Analyzing Coronavirus Effects on the Media and Job Market

Team Tech Yeah

https://github.com/bpowning/SI206-FinalProject-TechYeah.git

Course: SI 206: Data-Oriented Programming

Date: April 20, 2020

Group Members: Sophie Loesberg, Brooke Powning, Max Mittleman

Word count: 2085

Goals

As the world is greatly affected by the Coronavirus Pandemic, we as students wanted to explore how aspects of our everyday lives were impacted. Through this we determined to inspect how the media and job market were personally affected. This led us to using the New York Times API, the Github Jobs API, and the Coronavirus Live API. Our goal was to examine how coronavirus has increased mentions in the media and increased the number of remote job postings.

The goals that we achieved include a comprehensive view on the progression of mentions of coronavirus in the media as well as an understanding of how the United States compares to the globe in terms of confirmed deaths, and recovered coronavirus cases with increase in remote job listings as a result. Additionally, as a team we were able to gain an understanding of new concepts and apply those to a real world pandemic. As students, it was exciting to accomplish our goal of using data to gain insights into the effects of this global pandemic.

Challenges

New York Times API

- When I originally created my test database and tables, I had run into the obstacle of having duplicate entries for the same article. After testing where I should append certain attributes of an article in my loops through the dictionary, I decided to keep the original plan and filter out repeated entries in an ‘if’ statement, checking to make sure the attributes of the current article were not already in the lists that collect them to create the NYTCoronavirusPosts table.

- After finishing the construction of the table, I went back through the articles in the month data set and realized that there are articles about Coronavirus that don't have the keyword I was searching for in the articles. To widen my search, I looked to see if an article mentioned the disease in their headline or in their URL to also add it to my database, collecting all articles concerning this crisis and creating an accurate analysis of the NYT data.

Coronavirus API

- Our original idea with the Coronavirus data was to see a progression over time and compare that data with Github Jobs and New York Times. Due to the nature of the Coronavirus API we chose, it was only live data and would be constantly updated; due to this, we had to change our progression of how we would proceed with the Coronavirus data, such as comparing live data on the confirmed, deaths, and recovered cases by country and examining the United States specifically versus globally.

Github Jobs API

- When first discussing our project goal, we had a plan to see widespread effects of Coronavirus on the community, specifically the decreasing job market. However, as we dug deeper into this Github Jobs API, we noticed that it only showed developer jobs, which is not to the scale we originally wanted. To account for this, we scaled down the compared data to give a more appropriate comparison between the effects of Coronavirus by having the axes be in the hundreds. Another problem I faced was the API's broad search. There were not many parameters to help narrow my search for specific job data, so I had to account for that in my calculations.

Calculations

calculations.txt

Visualizations

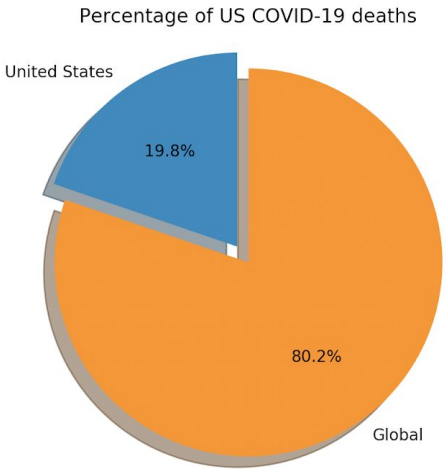

Figure 1. This graph shows the percentage of COVID-19 deaths in the US out of the global COVID deaths.

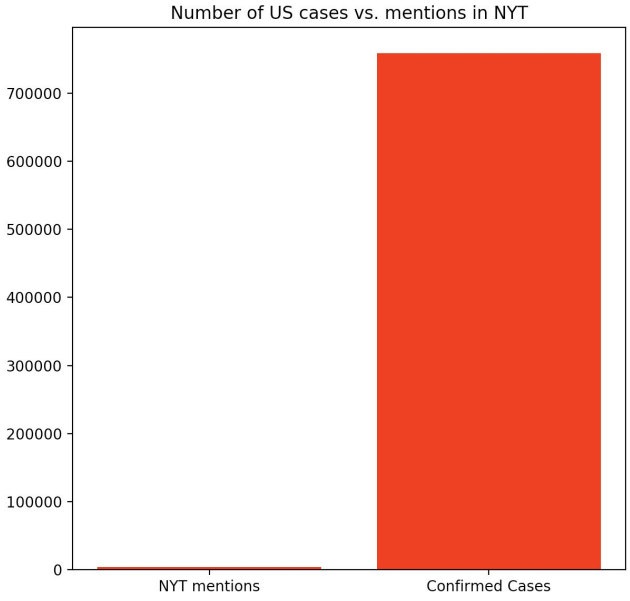

Figure 2. This graph shows the number of US confirmed cases of COVID-19 compared to the number of times COVID-19 is mentioned in NYT headlines.

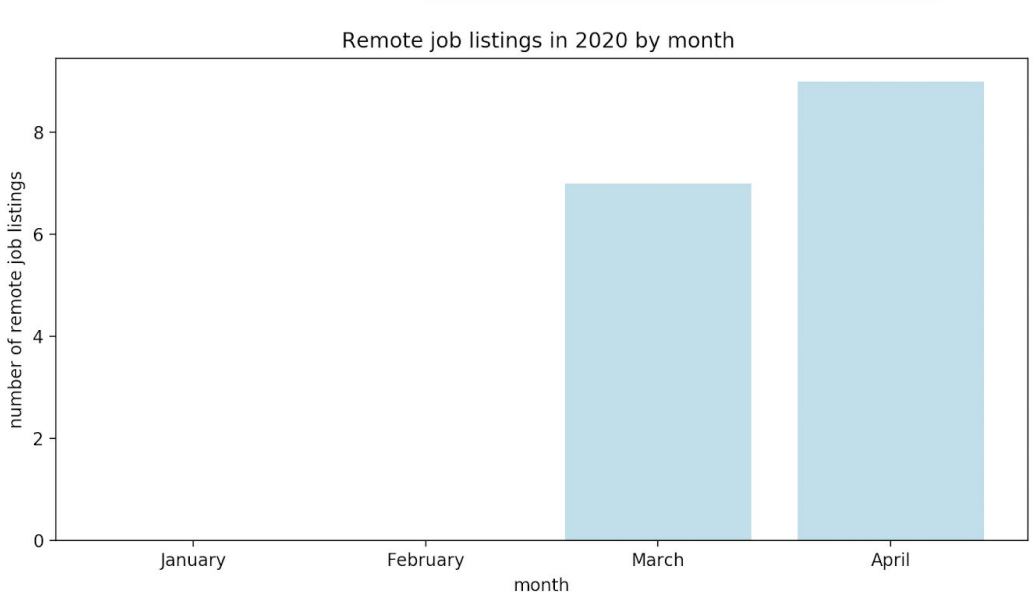

Figure 3. This graph shows the number of remote job listings on Github Jobs for every month in 2020.

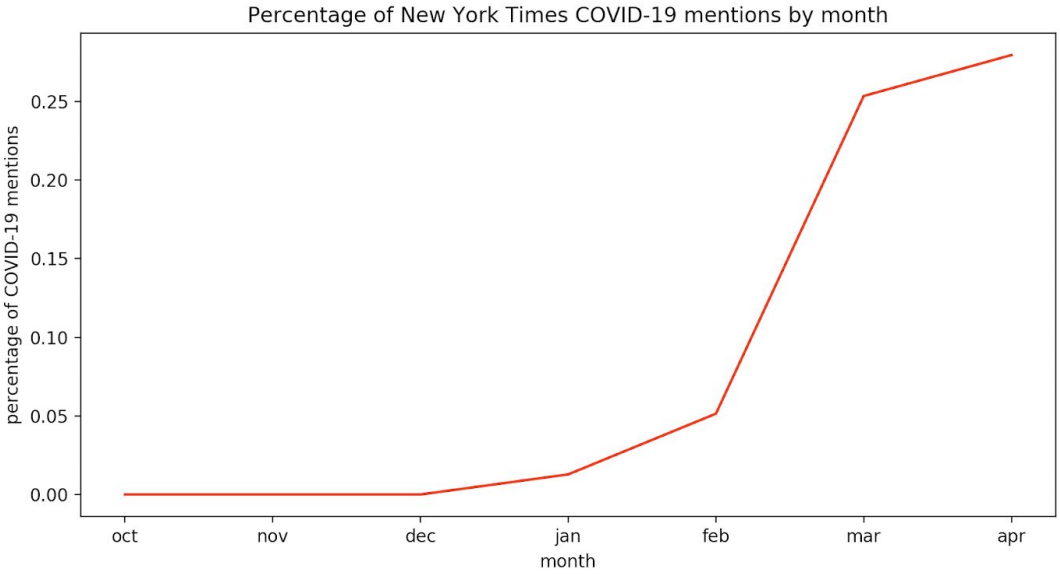

Figure 4. This graph shows the increase of the percentage of COVID-19 mentions in the NYT by month compared to the total number of articles published.

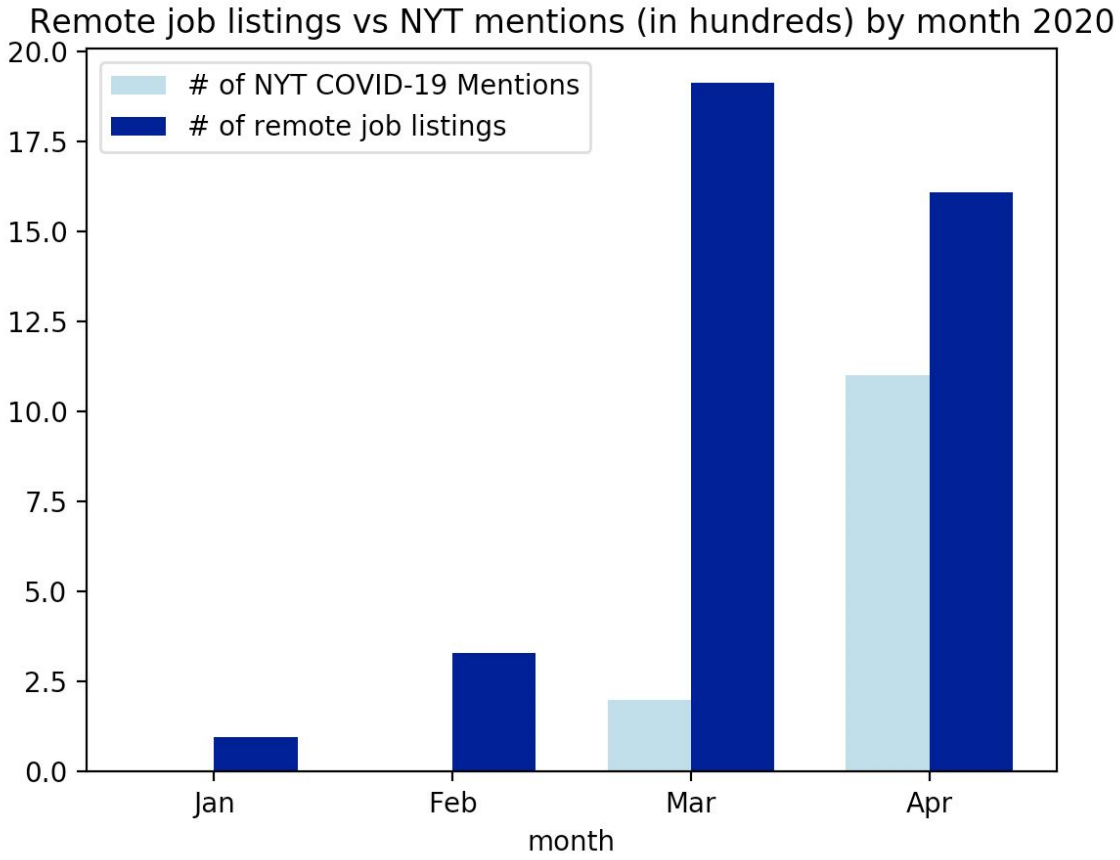

Figure 5. This graph shows the number of remote job listings by month compared with the number of NYT COVID-19 mentions by month (shown in hundreds).

Instructions

To run the code, you need to open up the githubjobs-api-reader.py, nyt-api-reader.py and covid_api.py. These three functions all make API calls to their respective API documentation. When those three files are run, the images of the visualizations will both appear in a pop up python screen and be saved as a png file under their names. The database will have been created

titled coronavirus.db which contains six different tables. Each API also has a JSON file which held our information title cache_jobs.json, covid_data.json, and nyt_covid_data.json.

Function Documentation

Coronavirus

1. read_cache	Takes a cache file as a parameter. Reads the JSON from the cache file and returns a dictionary from the cache data. If the file doesn't exist it returns an empty dictionary.
2. write_cache	This function encodes the cache dictionary (cache_dict) into JSON format and writes the JSON to the cache file (cache_file) to save the search results.
3. get_data	Takes the base url and cache file as parameters. read_cache is used on the cache file and if the base url is in the cache dictionary, the data is returned. If the base url is not in the cache dictionary, a request is made to get the data from the API and write that data as a Python object to the file. It returns the data.
4. numberOfConfirmed	Takes a string (the name of a country), cur and conn as parameters. It selects the total confirmed cases in the country of the string that is passed in the parameter and returns that integer.
5. numberOfDeaths	Takes a string (the name of a country), cur and conn as parameters. It selects the total deaths in the country of the string that is passed in the parameter and returns that integer.
6. numberOfRecovered	Takes a string (the name of a country), cur and conn as parameters. It selects the total recovered cases in the country of the string

	that is passed in the parameter and returns that integer.
7. percentageOfConfirmed	Takes a string (the name of a country), cur and conn as parameters. It uses the numberOfConfirmed function on the country passed in the parameters and then selects the total confirmed global cases. To find the percentage of the country's confirmed cases versus the global confirmed cases, the number of confirmed by country is divided by the global confirmed cases and the result is returned.
8. percentageOfDeaths	Takes a string (the name of a country), cur and conn as parameters. It uses the numberOfDeaths function on the country passed in the parameters and then selects the total global deaths. To find the percentage of the country's deaths versus the global deaths, the number of deaths by country is divided by the global deaths and the result is returned.
9. percentageOfRecovered	Takes a string (the name of a country), cur and conn as parameters. It uses the numberOfRecovered function on the country passed in the parameters and then selects the total recovered global cases. To find the percentage of the country's recovered cases versus the global recovered cases, the number of recovered by country is divided by the global recovered cases and the result is returned.

New York Times

1. read_cache	Takes a cache file as the parameter. Opens and reads the file and loads the data into a Python dictionary then returns the dictionary. If the file doesn't exist, it returns an empty dictionary.

2. write_cache	Takes a cache file and a cache dictionary as the parameters. Opens and writes the cache dictionary as a string to the file. It doesn't return anything.
3. url_for_month_2k19	Takes a numerical month to get the correct URL to access the NYT Archive API for the entered month in the year 2019. Returns the correct URL with month and API Key.
4. url_for_month_2k20	Takes a numerical month to get the correct URL to access the NYT Archive API for the entered month in the year 2020. Returns the correct URL with month and API Key.
5. get_data	Takes in a base_url and the cache_file (while is the JSON file) and creates the cache_dict, a python object storing the JSON data in a dictionary. If the base_url is in the cache_dict, it returns cache_dict at the base_url value. If the base_url is not in the cache_dict, it loads the base_url as a text file into the cache_dict at the base_url value, then uses the write_cache function to write the cache_dict to the file. The function then returns the cache_dict at the base_url value.
6. getNYTCoronaPostsNumberbyMonth	Takes in a chosen simple publish date as well as the cursor and connection objects to access data and calculate the number of posts by month with the matching simple publish date. It executes the order to select the articles and their attributes from NYTCoronavirusPosts where the inputted chosen simple publish date is equal to the article's simple publish date, and adds them to a list.

Github Jobs

1. read_cache	Takes a cache file and opens it. Loads file into a python dictionary and returns it. If it doesn't open, it returns an empty dictionary.
2. write_cache	Takes a cache file and a dictionary. It opens

	up the file as write file and dumps the the contents into the dictionary.
3. url_by_location	Takes a description of a job and returns a base url for the github jobs API with that job as a parameter.
4. url_by_description	Takes a location and returns a base url for the github jobs API with that loaction as a parameter.
5. get_data	Takes a url for an API call and a cache_file which in this case is set to jobs_file to make this API call easier. It reads the file using the read_cache function and returns the contents of the dictionary if that url has already been called. Otherwise it writes the content to the json file using write_cache.
6. get_listings_by_date	Takes a month, a year and a database connection. It returns a list of all of the job listings from that month and year from the database.
7. remote_listings_by_date	Takes a month, a year, and a database connection. It returns a list of all of the remote job listings from that month and year from the database.
8. remote_total_by_month	Takes a month and year and calls the remote_listings_by_date function. It applies the len function to that so it returns the number of remote listings.
9. month_totals_list	Takes a list of months and the year. It goes through each of the months, using the remote_total_by_month function and appending that to a list. It returns a list of the totals for each of the months passed into the function.
10. yearly_total	Takes a list of months and the year. It calls the months_totals_list function and iterates through that adding to an accumulator variable. It returns the total.

11. get_total_remote	Has no input. Returns the number of remote listings from the database.
12. get_percent_remote	Has no input. Returns the number of remote listings from the database by taking the total from get_total_remote and dividing it by the total listings in the database.
13. get_listings_by_type	Takes in a job type. Returns a list of all of the jobs in that type by joining the JobType and JobListings tables together on the JobType id.
14. num_listings_by_type	Takes in a job type. Applies the len function on get_listings_by_type to count the number of listings per job type.

Resources

Date	**Issue Description**	**Location of Resource**	**Result**
4/15/20	Gain understanding of how to utilize api from past example of Homework	Homework #7	Utilized past written code to provide a framework for gaining data from our new api
4/18/2020	Determine correct SQL description to select data from the database	https://www.w3schools.com/sql/exercise.asp?filename=exercise_select3	Used resource to write correct functions to determine coronavirus cases, deaths, and recovered by country and globally
4/19/2020	Read the json data easily	https://jsoneditoronline.org/	Was able to determine how to index into the data to extract what I wanted to examine.

4/19/2020	Selecting data from a table	Homework #8	Was able to understand how I can access data from a table with an identifying attribute and copying it to a separate list
4/19/2020	Creating Visualizations	https://matplotlib.org	Used to create proper visualizations with specific characteristics to compare our data
4/20/2020	Creating Visualizations	MatplotlibFiles.zip	Used the BarPlot-MIGA.py file to help create a side by side bar chart.