2/1/2025

Thanadol Tam

Analyzing the Crypto market

API Automation with Python and analysis with R

**Introduction – Project description**

**Project description:** This project involves using Python to pull data on the cryptocurrency market from CoinMarketCap’s API and then automating that process. Next, I take the data gathered and perform simple Data Analytics and Data Science techniques on the data gathered to generate visualizations.

**Automation with Python**

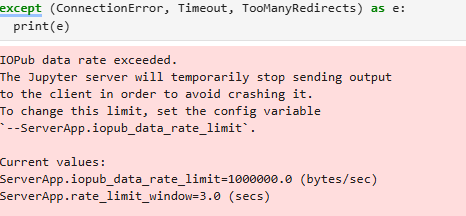
**Step 1: Pulling Data from CoinMarketCap’s API**

* Launched Jupyternote books on Anaconda and used CoinMarketCap’s API’s quick start guide to make an API pull request
* Edited API key to my account's unique API key



**Errors:**

* **Issue Encountered:** While running a Jupyter Notebook, I encountered an IOPub data rate exceeded error, which occurs when the notebook tries to process or print a large volume of data, exceeding the default data transfer limit and causing Jupyter to temporarily stop output to prevent crashes.

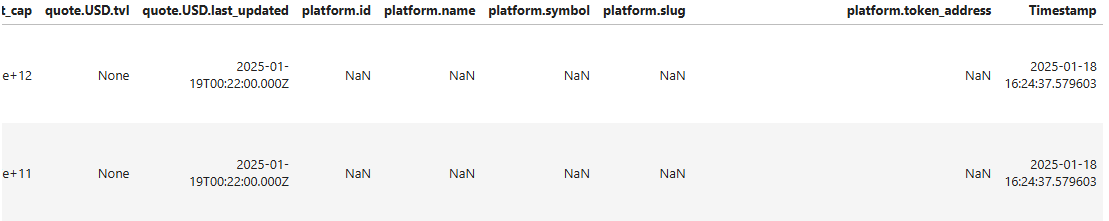


* **Solution Implemented:** To resolve this, I increased the data rate limit by running the command jupyter notebook --NotebookApp.iopub\_data\_rate\_limit=1e10, allowing Jupyter to handle larger data outputs without interruption.

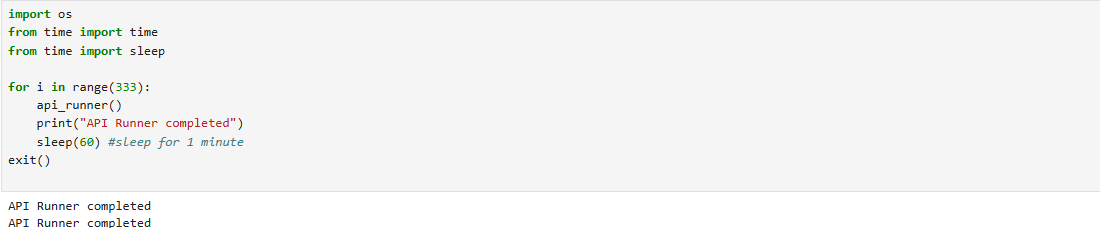
**Step 2: Creating a function for pulling from the API**

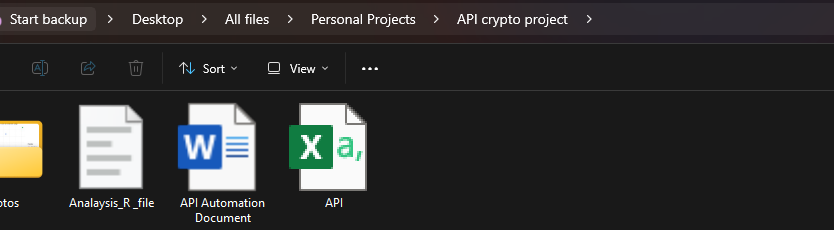
* Created a function “api\_runner” implementing the quick start guide
* Added column “Timestamp” to help track when the data was pulled from the API
* Created code that checks if a CSV file exists; if it doesn’t, it creates one and writes the data with headers, otherwise, it appends new data without rewriting the headers.
* The data pulled from API is uploaded as csv file on local desktop

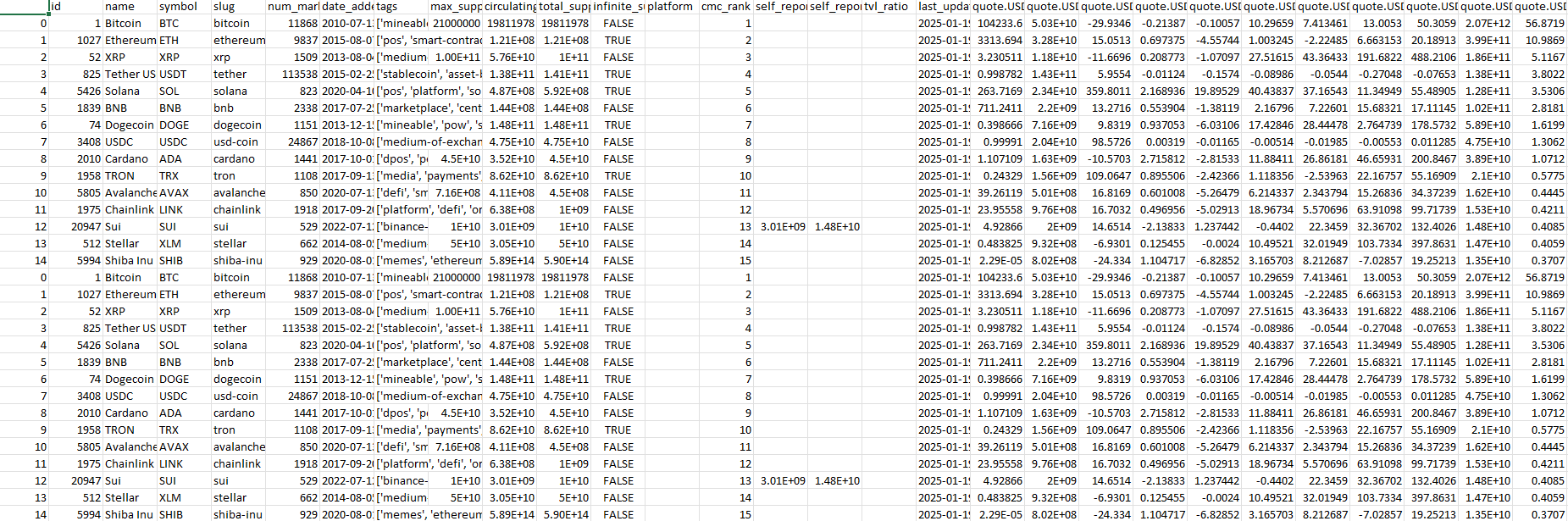


**Displaying example of data frame with “Timestamp” Column**

**Step 3: Creating Automation for pulling API data**

* Developed an automated script to retrieve cryptocurrency data from CoinMarketCap’s API while adhering to free-tier limitations.
* Configured the script to execute api\_runner() **333 times**, aligning with the maximum daily API request limit for free accounts.
* Implemented a **60-second delay** between each request to prevent rate limiting and allow time for data collection.
* For every time function is run prints out “API Runner completed” to confirm completion
* Data is successfully downloaded onto a local device as an csv file

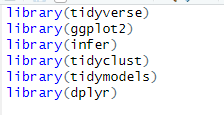




**Data Analysis With R**

**Notes:** this was a small trial of the workflow, so data was gathered over 15 minutes in 1-minute intervals.

**Step 1: Loading all libraries needed for Analysis**

* Loaded Multiple R libraries for analysis
* **Tidyverse** – Provides a collection of packages for data manipulation and visualization.
* **ggplot2** – Used for creating visualizations, including line graphs, histograms, and clustering plots.
* **Infer** – Supports statistical inference and bootstrapping techniques.
* **Tidyclust** – Facilitates clustering analysis, specifically for K-means clustering.
* **Tidymodels** – Provides a framework for modelling and machine learning, including K-means clustering.
* **dplyr** – Used for data manipulation, filtering, grouping, summarizing, and transforming data.

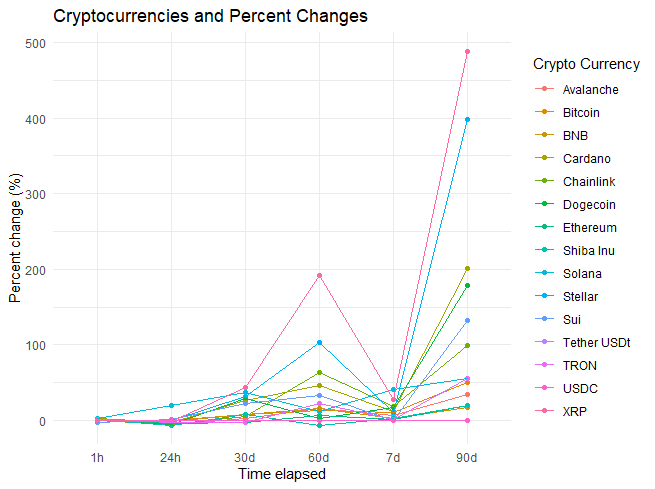
**Step 2: Loading csv file**

* Read csv file into R and assigned the data to variable name “cypto\_data”

**Step 3: Visualizations**

**Visualization 1: Line graph showing different cryptocurrency percent changes over time**

**Explanation of Code:**

* **Grouped the data** by cryptocurrency name and calculated the **average percent change** for each using summarise().
* **Transformed the dataset** into a long format using pivot\_longer(), renaming columns to store percent change types and their values separately.
* **Cleaned the column names** by removing the "quote.USD.percent\_change\_" prefix for better readability.
* **Created a visualization** using ggplot2, plotting percent changes on the x-axis and their values on the y-axis, with different colors representing each cryptocurrency.
* **Used both points and lines** to clearly display trends in percent changes over time.
* **Applied a minimal theme** to enhance readability and improve the overall appearance of the plot.

**Explanation of the Graph:**

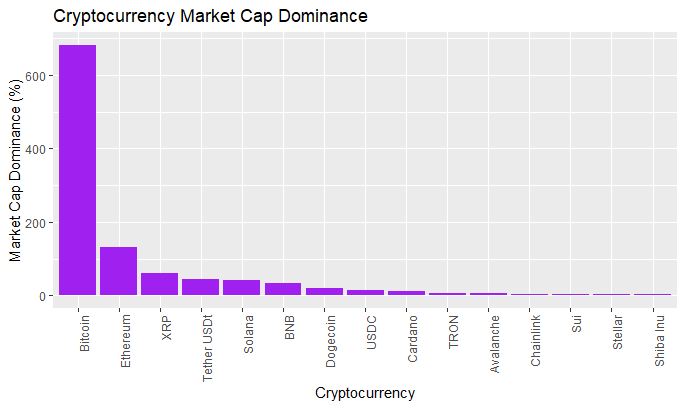
* X-Axis (Time Elapsed): Represents different time intervals (1 hour, 24 hours, 30 days, 60 days, 7 days, and 90 days), showing how cryptocurrency prices have changed over these periods.
* Y-Axis (Percent Change %): Measures the percentage change in cryptocurrency prices, indicating how much each asset has appreciated or depreciated over time.
* Color-Coded Lines: Each cryptocurrency is represented by a different color, as indicated in the legend on the right.

**Key Findings:**

1. **Higher Volatility Over Longer Timeframes:**
   * Cryptocurrencies show small fluctuations in the short term (1h to 24h).
   * As the time frame increases (30d, 60d, 7d, 90d), price changes become more pronounced, reflecting greater volatility.
2. **Avalanche (Pink) Shows the Most Significant Change:**
   * It experiences the highest percent increase by the 90-day mark, indicating a strong upward trend compared to other cryptos.
   * This suggests either strong adoption, market hype, or external influences (e.g., news, ecosystem developments).
3. **General Upward Trend for Most Cryptos:**
   * While price changes vary, most cryptocurrencies exhibit positive growth over time, indicating bullish momentum in the market.
   * Notably, some cryptos show a sharp rise at 60 days and then again at 90 days, suggesting recent market rallies.
4. **Stablecoins (Tether USDT, USDC) Remain Flat:**
   * Minimal percent change, as expected, since stablecoins are designed to maintain a fixed value and do not experience significant price fluctuations.
5. **Diverse Performance Across Cryptocurrencies:**
   * Some cryptos, like Solana, Ethereum, and Chainlink, show steady growth, while others like Shiba Inu and Dogecoin exhibit more irregular spikes, possibly due to speculative trading.

**Visualization 2: Market Cap distribution**

**Explanation of code:**

* **Converted timestamps** into POSIXct format to enable accurate time-based analysis.
* **Extracted time values** (HH:MM:SS) from timestamps for potential intraday trend analysis.
* **Created a market cap dominance bar chart**, using data from “quote.USD.market\_cap\_dominance” column visualizing the relative dominance of cryptocurrencies.
* **Reordered cryptocurrencies** in descending order based on market cap dominance for better comparison.
* **Used a bar chart with actual values (stat = "identity")** to display precise market dominance percentages.
* **Styled the bars with a purple fill** to enhance visual appeal.
* **Rotated x-axis labels (90 degrees)** to improve readability and prevent overlapping cryptocurrency names.

**Explanation of the Graph:**

* **X-Axis (Cryptocurrency):** Lists different cryptocurrencies, sorted in descending order based on their **market cap dominance**.
* **Y-Axis (Market Cap Dominance %):** Represents the percentage of total cryptocurrency market capitalization that each cryptocurrency holds.

**Key Findings:**

1. **Bitcoin Dominates the Market:**
   * Bitcoin has the highest market cap dominance, significantly surpassing all other cryptocurrencies.
   * This aligns with its position as the most widely adopted and valuable cryptocurrency.
2. **Ethereum Holds a Strong Second Place:**
   * Ethereum has a substantial market cap dominance, though much lower than Bitcoin.
   * Its strong position is likely due to its widespread use in **DeFi, smart contracts, and NFTs**.
3. **XRP, Tether (USDT), and Solana Follow:**
   * These cryptocurrencies have notable but much lower market cap dominance.
   * **Tether USDT** is a stablecoin, so its presence highlights its role in liquidity and trading.
4. **Remaining Cryptocurrencies Have Minimal Market Share:**
   * Cryptos like **BNB, Dogecoin, USDC, Cardano, TRON, and Avalanche** hold small portions of the market.
   * **Shiba Inu, Stellar, and Sui have the lowest dominance**, indicating limited impact compared to the leading cryptos.

**Data Science with R**

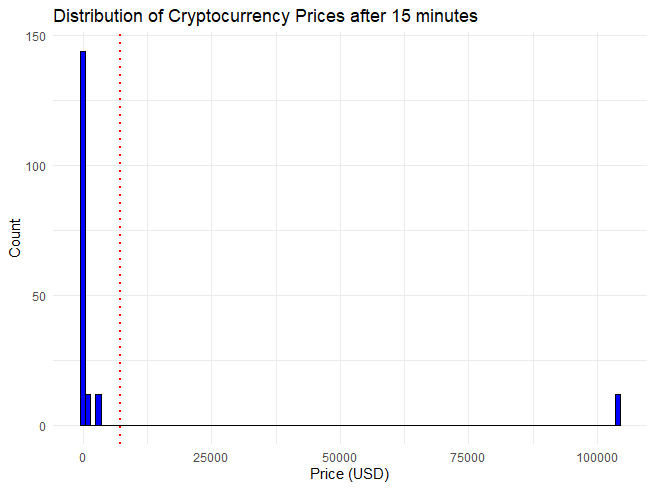
**Bootstrapping price distribution**

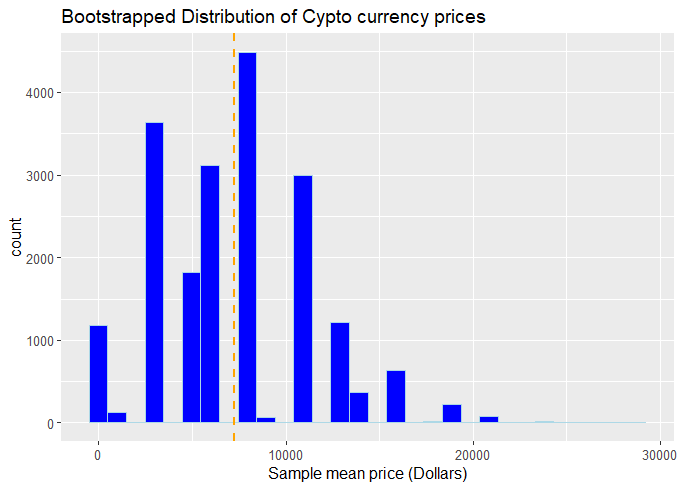
**Bootstrapping Cryptocurrency Price Data for Improved Accuracy**

In my initial analysis, I attempted to visualize the distribution of cryptocurrency prices using a non-bootstrapped dataset. However, due to the limitation of having only **15 minutes of data**, the resulting distribution lacked accuracy and was not representative of broader market trends. To address this issue, I decided to apply bootstrapping to generate a more reliable distribution and calculate an improved mean estimate.

**Initial Price Distribution**

To begin, I created a histogram of cryptocurrency prices using the available 15-minute dataset. The histogram grouped prices into **$1,000 intervals**, allowing for a basic understanding of the distribution. The mean price from this dataset was **$7,239.86**, and I marked this value on the histogram with a red dotted line for reference. However, due to the small sample size, this estimate was likely biased and not fully reflective of broader price behavior.



**Bootstrapping the Dataset for a More Accurate Mean**

To improve the reliability of my price distribution, I implemented a bootstrapping technique. I resampled the data 20,000 times, selecting 40 random samples per iteration with replacement. This method simulates a larger dataset and helps reduce sampling bias. By generating multiple replicates, I created a more robust representation of how cryptocurrency prices are distributed over time.

After resampling, I calculated the mean price for each of the 20,000 bootstrapped replicates. This provided a distribution of sample means rather than individual price values. Finally, I computed the overall mean of these sample means, which resulted in a more refined estimate of **$7,242.10.** This value was then plotted on the histogram as an orange dashed line to highlight the improved estimate compared to the initial mean.

**Conclusion**

By applying bootstrapping, I significantly improved the accuracy of my price distribution analysis, overcoming the constraints of having only 15 minutes of data. The bootstrapped mean ($7,242.10) was slightly higher than the initial mean ($7,239.86), indicating that the raw dataset underestimated the true mean price. This refined estimate provides a more reliable foundation for further analysis, ensuring that my conclusions about cryptocurrency prices are statistically sound. However, a note must be taken on the true mean price as the data I collected was done over a short interval (15 minutes) so to improve accuracy further next time more data can be collected.

**Clustering**

**Why Clustering**

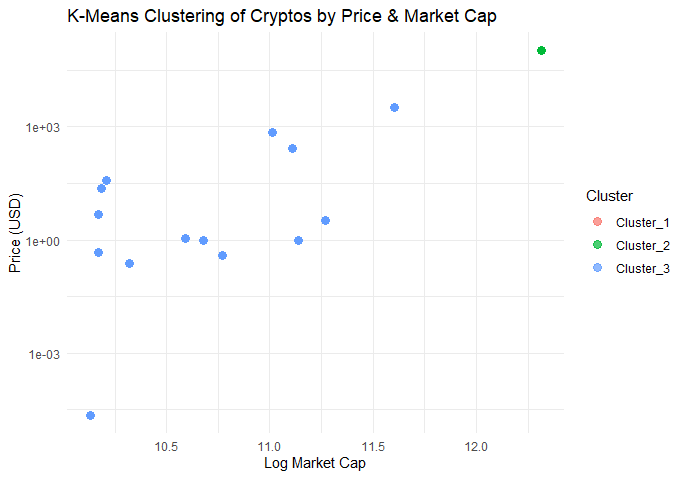
After analyzing the distribution of cryptocurrency prices and applying bootstrapping to improve accuracy, I am now performing clustering on the dataset using K-Means clustering. The goal of this analysis is to group cryptocurrencies based on their market capitalization and price, allowing for a clearer understanding of how different assets compare within the market.

K-Means clustering is a particularly useful technique in this case because it allows us to segment cryptocurrencies into meaningful groups based on numerical attributes, helping traders and analysts understand market trends and investment opportunities more effectively.

**Methodology: Applying K-Means Clustering**

To conduct clustering, I used a **K-Means algorithm with three clusters (K=3)** to classify cryptocurrencies based on their **price** and **market capitalization**. The following steps outline the clustering process:

1. **Data Preparation**
   * Selected **200 samples** from the bootstrapped dataset for clustering.
   * Extracted two key financial features: **price (USD)** and **market capitalization (USD)**.
   * Applied a **log transformation** to market capitalization (log\_market\_cap = log10(market\_cap + 1)) to handle large value disparities and improve cluster separation.
2. **Preprocessing the Data**
   * Standardized the dataset using step\_scale() and step\_center() to ensure all variables are on a comparable scale.
   * This step is crucial for K-Means clustering, as unscaled variables can lead to misleading results due to dominance by larger magnitude values.
3. **Building and Training the K-Means Model**
   * Defined a **K-Means clustering model with three clusters (K=3)**.
   * Trained the model using the **"stats"** engine, which implements the standard K-Means algorithm.
   * Assigned cluster labels to each cryptocurrency in the dataset.
4. **Visualization of Clustering Results**
   * Created a **scatter plot** where:
     + **X-axis represents the log-transformed market capitalization.**
     + **Y-axis represents the price of the cryptocurrency.**
     + **Different colors represent different clusters.**
   * Applied a **log scale to the Y-axis** to improve visualization and highlight relationships across a wide range of prices.

**Clustering Results and Observations**

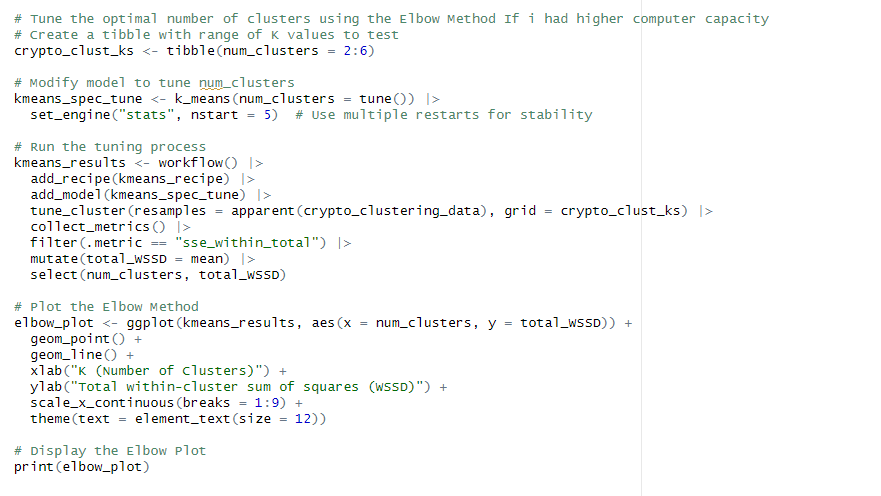
The clustering algorithm classified the cryptocurrencies into **three distinct groups**, which were visualized on a scatter plot. The **x-axis represents the log-transformed market capitalization**, while the **y-axis represents price (USD) on a logarithmic scale**. Each cluster is **color-coded** to highlight the differences in market behavior.

1. **Cluster 1 (Red) – Low Market Cap, Low Price Cryptos:**
   * This cluster consists of **small-cap cryptocurrencies** that have relatively **low prices** and limited market dominance.
   * Many of these assets are priced **below $1**, making them **highly speculative or newly emerging tokens**.
   * Their positioning towards the **left side** of the graph indicates that they have a **lower market capitalization** compared to other assets.
2. **Cluster 2 (Green) – High Market Cap, High Price Cryptos:**
   * This cluster represents **major cryptocurrencies with the highest price and market capitalization**, likely including **Bitcoin and Ethereum**.
   * The presence of a **single green data point in the upper right** suggests that Bitcoin or Ethereum was **isolated due to its extreme values** in both price and market cap.
   * These assets are typically **more stable and widely adopted**, making them the dominant players in the market.
3. **Cluster 3 (Blue) – Mid-Tier Cryptocurrencies:**
   * This cluster consists of **moderate-priced, mid-market cap cryptocurrencies**, which likely include assets such as **Solana, XRP, or BNB**.
   * Positioned in the **middle range** of the market capitalization scale, these cryptos are **established but not as dominant** as Bitcoin or Ethereum.
   * Their market behavior is often influenced by technological advancements, adoption rates, and investor sentiment.

**Issues and limitations**

While the clustering approach successfully grouped cryptocurrencies into **three broad categories**, a few **limitations** were observed:

* **Cluster Imbalance:** The **high-value cluster (Cluster 2) contains only one cryptocurrency**, likely Bitcoin or Ethereum, which suggests that its **dominance skews the clustering results**.
* **Overlapping Mid-Tier and Small-Cap Cryptos:** The **blue and red clusters have considerable overlap**, which indicates that the clustering might not fully distinguish between **small and mid-cap cryptocurrencies**.
* **Potential for More Clusters:** Given the diversity of cryptocurrencies, a **higher number of clusters (K > 3) may provide a more refined segmentation**.
* **Limitations in Tuning Cluster Count:**
  + I attempted to **tune the number of clusters (K) using the Elbow Method**, where I tested values ranging from **K=2 to K=6** and plotted the total within-cluster sum of squares (WSSD).
  + However, my computer lacked sufficient **processing power**, and attempts to run this tuning process frequently resulted in crashes, even after reducing the computational load.
  + This limitation prevented me from determining the **optimal number of clusters**, potentially impacting the accuracy of the segmentation.



**Next Steps for Improvement**

To refine the clustering results and gain deeper insights, I plan to implement the following improvements:

* **Optimize the Number of Clusters:** Given the computational limitations, I will explore **cloud-based or high-performance computing (HPC) resources** to successfully run the **Elbow Method** and determine the ideal **K-value**.
* **Incorporate Additional Variables:** Adding features like **volatility, liquidity, and trading volume** could improve cluster separation and provide a more **detailed classification**.
* **Try Alternative Clustering Methods:** Exploring **hierarchical clustering** or **DBSCAN** could offer **better-defined clusters**, especially for identifying outliers and unique market behaviors.

**Conclusion**

This analysis successfully segmented cryptocurrencies into three broad categories based on price and market capitalization. The results clearly distinguish between dominant assets (Bitcoin, Ethereum), mid-tier cryptocurrencies (e.g., Solana, XRP), and lower-value speculative tokens. While the clustering model provided meaningful insights, further optimization is needed to address cluster imbalances and improve classification accuracy. One of the biggest limitations was hardware constraints, which prevented me from performing Elbow Method tuning to determine the best number of clusters. This limitation suggests that additional computational resources or alternative clustering techniques may be required for more precise market segmentation. By refining the clustering approach, I aim to develop a more robust market segmentation model, enabling a deeper understanding of cryptocurrency price behaviors and investment opportunities.

**Project Final Thoughts**

This project was a valuable learning experience in data automation, statistical analysis, and machine learning. I gained hands-on experience with API integration in Python and data analysis in R, applying techniques like bootstrapping for accuracy and K-Means clustering for market segmentation. Challenges like limited data and computational constraints reinforced the importance of efficient coding and resource management. Moving forward, I want to continue developing my analytical skills by tackling more complex problems and utilizing advanced tools and techniques, such as deep learning, time-series forecasting, and cloud-based computing. This project strengthened my technical foundation and deepened my passion for data analytics and problem-solving, motivating me to explore even more sophisticated analytical approaches.