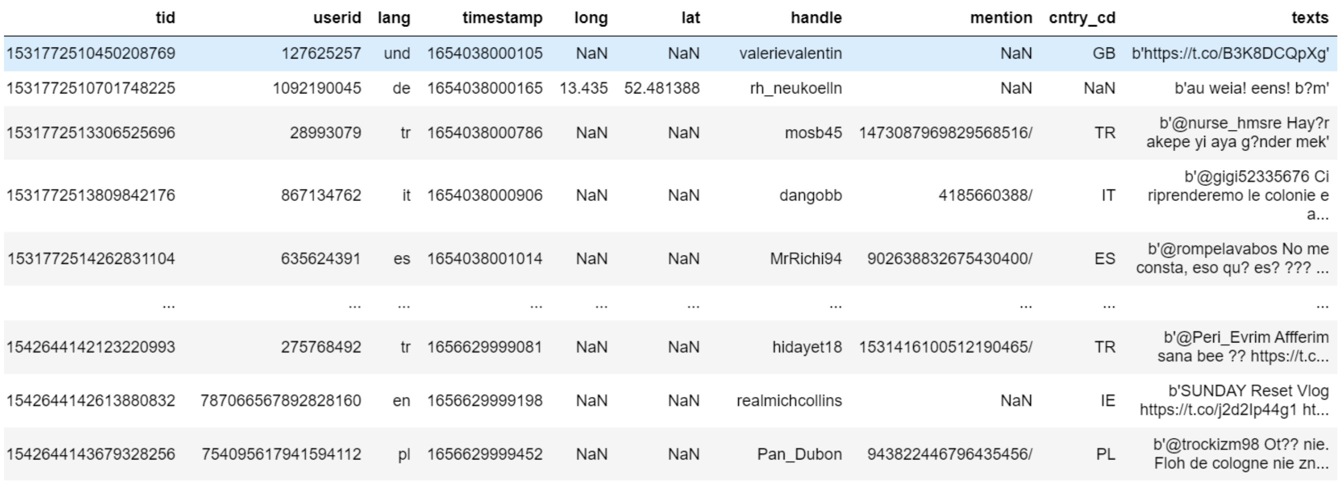
Coursework : ECMM443 Introduction to Data Science

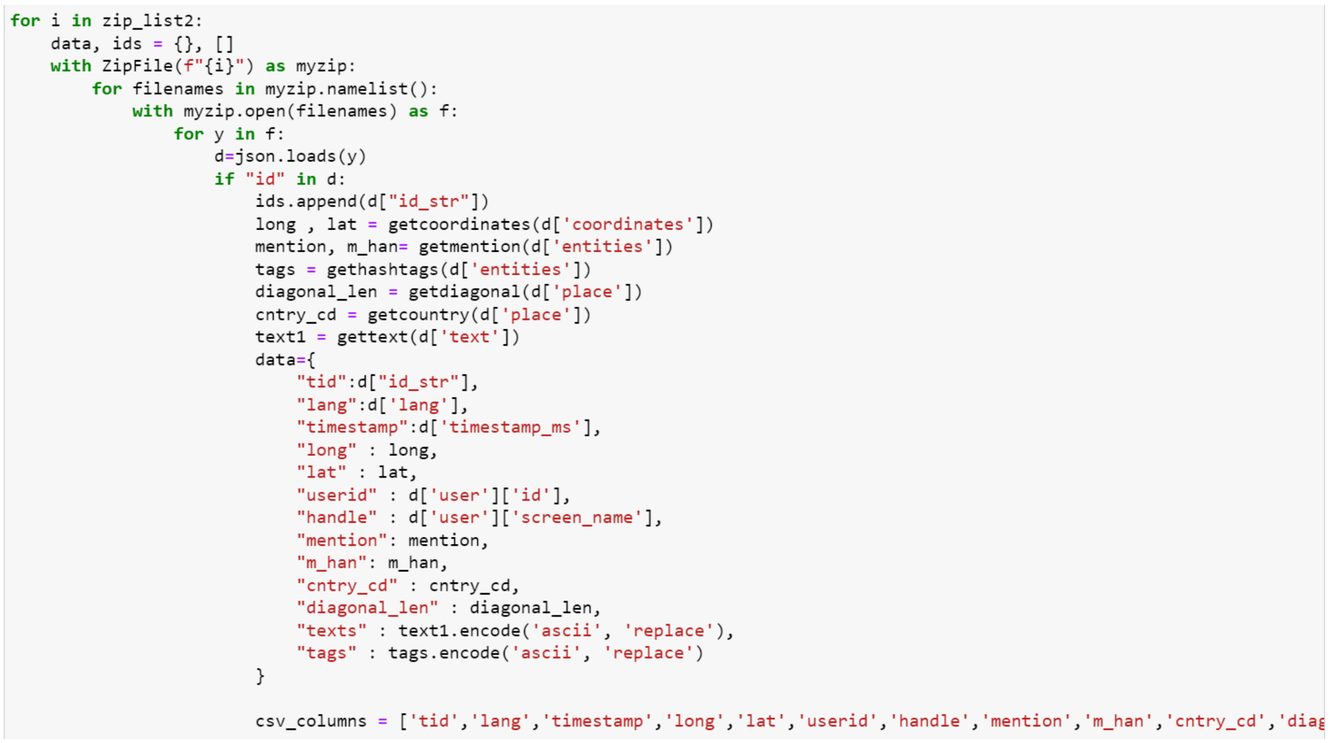
Data Preparation

Working with the zipped data size of 8.62GB is huge which is around 96GB on the hard drive. To continue further, the most viable option is to clean and take only the crucial information for analysis. Flat files are the easiest to manipulate and load the data. The data is extracted for every day into csv files to work with twitter ID, language, timestamp, coordinates (exact location of tweets), user ID, user's twitter handle, id of mentioned users, twitter handle of mentioned users, country, twitter texts. CSV files are then used to create pandas dataframe.



tid	userid	lang	timestamp	long	lat	handle	mention	cntry_cd	texts
1531772510450208769	127625257	und	1654038000105	NaN	NaN	valerievalentin	NaN	GB	b'https://t.co/B3K8DCQpXg'
1531772510701748225	1092190045	de	1654038000165	13.435	52.481388	rh_neukoelln	NaN	NaN	b'au weia! eens! b?m'
1531772513306525696	28993079	tr	1654038000786	NaN	NaN	mosb45	1473087969829568516/	TR	b'@nurse_hmsre Hay?r akepe yi aya g?nder mek'
1531772513809842176	867134762	it	1654038000906	NaN	NaN	dangobb	4185660388/	IT	b'@gigi52335676 Ci riprenderemo le colonie e a...
1531772514262831104	635624391	es	1654038001014	NaN	NaN	MrRichi94	902638832675430400/	ES	b'@rompelavabos No me consta, eso qu? es? ??? ...
...	...	...	...	...	...	...	...	...	...
1542644142123220993	275768492	tr	1656629999081	NaN	NaN	hidayet18	1531416100512190465/	TR	b'@Peri_Evrim Affferim sana bee ?? https://t.c...
1542644142613880832	787066567892828160	en	1656629999198	NaN	NaN	realmichcollins	NaN	IE	b'SUNDAY Reset Vlog https://t.co/j2d2Ip44g1 ht...
1542644143679328256	754095617941594112	pl	1656629999452	NaN	NaN	Pan_Dubon	943822446796435456/	PL	b'@trockizm98 Ot?? nie. Floh de cologne nie zn...

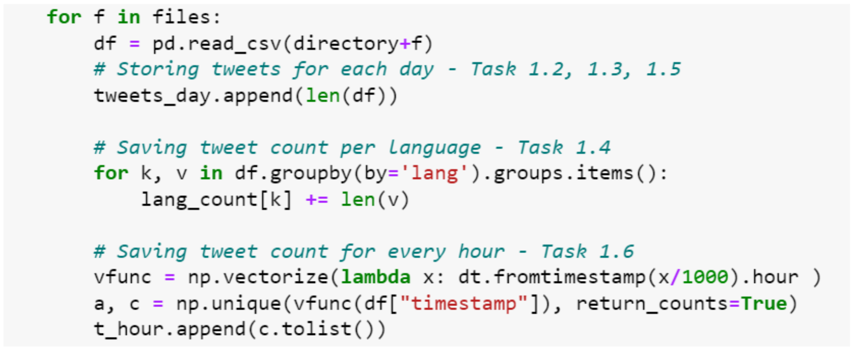
Table 1. Final table after data extraction and cleaning



```python
for i in zip_list2:
    data, ids = {}, []
    with ZipFile(f"{i}") as myzip:
        for filenames in myzip.namelist():
            with myzip.open(filenames) as f:
                for y in f:
                    d=json.loads(y)
                    if "id" in d:
                        ids.append(d["id_str"])
                        long , lat = getcoordinates(d['coordinates'])
                        mention, m_han= getmention(d['entities'])
                        tags = gethashtags(d['entities'])
                        diagonal_len = getdiagonal(d['place'])
                        cntry_cd = getcountry(d['place'])
                        text1 = gettext(d['text'])
                        data={
                            "tid":d["id_str"],
                            "lang":d['lang'],
                            "timestamp":d['timestamp_ms'],
                            "long" : long,
                            "lat" : lat,
                            "userid" : d['user']['id'],
                            "handle" : d['user']['screen_name'],
                            "mention": mention,
                            "m_han": m_han,
                            "cntry_cd" : cntry_cd,
                            "diagonal_len" : diagonal_len,
                            "texts" : text1.encode('ascii', 'replace'),
                            "tags" : tags.encode('ascii', 'replace')
                        }

                        csv_columns = ['tid','lang','timestamp','long','lat','userid','handle','mention','m_han','cntry_cd','diag
```

Figure 2. Code snippet for data preparation

Task 1



```python
for f in files:
    df = pd.read_csv(directory+f)
    # Storing tweets for each day - Task 1.2, 1.3, 1.5
    tweets_day.append(len(df))

    # Saving tweet count per language - Task 1.4
    for k, v in df.groupby(by='lang').groups.items():
        lang_count[k] += len(v)

    # Saving tweet count for every hour - Task 1.6
    vfunc = np.vectorize(lambda x: dt.fromtimestamp(x/1000).hour )
    a, c = np.unique(vfunc(df["timestamp"]), return_counts=True)
    t_hour.append(c.tolist())
```

Figure 3. Code snippet for preparing for Task 1

<u>Task 1:</u> 1.1

Total number of tweets are **15033548** after cleaning and removing anomalies. The tweets were identified by their unique “tweet Ids” and the rest of the garbage values were removed and then loaded into a dataframe. There were also duplicated data which was removed using panda’s inbuilt function.

<u>Task 1:</u> 1.2

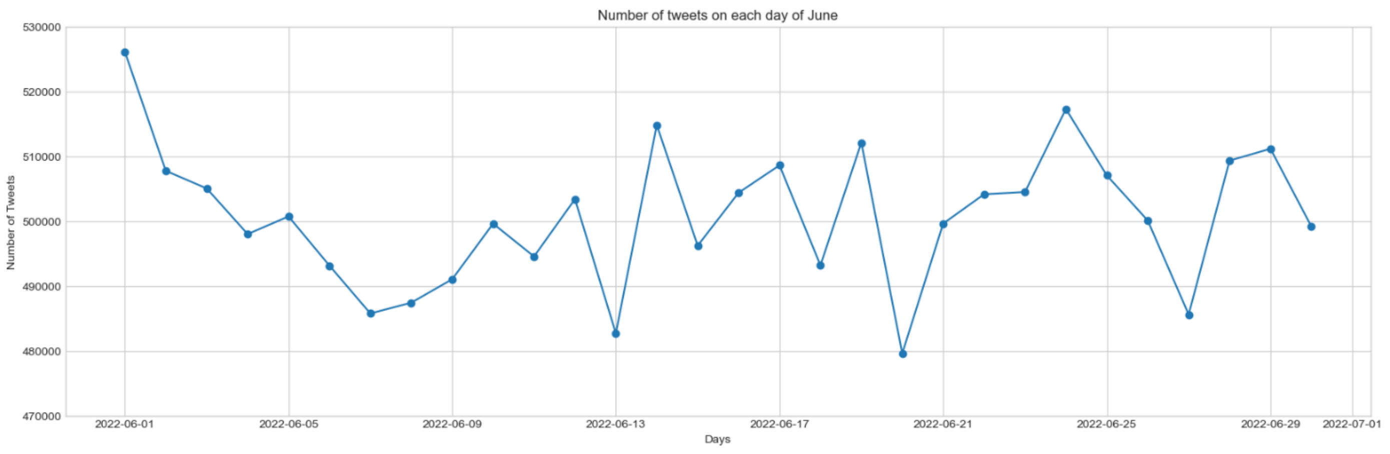


Figure 4. Plot showing no of tweets made per day for the month of June 2022

The tweets are higher in the beginning of the month and slow down around 7th June and then rise again. The low tweets can be observed on 13th, 20th and 27th June (all Mondays) which could be because of the Monday blues and people starting their work week after the weekend. The high peaks are observed on around 14th,19th and 24th June which could be due to some major event during these days. Towards the end of the month, tweets are looking to settle down but not too much.

Task 1: 1.3

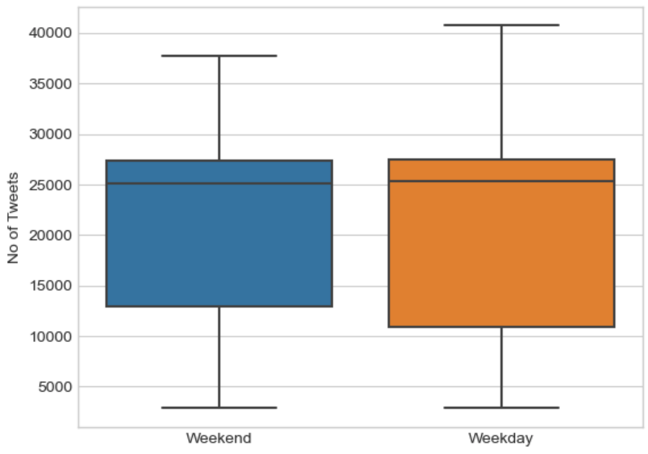


Figure 5. Box-whisker to compare weekend/weekdays

Features	Weekend	Weekday
Lowest value	~2500	~2500
Highest value	~38000	~42000
Median	~25000	~25500
Lower Quartile	~13000	~11000
Higher Quartile	~27000	~28000

Table 6. Box and whisker feature values

The values of the box-whisker features-the highest, lowest, median and quartile values on weekdays are greater than that of weekends. As for the spread of the data, it is visually obvious how people tweet more on the weekends as compared to the weekends. The median of both the days are almost the same which talks about the consistency. Even though people are busy during the weekdays, they are tweeting approximately with the same trend over weekends.

Task 1: 1.4

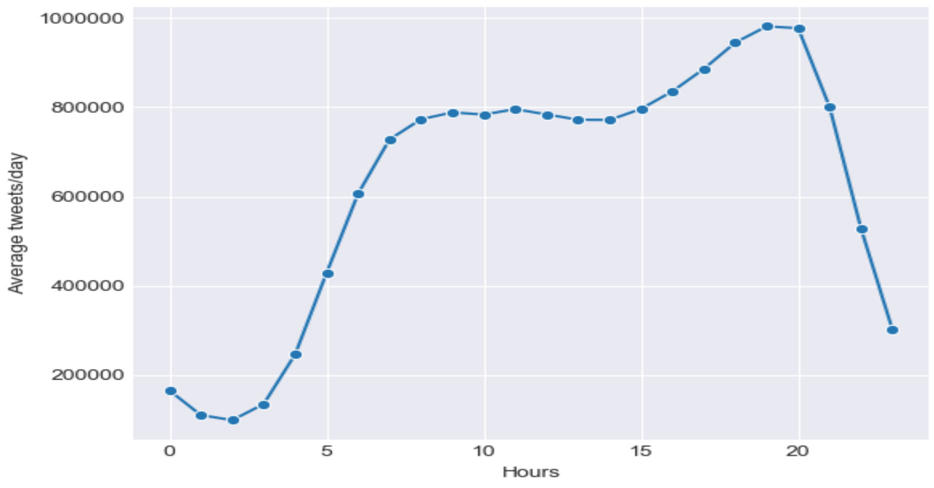


Figure 7. Graph showing average tweets per hour in the day

The average tweets per hour for over a time span of 30 days. It can be seen that tweets during the daytime are very high as compared to the late-night hours or very early hours. The gradual decline in tweets can be seen after 8pm. We see consistent increase in tweeting activity after 2am. From 10am to 3pm, there is a stability in the tweets which might be due to the peak productive working hours.

Task 2

Task 2: 2.1

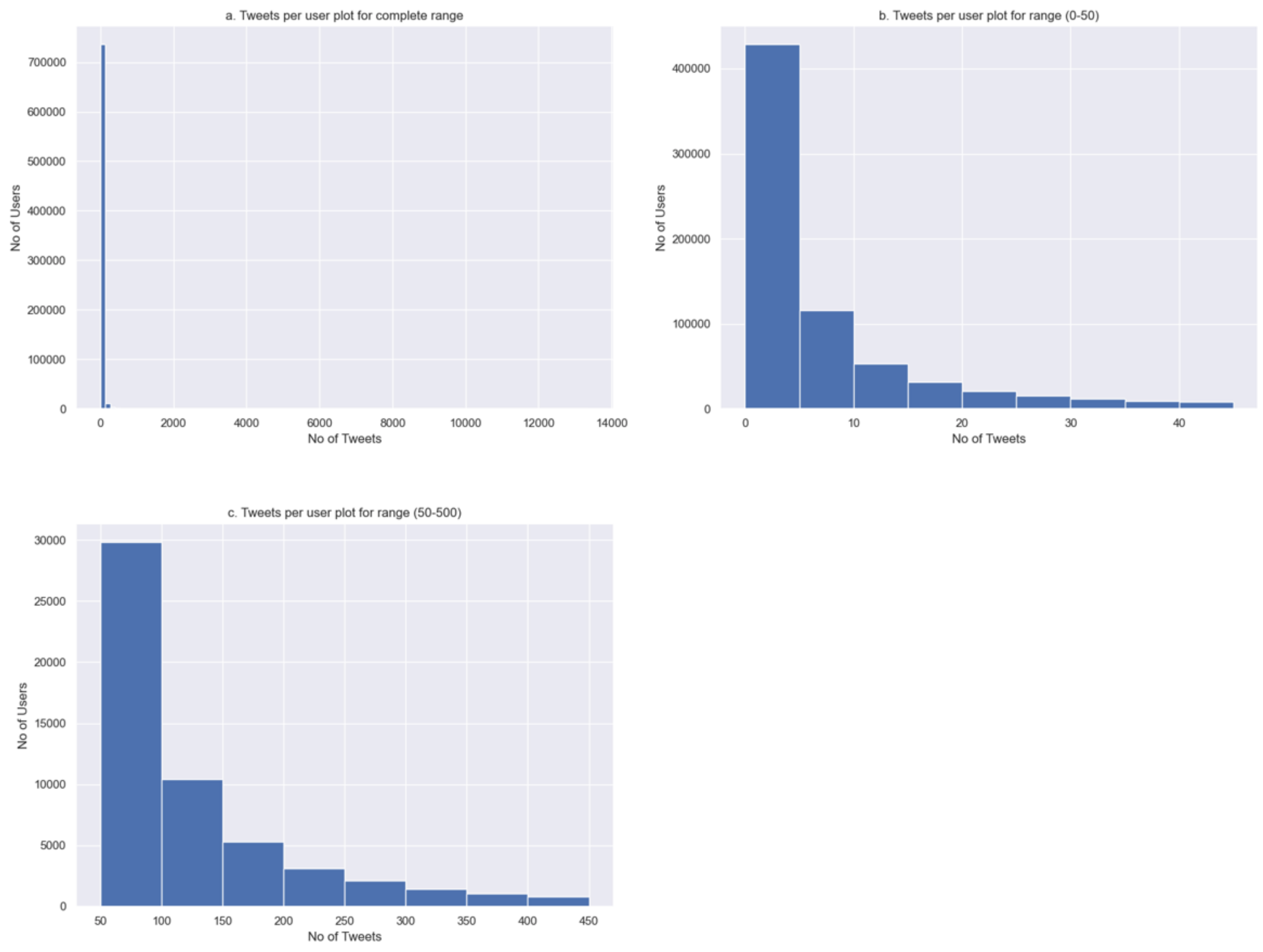
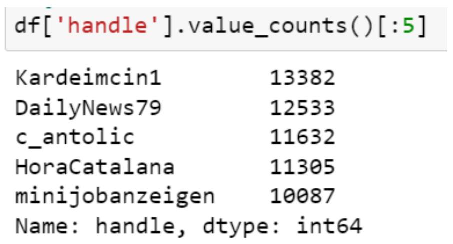


Figure 8. Histograms showing tweets per user

Figure 8 shows that majority of people tweet about 1-5 tweets per month. The number of tweets fall exponentially with the increase in number of tweets. There are also a few outliers that make up to 35000 tweets per month. Only 1 tweet was posted by almost 69% users, 15% tweeted twice in one month and the rest tweeted for a atleast once per day.

Task 2: 2.2

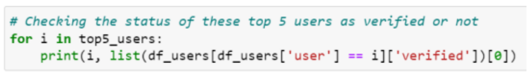


```
df['handle'].value_counts()[:5]

Kardeimcin1         13382
DailyNews79         12533
c_antolic           11632
HoraCatalana        11305
minijobanzeigen     10087
Name: handle, dtype: int64
```

Figure 9. Top 5 users

There are many ways to identify whether the accounts are automated/bots.

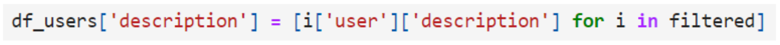
- Verified users: The raw files contain a tag which tells us whether the user is verified or not. For example, none of these accounts were verified



```
# Checking the status of these top 5 users as verified or not
for i in top5_users:
    print(i, list(df_users[df_users['user'] == i]['verified'])[0])
```

Figure 10. Code snippet for verified users

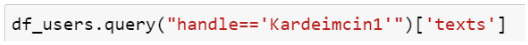
- User Description: The tag “description” gives an idea of the type of user. For example, “minijobanzeigen” has a description for sending out new job availability.



```
df_users['description'] = [i['user']['description'] for i in filtered]
```

Figure 11. Code snippet for description of users

- Spam Behaviour: Some tweets are in a repetitive fashion. For example, “Kardeimcin1” keeps posting the same kind of tweets with same links multiple times a day.



```
df_users.query("handle=='Kardeimcin1'")['texts']
```

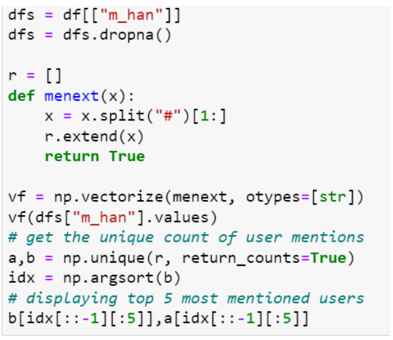
Figure 12. Code snippet for tweets by users

- Tweet count: Accounts like all of these are posting at a very high rate. Specially on some days more than the others.

SNo.	Name	No. of Tweets	Automated
1	Kardeimcin1	13376	Yes
2	DailyNews79	12518	Yes
3	c_antolic	11628	Yes
4	HoraCatalana	11294	Yes
5	minijobanzeigen	10087	Yes

Figure 13. Summary table for 5 top accounts

Task 2: 2.3



```
dfs = df[["m_han"]]
dfs = dfs.dropna()

r = []
def menext(x):
    x = x.split("#")[1:]
    r.extend(x)
    return True

vf = np.vectorize(menext, otypes=[str])
vf(dfs["m_han"].values)
# get the unique count of user mentions
a,b = np.unique(r, return_counts=True)
idx = np.argsort(b)
# displaying top 5 most mentioned users
b[idx[::-1][:5]],a[idx[::-1][:5]]
```

Figure 14. Top 5 people who were mentioned

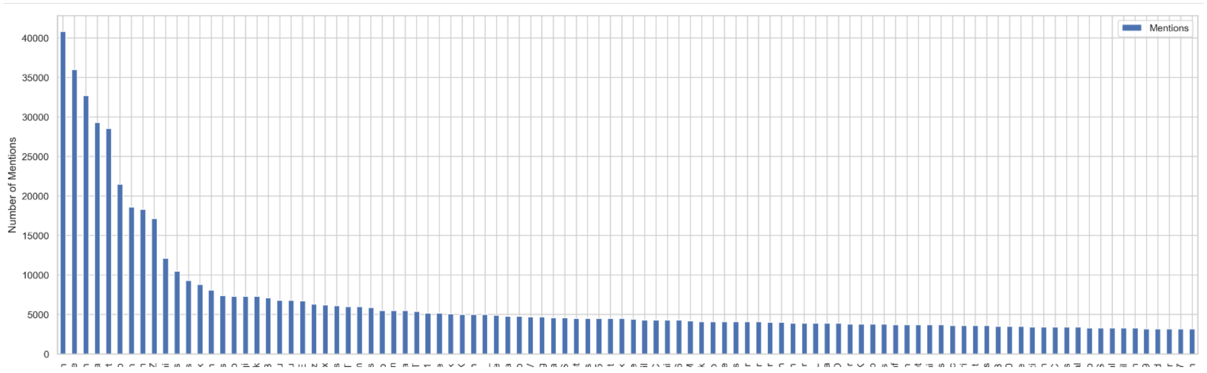
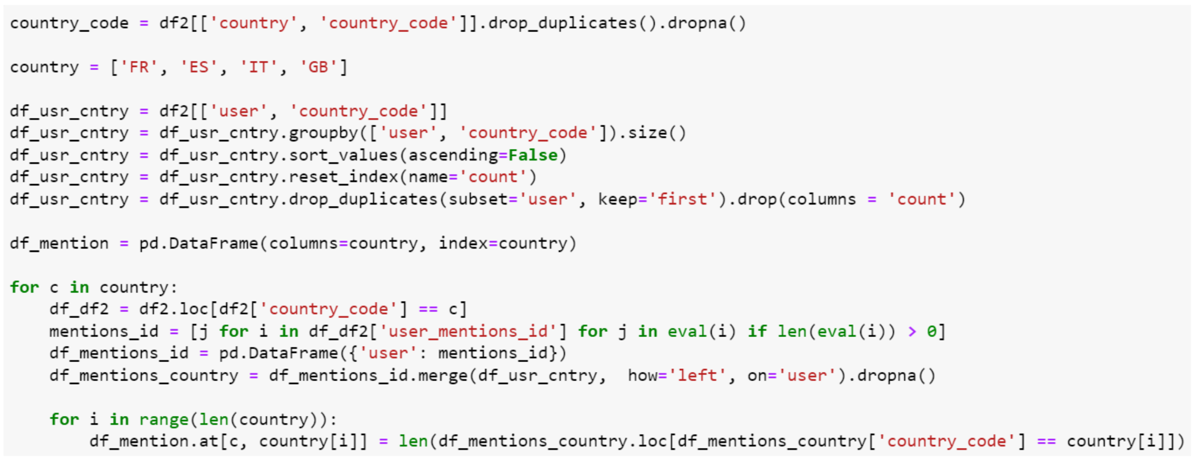


Figure 15. Plot for count of people mentioned

Above figure depicts the number of mentions per user for top 100 mentioned users. Youtube, RTErdogan, Boris Johnson is mentioned the most followed by Elon Musk and GBNEWS. The number of mentions per user keeps declining and gets quite close after a certain point. The more famous accounts or personalities are mentioned the most as per events and social situation.

- Youtube – for every YT video share, @Youtube is mentioned making it top in the ranking.
- RTErdogan – Turkish politician current president since 2014 (Recep Tayyip Erdoğan). He is very active on Twitter and tweets about subjects like high inflation where his post was re-tweeted and he was also mentioned in many of them. Being a political figure makes a good target for mentions in other people’s tweets.
- Boris Johnson – Earlier PM of the UK, resigned on 6th Sept. Twitter was already discussing his viable successor as the PM in June. Due to the instability of the UK government, he became one of the hottest political topics.
- Elon Musk – After providing a statement about the bankruptcy of the company, there was a high rise of Tweets tagging him. Employees working in the company started expressing their disappointments and fear of layoffs.
- GBNEWS– New channel with prime discussions such as Brexit, Queen’s Jubilee etc were broadcasted. It is one of the popular news channel, specially in the UK.

Task 2: 2.4



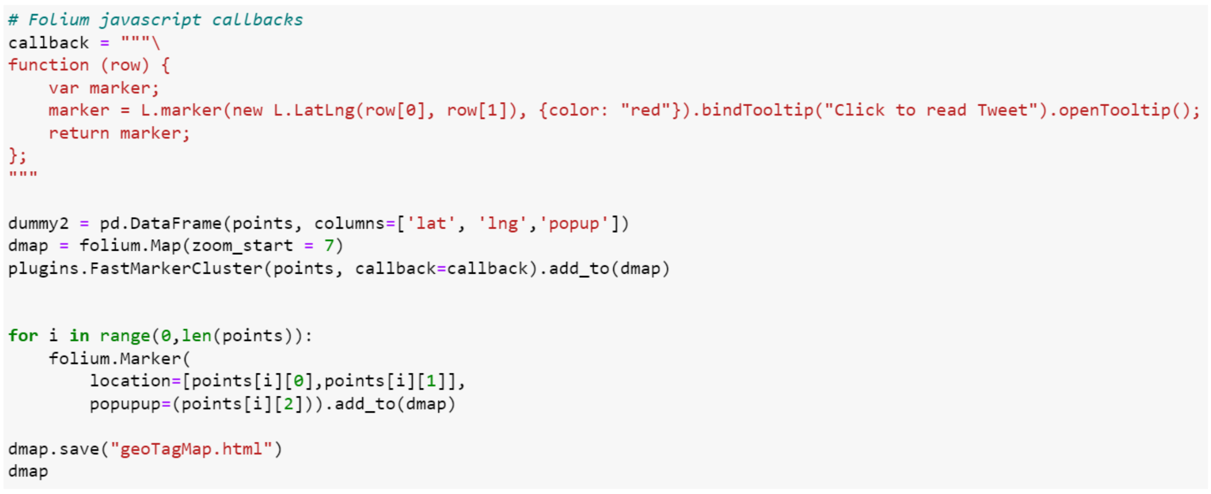
```
country_code = df2[['country', 'country_code']].drop_duplicates().dropna()

country = ['FR', 'ES', 'IT', 'GB']

df_usr_cntry = df2[['user', 'country_code']]
df_usr_cntry = df_usr_cntry.groupby(['user', 'country_code']).size()
df_usr_cntry = df_usr_cntry.sort_values(ascending=False)
df_usr_cntry = df_usr_cntry.reset_index(name='count')
df_usr_cntry = df_usr_cntry.drop_duplicates(subset='user', keep='first').drop(columns = 'count')

df_mention = pd.DataFrame(columns=country, index=country)

for c in country:
    df_df2 = df2.loc[df2['country_code'] == c]
    mentions_id = [j for i in df_df2['user_mentions_id'] for j in eval(i) if len(eval(i)) > 0]
    df_mentions_id = pd.DataFrame({'user': mentions_id})
    df_mentions_country = df_mentions_id.merge(df_usr_cntry,  how='left', on='user').dropna()

    for i in range(len(country)):
        df_mention.at[c, country[i]] = len(df_mentions_country.loc[df_mentions_country['country_code'] == country[i]])
```

Figure 16. Code snippet for 4 countries mentioning each other – FR/ES/IT/GB

We can observe that the countries mentioned themselves the most. Followed by the ones with some interesting events going on. For example, due to the government changing in the UK, it was also mentioned a lot.

Task 3

Task 3: 3.1



```
# Folium javascript callbacks
callback = """\
function (row) {
    var marker;
    marker = L.marker(new L.LatLng(row[0], row[1]), {color: "red"}).bindTooltip("Click to read Tweet").openTooltip();
    return marker;
};
"""

dummy2 = pd.DataFrame(points, columns=['lat', 'lng','popup'])
dmap = folium.Map(zoom_start = 7)
plugins.FastMarkerCluster(points, callback=callback).add_to(dmap)


for i in range(0,len(points)):
    folium.Marker(
        location=[points[i][0],points[i][1]],
        popupup=(points[i][2])).add_to(dmap)

dmap.save("geoTagMap.html")
dmap
```

Figure 17. Code snippet for GPS-tagged tweets

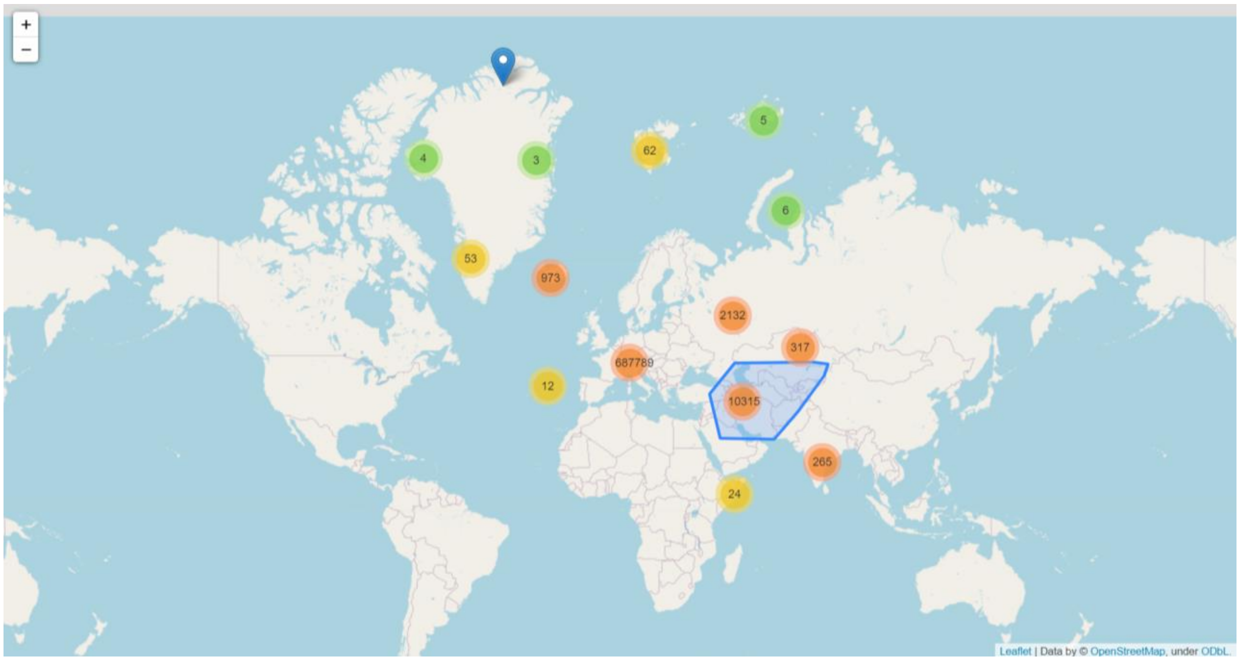
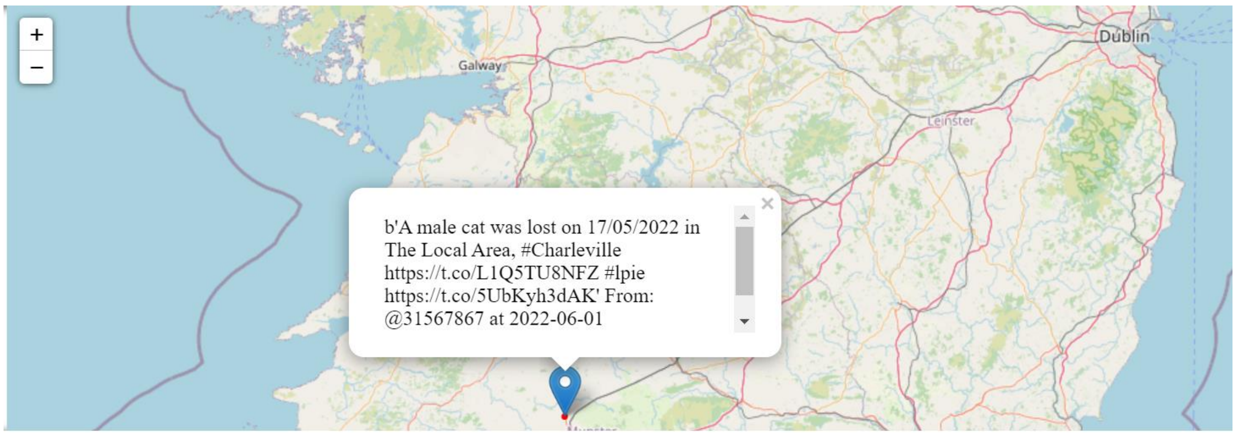


Figure 18. Screenshot for output - 1



b'A male cat was lost on 17/05/2022 in The Local Area, #Charleville https://t.co/L1Q5TU8NFZ #lpie https://t.co/5UbKyh3dAK' From: @31567867 at 2022-06-01

Figure 19. Screenshot for output – 2

Task 3: 3.2

Although the tweets are supposed to be from European boundary box, some of the tweets were from outside Europe also. We can see the tweets from Asia and also from North America. To look at UK, we see the big cities like London, being highly active on Twitter whereas smaller population countries like Ireland have tweeted low naturally. This concludes that Twitter is more popular in places with higher population than in smaller ones.

Task 3: 3.3

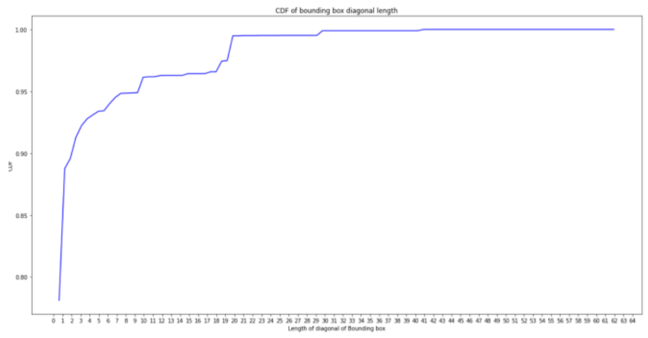


Figure 20. CFD of bounding box

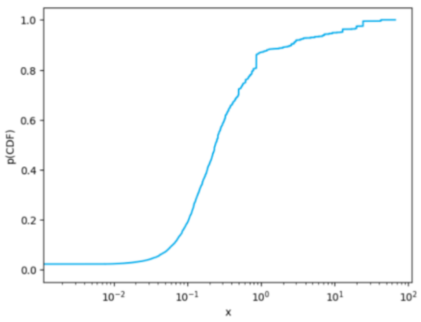
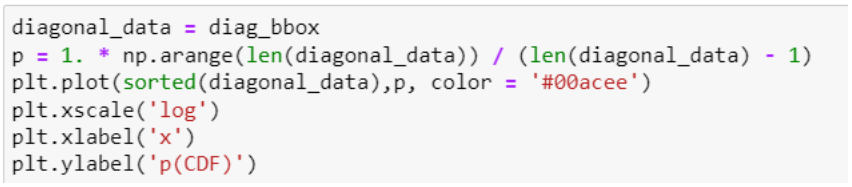


Figure 21. CFD of bounding box (log format)

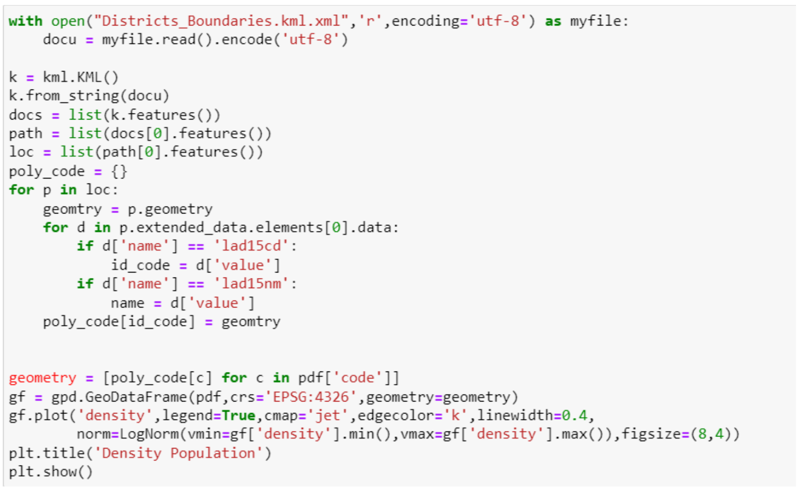


```
diagonal_data = diag_bbox
p = 1. * np.arange(len(diagonal_data)) / (len(diagonal_data) - 1)
plt.plot(sorted(diagonal_data),p, color = '#00acee')
plt.xscale('log')
plt.xlabel('x')
plt.ylabel('p(CDF)')
```

Figure 22. Code snippet for CFD of bounding box (log format)

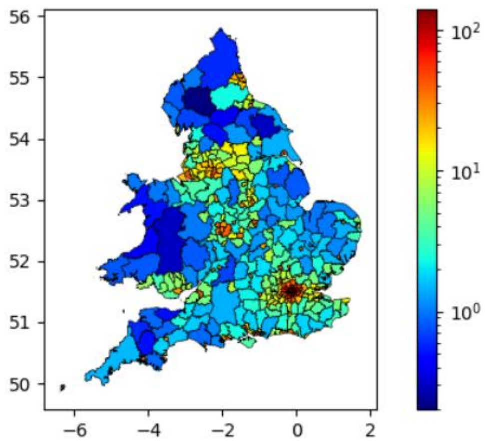
By observing figure 20, we can see that almost 90% of the data is within length 4 of the diagonal of the bounding box. This means that 90% of the bounding boxes are really small and the rest 10 are slightly bigger. We see that only 5% of the bounding boxes have a diagonal length greater than 11 (approx.). So only 5% of the total boxes are significantly larger ie the location is not at all precise and might cover larger areas. This could be due to low internet connection which was not good enough to provide a smaller/precise location of the user.

Task 3: 3.4



```
with open("Districts_Boundaries.kml.xml",'r',encoding='utf-8') as myfile:
    docu = myfile.read().encode('utf-8')

k = kml.KML()
k.from_string(docu)
docs = list(k.features())
path = list(docs[0].features())
loc = list(path[0].features())
poly_code = {}
for p in loc:
    geomtry = p.geometry
    for d in p.extended_data.elements[0].data:
        if d['name'] == 'lad15cd':
            id_code = d['value']
        if d['name'] == 'lad15nm':
            name = d['value']
    poly_code[id_code] = geomtry


geometry = [poly_code[c] for c in pdf['code']]
gf = gpd.GeoDataFrame(pdf,crs='EPSG:4326',geometry=geometry)
gf.plot('density',legend=True,cmap='jet',edgecolor='k',linewidth=0.4,
        norm=LogNorm(vmin=gf['density'].min(),vmax=gf['density'].max()),figsize=(8,4))
plt.title('Density Population')
plt.show()
```

Figure 23. Code snippet for additional data for density population in UK Figure

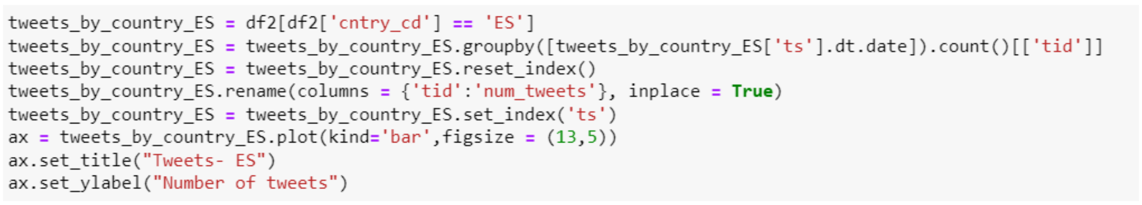


24. Plot for density population in UK

The colour gradient of the map helps to clearly understand the data for United Kingdom. The areas near the water bodies (mostly the country sides) are not highly populated. And we can see that metropolitan cities such as London as very densely populated. This is due to the better standard of living, accessibility, technology and corporate job opportunities. The northern part of the county, Scotland is amongst the lowest populated places due to the harsh and low temperatures which makes it difficult to reside. Similar to the north, the areas in Wales are also thinly populated. Decent population density can be observed in areas such as Manchester and Yorkshire.

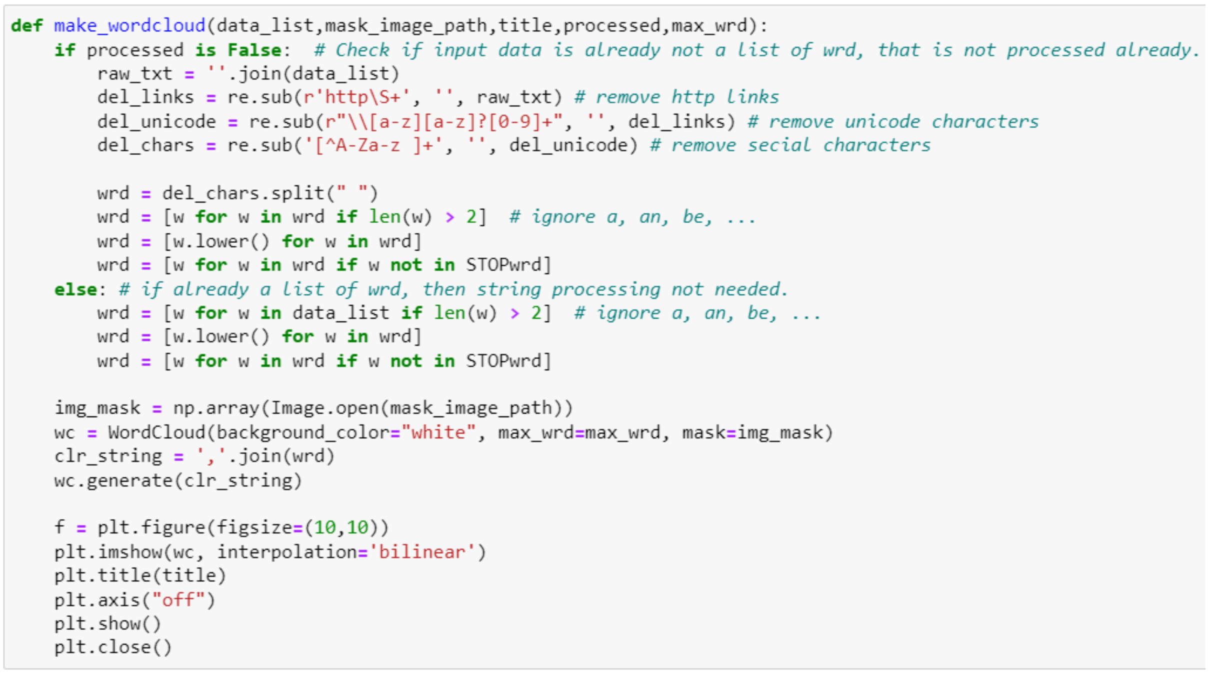
Task 4

Task 4: 4.1



```
tweets_by_country_ES = df2[df2['cntry_cd'] == 'ES']
tweets_by_country_ES = tweets_by_country_ES.groupby([tweets_by_country_ES['ts'].dt.date]).count()[['tid']]
tweets_by_country_ES = tweets_by_country_ES.reset_index()
tweets_by_country_ES.rename(columns = {'tid':'num_tweets'}, inplace = True)
tweets_by_country_ES = tweets_by_country_ES.set_index('ts')
ax = tweets_by_country_ES.plot(kind='bar',figsize = (13,5))
ax.set_title("Tweets- ES")
ax.set_ylabel("Number of tweets")
```

Figure 25. Code snippet for bar charts



```
def make_wordcloud(data_list,mask_image_path,title,processed,max_wrd):
    if processed is False:  # Check if input data is already not a list of wrd, that is not processed already.
        raw_txt = ''.join(data_list)
        del_links = re.sub(r'http\S+', '', raw_txt) # remove http links
        del_unicode = re.sub(r"\\[a-z][a-z]?[0-9]+", '', del_links) # remove unicode characters
        del_chars = re.sub('[^A-Za-z ]+', '', del_unicode) # remove secial characters

        wrd = del_chars.split(" ")
        wrd = [w for w in wrd if len(w) > 2]  # ignore a, an, be, ...
        wrd = [w.lower() for w in wrd]
        wrd = [w for w in wrd if w not in STOPwrd]
    else: # if already a list of wrd, then string processing not needed.
        wrd = [w for w in data_list if len(w) > 2]  # ignore a, an, be, ...
        wrd = [w.lower() for w in wrd]
        wrd = [w for w in wrd if w not in STOPwrd]

    img_mask = np.array(Image.open(mask_image_path))
    wc = WordCloud(background_color="white", max_wrd=max_wrd, mask=img_mask)
    clr_string = ','.join(wrd)
    wc.generate(clr_string)

    f = plt.figure(figsize=(10,10))
    plt.imshow(wc, interpolation='bilinear')
    plt.title(title)
    plt.axis("off")
    plt.show()
    plt.close()
```

Figure 26. Code snippet word cloud

Country 1: Spain (ES)

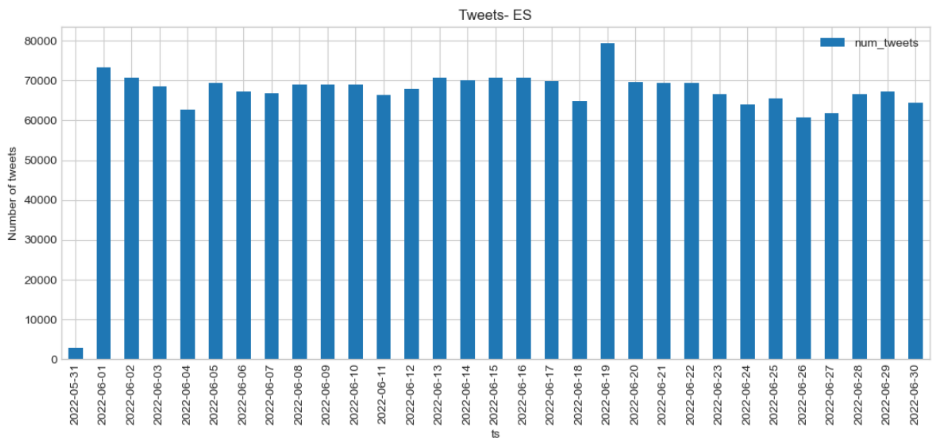


Figure 27. Unusual high activity day (ES) = 19th June 2022

Country 2: Italy (IT)

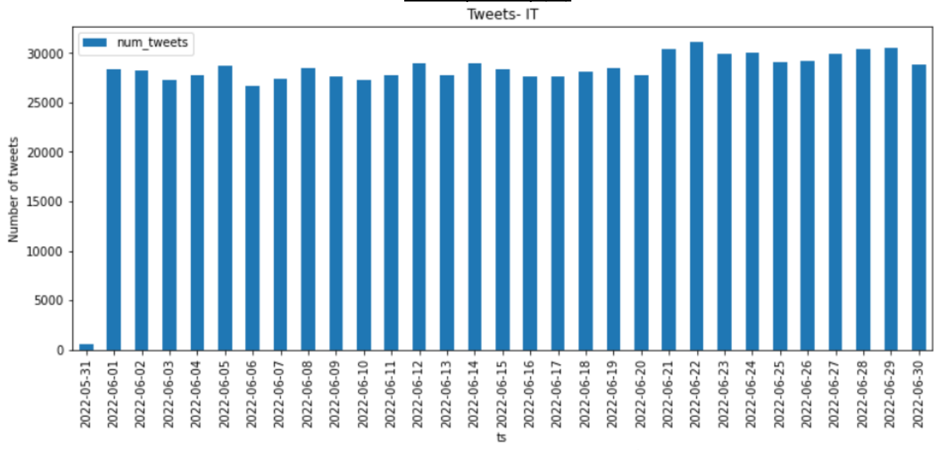


Figure 28. Unusual high activity day (IT) = 22nd June 2022

Country 3: France (FR)

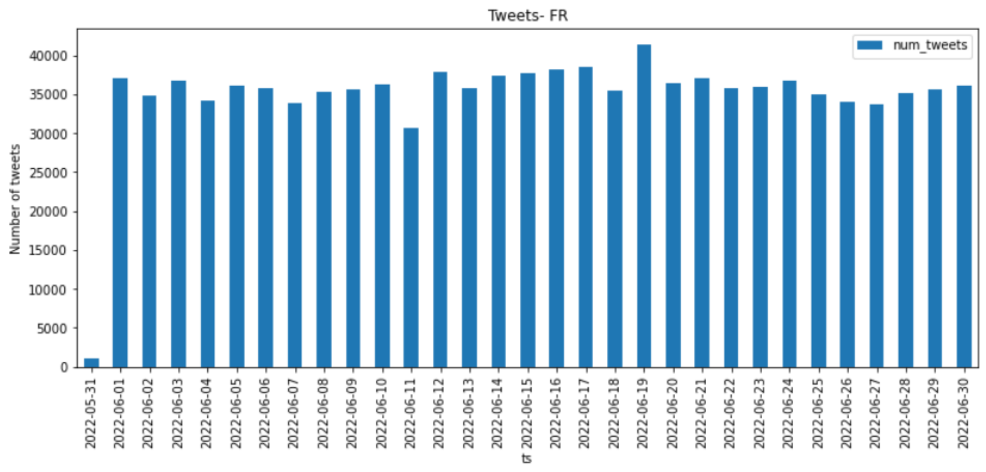


Figure 29. Unusual high activity day (FR) = 19th June 2022

We have selected 3 countries for analysis - Spain, Italy and France. Plotted a bar chart with number of tweets for each day of the month. We filtered only the data specific for each country and loaded the same into 3 different dataframes - one for each. This graph helps to identify the unusual days (ie. when there were maximum tweets created) as 22nd June for Italy and 19th June for both Spain and France. These high volumes, indicate a possibility of some event.

Task 4: 4.2

(a)

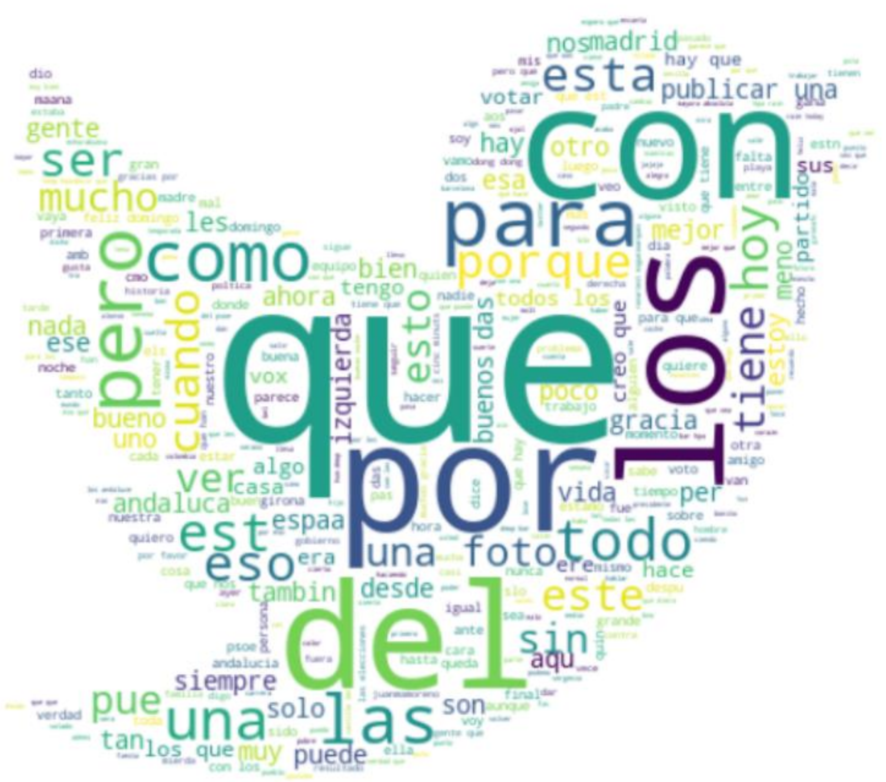


Figure 30. Word Cloud from tweets – ES

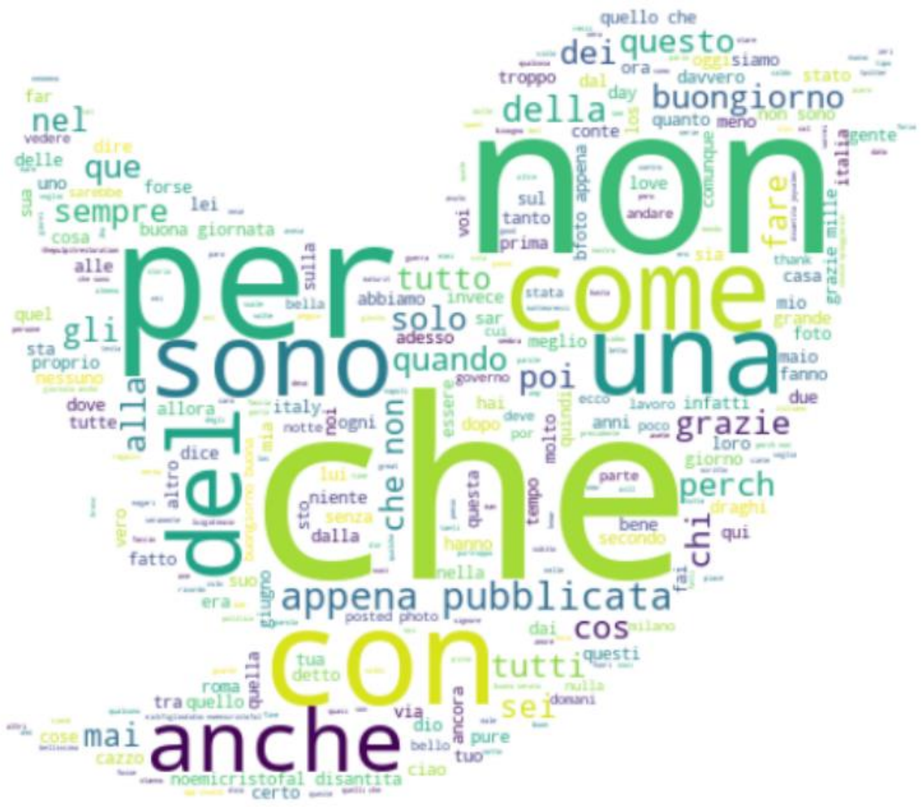


Figure 31. Word Cloud from tweets – IT

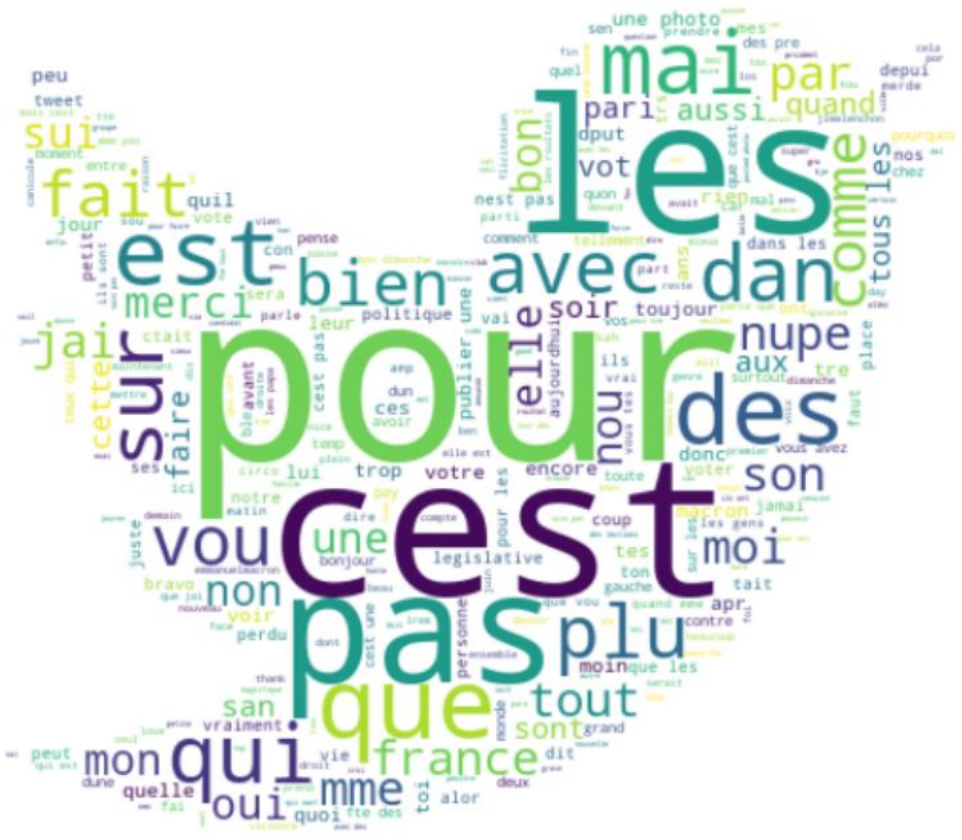


Figure 32. Word Cloud from tweets – FR

(b)

```
n = 50

txt_token = word_tokenize(comment_words)
tk_without_sw = [word for word in txt_token if not word in stopwords if len(re.findall(r'\w', word))>=2]
words = pd.DataFrame(Counter(tk_without_sw).most_common()[:n],
                             columns = ['word', 'word_count'])

bubble_chart = BubbleChart(area=words['word_count'], bubble_spacing=0.1)

bubble_chart.collapse()

rand_colors = []
for j in range(n):
    c = "#"+''.join([random.choice('ABCDEF0123456789') for i in range(6)])
    rand_colors.append(c)

fig, ax = plt.subplots(subplot_kw=dict(aspect="equal"), figsize = (10, 10))
bubble_chart.plot(ax, words['word'], colors = rand_colors)
ax.axis("off")
ax.relim()
ax.autoscale_view()
ax.set_title('Most frequent words')
plt.show()
```

Task 4: 4.3

i) Spain (19th June)

Looking at the words like partido (party), poltica (politics) one can concluse that the event is political. On further research it was found that the Popular Party(PP) had won on 19th June (Sunday) in the important regional elections. The country had a history of Socialist stronghold so such a victory was enjoyed in the country. Other words related to this event which can be found in the word cloud are gana(win),votar(vote) and equipo(equipment). The news headlines from this day reads “Spain's Popular Party boosted by Andalusia victory 18 months before general election” and “Spain: conservative People’s party wins unprecedented majority”

ii) Italy (22nd June)

22nd June was the last day of the Milan fashion week for men. The word cloud is filled with words related to this such as milano (Milan), pubblicata(published), italy (Italy), moda(fashion), stile(style), uomini(men) and ragazzi(boys). Some of the words depicting emotions such as love(love) and disantita(disheartened) can also be found. It is a reference point for the latest trends in men’s fashion. Since the fashion week is a popular even, maximum tweets were done on the last day. Headlines from online sources on the day read “Highlight : Milano Fashion Week Men’s”.

iii) France (19th June)

19th June was the day of the French legislative election results. People can be seen tweeting about the elections by using words like politique(politics) and legislative(legislative). Macron's minister lost her seat the same day. Word cloud contains macron(Macron) and contre(versus) to hint towards this event. The news articles read "French legislative elections: Voter turnout at 38.11% at 5pm, slightly down from first round". Macron, who was re-elected in April lost the majority in June elections making people tweet pros and cons of the political event.

Task 5: REFLECTION

Launched back in 2006, Twitter is one of the leading social media giants with more than 300 million active users [1] and 500 million tweets posted per day [2]. It has become a source of sentimental analysis, knowing about trends and happenings around the world which is a major focus for researchers and analysts these days. By using its powerful API's, the data can be extracted for any analysis. The metadata has so much information like demographics, personal details etc. It makes it easy to follow conversations and because of the hashtag USP and tweet limit, makes it interesting for text analysis. In this report we played around with Twitter data to analyse users, twitter trends by location, visualisation of data on maps etc. Some studies have identified natural disasters [3] [4] using twitter data.
With this being discussed, one should not ignore the drawbacks of this powerful platform which can easily be misused for spreading fake news or pushing political agendas which leads to hatred and sometimes crime as well. The automated bots and spam contribute to noise in the data leading to questionable credibility and the effort which would take to clean and prepare the data. Furthermore, the data can be biased as it is sampled from a small population and these biases reflects the posts made by people in one or another way. [7] mentions data created about Hurricane Sandy which was generated from Manhattan based on larger number of twitter users from the city portraying it as a hub of disaster. Only a handful of tweets were generated due to battery low after power cut. This bias was not reflecting truly on the actual scenario due to hurricane.
Another aspect are the ethical and legal issues arising from easy availability of Twitter data which can be resold without users realising that they had already provided consent. Contributing to biases on the platform is easy irrespective of whether intentional and unintentional which is because of the human nature too. For instance, a chatbot AI, Tay, was created by Microsoft but it started to make racist tweets and unfortunately, they had to shut it down. To monitor and keep the control Twitter's Terms of Service and Privacy Policy [8] has guidelines and by accepting the service agreement, users consent for their information to be used by third parties. The biggest example to all of this is the Cambridge Analytical Scandal.
Hence, twitter data helps in emerging stories, movements, disasters, etc. If this data can be used ethically it can give the best solutions and results for the betterment of people.

*References:**

[1] https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/
[2] https://www.internetlivestats.com/twitter-statistics/
[3] Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes twitter users: Real-time event detection by social sensors. In: Proceedings of the 19th International Conference on World Wide Web. pp. 851–860. WWW ’10, ACM, New York, NY, USA (2010), http://doi.acm.org/10.1145/1772690.1772777
[4] De Longueville, B., Smith, R.S., Luraschi, G.: Omg, from here, i can see the flames!”: A use case of mining location based social networks to acquire spatiotemporal data on forest fires. In: Proceedings of the 2009 International Workshop on Location Based Social Networks. pp. 73–80. LBSN ’09, ACM, New York, NY, USA (2009), http://doi.acm.org/10.1145/1629890.1629907
[5] Isa Inuwa-Dutse, Mark Liptrott, Ioannis Korkontzelos, Detection of spam-posting accounts on Twitter, Neurocomputing, Volume 315, 2018, Pages 496-511, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2018.07.044. [6] O. Varol, E. Ferrara, C.A. Davis, F. Menczer, A. FlamminiOnline human–bot interactions: detection, estimation, and characterization Proceedings of the International AAAI Conference on Web and Social Media, AAAI Press (2017), pp. 280-289
[7] https://hbr.org/2013/04/the-hidden-biases-in-big-data
[8] https://twitter.com/en/tos
*