```
!pip install emoji==1.7.0
```

```
Requirement already satisfied: emoji==1.7.0 in /usr/local/lib/python3.10/dist-packages (1.7.0)
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

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

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

```
import regex
import pandas as pd
import numpy as np
import emoji
from collections import Counter
import matplotlib.pyplot as plt
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 = regex.match(pattern, s)
    if result:
        return True
    return False

def find_author(s):
    s = s.split(":")
    if len(s)==2:
        return True
    else:
        return False

def getDatapoint(line):
    splitline = line.split(' - ')
    dateTime = splitline[0]
    date, time = dateTime.split(", ")
    message = " ".join(splitline[1:])
    if find_author(message):
        splitmessage = message.split(": ")
        author = splitmessage[0]
        message = " ".join(splitmessage[1:])
    else:
        author= None
    return date, time, author, message
```

```
cd /content/drive/MyDrive/shilpa
```

```
/content/drive/MyDrive/shilpa
```

```
data = []
conversation = '/content/drive/MyDrive/shilpa/chart1.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 = getDatapoint(line)
            messageBuffer.append(message)
        else:
            messageBuffer.append(line)
```

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

```
print(df.tail(20))
print(df.info())
print(df.Author.unique())
```

```
3523 2022-04-23   1:53 pm  +91 90042 80656
3524 2022-04-23   1:54 pm  +91 90042 80656
3525 2022-04-23   2:04 pm  +91 90042 80656
3526 2022-04-23   2:04 pm             None
3527 2022-04-25   7:36 am             None
3528 2022-04-25   8:38 am             None
3529 2022-04-25  11:00 am  +91 97683 13673
3530 2022-04-25  11:01 am  +91 97683 13673
3531 2022-04-25  11:06 am  +91 97683 13673
3532 2022-04-27   8:26 am  +91 70214 89118
3533 2022-04-28   5:14 pm  +91 99676 09749
3534 2022-04-28   7:40 pm  +91 70214 89118
3535 2022-04-29   8:17 am             None
3536 2022-04-29   8:17 am  +91 97683 13673

                                              Message
3517  +91 84259 79051: https://docs.google.com/sprea...
3518                                  <Media omitted>
3519                                  <Media omitted>
3520                                  <Media omitted>
3521                                  <Media omitted>
3522  Pratiksha Awate: https://docs.google.com/forms...
3523  web mining me same topic leke sabne huga h, ab...
3524  agar same topics mila kisi ka toh jisne bhi la...
3525                                            Haga
3526  +91 77383 28626: https://docs.google.com/sprea...
3527  +91 97683 13673: Take a break after 11:30am le...
3528      +91 97683 13673: https://youtu.be/oVEuOtRLNzc
3529  Abstract Introduction Objectives Methodology D...
3530                    Ye format bhi use kar sakte ho
3531                                  <Media omitted>
3532  *TODAY IS THE LAST DAY TO GIVE YOUR WEB MINING...
3533             NLP ka project kon kon submitted kiya?
3534                                                 😂
3535  +91 97683 13673: https://docs.google.com/sprea...
3536  Faculty ko forward kar ra hu....jisne bhi nhi ...
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3537 entries, 0 to 3536
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   Date     3537 non-null   datetime64[ns]
 1   Time     3537 non-null   object
 2   Author   3385 non-null   object
 3   Message  3537 non-null   object
dtypes: datetime64[ns](1), object(3)
memory usage: 110.7+ KB
None
[None 'Areej Clg' '+91 90042 80656' '+91 97683 13673' '+91 99672 73815'
 '+91 70214 89118' '+91 70217 45593' '+91 97695 24164' 'Pratiksha Awate'
 '+91 90040 75303' 'Preeti Clg Mumbai' 'Shraddha Panchal'
 'Anshu Clg Mumbai' '+91 87793 59887' '+91 70392 29744' 'Monika Kharkwal'
 '+91 82916 85824' '+91 99676 09749' '+91 84336 34677' '+91 77383 28626'
 'Shilpa Dhanure' '+91 82916 75179' '+91 87790 51155' '+91 84259 79051'
 '+91 96199 36420' '+91 90047 57892' '+91 70214 77723']
<ipython-input-8-66e13cebf647>:2: UserWarning: Could not infer format, so each element will be parsed individually, falling back
  df['Date'] = pd.to_datetime(df['Date'])
```

```
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-9-36b836a462c9>:2: UserWarning: Could not infer format, so each element will be parsed individually, falling back to
  df['Date']=pd.to_datetime(df['Date'])
<ipython-input-9-36b836a462c9>:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  data["positive"]=[sentiments.polarity_scores(i)["pos"] for i in data["Message"]]
<ipython-input-9-36b836a462c9>:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  data["negative"]=[sentiments.polarity_scores(i)["neg"] for i in data["Message"]]
<ipython-input-9-36b836a462c9>:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  data["neutral"]=[sentiments.polarity_scores(i)["neu"] for i in data["Message"]]
```

	Date	Time	contact	Message	positive	negative	neutral
2	2021-10-18	10:02 am	Areej Clg	Ohh	0.0	0.000	1.000
4	2021-10-18	10:57 am	Areej Clg	<Media omitted>	0.0	0.000	1.000
5	2021-10-18	11:53 am	+91 90042 80656	Guys saare lects ke end me ..jisne bhi notes b...	0.0	0.000	1.000
6	2021-10-18	11:54 am	+91 90042 80656	Offline chalu hoega toh kuch book me likhte h ...	0.0	0.103	0.897
7	2021-10-18	11:58 am	Areej Clg	Yes	1.0	0.000	0.000

```
total_messages = df.shape[0]
print(total_messages)
```

```
3537
```

```
media_messages = df[df["Message"]=='<Media omitted>'].shape[0]
print(media_messages)
```

```
314
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
def split_count(text):
    emoji_list = []
    data = regex.findall(r'\X',text)
    for word in data:
        if any(char in emoji.UNICODE_EMOJI for char in word):
            emoji_list.append(word)
    return emoji_list
df['emoji'] = df["Message"].apply(split_count)

emojis = sum(df['emoji'].str.len())
print(emojis)
```

```
0
```

```
URLPATTERN = r'(https?://\S+)'
df['urlcount'] = df.Message.apply(lambda x: regex.findall(URLPATTERN, x)).str.len()
links = np.sum(df.urlcount)

print("Chats between Aman and Sapna")
print("Total Messages: ", total_messages)
print("Number of Media Shared: ", media_messages)

print("Number of Links Shared", links)
```

```
Chats between Aman and Sapna
Total Messages:  3537
Number of Media Shared:  314
Number of Links Shared 74
```

```
media_messages_df = df[df['Message'] == '<Media omitted>']
messages_df = df.drop(media_messages_df.index)
messages_df['Letter_Count'] = messages_df['Message'].apply(lambda s : len(s))
messages_df['Word_Count'] = messages_df['Message'].apply(lambda s : len(s.split(' ')))
messages_df["MessageCount"]=1
```

Start coding or generate with AI.

```
total_emojis_list = list(set([a for b in messages_df.emoji for a in b]))
total_emojis = len(total_emojis_list)

total_emojis_list = list([a for b in messages_df.emoji for a in b])
emoji_dict = dict(Counter(total_emojis_list))
emoji_dict = sorted(emoji_dict.items(), key=lambda x: x[1], reverse=True)
for i in emoji_dict:
  print(i)

emoji_df = pd.DataFrame(emoji_dict, columns=['emoji', 'count'])
import plotly.express as px
fig = px.pie(emoji_df, values='count', names='emoji')
fig.update_traces(textposition='inside', textinfo='percent+label')
fig.show()
```

```
df = pd.DataFrame(data, columns=["Date", 'Time', 'Author', 'Message'])
df['Date'] = pd.to_datetime(df['Date'])
print(df.tail(20))
print(df.info())
print(df.Author.unique())
```

```
            Date      Time  Author  \
3509 2022-04-19   9:19 am     NaN
3510 2022-04-19   9:27 am     NaN
3511 2022-04-19  12:09 pm     NaN
3513 2022-04-19   8:24 pm     NaN
3514 2022-04-19   8:25 pm     NaN
3516 2022-04-20   3:21 pm     NaN
3518 2022-04-23   9:16 am     NaN
3519 2022-04-23   9:16 am     NaN
3520 2022-04-23   9:16 am     NaN
3521 2022-04-23  10:11 am     NaN
3523 2022-04-23   1:53 pm     NaN
3524 2022-04-23   1:54 pm     NaN
3525 2022-04-23   2:04 pm     NaN
3529 2022-04-25  11:00 am     NaN
3530 2022-04-25  11:01 am     NaN
3531 2022-04-25  11:06 am     NaN
3532 2022-04-27   8:26 am     NaN
3533 2022-04-28   5:14 pm     NaN
3534 2022-04-28   7:40 pm     NaN
3536 2022-04-29   8:17 am     NaN

                                          Message
3509                                         iara
```

```
3510                                          What no
3511  ML Project Documentation Title Intro Objective...
3513                                       Recordings
3514                                              NLP
3516  1hr hi milega toh better come atleast 15 min e...
3518                                  <Media omitted>
3519                                  <Media omitted>
3520                                  <Media omitted>
3521                                  <Media omitted>
3523  web mining me same topic leke sabne huga h, ab...
3524  agar same topics mila kisi ka toh jisne bhi la...
3525                                            Haga
3529  Abstract Introduction Objectives Methodology D...
3530                    Ye format bhi use kar sakte ho
3531                                  <Media omitted>
3532  *TODAY IS THE LAST DAY TO GIVE YOUR WEB MINING...
3533            NLP ka project kon kon submitted kiya?
3534                                                 😂
3536  Faculty ko forward kar ra hu....jisne bhi nhi ...
<class 'pandas.core.frame.DataFrame'>
Index: 3385 entries, 2 to 3536
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   Date     3385 non-null   datetime64[ns]
 1   Time     3385 non-null   object
 2   Author   0 non-null      float64
 3   Message  3385 non-null   object
dtypes: datetime64[ns](1), float64(1), object(2)
memory usage: 261.3+ KB
None
[nan]
```

```
total_messages = df.shape[0]
print(total_messages)
```

```
3385
```

```
media_messages = df[df["Message"]=='<Media omitted>'].shape[0]
print(media_messages)
```

```
314
```

```
text = " ".join(review for review in messages_df.Message)
print ("There are {} words in all the messages.".format(len(text)))
stopwords = set(STOPWORDS)
# Generate a word cloud image
wordcloud = WordCloud(stopwords=stopwords, background_color="white").generate(text)
# Display the generated image:
# the matplotlib way:
plt.figure( figsize=(10,5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

```
There are 122713 words in all the messages.
```


```
messages_df = df  # Assign the dataframe 'df' to 'messages_df' to make the 'Author' column available
l = ["Areej Clg", "Preeti Clg Mumbai"]
for i in range(len(l)):
  dummy_df = messages_df[messages_df['Author'] == l[i]]
  text = " ".join(review for review in dummy_df.Message)
```

```
stopwords = set(STOPWORDS)
# Generate a word cloud image
print('Author name',l[i])

# Check if text is empty after stop word removal
words = [word for word in text.split() if word.lower() not in stopwords]
if len(words) == 0:
  print(f"No words found for author {l[i]} after removing stop words. Skipping word cloud generation.")
  continue  # Skip to next author if no words are found

wordcloud = WordCloud(stopwords=stopwords, background_color="white").generate(" ".join(words))
# Display the generated image
plt.figure( figsize=(10,5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

```
Author name Areej Clg
No words found for author Areej Clg after removing stop words. Skipping word cloud generation.
Author name Preeti Clg Mumbai
No words found for author Preeti Clg Mumbai after removing stop words. Skipping word cloud generation.
```