Jagjot Singh 700036486

Introduction to Data Science
Twitter Data Analysis

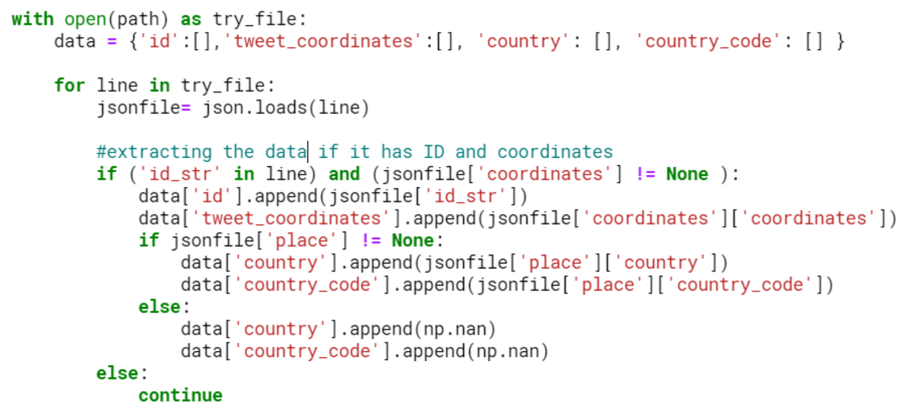
The data, fetched using Twitter API, ranges between the dates, March 1st, 2020 and March 31st ,2020 contains all the tweets during these day in the European region. The data is 92.3 GB in size after unzipping which introduces challenges due to limited RAM size at disposal. The analysis of data is divided into 4 parts, each trying to find different facets of the data to extract more information from data. This report will go through the methods used to analyse the data as large as it is, it will also show the pre-processing process done before the data could be peered into and will give my reflection on the analysis results and data.

Preprocessing and Reduction of the json files

The data used for the analysis was first uploaded in Kaggle and all the further coding was done on a Kaggle notebook, which provided the flexibility in storage but limited the available RAM to 16GB. The first step taken to reduce the data was to understand the data and json tags and selecting only the values which will be needed for the analysis.

The 9 selected tags are: **ID, created_at, text, lang, coordinates, place.country, place.country_code, user.screen_name, entities.user_mentions.**

Since the space was not an issue, to further reduce the size during performing the operation the tags were grouped in smaller 4 json file groups of 31 files (a file for each day), each group for each task in the coursework. For example: For task 2 json files were generated with tags **ID, coordinates, place.country and place.country_code** for only the lines which contained a value for coordinate tag. Once the data was extracted it was collected in a data frame to perform further analysis



```
with open(path) as try_file:
    data = {'id':[],'tweet_coordinates':[], 'country': [], 'country_code': [] }

    for line in try_file:
        jsonfile= json.loads(line)

        #extracting the data if it has ID and coordinates
        if ('id_str'  in line) and (jsonfile['coordinates'] != None ):
            data['id'].append(jsonfile['id_str'])
            data['tweet_coordinates'].append(jsonfile['coordinates']['coordinates'])
            if jsonfile['place'] != None:
                data['country'].append(jsonfile['place']['country'])
                data['country_code'].append(jsonfile['place']['country_code'])
            else:
                data['country'].append(np.nan)
                data['country_code'].append(np.nan)
        else:
            continue
```

Code snippet 1 - Data Extraction

Dealing with duplicates

Each ID has been taken as a unique tweet; this rule comes with acceptance of certain conditions and accepting and rejecting data based on this assumption. The conditions for acceptance of tweets are:

- The same tweets done by the same user multiple times will be considered as multiple tweets if tweet IDs are different. Which means it will cover automated or casual activities.
- The same tweets done by the different users will be considered different tweets, it could be bots, avid followers, or retweets, which shows the real-world activity. The texts will not be tinkered with but in Task 4 to get a clearer sentiment.

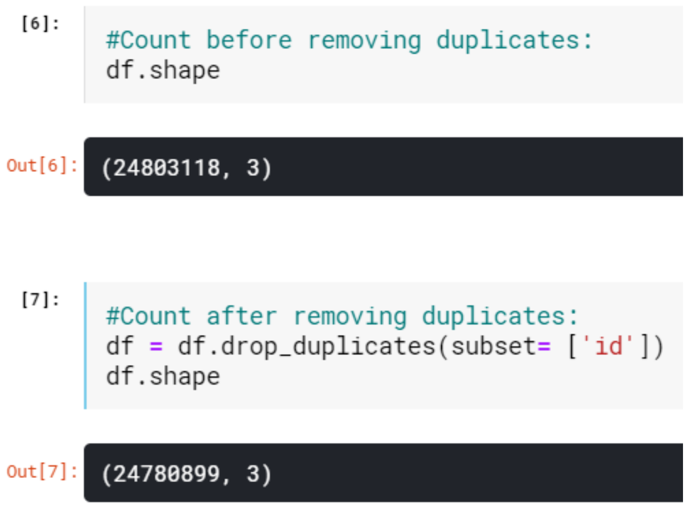
The possible duplicates which are rejected are:

- The tweets which are with same IDs.
- And the tweets with no ID at all have not been included in the count and have been considered as *anomalies* in the data.

Part 1: Basic Stats

Total number of tweets

The tasks in this part pertains with forming a timeseries and language usage analysis of the region. The data with tags **ID, created_at** and **lang** is extracted and is fed into a pandas dataframe. The duplicates in the data for repeated IDs are removed and total number of tweets after removing 22219 duplicate entries came out to be **24,780,899**.



```
[6]:
#Count before removing duplicates:
df.shape
```

```
Out[6]: (24803118, 3)
```

```
[7]:
#Count after removing duplicates:
df = df.drop_duplicates(subset= ['id'])
df.shape
```

```
Out[7]: (24780899, 3)
```

Code Snippet 2 – Total number of tweets

Time series of number of tweets by day

The number of tweets were calculated by extracting dates from **created_at** column and using group by method on it. The resulting table showed data with counts of **ID** rows against each date.

Comments: The graph for the data shows the biggest surge in tweet count on 12th March 2020, after which the number of tweets has been higher than the month average. This tells that some event with a long and continuous engagement could have happened around that time. This coincides with one of the first cases of covid-19 coming to the Europe and continuous activity as many countries in the region toughened the stance on the response and announced restrictions and lockdowns.

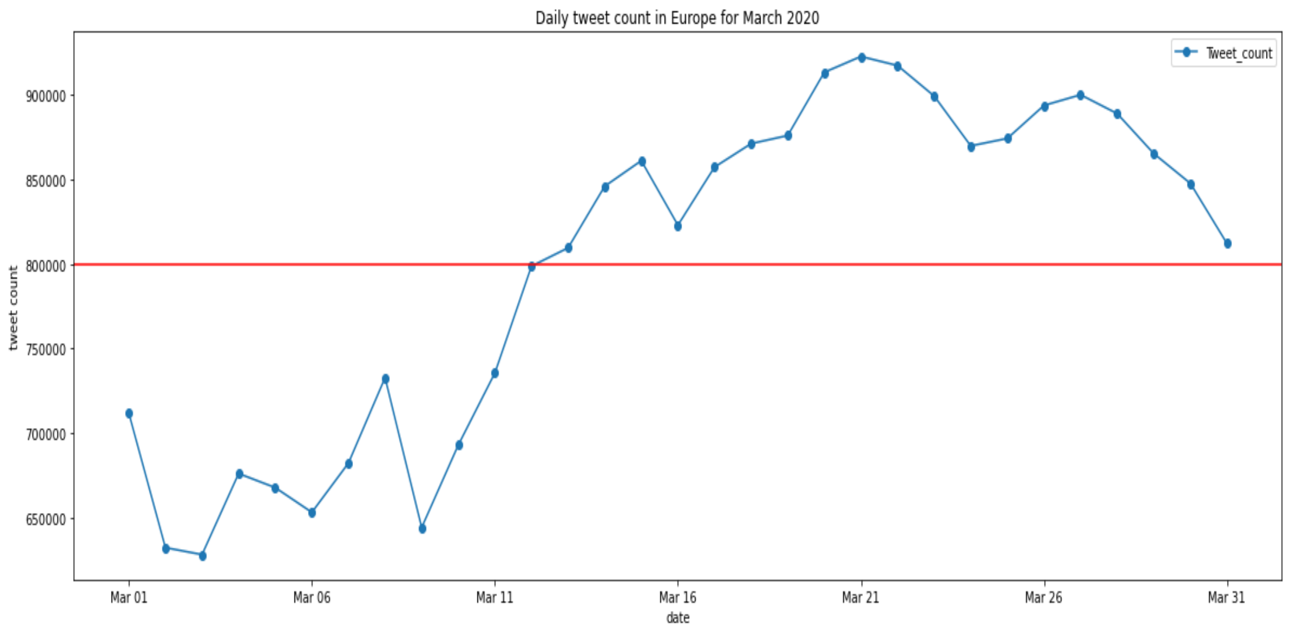
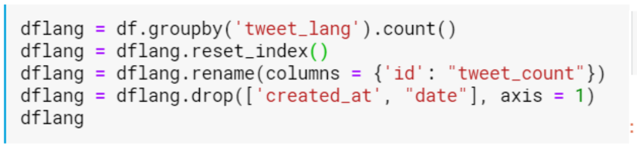


Figure 1 – Number of tweets per day in Europe during March 2020

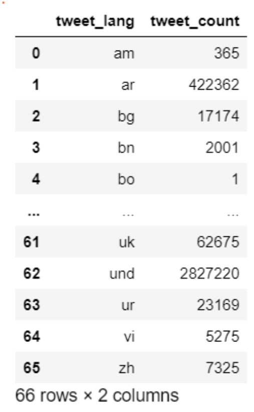
Number of Tweets by Language

Finding number of tweets by language will require the same approach as finding the tweets by date. This time the group by action will be performed on **lang** column.

According to twitter documentation twitter recognises 70 languages. These languages will be mapped against their encoder and will be translated to human recognisable words. Also, the language code '*und*' is used for tweets which are not understandable or just contain symbols, emojis or tags. BCP47 library can be imported to translate understandable language names from the encoded tags.

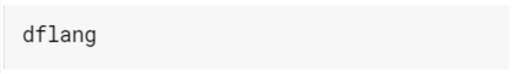


```
dflang = df.groupby('tweet_lang').count()
dflang = dflang.reset_index()
dflang = dflang.rename(columns = {'id': "tweet_count"})
dflang = dflang.drop(['created_at', "date"], axis = 1)
dflang
```

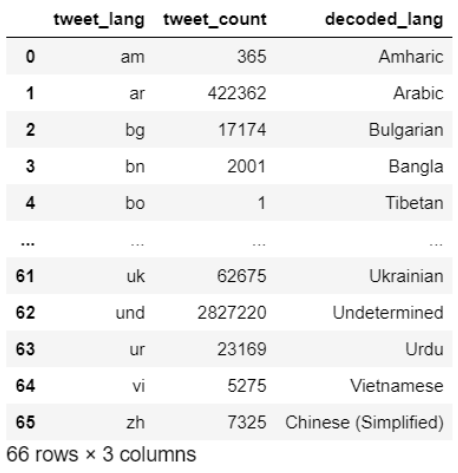


	tweet_lang	tweet_count
0	am	365
1	ar	422362
2	bg	17174
3	bn	2001
4	bo	1
...	...	...
61	uk	62675
62	und	2827220
63	ur	23169
64	vi	5275
65	zh	7325

66 rows × 2 columns

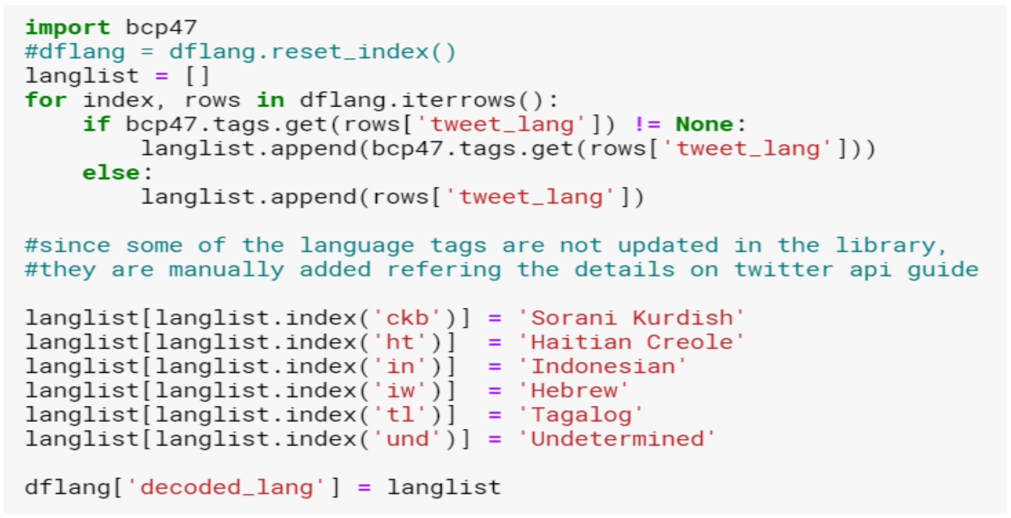


```
dflang
```



	tweet_lang	tweet_count	decoded_lang
0	am	365	Amharic
1	ar	422362	Arabic
2	bg	17174	Bulgarian
3	bn	2001	Bangla
4	bo	1	Tibetan
...	...	...	...
61	uk	62675	Ukrainian
62	und	2827220	Undetermined
63	ur	23169	Urdu
64	vi	5275	Vietnamese
65	zh	7325	Chinese (Simplified)

66 rows × 3 columns



```
import bcp47
#dflang = dflang.reset_index()
langlist = []
for index, rows in dflang.iterrows():
    if bcp47.tags.get(rows['tweet_lang']) != None:
        langlist.append(bcp47.tags.get(rows['tweet_lang']))
    else:
        langlist.append(rows['tweet_lang'])

#since some of the language tags are not updated in the library,
#they are manually added refering the details on twitter api guide

langlist[langlist.index('ckb')] = 'Sorani Kurdish'
langlist[langlist.index('ht')]  = 'Haitian Creole'
langlist[langlist.index('in')]  = 'Indonesian'
langlist[langlist.index('iw')]  = 'Hebrew'
langlist[langlist.index('tl')]  = 'Tagalog'
langlist[langlist.index('und')] = 'Undetermined'

dflang['decoded_lang'] = langlist
```

Code snippet 3 – Decoding bcp 47 language codes

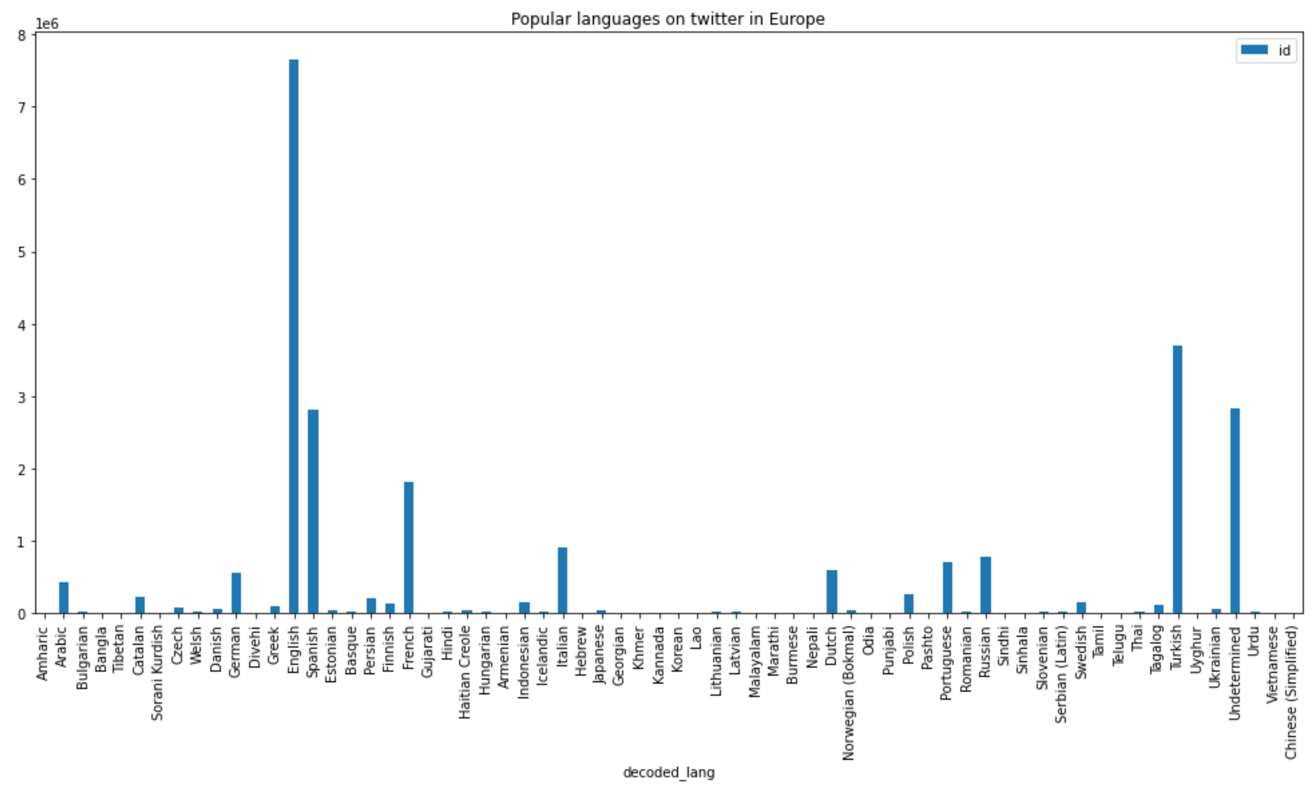


Figure 2 – Number of tweets by language

Comments: The bar chart for the language data shows that English is the most popular language used on twitter in European region followed by Turkish, which also makes sense because of the larger population of the country than most European countries. However, it is surprising to see small number of tweets in German. It is also worth noticing that many tweets in ‘undetermined’ languages, this could mean a behaviour of tagging people or posting just emojis as tweets, possibly in replies.

Comparisions between the number of tweets during weekday and weekends

To finding the data for weekday and weekends similar method as above analysis has been used hence to avoid being redundant code snippets will not be posted for the same. Using the same dataframe as used for finding tweets count on each day and using **created_at** column weekday and weekend can be characterised. Plotting the data using 'seaborn' will give figure 3.

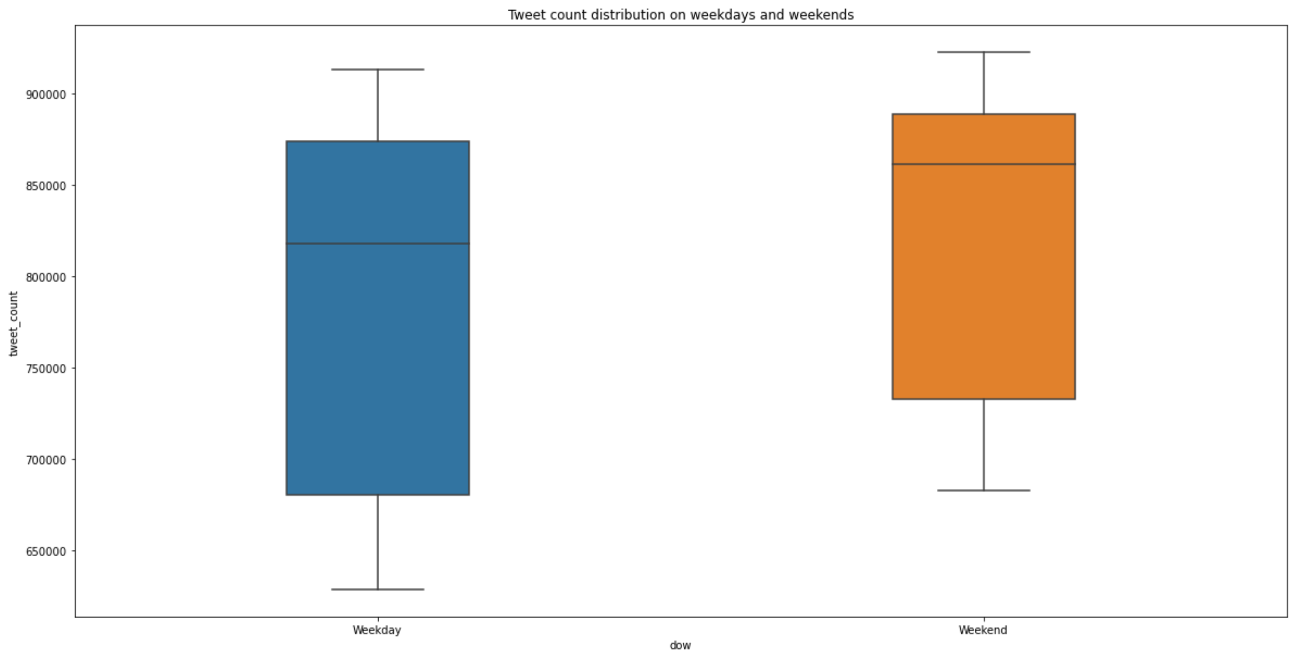


Figure 3 – Box and Whisker plot for tweet count on Weekdays vs Weekends

Comments: It can be observed that a greater number of tweets are posted during weekends than during weekdays. Both graphs are negatively skewed (i.e., smaller upper quartiles) showing that there have been short bursts of high volume of tweets in the count.

Average number of tweets each hour

The data is extracted using a similar method as in the above sections and the plotted bar graph is figure 4.

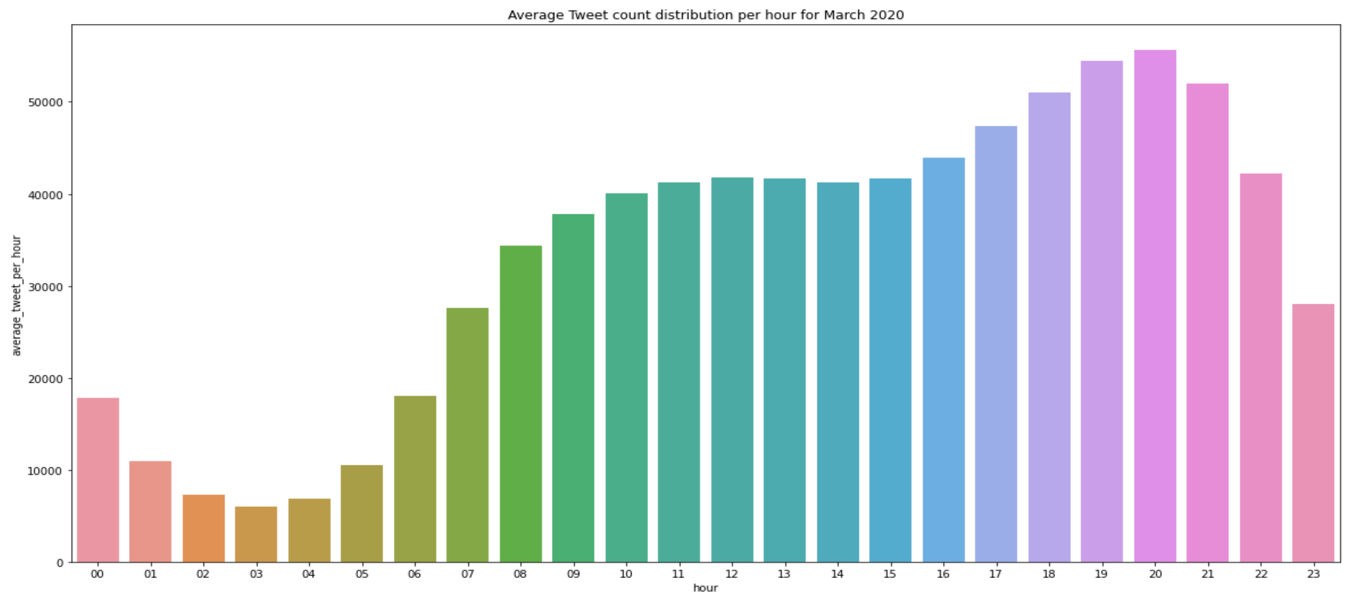


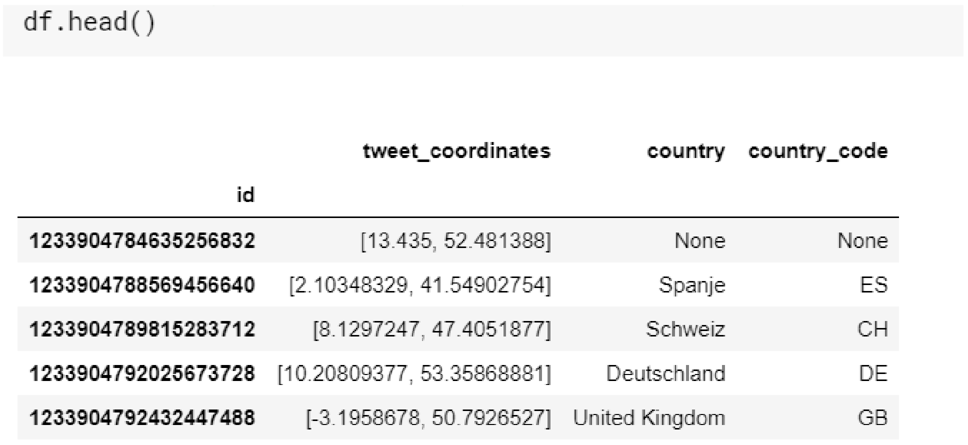
Figure 4 – Number of tweets each hour of the month

Comments: Number of tweets per hour peak during the evening between 5 pm and 10 pm. This also aligns with the daily routine of working men and women; most tweets are posted after office hours. We can also observe the steep depression in the count during wee hours which again indicates the obvious, that majority of users are sleeping.

Part 2: Plotting the data on a map: Choropleth

Preprocessing Data

The extraction of data has been explained in the example in data reduction section, moving ahead with pre-processing data. After removing duplicates and with already small selection of data for which **coordinates** tag in json file is *not null* the total number of rows come out to be 1403924 and the table looks like as below.

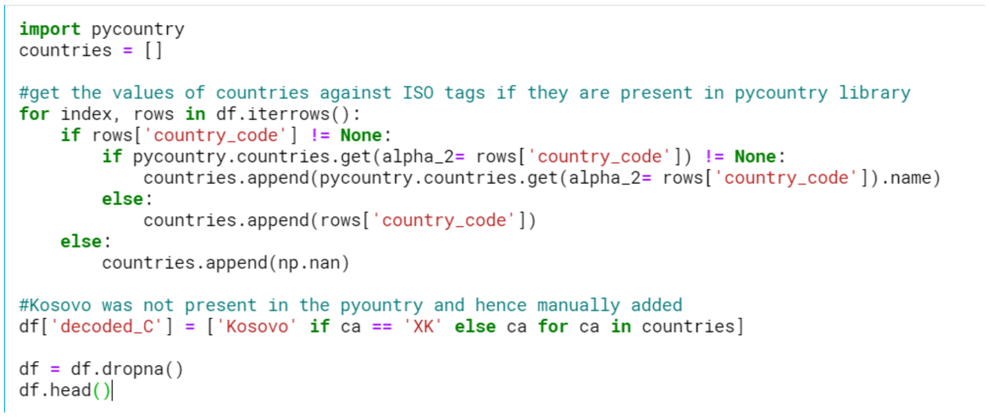


```
df.head()
```

id	tweet_coordinates	country	country_code
1233904784635256832	[13.435, 52.481388]	None	None
1233904788569456640	[2.10348329, 41.54902754]	Spanje	ES
1233904789815283712	[8.1297247, 47.4051877]	Schweiz	CH
1233904792025673728	[10.20809377, 53.35868881]	Deutschland	DE
1233904792432447488	[-3.1958678, 50.7926527]	United Kingdom	GB

Table 1 – Map plotting data

To make a choropleth map we need a geo-json file which contains the multi-polygons of the countries in in the data and mapping the names or the codes of the country with data we have extracted from the json to manage colour based on the number of the tweets in each country. As can be seen in Table 1, the names of the countries can be in local languages and ISO country codes are not of the same format as RFC7946 specifications that is used in geo-json. So, we will use the ISO codes to get the country names in English (Code snippet - 4).



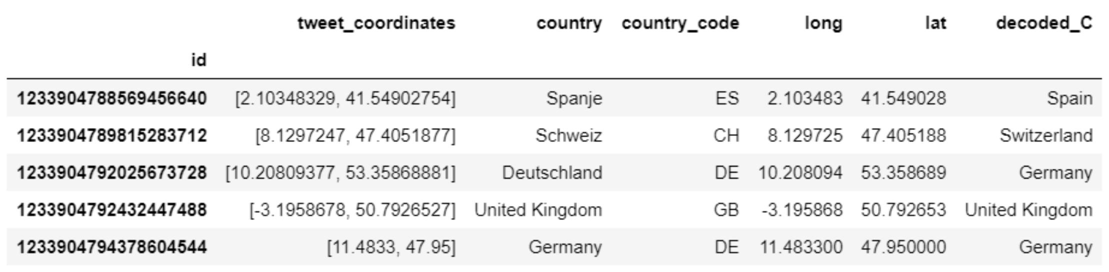
```
import pycountry
countries = []

#get the values of countries against ISO tags if they are present in pycountry library
for index, rows in df.iterrows():
    if rows['country_code'] != None:
        if pycountry.countries.get(alpha_2= rows['country_code']) != None:
            countries.append(pycountry.countries.get(alpha_2= rows['country_code']).name)
        else:
            countries.append(rows['country_code'])
    else:
        countries.append(np.nan)

#Kosovo was not present in the pyountry and hence manually added
df['decoded_C'] = ['Kosovo' if ca == 'XK' else ca for ca in countries]

df = df.dropna()
df.head()
```

Code snippet 4 – Converting country codes to Country names using pycountry



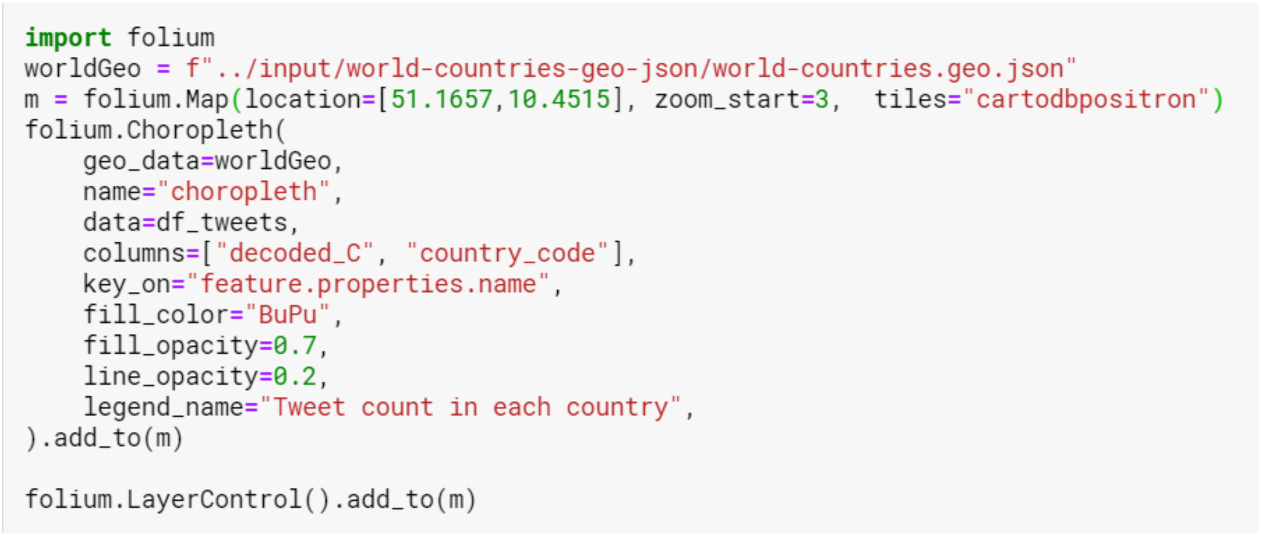
id	tweet_coordinates	country	country_code	long	lat	decoded_C
1233904788569456640	[2.10348329, 41.54902754]	Spanje	ES	2.103483	41.549028	Spain
1233904789815283712	[8.1297247, 47.4051877]	Schweiz	CH	8.129725	47.405188	Switzerland
1233904792025673728	[10.20809377, 53.35868881]	Deutschland	DE	10.208094	53.358689	Germany
1233904792432447488	[-3.1958678, 50.7926527]	United Kingdom	GB	-3.195868	50.792653	United Kingdom
1233904794378604544	[11.4833, 47.95]	Germany	DE	11.483300	47.950000	Germany

Table 2 – Processed map dataframe

Using *Code Snippet 4* and expanding latitude and longitude we get *Table 2* and after that the data is processed it can be used to get count table against the country names to plot a **choropleth** using **Folium**. The table will be made using groupby().count() yet again.

Plotting a choropleth

The dataframe with tweet count against the countries is made and is named ‘df_tweets’. The second set of data, the world map geo json file is uploaded in json after finding it on the internet and is named ‘worldGeo’. I preferred using a geo json file for the world countries because the data doesn’t fall strictly on Europe. Then using ‘folium.map()’ and ‘folium.choropleth()’ functions the map is plotted. As shown in the Code Snippet 5.



```
import folium
worldGeo = f"../input/world-countries-geo-json/world-countries.geo.json"
m = folium.Map(location=[51.1657,10.4515], zoom_start=3,  tiles="cartodbpositron")
folium.Choropleth(
    geo_data=worldGeo,
    name="choropleth",
    data=df_tweets,
    columns=["decoded_C", "country_code"],
    key_on="feature.properties.name",
    fill_color="BuPu",
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name="Tweet count in each country",
).add_to(m)

folium.LayerControl().add_to(m)
```

Code Snippet 5 – Plotting a choropleth using Folium

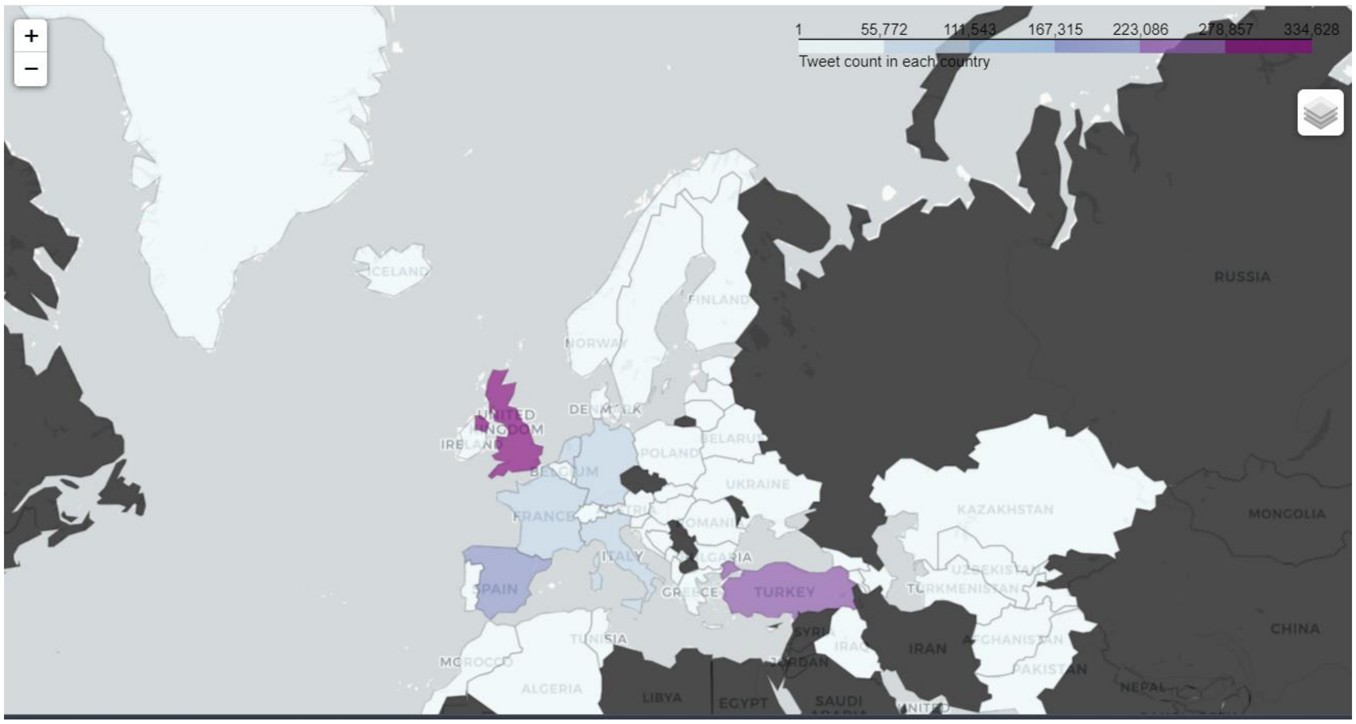


Figure 5 – Choropleth based on number of tweets from a country

Comments: The map shows the in darker colours the higher number of tweets from the coloured countries. It shows that UK, Turkey and Spain are have the most active twitter users in Europe. This claim is validated by Figure 2, which shows English, Turkish and Spanish are most common languages on twitter in Europe. This also shows that in these locations' users are tweeting from mobile devices with their locations services enabled.

Part 3: User Data Analysis

Number of Tweets per user

For user data analysis json data with tags **ID, created_at, user.screen_name, entities.user_mentions** is extracted. To count tweets per user is much like previous task. First, duplicate **ID** data is removed then using groupby() function on usernames and count() function to get number of tweets for each user. In resulting table, we can see the total number of users who used twitter during March 2020 in European region is **1,111,170**. This makes the average number of tweets for a user to be around 24 tweets a month. If we check this against the users in the given data, 83.5% of users post below average. In fact, 52.4% of users posted less than 5 tweets during March 2020. This becomes clearer seeing the count by scaling y-axis by log scale (Figure 6).

```
f, ax = plt.subplots(figsize=(15, 7))
sns.histplot(data = df_usertweets,x = 'Tweet_count', binwidth = 10)
plt.ylabel('Number of users')
plt.title('Number of tweets per user')
plt.xlim(0,2000)
ax.set_yscale('log')
plt.savefig('#3.1_tweets_per_user.png', bbox_inches = 'tight')
```

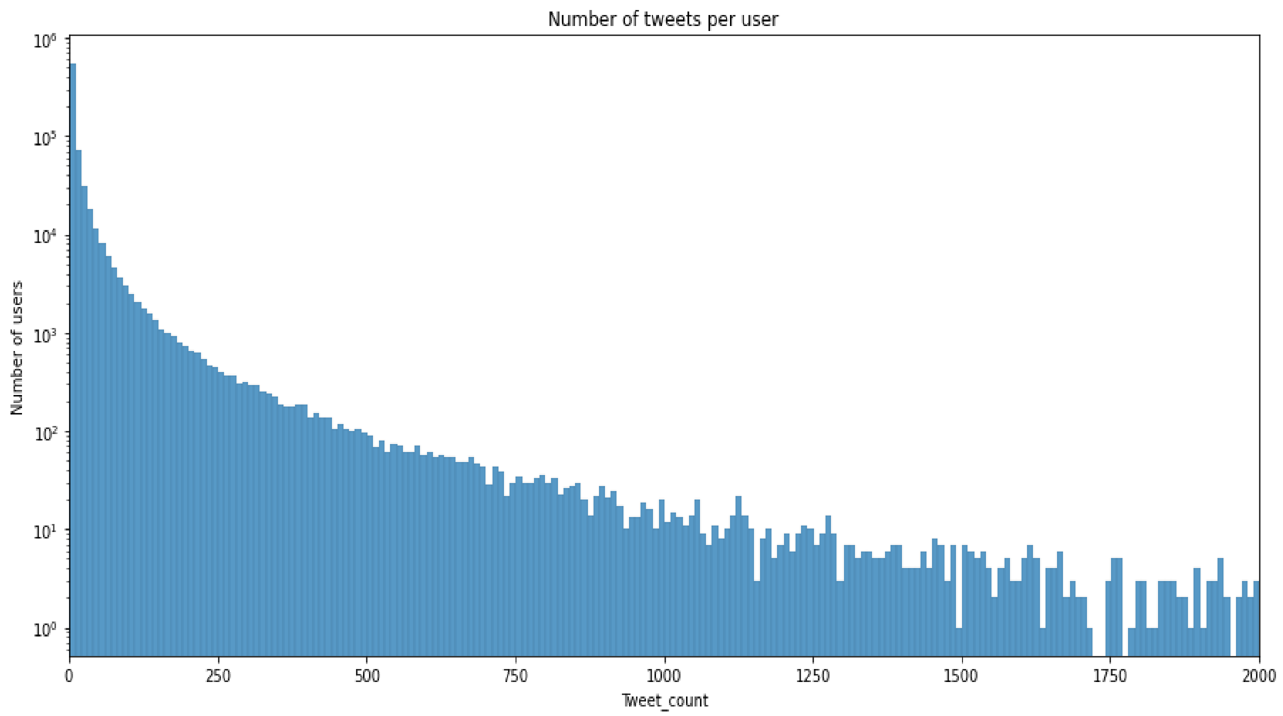
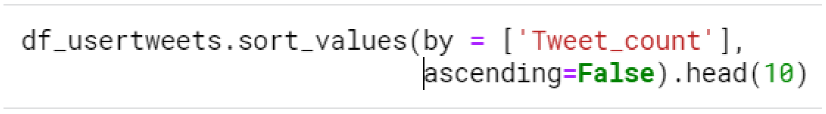


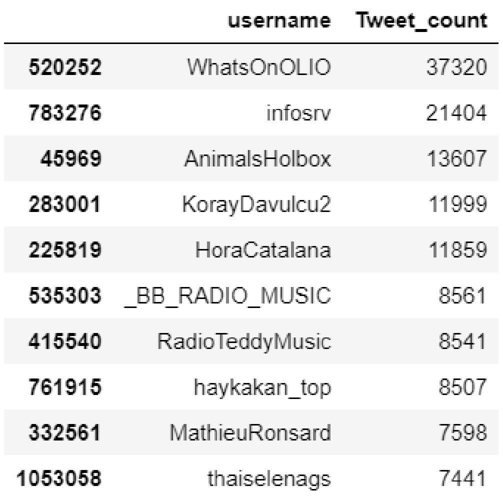
Figure 6 – Number of tweets per user

Most active users

The 10 most active users can be found by using the pandas dataframe used in the earlier part. The users are as shown in Code Snippet 6. Some of these accounts are automated accounts; there is some evidence shared below suggesting that other than the sheer volume of the tweets.



```
df_usertweets.sort_values(by = ['Tweet_count'],
                                  ascending=False).head(10)
```



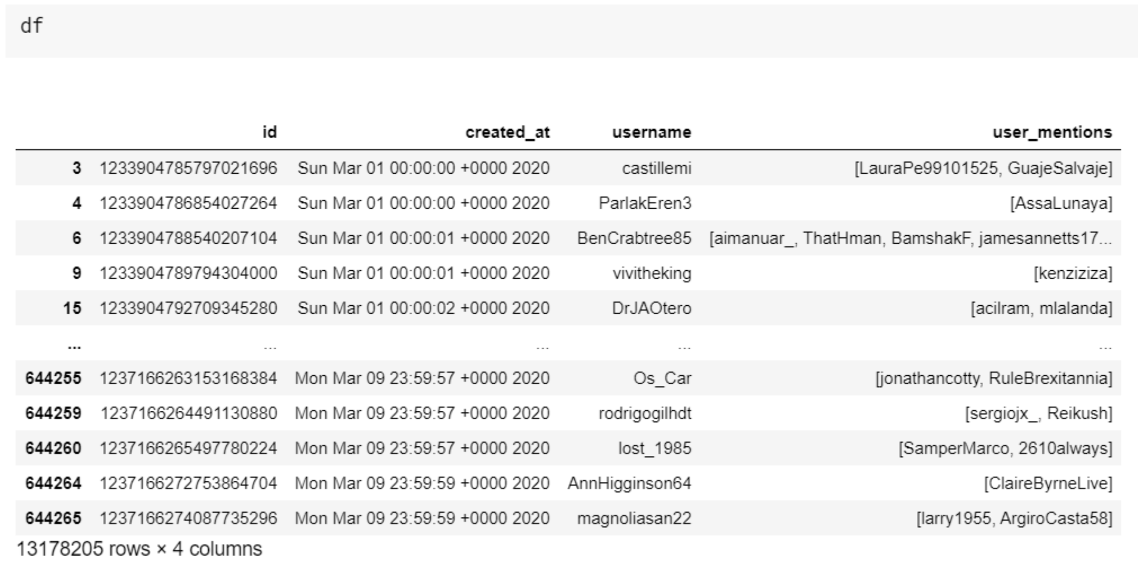
	username	Tweet_count
520252	WhatsOnOLIO	37320
783276	infosrv	21404
45969	AnimalsHolbox	13607
283001	KorayDavulcu2	11999
225819	HoraCatalana	11859
535303	_BB_RADIO_MUSIC	8561
415540	RadioTeddyMusic	8541
761915	haykakan_top	8507
332561	MathieuRonsard	7598
1053058	thaiselenags	7441

Code Snippet 6 – Most active users

- *WhatsOnOLIO* = the account of Olio, a food sharing app, which has 2M active users and when a user posts a food pick-up, it can be set to be posted on twitter automatically.
- *KorayDavulcu2* and *thaiselenags* = possibly spamming or automated accounts, the accounts do not exist now after huge post frequency during March. Looking for tags I found out they were fake profiles.
- *MathieuRonsard* = possibly automated account, low followers and all the tweets redirect to a suspicious link.
- *infosrv*: Account suspended by twitter, could be spamming.
- *AnimalsHolbox*: 0 followers, 0 following, 2.3 million tweets, possible spammer
- *HoraCatalana*: Automated account posts time every 5 minutes in Catalan, possibly using Socialoomph or tweetpi for automated scheduled posting.
- *_BB_RADIO_MUSIC*: German radio posts all the songs it plays, possibly automated
- *RadioTeddyMusic*: Another german radio station, posts all the songs it plays, possibly automated or a very monotonous job.

Number of mentions per user

The user mention data is taken as nested list (Table 3) as it was in json file and to be able to use it to count user mentions the data had to be flattened and then by using Counter library from Collection package repetition of each word is counted (Table 4).

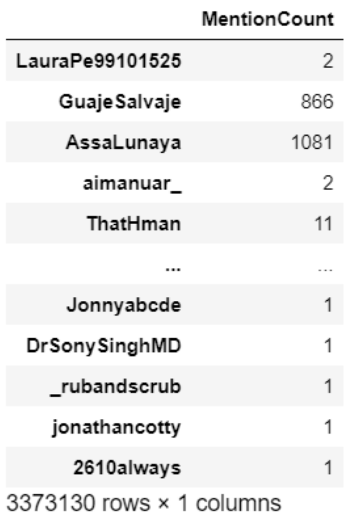


```
df
```

	id	created_at	username	user_mentions
3	1233904785797021696	Sun Mar 01 00:00:00 +0000 2020	castillemi	[LauraPe99101525, GuajeSalvaje]
4	1233904786854027264	Sun Mar 01 00:00:00 +0000 2020	ParlakEren3	[AssaLunaya]
6	1233904788540207104	Sun Mar 01 00:00:01 +0000 2020	BenCrabtree85	[aimanuar_, ThatHman, BamshakF, jamesannetts17...
9	1233904789794304000	Sun Mar 01 00:00:01 +0000 2020	vivitheking	[kenziziza]
15	1233904792709345280	Sun Mar 01 00:00:02 +0000 2020	DrJAOtero	[acilram, mlalanda]
...	...	...	...	...
644255	1237166263153168384	Mon Mar 09 23:59:57 +0000 2020	Os_Car	[jonathancotty, RuleBrexitannia]
644259	1237166264491130880	Mon Mar 09 23:59:57 +0000 2020	rodrigogilhdt	[sergiojx_, Reikush]
644260	1237166265497780224	Mon Mar 09 23:59:57 +0000 2020	lost_1985	[SamperMarco, 2610always]
644264	1237166272753864704	Mon Mar 09 23:59:59 +0000 2020	AnnHigginson64	[ClaireByrneLive]
644265	1237166274087735296	Mon Mar 09 23:59:59 +0000 2020	magnoliasan22	[larry1955, ArgiroCasta58]

13178205 rows × 4 columns

Table 3 – User data dataframe



	MentionCount
LauraPe99101525	2
GuajeSalvaje	866
AssaLunaya	1081
aimanuar_	2
ThatHman	11
...	...
Jonnyabcde	1
DrSonySinghMD	1
_rubandscrub	1
jonathancotty	1
2610always	1

3373130 rows × 1 columns

Table 4– User mention counts

There are 3 times as many users mentions as there are active users in the region and 21,715,621 users (about the population of New York) are mentioned. This comes out to be 7 user-mentions per user on average, but it can be seen most users do not get and there are very few users. Less than 3% of the mentioned users get mentioned more than once a day. The in a strange turn of numbers, there are only 100 users who get more than 100 mentions a day (Figure 7). The graph is taken on a log scale so all the data could fit.

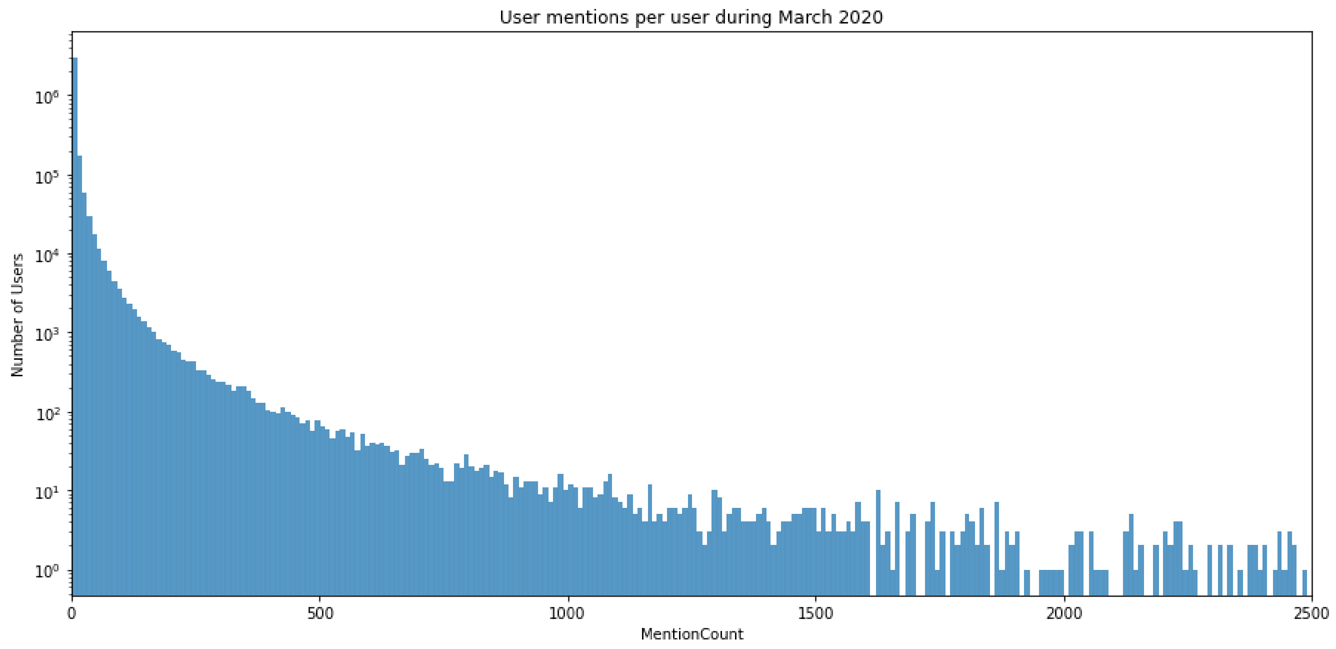
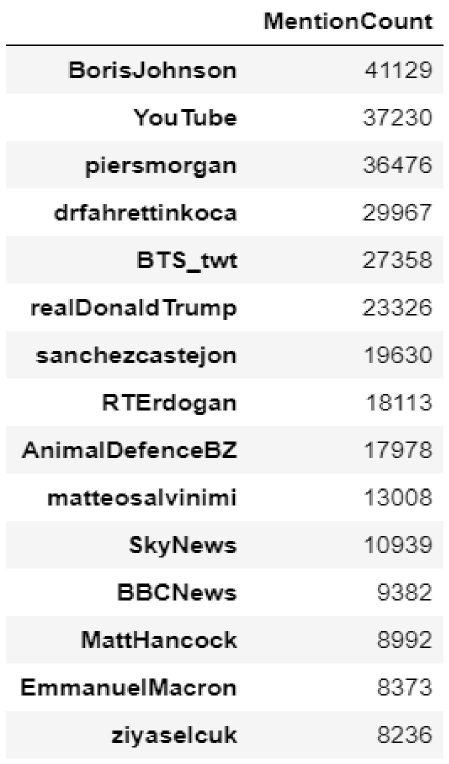


Figure 7 – User mentions per user during March 2020

Most mentioned users



	MentionCount
BorisJohnson	41129
YouTube	37230
piersmorgan	36476
drfahrettinkoca	29967
BTS_twt	27358
realDonaldTrump	23326
sanchezcastejon	19630
RTErdogan	18113
AnimalDefenceBZ	17978
matteosalvinimi	13008
SkyNews	10939
BBCNews	9382
MattHancock	8992
EmmanuelMacron	8373
ziyaselcuk	8236

Some of the most common mentions are the **world leaders** like Boris Johnson, Donald Trump, and Emmanuel Macron. March 2020 has been the time of covid-19 outbreak, restrictions, and lockdowns. And the people have been trying to reach the leaders on the mishaps or commending them for taking actions. This can also be seen by the presence of Matt Hancock being in the list.

News channels have been the first line of information on the new guidelines, people could be replying to the tweets of the news by the media.

Table 5 – Most mentioned users

Part 4: Event analysis

Identify days with high activity

For this task I took data from Ireland and UK so that the text analysis is in English and easy to deduce. The json tags extracted to make dataframe for this task are, **ID**, **created_at**, **place.country,** and **text.**

The data can is very light in size and the operations performed during Part 1 will be repeated to make plot a graph (*Figure 8)*. To spot unusual days, first we'll check for spikes. From the previous analysis we observed that number of tweets during the weekends are more than on weekdays. Keeping this information in mind, we can check weekdays with high number of tweets.

From the graph, for Ireland **12th and 17th March** have high surge in next day tweets and the graph for UK **26th March** is selected since it is the most active day for the UK.

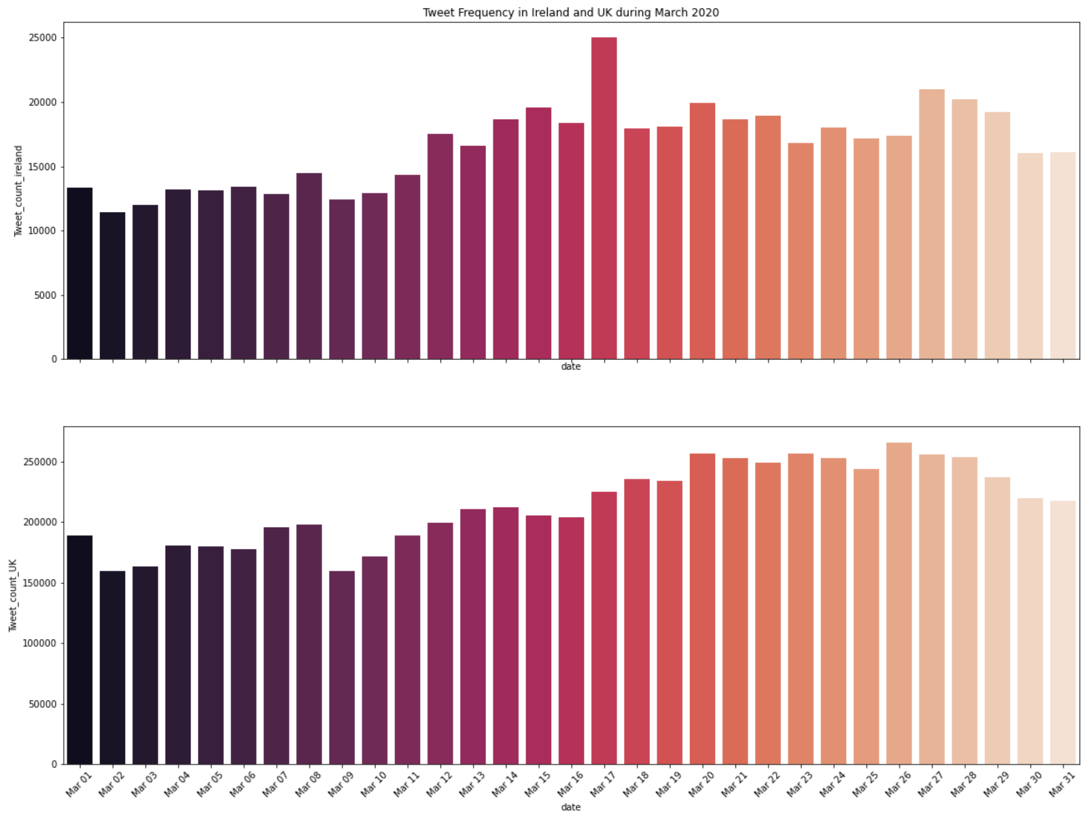


Figure 8 – Number tweets in Ireland (above) and UK during March 2020

Making Word Cloud for the event data

Preprocessing the data

To get the event processing the text data in terms of:

- Removing hyperlinks
- Removing User mentions
- Removing non=alpha numeric values
- Replacing \n with space (done during data extraction)
- Converting all data in to lowercase

To do this a combination of regex and replace() function is used and processed text is extracted (Code Snippet – 7).

```
df['processed_text'] = df['text'].str.lower().str.replace('(@[a-z0-9]+)\w+',' ')\
                          .str.replace('(http\S+)', ' ')\
                           .str.replace('([^0-9a-z \t])',' ').str.replace(' +',' ')
```

Code Snippet 7 – Data preprocessing for Word Cloud

Word Cloud: 12th March 2020, Ireland

The processed data is made into a single string and that text data is set as input to the WordCloud.generate() function (Code snippet 8 and Figure 9)

```
from PIL import Image
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
import matplotlib.pyplot as plt

# making a huge string with processed data value
text = " ".join(tweet for tweet in df_IR_12.values)

f, ax = plt.subplots(figsize= [20,10])
plt.imshow(WordCloud(stopwords = STOPWORDS,  collocations = False, width=2000,height=1000,
                     background_color = 'white').generate(text), interpolation='bilinear', )
plt.axis('off')
plt.savefig('#4.2_Ireland_12_Mar.png', bbox = 'tight')
```

Code Snippet 8 – Making a Word Cloud

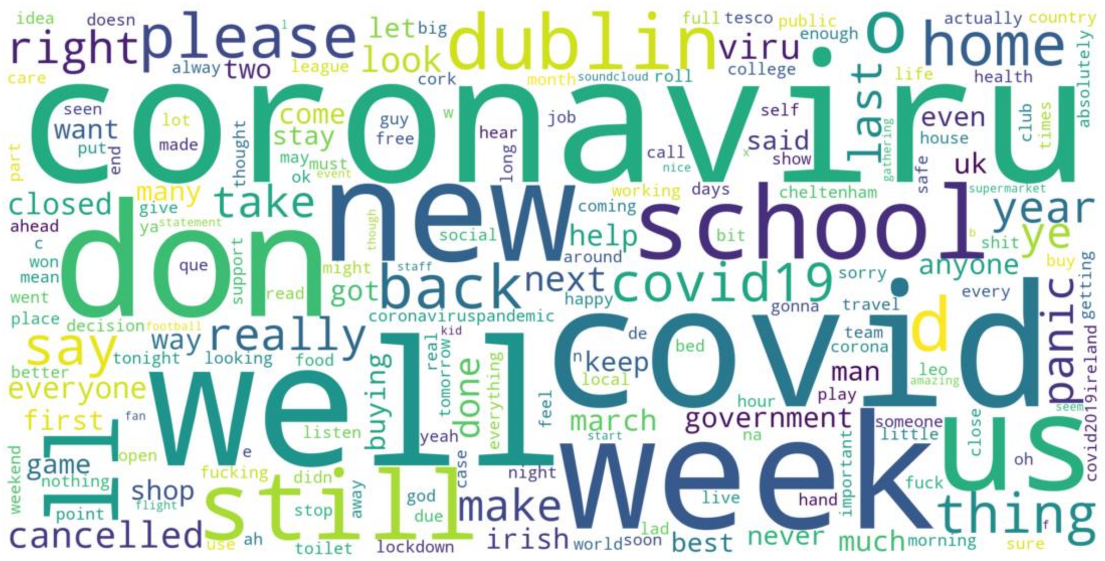
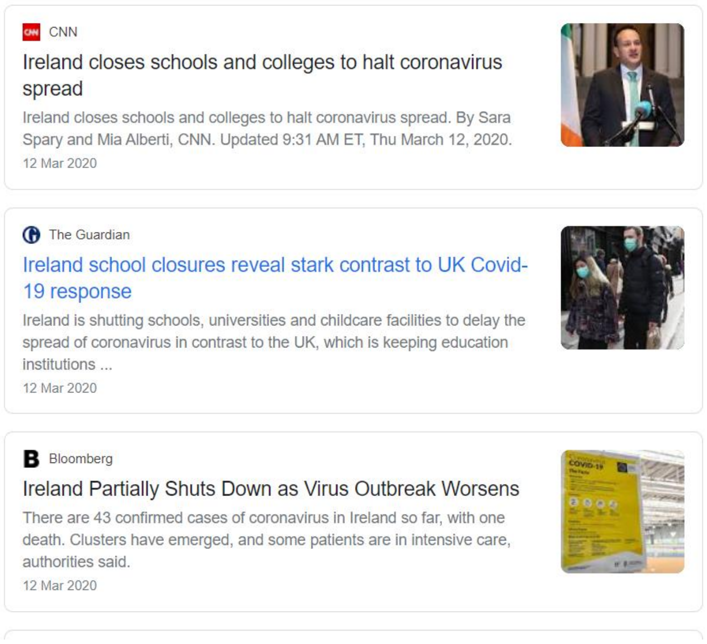


Figure 9 – Word cloud 12th Mar 2020, Ireland

The most common words are well done, coronavirus, school and week. This day Prime Minister of Ireland announced 2 weeks lock down and closed the schools down.


CNN

Ireland closes schools and colleges to halt coronavirus spread

Ireland closes schools and colleges to halt coronavirus spread. By Sara Spary and Mia Alberti, CNN. Updated 9:31 AM ET, Thu March 12, 2020.

12 Mar 2020

The Guardian

Ireland school closures reveal stark contrast to UK Covid-19 response

Ireland is shutting schools, universities and childcare facilities to delay the spread of coronavirus in contrast to the UK, which is keeping education institutions ...

12 Mar 2020

Bloomberg

Ireland Partially Shuts Down as Virus Outbreak Worsens

There are 43 confirmed cases of coronavirus in Ireland so far, with one death. Clusters have emerged, and some patients are in intensive care, authorities said.

12 Mar 2020

Figure 10 – Ireland News from 12th March 2020

Word Cloud: 17th March 2020, Ireland

It is St. Patrick's Day on 17th March and it does not come as a surprise that tweet count in Ireland skyrocketed that day.

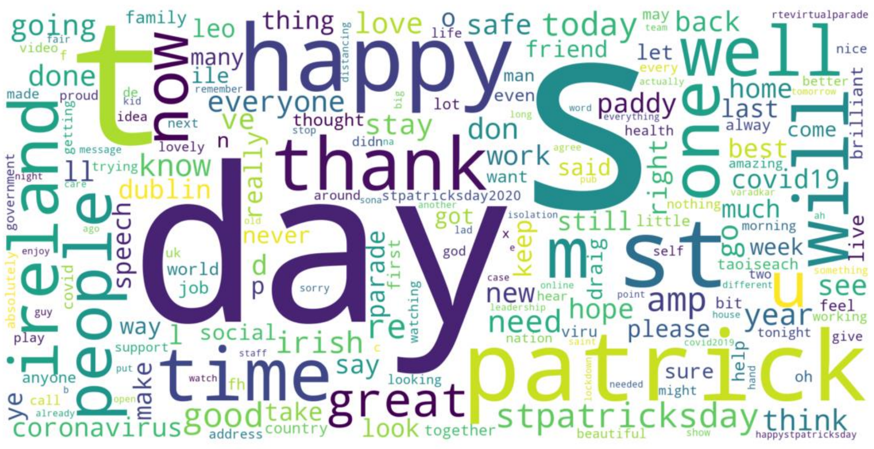
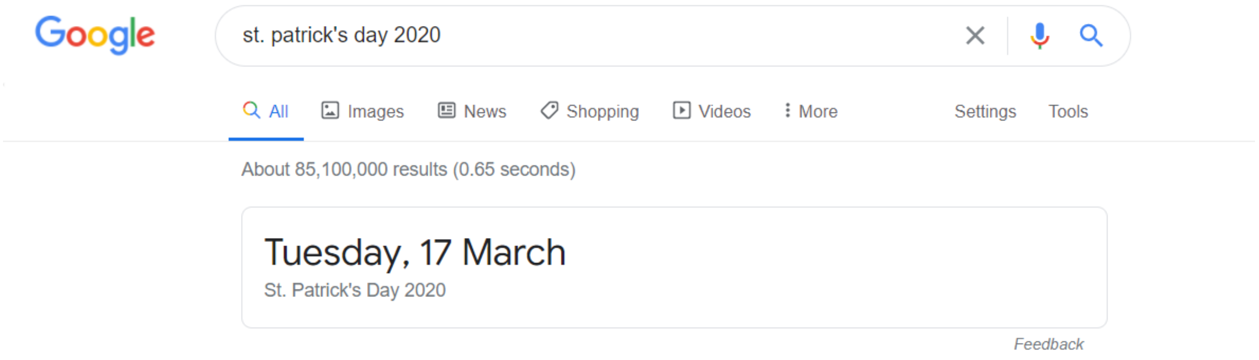


Figure 11 - Word Cloud: 17th March 2020, Ireland



Google

st. patrick's day 2020

All Images News Shopping Videos More Settings Tools

About 85,100,000 results (0.65 seconds)

Tuesday, 17 March

St. Patrick's Day 2020

Feedback

Figure 12 – St. Patrick's Day

Word Cloud: 26th March 2020, United Kingdom

This word cloud was made using a black silhouette as a mask. The key words are **'well', 'done'** and **'nhs'**. This marks the first time "Clap for carers" happened.

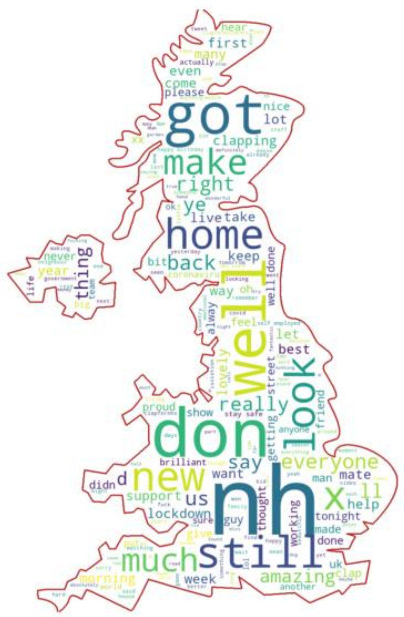
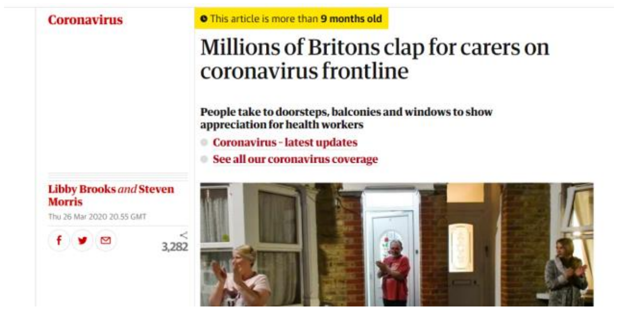


Figure 13 – Word Cloud 26th March, United Kingdom



Coronavirus

This article is more than 9 months old

Millions of Britons clap for carers on coronavirus frontline

People take to doorsteps, balconies and windows to show appreciation for health workers

Coronavirus - latest updates

See all our coronavirus coverage

Libby Brooks *and* Steven Morris

Thu 26 Mar 2020 20.55 GMT

3,282

Figure 14 – Guardian screenshot from 26th March 2020

Excerpt from wikipedia:

"On 26 March, the number of UK COVID-19 deaths increased by more than 100 in a day for the first time, rising to 578, while a total of 11,568 had tested positive for the virus.[113] At 8:00 pm that day, people from across the UK took part in applause in appreciation of health workers, later branded as Clap for Our Carers.[114][115] This gesture was repeated on the next nine Thursdays, up to 28 May.[116]"

Part 5: Reflection

The term “twitter data” alone yields over 7 million search results on Google Scholar. It is no surprise that there is an enormous amount of analysis and research being done on data from Twitter to find events in recent history or even some times, in the foreseeable future. It is also important to note that the data comes with a wide range of biases. Since around 80% of users use mobile devices for Twitter, low smartphone ownership amongst the poorer classes wipes out that section of the society for any analysis. For example, while inspecting an event, many more geo-tagged tweets can be traced back to the urban and privileged areas and this may paint a misleading picture of who was affected by the said event. The assumption that data will accurately reflect the social world and its voice revolving around events falls on a hollow ground as there are significant gaps since particular communities still have a far reach.

The strengths of Twitter as a data source to find events are many and this is advocated by the number of researches surrounding it. This is especially in the case of local events that can be updated in real-time quicker than news-outlet agencies can reach them and cover the event. The tweets remain based on the interpretation of the user though, so factual inaccuracies and miscommunication are bound to creep in. However, the nature of local, day-to-day events is such that most Twitter users rarely cover them.

The rate limit of Twitter API and the magnitude of data is both a strength and a weakness for the purpose. While it ensures an abundance of information and a larger dataset to base one’s findings on, noise and biases need to be mined out and efficient algorithms need to manage data of the scale. Not all tweets are geotagged; users can choose to disable location services while posting and this lack of information is sometimes a deterrent in the research to find events in a particular region. As most users use Twitter to voice their personal and political voice, the dataset yields result for searches into finding days like happy occasions or discontentment over political decisions but day-to-day events for reporting purposes are less common.

Most ethical & legal concerns surrounding users’ data from Twitter is regarding the user’s identity and tweets if these are to be produced in academic publications where gaining informed consent from the volume of users is difficult. But as long as the data is being employed to trace events, generally, ethical & legal challenges aren’t much of a hindrance.