```
pip install emoji

Collecting emoji
  Downloading emoji-2.12.1-py3-none-any.whl (431 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 431.4/431.4 kB 3.0 MB/s eta
0:00:00
ent already satisfied: typing-extensions>=4.7.0 in
/usr/local/lib/python3.10/dist-packages (from emoji) (4.11.0)
Installing collected packages: emoji
Successfully installed emoji-2.12.1
```

```
import re
import pandas as pd
import numpy as np
import emoji
from collections import Counter
import matplotlib.pyplot as plt
from PIL import Image
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator

def date_time(s):
    pattern='^([0-9]+)(\/)([0-9]+)(\/)([0-9]+), ([0-9]+):([0-9]+)[ ]?
(AM|PM|am|pm)? -'
    result=re.match(pattern, s)
    if result:
        return True
    return False

# Extract contacts
def find_contact(s):
    s=s.split(":")
    if len(s)==2:
        return True
    else:
        return False

# Extract Message
def getMassage(line):
    splitline=line.split(' - ')
    datetime= splitline[0];
    date, time= datetime.split(', ')
    message=" ".join(splitline[1:])

    if find_contact(message):
        splitmessage=message.split(": ")
        author=splitmessage[0]
        message=splitmessage[1]
    else:
        author=None
    return date, time, author, message
```

```
data=[]
conversation='chat.txt'
with open(conversation, encoding="utf-8") as fp:
    fp.readline()
    messageBuffer=[]
    date, time, author= None, None, None
    while True:
        line=fp.readline()
        if not line:
            break
        line=line.strip()
        if date_time(line):
            if len(messageBuffer) >0:
                data.append([date, time, author,
''.join(messageBuffer)])
            messageBuffer.clear()
            date, time, author, message=getMassage(line)
            messageBuffer.append(message)
        else:
            messageBuffer.append(line)

df=pd.DataFrame(data, columns=["Date", "Time", "contact", "Message"])
df['Date']=pd.to_datetime(df['Date'])

data=df.dropna()
from nltk.sentiment.vader import SentimentIntensityAnalyzer
sentiments=SentimentIntensityAnalyzer()
data["positive"]=[sentiments.polarity_scores(i)["pos"] for i in
data["Message"]]
data["negative"]=[sentiments.polarity_scores(i)["neg"] for i in
data["Message"]]
data["neutral"]=[sentiments.polarity_scores(i)["neu"] for i in
data["Message"]]

data.head()
```

```
<ipython-input-5-36b836a462c9>:2: UserWarning: Could not infer format,
so each element will be parsed individually, falling back to
`dateutil`. To ensure parsing is consistent and as-expected, please
specify a format.
  df['Date']=pd.to_datetime(df['Date'])

---------------------------------------------------------------------
-----
LookupError                               Traceback (most recent call
last)
<ipython-input-5-36b836a462c9> in <cell line: 6>()
      4 data=df.dropna()
      5 from nltk.sentiment.vader import SentimentIntensityAnalyzer
----> 6 sentiments=SentimentIntensityAnalyzer()
```

```
      7 data["positive"]=[sentiments.polarity_scores(i)["pos"] for i
in data["Message"]]
      8 data["negative"]=[sentiments.polarity_scores(i)["neg"] for i
in data["Message"]]

/usr/local/lib/python3.10/dist-packages/nltk/sentiment/vader.py in
__init__(self, lexicon_file)
    338
lexicon_file="sentiment/vader_lexicon.zip/vader_lexicon/vader_lexicon.
txt",
    339     ):
--> 340         self.lexicon_file = nltk.data.load(lexicon_file)
    341         self.lexicon = self.make_lex_dict()
    342         self.constants = VaderConstants()

/usr/local/lib/python3.10/dist-packages/nltk/data.py in
load(resource_url, format, cache, verbose, logic_parser,
fstruct_reader, encoding)
    748
    749     # Load the resource.
--> 750     opened_resource = _open(resource_url)
    751
    752     if format == "raw":

/usr/local/lib/python3.10/dist-packages/nltk/data.py in
_open(resource_url)
    874
    875     if protocol is None or protocol.lower() == "nltk":
--> 876         return find(path_, path + [""]).open()
    877     elif protocol.lower() == "file":
    878         # urllib might not use mode='rb', so handle this one
ourselves:

/usr/local/lib/python3.10/dist-packages/nltk/data.py in
find(resource_name, paths)
    581     sep = "*" * 70
    582     resource_not_found = f"\n{sep}\n{msg}\n{sep}\n"
--> 583     raise LookupError(resource_not_found)
    584
    585

LookupError:
**********************************************************************
  Resource vader_lexicon not found.
  Please use the NLTK Downloader to obtain the resource:

  >>> import nltk
  >>> nltk.download('vader_lexicon')

  For more information see: https://www.nltk.org/data.html
```

```
  Attempted to load
sentiment/vader_lexicon.zip/vader_lexicon/vader_lexicon.txt

  Searched in:
    - '/root/nltk_data'
    - '/usr/nltk_data'
    - '/usr/share/nltk_data'
    - '/usr/lib/nltk_data'
    - '/usr/share/nltk_data'
    - '/usr/local/share/nltk_data'
    - '/usr/lib/nltk_data'
    - '/usr/local/lib/nltk_data'
    - ''
**********************************************************************


pip install nltk

Requirement already satisfied: nltk in /usr/local/lib/python3.10/dist-
packages (3.8.1)
Requirement already satisfied: click in
/usr/local/lib/python3.10/dist-packages (from nltk) (8.1.7)
Requirement already satisfied: joblib in
/usr/local/lib/python3.10/dist-packages (from nltk) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in
/usr/local/lib/python3.10/dist-packages (from nltk) (2023.12.25)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-
packages (from nltk) (4.66.4)

df=pd.DataFrame(data, columns=["Date", "Time", "contact", "Message"])
df['Date']=pd.to_datetime(df['Date'])

data=df.dropna()
from nltk.sentiment.vader import SentimentIntensityAnalyzer
sentiments=SentimentIntensityAnalyzer()
data["positive"]=[sentiments.polarity_scores(i)["pos"] for i in
data["Message"]]
data["negative"]=[sentiments.polarity_scores(i)["neg"] for i in
data["Message"]]
data["neutral"]=[sentiments.polarity_scores(i)["neu"] for i in
data["Message"]]

data.head()

---------------------------------------------------------------------
-----
LookupError                               Traceback (most recent call
last)
<ipython-input-7-36b836a462c9> in <cell line: 6>()
      4 data=df.dropna()
```

```
      5 from nltk.sentiment.vader import SentimentIntensityAnalyzer
----> 6 sentiments=SentimentIntensityAnalyzer()
      7 data["positive"]=[sentiments.polarity_scores(i)["pos"] for i
in data["Message"]]
      8 data["negative"]=[sentiments.polarity_scores(i)["neg"] for i
in data["Message"]]

/usr/local/lib/python3.10/dist-packages/nltk/sentiment/vader.py in
__init__(self, lexicon_file)
    338
lexicon_file="sentiment/vader_lexicon.zip/vader_lexicon/vader_lexicon.
txt",
    339     ):
--> 340         self.lexicon_file = nltk.data.load(lexicon_file)
    341         self.lexicon = self.make_lex_dict()
    342         self.constants = VaderConstants()

/usr/local/lib/python3.10/dist-packages/nltk/data.py in
load(resource_url, format, cache, verbose, logic_parser,
fstruct_reader, encoding)
    748
    749     # Load the resource.
--> 750     opened_resource = _open(resource_url)
    751
    752     if format == "raw":

/usr/local/lib/python3.10/dist-packages/nltk/data.py in
_open(resource_url)
    874
    875     if protocol is None or protocol.lower() == "nltk":
--> 876         return find(path_, path + [""]).open()
    877     elif protocol.lower() == "file":
    878         # urllib might not use mode='rb', so handle this one
ourselves:

/usr/local/lib/python3.10/dist-packages/nltk/data.py in
find(resource_name, paths)
    581     sep = "*" * 70
    582     resource_not_found = f"\n{sep}\n{msg}\n{sep}\n"
--> 583     raise LookupError(resource_not_found)
    584
    585

LookupError:
**********************************************************************
  Resource vader_lexicon not found.
  Please use the NLTK Downloader to obtain the resource:

  >>> import nltk
  >>> nltk.download('vader_lexicon')
```

```
  For more information see: https://www.nltk.org/data.html

  Attempted to load
sentiment/vader_lexicon.zip/vader_lexicon/vader_lexicon.txt

  Searched in:
    - '/root/nltk_data'
    - '/usr/nltk_data'
    - '/usr/share/nltk_data'
    - '/usr/lib/nltk_data'
    - '/usr/share/nltk_data'
    - '/usr/local/share/nltk_data'
    - '/usr/lib/nltk_data'
    - '/usr/local/lib/nltk_data'
    - ''
**********************************************************************
```

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

```
[nltk_data] Downloading package vader_lexicon to /root/nltk_data...
```

```
True
```

```
df=pd.DataFrame(data, columns=["Date", "Time", "contact", "Message"])
df['Date']=pd.to_datetime(df['Date'])

data=df.dropna()
from nltk.sentiment.vader import SentimentIntensityAnalyzer
sentiments=SentimentIntensityAnalyzer()
data["positive"]=[sentiments.polarity_scores(i)["pos"] for i in
data["Message"]]
data["negative"]=[sentiments.polarity_scores(i)["neg"] for i in
data["Message"]]
data["neutral"]=[sentiments.polarity_scores(i)["neu"] for i in
data["Message"]]

data.head()
```

```
{"repr_error":"0","type":"dataframe","variable_name":"data"}
```

```
x=sum(data["positive"])
y=sum(data["negative"])
z=sum(data["neutral"])

def score(a,b,c):
    if (a>b) and (a>c):
        print("Positive ")
    if (b>a) and (b>c):
```

```
        print("Negative")
    if (c>a) and (c>b):
        print("Neutral")

score(x,y,z)
```

```
Neutral
```