ECMM443: Introduction to Data Science

Coursework

Table of Contents

Part 0: Data Ingestion

zipfile.ZipFile module was used to loop through all 720 zip files. Once inside each file, the following were loaded (not necessarily at the same time, but rather in a piece-meal manner, depending on which question was being answered) with the json library:

1. user→ID
2. created_at
3. timestamp_ms
4. id
5. extended→entities→user_mentions / entities→user_mentions
6. coordinates→coordinates
7. place→country
8. place→bounding_box→coordinates
9. extended_tweet→full_text/ text
10. extended_tweet→entities→hashtags/ entities→hashtags

multiprocessing.Pool module was used to parallelize the entire process. Once read, the data was stored in nested lists, which were then parsed, encoded in 'utf-8' format and written to a text file. These text files were later used while answering Questions 1 through 4.

Part 1: Basic Stats

1. The total number of unique tweets is 15,033,548. Overall, there are 15,040,709 entries in the files. Out of them there are 5,666 duplicate tweets occupying 6,839 places. 322 records do not have tweet ID (appearing as "None" or NULL depending on the method of data extraction). There are 753,481 unique users who have tweeted.

   Observation: There is a discrepancy between created_at and timestamp_ms. Twitter Documentation mentions that created_at is the time when the tweet was created according to local time. We observe that the minimum created_at value is 2022-05-31 23:00:00, i.e. 1 hour before the start of June 2022. We see that throughout the data, this shift of 1 hour remains between created_at and timestamp_ms (which holds the UTC time). This can be attributed to Daylight Savings time as between March and August, clocks are set back by 1 hour in the UK (*and much of the western hemisphere*), thus falling behind UTC.

   timestamp_ms is read as ts, which is then converted to datetime as follows:

```
[datetime.datetime.fromtimestamp(int(dt)/1000) for dt in ts]
```

   Going forward in this report, timestamp_ms is used as source time.

2. Time-series of tweets by day:

   Here is some basic code and a time series plot (shown in Fig 1):

```
# Group at day level
twt_daily = twt_df.groupby('ts_day').agg(num_tweets = ('tweet_id','nunique'))

# Plot
fig , ax = plt.subplots(figsize = (20 , 5))
ax.plot(twt_daily['num_tweets'] , color = 'orange');
ax.vlines(
    twt_daily.index ,
    ymin = min(twt_daily['num_tweets']) ,
    ymax = max(twt_daily['num_tweets']) ,
    color = 'blue' ,
    alpha = 0.1 ,
    ls = '--'
);
```

```
ax.set_xticks(twt_daily.index);
ax.set_xticklabels(labels = twt_daily.index, rotation = 90);

plt.xlim(min(twt_daily.index) , max(twt_daily.index));
```

Here is the result:

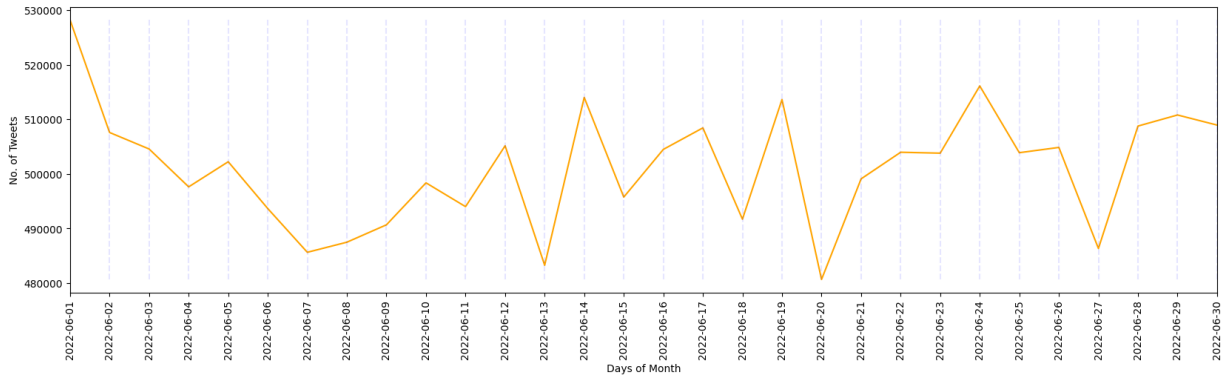


Fig 1: Time Series showing volume of tweets across the month of June'22

Comments:
Tweets seem to be falling slightly on Mondays/ Tuesdays, which makes sense as people resume work after the weekend. Volume generally rebounds back immediately.

3. Box and Whiskers Plot

```
# Preparing Boxplot
twt_df_grp = twt_df.groupby(['ts_day' , 'weekday']).agg(num_tweets = ('tweet_id' ,
'nunique')).reset_index()
# Grouping weekday numbers
twt_df_grp_wkday = twt_df_grp[twt_df_grp['weekday']==1]
t1 = twt_df_grp_wkday['num_tweets'].tolist()
# Grouping weekend numbers
twt_df_grp_wkend = twt_df_grp[twt_df_grp['weekday']==0]
t2 = twt_df_grp_wkend['num_tweets'].tolist()

week_tweet_dict = {
    'weekday':t1 ,
    'weekend':t2
    }
# Plot
fig , ax = plt.subplots()
ax.boxplot(week_tweet_dict.values());
ax.set_xticklabels(week_tweet_dict.keys());
ax.axhline(np.mean(t1) , ls = '--' , c = 'red' , label = 'weekday mean');
ax.axhline(np.mean(t2) , ls = '--' , c = 'olive' , label = 'weekend mean');
plt.legend(loc = 'best');
plt.title('Tweet Volume Difference');
```

The rather small difference is shown in Fig 2 below.

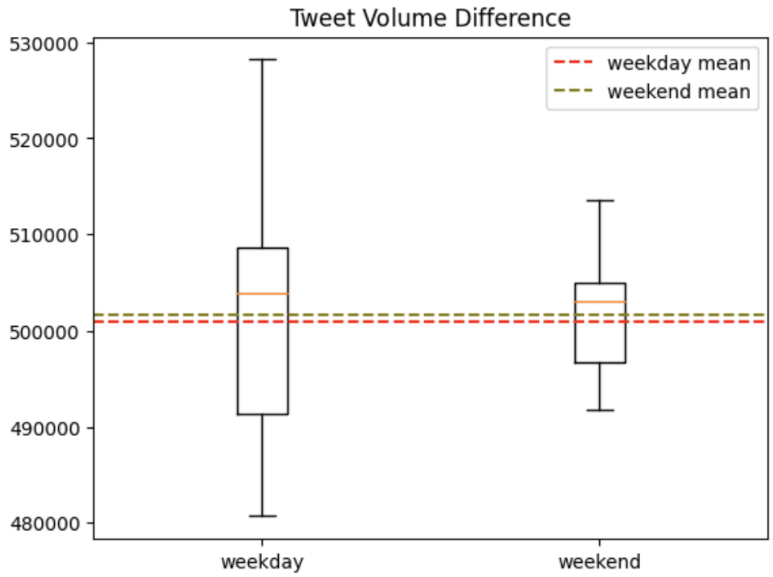


Fig 2: Boxplot showcasing the difference between volume of weekday tweets vs weekend tweets

From preliminary observation, it does not look like there is much difference. For more robust testing, the mean difference between the average number of their weekday and weekend tweets is generated for each user. In order to perform a t-test, this should approximate a normal distribution.

```
# Standardize
hypo_test_df['avg_weekend_tweets'] = (hypo_test_df['avg_weekend_tweets'] -
np.mean(hypo_test_df['avg_weekend_tweets']))/np.std(hypo_test_df['avg_weekend_tweets'])
hypo_test_df['avg_weekday_tweets'] = (hypo_test_df['avg_weekday_tweets'] -
np.mean(hypo_test_df['avg_weekday_tweets']))/np.std(hypo_test_df['avg_weekday_tweets'])
# Test for normality
hypo_test_df['mean_diff'] = hypo_test_df['avg_weekend_tweets'] -
hypo_test_df['avg_weekday_tweets']
(osm, osr), (slope, intercept, r) = scipy.stats.probplot(hypo_test_df['mean_diff'] , plot =
plt);
print(f'Correlation coefficient is {r}')
```

```
Correlation coefficient is 0.6261623577774496
```

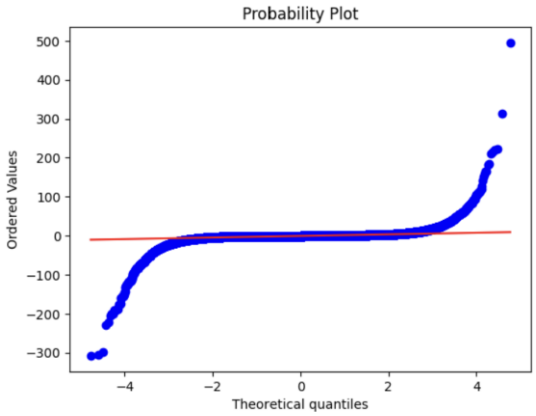


Fig 3: Quantile-quantile plot of data against a standard normal distribution

This does not look normal. However, with large N, t-test is not that sensitive to non-normal data, especially when the variances are equal (almost), as shown below.

```
np.var(hypo_test_df['avg_weekend_tweets']) = 0.9999999999999999
np.var(hypo_test_df['avg_week_tweets']) = 1.0000000000000002
```

Moreover, due to such large sample size (over 700,000), t-test can be assumed to be approximately valid.[1]

```
# Simulate CLT
X = np.array(hypo_test_df['mean_diff'])
sample_size = 100
N = 10000
collect = []
for _ in range(N):
    sample = [np.random.choice(X) for i in range(100)]
    sample_mean = np.mean(sample)
    collect.append(sample_mean)

fig , ax = plt.subplots()
freq , bins , patches = ax.hist(collect , bins = 100);
plt.title('Resampled Histogram');
ax.set_xlabel('Mean Diff');
ax.set_ylabel('Occurrences');
```

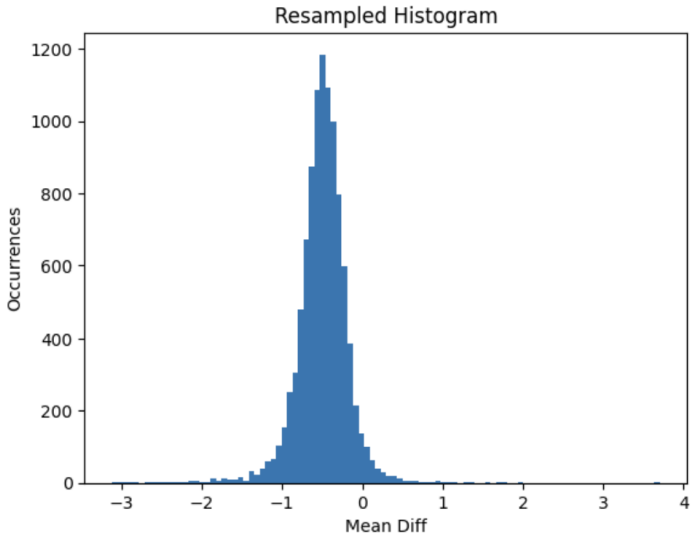


Fig 4: Resampled Histogram of difference between average weekend tweets and average weekday tweets

A paired two-sided t-test is performed, as equality (or inequality) of means is being tested – directionality of the difference between the means is not required.

```
%%time
# Standardize the data
avg_weekday_twt = (np.array(hypo_test_df['avg_weekday_tweets']) -
np.mean(np.array(hypo_test_df['avg_weekday_tweets'])))/np.std(np.array(hypo_test_df['avg_wee
kday_tweets']))
avg_weekdend_twt = (np.array(hypo_test_df['avg_weekend_tweets']) -
np.mean(np.array(hypo_test_df['avg_weekend_tweets'])))/np.std(np.array(hypo_test_df['avg_wee
kend_tweets']))

# Set up the Hypothesis test
# H0: There is no difference between the means; H1: there is difference between means
# Paired T Test - two sided
scipy.stats.ttest_rel(avg_weekday_twt , avg_weekdend_twt, alternative = 'two-sided')
```

[1] Skovlund, E. and Fenstad, G. (2001) "Should we always choose a nonparametric test when comparing two apparently nonnormal distributions?", *Journal of Clinical Epidemiology*, 54(1), pp. 86-92. doi: 10.1016/s0895-4356(00)00264-x.

Here is the result:

```
Ttest_relResult(statistic=2.654590994409255e-14, pvalue=0.9999999999999788)
```

Such a high p-value says:
1. There is definitely ***not*** enough evidence to ***reject*** the Null Hypothesis that the two means are equal.
2. The two datasets are most likely highly correlated; a correlation test would likely yield a very high coefficient.

```
scipy.stats.pearsonr(avg_weekday_twt , avg_weekdend_twt)
PearsonRResult(statistic=0.7548022023741843, pvalue=0.0)
```

Although not extremely high, the two samples (weekend tweets vs weekday tweets) are correlated to a fair degree (75%). To some extent, this may seem obvious, given we are comparing weekend tweets vs weekday tweets made by the same users.

4. Hour and day values are extracted via:

```
twt_df['hour'] = twt_df['ts'].dt.hour
twt_df['day'] = twt_df['ts'].dt.day_of_week
```

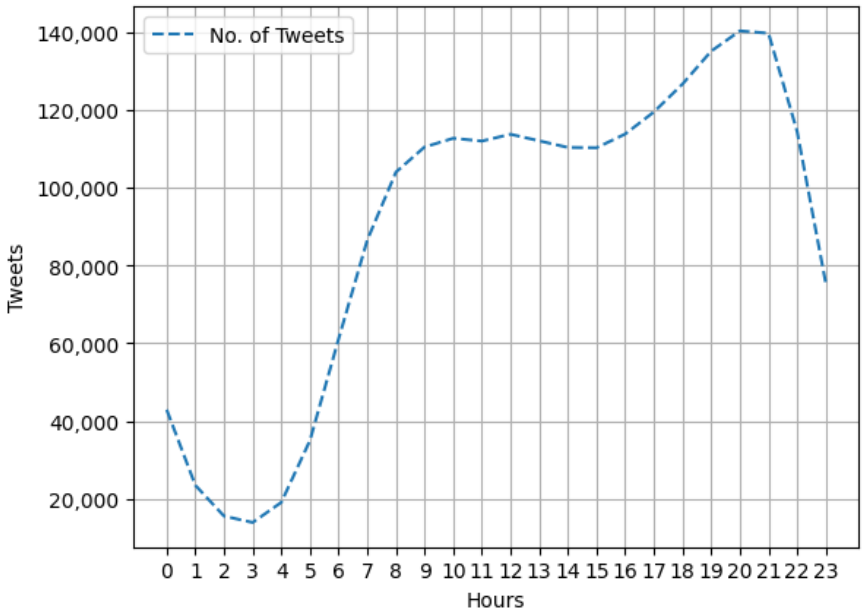


Fig 5: Time-series of tweets by hour, averaged over all days of the week

Comment on Pattern

Starting from 12 midnight, a decline in the number of tweets is observed as more and more people go to bed. This number starts picking up after 2 AM - 3AM as people start getting up (recall that there are a few time-zones captured in this data). The number climbs sharply till about 8 AM, which is likely when most users clock in to their daily work/ school, (or start commuting to work/ school etc.). Tweets stay plateaued at this range till around 3 PM, from where it starts hiking upwards again - people are likely leaving work/ school at this time. Number continues to increase until roughly 10 PM at night, which is when we see a sharp decline, likely due to users retiring for the day/ going to bed.

As a bonus, here is the pattern over the days separately instead of averaged across the week.

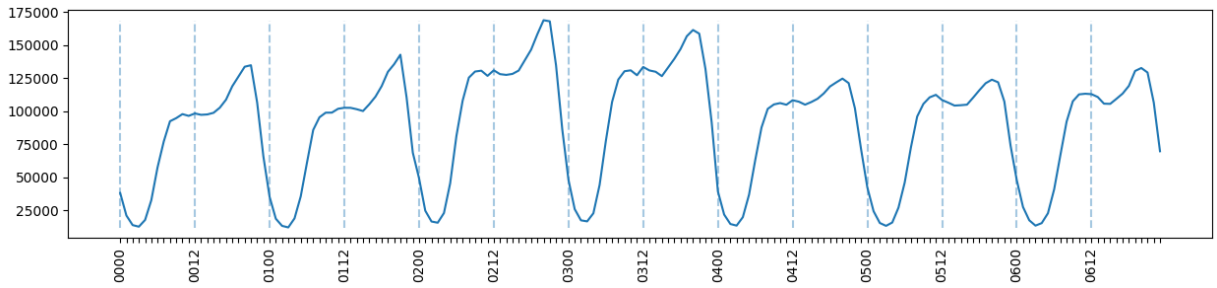


Fig 6: Time-series of tweet volume averaged at hours of a week. All 7 days from Monday to Sunday are represented here.

It seems like although the hourly pattern is mostly preserved in the number of tweets, Wednesdays and Thursdays see on average the highest volume of tweets.

Part 2: Users

1. Number of users against the number of tweets they have made.

```
twt_df_usr = twt_df.groupby('user').agg(num_twt = ('tweet_id' , 'nunique')).reset_index()
fig , ax = plt.subplots(figsize = (10 , 4));
freq , bins , patches = ax.hist(twt_df_usr.num_twt , bins = 250);
ax.set_ylabel('Users');
ax.set_xlabel('Number of tweets');
plt.title('Histogram for Number of Tweets made by  Users');
```

It is a highly right-skewed distribution, likely due to some accounts tweeting at an inordinate rate/ volume.

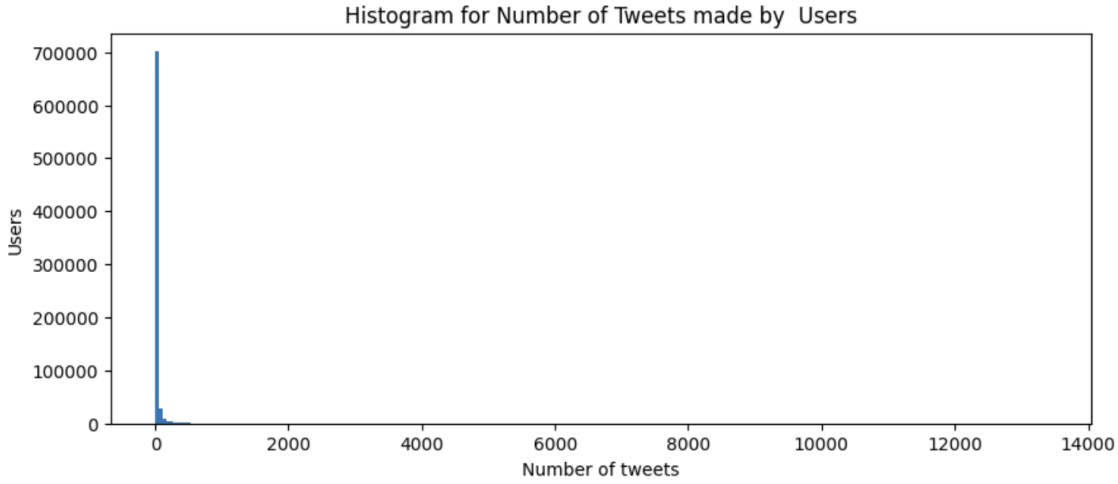


Fig 7: Histogram representing number of tweets made by users

It helps to understand that the majority of users (over 95%) have <100 tweets over the month of June.

```
print(f'{90}th quantile -> {np.quantile(twt_df_usr.num_twt , q = 0.9)}')
print(f'{95}th quantile -> {np.quantile(twt_df_usr.num_twt , q = 0.95)}')
print(f'{99}th quantile -> {np.quantile(twt_df_usr.num_twt , q = 0.99)}')
print(f'{100}th quantile -> {np.quantile(twt_df_usr.num_twt , q = 1.00)}')
```

Results:

```
90th quantile -> 37.0
95th quantile -> 76.0
99th quantile -> 289.0
100th quantile -> 13376.0
```

However, there are **some** users who have tweeted over 10,000 times, with the top user (by volume of tweets) having tweeted 13,376 times! Seeing that there are 30 days in our dataset, this user must have tweeted 13376/(30*24) = 18.6 times per hour, every hour of every day of the month.

Note: Perhaps, such high-volume users are bots. This will be explored further.

Log-transforming the data may give us some more tractable insights.

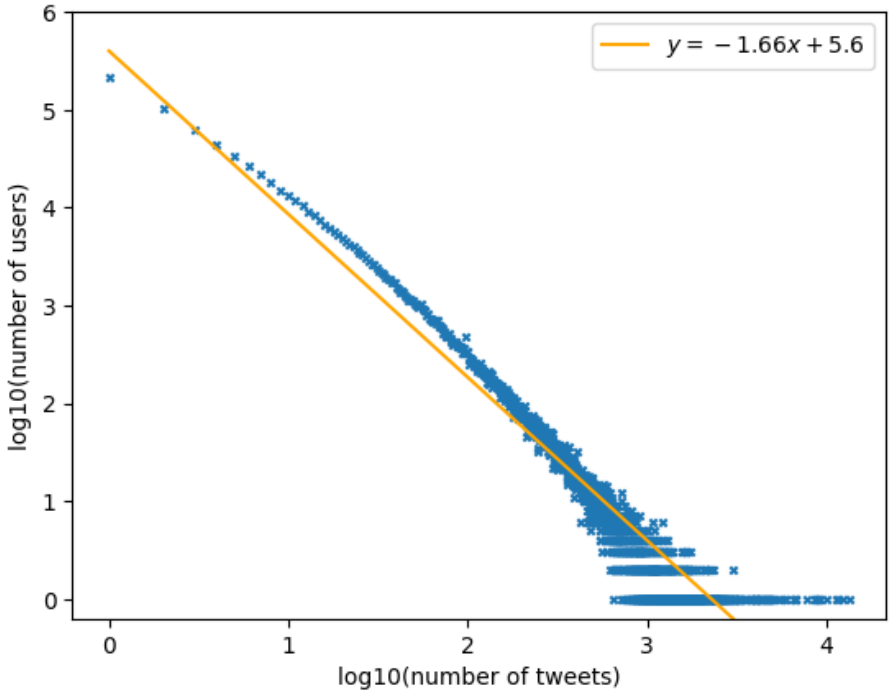


Fig 8: Log10 transforms of both axes – number of tweets, number of users shows a vaguely linear trend

The log-transformed data approximately follows a linear pattern. The orange line is a visual (*not very rigourous*) approximation of the equation the curve follows.

In the labelled equation, `Y = -1.66X + 5.6`, such that `Y = log10(y)` and `X = log10(x)`, where `y = number of users` and `x = number of tweets.`

Therefore, $y = 10^{5.6} x^{-1.666}$. A *power-law* relationship exists between users and tweets. If users doubled their current value, the number of tweets would become $2^{-1.666} = 0.314$ times their current value.

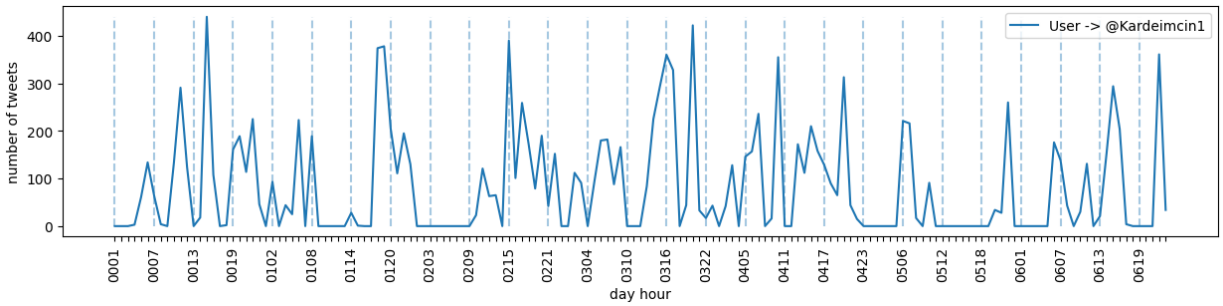
2. Top 5 Users by Total Tweets

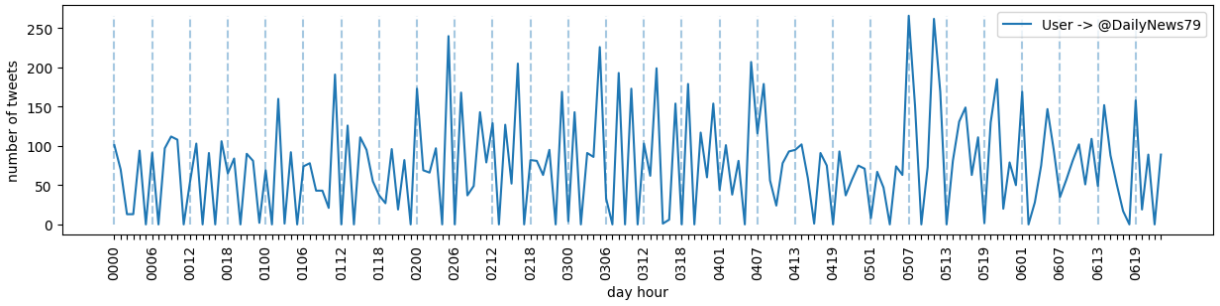
Grouping the number of tweets by user and selecting the top 5, we have the following:

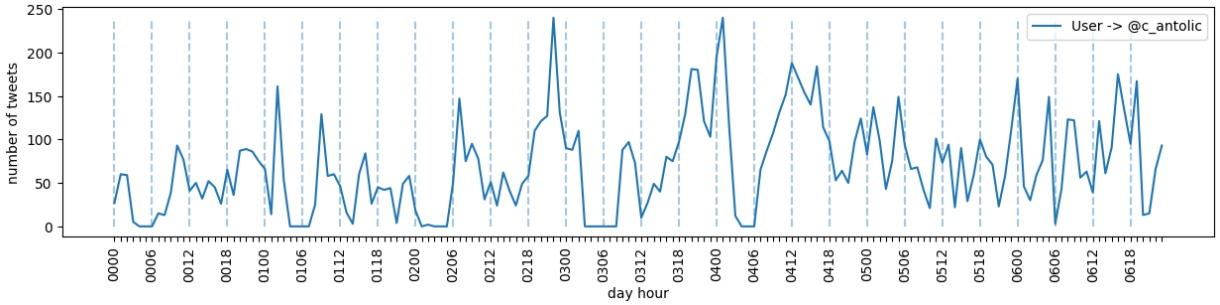
a. Büşra (@Kardeimcin1) – A Turkish account with politically inclined posts
b. Daily News Italy (@DailyNews79) – Twitter handle for Italian news service, which has been suspended at the time of writing this report.
c. Christian Antolic (@c_antolic) – German account tweeting single German words. At the time of writing this report, this account has been suspended.

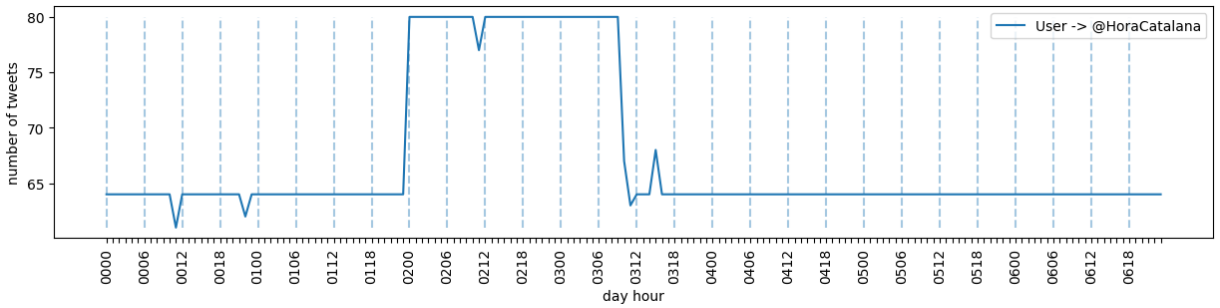
d. L’hora catalana (@HoraCatalana) – Posts the time in Catalan time system. Appears to be automated, based on frequency.
e. minijob-anzeigen.de (@minijobanzeigen) – German job advert website. The description of the account clearly mentions that it is automated, hence we can say it is a bot (*we have no reason not to trust the description*).

This can be further explored by looking at the patterns of their associated timestamps, i.e. the times when they have been posting content.









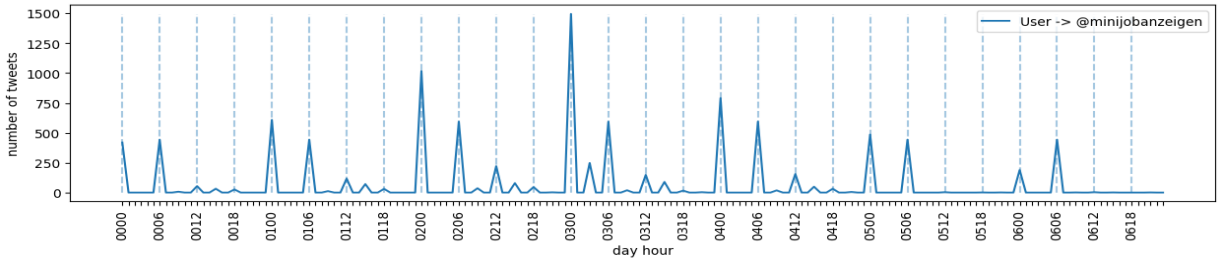


Fig 9: Time-series patterns of tweet volume from high-volume tweeters. All 7 days from Monday to Sunday are captured here at an hour level.

Comments:
Studying the top 3 graphs, there is no obvious pattern, which one might expect if it were a human that were making the tweets (perhaps, something close to Fig 6). For the fourth figure, a very uniform rate of tweets is observed (one can argue that it is far too uniform to be made by a human). The fact that it is the Twitter handle of a news channel can indicative of it being a bot. The last figure exhibits some patterns, but it is explicitly stated to be an automated account.

3. Users receiving the most mentions.

   For this, the user_mentions tag inside entities node was used. In case the tweet is truncated, all user mentions from within the extended_tweet root node were gathered.

```
# Mentions
mentions = []
try:
    # If there are multiple mentions, iterate thru them
    for mention in tweet.get('extended_tweet').get('entities').get('user_mentions'):
        mentions.append(mention.get('id'))
        # Add all mentions as a list element to the list of field
        list_of_field.append(mentions)
except AttributeError:
    try:
        # If there are multiple mentions, iterate thru them
        for mention in tweet.get('entities').get('user_mentions'):
        mentions.append(mention.get('id'))
        # Add all mentions as a list element to the list of field
        list_of_field.append(mentions)
    # If there are no mentions
    except:
        list_of_field.append("None")
```

Here are the top 5 accounts receiving the most mentions:

USER ID	***USER***
10228272	YouTube (Company)
68034431	Recep Tayyip Erdogan (President of Turkey)
3131144855	Boris Johnson (Former Prime Minister of the UK)
44196397	Elon Musk (Billionaire)
1354781680943431681	Baby Doge Swap (Cryptocurrency)[2]

Table 1: Highest-volume tweeters

[2] This is actually the 7th highest mentioned account. Two accounts belonging to political parties/personalities in the Indian state of Telengana are the 5th and 6th highest mentioned account. We have omitted them from the table as we do not believe European Twitter activity has much to do with said accounts.

These accounts are those belonging to companies, politicians and public figure(s).

4. How often do countries mention themselves/ each other?
   It is worth noting that only those countries explicitly present in the tweet json are considered. Four countries - France, Germany, Spain and Portugal are selected to compute patterns of how users from these countries mention (and are mentioned by) each other. Some further preprocessing is required as various countries are called by various names in various languages. For example, Germany is called *Allemagne* in French and *Deutschland* in German! As such, the names of countries are translated to their English counterparts[3].

   First, the user_id→mentioned_id dataframe is unpivoted.

```
%%time
# Changing from list representation to an actual list
twt_mentions_df_0['mentions'] = twt_mentions_df_0['mentions'].apply(lambda x :
json.loads(x))
twt_mentions_df = twt_mentions_df_0.explode('mentions')

print(f'the length of the exploded df is {len(twt_mentions_df)} and that of previous df is
{len(twt_mentions_df_0)}'
```

```
the length of the exploded df is 22502026 and that of previous df is 15033548
```

   Exploding has increased the number of records, as there may be multiple mentions in one tweet. From the tweet data, the user can be tied to their associated country by checking which country they tweet from the most.

   Note: The assumption here is that the country a user tweets most from is their country of residence/ origin for the purposes of this analysis.

   Since this establishes a link between user and country, it will also be helpful to locate the associated country of the mentioned user. This is required as the raw data does not hold any relation between mentioned user and their country. This also makes it such that any user who has not tweeted during the data capture period does not have an associated country – this is a caveat.

```
%%time
# Trying to get one primary country per user - the country column is only attached to the
user
twt_country_0 = twt_mentions_df_0[['user','country']].merge(pivoted_country_names,
left_on = 'country' , right_on = 'name' , how = 'left'
)
# Drop unnecessary cols
twt_country = twt_country_0.drop(['country' , 'name'] , axis = 1).rename(columns =
{'en':'country'})
# Diagnostics
twt_country.user.nunique() ,
len(twt_country[twt_country['country'].notnull()].drop_duplicates().user.unique())
```

```
Result → (753481, 459464)
```

```
%%time
country_df = twt_country[twt_country['country'].notnull()]
```

[3] https://www.kaggle.com/datasets/prasertk/country-name-in-different-languages

```
print(len(country_df) , len(country_df.user.unique()))

# How much have they tweeted from each country?
country_df_2 =  country_df.groupby(['user' , 'country']).agg(
    num_occr = ('user','count')).sort_values('num_occr' , ascending = False
    ).reset_index()

print(country_df_2.num_occr.sum() , len(country_df_2))
country_df_2['rank'] = country_df_2.groupby('user')['num_occr'].rank(method = 'first' ,
ascending = False)
# Main associated country based on volume of tweets
prim_country = country_df_2[country_df_2['rank']==1].drop(['num_occr' , 'rank'] , axis = 1)
print(f'length of df ->  {len(prim_country)}')
```

```
length of df -> 459464
```

Some people appear to have moved around quite a lot!

```
country_df_2[country_df_2['user']=='774694926135222272']
```

Here is the result!

user	country	num_occr	rank
774694926135222272	Germany	30	1
774694926135222272	Russia	16	2
774694926135222272	Spain	16	3
774694926135222272	France	12	4

Table 2: (Illustrative) Country having highest number of tweets from user is assigned as users' primary country (highlighted in yellow)

```
prim_country[prim_country['user']=='774694926135222272'].head(3)
```

user	country
774694926135222272	Germany

This account is interesting as it has tweeted from over 30 locations. Upon some investigation, it is revealed to be a bot, tweeting about various cities as a parody of Milton Glaser's "I ❤ NY" design.[4]

In this part, dask, a distributed extension to pandas was used to speed up processing times.

```
# drop tweets with no mentions
twt_mentions_df.dropna(inplace = True)
```

[4] https://www.creativereview.co.uk/i-love-ny-logo/

```
# Use dask to parallelize pandas stuff!
dask_mentions_df = from_pandas(twt_mentions_df, npartitions=8)

# Make data types consistent before joining
dask_mentions_df['user'] = dask_mentions_df['user'].astype('str')
dask_mentions_df['mentions'] = dask_mentions_df['mentions'].astype('str')
# Country df for joining later
dask_user_country_df = from_pandas(prim_country , npartitions=8)

# Joining country names to user and mentioned IDs
mentions = dask_mentions_df.merge(
    dask_user_country_df[['user' , 'country']] , on = 'user' , how = 'inner'
    ).merge(
        dask_user_country_df[['user' , 'country']] , left_on = 'mentions' , right_on =
'user' , how = 'inner'
    )

# See who's mentioning whom?
x_mentions_df = mentions.compute()
x_mentions_df.rename(columns = {'country_x':'user_country' ,
'country_y':'mentioned_country'}, inplace = True)
```

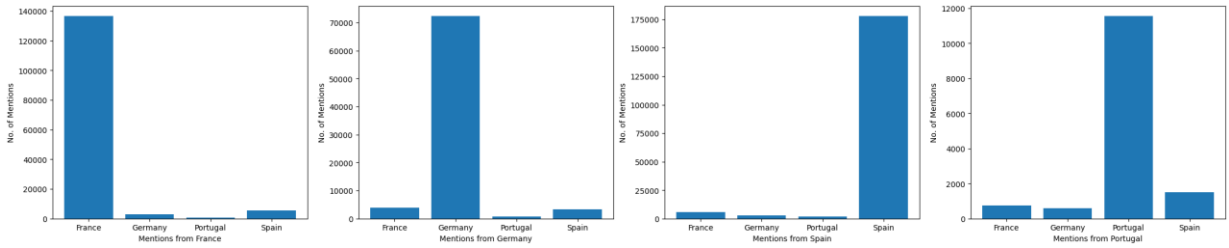


Fig 10: Cross-country mentions

It can be seen that countries mostly mention themselves!

Data is tabulated below in Table 3:

mentioned_country	France	Germany	Portugal	Spain
user_country				
France	136643	2822	729	5408
Germany	3832	72342	771	3315
Portugal	735	587	11546	1518
Spain	5541	2792	1591	177834

Table 3.1: Tabulation User of Country to Mentioned Country. Read as element i,j = x → Country i mentioned Country j x times.

mentioned_country	France	Germany	Portugal	Spain
user_country				
France	0.93846925	0.01938160	0.00500679	0.03714234
Germany	0.04774482	0.90134562	0.00960628	0.04130326
Portugal	0.05109133	0.040803559	0.802585847	0.105519255
Spain	0.02951139	0.014870205	0.008473674	0.947144729

Table 3.2: Percentage Tabulation of User Country to Mentioned Country

Part 3: Mapping

1. A Map of Europe that displays the use of Twitter across the continent, using only the GPS-tagged tweets.

```
# Use tweets and coordinates to create a mapping dataframe
mapping_df = pd.DataFrame(
    list(zip(tweets , coord)) , columns = ['tweet_id' , 'coordinates']
)
# Removing unnecessary rows
mapping_df = mapping_df[(mapping_df['tweet_id']!="None")&
(mapping_df['coordinates']!="None")].drop_duplicates()
len(mapping_df)
```

```
701961
```

Having extracted coordinates (longitude and latitude) from the twitter activity data, a map of Europe is overlaid[5] against the said activity, as showcased in Fig 11.

```
# Create custom colour map
cmap = LinearSegmentedColormap.from_list('custom blue', [(0,'#ffff00'),(1,'#8B0000')],
N=256)

# Make subplot
fig , ax = plt.subplots(figsize = (15 ,8))

# Overlay map
p = PatchCollection(patches , edgecolor = 'white' , lw = 1.5 , facecolor = 'black');
ax.add_collection(p);

# Hexbin for tweets
ax1 = ax.hexbin(x_coord , y_coord , gridsize = 1000 , bins = 'log' , cmap = cmap , alpha =
0.5);
cbar = fig.colorbar(ax1);
cbar.set_label('Number of Tweets');
```

[5] Shapefile for Europe and neighbouring countries was taken from: https://ec.europa.eu/eurostat/web/gisco/geodata/reference-data/administrative-units-statistical-units/nuts

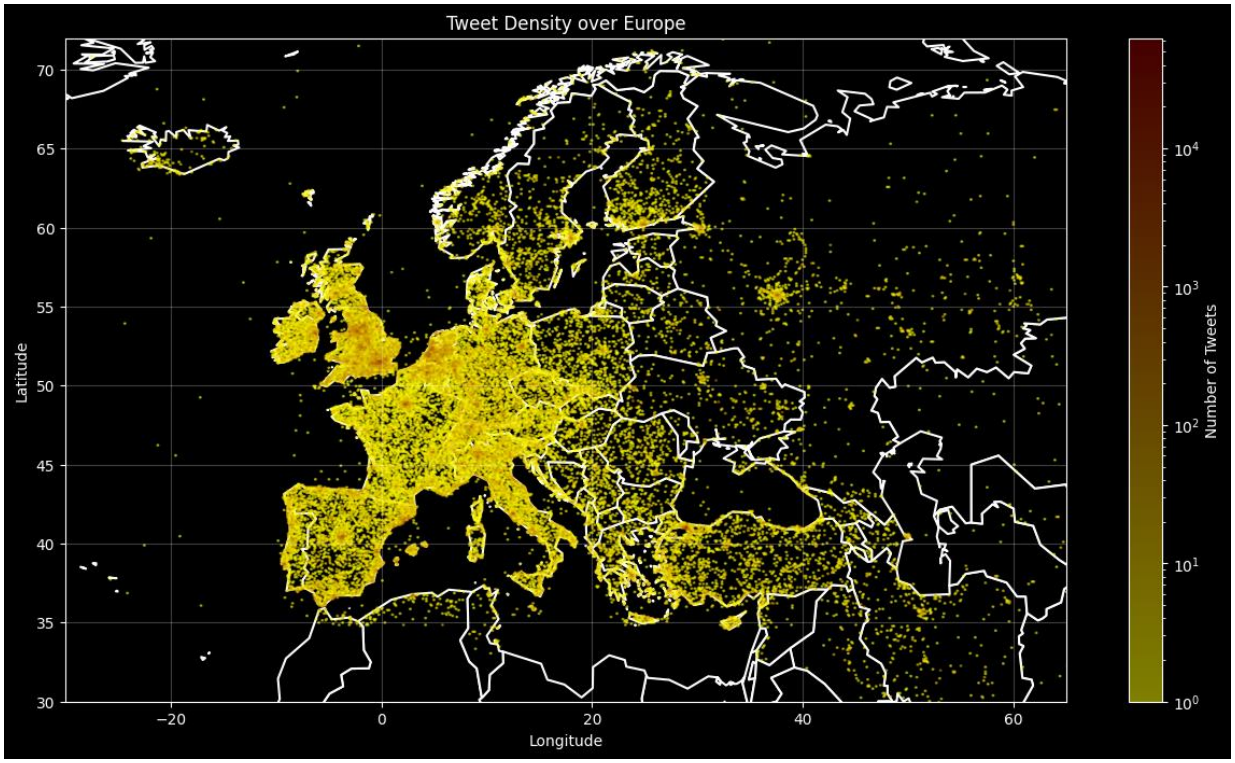


Fig 11: Twitter activity overlaid on top of geographical boundaries. Red signifies higher amount of tweets

2. **Patterns observed:**

1. The hotspots are centered around places of high population. The further north one goes, the lesser tweets show up - indicating (most likely) a drop in population. Rhe map is bright red over the capitals, especially Dublin, Paris, Madrid, Berlin, Amsterdam, Copenhagen, Warsaw, Istanbul etc.
2. Large concentrations of tweets also come from big cities which aren't capitals - examples, Barcelona, Malaga. These are also big population hubs.
3. There is a lot more activity along the coastlines, but this is likely due to the presence of big port cities like Nice and Barcelona.
4. Small islands of Ibiza and Malta are hotspots!
5. Some tweets arise from the middle of large water bodies – the Mediterranean Sea, the Atlantic Ocean, the Black Sea, the Caspian Sea.

tweet_id	coord	long	lat	Bot?
1.53334E+18	[-21.27, 46.22]	-21.27	46.222218	Bot
1.53351E+18	[-24.28, 53.92]	-24.28	53.921337	Bot
1.54062E+18	[-21.77, 59.35]	-21.77	59.353892	Bot

6. Tweets from Ukraine seem to be sparse – perhaps this can be attributed to internet outages[6]

3. The other tweets have a ‘place’ tag inside which is the ‘bounding box’ tag. The CDF of the diagonals is plotted in Fig 12. Some preprocessing →

```
length after removal of null tweets = 15,033,548
length after removal of null bounding boxes = 15,026,765, i.e. ~ 0.05% did not have proper bounding boxes
length after removal of non-null exact co-ord = 14,331,587, i.e. ~ 4.62% had exact coordinates
```

[6] https://freedomhouse.org/country/ukraine/freedom-net/2022

In order to get the diagonals, only two opposite pairs of points of the bounding box are required –the first (bb1) and the third (bb2) are chosen.

```
# Construct diagonals
bounding_box_df['diagonal'] = np.sqrt((bounding_box_df['bb1_long'] -
bounding_box_df['bb2_long'])**2 + (bounding_box_df['bb1_lat'] -
bounding_box_df['bb2_lat'])**2)
```

For ease of representation, a log scale is chosen for the y-axis on the right.

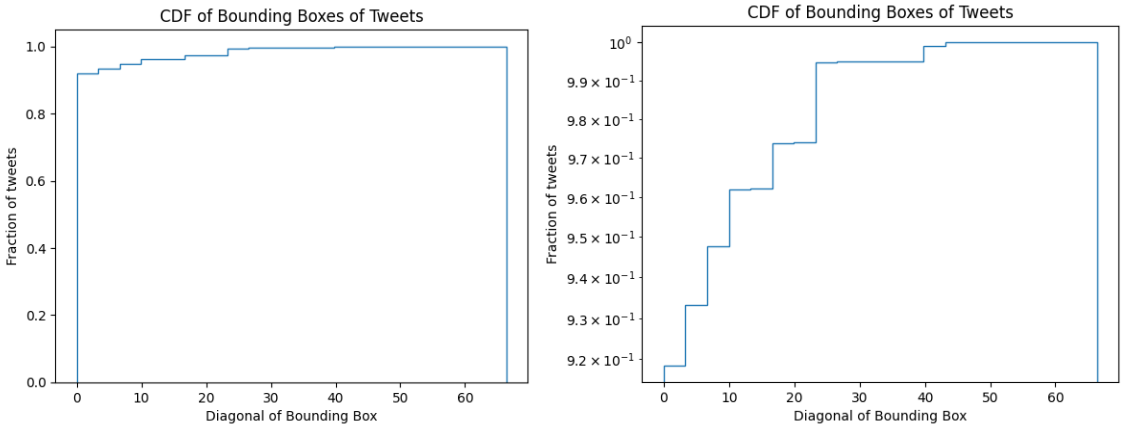


Fig 12.1: CDF of bounding box diagonals represented on i) equal axis and ii) log-transformed y-axis

Comments:
Over 90% of the data has small bounding boxes. There are however, tweets with huge bounding boxes seem to be originating firstly in Greenland & secondly, around Norway and Finland.

There may be few reasons for this. From a Euclidean geometrical perspective, it may seem odd that the bounding boxes are so big. Recall that, close to the poles, the longitudes bunch up together. Hence, the diagonals appear larger - it is the same reason why Greenland is disproportionately bigger on flat maps. However, Fig 12.3, showing two of the largest bounding boxes in the data can contest this theory.t

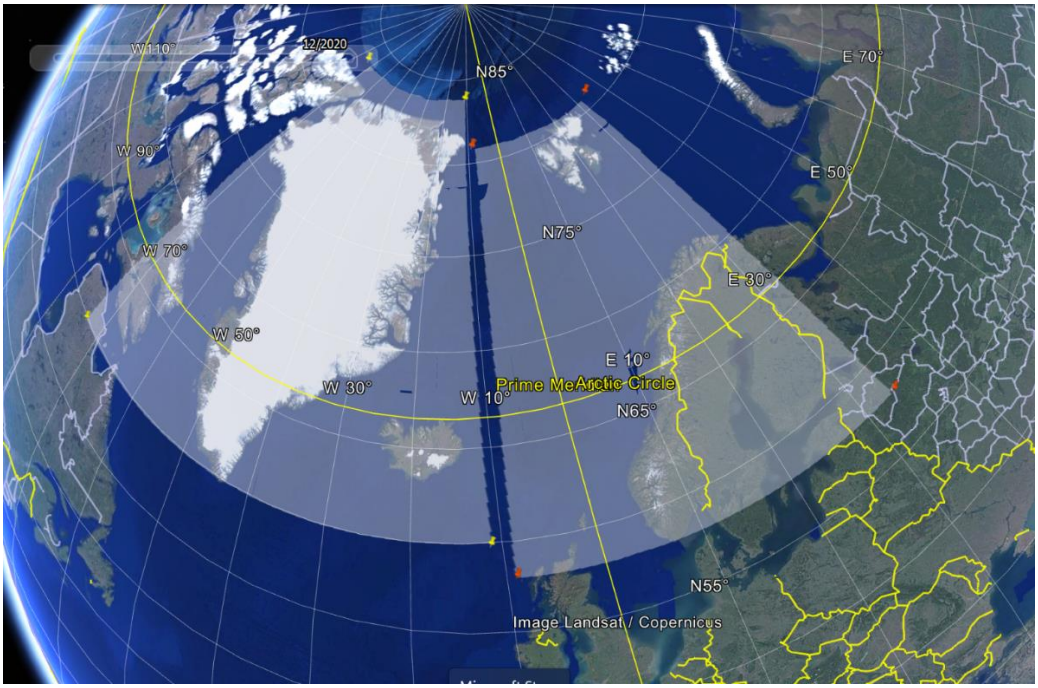


Fig 12.2: Two of the largest bounding boxes in the data

It appears that the initial theory is not quite applicable. Although the longitudes do concentrate closer to the North Pole, the bounding boxes are still massive in relation to most of the others in the set. All of the tweets with the largest size of bounding box have **the exact same bounding box** (pale white patch over Greenland, as shown in Fig 12.3). Second largest is that over Scandinavia (as shown in Fig 12.3). Perhaps there are only a few select cell towers in and around the Arctic, where the location data is pinged from. Without further data and exploration however, this is all hyperbolae.

4. Another additional spatial dataset is compared against the Twitter activity. For this, the greenery coverage[7] in the city of London is considered- roughly the coordinates are between (51.3 , -0.5) to (51.6, 0.3) in Fig 13. Roadmap of London is overlaid for clarity.

   Note: `OSGB36toWGS84` from the `bng_latlon` library was used to convert Northing-Easting to latitude-longitude.

```
plt.style.use('dark_background')
# Make subplot
fig , ax = plt.subplots(figsize = (10,6))
# Create custom colour map
cmap = LinearSegmentedColormap.from_list('custom blue', [(0,'#ffff00'),(1,'#8B0000')],
N=256)

p1 = PatchCollection(patches , edgecolor = 'white' , lw = 0.85 , facecolor = 'black');
ax.add_collection(p1);

p2 = PatchCollection(patches_greenery , edgecolor = 'green' , lw = 0.01 , facecolor =
'green');
ax.add_collection(p2);

# Scatter for tweets
ax1 = ax.hexbin(x_coord , y_coord , gridsize = 300 , bins = 'log',cmap = cmap, alpha =
0.75);
# ax.scatter(x_coord, y_coord,s=0.2,c='red')
cbar = fig.colorbar(ax1);
cbar.set_label('Number of Tweets');
```

[7] Shape-file for London greenery taken from - https://geospatialwandering.wordpress.com/2015/05/22/open-spaces-shapefile-for-london/comment-page-1/

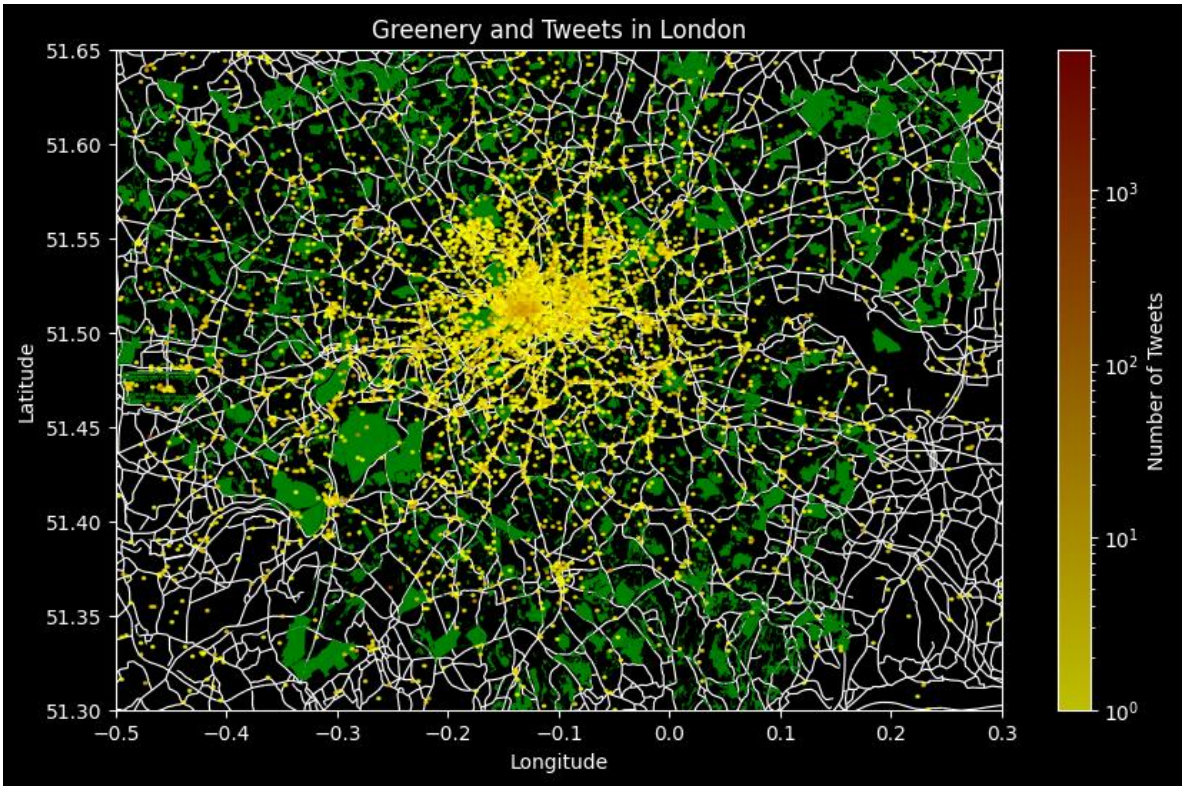


Fig 13: Though not immediately obvious, we can see that areas of greenery generally showcase lower tweet volume.

Apart from one hotspot in Richmond Park (the biggest green patch, slight left and below the centre of the image), other green parts generally show a low amount of twitter usage, if at all. This can perhaps be explained by the confounding variable - population density, to some extent. Parks are generally less crowded than spaces in the city centre.

Part 4: Events

1. United Kingdom, France and the Netherlands are selected as countries of interest. Similar to Part 2, names are standardized to their English representation/version.

   The tweet data of these three countries are placed in three separate dataframes as number of tweets by day. Then a technique adjacent to Bollinger bands[8] is used to determine days of unusual activity. Other anomaly detection techniques like isolation forests may also be considered, but are probably excessive for this.

```
# Anomaly Detection using Bollinger Bands-adjacent method
fig , ax = plt.subplots(figsize = (15,4))

rate = 3
tolerance = 1
ax.plot(ts_1 , c = 'olive' , label = f'Signal for {ts_1.name}');
ax.plot(ts_1.rolling(rate).mean() , lw = 1.5 , ls = "--" , c = 'red' , label = f'Rolling
mean for {rate} days');
ax.plot(ts_1.rolling(rate).mean() + tolerance*ts_1.rolling(rate).std() , lw = 1.5 , ls = ":"
, c = 'orange' , label = f'mean $\pm${tolerance}*std dev');
ax.plot(ts_1.rolling(rate).mean() - tolerance*ts_1.rolling(rate).std() , lw = 1.5 , ls = ":"
, c = 'orange');
ax.legend(loc='upper left');
ax.grid();
```

[8] Bollinger J., (1992) Stocks & Commodities V. 10:2 (47-51): Using Bollinger Bands

```
ax.set_xticks(ts_1.index);
ax.set_xticklabels(labels = ts_1.index, rotation = 90);
```

Here are successive outputs for UK (ts_1), France (ts_2) and the Netherlands (ts_3).

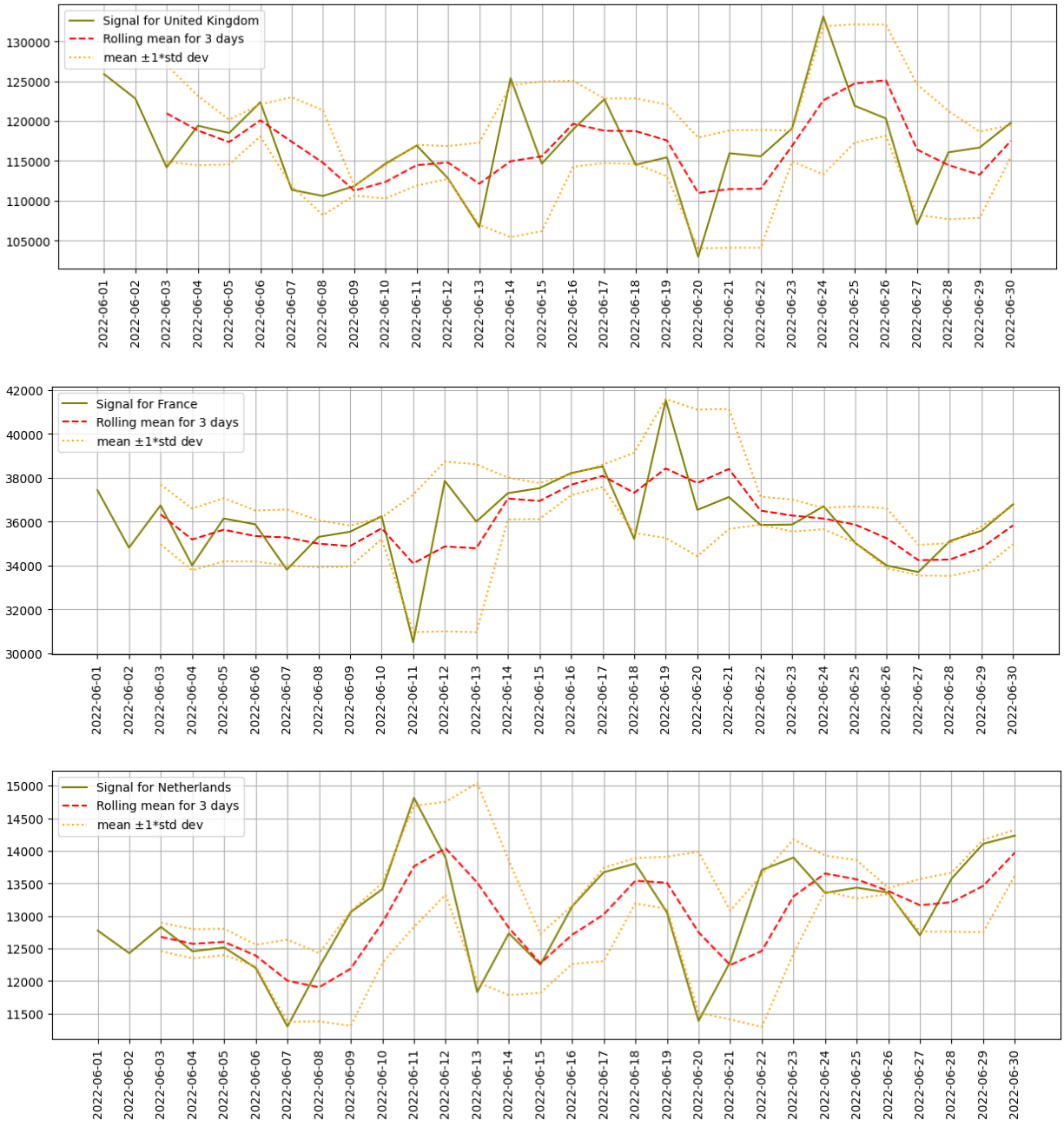


Fig 14: Time-series data on tweet volume in the UK, France and the Netherlands across June'22.

Days of excessive tweeting are filtered by checking if the value crosses the threshold of a 3-day rolling mean + 1 standard deviation. As such, from the above graph, the following days stand out as days of 'anomalous' activity:

- 24th June in United Kingdom
- 19th June in France
- 11th June in the Netherlands

2. A) The text from the twitter data is extracted to build word clouds from it. The 'extended_tweet' root node (where it is available) is checked for more text if it has been truncated in the root node 'text'.

Preprocessing - some stop-words have been removed in the respective languages to cast away uninformative words. This can of course be improved, but is approximately good enough for our purposes.

```
from spacy.lang.fr.stop_words import STOP_WORDS as fr_stop
from spacy.lang.en.stop_words import STOP_WORDS as en_stop
from spacy.lang.nl.stop_words import STOP_WORDS as nl_stop

en_lang_stopwords_list = list(en_stop) +
['https','co','t','c','d','l','j','s','good','thank','day','year','week','today','amp','time
','great','like','people','love','new','think','no','yes','going','need','work','right','kno
w','look','got','m','thanks',"I'm",'morning','hope','want','ve','way',"don't",'best','years'
,'come','don','photo','Happy','thing','night','posted','better','X','Friday','let','lot','li
ve','life','looking','Oh','London','world','weekend','lovely']
fr_lang_stopwords_list = list(fr_stop) + ['https','co','t','c','d','l','j','s','Foto']
nl_lang_stopwords_list = list(nl_stop) +
['https','co','t','c','d','l','j','s','i','la','le','a','to','the','you','and','it','dag','v
andaag','goed','Beautiful','that','gaat','Goedemorgen']
```

Define a simple function to plot word clouds.

```
def plot_wordcloud(df , stopwords_list):
    '''
    plots the word cloud given a dataframe and a list of stopwords
    params: df -> country and date dataframe
    params: stopwords_list -> list of stopwords language specific
    returns: None
    '''
    text = ' '.join(df['text'].tolist())
    word_cloud = WordCloud(background_color = 'black' , stopwords = stopwords_list,
collocations = False).generate(text)
    # Plot
    fig , ax = plt.subplots()
    ax.imshow(word_cloud, interpolation='bilinear')
    ax.axis("off")
```

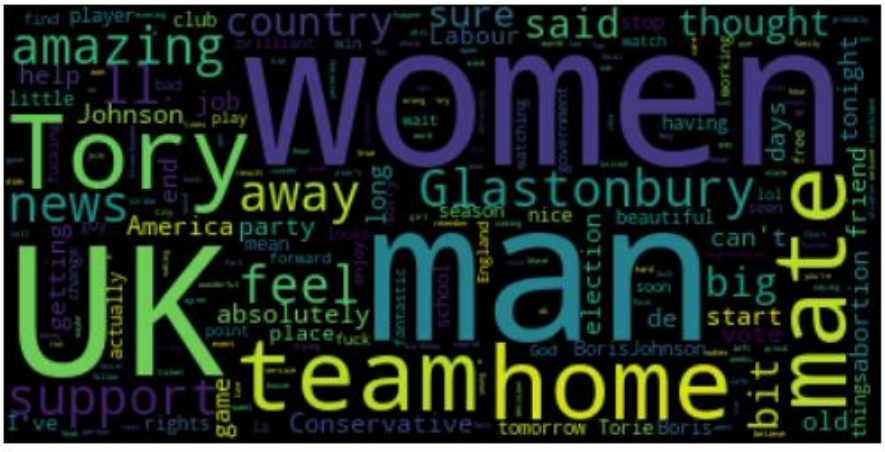


Fig 15: Word cloud showing prominent words in tweets originating in UK on 24th June, 2022

This was around the time Rowe-v-Wade was overturned in the USA[9]. Consequently, a lot of solidarity and support is seen from the UK[10], evinced by prominent mentions of "women", "America" and

[9] https://www.american.edu/cas/news/roe-v-wade-overturned-what-it-means-whats-next.cfm

[10] Graham, S. (2022) "We're horrified by the rejection of Roe v Wade—but abortion is not a universal right in the UK", *BMJ*, p. o1945. doi: 10.1136/bmj.o1945.

"abortion". Political discussions were also quite widespread on this day with prominent mentions of "Boris Johnson", "Conservative", "Labour" and "Tory".

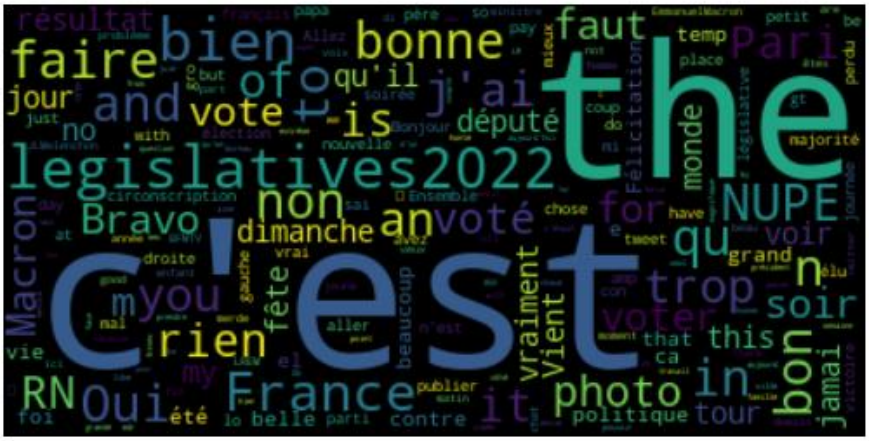


Fig 16: Word cloud showing prominent words in tweets originating in France on 19th June, 2022

Around this day in France, the Legislative Assembly 2022 was being held[11], marked by mentions of "legislatives2022", "NUPE[12]" and "Macron".

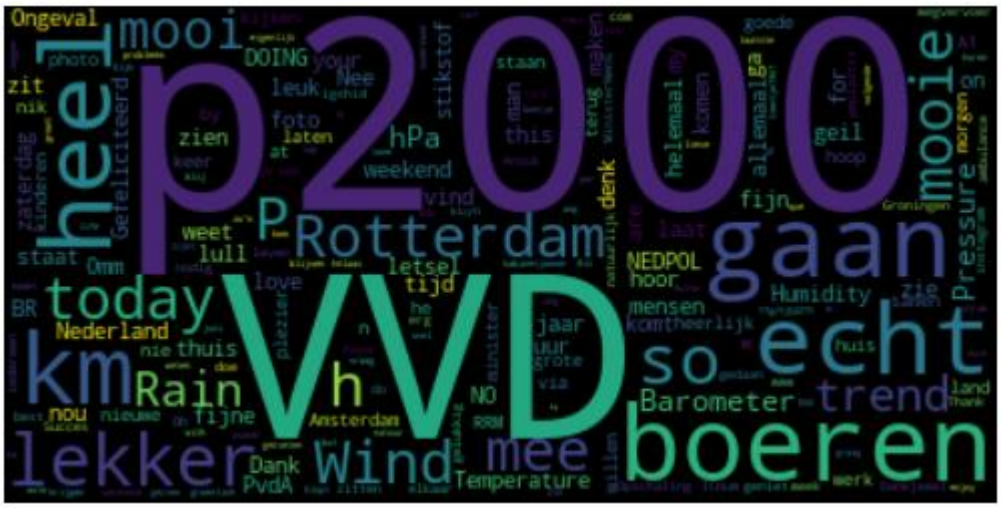


Fig 17: Word cloud showing prominent words in tweets originating in the Netherlands on 11th June, 2022

VVD is a political party in the Netherlands, which seems to be mentioned quite often. "Boeren" – the Dutch word for farmer is also seen to appear prominently. This may be in relation to Dutch farmer protests[13] that have been going strong since 2019. However, even more interesting is the word "p2000". Upon some investigation, it appears to be the Twitter equivalent of an emergency service call-out. "*The network is used by all emergency services and provides nationwide coverage. Several tests have shown the network can cope with the largest disasters when large numbers of emergency personnel need to be reached. The P2000 network is maintained by the Ministry of the Interior and Kingdom Relations*".[14] It thus stands to reason that this would be the mostly widely tweeted word.

[11] https://www.france24.com/en/france/20220619-live-france-votes-in-parliamentary-election-second-round
[12] https://en.wikipedia.org/wiki/New_Ecological_and_Social_People%27s_Union
[13] https://www.bbc.co.uk/news/world-europe-62335287
[14] https://en.wikipedia.org/wiki/P2000_(network)

B) Another way to characterize these days is with the use of hashtags. For these, 'extended_tweets' node was used again. The month's data is split for each country into two:

- that on the day of interest and
- that on every other day (i.e. apart from the day of interest).

All their hashtags were collected as lists. A small function helped to visualize the percentage occurrence of said hashtags.

```
def make_hashtag_df(hashtag_list):
    '''
    makes a df of the hashtag occurences
    params: hashtag_list_counted -> list containing all occurences of hashtags
    returns: df -> dataframe containing hashtags and occurences
    '''
    hashtag_list_counted = Counter(hashtag_list)
    L = []
    for el in hashtag_list_counted:
        L.append([el , hashtag_list_counted[el]])
    df = pd.DataFrame(L , columns = ['hashtag','occurence']).sort_values('occurence' ,
ascending = False)
    df['%occurence'] = df['occurence']/df['occurence'].sum()
    return df
```

For the United Kingdom, on the day of interest, i.e. 24th June, 2022, here are the most common hashtags →

hashtag	occurrence	fraction_occurence
LoveIsland	444	0.01015
Glastonbury2022	412	0.00942
RoeVsWade	373	0.00853

Table 4.1: Most common hashtags in UK, 24th June

Comparing that to other days,

hashtag	occurrence	fraction_occurrence
LoveIsland	11947	0.0104
PlatinumJubilee	6478	0.00564
loveisland	3187	0.00278

Table 4.2: Most common hashtags in UK, 24th June not included

Though LoveIsland seems to be a consistently predominant topic, the overturning of Roe Vs Wade seems to have been heavily discussed on this particular day. Glastonbury 2022, held between 22nd – 26th June, 2022 seems to have been a highly discussed topic as well.

For France, on the day of interest – 19th June, here are the most common hashtags →

hashtag	occurrence	fraction_occurrence
legislatives2022	1345	0.08898
NUPES	178	0.01178
chat	146	0.00966

Table 5.1: Most common hashtags in France, 19th June

Comparing that to other days,

hashtag	occurrence	fraction_occurrence
legislatives2022	8030	0.018948
chat	4085	0.009639
NUPES	3738	0.00882

Table 5.2: Most common hashtags in France, 19th June not included

Although we do not see different hashtags on the day of interest, it is worth noting that the fractional occurrence of "legislatives2022" has increased from 1.8% to 8.8%, likely hinting at the increased discussion around this topic on 19th June.

For the Netherlands, on the day of interest, i.e. 11th June, 2022, here are the most common hashtags →

hashtag	occurrence	fraction_occurence
p2000	403	0.07326
NEDPOL	168	0.03054
RRM	94	0.01709

Table 6.1: Most common hashtags in the Netherlands, 11th June

Comparing that to other days,

hashtag	occurrence	fraction_occurence
p2000	11277	0.081272
brugopen	2573	0.018543
ambulance	2317	0.016698

Table 6.2: Most common hashtags in the Netherlands, 11th June not included

It is interesting to see that ratio of p2000 tweets actually dropped slightly on the day of interest. Hence, the next obvious candidate for what special thing happened that day is #NEDPOL, referring to the football match that day between the Netherlands and Poland in the UEFA Nations League.

3. In summary, we have found the following days, characterized by the accompanying topics, to have the highest tweet volume across three countries – United Kingdom, France and the Netherlands:

Country	Date	Topic of Discussion	Validation
United Kingdom	24th June, 2022	Rowe v Wade being overturned	https://www.bbc.co.uk/news/world-us-canada-54513499 https://www.law.cornell.edu/supremecourt/text/410/113
France	19th June, 2022	Legislative Elections being held	http://fingfx.thomsonreuters.com/gfx/rngs/FRANCE-ELECTION/010041FQ34H/index.html#section/polling2
The Netherlands	11th June, 2022	Netherlands vs Poland football match	https://www.uefa.com/uefanationsleague/match/2034476--netherlands-vs-poland/

Table 7: Validation by news sources

Part 5: Reflection

Use of Twitter data across academia, industry, governments and media to inform decision-making by advancing better understanding of the myriad opinions and behaviour that humans display has increasingly led to concerns over ethics. When the public ticks the boxes and consents to their data becoming public, extractable, analysed and used, do they really understand the extent to which they are becoming accessible?

Twitter often becomes a convenient choice of data source for conducting analyses on elections, sports events, financial markets (Bollen, J., Mao, H. and Zeng, X. (2011))[15], public health crises (Signorini, A., Segre, A. and Polgreen, P. (2011))[16], extreme weather events, natural disasters (Bakshi, H. (2011)) [17] and many more. It often represents spatio-temporal patterns in far greater resolution and granularity than many traditional data sources (Malleson, N. and Andresen, M. (2015)) [18]. But it is, at its core, dealing with human participants, and their personal data. Although widespread public availability of Twitter data in conjunction with computational advancements in natural language processing has furthered social outreach efforts (*such as public health monitoring*), research ethics committees are yet to converge on ethically appropriate usage guidelines for this data source.

From a technical perspective, interpreting Twitter data at a population-level may be accompanied with statistical concerns. Researchers must exercise caution when attempting to generalize conclusions based on such data. For example, in order to make any statement from analysis of geo-tagged[19] tweets, one would need to know the percentage of the population that actually is represented through this[18]. Reports on misuse of social media data, be it intentional or unintentional, are not unknown (Boldyreva*, E., Grishina, N. and Duisembina, Y. (2018)) [20]. Twitter is but an example of the collective of social media data sources, rich with personal information of users, without adequate measures of data privacy protection in place (Alharthi, R., Alhothali, A. and Moria, K. (2019)) [21].The blame cannot be placed on the users of Twitter for not reading the small print (or the very long one), as it is important to recognise that an increasing number of websites and software make the process of having transparent information rather bureaucratic and exhausting, thus making people, with limited time, skip over. Information on any ethical issue must be publicised with greater emphasis and clarity.

If one were to draw parallels to traditional survey studies, guidelines explicitly letting survey participants know the extent of their participation/ contribution in these studies. However, this is a grey area with using Twitter data – the volume of tweets available provides a base level of anonymity. Moreover, reaching out to users is in breach of GDPR (https://blog.ukdataservice.ac.uk/is-using-twitter-data-ethical/) [22]. Thus, there needs to be more research into the ethical use of Twitter (and more broadly, social media) data. In a world where residential addresses can be reverse-engineered from a person's social media activity (https://nextshark.com/hibiki-sato-japanese-idol-obsessive-fan/) [23], putting them at risk of targeted assault, addressing not just data privacy issues, but data ethics concerns as a nuanced whole, is an urgent need.

[15] Bollen, J., Mao, H. and Zeng, X. (2011) "Twitter mood predicts the stock market", *Journal of Computational Science*, 2(1), pp. 1-8. doi: 10.1016/j.jocs.2010.12.007.
[16] Signorini, A., Segre, A. and Polgreen, P. (2011) "The Use of Twitter to Track Levels of Disease Activity and Public Concern in the U.S. during the Influenza A H1N1 Pandemic", *PLoS ONE*, 6(5), p. e19467. doi: 10.1371/journal.pone.0019467.
[17] Bakshi, H. (2011) *Framework for crawling and local event detection using twitter data*, *Rucore.libraries.rutgers.edu*.
[18] Malleson, N. and Andresen, M. (2015) "Spatio-temporal crime hotspots and the ambient population", *Crime Science*, 4(1). doi: 10.1186/s40163-015-0023-8.
[19] https://developer.twitter.com/en/docs/tutorials/filtering-tweets-by-location
[20] Boldyreva*, E., Grishina, N. and Duisembina, Y. (2018) "Cambridge Analytica: Ethics And Online Manipulation With Decision-Making Process", *The European Proceedings of Social and Behavioural Sciences*. doi: 10.15405/epsbs.2018.12.02.10.
[21] Alharthi, R., Alhothali, A. and Moria, K. (2019) "Detecting and Characterizing Arab Spammers Campaigns in Twitter", *Procedia Computer Science*, 163, pp. 248-256. doi: 10.1016/j.procs.2019.12.106
[22] *Part 3: Is using Twitter data ethical? – Data Impact blog* (2022). Available at: https://blog.ukdataservice.ac.uk/is-using-twitter-data-ethical/
[23] https://nextshark.com/hibiki-sato-japanese-idol-obsessive-fan/

Part 6: References

1. Skovlund, E. and Fenstad, G. (2001) "Should we always choose a nonparametric test when comparing two apparently nonnormal distributions?", *Journal of Clinical Epidemiology*, 54(1), pp. 86-92. doi: 10.1016/s0895-4356(00)00264-x.
2. –
3. https://www.kaggle.com/datasets/prasertk/country-name-in-different-languages
4. https://www.creativereview.co.uk/i-love-ny-logo/
5. Shapefile for Europe and neighbouring countries was taken from: https://ec.europa.eu/eurostat/web/gisco/geodata/reference-data/administrative-units-statistical-units/nuts
6. https://freedomhouse.org/country/ukraine/freedom-net/2022
7. Shape-file for London greenery taken from - https://geospatialwandering.wordpress.com/2015/05/22/open-spaces-shapefile-for-london/comment-page-1/
8. Bollinger J., (1992) Stocks & Commodities V. 10:2 (47-51): Using Bollinger Bands
9. https://www.american.edu/cas/news/roe-v-wade-overturned-what-it-means-whats-next.cfm
10. Graham, S. (2022) "We're horrified by the rejection of Roe v Wade—but abortion is not a universal right in the UK", *BMJ*, p. o1945. doi: 10.1136/bmj.o1945.
11. https://www.france24.com/en/france/20220619-live-france-votes-in-parliamentary-election-second-round
12. https://en.wikipedia.org/wiki/New_Ecological_and_Social_People%27s_Union
13. https://www.bbc.co.uk/news/world-europe-62335287
14. https://en.wikipedia.org/wiki/P2000_(network)
15. Bollen, J., Mao, H. and Zeng, X. (2011) "Twitter mood predicts the stock market", *Journal of Computational Science*, 2(1), pp. 1-8. doi: 10.1016/j.jocs.2010.12.007.
16. Signorini, A., Segre, A. and Polgreen, P. (2011) "The Use of Twitter to Track Levels of Disease Activity and Public Concern in the U.S. during the Influenza A H1N1 Pandemic", *PLoS ONE*, 6(5), p. e19467. doi: 10.1371/journal.pone.0019467.
17. Bakshi, H. (2011) *Framework for crawling and local event detection using twitter data*, *Rucore.libraries.rutgers.edu*.
18. Malleson, N. and Andresen, M. (2015) "Spatio-temporal crime hotspots and the ambient population", *Crime Science*, 4(1). doi: 10.1186/s40163-015-0023-8.
19. https://developer.twitter.com/en/docs/tutorials/filtering-tweets-by-location
20. Boldyreva*, E., Grishina, N. and Duisembina, Y. (2018) "Cambridge Analytica: Ethics And Online Manipulation With Decision-Making Process", The European Proceedings of Social and Behavioural Sciences. doi: 10.15405/epsbs.2018.12.02.10.
21. Alharthi, R., Alhothali, A. and Moria, K. (2019) "Detecting and Characterizing Arab Spammers Campaigns in Twitter", *Procedia Computer Science*, 163, pp. 248-256. doi: 10.1016/j.procs.2019.12.106
22. *Part 3: Is using Twitter data ethical? – Data Impact blog* (2022). Available at: https://blog.ukdataservice.ac.uk/is-using-twitter-data-ethical/
23. https://nextshark.com/hibiki-sato-japanese-idol-obsessive-fan/

Part 7: Appendix

Figures and Tables

Table/Figure	Description
Fig 1	*Time Series showing volume of tweets across the month of June'22*
Fig 2	*Boxplot showcasing the difference between volume of weekday tweets vs weekend tweets*
Fig 3	*Quantile-quantile plot of data against a standard normal distribution*
Fig 4	*Resampled Histogram of difference between average weekend tweets and average weekday tweets*
Fig 5	*Time-series of tweets by hour, averaged over all days of the week*
Fig 6	*Time-series of tweet volume averaged at hours of a week.*
Fig 7	*Histogram representing number of tweets made by users*
Fig 8	*Log10 transforms of both axes – number of tweets, number of users shows a vaguely linear trend*
Fig 9	*Time-series patterns of tweet volume from high-volume tweeters*
Table 1	*Highest-volume tweeters*
Table 2	*Country having highest number of tweets from user is assigned as users' primary country*
Fig 10	*Cross-country mentions*
Table 3.1	*Tabulation User of Country to Mentioned Country*
Table 3.2	*Percentage Tabulation of User Country to Mentioned Country*
Fig 11	*Twitter activity overlaid on top of geographical boundaries.*
Fig 12.1	*CDF of bounding box diagonals represented on i) equal axis and ii) log-transformed y-axis*
Fig 12.2	*Two of the largest bounding boxes in the data*
Fig 13	*Areas of greenery generally showcase lower tweet volume.*
Fig 14	*Time-series data on tweet volume in the UK, France and the Netherlands across June'22.*
Fig 15	*Word cloud showing prominent words in tweets originating in UK on 24th June, 2022*
Fig 16	*Word cloud showing prominent words in tweets originating in France on 19th June, 2022*
Fig 17	*Word cloud showing prominent words in tweets originating in the Netherlands on 11th June, 2022*
Table 4.1	*Most common hashtags in UK, 24th June*
Table 4.2	*Most common hashtags in UK, 24th June not included*
Table 5.1	*Most common hashtags in France, 19th June*
Table 5.2	*Most common hashtags in France, 19th June not included*
Table 6.1	*Most common hashtags in the Netherlands, 11th June*
Table 6.2	*Most common hashtags in the Netherlands, 11th June not included*
Table 7	*Validation by news sources*