📱💬 WhatsApp Chat Analysis 📊🧠

Joyashree Mandal

03/05/2025

Dive into the **insights of your WhatsApp group conversations** using data visualization techniques like **Word Clouds**. This is my self driven project. This analysis helps uncover patterns, frequent topics, and the chatting behavior of each participant—**turning raw chat logs into meaningful visuals**!

For full project Github Link:

https://github.com/joyashre/Whatsapp-chat-analysis

★ 💬 Data Extraction from Chats

If you don’t know how to extract messages from a chat, follow these steps:

1. 📂 Open the chat and click on the **three dots (⋮)** in the top-right corner.
2. 🔍 Select **More**, then choose **Explore Chat**.
3. 📤 Share the chat using any method (email, drive, etc.).

★ 📦 Importing Libraries

```
import regex
import pandas as pd
import numpy as np
import emoji
import plotly.express as px
from collections import Counter
import matplotlib.pyplot as plt
from os import path
from PIL import Image
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
```

🛠️ If Some Libraries Are Missing, Install Them Using:

```
pip install emoji
pip install plotly
pip install wordcloud
```

★ 📚 Library Descriptions

🔤 ***regex*** *: A RegEx, or Regular Expression, is a sequence of characters that forms a search pattern. RegEx can be used to check if a string contains the specified search pattern.*

📊 ***plotly :*** *Plotly's Python graphing library makes interactive, publication-quality graphs. It supports a wide range of chart types, including line plots, scatter plots, bar charts, histograms, and 3D graphs.*

☁️ ***wordcloud :*** *A word cloud, also known as a tag cloud, is a visual representation of text data where the size of each word indicates its frequency or importance. In python, the wordcloud library facilitates the creation of these visualizations.*

📁 ***Os :*** *Python has a built-in os module with methods for interacting with the operating system, like creating files and directories, management of files and directories, input, output, environment variables, process management, etc.*

😂 ***emoji****: Helps detect and extract emojis from text data.*

🧮 ***collections.Counter****: Quickly count frequencies of elements (like words or emojis).*

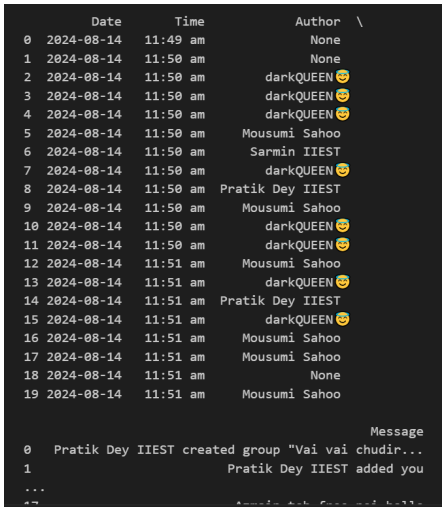
🖼️ ***PIL (Pillow)****: Used for image processing tasks such as loading images for word clouds.*

★ 💬 Chat Analysis:

we need to change the format of the date and time of messages. For this I will define a function that can detect whether each line starts with a date as it states that it is a unique message:

```
import re

def StartWithDateAndTime(s):
    pattern =
r'^\d{2}/\d{2}/\d{4},
\d{1,2}:\d{2}\u202f?(?:am|pm|AM|PM
)? -'
    result = re.match(pattern, s)
    if result:
        return True
    return False
```

- ➔ Now create a function to extract the usernames in the chats as Authors
- ➔ And creating a function to separate all the information from each other so that we could easily use the information as a pandas dataframe.
- ➔ Now if we see the data we find it like this



	Date	Time	Author	Message
0	2024-08-14	11:49 am	None	Pratik Dey IIEST created group "Vai vai chudir...
1	2024-08-14	11:50 am	None	Pratik Dey IIEST added you
2	2024-08-14	11:50 am	darkQUEEN😇	
3	2024-08-14	11:50 am	darkQUEEN😇	
4	2024-08-14	11:50 am	darkQUEEN😇	
5	2024-08-14	11:50 am	Mousumi Sahoo	
6	2024-08-14	11:50 am	Sarmin IIEST	
7	2024-08-14	11:50 am	darkQUEEN😇	
8	2024-08-14	11:50 am	Pratik Dey IIEST	
9	2024-08-14	11:50 am	Mousumi Sahoo	
10	2024-08-14	11:50 am	darkQUEEN😇	
11	2024-08-14	11:50 am	darkQUEEN😇	
12	2024-08-14	11:51 am	Mousumi Sahoo	
13	2024-08-14	11:51 am	darkQUEEN😇	
14	2024-08-14	11:51 am	Pratik Dey IIEST	
15	2024-08-14	11:51 am	darkQUEEN😇	
16	2024-08-14	11:51 am	Mousumi Sahoo	
17	2024-08-14	11:51 am	Mousumi Sahoo	
18	2024-08-14	11:51 am	None	
19	2024-08-14	11:51 am	Mousumi Sahoo	

➔ To get all the authors:

```
data.Author.unique()
```

output:

```
array([None, 'darkQUEEN😇',
'Mousumi Sahoo', 'Sarmin
IIEST',
       'Pratik Dey IIEST',
'Biswas'], dtype=object)
```

➔ To get media message:

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

```
304
```

Here I found 304 media messages.

➔ And also

```
Messages: 1233
Media: 304
Emojis: 475
Links: 11
```

➔ Now for author wise status

```
Stats of darkQUEEN😇 -
Messages Sent 288
Words per message 4.309027777777778
Media Messages Sent 185
Emojis Sent 119
Links Sent 0

Stats of Mousumi Sahoo -
Messages Sent 235
Words per message 2.821276595744681
Media Messages Sent 14
Emojis Sent 156
Links Sent 0

Stats of Sarmin IIEST -
Messages Sent 65
Words per message 3.076923076923077
Media Messages Sent 14
Emojis Sent 47
Links Sent 0
```

```
Stats of Pratik Dey IIEST -
Messages Sent 311
Words per message 3.4790996784565915
Media Messages Sent 78
Emojis Sent 144
Links Sent 0

Stats of Biswas -
Messages Sent 12
Words per message 3.1666666666666665
Media Messages Sent 13
Emojis Sent 1
Links Sent 0
```

➔ And for emoji:

```
total_emojis_list = list([a for b in messages_data.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)
```

[98]

```
('🤣', 97)
('😒', 48)
('😂', 37)
('🤩', 27)
('😩', 21)
('🥳', 17)
('😁', 16)
('🤭', 12)
('😑', 12)
('😇', 11)
('😉', 9)
('🤳', 9)
('🙂', 8)
('🖼', 7)
('🖼', 7)
('💃', 7)
('🥰', 7)
('😍', 7)
('🖼', 6)
('😐', 6)
```

★ 💬 WhatsApp Group Chat Analysis: Word Cloud ☁️

A **Word Cloud** is a visual representation of text data where the **most frequently used words** appear **larger and bolder**. It's a great way to quickly grasp what topics or phrases are commonly discussed in a chat.

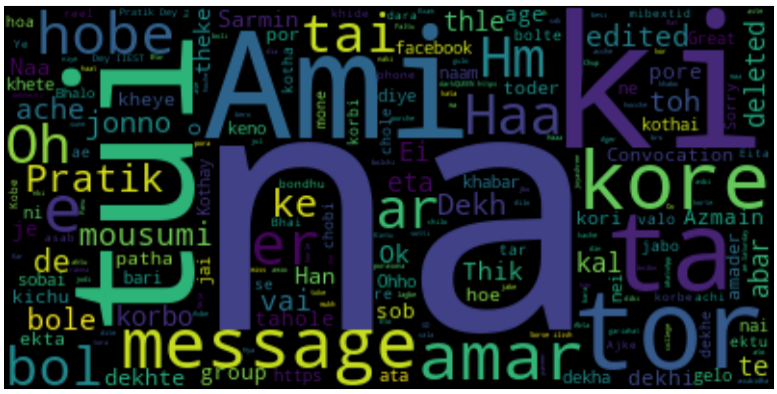
- 📌 **Group-Level Word Cloud**

Let's first generate a word cloud for the **entire group chat**:

```
text = " ".join(review for review in messages_data.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="black").generate(text)

# Display the generated image:
# the matplotlib way:
plt.figure( figsize=(10,5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

output:

🧮 There are 18244 words in all the messages.



- 👤 **Author-wise Word Cloud**

Now, let's take a look at the word usage **per individual** in the group:

```
l = ['darkQUEEN😇', 'Mousumi Sahoo', 'Sarmin IIEST', 'Pratik Dey IIEST',
'Biswas']
for i in range(len(l)):
  dummy_df = messages_data[messages_data['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])
  wordcloud = WordCloud(stopwords=stopwords,
background_color="white").generate(text)

  #Display the generated image
  plt.figure( figsize=(10,5))
  plt.imshow(wordcloud, interpolation='bilinear')
  plt.axis("off")
  plt.show()
```

output: