**README**

To conduct the project follow this instruction.

Software required:

1. Python
2. R
3. Kafka
4. MongoDB
5. Tableau

**STREAMING TWEET**

The first part of the project requires the streaming of the tweets, in particular it is necessary to receive the Consumer Key (API Key), Consumer Secret(API Secret), Access Token and Access Token Secret. In order to receive them apply to the following link: <https://developer.twitter.com/en>

Install MongoDB before running the codes in order to save directly in the collection the tweets streamed.

Once you have the keys, install Kafka and follow the ‘info for shell kafka.txt’ file in the codes ->streaming tweet folder.

After that you will have your Kafka and Zookeper running, run the python code ‘Twitter streaming.py’ in codes ->streaming tweet folder.

The code is implemented to be automatic, when you reach the volume of documents desired stop the process.

**REAL TIME DASHBOARDS**

To create the real time dashboards you need to use Python as primary software.

To build the Sentiment Analysis Graph, just need to run the ‘sentiment.py’ code in Data Management->codes->Creating Interactive Dash->Sentiment Graph folder.

It will automatically stream the tweets and analyze them through the sentiment analysis algorithm, once you stream them, plot them running on the shell the ‘sent\_graph.py’ file in the same folder.

Copy and paste the locally url shown in the shell.

To build the Real Time Tweet Map, first import Flask, after that just run the ‘frontend.py’file in Data Management->codes->Creating Interactive Dash->Real Time Map folder.

Pay attention to modify your KafkaClient host (generally it use the port 9092 locally, change it if you are using clouds).

Copy and paste the locally url shown in the shell to check the map.

**ANALYSIS**

The first part of analysis is totally used in R language on the TwitterCollection dataset using the code in Data Management->codes->PreProcessing and Analysis->Script\_R\_CANTSLEEP.R while the second is carried out in Python language on the 56 collections on the site: <https://github.com/mykabir/COVID19>.

For the second side just follow the script in Data Management->codes->PreProcessing and Analysis->Script\_Python\_CANTSLEEP.py

Delete duplicates and missing values ​​and clean the data , then apply sentiment analysis to the text of the Tweets and eliminate the missing locations. Enter the timezone name from the “timestate” dataset and compare the location with the cities and states in the same collection, then removing the unclassified ones.

Convert the UTC date to local dates with the help of the timezone name and subdivide them into night (00:00 -> 5:59) and day (6:00 -> 23:59).

Standardize the frequencies of the tweet volumes for each state and night/day and in parallel add the polarity variable obtained in real time.

Group all the collections into one and add the timezone name variable from the “timestate” dataset; then remove the excessive or useless states such as: not US territory (XX), Puerto Rico (PR), Guam (GU), Jakarta (ID), Germany (DE), Northern Mariana Islands (MP), Hawaii (HI) and Alaska (AK).

Turn the date into local datetime and separate night from day as explained above and finally standardize the volume of Tweet by state and by day / night.

Another analysis consists to mapping the network of retweet. For this part are necessary three datasets: the first containing six field sof the full collection, the other two useful for the geolocation of the tweets. For this analysis you need to use many R’s library: reader, diplyr, tidyr, lubridate, twitteR, devtools, plotly and wordcloud. After running the whole script you will obatain two datasets, one for the edges, the other for the nodes. These two serves to build the network in Gephi. After all the network was exported in a pdf file.

The analysis of fake news needs the same dataset of the precedent analysis, the one with six field of full collection and consists to truncate the text of the tweet to search references to three categories of fake news that we are considered. At the end the tweets have been grouped for day and country.