In [29]:

```
import pandas as pd
import numpy as np
import sklearn
import re
import seaborn as sns
!pip install wordcloud
from wordcloud import WordCloud,STOPWORDS
import nltk
nltk.download('abc')
import plotly.express as px
from nltk.corpus import abc
nltk.download('stopwords')
nltk.download('wordnet')
lst_stopwords = nltk.corpus.stopwords.words("english")
lst_stopwords[1:5]
!pip install textblob
!pip install -U kaleido
from textblob import TextBlob
import plotly.graph_objs as go
import matplotlib.pyplot as plt
import nltk
import collections
nltk.downloader.download('vader_lexicon')
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import warnings
warnings.filterwarnings("ignore")
```

```
Requirement already satisfied: wordcloud in /Users/nidhisoley/opt/anaconda3/lib/python3.8/site-packages (1.8.2.2)
Requirement already satisfied: pillow in /Users/nidhisoley/opt/anaconda3/lib/python3.8/site-packages (from wordclo
ud) (8.2.0)
Requirement already satisfied: matplotlib in /Users/nidhisoley/opt/anaconda3/lib/python3.8/site-packages (from wor
dcloud) (3.3.4)
Requirement already satisfied: numpy>=1.6.1 in /Users/nidhisoley/opt/anaconda3/lib/python3.8/site-packages (from w
ordcloud) (1.22.3)
Requirement already satisfied: cycler>=0.10 in /Users/nidhisoley/opt/anaconda3/lib/python3.8/site-packages (from m
atplotlib->wordcloud) (0.10.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in /Users/nidhisoley/opt/anaconda3/lib/pyt
hon3.8/site-packages (from matplotlib->wordcloud) (2.4.7)
Requirement already satisfied: python-dateutil>=2.1 in /Users/nidhisoley/opt/anaconda3/lib/python3.8/site-packages
(from matplotlib->wordcloud) (2.8.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /Users/nidhisoley/opt/anaconda3/lib/python3.8/site-packages (f
rom matplotlib->wordcloud) (1.3.1)
Requirement already satisfied: six in /Users/nidhisoley/opt/anaconda3/lib/python3.8/site-packages (from cycler>=0.
10->matplotlib->wordcloud) (1.15.0)
WARNING: You are using pip version 22.0.4; however, version 22.2.2 is available.
You should consider upgrading via the '/Users/nidhisoley/opt/anaconda3/bin/python -m pip install --upgrade pip' co
mmand.
```

```
[nltk_data] Downloading package abc to /Users/nidhisoley/nltk_data...
[nltk_data]   Package abc is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/nidhisoley/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     /Users/nidhisoley/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!

Requirement already satisfied: textblob in /Users/nidhisoley/opt/anaconda3/lib/python3.8/site-packages (0.17.1)
Requirement already satisfied: nltk>=3.1 in /Users/nidhisoley/opt/anaconda3/lib/python3.8/site-packages (from text
blob) (3.6.1)
Requirement already satisfied: click in /Users/nidhisoley/opt/anaconda3/lib/python3.8/site-packages (from nltk>=3.
1->textblob) (7.1.2)
Requirement already satisfied: joblib in /Users/nidhisoley/opt/anaconda3/lib/python3.8/site-packages (from nltk>=
3.1->textblob) (1.0.1)
Requirement already satisfied: tqdm in /Users/nidhisoley/opt/anaconda3/lib/python3.8/site-packages (from nltk>=3.1
->textblob) (4.59.0)
Requirement already satisfied: regex in /Users/nidhisoley/opt/anaconda3/lib/python3.8/site-packages (from nltk>=3.
1->textblob) (2021.4.4)
WARNING: You are using pip version 22.0.4; however, version 22.2.2 is available.
You should consider upgrading via the '/Users/nidhisoley/opt/anaconda3/bin/python -m pip install --upgrade pip' co
mmand.
Requirement already satisfied: kaleido in /Users/nidhisoley/opt/anaconda3/lib/python3.8/site-packages (0.2.1)
WARNING: You are using pip version 22.0.4; however, version 22.2.2 is available.
You should consider upgrading via the '/Users/nidhisoley/opt/anaconda3/bin/python -m pip install --upgrade pip' co
mmand.

[nltk_data] Downloading package vader_lexicon to
[nltk_data]     /Users/nidhisoley/nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!
```

1 Cleaning Text and preprocessing

In [2]:

```
df=pd.read_csv('vaccination_tweets.csv')
```

In [3]:
```
def utils_preprocess_text(text, flg_stemm=False, flg_lemm=True, lst_stopwords=None):
    ## clean (convert to lowercase and remove punctuations and characters and then strip)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub(r'\s+', ' ', text, flags=re.I)
    text = re.sub('\[.*?\]', '', text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub(r'[^\w\s]', '', str(text).lower().strip())

    ## Tokenize (convert from string to list)
    lst_text = text.split()
    ## remove Stopwords
    if lst_stopwords is not None:
        lst_text = [word for word in lst_text if word not in
                    lst_stopwords]

    ## Stemming (remove -ing, -ly, ...)
    if flg_stemm == True:
        ps = nltk.stem.porter.PorterStemmer()
        lst_text = [ps.stem(word) for word in lst_text]

    ## Lemmatisation (convert the word into root word)
    if flg_lemm == True:
        lem = nltk.stem.wordnet.WordNetLemmatizer()
        lst_text = [lem.lemmatize(word) for word in lst_text]

    ## back to string from list
    text = " ".join(lst_text)
    return text
```

In [4]:
```
xt_clean"] = df["text"].apply(lambda x: utils_preprocess_text(x, flg_stemm=False, flg_lemm=True,  lst_stopwords=lst_s
```

2 EDA

In [5]:
```
df=df.drop_duplicates(subset='user_name') #taking one tweet from one person
df=df.dropna()
```

In [7]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4197 entries, 0 to 11012
Data columns (total 17 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   id                4197 non-null   float64
 1   user_name         4197 non-null   object 
 2   user_location     4197 non-null   object 
 3   user_description  4197 non-null   object 
 4   user_created      4197 non-null   object 
 5   user_followers    4197 non-null   int64  
 6   user_friends      4197 non-null   int64  
 7   user_favourites   4197 non-null   int64  
 8   user_verified     4197 non-null   bool   
 9   date              4197 non-null   object 
 10  text              4197 non-null   object 
 11  hashtags          4197 non-null   object 
 12  source            4197 non-null   object 
 13  retweets          4197 non-null   int64  
 14  favorites         4197 non-null   int64  
 15  is_retweet        4197 non-null   bool   
 16  text_clean        4197 non-null   object 
dtypes: bool(2), float64(1), int64(5), object(9)
memory usage: 532.8+ KB
```

In [8]:
```
df
```

Out[8]:

	id	user_name	user_location	user_description	user_created	user_followers	user_friends	user_favourites	user_verified
0	1.340540e+18	Rachel Roh	La Crescenta-Montrose, CA	Aggregator of Asian American news; scanning di...	4/8/09 17:52	405	1692	3247	False
2	1.337860e+18	eli🇱🇹 🇪🇺 👌	Your Bed	heil, hydra 🖐☺	6/25/20 23:30	10	88	155	False
6	1.337850e+18	Gunther Fehlinger	Austria, Ukraine and Kosovo	End North Stream 2 now - the pipeline of corru...	6/10/13 17:49	2731	5001	69344	False
9	1.337840e+18	Ch.Amjad Ali	Islamabad	#ProudPakistani #LovePakArmy #PMIK @insafiansp...	11/12/12 4:18	671	2368	20469	False
10	1.337840e+18	Tamer Yazar	Turkey-Israel	Im Market Analyst, also Editor... working (fre...	9/17/09 16:45	1302	78	339	False
...	...	...	...	...	...	...	...	...	...
10999	1.461460e+18	💝 Poète Universel de Légende ClémentRomainFORTIN	✨✨✨🌍 ✨✨✨☀️ ✨✨✨🌒 ✨✨✨🎇	🎇 🌳 🌱🦇🐻 Million & discoveries, nature, walking,...	12/27/14 23:32	1261	1398	10114	False
11003	1.461280e+18	Margo Payne	Newbury West Berkshire England	#CommunityQueen, Vice President Newbury Lions,...	2/1/09 19:59	743	1067	13275	False
11010	1.461090e+18	Johnny Roque	Los Angeles, CA	I'm a dragon hunter, currently no dragons to h...	11/16/09 16:09	1456	773	5962	False
11011	1.461050e+18	Dr Giacomo Benedetto	United Kingdom	Jean Monnet Chair in European Politics.\nLates...	11/9/12 17:46	1747	1065	6501	False

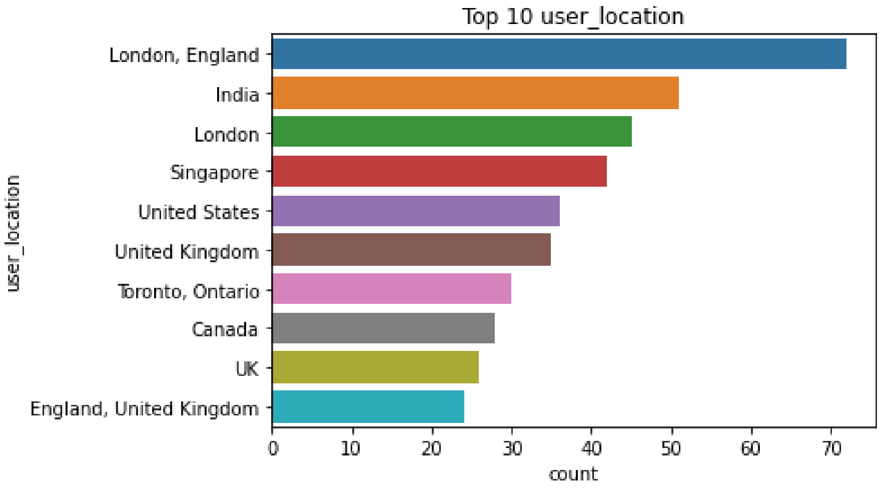
	id	user_name	user_location	user_description	user_created	user_followers	user_friends	user_favourites	user_verified
11012	1.460980e+18	Lincoln University - College of Agriculture (CA...	Jefferson City, MO	Lincoln University - College of Agriculture, E...	3/25/19 16:35	185	364	114	False

4197 rows × 17 columns

2.1 Top location from where the tweets are done

In [9]:

```
ds = df['user_location'].value_counts().reset_index()
ds.columns = ['user_location', 'count']
ds = ds.sort_values(['count'],ascending=False)
fig = sns.barplot(
    x=ds.head(10)["count"],
    y=ds.head(10)['user_location'],
    orientation='horizontal',
).set_title('Top 10 user_location')
```



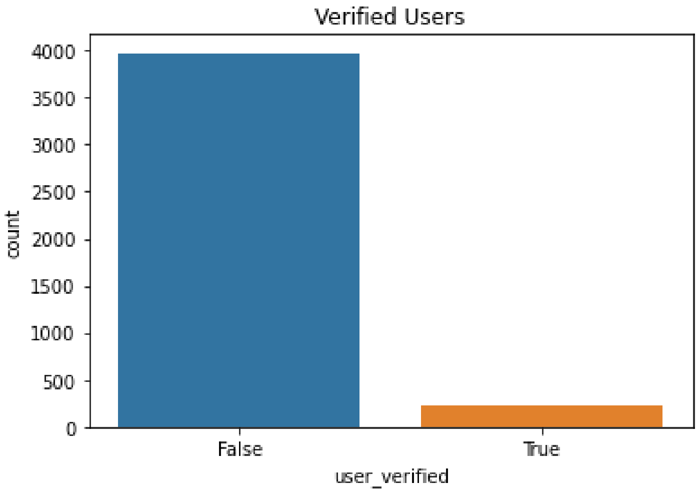
2.2 User verified or not

In [10]:

```
sns.countplot(data=df,x='user_verified').set_title('Verified Users')
```

Out[10]:

```
Text(0.5, 1.0, 'Verified Users')
```

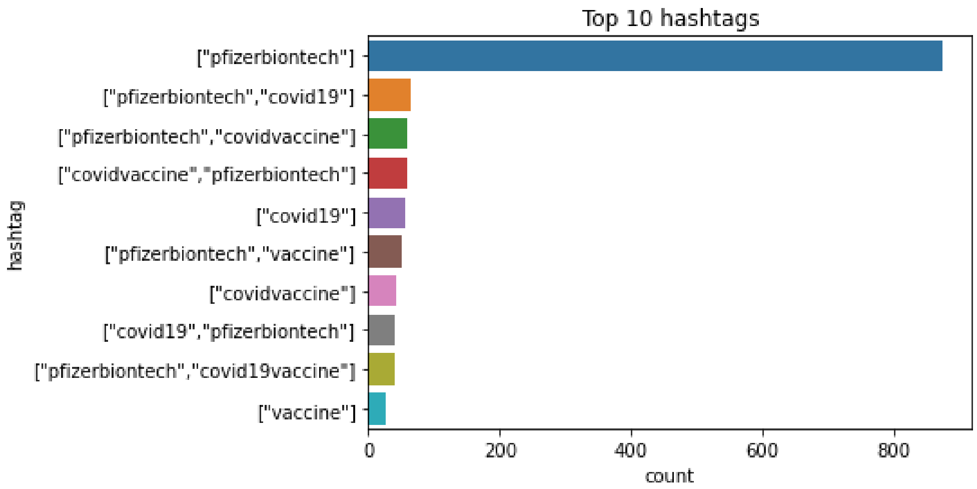


2.3 Top 10 hashtags

In [11]:

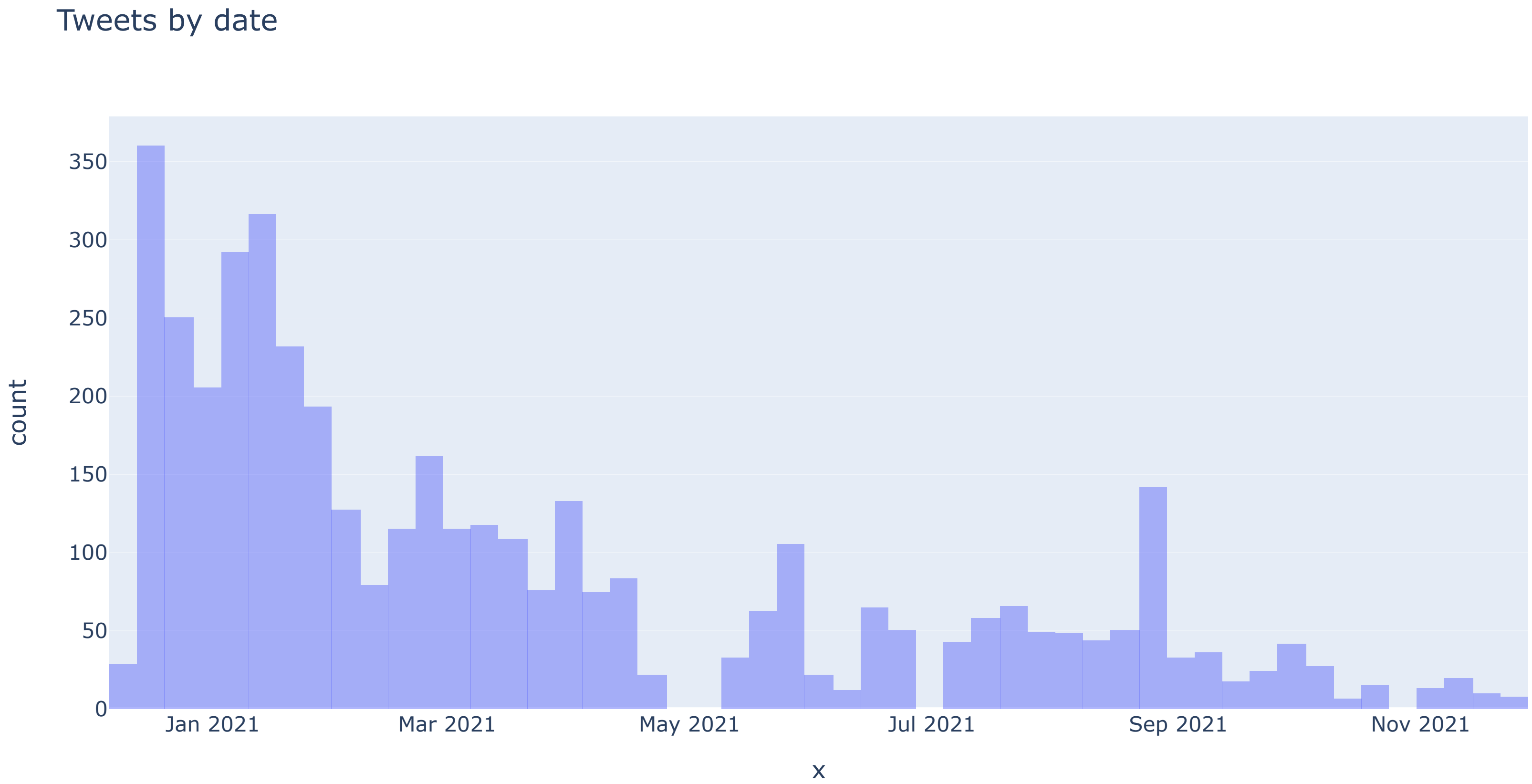
```
def split_hashtags(x):
    return str(x).replace('[', '').replace(']', '').split(',')
df1 = df.copy()
df1['hashtag'] = df1['hashtags'].apply(lambda row : split_hashtags(row))
df11 = df1.explode('hashtag')
df1['hashtag'] = df1['hashtag'].astype(str).str.lower().str.replace("'", '').str.replace(" ", '')
df1.loc[df1['hashtag']=='', 'hashtag'] = 'NO HASHTAG'


ds = df1['hashtag'].value_counts().reset_index()
ds.columns = ['hashtag', 'count']
ds = ds.sort_values(['count'],ascending=False)
fig = sns.barplot(
    x=ds.head(10)["count"],
    y=ds.head(10)['hashtag'],
    orientation='horizontal',
    #title='Top 20 hashtags',
    #width=800,
    #height=700
).set_title('Top 10 hashtags')
#fig.show()
```



2.4 Tweets by date

In [35]:
```
date=pd.to_datetime(df['date']).dt.date
px.histogram(df, x=date,  nbins=100,opacity=.5,title="Tweets by date")
```



3 Sentiment Analysis

In [13]:
```
r_followers','user_friends','user_favourites','user_verified','source',  'retweets', 'favorites',    'is_retweet'])
```

In [14]:

```
#Calculating Negative, Positive, Neutral and Compound values
def pos_neg(data):
    data=data
    for index, row in data.iteritems():
        score = SentimentIntensityAnalyzer().polarity_scores(row)
        neg = score['neg']
        neu = score['neu']
        pos = score['pos']
        comp = score['compound']
        if neg > pos:
            df.loc[index, 'sentiment'] = 'negative'
        elif pos > neg:
            df.loc[index, 'sentiment'] = 'positive'
        else:
            df.loc[index, 'sentiment'] = 'neutral'
        df.loc[index, 'neg'] = neg
        df.loc[index, 'neu'] = neu
        df.loc[index, 'pos'] = pos
        df.loc[index, 'compound'] = comp
```

In [15]:

```
pos_neg(df['text_clean']) #sentiment analysis of the cleaned tweet.
```

In [16]:

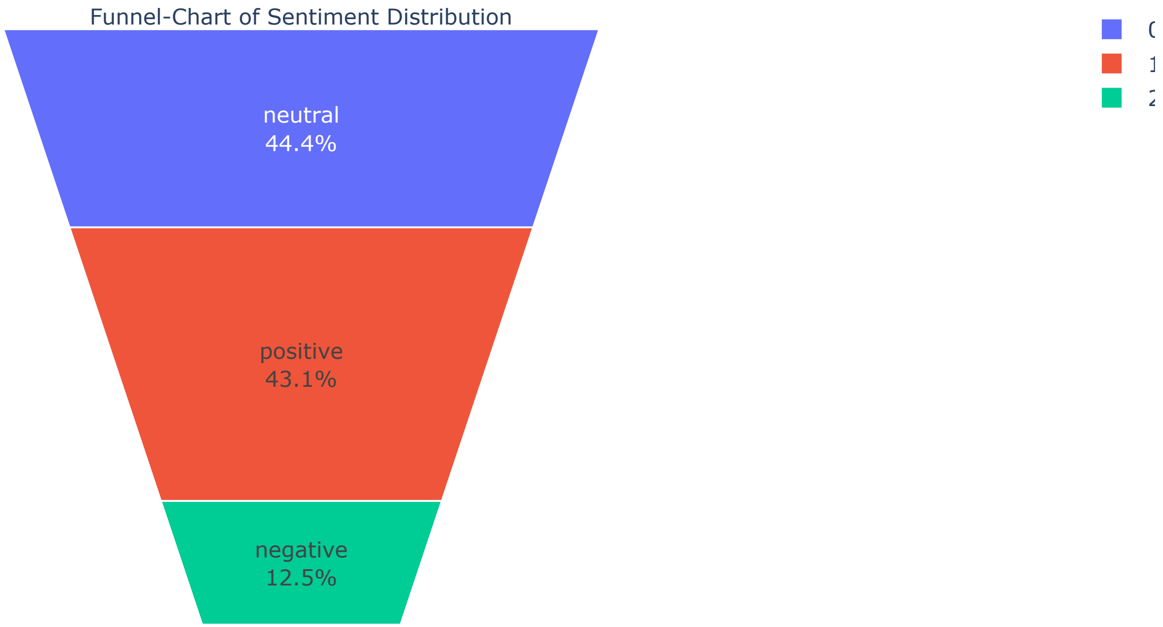
```
temp = df.groupby('sentiment').count()['text'].reset_index().sort_values(by='text',ascending=False)
temp.style.background_gradient(cmap='Purples')
```

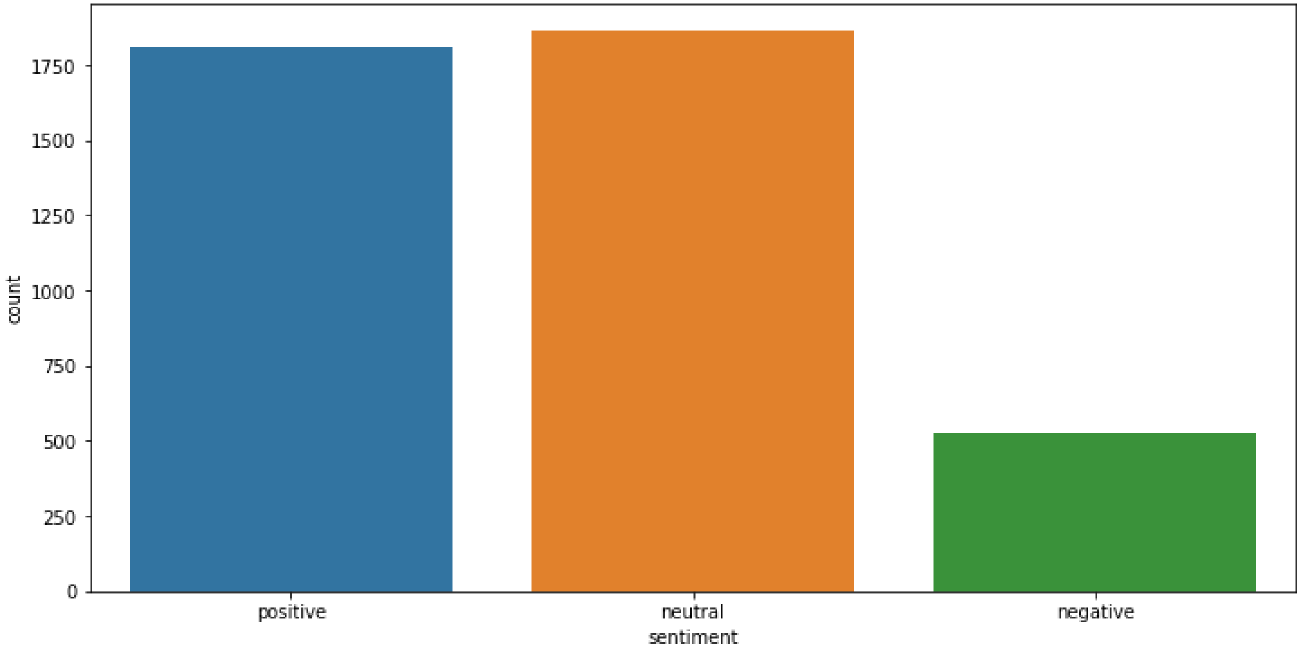
Out[16]:

	sentiment	text
1	neutral	1863
2	positive	1808
0	negative	526

In [17]:

```
plt.figure(figsize=(12,6))
sns.countplot(x='sentiment',data=df)
fig = go.Figure(go.Funnelarea(
    text =temp.sentiment,
    values = temp.text,
    title = {"position": "top center", "text": "Funnel-Chart of Sentiment Distribution"}
    ))
fig.show()
```

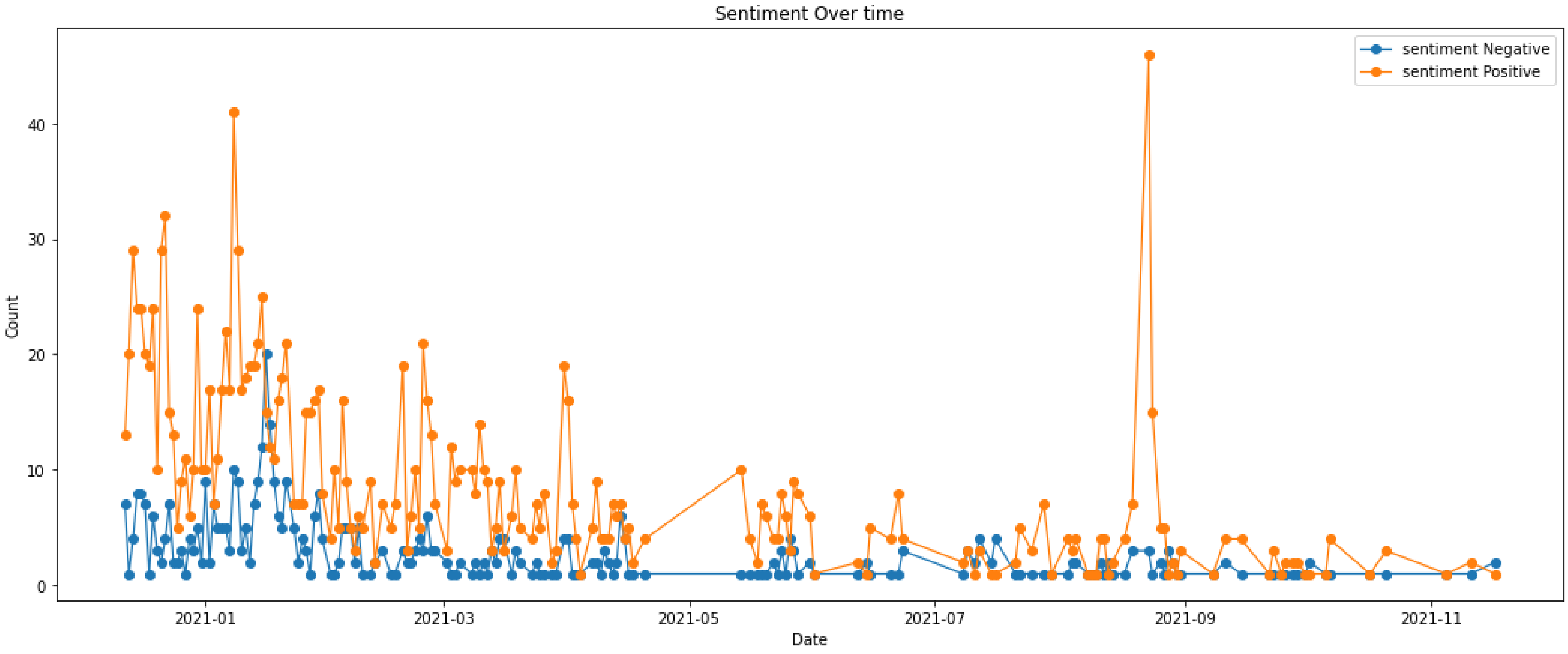




3.1 Change in sentiment with respect to time.

In [18]:
```
df['date'] = pd.to_datetime(df['date']).dt.date
negative_data = df[df['sentiment']=='negative'].reset_index()
positive_data = df[df['sentiment']=='positive'].reset_index()
grouped_data_neg = negative_data.groupby('date')['sentiment'].count().reset_index()
grouped_data_pos = positive_data.groupby('date')['sentiment'].count().reset_index()
merged_data = pd.merge(grouped_data_neg, grouped_data_pos, left_on='date', right_on='date', suffixes=(' Negative', '
merged_data
merged_data.plot(x='date', y=['sentiment Negative', 'sentiment Positive'], linewidth=1.2, figsize=(18, 7), marker='o
```

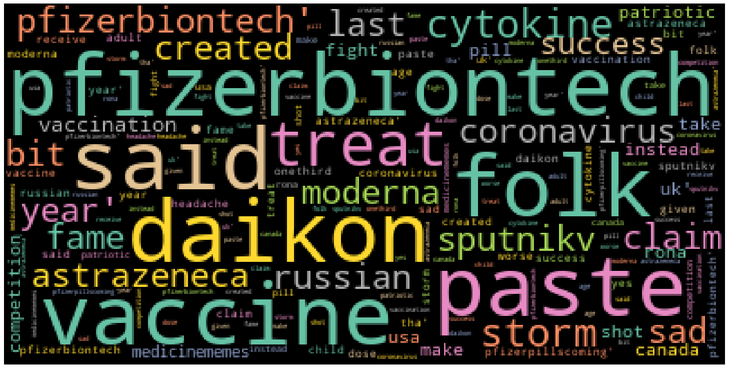
Out[18]:
```
<AxesSubplot:title={'center':'Sentiment Over time'}, xlabel='Date', ylabel='Count'>
```



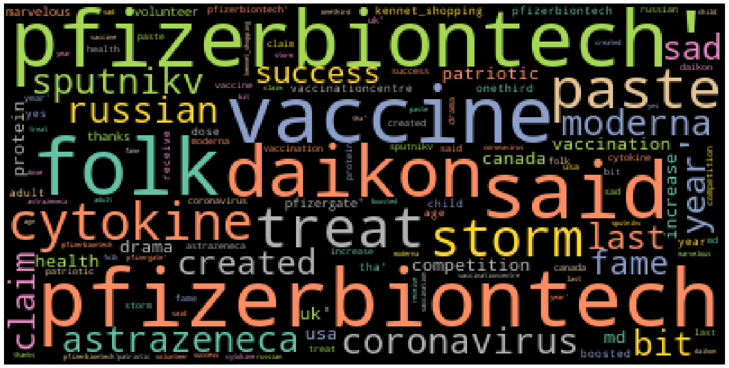
3.2 Word cloud for the cleaned tweet, positive tweets, negative tweets.

In [19]:
```
from PIL import Image
#Function to Create Wordcloud
def create_wordcloud(text):
    stopwords = set(STOPWORDS)
    wc = WordCloud(background_color='black', max_words=3000, stopwords=stopwords, repeat=True,colormap='Set2')
    wc.generate(str(text))
    wc.to_file('wc.png')
    path='wc.png'
    display(Image.open(path))
```

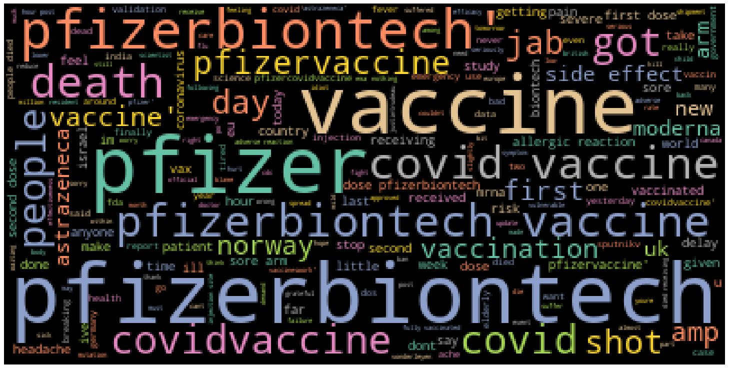
In [20]:
```
#Creating wordcloud for all tweets
create_wordcloud(df['text_clean'].values)
```



In [21]:
```
#Creating wordcloud for positive sentiment
tw_list_positive=df[df['sentiment']=='positive']
create_wordcloud(tw_list_positive['text_clean'].values)
```



In [22]:
```
#Creating wordcloud for negative sentiment
tw_list_negative=df[df['sentiment']=='negative']
create_wordcloud(tw_list_negative['text_clean'].values)
```



4 Sentiment Analysis for the top location of tweet

In [23]:
```
df.user_location.value_counts()
```

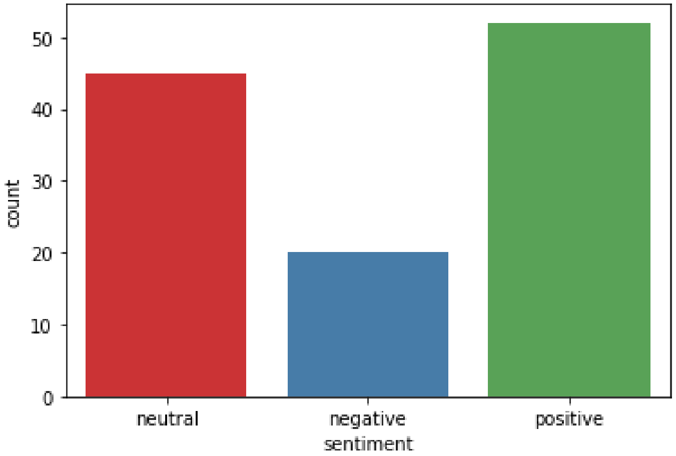
Out[23]:
```
London, England          72
India                    51
London                   45
Singapore                42
United States            36
                         ..
Black Hole                1
UK 🇬🇧  EU 🇪🇺  Earth 🌍         1
DC Metro Area, USA        1
Somewhere in Virginia     1
Halfway There             1
Name: user_location, Length: 2452, dtype: int64
```

In [24]:
```
london=df[(df['user_location']=='London, England') | (df['user_location']=='London')]
# pos_neg(london['text_clean'])
```

In [25]:
```
pos_neg(london['text_clean'])
```

In [26]:
```
sns.countplot(x="sentiment", data=london, palette="Set1")
print(london.sentiment.value_counts())
```

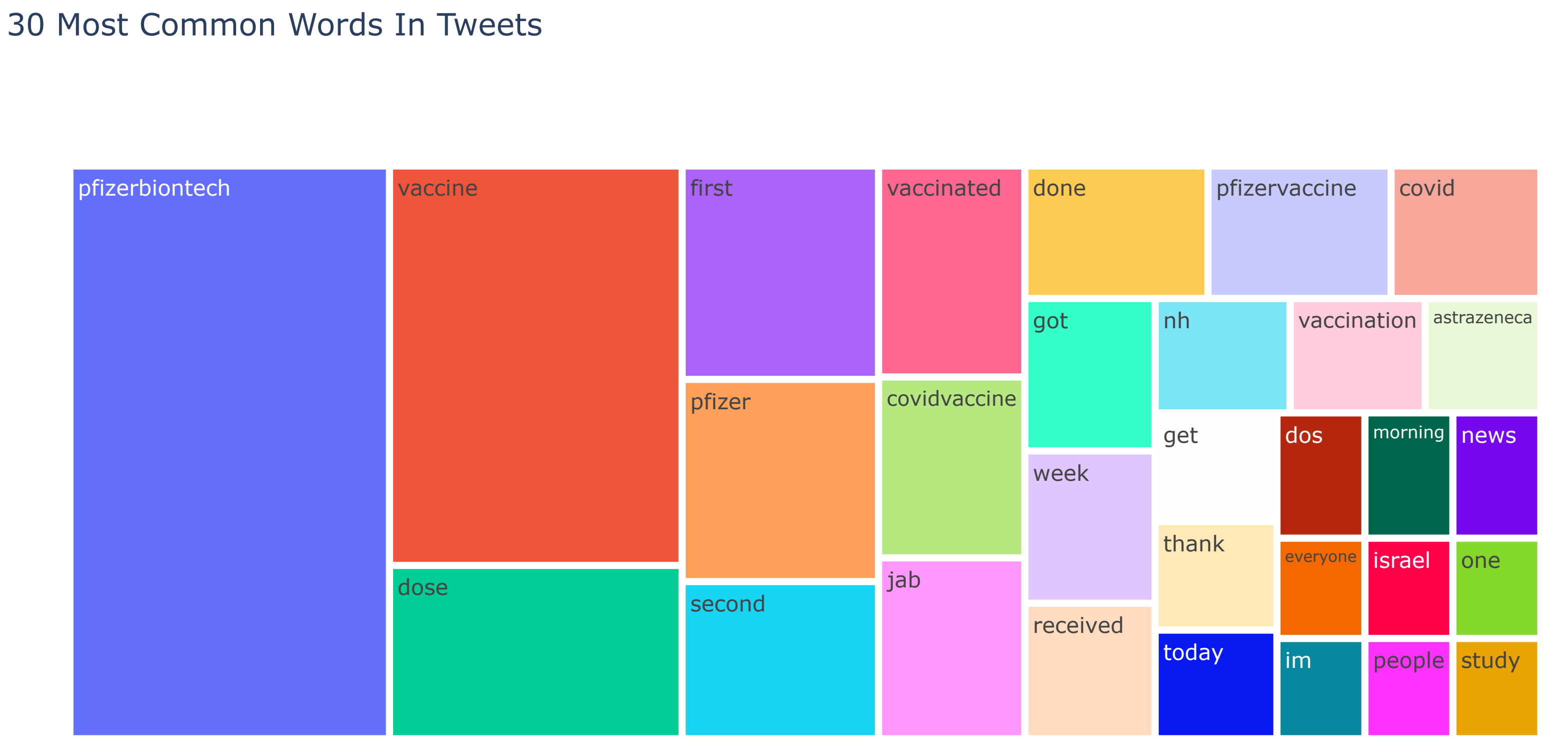
```
positive    52
neutral     45
negative    20
Name: sentiment, dtype: int64
```



4.1 Most common words used by the people of location with highest number of tweets

In [36]:

```
all_words=[]
# london=london.reset_index()
for i in range(len(london['text_clean'])):
    a=london['text_clean'][i].split()
    for i in a:
        all_words.append(i)
all_words=pd.Series(np.array(all_words))
common_words=all_words.value_counts()[:30].rename_axis('Common Words').reset_index(name='count')
fig = px.treemap(common_words, path=['Common Words'], values='count',title='30 Most Common Words In Tweets')
fig.show()
```



In [ ]: