Analyzing the Effects of Coronavirus Mentions on the Stock Market

SI 206 Final Project

Team: Coding with the Homies
Members: Tara Moore, Alex Sterchle, Brenna Doyle

Table of Contents

REPOSITORY:

https://github.com/brennadoylee/SI206_Final_Project

GOALS:

As COVID-19 has taken over the world, we were curious to see the effects that it was having on the economy as a whole. To most accurately depict this effect, we decided to inspect how the stock market was performing in relation to the frequency with which coronavirus was being talked about. Our goal was to determine if the stock market prices were being effected as COVID-19 was featured more in our twitter feeds. Our original goal was to collect all of the tweets containing 'COVID-19' and also take into account the sentiment analysis of these tweets and how the stock market might be affected by that factor. However, as the COVID-19 frenzy has spread, it quickly became clear that we would have to narrow it down to tweets from one twitter account, the CDC, as the volume of total tweets was astronomical. We also discovered that the sentiment analysis varied so heavily per day as new information is consistently released, that it would not be effective to further find a correlation between the stock prices and the sentiment analysis.

The primary goal that we achieved was the effect that the increased mention of COVID-19 had on the stock market, as well as the general trend of the stock market as well. We were able to gain a greater understanding of how COVID-19 is truly impacting the world around us through the data that we collected and visualized.

PROBLEMS FACED:

NASDAQ API:

- When attempting to decide what stock to choose, I decided to go for one that reflected the general trends of the market which ultimately led me to choose NASDAQ. The first problem I ran into was that the API I had wanted to use and had found for the plan we created, was actually not for personal use, but rather for the use of companies. I ultimately found IEX which allowed me to create an account and use their API. However, their API was very different from those I had previously worked with so I had to find other resources to figure out how to get the data I wanted.

Gas API:

- When I first started searching for API's to use, I learned there are many different oil stocks to pick from. I wanted to find an API that could reflect typical consumer trends, so I first chose petroleum gas stocks, however after gathering the cache dictionary learned this wasn't exactly what I was looking for. Instead, I picked conventional motor oil stock prices. The original plan for this API was to track oil stocks on a daily to weekly basis to later compare to the NASDAQ and Twitter APIs. This was especially difficult to compare to the Twitter API because the data was very different and also was on a much more frequent basis than my weekly Gas API.

Twitter API:

- The first problem that I had was regarding the Twitter API authentication process. They are very strict, and made me go through many hoops to get authenticated. I ultimately ended up speaking directly with Twitter HQ about the issue and API usage. When I first started pulling tweets, I was using the Twitter module rather than the Tweepy module. When using the Twitter module, I was having struggles filtering tweets for not only the @CDCgov username but also the words "COVID" and "coronavirus". Also, while using this module, there was not a simple way to extract tweets from a start date to an end date. Once I switched over to Tweepy,

extracting data like the time the tweet was created and the tweet id was much easier. Also, I was having a tough time trying to write my own tweet sentiment analyzer, and once I found the TextBlob module (through) Tweepy, I was able to write a quick and easy function to get my desired sentiment results.

CALCULATIONS:

Nasdaq Calculations:
https://github.com/brennadoylee/SI206_Final_Project/blob/master/NASDAQ_Monthly_Avg.txt

Gas Calculations:
https://github.com/brennadoylee/SI206_Final_Project/blob/master/Oil_Averages.txt

Twitter Calculations:
https://github.com/brennadoylee/SI206_Final_Project/blob/master/CDC_Twitter_Calculations.txt

VISUALIZATIONS:

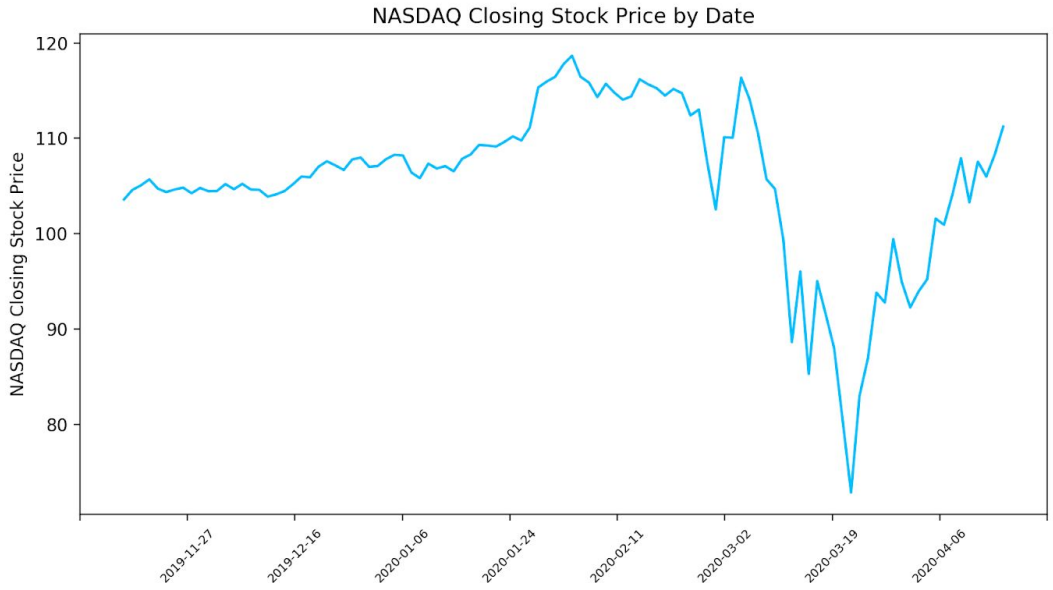

Figure 1. This graph shows the NASDAQ Closing Stock Price beginning in November, resulting in a dramatic decline in March.

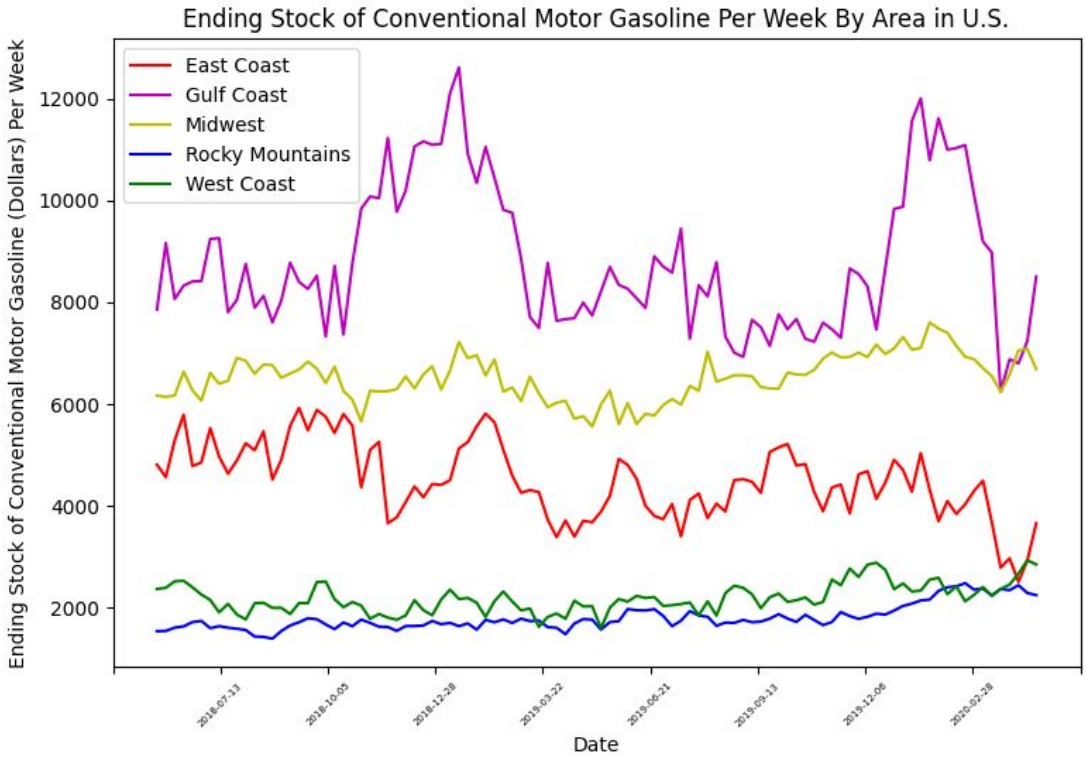

Figure 2. This graph shows the Ending Stock of Conventional Motor Gasoline Per Week by different areas within the United States. The weeks range from July 2018 to April 2020.

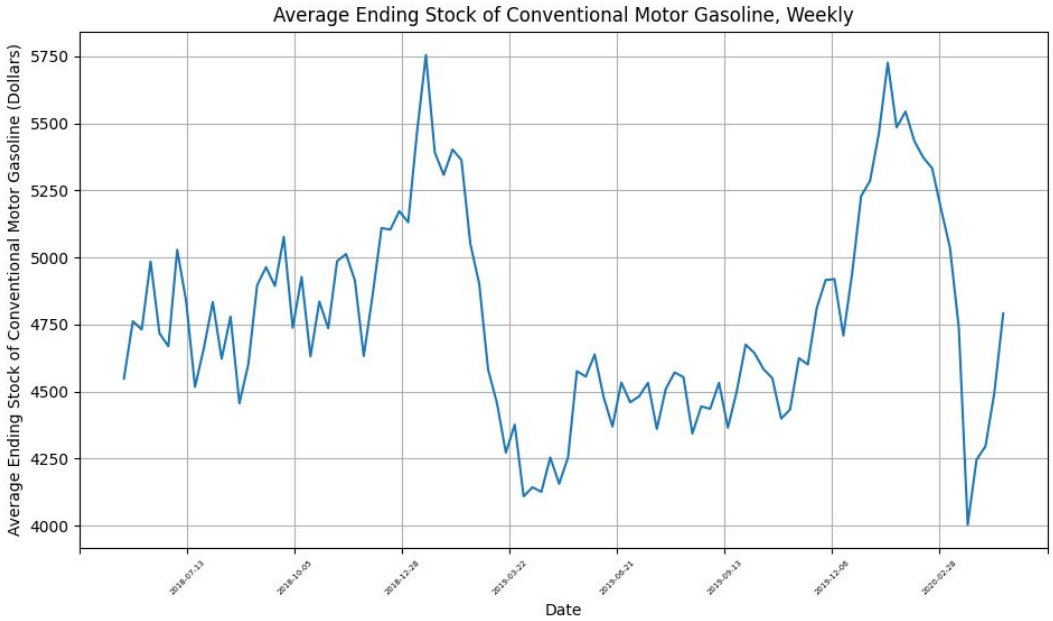

Figure 3. This graph shows the average ending stock of conventional motor gasoline weekly. This was the resulting graph from the Gas Calculations. The dates range from July 2018 to April 2020.

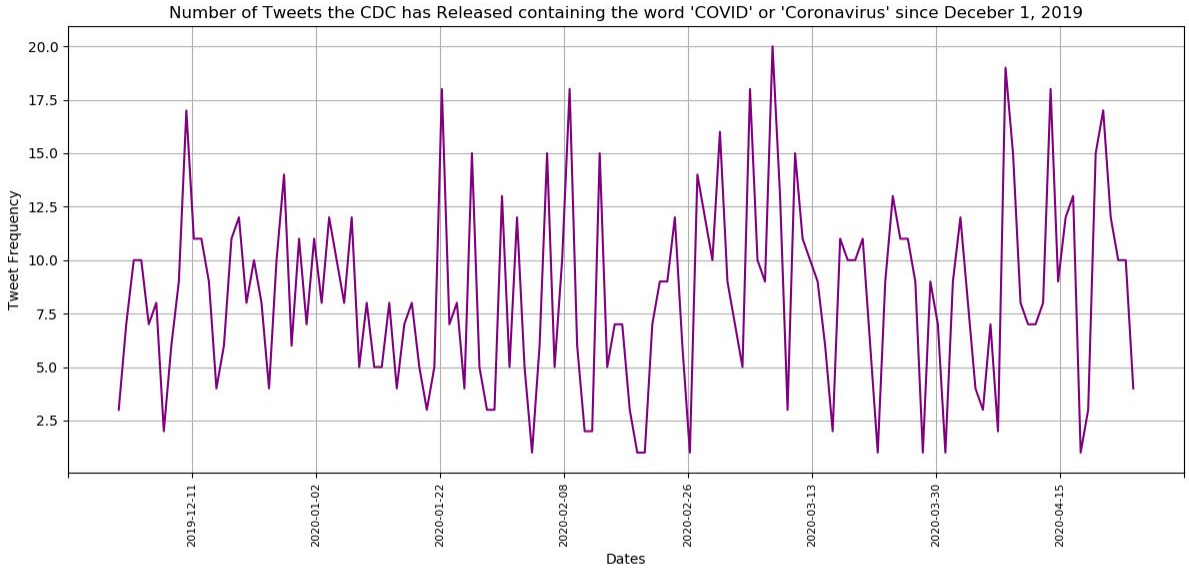

Figure 4. This graph shows the number of tweets the CDC has released containing the word "COVID" or "coronavirus" per date. The dates range from December 1, 2019 to April 26, 2020.

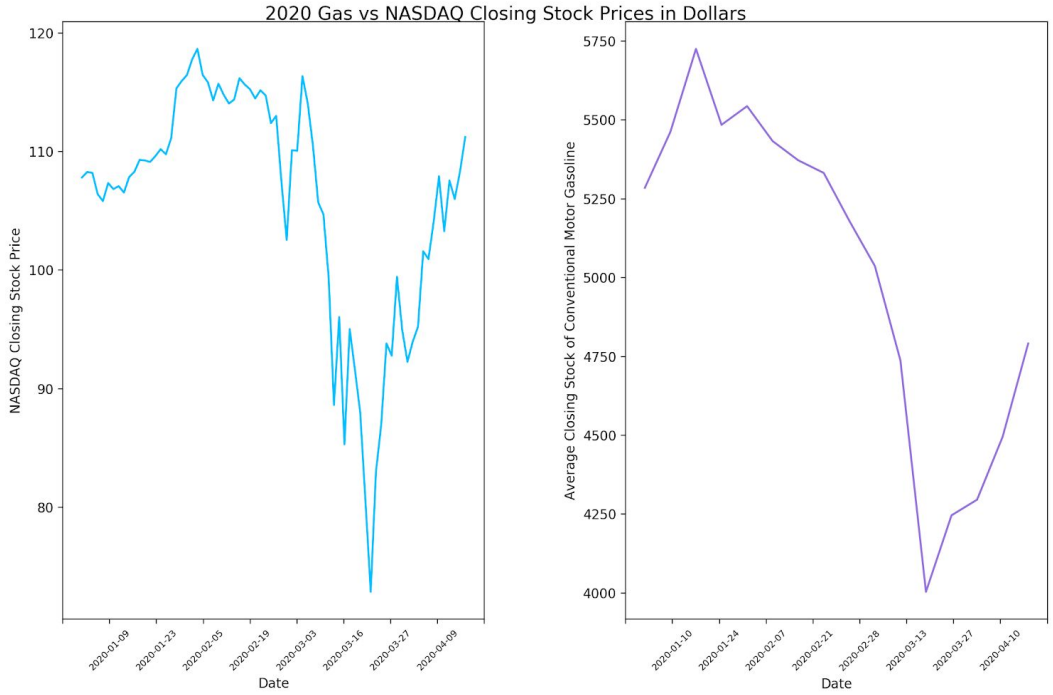

Figure 5. This graph shows the Ending Stock of NASDAQ per week versus the Ending Stock of Conventional Motor Gasoline Per Week by different areas within the United States. The weeks range from January 2020 to April 2020.

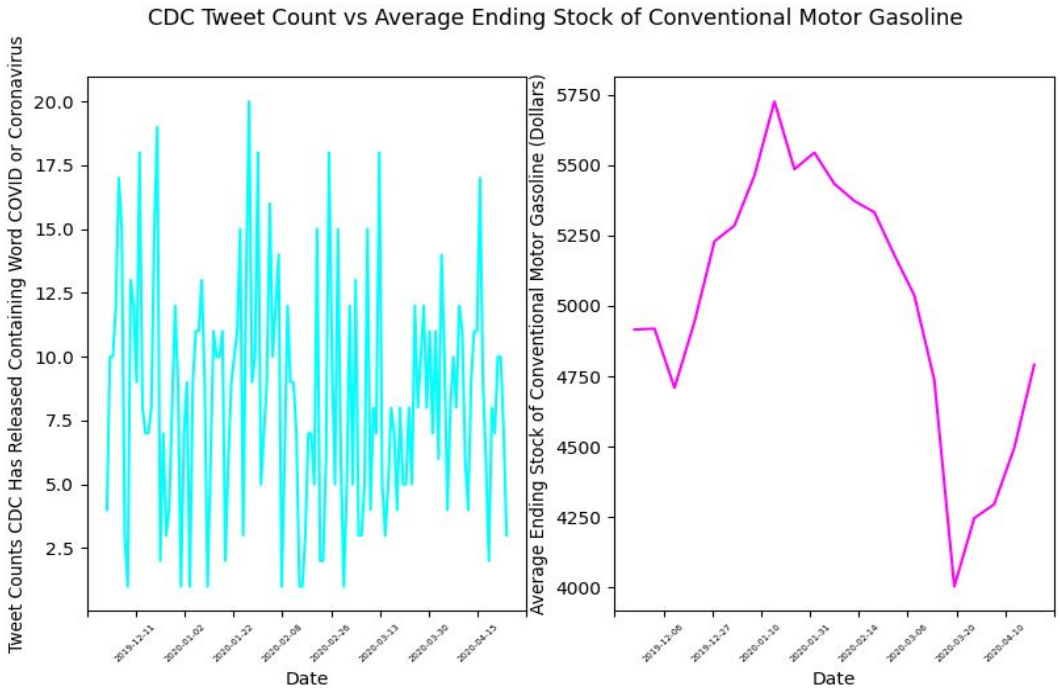

Figure 6. This graph shows the Ending Stock of Conventional Motor Gasoline Per Week by different areas within the United States versus the total count of tweets with "COVID" or "coronavirus" from the CDC per date. The weeks range from December 2019 to April 2020.

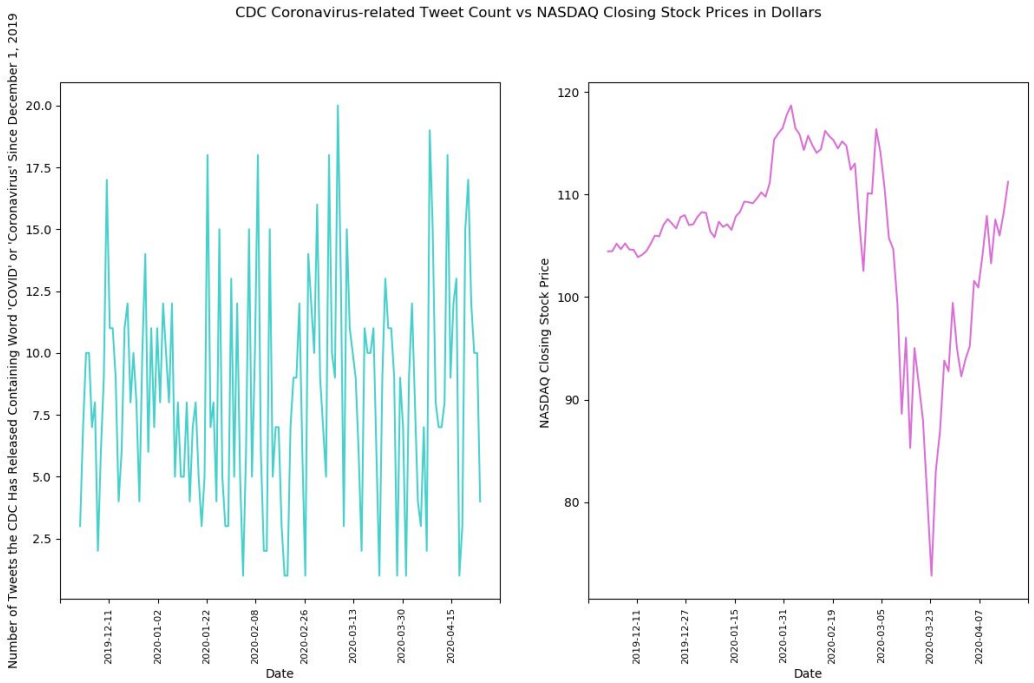

Figure 7. This graph shows the NASDAQ Closing Stock Price per date versus the total count of tweets with "COVID" or "coronavirus" from the CDC per date. The weeks range from December 2019 to April 2020.

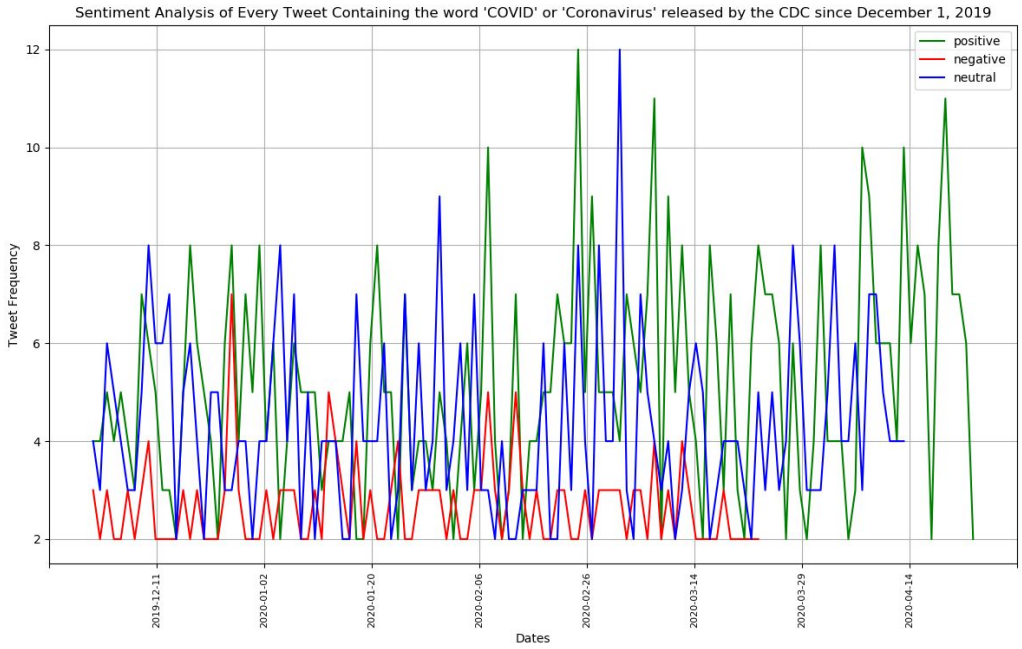

Figure 8. This graph shows the Sentiment Analysis of every tweet containing the word "COVID" or "coronavirus" released by the CDC. The three different lines represent the three different results offered by the sentiment analysis: Positive, Negative, or Neutral. The dates range from December 1, 2019, to April 26, 2020.

INSTRUCTIONS:

To run the code, you need to open the StocksAPI.py, gasprice.py, and TwitterAPI.py. These three python programs will insert data into the database "finalprojectdatabase.db" and create the visualizations listed above in a popup window that is then saved as a png. Also, you will need to install/import Tweepy, DateTime, time, TextBlob, Matplotlib, matplotlib.ticker, sqlite3, NumPy, pandas, iexfinance, unittest, Requests, re, and http.client onto your local computer in order to use the imported modules.

FUNCTION DOCUMENTATION:

gasprice.py:

FUNCTION	RESULT
read_cache	**Input:** - Filename for cache **Output:** - Returns dictionary to read from cache filename
write_cache	**Input:** - Filename for cache - Cache dictionary **Output:** - Writes dictionary to the cache file
get_url	**Input:** - Area id **Output:** - Returns url to call to get API
get_info	**Input:** - Url returned from get_url - Cache filename **Output:** - Returns the cache dictionary of that url with oil prices weekly
eastcoast_table	**Output:** - Returns table of East Coast motor oil stocks per week
gulfcoast_table	**Output:** - Returns table of Gulf Coast motor oil stocks per week
midwest_table	**Output:** - Returns table of Midwest motor oil stocks per week
rockymountain_table	**Output:** - Returns table of Rocky Mountain motor oil stocks per week
westcoast_table	**Output:** - Returns table of Gulf Coast motor oil stocks per week
join_tables	**Output:** - Joined table of all areas' motor oil stocks into one table

nasdaq.py:

FUNCTION	RESULT
getHistoricalPrices	**Input:** - Stock ticker - Start Date - End Date **Output:** - Request is made to obtain the historical prices from stock in date range inputted from API and write that data as a Python object - Returns the data
Ndate	**Input:** - Year, Month and Day **Output:** - Returns Closing Price for NASDAQ on that date
nasdaq	**Input:** - cursor and connection **Output:** - Retrieves the NASDAQ closing stock prices from database
monthlyNStock	**Input:** - list with tuple gathered from Database **Output:** - The average closing price for NASDAQ for each month

TwitterAPI.py:

FUNCTION	RESULT
tweet_analysis	**Input:** - A single Tweet pulled from the Twitter API **Output:** - A sentiment analysis of the tweet of either “Positive”, “Negative” or “Neutral” depending on its TextBlob sentiment score
get_all_tweets	**Input:** - The twitter username (@CDCgov) **Output:** - A list of lists containing the information of each pulled from

	the @CDCgov twitter in the form of [tweet ID number, created at date, sentiment analysis result, and the actual tweet]
covid_tweet_count	**Input:** - cursor and connection **Output:** - Retrieves data from the database (TotalTweets table) in the form of a list of tuples, where the tuple is: (tweet dates, number of coronavirus related tweets posted that day)
reverse	**Input:** - A list of dates **Output:** - The inputted list in reverse order
creating_weeks	**Input:** - A list of dates and the number of dates that go in each group **Output:** - A list containing lists of dates divided into groupings of seven, representing each week
tweets_per_week	**Input:** - The list of weeks outputted by divide_chunks **Output:** - A dictionary where the keys are the week number and the value is the average number of COVID-related tweets posted that week
tweet_sentiment	**Input:** - Cursor and connection **Output:** - Retrieves data from the database (TweetSentiment table) in the form of a list of tuples, where the tuple is: (tweet dates, sentiment analysis)
remove_duplicates	**Input:** - A list of dates **Output:** - The same list without duplicate elements
nasdaq_data	**Input:** - Cursor and connection **Output:** - Retrieves data from the database (TweetSentiment table) in the form of a list of tuples, where the tuple is: (dates, closing prices)

RESOURCES USED:

DATE	ISSUE DESCRIPTION	RESOURCE LOCATION	RESULTS
4/18	API understanding	Homework #7	Use previous homework to assist in API data collection
4/18	API Retrieval	https://iexcloud.io/console/	Used API and personal token to obtain NASDAQ stock information
4/18	Understand how the IEX API worked	https://addisonlynch.github.io/iexfinance/stable/	Able to use API's built in function to effectively scrape the data I needed (close_only = True feature so that I did not have unnecessary data collected and stored)
4/19	Pulling Tweets from a given start date to a given end date	https://stackoverflow.com/questions/49731259/tweepy-get-tweets-between-two-dates	We were able to use datetime.datetime to retrieve tweets from December 2019 to today instead of just the approximate 3000 most recent tweets (Tweepy's default function)
4/19	Tweet Sentiment Analysis	https://www.geeksforgeeks.org/twitter-sentiment-analysis-using-python/	We were able to quickly get the sentiment analysis from tweets by installing/importing TextBlob
4/20	API Retrieval	https://www.eia.gov/opendata/qb.php?category=388221	Received API and keys to different areas' conventional motor gasoline stock prices

4/20	Exporting to a database	Homework #8	We were able to more easily insert data into the database and subsequently pull data from our tables.
4/21	JSON Formatting	https://jsonformatter.org	We were able to more easily view and index into our dictionaries returned from our cache file
4/25	Creating Visualizations	matplotlibs.org	We were able to personalize the visualizations and make them more aesthetically appealing.
4/25	Creating Visualizations	Matplotlibs-v3.ppt - lineplot.py	We were able to use lineplot.py to properly format our own code when creating our visualizations