```
from google.colab import files
import pandas as pd

# Upload the CSV file
uploaded = files.upload()
```

Choose Files Corona_NLP_test.csv

- **Corona_NLP_test.csv**(text/csv) - 1002494 bytes, last modified: 5/5/2023 - 100% done

```
Saving Corona_NLP_test.csv to Corona_NLP_test.csv
```

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

```
# check for missing values
print(df.isna().sum())

# remove rows with missing values in the OriginalTweet column
df = df.dropna(subset=['OriginalTweet'])

# verify that missing values have been removed
print(df.isna().sum())
```

```
UserName           0
ScreenName         0
Location         834
TweetAt            0
OriginalTweet      0
Sentiment          0
dtype: int64
UserName           0
ScreenName         0
Location         834
TweetAt            0
OriginalTweet      0
Sentiment          0
dtype: int64
```

As we are only working the original tweet column i am not cleaning the data based on location

```
df.to_csv('Clean_Corona_NLP_test.csv', index=False)
```

Storing the clean data to Clean_Corona_NLP_test.csv file for re_use of data

a) Convert the text corpus into tokens:

```
import nltk
nltk.download('punkt')

from nltk.tokenize import word_tokenize

# tokenize tweets
df['tokens'] = df['OriginalTweet'].apply(lambda x: word_tokenize(x.lower()))
print(df['tokens'])
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
0       [trending, :, new, yorkers, encounter, empty, ...
1       [when, i, could, n't, find, hand, sanitizer, a...
2       [find, out, how, you, can, protect, yourself, ...
3       [#, panic, buying, hits, #, newyork, city, as,...
4       [#, toiletpaper, #, dunnypaper, #, coronavirus...
                              ...
3793    [meanwhile, in, a, supermarket, in, israel, --...
3794    [did, you, panic, buy, a, lot, of, non-perisha...
3795    [asst, prof, of, economics, @, cconces, was, o...
3796    [gov, need, to, do, somethings, instead, of, b...
3797    [i, and, @, forestandpaper, members, are, comm...
Name: tokens, Length: 3798, dtype: object
```

b) Perform stop word removal:

```
nltk.download('stopwords')

from nltk.corpus import stopwords

stop_words = set(stopwords.words('english'))
print(stop_words)
# remove stop words
df['tokens'] = df['tokens'].apply(lambda x: [word for word in x if word not in stop_words])
print(df['tokens'])
```

```
    {'me', "should've", "hasn't", 'very', 'then', 'down', 'while', 'such', "you've", 'a', "doesn't", 'shan', 'hers', "didn't", 'were', 'only
    0       [trending, :, new, yorkers, encounter, empty, ...
    1       [could, n't, find, hand, sanitizer, fred, meye...
    2       [find, protect, loved, ones, #, coronavirus, ....
    3       [#, panic, buying, hits, #, newyork, city, anx...
    4       [#, toiletpaper, #, dunnypaper, #, coronavirus...
                                   ...
    3793    [meanwhile, supermarket, israel, --, people, d...
    3794    [panic, buy, lot, non-perishable, items, ?, ec...
    3795    [asst, prof, economics, @, cconces, @, nbcphil...
    3796    [gov, need, somethings, instead, biar, je, rak...
    3797    [@, forestandpaper, members, committed, safety...
    Name: tokens, Length: 3798, dtype: object
    [nltk_data] Downloading package stopwords to /root/nltk_data...
    [nltk_data]   Package stopwords is already up-to-date!
```

from the output we can see that the stop words are removed

c) Count word frequencies

```
from collections import Counter

# count word frequencies
df['word_counts'] = df['tokens'].apply(lambda x: Counter(x))
print(df['word_counts'])
```

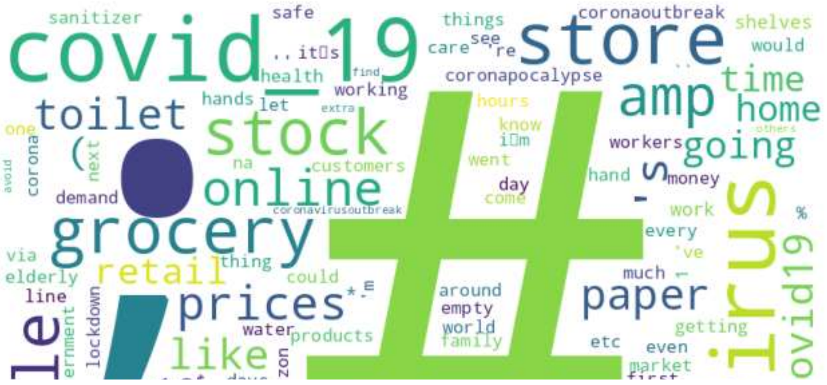
```
    0       {'trending': 1, ':': 3, 'new': 1, 'yorkers': 1...
    1       {'could': 1, 'n't': 1, 'find': 1, 'hand': 1, '...
    2       {'find': 1, 'protect': 1, 'loved': 1, 'ones': ...
    3       {'#': 11, 'panic': 1, 'buying': 1, 'hits': 1, ...
    4       {'#': 11, 'toiletpaper': 1, 'dunnypaper': 1, '...
                                   ...
    3793    {'meanwhile': 1, 'supermarket': 1, 'israel': 1...
    3794    {'panic': 1, 'buy': 1, 'lot': 1, 'non-perishab...
    3795    {'asst': 1, 'prof': 1, 'economics': 1, '@': 2,...
    3796    {'gov': 1, 'need': 1, 'somethings': 1, 'instea...
    3797    {'@': 1, 'forestandpaper': 1, 'members': 1, 'c...
    Name: word_counts, Length: 3798, dtype: object
```

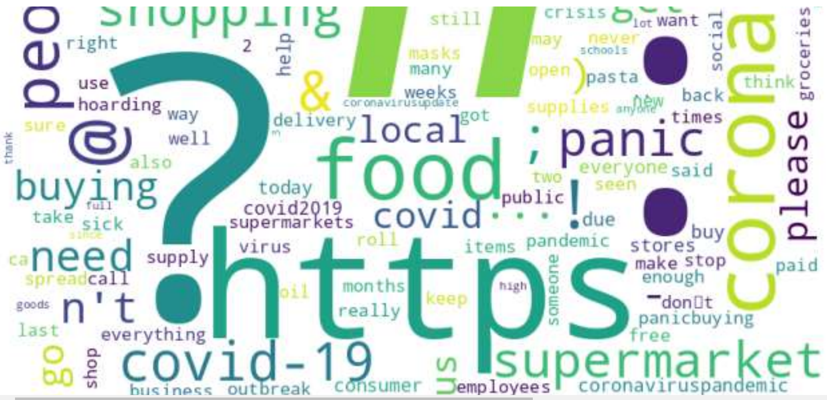
```
# merge all word counts into a single Counter object
all_word_counts = Counter()
for word_count in df['word_counts']:
    all_word_counts.update(word_count)

# display the count of each word
for word, count in all_word_counts.items():
    print(f"{word}: {count}")
    67,000: 1
    compiled: 1
    subscribers: 1
    ebsco: 1
    //t.co/x6dx5jljeh: 1
    hogan: 1
    tens: 1
    marylanders: 2
    //t.co/6jkuoojxvn: 2
    ricepolitics: 1
    mdcounties: 1
    craig: 1
    assembly: 1
    enact: 1
    dance: 1
    sing: 1
    //t.co/vllqgi3r16: 1
    screened: 1
    //t.co/tst4vjhbu4: 1
    asst: 1
    cconces: 1
    nbcphiladelphia: 1
    :33: 1
    //t.co/8tfynoro5l: 1
    somethings: 1
    biar: 1
    je: 1
    rakyat: 1
    assume: 1
    'lockdown: 1
    harini: 1
    semua: 1
    lagi: 1
    mudah: 1
    tu: 1
    tersebar: 1
    forestandpaper: 1
    end-users: 1
    //t.co/qf6hclcaeq: 1
    //t.co/xyvbnsfexa: 1
```

d) Create word clouds:

```
!pip install wordcloud

from wordcloud import WordCloud
import matplotlib.pyplot as plt

# create word cloud
all_word_counts = Counter()
for word_count in df['word_counts']:
    all_word_counts.update(word_count)
wordcloud = WordCloud(width=800, height=800, background_color='white').generate_from_frequencies(all_word_counts)

# display the generated image
plt.figure(figsize=(8,8))
plt.imshow(wordcloud)
plt.axis('off')
plt.show()
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/publ
Requirement already satisfied: wordcloud in /usr/local/lib/python3.10/dist-packages (1.8
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (fr
Requirement already satisfied: numpy>=1.6.1 in /usr/local/lib/python3.10/dist-packages (
Requirement already satisfied: pillow in /usr/local/lib/python3.10/dist-packages (from w
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-package
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packa
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-pa
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packa
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from
```





Colab paid products - Cancel contracts here

✓ 11s completed at 6:50 PM