**Project Overview**

This project is a cryptocurrency data pipeline designed to analyze the relationship between various cryptocurrencies and Bitcoin. It uses the CoinMarketCap API to fetch data and performs a series of transformations and calculations to provide insights such as:

1. The universe of cryptocurrencies.
2. Real-time pricing and ranking data.
3. Relationships between Bitcoin and other cryptocurrencies.
4. Average 24-hour percent changes for each cryptocurrency across multiple runs.

**Code Structure**

The project is divided into four steps, each implemented in a separate Python script for modularity and clarity.

**Step 1: Coin Universe Generator (step\_1\_coin\_universe\_generator.py)**

The purpose of this first Python file is to ensure the API key is functional and capable of returning data. This is a crucial validation step to verify the API connection before moving forward.

In the second section of the code, the script generates and saves the complete list of cryptocurrencies and their metadata, outputting a file named coin\_universe.csv. This file contains fields such as id, name, symbol, and rank, which are used in subsequent steps of the pipeline.

The output of this file will be a file named **coin\_universe.csv** containing the full list of cryptocurrencies and their metadata. Here is an example of what the produced csv will look like:

A table with numbers and letters

Description automatically generated

**Step 2: Pricing Data Fetcher (step\_2\_pricing\_data\_fetcher.py)**

This step focuses on fetching and saving real-time pricing data for a configurable list of cryptocurrencies. It uses the pre-defined file (coins\_to\_track.csv) to identify which cryptocurrencies to track and fetches their pricing and ranking data from the API.

The script outputs timestamped CSV files (e.g., **pricing\_data\_YYYYMMDD\_HHMMSS.csv**), ensuring that each run's data is stored independently for historical tracking and further analysis. Here is an example of what the produced csv will look like:

A screenshot of a computer

Description automatically generated

**Step 3: Bitcoin Relationship Analysis (step\_3\_bitcoin\_relationship\_analysis.py)**

This step is designed to analyze the relationship between Bitcoin and other cryptocurrencies by calculating the absolute difference in their 24-hour percent changes. Using the pricing data generated in Step 2, this script identifies how closely each cryptocurrency's price movement aligns with Bitcoin's.

The output of this step is a timestamped CSV file, named with the format: **btc\_relationship\_YYYYMMDD\_HHMMSS.csv**. This file includes the absolute difference values, making it easier to perform further trend analysis or comparisons.

You can run this step multiple times to get the bitcoin relationship for the latest timestamp. Here is an example of what the produced csv will look like:

A screenshot of a computer

Description automatically generated

**Step 4: Average Percent Change Calculator (step\_4\_average\_calculator.py)**

The purpose of this step is to calculate the average 24-hour percent change for each cryptocurrency across multiple runs. By aggregating data from several runs, it provides a view of how each cryptocurrency performs over time.

This step combines data from multiple timestamped CSV files generated in Step 3 (e.g., **btc\_relationship\_YYYYMMDD\_HHMMSS.csv**) and calculates the average 24-hour percent change for each cryptocurrency. The key function is it reads the data from all relevant CSV files in the specified directory, computes the averages, and saves the results to a summary file.

The output is a single CSV file named **average\_percent\_change\_summary.csv.** This file contains the symbols of the cryptocurrencies and their calculated average 24-hour percent changes, offering a comprehensive overview of their performance trends. Here is an example of what the produced csv will look like:

A screenshot of a computer

Description automatically generated

**Setup Instructions**

**1. Set Up the Environment**

1. Install PipEnv:
   1. pip install pipenv
2. Navigate to the project directory:
   1. cd /path/to/project
3. Create and activate a virtual environment:
   1. pipenv install
   2. pipenv shell
4. Install required dependencies:
   1. pipenv install pandas request

Here is a FlowChart of the overall workflow :

A diagram of a computer network

Description automatically generated

**2. Run the Steps**

1. **Generate the coin universe:**

python step\_1\_coin\_universe\_generator.py

1. **Fetch pricing data (run multiple times for historical data):**

python step\_2\_pricing\_data\_fetcher.py

1. **Analyze Bitcoin relationships (run for each pricing data file):**

python step\_3\_bitcoin\_relationship\_analysis.py

1. **Calculate average percent changes:**

python step\_4\_average\_calculator.py