Week 4 Exercise (group): Exploratory Data Analysis on Social Media Data

- member name1
- member name2
- member name3
- ...

1. Import necessary packages ¶

In [142]:
```
# import necessary packages here

!pip install emoji

import pandas as pd
import nltk
nltk.download('stopwords')
nltk.download('averaged_perceptron_tagger')
from nltk import ngrams
from nltk.collocations import BigramAssocMeasures, BigramCollocationFinde
from nltk.corpus import stopwords
import string
import seaborn as sns
import matplotlib.pyplot as plt
```

```
Requirement already satisfied: emoji in /opt/conda/lib/python3.10/site
-packages (2.2.0)

[nltk_data] Downloading package stopwords to /home/jovyan/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /home/jovyan/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
```

2. Read the data

The data is called `tweets.csv` in the same folder. More information about the data see here (https://www.kaggle.com/datasets/infamouscoder/mental-health-social-media)

The main column you will be working with is `post_text`

In [143]:
```
# df =
df = pd.read_csv("tweets.csv")
df.describe()

# explore the data characteristic using `df.describe()` or `df.info()`
```

Out[143]:

	Unnamed: 0	post_id	user_id	followers	friends	favourites	
count	20000.000000	2.000000e+04	2.000000e+04	20000.000000	20000.000000	20000.000000	2.0
mean	9999.500000	6.874728e+17	3.548623e+16	900.483950	782.428750	6398.235550	4.4
std	5773.647028	1.708396e+17	1.606083e+17	1899.913961	1834.817945	8393.072914	1.4
min	0.000000	3.555966e+09	1.472438e+07	0.000000	0.000000	0.000000	3.0
25%	4999.750000	5.931686e+17	3.242944e+08	177.000000	211.000000	243.000000	5.1
50%	9999.500000	7.637400e+17	1.052122e+09	476.000000	561.000000	2752.000000	1.3
75%	14999.250000	8.153124e+17	2.285923e+09	1197.000000	701.000000	8229.000000	5.2
max	19999.000000	8.194574e+17	7.631825e+17	28614.000000	28514.000000	39008.000000	1.0

3. Extract emojis

Use `emoji` package to extract emojis and put them into a new column called `emojis`

In [144]:
```
!pip install emoji
import emoji
import re
```

```
Requirement already satisfied: emoji in /opt/conda/lib/python3.10/site
-packages (2.2.0)
```

In [145]:
```
# define the function
def extract_emojis(post_text):
    emoji_pattern = re.compile("["
                               u"\U0001F600-\U0001F64F"
                               u"\U0001F300-\U0001F5FF"
                               u"\U0001F680-\U0001F6FF"
                               u"\U0001F1E0-\U0001F1FF"
                               "]+", flags=re.UNICODE)
    return emoji_pattern.findall(post_text)
# apply the function to your dataframe

df['emojis'] = df['post_text'].apply(extract_emojis)
df['emojis'].head()
```

Out[145]:
```
0    []
1    []
2    []
3    []
4    []
Name: emojis, dtype: object
```

4. Text Cleaning using Regular Expressions

1. Remove URLs
2. Remove mentions
3. Remove hashtags
4. Remove special characters
5. Remove extra space

Code can be found in [week 6 lecture 1 (https://github.com/yibeichan/COMM160DS/blob/main/week_6/lecture_part1.ipynb)](https://github.com/yibeichan/COMM160DS/blob/main/week_6/lecture_part1.ipynb) section `4.4 All-in-One`

Perform the analysis and save the results into a new column.

In [146]:
```
!pip install textblob
# define the function
def clean_text(post_text):
    text = re.sub(r"@\w+", "", post_text)  # Remove mentions
    text = re.sub(r"http\S+|www\S+|https\S+", "", post_text, flags=re.MUL
    text = text.lower()  # Convert to lowercase
    return post_text

# apply the function to your dataframe
df['cleaned_text'] = df['post_text'].apply(clean_text)
df['cleaned_text'].head()
```

```
Requirement already satisfied: textblob in /opt/conda/lib/python3.10/s
ite-packages (0.17.1)
Requirement already satisfied: nltk>=3.1 in /opt/conda/lib/python3.10/
site-packages (from textblob) (3.8.1)
Requirement already satisfied: joblib in /opt/conda/lib/python3.10/sit
e-packages (from nltk>=3.1->textblob) (1.2.0)
Requirement already satisfied: tqdm in /opt/conda/lib/python3.10/site-
packages (from nltk>=3.1->textblob) (4.65.0)
Requirement already satisfied: regex>=2021.8.3 in /opt/conda/lib/pytho
n3.10/site-packages (from nltk>=3.1->textblob) (2023.5.5)
Requirement already satisfied: click in /opt/conda/lib/python3.10/site
-packages (from nltk>=3.1->textblob) (8.1.3)
```

Out[146]:
```
0    It's just over 2 years since I was diagnosed w...
1    It's Sunday, I need a break, so I'm planning t...
2    Awake but tired. I need to sleep but my brain ...
3    RT @SewHQ: #Retro bears make perfect gifts and...
4    It’s hard to say whether packing lists are mak...
Name: cleaned_text, dtype: object
```

5. Analysis 1 (Rename the title with your chosen analysis)

Choose one analysis from (1)Sentiment Analysis, (2)N-grams and Phrase Analysis, (3)Collocation Analysis, (4)Part-of-Speech Tagging, (5)Named Entity Recognition, and (6)Dependency Parsing.

Perform the analysis and save the results into a new column.

In [147]:
```
# write your code here
from nltk import ngrams
from collections import Counter
import pandas as pd
import nltk
nltk.download('stopwords')
```

```
nltk.download('averaged_perceptron_tagger')
from nltk import ngrams
from nltk.collocations import BigramAssocMeasures, BigramCollocationFinde
from nltk.corpus import stopwords
import string

df = pd.read_csv("tweets.csv")









def generate_ngrams(text, n):
    tokens = text.apply(lambda x: x.split())
    return list(tokens.apply(lambda x : list(ngrams(x, n))))


text = df['post_text']
bigrams = generate_ngrams(text, 2)
bigrams [:2]
```

```
[nltk_data] Downloading package stopwords to /home/jovyan/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /home/jovyan/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
```

Out[147]:

```
[[("It's", 'just'),
  ('just', 'over'),
  ('over', '2'),
  ('2', 'years'),
  ('years', 'since'),
  ('since', 'I'),
  ('I', 'was'),
  ('was', 'diagnosed'),
  ('diagnosed', 'with'),
  ('with', '#anxiety'),
  ('#anxiety', 'and'),
  ('and', '#depression.'),
  ('#depression.', 'Today'),
  ('Today', "I'm"),
```

```
  ("I'm", 'taking'),
  ('taking', 'a'),
  ('a', 'moment'),
  ('moment', 'to'),
  ('to', 'reflect'),
  ('reflect', 'on'),
  ('on', 'how'),
  ('how', 'far'),
  ('far', "I've"),
  ("I've", 'come'),
  ('come', 'since.')],
 [("It's", 'Sunday,'),
  ('Sunday,', 'I'),
  ('I', 'need'),
  ('need', 'a'),
  ('a', 'break,'),
  ('break,', 'so'),
  ('so', "I'm"),
  ("I'm", 'planning'),
  ('planning', 'to'),
  ('to', 'spend'),
  ('spend', 'as'),
  ('as', 'little'),
  ('little', 'time'),
  ('time', 'as'),
  ('as', 'possible'),
  ('possible', 'on'),
  ('on', 'the'),
  ('the', '#A14...')]]
```

6. Analysis 2 (Rename the title with your chosen analysis)

Choose another analysis from (1)Sentiment Analysis, (2)N-grams and Phrase Analysis, (3)Collocation Analysis, (4)Part-of-Speech Tagging, (5)Named Entity Recognition, and (6)Dependency Parsing.

Perform the analysis and save the results into a new column.

In [*]:
```
# write your code here

import pandas as pd
import nltk
nltk.download('averaged_perceptron_tagger')

def pos_tagging(text):
    tokens = nltk.word_tokenize(text)
    tagged_tokens = nltk.pos_tag(tokens)
    return tagged_tokens

df['pos_tags'] = df['post_text'].apply(pos_tagging)


print(df['pos_tags'].head())
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /home/jovyan/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
```

7. Push Your Results to GitHub

As you did in previous weeks:

1. `git status`
2. `git add`
3. `git commit -m "type your message here"`
4. `git push`

If you can't push it to GitHub, it's okay to manually uploaded it.

In [ ]:

In [ ]: