Data Visualization

MSc in CSTE: CIDA option

Cranfield University

Submitted by:

Nnamdi Daniel Aghanya - SID: 460020

Paul Fontanges – SID: 461720

Word count: 2991

Introduction

Recent years have witnessed a transformative surge in digital currencies; most notably cryptocurrencies such as Bitcoin and Ethereum, which leverage blockchain technology to facilitate peer-to-peer transactions (Nakamoto, 2008; Rainer Böhme et al., 2015). As these currencies gain traction, they increasingly intersect with global financial systems, creating new avenues for cross-border transactions and reshaping conventional notions of monetary policy (Yermack, 2017). As Rzayev et al. (2024) point out, when cryptocurrencies peaked, their market value was close to $3 trillion, matching some of the biggest stocks in the world. The growth of cryptocurrency has led to numerous benefits such as: financial inclusion, the potential to enhance transaction efficiency, and reduced costs, which can positively impact economic growth and financial stability (Tong and Chen, 2021). However, there is growing apprehension regarding regulatory loopholes, financial instability, and national security risks (Prasad, 2023; Zetzsche et al., 2017).

Concerns over illicit financing, regulatory ambiguity, and systemic risks remain formidable obstacles to the mass adoption of digital currencies (Bis.org, 2022). From a security standpoint, policymakers and institutions must assess the vulnerability of digital assets to money laundering, cyber-attacks, and terrorism financing (Foley et al., 2019). Simultaneously, the quest for effective integration with existing financial infrastructures raises questions about compliance, consumer protection, and broader economic stability (Chan, 2023; Zetzsche et al., 2017). In an era where national interests converge with rapidly evolving financial technology, a data-driven lens becomes essential for separating genuine opportunity from hype.

To that end, the use of advanced data visualization techniques emerges as a crucial methodological approach; thus, this research employs large-scale datasets from Kaggle and advanced Python-based visualization libraries such as Matplotlib, Seaborn, and Plotly to analyze patterns in cryptocurrency adoption, blockchain network interactions, and relevant economic indicators. By the end of this study, the researcher's goal is to offer a clear, data-informed perspective on how digital currencies integrate into—or potentially disrupt—existing financial systems.

Aims and Objectives

This study aims to assess the interplay between digital currencies and global financial stability while recognizing the constraints of relying solely on a historical price dataset. Specifically, we seek to:

- **Highlight Data Gaps**: Examine why a daily price dataset is insufficient for comprehensive adoption, usage, and policy analyses.
- **Analyse Global Adoption**: Combine external socio-economic indicators to investigate cross-country trends in cryptocurrency transactions and market penetration.
- **Explore Macroeconomic Impact**: Illustrate relationships between digital currency use and indicators such as inflation or unemployment through advanced visual analytics.
- **Conduct Network Analysis**: Uncover patterns in blockchain transaction flows and identify anomalies that may affect national security.
- **Evaluate Policy Simulations**: Develop interactive dashboards using Plotly to test regulatory scenarios and measure their impact on compliance and fraud prevention.
- **Address Ethical Issues**: Balance transparency with privacy in data visualisation practices for sensitive financial information.

Methodology

In this section, we explore how to tackle the fascinating challenge of digital currency's impact on global finance and its security implications. Working with historical price data spanning open, high, low, close, volume, and market cap metrics, we faced an interesting constraint –

our dataset (SRK, 2021), while rich in trading information, couldn't directly speak to broader economic indicators or network patterns. This limitation pushed us to think creatively about extracting meaningful insights from price movements and trading behaviors, leading us down some unexpected but revealing analytical paths.

Data

The data used for this research is the "Cryptocurrency Historical Prices dataset" sourced from (SRK, 2021) on Kaggle—an online community and data science competition platform run by Google LLC. The dataset consists of 23 CSV files, each encompassing major digital currencies containing daily price history (from April 28, 2013 through 2021). Each CSV file contains daily records of Open, High, Low, Close, Volume, and Market Cap. After reading these files with Pandas, we handle missing or invalid values (e.g., zeros and negatives in logs) by applying data-cleaning routines (Baviskar, D. and Sankari, 2023). Additionally, calculations such as logarithmic transformations for the closing price and usage metrics (i.e., $Volume/Close$) are performed to facilitate more stable analyses and to minimize skewness (Baur, Hong and Lee, 2018). Due to uncertainty around IT lab environments and potential API installation constraints, we opted for manual file downloads, setting the stage for our initial data processing.

Visualization Approach

We built our analytical framework in Python. Two interactive dashboards were developed using Streamlit and Plotly Express for interactive visualizations. The first dashboard aims to analyze digital currency adoption trends (task a) by focusing on individual metrics (e.g., volume of transactions, log-close price, usage), enabling users to filter and observe trends in a single interface dynamically. The second dashboard addresses comparative analysis (task e), featuring multi-cryptocurrency comparisons of closing prices, volumes, and market capitalizations. The processing pipeline we developed handles several critical tasks:

- We implemented logarithmic transformations to make sense of the wild price swings common in crypto markets.
- We tackled computational edge cases with epsilon values ($1e - 10$), ensuring our calculations remained reliable and uncluttered.
- We built in careful cleaning protocols for those pesky infinite values that emerged during our transformations.

Implementation

In this section, we discuss the implementation of our cryptocurrency analysis dashboard using Python's data processing and visualization libraries. Given our dataset limitations, we focused on maximizing insights from available price and volume metrics, implementing two distinct dashboard approaches to address portions of tasks A and E.

Multi-Currency Analysis Dashboard

Our first implementation - analysis_dashboard.py—combines Streamlit's interface capabilities with Plotly Express's visualization tools to create a responsive analytical platform. At the core of our dashboard lies a sophisticated data processing pipeline, implemented through a cached loading function (@st.cache_data) that significantly improves performance by storing processed data in memory. This pipeline begins by reading our combined cryptocurrency dataset and applying several critical transformations. We implemented logarithmic transformations for closing prices to visualize exponential price movements. Handling computational edge cases through epsilon values (1e-10) to prevent mathematical errors with near-zero values. The pipeline also calculates usage metrics by examining the relationship between trading volume and closing prices, providing insights into market engagement patterns.

The dashboard visualization framework splits into two distinct view modes. The Individual Metrics view presents four key analytical perspectives: trading volume in USD, adjusted closing prices (revealing long-term price trends), usage metrics (showing market engagement), and total trading volume (aggregating across all cryptocurrencies). Each visualization employs interactive elements, allowing users to zoom, hover for detailed information, and filter specific cryptocurrencies. Combined Analysis view offers a different perspective, focusing on market relationships through pie charts showing volume distribution and a correlation heatmap revealing price relationships between distinct cryptocurrencies.

Single Currency Deep-Dive

Our second Python script 'historical_analysis.py' focuses on providing granular insights into individual cryptocurrency performance while maintaining the flexibility for multi-coin comparison. The analysis engine offers two primary modes of operation. In single-coin mode, the system generates six distinct visualizations that tell a comprehensive story of the cryptocurrency's performance. We track price movements through a multi-line plot showing open, high, low, and close prices, complemented by a monthly average price bar chart that smooths daily volatility to reveal longer-term trends.

To better understand market dynamics, we implemented a volume and market cap tracking system that visualizes these metrics on the same timeline, enabling users to spot correlations between price movements and market activity. We also developed a novel market metrics distribution visualization that provides a snapshot of market pricing data distributed in a pie chart. For a deeper analysis of market behavior, we introduced two specialized metrics. While volatility analysis focuses on the range of daily prices, our daily percentage change makes it possible to measure the degree and frequency of price swings. In multi-coin comparison mode, the system generates synchronized visualizations across selected cryptocurrencies, including direct price comparisons, volume analysis, market share distribution, and market cap trends. This comparative view is valuable for understanding how cryptocurrencies interact and compete within the broader market ecosystem.

Task Completion and Limitations

While our implementations successfully tackle aspects of the assigned tasks, the dataset's inherent limitations significantly shaped our analytical approach. For task A (Global Adoption Analysis), we could only examine adoption through trading volumes and price movements, falling short of the desired geographical and demographic analysis. The absence of country-specific transaction data meant we couldn't map adoption patterns across regions or correlate them with local economic indicators like GDP.

Our visualization system makes substantial progress on task E (Policy Simulation Dashboard) by providing interactive tools for market analysis through Plotly and Streamlit. The dashboard lets users explore historical trends, compare multiple currencies, and analyze market correlations. However, the dataset's limited scope—containing only price and volume data—meant we couldn't implement more sophisticated features like compliance rate tracking or fraud detection metrics that would be crucial for meaningful policy analysis.

The remaining tasks (B through D and F) proved impossible to address meaningfully with our dataset. The lack of macroeconomic indicators prevented any analysis of relationships between cryptocurrency adoption and economic factors like inflation or unemployment (task B). Similarly, the absence of blockchain transaction data made network analysis (task C) and security risk assessment (task D) unfeasible. While we could theoretically discuss ethical considerations in financial data visualization (task F), the public nature of our pricing dataset didn't present the privacy concerns that would arise with actual transactions or user data.

Results

Dataset description and summary statistics

The cryptocurrencies' characteristics include volume of transactions, opening time, opening price, highest price, lowest price, and closing price; all of these prices are expressed in US dollars. Daily cryptocurrency prices and volumes were collected from Kaggle. The observation period is from 29 April 2013 to 6 July 2021, resulting in a sample of 23 cryptocurrencies. Our final sample consists of 37082 observations.

Table 1 below shows the summary statistics. The close Price is the cryptocurrency's value in relation to the US dollar. The rationale for not using opening (opening price of the digital currency in the day), high (highest price of the currency during the day) and low (lowest prices during the day) prices because these do not closely resemble the corresponding actual values except the closing price (Oyedele et al., 2023). Furthermore, because cryptocurrency prices are so volatile, it is preferable to use the closing price rather than the opening, high, or low price because these do not always reflect the precise market trend (Anoop et al., 2025). Volume ($) is the volume of transactions in USD. We derived Usage by dividing Volume ($) by close price $. The result shows that the minimum closing price is $0.0001 and maximum closing price is $63,503 with an average (median) of $987.12 ($1.011) and a standard deviation of $5,094 from the average, suggesting high volatility in prices. Similarly, the average (median) transaction volume is $3.022 billion ($85.122 million), respectively and its standard deviation is $11.910 billion. The data show a lack of close alignment between the mean and median values, suggesting the transaction volume is skewed towards smaller values in the distribution. Also, the daily average usage is about 4.002 billion and its standard deviation is 15.511 billion. Table 2 presents the correlation matrix of prices among the cryptocurrencies. The results show that some currencies are positive and have a perfect linear relationship, suggesting the currencies are directly related direction. Similarly, Fig. 1 visualises the correlation heatmap of cryptocurrency markets. Most cryptocurrencies show positive price correlations, with particularly strong relationships among major assets. However, stablecoins (USDT, USDC) display distinct patterns, given their design to maintain stable values.

These findings suggest that while cryptocurrency adoption has grown substantially, it remains concentrated among a few major players. The emergence of high trading volumes in stablecoins indicates their crucial role in market infrastructure, potentially serving as a bridge between traditional finance and cryptocurrency markets. The data reveals a maturing market with increasing sophistication in trading patterns, though the concentration of volume in a few major currencies suggests that widespread adoption may still be in its early stages.

Table 1. Summary statistics for a sample 23 cryptocurrencies for the period 29 April 2013 to 6 July 2021

	Mean	Median	Standard Deviation	Min	Max
Price (Close Price) $	987.12	1.0011	5,093.70	0.0001	63,503.46
Volume ($)	3,022,541,604	85,128,048	11,909,631,166	0.0000	350,967,941,479
Usage (Volume/Close)	4,002,894,492	45,691,122	15,510,895,020	0.0000	538,644,290,943

Table 2. Cryptocurrency price Correlation Matrix

Symbol	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1. AAVE	1																						
2. ADA	0.8	1.0																					
3. ATOM	0.9	0.9	1.0																				
4. BNB	0.8	0.9	0.9	1.0																			
5. BTC	0.9	0.9	0.9	0.9	1.0																		
6. CRO	0.6	0.6	0.7	0.6	0.7	1.0																	
7. DOGE	0.6	0.8	0.7	0.9	0.7	0.4	1.0																
8. DOT	0.9	0.9	1.0	0.9	1.0	0.6	0.6	1.0															
9. EOS	0.7	0.3	0.6	0.2	0.1	0.3	0.2	0.8	1.0														
10. ETH	0.9	1.0	0.9	0.9	0.9	0.6	0.8	0.8	0.3	1.0													
11. LINK	0.9	0.8	0.9	0.9	0.9	0.7	0.7	0.9	0.0	0.9	1.0												
12. LTC	0.9	0.8	0.9	0.6	0.8	0.6	0.5	0.9	0.6	0.8	0.6	1.0											
13. MIOTA	0.9	0.5	1.0	0.3	0.4	0.7	0.3	0.9	0.7	0.5	0.2	0.8	1.0										
14. SOL	0.7	0.9	0.8	**1.0**	0.7	0.4	0.9	0.7	0.8	0.9	0.8	0.8	0.8	1.0									
15. TRX	0.7	0.8	0.9	0.7	0.7	0.6	0.6	0.8	0.6	0.8	0.6	0.8	0.6	0.9	1.0								
16. UNI	0.9	0.9	**1.0**	0.9	0.9	0.7	0.7	**1.0**	0.8	0.9	0.9	0.9	1.0	0.8	0.9	1.0							
17. USDC	0.1	-0.2	-0.1	-0.2	-0.3	-0.3	-0.2	0.0	-0.1	-0.3	-0.3	-0.3	-0.2	-0.1	-0.2	0.1	1.0						
18. USDT	0.1	-0.1	-0.2	-0.1	0.0	-0.3	0.0	-0.1	-0.1	0.0	-0.1	0.1	0.1	-0.1	-0.1	0.0	0.2	1.0					
19. WBTC	0.9	0.9	0.9	0.9	**1.0**	0.7	0.6	**1.0**	0.4	0.9	0.9	0.9	0.9	0.7	0.8	0.9	-0.3	-0.2	1.0				
20. XEM	0.6	0.5	0.9	0.2	0.5	0.6	0.2	0.8	0.5	0.5	0.2	0.8	0.8	0.4	0.6	0.6	-0.2	0.1	0.9	1.0			
21. XLM	0.9	0.8	1.0	0.6	0.8	0.6	0.5	0.9	0.7	0.8	0.5	0.9	0.7	0.8	0.9	0.9	-0.2	0.0	0.9	0.8	1.0		
22. XMR	0.8	0.8	0.9	0.6	0.8	0.7	0.5	0.8	0.6	0.8	0.5	**1.0**	0.9	0.9	0.8	0.9	-0.3	0.1	0.9	0.8	0.9	1.0	
23.XRP	0.7	0.7	0.8	0.5	0.6	0.5	0.5	0.7	0.7	0.7	0.3	0.9	0.8	0.9	0.8	0.8	-0.1	0.1	0.8	0.8	0.9	0.9	1.0

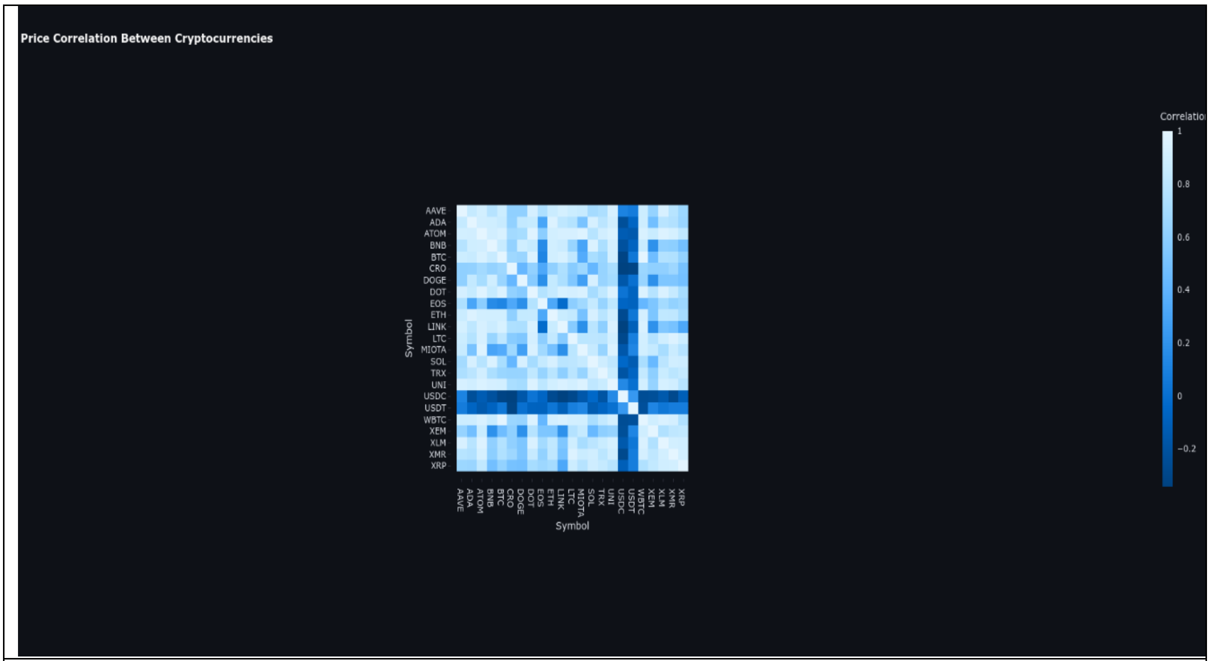

Figure 1. Price correlation Between Cryptocurrencies

Analysis of Global Adoption Trends

Our analysis of cryptocurrency adoption trends reveal several significant patterns across the examined period (2013-2021). Fig. 2 illustrates trading volume evolution across all major cryptocurrencies, showing a remarkable increase in market activity, particularly from 2017 onwards. However, there seems to be an anomaly in the graph and is highlighted for the period Jan 2020 to August 2020 and from Jan. 2021 to July 2021.

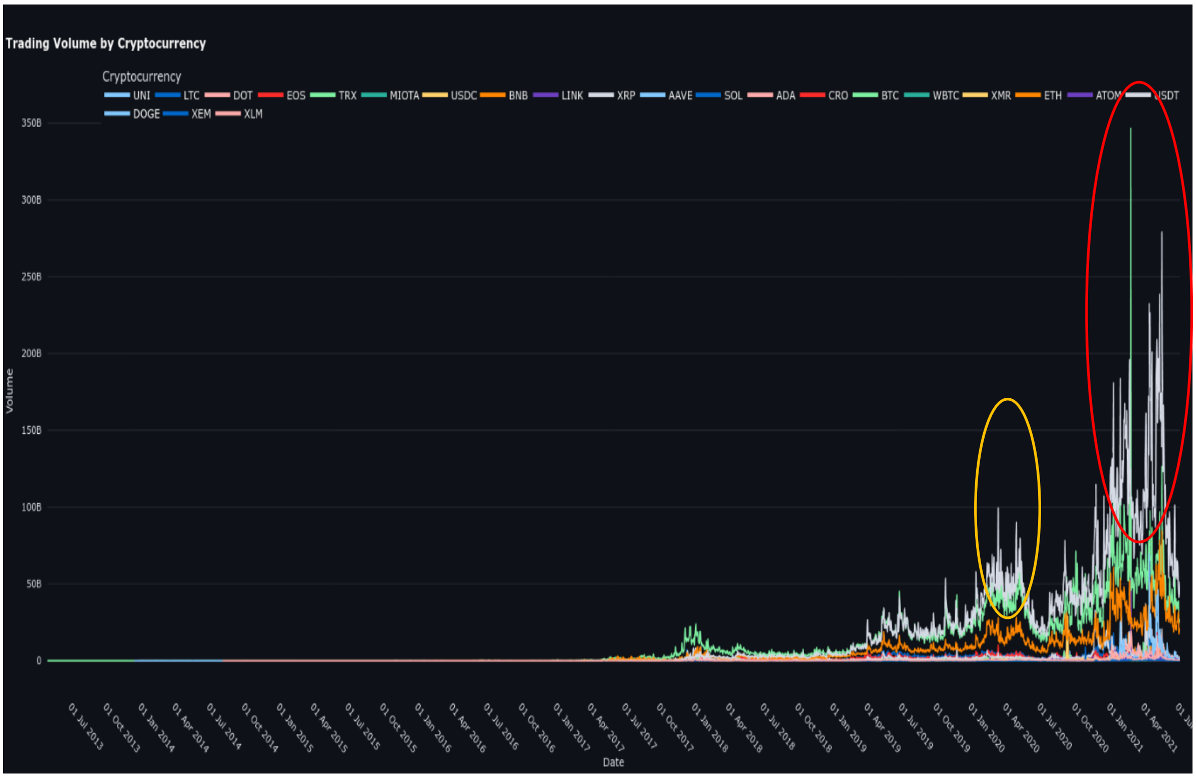

Figure 2. Volume of transactions in USD of 23 different cryptocurrencies from 29 April 2013 to 6 July 2021.

Next, we analyse market share dominance of certain cryptocurrencies. Fig. 3 reveals market share analysis in terms of trading volume. USDT (Tether) commands the largest share at 40.1% of total volume, followed by BTC (Bitcoin) at 29.1%, and ETH (Ethereum) at 13.6%. This distribution suggests that stablecoins (particularly USDT) play a crucial role in market liquidity and trading activity.

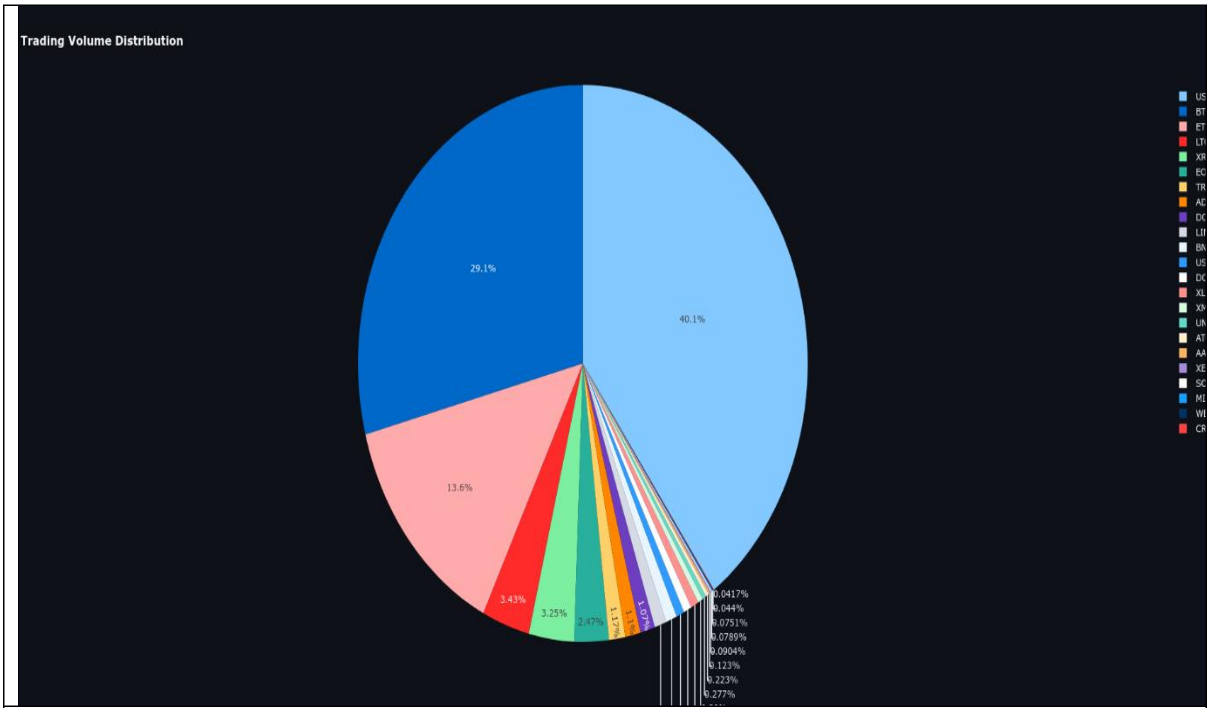

Figure 3. Trading Volume Distribution

Next, we aggregated the trading volume of all the cryptocurrencies during the sample period. Fig. 4 below provides visual evidence of trading volume in USD. It reveals that major activity started in 2017. The total trading volume across all cryptocurrencies demonstrates the overall market growth, with a dramatic surge in activity during 2020-2021. Three anomalies were noticed during this period and highlighted. This aggregated view shows that the market experienced several distinct phases of expansion, with each subsequent phase showing higher baseline activity levels than the previous one.

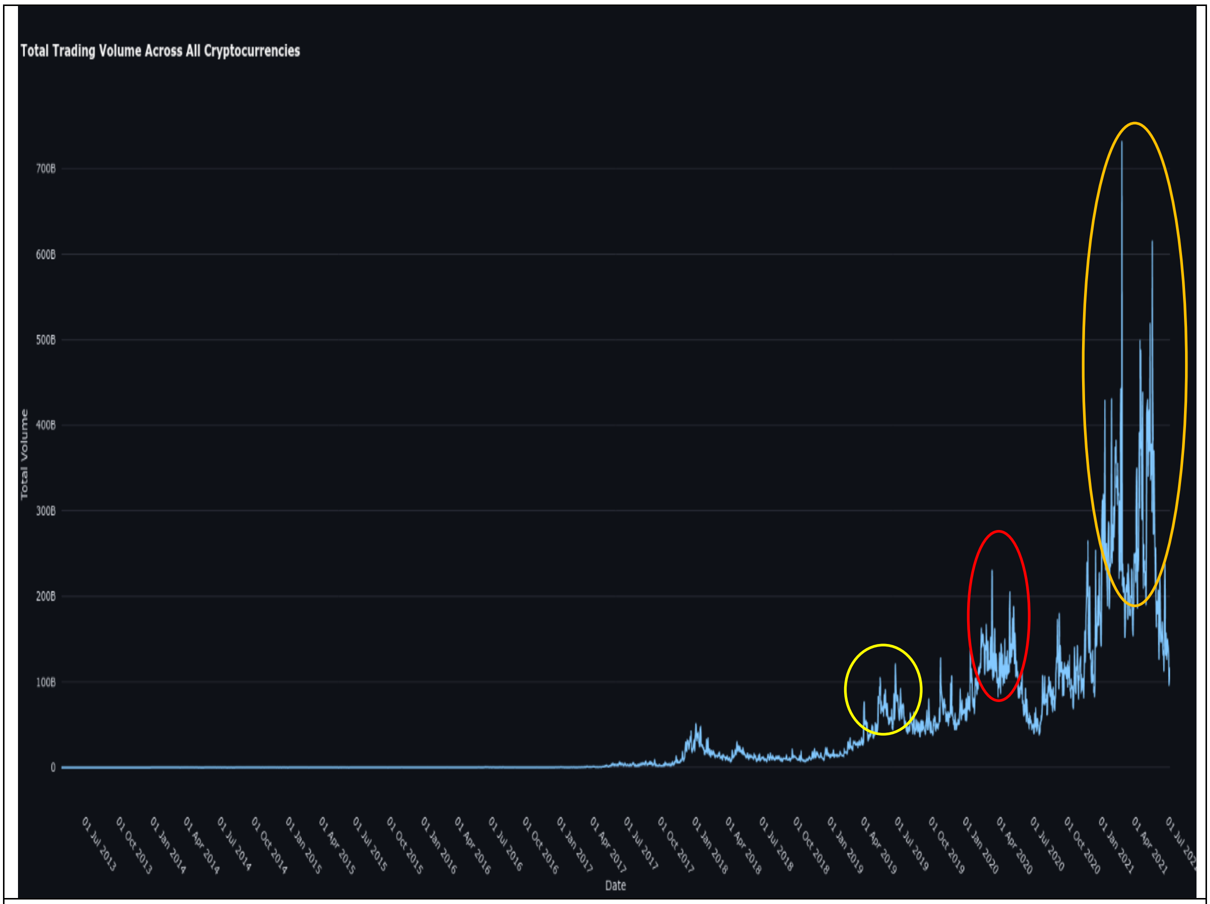

Fig 4. Total Trading Volume Across all Crypto Currencies

Next, we visualize the historical price of cryptocurrencies. Fig. 5 provides some insights regarding the prices of the cryptocurrencies during the sample period. The visualization reveals that while cryptocurrencies generally show correlated price movements, there are distinct periods of divergence, particularly during the 2020-2021 market expansion. We observed 3 anomalies: Oct 2017 to July 2018; July 2019 to April 2020 and from Jan 2021 to July 2021.

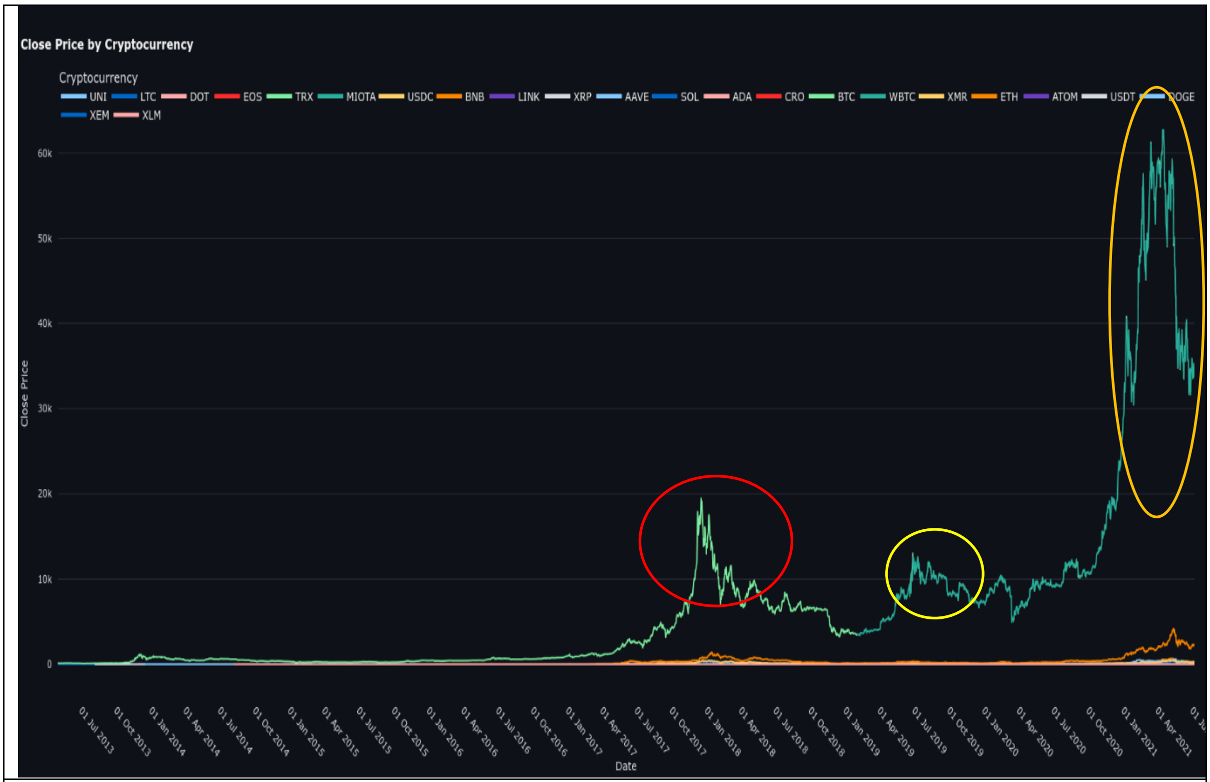

Fig. 5. Close Price of Cryptocurrency

Next, we show the usage trend metrics (volume/price ratio) across the 23 cryptocurrencies. Fig. 6 displays the trend. The metric shows increasing market usage of digital currency from 2019. We noticed anomalies around Nov. 2014; between Jan 2020 and July 2020, and between Jan 2021 and June 2021. The anomaly of usage was very high in very few currencies as highlighted.

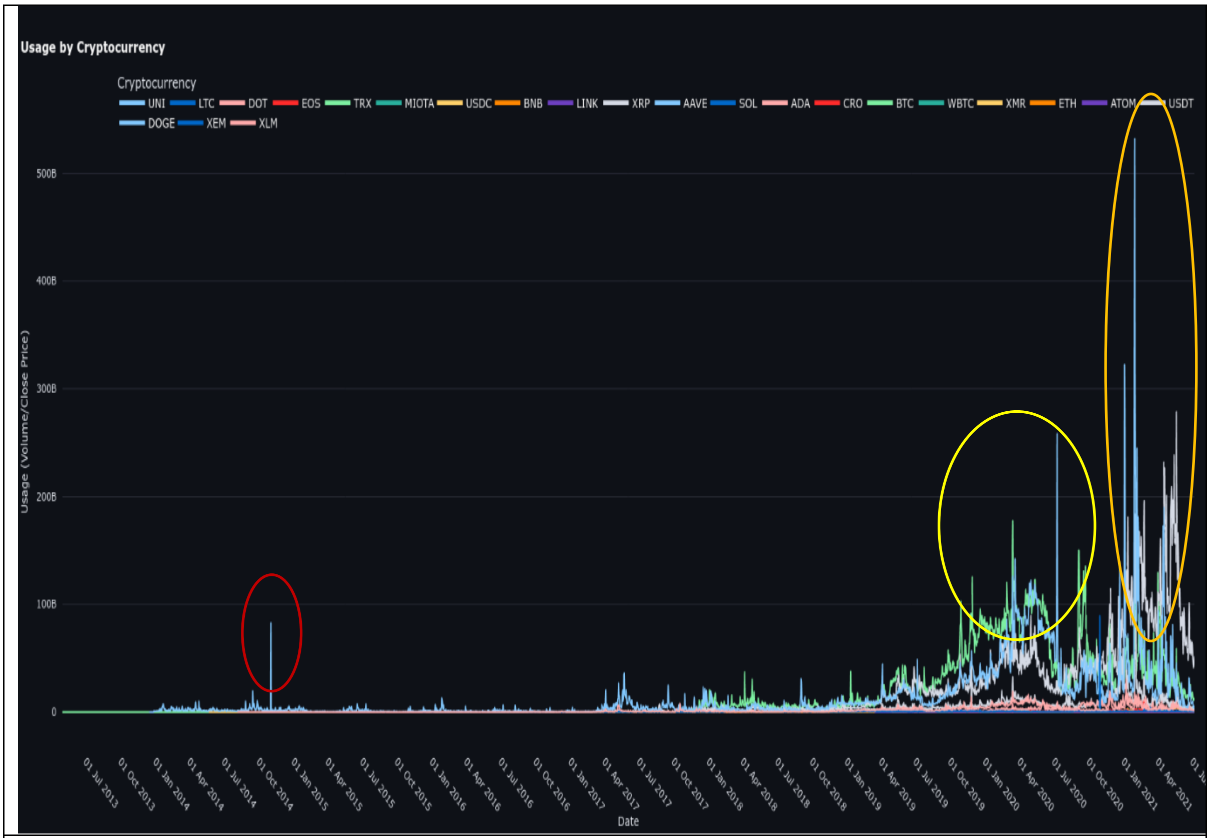

Figure 6: Cryptocurrency usage trend

Analysis of Market Dominance and Adoption Patterns

Regarding market dominance, our implementation of the policy simulation dashboard, while limited by the available data, successfully delivered several key visualization components for market analysis. Our analysis of cryptocurrency market dynamics reveals distinct patterns of market dominance and adoption across the examined period (2013-2021). Figure 7 presents the market share distribution by market capitalization, showing Bitcoin's clear dominance at 51.3% of total market capitalization, followed by Ethereum at 21.7%, while Tether account for 4.99%. The concentration of market value among a few major cryptocurrencies suggests a market structure where established Coinbase maintains significant advantages over newer entrants.

Figure 7. Market Capitalization Comparison

Ethical Considerations in Data Visualisation

Users of financial data can make better decision making they are given more relevant information (Pinsker, 2007). Accurate financial data helps users and investors to make better investment decisions by conducting more accurate financial analyses (Hodge et al., 2010). However, visualisation can confuse, mislead users by not telling the whole truth, leading to misrepresentation of information and creating harm to the users. Moreover, visualization brought on by inadequate data anonymization or bad visual choices can lead to disclosure of sensitive financial data.

Some strategies to balance transparency with privacy concerns are:

i. Ensure data privacy and security. There is the need to recognise the privacy policy of organisations by being aware of the connection between data sharing policies and privacy parameters (Bhattacharjee et al., 2020). Data visualisation must ensure that adequate security measures are implemented while also complying with the data protection rule of the organisation and the government.

ii. Transparency in data sources. For ethical data visualization of financial data, data sources must be kept accurate, transparent, and disclosed.

iii. Finding a Balance Between Complexity and Simplicity. To ensure ethical data presentation and successful communication, data visualization must strike the correct balance between simplicity and complexity.

Global Adoption Analysis: From the data we are using for the coursework, we do not have information on GDP and demographic penetration; and thus, cannot carry out the visualisation of some global adoption analysis.

Economic Impact Visualisation: From the link provided, we collected daily data on 23 cryptocurrencies. However, macroeconomic data such as inflation, unemployment are usually provided on monthly, quarterly, semi-annual or annual data; hence, cannot match them for visualisation. Secondly, the macroeconomic data was not provided to enable us carry out the visualisation. Thirdly, if we try to use other website for this data, we do not know the countries these digital currencies are located and hence cannot visualise the relationship between them.

National Security Implications: Given data available data, we could not visualise possible risks like money laundering, fraud, and unauthorized cross-border transactions. More data is required to enable us to apply anomaly detection techniques for the visualisation.

Policy Simulation Dashboard: From the data source, different regulatory policies on the cryptocurrency markets were not provided. The fraud detection efficiency and transaction compliance rates were not also provided. Thus, it is not possible to be able to use Plotly Dash to develop an interactive dashboard to analyse how various regulatory measures affect the digital currency markets.

Conclusion

This assignment sets out to examine the impact of digital currencies on global finance and their security implications through advanced data visualization techniques. While our initial scope was ambitious, the limitations of our dataset significantly shaped our analytical capabilities.

Our implementation successfully delivered two complementary dashboards that provide valuable insights into cryptocurrency market dynamics. The first dashboard effectively visualizes trading volumes, price movements, and market correlations, while the second offers deeper analysis of market capitalization distribution. These tools demonstrate the power of interactive visualization in understanding complex market behaviors, particularly during critical periods like the 2020-2021 market expansion.

However, this study also highlights critical gaps in publicly available cryptocurrency data. The inability to analyze geographical distribution, demographic penetration, or detailed transaction networks limited our capacity to address questions about global adoption patterns, security implications, and regulatory impacts. Future research would benefit from access to more comprehensive datasets including cross-border flows, demographic information, and granular transaction data. Such data would enable deeper insights into adoption patterns, security risks, and regulatory effectiveness in the rapidly evolving digital currency landscape.

Appendix

References

Anoop, C. V., Negi, N., & Aprem, A. (2025). Bayesian machine learning framework for characterizing structural dependency, dynamics, and volatility of cryptocurrency market using potential field theory. *Expert Systems with Applications*, *261*, 125475. https://doi.org/10.1016/j.eswa.2024.125475

Baur, D.G., Hong, K. and Lee, A.D. (2018). Bitcoin: Medium of exchange or speculative assets? *Journal of International Financial Markets, Institutions and Money*, [online] 54, pp.177–189. doi: https://doi.org/10.1016/j.intfin.2017.12.004.

Baviskar, V.S., Radha, D., & Sankari, S.U. (2023). Cryptocurrency Price Prediction and Analysis. In *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, [online] pp.1–7. doi: https://doi.org/10.1109/icccnt56998.2023.10308332.

Bhattacharjee, K., Chen, M., & Dasgupta, A. (2020). Privacy-preserving data visualization: reflections on the state of the art and research opportunities. *Computer Graphics Forum*, Vol. 39(3), pp. 675-692. https://doi.org/10.1111/cgf.14032

Bis.org. (2022). *III. The future monetary system*. [online] Available at: https://www.bis.org/publ/arpdf/ar2022e3.htm [Accessed 9 Jan. 2025].

Bohme, R., Christin, N., Edelman, B. and Moore, T. (2015). Bitcoin: Economics, Technology, and Governance. *The Journal of Economic Perspectives*, [online] 29(2), pp. 213–238. doi: https://doi.org/10.1257/jep.29.2.213.

Chan, S. (2023). China's Central Bank Digital Currency: Impact and Policy Implications. *An International Journal*, [online] 21(3), pp.141–157. Available at: https://muse.jhu.edu/pub/43/article/904731/pdf.

Foley, S., Karlsen, J.R. and Putniņš, T.J. (2019). Sex, Drugs, and Bitcoin: How Much Illegal Activity Is Financed through Cryptocurrencies? *Review of Financial Studies*, [online] 32(5), pp.1798–1853. doi: https://doi.org/10.1093/rfs/hhz015.

Hodge, F., and Pronk, M. (2006). The Impact of Expertise and Investors' Use of Online Financial Report Information. *Journal of Accounting Auditing Finance,* 21(3), 267–292. https://doi.org/10.1177/0148558X0602100304

Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [online] Available at: https://www.ussc.gov/sites/default/files/pdf/training/annual-national-training-seminar/2018/Emerging_Tech_Bitcoin_Crypto.pdf.

Oyedele, A. A., Ajayi, A. O., Oyedele, L. O., Bello, S. A., & Jimoh, K. O. (2023). Performance evaluation of deep learning and boosted trees for cryptocurrency closing price prediction. *Expert Systems with Applications*, *213*, 119233. https://doi.org/10.1016/j.eswa.2022.119233

Pinsker, R. (2007). Long Series of Information and Nonprofessional Investors' Belief Revision. *Behavioral Research in Accounting,* 19(1), 197–214. https://doi.org/10.2308/bria.2007.19.1.197

Prasad, E. (2023). How will digital technologies influence the international monetary system? *Oxford Review of Economic Policy*, [online] 39(2), pp.389–397. doi: https://doi.org/10.1093/oxrep/grad011.

Rzayev, K., Sakkas, A., & Urquhart, A. (2024). An adoption model of cryptocurrencies. *European Journal of Operational Research*. https://doi.org/10.1016/j.ejor.2024.11.024

SRK (2021). Cryptocurrency Historical Prices. [online] Kaggle.com. Available at: https://www.kaggle.com/datasets/sudalairajkumar/cryptocurrencypricehistory [Accessed 9 Jan. 2025].

Tong, W., & Chen, J. (2021). A study of the economic impact of central bank digital currency under global competition. *China Economic Journal*, [online] 14, pp. 78 - 101. https://doi.org/10.1080/17538963.2020.1870282.

Zetzsche, D.A., Buckley, R.P., Barberis, J.N. and Arner, D.W. (2017). *Regulating a Revolution: From Regulatory Sandboxes to Smart Regulation*. [online] FLASH: The Fordham Law Archive of Scholarship and History. Available at: https://ir.lawnet.fordham.edu/jcfl/vol23/iss1/2/ [Accessed 9 Jan. 2025].

Code

Analysis_Dashboard.py

```
import streamlit as st
import pandas as pd
import plotly.express as px
import numpy as np

st.set_page_config(page_title="Cryptocurrency Analysis Dashboard", layout=
"wide")
st.markdown(
    "<h1 style='text-align: center; color: #003366;'>Cryptocurrency Analys
is Dashboard</h1>",
    unsafe_allow_html=True,
)


def safe_log(series):
    """Safely calculate log values, replacing zeros and negatives with NaN
"""
    return np.log(series.replace({0: np.nan, np.inf: np.nan, -np.inf: np.n
an}))


def get_common_layout(yaxis_title):
    """Create a common layout configuration for plots with customizable y-
axis title"""
    return dict(
        legend_title="Cryptocurrency",
        xaxis_title="Date",
        yaxis_title=yaxis_title,
        height=600,
        xaxis=dict(
            tickformat="%d %b %Y", dtick="M3", tickangle=45  # Show ticks
every quarter
        ),
        legend=dict(
            orientation="h",
            yanchor="bottom",
            y=0.94,
            xanchor="right",
            x=1,
            font=dict(size=14),
            title=dict(font=dict(size=16)),
            itemsizing="constant",
            itemwidth=40,
        ),
    )


@st.cache_data
def load_and_process_data():
    try:
```

```
        df = pd.read_csv("./combined_crypto_data.csv", parse_dates=["Date"
])
        df["Volume"] = df["Volume"] + 1
        epsilon = 1e-10
        df["log_close"] = np.log(df["Close"].clip(lower=epsilon))
        df["usage"] = df["Volume"].clip(lower=epsilon) / df["Close"].clip(
lower=epsilon)
        df["log_usage"] = np.log(df["usage"].clip(lower=epsilon))

        df = df.replace([np.inf, -np.inf], np.nan)

        return df
    except Exception as e:
        st.error(f"Error processing data: {str(e)}")
        return None


df = load_and_process_data()
if df is not None:
    symbols = df["Symbol"].unique()
    st.sidebar.header("Analysis Options")
    view_mode = st.sidebar.radio(
        "Select View", ["Individual Metrics", "Combined Analysis"]
    )
    if view_mode == "Individual Metrics":
        col1, col2 = st.columns(2)
        with col1:
            st.subheader("Trading Volume Over Time")
            volume_df = df[df["Volume"] > 0].copy()
            if not volume_df.empty:
                fig_volume = px.line(
                    volume_df,
                    x="Date",
                    y="Volume",
                    color="Symbol",
                    title="Natural Log of Trading Volume by Cryptocurrency
",
                    labels={"Volume": "Volume", "Date": "Date"},
                    log_y=True,
                )
                fig_volume.update_layout(**get_common_layout("Log Volume")
)
                st.plotly_chart(fig_volume, use_container_width=True)
            else:
                st.warning("No valid volume data available for plotting")
        with col2:
            st.subheader("Log Close Price Over Time")
            close_df = df[df["Close"].notna()].copy()
            if not close_df.empty:
                fig_log_close = px.line(
                    close_df,
                    x="Date",
                    y="Close",
                    color="Symbol",
```

```
                title="Close Price by Cryptocurrency",
                labels={"Close": "Close Price", "Date": "Date"},
            )
            fig_log_close.update_layout(**get_common_layout("Close Pri
ce"))
            st.plotly_chart(fig_log_close, use_container_width=True)
        else:
            st.warning("No valid close price data available for plotti
ng")
    col3, col4, col5 = st.columns(3)
    with col3:
        st.subheader("Usage Metrics (Volume/Close Price)")
        usage_df = df[df["usage"].notna()].copy()
        if not usage_df.empty:
            fig_log_usage = px.line(
                usage_df,
                x="Date",
                y="usage",
                color="Symbol",
                title="Usage by Cryptocurrency",
                labels={"usage": "Usage", "Date": "Date"},
            )
            fig_log_usage.update_layout(
                **get_common_layout("Usage (Volume/Close Price)")
            )
            st.plotly_chart(fig_log_usage, use_container_width=True)
        else:
            st.warning("No valid usage data available for plotting")
    with col4:
        st.subheader("Total Trading Volume")
        daily_volume = df.groupby("Date")["Volume"].sum().reset_index(
)

        if not daily_volume.empty:
            fig_total_volume = px.line(
                daily_volume,
                x="Date",
                y="Volume",
                title="Total Trading Volume Across All Cryptocurrencie
s",

                labels={"Volume": "Total Volume", "Date": "Date"},
            )
            fig_total_volume.update_layout(**get_common_layout("Total
Volume"))
            st.plotly_chart(fig_total_volume, use_container_width=True
)
        else:
            st.warning("No valid total volume data available for plott
ing")
    with col5:
        st.subheader("Total Usage")
        daily_usage = df.groupby("Date")["usage"].sum().reset_index()
        if not daily_usage.empty:
            fig_total_usage = px.line(
                daily_usage,
```

```
                    x="Date",
                    y="usage",
                    title="Total Usage Across All Cryptocurrencies",
                    labels={"usage": "Total Usage", "Date": "Date"},
                )
                fig_total_usage.update_layout(**get_common_layout("Total U
sage"))
                st.plotly_chart(fig_total_usage, use_container_width=True)
            else:
                st.warning("No valid total usage data available for plotti
ng")
    else:
        st.subheader("Combined Analysis View")

        st.markdown("### Market Comparison Metrics")
        latest_data = (
            df.groupby("Symbol")
            .agg({"Volume": "sum", "Close": "last", "usage": "mean"})
            .reset_index()
        )
        col1, col2 = st.columns(2)
        with col1:
            if not latest_data.empty:
                fig_market_share = px.pie(
                    latest_data,
                    values="Volume",
                    names="Symbol",
                    title="Trading Volume Distribution",
                )
                st.plotly_chart(fig_market_share, use_container_width=True
)
            else:
                st.warning("No valid market share data available")
        with col2:
            if not latest_data.empty:
                fig_usage_compare = px.bar(
                    latest_data,
                    x="Symbol",
                    y="usage",
                    title="Average Usage by Cryptocurrency",
                    labels={"usage": "Average Usage (Volume/Close Price)"}
,
                )
                st.plotly_chart(fig_usage_compare, use_container_width=Tru
e)
            else:
                st.warning("No valid usage comparison data available")
        st.markdown("### Price Correlation Analysis")
        valid_price_df = df[df["Close"] > 0].copy()
        if not valid_price_df.empty:
            price_pivot = valid_price_df.pivot(
                index="Date", columns="Symbol", values="Close"
            )
            correlation = price_pivot.corr()
```

```
            print("\nCorrelation Matrix (Close Prices) - Lower Triangle:")
            print("=" * 50)
            pd.set_option("display.precision", 2)
            lower_triangle = pd.DataFrame(
                np.tril(correlation),
                index=correlation.index,
                columns=correlation.columns,
            )
            for i in range(len(lower_triangle.columns)):
                for j in range(i + 1, len(lower_triangle.columns)):
                    lower_triangle.iloc[i, j] = ""

            print(lower_triangle.to_string())
            print("=" * 50)
            fig_corr = px.imshow(
                correlation,
                labels=dict(color="Correlation"),
                title="Price Correlation Between Cryptocurrencies",
            )
            st.plotly_chart(fig_corr, use_container_width=True)
        else:
            st.warning("No valid data available for correlation analysis")
    st.markdown("---")
    st.markdown(
        "<p style='text-align: center;'>Analysis based on historical crypt
ocurrency data</p>",
        unsafe_allow_html=True,
    )
else:
    st.error(
        "Failed to load and process data. Please check your data file and
try again."
    )
```

Historical_Analysis.py

```
import streamlit as st
import pandas as pd
import plotly.express as px
import os


st.set_page_config(page_title="Cryptocurrency Historical Analysis", layout
="wide")
st.markdown("<h1 style='text-align: center; font-family: Arial, sans-serif
; color: #003366;'>Cryptocurrency Historical Analysis</h1>", unsafe_allow_
html=True)
@st.cache_data
def load_data():
    crypto_data = {}
    data_dir = "crypto_data"
    for file in os.listdir(data_dir):
        if file.endswith(".csv"):
            currency_name = file.split(".")[0]
            crypto_data[currency_name] = pd.read_csv(os.path.join(data_dir
, file))
    return crypto_data
data = load_data()
st.sidebar.header("Filter Options")
currencies = list(data.keys())
comparison_mode = st.sidebar.radio("Comparison Mode", ["Single Coin", "Com
pare Multiple Coins"], index=0)
if comparison_mode == "Single Coin":
    selected_coin = st.sidebar.radio("Select Cryptocurrency", currencies,
index=0)
    if selected_coin:
        coin = selected_coin
        df = data[coin]
        df['Date'] = pd.to_datetime(df['Date'])
        df = df.sort_values('Date')
        st.markdown(f"<h2 style='text-align: center; font-family: Arial, s
ans-serif; color: #003366;'>Analysis for {coin}</h2>", unsafe_allow_html=T
rue)
        col1, col2 = st.columns(2)
        with col1:
            st.subheader("Price Trends")
            fig_price = px.line(df, x='Date', y=['Open', 'High', 'Low', 'C
lose'],
                                title=f"{coin} Price Trends",
                                labels={"value": "Price (USD)", "Date": "D
ate"},
                                template="seaborn")
            fig_price.update_layout(legend_title="Price Type", xaxis_title
="Date", yaxis_title="Price (USD)")
            st.plotly_chart(fig_price, use_container_width=True)
        with col2:
            st.subheader("Monthly Average Close Prices")
            df['Month'] = df['Date'].dt.to_period('M')
```

```
            monthly_avg = df.groupby('Month')['Close'].mean().reset_index(
)
            monthly_avg['Month'] = monthly_avg['Month'].dt.to_timestamp()
            fig_monthly_avg = px.bar(monthly_avg, x='Month', y='Close',
                                     title=f"{coin} Monthly Average Close
Prices",
                                     labels={"Close": "Average Close Price
(USD)", "Month": "Month"},
                                     template="seaborn")
            fig_monthly_avg.update_layout(xaxis_title="Month", yaxis_title
="Average Close Price (USD)")
            st.plotly_chart(fig_monthly_avg, use_container_width=True)
        col3, col4 = st.columns(2)
        with col3:
            st.subheader("Volume and Market Cap")
            fig_volume = px.line(df, x='Date', y=['Volume', 'Marketcap'],
                                 title=f"{coin} Volume and Market Cap",
                                 labels={"value": "USD", "Date": "Date"},
                                 template="seaborn")
            fig_volume.update_layout(legend_title="Metrics", xaxis_title="
Date", yaxis_title="USD")
            st.plotly_chart(fig_volume, use_container_width=True)
        with col4:
            st.subheader("Market Metrics Distribution")
            metrics = pd.DataFrame({"Metric": ["Open", "High", "Low", "Clo
se"],
                                    "Value": [df['Open'].iloc[-1], df['Hig
h'].iloc[-1], df['Low'].iloc[-1], df['Close'].iloc[-1]]})
            fig_pie = px.pie(metrics, names='Metric', values='Value', titl
e=f"{coin} Market Metrics Distribution", template="seaborn")
            st.plotly_chart(fig_pie, use_container_width=True)
        st.subheader("Volatility Analysis")
        df['Daily Range'] = df['High'] - df['Low']
        fig_volatility = px.line(df, x='Date', y='Daily Range',
                                    title=f"{coin} Daily Price Range (Vola
tility)",
                                    labels={"Daily Range": "Price Range (U
SD)", "Date": "Date"},
                                    template="seaborn")
        fig_volatility.update_layout(xaxis_title="Date", yaxis_title="Pric
e Range (USD)")
        st.plotly_chart(fig_volatility, use_container_width=True)
        st.subheader("Daily Percentage Change")
        df['Daily Change (%)'] = df['Close'].pct_change() * 100
        fig_daily_change = px.line(df, x='Date', y='Daily Change (%)',
                                   title=f"{coin} Daily Percentage Change"
,
                                   labels={"Daily Change (%)": "Percentage
Change (%)", "Date": "Date"},
                                   template="seaborn")
        fig_daily_change.update_layout(xaxis_title="Date", yaxis_title="Pe
rcentage Change (%)")
        st.plotly_chart(fig_daily_change, use_container_width=True)
elif comparison_mode == "Compare Multiple Coins":
```

```
    selected_currencies = st.sidebar.multiselect("Select Cryptocurrencies"
, currencies, default=[currencies[0]])
    if selected_currencies:
        combined_data = []
        for coin in selected_currencies:
            temp_df = data[coin]
            temp_df['Currency'] = coin
            combined_data.append(temp_df)
        combined_df = pd.concat(combined_data)
        combined_df['Date'] = pd.to_datetime(combined_df['Date'])
        combined_df = combined_df.sort_values('Date')
        col1, col2 = st.columns(2)
        with col1:
            st.subheader("Closing Price Comparison")
            fig_close_compare = px.line(combined_df, x='Date', y='Close',
color='Currency',
                                        title="Closing Price Comparison",
                                        labels={"Close": "Price (USD)", "D
ate": "Date"},
                                        template="seaborn")
            fig_close_compare.update_layout(xaxis_title="Date", yaxis_titl
e="Price (USD)")
            st.plotly_chart(fig_close_compare, use_container_width=True)
        with col2:
            st.subheader("Volume Comparison")
            fig_volume_compare = px.line(combined_df, x='Date', y='Volume'
, color='Currency',
                                         title="Volume Comparison",
                                         labels={"Volume": "Transaction Vo
lume", "Date": "Date"},
                                         template="seaborn")
            fig_volume_compare.update_layout(xaxis_title="Date", yaxis_tit
le="Transaction Volume")
            st.plotly_chart(fig_volume_compare, use_container_width=True)
        col3, col4 = st.columns(2)
        with col3:
            st.subheader("Market Share Distribution")
            latest_market_cap = combined_df.groupby('Currency').apply(lamb
da x: x['Marketcap'].iloc[-1]).reset_index()
            latest_market_cap.columns = ['Currency', 'Marketcap']
            fig_pie_compare = px.pie(latest_market_cap, names='Currency',
values='Marketcap',
                                     title="Market Share by Market Cap", t
emplate="seaborn")
            st.plotly_chart(fig_pie_compare, use_container_width=True)
        with col4:
            st.subheader("Market Cap Comparison")
            fig_market_cap_compare = px.line(combined_df, x='Date', y='Mar
ketcap', color='Currency',
                                             title="Market Capitalization
Comparison",
                                             labels={"Marketcap": "Market
Cap (USD)", "Date": "Date"},
                                             template="seaborn")
```

```
            fig_market_cap_compare.update_layout(xaxis_title="Date", yaxis
_title="Market Cap (USD)")
            st.plotly_chart(fig_market_cap_compare, use_container_width=Tr
ue)
        col5, col6 = st.columns(2)
        # with col5:
        st.subheader("Average Volume Per Coin")
        avg_volume = combined_df.groupby('Currency')['Volume'].mean().rese
t_index()
        fig_avg_volume = px.bar(avg_volume, x='Currency', y='Volume',
                                title="Average Volume Per Coin",
                                labels={"Volume": "Average Volume", "Curre
ncy": "Cryptocurrency"},
                                template="seaborn")
        fig_avg_volume.update_layout(xaxis_title="Cryptocurrency", yaxis_t
itle="Average Volume")
        st.plotly_chart(fig_avg_volume, use_container_width=True)
st.markdown("---")
st.markdown("<p style='text-align: center; font-family: Arial, sans-serif;
color: #003366;'>Cryptocurrency Historical Analysis Dashboard</p>", unsafe
_allow_html=True)
```