020	1:30:22 PM	EXPE	Expedia	8.00	78.60	USD	1.23	0.00	509.55	
13	2131	Deposit	6/2/2020	8:19:54 PM	Null	Null	0.00	0.00			0.00	320.00	
14	2131	Market buy	6/3/2020	1:31:06 PM	NET	Cloudflare	20.00	30.55	USD	1.26	0.00	486.40	
15	2131	Deposit	6/4/2020	11:42:38 PM	Null	Null	0.00	0.00			0.00	520.00	
16	2131	Market buy	6/5/2020	1:32:00 PM	AVGO	Broadcom	2.00	320.00	USD	1.27	0.00	504.03	
17	2131	Deposit	6/30/2020	1:24:00 PM	Null	Null	0.00	0.00			0.00	6000.00	
18	2131	Market buy	6/30/2020	1:31:08 PM	GOOGL	Alphabet (Class A)	1.00	1396.62	USD	1.23	0.00	1134.57	
19	2131	Market buy	6/30/2020	1:31:11 PM	EXPE	Expedia	7.00	82.76	USD	1.23	0.00	470.64	
20	2131	Market buy	6/30/2020	1:31:36 PM	QCOM	Qualcomm	15.00	89.23	USD	1.23	0.00	1087.32	
21	2131	Market buy	6/30/2020	1:32:22 PM	META	Meta Platforms	3.00	219.94	USD	1.23	0.00	536.00	
22	2131	Market buy	6/30/2020	1:33:45 PM	FSLY	Fastly	8.00	82.40	USD	1.23	0.00	535.33	
23	2131	Market buy	6/30/2020	1:35:03 PM	NET	Cloudflare	18.00	35.43	USD	1.23	0.00	518.01	
24	2131	Market buy	6/30/2020	1:36:57 PM	AVGO	Broadcom	4.00	310.46	USD	1.23	0.00	1007.77	
25	2131	Deposit	6/30/2020	1:52:17 PM	Null	Null	0.00	0.00			0.00	1000.00	
26	2131	Dividend (Ordinary)	7/1/2020	9:02:28 AM	AVGO	Broadcom	2.00	2.76	USD	Not available	0.00	4.46	

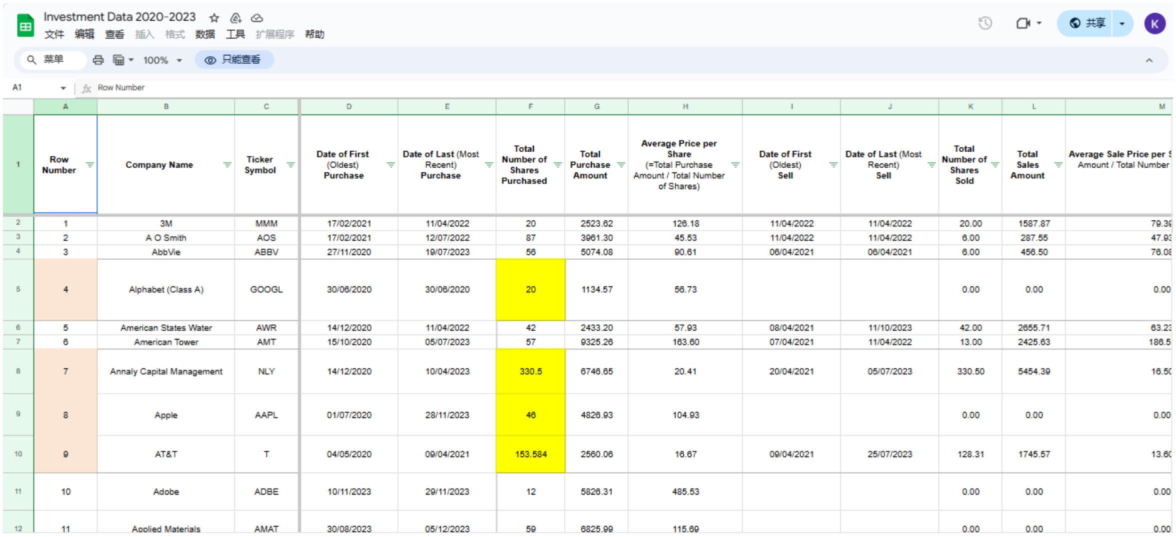
Figure 2.3.2.1: *A screenshot of the "Investment Transaction Dataset".*

• Company-Centred Metadata

This dataset is tailored to provide a detailed overview of company-specific trading activities from April 30, 2020, to December 12, 2023. Stored in CSV format and with a size of 20KB, it contains structured data related to 83 stocks managed across different types of accounts, encapsulating a detailed view of the buying and selling actions per company. The dataset is structured to provide insight into both realized and unrealized gains and losses, alongside dividend distributions and corporate actions that impact shareholding patterns. The description of the 19 attributes of the dataset are as follows:

- **Row Number *(Int)***: A sequential identifier for each transaction entry.
- **Company Name *(String)*:** The name of the company involved in the transaction.
- **Ticker Symbol *(String)***: Stock ticker symbol used to identify a specific stock.
- **Date of First Purchase *(Date)***: The date when shares were first purchased.
- **Date of Last Purchase *(Date)***: The date of the most recent share purchase.
- **Total Number of Shares Purchased *(Int)***: The sum of all shares purchased.
- **Total Purchase Amount *(Float)***: The value of all shares purchased.
- **Average Price per Share (Purchase) *(Float)***: Computed as the Total Purchase Amount divided by the Total Number of Shares Purchased.
- **Date of First Sale *(Date)***: The earliest recorded sale of shares.
- **Date of Last Sale *(Date)***: The date of the most recent sale of shares.
- **Total Number of Shares Sold *(Int)***: The total quantity of shares sold.

- **Total Sales Amount *(Float)***: The aggregate monetary value from selling shares.
- **Average Sale Price per Share *(Float)***: Calculated as the Total Sales Amount divided by the Total Number of Shares Sold.
- **Net Total Number of Shares *(Int)***: The difference between the Total Number of Shares Purchased and the Total Number of Shares Sold.
- **Current Share Price *(Float)***: The latest trading price of the shares.
- **Total Dividends as of 12 Dec 2023 *(Float)***: The total dividends.
- **Realized Capital Gain & Loss *(Float)***: The profit or loss realized from selling.
- **Unrealized Capital Gain & Loss *(Float)***: The profit or loss not yet realized.
- **Notes *(String)***: Any additional information to the transaction data.



Row Number	Company Name	Ticker Symbol	Date of First (Oldest) Purchase	Date of Last (Most Recent) Purchase	Total Number of Shares Purchased	Total Purchase Amount	Average Price per Share (=Total Purchase Amount / Total Number of Shares)	Date of First (Oldest) Sell	Date of Last (Most Recent) Sell	Total Number of Shares Sold	Total Sales Amount	Average Sale Price per Amount / Total Number
1	3M	MMM	17/02/2021	11/04/2022	20	2523.62	126.18	11/04/2022	11/04/2022	20.00	1587.87	79.3
2	A O Smith	AOS	17/02/2021	12/07/2022	87	3961.30	45.53	11/04/2022	11/04/2022	6.00	287.55	47.9
3	AbbVie	ABBV	27/11/2020	19/07/2023	56	5074.08	90.61	06/04/2021	06/04/2021	6.00	456.50	76.0
4	Alphabet (Class A)	GOOGL	30/06/2020	30/06/2020	20	1134.57	56.73			0.00	0.00	0.00
5	American States Water	AWR	14/12/2020	11/04/2022	42	2433.20	57.93	08/04/2021	11/10/2023	42.00	2655.71	63.2
6	American Tower	AMT	15/10/2020	05/07/2023	57	9325.26	163.60	07/04/2021	11/04/2022	13.00	2425.63	186.5
7	Annaly Capital Management	NLY	14/12/2020	10/04/2023	330.5	6746.65	20.41	20/04/2021	05/07/2023	330.50	5454.39	16.5
8	Apple	AAPL	01/07/2020	28/11/2023	46	4826.93	104.93			0.00	0.00	0.00
9	AT&T	T	04/05/2020	09/04/2021	153.584	2560.06	16.67	09/04/2021	25/07/2023	128.31	1745.57	13.6
10	Adobe	ADBE	10/11/2023	29/11/2023	12	5826.31	485.53			0.00	0.00	0.00
11	Applied Materials	AMAT	30/08/2023	05/12/2023	59	6825.99	115.69			0.00	0.00	0.00

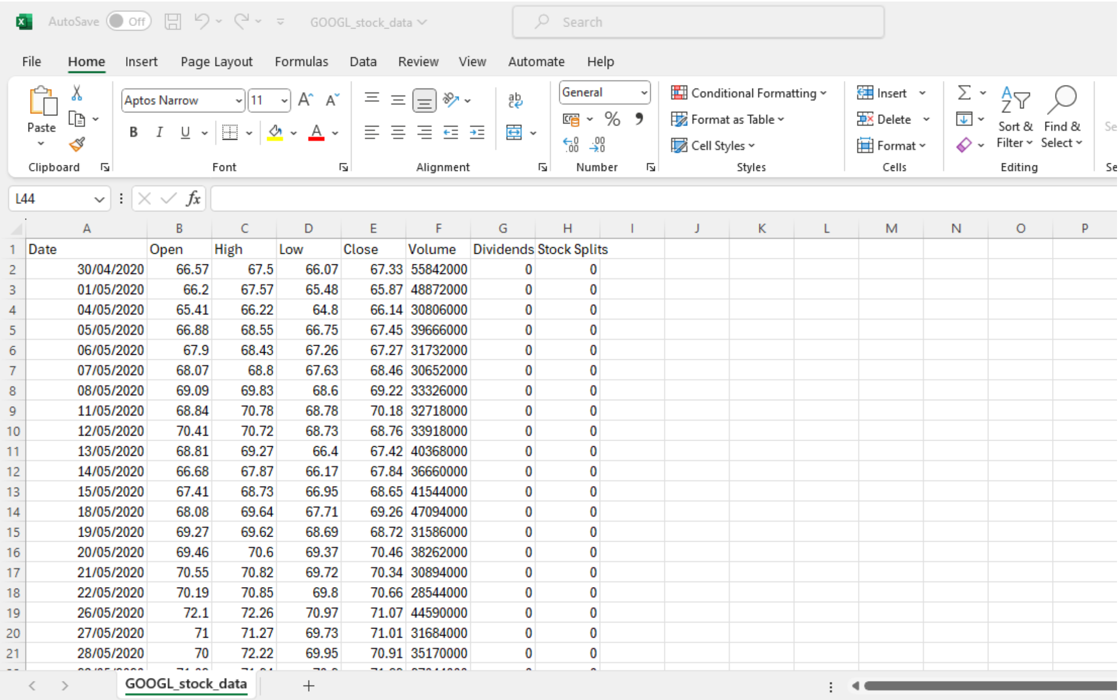
Figure 2.3.2.2: *A screenshot of the "Company-Centred Metadata".*

● Stock API Datasets

These stock datasets, which are generated for specific stocks appeared in the Investment Transaction Dataset from "Google Finance", provide an extensive history of daily stock market activity for individual stocks. Stored as individual files per stock, each dataset spans from April 30, 2020, to the present day, with each file approximately 50KB in size. These datasets play an important role in combining the user investment transaction data and the real-time market data, making them suitable for analyses on individual stock performances over time. The description of the 8 attributes of the dataset are as follows:

- **Date *(Date)***: The trading day for the recorded data.
- **Open *(Float)***: Opening price of the stock for the trading day.

- **High *(Float)***: Highest price of the stock during the trading day.
- **Low *(Float)***: Lowest price of the stock during the trading day.
- **Close *(Float)***: Closing price of the stock at the end of the trading day.
- **Volume *(Integer)***: Total volume of stock shares traded during the day.
- **Dividends *(Float)***: Dividends paid out on that trading day, if any.
- **Stock Splits *(Integer)***: Indicates any stock splits occurring on the trading day, represented by the split ratio (e.g., 0 for no split, 2 for a 2:1 split).



Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
30/04/2020	66.57	67.5	66.07	67.33	55842000	0	0
01/05/2020	66.2	67.57	65.48	65.87	48872000	0	0
04/05/2020	65.41	66.22	64.8	66.14	30806000	0	0
05/05/2020	66.88	68.55	66.75	67.45	39666000	0	0
06/05/2020	67.9	68.43	67.26	67.27	31732000	0	0
07/05/2020	68.07	68.8	67.63	68.46	30652000	0	0
08/05/2020	69.09	69.83	68.6	69.22	33326000	0	0
11/05/2020	68.84	70.78	68.78	70.18	32718000	0	0
12/05/2020	70.41	70.72	68.73	68.76	33918000	0	0
13/05/2020	68.81	69.27	66.4	67.42	40368000	0	0
14/05/2020	66.68	67.87	66.17	67.84	36660000	0	0
15/05/2020	67.41	68.73	66.95	68.65	41544000	0	0
18/05/2020	68.08	69.64	67.71	69.26	47094000	0	0
19/05/2020	69.27	69.62	68.69	68.72	31586000	0	0
20/05/2020	69.46	70.6	69.37	70.46	38262000	0	0
21/05/2020	70.55	70.82	69.72	70.34	30894000	0	0
22/05/2020	70.19	70.85	69.8	70.66	28544000	0	0
26/05/2020	72.1	72.26	70.97	71.07	44590000	0	0
27/05/2020	71	71.27	69.73	71.01	31684000	0	0
28/05/2020	70	72.22	69.95	70.91	35170000	0	0

Figure 2.3.2.3: *A screenshot of the "GOOGL Dataset", as part of "Stock API Datasets".*

Data is the fundamental and crucial element in all visualizations, and the combination of the three types of datasets offers a comprehensive insight into investor behaviours, market trends, and stock performances over time. In order to manage the valuable data effectively, the datasets are stored in Google Sheets, which provides a secure and reliable cloud-based platform for data management. This method of storage significantly reduces the risk of data loss and enhances accessibility, allowing researchers to access and share data seamlessly from any location. Utilizing Google Sheets ensures that all data remains up-to-date and available for ongoing and collaborative research, facilitating efficient data analysis and decision-making processes.

3 Project Specification

This Project Specification includes two sub-sections. Section 3.1 describes the feature specification and section 3.2 introduces technology choices.

3.1 Feature Specification

As suggested in Bob's Project Guidelines (Laramee, 2011), this section outlines the features of the software, including must-have features in section 3.1.1 and the optional features in section 3.1.2. The structured lists include both essential functionalities for the system’s basic operations and additional enhancements that could improve its overall performance, serving as a roadmap to guide the project to successful completion.

3.1.1 Must-have Features

The key features described in this section are intended to establish the fundamental functionalities of the system, which are vital for its effective operation and overall success. Guided by Shneiderman's Visualization Mantra, the approach prioritizes an "overview first, filtering and selection, and details-on-demand" thematic structure (Shneiderman, 1996). This methodology ensures that users are first presented with a comprehensive view of the data, followed by tools for focused filtering based on specific criteria, and ultimately, the ability to delve into detailed information on demand. This strategic framework ensures that the visualization system is both effective and user-friendly, addressing key user tasks and interactions efficiently.

1. **An interactive system** that enables dynamic visualization of financial data, through implementing a **dashboard application** with multiple pages and views.
2. **Seven distinct views** within the application, including Home, Buy/Sell, Dividend, Multiple, Single, Risk Factors, and Gain/Loss, switch through **Buttons**.
3. **A parallel coordinates plot** in the Risk Factors view to illustrate the performance and patterns across different companies based on Company-Centred Metadata.
4. **A multiple line chart** in the Multiple view, showing trends of all stocks included in the Investment Transaction Dataset.
5. **Stock history data** fetched from finance APIs to integrate real-time data analysis.

6. **Buy and sell points** on the multiple line chart for all stocks, the prices are according to the transaction record in the Investment Transaction Dataset.
7. **An input box** for user to enter **moving average** values and adjust the view of lines on the multiple line chart, with the default setting of 10 days.
8. **Two ticker boxes** for users to select whether to display trends following **the last sell or the last buy** on the multiple line chart.
9. **Colour mappings** to differentiate tickers, based on the average price of each stock over the period covered in the investment transaction dataset.
10. **Grouped legends** for the line and points of each distinct stock on multiple line chart, ensuring they respond consistently when users interact with the system**.**
11. **A stacked bar chart** to compare **dividends** of all companies in the Dividend view.
12. **A stacked bar chart** to track **dividend** changes over time in the Dividend view.
13. **Single line charts** for all individual stocks in the Single view, including a selection box for users to select specific stock tickers to view.
14. **An input box** for user to enter **moving average** values and add moving average lines in the single stock view, multiple lines of different moving averages are allowed.
15. **A stacked bar chart** to display monthly **buy and sell amounts** for the two distinct accounts in the Buy/Sell view.
16. **A stacked bar chart** to represent the **capital gain/loss and dividend gain** for each stock in the Gain/Loss view.
17. **User filter and selection options** should be implemented in all charts to ensure the interactivity, allowing users to select specific stocks or time periods for viewing and comparing between selected stocks over selected periods. Employ techniques such as "Focus and Context" to improve data navigation.

3.1.2 Optional Features

Optional features are enhancements that build upon the core functionalities of the system, as outlined in Bob's Project Guidelines (Laramee, 2011). These additions introduce advanced capabilities to the visualization tools and application architecture, improving efficiency and elevating the application to exceptional quality. By implementing these features, the system gains comprehensive enhancements that optimize performance and enrich the user experience.

1. **Different styles of charts** for the single stock view, including line, candle, ohlc, and area charts according to user selection, with the line chart set as the default.
2. **Time interval selection buttons** for all time-based charts with period selection option and use lines to separate the years for clearer visualization.
3. **Toggle view button** for users to toggle between two views on the dividend changes over time chart: one displaying the company name with different types of dividends, which is more detailed but complicated, and another showing only the company name, which provides less information but is clearer.
4. **Brush functionalities** for every axis of the parallel coordinate chart, allowing users to focus on stocks within the selected range of data. The parallel coordinate view should render selected companies with colour emphasis as the focal point while representing unselected entities transparently.
5. **Link brushing** for the multiple line chart and the parallel coordinate chart. The multiple line chart dynamically focuses on the selected stocks in parallel coordinate chart, turning the rest of the lines and their buy and sell points transparent.
6. **Display events** about the stock if an event of the company occurs, this happens when hovering over the data point on that specific day in the multiple line chart.
7. **Dynamic adjusts** to accommodate varying screen resolutions for all components.
8. **Legend interaction** for all charts, users could click once on the legend of a stock to remove it from the chart or double-click to isolate it for focused view and analysis.
9. **Detailed information** of the data point should display when the cursor hovers over each data point on any chart, implementing "focus and context." For instance, in the multiple line chart, hovering over a specific data point will reveal the name of ticker, the specific date and closing price of that data point on a specific line.
10. **Zooming and box selection functionalities** for all charts, enabling users to view every part of the charts and investigate the details freely.
11. **An informative tooltip** including the above functionalities should emerge when the cursor is positioned over any graphical element.

3.2 Technology Choices

For all technology choices, potential options will be evaluated based on a comprehensive set of criteria. Ideal technologies are open-source, offer cross-platform compatibility, and are supported by an active community or forum for developer support and issue resolution, as suggested in Bob's Project Guidelines (Laramee, 2011). This strategic approach ensures that the development environment supports extensive problem-solving and fosters innovation throughout the project's lifecycle, thereby enhancing both the efficiency of the development process and the quality of the final product.

This section details the tools and technologies selected for the implementation of this project. Section 3.2.1 delves into the choice of programming language, which is pivotal as it determines the ease of development, maintenance, and scalability of the application. Section 3.2.2 introduces the GUI library and other supplementary libraries selected for this project, which are essential for the overall user experience and the enhancement of specific functionalities. Section 3.2.3 discusses other technologies utilised in this project, such as GitHub, Finance APIs, PIP and Visual Studio Code.

3.2.1 Programming Languages

In the landscape of software development, particularly for projects demanding intricate data visualization capabilities, the choice of a programming language is crucial. It holds a profound influence on various aspects: the development process, system performance, future scalability, and the application's accessibility to users. Given the project's goal on developing an interactive customer-centric visualization tool for stock investments and based on the discussions with supervisor Professor Laramee, **Python** was chosen as the primary programming language, guided by several key considerations.

Apart from Python, several programming languages are also considered as potential candidates for the project. The first one of them is **Java**. With libraries such as Java Swing, Java is powerful for creating rich graphical user interface. However, it falls short in the web application domain, which may not be ideal for a project aiming for the ease of deployment. Java's approach is more suited to desktop applications rather than interactive, web-based applications.

The second candidate programming language is **R**, which is known for its strong statistical capabilities. However, its focus is more on data analysis rather than building interactive applications with its static nature. While libraries like Shiny and Plotly extend R's capabilities towards interactivity, they do not seamlessly integrate with web technologies as effectively as some modern web frameworks.

The other two languages considered are **JavaScript** and **TypeScript**. These languages are integral to web development and can be used to build highly interactive applications. They are well-suited for client-side scripting and could be a viable option if the project were solely focused on client-side operations. However, Python offers a more robust ecosystem for data manipulation and server-side logic, which are critical for this project.

Python is equipped with extensive libraries and frameworks conducive to data manipulation. Python's ecosystem includes powerful libraries such as Pandas for data manipulation, NumPy for numerical data, and Matplotlib for plotting graphs. More importantly, frameworks like Dash by Plotly provide the exact functionalities needed for building interactive, web-based dashboards, which delivers the precise functionality essential for the project's interactive nature.

Python's proficiency in data visualization is exemplified by “PyViz” (Figure 3.2.1.1), a comprehensive collection of visualization tools that enable developers to construct powerful plots and dashboards efficiently. With these visualization-focused tools, it stands out as a particularly resourceful asset for this project, providing sophisticated and user-centric visualizations crucial for analysing stock investments.

Being open-source, Python allows for a cost-effective development process with extensive libraries and tools at no additional cost. Community engagement and support within the Python ecosystem are also unparalleled. A robust community of developers offers a wealth of resources, which is indispensable for troubleshooting and expediting development. Additionally, Python's inherent cross-platform compatibility ensures the application can function across different operating systems. This means applications built with it can run on Windows, macOS, and Linux without significant modifications, thus amplifying its accessibility and reach.

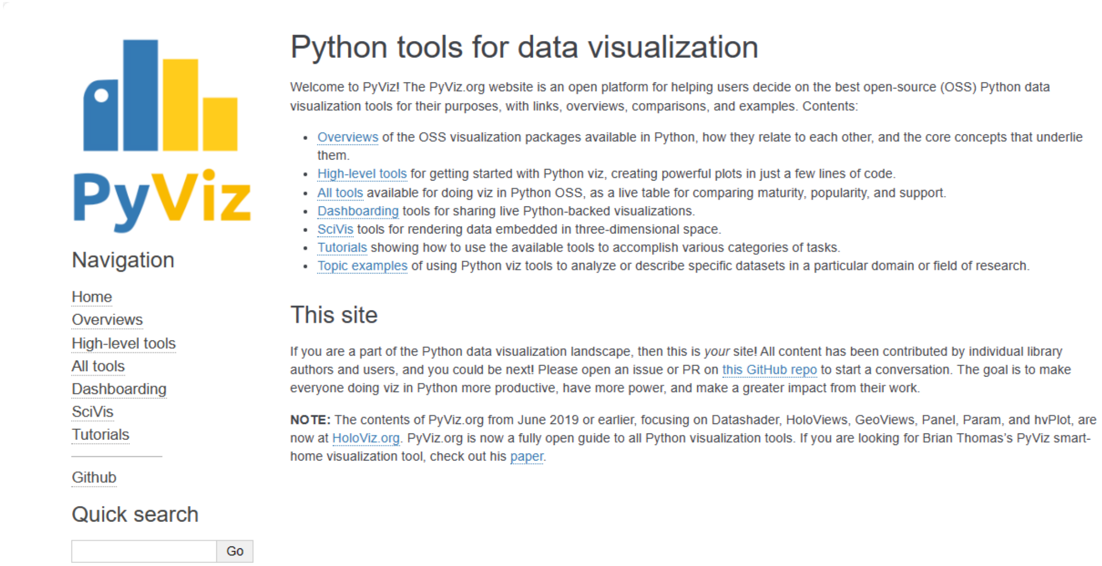


Figure 3.2.1.1: *A screenshot of the "PyViz" tools provided by python, (PyViz, 2024).*

The advantages of Python extend to its ease of integration, facilitating smooth interoperability with various data formats and sources. This seamless data handling is paramount for the financial datasets integral to the project. Python's clear and concise syntax enhances the development speed and simplifies the coding process, providing a swift turnaround crucial in the fast-evolving financial sector.

Given the need for robust data handling capabilities, extensive library support, and ease of building interactive web applications for this project, **Python** stands out as the most suitable programming language. The integration of Python with frameworks like Dash and visualization tools in PyViz provide the project with a sophisticated, user-centric visualization tool, poised to make a significant impact in the realm of stock investments.

3.2.2 Libraries

To build a dynamic and user-friendly visualization tool for stock investments, the libraries chosen for this project were carefully considered to ensure they offer robust functionality and can seamlessly integrate with the Python ecosystem. According to the discussions during the meetings and some blogs which compare data dashboarding tools and frameworks, such as "**Streamlit** vs. **Dash** vs. **Shiny** vs. **Voila** vs. **Flask** vs. **Jupyter**" https://www.datarevenue.com/en-blog/data-dashboarding-streamlit-vs-dash-vs-shiny-vs-voila, "Dash" by "Plotly" was selected as the primary library for this project.

Dash by Plotly: Serving as the cornerstone and the **GUI library** of this interactive tool, Dash by Plotly (Figure 3.2.2.1) has been instrumental in the development of the project's dashboard. It is a Python web application framework that enables the creation of interactive, web-based data dashboards without the need for JavaScript. Dash is inherently modular and responsive, making it an excellent choice for the project's GUI layer. Its compatibility with core Python data libraries allows for the creation of rich visualization components that are both performant and easy to use.

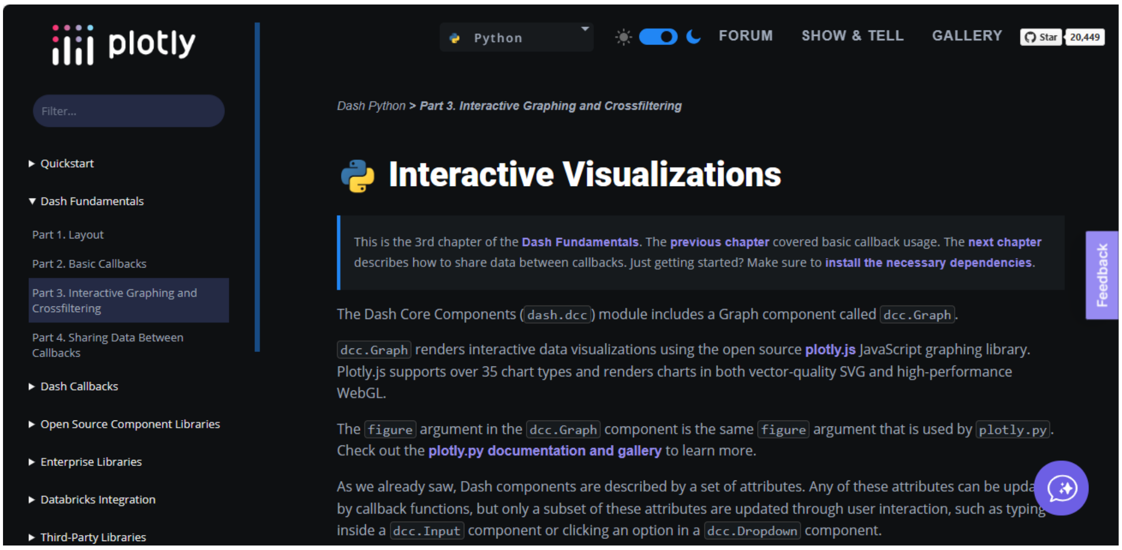


Figure 3.2.2.1: *A screenshot of the "Plotly" library provided by python, (Plotly, 2024).*

Pandas: Handling financial data requires powerful data manipulation capabilities, which is where Pandas comes in. It is an open-source library providing high-performance, easy-to-use data structures, and data analysis tools for Python. The library has been utilized extensively in the project to manipulate financial datasets, perform computations, and serve the processed data to the front end for visualization.

NumPy: Renowned for its numerical computation abilities, NumPy works hand-in-hand with Pandas to facilitate complex mathematical operations on large datasets. Its array-oriented computing provides an efficient means to store and manipulate large arrays of data, which is critical in performing real-time financial analysis.

Matplotlib: This library has been employed for its comprehensive set of plotting functions. Matplotlib integrates seamlessly with Pandas and NumPy, providing a solid foundation for creating static, animated, and interactive visualizations.

Each of these libraries plays a specific role in the application, working together to process data, generate visualizations, and present them through a user-friendly interface. "Dash" acts as the conductor, orchestrating the flow between data processing and visualization rendering, all within an interactive GUI that is accessible via web browsers.

3.2.3 Other Technologies

● GitHub

GitHub (Figure 3.2.3.1) utilizes Git, the most widely used modern version control system in the world, providing an efficient way to version control, revert to previous states, and manage code branches for feature development and testing. By leveraging GitHub as a part of the project's development process, the project benefits from a robust ecosystem for managing the software development process, enabling a more structured, collaborative, and quality-oriented approach. This choice reflects a commitment to best practices in software development and a recognition of the importance of community and open-source values in the modern development landscape. The codes for this project are available on GitHub via the link: https://github.com/KaiyiCHENkk/FinanceVis.

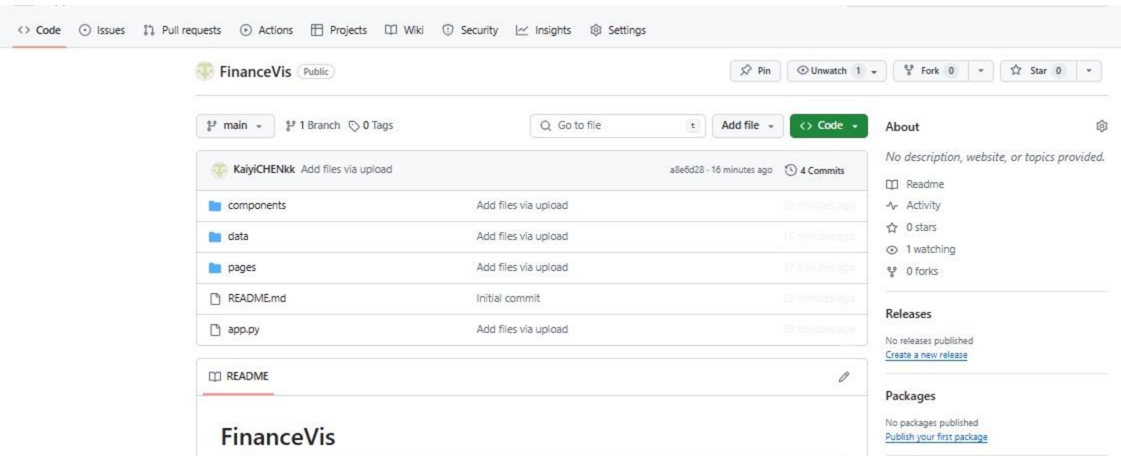


Figure 3.2.3.1: *A screenshot of this project on GitHub, (GitHub, 2024).*

● Finance APIs

Google Finance has been chosen for this project due to its robustness and reliability. It provides a wide range of financial data, including real-time market quotes, international exchanges, up-to-date financial news, and analytics. It's known for its precise data matching, which aligns well with the values in the investment transaction dataset. The

API offers a substantial history of data, critical for deep financial analysis, and is free of charge, which aligns with the project's aim to keep costs low while maintaining high data integrity. Its reputation for data accuracy and comprehensiveness sets it apart from other alternatives, which have occasionally faced issues in these areas.

Some other APIs had also been considered as candidates. For example, **Yahoo Finance** API is a popular choice among developers for accessing financial information, including stock prices, financial statements, and market trends. However, issues regarding the precision of its data have been noted, particularly as its data may not always align with actual stock values. This discrepancy has been highlighted in various forums, including Stack Overflow, where researchers and developers have shared concerns about data consistency. Although Yahoo Finance API offers a free tier, this potential lack of data accuracy was a significant factor in it not being the primary choice for this project.

Alpha Vantage API offers free APIs for historical and real-time data on stocks, forex, and cryptocurrencies. It's lauded for providing a simple, clean, and well-documented API interface. However, the free tier comes with limitations on API call frequency, which may pose a challenge for applications requiring high-frequency data updates. The historical data availability is extensive, but the granularity of data may vary according to various forums, necessitating validation against transaction datasets for precision.

IEX Cloud API provides financial data across stocks, currencies, cryptocurrencies, and other financial instruments. While it does offer a free starter tier, access to more comprehensive datasets and higher API request volumes are gated behind paid tiers. The API boasts a significant historical data range, and data accuracy is generally reliable. However, for financial analysis with a broad scope, the free tier might prove insufficient, prompting consideration of budget allowances for the necessary subscription plans.

Polygon.io offers extensive data coverage with high precision, which is pivotal for accurate financial analysis and modelling. The API is well-regarded for its data completeness and accuracy, matching closely with actual transaction datasets. While Polygon.io does provide a robust free tier, more advanced features, longer history data ranges and higher request volumes are available through its paid plans, making it a scalable option for this project.

● PIP

PIP, the package management system for Python, plays an integral role in managing the software libraries and dependencies. As Python's official package installer, PIP provides seamless access to the Python Package Index, which houses a comprehensive collection of software libraries and modules utilized throughout the project. This functionality is critical for integrating various Python packages such as Pandas for data manipulation, Plotly for creating interactive visualizations, and Dash for web application development, ensuring these components remain up-to-date and compatible. By simplifying the process of installing and updating libraries, PIP enhances the project's development efficiency, enabling a streamlined workflow that maintains focus on delivering high-quality financial visualization tools tailored to user needs.

● Visual Studio Code

VS Code (Figure 3.2.3.2) was selected as the primary development environment due to its robust support for code compilation and rich feature set, including build automation, testing, and deployment. This platform also boasts integrated Git support and a built-in console, which streamline version control and command-line operations, ensuring seamless integration with other technologies used in the project. The compatibility of VS Code with package management tools like PIP enhances the development workflow, making it an invaluable tool for coding, testing, and deploying the sophisticated visualizations that are central to this financial analysis tool.

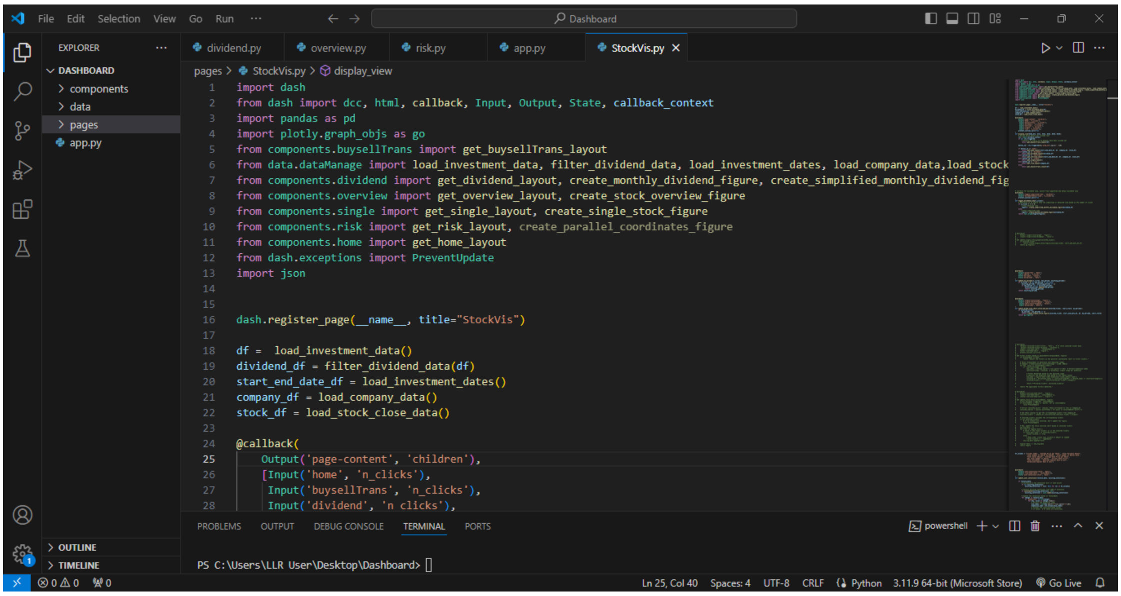


Figure 3.2.3.2: *A screenshot of this project on Visual Studio Code, (VS Code, 2024).*

4 Project Plan and Timetable

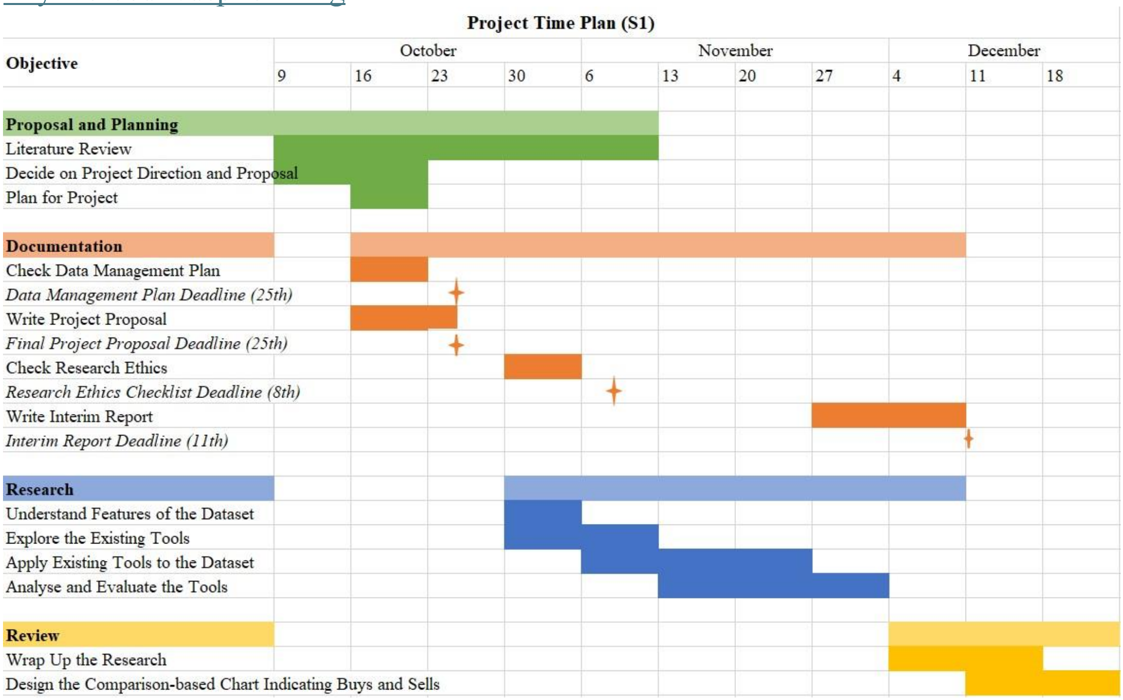
The project was broken down into smaller, manageable tasks using a divide-and-conquer approach, as suggested in Bob's Project Guidelines (Laramee, 2011). The timetable of this project is illustrated in the Gantt Charts below. To facilitate precise tracking of the schedule, according to Bob’s Minutes of Meeting Protocol, comprehensive details on the plan and weekly progress can be fetched from the documented weekly meeting minutes. The related meeting minutes can be fetched via the following links:

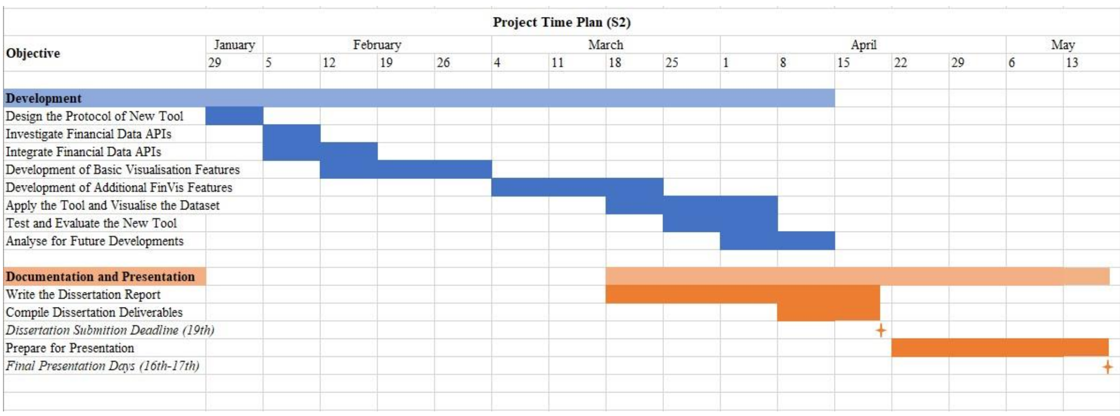
Year3 Project Student Group:

https://docs.google.com/document/d/1hHet4IjfqvmF2DtAVsJ0HWqDpQPLzHXgvTyzfZiHQ_4/edit?usp=sharing

StockVis Group:

https://docs.google.com/document/d/1jvSTDCbzuLoCJahr0fWbhZxJNiqvHVj1KQ0yNIy46b0/edit?usp=sharing





5 Project Design

The project design not only shapes the direction of the project but also addresses the key implementation challenges. This section provides a comprehensive overview of the visualization process and delves into the essential design components. Section 5.1 shows the visualization pipelines implemented in this project. Section 5.2 provides the process diagram of the development workflow.

5.1 Visualization Pipeline

The Visualization Pipeline is elaborated in a survey of visual analytic pipelines by Wang *et al.* (2016). Their work establishes a comprehensive framework for transforming raw data into meaningful visualizations. This framework highlights the systematic processes of data preparation, analytics, and visual representation crucial for interpreting complex datasets effectively. Figure 5.1.1 shows an example of the model design driven pipeline that they proposed. The survey emphasizes a structured approach to ensure that each step, from data acquisition to visual output, is optimized to enhance the analytical value and interpretability of the visual data presented.

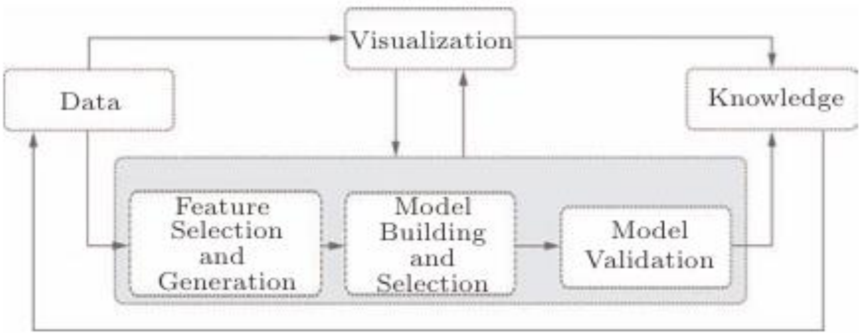


Figure 5.1.1: *Figure of model design driven visualisation pipeline by Wang et al. (2016).*

Building upon the foundational concepts in the survey, the pipeline by Endert et al. (2017) was adapted for this project to systematically illustrate the process of creating visual representations from source data to final views. This visualization pipeline (Figure 5.1.2) effectively transitions data through various stages. Specifically tailored to this project's needs, the adapted pipeline accommodates the unique requirements of domain-specific data, facilitating the development of customized visualizations that enable a deeper understanding and more refined visual displays.

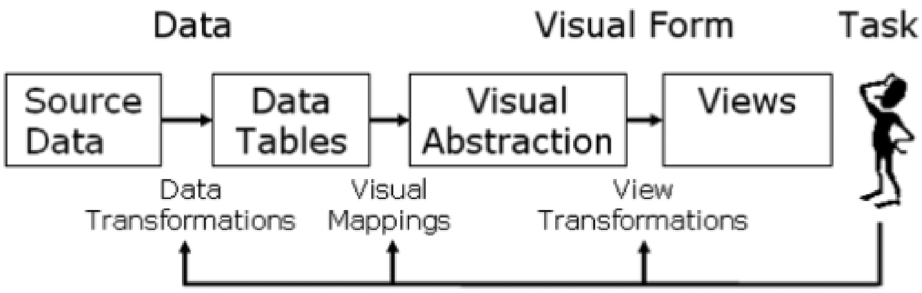


Figure 5.1.2: *Figure of visualisation pipeline by Endert et al. (2017).*

5.2 Process Diagram

The process diagram presents a streamlined workflow that captures the sequential steps and user interactions involved in the financial data visualization tool. The initial stage, Data Acquisition, involves gathering financial datasets from various sources, ensuring a robust foundation for analysis. This is followed by Data Modification, where the raw data undergoes cleansing, normalization, and transformation to align with the required analysis criteria. Once the data is refined, it transitions into the Data Format phase, where it's structured into a format amenable to visualization tools and techniques.

The user-centric nature of the tool is encapsulated in the branches leading to User Selection and User Option. User Selection influences Data Modification, allowing users to tailor the data processing based on their analytical focus. Similarly, User Option feeds into the visualization stage, where users' preferences shape the presentation and interactivity of the financial data. The culmination of this process is the Visualization step, where data is rendered into charts and graphs, facilitating user interpretation and analysis. This stage represents the graphical user interface of the tool, where the processed data is displayed in an interactive and visually engaging manner.

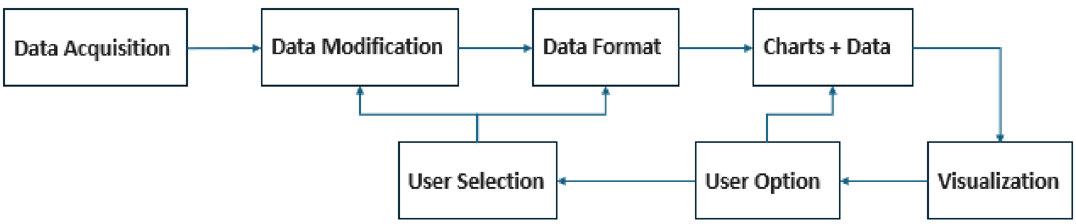


Figure 5.2.1: *Figure of the Process Diagram.*

6 Implementation

As outlined in Bob’s Project Guidelines (Laramee, 2011), this section systematically documents the implementation of the developed system. It covers all critical components as described in Section 3.1 Feature Specification. To present the implementation process more efficiently, related functionalities are integrated, and basic features implemented in the same view are described together in the same subsection. Apart from detailing the features and functionalities achieved for each component, this section will also discuss the challenges encountered during the project's development phase. Screenshots for each component are also included for illustration.

The implementation process has been divided into three stages. The first stage is using the existing visualisation tool to investigate the datasets, discovering their strengths and limitations before the development of the new tool, as discussed in Section 6.1. The second stage detailed in Section 6.2 is the basic implementations corresponding to Must-have Features discussed in Section 3.1.1. The final stage detailed in Section 6.3 is the enhancements which implements the optional features discussed in Section 3.1.2.

Figure 6.1: *A screenshot of the dashboard home page, which shows the default view.*

Figure 6.1 offers an overview of the system, a dashboard for customer-centric stock investment analysis. Positioned on the left corner are seven buttons, facilitating users to switch among various views that display different charts for detailed data examination.

The home page aggregates visual representations from other views, except the single stock page. At the top left section, there is a multiple line chart showcasing the trends of stocks appeared the Investment Transaction Dataset. This feature empowers users to compare stocks, adjust settings such as moving averages, and apply filters for focused analysis on specific stocks. Users also have the option to display trends following either the last sale or the last purchase. The bottom left section of the home page is dedicated to a parallel coordinate chart ticker linked to the multiple line chart, displaying the company metadata. This chart is colour-coordinated with the multiple line chart for visual consistency. User interaction options such as the brushing functionality is also implemented. At the centre of the dashboard is a stacked bar chart that displays the total dividends and capital gains or losses for each stock, sorted from the highest gain to the largest loss. In the upper right quadrant, another stacked bar chart illustrates the monthly buy and sell transactions across two accounts. Adjacent to this, two charts delve into dividend details: one represents the monthly dividend activities per ticker, and the other aggregates the total dividends per ticker, with a clear distinction between different types of dividends. The specific implementations for each view and chart will be detailed from Section 6.2 onwards, providing a thorough narrative of how each component contributes to the overarching functionality of the system.

6.1 Existing Tools

Before implementing the features to build the new Customer-Centric Visualization Tool for Stock Investments, an exploration into the existing tools was conducted using the Investment Transaction dataset. This focus is rooted in the need to develop solutions that not only enhance the clarity and accessibility of financial data but also contribute to a deeper understanding of specific consumer financial behaviour. Based on different kinds of visualisation resources available to researchers, it is a novel approach to analysing the strengths and limitations of the existing visualization tools based on a customer-centred investment dataset. This approach involves evaluating existing visualization methods and leveraging the unique properties of this dataset to uncover limitations and new possibilities in the realm of financial visualization. This research and experimentation process contributes to the development of a new customer-focused tool that is innovative and practically applicable in the world of financial data analysis.

Four popular visualisation tools are explored before the development of the new tool. They are Google Chart, Excel, Tableau and Power BI, discussed from Section 6.1.1 to Section 6.1.4. Some sample charts generated by these tools are presented and the tools' strengths and limitations are discovered. The charts generated by existing tools are for explorations only, which are not included in the new system developed for this project.

6.1.1 Google Chart

In the realm of financial data visualization, Google Charts emerges as a versatile and accessible tool, adept at integrating a variety of chart types into web applications. As explored in a study focused on business intelligence dashboards, Google Charts offers a diverse array of approximately 12 chart types, including pie, scatter, gauge, geochart, table, treemaps, combo, line, bar, column, area, and candlestick (Amin et al., 2022). This wide range facilitates the versatile representation of data from various sources, with the added advantage of easy integration into web pages. The tool's capability to display data across different browsers and platforms enhances its accessibility and user-friendliness. Its integration into web pages allows for the effective visualization of a broad spectrum of information, thereby playing a crucial role in facilitating informed decision-making (Amin et al., 2022). This aspect of Google Charts is particularly beneficial for projects that demand quick and clear visualization of data within a web environment. The tool's straightforward implementation process makes it a good choice for projects requiring rapid deployment of data visualization in web-based applications. As shown in Figure 6.1.1.1, Google Charts is proficient in presenting data in an easily understandable format.

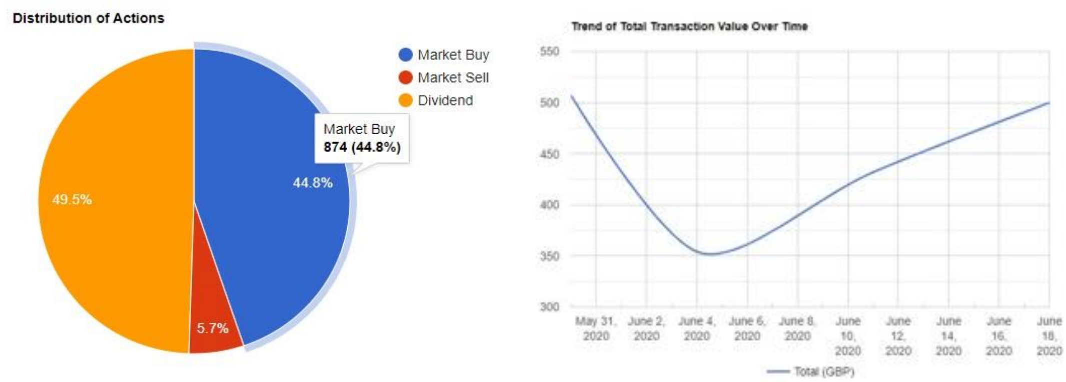


Figure 6.1.1.1: *A pie chart showing the distribution of number of actions in the Investment Transaction Dataset and a line chart showing the trend of transaction amount over a specific period, which indicates the activity level of the investment over this period. The charts are generated by Google Chart for exploration.*

Limitations:

Google Charts' ability to embed into web pages using JavaScript and to process data from sources like RESTful web services and server-side databases indicates a moderate level of versatility in handling data (Amin et al., 2022). However, when it comes to processing large CSV datasets, Google Charts shows limitations. The tool can handle standard datasets efficiently, but as the size and complexity increase, it may struggle with performance issues, leading to slower load times and potential difficulties in rendering intricate data patterns. This might hinder its applicability for in-depth financial analysis, where datasets are often extensive and multifaceted.

In terms of customization and interaction, Google Charts offers basic to moderate customization options. However, compared to more specialized data visualization tools, it falls short. The customization and interaction limitations could be significant when dealing with specialized financial data visualization needs, such as custom interactive elements or advanced data filtering and manipulation. Another limitation relevant to financial data visualization is Google Charts' dependency on internet connectivity. For scenarios where data security is paramount, or in environments with limited or unreliable internet access, this could be a significant drawback.

6.1.2 Excel

Excel, with its widespread availability and user familiarity, serves as a robust tool for creating financial dashboards. Excel's strengths lie in its capabilities for data aggregation and manipulation. It offers a suite of tools for sorting, splitting, and removing duplicate data, essential for preparing datasets for visualization (Kusleika, D., 2022). One of Excel's most powerful features for dashboard creation is the PivotTable, which allows users to filter, sort, and summarize data dynamically. This interactivity is key for financial analysis, where data conditions and parameters frequently change. For the visualization based on this Investment Transaction Data, dashboards in Excel that includes a variety of charts are considered, each tailored to specific financial actions like market sells, buys, dividends, deposits, and withdrawals (Figure 6.1.2.1). Each chart focuses on attributes relevant to the specific financial action, making the dashboard a comprehensive tool for analysing diverse financial transactions.

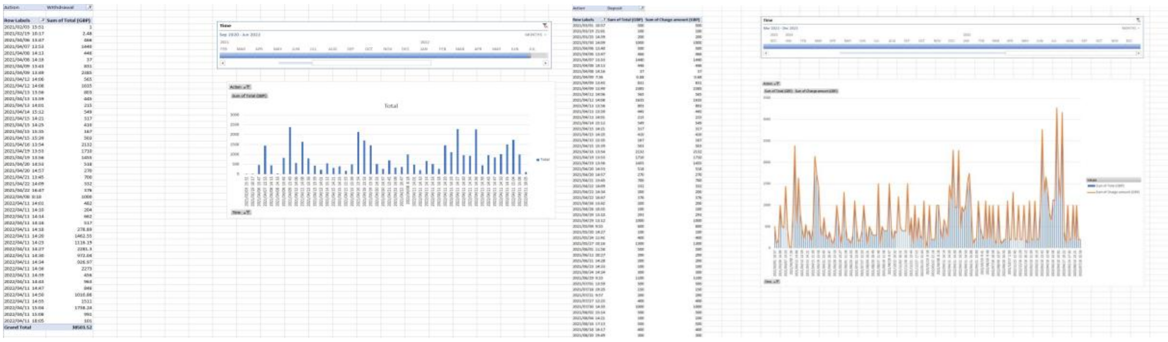


Figure 6.1.2.1: *A simple dashboard with Action filter and time filter created by Excel. The two screen shots show the withdraw amount and deposit trend, respectively.*

Limitations:

Excel offers a range of visualization options, but it lacks some of the more advanced features available in specialized visualization tools. This can be a constraint when trying to convey complex financial data insights through more sophisticated visual formats. Compared to other visualization tools, Excel dashboards tend to be less interactive. This can limit the user's ability to explore data dynamically, which is often crucial in financial analysis for understanding evolving market conditions and investment scenarios. Besides, while Excel allows for a certain degree of customization, it may not offer the same level of flexibility as some dedicated visualization tools. This can be a limitation when tailoring dashboards to specific user requirements or when trying to incorporate unique visualization elements. Additionally, managing extensive and complex datasets in Excel can be challenging. When dealing with large volumes of financial data, Excel might struggle with performance issues, leading to slower processing times and difficulties in rendering detailed visualizations efficiently. Effective use of Excel for financial visualization often requires a good understanding of its features and functionalities. Users without sufficient expertise in Excel may find it difficult to exploit its full potential for creating comprehensive and insightful financial dashboards.

6.1.3 Tableau

Tableau is highly regarded in financial data visualization for its advanced capabilities and user-friendly interface, ideal for intricate data analysis (Loth, A. and Sparkes, S., 2019). It offers an array of features for crafting interactive dashboards that blend diverse data sources, including live feeds essential for timely financial decisions. Users can easily navigate through complex data for trend analysis, portfolio tracking, and risk assessment. Tableau's seamless integration with various databases and APIs, coupled with its customizable visualization options, makes it a preferred choice for financial analysts looking to transform raw data into clear insights.

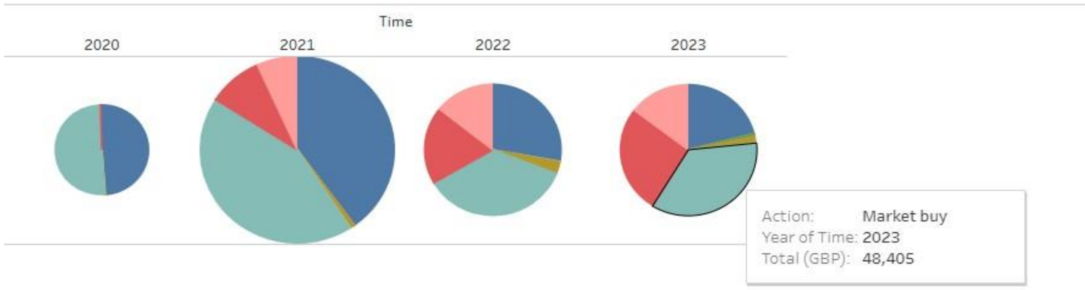


Figure 6.1.3.1: *An interactive pie chart showing the transaction distribution and total transaction amount over year by Tableau.*

Limitations:

Tableau's performance and the ease of creating visualizations can be highly dependent on the structure of the underlying data. In cases where data is not well-structured or standardized, users may face significant challenges in visualization. As organizations grow, the scale of data can increase exponentially. Tableau, although powerful, may not always scale as efficiently or cost-effectively compared to other enterprise-grade solutions that are designed for big data environments. For all its capabilities, Tableau can be resource-intensive, particularly when rendering very large or complex datasets, which can lead to longer processing times and may require more powerful hardware for optimal performance. Moreover, the most advanced features and capabilities of Tableau are often locked behind higher-tier licensing models. This makes these options cost-prohibitive and a significant investment for smaller firms or individual users.

6.1.4 Power BI

Power BI is a powerful data analytics tool that excels in data modelling and visualization, making it a strong candidate for creating financial dashboards. Its deep integration with the suite of Microsoft products and regular updates makes it an advanced platform for data analysis. The tool is designed to handle complex data sets and turn them into interactive, insightful visualizations. Analysts can utilize it to track metrics, identify trends, and make predictions based on historical data. The comprehensive set of features solidifies Power BI as a valuable resource in financial analytics and strategic planning.

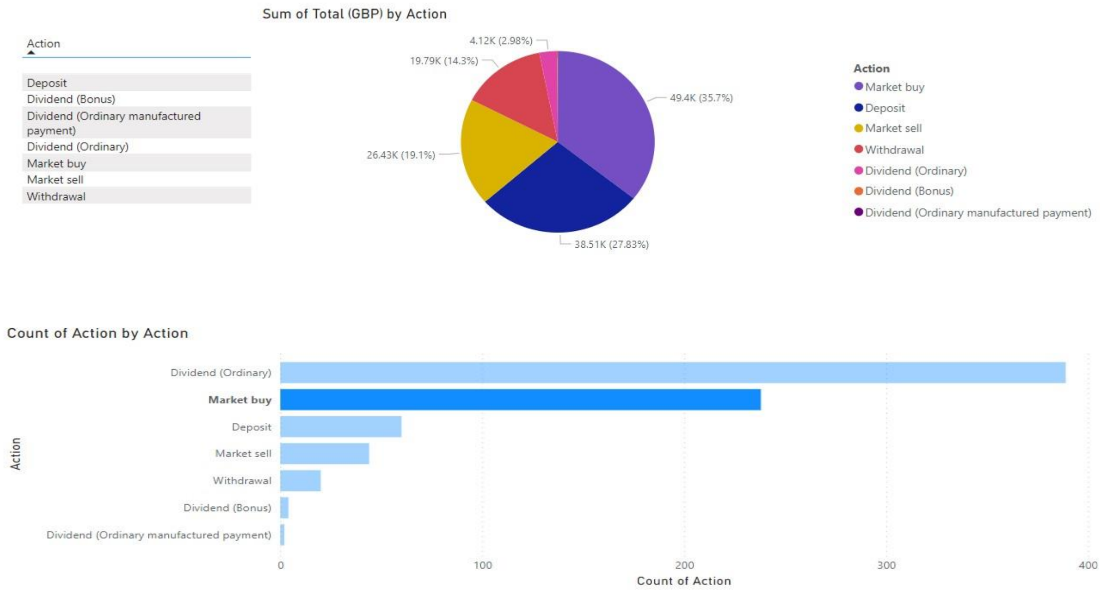


Figure 6.1.4.1: *An interactive pie chart showing the transaction distribution and a bar chart showing the count of each action during a selected period created by Power BI.*

Limitations:

Power BI relies heavily on the Microsoft ecosystem, which might present integration challenges with non-Microsoft products or data sources. Although Power BI integrates well within the Microsoft ecosystem, connecting with external or proprietary financial services and databases can sometimes be challenging and may require custom connectors or gateways. There can be limitations on the volume and frequency of live data feeds, especially in the free or Pro versions, which may not suffice for high-frequency trading or real-time financial monitoring. Besides, implementing row-level security in Power BI can be complex, and managing access for sensitive financial data may require advanced setup that is not immediately intuitive. While Power BI offers a variety of built-in visuals, creating custom visualizations for specific financial scenarios often necessitates familiarity with programming languages like JavaScript, which can be a limitation for analysts without a coding background.

In summary, while these tools provide robust data analysis capabilities, they often fall short in delivering tailored, customer-centric financial visualizations without requiring substantial coding skills for creating insightful charts. The tool developed in this project is designed to address these limitations by offering a clear and user-friendly financial visualization system that can effectively convey extensive information.

6.2 Basic Implementation

This section details the basic implementation of the system. Section 6.2.1 introduces the Dashboard User Interface, which serves as the foundation for displaying all views and charts implemented. Section 6.2.2 to 6.2.5 introduce four out of seven different views of the dashboard UI, including the Buy/Sell View, Gain/Loss View, Dividend View, and Single Stock View. Section 6.2.6 and 6.2.7 describe two main charts displayed in the dashboard: the Parallel Coordinate Chart and the Multiple Line Chart. The former is displayed in the Company View, showing the company metadata, and the latter in the Multiple Stock View, providing an overview of stock performance. Both are included in the home view discussed in Section 6.2.8. In addition to the creation of charts and views, basic implementations also include Colour Maps and User Options within the dashboard, discussed in Section 6.2.9 and 6.2.10, respectively. General functionalities such as filters and selections are implemented across all charts using various approaches, which will not be covered in a single section. Instead, each approach for this functionality will be detailed in its respective section to ensure clarity and depth of understanding.

6.2.1 Dashboard UI

The Dashboard UI is a central feature of this FinanceVis tool, pivotal to ensure a smooth system operation. Its implementation permeates throughout the system utilizing a user-centric design approach, which features multiple views including a dashboard (Figure 6.2.1.1) for a comprehensive financial analysis experience.

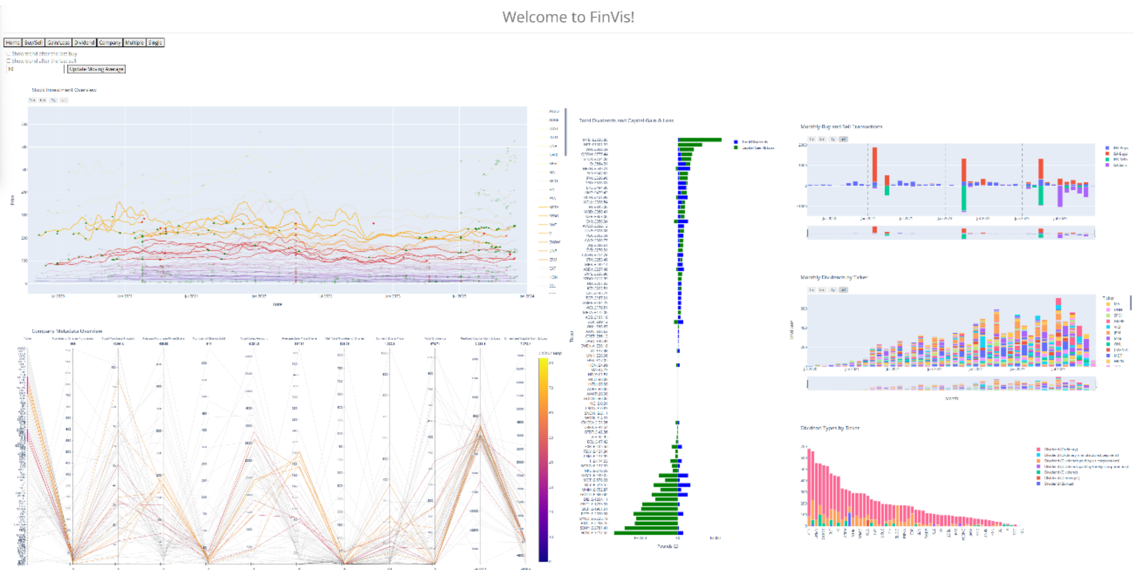


Figure 6.2.1.1: *The screenshot of the dashboard with user interactive activities.*

The fundamental UI structure and interactivity are facilitated through a blend of Python libraries associated with web application frameworks and front-end technologies such as ‘Flask’, ‘HTML’, and ‘JavaScript’. The Python library ‘Dash’, built on top of ‘Flask’, ‘Plotly.js’, and ‘React.js’, plays a pivotal role. Its approach consolidates the backend and frontend development into a unified Python codebase, minimizing the need to handle various technologies across different parts of the application. Instead of directly dealing with ‘Flask’ and front-end languages, it offers a Python-centric way to create interactive web applications while leveraging the flexible structure of ‘Flask’, which serves the UI components efficiently. The main executable ‘app.py’, as the entry point, initiates the Flask server and defines the routes to the different views of the dashboard. Within this file, the function ‘render_template’ from ‘Flask’ is utilized to serve ‘HTML’ templates, integrating ‘Jinja2’ template for dynamic content rendering.

The UI is structured to offer seven distinct views: Home, Buy/Sell, Dividend, Multiple, Single, Risk Factors, and Gain/Loss. Switching between these views is implemented through a series of Buttons (Figure 6.2.1.2) located in a consistent navigation area on the dashboard, making the transitions intuitive for the user. Each view corresponds to a separate page within the application, facilitating a separation of concerns and enhances the system’s maintainability and scalability. This modularity is reflected in the ‘pages’ folder, with ‘StockVis.py’ acting as the handler for these pages. This file also manages the layout across the system’s numerous pages and views. Dash’s layout is declared in “Pythonic” syntax which under the hood translates to the corresponding ‘HTML’, ‘CSS’, and ‘JavaScript’ for browser rendering. This simplifies the implementation process by allowing to use Python syntax to control the layout and style of the web pages, without being bogged down by cross-technology complexities.

Figure 6.2.1.2: *The screenshot of the buttons on the top left corner of the dashboard.*

Moreover, the dashboard is composed of a series of app ‘callbacks’ that respond to user interactions. User inputs such as button presses or dropdown selections, trigger callbacks that update the dashboard's visual components in real-time. For example, adjusting the moving average triggers callbacks that update the multiple line chart with the relevant data. Details will be covered in the following sections.

6.2.2 Buy/Sell View

The Buy/Sell view as illustrated in Figure 6.2.2.1 provides users with insights into their transaction history. This view is structured to represent transactional data through a stacked bar chart, which offers a clear visualization of monthly buy and sell activities across two different accounts. The implementation is managed by the ‘buySell.py’ file within the ‘components’ folder. ‘Dash’ and ‘Plotly’ was leveraged to create interactive and responsive visualizations. Using the ‘Plotly’ graphing library, the stacked bar chart is constructed with distinct colours representing buys and sells for two distinct accounts, which allows users to easily distinguish between the two types of transactions at a glance. The chart also enables users to hover over specific segments to see detailed information about the transactions, such as the exact amounts and dates.

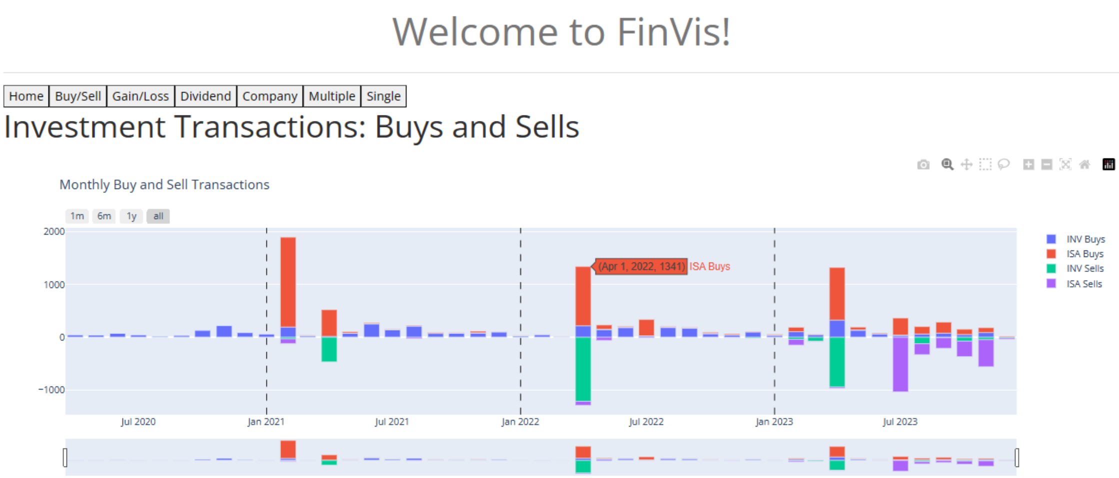


Figure 6.2.2.1: *A screenshot of the Buy/Sell View, with a stacked bar chart showing monthly buy and sell transactions, illustrating cursor hover interactions.*

The transaction data is processed and managed in ‘dataManage.py’ located in the ‘data’ directory. This file is responsible for handling the underlying datasets and preparing the data in a format suitable for visualization. It reads the transaction data from the datasets, computes monthly totals for buys and sells, and then passes this aggregated information to the ‘buySell.py’ script for visualization. Dash's callback functions in ‘StockVis.py’ are utilized to update the chart in response to user interactions. These callbacks enable dynamic updating of the visualization based on user-selected filters, such as selecting a specific time range or a particular account to display, as demonstrated in Figure 6.2.2.2. The implementation of the Buy/Sell view achieves a seamless blend of data processing, visualization, and user interaction within a single, cohesive Python environment.

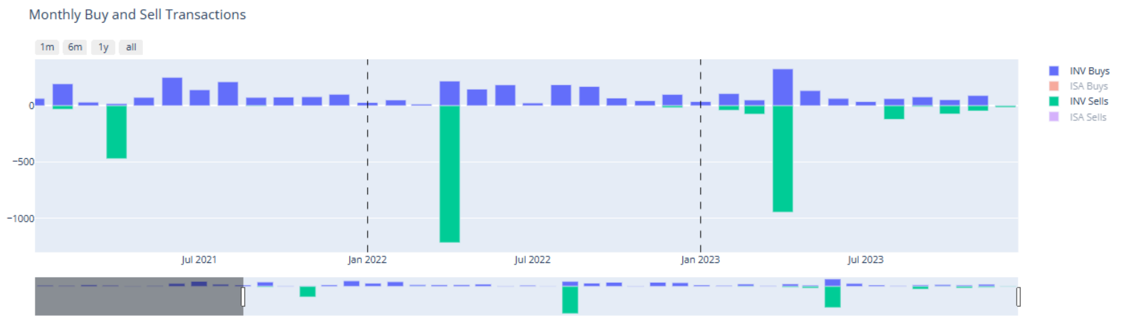


Figure 6.2.2.2: *A screenshot of the Buy/Sell View with filtered account and time period, illustrating user filter and selection interactions.*

6.2.3 Gain/Loss View

The Gain/Loss view of the system is designed to give the user insightful visualization of the investment outcomes in terms of capital gains, losses, and dividend earnings through a stacked bar chart. As illustrated in Figure 6.2.3.1, performance metrics are divided into two categories: capital gains and dividends, each distinguished by different colours in the chart. This categorical distinction is clearly annotated in the chart's legend, aiding users in quickly identifying the type of returns.

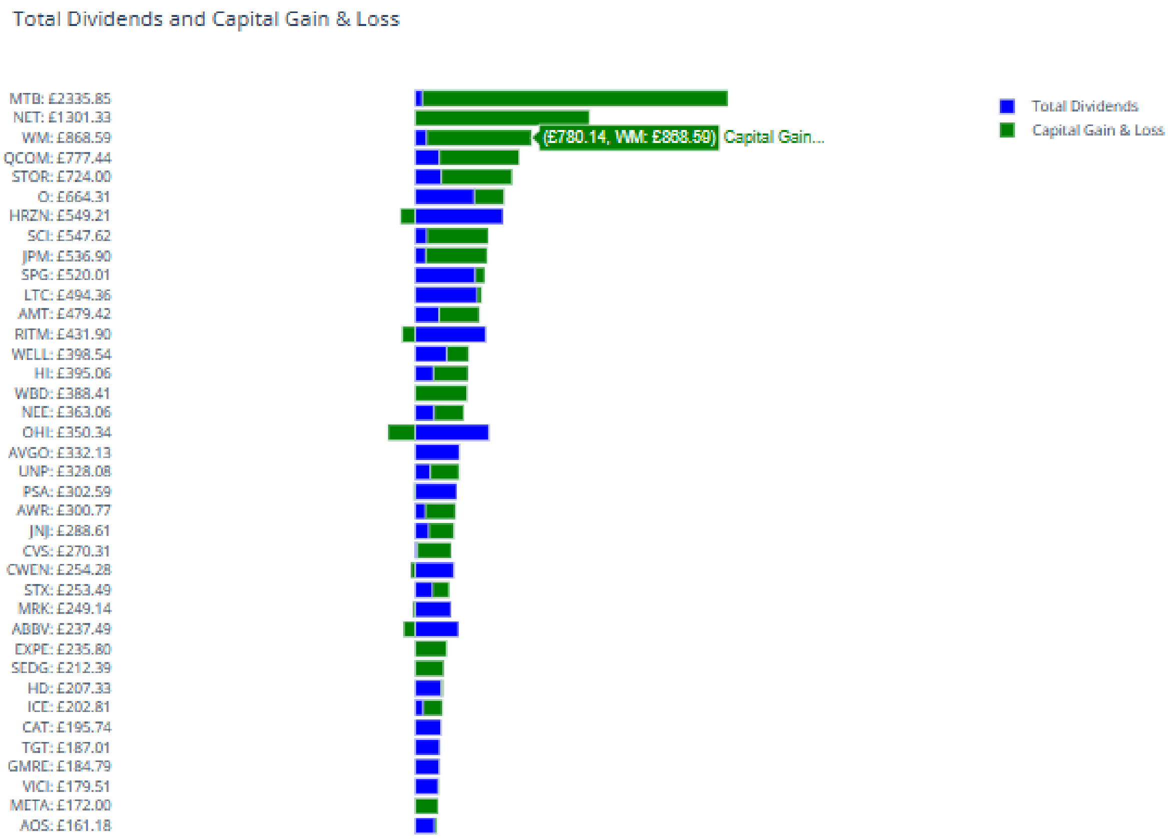


Figure 6.2.3.1: *A screenshot of the upper section of the Gain/Loss View.*

This stacked bar chart is structured with the Y-axis representing various companies in which investments have been made, and the X-axis showing the amount of investment performance. Two different bars represent different aspects of performance, allowing users to quickly discern the profitability of their investments. This view is implemented through the ‘gainLoss.py’ file within the ‘components’ directory. This file utilizes Python’s Dash and Plotly libraries to create and manage the visual representations of the financial data related to the outcomes. The data is processed by ‘dataManage.py’, which prepares and aggregates the financial metrics required for the visualization, including capital gains, capital losses, and dividends earned from stocks.

In this chart, users could filter and sort data dynamically. This is achieved through a series of interactive controls that allow users to isolate specific performance categories including gains & losses and dividends. As illustrated in Figure 6.2.3.2 and 6.2.3.3, when a specific category is selected for detailed analysis, the chart responds by reordering the companies based on the selected category’s performance metrics. This dynamic ranking is handled by the callback function (Figure 6.2.3.4) within Dash, which listen for user inputs and adjust the visualization accordingly. For instance, selecting ‘Total Dividends’ by double clicking on the legend would reorder the companies on the Y-axis according to the dividend, offering users a customized view that focuses on their area of interest.

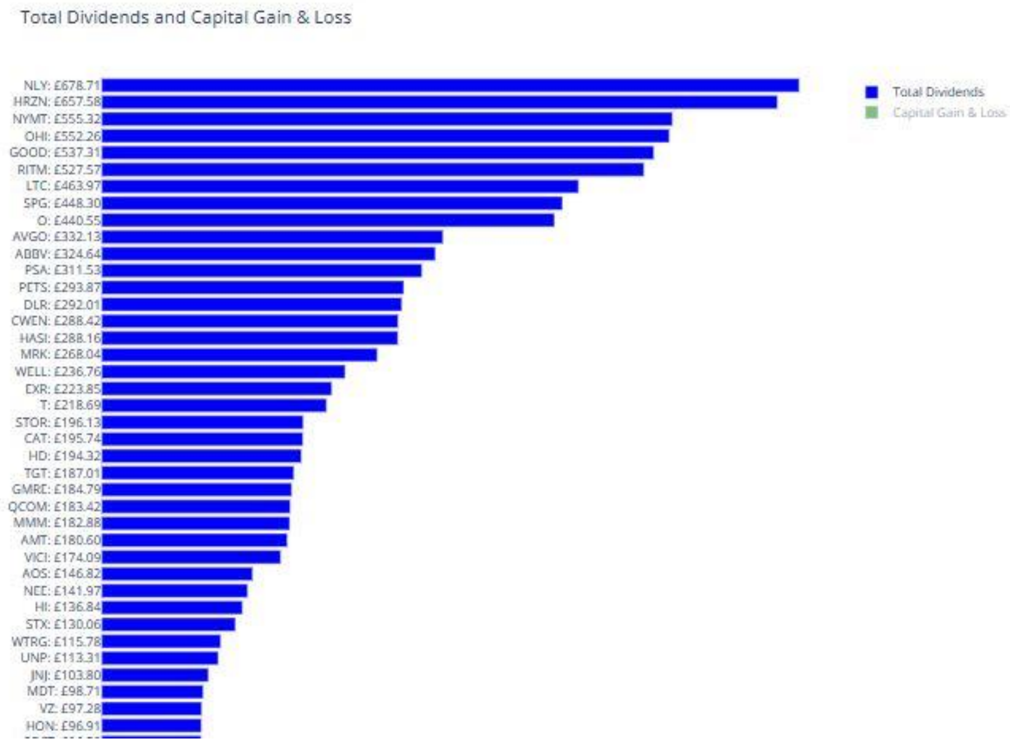


Figure 6.2.3.2: *A screenshot of the Gain/Loss View when Total Dividends is selected.*

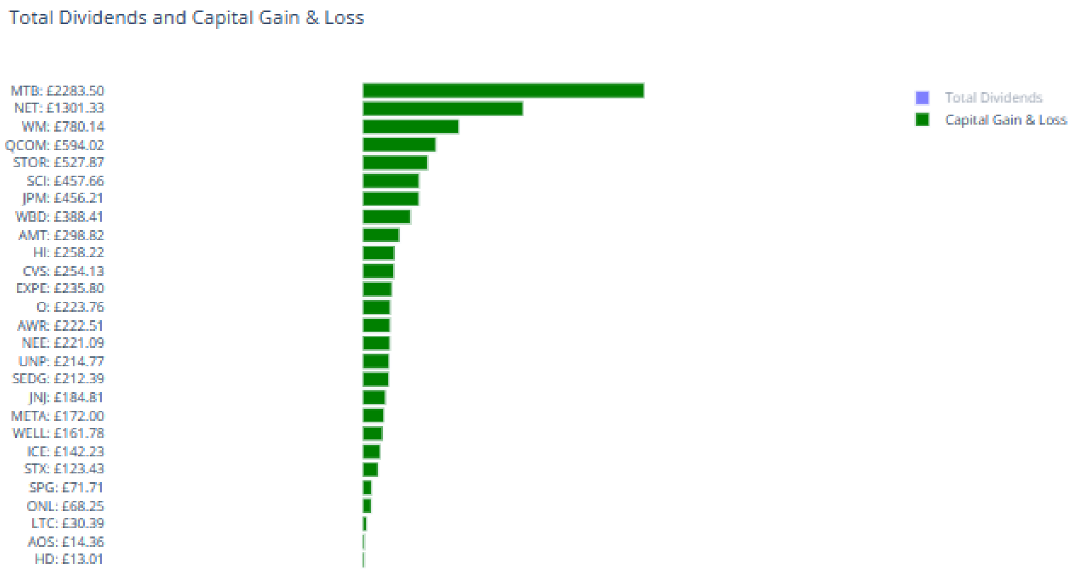


Figure 6.2.3.3: *Screenshot of the Gain/Loss View when Capital Gain & Loss is selected.*

```python
@callback(
    Output('gain-loss-chart', 'figure'),
    [Input('gain-loss-chart', 'restyleData')],
    [State('gain-loss-chart', 'figure')]
)
def update_chart(restyle_data, existing_figure):
    ctx = dash.callback_context
    if not ctx.triggered or not restyle_data:
        sort_column = 'Total'
    else:
        visible_traces = restyle_data[0]['visible']
        if visible_traces[0] is True:
            sort_column = 'Total Dividends'
        else:
            sort_column = 'Realized Capital Gain & Loss'

    fig = create_gain_loss_chart(company_df,sort_column)

    return fig
```

Figure 6.2.3.2: *A screenshot of the callback function for the Gain/Loss View.*

6.2.4 Dividend View

The Dividend view within the system specifically targets the dividend payments, structured through two distinct stacked bar charts. They effectively support investment decision-making by highlighting the frequency, consistency, and types of dividends paid by each stock in user's portfolio. Each chart serves a distinct purpose, providing both a snapshot and a detailed temporal analysis of dividend distributions. The first chart illustrates the total dividends per company (Figure 6.2.4.1), broken down by the type of dividend with each ticker on the x-axis and the total GBP on the y-axis. Different colours or segments of the bars represent the different types of dividends. This visualization helps quickly assess which stocks are paying what types of dividends and the relative proportions of these payments.

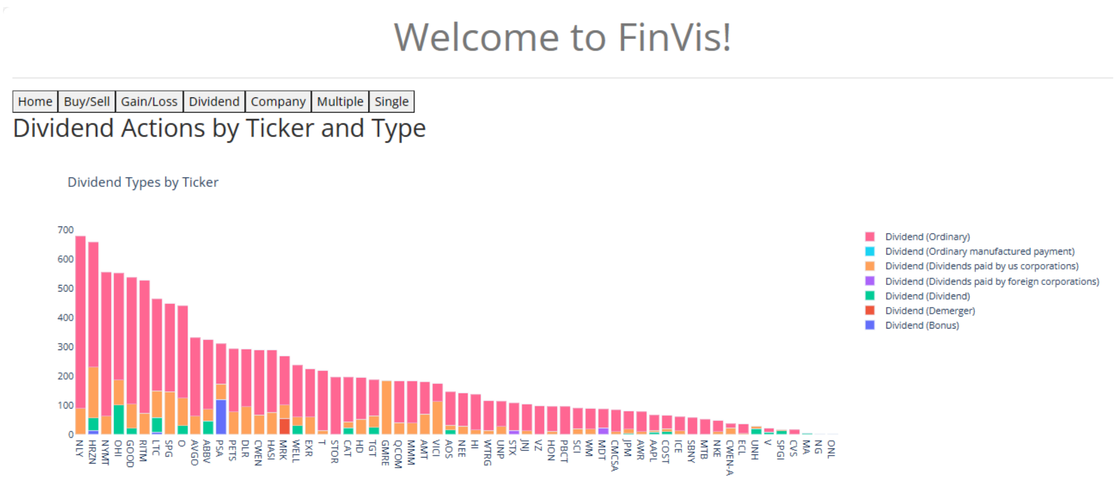


Figure 6.2.4.1: *A screenshot of the first stacked bar chart showing dividend value and type in the Dividend View. The data is sorted in descending order.*

The second chart (Figure 6.2.4.2) tracks how dividends for each company have varied across different time periods grouped by month, providing a temporal analysis of dividend trends. The x-axis represents time by month, and the y-axis represents the total GBP. The data processing for the Dividend View in the FinanceVis tool involves a series of Python functions designed to filter, aggregate, and prepare dividend data for visualization. Initially, the 'filter_dividend_data' function sifts through the dataset to include only relevant dividend transaction types, ensuring that the data is pertinent and streamlined. Subsequently, the 'aggregate_dividend_data' function groups this filtered data by ticker and type of dividend, summing up the total dividends paid and organizing them in a structured format for visualization. The 'aggregate_dividend_data_by_month' function extends the aggregation process by also incorporating the temporal dimension, grouping dividends by month, ticker, and type.

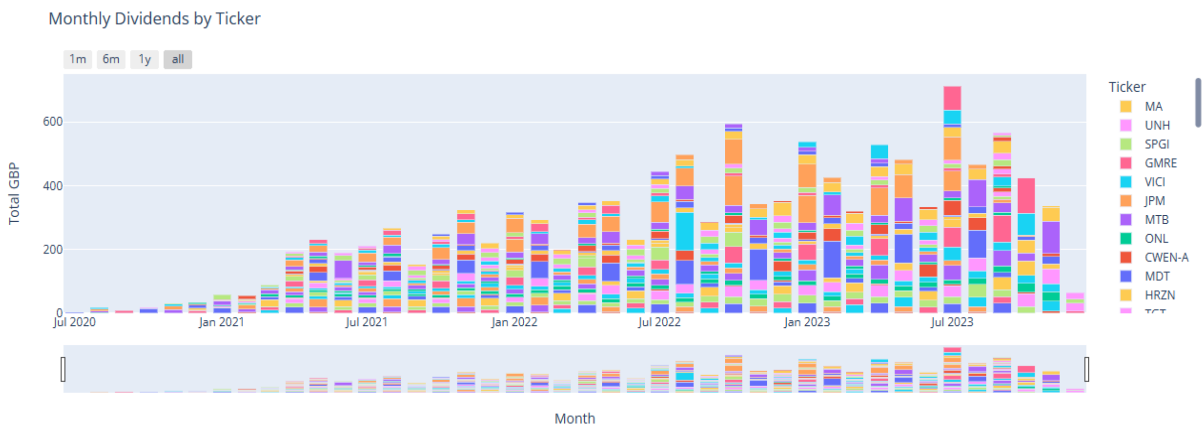


Figure 6.2.4.2: *A screenshot of the second stacked bar chart showing monthly dividend value over time in the Dividend View.*

Both charts were created by 'Plotly.py', given its ability to handle complex interactions and dynamic data updates efficiently. The charts are interactive, allowing users to hover over data points for more detailed information, filter based on specific criteria, and toggle between different views, which will be detailed in the enhancement section. Dash callbacks are extensively used to refresh and update the charts based on user interactions, such as selecting different companies or adjusting the time (Figure 6.2.4.3).

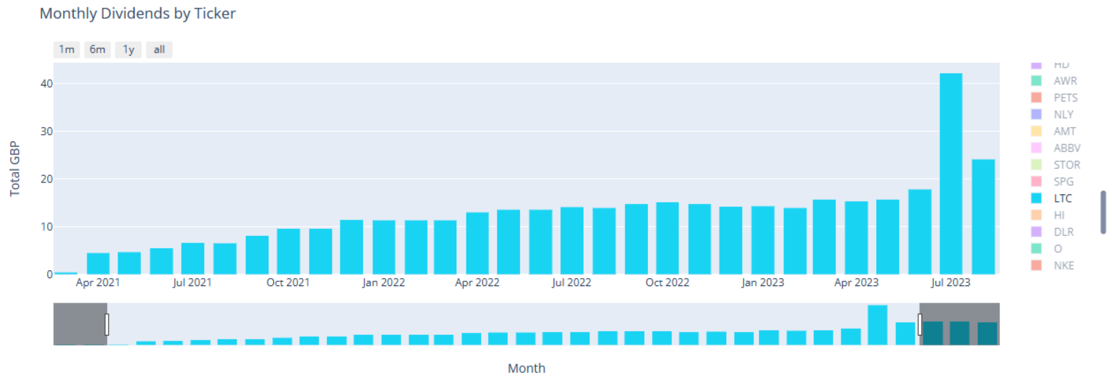


Figure 6.2.4.3: *A screenshot of the second stacked bar chart when ticker "LTC" was selected and time was filtered, showing the trend of dividend of a single stock over time.*

6.2.5 Single Stock View

The Single Stock View in the system offers a detailed examination of individual stocks through interactive and informative visualizations. This view focuses on the integration of price trends, moving averages, and distinct buy and sell points to aid users in assessing their trading decisions. The primary component of this view is a line chart that plots the daily closing prices and the transaction history for buys and sells of a selected stock over period between the first buy action to the last buy or sell action in the transaction dataset.

Figure 6.2.5.1: *The line chart showing "CMCSA" trend in the Single Stock View.*

Buy and sell points are superimposed on the closing price line chart as distinct markers. Buy points is indicated with green upward-pointing triangles, while sell points shown as red downward-pointing triangles, the position of these points is the exact buy and selling price from the transaction dataset. These markers provide visual cues that help traders quickly identify historical transaction points against the price trend, facilitating analysis of past buying or selling decisions, as shown in Figure 6.2.5.1.

In addition to the closing price, users can input the number of days for moving averages, which are then calculated and displayed as additional lines on the chart. These moving averages help smooth out price data to identify trends more clearly. Users also have the flexibility to view multiple moving averages simultaneously, as illustrated in Figure 6.2.5.2, aiding in comparative trend analysis and decision assessment.

Figure 6.2.5.2: *The line chart showing the trend and transaction history of ticker "CAT", with 10 days and 50 days moving average lines added.*

Interactive features and user interactions in the Single Stock View are sophisticatedly managed through dynamic data interaction and detailed tooltips. Dash callbacks are employed to efficiently handle real-time updates to the chart in response to user inputs such as stock selection, adjustments in moving average parameters, and changes in the observed period. Additionally, hovering over any point on the chart such as closing and moving average lines and buy/sell markers provide users with detailed information, including the exact price at that point, the specific date, and any other relevant metrics. This combination of these interactive functionalities enhances the user experience by providing immediate, in-depth data access directly on the visualization interface. Figure 6.2.5.3 illustrates the section box where user can select the ticker that they would like to investigate, the chart type selection box will be further detailed in enhancement section.

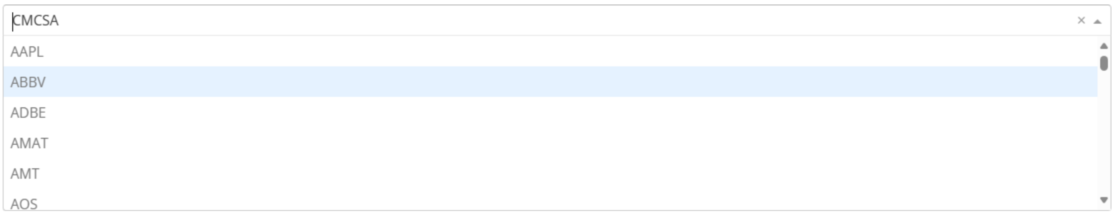


Figure 6.2.5.3: *The selection box on the top of the Single Stock View for ticker selection.*

6.2.6 Parallel Coordinate Chart

The Parallel Coordinate Chart in this tool provides a multidimensional representation of various financial metrics and risk indicators of tickers across different companies, as illustrated in Figure 6.2.6.1. This visualization enables the analysis of patterns and correlations between multiple data dimensions in a single interactive display. It is designed to handle a high volume of data points without losing performance, employing advanced rendering techniques to ensure smooth user interactions and transitions. Using the 'plotly.graph_objects', axes are carefully scaled and labelled to maintain readability and clarity. The chart plots multiple axes parallel to each other, each representing a different financial metric such as Average Purchase/Sell Price, the Total Purchase/Sales Amount, current Share Price and Realized/Unrealized Capital Gain & Loss, which serve as the risk indicators of the company. The data for creating this chart is loaded from 'dataManage.py', which reads the CSV file for the company metadata. Data points from different companies are displayed as lines intersecting these axes, where the position on each axis corresponds to the company's performance in that specific metric.

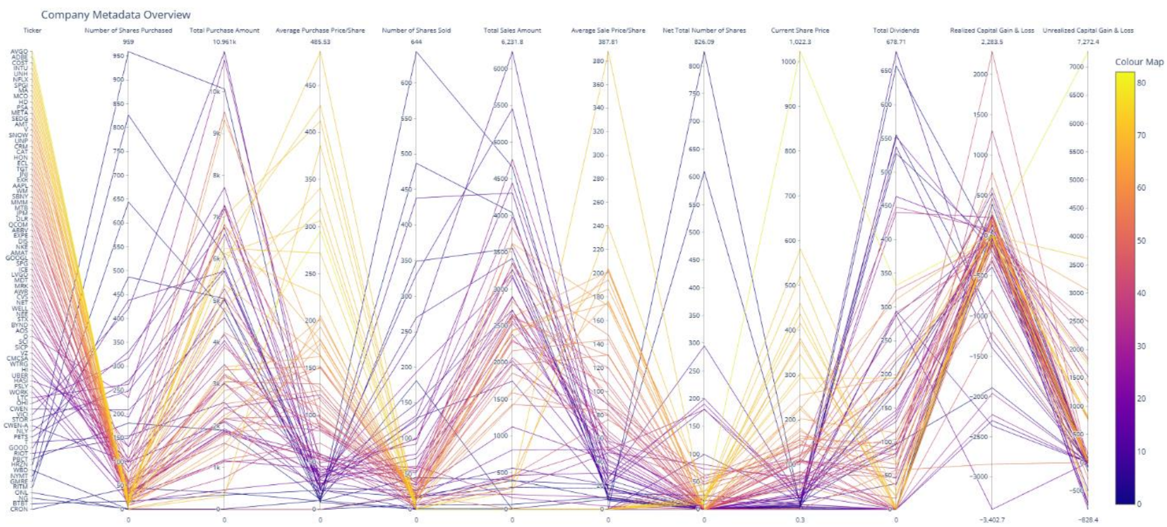


Figure 6.2.6.1: *The parallel coordinate chart showing the metadata of each company.*

The chart is constructed using the python library ‘plotly.graph_objects’, which allows for customizable visualizations. This ‘Graph Objects’ module is specifically used to create this parallel coordinate plot, providing tools to adjust properties like axis ranges, labels, and interactive features. Interactive brushing is enabled on each axis, allowing users to focus on specific data ranges or isolate companies based on their interest. This interactivity is crucial for detailed analysis on specific tickers selected by users. The “focus and context” technique is utilised here, the chart renders selected companies with colour and represents unselected entities transparently to provide a clear focus on the selected companies. Selection of multiple data ranges in the same axis is also allowed, and the selected ranges are automatically combined. The focused data is the intersection of the range selected in each coordinate, as illustrated in Figure 6.2.6.2.

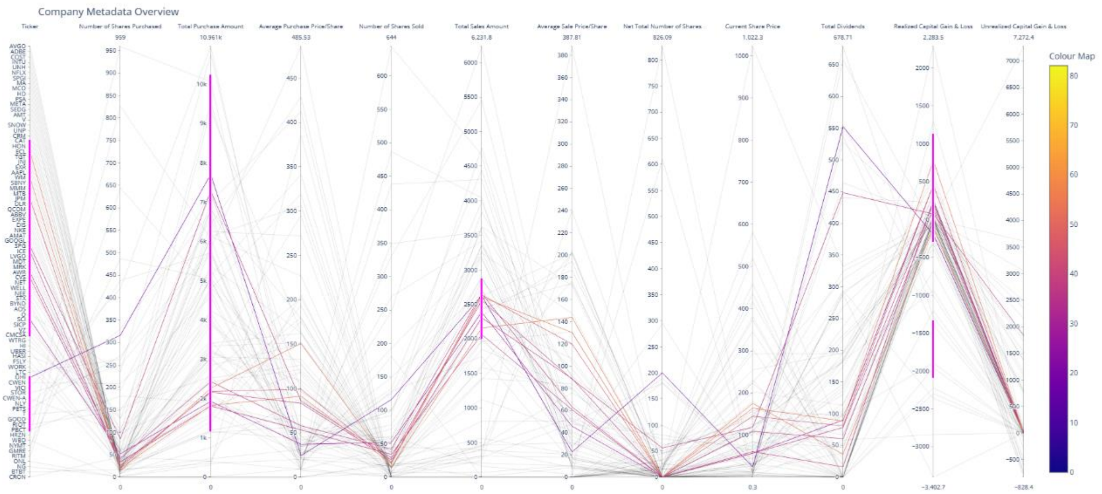


Figure 6.2.6.2: *The parallel coordinate chart with user brushing activities.*

Challenges were encountered when creating the ‘Ticker’ axis. This is because parallel coordinate plots typically require numerical data for plotting, but tickers are string characters. To include the tickers in a format compatible with the parallel coordinate plot, a unique numerical index is assigned to each ticker. This conversion is crucial as it allows the categorical data to be treated numerically. The chart is adjusted to include a 'Ticker' axis, with the values derived from the 'Ticker Index' and the range set from the min to the max of the unique ticker indices. This configuration uses the unique indices as the actual values plotted on the axis, while the corresponding tickers are displayed as labels. The ‘ticktext’ variable is used to replace the numerical indices with the actual ticker symbols on the axis labels, thus maintaining readability and relevance of the data displayed, implemented through the codes shown in Figure 6.2.6.3.

```python
# Adjusting dimensions to include 'Ticker' as the label with the values from 'Ticker Index'
dimensions = [
    dict(range=[min(unique_ticker_indices), max(unique_ticker_indices)],
         tickvals=list(df['Ticker Index'].unique()),
         ticktext=list(df['Ticker'].unique()),  # Keeping the label as 'Ticker'
         label='Ticker', values=df['Ticker Index']),
    dict(label='Number of Shares Purchased', values=df['Total Number of Shares Purchased']),
    dict(label='Total Purchase Amount', values=df['Total Purchase Amount']),
    dict(label='Average Purchase Price/Share', values=df['Average Price per Share']),
    dict(label='Number of Shares Sold', values=df['Total Number of Shares Sold']),
    dict(label='Total Sales Amount', values=df['Total Sales Amount']),
    dict(label='Average Sale Price/Share', values=df['Average Sale Price per Share']),
    dict(label='Net Total Number of Shares', values=df['Net Total Number of Shares']),
    dict(label='Current Share Price', values=df['Current Share Price']),
    dict(label='Total Dividends', values=df['Total Dividends']),
    dict(label='Realized Capital Gain & Loss', values=df['Realized Capital Gain & Loss']),
    dict(label='Unrealized Capital Gain & Loss', values=df['Unrealized Capital Gain & Loss']),
]
```

Figure 6.2.6.3: *The codes which address the challenges of incorporating categorical data into the parallel coordinate chart.*

6.2.7 Multiple Line Chart

The Multiple Line Chart in the system is a critical visualization for analysing the performance trends of multiple stocks simultaneously. This implementation allows users to visualize historical price data, buy and sell points, and various moving averages for a selection of stocks, providing a rich, layered analytical perspective. The multiple line chart is constructed using 'plotly.graph_objects', which is adept at handling complex data visualisations, enabling the detailed configuration of line properties, such as colour, thickness, and style for each stock, ensuring that each line is distinct and informative.

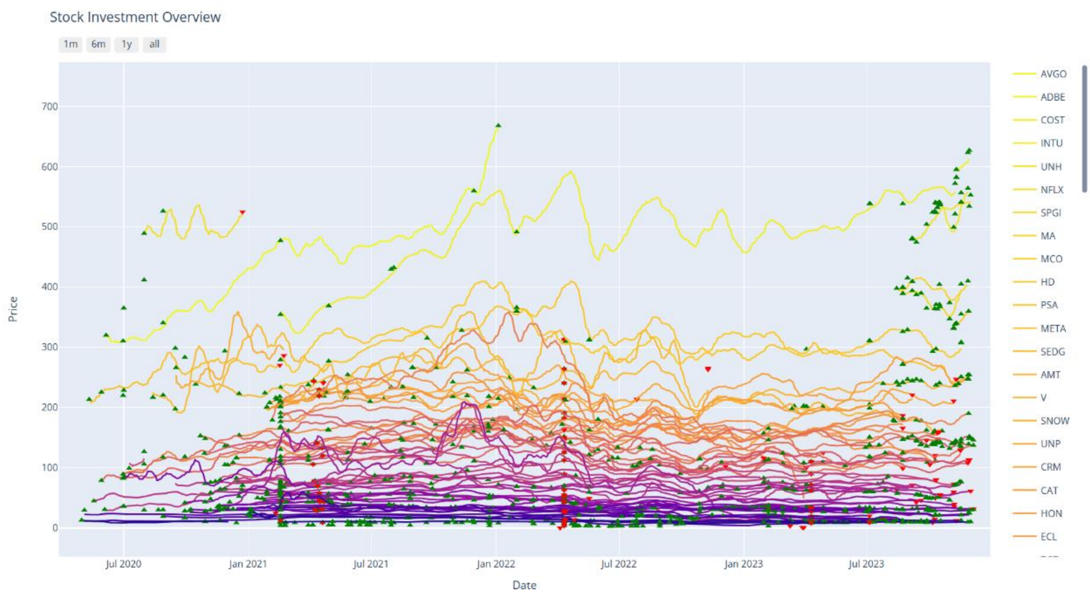


Figure 6.2.7.1: *The Multiple Line Chart showing the price history of all stocks in the Investment Transaction Dataset with Buy and Sell actions.*

Each stock in the dataset is represented as a separate line on the chart, allowing users to compare performance directly across different securities. Buy and Sell are highlighted on the stock lines using different markers, which help users to compare the investment decisions. Callbacks are used to manage real-time updates to the chart, accommodating user interactions such as selecting different stocks, adjusting time frames, or changing moving average settings, which will be further detailed in section 6.2.10.

In the implementation of this chart, significant emphasis is placed on the integration and preprocessing of datasets to ensure the visualization is accurate and insightful. To fetch the real-time stock data, the initial approach is using Yahoo Finance. However, during the implementation phase, it was observed that numerous data points representing buy and sell prices did not fall between the highest and lowest trading prices provided by Yahoo Finance. This discrepancy raised concerns about the accuracy and reliability of the data, so the data sourcing was shifted to Google Finance. Once the data is retrieved, it is processed for usability in the chart using the 'Pandas' library. This includes cleaning, normalization, and calculation of moving averages which are vital for trend analysis.

6.2.8 Home View

The Home View displays a dashboard (Figure6.1) that offers a comprehensive view for investment analysis, including the Multiple Stock Line Chart, Parallel Coordinate Chart, Gain/Loss Stacked Bar Chart, Buy/Sell Stacked Bar Chart and two Dividend Stacked Bar Charts, providing a quick glance at market trends, top performers, and other key metrics. The layout of the charts is effectively managed through 'HTML', ensuring a clear and user-friendly interface. The containers are set up for each chart, allocating them a specified location on the dashboard as illustrated in Figure 6.2.8.1.

```
html.Div([  # Sub-container for the Gain/Loss chart
    html.Div(style={'height': '200px'}),
    dcc.Graph(id='gain-loss-chart', figure=gain_loss_fig)
], style={'display': 'inline-block', 'width': '20%'}),

html.Div([  # Sub-container for Dividend and Buy/Sell charts
    html.Div(style={'height': '220px'}),
    dcc.Graph(id='buy_sell_fig-home', figure=buy_sell_fig),
    dcc.Graph(id='divi-time-home', figure=dividend_time_fig),
    dcc.Graph(id='divi-ticker-home', figure=dividend_ticker_fig)
], style={'display': 'inline-block', 'width': '30%'}),
```

Figure 6.2.8.1: *Part of the codes creating the sub-containers for the charts.*

By strategically placing interactive elements and visualizations within a flexibly styled container, the dashboard offers a dynamic user experience. The use of inline styles with 'percentage widths' ensures that the dashboard remains responsive and visually balanced across different screen sizes. Callbacks integrated in this view such as linked brushing between charts will be further detailed in the enhancement section. The 'dcc.Store' is included for holding user selections or computed data that needs to be shared across multiple components of the dashboard, enhancing the reactivity and efficiency of the dashboard without repetitive recalculations.

6.2.9 Colour Maps

Colour mapping is a crucial aspect of financial visualization as it helps to enhance the readability and interpretability of complex data by distinguishing different data elements visually, which helps to avoid confusion, reduce cognitive load, and enable quicker decision-making based on visual cues. The 'plasma_color_scale' (Figure 6.2.9.1) from the library 'plotly.express' was chosen to assign a unique colour to each ticker in this system. This is achieved by calculating equally spaced intervals on the colour scale based on the number of tickers.

This colour map was chosen because each ticker can be displayed distinctly, and it is especially clear for the multiple line chart under a light blue background. In addition, the buy and sell points marked in green and red can be shown on all lines clearly under this colour mapping, as these two colours are not appearing on 'Plasma'. Moreover, unlike other colour mappings such as 'HSV', which can also show distinct colours in a cyclic style, 'Plasma' is perceptually uniform sequential. This facilitates the opportunity to apply this colour scale to each ticker according to their average prices over the period covered in the investment transaction dataset in an ascending order. The colours could correspond to price levels, where the darkest blue indicates the lowest average price and the lightest yellow represents the highest. This contributes to a clear view of the Multiple Line Chart, where showing all lines clearly appeared to be a challenge. The tickers in the Parallel Coordinate Chart are also ordered to match their positions on the colour scale, ensuring a coherent and intuitive visual experience.

Figure 6.2.9.1: *The 'Plasma' Colour Map utilised in the system for multiple charts.*

6.2.10 User Options

Apart from selection and filtering options within each chart on the home page, two user options are integrated to update the multiple line chart to show different views. The first user option allows users to choose whether to show the price trend after the last sell action or the last buy action, which is implemented through two different ticker boxes. The second option is implemented through an input box for users to specify the value of the moving average for the chart, allowing them to smooth out price fluctuations over their desired timeframe. Showing neither of the trend after the last sell or buy action and the moving average of 10 days are set as default, as shown in Figure 6.2.10.1.

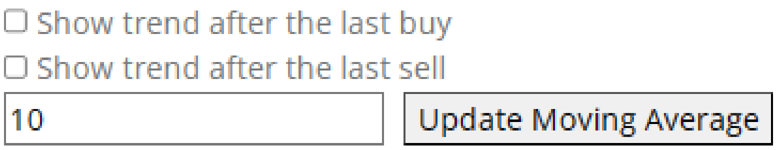


Figure 6.2.10.1: *The ticker boxes and the input box showing the default values.*

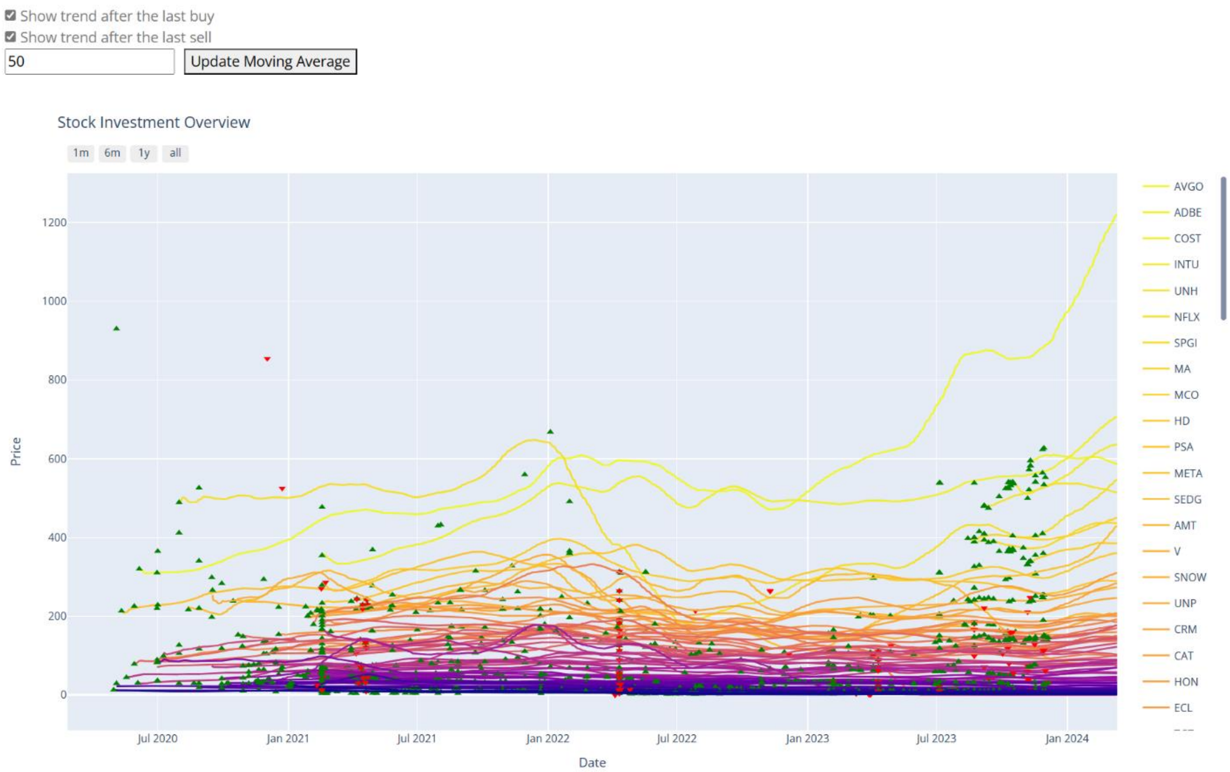


Figure 6.2.10.2: *A screenshot of the Multiple line chart when both ticker box are ticked and the moving average set to 50, presenting the lines more smoothly and showing the trends after the last transaction action in the Investment Transaction Dataset.*

The ticker boxes are incorporated using Dash's ‘dcc.Checklist’ component. When a user selects one of these options, Dash callbacks trigger an update to the multiple line chart to reflect the selected trend. The backend logic adjusts the data processing to filter and display the stock prices post the chosen transaction type. The input box is implemented with Dash’s ‘dcc.Input’ where users can enter a numeric value. When the button “Update Moving Average” is clicked, a Dash callback recalculates the moving averages based on the user-defined period and updates the chart accordingly.

6.3 Enhancements

This section details the implementation for enhancements in this system. Enhancements in different views will be discussed in the following sections. Section 6.3.1 introduces the Toggle View functionality within the Dividend view. Section 6.3.2 introduces four chart types for users to select in the Single Stock View. Section 6.3.3 is about grouping the legend in the Multiple Line Chart for the stock trend lines and their corresponding transaction markers. Section 6.3.4 details the Linked Brushing functionality between the Parallel Coordinate Chart and the Multiple Line Chart in the Home View. Section 6.3.5 and 6.3.6 introduces the Legend Interactions and Interactive Tooltips implemented in all charts. Section 6.3.7 discusses the techniques that ensure the system could dynamically adapt to different screen sizes.

6.3.1 Toggle Views in Dividend View

The Dividend Over Time Chart introduces a "Toggle View" button. This feature allows users to switch between two distinct modes of visualizing dividend data, catering to different analytical needs and preferences for clarity or detail. The simplified mode concentrates on presenting only the total dividend payments by company without distinguishing between different types of dividends. This mode offers a cleaner, more straightforward visualization, ideal for quick overviews, which is set as default as shown in Figure 6.2.4.2. The detailed mode which is illustrated in Figure 6.3.1.1 displays each company's dividend payments broken down by different types of dividends. This mode is designed for users who require a comprehensive analysis, detailing how various forms of dividends contribute to the overall dividend strategy of each company over time.

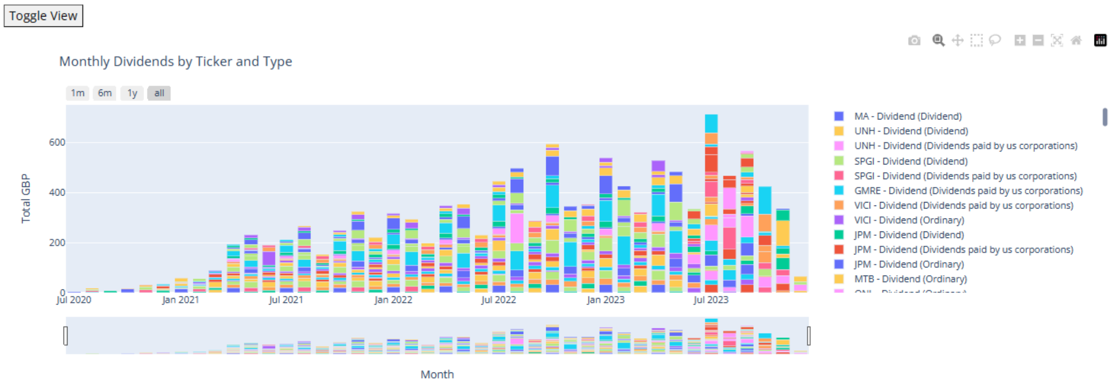


Figure 6.3.1.1: *A screenshot of the detailed mode when "Toggle View" button is clicked.*

The toggle functionality is implemented using a button triggered by click to switch between the Detailed and Simplified views through Dash callbacks. This control provides an intuitive way for users to choose their preferred data presentation style directly. The detailed mode is also ideal for users who would like to see the trend of specific types of dividends of a single stock over time, through selecting the related tickers as the example demonstrated in Figure 6.3.1.2.

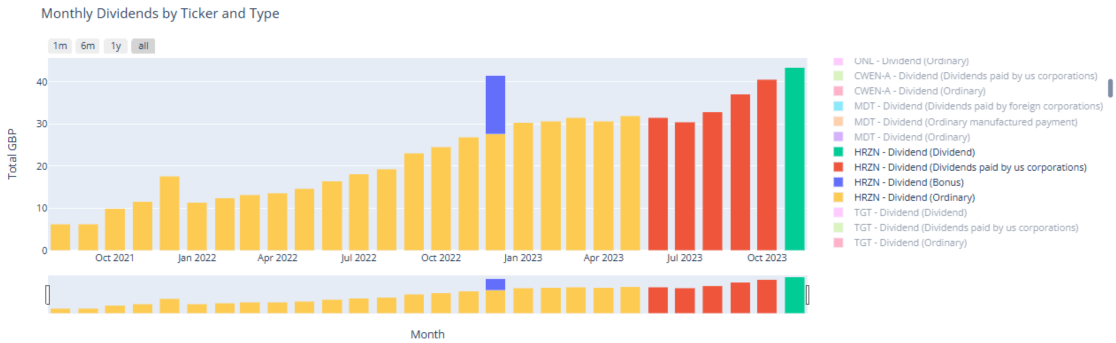


Figure 6.3.1.2: *The dividend trend of the ticker "HRZN" detailed by the dividend type.*

6.3.2 Chart Types in Single Stock View

The Single Stock View within the system has been enhanced by including a variety of chart types, allowing users to choose how they visualize stock price history according to their analysis preferences. Multiple options are provided through a selection box, including line charts (Figure 6.2.5.1), candlestick charts (Figure 6.3.2.1), OHLC (Open, High, Low, Close) charts (Figure 6.3.2.2), and area charts (Figure 6.3.2.3). The line chart is set as the default view, which provides a familiar and straightforward representation of stock price movements over time.

Figure 6.3.2.1: *The candlestick chart of ticker "BYND" with buys and sells markers.*

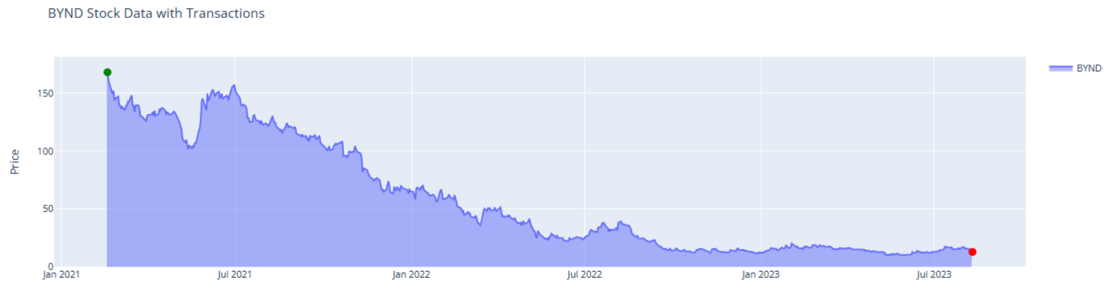


Figure 6.3.2.2: *The area chart of ticker "BYND" with buys and sells markers.*

Figure 6.3.2.3: *The OHLC chart of ticker "SNOW" with buys and sells markers.*

A selection box (Figure 6.3.2.4) is integrated into the Single Stock View, implemented using Dash's 'dcc.Dropdown' component. This dropdown menu lists the different chart types available, enabling users to select their preferred visualization method. Based on the user's selection, the backend logic dynamically updates the chart display. This is handled through a Dash callback that takes the selected chart type as input and outputs the corresponding chart for display.

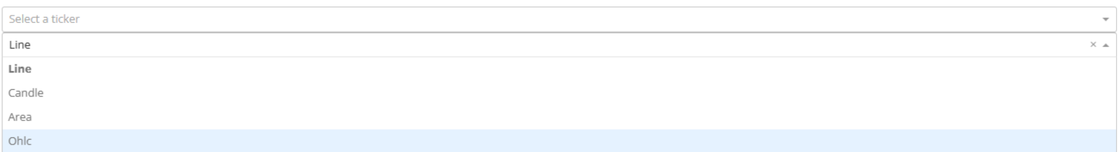


Figure 6.3.2.4: *The selection box for chart styles in the Single Stock View.*

6.3.3 Grouped Legends in Multiple Line Chart

Incorporating grouped legends for the Multiple Line Chart is especially crucial when visualizing stock price trend lines along with transaction markers like buy and sell points marked on them. This approach significantly enhances the clarity and interpretability of the chart by logically organizing related data series. In the implementation within the 'multiple.py', each stock ticker's price trend line and its associated buy and sell markers are grouped under the same legend entry. This grouping ensures that all the related series, including price data, buy points, and sell points are intuitively connected in the user interface, which react in the same way when the user interacts with the entire group. For example, when users select several tickers to focus on, their corresponding lines and transaction points are all being focused automatically, allowing users to interact with the legend to toggle the visibility of grouped elements together.

6.3.4 Linked Brushings in Home View

The functionality of linked brushing between the Multiple Line Chart and the Parallel Coordinate Chart on the dashboard is a sophisticated implementation that enhances data interactivity and coherence across different visualizations. This functionality adjusts the visibility and focus of corresponding data in the multiple line chart automatically when user dynamically focus on selected stocks in the parallel coordinate chart. As illustrated in Figure 6.3.4.1, when the stocks are selected in the parallel coordinate chart, their corresponding lines in the multiple line chart become more highlighted, while other lines become transparent to a certain degree. This visual distinction helps users to correlate data across charts without losing track of the broader data context. This linked brushing interaction not only enriches the user experience but also aids in detailed stock analysis by isolating noise from other less relevant data points.

The implementation of this functionality was one of the most challenging parts of this project. Typically, 'selectedData' in Plotly Callback is used to capture user selections, but with parallel coordinate charts, this property does not provide the necessary data, complicating the tracking of user-selected ranges on axes. This limitation necessitated an alternative method to effectively monitor and capture user interactions. To overcome this obstacle, the 'restyleData' property of the chart was utilized.

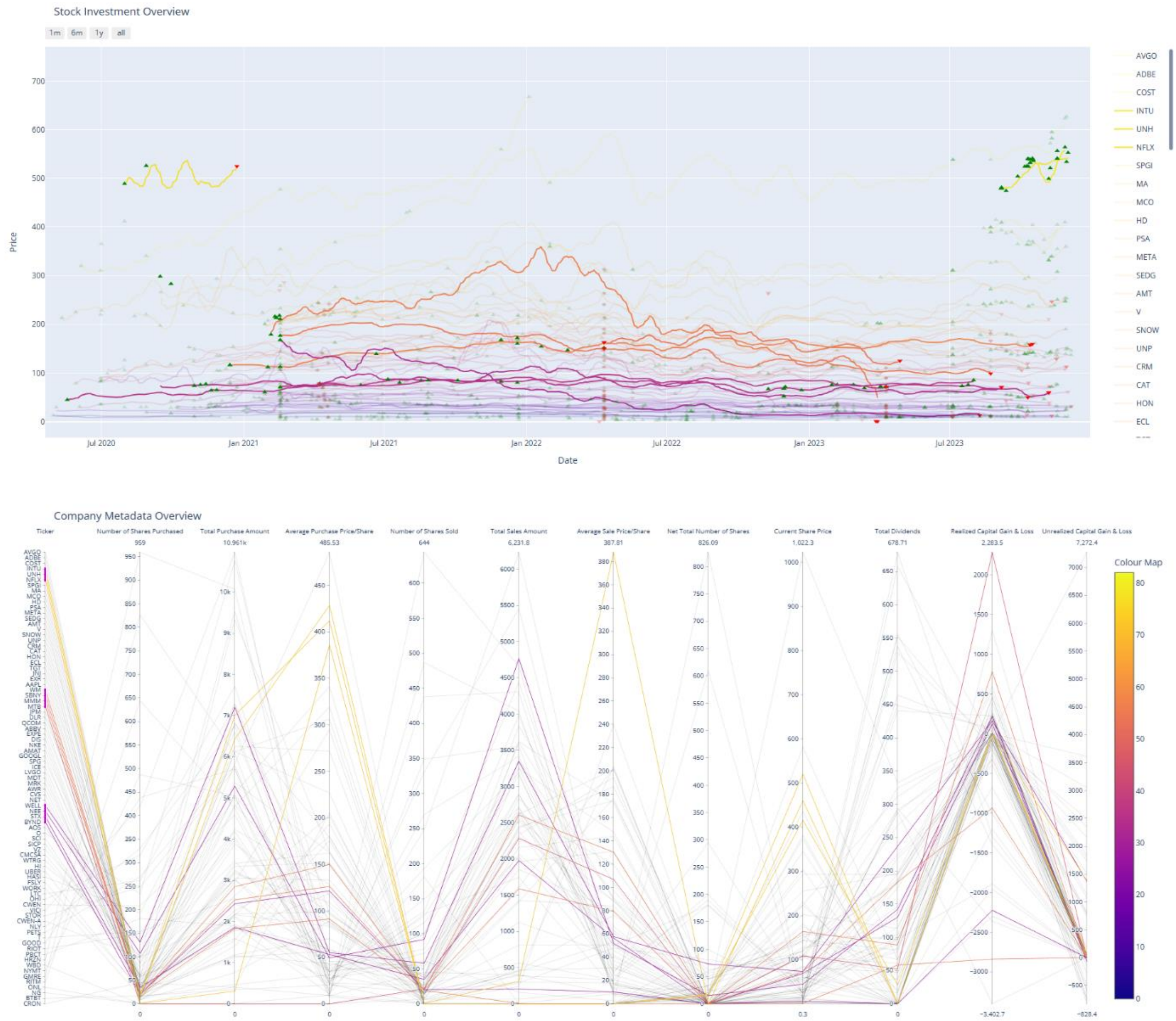


Figure 6.3.4.1: *A screenshot illustrating the linked brushing functionality between the Parallel Coordinate Chart and the Multiple Line Chart. Same stocks are focused on both charts, with the "Focus and Context" technique implemented.*

The 'restyleData' property captures low-level style changes, which indirectly include user interactions such as axis range selections. It provides a series of changes in a format that includes the axis index and the selected range. These data points are decoded to update a dictionary that tracks the current user selections by axis. If a selection is cleared, it updates the dictionary as well, ensuring that the selections are always current. The codes for fetching the user selected data are illustrated in Figure 6.3.4.2. After decoding the selections, the data is further processed to filter the indices of companies based on the selected ranges. Filters for different axis are iteratively applied to the company data and updates a set of valid indices. By dynamically focusing on user-selected stocks across charts, the tool provides a powerful means for focused financial analysis.

```
@callback(
    Output('user-selections-store', 'data'),
    [Input('risk-home-chart', 'restyleData')],
    State('user-selections-store', 'data')
)
def update_user_selections(restyle_data, existing_selections):
    if restyle_data:
        # Initialize a new selections dict if none exists
        if not existing_selections:
            existing_selections = {col: None for col in df_columns}
        # Convert existing selections from JSON
        if isinstance(existing_selections, str):
            existing_selections = json.loads(existing_selections)
        # Update the selections based on restyleData
        for change in restyle_data:
            if 'dimensions' in str(change):
                for key, value in change.items():
                    # Extract the dimension index
                    dim_index = int(key.split("[")[1].split("]")[0])
                    dimension_name = df_columns[dim_index]
                    if value:  # If there are selections
                        if not isinstance(value[0][0], list):
                            existing_selections[dimension_name] = [value[0]]  # Wrap single range in list of lists
                        else:
                            existing_selections[dimension_name] = value[0]
                    else:  # If the selection is cleared
                        existing_selections[dimension_name] = None
        # Convert updated selections back to JSON for storage
        return json.dumps(existing_selections)
    return existing_selections if existing_selections else json.dumps({col: None for col in df_columns})
```

Figure 6.3.4.2: *A screenshot of the codes for fetching the data selected by users on the Parallel Coordinate Chart.*

6.3.5 Interval Buttons in Time-Based Charts

For all Time-Based Charts, including the Multiple Line Chart, the Buy/Sell Transaction Chart, the Single Stock Charts and the Monthly Dividend Charts, two options for time filter were implemented. The first one is a time filter bar below each chart, which allows user to customise the period they want to investigate by dragging the filters bars. This is already implemented in basic developments. For enhancement, a more convenient way to filter and select time was implemented through the Interval Buttons, which are on the top left corner of each chart, with options including one month, six months, one year and all. An example of the Single Stock Line Chart when six months selected is shown in Figure 6.3.5.1. These buttons are configured in the x-axis settings and allow users to quickly adjust the time frame displayed on the chart with a more accurate value. The two filter features work together to create a user-friendly interface for navigating time-sensitive data, allowing users to effortlessly adjust their view to different time intervals. This makes it easier to analyse trends over specific periods without manually adjusting the range, thereby improving the efficiency of data exploration.

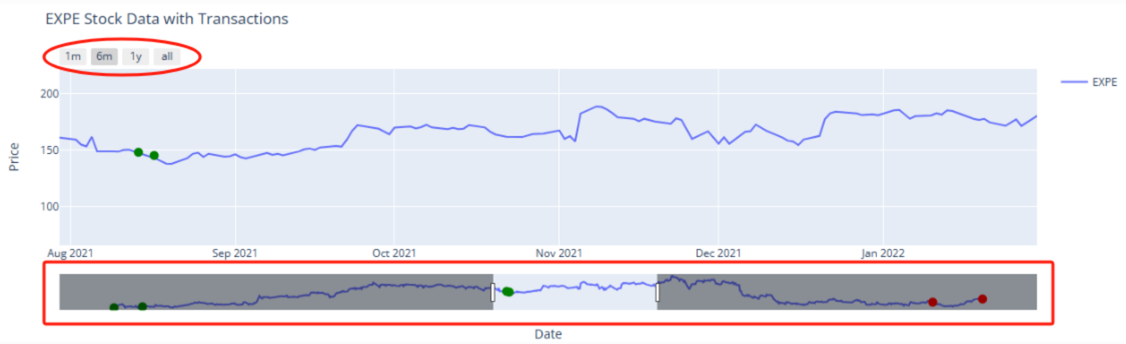


Figure 6.3.5.1: *A screenshot of Single Stock Line Chart when six months was selected.*

6.3.6 Legend Interactions in All Charts

Legend interactions are crucial for enhancing the usability and interactivity of charts in the entire system. These interactions allow users to control the visibility of various data series directly from the legend, helping to manage the complexity of visual information presented in multi-series charts. This functionality addressed one of the main challenges in the Multiple Line chart. By incorporating interactive legends that allow users to toggle specific lines on and off, users can focus on stocks of interest and effectively manage and display multiple stock data simultaneously without feeling overwhelmed. Users can also choose between directly toggle the lines off (Figure 6.3.6.1) through single click the legends on the Multiple Line Chart or set the unfocused lines to transparent (Figure 6.3.6.2) through selecting the tickers on PCP legend. Similar in the Single Stock View, users can also toggle the visibility of different data series such as moving averages. For Gain/Loss View, toggling individual legend entries allows users to focus exclusively on capital gains & losses or dividends.

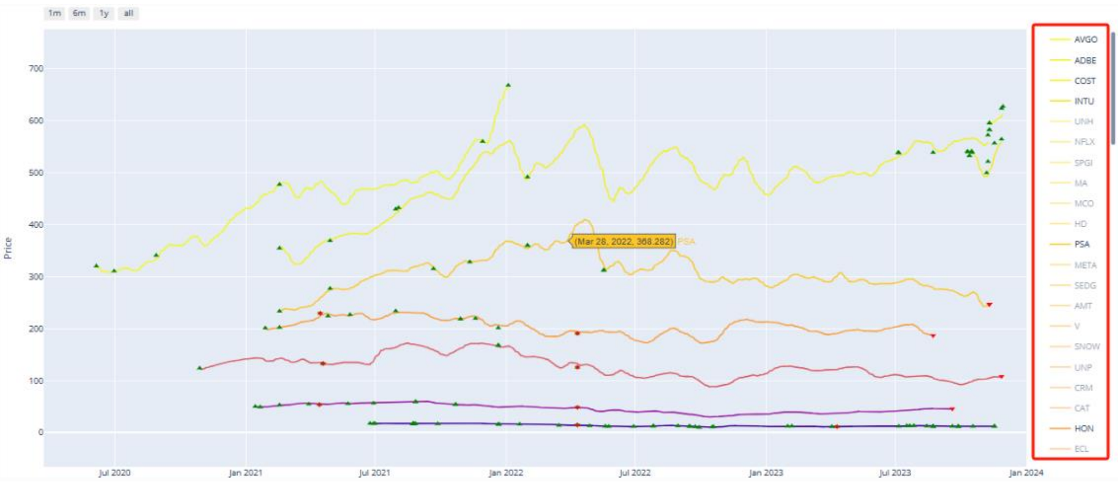


Figure 6.3.6.1: *A screenshot when unselected lines are toggled off.*

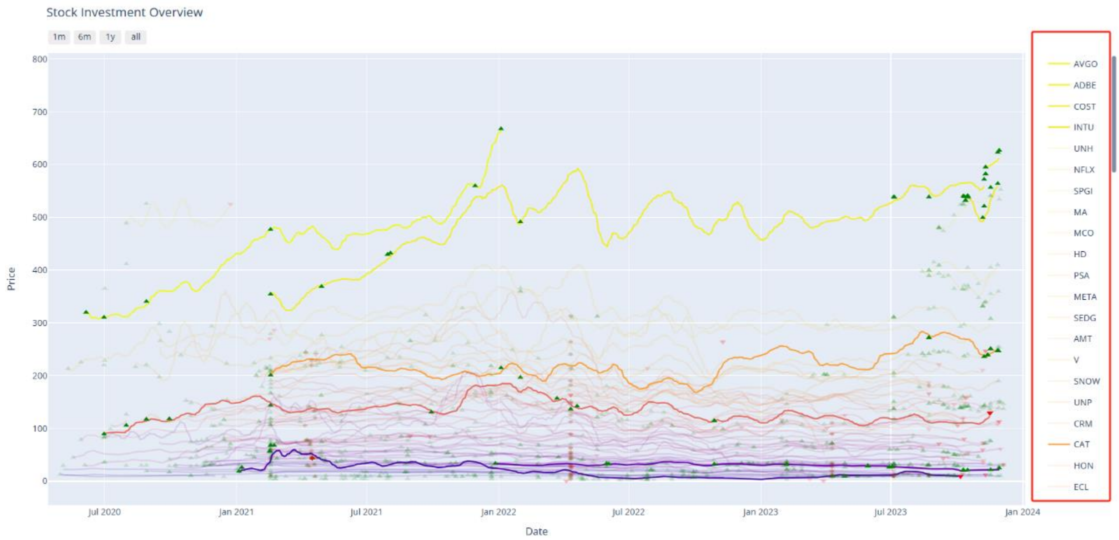


Figure 6.3.6.2: *A screenshot when unselected lines are set to transparent.*

In addition, double clicking on a legend item isolates this specific data series, and double clicking again returns the chart to its original state. This functionality is especially useful in Monthly Dividend Stacked Bar Charts when user want to investigate the dividend trend of a specific ticker. These legend interactions are applied to all charts. Legend controls are enabled by setting appropriate configuration options in the layout of the chart through 'Plotly'. This implementation not only enhances the data navigation experience within each chart but also supports a more focused and user-driven analysis, making complex datasets more accessible and easier to manage.

6.3.7 Zoom and Box Select in All Charts

The system also incorporates advanced navigation tools in all charts. Among the tools, the zooming and box selecting functionalities are most frequently used during visual analysis. They enable users to focus on specific areas of interest or to select multiple data points for deeper analysis. The Zoom tool allows users to magnify a particular segment of the chart, offering a closer look at the data. Users can select a specific portion of the chart and pan to navigate across different sections while remaining at the same zoom level. As illustrated in Figure 6.3.7.1, a portion of the Multiple Line Chart was selected, then the view toggles to Figure 6.3.7.2. In this view, detailed information such as the buy and sell points are clear and focused. The difference between the transaction price and the moving average price for that day is easy to identify through their positions. Users can also hover over the specific data points for detailed information.

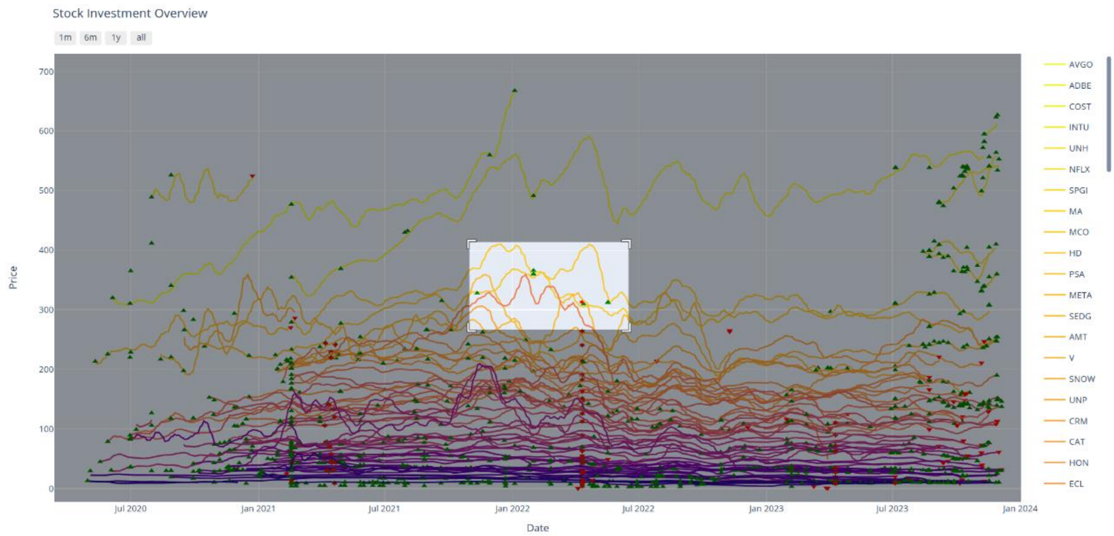


Figure 6.3.7.1: *A screenshot when zooming is applied to select a portion of the chart.*

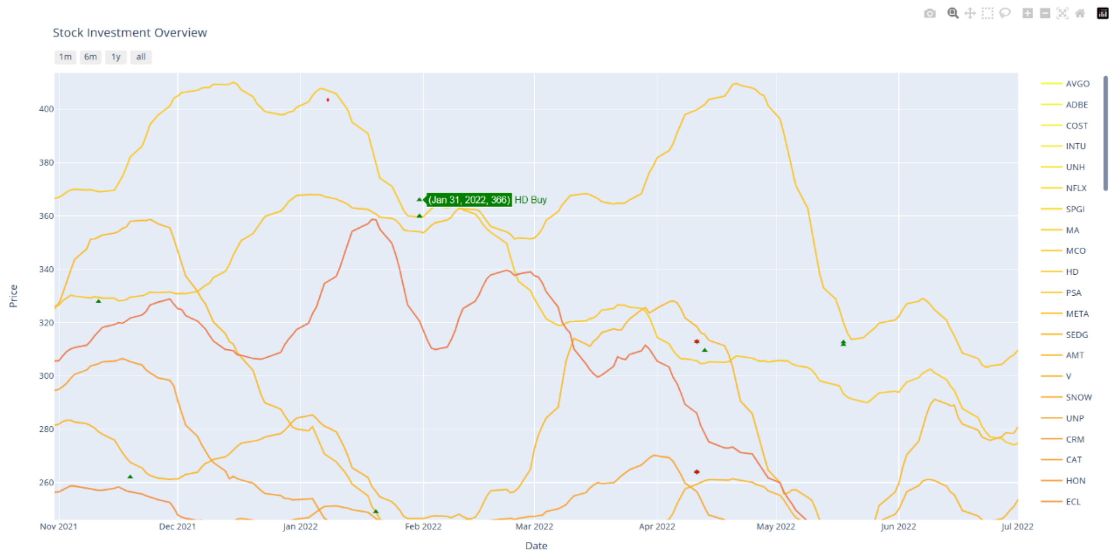


Figure 6.3.7.2: *A screenshot showing the detailed view of the portion after zooming.*

To accommodate detailed inspection, additional features like 'Zoom In and 'Zoom Out' are also included, which let users control the extent of zoom and quickly adjust the view according to their analytical needs. In addition to the zooming functionality, the box selection tool allows users to drag a rectangular selection box around a group of data points within the chart. The chart then highlights the selected points as shown in Figure 6.3.7.3. This is particularly useful for performing comparative analysis and focusing on clusters of data points to see how they behave within the context of the overall data set. Once a selection is made using 'Box Select', users can use other tools like 'Reset Axes' to focus solely on the selected area. Dynamic Dash callbacks are also integrated to update other components on the dashboard based on the selection.

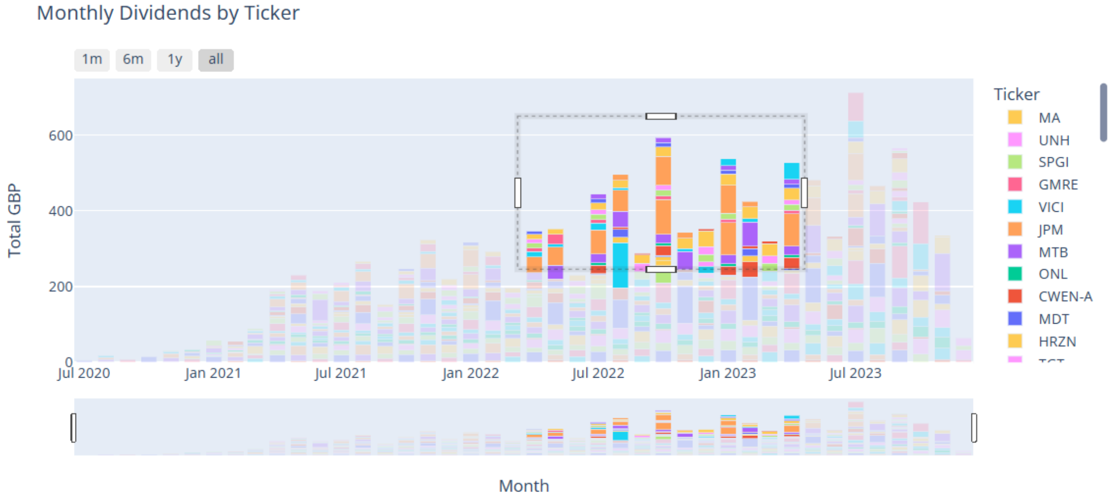


Figure 6.3.7.3: *A screenshot when box selection is used to focus on the*

6.3.8 Informative Tooltips in All Charts

The informative tooltips (Figure 6.3.8.1) were implemented in all charts, locating at their top right corner, which becomes visible when the user hovers over the chart area. This tooltip provides information related to all tools implemented in the chart, including Zoom, Pan, Box Select, Lasso Select, Auto Scale and Reset Axes.

Figure 6.3.8.1: *A screenshot of the tooltips on the top right corner of all charts.*

6.3.9 Adaptive Screen Resolution

To address the challenge posed by inconsistencies in chart displays across different devices, the adaptive screen resolution feature ensures that elements on the dashboard dynamically adjust to fit the viewing environment, maintaining usability and visual integrity no matter the device used. From a front-end perspective, 'Bootstrap' was employed to create a responsive layout that adapts to different screen sizes, ensuring accessibility across various devices. The dashboard UI components are designed with a combination of Bootstrap's grid system and custom 'CSS' for styling, achieving a balance between functionality and aesthetics.

7 Evaluation

This section provides a comprehensive analysis of the tool through a series of case studies designed to test its effectiveness in delivering personalized financial insights. By applying this tool to diverse scenarios, the evaluation confirms that the tool not only meets its intended design objectives but also excels in enhancing user decision-making through its dynamic and interactive visualizations. This rigorous testing process underscores the tool's potential to transform traditional investment analysis by providing deeper, more actionable insights tailored to individual investment profiles.

7.1 Results

This section outlines the results from a series of case studies designed to evaluate the effectiveness of this customer-centric visualization tool for analysing stock investment performance for anonymous investors. Throughout the implementation phase, each visual component and its interactive features were rigorously tested, ensuring their functionality and accuracy in the final presentation. These case studies illustrate how the tool supports investors by offering advanced visualization techniques and interactive analysis capabilities. It effectively demonstrates the ability to apply comparison between different stocks and assess whether the decisions of the investors are good or not.

Section 7.1.1 discusses Case Study A, which demonstrates the tool's ability to identify the abnormalities in the Investment Transaction Dataset. Section 7.1.2 discusses Case Study B, which conducted the comparison between the capital gains and losses and the dividend gains, with a particular focus on the tickers in the mREITs. Section 7.1.3 delves into the comparison on good and bad decisions made by the investors.

7.1.1 Case Study A: Identification of Abnormal Data

This case study focuses on the identification of abnormal data within the Investment Transaction Dataset. Abnormalities in AAPL and PBCT are identified through visual analysis. This showcases the tool's capability to detect and visually highlight outliers and anomalies in stock data, which are critical for investors monitoring for potential risks or unusual activity.

The abnormalities of AAPL and PBCT can be identified through Multiple Line Chart or the Single Stock Line Chart. The former provides an overview of obvious abnormal data points and the later provides a more detailed view. As shown in Figure 7.1.1.1, there are two data points that are significantly distant from any of the lines. The data points were identified as AAPL buy transactions through hover interactions on the related datapoints. It is also obvious to see that neither of the two lines nearest to the points are AAPL trend through colour mapped legend, which indicates that there definitely exist abnormalities in AAPL transaction data. Further detailed analysis is conducted through isolating this ticker in the single line chart, as shown in Figure 7.1.1.2.

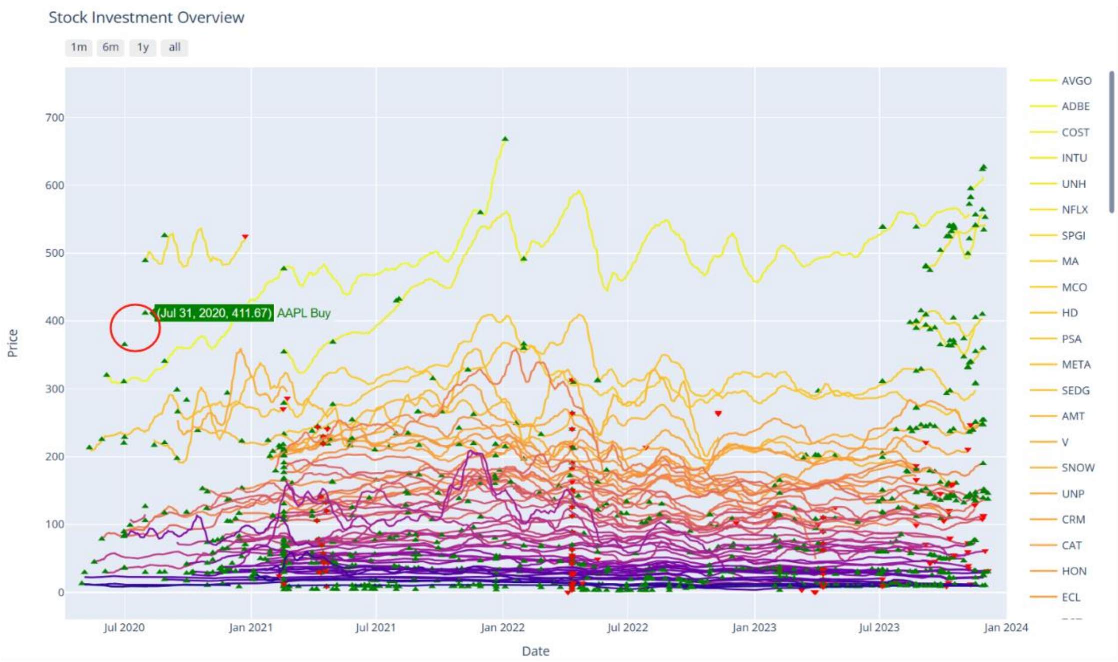


Figure 7.1.1.1: *A screenshot of the abnormal data identified in the multiple line chart.*

Figure 7.1.1.2: *A screenshot of the 'AAPL' trend with transaction markers.*

Through the isolated line chart of 'AAPL' with the transaction points, there are two abnormal buy transaction data points in total while the rest of them fits the trend well. Through further research on 'Apple' company events, conclusions can be made that this abnormality in data is due to the stock split. To be more specific, AAPL's 5th stock split took place on August 31, 2020. This was a 4 for 1 split, meaning for each share of AAPL owned pre-split, the shareholder now owned 4 shares. This explains why the purchase price before the split is much higher than the trend line, which appears to be abnormal in visual analysis. Similar abnormalities due to split can also be identified through the single line chart provided by this tool.

Through the single line chart, abnormal data was also identified in 'PBCT' sell point, as illustrated in Figure 7.1.1.3. While the other buy points align with the trend line well, there exists one sell marker showing that the stock was sell at £0, which is not possible. This indicates that there exist abnormalities in PBCT transaction data. Through further research on 'People's United Financial' company events, this abnormality is due to the corporate merger. To be more specific, on April 2, 2022, the People's United merged with M&T Bank and was fully integrated into M&T by the third quarter of 2022. The People's United Shares were converted to M&T bank shares, which seemed like PBCT was emptied at £0 but it actually remains in the M&T shares. Similar abnormalities due to merge can also be identified through this single line chart.

Figure 7.1.1.3: *A screenshot of the 'PBCT' trend with transaction markers.*

7.1.2 Case Study B: Comparison of Capital Gain/Loss vs Dividend

This case study delves into the performance of the stocks across the portfolio, with a particular focus on comparing the Capital Gain/Loss and Dividend Gain between tickers in mortgage Real Estate Investment Trusts (mREITs). The tickers included are NYMT, NLY, and RITM. This study aims to explore how fluctuations in U.S. mortgage rates impact the capital gains/losses and dividend yields of these entities, which are heavily invested in mortgage-backed securities. As mREITs generate income primarily through the interest accrued on these securities, their financial health is closely tied to national mortgage rate trends. The backdrop of rising mortgage rates over recent years sets the stage for this analysis. With rates climbing sharply, the potential for decreased housing affordability and increased default rates on mortgages could influence the valuation and performance of mREITs. This case study highlights the risk-return profile unique to mREITs in a changing economic landscape. Through interactive charts and detailed visual analysis, whether the higher yields from dividends compensate for the possible capital losses experienced by investors during periods of rate hikes will be assessed.

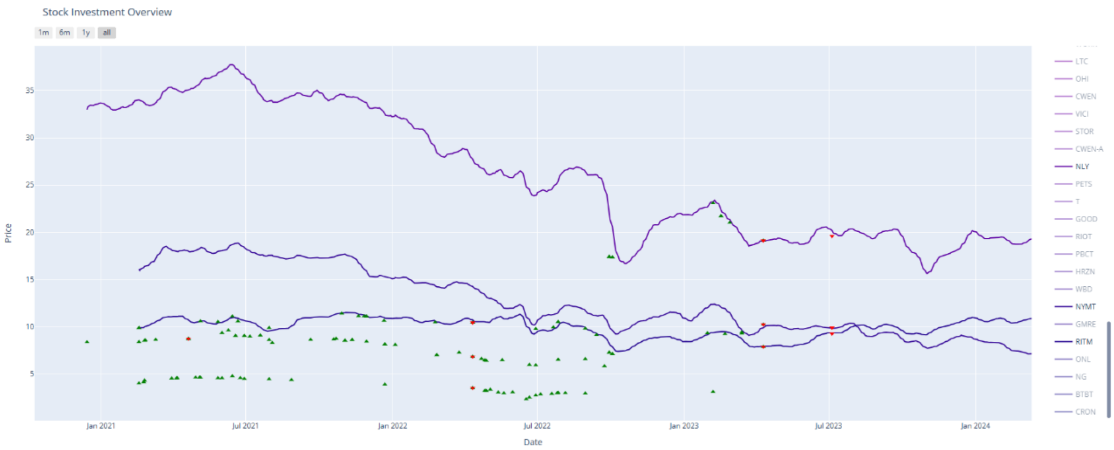


Figure 7.1.2.1: *The Multiple Line Chart with mREITs tickers isolated with the buy and sell points. The inconsistency of the data points is due to the reverse splits of the stocks.*

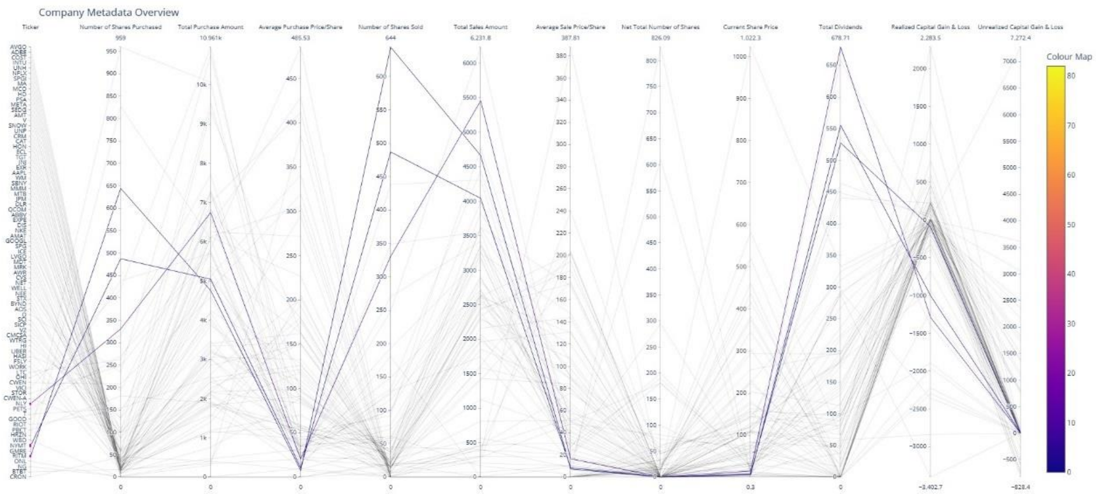


Figure 7.1.2.2: *The Parallel Coordinate Chart with focus on the mREITs tickers.*

The Multiple Line Chart illustrated in Figure 7.1.2.1 and the Parallel Coordinate Chart in Figure 7.1.2.2 with brushing together provides a comprehensive overview of the investment actions and price trends of the mREITs tickers. The 1 for 4 reverse splits happened to NLY and NYMT took place on September 26, 2022, and March 09, 2023, respectively. This reduces the number of shares and increases the market price per share.

From these visualisations, basic investment actions and performance of the mREIT stocks can be analysed. For **NLY**, initial shares were purchased on December 14, 2020, at a price of £8.35 per share. By April 10, 2023, when the last sell happened, the price was £19.1 per share, increased due to reverse split. However, the average purchase price per share over all transactions was slightly higher at £20.41 per share, in contrast to the current market price of £19.28 on March 11, 2024, and is also significantly higher than the average sell price at £16.50. This discrepancy has led to a capital loss of £1292.26 across 330.5 shares, highlighting a substantial decline in price per share.

For **NYMT**, shares were first purchased on February 17, 2021, for £3.99 each. The price also appeared higher at £9.88 due to the reverse split on July 5, 2023, when the last sell happened. However, the average purchase price over the investment period was £10.36 per share, which was higher than the average sale price at £8.3 and the current price at £7.12 per share by March 11, 2024. This has resulted in a total capital loss of £1004.37 for 487 shares, indicating a negative impact on the overall investment value.

For **RITM**, the stock was initially purchased on February 17, 2021, at £9.85 each, with subsequent purchases around similar price with fluctuations between a relatively small range. When considering all transactions, the average price per share comes to £7.39, slightly above the average sale price at £7.24. This has led to a modest capital loss of £95.67 for 644 shares, reflecting a slight underperformance in the investment.

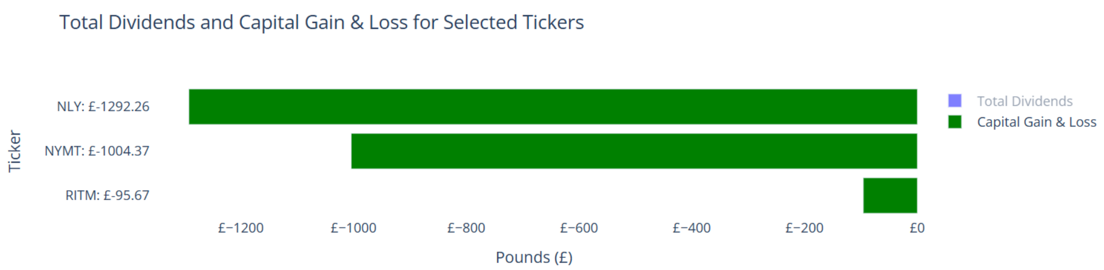


Figure 7.1.2.3: *The Capital Gain/Loss of the mREIT tickers.*

Regarding the Capital Gain and Loss of the three mREIT tickers, all tickers end up in high negative values, indicating the price of these tickers dropped significantly due to the environment. Among the stocks, the NLY shows the greatest loss, at the value of -£1292.26, followed by NYMT at -£1004.37. RITM was the best among them, which only end up in a loss of £95.67.

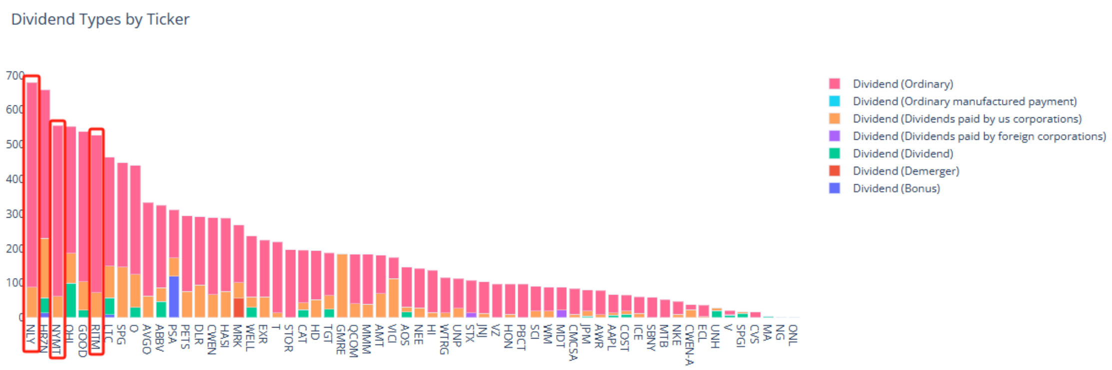


Figure 7.1.2.4: *The total dividend of tickers with mREITs highlighted in res boxes.*

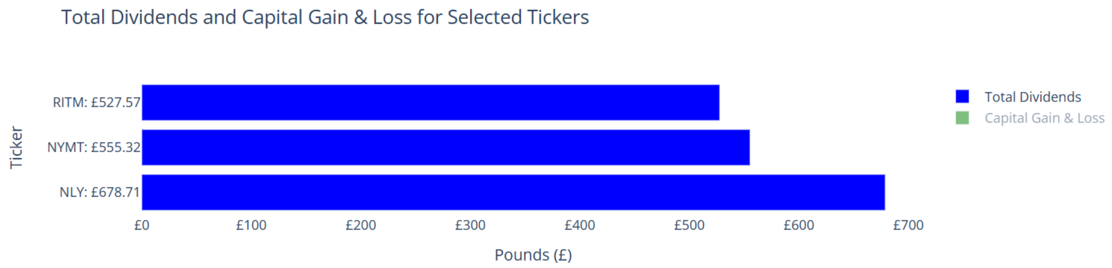


Figure 7.1.2.5: *The total dividend value of mREIT stocks.*

The dividend view for all stocks in Figure 7.1.2.4 shows that all three mREITs contribute to the top five among all tickers regarding the sum of dividends. Further detailed in Figure 7.1.2.5, NLY leads in dividend value at £678.71 among the three tickers, while NYMT and RITM allocated dividends of £555.32 and £527.57, respectively. This visual analysis illustrates the dividend performance for each stock, indicating that NLY yields the highest dividends, followed by NYMT, and RITM at a slightly lower value.

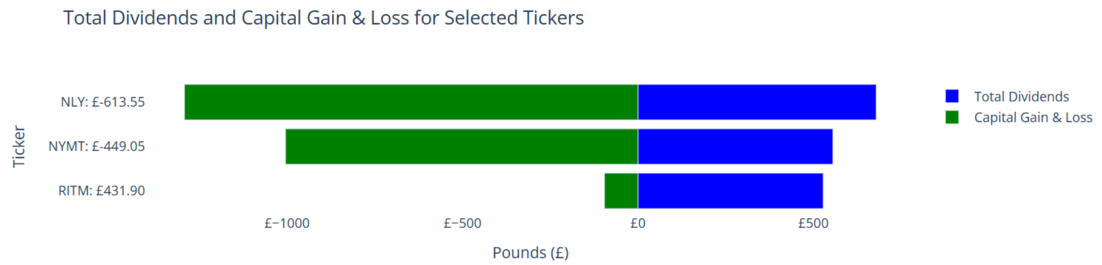


Figure 7.1.2.6: *The Sum of the Capital Gain/Loss and Total Dividends.*

However, regarding overall performance, as illustrated in Figure 7.1.2.6, RITM is the only stock ends up in positive values, indicating that the dividends of NLY and NYMT are unable to cover their capital losses. The net loss of NLY and NYMT are £613.55 and £449.05, while RITM results a net gain of £431.90.

7.1.3 Case Study C: Comparison of Good vs Bad Decisions

This case study delves into a strategic evaluation of buying and selling decisions within the stock market, distinguishing between successful trades and less optimal ones in the Investment Transaction Dataset. Employing the interactive data visualization provided by this tool, this examination scrutinizes each trade to see if actions taken by investors align with advantageous market trends as indicated by the moving averages. The buys executed below the moving average and sells above it are hypothesized to be favourable, potentially leading to gains, while trades that do not align with these benchmarks are subject to scrutiny as potentially less favourable decisions. This comparative analysis also illustrates the critical role of moving averages as benchmarks for good trading practices and to evaluate the strategic timing of entry and exit points in the stock market.

This analysis can be conducted through the Multiple Line Charts for obvious overview and the Single Stock Line Chart for details. For example, as illustrated in Figure 7.1.3.1, it is obvious that the Sale of SEDG at £265.01 made on Nov 9, 2022, was a very good decision. This is because the selling point is significantly higher than the 10 days moving average of that day, which is £228.96, securing a significant profit for the investor. This view provides an intuitive and clear visual for determining the success of buy and sell decisions, illustrating how trades that significantly diverge from the moving average trend line can result in pronounced capital gains or avert potential losses.

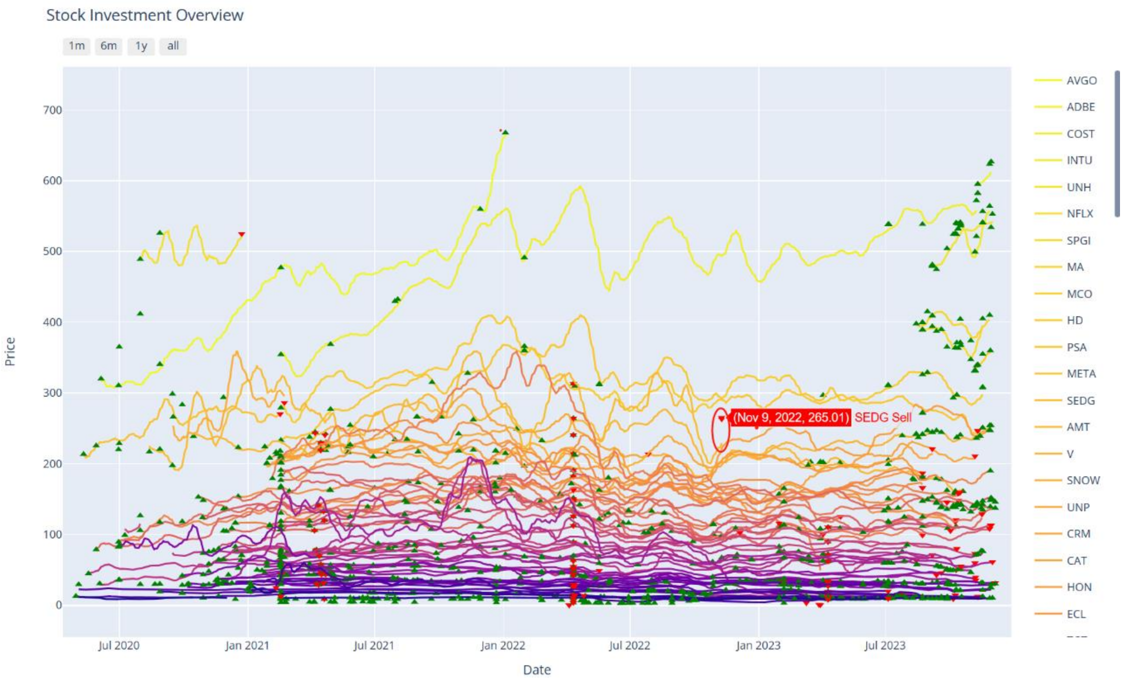


Figure 7.1.3.1: *The 'SEDG' Sell point on the Multiple Line Chart.*

Similar comparations can also be made in the Single Line Chart, as illustrated in Figure 7.1.3.2. The buy action done on April 11, 2022, at the price of £240.36 was significantly lower than the 10 days moving average, which is £256.62, making it a potentially profitable purchase. However, as the price trend goes down sharply after this buy action, it cannot be considered as a good decision anymore. This indicates that moving average should not be the only criterion for assessing the decisions, other related indicators such as the price trend should also be taken into account.

As a result, in addition to assessing good buy actions based on the moving averages, the buy actions done before a significant increase in a specific stock can also be considered as good decisions. As illustrated in Figure 7.1.3.3, the stock 'AVGO' experienced a tremendous increase in stock price after the buy actions, which reached the price of £1338.92 on March 11, 2024, which was significantly higher than the average purchase price at £295.01, contributing to a potential profit of £6785.2.

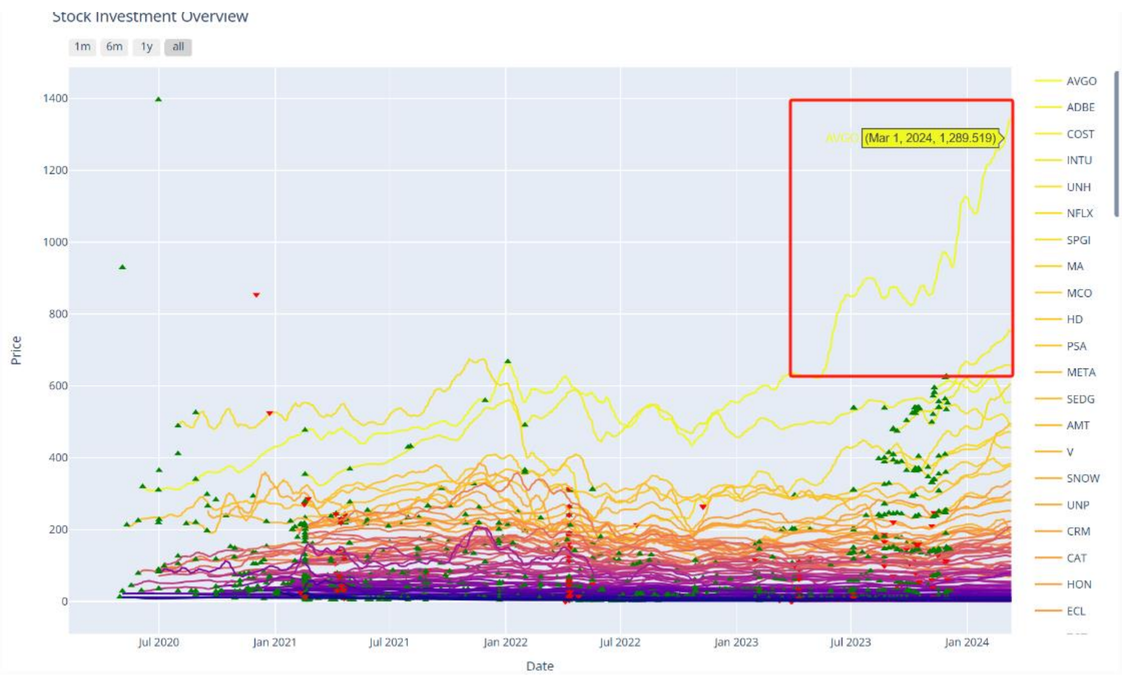


Figure 7.1.3.3: *A screenshot of the Multiple Line Chart indicating the significant increase in stock price of ticker 'AVGO'.*

The Figure 7.1.3.4 illustrates the overall performance of all stocks within the portfolio, aiding in the comparison to determine whether the invested stocks are profitable.

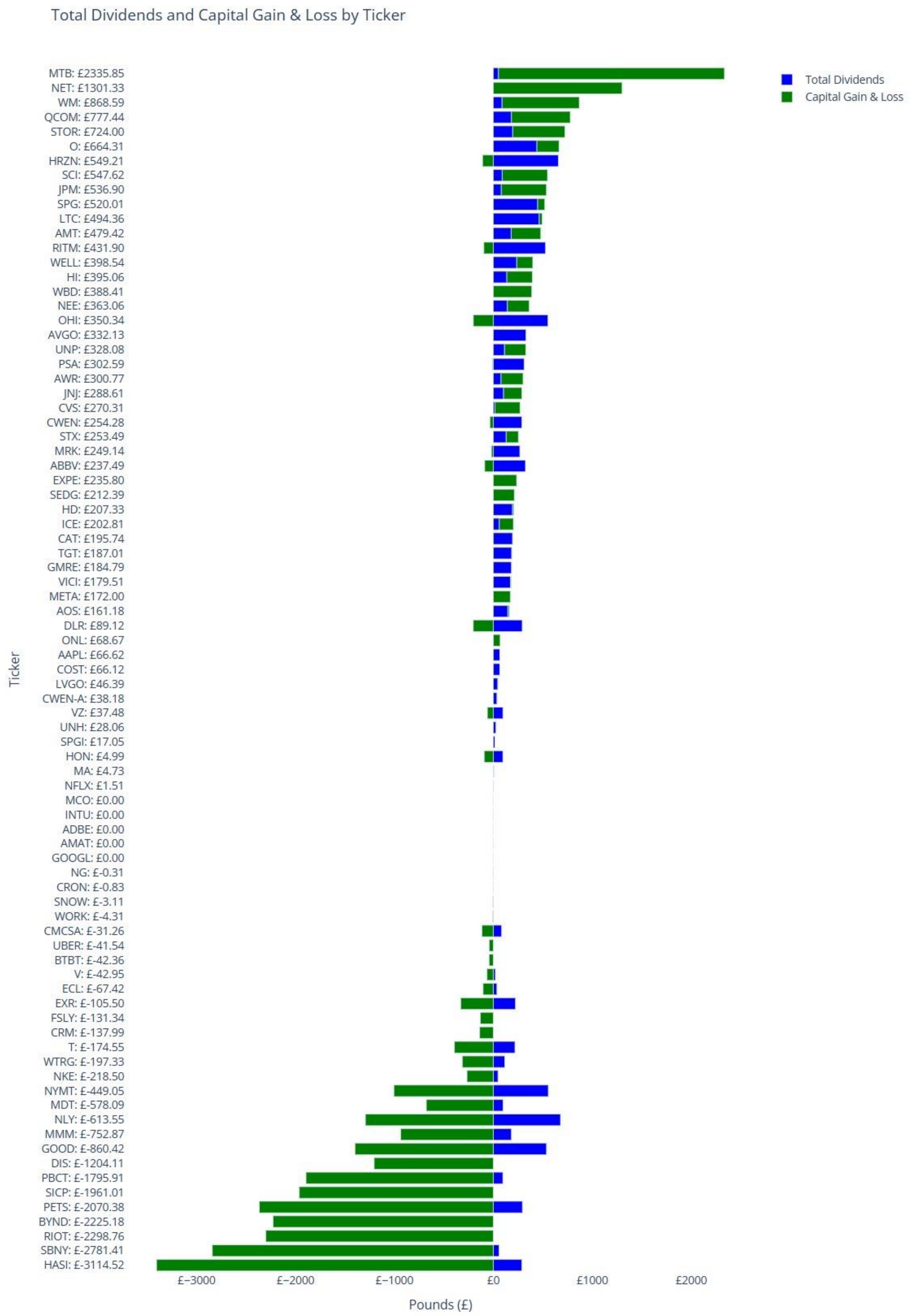


Figure 7.1.3.4: *A screenshot of the investment performance of 83 stocks, MTB leads in a net gain at £2335.85, while HASI results in the largest net loss of £3114.52.*

8 Conclusion

This thesis and the development works have successfully produced a customer-centric visualization tool for stock investments that transforms complex financial data into actionable insights tailored to individual investors. This innovative tool stands out from established systems such as Yahoo Finance by focusing on personalized data integration and analysis, thus enhancing user-specific investment strategies.

The developed tool effectively addressed some significant challenges such as enabling customer-centric comparisons in financial data visualization, balancing the complexity of detailed financial information with the need for a simple, navigable user interface. This was achieved by developing a meticulous design strategy that simplifies data presentation without sacrificing analytical depth.

Key results and advantages of this tool are listed below:

● Personalized Data Analysis: Provides a tailored visualization platform that integrates individual investment data for deep analysis.

● Dynamic Comparison Tools: Enables intuitive comparisons of investment decisions, visualizing their impacts under varying market conditions.

● Advanced Visualization Techniques: Utilizes complex methods such as parallel coordinate plots and multiple line charts for detailed financial insights.

● User-Friendly Interface: Merges complex data integration with a straightforward interface, making advanced analysis accessible to all users.

● Decision Making Support: Supports and evaluates investment strategies by offering and comparing transaction data real-time financial data.

In conclusion, this thesis encapsulates a forward-thinking venture into the realm of investment data visualization, marking a significant stride in the domain with the introduction of a customer-centric tool implemented on the first publicly available dataset of transactions from an anonymous investor.

9 Future Work

Building on the successful implementation of this customer-centric visualisation tool for stock investments, the scope for future enhancements is substantial, aiming to address potential challenges and extend the tool's capabilities. An integral area of focus will be the enhancement of data integration capabilities. Integrating additional diverse data sources such as social media sentiment and macroeconomic indicators could enrich the tool’s analytical depth, providing users with broader insights into factors that influence market dynamics. This integration will facilitate a more comprehensive decision-making process by considering both micro and macroeconomic factors.

Another significant extension involves incorporating machine learning algorithms to enable predictive analytics within the tool. By leveraging historical data and current market trends, machine learning models could predict future stock performance, thereby aiding investors in making proactive investment decisions. This predictive capability would transform the system from a reactive analytical tool to a proactive advisory platform, enhancing its utility and appeal to a broader user base.

To enhance user experience, further customization and personalization of the dashboard are crucial. Future iterations could focus on allowing users to create and save multiple dashboard configurations that cater to different analysis needs or investment scenarios. Additionally, expanding the platform to mobile devices would increase accessibility, enabling users to perform timely analyses and make informed decisions on-the-go. This expansion would ensure that the system meets the demands of a dynamic investment environment where decisions often need to be made quickly.

Lastly, considering the strict regulatory environment surrounding financial investments, incorporating regulatory compliance monitoring features would provide significant value. Such features could help users ensure that their investment activities remain within legal bounds and adhere to the latest financial regulations, reducing the risk of non-compliance and associated penalties.

Reference

Arleo, A. *et al.* (2023) 'Visual Exploration of Financial Data with Incremental Domain Knowledge', *Computer graphics forum*, 42(1), pp. 101–116. Available at: https://doi.org/10.1111/cgf.14723.

A. Leite, R. *et al.* (2020) 'NEVA: Visual Analytics to Identify Fraudulent Networks', *Computer graphics forum*, 39(6), pp. 344–359. Available at: https://doi.org/10.1111/cgf.14042.

Abad-Segura, E. *et al.* (2020) 'Financial Technology: Review of trends, approaches and management', *Mathematics (Basel)*, 8(6), p. 951. Available at: https://doi.org/10.3390/math8060951.

Arleo, A., Tsigkanos, C., Jia, C., Leite, R. A., Murturi, I., Klaffenböck, M., Dustdar, S., Wimmer, M., Miksch, S., & Sorger, J. (2019). Sabrina: Modeling and Visualization of Financial Data over Time with Incremental Domain Knowledge. 2019 IEEE Visualization Conference (VIS), 51–55. Available at: https://doi.org/10.1109/VISUAL.2019.8933598

Blyakhman, A. (2022) *SELECTING DATA VISUALIZATION TOOLS: Finance professionals must weigh a number of factors in choosing a program to suit the data consumption within their organizations, Strategic finance (Montvale, N.J.)*. Institute of Management Accountants, p. 62.

Endert, A. et al. (2017) 'The State of the Art in Integrating Machine Learning into Visual Analytics', Computer graphics forum, 36(8), pp. 458–486. Available at: https://doi.org/10.1111/cgf.13092.

Firat, E. E., Vytla, D., Singh, N. V., Jiang, Z., & Laramee, R. S. (2023). MoneyVis: Open Bank Transaction Data for Visualization and Beyond.

Guo, H. et al. (2022) 'RankFIRST: Visual Analysis for Factor Investment By Ranking Stock Timeseries', IEEE transactions on visualization and computer graphics, PP, pp. 1–10. Available at: https://doi.org/10.1109/TVCG.2022.3209414.

Julie Rodriguez, P.K. (2016) *Visualizing Financial Data*. 1st edn. Newark: Wiley. Available at: https://doi.org/10.1002/9781119183563.

Ko, S. et al. (2016) 'A Survey on Visual Analysis Approaches for Financial Data', Computer graphics forum, 35(3), pp. 599–617. Available at: https://doi.org/10.1111/cgf.12931.

Laramee, Bob, and S. Robert. "Bob's Project Guidelines: Writing a Dissertation for a BSc. in Computer Science." Innovation in Teaching and Learning in Information and Computer Sciences 10.1, 2011, pp. 43-54.

Leite, R. A., Gschwandtner, T., Miksch, S., Kriglstein, S., Pohl, M., Gstrein, E., & Kuntner, J. (2018). EVA: Visual Analytics to Identify Fraudulent Events. IEEE Transactions on Visualization and Computer https://doi.org/10.1109/TVCG.2017.2744758

Liu, S. et al. (2014) 'A survey on information visualization: recent advances and challenges', The Visual computer, 30(12), pp. 1373–1393. Available at: https://doi.org/10.1007/s00371-013-0892-3.

Lei, S. T., & Zhang, K. (2010). A visual analytics system for financial time-series data. Proceedings of the 3rd International Symposium on Visual Information Communication, 1–9. https://doi.org/10.1145/1865841.1865868

McNabb, L., & Laramee, R. S. (2017). Survey of Surveys (SoS)—Mapping The Landscape of Survey Papers in Information Visualization. Computer Graphics Forum, 36(3), 589–617. Available at: https://doi.org/10.1111/cgf.13212

Ren, J. (2024). Design and Implementation of Data Management and Visualisation Module in Financial Digital Management. Journal of Information & Knowledge Management, 2450023.

Shneiderman, B. (1996) 'The eyes have it: a task by data type taxonomy for information visualizations', in IEEE SYMPOSIUM ON VISUAL LANGUAGES, PROCEEDINGS. LOS ALAMITOS: IEEE, pp. 336–343. Available at: https://doi.org/10.1109/VL.1996.545307.

Shao, C. *et al.* (2022) 'IoT data visualization for business intelligence in corporate finance', *Information processing & management*, 59(1), p. 102736. Available at: https://doi.org/10.1016/j.ipm.2021.102736.

Sun Y, Xiong H, Yiu S M, et al. Bitvis: An interactive visualization system for bitcoin accounts analysis[C]//2019 Crypto Valley conference on blockchain technology (CVCBT). IEEE, 2019: 21-25.

Wang, X.-M. *et al.* (2016) 'A Survey of Visual Analytic Pipelines', *Journal of computer science and technology*, 31(4), pp. 787–804. Available at: https://doi.org/10.1007/s11390-016-1663-1.

Yue, X. et al. (2020) 'sPortfolio: Stratified Visual Analysis of Stock Portfolios', IEEE transactions on visualization and computer graphics, 26(1), pp. 601–610. Available at: https://doi.org/10.1109/TVCG.2019.2934660.