Affaan Mustafa
University of California, San Diego

Title of Project:

"Employing Causal Machine Learning to Identify and Predict Market Manipulation in Microcap Cryptocurrencies Through Social Media Analysis: A Comprehensive Study with High-Performance Computing"

Detailed Description of Project:

This project aims to employ advanced causal machine learning techniques alongside high-performance computing to uncover the dynamics of market manipulation in microcap cryptocurrencies, as indicated through social media sentiment. Building on preliminary investigations, the research will undertake a more detailed analysis using high-parameter data from diverse social media platforms to establish true causal relationships. Leveraging the San Diego Supercomputer Cluster, the project will utilize parallel computing algorithms and scalable data clustering to efficiently process vast datasets. The summer timeline is dedicated to developing methodologies, collecting and analyzing data, and disseminating findings.

Research Question:

How can the integration of advanced causal machine learning techniques with high-performance computing enhance the identification and understanding of market manipulation signals in microcap cryptocurrencies through social media sentiment analysis? How can we use our findings to combat economic and financial crime and increase social welfare?

Significance of the Project:

Achieving a deeper causal understanding of market manipulation through this approach has the potential to significantly influence regulatory practices and ensure market integrity. The research provides not only academic contributions but also practical tools for regulators and market participants to combat fraud.

Preliminary Work and Relevant Experience:

Preliminary research has explored the correlation between social media sentiment and market manipulation in microcap cryptocurrencies. My background in quantitative finance and machine learning, combined with hands-on experience from this initial work, uniquely positions me to advance this research using complex analytical techniques and high-performance computing resources.

Methods and Theoretical Frameworks:

This study will harness a variety of causal machine learning methodologies, such as propensity score matching and structural equation modeling, executed on the San Diego Supercomputer Cluster. Data will be managed using tensors and 2D arrays for efficient processing, with Kubernetes for scalable deployment. Theoretical frameworks will draw from

Affaan Mustafa
University of California, San Diego

financial market theory and behavioral finance, enriched with cutting-edge computational techniques.

Timeline of the Project:

Month 1: Development of methodologies and setup of data infrastructure on the supercomputer.

Month 2: Intensive data collection and preliminary analysis using parallel computing algorithms.

Month 3: Deep analysis and interpretation of results, leveraging optimized Python and Bash scripts for high-performance computing tasks. Preparation of findings for dissemination.

Funding Requirements:

Access to the San Diego Supercomputer Cluster is essential for processing the extensive parameter space required for this causal analysis. The requested funding will support the use of this high-performance computing resource, ensuring the project can efficiently process and analyze large-scale data.

List of References Cited:

References will include seminal works in causal analysis, machine learning, financial market manipulation, and high-performance computing applications. As well as the references in the below preliminary paper.

Discovering Market Manipulation Through Social Media Sentiment Analysis in Microcap Cryptocurrencies

August 13, 2023

Affaan Mustafa | University of California, San Diego

0.1 Abstract

This research delves into the effects of social media sentiment, with a primary focus on platforms such as Reddit and Twitter, on financial markets, specifically emphasizing the influence on microcap cryptocurrencies. The study begins with a literature review, highlighting key events such as the collective investment in heavily shorted stocks by Reddit users in 2021, an event that significantly altered the financial landscape. The research then narrows its focus to investigate the correlation between social media sentiment and price volatility in the microcap cryptocurrency markets, those defined as having a market cap under 1 million USD. This approach allows for a more controlled study, minimizing the influence of confounding variables and paving the way for causal inference. The exploration of strategies leveraged by cryptocurrency developers, including artificially-induced hype, underscores the potential for market manipulation. By applying existing analysis methods and statistical tools, this research seeks to demonstrate a link between price fluctuations in these microcap digital assets and the sentiment of the Reddit community. The findings indicate that societal and technological advancements are driving a new era in the financial sector. However, the research concludes by emphasizing the need for caution, as the impact of social media sentiment on financial markets is complex and multifaceted, and potentially subject to manipulation.

0.2 Introduction

As the digital era continues to evolve and internet 3.0 technologies proliferate, a significant transformation in our approach to currency is underway. The introduction of cryptocurrencies and brokerage platforms such as Robinhood has democratized investing, making it accessible to a broader demographic. This paper aims to address the question: does social media sentiment serve as an outlet to influence volatility in the financial markets, and could it amount to market manipulation?

In addressing this question, we narrow our focus to microcap cryptocurrencies, which we define as those with a market cap of under 1 million USD. We explore the correlation between social media sentiment, primarily from Reddit, and market indicators such as volume and price in these microcap cryptocurrency markets. This research illuminates the potential for market manipulation via artificially created hype and illustrates how aggregated individual investments can mirror the impact of a single large investment by a corporation. By using existing analysis methods and statistical tools, we aim to ascertain whether there is a correlation between the price of these microcap digital assets and the sentiment of the Reddit community.

This approach, focusing specifically on microcap cryptocurrencies, is intended to mitigate the influence of confounding variables that may skew results in larger cryptocurrency markets. This step towards causal inference aims to provide concrete evidence of market manipulation within this subset of the market. The analysis of notable events, such as the collective investment in shorted stocks by Reddit users in 2021, underscores the potential of social media sentiment to cause significant shifts in market prices. However, it also highlights the need for further investigation to understand the intricate dynamics between social media sentiment and financial market trends. Ultimately, this research seeks to provide a nuanced understanding of how social media sentiment influences microcap cryptocurrency markets, offering insights that could pave the way for future studies in this rapidly evolving field.

0.3   Background

Understanding this involves grappling with key contextual concepts. Cryptocurrencies are digital assets that use intricate algorithms and methodologies to authenticate transactions on a network known as the blockchain. This blockchain, as Frankenfield explains, is "a distributed ledger enforced by a disparate network of computers," rendering transactions publicly verifiable and incredibly secure. The idea of cryptocurrencies was born out of the desire for decentralization and secure access to personal assets without middleman or government interference. Since their inception in the late 2000s and early 2010s, cryptocurrencies have gained enormous popularity, with the collective market cap exceeding 1 trillion USD (Pound).

Traditional financial markets, encompassing the stock market, bond market, commodities market, and physical mineral markets, have also become easily accessible to the average person through phone apps and websites. The intuitive UI (User Interface) design of these platforms mimics features of popular social media, simplifying investment for the average consumer.

While companies employ analysts, investment bankers, and data scientists to optimize investments, the average investor often turns to social media for advice. This paper primarily focuses on Twitter, examining the correlation between user, developer, and company sentiment on this platform and the fluctuations of indicators such as volume and price in the collective financial markets. As Dulău (2) suggests, "The analysis of the tweets performed to extract the opinion about the topics referred to in their text, represents a good indicator of popularity." Twitter's large user community involved in cryptocurrency markets, along with its interactive nature, make it an ideal focus for this study.

An individual investor's small amount of money, when multiplied by millions of people, can equal the impact of a single large investment by a company. The wealthy population, often skeptical of emerging markets like cryptocurrency, is content with safer traditional methods of growing their wealth. However, someone with little money might opt for high-risk cryptocurrencies, given the potential to reap significant returns.

0.4   Literature Review [Dulău]

To investigate how social media sentiment can influence price fluctuations in cryptocurrencies, we must first establish a method for objectively and quantifiably linking social media sentiment to cryptocurrency. Dulău's study accomplishes this using java programs that extract "data related to cryptocurrencies from several sources, filtering and storing it, respectively applying suitable text processing algorithms on it" (Dulău 2). The next step involves categorizing emotions, which are complex and vary in interpretation, especially when conveyed through text. Despite these

caveats, it is practical to group emotions into "positive, negative and neutral categories" (Dulău 2), a common approach in research.

The challenge lies in overlaying this sentiment data derived from cryptocurrency-related tweets on Twitter and correlating it with significant changes in the markets. The question remains: Can this sentiment analysis tool determine a correlation between the price of a digital asset and the sentiment of the community? My experience in the cryptocurrency markets since 2016, along with the data I have gathered, suggests a direct correlation between the size of the cryptocurrency and the number of tweets about it. This thesis is supported by Dulău's findings, which show the sizes of cryptocurrencies by market cap in figure 3 and daily tweets about the top 3 cryptocurrencies in figure 4 (Dulău 4).

#	Name	Symbol	Market Cap
1	Bitcoin	BTC	$106,712,402,411
2	Ethereum	ETH	$45,751,321,457
3	Ripple	XRP	$18,822,950,002
4	Bitcoin Cash	BCH	$12,900,238,342
5	EOS	EOS	$7,281,904,659

Fig. 3: Ranking of cryptocurrencies based on market share

Fig. 4: Number of daily posts about top cryptocurrencies

However, this correlation does not imply that the number of tweets linearly corresponds with the market cap of a cryptocurrency. One reason for the lack of clear price correlations when using Twitter sentiment of the largest cryptocurrencies is their sheer size. An individual investor's tweet holds less impact on a large asset; a substantial change in price would require a significant number of people to either sell or buy the asset collectively. This could occur through an organized group effort or by chance, such as in response to major news.

0.5 Comparative Analysis

Take, for instance, a notable event in 2021 in which I was an active participant. Along with millions of other investors on a social media platform, Reddit, we collectively invested in heavily shorted stocks that were deemed to have no future. "Shorting" involves borrowing shares from the issuer, selling them, and hoping to buy them back at a lower price to return to the issuer, thereby pocketing the difference. Large corporations had shorted the stocks of companies like "GameStop", "AMC", "Nokia", and "Blackberry", which were presumed to be on the brink of bankruptcy or doomed for failure. The Reddit community rallied its users to purchase as many GameStop stocks as possible through the Robinhood app, raising the stock's price from 19 USD at the start of January to a peak of 480 USD (Wilkins). The hedge funds, caught in a squeeze, had to either double down on their shorts, forcing us to sell our positions, or buy back the shares they owed at a higher price, incurring a loss. The latter scenario unfolded, resulting in several hedge funds being bailed out after losing billions, so as to avoid bankruptcy (Wilkins). This episode, combined with the group's purchases, propelled the price and resulted in profit for us. It is a testament to how social media sentiment

can influence financial markets and cause substantial shifts in price—in this case, an upward shift.

However, this singular example is not sufficient to conclude that social media sentiment directly causes price shifts in markets. In this case, the group had a unified objective—to buy the stock—and there were few competing interests. Therefore, more investigation is required to understand the complex dynamics between social media sentiment and financial market trends.

Now that I've established the categorization of converting the sentiment of cryptocurrency users on Twitter, it is necessary to explore the impact of this sentiment on the markets. It's one thing to establish a correlation between the largest cryptocurrencies and the number of tweets about them. However, concluding that sentiment has a direct causation impact on the markets is another step.

To do this, we must analyze data and market trends from the top cryptocurrencies and compare them with their respective Twitter pages. This requires expanding our focus beyond just the user sentiment segment. To accurately assess whether sentiment influences the price fluctuations and volume of different cryptocurrencies, we must first consider all the players and factors involved. This includes not only considering individual investors and their opinions on a certain crypto but also delving into the marketing strategies of cryptocurrency developers themselves. We must consider how they sell their product, the strategies they use to generate positive or negative sentiment against a competitor, and how they communicate with their community and build their following.

0.6 Literature Review [Park and Lee]

Sentiment has been categorized into three sectors, but the platform where that sentiment is received, Twitter, is also relevant. Twitter offers multiple mechanisms of interaction. The most direct is the following-follower relationship, wherein someone actively chooses to follow an account. Next is the reply-mention network, where people can reply to tweets that appear on their feed without necessarily following the account. As Park and Lee state, "The Pearson correlation between Weiss ratings and indegree centralities was highest in the following–follower network ($r = 0.492$, $p < 0.001$), implying that the more credible a cryptocurrency is judged to be, the better connected it is" (Park and Lee 17).

Larger cryptocurrencies tend to have more reply-mention style of tweets, whereas "minor currencies made the strongest efforts to send information to promote their social media profile and achieve a better online presence" (Park and Lee 14). This results in the following-follower mechanic being more prominent in smaller cryptocurrencies, and as evidenced, it has a greater effect on the trends in those cryptocurrencies.

"As the tweet volume increased, cryptocurrencies were more likely to expand their friends in both inward and outward directions" (Park and Lee 20). This not only refers to minor cryptocurrency developers expanding their currency by branching out from following-follower sentiment into reply-mention, but it also implies that developers partner with each other to stimulate such a phenomenon.

Developers who blend themselves into a consumer role and interact frequently with the community, answering questions, posting ideas, and creating hype through extensive roadmaps, tend to generate a positive sentiment. From a psychological perspective, this seems natural and apparent. A developer who is vocal, truthful, transparent, and appears to be working hard eases fears and negative speculation in the community, creating a positive appearance. Easing the fear in consumers and stimulating their greed is a key factor of sentiment in the markets. A fear and greed index is

a tool that depicts the sentiment of the market on a scale ranging from extreme fear to extreme greed. It often indicates the direction of price and market volatility.

The conclusion of this segment is that, as Park and Lee state, "findings appear to support our argument that measuring the social media presence of a cryptocurrency may allow its more accurate evaluation" (Park and Lee 20). This supports my theory that social media trends impact individual perception of a cryptocurrency. For instance, Dogecoin was trending on Twitter after Elon Musk 'tweeted an image from The Godfather, captioning the meme: "One does not simply get into crypto, but when they do, they buy Doge"'. This led to an immediate surge in the price of Dogecoin, highlighting the power that social media, and certain influential figures, can have on the cryptocurrency market.

This isn't to suggest that any tweet about a cryptocurrency will immediately cause a spike in its price, nor that social media sentiment is the only factor that affects cryptocurrency prices. There are many other variables at play, including the overall state of the economy, the actions of governments and regulatory bodies, and technological advancements in the field of cryptocurrency. However, it does suggest that social media sentiment is a significant factor that should be taken into account when analyzing the cryptocurrency market.

While it may be challenging to assert with absolute certainty that sentiment directly influences market movement in all instances, it is nevertheless crucial to consider market sentiment as a consequential factor in market shifts, as demonstrated earlier. Substantial evidence suggests a correlation between sentiment and price fluctuation. My theory posits that even if the initial trigger isn't connected to social media, the public's reaction to such an event can either amplify the price movement in the same direction or reverse it.

0.7   Literature Review [Dipple et al.]

The rational expectations theory, a popular concept in macroeconomics, asserts that people's current expectations of the economy can influence the future state of the economy (Tardi). Applying this theory to the cryptocurrency market and social media sentiment brings us a step closer to establishing causation. As Dipple et al. pointed out, "a large volume of tweets can be produced in response to an event" (13). Although it is self-evident, proving that people's perceptions and reactions on social media lead to further shifts in the markets is not straightforward. Dipple et al. further noted that "Social Media predictions perform significantly better than the prediction without correlation on average" (13). Therefore, the correlation between social media sentiment and price fluctuations is not merely conjectural. There is tangible evidence—albeit contingent on an event occurring—that suggests that social media sentiment can directly influence price fluctuations in the market in the same direction as people's sentiment.

0.8   Comparative Analysis

Establishing causality is crucial to validate sentiment as a determinant factor in impacting price fluctuations and market movements in cryptocurrency. I've personally profited significantly by using social media as an indicator. How is this possible if sentiment doesn't impact price but price impacts sentiment? The answer is that sentiment indeed influences price in many circumstances. I recall investing in a cryptocurrency called "Safemoon'' in early 2021 after watching a TikTok video about it. Initially, sentiment towards the coin was mixed, with many viewing it as a gamble. However, as the developers implemented effective marketing strategies on Twitter—spending significantly on branding, advertising, livestreams, and audits to earn people's trust—the positive sentiment

outweighed the negative, and the price soared over the subsequent months, netting me a return of over 100x.

0.9   Literature Review [Al Nemer et al.]

As Al Nemer et al. observed, investor sentiment "drives cryptocurrencies' price changes across frequencies and investment horizons, particularly in the long run" (17). While this study focused on larger cryptocurrencies over extended periods, there are anecdotal instances where social media sentiment has impacted markets on a short scale, in addition to empirical evidence. Filtering out noise from unwanted variables such as bots—which will be discussed more in the counterargument section of this paper—as well as creating models to distinguish between causality and correlation is what one study achieved.

0.10   Literature Review [Kraaijeveld and Smedt]

Through causality testing, it demonstrated that "Twitter sentiment has significant predictive power for the price returns of Bitcoin, Bitcoin Cash and Litecoin" (Kraaijeveld and Smedt 29). Furthermore, "Twitter sentiment is able to have strong predictive power for the next 1-2 day(s) returns" (Kraaijeveld and Smedt 14). Thus, I have established that social media sentiment does indeed influence trends in the financial markets.

However, it is crucial to acknowledge the limitations of this research and the external factors not considered during the study that could have unpredictable outcomes. The complex nature of these markets includes numerous moving parts and areas where manipulation can occur. One such manipulation is the concept of artificial hype, which involves creating a false positive sentiment stemming from bot accounts or marketers. Even the study recognized this, noting, "the possible effects of Twitter bots on sentiment and/or prices are not researched" (Kraaijeveld and Smedt 14). Regardless of how well these bots were filtered out, they accounted for about "1% - 14% of the Tweets posted" (Kraaijeveld and Smedt 27). These tactics are typically employed to lure unsuspecting people into scams where the developer absconds with investor money or to give the impression that a coin is more desirable than it actually is, enabling early adopters to sell their coins to late arrivals drawn in by the sentiment.

In many large stocks and cryptocurrency markets, investor sentiment is not the only deciding factor. External factors like foreign policy, disaster, inflation, recession, and other variables can affect the markets to the extent that even if sentiment is positive, the markets may decline. It is crucial to recognize this before making investment decisions based solely on Twitter sentiment.

0.11   Exploratory Research

We have conducted thorough literature review now, most of the existing literature focuses on establishing correlation between price and sentiment on a larger scale. No significant steps towards causal inference have been made and neither has exploration been done into the idea that developers or paid promoters are responsible for market manipulation by artificially creating hype and pumping cryptocurrencies to entice the outside investor to invest and then pull all their money from the project. The purpose of this study is to conduct an exploratory study in which we take steps towards causal inference by reducing the noise and confounding variables in our correlation analysis. We approach this problem by first narrowing down the scope of the analysis. We focus on cryptocurrencies we coin as "Microcap" which are cryptocurrencies that are of a stated or estimated

market value of under 1,000,000 USD. We then focus on the timeframe of the past 6 months to have more recent data that is less affected by noise created after the event.

0.12   Methodology

Step 1 - Sample Selection

1. **Objective**:
   - The first step of our analysis is to identify the population of microcap cryptocurrencies and randomly select a subset for detailed examination.
2. **Data Source**:
   - Leveraging CoinMarketCap's API, we fetch data for a broad range of cryptocurrencies.
3. **Filtering Criteria**:
   - We narrow our focus to those considered 'microcap commodities'. These are defined as cryptocurrencies with a market capitalization of under 1 million USD.
4. **Selection Algorithm**:
   - From this filtered list, we use a random selection algorithm to pick 500 microcap cryptocurrencies.
5. **Outcome**:
   - These 500 cryptocurrencies will be utilized as our representative sample for the subsequent stages of our analysis.
6. **Rationale**:
   - By focusing on microcaps, we aim to investigate a unique segment of the cryptocurrency market that may exhibit distinct behaviors or characteristics.
7. **Implications**:
   - The selection of this sample sets the stage for the rest of the study, defining the scope and focus of the analysis.

```
[18]: import requests
      import pandas as pd
      import random
      import json
      import datetime
      import time

      # Load the environment variables

      # We use CMC to get a random sample of 500 microcap cryptocurrencies created in
       ↪the year 2023 that we will then analyze on Twitter and Reddit.

      # Calls the CoinMarketCap API and stores the results in a DataFrame.
      # The DataFrame is then filtered to only include microcap cryptocurrencies.
      # The microcap cryptocurrencies are then filtered to only include
       ↪cryptocurrencies that were added to CoinMarketCap in 2023.
      # The final DataFrame is then randomly sampled to include 500 cryptocurrencies.
       ↪The symbols of the 500 cryptocurrencies are then stored in a list.

      # Calls the CMC API
```

```python
API_KEY = 'your API key'

url = "https://pro-api.coinmarketcap.com/v1/cryptocurrency/listings/latest"
headers = {
    'Accepts': 'application/json',
    'X-CMC_PRO_API_KEY': API_KEY,  # Replace with your actual API key
}

dfs = []
batch_limit = 5000  # Number of cryptocurrencies to fetch in each batch
total_limit = 50000  # Total number of cryptocurrencies to fetch, 50,000.

for start in range(1, total_limit, batch_limit):
    params = {
        'start': str(start),
        'limit': str(batch_limit),
        'convert': 'USD',
        'market_cap_max': '1000000',  # Filter by maximum market cap of $1␣
  ↪million
    }
    response = requests.get(url, headers=headers, params=params)
    data = response.json()

    if 'data' in data:
        dfs.append(pd.DataFrame(data['data']))

# Concatenate all dataframes into a single dataframe
df = pd.concat(dfs, ignore_index=True)

# Preprocess microcaps by dates being within 2023
df['date_added'] = pd.to_datetime(df['date_added'])
start_date = '2023-01-01'
end_date = '2023-07-01'

# Filter the DataFrame to include only rows within the date range
df = df[(df['date_added'] >= start_date) & (df['date_added'] <= end_date)]

# Randomly sample 500 cryptocurrencies from the DataFrame
sample_df = df.sample(n=500, random_state=42)

# Store the symbols of the 500 cryptocurrencies in a list
symbols = sample_df['symbol'].tolist()

# Print the count of sampled symbols
print(len(symbols))
```

500

Step 2 - Data Collection

1. **Objective**:
   - This stage is dedicated to collecting data about mentions of the selected microcap cryptocurrencies from online platforms such as Twitter and Reddit.
2. **Data Sources**:
   - Various APIs, including CoinMarketCap, Twitter, and Reddit, are utilized. Due to the stringent usage limits of Twitter's API, the study primarily relies on data from Reddit.
3. **Subreddit Selection**:
   - A manually compiled list of 30 subreddits known for promoting or discussing microcap cryptocurrencies is used to guide the collection process.
4. **Search Criteria**:
   - The search focuses on posts within these subreddits that mention the selected microcap cryptocurrencies, concentrating on data from the past six months.
5. **Data Points**:
   - For each post encountered, key details such as the title, content, number of upvotes, subreddit, date, and author are gathered, creating a rich dataset.
6. **Storage**:
   - The collected data is stored in CSV files for future reference and ease of manipulation during subsequent phases of the research.
7. **Rationale**:
   - The collection of this data enables the study to investigate the online discussion and sentiment surrounding microcap cryptocurrencies.
8. **Implications**:
   - The success of this step is crucial for the rest of the analysis as it forms the primary dataset that will be explored and analyzed in later stages.
9. **Note on Twitter**:
   - While Twitter data could be valuable, the API limitations necessitated a focus on Reddit. Code for Twitter data collection can be provided to researchers interested in expanding the data scope.

```
[166]:  # Now that we have a list of  microcap cryptocurrencies, we can use the Twitter␣
        ↪API to search for tweets that mention these cryptocurrencies.

        ## initialize the twitter API

        consumer_key = 'your twitter client ID'
        consumer_secret = 'your twitter client secret'

        # Define the bearer token associated with your project
        bearer_token = 'your twitter bearer token'

        # Define the keywords for searching tweets that mention the 49 microcap␣
        ↪cryptocurrencies

        keywords = [f'\${symbol}' for symbol in symbols]

        # Join the keywords with OR operator for the Twitter API query
```

```python
query = " OR ".join(keywords)

# Define the start and end dates for the Twitter API query

end_date = datetime.datetime.now()
start_date = end_date - datetime.timedelta(days=180)  # 6 months

# Define the parameters for the API request [notice that we are using the v2␣
 ↪endpoint, not v1, therefore the recent endpoint is different and only allows␣
 ↪for 100 tweets per request as well as 7 day limit]

start_time = "your start time"
end_time = "your start time + 7 days"

# Define the parameters for the API request
params = {
    "query": query,  # use the query to search for tweets
    "tweet.fields": "created_at",
    "start_time": start_time,
    "end_time": end_time,
    "expansions": "author_id",
    "user.fields": "public_metrics",
    "max_results": 100,
}


# Define the headers with bearer token
headers = {
    'Authorization': f'Bearer {bearer_token}'
}

# Fetch multiple pages of tweets and store filtered data until reaching 100␣
 ↪tweets that meet the criteria of having at least 5000 followers, higher␣
 ↪limits can be achieved by using the paid version of the API]
# be aware you may need to call multiple times to get a larger sample size
data = []
while len(data) < 100:
    if len(data) > 0:
        params['next_token'] = next_token
    response = requests.get("https://api.twitter.com/2/tweets/search/recent",␣
 ↪params=params, headers=headers)
    if response.status_code != 200:
        raise Exception("Request returned an error: {} {}".format(response.
 ↪status_code, response.text))
    json_response = response.json()

    # Extract the needed data and store filtered tweets in the list
```

```python
    for tweet in json_response['data']:
        user = next((u for u in json_response['includes']['users'] if u['id']
↪== tweet['author_id']), None)
        if user and user['public_metrics']['followers_count'] >= 5000:
            data.append([tweet['text'], user['username'],
↪user['public_metrics']['followers_count']])
            if len(data) == 100:
                break

    # Get the next token from the response, if it exists
    next_token = json_response.get('meta', {}).get('next_token')
    if not next_token:
        break  # no more pages

# Convert the list to a DataFrame
df_inuse = pd.DataFrame(data, columns=['tweet', 'username', 'followers'])
```

```python
# check the dataframe to make sure it loaded correctly

print(df_inuse.head())
```

```python
#check the number of tweets in the dataframe

df_inuse.count()
```

```python
#if you have more api calls to make, you can use the next token to make the
↪next call

# Define the parameters for the API request

consumer_key = 'your client ID'
consumer_secret = 'your client secret'

# Define the bearer token associated with your project
bearer_token = 'your bearer token'

# Define the keywords for searching tweets that mention the 49 microcap
↪cryptocurrencies
keywords = [f'\${symbol}' for symbol in symbols]

# Join the keywords with OR operator for the Twitter API query
query = " OR ".join(keywords)

# Define the start and end dates for the Twitter API query

end_date = datetime.datetime.now()
start_date = end_date - datetime.timedelta(days=180)  # 6 months
```

```python
# Define the parameters for the API request [notice that we are using the v2
 ↪endpoint, not v1, therefore the recent endpoint is different and only allows
 ↪for 100 tweets per request as well as 7 day limit]

start_time = "your start time"
end_time = "your start time + 7 days"

# Define the parameters for the API request
params = {
    "query": query,  # use the query to search for tweets
    "tweet.fields": "created_at",
    "start_time": start_time,
    "end_time": end_time,
    "expansions": "author_id",
    "user.fields": "public_metrics",
    "max_results": 100,
}


# Define the headers with bearer token
headers = {
    'Authorization': f'Bearer {bearer_token}'
}


# Fetch multiple pages of tweets and store filtered data until reaching 50
 ↪tweets
data2 = []
while len(data2) < 100:
    if len(data2) > 0:
        params['next_token'] = next_token
    response = requests.get("https://api.twitter.com/2/tweets/search/recent",
 ↪params=params, headers=headers)
    if response.status_code != 200:
        raise Exception("Request returned an error: {} {}".format(response.
 ↪status_code, response.text))
    json_response = response.json()

    # Extract the needed data and store filtered tweets in the list
    for tweet in json_response['data']:
        user = next((u for u in json_response['includes']['users'] if u['id']
 ↪== tweet['author_id']), None)
        if user and user['public_metrics']['followers_count'] >= 5000:
            data.append([tweet['text'], user['username'],
 ↪user['public_metrics']['followers_count']])
            if len(data) == 100:
                break
```

```
    # Get the next token from the response, if it exists
    next_token = json_response.get('meta', {}).get('next_token')
    if not next_token:
        break  # no more pages

# Convert the list to a DataFrame
df_new = pd.DataFrame(data, columns=['tweet', 'username', 'followers'])

# Concatenate the new DataFrame with the old DataFrame
df_combined = pd.concat([df_inuse, df_new], ignore_index=True)

# Remove duplicates
df_combined = df_combined.drop_duplicates()

# Reset the index
df_combined = df_combined.reset_index(drop=True)
```

```
[19]: # save the dataframe to an excel file
      # move on to the next step and for the reddit data

      # df_inuse.to_excel('tweets.xlsx', index=False)
      sample_df['date_added'] = sample_df['date_added'].dt.tz_localize(None)
      sample_df.to_excel('microcaps.xlsx', index=False)
```

Data Scraping Summary

1. **Results**:
   - The scraping process yielded 50 tweets before hitting the API limit and a substantial 36158 posts from 30 selected subreddits.
2. **Focus on Reddit Data**:
   - Due to the limited number of tweets obtained, the study's focus will be on the Reddit data, which contains potential mentions of one or multiple of the 500 analyzed cryptocurrencies.
3. **Tickers and Unique IDs**:
   - It's important to note that multiple cryptocurrencies may share the same ticker symbol. However, CoinMarketCap's API assigns a unique ID to each currency, ensuring clarity in the analysis.
4. **List of Cryptocurrencies and Subreddits**:
   - The tickers for the 500 randomly sampled cryptocurrencies and the 30 subreddits scraped are documented in the code blocks below.
5. **Rationale**:
   - By focusing on Reddit data and utilizing unique IDs, the study aims to provide a precise and robust analysis of the online discussion surrounding microcap cryptocurrencies.

```
[1]: import praw
     import datetime
     import time
```

```python
import pandas as pd
import random

# Read the list of microcaps from the Excel file previously created when
 ↪scraping the list of microcaps from CoinMarketCap
df_microcaps = pd.read_excel('microcaps.xlsx')
df_microcaps_100 = df_microcaps.sample(n=100) # take a subset of 100 microcaps
 ↪from our 500 microcaps for this call to handle reddit api better.
microcaps = df_microcaps_100['symbol'].to_list()
print(microcaps)

# Scrape the subreddits for the 100 microcap subset of our 500 microcaps sample
 ↪we previously identified through random sampling using CoinMarketCap
reddit = praw.Reddit(
    client_id="your client id",
    client_secret="your client secret",
    user_agent="microcap_cryptos",
)


# sample size of 30 subreddits to scrape all specified subreddits deal with
 ↪microcaps

subreddits = ['SatoshiBets', 'Shitcoins', 'MoonBets', 'PancakeswapICO',
 ↪'AllCryptoBets', 'CryptoNews', 'airdrops',
              'ethtrader', 'CryptoMoonShots', 'shitcoinmoonshots',
 ↪'Crypto_General', 'CryptoMars', 'crypto_currency',
              'SatoshiStreetBets', 'BountyICO', 'CryptoCurrencyClassic',
 ↪'bscbombs', 'CryptoIDOS', 'SHIBADULTS',
              'dogecoin', 'BSCMoonShots', 'BitcoinDiscussion', 'Shibainucoin',
 ↪'CryptoMoon', 'Shibu_Inu', 'ico',
              'ICOAnalysis', 'CryptocurrencyICO', 'CryptoMarsShots',
 ↪'cryptostreetbets']  # list of subreddits you're interested in
keywords = microcaps  # list of keywords you're interested in

# search all posts in all 30 subreddits for the 100 microcaps, collect the
 ↪title, content, upvotes, subreddit, date, and author of each post, and save
 ↪the results to a csv file.

all_posts = []

for subreddit_name in subreddits:
    posts = []
    subreddit = reddit.subreddit(subreddit_name)
    for keyword in keywords:
        for submission in subreddit.search(keyword, time_filter='all'):
            posts.append({
```

```python
                'Title': submission.title,
                'Content': submission.selftext,
                'Ups': submission.ups,
                'Subreddit': str(subreddit),
                'Date': datetime.datetime.fromtimestamp(submission.created),
                'Author': str(submission.author),
            })
    df = pd.DataFrame(posts)
    df.to_csv(f'{subreddit_name}_posts.csv', index=False)
    all_posts.extend(posts)
    print(f"Finished scraping subreddit: {subreddit_name}")
    time.sleep(60)  # delay for 60 seconds

    # save the results to a csv file

df_all = pd.DataFrame(all_posts)
df_all.to_csv('all_posts.csv', index=False)
```

```
['BABYBNBTIGER', 'WONKACAP', 'POORPEPE', 'BIZ', 'LGC', 'W$C', 'JW', 'WEWE',
'GP', 'BLC', 'PILADYS', 'TULIP', 'GHA', 'OXO', 'DUDES', 'PEPEPUNK', 'STAR',
'ARCHIVE', 'MIND', 'APECON', 'TOX', 'FRXETH', 'BABYJESUS', 'BEEP', 'MAP',
'MAFIA', 'STR', 'BUNNY INU', 'Jpepe', 'ASTROPEPE', 'MDCX', 'ORCH', 'RAB', 'AUT',
'ZKPAD', 'POT', 'MIDGET', 'BTW', 'LST', '$CHILL', 'PIKA', 'WAIFU', 'PRP', 'USP',
'KEKE COIN', 'NOMNOM', 'BGPT', 'CHUNKS', 'ZACH', 'ACN', 'TIMMY', 'OX', 'SPEPE',
'KABS', 'PEPMCITY', 'FUNC', 'BBO', 'FLUT', 'SMUD', 'MLD', 'LOONEY', 'OBTC',
'$BOLT', 'BKASPA', 'SUPER', 'LPEPE', 'RAGE', 'PSALE', 'PEPELINDA', 'OZONE',
'RAM', 'GYM AI', 'SLURP', 'AMBR', 'PEPELON', '$HOLA', 'FIT', 'WWEMIX', 'FLONA',
'HAM', 'STONKS', 'BCG', 'SCAM', 'GLG', 'PPUSDT', 'RSC', 'ZNX', 'MIRACLE',
'PEPECZ', 'BOLT', 'TOMC', 'DOODIE', '$KENTO', 'ALIT', 'GROOMER', 'GRPEPE',
'TWELVE', 'PUNCH', 'APN', 'VELA']
Finished scraping subreddit: SatoshiBets
Finished scraping subreddit: Shitcoins
Finished scraping subreddit: MoonBets
Finished scraping subreddit: PancakeswapICO
Finished scraping subreddit: AllCryptoBets
Finished scraping subreddit: CryptoNews
Finished scraping subreddit: airdrops
Finished scraping subreddit: ethtrader
Finished scraping subreddit: CryptoMoonShots
Finished scraping subreddit: shitcoinmoonshots
Finished scraping subreddit: Crypto_General
Finished scraping subreddit: CryptoMars
Finished scraping subreddit: crypto_currency
Finished scraping subreddit: SatoshiStreetBets
Finished scraping subreddit: BountyICO
Finished scraping subreddit: CryptoCurrencyClassic
Finished scraping subreddit: bscbombs
Finished scraping subreddit: CryptoIDOS
```

```
Finished scraping subreddit: SHIBADULTS
Finished scraping subreddit: dogecoin
Finished scraping subreddit: BSCMoonShots
Finished scraping subreddit: BitcoinDiscussion
Finished scraping subreddit: Shibainucoin
Finished scraping subreddit: CryptoMoon
Finished scraping subreddit: Shibu_Inu
Finished scraping subreddit: ico
Finished scraping subreddit: ICOAnalysis
Finished scraping subreddit: CryptocurrencyICO
Finished scraping subreddit: CryptoMarsShots
Finished scraping subreddit: cryptostreetbets
```

```python
[10]: import pandas as pd

      # Read the CSV file into a new DataFrame
      df_all = pd.read_csv('all_posts.csv')

      # Get the number of posts
      num_posts = len(df_all)
      print(f"Number of posts: {num_posts}")
```

```
Number of posts: 36158
```

Step 3 - Data Preprocessing:

1. **Loading Data**:
   - The collected Reddit posts are loaded from the CSV file using the pandas library.
2. **Date Filtering**:
   - The 'Date' column is converted to datetime format, and posts from the year 2023 are filtered out using a boolean mask.
3. **Time Granularity Adjustment**:
   - The time of each post is rounded to the nearest 15 minutes, and the posts are grouped by the rounded time, content, and title.
4. **Aggregation**:
   - The upvotes for posts falling within each group are summed up, measuring the overall impact and reach.
5. **Text Cleaning**:
   - The text is cleaned by removing punctuation, converting to lowercase, tokenizing into words, and lemmatizing. Common "stop words" are also removed.
6. **Preparation for Analysis**:
   - The processed text and other relevant details of each post are prepared for sentiment analysis.
7. **Outcome**:
   - By the end of this step, a preprocessed and aggregated dataset is ready, setting the stage for sentiment analysis.

```python
[2]: import nltk
     from nltk.corpus import stopwords
```

```python
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
import re
from collections import defaultdict

# Load the reddit data
reddit_posts = pd.read_csv('all_posts.csv')

# Load the microcaps data
microcaps = pd.read_excel('microcaps.xlsx')

# Get the CMC IDs as a list
cmc_id_list = microcaps['id'].tolist()

# Convert the 'Date' column to datetime
reddit_posts['Date'] = pd.to_datetime(reddit_posts['Date'])

# Create a boolean mask for the year 2023
mask = (reddit_posts['Date'].dt.year == 2023)

# Apply the mask to the DataFrame
reddit_posts = reddit_posts.loc[mask]

# Round the time to the nearest 15 minutes
reddit_posts['time_rounded'] = reddit_posts['Date'].dt.round('15min')

# Group by the rounded time and the post text, and aggregate the upvotes
reddit_posts_agg = reddit_posts.groupby(['time_rounded', 'Content', 'Title']).
 ↪agg({'Ups': 'sum'}).reset_index()

stop_words = set(stopwords.words('english'))
lemmatizer = WordNetLemmatizer()

def preprocess_text(text):
    text = re.sub(r'[^\w\s]', '', text)  # remove punctuations
    text = text.lower()  # convert to lowercase
    tokenized_text = word_tokenize(text)  # tokenization
    text = [lemmatizer.lemmatize(word) for word in tokenized_text if word not␣
 ↪in stop_words]  # lemmatization and stopwords removal
    return text

reddit_posts_agg['processed_text'] = reddit_posts_agg['Content'].
 ↪apply(preprocess_text)

# Print the number of remaining posts after aggregation
remaining_post_count = reddit_posts_agg.shape[0]
print(f"Number of remaining posts after aggregation: {remaining_post_count}")
```

```python
# Get the microcap tickers as a list
microcap_tickers = microcaps['symbol'].tolist()

# Create a dictionary to store the post counts for each ticker
tickers_with_posts = defaultdict(dict)

# Iterate through the tickers and count the number of posts for each ticker
for ticker in microcap_tickers:
    mask = reddit_posts_agg['Content'].str.contains(ticker, case=False) |␣
 ↪reddit_posts_agg['Title'].str.contains(ticker, case=False)
    posts_for_ticker = reddit_posts_agg[mask]
    if len(posts_for_ticker) > 10:
        for date, group in posts_for_ticker.
 ↪groupby(posts_for_ticker['time_rounded'].dt.date):
            tickers_with_posts[ticker][date] = len(group)

# Print the dictionary mapping tickers to post counts in descending order
print("Ticker - Number of Posts")
for ticker, posts_data in tickers_with_posts.items():
    total_posts = sum(posts_data.values())
    print(f"{ticker}: {total_posts}")

# Print the count of tickers
ticker_count = len(tickers_with_posts)
print(f"Total number of microcaps: {ticker_count}")

# Create a dictionary mapping from ticker to id for all tickers
ticker_to_id_all = dict(zip(microcap_tickers, cmc_id_list))

# Create a dictionary mapping from ticker to id for the tickers in␣
 ↪tickers_with_posts
ticker_to_id = {ticker: ticker_to_id_all[ticker] for ticker in␣
 ↪tickers_with_posts if ticker in ticker_to_id_all}
```

```
Number of remaining posts after aggregation: 2866
Ticker - Number of Posts
RNM: 32
MIND: 631
DUO: 196
PEPE: 119
TRUTH: 12
SPC: 25
EQUAL: 61
ROPE: 133
RAB: 146
WAIFU: 15
```

```
GOLD: 60
CLOUD: 26
ALL: 1874
WC: 51
CAT: 946
AMC: 31
YDOGE: 13
AVO: 195
DIA: 445
RAM: 960
RIN: 1360
USDT+: 41
NALS: 65
ADS: 360
BOB: 13
SMU: 14
OXO: 21
ROE: 23
BS: 1707
HUNDRED: 18
FUNC: 240
FOMO: 12
USH: 85
GOV: 141
PAI: 293
AI: 2559
ORD: 781
HOW: 1295
XI: 257
GC: 23
STR: 1669
FUR: 221
SCAM: 199
TREND: 113
RAGE: 369
OXC: 21
MOD: 284
MBIT: 15
DROPS: 61
FAST: 247
JW: 12
USP: 11
BORAT: 248
CAL: 931
MAY: 305
MEME: 371
PEO: 389
HAM: 37
```

```
TRAC: 1116
TIME: 1107
THL: 229
OZONE: 258
GEN: 1058
STAR: 802
MG: 97
ALAB: 108
UND: 1246
IND: 1626
MPI: 55
DRA: 170
BTW: 18
FIT: 706
ERIC: 66
COS: 834
ASTRO: 34
ALIT: 567
PAT: 484
FOUR: 31
SCALE: 66
TITI: 144
CHT: 11
MAFIA: 256
MAP: 597
RSC: 29
IBIT: 13
SUPER: 243
RICK: 190
BEAST: 16
MEDIT: 16
MINT: 333
TAIL: 310
EG: 1467
GP: 233
SECT: 103
LST: 27
PP: 2281
BIZ: 15
BANK: 77
OX: 388
ONLINE: 491
MOOT: 18
URS: 389
BIRD: 12
VOC: 33
POT: 721
ARCHIVE: 47
```

```
CUT: 184
THEO: 12
AUT: 882
CURVE: 165
LBR: 17
GAIN: 324
Total number of microcaps: 112
```

Interim Note:

After filtering out cryptocurrencies with fewer than 10 posts, the sample size is reduced to 112 microcaps (note that our sample for post selection was 100 out of the 500, meaning that some posts had mentions of multiple separate cryptos potentially). The post count also decreases from 36158 to 2866 due to the aggregation techniques employed. A dictionary is printed that maps each microcap ticker to its post count in descending order, providing insight into which microcap cryptocurrencies are most discussed on Reddit. Additionally, a dictionary is created from this filtered set, mapping the tickers to their respective CMC ID's while retaining post data and time of post. This step facilitates the collection of price data from CoinMarketCap's API by connecting the tickers to their respective CMC ID's.

Step 4 - Sentiment Analysis:

1. **Objective**:
   - The goal is to determine the sentiment of each Reddit post in the dataset using the VADER sentiment analysis tool.
2. **Tool Selection**:
   - VADER (Valence Aware Dictionary and sEntiment Reasoner) from the nltk library is used. It's designed to work well with social media text.
3. **Sentiment Scoring**:
   - A SentimentIntensityAnalyzer object is created, and a function 'get_sentiment' is defined to return the compound sentiment score, normalized between -1 (most extreme negative) and +1 (most extreme positive).
4. **Application to Dataset**:
   - The function is applied to the content of each Reddit post, and a new column 'sentiment' is created to store the sentiment scores.
5. **Analysis**:
   - Descriptive statistics are used to understand the distribution of sentiment scores in the dataset.
6. **Outcome**:
   - The sentiment analysis phase yields sentiment scores for each post, contributing to the understanding of public perception of the selected microcap cryptocurrencies.

```python
[13]: from nltk.sentiment.vader import SentimentIntensityAnalyzer

sia = SentimentIntensityAnalyzer()

def get_sentiment(text):
    sentiment = sia.polarity_scores(text)
    return sentiment['compound']
```

```
reddit_posts_agg['sentiment'] = reddit_posts_agg['Content'].apply(get_sentiment)
reddit_posts_agg['sentiment'].describe()
```

[13]:
```
count    2866.000000
mean        0.820497
std         0.365466
min        -0.989300
25%         0.827200
50%         0.975300
75%         0.994000
max         0.999700
Name: sentiment, dtype: float64
```

Interim Note - Sentiment Analysis Findings and Price Data Collection:

1. **Findings from Sentiment Analysis**:
   - The sentiment analysis reveals that the compound sentiment is overwhelmingly positive on a scale from -1 to 1.
   - This finding supports the theory of targeted promotion and shilling of these cryptocurrencies, possibly motivated by hidden incentives.
   - Such practices may be orchestrated by developer-led pump and dump schemes or insider groups profiting from tiered rankings, with delayed release of insider information based on subscription level.
   - These activities are often organized through online communities on platforms like Telegram and Discord.
2. **Preparation for Next Step - Price Data Collection**:
   - Before progressing to the next phase, price data for all 112 microcap cryptocurrencies over the past 6 months is scraped from the CoinMarketCap (CMC) API.
   - Price quotes are taken at 15-minute intervals to align with the aggregation of Reddit posts, ensuring consistent time granularity.
3. **Outcome**:
   - With the sentiment analysis complete and the price data collected, the stage is set for the subsequent step of multivariate regression analysis.
   - The positive sentiment identified adds a layer of complexity to the analysis, hinting at underlying manipulative practices within the market for microcap cryptocurrencies.

[15]:
```python
import requests

# Define your CoinMarketCap API key
cmc_API_KEY = 'your api key'

# Define the base URL for the CoinMarketCap API
cmc_url = 'https://pro-api.coinmarketcap.com/v3/cryptocurrency/quotes/
 ↪historical'

# Define the headers for the API request
headers = {
```

```python
    'Accepts': 'application/json',
    'X-CMC_PRO_API_KEY': cmc_API_KEY,
}

def get_historical_price(cmc_id, year):
    # Define the time ranges for each month from January to June
    time_ranges = [
        (datetime.date(year, month, 1).isoformat(),
         datetime.date(year, month+1, 1).isoformat() if month < 6 else datetime.
 ↪date(year, month, 30).isoformat())
        for month in range(1, 7)
    ]

    all_quotes = []
    for time_start, time_end in time_ranges:
        parameters = {
            'id': cmc_id,
            'time_start': time_start,
            'time_end': time_end,
            'interval': '15m',
            'convert': 'USD',
            'aux': 
 ↪'price,volume,market_cap,circulating_supply,total_supply,quote_timestamp,is_active,is_fiat'
            'skip_invalid': True
        }

        response = requests.get(cmc_url, headers=headers, params=parameters)

        # Check if the request was successful
        if response.status_code != 200:
            print(f'Request for ID {cmc_id} not successful. Status code: 
 ↪{response.status_code}')
            print(f'Response: {response.text}')
            continue

        response_json = response.json()
        for item in response_json['data'][str(cmc_id)]['quotes']:
            quote = {
                'id': cmc_id,
                'timestamp': item.get('timestamp'),
                'price': item['quote']['USD'].get('price', None),
                'volume_24hr': item['quote']['USD'].get('volume_24hr', None),
                'market_cap': item['quote']['USD'].get('market_cap', None),
                'conversion_timestamp': item['quote']['USD'].get('timestamp', 
 ↪None),
            }
            all_quotes.append(quote)
```

23

```python
        return all_quotes

all_dataframes = []
for ticker, cmc_id in ticker_to_id.items():
    quotes = get_historical_price(cmc_id, 2023)  # Replace 2023 with the year␣
 ↪you want
    df_price = pd.DataFrame(quotes)

    # Convert the 'timestamp' to datetime and set it as the index
    df_price['timestamp'] = pd.to_datetime(df_price['timestamp'])
    df_price.set_index('timestamp', inplace=True)

    all_dataframes.append(df_price)

# Concatenate all the dataframes into a single dataframe
df_all = pd.concat(all_dataframes, ignore_index=True)

# save our price data as a csv file
df_all.to_csv('all_prices.csv')
```

```python
# saving our final 112 microcaps for analysis into a seperate file

final_microcaps = pd.DataFrame({'symbol': list(ticker_to_id.keys())})
final_microcaps.to_excel('final_microcaps.xlsx', index=False)
```

Interim Note - Visualization Tool for Correlation Analysis:

1. **Tool Creation**:
   - For a tangible understanding of potential correlations, a specialized tool is designed to select a random cryptocurrency from the pool of 112 microcaps.
   - This tool presents the historical price data, post count, and the number of upvotes per post, enabling an intuitive visualization of any correlation.
2. **Example Visualization - Ticker 'AVO'**:
   - An illustration of this visualization is provided for the ticker 'AVO'.
   - The first graph, displayed below the code block, portrays the price trend for 'AVO' over the past 6 months.
   - The second output, situated beneath the historical price graph, represents the Reddit posts on the same timescale. Each post is depicted as a dot, with the height indicating the number of upvotes that post received.
3. **Outcome**:
   - This visual representation aids in recognizing a preliminary correlation between the selected variables.
   - It serves as a precursor to the actual correlation analysis, establishing a foundation for more rigorous examination.

```python
import random
import seaborn as sns
```

```python
import matplotlib.pyplot as plt
from matplotlib.dates import DateFormatter

df_all = pd.read_csv('all_prices.csv')

# Set the seed for reproducibility
random.seed(54)

# Select a random ticker from the tickers with posts
random_crypto = random.choice(list(tickers_with_posts.keys()))
random_crypto_id = ticker_to_id[random_crypto]

print(f"Random cryptocurrency with posts: {random_crypto}")
print(f"Total number of filtered posts: {sum(tickers_with_posts[random_crypto].
 ↪values())}")

# Retrieve historical price data for the randomly selected cryptocurrency
df_price_random = df_all[df_all['id'] == random_crypto_id]

# Convert the 'timestamp' to datetime and set it as the index
df_price_random['conversion_timestamp'] = pd.
 ↪to_datetime(df_price_random['conversion_timestamp'])
df_price_random.set_index('conversion_timestamp', inplace=True)

# Plot the historical price data with a fixed x-axis range
plt.plot(df_price_random.index, df_price_random['price'], label='Price')
plt.xlabel('Time')
plt.ylabel('Price')
plt.title(f'Historical Price for {random_crypto}')
plt.xticks(rotation=45, ha='right')  # Rotate x-axis labels if needed
plt.gca().xaxis.set_major_formatter(DateFormatter("%m-%d"))  # Format x-axis␣
 ↪dates
plt.legend()
plt.xlim(df_price_random.index.min(), df_price_random.index.max())  # Set fixed␣
 ↪x-axis range
plt.show()

# Plot the number of upvotes against time using the same x-axis range as the␣
 ↪price plot
random_crypto_df = reddit_posts_agg[reddit_posts_agg['Content'].str.
 ↪contains(random_crypto, case=False) | reddit_posts_agg['Title'].str.
 ↪contains(random_crypto, case=False)]
upvotes = random_crypto_df['Ups']
upvote_dates = random_crypto_df['time_rounded'].apply(lambda x: x.date())
plt.scatter(upvote_dates, upvotes, label='Upvotes', color = 'red')
plt.xlabel('Time')
```

```
plt.ylabel('Upvotes')
plt.title(f'Reddit Posts for {random_crypto}')
plt.xticks(rotation=45, ha='right')  # Rotate x-axis labels if needed
plt.gca().xaxis.set_major_formatter(DateFormatter("%m-%d"))  # Format x-axis
   ↪dates
plt.tight_layout()
plt.xlim(df_price_random.index.min(), df_price_random.index.max())  # Set fixed
   ↪x-axis range
plt.show()
```

Random cryptocurrency with posts: AVO
Total number of filtered posts: 195

Historical Price for AVO

Reddit Posts for AVO
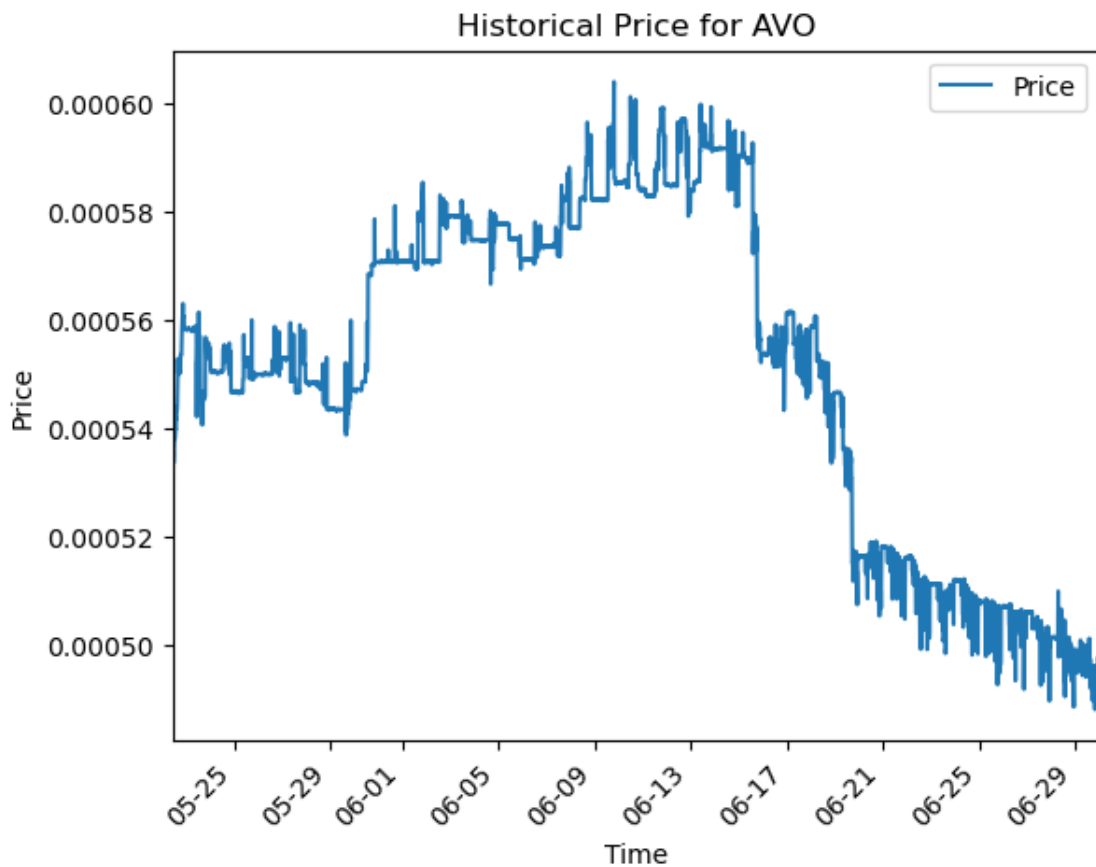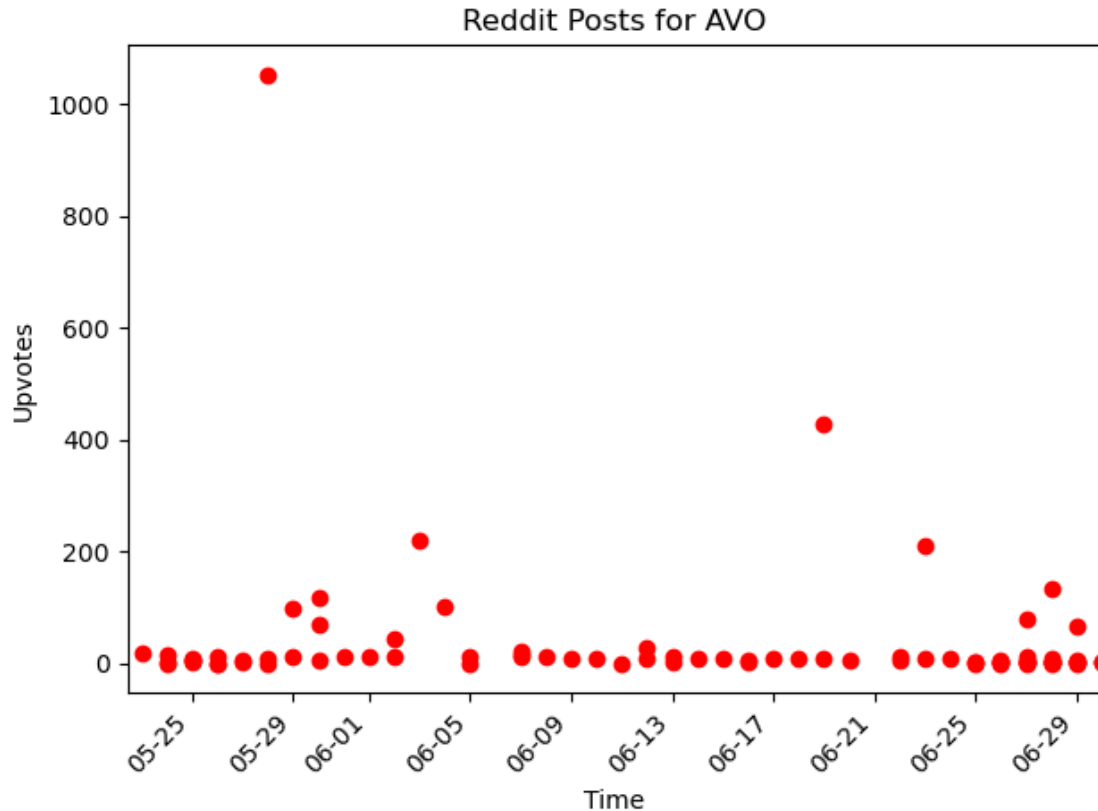
Interim Note - Superimposed Visualization for Correlation Analysis:

1. **Selection of Cryptocurrency - Ticker 'DROPS':**
   - In this visualization step, another random cryptocurrency is chosen from our list of 112, specifically the ticker 'DROPS'.
2. **Superimposed Visualization Technique:**
   - The visualization involves superimposing the graph of Reddit posts on top of the graph of historical price data for 'DROPS'.
   - This approach enables a direct comparison of price trends with Reddit activity on the same graph.
3. **Data Transformation:**
   - For 'DROPS', the upvotes are transformed logarithmically and relatively to align with the price graph.
   - The transformation ensures that the highest number of upvotes takes the value of 1 and the lowest takes the value of 0.
4. **Outcome:**
   - The superimposed visualization provides a more intricate view of the potential correlation between price and Reddit activity.
   - By juxtaposing these metrics on a shared graph, it facilitates an intuitive understanding of how price movements may align with changes in Reddit discussions and upvotes.

27

```
[65]:  import random
       import seaborn as sns
       import matplotlib.pyplot as plt
       import numpy as np
       import pandas as pd

       # Load data
       df_all = pd.read_csv('all_prices.csv')

       # Select a random ticker from the tickers with posts
       random_crypto = random.choice(list(tickers_with_posts.keys()))
       random_crypto_id = ticker_to_id[random_crypto]

       print(f"Random cryptocurrency with posts: {random_crypto}")
       print(f"Total number of filtered posts: {sum(tickers_with_posts[random_crypto].
        ↪values())}")

       # Retrieve historical price data for the randomly selected cryptocurrency
       df_price_random = df_all[df_all['id'] == random_crypto_id]

       # Convert the 'timestamp' to datetime and set it as the index
       df_price_random['conversion_timestamp'] = pd.
        ↪to_datetime(df_price_random['conversion_timestamp'])
       df_price_random.set_index('conversion_timestamp', inplace=True)

       # Create a figure and axis object for the plot
       fig, ax = plt.subplots()

       # Plot the historical price data with a fixed x-axis range
       ax.plot(df_price_random.index, df_price_random['price'], label='Price',
        ↪color='green')
       ax.set_xlabel('Time')
       ax.set_ylabel('Price')
       ax.set_title(f'Price Overlayed on Reddit Posts for {random_crypto}')
       ax.legend()

       # Plot the number of upvotes against time using the same x-axis range as the
        ↪price plot
       random_crypto_df = reddit_posts_agg[reddit_posts_agg['Content'].str.
        ↪contains(random_crypto, case=False) | reddit_posts_agg['Title'].str.
        ↪contains(random_crypto, case=False)]
       upvotes = random_crypto_df['Ups']
       upvote_dates = random_crypto_df['time_rounded'].apply(lambda x: x.date())

       # Calculate the logarithm of upvotes with a scaling factor for volatility
        ↪adjustment
       scaling_factor = 0.5  # Adjust the scaling factor as desired
```

```
log_upvotes = np.log(upvotes + 1) * scaling_factor

# Normalize the transformed upvotes relative to the highest upvote value
max_upvotes = np.max(log_upvotes)
normalized_upvotes = log_upvotes / max_upvotes

# Create a twin y-axis for the normalized upvotes
ax2 = ax.twinx()
ax2.scatter(upvote_dates, normalized_upvotes, label='Upvotes', color='red',␣
 ↪alpha=0.1)  # Set color to green
ax2.set_ylim(0, 1)   # Set y-axis limits from 0 to 1
ax2.set_ylabel('Normalized Upvotes')

plt.xticks(rotation=45, ha='right')  # Rotate x-axis labels if needed
plt.tight_layout()

plt.show()
```
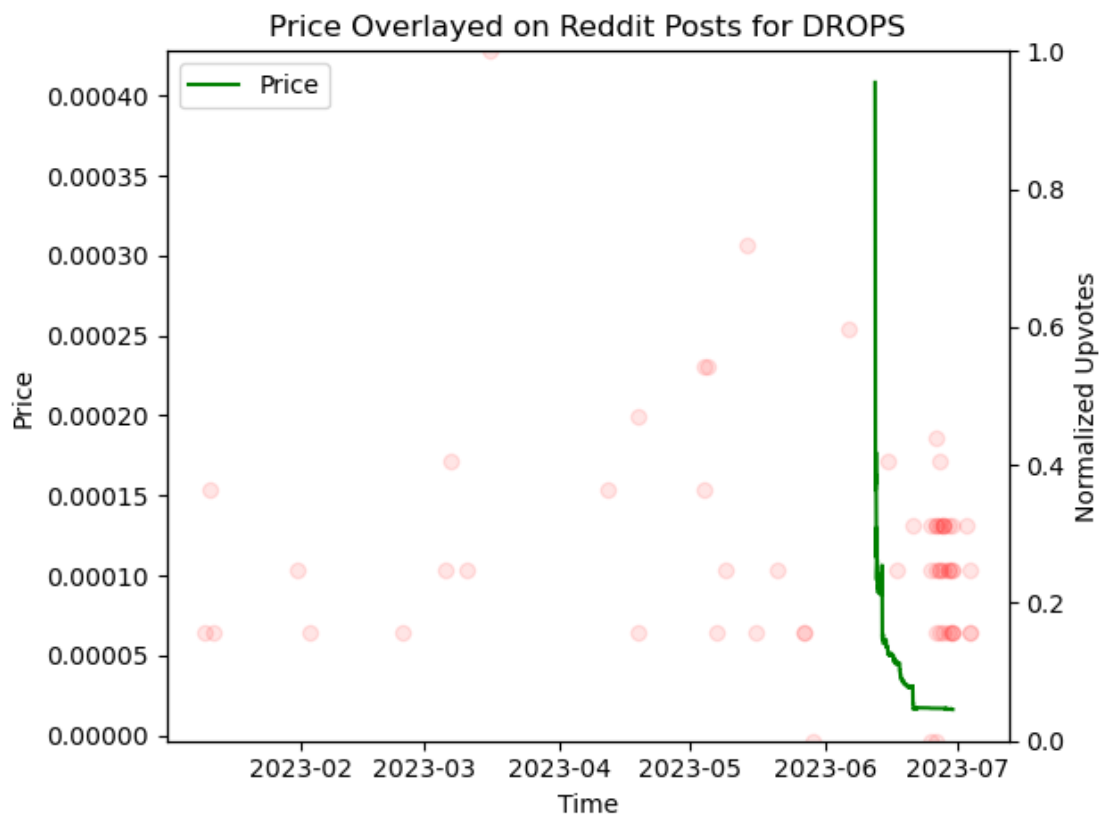
Random cryptocurrency with posts: DROPS
Total number of filtered posts: 61

Price Overlayed on Reddit Posts for DROPS

Step 5 - Correlation Analysis:

1. **Objective**:
   - This step focuses on quantifying the relationship between the number of Reddit posts (which represents discussion volume) and the price of each cryptocurrency within our selected microcaps.

2. **Correlation Calculation**:
   - We employ statistical methods to calculate the correlation between these variables.
   - The correlation coefficient ranges from -1 to +1, where:
     - **Positive Correlation**: Indicates that as the number of Reddit posts increases, the price of the cryptocurrency tends to increase as well, and vice versa.
     - **Negative Correlation**: Implies that as the number of Reddit posts increases, the price of the cryptocurrency tends to decrease, and vice versa.
     - **Zero Correlation**: Suggests that there is no linear relationship between the variables.

3. **Interpretation and Implications**:
   - The correlation analysis helps in understanding how discussion volume on Reddit may influence the price of microcap cryptocurrencies.
   - It provides insights into the potential impact of social media discussions on market dynamics.
   - This information can be instrumental in devising trading strategies, detecting manipulation, or understanding investor behavior related to these specific cryptocurrencies.

4. **Limitations**:
   - It's essential to recognize that correlation does not imply causation. A high correlation may indicate an association but does not prove that changes in Reddit posts cause changes in price.
   - Other confounding factors might influence the observed relationship.

5. **Further Exploration**:
   - Additional analysis could involve looking at the correlation with other variables like sentiment, upvotes, or using lagged versions of the variables to investigate potential lead-lag relationships.

```python
[67]: import warnings
      from scipy import stats
      warnings.filterwarnings('ignore')

      # Create a DataFrame to store the correlation results
      correlation_df = pd.DataFrame(index=list(tickers_with_posts.keys()),
       ↪columns=['Correlation', 'P-value', 'Post Count'])

      # Create a list to store the selected cryptocurrencies
      selected_cryptos = []

      # Iterate through the tickers with posts and perform correlation analysis
      for ticker in tickers_with_posts.keys():
          try:
              # Filter posts based on the ticker
```

```python
        mask = reddit_posts_agg['Content'].str.contains(ticker, case=False) |↩
    ↪reddit_posts_agg['Title'].str.contains(ticker, case=False)
        ticker_df = reddit_posts_agg[mask]

        # Retrieve historical price data for the ticker
        ticker_id = ticker_to_id.get(ticker)
        if ticker_id is None:
            raise KeyError(f"Ticker ID not found for ticker: {ticker}")
        ticker_price_df = df_all[df_all['id'] == ticker_id]

        # Convert conversion_timestamp column to datetime64[ns]
        ticker_price_df.loc[:, 'conversion_timestamp'] = pd.
    ↪to_datetime(ticker_price_df['conversion_timestamp'])

        # Convert the conversion_timestamp column to UTC datetime format
        ticker_price_df.loc[:, 'conversion_timestamp'] =↩
    ↪ticker_price_df['conversion_timestamp'].dt.tz_convert('UTC')

        # Convert the time_rounded column to UTC datetime format
        ticker_df.loc[:, 'time_rounded'] = pd.
    ↪to_datetime(ticker_df['time_rounded']).dt.tz_localize('UTC')

        # Merge the price and posts data on the common timestamp
        merged_df = pd.merge(ticker_price_df, ticker_df,↩
    ↪left_on='conversion_timestamp', right_on='time_rounded', how='inner')

        # Calculate the correlation coefficient and p-value
        correlation_coefficient, p_value = stats.pearsonr(merged_df['price'],↩
    ↪merged_df['Ups'])

        # Store the correlation coefficient, p-value, and post count in the↩
    ↪DataFrame
        correlation_df.loc[ticker, 'Correlation'] = correlation_coefficient
        correlation_df.loc[ticker, 'P-value'] = p_value
        correlation_df.loc[ticker, 'Post Count'] = ticker_df.shape[0]

        # Add the ticker to the selected cryptocurrencies list
        selected_cryptos.append(ticker)

    except KeyError as e:
        print(f"KeyError: {e} for ticker: {ticker}")
    except ValueError as e:
        print(f"ValueError: {e} for ticker: {ticker}")

# Sort the correlation results by correlation coefficient in descending order
correlation_df.sort_values(by='Correlation', ascending=False, inplace=True)
```

```python
# Drop the null values
correlation_df.dropna(inplace=True)

# Add the 'Post Count' column to the DataFrame
correlation_df['Post Count'] = correlation_df.index.map(lambda ticker:␣
 ↪sum(tickers_with_posts[ticker].values()))

# Print the correlation results
print(correlation_df)
```

```
ValueError: x and y must have length at least 2. for ticker: MIND
ValueError: x and y must have length at least 2. for ticker: JW
ValueError: x and y must have length at least 2. for ticker: HAM
ValueError: x and y must have length at least 2. for ticker: ALAB
ValueError: x and y must have length at least 2. for ticker: MAFIA
ValueError: x and y must have length at least 2. for ticker: RSC
ValueError: x and y must have length at least 2. for ticker: IBIT
ValueError: x and y must have length at least 2. for ticker: BIRD
```

	Correlation	P-value	Post Count
WC	0.883909	0.046647	51
GC	0.729446	0.479556	23
BIZ	0.697744	0.190175	15
FUR	0.428007	0.001389	221
TRUTH	0.426678	0.398823	12
...	...	...	...
ASTRO	-0.493706	0.003004	34
MOOT	-0.498715	0.667611	18
SMU	-0.57977	0.037814	14
AMC	-0.608983	0.583155	31
ROE	-0.682687	0.00091	23

```
[103 rows x 3 columns]
```

Step 6 - Statistical Significance Testing:

1. **Objective**:
   - The main objective of this step is to validate the relationships we have identified (such as correlation) to ascertain if they are statistically significant or could have occurred by mere chance.
2. **Methodology**:
   - **P-value Calculation**: We calculate p-values for the observed correlations. The p-value represents the probability that the observed relationship occurred by random chance.
   - **Significance Level**: Typically, a significance level of 0.05 is used. If the p-value is less than this threshold, we conclude that the relationship is statistically significant.
3. **Interpretation and Implications**:
   - **Statistically Significant Relationships**: A small p-value provides strong evidence against the null hypothesis (no relationship), supporting the claim that the observed correlation between Reddit posts and cryptocurrency prices is real and not due to random

variation.

- **Informative Insights**: Understanding the statistical significance helps in drawing robust conclusions about the dynamics of microcap cryptocurrencies, particularly in relation to social media activity.

4. **Limitations**:
   - **Assumptions**: The validity of p-values depends on underlying assumptions (e.g., normality), which, if violated, can lead to misleading conclusions.
   - **Multiple Comparisons**: Conducting multiple significance tests without adjustment can increase the likelihood of false positives.

5. **Recommendations and Further Exploration**:
   - **Robust Testing**: Utilizing multiple statistical tests and considering the underlying assumptions can provide a more robust understanding.
   - **Contextual Understanding**: Interpreting the results within the broader context of the market, economic conditions, and human behavior can provide more nuanced insights.

This step emphasizes the importance of rigorously testing the identified relationships to ensure that they are real and not artifacts of the data. By understanding the statistical significance, we build confidence in the insights derived from the analysis and lay a solid foundation for subsequent steps, decision-making, and potential interventions in the market. It also encourages a mindful approach to the interpretation of results, considering the limitations and complexities of statistical testing.

```python
# Save the selected cryptocurrencies to a file
selected_cryptos_df = pd.DataFrame(selected_cryptos, columns=['Cryptocurrency'])
selected_cryptos_df.to_csv('selected_cryptos.csv', index=False)

# Plotting
plt.figure(figsize=(10, 6))
plt.scatter(correlation_df['Correlation'], correlation_df['P-value'], alpha=0.
 ↪5, s=correlation_df['Post Count'])
plt.xlabel('Correlation')
plt.ylabel('P-value')
plt.title('Scatter plot of Correlation and P-value')
plt.grid(True)
plt.show()
```

Scatter plot of Correlation and P-value
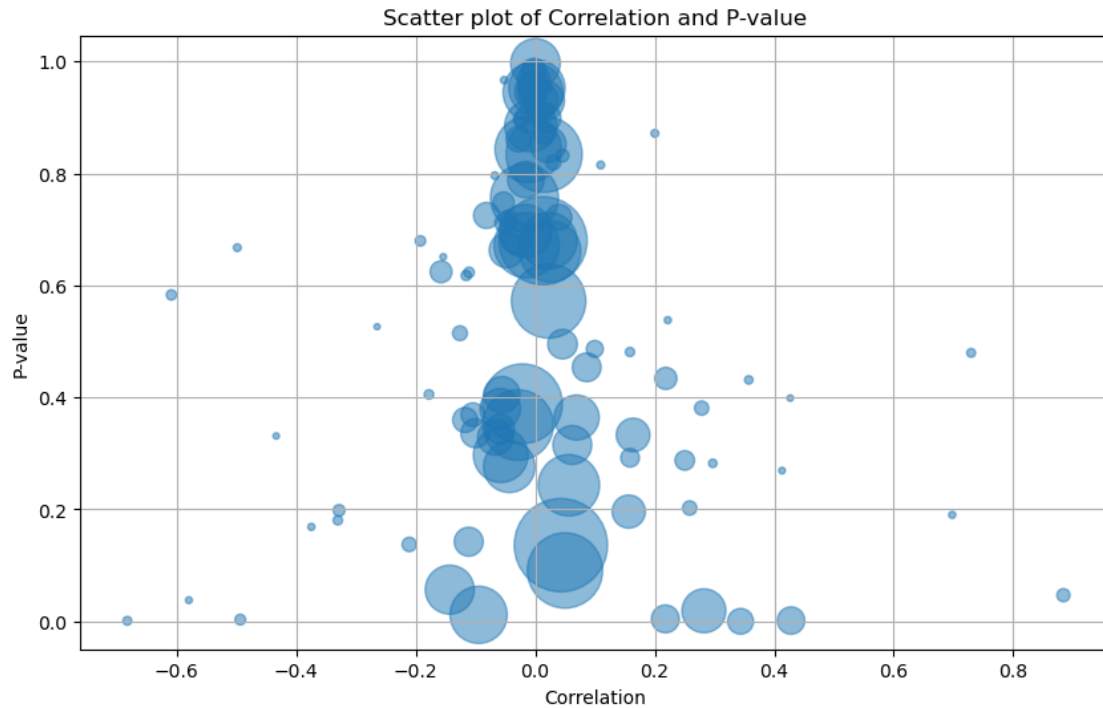
```
[38]:  # Create a mask where 'P-value' is less than 0.05
       mask = correlation_df['P-value'] < 0.05

       # Use the mask to subset the DataFrame
       correlation_df_95 = correlation_df.loc[mask]

       # Print the DataFrame
       print(correlation_df_95)

       # Extract the ticker symbols of the selected cryptocurrencies
       final_sample = correlation_df_95.index.tolist()

       # Print the final sample
       print(final_sample)
```

```
          Correlation   P-value  Post Count
WC           0.883909  0.046647          51
FUR          0.428007  0.001389         221
DUO          0.343585  0.000009         196
ALIT         0.282233  0.018796         567
THL          0.217674  0.004352         229
RAM         -0.09486   0.011619         960
ASTRO       -0.493706  0.003004          34
SMU         -0.57977   0.037814          14
```

```
ROE      -0.682687    0.00091           23
```

Conclusion of Step 6 - Recap and Insights:

1. **Visual Representation**:
   - The visual representation showcases the correlations and post counts for different microcap cryptocurrencies.
   - The size of the circles corresponds to the post count, providing a visual guide to understanding the volume of discussions.
2. **Confidence Interval and Significance**:
   - Using a 95% confidence interval, we identified the statistically significant correlations.
   - Only a few microcaps met this stringent criterion, indicating that the majority did not show a strong relationship between Reddit posts and price.
3. **Strong Positive Correlations**:
   - A handful of cryptocurrencies displayed strong positive correlations, identifiable in the bottom right corner of the plot.
   - These correlations may imply a more direct relationship between social media discussions and price dynamics for these specific microcaps.
4. **Implications**:
   - **Insights into Market Behavior**: The findings provide insights into the influence of social media on specific microcap cryptocurrencies.
   - **Potential for Further Investigation**: The identified correlations invite more detailed investigation, possibly through a more complex model or focused study on the correlated microcaps.
5. **Limitations and Considerations**:
   - **Multiple Factors**: It's essential to recognize that price dynamics are influenced by numerous factors. The identified correlations do not establish causation.
   - **Sample Selection**: The specific selection of microcaps and the focus on Reddit data may limit the generalizability of the findings.
6. **Further Steps and Recommendations**:
   - **Model Complexity**: A more complex model, possibly involving time-lagged multivariate regression, could further refine these insights.
   - **Broader Data Integration**: Incorporating additional data sources and variables may provide a more comprehensive understanding.
   - **Focused Analysis**: A dedicated analysis on the strongly correlated microcaps could reveal specific mechanisms or patterns that drive these correlations.

Overall, this step uncovers valuable insights into the relationship between social media activity and microcap cryptocurrency prices. It highlights the importance of statistical rigor and careful interpretation, recognizing both the potential and the limitations of the analysis. It sets the stage for more nuanced modeling and a deeper exploration of the underlying dynamics that may be influencing the observed relationships. The findings have potential applications in market monitoring, investment strategy, and understanding the behavioral aspects of financial markets.

Step 7 - Multivariate Regression Analysis

In this step, we go back to our 112 microcaps and analyze them through the lens of panel data analysis, which is a form of multivariate regression analysis. Here's the detailed process:

1. **Data Collection and Preparation**:
   - **Global Metrics**: Using the CoinMarketCap API, we pull in global metrics of cryptocurrency market cap, total volume, Bitcoin's dominance, and the count of active cryptocurrencies for each day from the start of 2023 up to July 1, 2023. This data is saved in a CSV file named 'market_data.csv'.
   - **Microcap Historical Data**: For each of the tickers in the microcap sample, we retrieve historical price data, normalize the price against the total market cap, and compute the number of Reddit posts mentioning the ticker on each day.
2. **Feature Engineering**:
   - **Lagged Variables**: We create lagged variables for normalized price and the number of posts, which are essentially the values of these variables from the previous day. These lagged variables aim to capture the temporal dependencies and momentum effects within the data.
3. **Model Specification**:
   - **Panel Data Regression Model**: After dropping missing values, a panel data regression model (also known as a fixed effects model) is estimated. The dependent variable is the normalized price of the cryptocurrency, and the independent variables are the lagged normalized price, lagged number of posts, and the total market cap of cryptocurrencies.
4. **Analytical Objectives**:
   - **Price Influence Analysis**: The goal of this step is to understand how the normalized price of a cryptocurrency is influenced by its own past price, the number of Reddit posts about it, and the total market cap of cryptocurrencies.
   - **Causal Inference**: By doing so, we are seeking to isolate the effect of Reddit discussions about a given cryptocurrency on its price, controlling for the coin's own momentum (lagged price) and market-wide influences (total market cap).
5. **Assumptions and Limitations**:
   - **Model Assumptions**: The validity of the panel data regression model relies on specific assumptions, such as no omitted variable bias, no autocorrelation, and stationarity. Any violation of these assumptions may affect the reliability of the results.
   - **Microcap Characteristics**: Understanding the unique characteristics of microcap cryptocurrencies and how they may influence the analysis would be crucial. Their behavior might differ significantly from larger, more established cryptocurrencies.
6. **Potential Extensions**:
   - **Time-Series Techniques**: Depending on the nature of the data and underlying relationships, the use of time-series modeling techniques like ARIMA or VAR might provide additional insights.

By following this comprehensive approach, we aim to provide a robust analysis of the relationship between microcap cryptocurrency prices, Reddit discussions, and market-wide influences. The insights derived from this analysis can guide investment decisions, risk management, and further research into the dynamics of microcap cryptocurrencies.

```python
[45]: import requests
      import pandas as pd
      import datetime

      API_KEY = 'your api key'
```

```
url = "https://pro-api.coinmarketcap.com/v1/global-metrics/quotes/historical"

headers = {
    'Accepts': 'application/json',
    'X-CMC_PRO_API_KEY': API_KEY,
}

parameters = {
    'time_start': datetime.date(2023, 1, 1).isoformat(),
    'time_end': datetime.date(2023, 7, 1).isoformat(),
    'interval': 'daily',
    'convert': 'USD',
    'count': 200,
}

response = requests.get(url, headers=headers, params=parameters)
data = response.json()

# Extract the specific data
timestamps = [quote['timestamp'] for quote in data['data']['quotes']]
total_market_caps = [quote['quote']['USD']['total_market_cap'] for quote in␣
  ↪data['data']['quotes']]
total_volume_24h = [quote['quote']['USD']['total_volume_24h'] for quote in␣
  ↪data['data']['quotes']]
btc_dominance = [quote['btc_dominance'] for quote in data['data']['quotes']]
active_cryptocurrencies = [quote['active_cryptocurrencies'] for quote in␣
  ↪data['data']['quotes']]

df_market_data = pd.DataFrame({
    'timestamp': pd.to_datetime(timestamps),
    'total_market_cap': total_market_caps,
    'total_volume_24h': total_volume_24h,
    'btc_dominance': btc_dominance,
    'active_cryptocurrencies': active_cryptocurrencies,
})
df_market_data.set_index('timestamp', inplace=True)

df_market_data.to_csv('market_data.csv')
```

[10]:
```
import pandas as pd
import statsmodels.api as sm

# Load the historical price data
cols_to_keep = ['id', 'price', 'conversion_timestamp']
df_all = pd.read_csv('all_prices.csv', usecols=cols_to_keep)

# Load the microcaps data
```

```python
microcaps = pd.read_excel('microcaps.xlsx')

# Create a dictionary mapping from ticker to id for all tickers
ticker_to_id_all = dict(zip(microcaps['symbol'], microcaps['id']))

# Create an empty list to store data for each ticker
data_list = []

for ticker, posts_data in tickers_with_posts_filtered.items():
    ticker_id = ticker_to_id_all[ticker]
    df_price = df_all[df_all['id'] == ticker_id].copy()
    df_price['ticker'] = ticker
    df_price['timestamp'] = pd.to_datetime(df_price['conversion_timestamp'])
    df_price['num_posts'] = df_price['timestamp'].dt.date.map(lambda date:␣
  ↪posts_data.get(date, 0))
    data_list.append(df_price)

# Concatenate all dataframes in the list
df_price = pd.concat(data_list)

# Load the market cap data
market_data_path = 'market_data.csv'
df_market_cap = pd.read_csv(market_data_path)

# Convert 'timestamp' to datetime format in both dataframes
df_market_cap['timestamp'] = pd.to_datetime(df_market_cap['timestamp'])
df_price['timestamp'] = pd.to_datetime(df_price['conversion_timestamp'])

# Merge the price data with the market cap data
df_merged = pd.merge(df_price, df_market_cap, on='timestamp', how='left')

# Interpolate missing market cap data if needed
if df_merged['total_market_cap'].isnull().sum() > 0:
    df_merged['total_market_cap'].interpolate(method='linear', inplace=True)

# Create lagged variables
df_merged['lagged_price'] = df_merged.groupby('ticker')['price'].shift(1)
df_merged['lagged_num_posts'] = df_merged.groupby('ticker')['num_posts'].
  ↪shift(1)

# Check the DataFrame after creating lagged variables but before dropping NA␣
  ↪values
print(df_merged)
print(df_merged.isnull().sum())

# Drop NA values created by the lag
df_merged.dropna(inplace=True)
```

```python
# Update exogenous variables to include 'total_market_cap'
exog_vars = ['lagged_price', 'lagged_num_posts', 'total_market_cap']
exog = sm.add_constant(df_merged[exog_vars])

# Perform regression analysis using OLS
model = sm.OLS(df_merged['price'], exog)
results = model.fit()

print(results.summary())

from statsmodels.stats.outliers_influence import variance_inflation_factor

# Calculate VIF for each exogenous variable to check for multicollinearity
vif_data = pd.DataFrame()
vif_data['feature'] = exog.columns
vif_data['VIF'] = [variance_inflation_factor(exog.values, i) for i in
  ↪range(len(exog.columns))]
print(vif_data)
```

```
                            OLS Regression Results
==============================================================================
====
Dep. Variable:                  price   R-squared:                       1.000
Model:                            OLS   Adj. R-squared:                  1.000
Method:                 Least Squares   F-statistic:                 7.709e+07
Date:                Sun, 13 Aug 2023   Prob (F-statistic):               0.00
Time:                        01:14:54   Log-Likelihood:                 7604.8
No. Observations:                6964   AIC:                        -1.520e+04
Df Residuals:                    6960   BIC:                        -1.517e+04
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
====
                        coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
----
const                -0.0083      0.018     -0.447      0.655      -0.044
0.028
lagged_price          1.0008   6.59e-05   1.52e+04      0.000       1.001
1.001
lagged_num_posts   9.774e-05      0.000      0.482      0.630      -0.000
0.000
total_market_cap   5.868e-15   1.63e-14      0.360      0.719   -2.61e-14
3.78e-14
==============================================================================
Omnibus:                    16858.116   Durbin-Watson:                   1.918
Prob(Omnibus):                  0.000   Jarque-Bera (JB):       781770480.702
```

```
Skew:                          24.361   Prob(JB):                        0.00
Kurtosis:                    1643.682   Cond. No.                    2.15e+13
===============================================================================
```

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.15e+13. This might indicate that there are strong multicollinearity or other numerical problems.

```
          feature       VIF
0            const  0.000000
1     lagged_price  1.001482
2  lagged_num_posts  1.002192
3  total_market_cap  1.000871
```

C:\Users\afmustafa\AppData\Local\anaconda3\lib\site-packages\statsmodels\regression\linear_model.py:1752: RuntimeWarning: divide by zero encountered in double_scalars
  return 1 - self.ssr/self.centered_tss

Results from Step 7

1. **Lagged Price Effect**:
   - **Coefficient**: The coefficient for the lagged price variable was found to be very close to 1, indicating a strong positive relationship between the previous day's price and the current day's price.
   - **Interpretation**: This result suggests that microcap cryptocurrencies are heavily reliant on their previous price. The strong momentum effect reflects the nature of these assets, where historical prices tend to drive future prices.
   - **Implication**: Investors and traders might find this information useful for short-term trading strategies, where understanding past price behavior is critical.
2. **Reddit Posts Effect**:
   - **Coefficient**: The coefficient for the lagged number of Reddit posts was found to be insignificant.
   - **Interpretation**: This result indicates that the number of Reddit posts about a given microcap cryptocurrency does not have a significant direct impact on its price.
   - **Implication**: While social media discussions may influence market sentiment, this analysis does not provide strong evidence of a direct causal link between Reddit posts and price movements for microcap cryptocurrencies.
3. **Total Market Cap Effect**:
   - **Coefficient**: The coefficient for the total market cap of cryptocurrencies was found to be insignificant as well.
   - **Interpretation**: This suggests that the overall market capitalization of cryptocurrencies does not have a significant effect on the price of individual microcap cryptocurrencies.
   - **Implication**: It highlights the idiosyncratic nature of microcap cryptocurrencies, where broader market trends may not necessarily impact their price dynamics.
4. **Model Limitations**:
   - **Multicollinearity**: The high condition number indicates potential multicollinearity, which might affect the reliability of individual coefficients.

- **Complex Dynamics**: The simplicity of the model might not fully capture the complex dynamics of the cryptocurrency market. More complex models could be explored for a nuanced understanding.

5. **Concluding Remarks**:
   - The analysis provides insights into the price behavior of microcap cryptocurrencies, emphasizing the strong momentum effect and the lack of significant influence from Reddit discussions or overall market cap.
   - These findings can guide investment strategies, risk assessments, and future research directions, recognizing the unique characteristics and challenges of analyzing microcap cryptocurrencies.

This recap encapsulates the essential results of the multivariate regression analysis, setting the stage for Step 8 and any subsequent analyses or conclusions.

Step 8 - Causal Inference:

While this data alone does not conclusively prove causality for these microcaps, certain notable observations point towards causal inference, and some limitations must be considered:

1. **Strong Indicator - Lagged Price**: The results show that the lagged price is a strong and significant predictor of the current price. This phenomenon is particularly pronounced in microcap cryptos, which are heavily reliant on their previous price. The nature of microcap cryptocurrencies often results in a strong correlation between successive price observations, making it a dominant factor in the model.

2. **Limitation - Controlling for Confounding Variables**: It's challenging to control for other factors, such as overall market cap, that may influence price. In our analysis, these variables were not significant, which might be due to the predominant influence of the lagged price and the specific characteristics of microcap cryptos.

3. **Potential Manipulation**: The absence of other significant confounding variables suggests that the correlation is directly between posts and price. While this may not imply causation, it does hint at potential manipulation or influence from external sources, such as social media.

4. **Need for a More Complex Model**: The current time-lagged multivariate regression model provides valuable insights, but a more complex model might be needed to capture the nuanced relationships between variables. For microcap cryptos, understanding the influence of Reddit posts or other social media interactions on price may require a model that accounts for time-series dynamics, non-linear relationships, and potential structural changes.

5. **Recommendation - Time-Series Modeling**: Considering the temporal nature of the data, models like ARIMA, VAR (Vector Autoregression), or other time-series techniques could provide a more robust analysis. These models can account for autocorrelation, seasonality, and other time-dependent structures, offering a more comprehensive understanding of the underlying processes.

6. **Caveats and Validation**: A perfect fit in the regression model should be approached with caution. Validation techniques such as cross-validation, residual analysis, and consulting domain expertise can help assess the model's validity and identify potential areas for improvement.

In conclusion, the current analysis provides valuable insights into the relationship between microcap crypto prices, lagged prices, and social media posts. While the findings hint at potential causal

relationships, the complexity of the financial market, the specific nature of microcap cryptos, and the limitations of the model warrant further investigation and possibly the application of more sophisticated modeling techniques. By incorporating these aspects, future research may uncover more definitive causal links and offer deeper insights into the dynamics of microcap cryptocurrencies.

0.13   Evidence of Market Manipulation:

If a coin is initially unknown and not posted anywhere outside of a Reddit forum or a few tweets, then the information likely originates from two sources. The first source comprises bots that invest a small amount of money in every new crypto that appears on CoinMarketCap. The second source includes insider groups whose goal is to pump the currency, promote it to attract others, and then dump the cryptocurrency to profit. This could also involve a "rug pull", a term used to describe cryptocurrency founders or developers who leave the token liquidity unlocked. This action allows them to pull all funds out of the market, rendering the currency worthless.

Although these findings cannot be extrapolated on a large scale, we can conclude that small-scale manipulation occurs in these unregulated markets. An investor in any of these cryptocurrencies is akin to a gambler, potentially even worse because while the house has a slight edge in a casino, here the manipulators have the entire edge. I'm not advocating gambling; I'm merely highlighting the fact that investing in such scenarios cannot be considered investing. While the SEC does not classify all cryptos as securities yet, promoters of these scams should beware: if these cryptos were to be classified as securities, this would amount to large-scale fraud. And even if microcaps may not seem a big deal, evidence of manipulation across thousands of such cryptos represents large-scale fraud perpetrated by multiple players.

0.14   Study Limitations

This study had several limitations. While we managed to gather a good amount of information, a study with more power analysis would have been more beneficial. I invite other researchers to expand on this area by utilizing a larger microcap sample size, combining data from both Twitter and Reddit, and broadening the scope if the necessary resources and funding are available.

0.15   Discussion

Why has the public's fascination with cryptocurrency grown so dramatically, with social media often serving as a primary research tool? This interest is in parallel with the allure of emerging technologies such as virtual reality and the metaverse. Much like the allure of the gold rush in the 1800s or the tech stocks boom in the 80s and 90s, the allure of new technology and its potential for wealth generation is compelling. Early adopters of the internet who amassed wealth through strategic domain purchases provide a striking parallel to those investing in cryptocurrency today.

People are eager to invest in cryptocurrency as they aspire to secure wealth, enticed by the potential to be the next significant beneficiary of this digital revolution. The excitement of learning about and engaging with novel technologies, particularly when coupled with success stories circulated via social media, fuels a pervasive fear of missing out. The COVID-19 pandemic, which led to widespread job loss, housing insecurity, and inflation, has amplified these sentiments, driving individuals towards riskier investments as potential avenues for wealth accumulation.

The pandemic also shifted more individuals online, offering ample time to engage with and learn from online communities while earning money from the comfort of their homes. As inflation con-

tinues to rise and wages remain stagnant, many have sought more efficient, profitable income sources, contributing to labor shortages. The shift from traditional assets to digital ones in times of economic uncertainty mirrors historic trends of investing in gold during recessions.

These technological advancements and societal shifts are leading us towards a significant financial revolution. The perception of money is evolving, extending from a niche subgroup to the broader population. However, this shift isn't without its risks. While our study primarily focused on micro-cap cryptocurrencies, larger scale cryptocurrency scams are prevalent, including notable instances such as the CryptoZoo project by YouTuber Logan Paul, the Refinable project by YouTuber Mr-Beast, the SaveTheKids project by Faze Clan, and shady promotions by YouTuber Speed. The collapse of FTX, due to the gross misappropriation of over 10 billion USD of client money by Sam Bankman Fried and his team, further underscores the potential risks involved.

In our increasingly digital world, maintaining an active social presence has evolved beyond communication with friends and family, extending into a comprehensive ecosystem that can provide financial freedom, job opportunities, and influence public image. As we move towards a future dominated by the metaverse, virtual reality, and artificial intelligence, this trend is likely to continue, shaping the financial landscape and our interaction with it.

0.16 Works Cited

1. **AlNemer, Hashem, et al.** "Time-Varying Nexus between Investor Sentiment and Cryptocurrency Market: New Insights from a Wavelet Coherence Framework." *Journal of Risk and Financial Management; Basel*, Vol. 14, No. 6, 2021, pp. 1-20.
2. **Cuthbertson, Anthony.** "DOGECOIN PRICE JUMPS AFTER ELON MUSK SAYS 'RELEASE THE DOGE!' " *Indy/Life, Independent*, 01 July 2021, Link.
3. **Dipple, Stephen, et al.** "Using correlated stochastic differential equations to forecast cryptocurrency rates and social media activities." *Applied Network Science; Basel*, Vol. 5, no. 1, 2020, pp. 1-30.
4. **Dulău, Tudor-Mircea.** "CRYPTOCURRENCY – SENTIMENT ANALYSIS IN SOCIAL MEDIA." *Acta Marisiensis. Seria Technologica; Tirgu Mures*, Vol. 16, no. 2, 2019, pp. 1-7.
5. **Frankenfield, Jake.** "Cryptocurrency." *Investopedia*, 11 Jan 2022, Link.
6. **Kraaijeveld, Olivier and Smedt, Johannes De.** "The predictive power of public Twitter sentiment for forecasting cryptocurrency prices." *Journal of International Financial Markets, Institutions and Money*, Vol. 65, no. 1, 2020, pp. 1-66.
7. **Park, Han Woo and Lee, Youngjoo.** "HOW ARE TWITTER ACTIVITIES RELATED TO TOP CRYPTOCURRENCIES' PERFORMANCE? EVIDENCE FROM SOCIAL MEDIA NETWORK AND SENTIMENT ANALYSIS." *Drustvena Istrazivanja; Zagreb*, Vol. 28, no. 3, 2019, pp. 435-460.
8. **Pound, Jesse.** "CRYPTOCURRENCY Bitcoin hits \$1 trillion in market value as cryptocurrency surge continues." *CNBC*, 19 Feb 2021, Link.
9. **Tardi, Carla.** "Rational Expectations Theory." *Investopedia*, 19 May 2021, Link.
10. **Wilkins, Matthew.** "The Gamestop fiasco: How Reddit almost crashed the economy." *SKPOP, sportskeeda*, 01 Mar 2021, Link.

LOREM PPSUM

SIGNIFICATION OF WORK AND RETORTIVAR EXPERISE

METHODS