```
pip install emoji
```

```
Requirement already satisfied: emoji in /usr/local/lib/python3.10/dist-packages (2.14.0)
```

```
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

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/chart45.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(100)) # Corrected the typo from 'taiCl' to 'tail'
print(df.info())
print(df.Author.unique())
```

```
           Date      Time           Author  \
3437 2022-05-04  12:55 pm  +91 96199 36420
3438 2022-05-04  12:55 pm  +91 96199 36420
3439 2022-05-04  12:56 pm  +91 90042 80656
3440 2022-05-04  12:56 pm  Monika Kharkwal
3441 2022-05-04  12:56 pm  +91 90042 80656
...         ...       ...              ...
3532 2024-04-27   8:26 am  +91 70214 89118
3533 2024-04-28   5:14 pm  +91 99676 09749
3534 2024-04-28   7:40 pm  +91 70214 89118
3535 2024-04-29   8:17 am             None
3536 2024-04-29   8:17 am  +91 97683 13673

                                       Message
3437                            Home security....
3438                       Change nahi karungaa 😅 abhe
3439                                  Bacha kya h
3440                           Tumhara project hna
3441                              Topics batao toh
...                                            ...
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 ...

[100 rows x 4 columns]
<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' 'Pratiksha Awate' '+91 97683 13673' '+91 99672 73815'
 '+91 70214 89118' '+91 97695 24164' '+91 90040 75303' 'Preeti Clg Mumbai'
 'Shraddha Panchal' '+91 90042 80656' '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-29-ddba0f74192b>:2: UserWarning:

Could not infer format, so each element will be parsed individually, falling back to `dateutil`. To ensure parsing is consistent and
```

```
df.head()
```

	Date	Time	Author	Message
0	2021-09-10	9:32 am	None	Areej Clg created group "notes~"
1	2021-10-18	9:07 am	None	You joined using this group's invite link
2	2021-10-18	10:02 am	Areej Clg	Ohh
3	2021-10-18	10:02 am	None	Areej Clg added +91 99676 09749
4	2021-10-18	10:57 am	Areej Clg	<Media omitted>

Next steps: Generate code with df | View recommended plots | New interactive sheet

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

```
3537
```

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

```
314
```

```
import emoji
import regex
```

```
import pandas as pd

def split_count(text):
    emoji_list = []
    data = regex.findall(r'\X',text)
    for word in data:
        # Use emoji.is_emoji() to check for emojis
        if any(emoji.is_emoji(char) 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)
```

```
789
```

```
def split_count(text):
    emoji_list = []
    data = regex.findall(r'\X', text)
    for word in data:
        # Use emoji.is_emoji() instead of UNICODE_EMOJI
        if any(emoji.is_emoji(char) 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)
```

```
789
```

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

```
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.
```

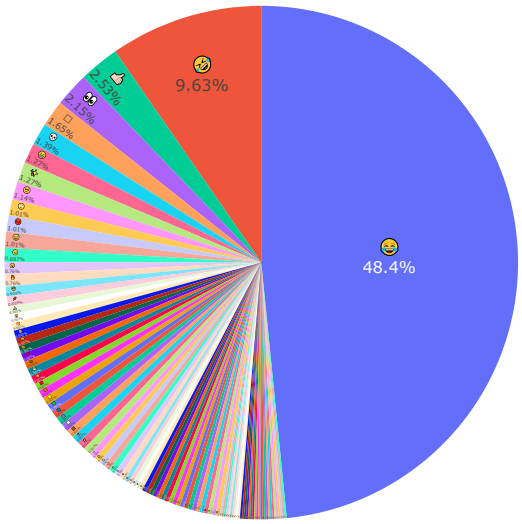
```
total_emojis_list = list(set([a for b in messages_df.emoji for a in b]))
total_emojis = len(total_emojis_list)

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()
```

```
('🤣', 76)
('👍', 20)
('👀', 17)
('', 13)
('💀', 11)
('😁', 10)
('✨', 10)
('🥺', 9)
('😊', 8)
('😡', 8)
('😅', 8)
('😒', 7)
('😭', 6)
('🔥', 6)
('😎', 5)
('📌', 5)
('🙏', 4)
('😆', 4)
('🙂', 4)
('😐', 4)
('🤲', 4)
('🤦\u200d🚓', 4)
('❤️', 4)
('🐲', 4)
('😵', 4)
('😑', 4)
('🟥', 4)
('🟧', 4)
('🟨', 4)
('🟩', 4)
('🟦', 4)
('🟪', 4)
('⬜', 4)
('🟫', 4)
('🚺', 4)
('💥', 4)
('▫', 4)
('✌', 3)
('🤷\u200d♀', 3)
('🙃', 3)
('😉', 3)
('☝', 3)
('👇', 3)
('🔶', 3)
('🤷', 2)
('🙂', 2)
('🤹\u200d♂', 2)
('❤️', 2)
('🙂', 2)
('😔', 2)
('🤦\u200d♂', 2)
('🐍', 2)
('🐛', 2)
('😶', 2)
('🚨', 2)
('😯', 2)
('♥', 2)
('👍', 2)
('😑', 2)
('♡', 2)
('💻', 2)
('📍', 2)
('🏷', 2)
('📆', 2)
('⏱', 2)
('📱', 2)
('🔰', 2)
('📄', 2)
('💿', 2)
('😄', 1)
('😩', 1)
('😇', 1)
('🤔', 1)
('😅', 1)
('😉', 1)
('😘', 1)
('😔', 1)
('', 1)
('🤷\u200d♂', 1)
('🤪', 1)
('😲', 1)
('🤹\u200d♀', 1)
('😢', 1)
('✌', 1)
('🥺', 1)
('😱', 1)
```

```
('😉', 1)
('🏫', 1)
('📝', 1)
('🆓', 1)
('📜', 1)
('💰', 1)
('🌐', 1)
('🙏', 1)
('☝', 1)
('🤟', 1)
('😅', 1)
('😔', 1)
('👆', 1)
```




```
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())
```

```
     3525 2023-04-23   2:04 pm  +91 90042 80656
     3526 2023-04-23   2:04 pm             None
     3527 2023-04-25   7:36 am             None
     3528 2023-04-25   8:38 am             None
     3529 2023-04-25  11:00 am  +91 97683 13673
     3530 2024-04-25  11:01 am  +91 97683 13673
     3531 2024-04-25  11:06 am  +91 97683 13673
     3532 2024-04-27   8:26 am  +91 70214 89118
     3533 2024-04-28   5:14 pm  +91 99676 09749
     3534 2024-04-28   7:40 pm  +91 70214 89118
     3535 2024-04-29   8:17 am             None
     3536 2024-04-29   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
```

```
 #   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' 'Pratiksha Awate' '+91 97683 13673' '+91 99672 73815'
 '+91 70214 89118' '+91 97695 24164' '+91 90040 75303' 'Preeti Clg Mumbai'
 'Shraddha Panchal' '+91 90042 80656' '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-38-66e13cebf647>:2: UserWarning:

Could not infer format, so each element will be parsed individually, falling back to `dateutil`. To ensure parsing is consistent
```

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

```
3537
```

```
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 161033 words in all the messages.
```


```
messages_df = df  # Assign the dataframe 'df' to 'messages_df' to make the 'Author' column available
l = ["Areej Clg", "Pratiksha Awate"]
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))
```