Front End UI/UX Design Project

Velar Technologies

January 2025

1 Cryptocurrency Correlation Matrix Dashboard

Objective: Create an interactive web dashboard that fetches crypto prices from a public cryptocurrency API and displays price correlations among assets.

Correlation is a one statistical measurement of how two variables are related to each other. For the purposes of this project, we will utilize the Pearson Correlation. A correlation coefficient between two assets is typically expressed as a number between [-1, 1] where +1 indicates the price of two assets move in the same direction and with the same magnitude. A coefficient of -1 indicates that they move with the same magnitude in opposite directions (i.e asset A appreciates 5% while asset B depreciates 5%). More information can be found here: Wikipedia Pearson Correlation Coefficient

Select a cryptocurrency exchange with a public API that provides historical pricing data on regularly sampled time boundaries. Examples in this document will be selected from the Kraken spot exchange, although you are free to utilize any exchange that provides such data and is not restricted in either your jurisdiction or Singapore. Kraken refers to this data as `OHLC` and the documentation can be found at the Kraken API documentation. Note that there are examples for fetching data using `curl` and Python on the right hand side. Other exchanges sometimes refer to this data as `candles` or `bars`. Exchanges often offer multiple time frame intervals over which the data is aggregated. For this project, feel free to select a single "interval" that is less than or equal to 1 day (e.g. 1 minute) or make it an option for the user.

First, fetch the list of available assets and display them in some sort of a list so that the user can select among the available assets. For Kraken, this can be obtained by querying Get Tradable Asset Pairs. Note that most exchanges (including Kraken) have API rate limits that limit the frequency you can query, so you may need a mechanism for handling these limits (i.e. throttling queries and/or retrying failed attempts). For the selected assets, obtain the pricing data and compute the correlation for each pair. Note that this computation can be expensive, so you may want to limit the amount of assets that the user can select. For example, you might want to limit the set of selectable assets to `N=10` assets. Display the correlation on the web page in matrix form. The resulting correlation matrix should be symmetric: $A = A^T$, so feel free to show the upper or lower triangular if it is more clear. You may also want to apply a color map based on the correlation values so that large values are more salient.

You are free to choose whichever display platforms and tools you like (e.g. React, Dash, Streamlit, etc.). You may also use libraries to compute the correlations, although the computation must be done in your code. Upon completion of this task, please provide the following deliverables:

1. Git repository containing the code for hosting the dashboard. To send via email, one option is to create a git bundle file (e.g. `git bundle create solution.bundle --branches --tags`).
2. Instructions on how to run the code and view/interact with the webpage.
3. Screenshot(s) highlighting functionality and showing the resulting correlations.
4. Commentary on how to structure the code so that the data could be updated in real-time.