Automating Crypto Website API Pull Using Python

Purpose:

This Python script automates the retrieval, processing, and analysis of cryptocurrency data from the CoinMarketCap API, aiming to provide valuable insights for data analysts and cryptocurrency enthusiasts.

Libraries Used:

- `requests`: Handles HTTP requests to the CoinMarketCap API.
- `json`: Deals with JSON data.
- `pandas`: Manages data in DataFrame format.
- `seaborn` and `matplotlib.pyplot`: Used for data visualization.

Use Case:

Analyzing Cryptocurrency Trends

Data Acquisition:

- API Retrieval: Utilizes the CoinMarketCap API to fetch real-time cryptocurrency data.
- Data Normalization: Transforms the obtained JSON data into a structured DataFrame (`df`) for analysis.

Data Analysis Perspectives:

1. Temporal Trends:
- Time Series Analysis: Visualizes price fluctuations of specific cryptocurrencies (e.g., Bitcoin) over time using line plots.
- Percent Change Calculation: Analyzes hourly, daily, weekly, and monthly percent changes to identify trends and volatility.

2. Comparative Analysis:
- Mean Percentage Changes: Calculates the mean percentage changes across cryptocurrencies to compare their performance.
- Categorical Visualization: Utilizes categorical plots to compare the performance of different cryptocurrencies across various timeframes.

Insights & Applications:

- Identifying Trends: Observes temporal trends to understand short-term and long-term performance.
- Comparative Performance: Compares the relative performance of multiple cryptocurrencies, aiding in investment or strategic decisions.
- Volatility Analysis: Analyzes percent changes to gauge the volatility of cryptocurrencies over different time intervals.

Notes & Instructions:

- The script requires specific Anaconda Prompt adjustments for data retrieval due to data rate limitations *(`jupyter notebook --NotebookApp.iopub_data_rate_limit=1e10`).*
- Sections for CSV operations are available for data storage but currently commented out.

Instructions for Use:

1. Setup: - Ensure necessary libraries (`requests`, `pandas`, `seaborn`, `matplotlib`) are installed.
2. Execution:
- Run the script to fetch cryptocurrency data and observe trends through generated visualizations.

3. Customization:
- Modify paths or filenames for CSV operations if uncommented, enabling data storage for future analysis.

Use Case Demonstration:

- Provides a comprehensive approach for data analysts to understand, compare, and visualize cryptocurrency trends for strategic decision-making or further in-depth analysis.