1. Summarizing project idea and outline of strategy in a Juptyer notebook

Here, in this project I used two websites to scrape data,

1. https://coinmarketcap.com/
2. https://www.coingecko.com/

Objective:

- I scraped the data using BeautifulSoup package and stored this data to the backend using MYSQL. Later, performed some visualizations on the scraped data to see the daily trends in the Marketcap, volume, opening and closing prices of Bitcoin and Ethereum.

Scraping on https://coinmarketcap.com

Import necerssary packages

In [222…

```
import requests
import pandas as pd
import numpy as np
import json
import time
import seaborn as sns
import plotly.offline as py
import plotly.graph_objs as go
py.init_notebook_mode()
from bs4 import BeautifulSoup
import pandas_datareader.data as pdr
import datetime as dt
import matplotlib.pyplot as plt
%matplotlib inline
```

2. Picked BeautifulSoup and parsed and extracted the data to MySql

Scraping the data from the website using the beautifulSoup

In [223…

```
cmc= requests.get("https://coinmarketcap.com/trending-cryptocurrencies/")
soup=BeautifulSoup(cmc.content,'html.parser')
#print(soup)
```

In [224…

```
data = soup.find("script", id="__NEXT_DATA__", type="application/json")
coins = {}
coin_data = json.loads(data.contents[0])

listings = coin_data['props']['initialState']['cryptocurrency']['trendingCoins']['data']

for i in listings:
  coins[str(i['id'])] = i['slug']
```

In [225…

```
#print(listings)
```

In [226…

```
coins
```

Out[226…

```
{'5426': 'solana',
 '3945': 'harmony',
 '18363': 'gocryptome',
 '1027': 'ethereum',
 '1839': 'bnb',
 '4030': 'algorand',
 '6193': 'cream-finance',
 '4172': 'terra-luna',
 '3890': 'polygon',
 '3513': 'fantom',
 '2010': 'cardano',
 '7080': 'gala',
 '3794': 'cosmos',
 '5994': 'shiba-inu',
 '4118': 'the-force-protocol',
 '1': 'bitcoin',
 '4807': 'certik',
 '1958': 'tron',
 '7288': 'venus',
 '6783': 'axie-infinity',
 '2682': 'holo',
 '512': 'stellar',
 '4206': 'wink',
 '5647': 'kadena',
 '3635': 'cronos',
 '1104': 'augur',
 '2585': 'centrality',
 '5552': 'hathor',
 '2398': 'selfkey',
 '6210': 'the-sandbox'}
```

Inspect the website's HTML source and identified the right URLs to download.

- Function that extracts required data from the json when the parameters like start and end date are passed. Here the date format is "YYYYMMDD"

In [227…

```
marketCap = list()
volume = list()
symbol = list()
timestamp = list()
name= list()
slug = list()
#name = list()
Open= list()
Close=list()
High= list()
Low= list()
roi=list()
roi_dict=dict()

def histDataFetch(startDate, endDate):
    #coins = {'1': 'bitcoin'}
    for i in coins:
        page = requests.get('https://coinmarketcap.com/currencies/{0}/historical-data/?sta

        soup=BeautifulSoup(page.text,'html.parser')

        data= soup.find('script', id ='__NEXT_DATA__', type='application/json')

        #print(data.contents)
        historical_data = json.loads(data.contents[0])
        data = historical_data['props']['initialProps']['pageProps']['historicalData']['da
        quotes = historical_data['props']['initialProps']['pageProps']['historicalData']['
        roi_data =historical_data['props']['initialProps']['pageProps']['info']['statistic
        roi_dict[coins[i]] = roi_data
        #print(quotes)
        for item in quotes:
            marketCap.append(item['quote']['marketCap'])
            volume.append(item['quote']['volume'])
            timestamp.append(item['quote']['timestamp'])
            symbol.append(data['symbol'])
            slug.append(coins[i])
            name.append(data['name'])
            Open.append(item['quote']['open'])
            Close.append(item['quote']['close'])
            High.append(item['quote']['high'])
            Low.append(item['quote']['low'])
        #return quotes
```

Function calling

In [228…

```
histDataFetch(20210201, 202202016)
```

3.Parse and Explore your data and extract information.

In [229…

```
cryptodata = pd.DataFrame(columns = ["name", "symbol", "slug", "timestamp", "marketCap", '
roi_data= pd.DataFrame(roi_dict.items(),columns=['slug','roi'])
#roi_data.columns=['slug','roi']
cryptodata["name"] = name
cryptodata["symbol"]= symbol
cryptodata["slug"] = slug
cryptodata["timestamp"] = timestamp
cryptodata["marketCap"] = marketCap
cryptodata["volume"] = volume
cryptodata["Open"] = Open
cryptodata["Close"] =Close
cryptodata["High"] = High
cryptodata["Low"] =Low

crypto= cryptodata.copy()
```

In [230…

```
cryptodata.head()
crypto
```

Out[230…

	name	symbol	slug	timestamp	marketCap	volume	Open	Close	
0	Solana	SOL	solana	2022-02-21T23:59:59.999Z	2.658221e+10	2.892901e+09	90.883023	83.123162	95.
1	Solana	SOL	solana	2022-02-22T23:59:59.999Z	2.765098e+10	2.046554e+09	83.106396	86.465385	87.
2	Solana	SOL	solana	2022-02-23T23:59:59.999Z	2.715107e+10	2.293962e+09	86.470797	84.902785	92.
3	Solana	SOL	solana	2022-02-24T23:59:59.999Z	2.853607e+10	5.253579e+09	84.863582	89.194748	92.
4	Solana	SOL	solana	2022-02-25T23:59:59.999Z	2.962508e+10	3.069243e+09	89.228131	92.598823	94
...	...	...	...	...	...	...	...	...	
202	The Sandbox	SAND	the-sandbox	2022-02-23T23:59:59.999Z	3.264927e+09	1.079628e+09	3.158376	2.998249	3.

	name	symbol	slug	timestamp	marketCap	volume	Open	Close	
203	The Sandbox	SAND	the-sandbox	2022-02-24T23:59:59.999Z	3.296947e+09	1.640270e+09	2.996292	3.027653	3.(
204	The Sandbox	SAND	the-sandbox	2022-02-25T23:59:59.999Z	3.444999e+09	1.159279e+09	3.028084	3.163612	3
205	The Sandbox	SAND	the-sandbox	2022-02-26T23:59:59.999Z	3.368408e+09	7.737934e+08	3.163691	3.093277	3.
206	The Sandbox	SAND	the-sandbox	2022-02-27T23:59:59.999Z	3.159387e+09	9.447222e+08	3.092453	2.901329	3.

207 rows × 10 columns

In [231…

```
cryptodata['timestamp']=pd.to_datetime(cryptodata['timestamp']).dt.strftime("%Y-%m-%d %H:%
```

In [232…

```
crypto.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 207 entries, 0 to 206
Data columns (total 10 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   name       207 non-null    object 
 1   symbol     207 non-null    object 
 2   slug       207 non-null    object 
 3   timestamp  207 non-null    object 
 4   marketCap  207 non-null    float64
 5   volume     207 non-null    float64
 6   Open       207 non-null    float64
 7   Close      207 non-null    float64
 8   High       207 non-null    float64
 9   Low        207 non-null    float64
dtypes: float64(6), object(4)
memory usage: 16.3+ KB
```

In [233…

```
crypto.head()
```

Out[233…

	name	symbol	slug	timestamp	marketCap	volume	Open	Close	High
0	Solana	SOL	solana	2022-02-21T23:59:59.999Z	2.658221e+10	2.892901e+09	90.883023	83.123162	95.904091
1	Solana	SOL	solana	2022-02-22T23:59:59.999Z	2.765098e+10	2.046554e+09	83.106396	86.465385	87.163976
2	Solana	SOL	solana	2022-02-23T23:59:59.999Z	2.715107e+10	2.293962e+09	86.470797	84.902785	92.738442
3	Solana	SOL	solana	2022-02-24T23:59:59.999Z	2.853607e+10	5.253579e+09	84.863582	89.194748	92.279393
4	Solana	SOL	solana	2022-02-25T23:59:59.999Z	2.962508e+10	3.069243e+09	89.228131	92.598823	94.77810

Save the data in a csv file

In [234…

```
crypto.to_csv("/Users/anitateladevalapalli/Documents/cryptoScrapedData.csv", index=False)
```

Extracted the data to the backend and saved in the database. Used MYSQL as a platform

In [235…

```
pip install mysql-connector-python
```

```
Requirement already satisfied: mysql-connector-python in ./opt/anaconda3/lib/python3.9/sit
e-packages (8.0.28)
Requirement already satisfied: protobuf>=3.0.0 in ./opt/anaconda3/lib/python3.9/site-packa
ges (from mysql-connector-python) (3.19.4)
Note: you may need to restart the kernel to use updated packages.
```

In [236…

```
import mysql.connector as msql
from mysql.connector import Error
try:
    conn = msql.connect(host='localhost', user='root',
                        password='Anita')#give ur username, password
    if conn.is_connected():
        cursor = conn.cursor()
        cursor.execute("CREATE DATABASE crypto")
        print("Database is created")
except Error as e:
    print("Error while connecting to MySQL", e)
```

```
Error while connecting to MySQL 1007 (HY000): Can't create database 'crypto'; database exi
sts
```

In [237…

```
try:
    conn = msql.connect(host='localhost', database='crypto', user='root', password='Anita'
    if conn.is_connected():
        print("Connection is ..")
        cursor = conn.cursor()

        cursor.execute("use crypto;")
        record = cursor.fetchone()

        print("You're connected to database: ", record)
        cursor.execute('DROP TABLE IF EXISTS crypto_data;')
        print('Creating table....')
# in the below line please pass the create table statement which you want #to create
        cursor.execute("CREATE TABLE crypto_data(name varchar(255),symbol varchar(255),slu
        print("Table is created....")
        #loop through the data frame
        for i,row in cryptodata.iterrows():
            #here %S means string values
            sql = "INSERT INTO crypto.crypto_data VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s);'
            cursor.execute(sql, tuple(row))
            print("Record inserted")
            # the connection is not auto committed by default, so we must commit to save c
            conn.commit()
except Error as e:
            print("Error while connecting to MySQL", e)
```

```
Connection is ..
You're connected to database:  None
Creating table....
Table is created....
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
Record inserted
```

4.Organize, Clean and validate your extracted data in CSV format.

In [238…

```
df = pd.read_csv("/Users/anitateladevalapalli/Documents/cryptoScrapedData.csv",parse_dates
```

In [239…

```
df.head()
```

Out[239…

	name	symbol	slug	marketCap	volume	Open	Close	I
timestamp								
2022-02-21 23:59:59.999000+00:00	Solana	SOL	solana	2.658221e+10	2.892901e+09	90.883023	83.123162	95.904
2022-02-22 23:59:59.999000+00:00	Solana	SOL	solana	2.765098e+10	2.046554e+09	83.106396	86.465385	87.163
2022-02-23 23:59:59.999000+00:00	Solana	SOL	solana	2.715107e+10	2.293962e+09	86.470797	84.902785	92.738
2022-02-24 23:59:59.999000+00:00	Solana	SOL	solana	2.853607e+10	5.253579e+09	84.863582	89.194748	92.279
2022-02-25 23:59:59.999000+00:00	Solana	SOL	solana	2.962508e+10	3.069243e+09	89.228131	92.598823	94.777

In [240…

```
df.isnull().sum()
```

Out[240…

```
name         0
symbol       0
slug         0
marketCap    0
volume       0
Open         0
Close        0
High         0
Low          0
dtype: int64
```

In [241…

```
df['marketCap'].corr(df['volume'])
```

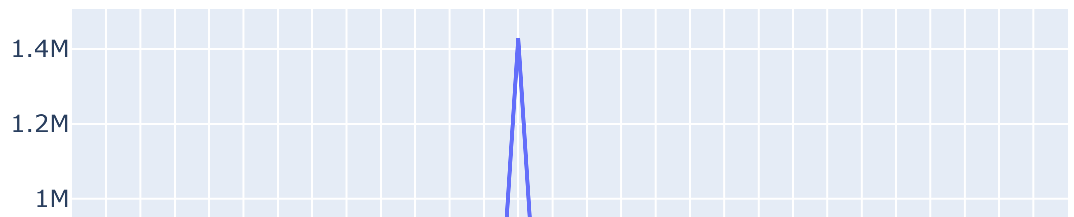
Out[241…

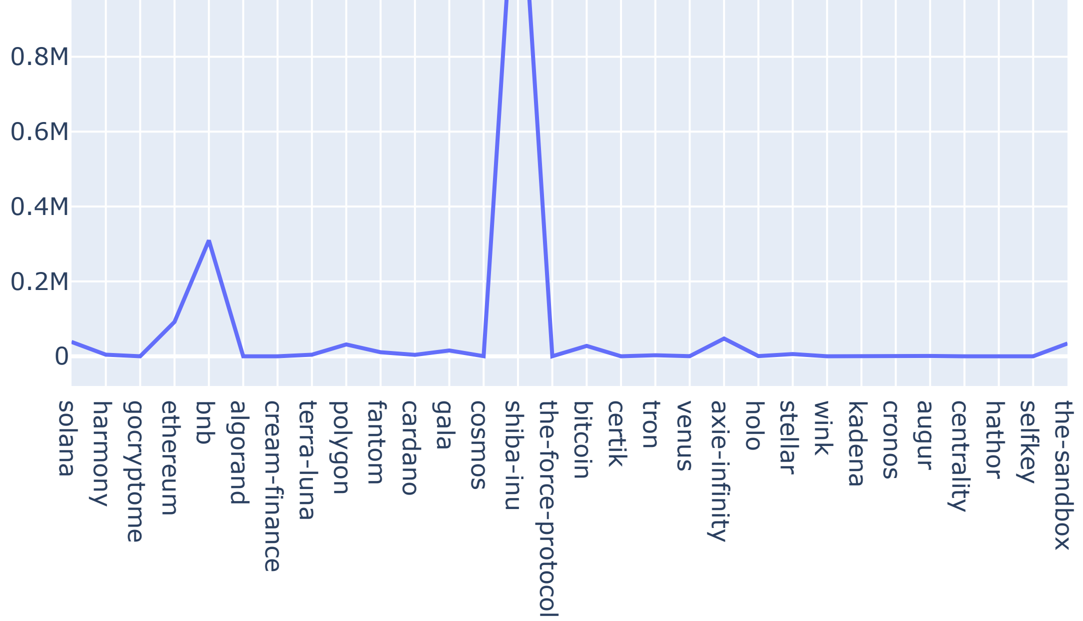
```
0.9285079038919382
```

visualizations

In [242…

```
plot3 = go.Scatter(x=roi_data['slug'], y=roi_data['roi'])
py.iplot([plot3])
```





Analysis 1:

- We can see that the Return on Investment is high for the coin, "shiba-inu" followed bt "bnb"

In [243…

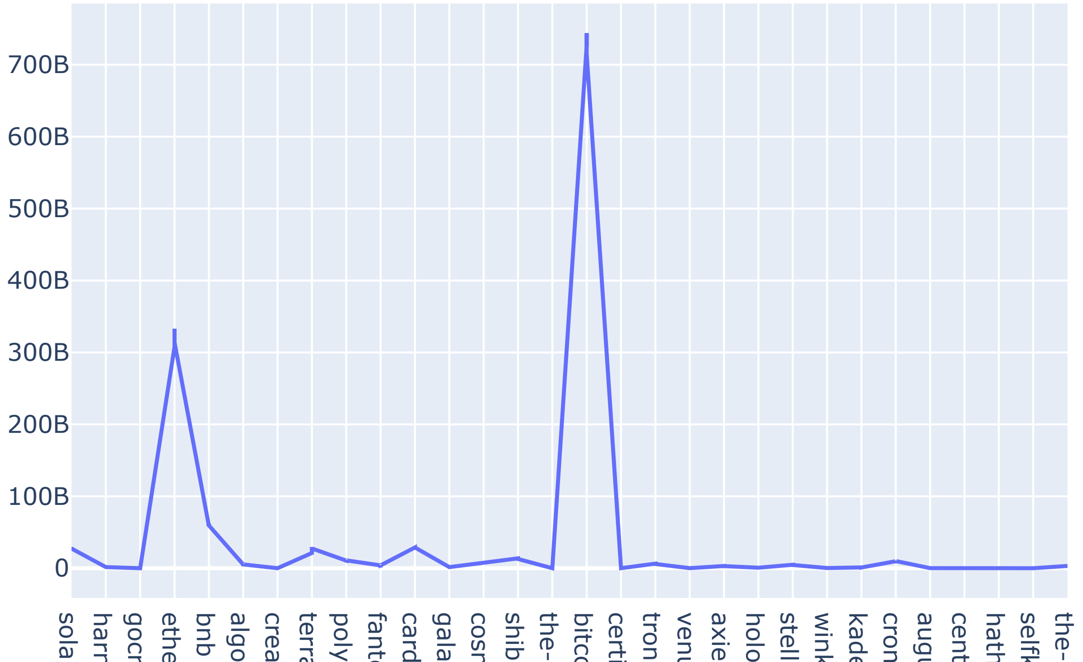
```
df['slug'].unique()
```

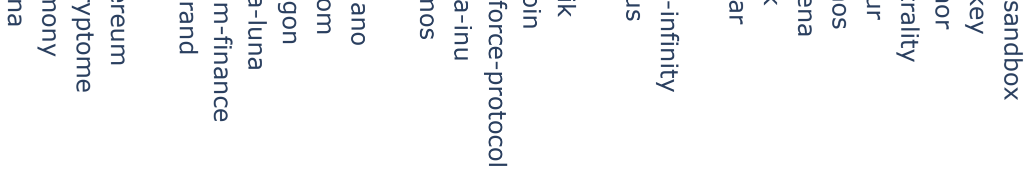
Out[243…

```
array(['solana', 'harmony', 'gocryptome', 'ethereum', 'bnb', 'algorand',
       'cream-finance', 'terra-luna', 'polygon', 'fantom', 'cardano',
       'gala', 'cosmos', 'shiba-inu', 'the-force-protocol', 'bitcoin',
       'certik', 'tron', 'venus', 'axie-infinity', 'holo', 'stellar',
       'wink', 'kadena', 'cronos', 'augur', 'centrality', 'hathor',
       'selfkey', 'the-sandbox'], dtype=object)
```

In [244…

```
plot_mc = go.Scatter(x=df['slug'], y=df['marketCap'])
py.iplot([plot_mc])
```



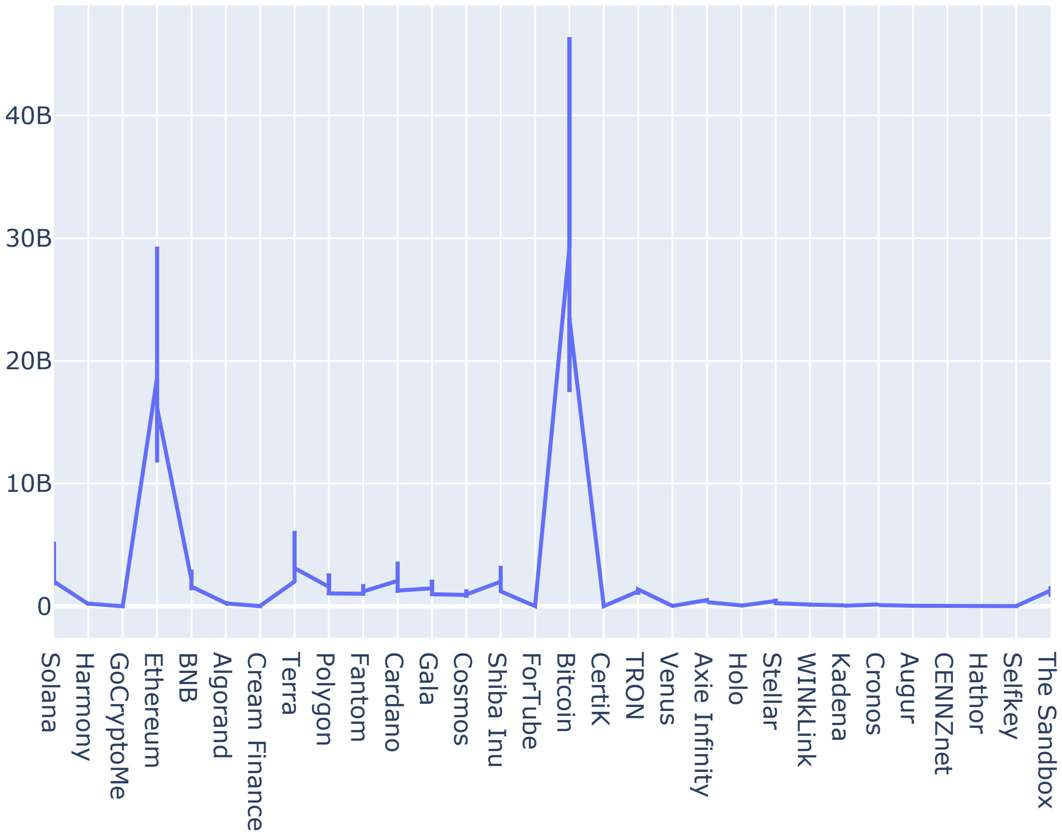


Analysis-2:

- The MarketCap of Bitcoin remained high and then followed by Ethereum

In [245…

```
plot2 = go.Scatter(x=df['name'], y=df['volume'])
py.iplot([plot2])
```



Analysis -3:

- From section 4, we calculated the correlation between Marketcap and Volume which was found to be 91%. Hence, increase in volume will definetely increase the marketcap because Market capitalization is the product of share price and the number of outstanding shares and volume is the number of shares traded.

In [246…

```
df= df.drop(['name','symbol'],axis=1)
```

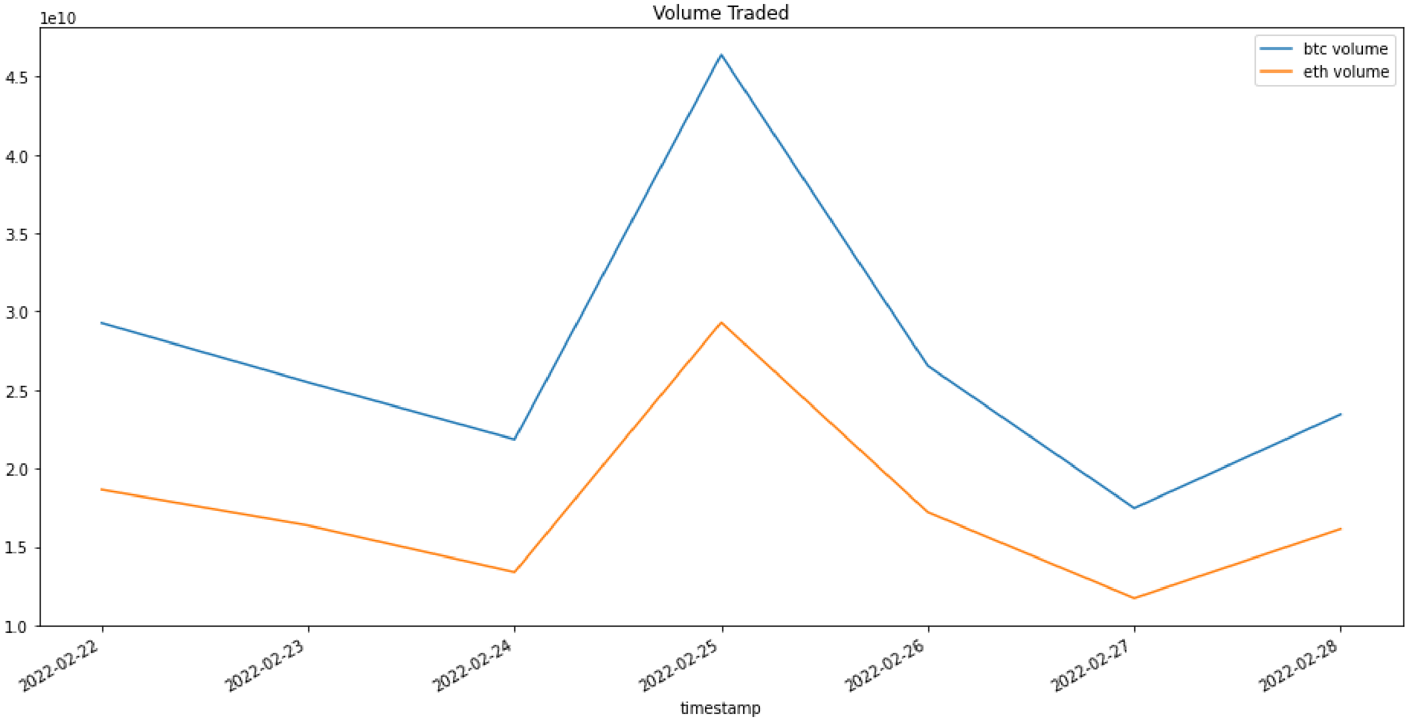
In [247…

```
btc=df[df["slug"] == 'bitcoin']
eth=df[df["slug"]=='ethereum']
btc['volume'].plot(label='btc volume',figsize=(16,8),title='Volume Traded')
```

```
eth['volume'].plot(label='eth volume',figsize=(16,8),title='Volume Traded')
plt.legend()
```

Out[247…

```
<matplotlib.legend.Legend at 0x7fb6c04b3850>
```

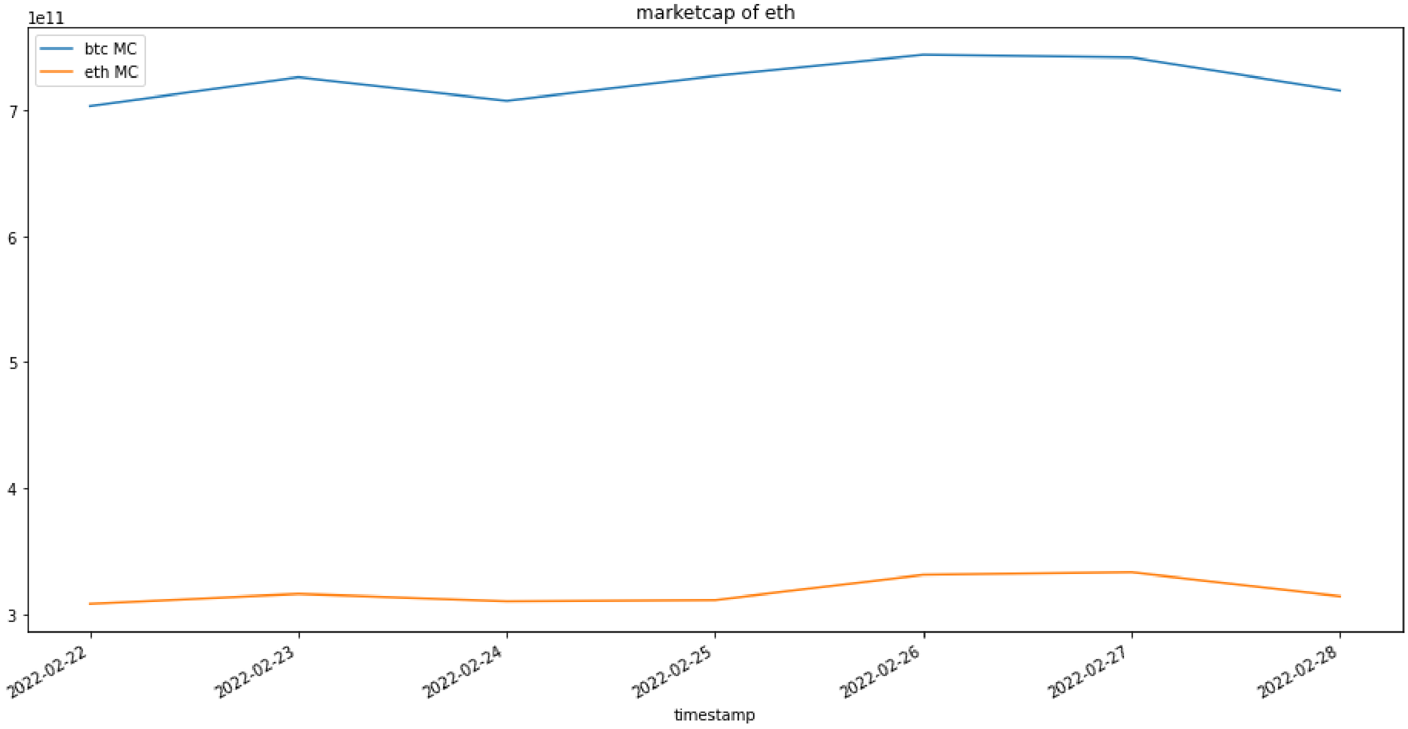


In [248…

```
btc['marketCap'].plot(label='btc MC',figsize=(16,8),title='marketcap of btc')
eth['marketCap'].plot(label='eth MC',figsize=(16,8),title='marketcap of eth')
plt.legend()
```

Out[248…

```
<matplotlib.legend.Legend at 0x7fb698ac4e20>
```

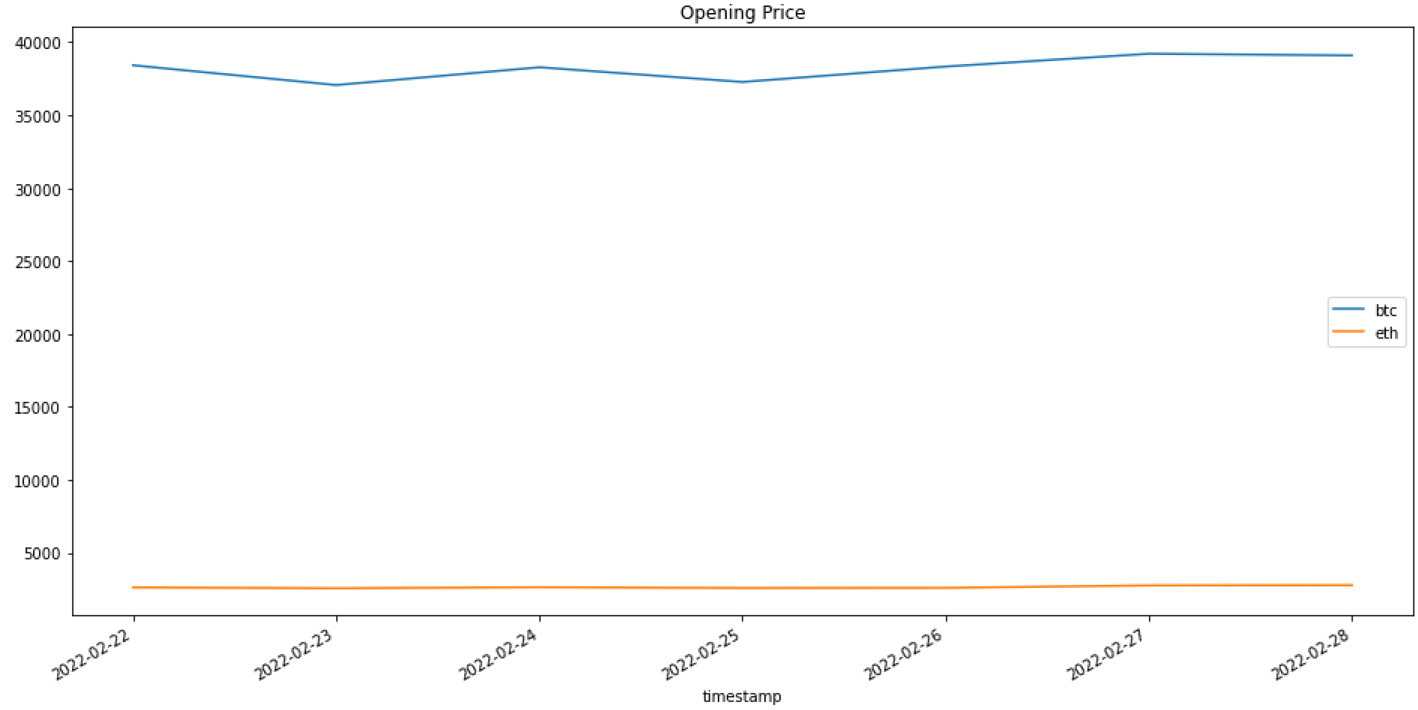


In [249…

```
btc['Open'].plot(label='btc',figsize=(16,8),title='Opening price')
eth['Open'].plot(label='eth',figsize=(16,8),title='Opening Price')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7fb6c09ee820>
```

Out[249…



https://www.marca.com/en/lifestyle/us-news/2022/01/30/61f5ab0122601d9e6c8b45bc.html

The above link redirects to the page where we can see why the opening price of bitcoin has been dropped

In [250…

```
btc['volume'].max()
```

Out[250…

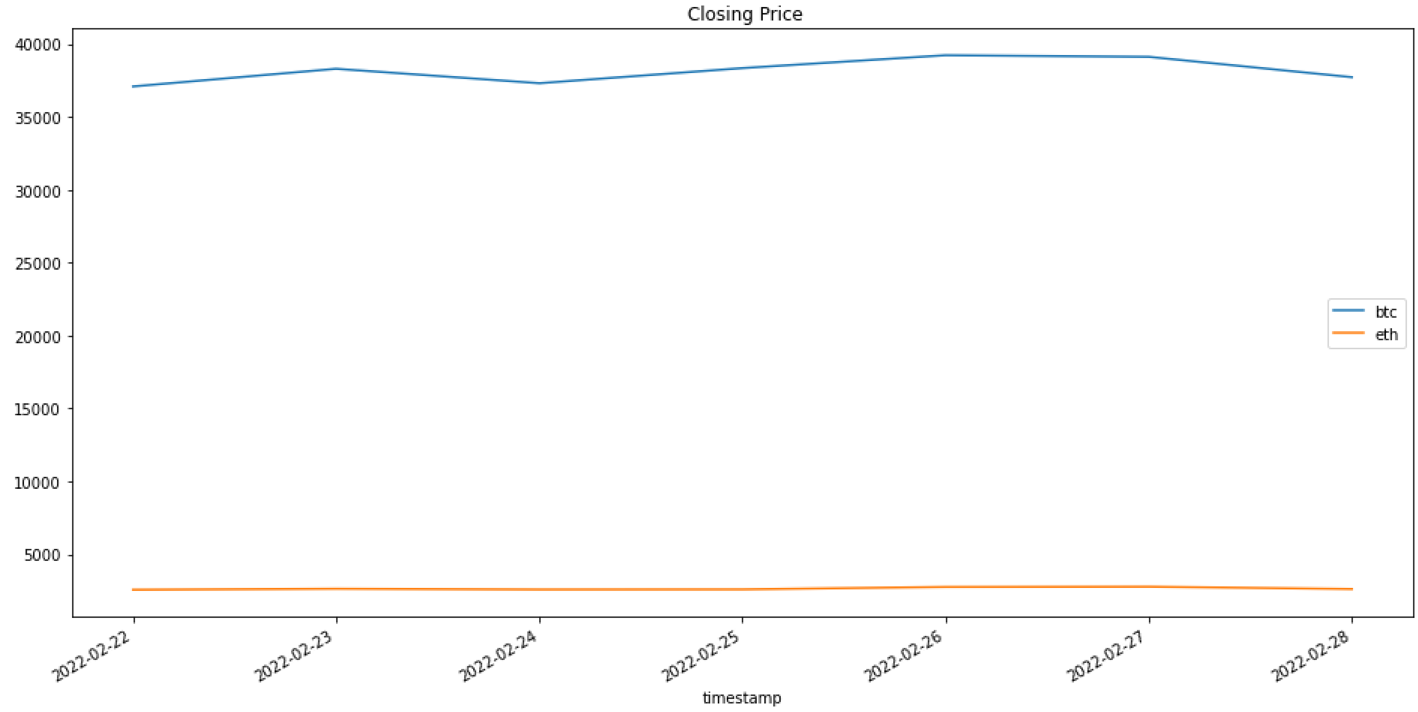
```
46383802092.57
```

In [251…

```
btc['Close'].plot(label='btc',figsize=(16,8),title='Closing price')
eth['Close'].plot(label='eth',figsize=(16,8),title=' Closing Price')
plt.legend()
```

Out[251…

```
<matplotlib.legend.Legend at 0x7fb6c1839880>
```

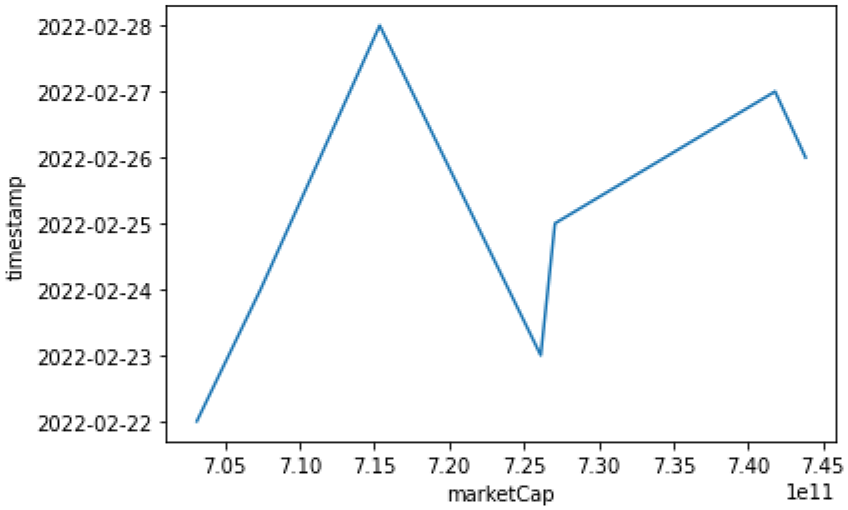


In [252…

```
sns.lineplot(data=btc, x="marketCap", y="timestamp")
```

Out[252…

```
<AxesSubplot:xlabel='marketCap', ylabel='timestamp'>
```



total amount of money being traded (e.g. 100 units of stock at 10$eachversus100000unitsofstockat$ 1 each)

In [253…

```
btc['Total Traded'] = btc['Open']*btc['volume']
eth['Total Traded']=eth['Open']*eth['volume']
```

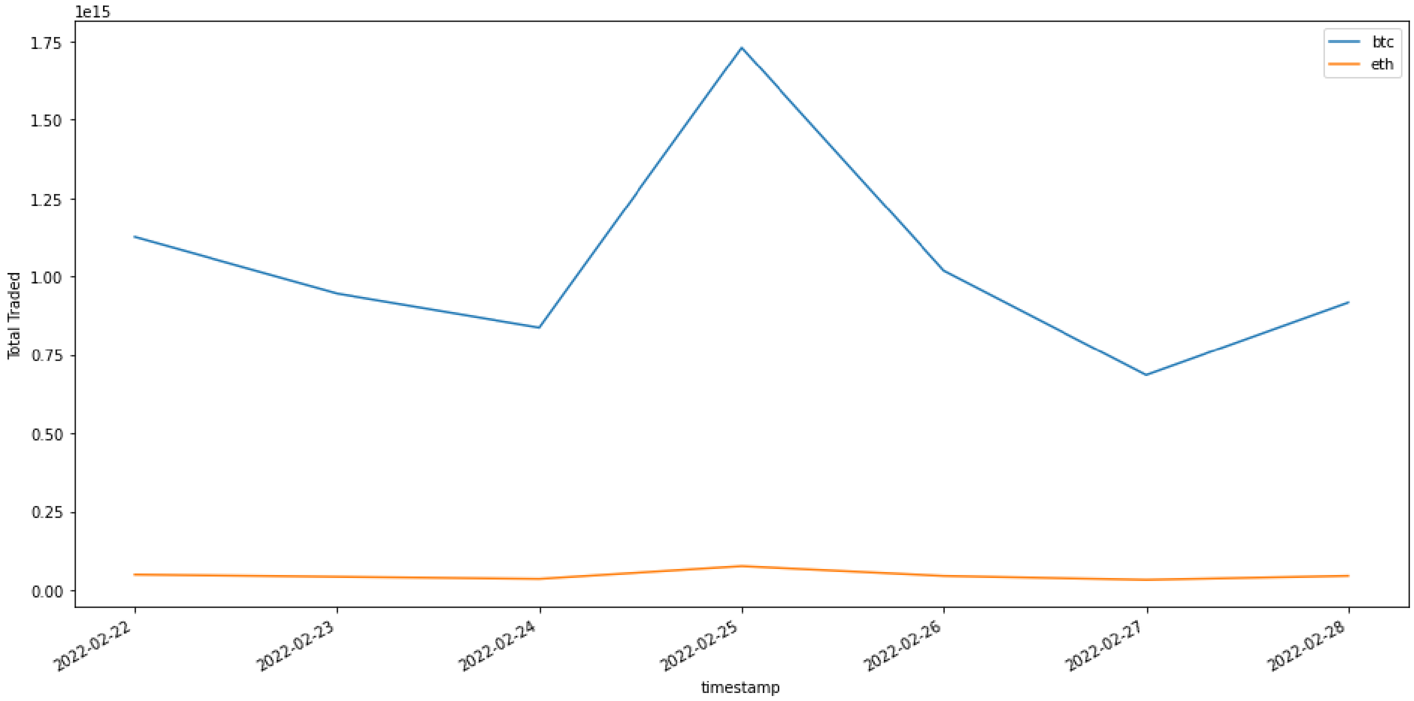
```
/var/folders/6f/c2b7vdpx247cstzj573kd1k40000gn/T/ipykernel_60630/1225935031.py:1: SettingW
ithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_gu
ide/indexing.html#returning-a-view-versus-a-copy

/var/folders/6f/c2b7vdpx247cstzj573kd1k40000gn/T/ipykernel_60630/1225935031.py:2: SettingW
ithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_gu
ide/indexing.html#returning-a-view-versus-a-copy
```

In [254…

```
btc['Total Traded'].plot(label='btc',figsize=(16,8))
eth['Total Traded'].plot(label='eth')
plt.legend()
plt.ylabel('Total Traded')
```

Out[254…

```
Text(0, 0.5, 'Total Traded')
```



Analysis-4

- The reason for sudden increase in total volume traded is explained in the article below https://time.com/nextadvisor/investing/cryptocurrency/latest-crypto-news/

Exponential Weighted Moving average for 2 days vs Exponential Weighted Moving average of Opening price for 4 days of bitcoin

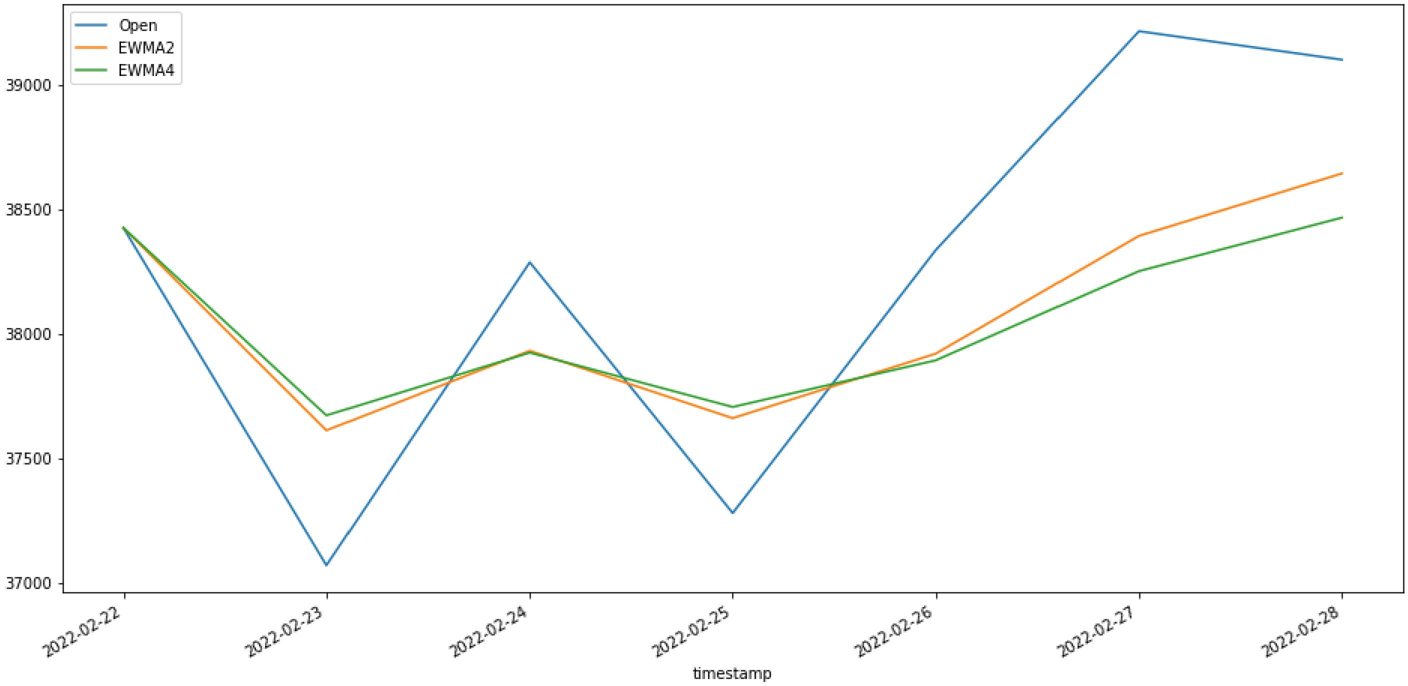
In [255…

```
btc['EWMA2'] = btc['Open'].ewm(2).mean()
btc['EWMA4'] = btc['Open'].ewm(4).mean()
btc[['Open','EWMA2','EWMA4']].plot(label='btc',figsize=(16,8))
plt.legend()
```

```
/var/folders/6f/c2b7vdpx247cstzj573kd1k40000gn/T/ipykernel_60630/3695009140.py:1: SettingW
ithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_gu
ide/indexing.html#returning-a-view-versus-a-copy

/var/folders/6f/c2b7vdpx247cstzj573kd1k40000gn/T/ipykernel_60630/3695009140.py:2: SettingW
ithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_gu
ide/indexing.html#returning-a-view-versus-a-copy
```

Out[255…

```
<matplotlib.legend.Legend at 0x7fb6c09eefd0>
```



In [256…

```
from pandas.plotting import scatter_matrix
```

In [257…

```
crypt_comp = pd.concat([btc['Open'],eth['Open']],axis=1)
```

In [258…

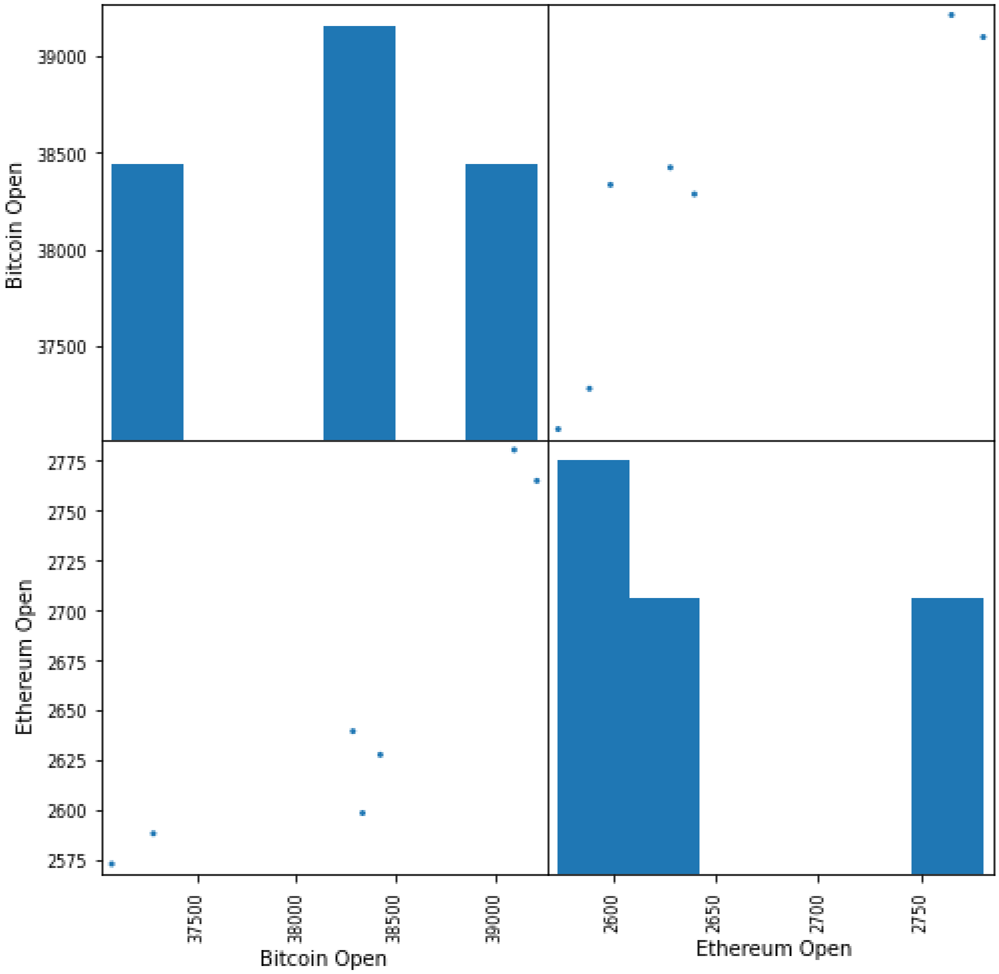
```
crypt_comp.columns = ['Bitcoin Open',' Ethereum Open']
```

In [259…

```
scatter_matrix(crypt_comp,figsize=(8,8),alpha=1,hist_kwds={'bins':6})
```

Out[259…

```
array([[<AxesSubplot:xlabel='Bitcoin Open', ylabel='Bitcoin Open'>,
        <AxesSubplot:xlabel=' Ethereum Open', ylabel='Bitcoin Open'>],
       [<AxesSubplot:xlabel='Bitcoin Open', ylabel=' Ethereum Open'>,
        <AxesSubplot:xlabel=' Ethereum Open', ylabel=' Ethereum Open'>]],
      dtype=object)
```



We see a positve correlation

Daily Percentage Change

First we will begin by calculating the daily percentage change. Daily percentage change is defined by the following formula: $r_t = \frac{p_t}{p_{t-1}} - 1$

This defines r_t (return at time t) as equal to the price at time t divided by the price at time t-1 (the previous day) minus 1. Basically this just informs you of your percent gain (or loss) if you bought the crypto on day and then sold it the next day. While this isn't necessarily helpful for attempting to predict future values of the crypto, its very helpful in analyzing the volatility of the coin. If daily returns have a wide distribution, the coin is more volatile from one day to the next. Let's calculate the percent returns and then plot them with a histogram, and decide which coin is the most stable!

In [260…

```
btc['returns'] = btc['Close'].pct_change(1)
eth['returns'] = eth['Close'].pct_change(1)
```

```
/var/folders/6f/c2b7vdpx247cstzj573kd1k40000gn/T/ipykernel_60630/1884753176.py:1: SettingW
ithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_gu
ide/indexing.html#returning-a-view-versus-a-copy

/var/folders/6f/c2b7vdpx247cstzj573kd1k40000gn/T/ipykernel_60630/1884753176.py:2: SettingW
ithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_gu
ide/indexing.html#returning-a-view-versus-a-copy
```

In [261…

```
btc['returns'].fillna(btc['returns'].mean())
eth['returns'].fillna(eth['returns'].mean())
```

Out[261…

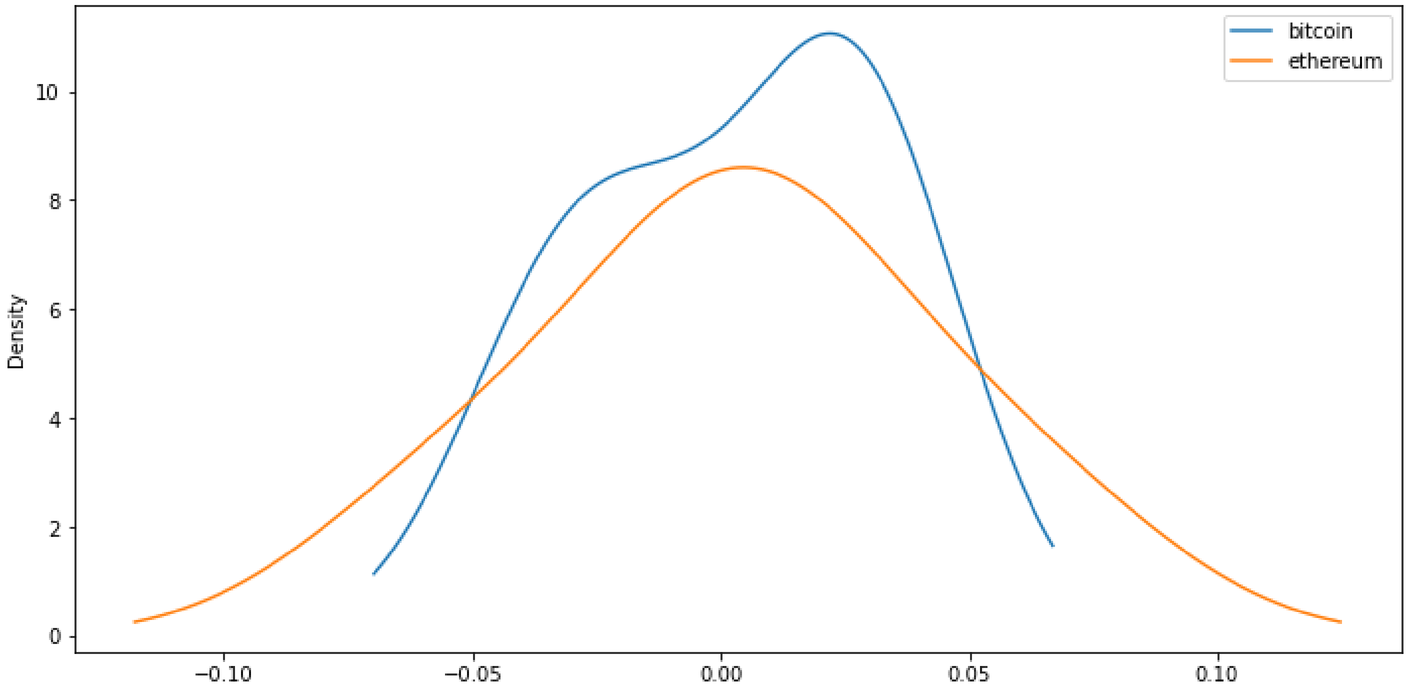
```
timestamp
2022-02-21 23:59:59.999000+00:00    0.003777
2022-02-22 23:59:59.999000+00:00    0.025442
2022-02-23 23:59:59.999000+00:00   -0.018543
2022-02-24 23:59:59.999000+00:00    0.002975
2022-02-25 23:59:59.999000+00:00    0.064074
2022-02-26 23:59:59.999000+00:00    0.005996
2022-02-27 23:59:59.999000+00:00   -0.057283
Name: returns, dtype: float64
```

In [262…

```
btc['returns'].plot(kind='kde',label='bitcoin',figsize=(12,6))
eth['returns'].plot(kind='kde',label='ethereum')
plt.legend()
```

Out[262…

```
<matplotlib.legend.Legend at 0x7fb6c05ee280>
```



Analysis -5

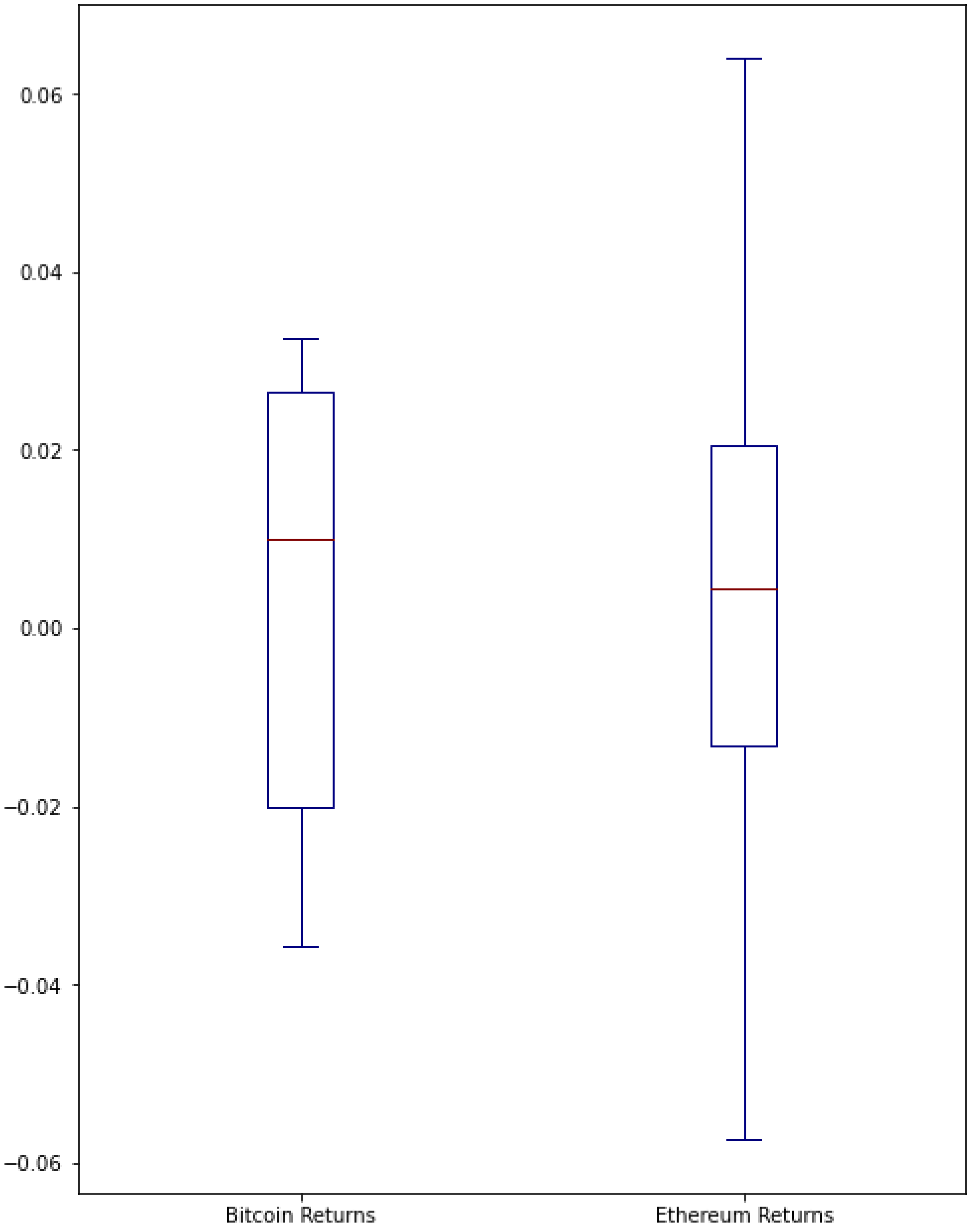
- We can clearly see that ethereum is much more stable than bitcoin from above plot

In [263…

```
box_df = pd.concat([btc['returns'],eth['returns']],axis=1)
box_df.columns = ['Bitcoin Returns','Ethereum Returns']
box_df.plot(kind='box',figsize=(8,11),colormap='jet')
```

Out[263…

```
<AxesSubplot:>
```



While daily returns are useful, it doesn't give the investor a immediate insight into the gains he had made till date, especially if the coin is very volatile. Cumulative return is computed relative to the day investment is made. If cumulative return is above one, you are making profits else you are in loss.

In [264…

```
btc['Cumulative Return'] = (1 + btc['returns']).cumprod()
eth['Cumulative Return'] = (1 + eth['returns']).cumprod()
```

```
/var/folders/6f/c2b7vdpx247cstzj573kd1k40000gn/T/ipykernel_60630/3264051095.py:1: SettingW
ithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_gu
ide/indexing.html#returning-a-view-versus-a-copy

/var/folders/6f/c2b7vdpx247cstzj573kd1k40000gn/T/ipykernel_60630/3264051095.py:2: SettingW
ithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

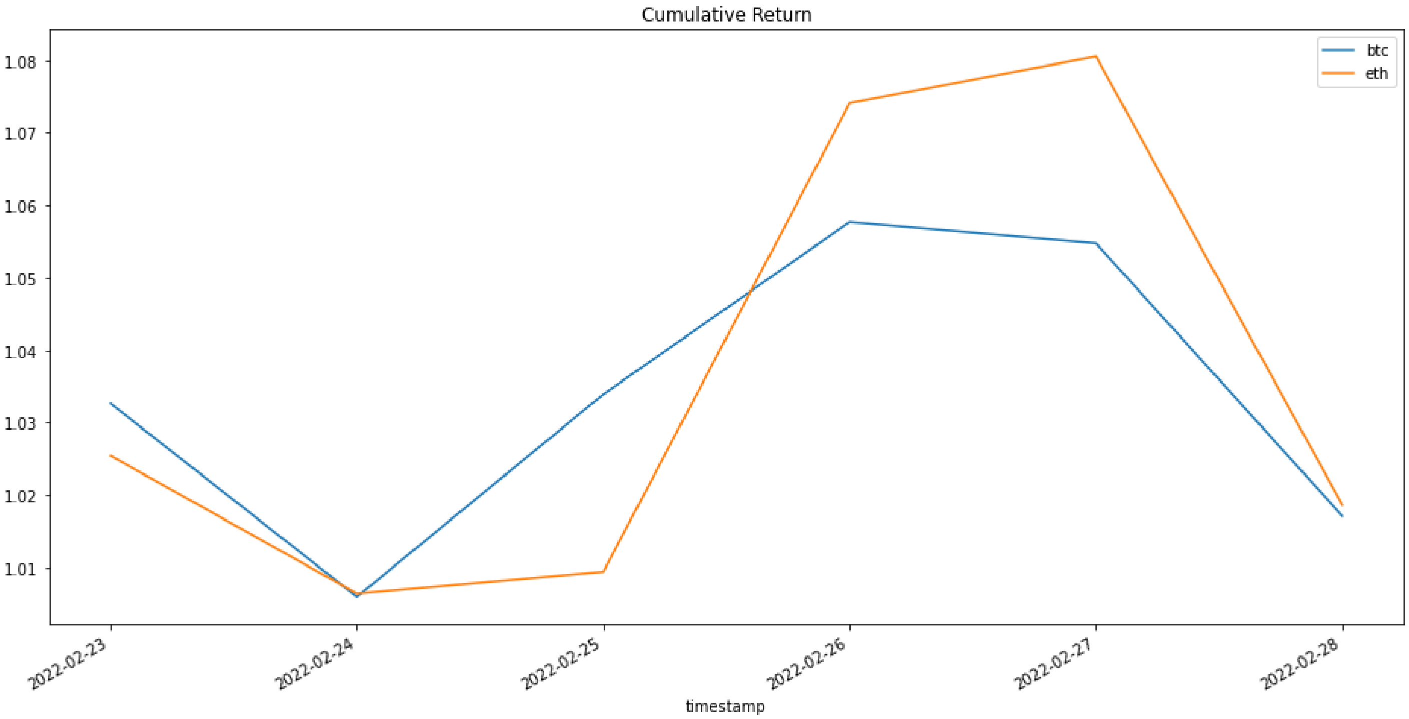
```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_gu
ide/indexing.html#returning-a-view-versus-a-copy
```

In [265…

```
btc['Cumulative Return'].plot(label='btc',figsize=(16,8),title='Cumulative Return')
eth['Cumulative Return'].plot(label='eth')
plt.legend()
```

Out[265…

```
<matplotlib.legend.Legend at 0x7fb6a8f35e50>
```



The investor who purchased Ether coins was in more profit

In [266…

```
btc.columns
```

Out[266…

```
Index(['slug', 'marketCap', 'volume', 'Open', 'Close', 'High', 'Low',
       'Total Traded', 'EWMA2', 'EWMA4', 'returns', 'Cumulative Return'],
      dtype='object')
```

Candlestick plot of bitcoin

In [267…

```
data = [go.Candlestick(x=btc.index,
                       open=btc.Open,
                       high=btc.High,
                       low=btc.Low,
                       close=btc.Close)]
```

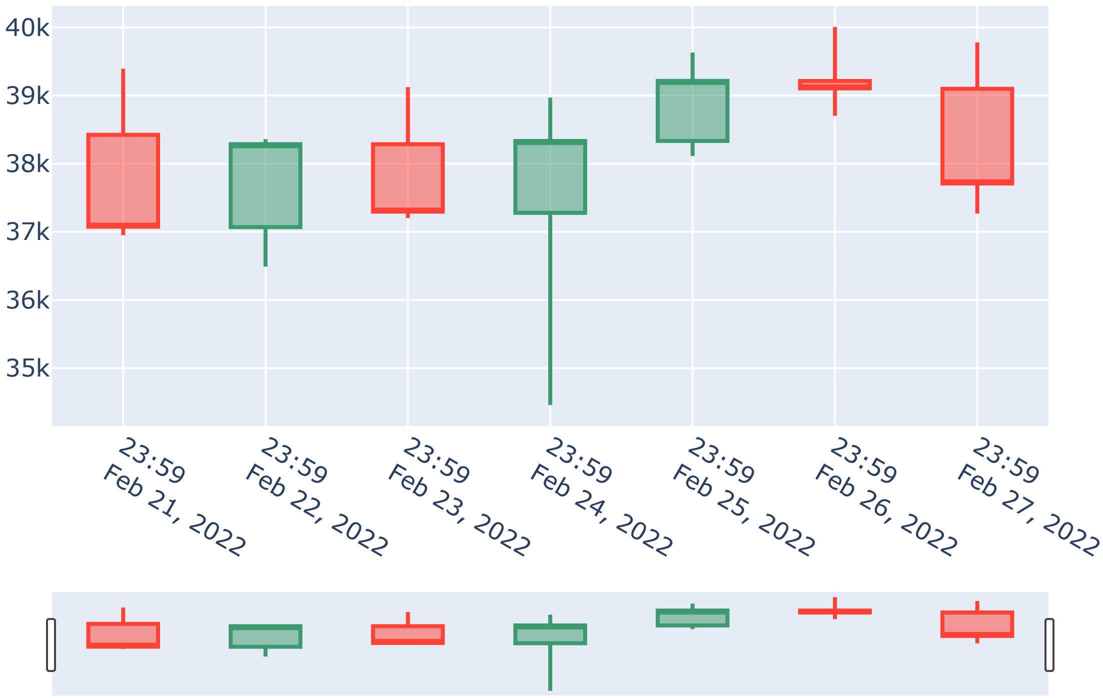
In [268…

```
layout = go.Layout(title='Bitcoin Candlestick with Range Slider',
                   xaxis={'rangeslider':{'visible':True}})
```

In [269…

```
fig = go.Figure(data=data,layout=layout)
py.iplot(fig,filename='bitcoin_candlestick')
```

Bitcoin Candlestick with Range Slider



Website -2

In [270…

```python
import requests
import pandas as pd
import numpy as np
import json
import time
import seaborn as sns
import plotly.offline as py
import plotly.graph_objs as go
from bs4 import BeautifulSoup
import pandas_datareader.data as pdr
import datetime as dt
import matplotlib.pyplot as plt
%matplotlib inline
import csv
from lxml import html
```

In [271…

```python
def fetch_coingecko_html():
    # make a request to the target website
    r = requests.get("https://www.coingecko.com")
    if r.status_code == 200:
        # if the request is successful return the HTML content
        return r.text
    else:
        # throw an exception if an error occurred
        raise Exception("an error occurred while fetching coingecko html")
```

In [272…

```python
def extract_crypto_info(html):
    # parse the HTML content with Beautiful Soup
    soup = BeautifulSoup(html, "html.parser")
    # find all the cryptocurrency elements
    coin_table = soup.find("div", {"class": "coin-table"})
    crypto_elements = coin_table.find_all("tr")[1:]

    # iterate through our cryptocurrency elements
    cryptos = []
    for crypto in crypto_elements:
        # extract the information needed using our observations
        cryptos.append({
            "name": crypto.find("td", {"class": "coin-name"})["data-sort"],
            "price": crypto.find("td", {"class": "td-price price text-right pl-0"}).text.s
            "change_1h": crypto.find("td", {"class": "td-change1h"}).text.strip(),
            "change_24h": crypto.find("td", {"class": "td-change24h"}).text.strip(),
            "change_7d": crypto.find("td", {"class": "td-change7d"}).text.strip(),
            "volume": crypto.find("td", {"class": "td-liquidity_score"}).text.strip(),
            "market_cap": crypto.find("td", {"class": "td-market_cap"}).text.strip()
        })

    return cryptos
```

In [273…

```
html = fetch_coingecko_html()

# extract our data from the HTML document
cryptos = extract_crypto_info(html)

# display the scraper results
for crypto in cryptos:
    print(crypto, "\n")
```

```
{'name': 'Bitcoin', 'price': '$37,694.67', 'change_1h': '-0.4%', 'change_24h': '-2.7%', 'c
hange_7d': '-2.1%', 'volume': '$19,109,285,099', 'market_cap': '$713,044,206,175'}

{'name': 'Ethereum', 'price': '$2,613.01', 'change_1h': '-0.4%', 'change_24h': '-4.7%', 'c
hange_7d': '-0.7%', 'volume': '$17,141,182,101', 'market_cap': '$312,228,410,324'}

{'name': 'Tether', 'price': '$1.00', 'change_1h': '-0.1%', 'change_24h': '0.4%', 'change_7
d': '0.1%', 'volume': '$43,872,567,762', 'market_cap': '$79,697,501,124'}

{'name': 'BNB', 'price': '$357.78', 'change_1h': '-0.9%', 'change_24h': '-2.7%', 'change_7
d': '-6.2%', 'volume': '$1,224,582,043', 'market_cap': '$60,062,991,135'}

{'name': 'USD Coin', 'price': '$1.00', 'change_1h': '-0.1%', 'change_24h': '-0.1%', 'chang
e_7d': '-0.0%', 'volume': '$3,634,041,761', 'market_cap': '$53,470,847,271'}

{'name': 'XRP', 'price': '$0.719535', 'change_1h': '-0.3%', 'change_24h': '-1.2%', 'change
_7d': '-7.5%', 'volume': '$3,667,262,364', 'market_cap': '$34,416,898,378'}

{'name': 'Terra', 'price': '$72.58', 'change_1h': '-0.9%', 'change_24h': '-4.2%', 'change_
7d': '46.3%', 'volume': '$2,239,503,242', 'market_cap': '$27,435,290,641'}

{'name': 'Cardano', 'price': '$0.851279', 'change_1h': '-1.0%', 'change_24h': '-1.2%', 'ch
ange_7d': '-8.9%', 'volume': '$925,144,342', 'market_cap': '$27,243,112,852'}

{'name': 'Solana', 'price': '$84.75', 'change_1h': '-1.3%', 'change_24h': '-3.1%', 'change
_7d': '-6.7%', 'volume': '$1,536,132,312', 'market_cap': '$27,090,676,946'}

{'name': 'Polkadot', 'price': '$17.30', 'change_1h': '-1.4%', 'change_24h': '-2.2%', 'chan
ge_7d': '2.1%', 'volume': '$970,403,015', 'market_cap': '$18,799,861,084'}

{'name': 'Binance USD', 'price': '$1.00', 'change_1h': '-0.3%', 'change_24h': '0.3%', 'cha
nge_7d': '0.3%', 'volume': '$3,201,196,879', 'market_cap': '$18,112,473,916'}

{'name': 'Avalanche', 'price': '$73.33', 'change_1h': '-2.5%', 'change_24h': '-7.6%', 'cha
nge_7d': '-5.3%', 'volume': '$1,123,780,981', 'market_cap': '$18,071,352,975'}
```

```
{'name': 'Dogecoin', 'price': '$0.122546', 'change_1h': '-0.8%', 'change_24h': '-2.1%', 'change_7d': '-10.7%', 'volume': '$799,833,224', 'market_cap': '$16,272,217,048'}

{'name': 'TerraUSD', 'price': '$1.00', 'change_1h': '-0.2%', 'change_24h': '0.5%', 'change_7d': '0.2%', 'volume': '$312,935,183', 'market_cap': '$12,875,706,241'}

{'name': 'Shiba Inu', 'price': '$0.000023041024', 'change_1h': '-2.0%', 'change_24h': '-3.0%', 'change_7d': '-9.8%', 'volume': '$803,427,624', 'market_cap': '$12,612,935,407'}

{'name': 'Cronos', 'price': '$0.393836', 'change_1h': '-0.5%', 'change_24h': '-3.8%', 'change_7d': '-2.9%', 'volume': '$93,077,021', 'market_cap': '$9,924,713,218'}

{'name': 'Wrapped Bitcoin', 'price': '$37,655.43', 'change_1h': '-0.4%', 'change_24h': '-2.5%', 'change_7d': '-2.3%', 'volume': '$309,086,105', 'market_cap': '$9,878,974,411'}

{'name': 'Polygon', 'price': '$1.43', 'change_1h': '-1.3%', 'change_24h': '-3.0%', 'change_7d': '-5.4%', 'volume': '$834,493,125', 'market_cap': '$9,841,423,980'}

{'name': 'Dai', 'price': '$1.00', 'change_1h': '-0.0%', 'change_24h': '0.4%', 'change_7d': '0.1%', 'volume': '$283,656,484', 'market_cap': '$9,221,205,576'}

{'name': 'Cosmos', 'price': '$26.61', 'change_1h': '-1.0%', 'change_24h': '-8.2%', 'change_7d': '1.2%', 'volume': '$683,179,804', 'market_cap': '$7,675,167,237'}

{'name': 'Litecoin', 'price': '$102.65', 'change_1h': '-0.4%', 'change_24h': '-3.7%', 'change_7d': '-7.4%', 'volume': '$555,387,760', 'market_cap': '$7,145,343,978'}

{'name': 'Chainlink', 'price': '$13.61', 'change_1h': '-0.3%', 'change_24h': '-4.6%', 'change_7d': '-7.1%', 'volume': '$585,485,015', 'market_cap': '$6,345,889,538'}

{'name': 'TRON', 'price': '$0.058054853110', 'change_1h': '-0.5%', 'change_24h': '-1.7%', 'change_7d': '-6.8%', 'volume': '$1,118,557,721', 'market_cap': '$5,902,250,549'}

{'name': 'Bitcoin Cash', 'price': '$302.73', 'change_1h': '-0.7%', 'change_24h': '-1.9%', 'change_7d': '-0.3%', 'volume': '$1,226,716,351', 'market_cap': '$5,742,405,189'}

{'name': 'FTX Token', 'price': '$41.39', 'change_1h': '-0.5%', 'change_24h': '-2.9%', 'change_7d': '0.8%', 'volume': '$79,691,952', 'market_cap': '$5,699,930,955'}

{'name': 'LEO Token', 'price': '$5.95', 'change_1h': '-0.1%', 'change_24h': '-1.9%', 'change_7d': '3.5%', 'volume': '$500,653', 'market_cap': '$5,614,391,881'}

{'name': 'Near', 'price': '$8.43', 'change_1h': '-1.0%', 'change_24h': '-4.3%', 'change_7d': '-9.5%', 'volume': '$219,365,235', 'market_cap': '$5,398,298,562'}

{'name': 'Algorand', 'price': '$0.791247', 'change_1h': '-1.0%', 'change_24h': '-5.3%', 'change_7d': '-6.6%', 'volume': '$133,380,180', 'market_cap': '$5,227,746,796'}

{'name': 'Lido Staked Ether', 'price': '$2,608.69', 'change_1h': '-0.3%', 'change_24h': '-4.3%', 'change_7d': '-1.1%', 'volume': '$3,434,592', 'market_cap': '$5,176,676,398'}

{'name': 'Stellar', 'price': '$0.178819', 'change_1h': '-1.5%', 'change_24h': '-4.4%', 'change_7d': '-8.5%', 'volume': '$196,485,554', 'market_cap': '$4,456,697,168'}

{'name': 'Uniswap', 'price': '$9.67', 'change_1h': '-1.3%', 'change_24h': '10.1%', 'change_7d': '3.1%', 'volume': '$341,861,108', 'market_cap': '$4,400,843,117'}

{'name': 'OKB', 'price': '$16.63', 'change_1h': '-1.4%', 'change_24h': '-4.0%', 'change_7d': '-12.2%', 'volume': '$90,960,513', 'market_cap': '$4,377,657,911'}

{'name': 'Hedera', 'price': '$0.209406', 'change_1h': '-0.6%', 'change_24h': '-3.9%', 'change_7d': '-3.0%', 'volume': '$53,742,653', 'market_cap': '$4,079,728,099'}

{'name': 'Fantom', 'price': '$1.57', 'change_1h': '-1.5%', 'change_24h': '-9.2%', 'change_7d': '-9.4%', 'volume': '$883,899,677', 'market_cap': '$3,975,424,940'}
```

```
{'name': 'Decentraland', 'price': '$2.50', 'change_1h': '-3.3%', 'change_24h': '-3.2%', 'change_7d': '-8.0%', 'volume': '$625,352,853', 'market_cap': '$3,721,878,007'}

{'name': 'Ethereum Classic', 'price': '$27.25', 'change_1h': '-0.8%', 'change_24h': '-3.5%', 'change_7d': '-0.8%', 'volume': '$298,074,322', 'market_cap': '$3,623,291,250'}

{'name': 'Internet Computer', 'price': '$17.15', 'change_1h': '-0.8%', 'change_24h': '-0.9%', 'change_7d': '-8.5%', 'volume': '$166,942,580', 'market_cap': '$3,534,962,649'}

{'name': 'Axie Infinity', 'price': '$47.65', 'change_1h': '-1.2%', 'change_24h': '-2.5%', 'change_7d': '-9.5%', 'volume': '$264,334,462', 'market_cap': '$3,463,299,967'}

{'name': 'Filecoin', 'price': '$19.99', 'change_1h': '-1.7%', 'change_24h': '7.4%', 'change_7d': '0.6%', 'volume': '$537,285,275', 'market_cap': '$3,376,198,433'}

{'name': 'The Sandbox', 'price': '$2.87', 'change_1h': '-1.8%', 'change_24h': '-4.0%', 'change_7d': '-10.9%', 'volume': '$663,109,335', 'market_cap': '$3,110,110,227'}

{'name': 'Klaytn', 'price': '$1.18', 'change_1h': '0.2%', 'change_24h': '-1.8%', 'change_7d': '-6.5%', 'volume': '$31,204,373', 'market_cap': '$3,094,391,199'}

{'name': 'VeChain', 'price': '$0.045115433327', 'change_1h': '-0.9%', 'change_24h': '-4.0%', 'change_7d': '-8.7%', 'volume': '$250,079,260', 'market_cap': '$3,006,377,487'}

{'name': 'Elrond', 'price': '$134.62', 'change_1h': '-1.2%', 'change_24h': '-2.5%', 'change_7d': '-13.5%', 'volume': '$129,768,785', 'market_cap': '$2,948,776,670'}

{'name': 'cETH', 'price': '$52.43', 'change_1h': '-0.6%', 'change_24h': '-3.6%', 'change_7d': '-1.0%', 'volume': '$513.80', 'market_cap': '$2,938,755,992'}

{'name': 'Frax', 'price': '$1.00', 'change_1h': '0.0%', 'change_24h': '0.1%', 'change_7d': '0.1%', 'volume': '$36,305,300', 'market_cap': '$2,854,335,048'}

{'name': 'cUSDC', 'price': '$0.022548452632', 'change_1h': '-0.2%', 'change_24h': '0.1%', 'change_7d': '-0.1%', 'volume': '$0.000000000000', 'market_cap': '$2,826,419,880'}

{'name': 'Magic Internet Money', 'price': '$0.994695', 'change_1h': '-0.6%', 'change_24h': '-0.1%', 'change_7d': '-0.0%', 'volume': '$56,626,976', 'market_cap': '$2,770,398,683'}

{'name': 'Osmosis', 'price': '$8.98', 'change_1h': '-0.8%', 'change_24h': '-6.3%', 'change_7d': '5.2%', 'volume': '$81,896,693', 'market_cap': '$2,760,790,556'}

{'name': 'Monero', 'price': '$149.76', 'change_1h': '-0.8%', 'change_24h': '-2.7%', 'change_7d': '-2.9%', 'volume': '$88,921,712', 'market_cap': '$2,702,082,260'}

{'name': 'Tezos', 'price': '$3.07', 'change_1h': '-1.5%', 'change_24h': '-7.1%', 'change_7d': '-6.5%', 'volume': '$158,997,035', 'market_cap': '$2,687,033,566'}

{'name': 'Theta Network', 'price': '$2.68', 'change_1h': '-2.6%', 'change_24h': '-2.3%', 'change_7d': '-9.9%', 'volume': '$198,475,256', 'market_cap': '$2,679,045,507'}

{'name': 'ECOMI', 'price': '$0.004043449591', 'change_1h': '-0.3%', 'change_24h': '-6.4%', 'change_7d': '-18.1%', 'volume': '$2,775,590', 'market_cap': '$2,352,128,129'}

{'name': 'Helium', 'price': '$22.48', 'change_1h': '-1.6%', 'change_24h': '-5.5%', 'change_7d': '-2.2%', 'volume': '$15,824,734', 'market_cap': '$2,255,816,955'}

{'name': 'The Graph', 'price': '$0.332918', 'change_1h': '-1.0%', 'change_24h': '-0.4%', 'change_7d': '-15.6%', 'volume': '$115,856,053', 'market_cap': '$2,045,603,373'}

{'name': 'EOS', 'price': '$2.08', 'change_1h': '-1.1%', 'change_24h': '-2.4%', 'change_7d': '-5.9%', 'volume': '$354,300,342', 'market_cap': '$2,045,093,555'}

{'name': 'cDAI', 'price': '$0.021883262123', 'change_1h': '-0.2%', 'change_24h': '0.0%', 'change_7d': '-0.1%', 'volume': '$165.96', 'market_cap': '$1,995,992,138'}
```

```
{'name': 'IOTA', 'price': '$0.715505', 'change_1h': '0.4%', 'change_24h': '-0.9%', 'change_7d': '-9.7%', 'volume': '$37,364,794', 'market_cap': '$1,986,344,707'}

{'name': 'Theta Fuel', 'price': '$0.162775', 'change_1h': '-0.9%', 'change_24h': '-1.3%', 'change_7d': '-11.5%', 'volume': '$23,031,116', 'market_cap': '$1,886,178,774'}

{'name': 'Flow', 'price': '$5.65', 'change_1h': '-1.9%', 'change_24h': '2.1%', 'change_7d': '-2.8%', 'volume': '$195,022,836', 'market_cap': '$1,864,477,714'}

{'name': 'Aave', 'price': '$129.20', 'change_1h': '-1.9%', 'change_24h': '-1.7%', 'change_7d': '-6.2%', 'volume': '$224,350,934', 'market_cap': '$1,757,892,438'}

{'name': 'BitTorrent', 'price': '$0.000001846362', 'change_1h': '-0.7%', 'change_24h': '-2.7%', 'change_7d': '-6.7%', 'volume': '$79,546,943', 'market_cap': '$1,717,215,087'}

{'name': 'Gala', 'price': '$0.225897', 'change_1h': '-2.5%', 'change_24h': '-5.5%', 'change_7d': '-10.5%', 'volume': '$654,743,031', 'market_cap': '$1,699,155,741'}

{'name': 'PancakeSwap', 'price': '$6.19', 'change_1h': '-0.7%', 'change_24h': '-3.2%', 'change_7d': '-16.0%', 'volume': '$69,312,836', 'market_cap': '$1,682,773,404'}

{'name': 'Harmony', 'price': '$0.138923', 'change_1h': '-1.3%', 'change_24h': '-7.0%', 'change_7d': '-8.0%', 'volume': '$135,789,461', 'market_cap': '$1,637,928,652'}

{'name': 'Maker', 'price': '$1,783.82', 'change_1h': '-1.6%', 'change_24h': '-6.3%', 'change_7d': '0.1%', 'volume': '$52,427,561', 'market_cap': '$1,605,180,331'}

{'name': 'Bitcoin SV', 'price': '$81.80', 'change_1h': '-0.1%', 'change_24h': '-1.9%', 'change_7d': '-3.6%', 'volume': '$61,782,775', 'market_cap': '$1,552,758,457'}

{'name': 'Huobi BTC', 'price': '$37,768.42', 'change_1h': '-0.4%', 'change_24h': '-2.2%', 'change_7d': '-0.6%', 'volume': '$1,352,771', 'market_cap': '$1,505,263,940'}

{'name': 'TrueUSD', 'price': '$1.00', 'change_1h': '0.0%', 'change_24h': '0.5%', 'change_7d': '0.1%', 'volume': '$72,965,698', 'market_cap': '$1,484,103,867'}

{'name': 'JUNO', 'price': '$32.51', 'change_1h': '-0.5%', 'change_24h': '-10.3%', 'change_7d': '18.0%', 'volume': '$9,865,626', 'market_cap': '$1,442,194,266'}

{'name': 'KuCoin Token', 'price': '$18.95', 'change_1h': '-0.4%', 'change_24h': '-2.7%', 'change_7d': '-0.6%', 'volume': '$10,831,712', 'market_cap': '$1,440,804,815'}

{'name': 'Arweave', 'price': '$28.49', 'change_1h': '-1.4%', 'change_24h': '6.3%', 'change_7d': '8.4%', 'volume': '$34,264,805', 'market_cap': '$1,423,309,217'}

{'name': 'Huobi Token', 'price': '$9.11', 'change_1h': '-0.1%', 'change_24h': '0.7%', 'change_7d': '-5.1%', 'volume': '$37,035,243', 'market_cap': '$1,422,935,731'}

{'name': 'Quant', 'price': '$106.04', 'change_1h': '0.1%', 'change_24h': '-1.6%', 'change_7d': '-2.1%', 'volume': '$28,871,743', 'market_cap': '$1,420,741,141'}

{'name': 'NEO', 'price': '$19.74', 'change_1h': '-1.1%', 'change_24h': '-3.4%', 'change_7d': '-15.8%', 'volume': '$137,418,115', 'market_cap': '$1,388,351,838'}

{'name': 'eCash', 'price': '$0.000073193736', 'change_1h': '-1.0%', 'change_24h': '-1.4%', 'change_7d': '-3.7%', 'volume': '$20,901,174', 'market_cap': '$1,385,940,401'}

{'name': 'Radix', 'price': '$0.133716', 'change_1h': '-3.2%', 'change_24h': '-9.1%', 'change_7d': '-0.1%', 'volume': '$148,022', 'market_cap': '$1,315,016,211'}

{'name': 'Humans.ai', 'price': '$0.122604', 'change_1h': '-0.2%', 'change_24h': '-0.7%', 'change_7d': '-13.1%', 'volume': '$513,250', 'market_cap': '$1,309,127,902'}

{'name': 'Amp', 'price': '$0.026738405194', 'change_1h': '-0.6%', 'change_24h': '-3.1%', 'change_7d': '-1.8%', 'volume': '$8,062,666', 'market_cap': '$1,282,498,443'}
```

```
{'name': 'Enjin Coin', 'price': '$1.35', 'change_1h': '-1.8%', 'change_24h': '-3.6%', 'cha
nge_7d': '-12.1%', 'volume': '$131,829,230', 'market_cap': '$1,260,601,040'}

{'name': 'Zcash', 'price': '$101.05', 'change_1h': '-1.9%', 'change_24h': '-4.9%', 'change
_7d': '-6.7%', 'volume': '$150,049,241', 'market_cap': '$1,227,997,465'}

{'name': 'Celsius Network', 'price': '$2.90', 'change_1h': '0.2%', 'change_24h': '-6.8%',
'change_7d': '-7.7%', 'volume': '$2,054,041', 'market_cap': '$1,227,518,550'}

{'name': 'Frax Share', 'price': '$20.63', 'change_1h': '-0.3%', 'change_24h': '3.1%', 'cha
nge_7d': '-4.5%', 'volume': '$8,357,101', 'market_cap': '$1,189,858,625'}

{'name': 'Stacks', 'price': '$1.14', 'change_1h': '-0.3%', 'change_24h': '-2.6%', 'change_
7d': '-9.0%', 'volume': '$20,020,153', 'market_cap': '$1,188,652,135'}

{'name': 'Waves', 'price': '$11.41', 'change_1h': '1.0%', 'change_24h': '-7.9%', 'change_7
d': '20.8%', 'volume': '$299,153,850', 'market_cap': '$1,130,386,461'}

{'name': 'NEXO', 'price': '$1.92', 'change_1h': '0.1%', 'change_24h': '-3.7%', 'change_7
d': '0.6%', 'volume': '$3,933,499', 'market_cap': '$1,076,116,638'}

{'name': 'Kusama', 'price': '$115.87', 'change_1h': '-1.4%', 'change_24h': '-3.8%', 'chang
e_7d': '-16.8%', 'volume': '$34,075,751', 'market_cap': '$1,039,304,644'}

{'name': 'THORChain', 'price': '$3.45', 'change_1h': '-1.3%', 'change_24h': '-7.1%', 'chan
ge_7d': '-10.0%', 'volume': '$44,861,555', 'market_cap': '$1,036,538,587'}

{'name': 'Pax Dollar', 'price': '$1.00', 'change_1h': '-0.2%', 'change_24h': '0.1%', 'chan
ge_7d': '0.0%', 'volume': '$15,666,050', 'market_cap': '$1,022,303,070'}

{'name': 'BitDAO', 'price': '$1.18', 'change_1h': '0.0%', 'change_24h': '-3.1%', 'change_7
d': '-20.0%', 'volume': '$36,460,208', 'market_cap': '$1,011,591,916'}

{'name': 'Kadena', 'price': '$5.89', 'change_1h': '-0.2%', 'change_24h': '-7.9%', 'change_
7d': '-4.6%', 'volume': '$14,043,258', 'market_cap': '$1,010,886,341'}

{'name': 'Basic Attention Token', 'price': '$0.662959', 'change_1h': '-0.9%', 'change_24
h': '-2.9%', 'change_7d': '-6.1%', 'volume': '$69,502,542', 'market_cap': '$990,190,349'}

{'name': 'GateToken', 'price': '$6.36', 'change_1h': '-0.1%', 'change_24h': '-0.0%', 'chan
ge_7d': '1.3%', 'volume': '$3,499,153', 'market_cap': '$965,547,084'}

{'name': 'Celo', 'price': '$2.36', 'change_1h': '-0.5%', 'change_24h': '0.9%', 'change_7
d': '-8.1%', 'volume': '$32,755,273', 'market_cap': '$958,588,004'}

{'name': 'Anchor Protocol', 'price': '$3.66', 'change_1h': '-0.6%', 'change_24h': '2.7%',
'change_7d': '65.7%', 'volume': '$48,214,680', 'market_cap': '$938,162,153'}

{'name': 'Dash', 'price': '$87.37', 'change_1h': '-0.9%', 'change_24h': '-4.4%', 'change_7
d': '-11.7%', 'volume': '$106,143,579', 'market_cap': '$925,226,780'}

{'name': 'NEM', 'price': '$0.099631475047', 'change_1h': '-1.2%', 'change_24h': '0.9%', 'c
hange_7d': '0.8%', 'volume': '$19,303,859', 'market_cap': '$895,431,879'}

{'name': 'Loopring', 'price': '$0.719976', 'change_1h': '-1.2%', 'change_24h': '-3.7%', 'c
hange_7d': '-11.5%', 'volume': '$116,339,145', 'market_cap': '$895,017,098'}

{'name': 'Convex Finance', 'price': '$17.16', 'change_1h': '-1.4%', 'change_24h': '-9.7%',
'change_7d': '-26.3%', 'volume': '$7,187,733', 'market_cap': '$892,272,601'}

{'name': 'Chiliz', 'price': '$0.166327', 'change_1h': '-0.9%', 'change_24h': '-2.5%', 'cha
nge_7d': '-11.4%', 'volume': '$69,126,532', 'market_cap': '$886,815,384'}

{'name': 'Curve DAO Token', 'price': '$2.17', 'change_1h': '-0.4%', 'change_24h': '-5.6%',
'change_7d': '-15.2%', 'volume': '$168,879,354', 'market_cap': '$848,402,400'}
```

In [274…

```
import json

# save the results locally in JSON
with open("coingecko.json", "w") as f:
    f.write(json.dumps(cryptos, indent=2))
```

In [275…

```
crypto_header=['name', 'price','change_1h','change_24h','change_7d','volume','market_cap']
with open("/Users/anitateladevalapalli/Documents/coingecko.csv", 'w') as csvfile:
    writer = csv.DictWriter(csvfile, fieldnames = crypto_header)
    writer.writeheader()
    writer.writerows(cryptos)
```

In [276…

```
crypto_df = pd.read_csv("/Users/anitateladevalapalli/Documents/coingecko.csv")
```

In [277…

```
crypto_df.head()
```

Out[277…

	name	price	change_1h	change_24h	change_7d	volume	market_cap
0	Bitcoin	$37,694.67	-0.4%	-2.7%	-2.1%	$19,109,285,099	$713,044,206,175
1	Ethereum	$2,613.01	-0.4%	-4.7%	-0.7%	$17,141,182,101	$312,228,410,324
2	Tether	$1.00	-0.1%	0.4%	0.1%	$43,872,567,762	$79,697,501,124
3	BNB	$357.78	-0.9%	-2.7%	-6.2%	$1,224,582,043	$60,062,991,135
4	USD Coin	$1.00	-0.1%	-0.1%	-0.0%	$3,634,041,761	$53,470,847,271

In [278…

```
crypto_df.isna().sum()
```

Out[278…

```
name          0
price         0
change_1h     0
change_24h    0
change_7d     0
volume        0
market_cap    0
dtype: int64
```

In [279…

```
#%timeit [x.strip('$') for x in crypto_df.price]
crypto_df['change_1h'] = crypto_df['change_1h'].str.replace('%', '').astype(float)




crypto_df['change_24h'] = crypto_df['change_24h'].str.replace('%', '').astype(float)




crypto_df['change_7d'] = crypto_df['change_7d'].str.replace('%', '').astype(float)
```

In [280…

```
crypto_df['price'] = crypto_df['price'].str.replace('$', '')



crypto_df['market_cap'] = crypto_df['market_cap'].str.replace('$', '')



crypto_df['volume'] = crypto_df['volume'].str.replace('$', '')
```

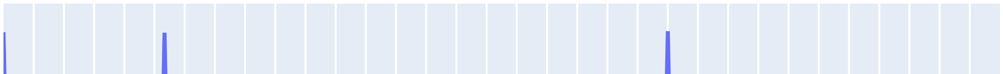
```
/var/folders/6f/c2b7vdpx247cstzj573kd1k40000gn/T/ipykernel_60630/1078236830.py:1: FutureWa
rning:

The default value of regex will change from True to False in a future version. In additio
n, single character regular expressions will *not* be treated as literal strings when rege
x=True.

/var/folders/6f/c2b7vdpx247cstzj573kd1k40000gn/T/ipykernel_60630/1078236830.py:7: FutureWa
rning:

The default value of regex will change from True to False in a future version. In additio
n, single character regular expressions will *not* be treated as literal strings when rege
x=True.

/var/folders/6f/c2b7vdpx247cstzj573kd1k40000gn/T/ipykernel_60630/1078236830.py:14: FutureW
arning:

The default value of regex will change from True to False in a future version. In additio
n, single character regular expressions will *not* be treated as literal strings when rege
x=True.
```

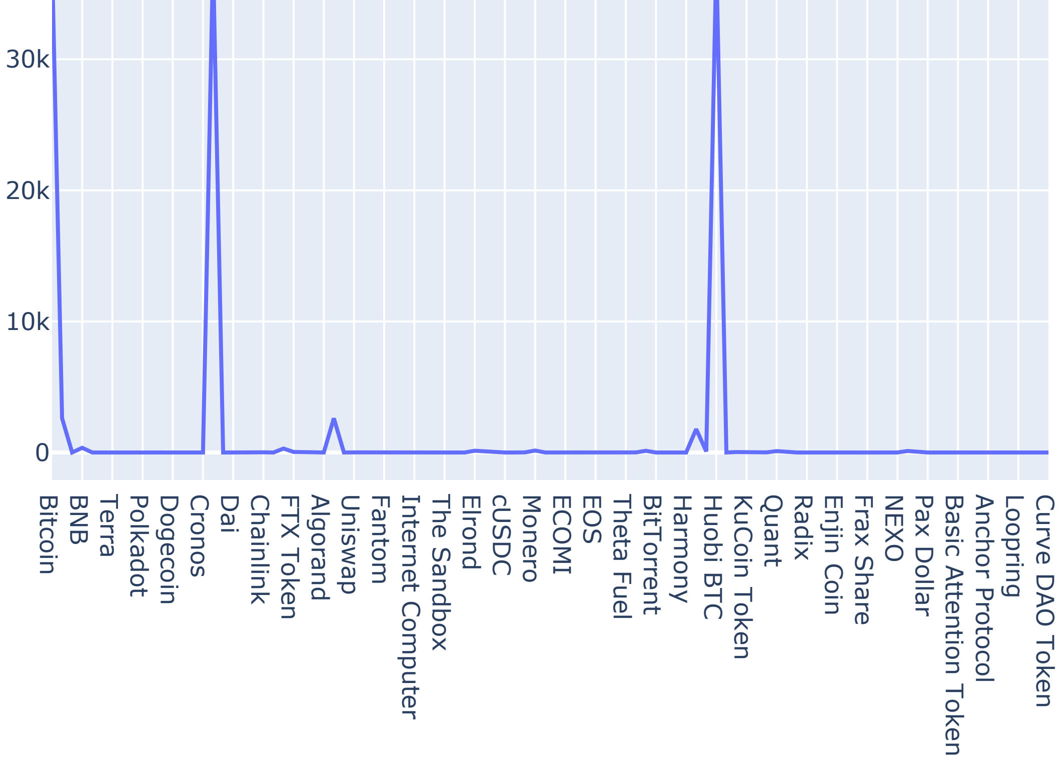
In [281…

```
crypto_df['price'] = crypto_df['price'].str.replace(',', '')


crypto_df['market_cap'] = crypto_df['market_cap'].str.replace(',', '')

crypto_df['volume'] = crypto_df['volume'].str.replace(',', '')



crypto_df['price'] = crypto_df['price'].astype(float)

crypto_df['market_cap'] = crypto_df['market_cap'].astype(float)

crypto_df['volume'] = crypto_df['volume'].astype(float)
```

In [282…

```
plota = go.Scatter(x=crypto_df['name'], y=crypto_df['price'])
py.iplot([plota])
```





In [283…

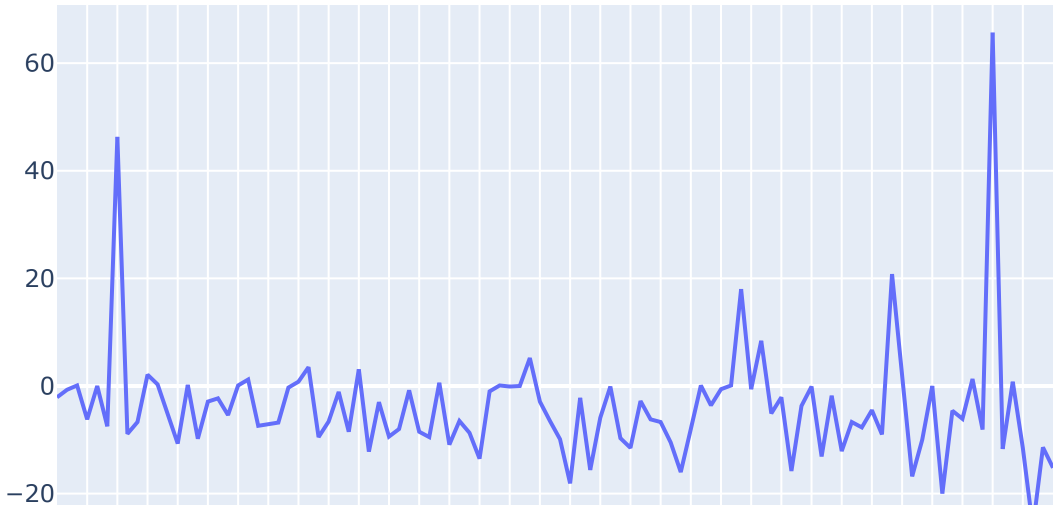
```
crypto_df.head()
```

Out[283…

	name	price	change_1h	change_24h	change_7d	volume	market_cap
0	Bitcoin	37694.67	-0.4	-2.7	-2.1	1.910929e+10	7.130442e+11
1	Ethereum	2613.01	-0.4	-4.7	-0.7	1.714118e+10	3.122284e+11
2	Tether	1.00	-0.1	0.4	0.1	4.387257e+10	7.969750e+10
3	BNB	357.78	-0.9	-2.7	-6.2	1.224582e+09	6.006299e+10
4	USD Coin	1.00	-0.1	-0.1	-0.0	3.634042e+09	5.347085e+10

In [284…

```
plot_mc = go.Scatter(x=crypto_df['name'], y=crypto_df['change_7d'])
py.iplot([plot_mc])
```

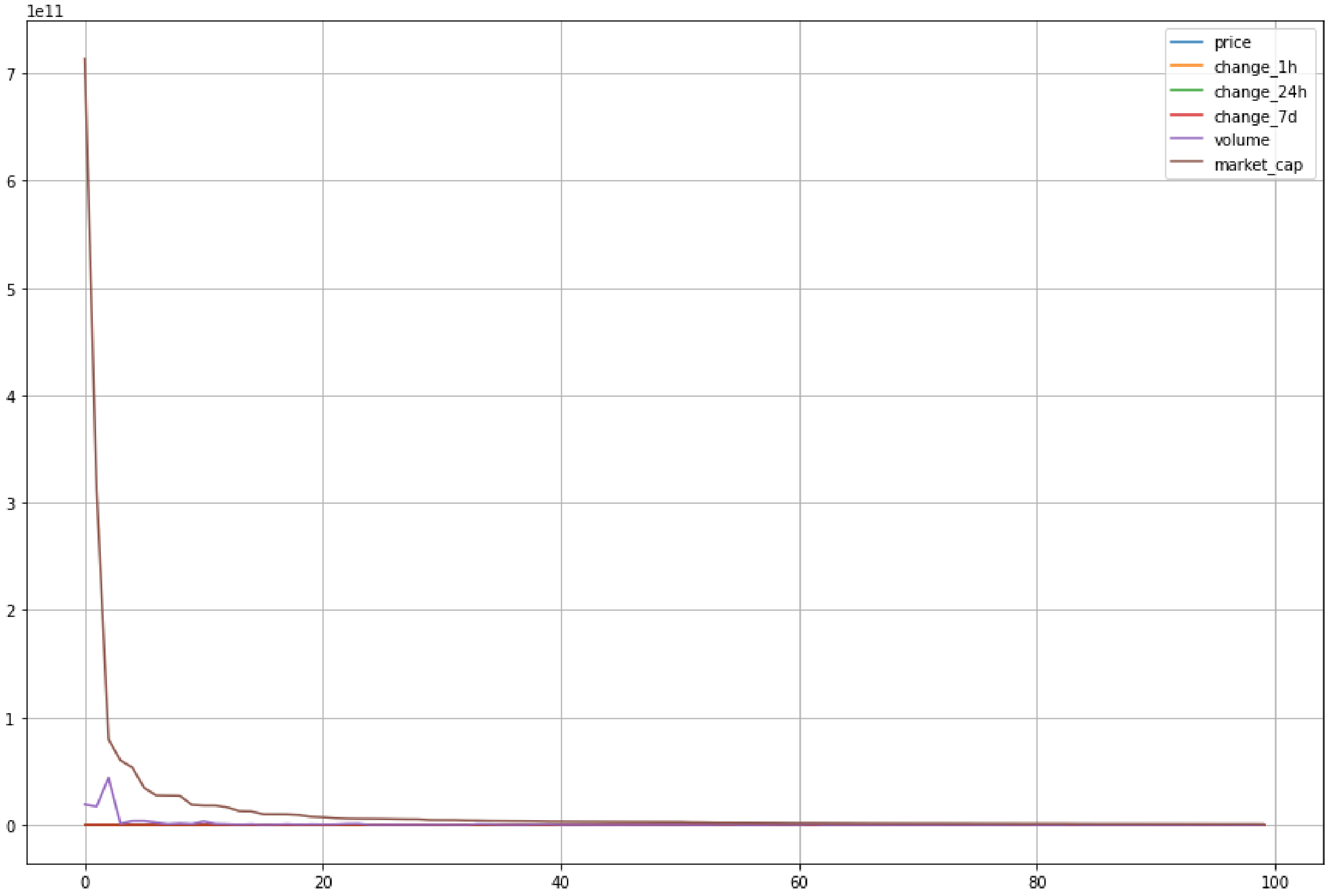


In [285…

```
crypto_df.plot(grid=True, figsize=(15, 10))
```

Out[285…

```
<AxesSubplot:>
```

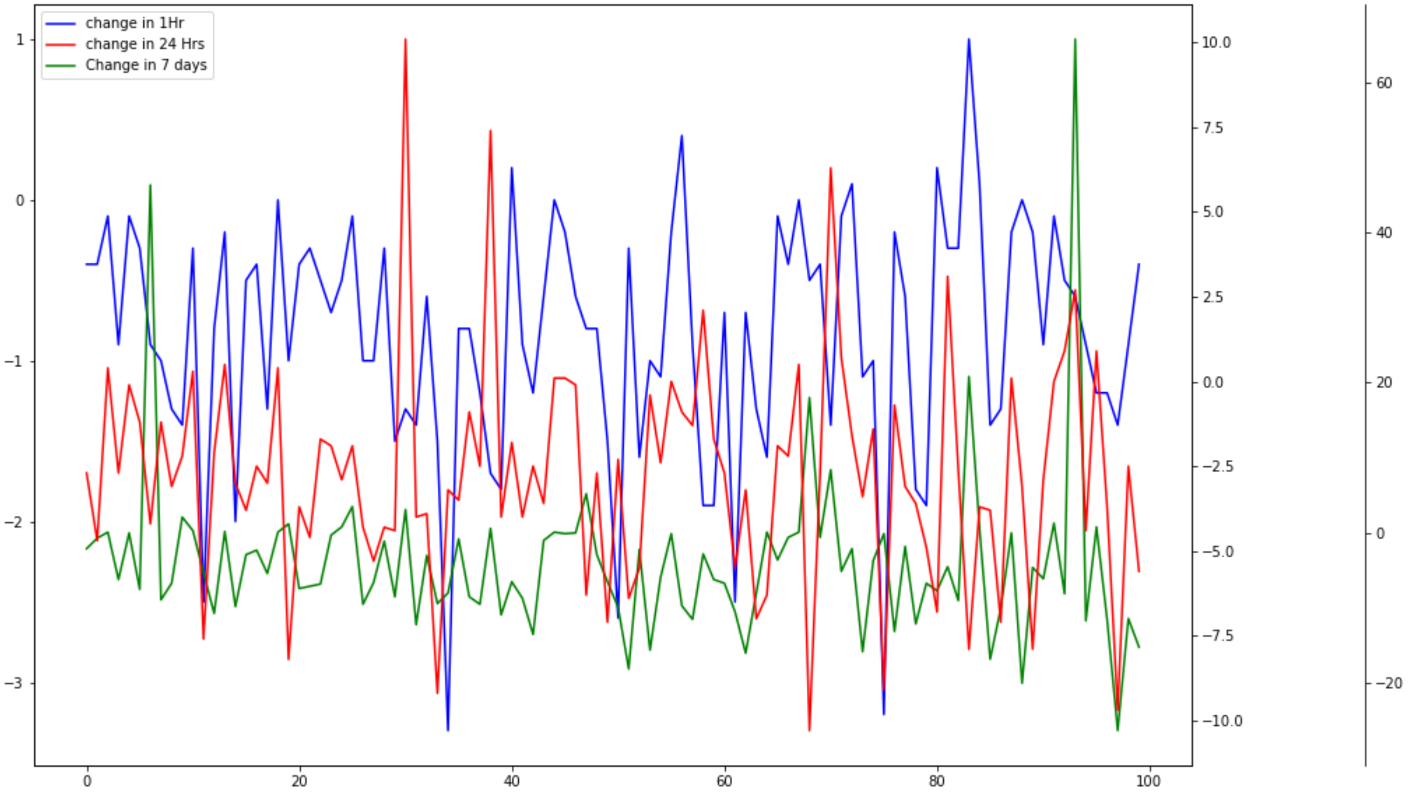


In [286…

```
fig, ax1 = plt.subplots(figsize=(20, 10))
ax2 = ax1.twinx()
rspine = ax2.spines['right']
rspine.set_position(('axes', 1.15))
ax2.set_frame_on(True)
ax2.patch.set_visible(False)
fig.subplots_adjust(right=0.7)
crypto_df['change_1h'].plot(ax=ax1, style='b-')
crypto_df['change_24h'].plot(ax=ax1, style='r-', secondary_y=True)
crypto_df['change_7d'].plot(ax=ax2, style='g-')
# legend
ax2.legend([ax1.get_lines()[0],
            ax1.right_ax.get_lines()[0],
            ax2.get_lines()[0]],
           ['change in 1Hr', 'change in 24 Hrs', 'Change in 7 days'])
```

```
<matplotlib.legend.Legend at 0x7fb6c138db20>
```

Out[286…



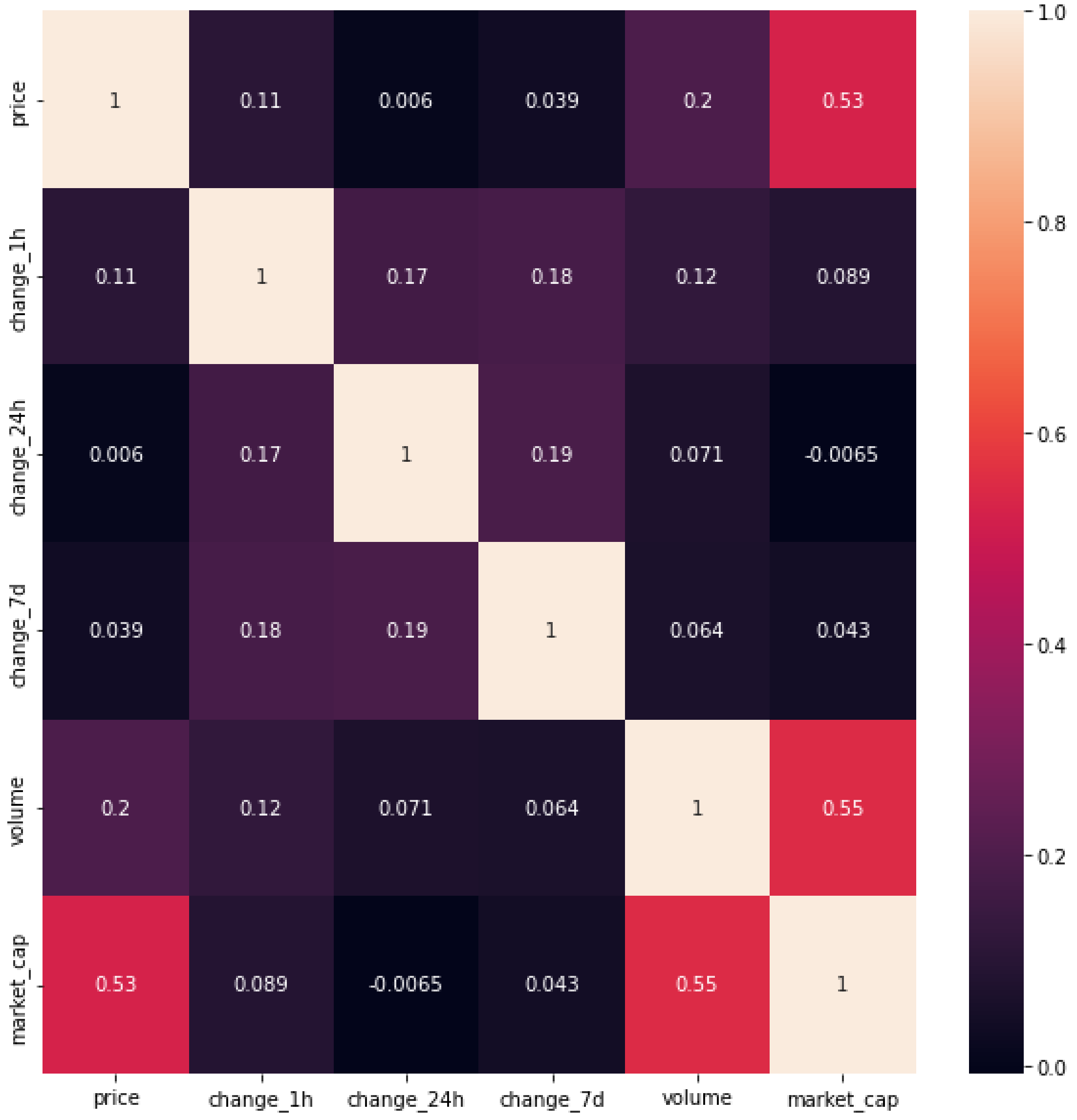
In [287…

```python
import seaborn as sns
import matplotlib.pyplot as plt
# Compute the correlation matrix
corr = crypto_df.corr()
# Generate a mask for the upper triangle
mask = np.zeros_like(corr, dtype=np.bool)
mask[np.triu_indices_from(mask)] = True
# Set up the matplotlib figure
f, ax = plt.subplots(figsize=(10, 10))
# Draw the heatmap with the mask and correct aspect ratio
sns.heatmap(corr, annot=True)
```

```
/var/folders/6f/c2b7vdpx247cstzj573kd1k40000gn/T/ipykernel_60630/2490745844.py:6: Deprecat
ionWarning:

`np.bool` is a deprecated alias for the builtin `bool`. To silence this warning, use `bool
` by itself. Doing this will not modify any behavior and is safe. If you specifically want
ed the numpy scalar type, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/releas
e/1.20.0-notes.html#deprecations
```

Out[287…

```
<AxesSubplot:>
```


price
change_1h
change_24h
change_7d
volume
market_cap
1 0.11 0.006 0.039 0.2 0.53
0.11 1 0.17 0.18 0.12 0.089
0.006 0.17 1 0.19 0.071 -0.0065
0.039 0.18 0.19 1 0.064 0.043
0.2 0.12 0.071 0.064 1 0.55
0.53 0.089 -0.0065 0.043 0.55 1
price
change_1h
change_24h
change_7d
volume
market_cap
1.0
0.8
0.6
0.4
0.2
0.0