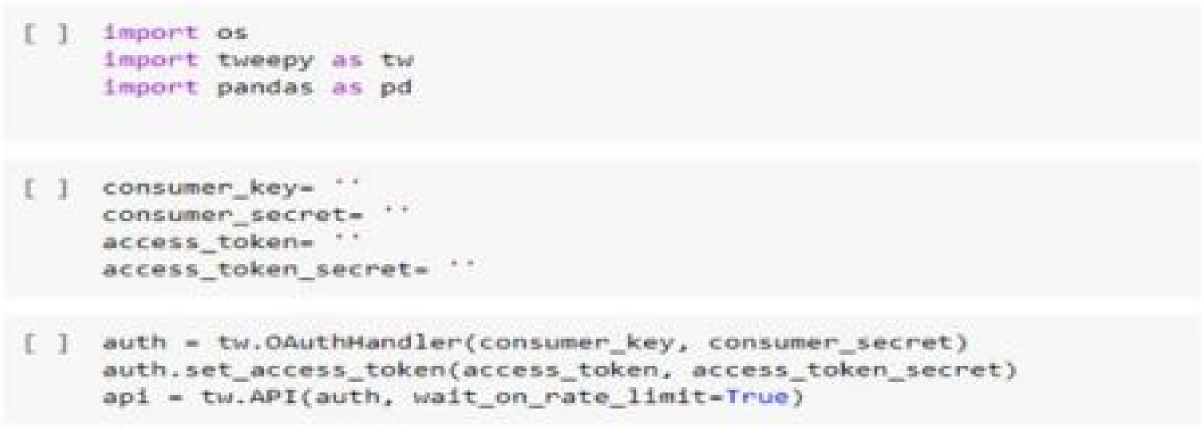
Sentiment Analysis

TOPIC/HASHTAG: #BTS

BTS (*Bangtan Sonyeondan*), also known as the **Bangtan Boys**, is a seven-member South Korean boy band that began formation in 2010 and debuted in 2013 under Big Hit Entertainment. Their lyrics, often focused on personal and social commentary, touch on the themes of mental health, troubles of school-age youth, loss, the journey towards loving oneself, and individualism.

Step 1: Importing all the necessary libraries and inserting the four keys given by twitter developer account.

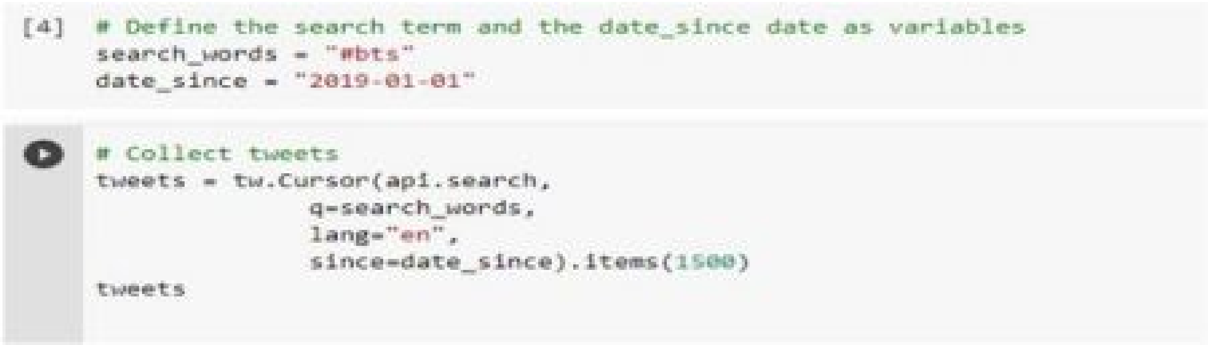


```
[ ]  import os
     import tweepy as tw
     import pandas as pd
```

```
[ ]  consumer_key= ''
     consumer_secret= ''
     access_token= ''
     access_token_secret= ''
```

```
[ ]  auth = tw.OAuthHandler(consumer_key, consumer_secret)
     auth.set_access_token(access_token, access_token_secret)
     api = tw.API(auth, wait_on_rate_limit=True)
```

Step 2: Defining the search term which is 'BTS' in my case and the date_since date as variables and collecting the possible tweets from 1st January 2019.

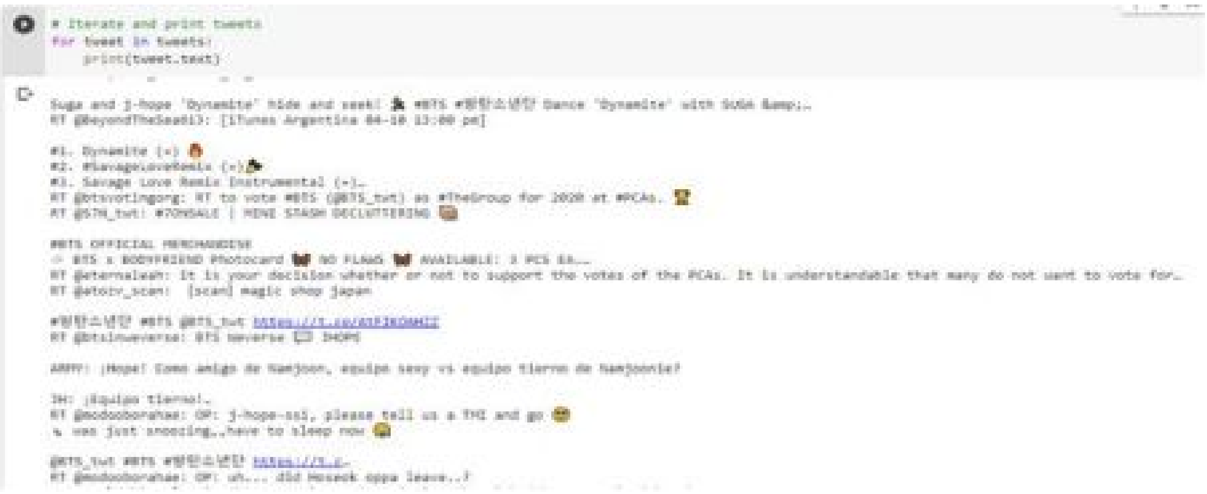


```
[4]  # Define the search term and the date_since date as variables
     search_words = "#bts"
     date_since = "2019-01-01"
```

```
     # Collect tweets
     tweets = tw.Cursor(api.search,
                        q=search_words,
                        lang="en",
                        since=date_since).items(1500)
     tweets
```

Here we are collecting 1500 items or tweets on Mental Health from 1st January, 2019 using the Twitter API. And the language in which the tweets must be is mentioned as en which is English.

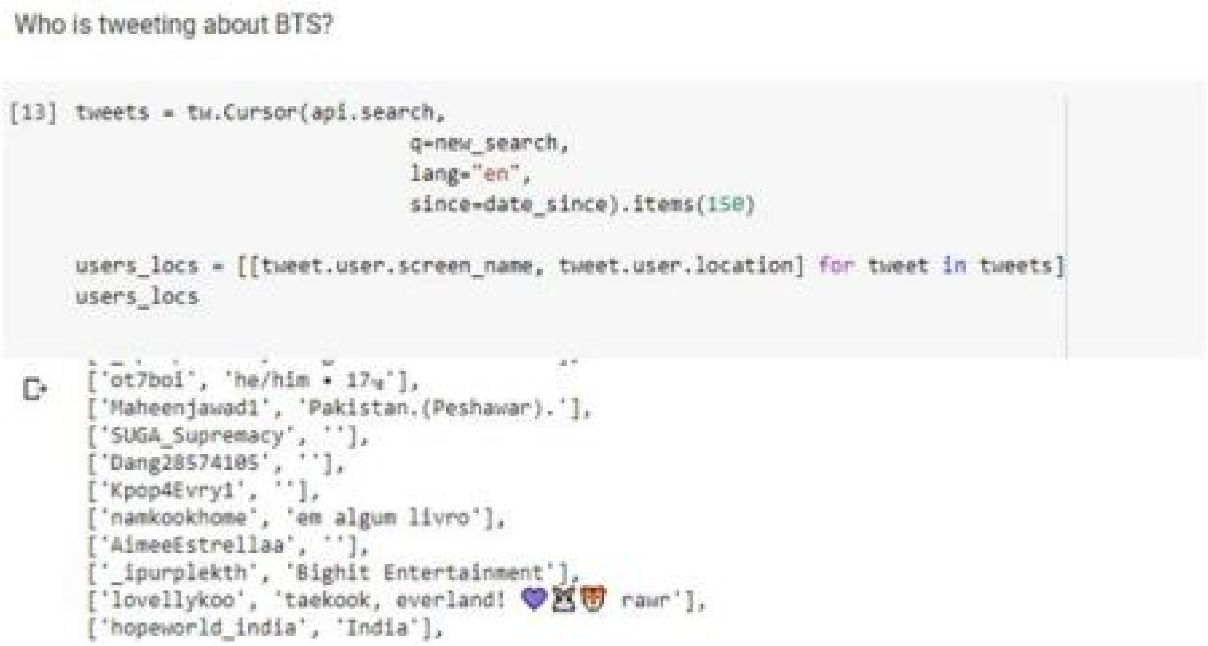
Step 3: Iterating and Printing Tweets



```
# Iterate and print tweets
for tweet in tweets:
    print(tweet.text)
```

```
Suga and j-hope 'Dynamite' hide and seek! 🕺 #BTS #방탄소년단 Dance 'Dynamite' with SUGA &amp;…
RT @BeyondTheSea613: [iTunes Argentina 04-10 13:00 pm]

#1. Dynamite (=) 🔥
#2. #SavageLoveRemix (=)🕺
#3. Savage Love Remix Instrumental (=)…
RT @btsvotingorg: RT to vote #BTS (@BTS_twt) as #TheGroup for 2020 at #PCAs. 🏆
RT @S7N_twt: #7ONSALE | MINI STASH DECLUTTERING 📦

#BTS OFFICIAL MERCHANDISE
☆ BTS x BODYFRIEND Photocard 🦋 NO FLAWS 🦋 AVAILABLE: 3 PCS EA…
RT @eternaleah: It is your decision whether or not to support the votes of the PCAs. It is understandable that many do not want to vote for…
RT @atoiv_scan:  [scan] magic shop japan

#방탄소년단 #BTS @BTS_twt https://t.co/[illegible]
RT @btsinueverse: BTS Weverse 🛍 SHOPS

ARMY: ¡Hope! Como amigo de Namjoon, equipo sexy vs equipo tierno de Namjoonie?

JH: ¡Equipo tierno!…
RT @modooborahae: OP: j-hope-ssi, please tell us a TMI and go 😊
↳ was just snoozing…have to sleep now 😪

@BTS_twt #BTS #방탄소년단 https://t.c…
RT @modooborahae: OP: uh... did Hoseok oppa leave..?
```

Step 4: The following code will tell us who is tweeting about Mental Health

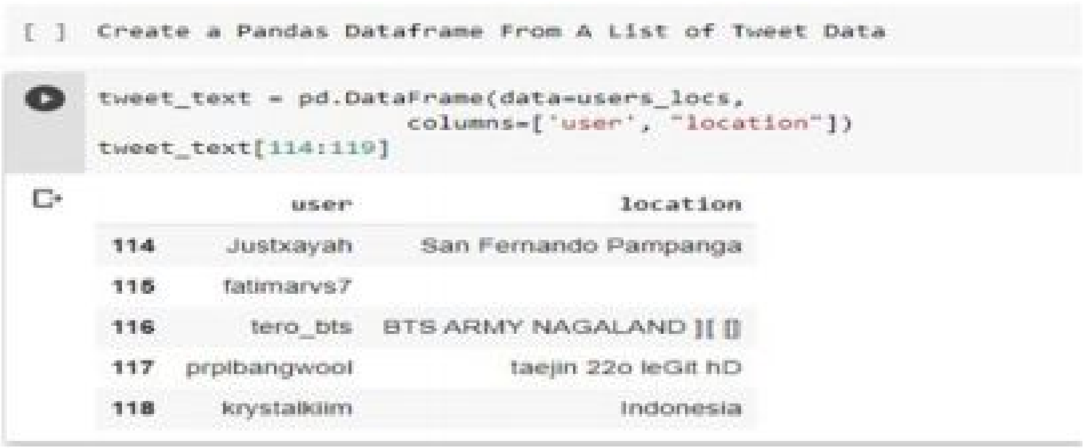


Who is tweeting about BTS?

```
[13] tweets = tw.Cursor(api.search,
                            q=new_search,
                            lang="en",
                            since=date_since).items(150)

     users_locs = [[tweet.user.screen_name, tweet.user.location] for tweet in tweets]
     users_locs
```

```
['ot7boi', 'he/him • 17ᵧ'],
['Maheenjawad1', 'Pakistan.(Peshawar).'],
['SUGA_Supremacy', ''],
['Dang28574185', ''],
['Kpop4Evry1', ''],
['namkookhome', 'em algum livro'],
['AimeeEstrellaa', ''],
['_ipurplekth', 'Bighit Entertainment'],
['lovellykoo', 'taekook, everland! 💜🐰🐯 rawr'],
['hopeworld_india', 'India'],
```

The names of the users are on the left side along with their locations on the right. **Step 5: Creating a Pandas Data frame From A List of Tweet Data**



```
[ ]  Create a Pandas Dataframe From A List of Tweet Data
```

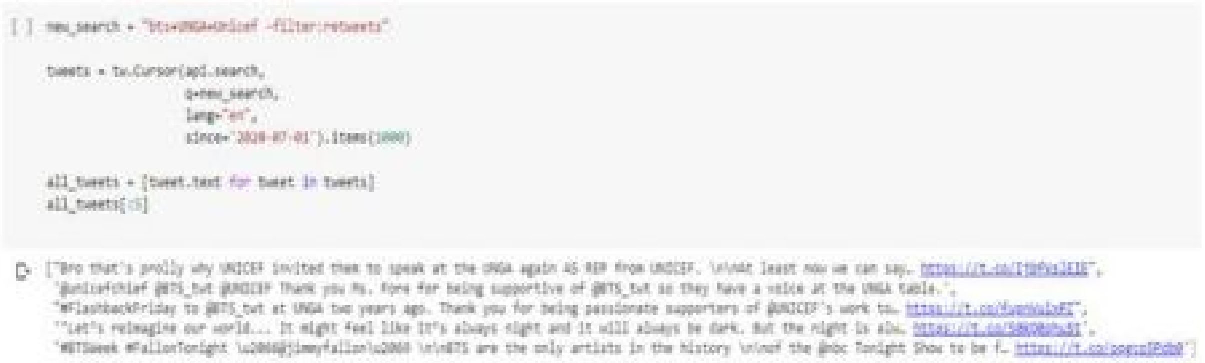
```
tweet_text = pd.DataFrame(data=users_locs,
                          columns=['user', "location"])
tweet_text[114:119]
```

```
            user                    location
114    Justxayah     San Fernando Pampanga
115    fatimarvs7
116     tero_bts   BTS ARMY NAGALAND ][ []
117  prpibangwool          taejin 22o leGit hD
118    krystalkiim                 Indonesia
```

We have converted all the user's name and location in a structured format of rows and columns called as dataframe for better and easy understanding.

Step 6: Customizing Twitter Queries

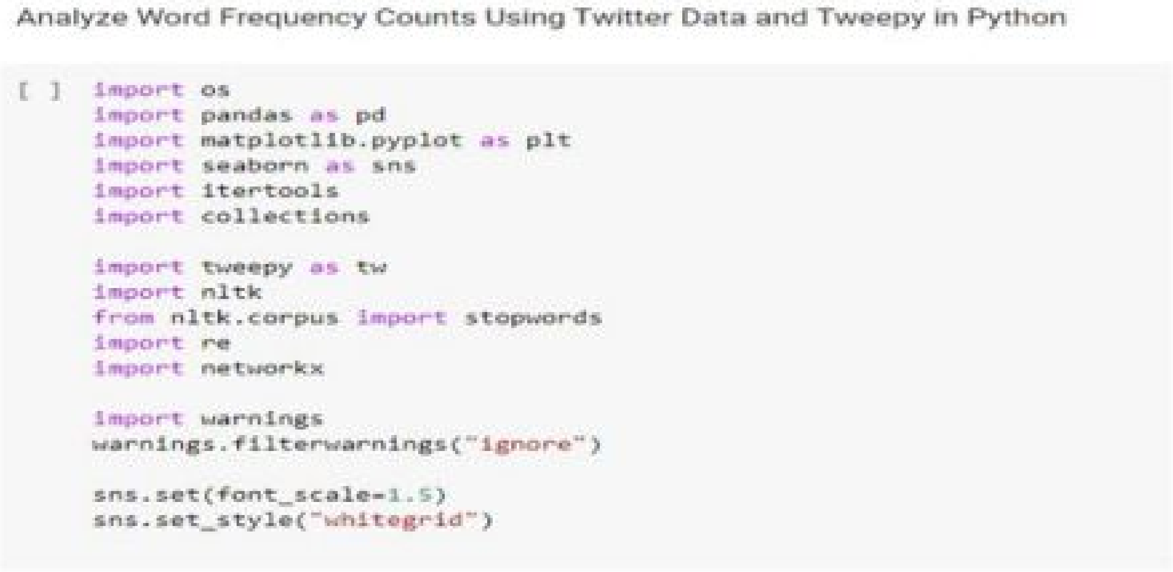
For instance, if you search for BTS + Unicef, Twitter will return all tweets that contain both of those words (in a row) in each tweet.



```
[ ] new_search = "bts+UNGA+Unicef -filter:retweets"

    tweets = tw.Cursor(api.search,
                  q=new_search,
                  lang="en",
                  since='2020-07-01').items(1000)

    all_tweets = [tweet.text for tweet in tweets]
    all_tweets[:5]
```

```
["Bro that's prolly why UNICEF invited them to speak at the UNGA again AS REP from UNICEF. \n\nAt least now we can say… https://t.co/[illegible]",
 '@unicefchief @BTS_twt @UNICEF Thank you Ms. Fore for being supportive of @BTS_twt so they have a voice at the UNGA table.',
 "#FlashbackFriday to @BTS_twt at UNGA two years ago. Thank you for being passionate supporters of @UNICEF's work to… https://t.co/[illegible]",
 '"Let's reimagine our world... It might feel like it's always night and it will always be dark. But the night is alw… https://t.co/[illegible]',
 '#BTSWeek #FallonTonight \u2066@jimmyfallon\u2069 \n\nBTS are the only artists in the history \n\nof the @nbc Tonight Show to be f… https://t.co/[illegible]']
```

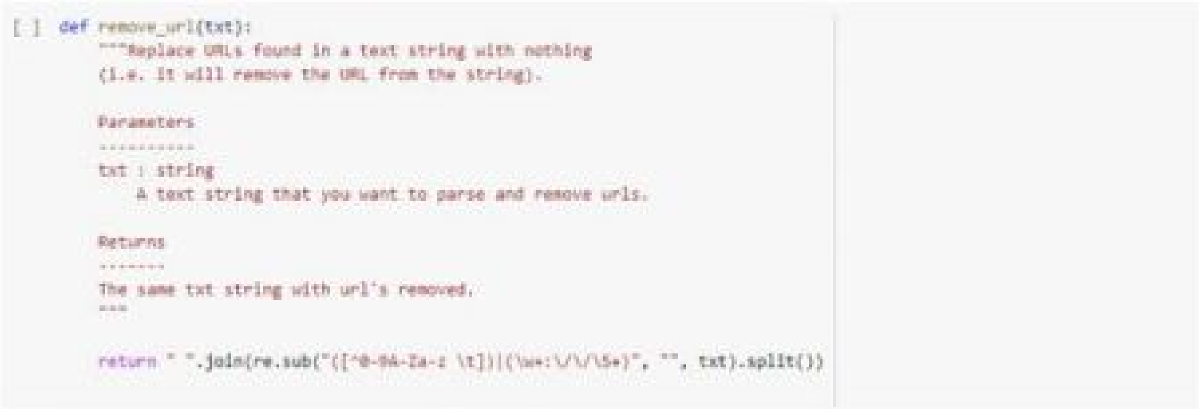
Step 7: Analysing Word Frequency Counts Using Twitter Data and Tweepy in Python



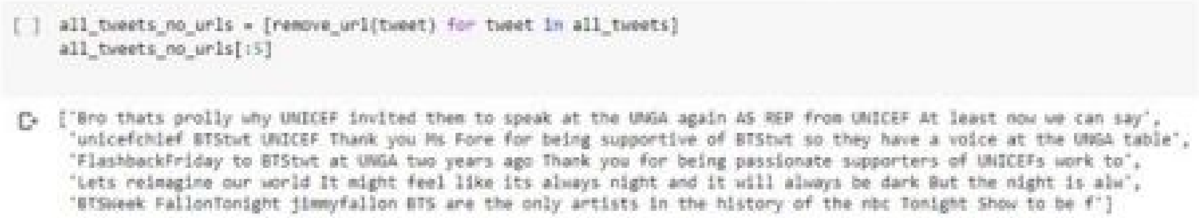
Analyze Word Frequency Counts Using Twitter Data and Tweepy in Python

```
[ ]  import os
     import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns
     import itertools
     import collections

     import tweepy as tw
     import nltk
     from nltk.corpus import stopwords
     import re
     import networkx

     import warnings
     warnings.filterwarnings("ignore")

     sns.set(font_scale=1.5)
     sns.set_style("whitegrid")
```

As we have to go ahead with finding the frequency of words in the tweets. The first step involved is removing all the URL's attached to the tweets.



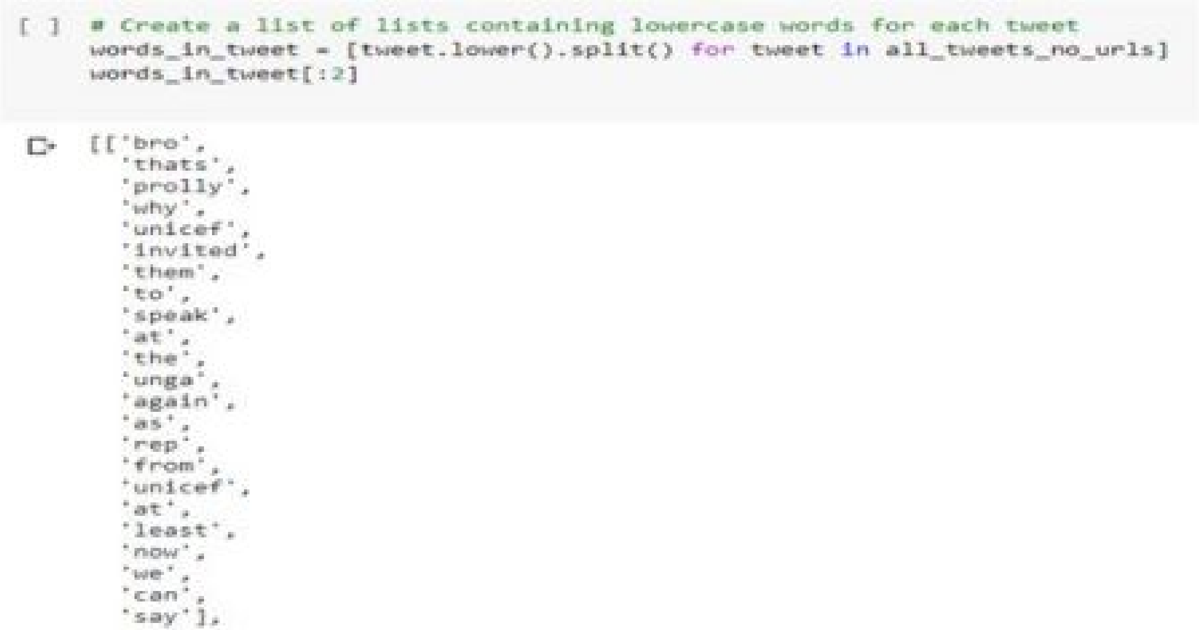
```
[ ] def remove_url(txt):
        """Replace URLs found in a text string with nothing
        (i.e. it will remove the URL from the string).

        Parameters
        ----------
        txt : string
            A text string that you want to parse and remove urls.

        Returns
        -------
        The same txt string with url's removed.
        """

        return " ".join(re.sub("([^0-9A-Za-z \t])|(\w+:\/\/\S+)", "", txt).split())
```



```
[ ] all_tweets_no_urls = [remove_url(tweet) for tweet in all_tweets]
    all_tweets_no_urls[:5]
```

```
['Bro thats prolly why UNICEF invited them to speak at the UNGA again AS REP from UNICEF At least now we can say',
 'unicefchief BTStwt UNICEF Thank you Ms Fore for being supportive of BTStwt so they have a voice at the UNGA table',
 'FlashbackFriday to BTStwt at UNGA two years ago Thank you for being passionate supporters of UNICEFs work to',
 'Lets reimagine our world It might feel like its always night and it will always be dark But the night is alw',
 'BTSWeek FallonTonight jimmyfallon BTS are the only artists in the history of the nbc Tonight Show to be f']
```

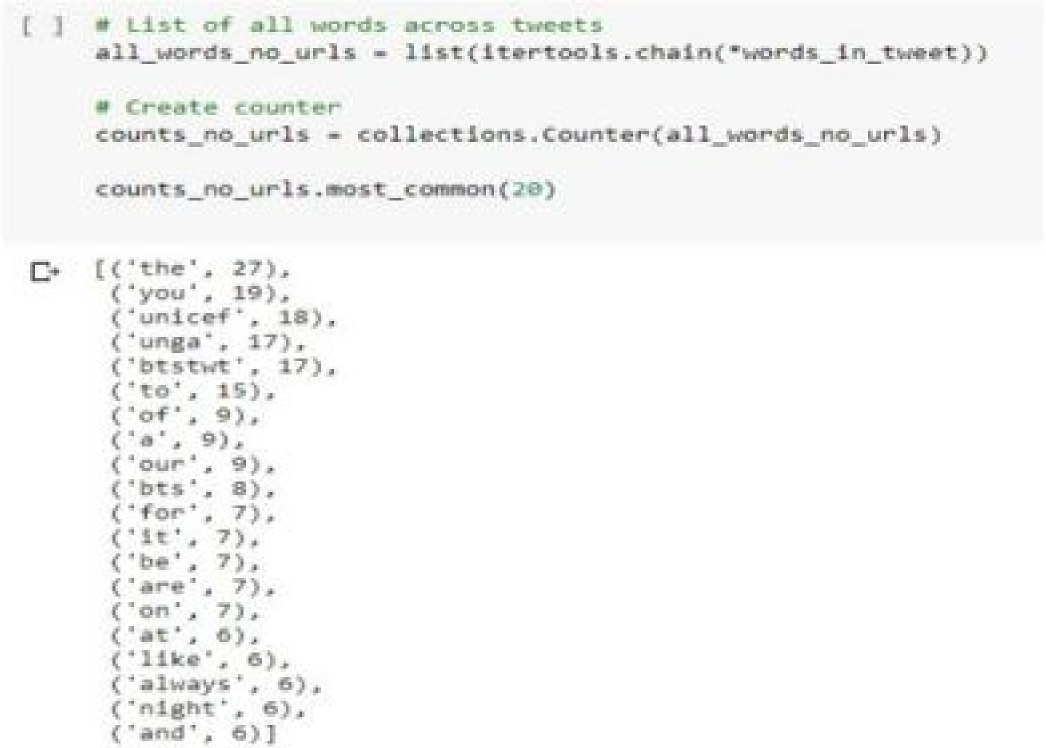
Step 8: Creating a list of lists containing lowercase words for each tweet



```
[ ] # Create a list of lists containing lowercase words for each tweet
    words_in_tweet = [tweet.lower().split() for tweet in all_tweets_no_urls]
    words_in_tweet[:2]
```

```
[['bro',
  'thats',
  'prolly',
  'why',
  'unicef',
  'invited',
  'them',
  'to',
  'speak',
  'at',
  'the',
  'unga',
  'again',
  'as',
  'rep',
  'from',
  'unicef',
  'at',
  'least',
  'now',
  'we',
  'can',
  'say'],
```

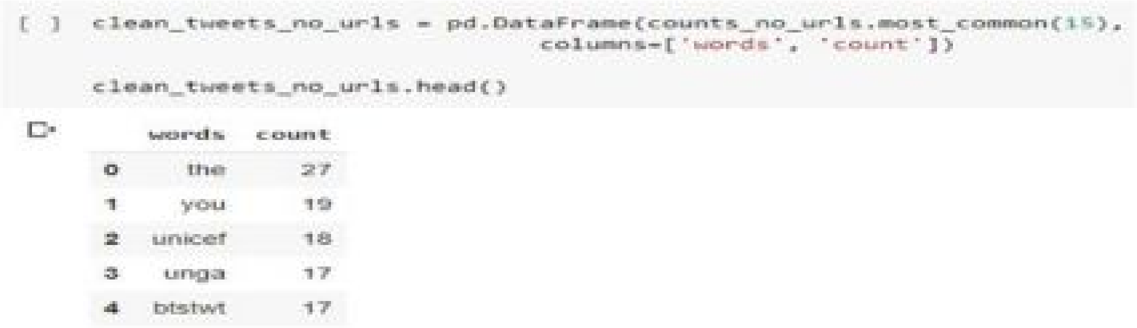
Step 9: Listing of all words across tweets and Creating counter



```
[ ]  # List of all words across tweets
     all_words_no_urls = list(itertools.chain(*words_in_tweet))

     # Create counter
     counts_no_urls = collections.Counter(all_words_no_urls)

     counts_no_urls.most_common(20)
```

```
[('the', 27),
 ('you', 19),
 ('unicef', 18),
 ('unga', 17),
 ('btstwt', 17),
 ('to', 15),
 ('of', 9),
 ('a', 9),
 ('our', 9),
 ('bts', 8),
 ('for', 7),
 ('it', 7),
 ('be', 7),
 ('are', 7),
 ('on', 7),
 ('at', 6),
 ('like', 6),
 ('always', 6),
 ('night', 6),
 ('and', 6)]
```

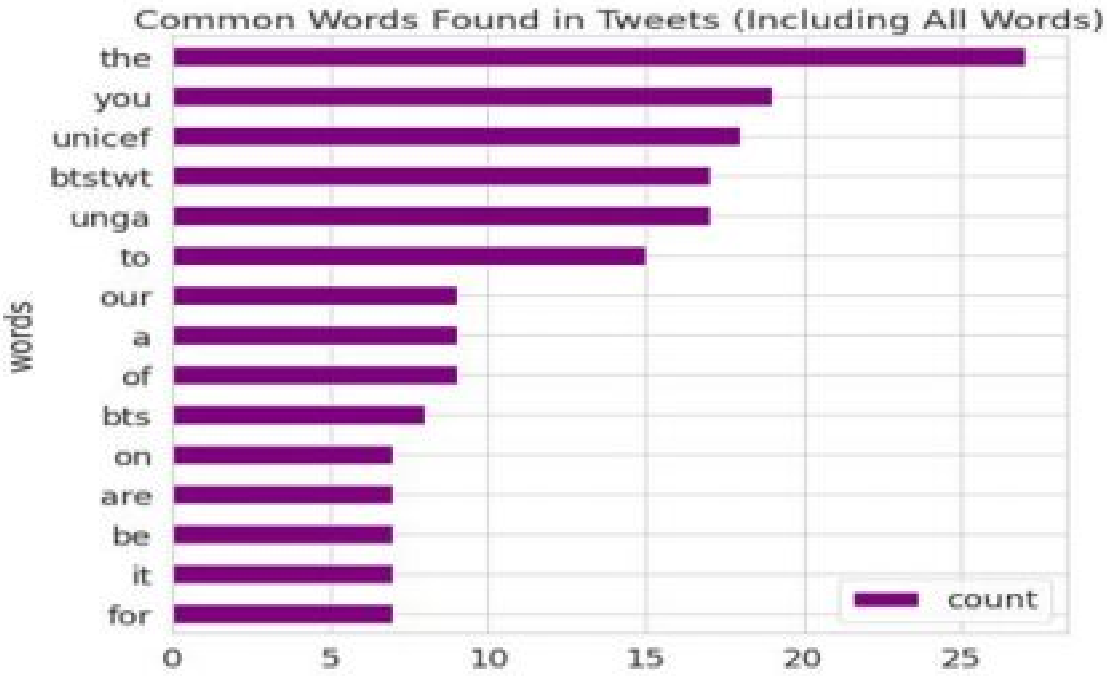
Step 10: Creating a data frame after cleaning tweets



```
[ ]  clean_tweets_no_urls = pd.DataFrame(counts_no_urls.most_common(15),
                                      columns=['words', 'count'])

     clean_tweets_no_urls.head()
```

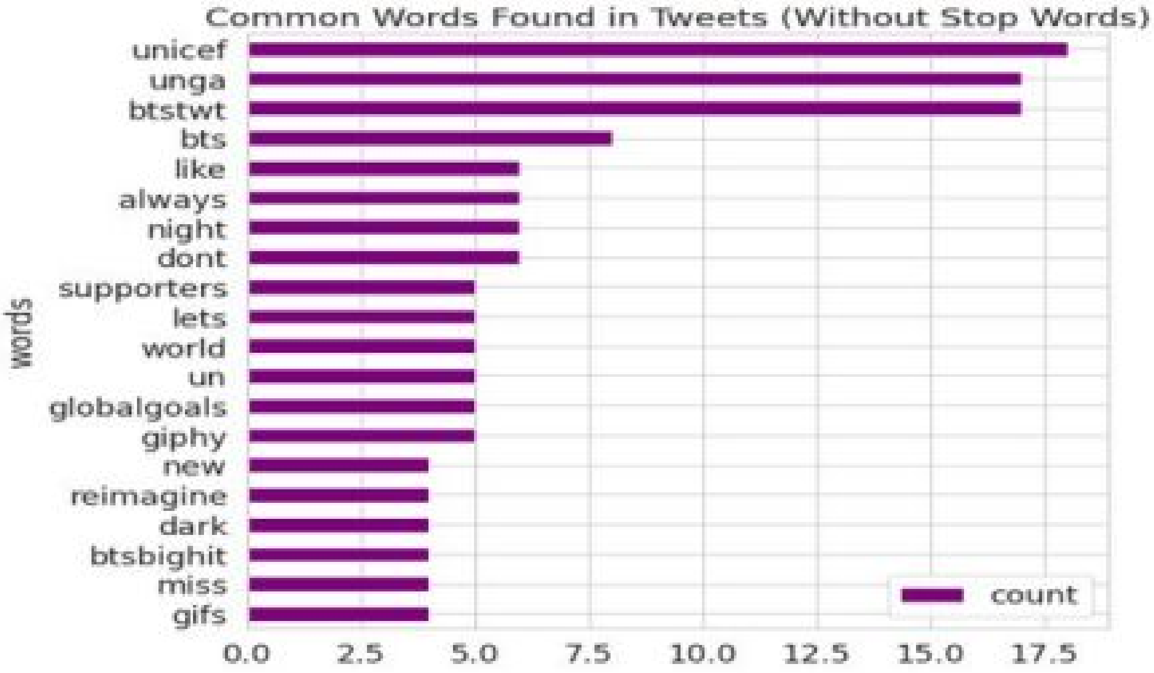
	words	count
0	the	27
1	you	19
2	unicef	18
3	unga	17
4	btstwt	17

Step 11: Plotting a bar graph of most common words used in tweets

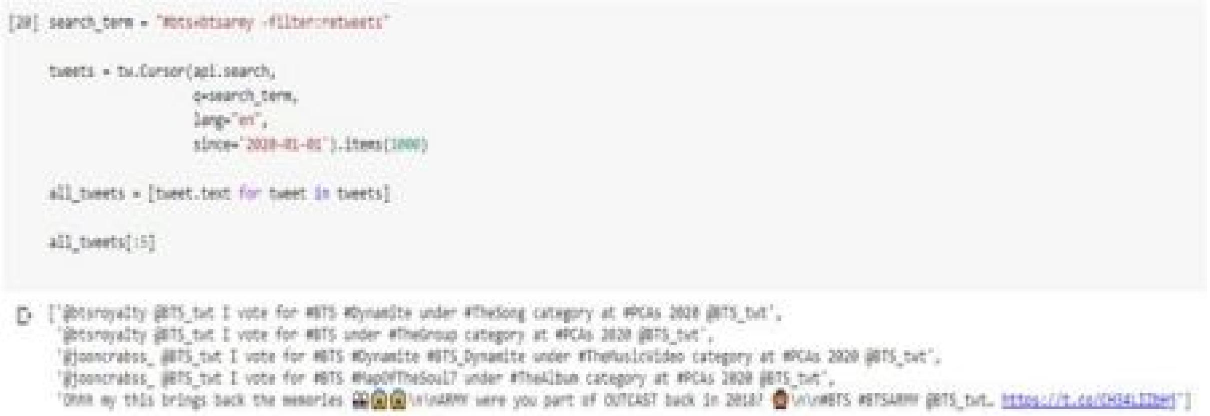


Here we are visualising the first 15 frequently appearing words. But as we can see we also have words such as be, of, on, in etc. These are called stopwords. Therefore we go ahead with the process of removing stop words so that we can get a clear understanding and do not deviate from the objective of the analysis.

Step 12: Removing Stop words and then plotting a bar graph again



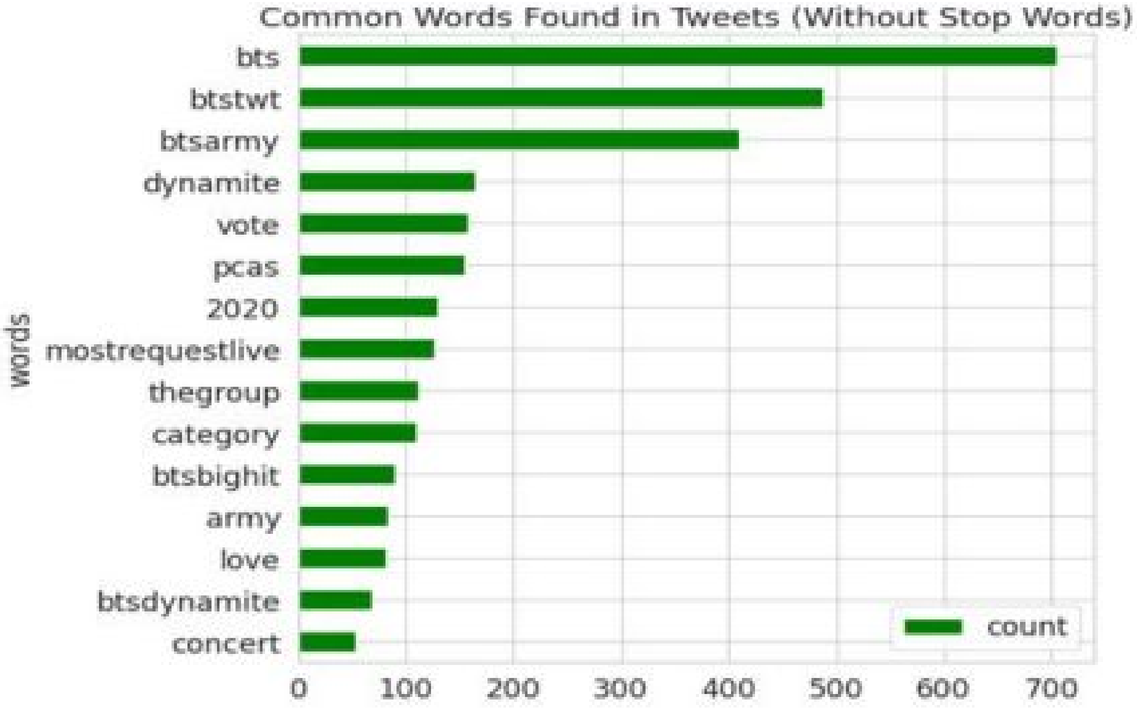
From here we can see that words such as globalgoals, supporters, UN, reimagine, new, world all indicate that people admire BTS. For further we also plot a word cloud for more clarity.



```
[20] search_term = "#bts+btsarmy -filter:retweets"

     tweets = tw.Cursor(api.search,
                        q=search_term,
                        lang="en",
                        since='2020-01-01').items(1000)

     all_tweets = [tweet.text for tweet in tweets]

     all_tweets[:5]
```

```
['@btsroyalty @BTS_twt I vote for #BTS #Dynamite under #TheSong category at #PCAs 2020 @BTS_twt',
 '@btsroyalty @BTS_twt I vote for #BTS under #TheGroup category at #PCAs 2020 @BTS_twt',
 '@jooncrabss_ @BTS_twt I vote for #BTS #Dynamite #BTS_Dynamite under #TheMusicVideo category at #PCAs 2020 @BTS_twt',
 '@jooncrabss_ @BTS_twt I vote for #BTS #MapOfTheSoul7 under #TheAlbum category at #PCAs 2020 @BTS_twt',
 'Ohhh my this brings back the memories 😱😱😱\n\nARMY were you part of OUTCAST back in 2018? 🙈\n\n#BTS #BTSARMY @BTS_twt… https://t.co/CH34LIZbM']
```

We took another customize query “#BTS+btsarmy” on which will perform Sentiment analysis. Remove the url and stopwords and found the most common words in the tweets. Below is the image for same:



Step 13: Creating Text Blobs

Step 14: Calculating Polarity of tweets

Here we are identifying the polarity values for each tweet where all of tweets have positive polarity.

The Histogram shows the polarity of tweets. Here you can see tweets have high positive polarity. The highest number of tweets lies between the polarity ranges of 0.25 to 1.00. If the tweets have polarity value positive which shows that people have very high positive sentiments. And if polarity is negative, it shows that people have negative sentiments towards the subject. Here we can come with a conclusion that people highly admire and like BTS, having so much high positive sentiment towards them.