README

Ishan Gupta(3034872764), Jacqueline Bohrer(3034749706), Anurag Bhuwania(3034708275)

YouTube video link - **https://youtu.be/pCvVDvYsdyk**

Our final project is intended to track the spread of diseases (based on the settings) using data from Tweets. The project works on three main levels: the retrieval of live data from twitter, the processing of the JSON object (on the basis of the tweet and its geolocation settings) and finally the plotting of data in real time.

For the first part of the project (the retrieval of data) we first had to go through the entire process of requesting an API from Twitter after doing which we got a 'consumer key', a 'consumer secret', an 'access token' and an 'access token secret' using with a class of StreamListener which we got from "tweepy.Streaming". However, initially we had been using static data, but soon realised our mistake and had to understand how to retrieve a stream of data instead. The data we streamed came in the form of JSON objects which brought us to the next part of the project.

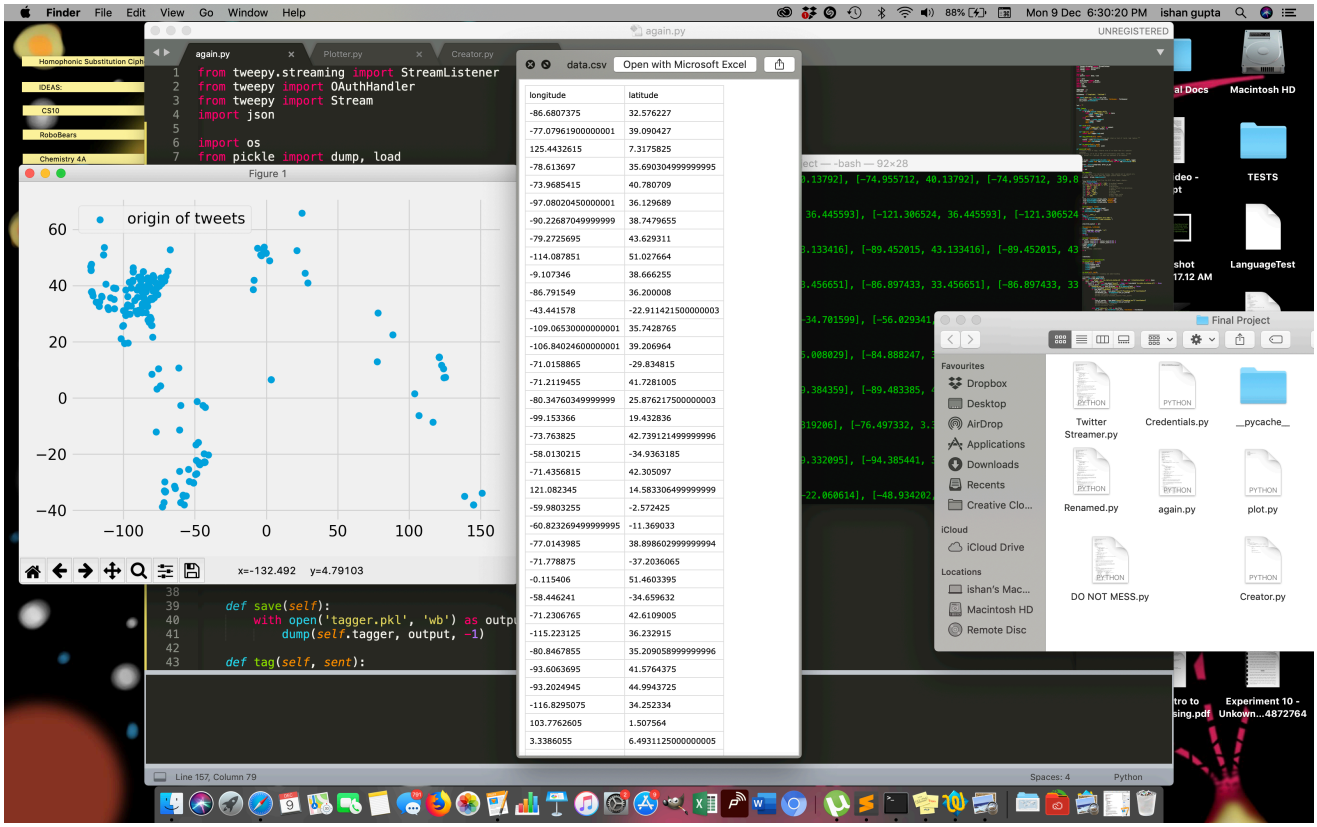
For the second part of the project, we had to work with the JSON object. Initially when we tried to extract data from the JSON, we got an error. Soon, we realised that twitter sent the JSON object in the form of a string and hence we converted it back into a JSON to work with it. Once this was done we analysed big chunks of data to understand what information we needed (shown below) . We also learn that JSON objects worked like dictionaries and contained multiple dictionaries with them. Our knowledge of dictionaries obtained in labs made it a lot easier to deal with the information.

Shown below is an example of the JSON objects we received:

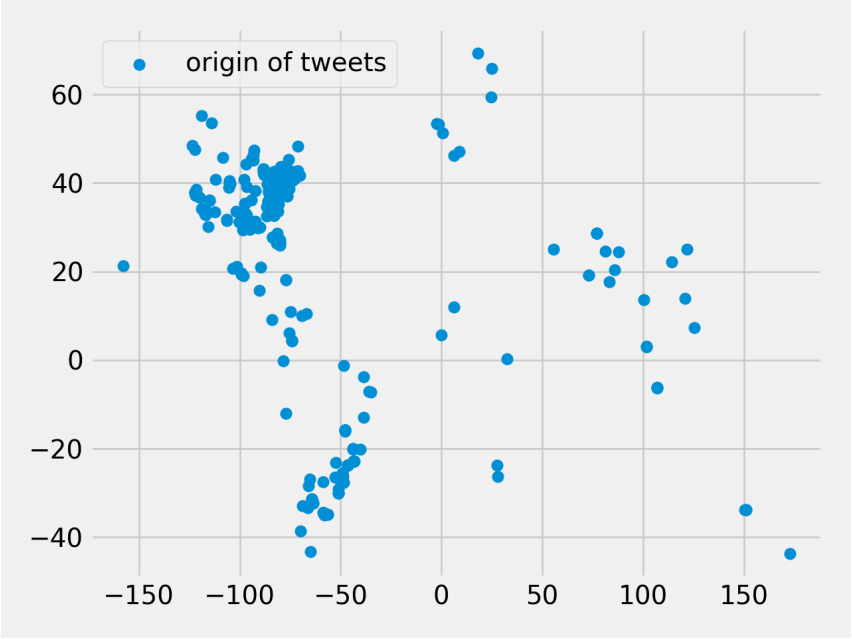
{"created_at":"Tue Dec 10 06:31:49 +0000 2019","id":1204287585683755008,"id_str":"1204287585683755008","text":"RT @ahmovibe: A visitinha aqui \u00e9 constante n\u00e9? Sinta-se avontade!!! \ud83d\ude18\ud83d\ude05","source":"\u003ca href=\"http:\/\/twitter.com\/download\/iphone\" rel=\"nofollow\"\u003eTwitter for iPhone\u003c\/a\u003e","truncated":false,"in_reply_to_status_id":null,"in_reply_to_status_id_str":null,"in_reply_to_user_id":null,"in_reply_to_user_id_str":null,"in_reply_to_screen_name":null,"user":{"id":3633218536,"id_str":"3633218536","name":"\ud83c\udf38","screen_name":"thaianesouza99","location":null,"url":"https:\/\/www.instagram.com\/thaianesoouza\/","description":"\ud83d\udccdRio de Janeiro","translator_type":"none","protected":false,"verified":false,"followers_count":388,"friends_count":214,"listed_count":0,"favourites_count":6282,"statuses_count":18546,"created_at":"Sat Sep 12 16:53:11 +0000 2015","utc_offset":null,"time_zone":null,"geo_enabled":true,"lang":null,"contributors_enabled":false,"is_translator":false,"profile_background_color":"C0DEED","profile_background_image_url":"http:\/\/abs.twimg.com\/images\/themes\/theme1\/bg.png","profile_background_image_url_https":"https:\/\/abs.twimg.com\/images\/themes\/theme1\/bg.png","profile_background_tile":false,"profile_link_color":"1DA1F2","profile_sidebar_border_color":"C0DEED","profile_sidebar_fill_color":"DDEEF6","profile_text_color":"333333","profile_use_background_image":true,"profile_image_url":"http:\/\/pbs.twimg.com\/profile_images\/1189266394967367681\/bvElegwN_normal.jpg","profile_image_url_https":"https:\/\/pbs.twimg.com\/profile_images\/1189266394967367681\/bvElegwN_normal.jpg","profile_banner_url":"https:\/\/pbs.twimg.com\/profile_banners\/3633218536\/1575553142","default_profile":true,"default_profile_image":false,"following":null,"follow_request_sent":null,"notifications":null},"geo":null,"coordinates":null,"place":null,"contributors":null,"retweeted_status":{"created_at":"Mon Dec 09 21:27:10 +0000 2019","id":1204150521029939200,"id_str":"1204150521029939200","text":"A visitinha aqui \u00e9 constante n\u00e9? Sinta-se avontade!!! \ud83d\ude18\ud83d\ude05","source":"\u003ca href=\"http:\/\/twitter.com\/download\/android\" rel=\"nofollow\"\u003eTwitter for Android\u003c\/a\u003e","truncated":false,"in_reply_to_status_id":null,"in_reply_to_status_id_str":null,"in_reply_to_user_id":null,"in_reply_to_user_id_str":null,"in_reply_to_screen_name":null,"user":{"id":3338976257,"id_str":"3338976257","name":"S\u00e3 A VERDADE \ud83d\ude18","screen_name":"ahmovibe","location":"Bangu, Rio de Janeiro","url":null,"description":"* ATIVE AS NOTIFICA\u00e7\u00f5ES !! Parceria via : DM","translator_type":"none","protected":false,"verified":false,"followers_count":109830,"friends_count":165,"listed_count":26,"favourites_count":4813,"statuses_count":2993,"created_at":"Sun Jun 21 13:09:08 +0000 2015","utc_offset":null,"time_zone":null,"geo_enabled":true,"lang":null,"contributors_enabled":false,"is_translator":false,"profile_background_color":"C0DEED","profile_background_image_url":"http:\/\/abs.twimg.com\/images\/themes\/theme1\/bg.png","profile_background_image_url_https":"https:\/\/abs.twimg.com\/images\/themes\/theme1\/bg.png","profile_background_tile":false,"profile_link_color":"1DA1F2","profile_sidebar_border_color":"C0DEED","profile_sidebar_fill_color":"DDEEF6","profile_text_color":"333333","profile_use_background_image":true,"profile_image_url":"http:\/\/pbs.twimg.com\/profile_images\/1086276139717251073\/Jm2QY05q_normal.jpg","profile_image_url_https":"https:\/\/pbs.twimg.com\/profile_images\/1086276139717251073\/Jm2QY05q_normal.jpg","profile_banner_url":"https:\/\/pbs.twimg.com\/profile_banners\/3338976257\/1572977644","default_profile":true,"default_profile_image":false,"following":null,"follow_request_sent":null,"notifications":null},"geo":null,"coordinates":null,"place":null,"contributors":null,"is_quote_status":false,"quote_count":30,"reply_count":0,"retweet_count":312,"favorite_count":345,"entities":{"hashtags":[],"urls":[],"user_mentions":[],"symbols":[]},"favorited":false,"retweeted":false,"filter_level":"low","lang":"pt"},"is_quote_status":false,"quote_count":0,"reply_count":0,"retweet_count":0,"favorite_count":0,"entities":{"hashtags":[],"urls":[],"user_mentions":[{"screen_name":"ahmovibe","name":"S\u00e3 A VERDADE \ud83d\ude18","id":3338976257,"id_str":"3338976257","indices":[3,12]}],"symbols":[]},"favorited":false,"retweeted":false,"filter_level":"low","lang":"pt","timestamp_ms":"1575959589665"}

After this, the last and probably the hardest part of the project was figuring our a way to update our graph in real-time. For the purpose of plotting we were using MatPlotLib. And our initial goal was to update the graph using the show() function, however, we learnt that the show() function actually blocks the entire script from running and hence we had to think of an alternate way. After trying everything from switching off the block (in which case the graph just didn't show up) to trying to import data into another script every-time new data was obtained, we finally found our solution in creating a csv file and storing each new datum as a row in the csv file and then running a real time

grapher that read this file every certain amount of time. Finally, we obtained results that looked something like this:



And it gave us a plot that looked like this after 20 minutes of updating.



Finally, the program ran on three scripts. The first was the credentials script which contained all the access keys etc give to us by twitter. The second was the main script that streamed the data, used the language processing feature, created the csv file, dealt with the JSON and finally saved the relevant data in the csv file. The third and final was the script that plotted the updated data from the csv file. Therefore **to run the program** all three scripts should be in the first folder firstly. And then, the

main (2nd script) should be run through terminal and the 3rd and final script should be run using the Build feature under 'Tools' on Sublime text. There may be other ways of running it but this one worked best for us.

Name of Function (4 main)	**Domain (inputs)**	**Range (outputs)**	**Bheaviour**
on_data	(self, data) - data (from twitter) and because its a method in a class - self	nothing (but on the script True)	It is the method of the StdOutListener Class that streams the data and receives it. and instead of returning the data, we decided to process it in the function itself.
clean_data	(json_data) - it takes in the JSON format data after being converted from string	either the location of the tweet (if it meets the passive/active voice parameters, other filters) or "no".	The clean_data has several conditions that ensure that the data is worth considering and also it filters out the data in the process to give us the coordinates of the tweets origin finally
save	(list_long_lat) - it takes in a list of longitude, latitude	It opens the csv file named "Data.csv" and writes down the new data in it in the correct columns. It also prints the longitude and latitude to indicate updates	prints the longitude and latitude out. And it also opens the csv file and saves the data in it.
triangulate	(coordinates) - twitter gives 4 sets of coordinates to indicate an approximate location within them. Hence they were processed in this function.	It outputs the average of the four coordinates (square on the map) to determine the approximate location of the origin of the tweet.	Adds all values to the longitude and latitude lists and also the final coordinates that are plotted on the map.

In the following project, lists and dictionaries were an essential part! We dealt with dictionaries (in the form of JSON) extensively to filter out the data we needed. Additionally, lists were used to store all the valid longitudes and latitudes and also to input data into the rows for csv files. Additionally, while creating the plot, we sent the information into the csv by creating a dictionary for longitude

and latitude. Local variables were also used extensively within the functions to make sure they were working.
Lists/ Dictionaries (where they are in the code): triangulate() - Function - Lists; save() - Function - Lists; clean_data() - Function - Dictionaries and Lists. Plotter.py - Script - Dictionaries (for extracting longitude/latitude from csv for plotting)

Global Variables: LONGITUDES [], LATITUDES [] and json_data = 0

local script variables: save(**list_long_lat**); triangulate(**coordinates**); on_data(self, **data**)

For the purpose of this project, we used python 3.6.5 and the list of libraries we used along with their links is below:

To download these on Macs - you can run "Sudo pip install *package name*"

tweepy - https://github.com/tweepy/tweepy
json - https://docs.python.org/3/library/json.html
csv - https://docs.python.org/3/library/csv.html
time - https://docs.python.org/3/library/time.html
os - https://docs.python.org/3/library/os.html
pickle - https://docs.python.org/3/library/pickle.html
nltk - https://docs.python.org/3/library/json.html
nltk.corpus - https://www.nltk.org/api/nltk.corpus.html
matplotlib - https://pypi.org/project/matplotlib/
pandas - https://pandas.pydata.org

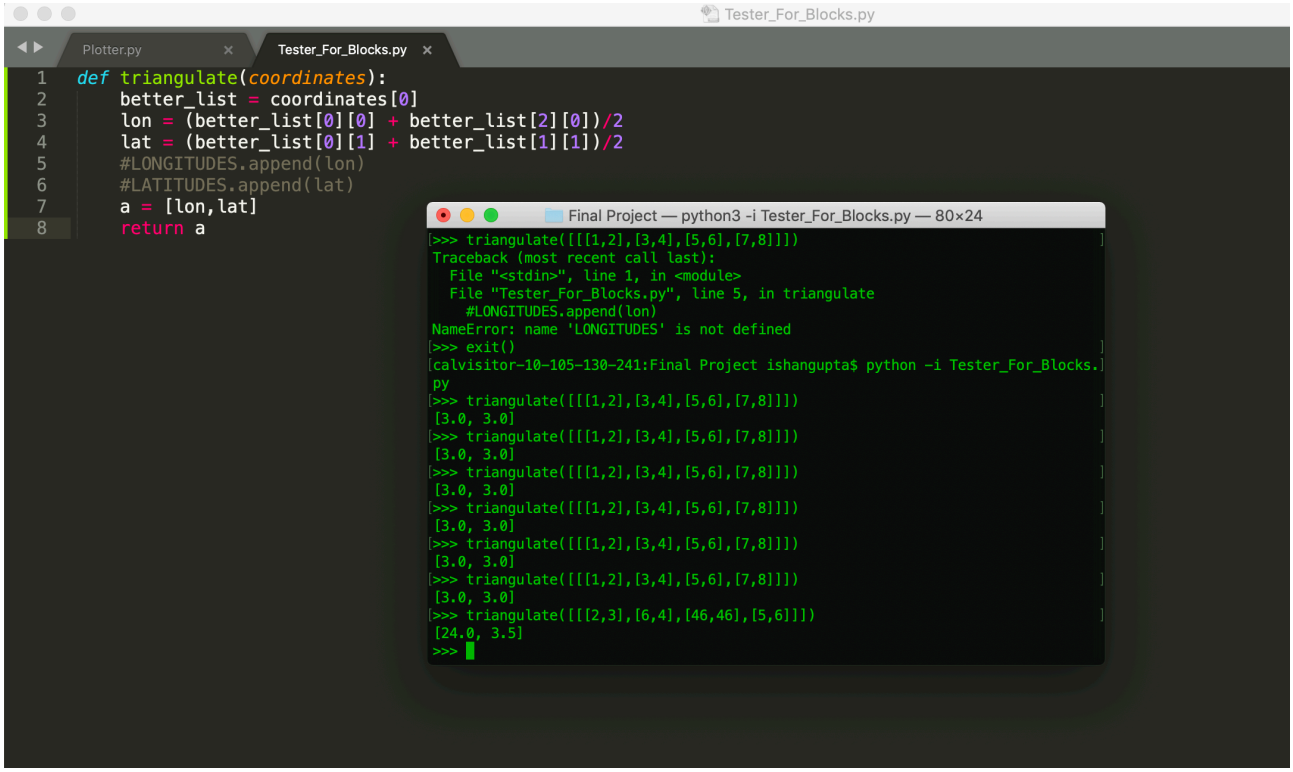
In addition to all this, we had a feature that tested for passive and active voice between lines 24 and 98. We made changes to the function to understand it in our own way as we learnt how to make it. However, we did not understand all of it in totality as is used traces of machine learning which was way beyond your scope. However, after talking to several 61A kids we figured out what we were doing. Using the nltk library, and the brown data set a neural network was trained to differentiate between passive and active sentences. However, once again, while we tried our very best to understand this code and also looked for simpler solutions, we wanted the most effective way to work with out data which in this case happened to be much beyond the scope of our class. But, we genuinely learnt a lot.

The link from where we referred to the code was - https://github.com/cowlicks/ispassive/blob/master/ispassive.py
and where we understood it was from - https://www.youtube.com/watch?v=5ctbvkAMQO4 , And a lot of CS61A kids. https://www.youtube.com/watch?v=fOvTtapxa9c

As mentioned above, to run the scripts properly, it is recommended that the file "**IshanGupta_JacquelineBohrer_AnuragBhuwania_Main_Script.py**" is run through terminal while the other is built from the text editor.

Additionally, we tested all blocks but the ones shown was the easiest to test outside the entire script. The average ion the polygon was taken in a unique way because it was realised that twitter gave

coordinates in a square. Thus making it easier to calculate the average. We also noticed that twitter gave it in the form coordinates = [[long1, lat1], [long1, lat2], [long2, lat1], [long2, lat2]]
Therefore the algorithm (as can be sen in the function) was built to average long1 (coordinates[0][0]) and long2 (coordinates[2][0]) AND lat1 (coordinates[0][1]) and lat2 (coordinates[1][1]). For this reason the numbers in the example do not yield the actual average but only the average assuming it was in the same format stated above.



EXTRA CREDIT:

We think we deserve extra credit because our project at its core is extremely useful. Unlike games and simulators (which we also considered making) our project has real world application and can be used to track diseases and allow predictions to be made by whomsoever may want to use the program. To begin with, we were all very new to webscraping and the concept of downloading information from the internet was hard to grasp. But, using Tweepy and gaining an API after requesting it (which took a lot of effort to understand in totality) we felt much more confident to us coding for real world application. Additionally, through the twitter API and streaming features we were forced to use/understand OOP (classes and methods) and also work with JSON objects (which we now know closely resemble dictionaries). Additionally, the hardest part for us was figuring out the real-time data plotting for which we had to use the animation feature within matplotlib. This also require us to write all the data into a csv file (and understanding how that worked) and then plotting it as it updated. In the process of this project we were also introduced to nltk through which we learnt a lot about natural language processing and machine learning, however, that code was not ours although we did adapt it at points and understand how to use it. Lastly, we were introduce to a vast variety of new topics, coding methods, techniques, concepts and so much more! It was really hard to grasp all of it quickly and apply it, but our teamwork made it possible. We feel like we have created something of value and hopefully something of use for the future. Overall, we enjoyed every stage of this project immensely and would love to take this project much further with better tweet filtration using nltk again.

Citations:

http://docs.tweepy.org/en/latest/

https://github.com/CoreyMSchafer/code_snippets/blob/master/Python/Matplotlib/09-LiveData/finished_code.py

https://stackoverflow.com/questions/4098131/how-to-update-a-plot-in-matplotlib

https://pythonspot.com/matplotlib-update-plot/

https://pythonprogramming.net/python-matplotlib-live-updating-graphs/

https://gist.github.com/rlabbe/ea3444ef48641678d733

https://block.arch.ethz.ch/blog/2016/08/dynamic-plotting-with-matplotlib/

https://github.com/cowlicks/ispassive/blob/master/ispassive.py

https://www.youtube.com/watch?v=fOvTtapxa9c

https://www.youtube.com/watch?v=5ctbvkAMQO4

https://towardsdatascience.com/extracting-twitter-data-pre-processing-and-sentiment-analysis-using-python-3-0-7192bd8b47cf

https://www.youtube.com/watch?v=wlnx-7cm4Gg&t=760s