In [2]:
```
import pandas as pd
df = pd.read_csv('data_visualization.csv')
```

```
C:\Users\prati\AppData\Local\Temp\ipykernel_2444\1333053867.py:2: DtypeWarning: Columns (22,24) have mixed types. Specify dtype
option on import or set low_memory=False.
  df = pd.read_csv('data_visualization.csv')
```

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 33590 entries, 0 to 33589
Data columns (total 36 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   id               33590 non-null  int64
 1   conversation_id  33590 non-null  int64
 2   created_at       33590 non-null  object
 3   date             33590 non-null  object
 4   time             33590 non-null  object
 5   timezone         33590 non-null  int64
 6   user_id          33590 non-null  int64
 7   username         33590 non-null  object
 8   name             33590 non-null  object
 9   place            85 non-null     object
 10  tweet            33590 non-null  object
 11  language         33590 non-null  object
 12  mentions         33590 non-null  object
 13  urls             33590 non-null  object
 14  photos           33590 non-null  object
 15  replies_count    33590 non-null  int64
 16  retweets_count   33590 non-null  int64
 17  likes_count      33590 non-null  int64
 18  hashtags         33590 non-null  object
 19  cashtags         33590 non-null  object
 20  link             33590 non-null  object
 21  retweet          33590 non-null  bool
 22  quote_url        1241 non-null   object
 23  video            33590 non-null  int64
 24  thumbnail        9473 non-null   object
 25  near             0 non-null      float64
 26  geo              0 non-null      float64
 27  source           0 non-null      float64
 28  user_rt_id       0 non-null      float64
 29  user_rt          0 non-null      float64
 30  retweet_id       0 non-null      float64
 31  reply_to         33590 non-null  object
 32  retweet_date     0 non-null      float64
 33  translate        0 non-null      float64
 34  trans_src        0 non-null      float64
 35  trans_dest       0 non-null      float64
dtypes: bool(1), float64(10), int64(8), object(17)
memory usage: 9.0+ MB
```

In [4]:
```
df['tweet'][10]
```

Out[4]:
```
'We are pleased to invite you to the EDHEC DataViz Challenge grand final for a virtual exchange with all Top 10 finalists to see how data visualization creates impact and can bring out compelling stories in support of @UNICEF’s mission.  https://t.co/Vbj9B48VjV'
```

In [5]:
```
import nltk
nltk.download('vader_lexicon')
from nltk.sentiment.vader import SentimentIntensityAnalyzer
sid = SentimentIntensityAnalyzer()
import re
import pandas as pd
import nltk
nltk.download('words')
words = set(nltk.corpus.words.words())
```

```
[nltk_data] Downloading package vader_lexicon to
[nltk_data]     C:\Users\prati\AppData\Roaming\nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!
[nltk_data] Downloading package words to
[nltk_data]     C:\Users\prati\AppData\Roaming\nltk_data...
[nltk_data]   Package words is already up-to-date!
```

In [12]:
```
sentence = df['tweet'][0]
sid.polarity_scores(sentence)['compound']
```

Out[12]:
```
0.7089
```

In [14]:
```
import re
import nltk
nltk.download('punkt')
words = set(nltk.corpus.words.words())

def cleaner(tweet):
    tweet = re.sub("@[A-Za-z0-9]+", "", tweet)
    tweet = re.sub(r"(?:\@|http?\://|https?\://|www)\S+", "", tweet)
    tweet = " ".join(tweet.split())
    tweet = tweet.replace("#", "").replace("_", " ")
```

```
    tweet = " ".join(w for w in nltk.word_tokenize(tweet) if w.lower() in words or not w.isalpha())
    return tweet

df['tweet_clean'] = df['tweet'].apply(cleaner)
```

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

In [15]:
```
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer

nltk.download('vader_lexicon')

word_dict = {
    'manipulate': -1,
    'manipulative': -1,
    'jamescharlesiscancelled': -1,
    'jamescharlesisoverparty': -1,
    'pedophile': -1,
    'pedo': -1,
    'cancel': -1,
    'cancelled': -1,
    'cancelculture': 0.4,
    'teamtati': -1,
    'teamjames': 1,
    'teamjamescharles': 1,
    'liar': -1
}

sid = SentimentIntensityAnalyzer()

sid.lexicon.update(word_dict)

list1 = []
for tweet_clean in df['tweet_clean']:
    scores = sid.polarity_scores(str(tweet_clean))
    list1.append(scores['compound'])
```

```
[nltk_data] Downloading package vader_lexicon to
[nltk_data]     C:\Users\prati\AppData\Roaming\nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!
```

In [16]:
```
import pandas as pd

df['sentiment'] = pd.Series(list1)

def sentiment_category(sentiment):
    label = ''
    if sentiment > 0:
        label = 'positive'
    elif sentiment == 0:
        label = 'neutral'
    else:
        label = 'negative'
    return label

df['sentiment_category'] = df['sentiment'].apply(sentiment_category)
```

In [17]:
```
df = df[['tweet', 'date', 'id', 'sentiment', 'sentiment_category']]
df.head()
```

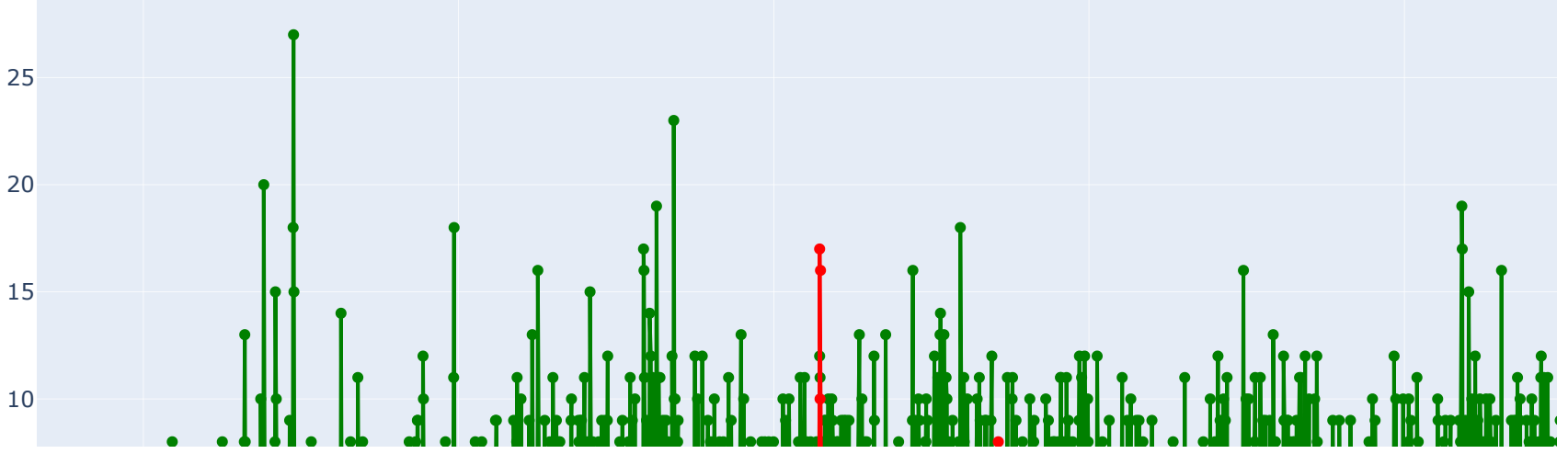
Out[17]:

	tweet	date	id	sentiment	sentiment_category
0	Take your storytelling to the next level using...	2021-06-20	1406335989484822531	0.7089	positive
1	Choosing Fonts for Your Data Visualization | b...	2021-06-19	1406292636789526537	0.0000	neutral
2	This data visualization shows where our greate...	2021-06-19	1406082288035811330	0.0000	neutral
3	Looking for examples of stellar charts made so...	2021-06-18	1405948260796100610	0.4019	positive
4	With #WISQARS Data Visualization, you can disp...	2021-06-18	1405942146960613376	-0.4215	negative

In [18]:
```
pos = df[df['sentiment_category'] == 'positive'].groupby('date', as_index=False)['id'].count()

neg = df[df['sentiment_category'] == 'negative'].groupby('date', as_index=False)['id'].count()
```

In [19]:
```
import plotly.graph_objs as go

fig = go.Figure()

fig.add_trace(go.Scatter(x=pos['date'], y=pos['id'],
                         name='Positive Sentiment',
                         mode='markers+lines',
                         line=dict(shape='linear'),
                         connectgaps=True,
                         line_color='green'
                         ))

fig.add_trace(go.Scatter(x=neg['date'], y=neg['id'],
                         name='Negative Sentiment',
```

```
                         mode='markers+lines',
                         line=dict(shape='linear'),
                         connectgaps=True,
                         line_color='red'
                         ))

fig.show()
```



In [ ]: