MINI PROJECT REPORT

ON

BITCOIN SHARE VALUES****

SUBMITTED BY

PRANEETHWARAN.K

NC.SC.U4CSE24222

FOR

23CSE101-COMPUTATIONAL PROBLEM SOLVING

I SEMESTER

B.TECH.-CSE

**SCHOOL OF COMPUTING**

**AMRITA VISHWA VIDHYAPEETHAM, NAGERCOIL**

**Index**

**Chapter Name Page No.**

**1. Introduction**

**2. Problem Statement**

**3. Objectives**

**4. Python Libraries used in the project**

**5. Modules of project**

**6. Code**

**7. Output Screenshots**

**8. Application of the project**

**9. Limitations of the project**

**10. Bibliography**

**11. GitHub link of the project**

**INTRODUCTION:**

1. Bitcoin is a cryptocurrency introduced in 2009 by an anonymous entity called Satoshi Nakamoto.
2. It operates on blockchain technology, ensuring transparency and security.
3. Bitcoin's price is highly volatile due to limited supply and demand.
4. Understanding historical price data can assist in making informed investment decisions.
5. The project uses Python to analyse Bitcoin price trends over a specified period.
6. The focus is on closing prices, a common metric for financial analysis.
7. Data from Yahoo Finance is utilized, offering free and reliable cryptocurrency data.
8. Visualizations help identify patterns such as price, and long-term trends.
9. The project serves as a foundation for creating more advanced cryptocurrency analytics tools.
10. The script is modular, making it easy to extend for other cryptocurrencies or advanced analyses.
11. DATA VISUALIZATION helps the reader or the viewer to understand the data very clear and in the right manner.

PROBLEM STATEMENT:

1. This project aims to bridge these gaps using Python's powerful libraries.
2. Understanding price trends requires access to accurate historical data.
3. Manual analysis of large datasets is time-consuming and error-prone.
4. Traditional financial tools may not support cryptocurrency-specific analyses.
5. A lack of visual tools makes it harder to interpret price trends and patterns.
6. Bitcoin's extreme volatility poses challenges for investors and researchers.
7. A lack of visual tools makes it harder to interpret price trends and patterns.
8. Identifying correlations between metrics like volume and price changes is difficult.
9. Many available tools require expensive subscriptions or technical expertise
10. Investors lack an easy way to analyse historical performance over custom time frames
11. Tools for comparing cryptocurrencies or validating investment hypotheses are limited.

**OBJECTIVES:**

1. Fetch historical Bitcoin data for a user-defined date range.
2. Compute statistical metrics like mean, median, and standard deviation.
3. Display descriptive statistics to summarize Bitcoin's performance.
4. Generate plots of Bitcoin's closing prices to visualize trends.
5. Identify key insights, such as periods of high volatility or steady growth.
6. Provide a modular framework to adapt the analysis for other cryptocurrencies.
7. Highlight data anomalies, such as sudden spikes or crashes.
8. Enable users to detect long-term trends using simple visual tools.
9. Create reusable code that can be extended for machine learning applications.
10. Offer an accessible way for non-technical users to analyse Bitcoin data.

**Python Libraries Used in the Project**

1. **Y finance**:
   * Fetches historical cryptocurrency prices, including Open, High, Low, Close prices, and Volume.
   * Provides seamless integration with Yahoo Finance's APIs.
2. **pandas**:
   * Handles structured data using Data Frames for easy manipulation.
   * Computes summary statistics like mean, median, and variance.
3. **Matplotlib. Pyplot**:
   * Visualizes the data using line plots and scatter plots.
   * Enables customization of titles, labels, and legends for clarity.
4. **numpy**:
   * Supports numerical operations like array manipulation and statistical computations.
   * Simplifies calculations of averages and variances.
5. **seaborn (optional)**:
   * Enhances plots with advanced styling and heatmaps.

6.**datetime (optional)**:

Helps parse and manipulate date ranges for analysis.

**7.sys (optional)**:

Captures and handles system-related errors during runtime.

**8.os (optional)**:

Manages file operations if data is stored or retrieved locally

INFO ABOUT LIBRARY USED:

Each library is open-source, ensuring accessibility and flexibility. Combining these libraries enables comprehensive data analysis and visualization.

The optional modules used in the python code helps the user to understand the values at the accurate date and the time and the seaborn module helps the graph in the decorative manner the sys module help the code to execute faster without error during the runtime.

**5. Modules of the Project**

1. **Data Fetching**:

Retrieves historical Bitcoin price data using y finance.

Accepts user-defined start and end dates for customization.

1. **Data Cleaning (optional)**:

Handles missing or erroneous data points for accuracy.

1. **Exploratory Analysis**:

Provides descriptive statistics such as count, mean, and standard deviation.

Displays the first and last few rows to give and overview of the dataset.

1. **Data Visualization**:

Plots closing prices to show trends over time.

Highlights price fluctuations using gridlines and legends.

**5**.**Error Handling**:

* + Ensures that invalid inputs or missing data do not crash the program.

**6.Main Controller**:

Orchestrates the script's flow from fetching to visualization.

**7.Modularity**:

Each function operates independently, allowing for easy updates or additions.

**8.Flexibility**:

Adaptable for analyzing other cryptocurrencies like Ethereum or Litecoin.

**9.Scalability**:

Supports integration with machine learning or predictive analytics in future iterations.

**10.Documentation**:

Includes comments for better understanding and future enhancements.

PYTHON CODE

import yfinance as yf

import matplotlib.pyplot as plt

import pandas as pd

# Fetch Bitcoin price data

def fetch\_bitcoin\_data(start\_date="2000-01-01", end\_date="2025-12-31"):

    btc = yf.Ticker("BTC-USD")  # Bitcoin in USD

    btc\_data = btc.history(start=start\_date, end=end\_date)

    return btc\_data

# Exploratory analysis

def explore\_bitcoin\_data(btc\_data):

    print("Summary statistics:")

    print(btc\_data.describe())

    print("\nFirst few rows of the data:")

    print(btc\_data.head())

    print("\nLast few rows of the data:")

    print(btc\_data.tail())

# Plot the closing price

def plot\_bitcoin\_data(btc\_data):

    if 'Close' in btc\_data.columns:

        plt.figure(figsize=(10, 5))

        plt.plot(btc\_data.index, btc\_data['Close'], label="Closing Price", color='black')

        plt.title("Bitcoin Price Over Time")

        plt.xlabel("Date")

        plt.ylabel("Price (USD)")

        plt.legend()

        plt.grid(True)

        plt.show()

    else:

        print("The 'Close' column is not available in the data.")

# Main function to run the analysis

if \_\_name\_\_ == "\_\_main\_\_":  # Fixed the typo "\_\_name\_\_" instead of "\_name\_"

    start\_date = "2000-01-01"  # Start of analysis

    end\_date = "2025-12-31"    # End of analysis

    # Fetch data

    btc\_data = fetch\_bitcoin\_data(start\_date, end\_date)

    # Explore data

    explore\_bitcoin\_data(btc\_data)

    # Plot data

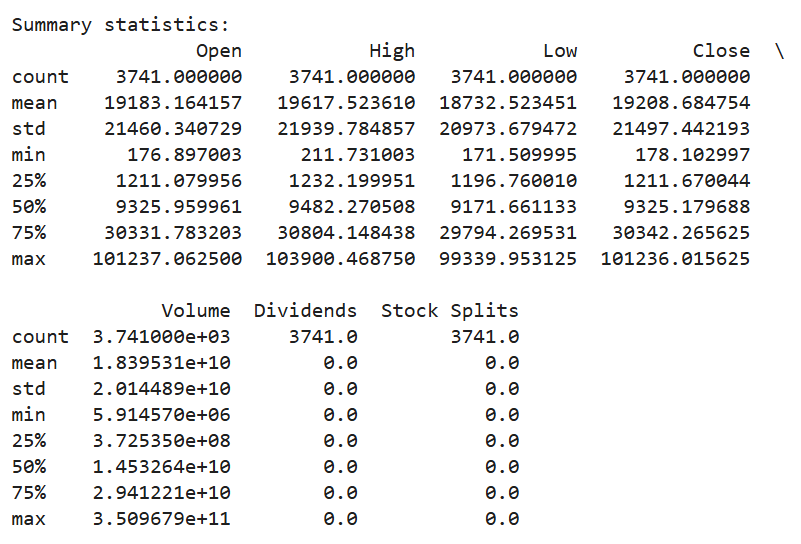
    plot\_bitcoin\_data(btc\_data)

ENVIRONMENT USED:

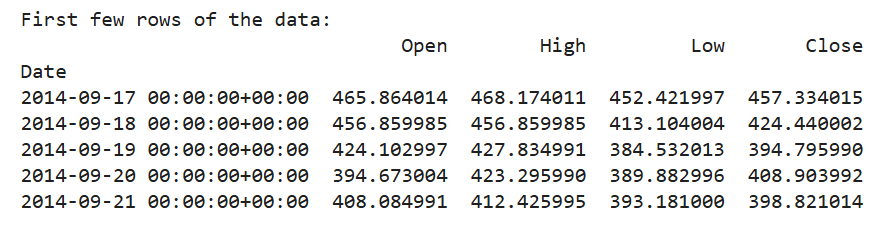
The above code is done in the Google Colab. As consists of the default backend Python3.

OUTPUT

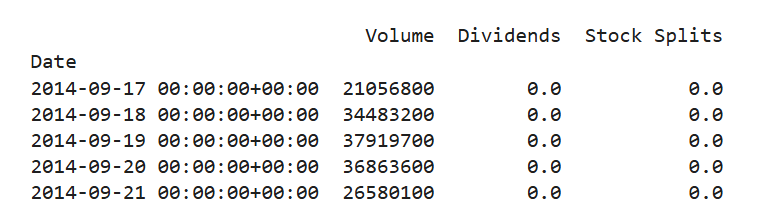
Summary statistics:



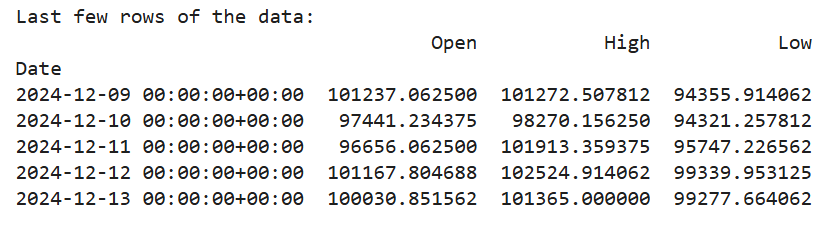
First few rows of data:



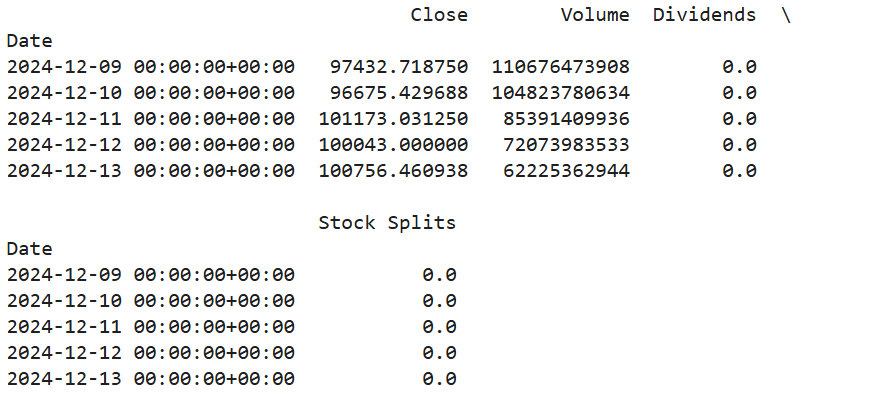
Stocks split:



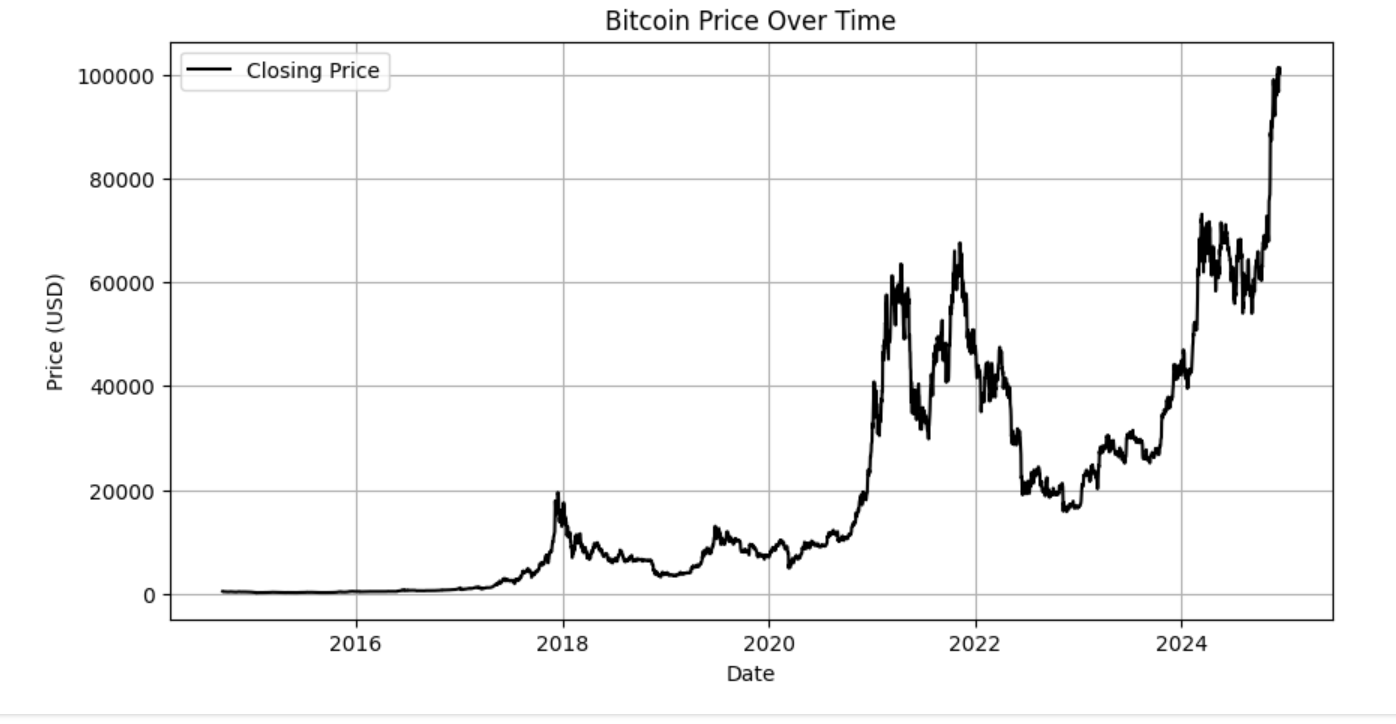
Last few rows of data:



Stocks splits:



VISUALIZATION:



The graph is plotted in the dollar as converting to the Indian money

**1 USD = 84.7820 INR**

The price of the bitcoin in the month of December is

**Bitcoin (BTC)**

**Bitcoin-₹85,52,178.31(‎-0.66%)**

**Applications of the Project**

1. **Investor Insights:**

Helps investors analyze historical price trends for better decision-making.

1. **Market Research:**

Identifies periods of high or low volatility for market study.

1. **Risk Management:**

Assesses historical price fluctuations to manage financial risk.

1. **Portfolio Optimization:**

Provides data for balancing cryptocurrency investments.

1. **Educational Use:**

Teaches students how to analyze financial data programmatically.

**6.Pattern Detection:**

Highlights recurring patterns, such as price rallies post-halving.

**7.Tool for Researchers**:

Assists in studying correlations between price and external factors.

**8.Customizable Framework:**

Adaptable for other cryptocurrencies or stock data.

**9.Foundation for Predictive Models:**

Can be extended to include machine learning for price predictions.

**10.Public Awareness:**

Informs non-technical users about Bitcoin's historical performance.

**LIMITATIONS OF THE PROJECT:**

**1.** **Dependence on Yahoo Finance**:

The program relies entirely on the availability of Yahoo Finance APIs.

2.**Static Analysis**:

Focuses on historical data, with no real-time tracking or updates.

**3.Limited to Bitcoin**:

Requires modifications for analyzing other financial instruments.

**4.No Predictive Modeling**:

Does not forecast future prices or trends.

**5.** **Simplistic Visuals**:

Basic plots; lacks advanced visualizations like candlestick charts.

**6.**  **No Risk Metrics:**

Omits important metrics like Sharpe ratio or Value at Risk (VaR).

**7.Data Gaps:**

Missing data points can skew analysis.

**8.No Sentiment Analysis:**

Ignores external factors like news or social media sentiment.

**9.No Multi-Currency Analysis:**

Cannot compare multiple cryptocurrencies simultaneously.

**10.Not Suitable for Real-Time Trading:**

Lacks features required for high-frequency trading or live decision-making.

**BIBLIOGRAPHY:**

**Python Software Foundation.**GOOGLE COLAB

**Pandas Development Team.**Pandas: Python Data Analysis Library. Available at: https://pandas.pydata.org/

**NumPy Community.**NumPy: The fundamental package for scientific computing with Python. Available at:[**https://numpy.org/**](https://numpy.org/)

**Matplotlib Developers.**Matplotlib: Visualization with Python. Available at:[**https://matplotlib.org/**](https://matplotlib.org/)

**Cryptocurrency APIs (e.g., Coin Gecko, Coin Market Cap).**Used for retrieving Bitcoin price data. Documentation:

* Coin Gecko API: https://www.coingecko.com/en/api
* Coin MarketCap API: https://coinmarketcap.com/api/

**Kaggle Datasets.**Public datasets used for cryptocurrency historical analysis. Example: Bitcoin Historical Data. Available at: [**https://www.kaggle.com/**](https://www.kaggle.com/)

**OpenAI ChatGPT.**Assistance in developing and reviewing the Python code. Available at:[**https://openai.com/**](https://openai.com/)

**Official Python Documentation.**Modules and functions used in the code, such as datetime, statistics, or requests. Available at:[**https://docs.python.org/3/**](https://docs.python.org/3/)

**Books and Tutorials:**

"Mastering Bitcoin: Unlocking Digital Cryptocurrencies" by Andreas M. Antonopoulos.

**GITHUB LINK:**