In [1]:
```
import json
import os
import pandas as pd
import zipfile

parent_directory = os.fsencode('S:\MSc Data Science with AI\IDS\Coursework\Twitte
file_count = 0
total_tweet = 0
tweet_dict = {"id": [], "text": [], "tweet_created_at": [], "timestamp_ms": []}

for folder in os.listdir(parent_directory):
    folder_name = os.fsdecode(folder)
    with zipfile.ZipFile(parent_directory.decode("utf-8") + '/' + folder_name) as
        for item in zip_archive.filelist:
            print(item.filename)
            json_file_data = zip_archive.read(item.filename)
            json_file_data_string = json_file_data.decode("utf-8")
            json_file_data_list = json_file_data_string.split("\n")

            for tweet in json_file_data_list:
                if tweet:
                    total_tweet += 1
                    json_tweet = json.loads(tweet)
                    if 'id_str' in json_tweet and 'created_at' and 'timestamp_ms'
                        tweet_dict['id'].append(json_tweet['id_str'])
                        tweet_dict['text'].append(json_tweet['text'])
                        tweet_dict['tweet_created_at'].append(json_tweet['created
                        tweet_dict['timestamp_ms'].append(json_tweet['timestamp_m

            # Capturing high level tweet statistics
            file_count += 1
```

```
geoEurope/geoEurope_2022060100.json
geoEurope/geoEurope_2022060101.json
geoEurope/geoEurope_2022060102.json
geoEurope/geoEurope_2022060103.json
geoEurope/geoEurope_2022060104.json
geoEurope/geoEurope_2022060105.json
geoEurope/geoEurope_2022060106.json
geoEurope/geoEurope_2022060107.json
geoEurope/geoEurope_2022060108.json
geoEurope/geoEurope_2022060109.json
geoEurope/geoEurope_2022060110.json
geoEurope/geoEurope_2022060111.json
geoEurope/geoEurope_2022060112.json
geoEurope/geoEurope_2022060113.json
geoEurope/geoEurope_2022060114.json
geoEurope/geoEurope_2022060115.json
geoEurope/geoEurope_2022060116.json
geoEurope/geoEurope_2022060117.json
geoEurope/geoEurope_2022060118.json
```

Part 1

Q1- Count the total number of tweets, describing how you deal with duplicates or other anomalies in the data set. [5 marks]

Total Number of tweets:- 15040709 Total number of tweets with a msg body: 15040386 Total number of unique tweets: 14627084

In [2]:
```
print("Total tweets- ",total_tweet)
print("Total Files - ",file_count)
```

```
Total tweets-  15040709
Total Files -  720
```

In [3]:
```
# len(tweet_dict['text'])
df = pd.DataFrame.from_dict(tweet_dict)
print(df.columns)
```

```
Index(['id', 'text', 'tweet_created_at', 'timestamp_ms'], dtype='object')
```

In [4]:
```
print(df['text'])
```

```
0                                        https://t.co/B3K8DCQpXg (https://t.co/B3K
8DCQpXg)
1                                              au weia! eens! bäm
2                    @nurse_hmsre Hayır akepe yi aya gönder mek
3              @gigi52335676 Ci riprenderemo le colonie e anc...
4              @rompelavabos No me consta, eso qué es? 😂😂😂 ht...
                                    ...
15040382       @Peri_Evrim Affferim sana bee 😂🏆 https://t.co/... (https://t.c
o/...)
15040383       SUNDAY Reset Vlog https://t.co/j2d2Ip44g1 (https://t.co/j2d2Ip44g1)
http...
15040384       @trockizm98 Otóż nie. Floh de cologne nie znam...
15040385       @clawdialopez Mi niña, enseguida juntisssss ❤️...
15040386         @KrzysztofNorw Lepszy live z meczu niz trybunka
Name: text, Length: 15040387, dtype: object
```

In [5]:
```
print(len(df.text.unique()))
```

```
14627084
```

Q2- Plot a time-series of the number of tweets by day using the whole dataset and comment on what you see. [5 marks]

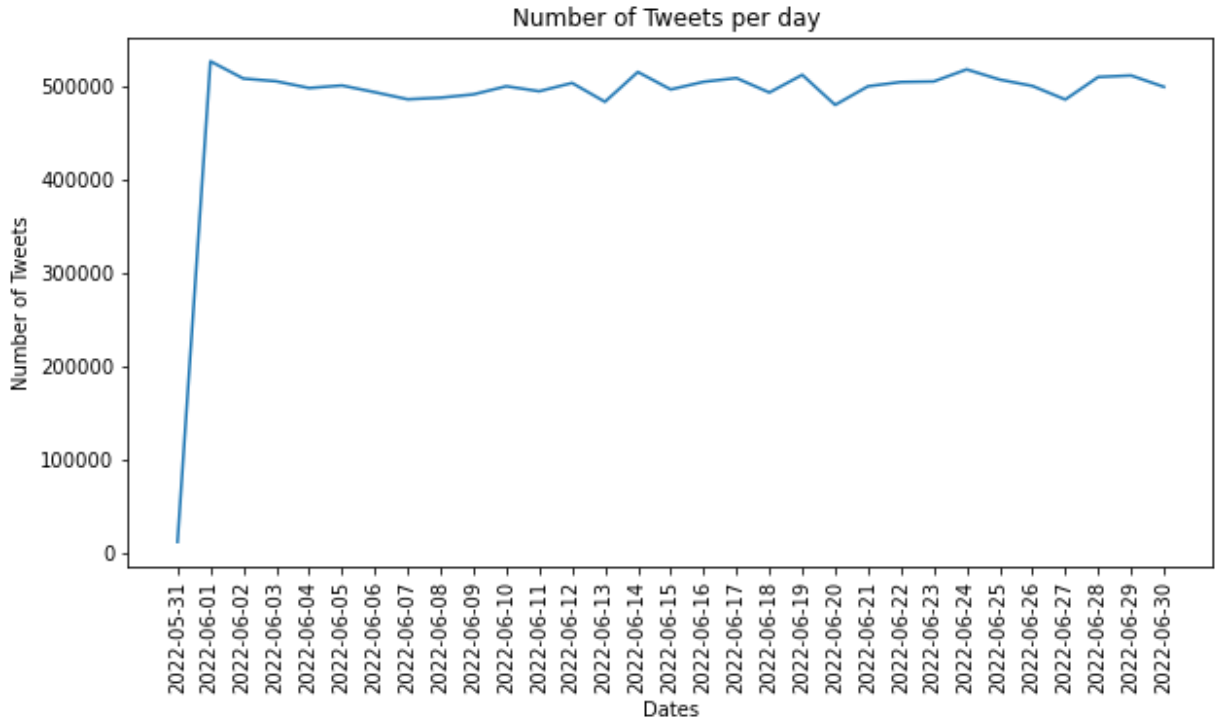
In [6]:
```
import matplotlib.pyplot as plt
```

In [7]:
```
def tweets_per_day(df):
    df['tweet_created_at'] = pd.to_datetime(df['tweet_created_at'], format='%a %b
    return df['text'].groupby(df['tweet_created_at'].dt.date).count()
  # return df['Tweets'].groupby(df['Date'].dt.date).count()
  # if you want output to be `Series`

tweets_per_day_temp = tweets_per_day(df)
```

In [8]:
```
print(tweets_per_day_temp,"\n")
print((tweets_per_day_temp[0]))
```

```
tweet_created_at
2022-05-31     12156
2022-06-01    526611
2022-06-02    508119
2022-06-03    505217
2022-06-04    497994
2022-06-05    500701
2022-06-06    493504
2022-06-07    485929
2022-06-08    487639
2022-06-09    491178
2022-06-10    499855
2022-06-11    494527
2022-06-12    503368
2022-06-13    483254
2022-06-14    515140
2022-06-15    496458
2022-06-16    504635
2022-06-17    508597
2022-06-18    493172
2022-06-19    512085
2022-06-20    479874
2022-06-21    499855
2022-06-22    504400
2022-06-23    504988
2022-06-24    517877
2022-06-25    507040
2022-06-26    500194
2022-06-27    485659
2022-06-28    509762
2022-06-29    511371
2022-06-30    499228
Name: text, dtype: int64

12156
```

In [9]:
```
# x axis - time
# y axis - no. of tweets
dates, no_of_tweets_daily = [], []
for a,b in tweets_per_day_temp.items():
    dates.append(a)
    no_of_tweets_daily.append(b)



figure = plt.figure(figsize=(10, 5))
plt.plot(dates, no_of_tweets_daily)
plt.xticks(dates, rotation ="vertical")
plt.xlabel("Dates")
plt.ylabel("Number of Tweets")
plt.title("Number of Tweets per day ")

# fig.savefig("test.png")
plt.show()
```

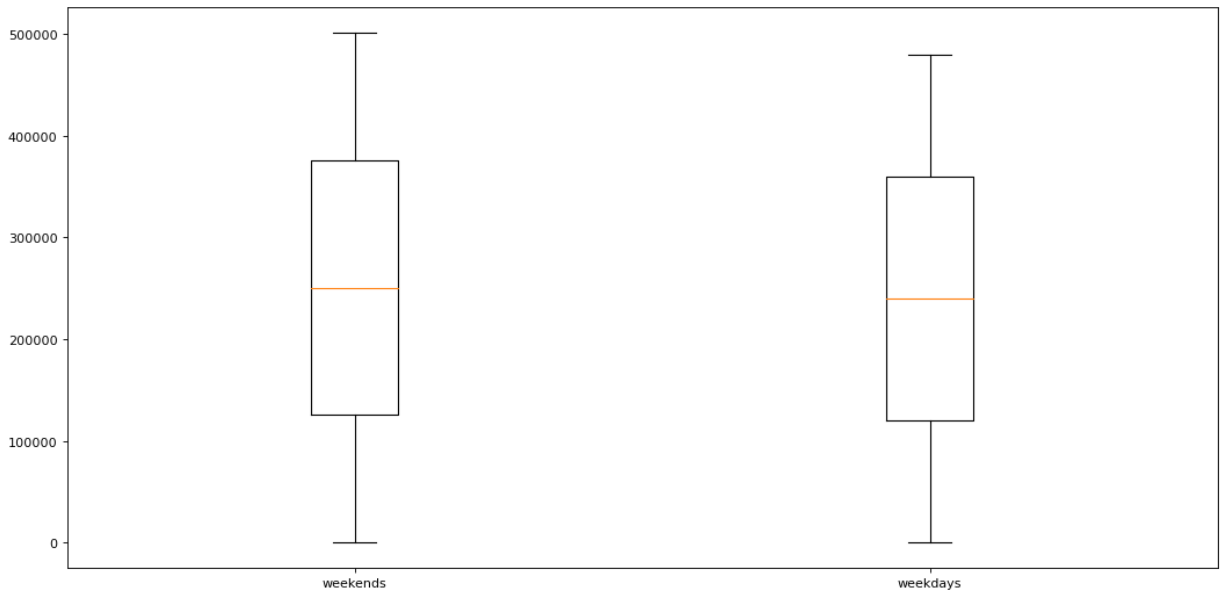


Q1.3- Use box and whisker diagrams to compare the average number of tweets on weekdays to the numbers for weekend days. Are there statistically significant differences between the number of tweets on weekdays and weekends? [5 marks]

In [10]:
```
Number_of_tweets = {'weekends': 0, 'weekdays': 0}
Days = []
count1 = 0

df['tweet_created_at'] = pd.to_datetime(df['tweet_created_at'])
df['tweet_created_at'] = df['tweet_created_at'].dt.date
for i in df['tweet_created_at']:
    if i.weekday() > 4:
        Number_of_tweets['weekends'] += 1  # Its a weekend
    else:
        Number_of_tweets['weekdays'] += 1  # Its a weekday

    if i in Days:
        continue
    else:
        Days.append(i)
```

In [11]:
```
No_of_week_days = 0
No_of_weekend_days = 0
Avg_values = []


for i in Days:
    if i.weekday() > 4:
        No_of_weekend_days += 1
    else:
        No_of_week_days += 1  # Its a weekday


print(No_of_weekend_days)
print(No_of_week_days)


labels, data = Number_of_tweets.keys(), Number_of_tweets.values()
count = 0
for val in data:
    if count == 0:
        Avg_values.append(val/No_of_weekend_days)  # total no of weekend tweets /
    else:
        Avg_values.append(val/No_of_week_days)    # total no of week day tweets /
    count = 1

print(No_of_week_days, No_of_weekend_days)

print(Avg_values)
values = [[No_of_weekend_days,Avg_values[0]], [No_of_week_days,Avg_values[1]]]
plt.figure(figsize=(16, 8), dpi=80)
plt.boxplot(values)
plt.xticks(range(1, len(labels) + 1), labels)
plt.show()
```

```
8
23
23 8
[501135.125, 479622.0]
```



In [12]:
```
print(Days)
#print(cnt)
#print(cnt_wkd)
print(Number_of_tweets['weekdays'])
print(Number_of_tweets['weekends'])
```

```
[datetime.date(2022, 5, 31), datetime.date(2022, 6, 1), datetime.date(2022, 6,
2), datetime.date(2022, 6, 3), datetime.date(2022, 6, 4), datetime.date(2022,
6, 5), datetime.date(2022, 6, 6), datetime.date(2022, 6, 7), datetime.date(202
2, 6, 8), datetime.date(2022, 6, 9), datetime.date(2022, 6, 10), datetime.date
(2022, 6, 11), datetime.date(2022, 6, 12), datetime.date(2022, 6, 13), datetim
e.date(2022, 6, 14), datetime.date(2022, 6, 15), datetime.date(2022, 6, 16), da
tetime.date(2022, 6, 17), datetime.date(2022, 6, 18), datetime.date(2022, 6, 1
9), datetime.date(2022, 6, 20), datetime.date(2022, 6, 21), datetime.date(2022,
6, 22), datetime.date(2022, 6, 23), datetime.date(2022, 6, 24), datetime.date(2
022, 6, 25), datetime.date(2022, 6, 26), datetime.date(2022, 6, 27), datetime.d
ate(2022, 6, 28), datetime.date(2022, 6, 29), datetime.date(2022, 6, 30)]
11031306
4009081
```

Q1.4- Plot a time-series of the number of tweets by hour, averaged over all weekdays and comment on what you see. [5 marks]

In [ ]:

In [ ]:

Part-4

In [13]:
```
import json
import os
import pandas as pd
import zipfile
```

In [14]:
```
parent_directory = os.fsencode('TwitterJune2022')
file_count = 0
total_tweet = 0
tweet_dict = {"text": [], "country": [], "tweet_created_at": []}

for folder in os.listdir(parent_directory):
    folder_name = os.fsdecode(folder)
    with zipfile.ZipFile(parent_directory.decode("utf-8") + '/' + folder_name) as
        for item in zip_archive.filelist:
            print(item.filename)
            json_file_data = zip_archive.read(item.filename)
            json_file_data_string = json_file_data.decode("utf-8")
            json_file_data_list = json_file_data_string.split("\n")

            for tweet in json_file_data_list:
                if tweet:
                    total_tweet += 1
                    json_tweet = json.loads(tweet)
                    if 'id_str' in json_tweet and 'created_at' and 'timestamp_ms'
                        try:
                            #tweet_dict['id'].append(json_tweet['id_str'])
                            tweet_dict['text'].append(json_tweet['text'])
                            tweet_dict['country'].append(json_tweet['place']['cou
                            tweet_dict['tweet_created_at'].append(json_tweet['cre
                        except TypeError:
                            pass

            # Capturing high level tweet statistics
            file_count += 1
```

```
geoEurope/geoEurope_2022060100.json
geoEurope/geoEurope_2022060101.json
geoEurope/geoEurope_2022060102.json
geoEurope/geoEurope_2022060103.json
geoEurope/geoEurope_2022060104.json
geoEurope/geoEurope_2022060105.json
geoEurope/geoEurope_2022060106.json
geoEurope/geoEurope_2022060107.json
geoEurope/geoEurope_2022060108.json
geoEurope/geoEurope_2022060109.json
geoEurope/geoEurope_2022060110.json
geoEurope/geoEurope_2022060111.json
geoEurope/geoEurope_2022060112.json
geoEurope/geoEurope_2022060113.json
geoEurope/geoEurope_2022060114.json
geoEurope/geoEurope_2022060115.json
geoEurope/geoEurope_2022060116.json
geoEurope/geoEurope_2022060117.json
geoEurope/geoEurope_2022060118.json
```

In [15]:
```
c_df = pd.DataFrame.from_dict(tweet_dict, orient='index')
c_df
```

Out[15]:

	0	1	2	3	4
text	https://t.co/B3K8DCQpXg	au weia! eens! bäm	@nurse_hmsre Hayır akepe yi aya gönder mek	@gigi52335676 Ci riprenderemo le colonie e anc...	@rompelavabos No me consta, eso qué es? 😂😂😂 ht...
country	Royaume-Uni	Türkiye	Italia	España	Italia
tweet_created_at	Tue May 31 23:00:00 +0000 2022	Tue May 31 23:00:00 +0000 2022	Tue May 31 23:00:00 +0000 2022	Tue May 31 23:00:01 +0000 2022	Tue May 31 23:00:01 +0000 2022

3 rows × 15040387 columns

In [16]:
```
c = pd.DataFrame(c_df)
c_data = c.T
c_drop_data = c_data.dropna(how='any')
```

In [17]:
```
c_drop_data
```

Out[17]:

	text	country	tweet_created_at
0	https://t.co/B3K8DCQpXg	Royaume-Uni	Tue May 31 23:00:00 +0000 2022
1	au weia! eens! bäm	Türkiye	Tue May 31 23:00:00 +0000 2022
2	@nurse_hmsre Hayır akepe yi aya gönder mek	Italia	Tue May 31 23:00:00 +0000 2022
3	@gigi52335676 Ci riprenderemo le colonie e anc...	España	Tue May 31 23:00:01 +0000 2022
4	@rompelavabos No me consta, eso qué es? 😂😂😂 ht...	Italia	Tue May 31 23:00:01 +0000 2022
...	...	...	...
15033599	Kimse suskunluğumu asaletimden sanmasın, ite k...	Türkiye	Thu Jun 30 22:59:59 +0000 2022
15033600	@DrGozen +1	Ireland	Thu Jun 30 22:59:59 +0000 2022
15033601	Finally getting around to watching Miranda @me...	Poland	Thu Jun 30 22:59:59 +0000 2022
15033602	@az_tb_77 و عافیت بشید با ساله ۱۲۰ میگم تبریک ...	España	Thu Jun 30 22:59:59 +0000 2022
15033603	@ugciaaxsbh3ARKo @MAQBOOL85432875 https://t.co...	Polska	Thu Jun 30 22:59:59 +0000 2022

15033604 rows × 3 columns

Part 4.1

In [18]:
```
UK_cdf = pd.DataFrame(c_drop_data, columns=['text' , 'country' , 'tweet_created_a
UK_data = UK_cdf.loc[UK_cdf['country'] == 'United Kingdom']
UK_data
```

Out[18]:

	text	country	tweet_created_at
5	01:00\nTemp. 15,0°C App. 15,9°C\nUmid. 95% \nP...	United Kingdom	Tue May 31 23:00:01 +0000 2022
9	"Open arms of the sea"\n#NFTCommunity #NFTdrop...	United Kingdom	Tue May 31 23:00:01 +0000 2022
15	I started 26 days 10 hours and 5 minutes ago. ...	United Kingdom	Tue May 31 23:00:02 +0000 2022
16	LSZH 312250Z AUTO 22006KT 200V260 9999 NSC 15/...	United Kingdom	Tue May 31 23:00:02 +0000 2022
21	DOING (1:00 uur)	United Kingdom	Tue May 31 23:00:03 +0000 2022
...	...	...	...
15033579	@kobou_ J'sais pas y'a une meuf elle me dit j'...	United Kingdom	Thu Jun 30 22:59:53 +0000 2022
15033581	@guidotolomei Grazie per la spiegazione, notiz...	United Kingdom	Thu Jun 30 22:59:54 +0000 2022
15033586	@Happydog___ I WOULD SO ADOPT THIS CUTE SWEET PUP	United Kingdom	Thu Jun 30 22:59:56 +0000 2022
15033587	@JMJM0_ 🙄	United Kingdom	Thu Jun 30 22:59:56 +0000 2022
15033596	@_Bariskdlr53 Amin 👏👏👏🧿	United Kingdom	Thu Jun 30 22:59:58 +0000 2022

3465192 rows × 3 columns

In [19]:
```
def tweets_per_day(df):

    df['tweet_created_at'] = pd.to_datetime(df['tweet_created_at'], format='%a %b

    a = df[['text']].groupby(df['tweet_created_at'].dt.date).count()

    #b = df[['text']].count()

    return a #,b

tweets_per_day(UK_data)
```

```
C:\Users\alama\AppData\Local\Temp/ipykernel_8284/2242918466.py:3: SettingWithCo
pyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/sta
ble/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pyd
ata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
opy)
  df['tweet_created_at'] = pd.to_datetime(df['tweet_created_at'], format='%a %b
%d %H:%M:%S +0000 %Y')
```

Out[19]:

tweet_created_at	text
2022-05-31	2608
2022-06-01	124478
2022-06-02	120872
2022-06-03	112177
2022-06-04	118246
2022-06-05	115927
2022-06-06	121339
2022-06-07	110156
2022-06-08	109072
2022-06-09	110357
2022-06-10	113415
2022-06-11	115663
2022-06-12	110659
2022-06-13	105194
2022-06-14	124313
2022-06-15	113348
2022-06-16	117714
2022-06-17	121227
2022-06-18	113816

tweet_created_at	text
2022-06-19	113115
2022-06-20	101609
2022-06-21	114588
2022-06-22	114540
2022-06-23	118437
2022-06-24	131937
2022-06-25	122537
2022-06-26	116126
2022-06-27	105377
2022-06-28	115029
2022-06-29	115326
2022-06-30	115990

In [20]:
```
ukdataframe = pd.DataFrame(tweets_per_day(UK_data))
ukdataframe
```

```
C:\Users\alama\AppData\Local\Temp/ipykernel_8284/2242918466.py:3: SettingWithCo
pyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/sta
ble/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pyd
ata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
opy)
  df['tweet_created_at'] = pd.to_datetime(df['tweet_created_at'], format='%a %b
%d %H:%M:%S +0000 %Y')
```

Out[20]:

tweet_created_at	text
2022-05-31	2608
2022-06-01	124478
2022-06-02	120872
2022-06-03	112177
2022-06-04	118246
2022-06-05	115927
2022-06-06	121339
2022-06-07	110156
2022-06-08	109072
2022-06-09	110357
2022-06-10	113415
2022-06-11	115663
2022-06-12	110659
2022-06-13	105194
2022-06-14	124313
2022-06-15	113348
2022-06-16	117714
2022-06-17	121227
2022-06-18	113816
2022-06-19	113115
2022-06-20	101609
2022-06-21	114588
2022-06-22	114540
2022-06-23	118437
2022-06-24	131937

tweet_created_at	text
2022-06-25	122537
2022-06-26	116126
2022-06-27	105377
2022-06-28	115029
2022-06-29	115326
2022-06-30	115990

In [21]:
```
max_uk = ukdataframe['text'].max()
max_uk
```

Out[21]: 131937

In [22]:
```
ukdataframe['text'].idxmax(skipna = True)
```

Out[22]: datetime.date(2022, 6, 24)

In [23]:
```
#France
```

In [24]:
```
France_cdf = pd.DataFrame(c_drop_data, columns=['text' , 'country' , 'tweet_creat
France_data = France_cdf.loc[France_cdf['country'] == 'France']
France_data
```

Out[24]:

	text	country	tweet_created_at
12	Бам Бам\nБайрактар!	France	Tue May 31 23:00:02 +0000 2022
49	https://t.co/5fOnpul5Cc	France	Tue May 31 23:00:07 +0000 2022
97	@mrbenjitaylor @thisisbask Looks epic Ben cong...	France	Tue May 31 23:00:18 +0000 2022
116	@ilPellicano_ @gianpi36590925 Sveglissima. Qui...	France	Tue May 31 23:00:21 +0000 2022
127	00:49 Temp. 19.4°C, Hum. 80%, Dewp. 15°C, Bar....	France	Tue May 31 23:00:25 +0000 2022
...	...	...	...
15033506	Sıradaki şarkı maymunlara gelsin	France	Thu Jun 30 22:59:33 +0000 2022
15033534	July 1… Day 1! 💞\n#angatbuhay \n#csmyway @ Bur...	France	Thu Jun 30 22:59:42 +0000 2022
15033538	🤝طولت بالي كل مالها و تـ ـقـ صـ ر	France	Thu Jun 30 22:59:43 +0000 2022
15033584	آیا برایت من شد؟	France	Thu Jun 30 22:59:55 +0000 2022
15033591	i do not have the restraint necessary. but ple...	France	Thu Jun 30 22:59:57 +0000 2022

1038675 rows × 3 columns

In [25]:
```
def tweets_per_day(df):

    df['tweet_created_at'] = pd.to_datetime(df['tweet_created_at'], format='%a %b

    a = df[['text']].groupby(df['tweet_created_at'].dt.date).count()

    #b = df[['text']].count()

    return a #,b

tweets_per_day(France_data)
```

```
C:\Users\alama\AppData\Local\Temp/ipykernel_8284/582256297.py:3: SettingWithCop
yWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/sta
ble/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pyd
ata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
opy)
  df['tweet_created_at'] = pd.to_datetime(df['tweet_created_at'], format='%a %b
%d %H:%M:%S +0000 %Y')
```

Out[25]:

tweet_created_at	text
2022-05-31	936
2022-06-01	35507
2022-06-02	33466
2022-06-03	35330
2022-06-04	32649
2022-06-05	34351
2022-06-06	34247
2022-06-07	32464
2022-06-08	33702
2022-06-09	33981
2022-06-10	34923
2022-06-11	29279
2022-06-12	36456
2022-06-13	34429
2022-06-14	35953
2022-06-15	36269
2022-06-16	36785
2022-06-17	37065
2022-06-18	33839

tweet_created_at	text
2022-06-19	39714
2022-06-20	35081
2022-06-21	35743
2022-06-22	34320
2022-06-23	34564
2022-06-24	35490
2022-06-25	33749
2022-06-26	32650
2022-06-27	32420
2022-06-28	33983
2022-06-29	34373
2022-06-30	34957

In [26]:
```
France_df = pd.DataFrame(tweets_per_day(France_data))
France_df
```

```
C:\Users\alama\AppData\Local\Temp/ipykernel_8284/582256297.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  df['tweet_created_at'] = pd.to_datetime(df['tweet_created_at'], format='%a %b %d %H:%M:%S +0000 %Y')
```

Out[26]:

tweet_created_at	text
2022-05-31	936
2022-06-01	35507
2022-06-02	33466
2022-06-03	35330
2022-06-04	32649
2022-06-05	34351
2022-06-06	34247
2022-06-07	32464
2022-06-08	33702
2022-06-09	33981
2022-06-10	34923
2022-06-11	29279
2022-06-12	36456
2022-06-13	34429
2022-06-14	35953
2022-06-15	36269
2022-06-16	36785
2022-06-17	37065
2022-06-18	33839
2022-06-19	39714
2022-06-20	35081
2022-06-21	35743
2022-06-22	34320
2022-06-23	34564
2022-06-24	35490

tweet_created_at	text
2022-06-25	33749
2022-06-26	32650
2022-06-27	32420
2022-06-28	33983
2022-06-29	34373
2022-06-30	34957

In [27]:
```
max_france = France_df['text'].max()
max_france
```

Out[27]: 39714

In [28]:
```
France_df['text'].idxmax(skipna = True)
```

Out[28]: datetime.date(2022, 6, 19)

In [29]:
```
#Turkey
```

In [30]:
```
Turkey_cdf = pd.DataFrame(c_drop_data, columns=['text' , 'country' , 'tweet_creat
Turkey_data = Turkey_cdf.loc[Turkey_cdf['country'] == 'Turkey']
Turkey_data
```

Out[30]:

	text	country	tweet_created_at
35	banyoda soğuk suyun altında uyusam abartmıs ol...	Turkey	Tue May 31 23:00:06 +0000 2022
39	Mayıs bitti :((((	Turkey	Tue May 31 23:00:07 +0000 2022
59	Excessive queuing for airports, queuing for sc...	Turkey	Tue May 31 23:00:10 +0000 2022
78	@RobertaFavalor2 Ma che meraviglia questo bell...	Turkey	Tue May 31 23:00:14 +0000 2022
168	#BGT this is really becoming a male-dominated ...	Turkey	Tue May 31 23:00:38 +0000 2022
...	...	...	...
15033477	@NOS ... niet met "PLEK" maar met "PLAATS"...	Turkey	Thu Jun 30 22:59:26 +0000 2022
15033500	@Emilia81439113 Parece que los puntuales somos...	Turkey	Thu Jun 30 22:59:32 +0000 2022
15033539	@nw_nicholas You're going to need to brush up ...	Turkey	Thu Jun 30 22:59:44 +0000 2022
15033566	@IsabelA62887947 @budino_antonio Buenas noches...	Turkey	Thu Jun 30 22:59:50 +0000 2022
15033574	After the shop employees called the police and...	Turkey	Thu Jun 30 22:59:53 +0000 2022

365194 rows × 3 columns

In [31]:
```
def tweets_per_day(df):

    df['tweet_created_at'] = pd.to_datetime(df['tweet_created_at'], format='%a %b

    a = df[['text']].groupby(df['tweet_created_at'].dt.date).count()

    #b = df[['text']].count()

    return a #,b

tweets_per_day(Turkey_data)
```

```
C:\Users\alama\AppData\Local\Temp/ipykernel_8284/2592286055.py:3: SettingWithCo
pyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/sta
ble/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pyd
ata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
opy)
  df['tweet_created_at'] = pd.to_datetime(df['tweet_created_at'], format='%a %b
%d %H:%M:%S +0000 %Y')
```

Out[31]:

tweet_created_at	text
2022-05-31	268
2022-06-01	12604
2022-06-02	11703
2022-06-03	11910
2022-06-04	11415
2022-06-05	11319
2022-06-06	11431
2022-06-07	11874
2022-06-08	11387
2022-06-09	11880
2022-06-10	12159
2022-06-11	12306
2022-06-12	12500
2022-06-13	12070
2022-06-14	11906
2022-06-15	11530
2022-06-16	11190
2022-06-17	11492
2022-06-18	11644

tweet_created_at	text
2022-06-19	11477
2022-06-20	11628
2022-06-21	12826
2022-06-22	12994
2022-06-23	12601
2022-06-24	12552
2022-06-25	12391
2022-06-26	12528
2022-06-27	13326
2022-06-28	13560
2022-06-29	13796
2022-06-30	12927

In [32]:
```
Turkey_df = pd.DataFrame(tweets_per_day(Turkey_data))
Turkey_df
```

```
C:\Users\alama\AppData\Local\Temp/ipykernel_8284/2592286055.py:3: SettingWithCo
pyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/sta
ble/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pyd
ata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
opy)
  df['tweet_created_at'] = pd.to_datetime(df['tweet_created_at'], format='%a %b
%d %H:%M:%S +0000 %Y')
```

Out[32]:

tweet_created_at	text
2022-05-31	268
2022-06-01	12604
2022-06-02	11703
2022-06-03	11910
2022-06-04	11415
2022-06-05	11319
2022-06-06	11431
2022-06-07	11874
2022-06-08	11387
2022-06-09	11880
2022-06-10	12159
2022-06-11	12306
2022-06-12	12500
2022-06-13	12070
2022-06-14	11906
2022-06-15	11530
2022-06-16	11190
2022-06-17	11492
2022-06-18	11644
2022-06-19	11477
2022-06-20	11628
2022-06-21	12826
2022-06-22	12994
2022-06-23	12601
2022-06-24	12552

tweet_created_at	text
2022-06-25	12391
2022-06-26	12528
2022-06-27	13326
2022-06-28	13560
2022-06-29	13796
2022-06-30	12927

In [33]:
```
max_turkey = Turkey_df['text'].max()
max_turkey
```

Out[33]: 13796

In [34]:
```
Turkey_df['text'].idxmax(skipna = True)
```

Out[34]: datetime.date(2022, 6, 29)

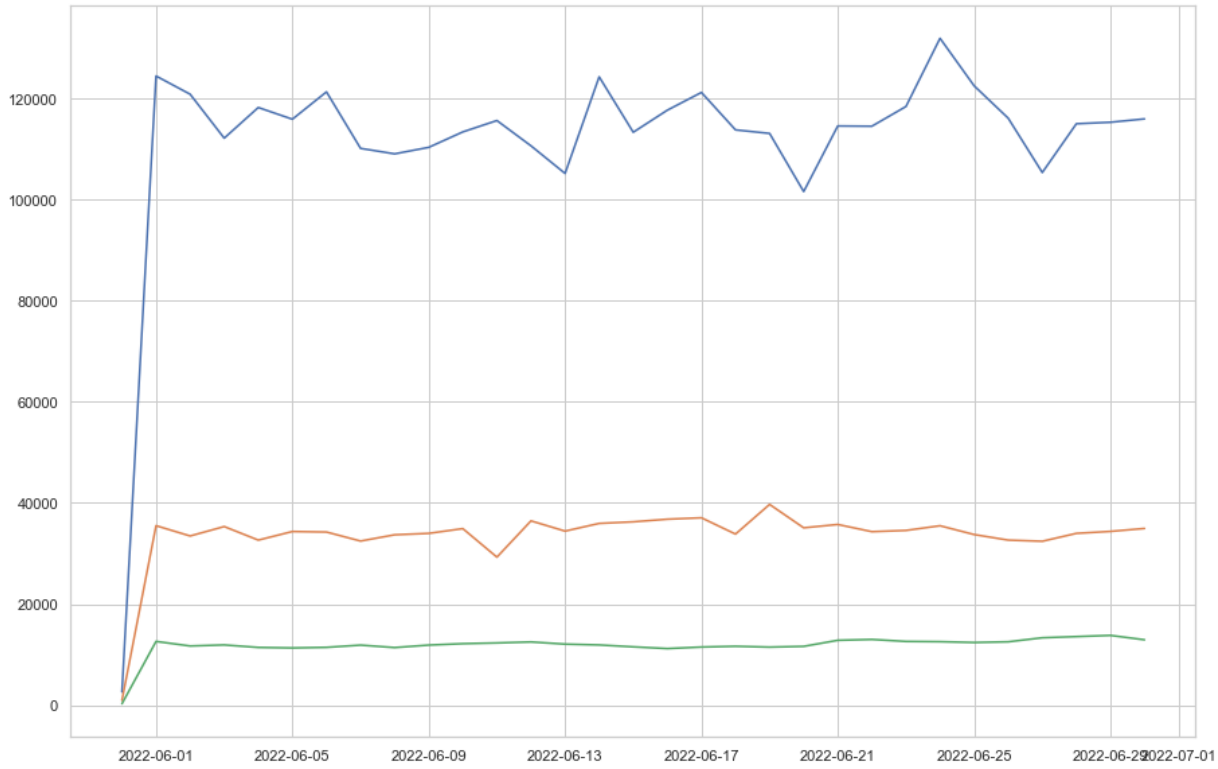
In [35]:
```
#Plot
import matplotlib.pyplot as plt
```

In [36]:
```
u=ukdataframe.copy()

f=France_df.copy()

t=Turkey_df.copy()

plotdata = u.merge(f,on='tweet_created_at').merge(t,on='tweet_created_at')

print(plotdata)
```

tweet_created_at	text_x	text_y	text
2022-05-31	2608	936	268
2022-06-01	124478	35507	12604
2022-06-02	120872	33466	11703
2022-06-03	112177	35330	11910
2022-06-04	118246	32649	11415
2022-06-05	115927	34351	11319
2022-06-06	121339	34247	11431
2022-06-07	110156	32464	11874
2022-06-08	109072	33702	11387
2022-06-09	110357	33981	11880
2022-06-10	113415	34923	12159
2022-06-11	115663	29279	12306
2022-06-12	110659	36456	12500
2022-06-13	105194	34429	12070
2022-06-14	124313	35953	11906
2022-06-15	113348	36269	11530
2022-06-16	117714	36785	11190
2022-06-17	121227	37065	11492
2022-06-18	113816	33839	11644
2022-06-19	113115	39714	11477
2022-06-20	101609	35081	11628
2022-06-21	114588	35743	12826
2022-06-22	114540	34320	12994
2022-06-23	118437	34564	12601
2022-06-24	131937	35490	12552
2022-06-25	122537	33749	12391
2022-06-26	116126	32650	12528
2022-06-27	105377	32420	13326
2022-06-28	115029	33983	13560
2022-06-29	115326	34373	13796
2022-06-30	115990	34957	12927

In [37]:
```
import seaborn as sns

import numpy as np

sns.set_theme(style="whitegrid")



figure = plt.figure(figsize=(15, 10))

plt.plot(plotdata)
```

Out[37]:
```
[<matplotlib.lines.Line2D at 0x2109756c130>,
 <matplotlib.lines.Line2D at 0x210d9a9eaf0>,
 <matplotlib.lines.Line2D at 0x210d9aad940>]
```



Part 4.2

In [38]:
```
import re
import string
import nltk
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
import nltk as nlp
import nltk.corpus
from nltk.corpus import stopwords
from nltk import word_tokenize
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
from collections import Counter
from nltk.util import everygrams
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\alama\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\alama\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\alama\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

In [42]:
```
### Data Preprocessing/cleaning
```

In [43]:
```
cleaned_tweet_list = []
for cleaned_tweet in c_data.text:
    cleaned_tweet = cleaned_tweet.lower() # convert tweet to lowercase
    cleaned_tweet = re.sub(r'@[A-Za-z0-9-_]+', '', cleaned_tweet) # Removed @ment
    cleaned_tweet = re.sub(r"'[A-Za-z0-9-_]+", '', cleaned_tweet) # Removed 's
    cleaned_tweet = re.sub(r'https?:\/\/\S+', '', cleaned_tweet) # Remove the hyp
    cleaned_tweet = re.sub(r'(\s*)amp(\s*)+', '', cleaned_tweet) # Remove amp (be
    cleaned_tweet = ''.join(word for word in cleaned_tweet if not word.isdigit())
    cleaned_tweet = nltk.word_tokenize(cleaned_tweet) # Tokenization
    table = str.maketrans('', '', string.punctuation)
    stripped = [word.translate(table) for word in cleaned_tweet] # Remove Punctuc
    cleaned_tweet = [word for word in stripped if word.isalpha()] # Remove remair
    cleaned_tweet = [word for word in cleaned_tweet if not word in
                     set(stopwords.words("english"))] # Remove stopwords
    lemma = nlp.WordNetLemmatizer()
    cleaned_tweet = [lemma.lemmatize(word) for word in cleaned_tweet]
    cleaned_tweet = " ".join(word for word in cleaned_tweet if len(word) > 1) # L
    cleaned_tweet_list.append(cleaned_tweet)
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_8284/2396430394.py in <module>
     11     stripped = [word.translate(table) for word in cleaned_tweet] # Remo
ve Punctuation
     12     cleaned_tweet = [word for word in stripped if word.isalpha()] # Rem
ove remaining tokens that are not alphabetic (emoji and non-english words)
---> 13     cleaned_tweet = [word for word in cleaned_tweet if not word in
     14                     set(stopwords.words("english"))] # Remove stopwords
     15     lemma = nlp.WordNetLemmatizer()

~\AppData\Local\Temp/ipykernel_8284/2396430394.py in <listcomp>(.0)
     12     cleaned_tweet = [word for word in stripped if word.isalpha()] # Rem
ove remaining tokens that are not alphabetic (emoji and non-english words)
     13     cleaned_tweet = [word for word in cleaned_tweet if not word in
---> 14                     set(stopwords.words("english"))] # Remove stopwords
     15     lemma = nlp.WordNetLemmatizer()
     16     cleaned_tweet = [lemma.lemmatize(word) for word in cleaned_tweet]

~\anaconda3\lib\site-packages\nltk\corpus\reader\wordlist.py in words(self, fil
eids, ignore_lines_startswith)
     19         return [
     20             line
---> 21             for line in line_tokenize(self.raw(fileids))
     22             if not line.startswith(ignore_lines_startswith)
     23         ]

~\anaconda3\lib\site-packages\nltk\corpus\reader\api.py in raw(self, fileids)
    216         for f in fileids:
    217             with self.open(f) as fp:
--> 218                 contents.append(fp.read())
    219         return concat(contents)
    220

~\anaconda3\lib\site-packages\nltk\data.py in __exit__(self, type, value, trace
back)
   1165
```

```
   1166     def __exit__(self, type, value, traceback):
-> 1167         self.close()
   1168
   1169     def xreadlines(self):

~\anaconda3\lib\site-packages\nltk\data.py in close(self)
   1194         Close the underlying stream.
   1195         """
-> 1196         self.stream.close()
   1197
   1198     # //////////////////////////////////////////////////////////

KeyboardInterrupt:
```

In [ ]:
```
c_data["cleaned_tweet"] = cleaned_tweet_list
c_data
```

In [ ]:
```
c_data.dropna(how='any')
```

In [ ]:
```
c_data.to_csv('cleaned_tweets.csv')
```

In [ ]:
```
# Removing whitespaces by splitting cleaned_tweet for counter
def split_name(cleaned_tweet):
    split = str(cleaned_tweet).split()
    return split

# Store the data in a list for text visualization
tweets_count_list = []
for x in cleaned_tweet_list:
    for y in split_name(x):
        tweets_count_list.append(y)
```

In [ ]:
```
!pip install wordcloud
```

In [ ]:
```
import seaborn as sns
from wordcloud import WordCloud
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
```

In [ ]:
```

```

In [ ]:
```

```