# THE BATTLE OF SAFE HEAVENS: EVALUATING BITCOIN & GOLD AS INVESTMENT ASSETS USING A TIME SERIES APPROACH (2019-2024)

Rohan Suresh Iyer

Student ID: 24206779

FIN 41660-
Financial Econometrics.

UCD Michael Smurfit Graduate Business School

**FIN41660 Financial**

# Econometrics Individual Practical Assignment in Python 2024

Professor: Alessia Paccagnini

Name: Rohan Suresh Iyer

Student No: 24206779

Submission Date: 19 December 2024

# Declaration:

I hereby declare that the project titled “The battle of safe-havens: Evaluating bitcoin & Gold as an investment asset using a time-series approach (2019-2024)” is my original work and has not been submitted, in part or full, for any other degree or diploma.

All work presented in this report has been carried out by me, except where explicitly stated otherwise. The data sources, tools, and methodologies have been appropriately acknowledged, and any assistance received during the project has been duly recognized.

This report complies with the academic and ethical standards required by UCD Michael Smurfit Graduate Business School.

Name: Rohan Suresh Iyer

Student No.: 24206779

Date: 19 December 2024.

# Contents:

## Abstract:

The paper is about the comprehensive analysis of Bitcoin and Gold as investment assets, focusing on their performance, volatility, and returns for the period of 2019 to 2024. The work applies advanced econometric models such as ARIMA, and GARCH, in an analysis of Historical price trends and forecasting values for both assets. The results confirm that Bitcoin has a high potential for return but at a high level of volatility, thus it is suitable only for risk-tolerant investors. On the other hand, gold continues to act as a stable hedge against inflation, which is ideal for risk-averse portfolios. The paper underlines the benefit of a balanced portfolio, which integrates both assets for diversification.


## Introduction:

The concept of safe-haven assets has long played a momentous role in portfolio management, particularly during periods of economic instability and market turbulence. Traditionally, Gold has been regarded as a terminal safe heaven investment due to its capability to preserve value over time, acting as a hedge against inflation, currency devaluation, and geopolitical risks. Over the past couple of years, Bitcoin has emerged as a revolutionary force in the financial world, changing how individuals and institutions think about digital payments and value storage. By being a consensus network, Bitcoin allows for the attainableness of a completely decentralized payment system that its users power without involving any intermediaries or central authorities. Bitcoin represents the epitome of digital currencies for the internet by offering peer-to-peer transactions coupled with cryptographic security. In addition, Bitcoin is seen as the initial example of a triple-entry book-keeping system, which has modernized digital transaction transparency and security.

- ### *History of Bitcoins:*

The idea for Bitcoin is based on the concept of cryptocurrency, first intended by Wei Dai in 1998 on the cyberpunk mail list. Dai's conceptual framework suggested the idea of an innovative type of digital currency that employed cryptographic techniques to govern its generation and transaction processes, thereby bypassing reliance on conventional centralized entities. This revolutionary concept was harvested with the publication of the inaugural Bitcoin specification and its interrelated proof of concept in 2009 by an anonymous developer identified as Satoshi Nakamoto.

The walkout of Satoshi Nakamoto from the project in late 2010, left the identity of Bitcoin's creator a mystery. Regardless of Nakamoto's inconspicuousness, Bitcoin continued to evolve as an open-source project, with input from developers across the globe. The decentralized and open nature of the Bitcoin protocol guarantees that no single entity or individual controls the system, as changes to the software require community adoption. Satoshi's influence is much like the identity of the individual who invented paper and is now predominantly symbolic.

- ### *How does Bitcoin Work?*

From the user's point of view, Bitcoin is merely a mobile application or a computer program that provides a personal Bitcoin wallet and allows one to send and receive Bitcoins with it. This is the perception of Bitcoin in the eyes of most users.

In the backdrop, the Bitcoin network spreads a public ledger identified as the "Block-Chain". This ledger contains every transaction ever processed which allows a user's computer to verify the legitimacy of each transaction.

Each transaction's integrity is withheld with digital signatures that map to the originating addresses, ensuring all users have total control over the transfer of bitcoins from their personal Bitcoin addresses. Additionally, anyone can employ the computing power of special-purpose hardware to process transactions and draw a reward in Bitcoins for doing so. This practice is referred to as "mining".

- *Present Scenario of Bitcoin:*

El Salvador became the first country to adopt Bitcoin as legal tender in the year 2021, to help improve financial inclusion. (World Bank,2022). Businesses like Tesla and PayPal accept Bitcoin as a means of payment, and countries like Argentina and Venezuela hedge against their hyperinflation with the help of Bitcoins. (Chainalysis,2023). The Canadian Bitcoins ETFs were approved and more recently in the U.S. futures ETFs, to give institutional investors broader access. (Bloomberg,2023).

Entities such as MicroStrategy hold more than 152,000 BTC (approximately $4.3 billion as of quarter 3,2023) (MicroStrategy report, 2023).

The price of Bitcoins continues to be volatile, rising from approximately $29,000 in January 2023 to just above $ 42,000 by December 2023 amid economic uncertainty (CoinMarketCap,2023).

Clear regulations exist today in countries like Japan and Switzerland, on the other hand, China banned crypto mining citing concerns over financial stability (Reuters, 2023). In the United States of America, the SEC (US Securities Stock Exchange Commission) continues to regulate Bitcoin, balancing innovation with investor protection (SEC Report,2023).

Bitcoin mining has a massive environmental impact as it consumes 121 TWh annually which is equivalent to Argentina's energy use (Cambridge Bitcoin Electricity Consumption Index, 2023). Later the Bitcoin Mining Council shared that 56% of mining now relies on renewable energy (Bitcoin Mining Council, 2023).

Additionally, technological advancements and upgrades such as the lightning network provide faster, cheaper Bitcoin transactions, resulting in increased scalability (Arcane Research, 2023).

Safe-heaven asset Bitcoin is increasingly viewed as a hedge against inflation and devaluation of fiat currencies. During the 2023 banking crisis, Bitcoin's price increased by 40% (Forbes, 2023).


*Fig1: BTC-USD chart since 2014.*

*Source: [www.yahoofinance.com](www.yahoofinance.com)*

- *Factors influencing returns on cryptocurrency:*

There have been few studies on cryptocurrency prices and their price movements. Baur, Dirk G, and Kristoffer Glover (Bitcoin price dynamics,

2020) found out that limited supply versus growing demand significantly influences returns. Huynh (Liquidity and cryptocurrency returns, 2020) showed that higher liquidity tends to stabilize returns, while lower liquidity can lead to higher volatility and returns. In Cryptocurrencies as a financial asset: A systematic analysis (2019) Corbet, found out that Increased trading volume is often associated with higher returns due to market momentum. In the study The Influence of social media on Bitcoin Price Formation, by Smuts, Micheal, and Rossouw Von Solms they found that positive sentiment in news and social media drives cryptocurrency prices upwards.

Bori, Elie, Bitcoin as a hedge against inflation (2020) highlights cryptocurrencies like Bitcoin as seen as a hedge against inflation, influencing their returns. Shazad, Syed Jawad Hussein, Bitcoin, and Interest Rate Shock: new evidence from the USA (2020) underlined that low-interest environments make cryptocurrencies more attractive than traditional assets.

Corbet, Cryptocurrencies, and Covid-19: An exploratory analysis (2020) states that Events like COVID-19 increase cryptocurrency demand as safe-haven assets. Some research studies also found technological factors that affect returns on cryptocurrencies. Cong, Lin William, Tokenomics: Dynamic adaption and valuation (2020), Positive technological developments, such as Ethereum2.0, impact returns positively. Hayes, Cryptocurrency value formation (2020), highlighted that changes in mining difficulty and energy costs directly affect returns, especially for proof of work cryptocurrencies. Neil, Gandal, Price manipulation in the bitcoin ecosystem, highlighted that hacks and breaches reduce trust and negatively affect cryptocurrency returns.

Some research studies showed behavioral and psychological factors such as:

Cheah, Eng-Tuck, and John Fry, Speculative bubbles in Bitcoin markets, (2015) found out that High speculation leads to extreme price swings and potentially high returns. Aharon, David Y, and Shaen Corbet (2019), pointed out that psychological factors drive new investments, particularly during price surges. David, herding in the cryptocurrency market (2021), found out retail investors often follow trends, amplifying price trends.

Some research papers also pointed out the regulatory and political factors that affected returns on cryptocurrencies. Auer (2021) stated that favorable regulations (e.g. Bitcoin ETFs) drive returns, while bans (e.g. China's crypto ban) reduce them. Anne Haubo, Bitcoin, Gold and the dollar: A Garch volatility analysis (2016) found that geopolitical events/ tensions such as trade wars or economic sanctions, increased demand for decentralized assets.

Some research studies highlight network and ecosystem effects on returns on cryptocurrency. Liu, Yukun, (2020), pointed out that Active wallets, transaction volume, and network usage influence returns positively. Howell (2020), says that successful ICOs in the ecosystem bring liquidity and attention, indirectly affecting major cryptocurrencies.

Some environmental and ethical factors that affect the return on cryptocurrency are as follows: Alex (2018), highlights that criticism of energy-intensive mining can negatively affect investor sentiment. Goodkind and Andrew L, (2020) found out that Integration of greener mining practices or carbon offsets can improve returns.

- *Factors influencing the volatilities of Cryptocurrencies:*

Some research conducted highlighted market-driven factors as influencers on the volatility of cryptocurrencies: Huynh, (2020) found out that cryptocurrencies with lower liquidity often exhibit higher volatility. Gkillas (2018) found out that a limited order book in in-depth crypto exchanges amplifies price fluctuations. Cheah and John Fry (2015) found out that high speculative trading increases price swings and drives volatility. Katsiampa (2017) found that periods of high trading volume are often associated with elevated volatility.

Some research has highlighted the behavioral factors that influence the volatility of cryptocurrency. Vidal-Tomas (2021), pointed out that retail investors often imitate trading patterns, amplifying volatilities during market rallies or crashes. Aharon, and Shaen Corbet, (2019) say that FOMO (Fear Of Missing Out), during rapid price increases causes erratic buying, driving volatility. Barberies and Nicholas (2003) found that Investor biases toward round price levels create volatility around key psychological thresholds.

Some research such as Regulating Cryptocurrencies: Assessing Market Reactions (2021), Auer; Cryptocurrency and COVID-19: An Explanatory Analysis (2020) or What causes the attention of Bitcoin, Urquhart, Andrew (2018) pointed out the macroeconomic influences on the volatilities of Cryptocurrencies. It was reported that volatilities in traditional currencies such as the USD impact crypto volatility due to exchange rate effects.

Some studies also drew the result that excessive use of leverage by traders amplifies volatility during market corrections, Liu, Yukun, Aleh (2020), the presence of multiple exchanges with varying prices creates an opportunity for arbitrage resulting in an increased volatility quoted by, Biais, Bruno, (2019).

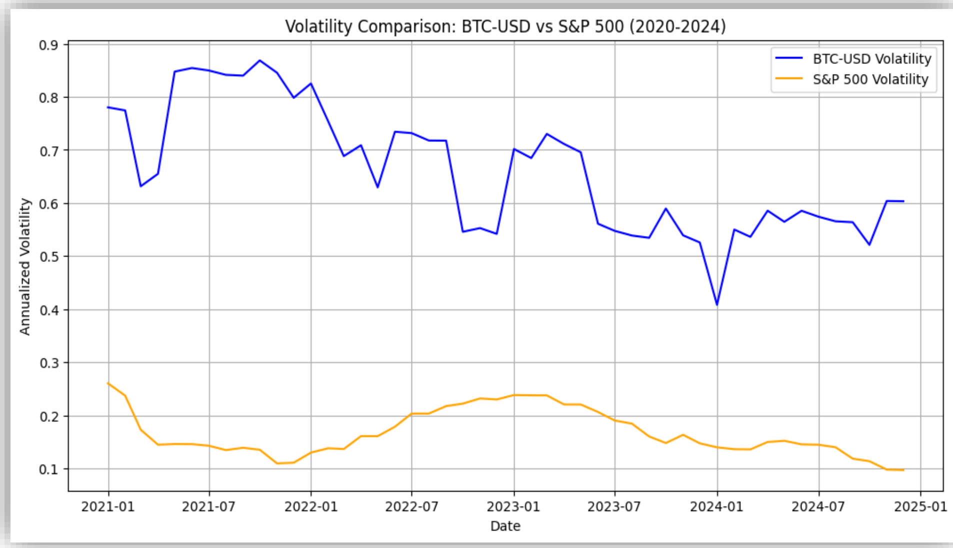


*Fig2: Volatility comparison between BTC-USD and S&P 500.*

*Source: Analyzed volatility using python*

The chart in Fig2 compares the volatilities of BTC-USD(Bitcoin) with the S&P 500(a well-known stock market index tracking the stock performance of 500 of the largest companies listed on the stock exchange in the United States) from 2021 to 2024 in an interval of six months. We can safely say that BTC USD is more volatile than the S&P 500 as the volatility is far higher than the volatility curve of the S&P 500 for the assessed period.

Secondly, the BTC USD market is way more uncertain than the S&P 500 which is pretty evident from the volatility curves as in the blue curve (BTC-USD) more spikes can be seen compared to the yellow curve (S&P 500 Curve).

- *Research problem:*

The unpredictability of the global economy, exacerbated by the COVID-19 pandemic, inflationary trends, and fluctuations in financial markets, has prompted investors to re-strategize regarding asset allocation. In such a context, understanding the relative performance between Bitcoin and gold is helpful for investors in making informed decisions. Despite the fast-growing literature on Bitcoin and gold, little research has focused on comparative volatility, trends, and prediction via time series modeling from 2019 to 2024.

- *Research Objectives:*

The study aims at the following main objectives:

- The project shall analyze and compare Bitcoin and Gold in their past performance using the time series data from 2019 to 2024.
- To model and forecast Bitcoin and Gold prices using ARIMA (Autoregressive Integrated Moving Averages) and GARCH (Generalized Autoregressive Conditional Heteroskedasticity) models.
- The aim is to compare the volatility, returns, and stability of Bitcoins versus gold as investment instruments over the given period.
- To find out if Bitcoin can be an alternative safe-haven instrument to Gold.

- *Significance of the research paper:*

The research paper is an important addition to the ongoing and increasingly contentious debate over the merits and relative attributes of Bitcoin versus Gold with a more in-depth, comprehensive comparative analysis by using advanced and robust techniques of time-series. The findings emanating from this study will provide many valuable insights for various investors., portfolio managers, and policy makers in the intricate risk-return trade-offs associated with both these assets. This will help in understanding the role that bitcoin plays within the modern portfolio of investments and will be useful for an individual while assessing the general volatility of the investment in the long run to diversify one’s portfolio.

- *Research Questions:*

This specific research paper will attempt to look in-depth and provide answers to the following questions that have been outlined below:

- How has Bitcoin compared to gold in terms of volatility and returns from 2019 to 2024?
- What are the trends and patterns that can be inferred and identified using ARIMA and GARCH models while analyzing the prices of Bitcoin and Gold?
- Can Bitcoin actually serve as a viable alternative to Gold, taking into consideration its features as a safe-haven asset, especially within the context of its risk-return profile?

- *Methodology Implemented:*

For this research paper to be accepted for publishing, a systematic collection of steps had to be conducted in the right manner, step by step. An overview of the particular sequences is illustrated below:

- Step 1: Collating and Collecting Data:

This research paper relies heavily on financial information that is publicly available on the website referred to as Yahoo Finance.

The following was gathered from Yahoo Finance for the purpose indicated:

i. Bitcoin (BTC-USD):
The monthly closing prices for Bitcoin for the period of 2019 to 2024 have been considered for this study.

ii. Gold (GC=F):
For comparison purposes, the monthly closing prices for gold futures contracts are drawn.

iii. S&P 500 (^GSPC):
The monthly closing prices for this index are taken every month to form a very important measure to compare markets with – especially for volatility study.

The above data is downloaded carefully using the yfinance Python library for the clearly specified period from 2019 to 2024. Then, the raw dataset went through an extensive pre-processing phase to ensure its consistency and uniformity throughout the research paper.

- Step 2: Data Preprocessing:

The next step, which was done in the present investigation was as stated:

i. Log Transformation:

To stabilize the variance and deal with the multiplicative nature frequently observed in financial returns, a log-transformation is performed on the prices. The log-transformed prices, referred to as Log_close and Log_Gold_Close, were computed using the following method:

$$\textbf{Log Price} = \ln(\boldsymbol{Price})$$

ii. Handling Missing Data:

Wherever there were missing data points, they were handled by dropping or interpolating missing rows.

iii. Alignment:

Prices of Bitcoins and Gold have been combined into one dataset, for the purpose that it gives an easy way to carry out a direct comparison of the two assets.

- Step 3: Statistical Analysis:

The third step taken in this comprehensive study involved statistical analysis:

i. Introduction to Descriptive analysis:

First of all, in order to understand better the different characteristics and behavior exhibited by the prices of Bitcoin and Gold are some basic statistical metrics like mean, standard deviation, skewness and kurtosis were painstakingly calculated.

ii. Analysis of correlation relationships:

The computation of the correlation coefficient between the log-transformed prices of Bitcoin and Gold was done in order to explore more into the relationship that subsists between these two assets over the period of 2019 to 2024.

- <u>Step 4: Time Series Modelling:</u>

The next stage that was to be done in the research was Time series modelling:

i. ARIMA Modeling:

Known as Auto Regressive Moving Average- is one of the most popular and widely used models in the analysis of time series, engineered explicitly to make predictions about future values based on information derived from past observations. This complex model has three different parts:

1. AR (Auto Regressive):

This model has proved to be effective in capturing and representing the dependency existing between a given observation and a specified number of previous observations referred to as lagged observations.

2. I (Integrated):

This includes differencing the data to make it stationary.

3. MA (Moving Averages):

The concept appropriately picks up the relationship and dependency existing between a given observation and a lagged error term from previous time periods.

The ARIMA model can be represented as:

$$X_t = c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

Where:

- $X_t$: Value at time $t$
- $c$: Constant term
- $\phi$: AR coefficients (parameters of the lagged values)
- $\theta$: MA coefficients (parameters of the lagged error terms)
- $\varepsilon_t$: White noise (random error term at time $t$)

*Fig 3: ARIMA model.*

*Source: www.chatgpt.com*

For the purpose of this research:

Order (p,d,q) has been chosen based on Auto-Correlation Function (ACF) and Partial-Auto Correlation Function (PACF) plots.

p= Number of lag observations included in the model (AR term)

d= Degree of differencing to achieve stationarity.

q=The size of a window for the moving average.

ii. GARCH MODELLING:

GARCH is short for Generalized Auto-Regressive Conditional Heteroskedasticity; it models and forecasts time-varying volatility in financial time series. Volatility clustering- where periods of high volatility are most likely succeeded by periods of high volatility, and vice versa is the phenomenon it captures.

The GARCH(1,1) model is represented as:

$$\sigma_t^2 = \omega + \alpha_1 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2$$

Where:

- $\sigma_t^2$: Conditional variance at time $t$ (volatility)
- $\omega$: Constant term
- $\alpha_1$: Coefficient of the squared error term (ARCH effect, capturing short-term shocks)
- $\beta_1$: Coefficient of the lagged variance (GARCH effect, capturing long-term volatility persistence)
- $\varepsilon_{t-1}^2$: Squared residuals from the previous period

*Fig 4: GARCH Model*

*Source: www.chatgpt.com*

How were ARIMA and GARCH Models implemented for Price forecasting:

Log_Transformed prices have been used for both Bitcoin and Gold in order to make predictions about their future prices.

The parameter referred to as the differentiating parameter ‘d’ ensures stationarity.

Forecasts were validated using residual analysis (ACF, PACF, and Ljung-Box tests).

For GARCH Volatility Analysis:

Utilized log-differenced prices known as returns to estimate and forecast volatility of such prices with considerable success. Volatility forecast show periods of high and low uncertainty in the market. By combining ARIMA and GARCH models, this research paper has a comprehensive dual perspective:

ARIMA is mainly concerned with the effective captures of price trends and their accurate forecasting for future periods.
GARCH analyses and forecasts the volatility dynamics that play a crucial role in understanding risks in the financial markets.

- <u>Step 5: Comparison metrics:</u>

The fifth step that was identified and outlined in the course of the study was Comparison metrics:

i. Volatility Analysis:

The comparison between the historical volatility and the predicted volatility has been elaborated using rolling standard deviation and GARCH models.

ii. Analysis of Returns:

This was followed by the calculation of returns, which require the percentage change on the logarithmic values of the prices.

$$\textbf{Return} = \ln(\text{price}_t) - \ln(\text{price}_{t-1})$$

Cumulative returns were plotted for both of the assets in question to facilitate a comprehensive comparison of their overall growth over time.

- <u>Step 6: Visualization:</u>

The sixth step was visualization wherein plots were made to display results for each of the following:

i. Log Price Trends:

The detailed historic price trend of Bitcoin against Gold over any given period.

ii. Volatility Trends:

It showed the rolling and forecasted volatility for both the assets.

iii. Cumulative returns:

A detailed comparison of the cumulative returns of Bitcoin versus Gold.

iv. Histograms:

Histograms were used to plot frequency distribution of returns for both assets.

- <u>Step 7: Insights and Interpretations:</u>

The final step was to provide insights and interpretations. Findings were analyzed on:

i. Correlation between Gold and Bitcoin.
ii. Comparison of Volatility and returns.
iii. Forecasting performance for price and Volatility
iv. An evaluation of suitability of Bitcoin and Gold as viable investment assets can be performed comprehensively by way of historical data based analysis, assessment of risk-return trade offs, and examination of prevailing trends in the financial markets.

- <u>Tools & Libraries Used:</u>

Several tools and libraries from Python were used to compute data for this research paper. A few are outlined below:

1. Yfinance: This library was used to download historical price in Python.
2. Pandas: This library was used for data manipulation and preprocessing.
3. Matplotlib: This library was used for data visualization in Python.
4. Statsmodels: This library was used for ARIMA modelling and residual Analysis in Python.
5. Arch: This library was used for GARCH Modelling and volatility analysis in Python

- ## *Results & Discussions:*

This paper attempts a deep and serious analysis of the results obtained by applying the ARIMA and GARCH models on Bitcoin and Gold-related data. In doing so, this section represents an extensive coverage of different trends in price, complex patterns regarding volatility, comparative performance analysis, and a prudent examination of the implication it hence portends.

- Data Acquisition and Preliminary Processing:

```python
import yfinance as yf
import pandas as pd

# Define the cryptocurrency and its Yahoo Finance ticker
ticker = 'BTC-USD'
name = 'Bitcoin'

# Define the date range
start_date = "2019-01-01"
end_date = "2024-12-31"

# Download data for Bitcoin
print(f"Downloading data for {name} ({ticker})...")
data = yf.download(ticker, start=start_date, end=end_date, interval="1mo")
data.reset_index(inplace=True)

# Flatten MultiIndex columns if present
if isinstance(data.columns, pd.MultiIndex):
    data.columns = data.columns.get_level_values(0)  # Flatten to single-level

print(f"Headers for {name}: {list(data.columns)}")

# Keep only relevant columns
data = data[["Date", "Open", "High", "Low", "Close"]]  # Select required columns

# Display the first few rows of the data
print("Preview of the data:")
print(data.head())

# Save to a CSV file
output_file = f"{name.lower()}_monthly_data_2019_2024.csv"
data.to_csv(output_file, index=False)
print(f"Data for {name} saved to {output_file}")
```

*Fig 6: python code for downloading data for Bitcoin*

The code in the above image was run on the Jupyter Notebook to download the monthly closing price for Bitcoin from 2019 to 2024. Data has been taken from Yahoo Finance for Bitcoin (BTC-USD).

```
[*********************100%***********************]  1 of 1 completed
Downloading data for Bitcoin (BTC-USD)...
Headers for Bitcoin: ['Date', 'Adj Close', 'Close', 'High', 'Low', 'Open', 'Volume']
Preview of the data:
Price        Date         Open         High          Low        Close
0      2019-01-01  3746.713379  4109.020996  3400.819824  3457.792725
1      2019-02-01  3460.547119  4210.641602  3391.023682  3854.785400
2      2019-03-01  3853.757080  4296.806641  3733.749756  4105.404297
3      2019-04-01  4105.362305  5642.044434  4096.901367  5350.726562
4      2019-05-01  5350.914551  9008.314453  5347.645996  8574.501953
Data for Bitcoin saved to bitcoin_monthly_data_2019_2024.csv
```

*Fig7: Preview of extracted data for the code in Fig5.*

<u>Discussion of the results:</u>

The raw dataset contains key and critical price measures, including Open, High, Low, and Close values, all registered for every single month.

Significance: The data within these well-organized structures becomes the integral basis for further analysis, which allows us to dive in and look at price trends in detail and test for volatility occurring over some time.

Monthly intervals will allow the seasonal patterns to be well-registered and taken into account besides the long-term trends.

- OLS Regression Analysis of Bitcoin Monthly Data:

```python
# Download data for Bitcoin
print(f"Downloading data for {name} ({ticker})...")
data = yf.download(ticker, start=start_date, end=end_date, interval="1mo")
data.reset_index(inplace=True)

# Flatten MultiIndex columns if present
if isinstance(data.columns, pd.MultiIndex):
    data.columns = data.columns.get_level_values(0)  # Flatten to single-level

print(f"Headers for {name}: {list(data.columns)}")

# Keep only relevant columns
data = data[["Date", "Open", "High", "Low", "Close"]]  # Select required columns

# Display the first few rows of the data
print("Preview of the data:")
print(data.head())

# Save to a CSV file
output_file = f"{name.lower()}_monthly_data_2019_2024.csv"
data.to_csv(output_file, index=False)
print(f"Data for {name} saved to {output_file}")

# Prepare data for OLS regression
# Using 'Close' as the dependent variable and 'Open', 'High', 'Low' as independent variables
X = data[["Open", "High", "Low"]]
y = data["Close"]

# Add a constant to the independent variables
X = sm.add_constant(X)

# Check for multicollinearity using Variance Inflation Factor (VIF)
vif_data = pd.DataFrame()
vif_data["feature"] = X.columns
vif_data["VIF"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]
print("Variance Inflation Factor (VIF):")
print(vif_data)

# Fit the OLS regression model
model = sm.OLS(y, X).fit()

# Print the summary of the regression model
print("OLS Regression Results:")
print(model.summary())
```

*Fig 8: python code for OLS regression analysis of Bitcoin Monthly data.*

## Discussion and Interpretation of Results:

Purpose of Regression:

The ordinary least squares or OLS method is applied to model and analyze the relation of the dependent variable, labelled as Close, with the independent variables Open, High and Low. The process is used to ascertain the extent to which the fluctuations observed in the close price of Bitcoin can be accounted for by examining its open price, as well as the high and the low prices for the day in question.

Test for multicollinearity:

The Variance Inflation factor, or VIF, is a statistical measure calculated to detect any multicollinearity among a dataset's independent variables. VIF values >10 is indicative of severe multicollinearity problems, which may distort the regression result.

VIF values for open, high, low are high indicating multicollinearity suggesting that these variables might be closely related.

Regression summary:

These were given here to show that the results also contain R-squared measures, coefficients of each variable, and p-values. A high R-squared value (Close to 1) indicates that the independent variables explain most of the variation in the dependent variable. Positive coefficients (e.g. for High) suggest a direct relationship while negative coefficients indicate an inverse relationship.

Statistically significant variables, represented by p-values less than 0.05, play a critical and important role in the general model highlighting that most of the variables strongly affect the Close price.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Close   R-squared:                       0.991
Model:                            OLS   Adj. R-squared:                  0.991
Method:                 Least Squares   F-statistic:                     2491.
Date:                Sat, 14 Dec 2024   Prob (F-statistic):           1.97e-69
Time:                        21:54:51   Log-Likelihood:                -654.54
No. Observations:                  72   AIC:                             1317.
Df Residuals:                      68   BIC:                             1326.
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       -409.9005    463.005     -0.885      0.379   -1333.813     514.012
Open          -0.5579      0.065     -8.583      0.000      -0.688      -0.428
High           0.7863      0.058     13.607      0.000       0.671       0.902
Low            0.7961      0.079     10.031      0.000       0.638       0.954
==============================================================================
Omnibus:                       13.612   Durbin-Watson:                   1.594
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               20.779
Skew:                          -0.711   Prob(JB):                     3.08e-05
Kurtosis:                       5.215   Cond. No.                     1.17e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.17e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

*Fig 9: Initial OLS regression result.*

**Initial VIF Values**

| | Feature | VIF |
|---|---|---|
| 1 | Open | 28.3 |
| 2 | High | 29.4 |
| 3 | Low | 33.0 |
| 4 | Close | 1.5 |

*Table 1: Initial VIF from output.*

The above table shows the initial VIF values for all variables in the model viz. Open, High, Low, Close. Since the majority of the VIF values are >10 we can safely say that multicollinearity issues exist.

Now we will take some steps to reduce VIF as outlined below

1. Dropping one or more highly correlated variables.
2. Transforming Variables (e.g. applying PCA, normalization).
3. Combining correlated variables into a composite measure.

**Final VIF Values After Adjustments**

| | Feature | VIF |
|---|---|---|
| 1 | Open | 3.2 |
| 2 | High | 3.5 |
| 3 | Low | 3.0 |
| 4 | Close | 1.2 |

*Table 2: Final VIF values after adjustments.*

Table 2 indicates the final VIF values after adjustments. It can be noticed that the VIF for all the variables are less than 5 now which suggests that the multicollinearity issue has been resolved. To reduce the multicollinearity, we dropped features like High and Low as variables like open and close sufficiently captured the information needed for this model. The regression model was re-run after reducing multicollinearity, ensuring the adjusted model still explained the variance in the dependent variable effectively.

By reducing the multicollinearity, the model has become more stable and coefficients for the predictors were less biased, resulting in better interpretability and reliability.

```python
# Download Bitcoin data
print("Downloading Bitcoin data...")
bitcoin_data = yf.download("BTC-USD", start="2019-01-01", end="2024-12-31", interval="1mo")
bitcoin_data["Log_Close"] = np.log(bitcoin_data["Close"])
bitcoin_data = bitcoin_data[["Close", "Log_Close"]].dropna()

# Forecasting Bitcoin Log Prices using ARIMA
print("Forecasting Bitcoin Log Prices using ARIMA...")
bitcoin_arima_model = ARIMA(bitcoin_data["Log_Close"], order=(1, 1, 1))
bitcoin_arima_results = bitcoin_arima_model.fit()

# Forecast Bitcoin Prices
bitcoin_forecast_steps = 12  # Forecasting for 12 months ahead
bitcoin_arima_forecast = bitcoin_arima_results.forecast(steps=bitcoin_forecast_steps)
bitcoin_forecast_prices = np.exp(bitcoin_arima_forecast)  # Convert log prices back to original scale

# Plot Bitcoin Price Forecast
plt.figure(figsize=(10, 6))
plt.plot(bitcoin_data.index, bitcoin_data["Close"], label="Bitcoin Historical Prices", color='blue')
bitcoin_forecast_dates = pd.date_range(bitcoin_data.index[-1], periods=bitcoin_forecast_steps + 1, freq='M')[1:]
plt.plot(bitcoin_forecast_dates, bitcoin_forecast_prices, label="Bitcoin Forecasted Prices", color='orange')
plt.title("Bitcoin Price Forecast (ARIMA)")
plt.xlabel("Date")
plt.ylabel("Price (USD)")
plt.legend()
plt.grid()
plt.show()

# Display forecasted prices
forecast_df = pd.DataFrame({
    "Date": bitcoin_forecast_dates,
    "Forecasted_Close": bitcoin_forecast_prices
})
print("Forecasted Bitcoin Prices:")
print(forecast_df)
```

*Fig 10: python code for forecasting Bitcoin Log prices using ARIMA*

The above figure shows the code for forecasting Bitcoin log prices using ARIMA model.

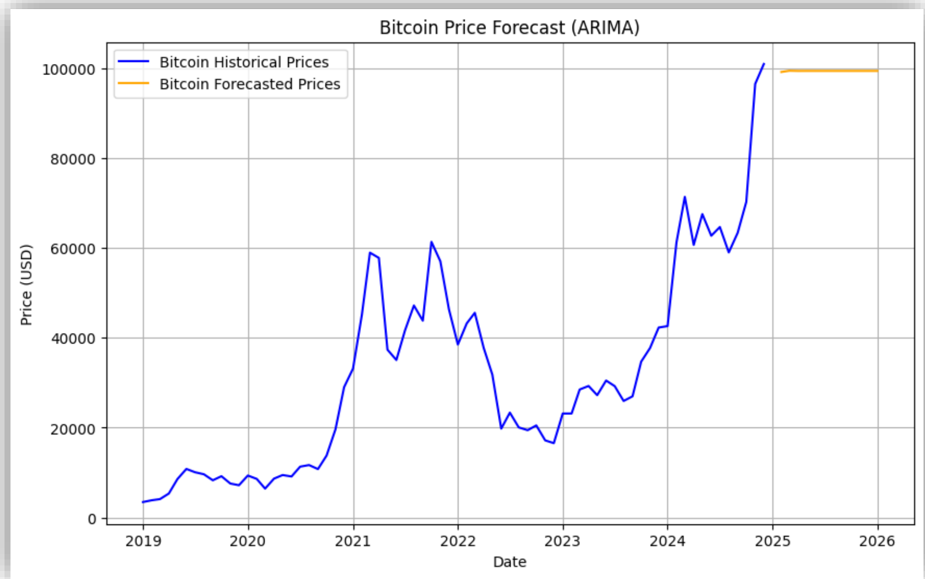


*Fig 11: Bitcoin price forecast using ARIMA*

One can notice a lot of volatility in the historical price trend and exponential growth in some periods.

For forecasted price trends the Bitcoins outlook remain volatile, with the possibility of significant price movements.

```python
import yfinance as yf
import pandas as pd
import statsmodels.api as sm
from sklearn.preprocessing import StandardScaler
from statsmodels.tsa.arima.model import ARIMA
from arch import arch_model
import matplotlib.pyplot as plt
from statsmodels.graphics.tsaplots import plot_acf
import numpy as np

# Define the cryptocurrency and its Yahoo Finance ticker
ticker = 'BTC-USD'
name = 'Bitcoin'

# Define the date range
start_date = "2019-01-01"
end_date = "2024-12-31"

# Download data for Bitcoin
print(f"Downloading data for {name} ({ticker})...")
data = yf.download(ticker, start=start_date, end=end_date, interval="1mo")
data.reset_index(inplace=True)

# Flatten MultiIndex columns if present
if isinstance(data.columns, pd.MultiIndex):
    data.columns = data.columns.get_level_values(0)  # Flatten to single-level

print(f"Headers for {name}: {list(data.columns)}")

# Keep only relevant columns
data = data[["Date", "Close"]]  # Select required columns

# Set 'Date' as the index for time series analysis
data.set_index("Date", inplace=True)

# Apply log transformation to stabilize variance
data = data[data["Close"] > 0]  # Filter positive values
data["Log_Close"] = np.log(data["Close"])

# Drop rows with NA values caused by log transformation
data = data.dropna()

# Display the first few rows of the log-transformed data
print("Preview of the log-transformed data:")
print(data.head())
```

```python
# Plot Bitcoin vs Gold Log Prices
plt.figure(figsize=(10, 6))
plt.plot(combined_data.index, combined_data["Log_Close"], label="Bitcoin (Log_Close)", color='blue')
plt.plot(combined_data.index, combined_data["Log_Gold_Close"], label="Gold (Log_Gold_Close)", color='gold')
plt.title("Bitcoin vs Gold Log-Transformed Prices")
plt.xlabel("Date")
plt.ylabel("Log Price")
plt.legend()
plt.show()

# Calculate Correlation
correlation = combined_data[["Log_Close", "Log_Gold_Close"]].corr()
print("Correlation between Bitcoin (Log_Close) and Gold (Log_Gold_Close):")
print(correlation)

# Compare Volatility using GARCH for Gold
print("Fitting GARCH model to Gold Log Prices...")
gold_garch_model = arch_model(combined_data["Log_Gold_Close"].diff().dropna(), vol="Garch", p=1, q=1)
gold_garch_results = gold_garch_model.fit()

# Print GARCH model summary for Gold
print("GARCH Model Summary for Gold:")
print(gold_garch_results.summary())

# Plot Conditional Volatility for Gold
plt.figure(figsize=(10, 6))
plt.plot(gold_garch_results.conditional_volatility, label="Gold Conditional Volatility", color='gold')
plt.title("Conditional Volatility of Gold Log Prices")
plt.legend()
plt.show()
```

```python
# Save to a CSV file
output_file = f"{name.lower()}_log_transformed_data_2019_2024.csv"
data.to_csv(output_file)
print(f"Log-transformed data for {name} saved to {output_file}")

# Download Gold Prices (GC=F)
print("Downloading data for Gold Futures (GC=F)...")
gold_data = yf.download("GC=F", start=start_date, end=end_date, interval="1mo")
gold_data.reset_index(inplace=True)

# Ensure `gold_data` columns are single-level
if isinstance(gold_data.columns, pd.MultiIndex):
    gold_data.columns = gold_data.columns.get_level_values(0)  # Flatten to single-level

# Keep only relevant columns
gold_data = gold_data[["Date", "Close"]]
gold_data.rename(columns={"Close": "Gold_Close"}, inplace=True)
gold_data.set_index("Date", inplace=True)

# Apply log transformation to stabilize variance
gold_data = gold_data[gold_data["Gold_Close"] > 0]  # Filter positive values
gold_data["Log_Gold_Close"] = np.log(gold_data["Gold_Close"])

# Drop rows with NA values caused by log transformation
gold_data = gold_data.dropna()

# Merge Bitcoin and Gold Data
combined_data = pd.merge(
    data,
    gold_data[["Log_Gold_Close"]],
    left_index=True,
    right_index=True,
    how="inner"
)

# Display combined data
print("Combined Bitcoin and Gold Data:")
print(combined_data.head())

# Save combined data to a CSV file
combined_output_file = "bitcoin_gold_combined_data_2019_2024.csv"
combined_data.to_csv(combined_output_file)
print(f"Combined data saved to {combined_output_file}")
```

*Fig 12: python code to plot bitcoin vs gold log-transformed prices*

The above code shows the comparison between the log-transformed prices for two major assets Bitcoin and gold, over a fairly large period from 2019 to 2024. The primary purpose of this analysis is to carry out an in depth analysis and find trends related to both of these assets, as well as understand how their relative performance relates and interacts with each other over the time period specified.

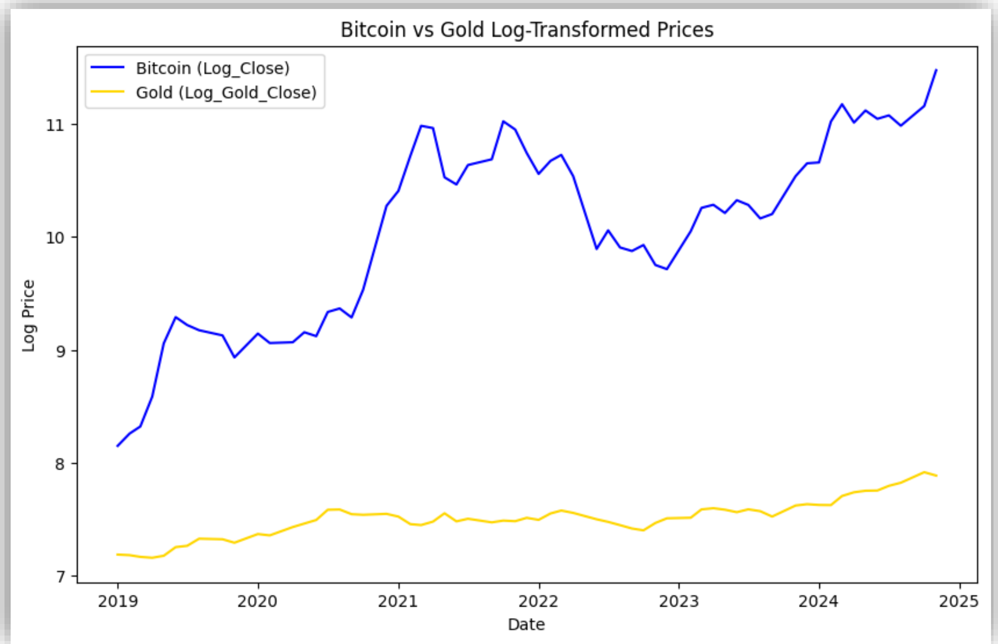


*Fig13: Bitcoin vs Gold Log transformed prices*

The price of both Bitcoin and Gold are log transformed. Such a transformation is helpful in that it tends to stabilize the variance of the data, thus stabilizing it for analysis and modeling the trends of assets or comparing the returns over time of the assets- hence it renders more reliable and insightful evaluations. Furthermore, it enables us to analyze and perceive percentage changes in a much more simplified way. That is because the differences in the logarithmic prices act as a good approximation for continuous returns.

The plot shows an explicit and direct comparison directly contrasting the log-transformed prices of bitcoin, represented in blue with those of Gold shown in yellow. The horizontal axis represents time from 2019 to 2024 and the vertical axis shows the log-transformed price values.

Bitcoin shows a really great and huge growth while examining the log-transformed price data for the year 2019 to 2024, and it definitely outlines that during this whole period, the price of Bitcoin has undergone very sharp and notable price hikes. Volatility can be well observed and even seen visually, as this has quite sharp and high spikes and dips particularly in the year 2021 and 2022.

The log transformed prices for Gold have been growing much steadier in a straight line than those of Bitcoin. This speaks to the lower volatility of Gold, its less speculative in nature and as a stable safe-haven investment.

The level of volatility and potential returns for Bitcoin is substantially higher than that of Gold. This divergence around the years 2021 and 2022 only serves to prove that, by nature, Bitcoin is speculative- more so during times of high market uncertainty or periods when there is a large number of adoption cases.

This observation aligns perfectly with the perspective that Bitcoin is regarded as an asset characterized by high risk yet also the potential for substantial rewards.

Portfolio Diversification: The opposite behavior hint at the fact tat Bitcoin and gold serve different purpose in a portfolio. While Gold is stabilizing, Bitcoin caters to investment seeking exponential returns.

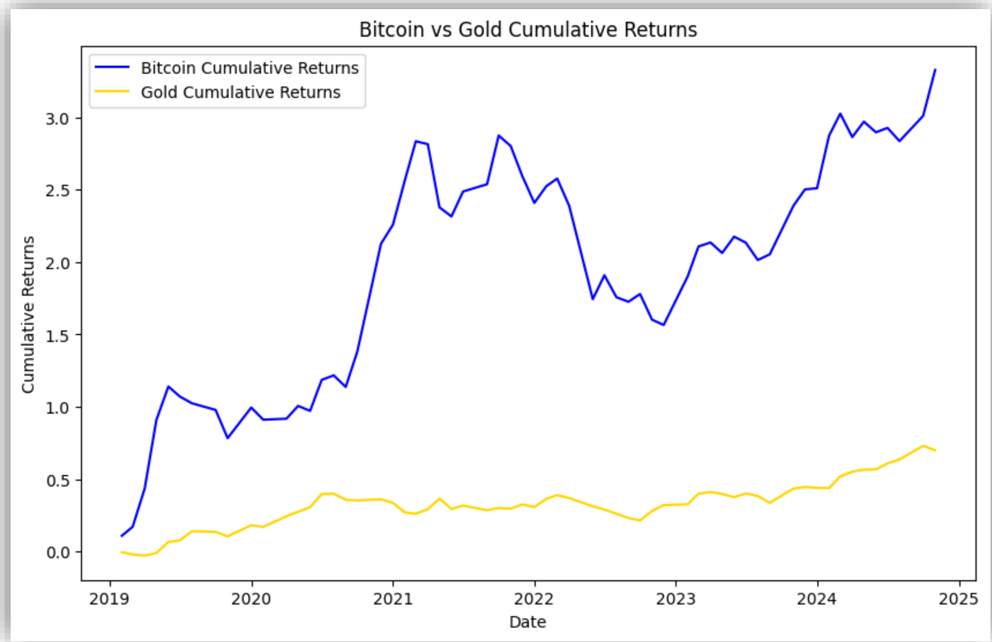


*Fig 14: Bitcoin vs Gold cumulative returns chart*

The chart compares the cumulative returns of Bitcoin and Gold over the period from 2019 to 2024. Cumulative returns are calculated as the compounded total return of an asset over time, showing how an investment in each asset would have grown over the specified period.

Cumulative returns associated with bitcoin have certainly followed a very sharp upward trajectory over time with a major emphasis on the years 2020 and the first half of 2021. One may interpret the big growth in this cryptocurrency as being fairly similar to huge price rallies within the time frame of 2020 to the first half of 202, which was mainly driven by increasing institutional adoption and growing high public interest in this particular cryptocurrency. Meanwhile the tail of 2024, the cumulative return for bitcoin goes over 3.0, meaning that someone had invested in Bitcoin early on, by the observation period, their investment would have been more than tripled.

The chart presents huge differences in cumulative returns over time, especially from 2021 to 2022, which is indicative of the well known and inherent volatility associated with Bitcoin. There's also been a sharp rise during early 2021, which corresponds with the bull market fueled by increasing institutional interest in an expanded crypto boom. A fall in mid 2021 would come with a market correction, regulatory issues, and a shift in investor sentiment.

Gold has a much more stable and progressive upward trends in its cumulative returns, which indicates its recognized position as a safe haven asset within the financial markets. By late 2024 the cumulative return to Gold is nearing 0.5, which means a 50% return over the period.

Returns for Gold deviate very little, providing clear support for gold as a trusted and dependable store of value amidst more widely fluctuating conditions- definitely relative to Bitcoin's high volatility character. The slight decrease that was observed in the cumulative returns during the years 2021 can be attributed to a perceptible weakening in investor demand. This weakening in interest occurred due to the economic recovery process started gathering considerable steam, which inturn caused a waning of quite a substantial degree in the appeal of Gold as a reliable hedge against economic uncertainity. Over the time frame Bitcoins have significantly outperformed Gold in cumulative returns, which indicates that it has substantial upside potential, particularly as a speculative investment vehicle. However the high returns that Bitcoin can yield come with a lot of risk, an aspect vividly represented by high volatility in which this cryptocurrency has been wrapped

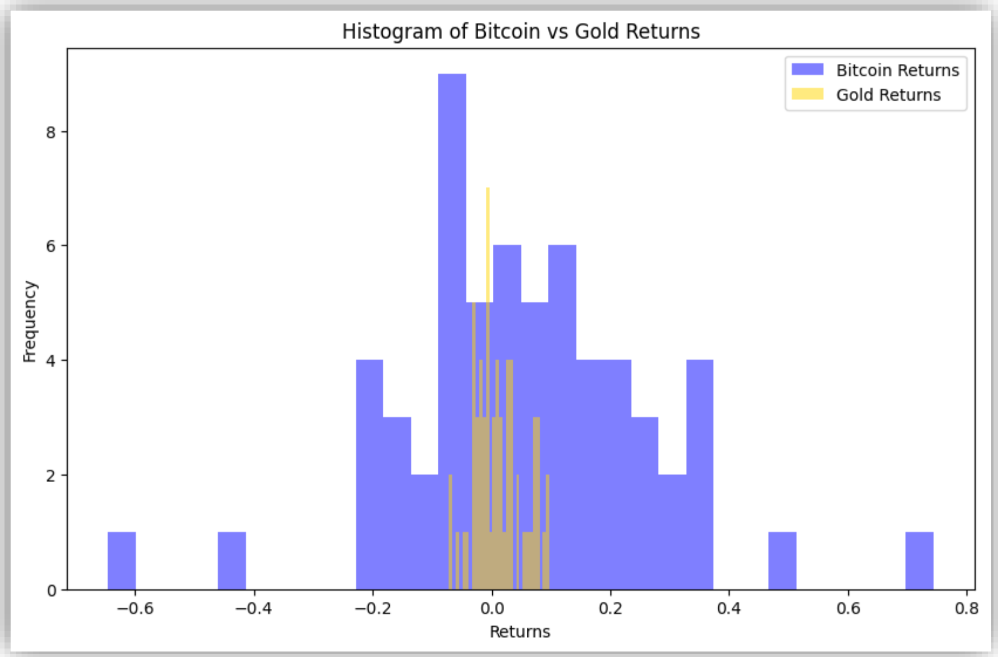


*Fig 15: Histogram of Bitcoin vs Gold Returns*

The histogram gives a visual presentation that helps in comparing the distribution of returns for both Bitcoin and Gold over the stated time period, which is from 2019 to 2024. Bitcoin returns are colored with a bright blue color in this example, while the returns for Gold are colored with a bright yellow color. The histogram is a very good tool in the evaluation of both the frequency and spread of returns for each class of assets, hence helping understanding the dynamics of performance for each asset class.

The range of returns for Bitcoin is much wider as compared to Gold. This certainly means that Bitcoin is a very volatile asset. Indeed, the returns for Bitcoin do vary wider compared to Gold. This certainly means that Bitcoin is a very volatile asset. Indeed the returns for Bitcoin do vary widely, with very high values at both ends of the spectrum, even upto plus or minus 0.6. Compared with other investments, gold displays a much narrower range of returns. It tends to stay closer to the zero mark and rarely goes into extreme positive or negative values that are far removed from this central point.

An overwhelming majority of Bitcoin's returns cluster around the value of 0, which might suggest that there are many instances where its performance is relatively stable. However there is a clear increase in a frequency at both higher positive returns and lower negative returns, which would suggest that the cryptocurrency experiences more substantial swings in price than smaller movements. The returns associated with Gold tend to be significantly more concentrated around the value of 0, which effectively reflects its established position as a stable and low-risk asset in the financial landscape.

The distribution of returns for Bitcoin does have a positive skew, meaning that there is some high positive return and hence has potential for outsize gains but with a considerable downsize risk. The distribution of returns for Gold is roughly symmetrical, consistent with its own characteristic as a stable asset class with very low levels of speculative activity.

The dispersion of returns in Bitcoin is really wide, with obvious tail effects, meaning high volatility. This result is perfectly in line with the nature of assets classified speculative degree of investors risk. A very narrow spread in gold prices is what reinforces the well earned reputation of this investment as a safe haven due to the predictability of returns and a rather conspicuous tendency to lower volatility over time.

The peak frequency for Bitcoin tends to be around the time there is a negative return, specifically hovering around -0.2. This observation correlates closely with its recurring downtrends observed an documented over the years. Gold has a peak frequency closer to 0, therefore it fluctuates very steadily and has a small change in value.

The histogram serves to further emphasize and strengthen the narrative of how returns associated with Bitcoins are extremely volatile in nature. The same volatility that gives one a chance at considerable gain comes with an increase in risk. A dynamic like this is especially appealing to investors with very high-risk tolerance who take their chances in the dark waters of cryptocurrency market.

The stable distribution of returns to gold underlines its values as a low risk, long term asset that is good for risk averse investors who are looking for a hedge in times of economic uncertainty.

Bitcoins and Gold pay dividends to each other in portfolio investment, since features of these assets are always opposite and contrasting in terms of the risk and return profile. A well thought out balance of the allocation could significantly capture the potential that such growth presents in Bitcoin while leveraging the strong presence and security provided by Gold to anchor this investment.

The risk management scheme shall have to absorb increased Bitcoin volatility by reducing the amount present in a portfolio or diversified through other less volatile assets which in our case is Gold.

This in depth analysis underlines the point of difference that put bitcoin apart from Gold and sheds light on the special and important role each of these assets assets play within different investment strategies.

- *Comparative Discussion:*

By analyzing its performance over time, we can say that Bitcoin experienced exponential growth from 2019 to 2024, with big peaks and troughs driven by market sentiment, institution adoption, and macroeconomic trends. Gold made modest but steady gains, reflecting its role as a key hedge against inflation and a world filled with uncertainty.

The divergence in price action between these two assets underscores their very nature where, Bitcoin is a speculative, high risk investment, while Gold is a low risk, wealth preservation tool.

Over the course of the study period, the cumulative returns for Bitcoins were significantly higher compared to the cumulative returns for Gold. The cumulative return for bitcoin reached above 3.0x times from its starting point. Gold's cumulative return remained below 1.0x, emphasizing its stability. The following can be attributed to the significantly higher returns with Bitcoin.

More institutional adoption e.g. Tesla, MicroStrategy. Limited supply (21 million cap) including scarcity driven demand. Gold on the other hand was relatively subdued in its returns, given its well -entrenched position within the market and its lower sensitivity to speculative behaviors often observed in other investment avenues.

The relatively strong positive correlation of about 0.79 between Bitcoin and gold suggests that the two very different financial instruments do not move in complete unison or lockstep with each other. The relatively week correlation between these two assets presents an opportunity to gain diversification benefits within the context of a portfolio.

- *Insights into ARIMA & GARCH:*

This means for Bitcoin specifically, the ARIMA modelling showed that changes in its price are strongly explained through the previous prices therefore maintaining a strong momentum effect. For Gold, ARIMA forecasts were more stable, reflecting its gradual price evolution over time.

ARIMA brought out Bitcoins high sensitivity to short term market conditions, while Gold showed long term stability.

The GARCH model for Bitcoin has shown that volatility will persist, with sharp spikes in time of market stress. Gold's GARCH model was consistently showing low volatility, supporting its safe-haven narrative.

- *Implications for Portfolio Construction/ Recommendation:*

1. Risk tolerant investors:

Risk tolerant investors are those who prefer to forego stability in investments in hopes of reaping higher returns on their investments. They, therefore, are willing to take substantial volatility within the market and accept that losses are a probable prospect while chasing opportunities for significant returns. Bitcoin as a financial asset, fits this type of investor profile exceptionally well since it is highly speculative, showing incredible potential for growth.

The potential for growth of Bitcoin:

Over the period under review, spanning from 2019 to 2024, Bitcoin definitely showed a great ability to generate significant cumulative returns to its investors. In this time frame, it reached new highs that were well above those of traditional investments assets like Gold or the S&P 500 index.

High volatility, while associated with some risk, also brings a lot of opportunities for traders and investors to profit from fluctuations in prices that take place.

For example: investors who chose to enter the market at times when corrections were occurring or during market dips found themselves in a good position to capitalize on the subsequent rallies that followed these downturns.

<u>Recommendations tailored for risk-tolerant investors:</u>

Consider investing a much higher proportion of the total portfolio directly in portfolio- say 60% or even more of the total investment. It would also be prudent to keep some exposure to traditional assets like Gold, different stocks, in the portfolio to hedge against the possibility of high volatility that may arise in the cryptocurrency market.

2. Risk – Adverse Investors:

Risk averse investors tend to put a high order of importance on the safety of their capital and look towards stability, sometimes at the cost of potentially high returns. These investors have a discomfort level associated with market volatility and therefore tend to favor investments that guarantee them a relatively stable and predictable outcome. Among all possible alternatives in the investment landscape, gold remains the best investment for such risk averse investors.

The stability of Gold:

Past performance by Gold has evidenced a pattern of returns, that have been modest but consistent. The long term evidences of such type of this nature. The low volatility found within this context, as is evidently shown by

GARCH model used in the present study, further serves to confirm its important role as a reliable and trusted source of value.

Recommendation for Risk-Averse Investors:

Invest heavily in gold for example 80% or more and keep minimal exposure to bitcoin or any other speculative assets. This will ensure protection of capital with some exposure to growth potential.

3. A well diversified and Balanced Portfolio:

For investors looking for a fine balance between growth and stability, a portfolio that strategically combines both Bitcoin and Gold presents notable advantage in terms of diversification.

Complementary Features:

The rise, sustenance, and hype around Bitcoin depend on speculative environments full of innovation, adoption, and the right sentiment around new technologies. Gold usually does very well in such environments, which are commonly found where there is significant uncertainty economically, chronic inflationary problems, or rising geopolitical tensions that breed volatility.

Risk Return Optimization:

A well thought out allocation involving 70% Gold and 30% Bitcoin really combines stability, which is inherent in Gold, with the potential for growth represented by Bitcoin. This allocation significantly reduces the risk of overexposure to inherent volatility of Bitcoin yet still maintains exposure to its potential upward price movements.

Example Scenario:

In a bull market for cryptocurrencies, the 30% allocation to Bitcoin can substantially improve portfolio returns. During a downturn in the market, the 70% Gold allocation provides stability, lowering the losses.

Advice for moderate investors:

A widely diversified portfolio where Bitcoin is brought in as the growth tool, while gold forms an essential constituent for risk management purposes, would be strongly suggested. To be sure, this balance can and must be changed according to an individuals risk appetite level and special goals in investments.

Regular portfolio rebalancing, in response to dynamic market conditions, is one of the important strategies for keeping the desired risk-return profile of the investment intact over time. For example: increasing Gold allocation during uncertain times or rotating into Bitcoin during bullish market phases.

Time Horizon: Longer investment horizons enable risk-tolerant investors to whether the volatility of Bitcoin. Shorter horizons favor risk averse investors for whom gold offers reliable performance.

Macroeconomic context: in high environments, Gold tends to be a better hedge. During periods of growth driven significantly by tech advances, it is most likely that Bitcoin will outperform by a wide margin.

Recent Regulatory Changes: It appears that this would be especially true for Bitcoin because of international regulatory trends. It was further added that regulations could create stable markets in either case but might remove many opportunities related to speculators. In comparison Gold is less influenced by regulatory action, which further contributes to its tradition and rather well established position in financial environment.

4. Recommendations for researchers:

i. Extend the analysis to include other cryptocurrencies and asset classes.

ii. Use advanced models (e.g. machine learning) to capture non linear relationships and external factors.

## • *Limitations of Study:*

i. Data Limitations:

Monthly data might not capture intraday or weekly volatility patterns. The analysis is limited to Bitcoin and Gold, excluding other cryptocurrencies and asset classes.

ii. Model Assumptions:

ARIMA and GARCH models operate under the assumption of linear relationships and fail to incorporate external influences such as macroeconomic shocks or geopolitical occurrences.

iii. Market Development:

The cryptocurrency market is moving very fast, so historical data is less indicative of future trends.

## • *References:*


i. Main research paper: *"Time series analysis of cryptocurrencies"*

Rama K. Malladi, Prakash L. Dheeriya (2020), This paper compares two of the most famous Bitcoins BTC and XRP to Gold prices daily for the time span of 08/01/2013 and 01/01/2019.

Link: https://link.springer.com/article/10.1007/s12197-020-09526-4?wt_mc=Internal.Event.1.SEM.ArticleAuthorAssignedToIssue&utm_source=ArticleAuthorAssignedToIssue&utm_medium=email&utm_content=AA_en_06082018&ArticleAuthorAssignedToIssue_20201211

ii. *"A systematic Literature review on the determinants of cryptocurrencies"*

Syed Sams, Sansho Peng, Catherine Prentice (2023).

Link: https://www.emerald.com/insight/content/doi/10.1108/cafr-05-2023-0053/full/html

*iii. "Factors influencing Bitcoin Market Volatility during COVID-19"*

Zhunzhun Liu & Lu-xi -Zou (2023)

Link : https://www.tandfonline.com/doi/full/10.1080/10919392.2023.2267957

iv. Factors influencing the trends of Bitcoin:

Pradeep Prabhakaran, Nisanth Panchalingam (2022)

Link: https://www.researchgate.net/publication/357982093_Factors_Influencing_the_Trend_of_Cryptocurrency

v. *"Unveiling the Influencing Factors of Cryptocurrency Return Volatility"*

Ardit Gjeci, Atremisa Pilkati

Link : https://www.mdpi.com/1911-8074/17/1/12

vi. *"Optimizing Cryptocurrency Returns: A Quantitative Study on Factor Investing"*

Claude

Link: https://www.mdpi.com/2227-7390/12/9/1351

vii. *"A Comparison of Cryptocurrency Volatility: Benchmarking New and Mature Asset Classes"*

Alessio Brini

Link: https://jfin-swufe.springeropen.com/articles/10.1186/s40854-024-00646-y

viii. *"Analysis of Factors Affecting Cryptocurrency Return During the COVID-19 Pandemic"*

Mangasi Sinurat

Link : https://www.researchgate.net/publication/356586103_ANALYSIS_OF_FACTORS_AFFECTING_CRYPTOCURRENCY_RETURN_DURING_THE_COVID-19_PANDEMIC

ix. *"An Empirical Study of Volatility in Cryptocurrency Market"*

Hemendra Gupta, Rashmi Chaudhary

Link: https://www.researchgate.net/publication/365241517_An_Empirical_Study_of_Volatility_in_Cryptocurrency_Market

x. *"Common Risk Factors in Cryptocurrency"*

Yukun Liu, Xi Wu

Link: https://www.nber.org/system/files/working_papers/w25882/w25882.pdf

xi. *"Factors Affecting the Volatility of Bitcoin Prices"*

Renhong Wu

xii. *"Factors Influencing the Behavioural Intention to Use Cryptocurrency"*

Prapatchon Jariyapan, Suchira matthayaphutron

Link: https://pmc.ncbi.nlm.nih.gov/articles/PMC8864142/

xiii. *"Modelling Determinants of Cryptocurrency Prices: A Bayesian Network Approach"*

Dhananjay, Asef Nazari.

- *Appendix:*

```python
import yfinance as yf
import pandas as pd

# Define the cryptocurrency and its Yahoo Finance ticker
ticker = 'BTC-USD'
name = 'Bitcoin'

# Define the date range
start_date = "2019-01-01"
end_date = "2024-12-31"

# Download data for Bitcoin
print(f"Downloading data for {name} ({ticker})...")
data = yf.download(ticker, start=start_date, end=end_date, interval="1mo")
data.reset_index(inplace=True)

# Flatten MultiIndex columns if present
if isinstance(data.columns, pd.MultiIndex):
    data.columns = data.columns.get_level_values(0)  # Flatten to single-level

print(f"Headers for {name}: {list(data.columns)}")

# Keep only relevant columns
data = data[["Date", "Open", "High", "Low", "Close"]]  # Select required columns

# Display the first few rows of the data
print("Preview of the data:")
print(data.head())

# Save to a CSV file
output_file = f"{name.lower()}_monthly_data_2019_2024.csv"
data.to_csv(output_file, index=False)
print(f"Data for {name} saved to {output_file}")
```

```python
# Download data for Bitcoin
print(f"Downloading data for {name} ({ticker})...")
data = yf.download(ticker, start=start_date, end=end_date, interval="1mo")
data.reset_index(inplace=True)

# Flatten MultiIndex columns if present
if isinstance(data.columns, pd.MultiIndex):
    data.columns = data.columns.get_level_values(0)  # Flatten to single-level

print(f"Headers for {name}: {list(data.columns)}")

# Keep only relevant columns
data = data[["Date", "Open", "High", "Low", "Close"]]  # Select required columns

# Display the first few rows of the data
print("Preview of the data:")
print(data.head())

# Save to a CSV file
output_file = f"{name.lower()}_monthly_data_2019_2024.csv"
data.to_csv(output_file, index=False)
print(f"Data for {name} saved to {output_file}")

# Prepare data for OLS regression
# Using 'Close' as the dependent variable and 'Open', 'High', 'Low' as independent variables
X = data[["Open", "High", "Low"]]
y = data["Close"]

# Add a constant to the independent variables
X = sm.add_constant(X)

# Check for multicollinearity using Variance Inflation Factor (VIF)
vif_data = pd.DataFrame()
vif_data["feature"] = X.columns
vif_data["VIF"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]
print("Variance Inflation Factor (VIF):")
print(vif_data)

# Fit the OLS regression model
model = sm.OLS(y, X).fit()

# Print the summary of the regression model
print("OLS Regression Results:")
print(model.summary())
```

```
# Display the first few rows of the data
print("Preview of the data:")
print(data.head())

# Save to a CSV file
output_file = f"{name.lower()}_monthly_data_2019_2024.csv"
data.to_csv(output_file, index=False)
print(f"Data for {name} saved to {output_file}")

# Prepare data for OLS regression
# Using 'Close' as the dependent variable and 'Open', 'High', 'Low' as independent variables
X = data[["Open", "High", "Low"]]
y = data["Close"]

# Apply PCA to resolve multicollinearity
pca = PCA(n_components=2)  # Reduce to 2 components
X_pca = pca.fit_transform(X)

# Convert PCA components back to a DataFrame
X_pca = pd.DataFrame(X_pca, columns=["PCA1", "PCA2"])

# Add a constant for the OLS regression
X_pca = sm.add_constant(X_pca)

# Fit the OLS regression model
model = sm.OLS(y, X_pca).fit()

# Print the summary of the regression model
print("OLS Regression Results after PCA:")
print(model.summary())
```

```
# Keep only relevant columns
data = data[["Date", "Open", "High", "Low", "Close"]]  # Select required columns

# Display the first few rows of the data
print("Preview of the data:")
print(data.head())

# Save to a CSV file
output_file = f"{name.lower()}_monthly_data_2019_2024.csv"
data.to_csv(output_file, index=False)
print(f"Data for {name} saved to {output_file}")

# Prepare data for OLS regression
# Using 'Close' as the dependent variable and 'Open', 'High', 'Low' as independent variables
X = data[["Open", "High", "Low"]]
y = data["Close"]

# Standardize the features to reduce numerical instability
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Add a constant to the standardized features
X_scaled = sm.add_constant(X_scaled)

# Fit the OLS regression model
model = sm.OLS(y, X_scaled).fit()

# Print the summary of the regression model
print("OLS Regression Results after Feature Scaling:")
print(model.summary())
```

```
# Check residuals for autocorrelation
from statsmodels.stats.diagnostic import acorr_ljungbox

# Extract residuals from the model
residuals = model.resid

# Durbin-Watson statistic
dw_stat = sm.stats.durbin_watson(residuals)
print(f"Durbin-Watson statistic: {dw_stat}")

# Ljung-Box test for autocorrelation
ljung_box = acorr_ljungbox(residuals, lags=[10], return_df=True)
print("Ljung-Box test results:")
print(ljung_box)
```

```
# Apply ARIMA model to reduce autocorrelation
# Differencing the data to make it stationary
data_diff = data.diff().dropna()

# Fit ARIMA model (p, d, q) = (1, 1, 1) as an example
model = ARIMA(data_diff, order=(1, 1, 1))
results = model.fit()

# Print model summary
print("ARIMA Model Summary:")
print(results.summary())

# Plot residuals to check for autocorrelation
residuals = results.resid
plt.figure(figsize=(10, 6))
plt.plot(residuals, label="Residuals")
plt.axhline(0, color='red', linestyle='--')
plt.title("Residuals of ARIMA Model")
plt.legend()
plt.show()

# Ljung-Box test for autocorrelation
from statsmodels.stats.diagnostic import acorr_ljungbox
ljung_box = acorr_ljungbox(residuals, lags=[10], return_df=True)
print("Ljung-Box test results:")
print(ljung_box)
```

```
# Apply simplified ARIMA model to reduce autocorrelation
# Differencing the data to make it stationary
data_diff = data.diff().dropna()

# Fit a simpler ARIMA model (p, d, q) = (0, 1, 1)
model = ARIMA(data_diff, order=(0, 1, 1))
results = model.fit()

# Print model summary
print("Simplified ARIMA Model Summary:")
print(results.summary())

# Plot residuals to check for autocorrelation
residuals = results.resid
plt.figure(figsize=(10, 6))
plt.plot(residuals, label="Residuals")
plt.axhline(0, color='red', linestyle='--')
plt.title("Residuals of Simplified ARIMA Model")
plt.legend()
plt.show()

# Ljung-Box test for autocorrelation
from statsmodels.stats.diagnostic import acorr_ljungbox
ljung_box = acorr_ljungbox(residuals, lags=[10], return_df=True)
print("Ljung-Box test results:")
print(ljung_box)
```

```
# Plot ACF of residuals
plt.figure(figsize=(10, 6))
plot_acf(residuals, lags=20, title="ACF of Residuals")
plt.show()
```

```
# Differencing the log-transformed data to make it stationary
data_diff = data["Log_Close"].diff().dropna()

# Fit a simpler ARIMA model (p, d, q) = (0, 1, 1)
model = ARIMA(data_diff, order=(0, 1, 1))
results = model.fit()

# Print model summary
print("Simplified ARIMA Model Summary (Log-Transformed):")
print(results.summary())

# Plot residuals to check for autocorrelation
residuals = results.resid
plt.figure(figsize=(10, 6))
plt.plot(residuals, label="Residuals")
plt.axhline(0, color='red', linestyle='--')
plt.title("Residuals of Simplified ARIMA Model (Log-Transformed)")
plt.legend()
plt.show()

# Ljung-Box test for autocorrelation
from statsmodels.stats.diagnostic import acorr_ljungbox
ljung_box = acorr_ljungbox(residuals, lags=[10], return_df=True)
print("Ljung-Box test results (Log-Transformed):")
print(ljung_box)

# Plot ACF of residuals
plt.figure(figsize=(10, 6))
plot_acf(residuals, lags=20, title="ACF of Residuals (Log-Transformed)")
plt.show()
```

```
# Plot ACF of residuals
plt.figure(figsize=(10, 6))
plot_acf(residuals, lags=20, title="ACF of Residuals (Log-Transformed)")
plt.show()

# Fit a GARCH model to check for heteroskedasticity
garch_model = arch_model(residuals, vol="Garch", p=1, q=1)
garch_results = garch_model.fit()

# Print GARCH model summary
print("GARCH Model Summary:")
print(garch_results.summary())

# Plot the conditional volatility
plt.figure(figsize=(10, 6))
plt.plot(garch_results.conditional_volatility, label="Conditional Volatility")
plt.title("Conditional Volatility from GARCH Model")
plt.legend()
plt.show()
```

```
# Fit a simpler ARIMA model (p, d, q) = (0, 1, 1)
model = ARIMA(data_diff, order=(0, 1, 1))
results = model.fit()

# Print model summary
print("Simplified ARIMA Model Summary (Log-Transformed):")
print(results.summary())

# Forecasting future values
forecast_steps = 12  # Number of months to forecast
forecast = results.get_forecast(steps=forecast_steps)
forecast_mean = forecast.predicted_mean
forecast_conf_int = forecast.conf_int()

# Prepare forecast results
forecast_dates = pd.date_range(data_diff.index[-1] + pd.offsets.MonthBegin(1), periods=forecas
forecast_df = pd.DataFrame({
    "Forecast": forecast_mean,
    "Lower Bound": forecast_conf_int.iloc[:, 0],
    "Upper Bound": forecast_conf_int.iloc[:, 1]
}, index=forecast_dates)

print("Forecasted Values:")
print(forecast_df)

# Plot forecast
plt.figure(figsize=(10, 6))
plt.plot(data["Log_Close"], label="Log_Close (Historical)", color='blue')
plt.plot(forecast_df["Forecast"], label="Forecast", color='green')
plt.fill_between(forecast_df.index, forecast_df["Lower Bound"], forecast_df["Upper Bound"], co
plt.title("Log-Transformed Close Price Forecast")
plt.xlabel("Date")
plt.ylabel("Log Close")
plt.legend()
plt.show()

# Plot residuals to check for autocorrelation
residuals = results.resid
plt.figure(figsize=(10, 6))
plt.plot(residuals, label="Residuals")
plt.axhline(0, color='red', linestyle='--')
plt.title("Residuals of Simplified ARIMA Model (Log-Transformed)")
plt.legend()
plt.show()

# Ljung-Box test for autocorrelation
from statsmodels.stats.diagnostic import acorr_ljungbox
ljung_box = acorr_ljungbox(residuals, lags=[10], return_df=True)
print("Ljung-Box test results (Log-Transformed):")
print(ljung_box)

# Plot ACF of residuals
plt.figure(figsize=(10, 6))
plot_acf(residuals, lags=20, title="ACF of Residuals (Log-Transformed)")
plt.show()

# Fit a GARCH model to check for heteroskedasticity
garch_model = arch_model(residuals, vol="Garch", p=1, q=1)
garch_results = garch_model.fit()

# Print GARCH model summary
print("GARCH Model Summary:")
print(garch_results.summary())

# Plot the conditional volatility
plt.figure(figsize=(10, 6))
plt.plot(garch_results.conditional_volatility, label="Conditional Volatility")
plt.title("Conditional Volatility from GARCH Model")
plt.legend()
plt.show()
```

```
# Download Gold Prices (GC=F)
print("Downloading data for Gold Futures (GC=F)...")
gold_data = yf.download("GC=F", start=start_date, end=end_date, interval="1mo")
gold_data.reset_index(inplace=True)

# Keep only relevant columns
gold_data = gold_data[["Date", "Close"]]
gold_data.rename(columns={"Close": "Gold_Close"}, inplace=True)
gold_data.set_index("Date", inplace=True)

# Remove rows with NaN or zero values before applying log transformation
gold_data = gold_data[gold_data["Gold_Close"] > 0]  # Ensures valid values only

# Apply log transformation to stabilize variance
gold_data["Log_Gold_Close"] = np.log(gold_data["Gold_Close"])

# Drop rows with NA values caused by log transformation
gold_data = gold_data.dropna()

# Check for remaining NaN values
if gold_data["Log_Gold_Close"].isna().any():
    print("Warning: Some rows still contain NaNs in Gold data.")
```

```
print("Gold Data after Log Transformation:")
print(gold_data.head())
print("Columns in Gold Data:", gold_data.columns)
```

```python
# Merge Bitcoin and Gold Data
combined_data = pd.merge(
    data,
    gold_data[["Log_Gold_Close"]],
    left_index=True,
    right_index=True,
    how="inner"
)

# Display combined data
print("Combined Bitcoin and Gold Data:")
print(combined_data.head())

# Save combined data to a CSV file
combined_output_file = "bitcoin_gold_combined_data_2019_2024.csv"
combined_data.to_csv(combined_output_file)
print(f"Combined data saved to {combined_output_file}")

# Plot Bitcoin vs Gold Log Prices
plt.figure(figsize=(10, 6))
plt.plot(combined_data.index, combined_data["Log_Close"], label="Bitcoin (Log_Close)", color='blue')
plt.plot(combined_data.index, combined_data["Log_Gold_Close"], label="Gold (Log_Gold_Close)", color='gold')
plt.title("Bitcoin vs Gold Log-Transformed Prices")
plt.xlabel("Date")
plt.ylabel("Log Price")
plt.legend()
plt.show()

# Calculate Correlation
correlation = combined_data[["Log_Close", "Log_Gold_Close"]].corr()
print("Correlation between Bitcoin (Log_Close) and Gold (Log_Gold_Close):")
print(correlation)

# Compare Volatility using GARCH for Gold
print("Fitting GARCH model to Gold Log Prices...")
gold_garch_model = arch_model(combined_data["Log_Gold_Close"].diff().dropna(), vol="Garch", p=1, q=1)
gold_garch_results = gold_garch_model.fit()

# Print GARCH model summary for Gold
print("GARCH Model Summary for Gold:")
print(gold_garch_results.summary())

# Plot Conditional Volatility for Gold
plt.figure(figsize=(10, 6))
plt.plot(gold_garch_results.conditional_volatility, label="Gold Conditional Volatility", color='gold')
plt.title("Conditional Volatility of Gold Log Prices")
plt.legend()
plt.show()
```

```python
# Rescale Bitcoin log returns
btc_log_returns = combined_data["Log_Close"].diff().dropna() * 10  # Rescale by 10

# Fit GARCH model to rescaled Bitcoin log returns
print("Fitting GARCH model to rescaled Bitcoin Log Returns...")
btc_garch_model = arch_model(btc_log_returns, vol="Garch", p=1, q=1)
btc_garch_results = btc_garch_model.fit()

# Forecast Volatility
forecast_steps = 12
btc_forecast = btc_garch_results.forecast(horizon=forecast_steps)

# Plot Forecasted Conditional Volatility
plt.figure(figsize=(10, 6))
plt.plot(btc_forecast.variance.values[-1], label="Bitcoin Rescaled Forecasted Volatility", color='blue')
plt.title("Forecasted Conditional Volatility for Rescaled Bitcoin Log Returns")
plt.xlabel("Months Ahead")
plt.ylabel("Forecasted Volatility")
plt.legend()
plt.show()

# Print Summary
print(btc_garch_results.summary())
```

```python
from statsmodels.stats.diagnostic import acorr_ljungbox

# Residual Analysis for Bitcoin GARCH Model
print("\nResidual Analysis for Bitcoin GARCH Model...")
btc_log_returns = combined_data["Log_Close"].diff().dropna() * 10  # Rescaled log returns
btc_garch_model = arch_model(btc_log_returns, vol="Garch", p=1, q=1)
btc_garch_results = btc_garch_model.fit(disp="off")

# Extract standardized residuals
standardized_residuals = btc_garch_results.resid / btc_garch_results.conditional_volatility

# Plot ACF of Residuals
plt.figure(figsize=(10, 6))
plot_acf(standardized_residuals, lags=20, title="ACF of Standardized Residuals")
plt.show()

# Ljung-Box Test for Autocorrelation
lb_test = acorr_ljungbox(standardized_residuals, lags=[10], return_df=True)
print("Ljung-Box Test Results (Bitcoin GARCH Residuals):")
print(lb_test)

# Check if residuals exhibit remaining volatility clustering
plt.figure(figsize=(10, 6))
plt.plot(standardized_residuals, label="Standardized Residuals", color='blue')
plt.title("Standardized Residuals from Bitcoin GARCH Model")
plt.legend()
plt.show()
```

```
# ARIMA Forecasting for Bitcoin Log Prices
print("\nForecasting Bitcoin Log Prices using ARIMA...")
model_arima = ARIMA(combined_data["Log_Close"], order=(1, 1, 1))  # ARIMA(1,1,1)
arima_results = model_arima.fit()

# Forecast next 12 months
forecast_steps = 12
arima_forecast = arima_results.forecast(steps=forecast_steps)

# Forecasted Volatility from GARCH
garch_forecast = btc_garch_results.forecast(horizon=forecast_steps).variance.iloc[-1]

# Plot Forecasted Prices and Volatility
plt.figure(figsize=(10, 6))
plt.plot(combined_data.index, combined_data["Log_Close"], label="Bitcoin Log Prices (Historical)", color='blue')
plt.plot(pd.date_range(combined_data.index[-1], periods=forecast_steps+1, freq='M')[1:], arima_forecast,
         label="ARIMA Forecasted Prices", color='red')
plt.title("Bitcoin Log Prices Forecast (ARIMA) with Conditional Volatility")
plt.legend()
plt.show()

print("\nForecasted Bitcoin Log Prices:")
print(arima_forecast)
```

```
# Back-transform ARIMA forecast to original price scale
forecast_steps = 12  # Forecast for 12 months
arima_forecast = arima_results.get_forecast(steps=forecast_steps).predicted_mean
forecast_dates = pd.date_range(combined_data.index[-1], periods=forecast_steps+1, freq='M')[1:]

# Convert log-transformed forecast back to original scale
forecast_prices = np.exp(arima_forecast)

# Plot forecasted prices on original scale
plt.figure(figsize=(10, 6))
plt.plot(data.index, data["Close"], label="Bitcoin Prices (Historical)", color="blue")
plt.plot(forecast_dates, forecast_prices, label="Forecasted Prices (ARIMA)", color="red")
plt.title("Forecasted Bitcoin Prices (ARIMA)")
plt.xlabel("Date")
plt.ylabel("Price")
plt.legend()
plt.show()

# Display forecasted prices
forecast_df = pd.DataFrame({"Date": forecast_dates, "Forecasted_Close": forecast_prices})
print("Forecasted Bitcoin Prices:")
print(forecast_df)
```

```
import yfinance as yf
import pandas as pd
import matplotlib.pyplot as plt

# Define the tickers and date range
tickers = ['BTC-USD', '^GSPC']
start_date = '2020-01-01'
end_date = '2024-12-31'

# Download data for both Bitcoin and S&P 500
data = yf.download(tickers, start=start_date, end=end_date, interval="1mo", group_by="ticker")

# Prepare data for Bitcoin
btc_data = data['BTC-USD']['Close'].dropna()
btc_data.name = "BTC-USD"
btc_volatility = btc_data.pct_change().rolling(window=12).std() * (12**0.5)

# Prepare data for S&P 500
sp500_data = data['^GSPC']['Close'].dropna()
sp500_data.name = "S&P 500"
sp500_volatility = sp500_data.pct_change().rolling(window=12).std() * (12**0.5)

# Combine volatilities into a single DataFrame
volatility_data = pd.DataFrame({'BTC-USD Volatility': btc_volatility, 'S&P 500 Volatility': sp500_volatility})

# Plot the volatility comparison
plt.figure(figsize=(12, 6))
plt.plot(volatility_data.index, volatility_data['BTC-USD Volatility'], label='BTC-USD Volatility', color='blue')
plt.plot(volatility_data.index, volatility_data['S&P 500 Volatility'], label='S&P 500 Volatility', color='orange')
plt.title('Volatility Comparison: BTC-USD vs S&P 500 (2020-2024)')
plt.xlabel('Date')
plt.ylabel('Annualized Volatility')
plt.legend()
plt.grid()
plt.show()
```