SI 206 Final Report

Github link: https://github.com/hwarif/SI206-Final-Project.git

Our Team

Nick Sidor

School of Information

UX Research

Alaa Shahin

School of Information

UX Designer

Ari Feldberg

School of Information

Information Analysis

Goals

Goals

1. Use at least 2 API's and/or 1 website
2. Create at least 3 visualizations
3. Write a calculation using the collected data and write the result to a text file.
4. At least 1 API must have 2 tables that share a key.
5. Compare cryptocurrency rankings to amount of Twitter followers.

Goals Achieved

Goals Achieved

1. Used 3 API's (Coingecko API, Tweepy API, and General Mills API for Google Trends).
2. Created 3 Visualizations (dual trendline, dual bar graphs and a stacked bar graph).
3. Wrote a calculation using the collected data and write the result to a text file.

   (cryptodata.txt, followerdata.txt)
4. An API that uses two tables and share a key.
5. Compared Crypto Rankings to amount of twitter followers.

GET DATA

Get Data Requirements	Achieved
3 API's	✔
Store 100 items in one table per API	✔
Have at least 1 table per API	✔
At least 1 API must have 2 tables that share a key.	✔
Limited the amount of data to 25 collected/stored at a time up to the first 100	✔

Application Programming Interface

	Name	Link	Purpose
API 1	Coingecko API	https://www.coingecko.com/api/documentations/v3	Collecting crypto data including price and rank
API 2	Tweepy API	https://docs.tweepy.org/en/latest/install.html	Collecting the amount of followers for each coin
API 3	General Mills API for Google Trends	https://github.com/cryptowatch/cw-sdk-python	Collecting Google trends data and timestamps on Dogecoin.

Problems

Problems

1. We were originally looking for total amount of crypto mentions on Twitter, but the API stops at 500 tweets. Because the amount of crypto tweets exceed 500 all of our data points came out to exactly that number. We knew that to be incorrect.

Solution: We focused our scope to look at the amount of followers for the top ten coins and compared their rankings.

2. Some of the Cryptocurrency sites did not allow web scraping. We were not able to use Beautifulsoup with coindesk.com.

Solution: Instead of using beautiful soup on a website, we utilized the API's that other Crypto market sites offered.

Problems

3. While working on our first visualization our data values were cluttering the graph and did not look visually appealing. We needed to find a way to adjust the visualization.

Solution: We changed the bar values in our visualization to display only the x and y axis data. The additional values are display when hovering over the graph.

4. Figuring out how to join tables and implement them into our code was challenge for the group.

Solution: We referred back to the lecture slides to get a further understanding on the concept until we were able to work it out in our project.

Calculations

```
Average % Price Change of the Top 100 Cryptocurrencies
```

The average percent change of the price of the top 100 cryptocurrencies is -2.8016%.

```
        Top 5 Cryptocurrency Market Caps (USD)
```

#1: bitcoin ($899380012105)
#2: ethereum ($260597224001)
#3: binancecoin ($76529075685)
#4: tether ($50279180985)
#5: ripple ($46116475567)

```
        Top 10 Cryptocurrency Coin Prices (USD)
```

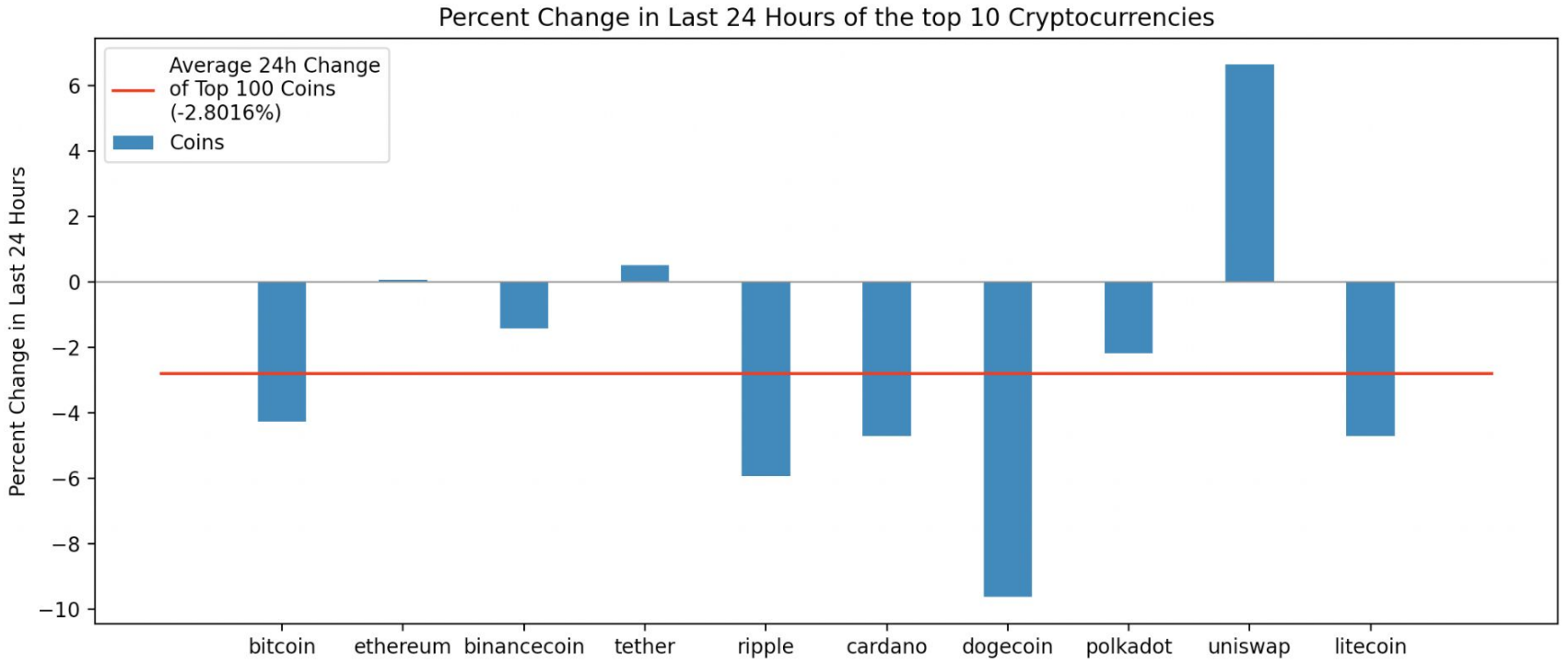
#1: bitcoin ($48555)
#2: ethereum ($2276.99)
#3: binancecoin ($499.47)
#4: tether ($1.0)
#5: ripple ($1.02)
#6: cardano ($1.07)
#7: dogecoin ($0.245613)
#8: polkadot ($29.34)
#9: uniswap ($33.13)
#10: litecoin ($218.38)

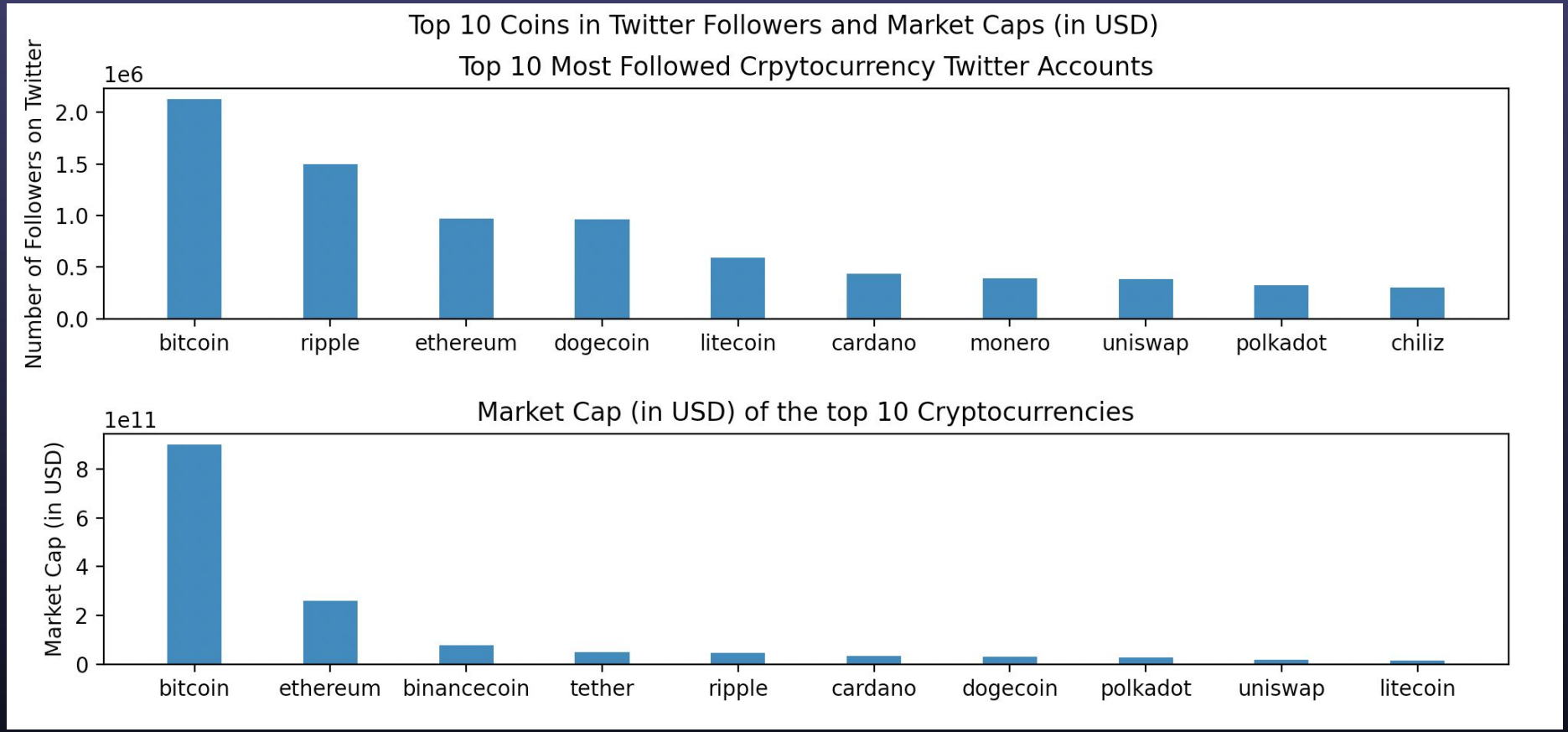
cryptodata.txt

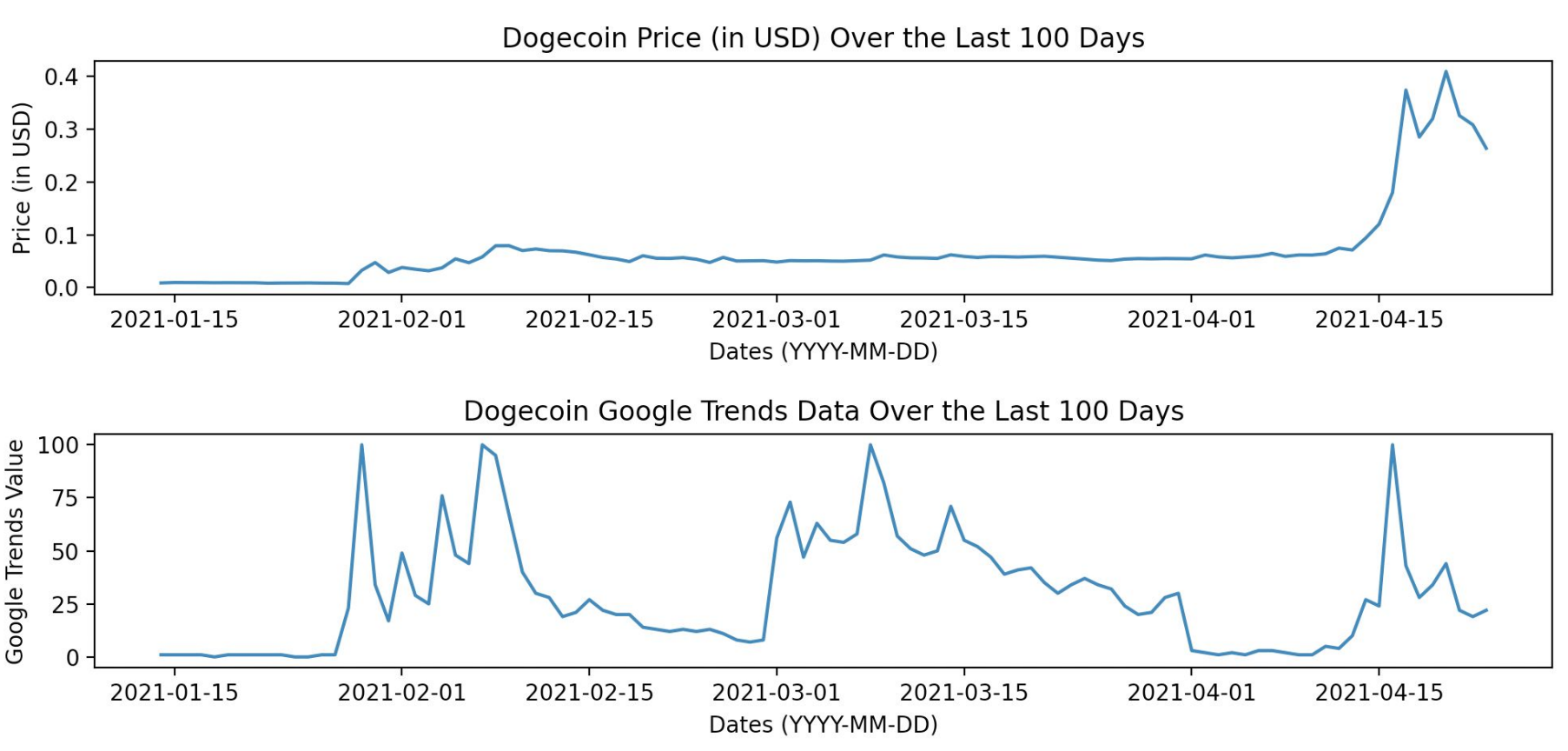
followerdata.txt

```
2126425,bitcoin
1501905,ripple
972479,ethereum
963325,dogecoin
589071,litecoin
434587,cardano
394696,monero
387690,uniswap
322763,polkadot
301684,chiliz
291537,eos
249009,chainlink
232566,qtum
220791,zilliqa
174539,iota
170207,solana
120770,decentraland
118053,nano
112134,bancor
108802,cosmos
105963,tezos
93500,bitcoingold
85657,algorand
78799,filecoin
77981,ankr
59012,thorchain
58560,ravencoin
56202,swissborg
33445,helium
29896,dash
26887,0x
23789,feiprotocol
20991,ethereumclassic
10529,piratechain
7334,amptoken
6905,paxosstandard
5183,tron
3928,maticnetwork
3887,maker
2824,tether
2435,zcash
2210,neo
1491,binancecoin
973,stellar
906,near
811,flow
548,waves
300,nem
279,digibyte
235,dai
168,okb
132,usdcoin
109,wazirx
101,thegraph
88,ontology
55,bitcoincashsv
31,aave
15,huobitoken
12,hederahashgraph
```

Visualizations


Percent Change in Last 24 Hours of the top 10 Cryptocurrencies
Average 24h Change of Top 100 Coins (-2.8016%)
Coins
Percent Change in Last 24 Hours
6
4
2
0
−2
−4
−6
−8
−10
bitcoin
ethereum
binancecoin
tether
ripple
cardano
dogecoin
polkadot
uniswap
litecoin


Top 10 Coins in Twitter Followers and Market Caps (in USD)
Top 10 Most Followed Crpytocurrency Twitter Accounts
1e6
Number of Followers on Twitter
2.0
1.5
1.0
0.5
0.0
bitcoin
ripple
ethereum
dogecoin
litecoin
cardano
monero
uniswap
polkadot
chiliz
Market Cap (in USD) of the top 10 Cryptocurrencies
1e11
Market Cap (in USD)
8
6
4
2
0
bitcoin
ethereum
binancecoin
tether
ripple
cardano
dogecoin
polkadot
uniswap
litecoin


Dogecoin Price (in USD) Over the Last 100 Days
Price (in USD)
0.0
0.1
0.2
0.3
0.4
2021-01-15
2021-02-01
2021-02-15
2021-03-01
2021-03-15
2021-04-01
2021-04-15
Dates (YYYY-MM-DD)
Dogecoin Google Trends Data Over the Last 100 Days
Google Trends Value
0
25
50
75
100
2021-01-15
2021-02-01
2021-02-15
2021-03-01
2021-03-15
2021-04-01
2021-04-15
Dates (YYYY-MM-DD)

Instructions

Installation

Make sure that these libraries are installed on your machine. Do this by typing “pip install ……” into your terminal.

pip install **tweepy**
pip install **numpy**
pip install **matplotlib**
pip install **mplcursors**
pip install **sqlite3**
pip install **pycoingecko**
pip install **pytrends**

Instructions to Run Code

1. Open the zip file and run “crypto_data.py” **four** times. This will create the database (crypto.db) and set up two tables named “Top100Coins” and “Top100CoinsIDs”. On the fourth time running, it will create a text file called “cryptodata.txt” which contains the average percent price change of the top 100 cryptocurrencies in the last 24 hours, as well as other data.

2. The second file that should be run is “twitter_data.py”. This file should also be run **four** times. This file creates one table called “Top100CoinsFollowers” and inserts the Twitter follower counts of the top 100 cryptocurrencies (if an account exists) into the table. The first time it should take 50-60 seconds, as it is collecting data from the tweepy API, however the next three runs should be quicker due to the previously collected data being cached in a text file called “followerdata.txt”. On the first run, the percentage complete will display as it runs.

3. The third file to be run is “doge.py”. This file should also be ran **four** times. This file creates three tables called “DogecoinUSDPast”, “DogecoinTrends”, and “DogeDateIDs”. This file collects the previous 100 days of Dogecoin prices (in USD) and also Google Trends value for the search term “dogecoin”, and inserts the values into the tables. Each time running should take around 20 seconds.

4. The final file that should be run is “visualizations.py”. This file only needs to run **once**.

To see the next visualization, close the window that appears.

5. To see the calculations made from the data, open “followerdata.txt” and

“cryptodata.txt” which

should have been created while running the other four files.

6. Open “crypto.db” to see the completed database, which should contain 6 tables total.

Code Documentation

crypto_data.py file

```
# calculates average change in price of the top 100 cryptocurrencies (in market cap values) over the last 24 hours
def getAvgChange(cur, conn):
```

```
# gets top 100 cryptocurrencies and returns a list of important data about each coin
def getTop100Coins():
```

```
# adds the top 100 coins to two tables within crypto.db
def addTopCoins(cur, conn):
```

crypto_data.py file

```
# creates the database crypto.db
def setUpDatabase(db_name):
```

```
# sets up two tables within crypto.db
def setUpTables(cur, conn):
```

```
# writes data calculated within this file to a txt file (cryptodata.txt)
def write_data_to_file(filename, cur, conn):
```

twitter_data.py file

```
# joins Top100CoinsIDs and Top100Coins to return the names of coins with their ids
def joinTables(cur, conn):
```

```
# will try to read in from a file containing follower count data
# if unable to read from file (does not exist yet/first time running file), it will get the proper follower count data,
# create a list with it, and create/write it to a file. doing this helps speed up run time because getting data from
# tweepy takes a while, so storing it in a cache file helps speed up the process.
def getSortedFollowerCounts(topCoins):
```

```
# adds the necessary information to the table
def addToTable(cur, conn):
```

```
# sets up one table within crypto.db
def setUpTables(cur, conn):
```

doge.py file

```
# returns the last 100 days of Google Trends values for Dogecoin
def getDogecoinTrends():
```

```
# returns the last 25 days of doge price (in USD) that were not already added to the table from coingecko
def getPrevDOGE(cur, conn):
```

```
# adds Google Trends values to tables and also adds dates to a separate table, with both having the same id/key
def addTrendsToTable(cur, conn):
```

```
# adds dogecoin price to DogecoinUSDPast table 25 prices at a time
def addPriceToTable(cur, conn):
```

```
# sets up three tables within crypto.db
def setUpTables(cur, conn):
```

visualizations.py file

```
# returns tuple of market caps and coin names through table joins
def joinTablesTwitter(cur, conn):
```

```
# returns tuple of percent price change and coin names through table joins
def joinTablesCrypto(cur, conn):
```

```
# returns tuple of dates and past prices through table joins
def joinTablesDogePrice(cur, conn):
```

```
# returns tuple of dates and trend values through table joins
def joinTablesTrends(cur, conn):
```

```
# calculates average change in price of the top 100 cryptocurrencies (in market cap values) over the last 24 hours
def getAvgChange(cur, conn):
```

visualizations.py file

```
# makes bar graph of the percent change of the top 10 cryptocurrencies (market cap values), along with an average line
def makeCryptoBar(cur, conn):
```

```
# makes two stacked bar graphs comparing the top coin market caps and the top coin twitter account follower counts,
# showing a correlation between coin market cap and twitter popularity
def makeTwitterBar(cur, conn):
```

```
# makes two trend lines comparing dogecoin price (in USD) and its google trend value, showing a correlation between
# coin price and search activity
def dogeTrendline(cur, conn):
```

Resources

Date	Issue Description	Location of Resource	Result (did it solve the issue?)
4/11	Working with Beautifulsoup on coindesk.com	https://stackoverflow.com/questions/50254397/web-scraping-with-python-request-and-beautiful-soup	No, we used an API from a stable site
4/11	Setting up a coinbase API	https://help.coinbase.com/en/pro/other-topics/api/how-do-i-create-an-api-key-for-coinbase-pro	We learned the site required a customer account and went with coingecko instead.
4/11	Setting up Tweepy API	https://docs.tweepy.org/en/v3.5.0/getting_started.html#introduction	Yes, this resolved issue and told use to be patient while waiting for permission.
4/11	Setting up Coingecko API	https://www.coingecko.com/api/documentations/v3	Yes, It helped us set up our first API
4/16	Writing to file: fail if does not exist - succeed if exists	https://stackoverflow.com/questions/2967194/open-in-python-does-not-create-a-file-if-it-doesnt-exist	Yes, this helped us write to a file and run a 'try & except'.

Date	Issue Description	Location of Resource	Result (did it solve the issue?)
4/19	Integer for average crypto gain too long. Needed to shorten decimal points	https://www.programiz.com/python-programming/methods/built-in/round	Yes, round() solved the issue
4/19	Calculation to convert a Unix timestamp	https://stackoverflow.com/questions/3309012/what-is-the-formula-for-calculating-a-timestamp	Yes, we were able to convert the timestamp to a more legible set of integers
4/22	General Mills API for Google Trends	https://github.com/cryptowatch/cw-sdk-python	Yes, It allowed us to access Google Trends Data
4/23	Managing the data in our dogecoin trendline	https://pandas.pydata.org/	Yes, we used the panda library to manage our data in the third visualization
4/23	Checking the accuracy of our Dogecoin trendline	https://trends.google.com/trends/explore?geo=US&q=%2Fm%2F0zmxk9t	Yes, it helped to double check the accuracy of Dogecoin trends data

Tables

Tables created from crypto_data.py file

Table: Top100Coins

	coin_rank	curr_usd_price	curr_usd_mkt_cap	perc_price_change
	Filter	Filter	Filter	Filter
1	1	48555	899380012105	-4.2507
2	2	2276.99	260597224001	0.05869
3	3	499.47	76529075685	-1.42454
4	4	1	50279180985	0.51665
5	5	1.02	46116475567	-5.92607
6	6	1.07	34038818574	-4.70784
7	7	0.245613	31245735705	-9.6192
8	8	29.34	28697812081	-2.169
9	9	33.13	17063436304	6.63681
10	10	218.38	14495870642	-4.70972
11	11	746.91	13856924320	-5.44885
12	12	1	13622417004	0.3317
13	13	31.23	13044197204	-2.712
14	14	47.36	12510385138	15.2524
15	15	0.171066	10959358761	-8.47884
16	16	0.414118	9435282429	-3.85483
17	17	133.26	9095407249	-2.64811
18	18	8.85	8738574057	-0.43776
19	19	48403	7543776424	-4.44403
20	20	0.101341	7189247367	-4.49692
21	21	1.01	7152358304	1.03879
22	22	17.46	6446115152	10.40579
23	23	351.21	6236854434	-11.12639
24	24	2.26	5544280698	5.85182

Table: Top100CoinsID

	coin_rank	coin_name
	Filter	Filter
1	1	bitcoin
2	2	ethereum
3	3	binancecoin
4	4	tether
5	5	ripple
6	6	cardano
7	7	dogecoin
8	8	polkadot
9	9	uniswap
10	10	litecoin
11	11	bitcoin-cash
12	12	usd-coin
13	13	chainlink
14	14	solana
15	15	vechain
16	16	stellar
17	17	filecoin
18	18	theta-token
19	19	wrapped-bitcoin
20	20	tron
21	21	binance-usd
22	22	terra-luna
23	23	monero
24	24	klay-token

Table created from twitter_data.py file

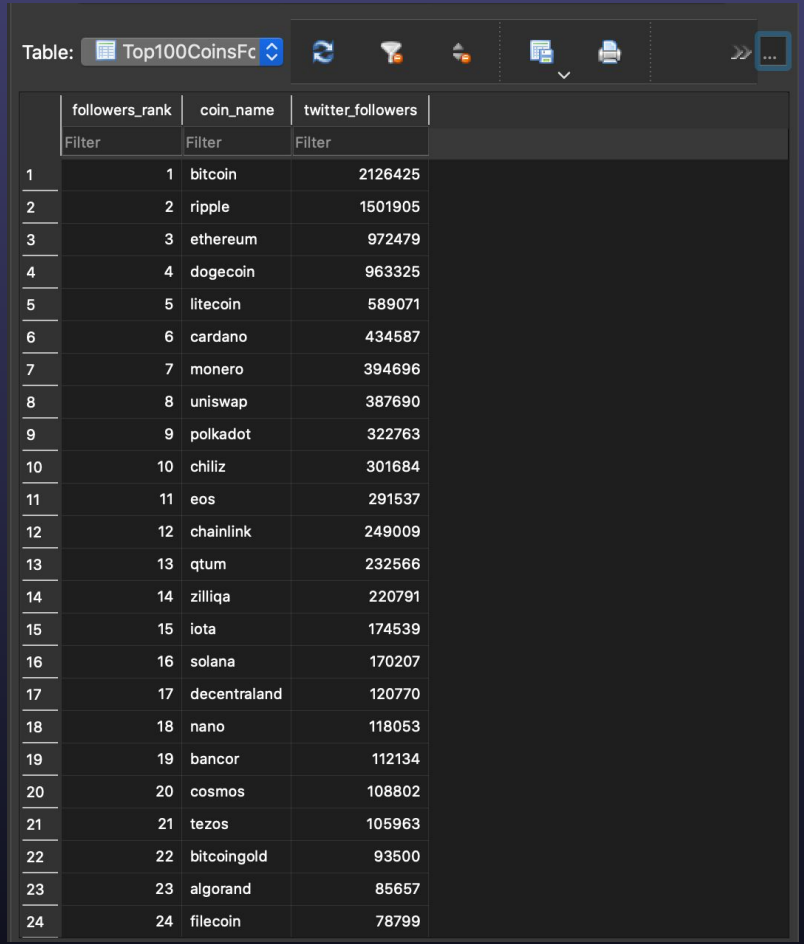


Table: Top100CoinsFc

	followers_rank	coin_name	twitter_followers
	Filter	Filter	Filter
1	1	bitcoin	2126425
2	2	ripple	1501905
3	3	ethereum	972479
4	4	dogecoin	963325
5	5	litecoin	589071
6	6	cardano	434587
7	7	monero	394696
8	8	uniswap	387690
9	9	polkadot	322763
10	10	chiliz	301684
11	11	eos	291537
12	12	chainlink	249009
13	13	qtum	232566
14	14	zilliqa	220791
15	15	iota	174539
16	16	solana	170207
17	17	decentraland	120770
18	18	nano	118053
19	19	bancor	112134
20	20	cosmos	108802
21	21	tezos	105963
22	22	bitcoingold	93500
23	23	algorand	85657
24	24	filecoin	78799

Tables created from doge.py file

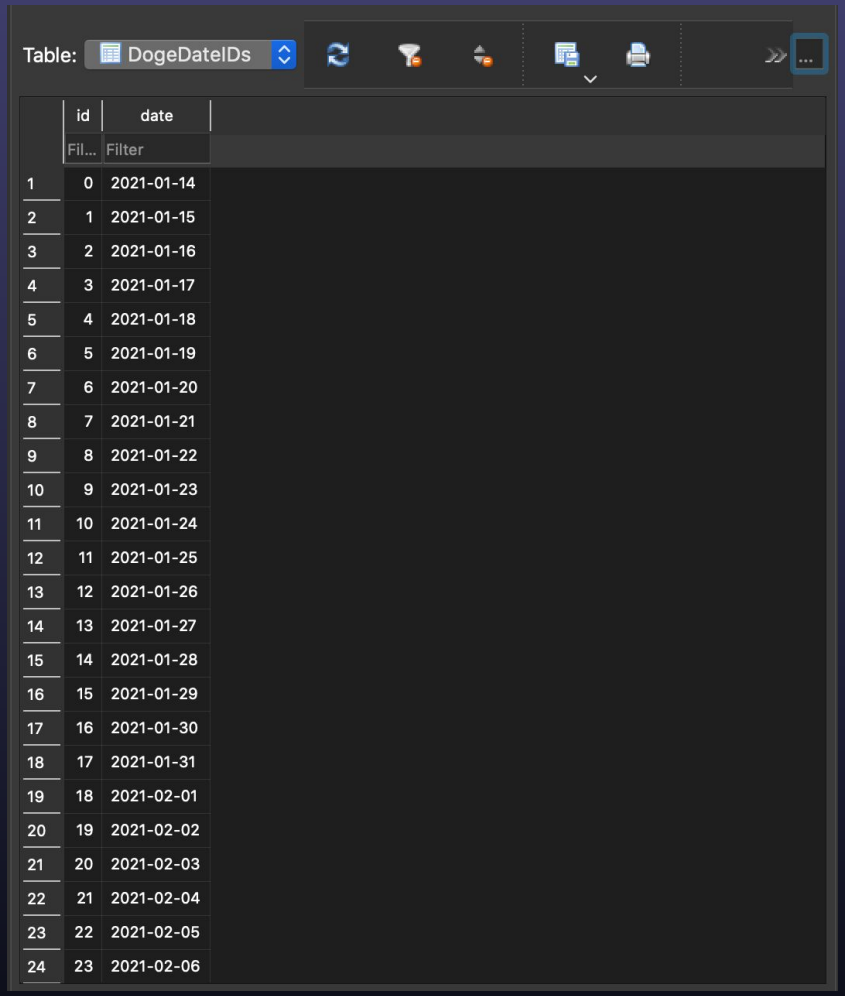
Table: DogecoinTrend

	id	trend_value
	Fil...	Filter
1	0	1
2	1	1
3	2	1
4	3	1
5	4	0
6	5	1
7	6	1
8	7	1
9	8	1
10	9	1
11	10	0
12	11	0
13	12	1
14	13	1
15	14	23
16	15	100
17	16	34
18	17	17
19	18	49
20	19	29
21	20	25
22	21	76
23	22	48
24	23	44

Table: DogecoinUSDP

	id	usd_price
	Fil...	Filter
1	0	0.00865
2	1	0.00947
3	2	0.00933
4	3	0.00926
5	4	0.00903
6	5	0.00916
7	6	0.00905
8	7	0.00905
9	8	0.00816
10	9	0.00852
11	10	0.00857
12	11	0.00873
13	12	0.00837
14	13	0.00828
15	14	0.00744
16	15	0.03264
17	16	0.04735
18	17	0.02852
19	18	0.03799
20	19	0.03459
21	20	0.03162
22	21	0.03738
23	22	0.05428
24	23	0.04706

Tables created from doge.py file



	id	date
	Fil...	Filter
1	0	2021-01-14
2	1	2021-01-15
3	2	2021-01-16
4	3	2021-01-17
5	4	2021-01-18
6	5	2021-01-19
7	6	2021-01-20
8	7	2021-01-21
9	8	2021-01-22
10	9	2021-01-23
11	10	2021-01-24
12	11	2021-01-25
13	12	2021-01-26
14	13	2021-01-27
15	14	2021-01-28
16	15	2021-01-29
17	16	2021-01-30
18	17	2021-01-31
19	18	2021-02-01
20	19	2021-02-02
21	20	2021-02-03
22	21	2021-02-04
23	22	2021-02-05
24	23	2021-02-06

Thank you