Assignment Week 9 - 10 Term Project Milestone- 4 Digital Currency Cleaning/Formatting using API connection of the given Data

'''

Name : Karthikeyan Chellamuthu

Date : 05-22-2022

'''

In [1]:
```
# Importing necessary libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from requests import Request, Session
from requests.exceptions import ConnectionError, Timeout, TooManyRedirects
import json

import os
%matplotlib inline

import warnings
warnings.filterwarnings("ignore")
```

Step 1: Reading data using "requests"/API library's in Python

'''In this week we are trying to analyze if the digital currency is suitable for investment?. In the previous milestones, we have analyzed the csv files having historical trading data for various digital currency and see the price fluctuation is high compared to the trading price of stocks and web data showing metadata about digital currency. We have applied all the required transformations on csv files (data) and web data, and created final datasets with can be used to make a join with other type of datasets. As part of this milestone, we are going to analyze API data. I have created a API in coinmarketcap to get the latest data for the digital currency. The dataset contains so many digital currency. However, we will consider only the digital currency that we have taken into consideration during previous milestones. We can make a request based on number of digitalcurrency or based on name.
https://pro-api.coinmarketcap.com/v1/cryptocurrency/quotes/latest '''

In [21]:
```
# A Json file has been created  to read API key and request for the required data
#  Read data from json file and create headers
with open("api_key.json") as f:
    json_data = json.load(f)
    f.close()
```

In [11]:
```
# Assigning the json_data to headers variable and assign API url to url variable

headers = json_data
url = 'https://pro-api.coinmarketcap.com/v1/cryptocurrency/listings/latest'
```

In [113…
```
# Creating a list of digitalcurrency which we need to request in API to get current

crypto_currencies = ['ADA','BCH','DOGE','BTC','ETC','ETH','LTC','NEO']
```

In [38]:
```
# Requesting Digital currency data from API and Declaring session required to reques

session = Session()
session.headers.update(headers)
```

In [94]:
```
# Take a DOGE sample digital coing  by calling the API request to print/display

parameters['symbol'] = crypto_currencies[2]
print(parameters)

try:
    response = session.get(url, params=parameters)
    data = json.loads(response.text)
    print(data)
except (ConnectionError, Timeout, TooManyRedirects) as e:
        print(e)
```

```
{'symbol': 'DOGE'}
{'status': {'timestamp': '2022-02-16T01:17:35.018Z', 'error_code': 0, 'error_messag
e': None, 'elapsed': 30, 'credit_count': 1, 'notice': None}, 'data': {'DOGE': {'id':
74, 'name': 'Dogecoin', 'symbol': 'DOGE', 'slug': 'dogecoin', 'num_market_pairs': 44
6, 'date_added': '2013-12-15T00:00:00.000Z', 'tags': ['mineable', 'pow', 'scrypt',
'medium-of-exchange', 'memes', 'payments', 'binance-smart-chain', 'doggone-doggere
l'], 'max_supply': None, 'circulating_supply': 132670764299.89409, 'total_supply': 1
32670764299.89409, 'is_active': 1, 'platform': None, 'cmc_rank': 11, 'is_fiat': 0,
'self_reported_circulating_supply': None, 'self_reported_market_cap': None, 'last_up
dated': '2022-02-16T01:16:00.000Z', 'quote': {'USD': {'price': 0.1501636704859218,
'volume_24h': 678306650.7534797, 'volume_change_24h': -13.1853, 'percent_change_1h':
-0.72407569, 'percent_change_24h': 3.33607896, 'percent_change_7d': -5.11055103, 'pe
rcent_change_30d': -15.14272417, 'percent_change_60d': -10.68758088, 'percent_change
_90d': -37.02955627, 'market_cap': 19922328933.444695, 'market_cap_dominance': 0.999
2, 'fully_diluted_market_cap': 19922328933.44, 'last_updated': '2022-02-16T01:16:00.
000Z'}}}}}
```

'''The inference is the given data in nested dictionary format. Hence before reading the data into dataframe, we have to flattened the data so that we can create dataframe for the dataset. Dreating a function called "flatten" to flatten the nested dictionary into linear format'''

In [63]:
```
# Function to flatten the nested dictionary into linear format

import collections

def flatten(d, parent_key='', sep='_'):
    items = []
    for k, v in d.items():
        new_key = parent_key + sep + k if parent_key else k
        if isinstance(v, collections.MutableMapping):
            items.extend(flatten(v, new_key, sep=sep).items())
```

```
        else:
            items.append((new_key, v))
    return dict(items)
```

In [95]:

```
# Flattening the dataset for Doge digital coin from the above steps

flatten_data = flatten(data)
data_columns = flatten_data.keys()
print("Print the column names:")
print(data_columns)
```

```
Print the column names:
dict_keys(['status_timestamp', 'status_error_code', 'status_error_message', 'status_
elapsed', 'status_credit_count', 'status_notice', 'data_DOGE_id', 'data_DOGE_name',
'data_DOGE_symbol', 'data_DOGE_slug', 'data_DOGE_num_market_pairs', 'data_DOGE_date_
added', 'data_DOGE_tags', 'data_DOGE_max_supply', 'data_DOGE_circulating_supply', 'd
ata_DOGE_total_supply', 'data_DOGE_is_active', 'data_DOGE_platform', 'data_DOGE_cmc_
rank', 'data_DOGE_is_fiat', 'data_DOGE_self_reported_circulating_supply', 'data_DOGE
_self_reported_market_cap', 'data_DOGE_last_updated', 'data_DOGE_quote_USD_price',
'data_DOGE_quote_USD_volume_24h', 'data_DOGE_quote_USD_volume_change_24h', 'data_DOG
E_quote_USD_percent_change_1h', 'data_DOGE_quote_USD_percent_change_24h', 'data_DOGE
_quote_USD_percent_change_7d', 'data_DOGE_quote_USD_percent_change_30d', 'data_DOGE_
quote_USD_percent_change_60d', 'data_DOGE_quote_USD_percent_change_90d', 'data_DOGE_
quote_USD_market_cap', 'data_DOGE_quote_USD_market_cap_dominance', 'data_DOGE_quote_
USD_fully_diluted_market_cap', 'data_DOGE_quote_USD_last_updated'])
```

In [97]:

```
# Attribute Formatting and  removing DOGE

column_list = []
for col in data_columns:
    if 'DOGE' in col:
        col = col.replace('DOGE_','')
    column_list.append(col)
print(column_list)
```

```
['status_timestamp', 'status_error_code', 'status_error_message', 'status_elapsed',
'status_credit_count', 'status_notice', 'data_id', 'data_name', 'data_symbol', 'data
_slug', 'data_num_market_pairs', 'data_date_added', 'data_tags', 'data_max_supply',
'data_circulating_supply', 'data_total_supply', 'data_is_active', 'data_platform',
'data_cmc_rank', 'data_is_fiat', 'data_self_reported_circulating_supply', 'data_self
_reported_market_cap', 'data_last_updated', 'data_quote_USD_price', 'data_quote_USD_
volume_24h', 'data_quote_USD_volume_change_24h', 'data_quote_USD_percent_change_1h',
'data_quote_USD_percent_change_24h', 'data_quote_USD_percent_change_7d', 'data_quote
_USD_percent_change_30d', 'data_quote_USD_percent_change_60d', 'data_quote_USD_perce
nt_change_90d', 'data_quote_USD_market_cap', 'data_quote_USD_market_cap_dominance',
'data_quote_USD_fully_diluted_market_cap', 'data_quote_USD_last_updated']
```

In [124…

```
# Create a for Loop through the list of digital currency  declared in the previous s

df_list = []
for crypto in crypto_currencies:

    parameters['symbol'] = crypto
    try:
        response = session.get(url, params=parameters)
        data = json.loads(response.text)
        print("\nPrinting the details for the cryptocurrency: {}".format(crypto))
        print(data)

        flatten_data = flatten(data)
        print(flatten_data)
```

```
        df = pd.DataFrame(flatten_data)
        df.columns = column_list
        print('The shape of dataframe: {}'.format(df.shape))
        df_list.append(df)

    except (ConnectionError, Timeout, TooManyRedirects) as e:
        print(e)

df = pd.concat(df_list, axis=0)  # Adding the dataframe to df_list
```

```
Printing the details for the cryptocurrency: ADA
{'status': {'timestamp': '2022-02-16T01:49:28.198Z', 'error_code': 0, 'error_messag
e': None, 'elapsed': 35, 'credit_count': 1, 'notice': None}, 'data': {'ADA': {'id':
2010, 'name': 'Cardano', 'symbol': 'ADA', 'slug': 'cardano', 'num_market_pairs': 36
4, 'date_added': '2017-10-01T00:00:00.000Z', 'tags': ['mineable', 'dpos', 'pos', 'pl
atform', 'research', 'smart-contracts', 'staking', 'binance-smart-chain', 'cardano-e
cosystem', 'cardano'], 'max_supply': 45000000000, 'circulating_supply': 33613420243.
99, 'total_supply': 34105094650.087, 'is_active': 1, 'platform': None, 'cmc_rank':
7, 'is_fiat': 0, 'self_reported_circulating_supply': None, 'self_reported_market_ca
p': None, 'last_updated': '2022-02-16T01:48:00.000Z', 'quote': {'USD': {'price': 1.0
990844571343013, 'volume_24h': 1308453639.4361248, 'volume_change_24h': 9.3532, 'per
cent_change_1h': -0.52848559, 'percent_change_24h': 2.34419718, 'percent_change_7d':
-6.82780106, 'percent_change_30d': -21.00901256, 'percent_change_60d': -9.37611845,
'percent_change_90d': -41.95378512, 'market_cap': 36943987741.292885, 'market_cap_do
minance': 1.8588, 'fully_diluted_market_cap': 49458800571.04, 'last_updated': '2022-
02-16T01:48:00.000Z'}}}}}
{'status_timestamp': '2022-02-16T01:49:28.198Z', 'status_error_code': 0, 'status_err
or_message': None, 'status_elapsed': 35, 'status_credit_count': 1, 'status_notice':
None, 'data_ADA_id': 2010, 'data_ADA_name': 'Cardano', 'data_ADA_symbol': 'ADA', 'da
ta_ADA_slug': 'cardano', 'data_ADA_num_market_pairs': 364, 'data_ADA_date_added': '2
017-10-01T00:00:00.000Z', 'data_ADA_tags': ['mineable', 'dpos', 'pos', 'platform',
'research', 'smart-contracts', 'staking', 'binance-smart-chain', 'cardano-ecosyste
m', 'cardano'], 'data_ADA_max_supply': 45000000000, 'data_ADA_circulating_supply': 3
3613420243.99, 'data_ADA_total_supply': 34105094650.087, 'data_ADA_is_active': 1, 'd
ata_ADA_platform': None, 'data_ADA_cmc_rank': 7, 'data_ADA_is_fiat': 0, 'data_ADA_se
lf_reported_circulating_supply': None, 'data_ADA_self_reported_market_cap': None, 'd
ata_ADA_last_updated': '2022-02-16T01:48:00.000Z', 'data_ADA_quote_USD_price': 1.099
0844571343013, 'data_ADA_quote_USD_volume_24h': 1308453639.4361248, 'data_ADA_quote_
USD_volume_change_24h': 9.3532, 'data_ADA_quote_USD_percent_change_1h': -0.52848559,
'data_ADA_quote_USD_percent_change_24h': 2.34419718, 'data_ADA_quote_USD_percent_cha
nge_7d': -6.82780106, 'data_ADA_quote_USD_percent_change_30d': -21.00901256, 'data_A
DA_quote_USD_percent_change_60d': -9.37611845, 'data_ADA_quote_USD_percent_change_90
d': -41.95378512, 'data_ADA_quote_USD_market_cap': 36943987741.292885, 'data_ADA_quo
te_USD_market_cap_dominance': 1.8588, 'data_ADA_quote_USD_fully_diluted_market_cap':
49458800571.04, 'data_ADA_quote_USD_last_updated': '2022-02-16T01:48:00.000Z'}
The shape of dataframe: (10, 36)

Printing the details for the cryptocurrency: BCH
{'status': {'timestamp': '2022-02-16T01:49:28.395Z', 'error_code': 0, 'error_messag
e': None, 'elapsed': 29, 'credit_count': 1, 'notice': None}, 'data': {'BCH': {'id':
1831, 'name': 'Bitcoin Cash', 'symbol': 'BCH', 'slug': 'bitcoin-cash', 'num_market_p
airs': 568, 'date_added': '2017-07-23T00:00:00.000Z', 'tags': ['mineable', 'pow', 's
ha-256', 'marketplace', 'medium-of-exchange', 'store-of-value', 'enterprise-solution
s', 'payments', 'binance-chain', 'binance-smart-chain'], 'max_supply': 21000000, 'ci
rculating_supply': 18984206.25, 'total_supply': 18984206.25, 'is_active': 1, 'platfo
rm': None, 'cmc_rank': 28, 'is_fiat': 0, 'self_reported_circulating_supply': None,
'self_reported_market_cap': None, 'last_updated': '2022-02-16T01:48:00.000Z', 'quot
e': {'USD': {'price': 337.56086978310043, 'volume_24h': 4584883696.589585, 'volume_c
hange_24h': -1.5602, 'percent_change_1h': -1.10111977, 'percent_change_24h': 0.13520
53, 'percent_change_7d': 0.9207836, 'percent_change_30d': -12.37070867, 'percent_cha
nge_60d': -20.15238398, 'percent_change_90d': -43.6830922, 'market_cap': 6408325173.
891771, 'market_cap_dominance': 0.3225, 'fully_diluted_market_cap': 7088778265.45,
'last_updated': '2022-02-16T01:48:00.000Z'}}}}}
```

{'status_timestamp': '2022-02-16T01:49:28.395Z', 'status_error_code': 0, 'status_err
or_message': None, 'status_elapsed': 29, 'status_credit_count': 1, 'status_notice':
None, 'data_BCH_id': 1831, 'data_BCH_name': 'Bitcoin Cash', 'data_BCH_symbol': 'BC
H', 'data_BCH_slug': 'bitcoin-cash', 'data_BCH_num_market_pairs': 568, 'data_BCH_dat
e_added': '2017-07-23T00:00:00.000Z', 'data_BCH_tags': ['mineable', 'pow', 'sha-25
6', 'marketplace', 'medium-of-exchange', 'store-of-value', 'enterprise-solutions',
'payments', 'binance-chain', 'binance-smart-chain'], 'data_BCH_max_supply': 2100000
0, 'data_BCH_circulating_supply': 18984206.25, 'data_BCH_total_supply': 18984206.25,
'data_BCH_is_active': 1, 'data_BCH_platform': None, 'data_BCH_cmc_rank': 28, 'data_B
CH_is_fiat': 0, 'data_BCH_self_reported_circulating_supply': None, 'data_BCH_self_re
ported_market_cap': None, 'data_BCH_last_updated': '2022-02-16T01:48:00.000Z', 'data
_BCH_quote_USD_price': 337.56086978310043, 'data_BCH_quote_USD_volume_24h': 45848836
96.589585, 'data_BCH_quote_USD_volume_change_24h': -1.5602, 'data_BCH_quote_USD_perc
ent_change_1h': -1.10111977, 'data_BCH_quote_USD_percent_change_24h': 0.1352053, 'da
ta_BCH_quote_USD_percent_change_7d': 0.9207836, 'data_BCH_quote_USD_percent_change_3
0d': -12.37070867, 'data_BCH_quote_USD_percent_change_60d': -20.15238398, 'data_BCH_
quote_USD_percent_change_90d': -43.6830922, 'data_BCH_quote_USD_market_cap': 6408325
173.891771, 'data_BCH_quote_USD_market_cap_dominance': 0.3225, 'data_BCH_quote_USD_f
ully_diluted_market_cap': 7088778265.45, 'data_BCH_quote_USD_last_updated': '2022-02
-16T01:48:00.000Z'}
The shape of dataframe: (10, 36)

Printing the details for the cryptocurrency: DOGE
{'status': {'timestamp': '2022-02-16T01:49:28.512Z', 'error_code': 0, 'error_messag
e': None, 'elapsed': 30, 'credit_count': 1, 'notice': None}, 'data': {'DOGE': {'id':
74, 'name': 'Dogecoin', 'symbol': 'DOGE', 'slug': 'dogecoin', 'num_market_pairs': 44
6, 'date_added': '2013-12-15T00:00:00.000Z', 'tags': ['mineable', 'pow', 'scrypt',
'medium-of-exchange', 'memes', 'payments', 'binance-smart-chain', 'doggone-doggere
l'], 'max_supply': None, 'circulating_supply': 132670764299.89409, 'total_supply': 1
32670764299.89409, 'is_active': 1, 'platform': None, 'cmc_rank': 11, 'is_fiat': 0,
'self_reported_circulating_supply': None, 'self_reported_market_cap': None, 'last_up
dated': '2022-02-16T01:48:00.000Z', 'quote': {'USD': {'price': 0.15019767742454243,
'volume_24h': 653018671.2891237, 'volume_change_24h': -15.7434, 'percent_change_1h':
-0.31295699, 'percent_change_24h': 1.05521227, 'percent_change_7d': -4.81858098, 'pe
rcent_change_30d': -14.67349055, 'percent_change_60d': -10.49105492, 'percent_change
_90d': -37.41847075, 'market_cap': 19926840659.982994, 'market_cap_dominance': 1.002
2, 'fully_diluted_market_cap': 19926840659.98, 'last_updated': '2022-02-16T01:48:00.
000Z'}}}}}
{'status_timestamp': '2022-02-16T01:49:28.512Z', 'status_error_code': 0, 'status_err
or_message': None, 'status_elapsed': 30, 'status_credit_count': 1, 'status_notice':
None, 'data_DOGE_id': 74, 'data_DOGE_name': 'Dogecoin', 'data_DOGE_symbol': 'DOGE',
'data_DOGE_slug': 'dogecoin', 'data_DOGE_num_market_pairs': 446, 'data_DOGE_date_add
ed': '2013-12-15T00:00:00.000Z', 'data_DOGE_tags': ['mineable', 'pow', 'scrypt', 'me
dium-of-exchange', 'memes', 'payments', 'binance-smart-chain', 'doggone-doggerel'],
'data_DOGE_max_supply': None, 'data_DOGE_circulating_supply': 132670764299.89409, 'd
ata_DOGE_total_supply': 132670764299.89409, 'data_DOGE_is_active': 1, 'data_DOGE_pla
tform': None, 'data_DOGE_cmc_rank': 11, 'data_DOGE_is_fiat': 0, 'data_DOGE_self_repo
rted_circulating_supply': None, 'data_DOGE_self_reported_market_cap': None, 'data_DO
GE_last_updated': '2022-02-16T01:48:00.000Z', 'data_DOGE_quote_USD_price': 0.1501976
7742454243, 'data_DOGE_quote_USD_volume_24h': 653018671.2891237, 'data_DOGE_quote_US
D_volume_change_24h': -15.7434, 'data_DOGE_quote_USD_percent_change_1h': -0.3129569
9, 'data_DOGE_quote_USD_percent_change_24h': 1.05521227, 'data_DOGE_quote_USD_percen
t_change_7d': -4.81858098, 'data_DOGE_quote_USD_percent_change_30d': -14.67349055,
'data_DOGE_quote_USD_percent_change_60d': -10.49105492, 'data_DOGE_quote_USD_percent
_change_90d': -37.41847075, 'data_DOGE_quote_USD_market_cap': 19926840659.982994, 'd
ata_DOGE_quote_USD_market_cap_dominance': 1.0022, 'data_DOGE_quote_USD_fully_diluted
_market_cap': 19926840659.98, 'data_DOGE_quote_USD_last_updated': '2022-02-16T01:48:
00.000Z'}
The shape of dataframe: (8, 36)

Printing the details for the cryptocurrency: BTC
{'status': {'timestamp': '2022-02-16T01:49:28.636Z', 'error_code': 0, 'error_messag
e': None, 'elapsed': 33, 'credit_count': 1, 'notice': None}, 'data': {'BTC': {'id':
1, 'name': 'Bitcoin', 'symbol': 'BTC', 'slug': 'bitcoin', 'num_market_pairs': 9152,

'date_added': '2013-04-28T00:00:00.000Z', 'tags': ['mineable', 'pow', 'sha-256', 'st
ore-of-value', 'state-channel', 'coinbase-ventures-portfolio', 'three-arrows-capital
-portfolio', 'polychain-capital-portfolio', 'binance-labs-portfolio', 'blockchain-ca
pital-portfolio', 'boostvc-portfolio', 'cms-holdings-portfolio', 'dcg-portfolio', 'd
ragonfly-capital-portfolio', 'electric-capital-portfolio', 'fabric-ventures-portfoli
o', 'framework-ventures-portfolio', 'galaxy-digital-portfolio', 'huobi-capital-portf
olio', 'alameda-research-portfolio', 'a16z-portfolio', '1confirmation-portfolio', 'w
inklevoss-capital-portfolio', 'usv-portfolio', 'placeholder-ventures-portfolio', 'pa
ntera-capital-portfolio', 'multicoin-capital-portfolio', 'paradigm-portfolio'], 'max
_supply': 21000000, 'circulating_supply': 18959431, 'total_supply': 18959431, 'is_ac
tive': 1, 'platform': None, 'cmc_rank': 1, 'is_fiat': 0, 'self_reported_circulating_
supply': None, 'self_reported_market_cap': None, 'last_updated': '2022-02-16T01:48:0
0.000Z', 'quote': {'USD': {'price': 44035.09941170733, 'volume_24h': 22241330229.810
623, 'volume_change_24h': 2.5446, 'percent_change_1h': -0.44794852, 'percent_change_
24h': 1.35751896, 'percent_change_7d': -0.05794646, 'percent_change_30d': 2.6474810
2, 'percent_change_60d': -4.06719124, 'percent_change_90d': -27.3678098, 'market_ca
p': 834880428874.4056, 'market_cap_dominance': 42.0145, 'fully_diluted_market_cap':
924737087645.85, 'last_updated': '2022-02-16T01:48:00.000Z'}}}}}
{'status_timestamp': '2022-02-16T01:49:28.636Z', 'status_error_code': 0, 'status_err
or_message': None, 'status_elapsed': 33, 'status_credit_count': 1, 'status_notice':
None, 'data_BTC_id': 1, 'data_BTC_name': 'Bitcoin', 'data_BTC_symbol': 'BTC', 'data_
BTC_slug': 'bitcoin', 'data_BTC_num_market_pairs': 9152, 'data_BTC_date_added': '201
3-04-28T00:00:00.000Z', 'data_BTC_tags': ['mineable', 'pow', 'sha-256', 'store-of-va
lue', 'state-channel', 'coinbase-ventures-portfolio', 'three-arrows-capital-portfoli
o', 'polychain-capital-portfolio', 'binance-labs-portfolio', 'blockchain-capital-por
tfolio', 'boostvc-portfolio', 'cms-holdings-portfolio', 'dcg-portfolio', 'dragonfly-
capital-portfolio', 'electric-capital-portfolio', 'fabric-ventures-portfolio', 'fram
ework-ventures-portfolio', 'galaxy-digital-portfolio', 'huobi-capital-portfolio', 'a
lameda-research-portfolio', 'a16z-portfolio', '1confirmation-portfolio', 'winklevoss
-capital-portfolio', 'usv-portfolio', 'placeholder-ventures-portfolio', 'pantera-cap
ital-portfolio', 'multicoin-capital-portfolio', 'paradigm-portfolio'], 'data_BTC_max
_supply': 21000000, 'data_BTC_circulating_supply': 18959431, 'data_BTC_total_suppl
y': 18959431, 'data_BTC_is_active': 1, 'data_BTC_platform': None, 'data_BTC_cmc_ran
k': 1, 'data_BTC_is_fiat': 0, 'data_BTC_self_reported_circulating_supply': None, 'da
ta_BTC_self_reported_market_cap': None, 'data_BTC_last_updated': '2022-02-16T01:48:0
0.000Z', 'data_BTC_quote_USD_price': 44035.09941170733, 'data_BTC_quote_USD_volume_2
4h': 22241330229.810623, 'data_BTC_quote_USD_volume_change_24h': 2.5446, 'data_BTC_q
uote_USD_percent_change_1h': -0.44794852, 'data_BTC_quote_USD_percent_change_24h':
1.35751896, 'data_BTC_quote_USD_percent_change_7d': -0.05794646, 'data_BTC_quote_USD
_percent_change_30d': 2.64748102, 'data_BTC_quote_USD_percent_change_60d': -4.067191
24, 'data_BTC_quote_USD_percent_change_90d': -27.3678098, 'data_BTC_quote_USD_market
_cap': 834880428874.4056, 'data_BTC_quote_USD_market_cap_dominance': 42.0145, 'data_
BTC_quote_USD_fully_diluted_market_cap': 924737087645.85, 'data_BTC_quote_USD_last_u
pdated': '2022-02-16T01:48:00.000Z'}
The shape of dataframe: (28, 36)

Printing the details for the cryptocurrency: ETC
{'status': {'timestamp': '2022-02-16T01:49:28.847Z', 'error_code': 0, 'error_messag
e': None, 'elapsed': 32, 'credit_count': 1, 'notice': None}, 'data': {'ETC': {'id':
1321, 'name': 'Ethereum Classic', 'symbol': 'ETC', 'slug': 'ethereum-classic', 'num_
market_pairs': 286, 'date_added': '2016-07-24T00:00:00.000Z', 'tags': ['mineable',
'pow', 'ethash', 'platform', 'smart-contracts', 'binance-smart-chain', 'dcg-portfoli
o'], 'max_supply': 210700000, 'circulating_supply': 132996149.32173578, 'total_suppl
y': 210700000, 'is_active': 1, 'platform': None, 'cmc_rank': 37, 'is_fiat': 0, 'self
_reported_circulating_supply': None, 'self_reported_market_cap': None, 'last_update
d': '2022-02-16T01:48:00.000Z', 'quote': {'USD': {'price': 32.829660868750615, 'volu
me_24h': 525840993.66742784, 'volume_change_24h': -6.1383, 'percent_change_1h': -1.2
0876973, 'percent_change_24h': 1.98689495, 'percent_change_7d': 1.69659537, 'percent
_change_30d': 2.82919274, 'percent_change_60d': -3.94587318, 'percent_change_90d': -
36.29406949, 'market_cap': 4366218479.082303, 'market_cap_dominance': 0.2197, 'fully
_diluted_market_cap': 6917209545.05, 'last_updated': '2022-02-16T01:48:00.000Z'}}}}}
{'status_timestamp': '2022-02-16T01:49:28.847Z', 'status_error_code': 0, 'status_err
or_message': None, 'status_elapsed': 32, 'status_credit_count': 1, 'status_notice':
None, 'data_ETC_id': 1321, 'data_ETC_name': 'Ethereum Classic', 'data_ETC_symbol':

'ETC', 'data_ETC_slug': 'ethereum-classic', 'data_ETC_num_market_pairs': 286, 'data_ETC_date_added': '2016-07-24T00:00:00.000Z', 'data_ETC_tags': ['mineable', 'pow', 'ethash', 'platform', 'smart-contracts', 'binance-smart-chain', 'dcg-portfolio'], 'data_ETC_max_supply': 210700000, 'data_ETC_circulating_supply': 132996149.32173578, 'data_ETC_total_supply': 210700000, 'data_ETC_is_active': 1, 'data_ETC_platform': None, 'data_ETC_cmc_rank': 37, 'data_ETC_is_fiat': 0, 'data_ETC_self_reported_circulating_supply': None, 'data_ETC_self_reported_market_cap': None, 'data_ETC_last_updated': '2022-02-16T01:48:00.000Z', 'data_ETC_quote_USD_price': 32.829660868750615, 'data_ETC_quote_USD_volume_24h': 525840993.66742784, 'data_ETC_quote_USD_volume_change_24h': -6.1383, 'data_ETC_quote_USD_percent_change_1h': -1.20876973, 'data_ETC_quote_USD_percent_change_24h': 1.98689495, 'data_ETC_quote_USD_percent_change_7d': 1.69659537, 'data_ETC_quote_USD_percent_change_30d': 2.82919274, 'data_ETC_quote_USD_percent_change_60d': -3.94587318, 'data_ETC_quote_USD_percent_change_90d': -36.29406949, 'data_ETC_quote_USD_market_cap': 4366218479.082303, 'data_ETC_quote_USD_market_cap_dominance': 0.2197, 'data_ETC_quote_USD_fully_diluted_market_cap': 6917209545.05, 'data_ETC_quote_USD_last_updated': '2022-02-16T01:48:00.000Z'}
The shape of dataframe: (7, 36)

Printing the details for the cryptocurrency: ETH
{'status': {'timestamp': '2022-02-16T01:49:29.005Z', 'error_code': 0, 'error_message': None, 'elapsed': 36, 'credit_count': 1, 'notice': None}, 'data': {'ETH': {'id': 1027, 'name': 'Ethereum', 'symbol': 'ETH', 'slug': 'ethereum', 'num_market_pairs': 5502, 'date_added': '2015-08-07T00:00:00.000Z', 'tags': ['mineable', 'pow', 'smart-contracts', 'ethereum-ecosystem', 'binance-smart-chain', 'coinbase-ventures-portfolio', 'three-arrows-capital-portfolio', 'polychain-capital-portfolio', 'binance-labs-portfolio', 'blockchain-capital-portfolio', 'boostvc-portfolio', 'cms-holdings-portfolio', 'dcg-portfolio', 'dragonfly-capital-portfolio', 'electric-capital-portfolio', 'fabric-ventures-portfolio', 'framework-ventures-portfolio', 'hashkey-capital-portfolio', 'kenetic-capital-portfolio', 'huobi-capital-portfolio', 'alameda-research-portfolio', 'a16z-portfolio', '1confirmation-portfolio', 'winklevoss-capital-portfolio', 'usv-portfolio', 'placeholder-ventures-portfolio', 'pantera-capital-portfolio', 'multicoin-capital-portfolio', 'paradigm-portfolio', 'injective-ecosystem'], 'max_supply': None, 'circulating_supply': 119597059.0615, 'total_supply': 119597059.0615, 'is_active': 1, 'platform': None, 'cmc_rank': 2, 'is_fiat': 0, 'self_reported_circulating_supply': None, 'self_reported_market_cap': None, 'last_updated': '2022-02-16T01:48:00.000Z', 'quote': {'USD': {'price': 3144.985914094705, 'volume_24h': 13839701953.315847, 'volume_change_24h': 7.9237, 'percent_change_1h': -0.50892241, 'percent_change_24h': 4.51655051, 'percent_change_7d': 0.98779345, 'percent_change_30d': -5.29565829, 'percent_change_60d': -17.78466341, 'percent_change_90d': -27.17887608, 'market_cap': 376131066115.57, 'market_cap_dominance': 18.9249, 'fully_diluted_market_cap': 376131066115.57, 'last_updated': '2022-02-16T01:48:00.000Z'}}}}}
{'status_timestamp': '2022-02-16T01:49:29.005Z', 'status_error_code': 0, 'status_error_message': None, 'status_elapsed': 36, 'status_credit_count': 1, 'status_notice': None, 'data_ETH_id': 1027, 'data_ETH_name': 'Ethereum', 'data_ETH_symbol': 'ETH', 'data_ETH_slug': 'ethereum', 'data_ETH_num_market_pairs': 5502, 'data_ETH_date_added': '2015-08-07T00:00:00.000Z', 'data_ETH_tags': ['mineable', 'pow', 'smart-contracts', 'ethereum-ecosystem', 'binance-smart-chain', 'coinbase-ventures-portfolio', 'three-arrows-capital-portfolio', 'polychain-capital-portfolio', 'binance-labs-portfolio', 'blockchain-capital-portfolio', 'boostvc-portfolio', 'cms-holdings-portfolio', 'dcg-portfolio', 'dragonfly-capital-portfolio', 'electric-capital-portfolio', 'fabric-ventures-portfolio', 'framework-ventures-portfolio', 'hashkey-capital-portfolio', 'kenetic-capital-portfolio', 'huobi-capital-portfolio', 'alameda-research-portfolio', 'a16z-portfolio', '1confirmation-portfolio', 'winklevoss-capital-portfolio', 'usv-portfolio', 'placeholder-ventures-portfolio', 'pantera-capital-portfolio', 'multicoin-capital-portfolio', 'paradigm-portfolio', 'injective-ecosystem'], 'data_ETH_max_supply': None, 'data_ETH_circulating_supply': 119597059.0615, 'data_ETH_total_supply': 119597059.0615, 'data_ETH_is_active': 1, 'data_ETH_platform': None, 'data_ETH_cmc_rank': 2, 'data_ETH_is_fiat': 0, 'data_ETH_self_reported_circulating_supply': None, 'data_ETH_self_reported_market_cap': None, 'data_ETH_last_updated': '2022-02-16T01:48:00.000Z', 'data_ETH_quote_USD_price': 3144.985914094705, 'data_ETH_quote_USD_volume_24h': 13839701953.315847, 'data_ETH_quote_USD_volume_change_24h': 7.9237, 'data_ETH_quote_USD_percent_change_1h': -0.50892241, 'data_ETH_quote_USD_percent_change_24h': 4.51655051, 'data_ETH_quote_USD_percent_change_7d': 0.98779345, 'data_ETH_quote_USD_percent_change_30d': -5.29565829, 'data_ETH_quote_USD_percent_change_60d': -17.78466

341, 'data_ETH_quote_USD_percent_change_90d': -27.17887608, 'data_ETH_quote_USD_mark
et_cap': 376131066115.57, 'data_ETH_quote_USD_market_cap_dominance': 18.9249, 'data_
ETH_quote_USD_fully_diluted_market_cap': 376131066115.57, 'data_ETH_quote_USD_last_u
pdated': '2022-02-16T01:48:00.000Z'}
The shape of dataframe: (30, 36)

Printing the details for the cryptocurrency: LTC
{'status': {'timestamp': '2022-02-16T01:49:29.144Z', 'error_code': 0, 'error_messag
e': None, 'elapsed': 30, 'credit_count': 1, 'notice': None}, 'data': {'LTC': {'id':
2, 'name': 'Litecoin', 'symbol': 'LTC', 'slug': 'litecoin', 'num_market_pairs': 733,
'date_added': '2013-04-28T00:00:00.000Z', 'tags': ['mineable', 'pow', 'scrypt', 'med
ium-of-exchange', 'binance-chain', 'binance-smart-chain'], 'max_supply': 84000000,
'circulating_supply': 69647781.8613374, 'total_supply': 84000000, 'is_active': 1, 'p
latform': None, 'cmc_rank': 20, 'is_fiat': 0, 'self_reported_circulating_supply': No
ne, 'self_reported_market_cap': None, 'last_updated': '2022-02-16T01:48:00.000Z', 'q
uote': {'USD': {'price': 129.77413264530236, 'volume_24h': 763697480.2316202, 'volum
e_change_24h': 5.844, 'percent_change_1h': -0.5312272, 'percent_change_24h': 1.62818
516, 'percent_change_7d': -3.47882758, 'percent_change_30d': -11.2997567, 'percent_c
hange_60d': -10.1241145, 'percent_change_90d': -43.49101459, 'market_cap': 903848048
1.724283, 'market_cap_dominance': 0.4546, 'fully_diluted_market_cap': 10901027142.2
1, 'last_updated': '2022-02-16T01:48:00.000Z'}}}}}
{'status_timestamp': '2022-02-16T01:49:29.144Z', 'status_error_code': 0, 'status_err
or_message': None, 'status_elapsed': 30, 'status_credit_count': 1, 'status_notice':
None, 'data_LTC_id': 2, 'data_LTC_name': 'Litecoin', 'data_LTC_symbol': 'LTC', 'data
_LTC_slug': 'litecoin', 'data_LTC_num_market_pairs': 733, 'data_LTC_date_added': '20
13-04-28T00:00:00.000Z', 'data_LTC_tags': ['mineable', 'pow', 'scrypt', 'medium-of-e
xchange', 'binance-chain', 'binance-smart-chain'], 'data_LTC_max_supply': 84000000,
'data_LTC_circulating_supply': 69647781.8613374, 'data_LTC_total_supply': 84000000,
'data_LTC_is_active': 1, 'data_LTC_platform': None, 'data_LTC_cmc_rank': 20, 'data_L
TC_is_fiat': 0, 'data_LTC_self_reported_circulating_supply': None, 'data_LTC_self_re
ported_market_cap': None, 'data_LTC_last_updated': '2022-02-16T01:48:00.000Z', 'data
_LTC_quote_USD_price': 129.77413264530236, 'data_LTC_quote_USD_volume_24h': 76369748
0.2316202, 'data_LTC_quote_USD_volume_change_24h': 5.844, 'data_LTC_quote_USD_percen
t_change_1h': -0.5312272, 'data_LTC_quote_USD_percent_change_24h': 1.62818516, 'data
_LTC_quote_USD_percent_change_7d': -3.47882758, 'data_LTC_quote_USD_percent_change_3
0d': -11.2997567, 'data_LTC_quote_USD_percent_change_60d': -10.1241145, 'data_LTC_qu
ote_USD_percent_change_90d': -43.49101459, 'data_LTC_quote_USD_market_cap': 90384804
81.724283, 'data_LTC_quote_USD_market_cap_dominance': 0.4546, 'data_LTC_quote_USD_fu
lly_diluted_market_cap': 10901027142.21, 'data_LTC_quote_USD_last_updated': '2022-02
-16T01:48:00.000Z'}
The shape of dataframe: (6, 36)

Printing the details for the cryptocurrency: NEO
{'status': {'timestamp': '2022-02-16T01:49:29.326Z', 'error_code': 0, 'error_messag
e': None, 'elapsed': 33, 'credit_count': 1, 'notice': None}, 'data': {'NEO': {'id':
1376, 'name': 'Neo', 'symbol': 'NEO', 'slug': 'neo', 'num_market_pairs': 239, 'date_
added': '2016-09-08T00:00:00.000Z', 'tags': ['platform', 'enterprise-solutions', 'sm
art-contracts'], 'max_supply': 100000000, 'circulating_supply': 70538831, 'total_sup
ply': 100000000, 'is_active': 1, 'platform': None, 'cmc_rank': 63, 'is_fiat': 0, 'se
lf_reported_circulating_supply': None, 'self_reported_market_cap': None, 'last_updat
ed': '2022-02-16T01:48:00.000Z', 'quote': {'USD': {'price': 22.511006488075502, 'vol
ume_24h': 92461108.66798992, 'volume_change_24h': 8.9861, 'percent_change_1h': -0.95
333309, 'percent_change_24h': 3.07403896, 'percent_change_7d': -2.47597606, 'percent
_change_30d': -9.28962506, 'percent_change_60d': -10.66846433, 'percent_change_90d':
-48.22616783, 'market_cap': 1587900082.3022614, 'market_cap_dominance': 0.0799, 'ful
ly_diluted_market_cap': 2251100648.81, 'last_updated': '2022-02-16T01:48:00.000
Z'}}}}}
{'status_timestamp': '2022-02-16T01:49:29.326Z', 'status_error_code': 0, 'status_err
or_message': None, 'status_elapsed': 33, 'status_credit_count': 1, 'status_notice':
None, 'data_NEO_id': 1376, 'data_NEO_name': 'Neo', 'data_NEO_symbol': 'NEO', 'data_N
EO_slug': 'neo', 'data_NEO_num_market_pairs': 239, 'data_NEO_date_added': '2016-09-0
8T00:00:00.000Z', 'data_NEO_tags': ['platform', 'enterprise-solutions', 'smart-contr
acts'], 'data_NEO_max_supply': 100000000, 'data_NEO_circulating_supply': 70538831,
'data_NEO_total_supply': 100000000, 'data_NEO_is_active': 1, 'data_NEO_platform': No

```
ne, 'data_NEO_cmc_rank': 63, 'data_NEO_is_fiat': 0, 'data_NEO_self_reported_circulat
ing_supply': None, 'data_NEO_self_reported_market_cap': None, 'data_NEO_last_update
d': '2022-02-16T01:48:00.000Z', 'data_NEO_quote_USD_price': 22.511006488075502, 'dat
a_NEO_quote_USD_volume_24h': 92461108.66798992, 'data_NEO_quote_USD_volume_change_24
h': 8.9861, 'data_NEO_quote_USD_percent_change_1h': -0.95333309, 'data_NEO_quote_USD
_percent_change_24h': 3.07403896, 'data_NEO_quote_USD_percent_change_7d': -2.4759760
6, 'data_NEO_quote_USD_percent_change_30d': -9.28962506, 'data_NEO_quote_USD_percent
_change_60d': -10.66846433, 'data_NEO_quote_USD_percent_change_90d': -48.22616783,
'data_NEO_quote_USD_market_cap': 1587900082.3022614, 'data_NEO_quote_USD_market_cap_
dominance': 0.0799, 'data_NEO_quote_USD_fully_diluted_market_cap': 2251100648.81, 'd
ata_NEO_quote_USD_last_updated': '2022-02-16T01:48:00.000Z'}
The shape of dataframe: (3, 36)
```

In [123…

```
# Display some sample records from the dataframe

df.head()
```

Out[123…

	status_timestamp	status_error_code	status_error_message	status_elapsed	status_credit_count	sta
0	2022-02-16T01:48:16.954Z	0	None	27	1	
1	2022-02-16T01:48:16.954Z	0	None	27	1	
2	2022-02-16T01:48:16.954Z	0	None	27	1	
3	2022-02-16T01:48:16.954Z	0	None	27	1	
4	2022-02-16T01:48:16.954Z	0	None	27	1	

5 rows × 36 columns

In [125…

```
# Calculate the number of rows and columns in the Final dataframe

print("Total number of rows and columns present in the  final dataframe: {}".format(
```

```
Total number of rows and columns present in the dataframe: (102, 36)
```

In [130…

```
# Display the attributes names present in the dataframe df

df.columns
```

Out[130…

```
Index(['status_timestamp', 'status_error_code', 'status_error_message',
       'status_elapsed', 'status_credit_count', 'status_notice', 'data_id',
       'data_name', 'data_symbol', 'data_slug', 'data_num_market_pairs',
       'data_date_added', 'data_tags', 'data_max_supply',
       'data_circulating_supply', 'data_total_supply', 'data_is_active',
       'data_platform', 'data_cmc_rank', 'data_is_fiat',
       'data_self_reported_circulating_supply',
       'data_self_reported_market_cap', 'data_last_updated',
       'data_quote_USD_price', 'data_quote_USD_volume_24h',
       'data_quote_USD_volume_change_24h', 'data_quote_USD_percent_change_1h',
       'data_quote_USD_percent_change_24h', 'data_quote_USD_percent_change_7d',
       'data_quote_USD_percent_change_30d',
       'data_quote_USD_percent_change_60d',
       'data_quote_USD_percent_change_90d', 'data_quote_USD_market_cap',
```

```
       'data_quote_USD_market_cap_dominance',
       'data_quote_USD_fully_diluted_market_cap',
       'data_quote_USD_last_updated'],
      dtype='object')
```

Tether (USDT) cryptocurrency has additional 4 columns as mentioned below compared to other cryptocurrencies taken into consideration. So, we have to format the dataset so that we can merge USDT with the dataframe created above

data_platform_id
data_platform_name
data_platform_symbol
data_platform_slug
data_platform_token_address

In [156…

```
# Step2: Adding additional dataset and formatting columns

# USDT has more number of columns compared to other digital currency and  Reading th

parameters['symbol'] = 'USDT'
print(parameters)

try:
    response = session.get(url, params=parameters)
    usdt_data = json.loads(response.text)
    print(usdt_data)
except (ConnectionError, Timeout, TooManyRedirects) as e:
        print(e)

flatten_usdt_data = flatten(usdt_data)
usdt_data_columns = flatten_usdt_data.keys()
print("Print the column names for USDT dataset: ")
print(usdt_data_columns)

usdt_column_list = []
for col in usdt_data_columns:
    if 'USDT' in col:
        col = col.replace('USDT_','')
    usdt_column_list.append(col)
print(usdt_column_list)

usdt_df = pd.DataFrame(flatten_usdt_data)
usdt_df.columns = usdt_column_list
```

```
{'symbol': 'USDT'}
{'status': {'timestamp': '2022-02-16T03:55:30.342Z', 'error_code': 0, 'error_messag
e': None, 'elapsed': 36, 'credit_count': 1, 'notice': None}, 'data': {'USDT': {'id':
825, 'name': 'Tether', 'symbol': 'USDT', 'slug': 'tether', 'num_market_pairs': 2798
7, 'date_added': '2015-02-25T00:00:00.000Z', 'tags': ['payments', 'stablecoin', 'ass
et-backed-stablecoin', 'binance-smart-chain', 'avalanche-ecosystem', 'solana-ecosyst
em', 'moonriver-ecosystem', 'injective-ecosystem'], 'max_supply': None, 'circulating
_supply': 78641051388.78949, 'total_supply': 81064697052.54242, 'platform': {'id': 1
027, 'name': 'Ethereum', 'symbol': 'ETH', 'slug': 'ethereum', 'token_address': '0xda
c17f958d2ee523a2206206994597c13d831ec7'}, 'is_active': 1, 'cmc_rank': 3, 'is_fiat':
0, 'self_reported_circulating_supply': None, 'self_reported_market_cap': None, 'last
_updated': '2022-02-16T03:54:00.000Z', 'quote': {'USD': {'price': 1.000576060314811,
'volume_24h': 51821747736.79891, 'volume_change_24h': 2.6403, 'percent_change_1h': -
0.00106502, 'percent_change_24h': 0.00648862, 'percent_change_7d': -0.01388727, 'per
cent_change_30d': 0.03025311, 'percent_change_60d': 0.01691945, 'percent_change_90
d': 0.00969067, 'market_cap': 78686353377.60959, 'market_cap_dominance': 3.9588, 'fu
lly_diluted_market_cap': 81111395207.45, 'last_updated': '2022-02-16T03:54:00.000
```

```
Z'}}}}}
Print the column names for USDT dataset:
dict_keys(['status_timestamp', 'status_error_code', 'status_error_message', 'status_
elapsed', 'status_credit_count', 'status_notice', 'data_USDT_id', 'data_USDT_name',
'data_USDT_symbol', 'data_USDT_slug', 'data_USDT_num_market_pairs', 'data_USDT_date_
added', 'data_USDT_tags', 'data_USDT_max_supply', 'data_USDT_circulating_supply', 'd
ata_USDT_total_supply', 'data_USDT_platform_id', 'data_USDT_platform_name', 'data_US
DT_platform_symbol', 'data_USDT_platform_slug', 'data_USDT_platform_token_address',
'data_USDT_is_active', 'data_USDT_cmc_rank', 'data_USDT_is_fiat', 'data_USDT_self_re
ported_circulating_supply', 'data_USDT_self_reported_market_cap', 'data_USDT_last_up
dated', 'data_USDT_quote_USD_price', 'data_USDT_quote_USD_volume_24h', 'data_USDT_qu
ote_USD_volume_change_24h', 'data_USDT_quote_USD_percent_change_1h', 'data_USDT_quot
e_USD_percent_change_24h', 'data_USDT_quote_USD_percent_change_7d', 'data_USDT_quote
_USD_percent_change_30d', 'data_USDT_quote_USD_percent_change_60d', 'data_USDT_quote
_USD_percent_change_90d', 'data_USDT_quote_USD_market_cap', 'data_USDT_quote_USD_mar
ket_cap_dominance', 'data_USDT_quote_USD_fully_diluted_market_cap', 'data_USDT_quote
_USD_last_updated'])
['status_timestamp', 'status_error_code', 'status_error_message', 'status_elapsed',
'status_credit_count', 'status_notice', 'data_id', 'data_name', 'data_symbol', 'data
_slug', 'data_num_market_pairs', 'data_date_added', 'data_tags', 'data_max_supply',
'data_circulating_supply', 'data_total_supply', 'data_platform_id', 'data_platform_n
ame', 'data_platform_symbol', 'data_platform_slug', 'data_platform_token_address',
'data_is_active', 'data_cmc_rank', 'data_is_fiat', 'data_self_reported_circulating_s
upply', 'data_self_reported_market_cap', 'data_last_updated', 'data_quote_USD_pric
e', 'data_quote_USD_volume_24h', 'data_quote_USD_volume_change_24h', 'data_quote_USD
_percent_change_1h', 'data_quote_USD_percent_change_24h', 'data_quote_USD_percent_ch
ange_7d', 'data_quote_USD_percent_change_30d', 'data_quote_USD_percent_change_60d',
'data_quote_USD_percent_change_90d', 'data_quote_USD_market_cap', 'data_quote_USD_ma
rket_cap_dominance', 'data_quote_USD_fully_diluted_market_cap', 'data_quote_USD_last
_updated']
```

In [157…

```
# UDST dataframe has been created

usdt_df.head()
```

Out[157…

	status_timestamp	status_error_code	status_error_message	status_elapsed	status_credit_count	sta
0	2022-02-16T03:55:30.342Z	0	None	36	1	
1	2022-02-16T03:55:30.342Z	0	None	36	1	
2	2022-02-16T03:55:30.342Z	0	None	36	1	
3	2022-02-16T03:55:30.342Z	0	None	36	1	
4	2022-02-16T03:55:30.342Z	0	None	36	1	

5 rows × 40 columns

In [158…

```
# Display the sample values and shape

print("The shape of the usdt dataframe: {}".format(usdt_df.shape))
```

```
The shape of the usdt dataframe: (8, 40)
```

In [159…

```
# Finding additional attributes  present in UDST dataset

usdt_extra_col = [col for col in usdt_column_list if col not in column_list]
usdt_extra_col.remove('data_platform_name')
print(usdt_extra_col)
```

```
['data_platform_id', 'data_platform_symbol', 'data_platform_slug', 'data_platform_to
ken_address']
```

In [160…

```
# Drop extra columns in the dataframe

usdt_df = usdt_df.drop(columns= usdt_extra_col)
```

In [162…

```
# Print or display the sample values of usdt

usdt_df.head()
```

Out[162…

	status_timestamp	status_error_code	status_error_message	status_elapsed	status_credit_count	sta
0	2022-02-16T03:55:30.342Z	0	None	36	1	
1	2022-02-16T03:55:30.342Z	0	None	36	1	
2	2022-02-16T03:55:30.342Z	0	None	36	1	
3	2022-02-16T03:55:30.342Z	0	None	36	1	
4	2022-02-16T03:55:30.342Z	0	None	36	1	

5 rows × 36 columns

In [163…

```
# Combining the usdt dataset with original dataframe and One of the column name is n

usdt_df.rename(columns={'data_platform_name': 'data_platform'}, inplace=True)
```

In [166…

```
#Display the count of rows and columns present in usdt dataframe

print("The number of rows and columns present in the usdt dataframe: {}".format(usdt
```

```
The number of rows and columns present in the usdt dataframe: (8, 36)
```

In [168…

```
# Combining the usdt dataset with original dataframe

crypto_raw_df = pd.concat([df, usdt_df], axis=0)
```

In [169…

```
# Display  the shape of final dataframe

print("The number of rows and columns present in the final dataframe: {}".format(cry
```

```
The number of rows and columns present in the final dataframe: (110, 36)
```

From the above result, we could see the total number of records in the final dataframe is 110 (102+8) which is correct as expected. The dataframe contains the data for below cryptocurrencies.
'ADA','BCH','DOGE','BTC','ETC','ETH','LTC','NEO' and 'USDT

In [170…

```
# Step 3: Handling null /missing values from the dataset

# Calculating number of null rows present in the dataset across all columns

print("Number of Null rows in the dataset: {}".format(crypto_raw_df.isnull().sum()))
```

```
Number of Null rows in the dataset: status_timestamp                         0
status_error_code                              0
status_error_message                         110
status_elapsed                                 0
status_credit_count                            0
status_notice                                110
data_id                                        0
data_name                                      0
data_symbol                                    0
data_slug                                      0
data_num_market_pairs                          0
data_date_added                                0
data_tags                                      0
data_max_supply                               46
data_circulating_supply                        0
data_total_supply                              0
data_is_active                                 0
data_platform                                102
data_cmc_rank                                  0
data_is_fiat                                   0
data_self_reported_circulating_supply        110
data_self_reported_market_cap                110
data_last_updated                              0
data_quote_USD_price                           0
data_quote_USD_volume_24h                      0
data_quote_USD_volume_change_24h               0
data_quote_USD_percent_change_1h               0
data_quote_USD_percent_change_24h              0
data_quote_USD_percent_change_7d               0
data_quote_USD_percent_change_30d              0
data_quote_USD_percent_change_60d              0
data_quote_USD_percent_change_90d              0
data_quote_USD_market_cap                      0
data_quote_USD_market_cap_dominance            0
data_quote_USD_fully_diluted_market_cap        0
data_quote_USD_last_updated                    0
dtype: int64
```

From the above result, we could see below columns or fields doesn't make any value to the dataset. Among these values, couple of fields are having null values for all the records. So, we can remove these columns from the dataframe

status_error_code

status_error_message

status_elapsed

status_credit_count

status_notice

In [172…
```
# Cleanse the unwanted columns from the dataframe

unwanted_cols = ['status_error_code','status_error_message','status_elapsed','status
digital_df = crypto_raw_df.drop(columns= unwanted_cols)
```

In [173…
```
# Display the dataset after removing null and unwated columns

digital_df.head()
```

Out[173…

	status_timestamp	data_id	data_name	data_symbol	data_slug	data_num_market_pairs	data_date
0	2022-02-16T01:49:28.198Z	2010	Cardano	ADA	cardano	364	2 01T00:00:
1	2022-02-16T01:49:28.198Z	2010	Cardano	ADA	cardano	364	2 01T00:00:
2	2022-02-16T01:49:28.198Z	2010	Cardano	ADA	cardano	364	2 01T00:00:
3	2022-02-16T01:49:28.198Z	2010	Cardano	ADA	cardano	364	2 01T00:00:
4	2022-02-16T01:49:28.198Z	2010	Cardano	ADA	cardano	364	2 01T00:00:

5 rows × 31 columns

In [175…
```
# Calculate the total number of rows and columns from the latest dataset

print("Total number of rows and columns present in the dataset: {}".format(digital_d
```

```
Total number of rows and columns present in the dataset: (110, 31)
```

In [176…
```
# Step 4: Handling  null rows from the dataset

# Calculating number of null rows present in the dataset

print("Number of Null rows in the dataset: {}".format(digital_df.data_symbol.isnull(
```

```
Number of Null rows in the dataset: 0
```

We have considered "data_symbol" field to check for null values. This is a mandatory field which should have values populated. On checking, we could see there is no null row present in the dataset.

In [177…
```
# Step 5: Replace Headers with more meaningful names

# Printing the column names present in the dataframe

digital_df.columns
```

Out[177…
```
Index(['status_timestamp', 'data_id', 'data_name', 'data_symbol', 'data_slug',
       'data_num_market_pairs', 'data_date_added', 'data_tags',
       'data_max_supply', 'data_circulating_supply', 'data_total_supply',
       'data_is_active', 'data_platform', 'data_cmc_rank', 'data_is_fiat',
       'data_self_reported_circulating_supply',
       'data_self_reported_market_cap', 'data_last_updated',
```

```
       'data_quote_USD_price', 'data_quote_USD_volume_24h',
       'data_quote_USD_volume_change_24h', 'data_quote_USD_percent_change_1h',
       'data_quote_USD_percent_change_24h', 'data_quote_USD_percent_change_7d',
       'data_quote_USD_percent_change_30d',
       'data_quote_USD_percent_change_60d',
       'data_quote_USD_percent_change_90d', 'data_quote_USD_market_cap',
       'data_quote_USD_market_cap_dominance',
       'data_quote_USD_fully_diluted_market_cap',
       'data_quote_USD_last_updated'],
      dtype='object')
```

In [180…

```
# Display few records using head command

digital_df.head()
```

Out[180…

	status_timestamp	data_id	data_name	data_symbol	data_slug	data_num_market_pairs	data_date
0	2022-02-16T01:49:28.198Z	2010	Cardano	ADA	cardano	364	2 01T00:00:
1	2022-02-16T01:49:28.198Z	2010	Cardano	ADA	cardano	364	2 01T00:00:
2	2022-02-16T01:49:28.198Z	2010	Cardano	ADA	cardano	364	2 01T00:00:
3	2022-02-16T01:49:28.198Z	2010	Cardano	ADA	cardano	364	2 01T00:00:
4	2022-02-16T01:49:28.198Z	2010	Cardano	ADA	cardano	364	2 01T00:00:

5 rows × 31 columns

In [181…

```
#Change or rename the attributs

digital_df.rename(columns={'data_id': 'crypto_id', 'data_name': 'crypto_name', 'data
```

In [182…

```
# Display the columns name after converting the column names

digital_df.head()
```

Out[182…

	status_timestamp	crypto_id	crypto_name	symbol	crypto_slug	num_market_pairs	found_da
0	2022-02-16T01:49:28.198Z	2010	Cardano	ADA	cardano	364	2017-' 01T00:00:00.00
1	2022-02-16T01:49:28.198Z	2010	Cardano	ADA	cardano	364	2017-' 01T00:00:00.00
2	2022-02-16T01:49:28.198Z	2010	Cardano	ADA	cardano	364	2017-' 01T00:00:00.00
3	2022-02-16T01:49:28.198Z	2010	Cardano	ADA	cardano	364	2017-' 01T00:00:00.00
4	2022-02-16T01:49:28.198Z	2010	Cardano	ADA	cardano	364	2017-' 01T00:00:00.00

5 rows × 31 columns

In [183…
```
# Displaying the column names after renaming

digital_df.columns
```

Out[183…
```
Index(['status_timestamp', 'crypto_id', 'crypto_name', 'symbol', 'crypto_slug',
       'num_market_pairs', 'found_date', 'tags', 'maximum_num_supply',
       'circulating_supply', 'total_num_supply', 'active_flag', 'platform',
       'crypto_cmc_rank', 'is_fiat', 'circulating_supply', 'market_cap',
       'last_update_dt', 'usd_price', 'usd_volume_24h',
       'usd_volume_change_24h', 'usd_percent_change_1h',
       'usd_percent_change_24h', 'usd_percent_change_7d',
       'usd_percent_change_30d', 'usd_percent_change_60d',
       'usd_percent_change_90d', 'usd_market_cap', 'usd_market_cap_dominance',
       'usd_fully_diluted_market_cap', 'usd_last_updated_tm'],
      dtype='object')
```

In [185…
```
# Step 6: Replace datatypes based on column values

# Display the datatypes of all the columns present in the dataframe

digital_df.dtypes
```

Out[185…
```
status_timestamp               object
crypto_id                       int64
crypto_name                    object
symbol                         object
crypto_slug                    object
num_market_pairs                int64
found_date                     object
tags                           object
maximum_num_supply             object
circulating_supply            float64
total_num_supply              float64
active_flag                     int64
platform                       object
crypto_cmc_rank                 int64
is_fiat                         int64
circulating_supply             object
market_cap                     object
last_update_dt                 object
usd_price                     float64
usd_volume_24h                float64
usd_volume_change_24h         float64
usd_percent_change_1h         float64
usd_percent_change_24h        float64
usd_percent_change_7d         float64
usd_percent_change_30d        float64
usd_percent_change_60d        float64
usd_percent_change_90d        float64
usd_market_cap                float64
usd_market_cap_dominance      float64
usd_fully_diluted_market_cap  float64
usd_last_updated_tm            object
dtype: object
```

From the above result, we could see float and int columns are populated with correct datatypes before. Now, we would want to change the date and timestamp columns

In [192…
```
# Change the attributes date and time values to datetime datatype.
```

```
digital_df['usd_last_updated_tm'] =  pd.to_datetime(digital_df['usd_last_updated_tm'
digital_df['status_timestamp'] =  pd.to_datetime(digital_df['status_timestamp'])
digital_df['last_update_dt'] =  pd.to_datetime(digital_df['last_update_dt'])
digital_df['found_date'] =  pd.to_datetime(digital_df['found_date'])
```

In [193…

```
# Display the datatypes after the conversion

digital_df.dtypes
```

Out[193…

```
status_timestamp                datetime64[ns, UTC]
crypto_id                                     int64
crypto_name                                  object
symbol                                       object
crypto_slug                                  object
num_market_pairs                              int64
found_date                      datetime64[ns, UTC]
tags                                         object
maximum_num_supply                           object
circulating_supply                          float64
total_num_supply                            float64
active_flag                                   int64
platform                                     object
crypto_cmc_rank                               int64
is_fiat                                       int64
circulating_supply                           object
market_cap                                   object
last_update_dt                  datetime64[ns, UTC]
usd_price                                   float64
usd_volume_24h                              float64
usd_volume_change_24h                       float64
usd_percent_change_1h                       float64
usd_percent_change_24h                      float64
usd_percent_change_7d                       float64
usd_percent_change_30d                      float64
usd_percent_change_60d                      float64
usd_percent_change_90d                      float64
usd_market_cap                              float64
usd_market_cap_dominance                    float64
usd_fully_diluted_market_cap                float64
usd_last_updated_tm             datetime64[ns, UTC]
dtype: object
```

In [194…

```
# Step 7: Find and remove duplicates from the dataset

#  Checking if there are any duplicate values within the dataframe, at a row level

duplicate_df = digital_df[digital_df.duplicated()]
```

In [195…

```
# Display the redundents to check if there is any duplicates present in the datafram

duplicate_df
```

Out[195…

status_timestamp	crypto_id	crypto_name	symbol	crypto_slug	num_market_pairs	found_date	ta

0 rows × 31 columns

In [196…
```
#  Calculate the size of dataframe after removing the duplicates

digital_df.shape
```

Out[196…
```
(110, 31)
```

We can conclued or could see that there is no duplicates present in the dataframe. The size remains same as before

In [200…
```
# Index resetting values of the dataframe

digital_df = digital_df.reset_index()
del digital_df['index']
```

In [202…
```
# Display  the records present in the dataframe

del digital_df['level_0']
digital_df
```

Out[202…

	status_timestamp	crypto_id	crypto_name	symbol	crypto_slug	num_market_pairs	four
0	2022-02-16 01:49:28.198000+00:00	2010	Cardano	ADA	cardano	364	201 00:00:0(
1	2022-02-16 01:49:28.198000+00:00	2010	Cardano	ADA	cardano	364	201 00:00:0(
2	2022-02-16 01:49:28.198000+00:00	2010	Cardano	ADA	cardano	364	201 00:00:0(
3	2022-02-16 01:49:28.198000+00:00	2010	Cardano	ADA	cardano	364	201 00:00:0(
4	2022-02-16 01:49:28.198000+00:00	2010	Cardano	ADA	cardano	364	201 00:00:0(
...	...	...	...	...	...	...	
105	2022-02-16 03:55:30.342000+00:00	825	Tether	USDT	tether	27987	201 00:00:0(
106	2022-02-16 03:55:30.342000+00:00	825	Tether	USDT	tether	27987	201 00:00:0(
107	2022-02-16 03:55:30.342000+00:00	825	Tether	USDT	tether	27987	201 00:00:0(
108	2022-02-16 03:55:30.342000+00:00	825	Tether	USDT	tether	27987	201 00:00:0(
109	2022-02-16 03:55:30.342000+00:00	825	Tether	USDT	tether	27987	201 00:00:0(

110 rows × 31 columns

From the data, we will create another dataframe with precentage fields

In [204…
```
# Step 8: Fuzzy matching and  Pictorial representation of percentage change

# Adding a subset of dataframe with usd percent change

digital_df_pic = digital_df.loc[:, ['symbol','usd_percent_change_1h', 'usd_percent_c
```

In [205…
```
# Cleance duplicates from the dataframe

digital_df_pic.drop_duplicates(inplace = True)
```

In [207…
```
#Print the total record count and sample records from the dataframe

print("Total number of rows and columns present in the dataframe: {}".format(digital
```

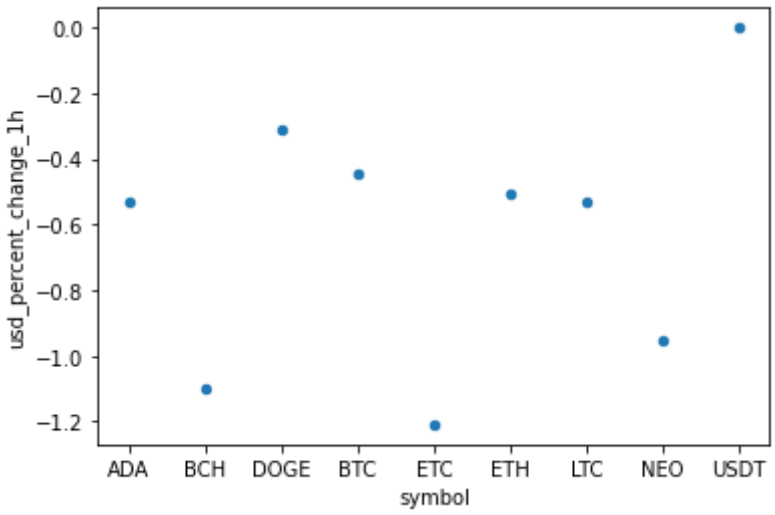
```
Total number of rows and columns present in the dataframe: (9, 7)
```

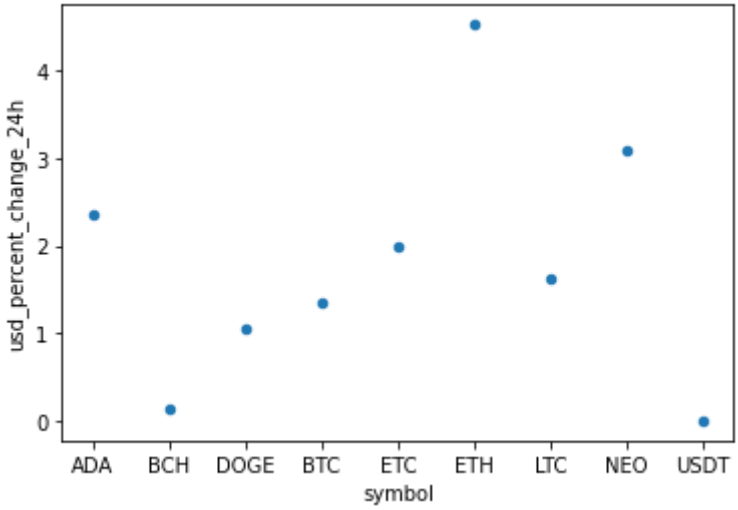
In [208…
```
digital_df_pic.head()
```

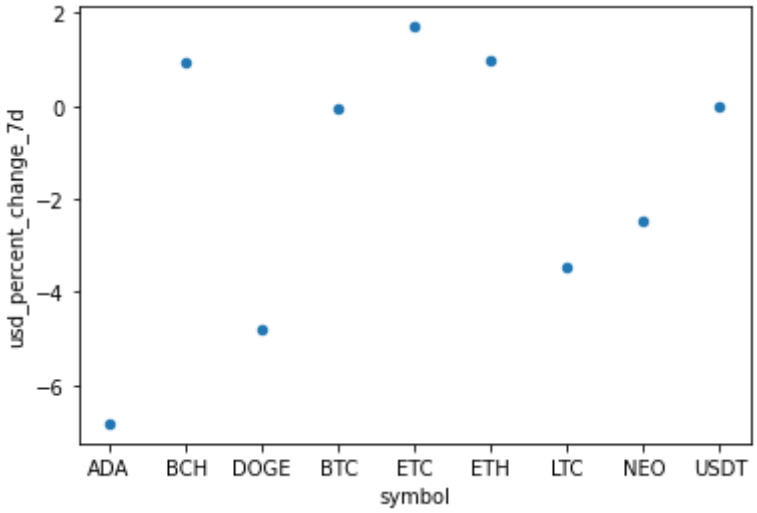
Out[208…

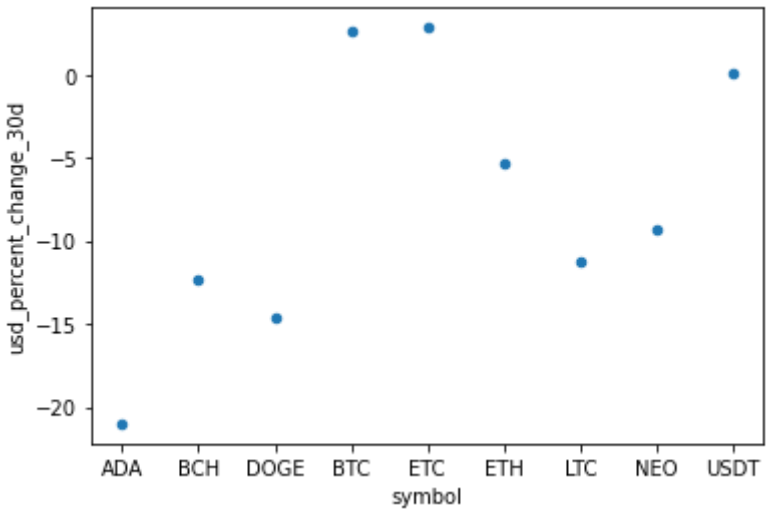
	symbol	usd_percent_change_1h	usd_percent_change_24h	usd_percent_change_7d	usd_percent_ch
0	ADA	-0.528486	2.344197	-6.827801	-
10	BCH	-1.101120	0.135205	0.920784	-
20	DOGE	-0.312957	1.055212	-4.818581	-
28	BTC	-0.447949	1.357519	-0.057946	
56	ETC	-1.208770	1.986895	1.696595	

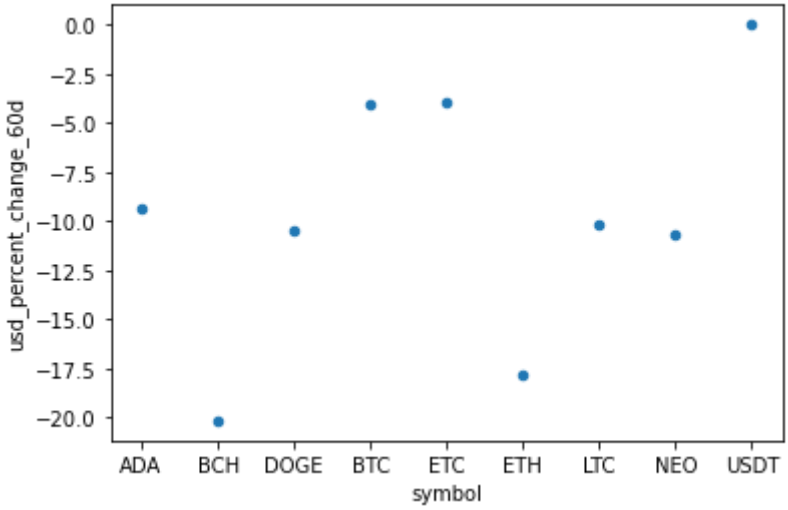
In [213…
```
## Plotting scatter plot for all the above columns taken into consideration

for col in list(digital_df_pic.columns):
    if col != 'symbol':
        digital_df_pic.plot.scatter(x = 'symbol', y = col)
```

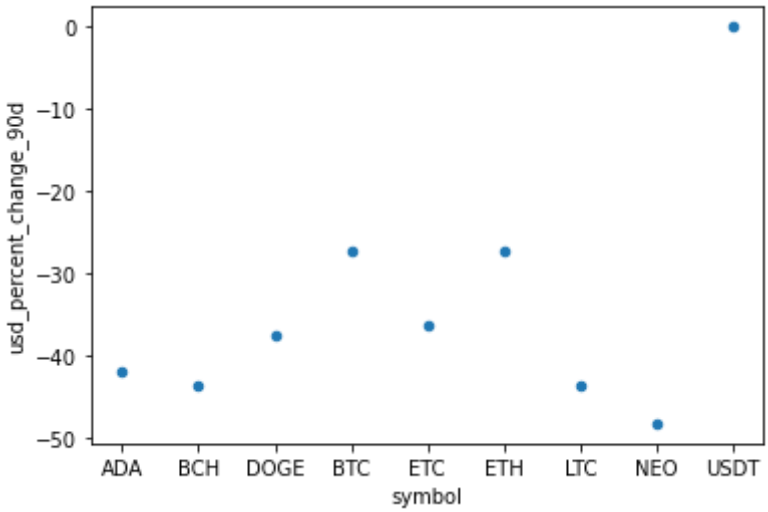



usd_percent_change_24h
4
3
2
1
0
ADA
BCH
DOGE
BTC
ETC
ETH
LTC
NEO
USDT
symbol


usd_percent_change_7d
2
0
−2
−4
−6
ADA
BCH
DOGE
BTC
ETC
ETH
LTC
NEO
USDT
symbol


usd_percent_change_30d
0
−5
−10
−15
−20
ADA
BCH
DOGE
BTC
ETC
ETH
LTC
NEO
USDT
symbol





Final finding from the graph shows that we can see the percentage change value across the days is very high compared to stocks and other investment.

In [214…

```
# Step 9: Saving the final output to CSV file

# Store the final dataframe to csv file which can be used to make join with other da

digital_df.to_csv("Crypto_Currencies_api_dataset.csv", index=False)
```

In [ ]: