

Reading the data

Operations Done

- * Reading from txt file
- * Splitting it to rows using regex
- * convert to DataFrame

```
In [1]: import re
import pandas as pd

# Specify the path to your text file containing chat data
file_path = '_chat.txt'

# Initialize empty lists to store components
dates = []
times = []
usernames = []
messages = []

# Read data from the text file
with open(file_path, 'r', encoding='utf-8') as file:
    chat_data = file.readlines()

# Extract components and store in lists
for line in chat_data:
    date_match = re.search(r'^\[([.*?])\]', line)
    if date_match:
        date_time_str = date_match.group(1)
        datetime_parts = re.split(r'[,\s]+' , date_time_str)
        if len(datetime_parts) == 3:
            date, time, am_pm = datetime_parts
            dates.append(date)
            times.append(time + " " + am_pm)
        else:
            dates.append(None)
            times.append(None)

    match = re.search(r'\]\s([.*?]):\s', line)
    if match:
        username = match.group(1)
        usernames.append(username)
        message = re.sub(r'^\[([.*?])\]', '', line).strip()
        messages.append(message)
    else:
        usernames.append(None)
        messages.append(None)
    else:
        dates.append(None)
        times.append(None)
        usernames.append(None)
        messages.append(None)
```

```
# Create a pandas DataFrame
data = {
    'Date': dates,
    'Time': times,
    'Username/Number': usernames,
    'Message': messages
}

df = pd.DataFrame(data)

# Display the DataFrame
print(df)
```

	Date	Time	Username/Number
0	28/04/23	11:02:30 AM	CodeSapiens - Code 'Seivom' \
1	28/04/23	11:02:30 AM	Product Manager
2	28/04/23	11:03:11 AM	CodeSapiens - Code 'Seivom'
3	28/04/23	11:06:18 AM	CodeSapiens - Code 'Seivom'
4	28/04/23	11:08:21 AM	CodeSapiens - Code 'Seivom'
...
10023	01/10/23	1:32:54 PM	+91 99441 40269
10024	01/10/23	1:57:53 PM	~ Mr.Coder
10025	01/10/23	1:58:48 PM	~ Mr.Coder
10026	01/10/23	1:59:34 PM	~ Mr.Coder
10027	01/10/23	2:19:39 PM	+91 93610 85806

	Message
0	CodeSapiens - Code 'Seivom': Messages and cal...
1	Product Manager: Product Manager created this...
2	CodeSapiens - Code 'Seivom': Product Manager ...
3	CodeSapiens - Code 'Seivom': You changed this...
4	CodeSapiens - Code 'Seivom': You changed the ...
...	...
10023	+91 99441 40269: +91 99441 40269 left
10024	~ Mr.Coder: enakum athala pudikathu sir..so pr...
10025	~ Mr.Coder: kandipaaga sir
10026	~ Mr.Coder: fine sir
10027	+91 93610 85806: +91 93610 85806 joined u...

[10028 rows x 4 columns]

Basic EDA

```
In [2]: import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [3]: ## Checking the shape of the DataFrame
df.shape
```

Out[3]: (10028, 4)

```
In [4]: ## Displaying the basic info about the DataFrame
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10028 entries, 0 to 10027
Data columns (total 4 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   Date                   5763 non-null   object
1   Time                   5763 non-null   object
2   Username/Number        5763 non-null   object
3   Message                5763 non-null   object
dtypes: object(4)
memory usage: 313.5+ KB
```

```
In [5]: ## Displaying Basic statistical Description about the DataFrame
df.describe(include='all')
```

```
Out[5]:
```

	Date	Time	Username/Number	Message
count	5763	5763	5763	5763
unique	93	4802	1007	5514
top	03/09/23	7:07:44 AM	ThiyagaB	ThiyagaB: POLL:
freq	341	11	1218	20

```
In [6]: ## Displaying the head of the DF
df.head()
```

```
Out[6]:
```

	Date	Time	Username/Number	Message
0	28/04/23	11:02:30 AM	CodeSapiens - Code 'Seivom'	CodeSapiens - Code 'Seivom': Messages and cal...
1	28/04/23	11:02:30 AM	Product Manager	Product Manager: Product Manager created this...
2	28/04/23	11:03:11 AM	CodeSapiens - Code 'Seivom'	CodeSapiens - Code 'Seivom': Product Manager ...
3	28/04/23	11:06:18 AM	CodeSapiens - Code 'Seivom'	CodeSapiens - Code 'Seivom': You changed this...
4	28/04/23	11:08:21 AM	CodeSapiens - Code 'Seivom'	CodeSapiens - Code 'Seivom': You changed the ...

```
In [7]: ## Printing number of unique values present in each column
df.nunique()
```

```
Out[7]: Date                93
Time                4802
Username/Number      1007
Message              5514
dtype: int64
```

```
In [8]: ## Getting the unique values to check for values other than dates
df['Date'].unique()
```

```
Out[8]: array(['28/04/23', '05/05/23', None, '06/05/23', '08/05/23', '09/05/23',
              '10/05/23', '11/05/23', '13/05/23', '14/05/23', '18/05/23',
              '20/05/23', '21/05/23', '23/05/23', '26/05/23', '28/05/23',
              '29/05/23', '01/06/23', '03/06/23', '04/06/23', '05/06/23',
              '06/06/23', '07/06/23', '09/06/23', '10/06/23', '11/06/23',
              '12/06/23', '14/06/23', '16/06/23', '17/06/23', '23/06/23',
              '24/06/23', '08/07/23', '01/08/23', '02/08/23', '04/08/23',
              '05/08/23', '06/08/23', '07/08/23', '08/08/23', '09/08/23',
              '10/08/23', '11/08/23', '12/08/23', '13/08/23', '14/08/23',
              '15/08/23', '16/08/23', '17/08/23', '18/08/23', '19/08/23',
              '20/08/23', '21/08/23', '22/08/23', '23/08/23', '24/08/23',
              '25/08/23', '26/08/23', '27/08/23', '28/08/23', '29/08/23',
              '30/08/23', '31/08/23', '01/09/23', '02/09/23', '03/09/23',
              '04/09/23', '05/09/23', '06/09/23', '07/09/23', '08/09/23',
              '09/09/23', '10/09/23', '11/09/23', '12/09/23', '13/09/23',
              '14/09/23', '15/09/23', '16/09/23', '17/09/23', '18/09/23',
              '19/09/23', '20/09/23', '21/09/23', '22/09/23', '23/09/23',
              '24/09/23', '25/09/23', '26/09/23', '27/09/23', '28/09/23',
              '29/09/23', '30/09/23', '01/10/23'], dtype=object)
```

```
In [9]: ## Converting the column from "Object" to "DateTime"
df['Date'] = pd.to_datetime(df['Date'], format='%d/%m/%y')
```

```
In [10]: ## Checking if changes have made to DF using info()
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10028 entries, 0 to 10027
Data columns (total 4 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   Date                  5763 non-null   datetime64[ns]
1   Time                  5763 non-null   object
2   Username/Number       5763 non-null   object
3   Message               5763 non-null   object
dtypes: datetime64[ns](1), object(3)
memory usage: 313.5+ KB
```

Gathering Info about Users

```
In [11]: # Get unique values in the 'Username/Number' column
unique_values = df['Username/Number'].unique()

# Print the unique values
for value in unique_values:
    print(value)
```

CodeSapiens - Code 'Seivom'

Product Manager

+91 86109 86964

~ Harisangar A P

~ 👑

~ Tryphosa Evangeline

~ Shandhini Mohankumar

~ Sri 💖

~ S Divya Darsshini ✨😊

~ .^.

~ *Poorna*....

~ Monzzz 🍷

~ (◡‿◡)

~ ~ Bhuva ❤️

~ Sowmiya Saravanan

~ Harish.EA

+91 90257 16650

~ SIVASARAN

~ Sanjay Saravanan

~ .

~ Keerthana ✨👉

~ Sanjena. G

~ Madhubaalika

~ Prashant.SS

~ swetha

~ Pooja 💖 ✨

~ KAVS ✨

~ ~mathi ✨

~ Loki Skywalker

ThiyagaB

~ Arun

~ P M Kiruthiga

~ Sivanipriya 🍀

~ karthick

~ Keerthi💖💖

~ M. Ilampirai

~ Karan

~ Lingeshwaran.V

~ Abishek 💖💖

~ Karthick

~ Vignesh

~ 🌸🌸mohan🌸🌸,

~ VINOTH

None

~ Karthi

~ ○○○○..❤️..○○○○

~ ✨

+91 97150 56251

~ Naren

Ganesh SG INDIA

+91 96593 39945

~ Nisha ❤️

~ Swetha

+91 74486 61633

~ ^_^

~ Sharu

Vinith Walmart
Koushik PEC
~ ...
+65 8819 8327
~ Srini
~ Adithya♥
~ ROHITH B 🏆♥
~ SK
~ S.Vairaperumal
~ Shakith. A
~ Shunmathi
~ yuvapriya245
~ Yukta
~ Dukatu
~ Nanda Kumaran
~ Lokesh :)
~ Sowmya😁♥
~ Pavithra
~ Mathumitha Mathiyalagan
~ Ahalya VS
~ Nathin kishore T
~ Rei Kiriyama
~ 😊
~ Jaasim Hameed
~ Vishruthi
~ Mohamed Imran
~ ｲﾝﾓﾛﾓﾓ
~ Elumalai PRO
~ NithishaKumar😁😁
~ Sangeetha Sambath
~ Balaji
~ Sanjay Ganesh
~ 🤔
~ Sourish Cidambaram
~ selva 🌸
+91 76048 65800
~ Sonali💎
~ kowsiks
~ Sriram
~ Shreyas Ramanathan
~ 🏆👉
+91 82481 22029
~ Varun Raghav
~ sharada
~ SOMASUNDARAM K
~ Gurumoorthi😊
~ Prani
~ shriram
~ Deepan.B
~ Thyagu
~ rajesh
~ vamsi
~ Deepak Sridhar
~ Surendiran💎
~ Ramyaa AP
~ Keerthanaa♥

~ ~Sowmi 🐾🐾
~ Sulochana
~ Sushmitha
~ Subash Lakshmanan
~ Harini Natarajan
~ Swasti__ka
~ Aadhitya sriram
~ Karthik Krishna
~ Vishal
~ Me
~ Mr.SPEED... 🤖 😊
~ R.MUTHUKRISHNANGOMATHI
~ Samyuktha
~ Arun S
~ Gk
~ SARUMATHI
~ 🤖
~ Sree
~ Sowmitha Moorthy
~ Neeraj
~ Adithiyan Murali
~ Keshav
~ srini
~ Deva Abinaya
~ Sree Ram.T.R
~ Revathi
~ ~TheDiablo 🤖~
~ Salai Kowshikan
~ ✨Abhinav ✨
~ Santhosh
~ Swetha Barade
+91 73056 00331
~ 🤖
~ Anagha
~ Akshaya
~ Mariam Bobby
~ Raghunath
~ Rahul Balaji
~ Yuva 🤖 🤖
~ Vishal. P
~ Harshini
~ Suriya Prakash 🤖
~ Arun Karthick Saravanan
~ Ezhil Dhiraviya
~ Mowni 🤖
~ Akash K
~ A7WiN ✨
~ D E I V A
~ Mathan Kumar
~ Haridass
~ RaZee... 🤖
~ Suren Kumar
~ 🤖 Rāgūl 🤖
~ Shakthidharan
~ Dharshan Sidharth
~ Sxnjay XD

~ Krishna
~ Harini N
~ SRI DHARANI
~ Subha ;)
~ KisHoRE
~ Hardik Pandya
+91 97890 54094
~ SS
~ Dharani Sengottuvelu
~ Deepak Chandhru
~ Sahana Sara 🐼
~ Nivedha Madhavan
~ JDNaveen
~ Gadiraju Dinesh 🧑
~ Shewak
~ Kailash
~ Vasundaraa
SHIVABALAN MANI 🐼
~ Jayasri
+91 6379 697 899
~ S❀H❀N❀
~ Yaamini
~ Subhash Krishnasamy
+91 93455 41653
+91 74183 70961
+91 79042 97172
+971 52 998 0148
~ Jack💖
+91 86100 86885
~ Ishwarya
~ Sakthi
~ Lakshmi Praba
~ sanakkian
~ HEMACHANDAR K
~ Balasurya
+91 97917 98238
~ Roshini🐼
+91 98846 24057
~ Usha
~ PRAJIN
~ Punitha
~ Chai Ganesh J
~ Nishanth TR
+91 73586 93653
~ Madhan Sai
~
+91 6374 109 779
~ Shreya Karthikeyan
~ Mohit R
~ ♥Nitin-Nithya♥
+91 93456 61373
~ Entrepreneur ♥💎
~ Rajesh
~ Jothi
~ madhavanlogith@gmail.com
~ KiranKumar

~ nagappans
+91 99406 17737
+91 6381 086 085
~ 🍷
~ Prabakaran
~ Raju
+91 93449 44089
~ Aashath💕
~ ✨AJ
~ prasanth
+91 88700 79990
~ Kisssorre
~ Prani18🤔
+91 88254 80770
+91 88384 56818
~ Manoj
+91 98435 08164
~
+91 98946 47686
~ ✨💕Madhan💕✨
+91 70923 03309
~ KATHIRESAN T
~ Gajendra Babu
~ 🐱భూపతి🐱
+91 74185 51375
~ CR7 GAMING YT - தமிழ்
+91 75502 80105
~ Ram
~ Allen
~ Aakaxh
+91 73050 63094
~ **Ravindar**
~ Thirumurugan A
+91 98412 04362
+91 6379 432 017
~ TheRIKE- லி லி
+91 81226 62347
+91 93843 50861
~ Ak Aadhi
+91 6383 517 170
+91 93635 45844
+91 83448 19309
+91 6382 010 323
~ G.Manoj
~ Betsee Natasha
~ Gayu💕
~ Chamini
+91 81247 07525
~ **Harshini**
+91 90435 87515
~ M
+91 93425 63753
~ Aishu❄️
+91 87781 34325
+91 72001 12388
~ Sangeetha

+91 95974 87780
+91 99622 67669
+91 98840 06155
~ Dharmarajan
~ Mano Ranjitham ❤️
+91 6382 580 602
~ Jagan
~ Divya
~ unuV
+91 76959 91511
+91 6369 296 151
~ SAN
+91 98439 40535
+91 97102 71233
+91 80724 05648
+91 94446 40117
+91 99401 79612
+91 6382 650 792
~ 💧 Pradeep Krishna 💧
~ Hari
~ Dhileep
~ ..
~ ❤️ GK ❤️
~ ΔRI 💎
~ ❤️ Medha.M.J ❤️
~ !!!!
~ K.M.KUMARAVEL
~ bhuvanesh ramesh
~ ilai....
~ Rakshaka ❤️
~ **GOPAL**
~ Aswin T
~ Bharathi
~ Sridharan ✨
+91 95000 63013
~ Harshu
~ ❄️
+91 93844 64558
~ Sri Rama Pandian H U 😊😎😊😊
+91 81227 74857
+91 90922 34285
+91 84382 05064
+91 86673 04549
~ NithiYogi
+91 70104 54417
~ Selva Ganesh S
~ Hariguru
~ Visvess😊
~ Dhinagar
~ **Sairam** ✨
~ Dhiv❤️
~ Aishu❤️
~ R.S .Moses Iraianbu
~ Raghul V
~ Balamurugan S
~ Asmath Akbar Ali

~ 😊PETER SAM😊
+91 81488 72716
~ Fadhil
+91 93618 50445
~ Rajavel
~ Lakshman Kumar
~ Nivedhashree
+91 73580 05636
Vasanth Ravichandran
~ Anukraha ❤️
~ **Vaseekaran M**
+91 87783 81543
~ 🌀
~ BARATH
+91 87543 64622
~ Jagadheeswari
~ Nikaash Arulkon
~ Kowshalya..
~ srivatsan
~ Stephen ❤️
~ Logesh
~ Subi Pinsha
~ Mahima
~ 💎❤️Dhana❤️💎
~ KK
~ Muruganantham
~ MUGIL
+91 89256 52309
~ Dinesh
~ ~ Sanjith Shylesh
~ 자얀스 🍀💎
~ AchuMaheswaran....~❤️
~ VJ
~ Soumya Renganathen
~ Purushothaman Srinivasan
~ Kishore
+91 99443 70355
~ MAJOR.B
~ Hari😊
~ Aakash K K
~ Razeen
~ Abinaya S M
~ GOPINATH V
~ thaladev
+91 80568 90678
~ 😊
~ Asvajith
+91 90806 68298
~ Muthurama A
~ Sridhishnee
~ ~srivaigunth~
~ Tejashwini Satash
~ Dharshuu
+91 82472 21689
~ BALAJI S 😊
+91 76674 76679

+91 95144 97762
~ Sowmya A
~ 🐼 Sree 🐼
~ Shankar Sai
~ Vaishu
~ _its_me_bala__
+91 70100 49265
~ Abinaya
+91 93420 32287
~ GuberakannaN ThiyagarajaN
~ Hema
+91 6382 558 669
~ Ashraf Basha
+91 81225 94059
+91 6382 058 580
~ Kavin Bharathi
~ 💎Nithish Kumar💎
+91 6374 938 073
+91 86673 29935
+91 90432 53617
~ Shamim Ahamed
~ **Balzz_x** 🌐 🌐
~ Pranesh
+91 76958 66009
~ Tony
+91 98659 57555
+91 83005 81670
~ ASWIN KUMAR
+91 90873 09810
~ Nithish
~ Mahaveer@
+91 93446 01094
~ Manoj S
~ Vishakan N
~ Hashir Yousuf
~ Rishi Kumar.U
~ Yash
+91 90478 99465
~ Navin
~ Car Addict Saravanan
~ Pharaoh Suriya
~ Praveen
~ Sanjay Kumar
+91 6301 241 409
+91 99521 72138
~ Vishnupriya Shanmugam
~ Mirudula
~ N G K
~ Nithish .k 😊 😊
~ Sandy 💎
~ miracle 🙄 😊
~ Karthica
~ Yashwant
+91 99417 94229
~ 📱
~ KAVIYA G

+91 82204 66177
~ Lakshay
~ Veni
+91 6382 332 554
~ Joker
~ FIDA
~ Subhasri Muthusamy
~ A
~ Sisters Love☐💞💞💞...
+91 88380 24203
+91 97868 14340
+91 94457 44690
~ Prasitha🐼
~ JOTHIRUBAN M
+91 73059 89697
~ C² SANTHOSH
~ Harini🌹
~ 🌸🌸🌸Purna🌸🌸🌸v🌸🌸vikraman🌸🌸
~ krishna♥
+91 88704 73893
+91 80562 74738
+91 90478 73653
~ Pulipati Surya
+91 96594 02034
~ ITACHI
~ Dhayalan 😊😊😊😊
~ jaya_raja
~ °ZERO
~ Abishek.N
~ Ajay 😊😊
+91 99440 98407
~ INBA GANESH
~ S.Dhanush Kumar
~ تنزيل محمد ♥
~ Aravindan P B
~ abisheks
~ Mohith
~ THANUSH
+91 88072 86899
+91 93470 97459
+91 98481 73257
+91 88009 27972
~ 彡★ S.Adhithya ★彡
+91 98849 76665
~ Haemowaek
+91 72004 80691
~ Rohith
~ Jeevi *°-°*
~ Aswanth Babu
+91 98404 14145
+91 99621 11461
~ skandaalogistics
~ Sujii ✨
~ Monish
~ VESHVAA R A 12-A
~ Sowmya

~ Harikumar G
~ Idris
~ Jayasuriya 🤖
~ ÇmarE_ñareşh_mañi
~ Mr.Coder
+91 99410 10111
~ bala💖
~ Jagdish
+91 96776 68258
~ Deepak B
~ Narendran.G
~ Ram💖
~ Rithish
+91 6383 449 247
~ Kalyaan Mahendar
~ 💖
+91 91 77047 301
~ Chokkalingam 💖
~ Densingh
~ Aakash
~ AJAY J
~ Diwakar S
~ DIVAKAR 😊
~ Riya:)
~ ✨ M S SUGUMAR ✨
~
~ Abhi
~ VIGNESHWARAN
~ MANOJ
~ Mouni💖😊
~ _vicky...
~ Nishanth
+91 6374 142 765
~ Jp
~ ぞ榮現
~ HARINI PREMKUMAR 🤖🤖
~ Krithiga Jayalakshmy
~ Bhuvan
~ Barath
~ Hari Hara Sudhan J
~ AK ✨
~ ROHIT VIJAYAN
~ ...
~ Naveen💎
~ Shahana
~ Teja
~ RadhikaGaneshkumar
~ Rupesh
~ Divi
~ Darunesh/Sathya
~ Likesh Abhimanyu
~ Kanagavel ✨
~ Rubakrishna°
~ Sheegan Sri 🤖
~ Niktha
~

+91 78068 22194
~ R.Jayanth Raj
~ Sachin
~ JOHN YUGANDHAR
~ 🙌 GOKULRAJAN.R.M 🙌
~ ❤️🏠SUNIL. 🏠❤️
+91 6383 985 664
~ Keerthana 🥰
~ Tinku Abenas
~ Nithesh
~ Sandeep Krishna M
+91 78128 79331
~ Rahul
+91 93613 21901
~ PVVishal
~ A.Keerthi Vasan
+91 98411 97144
~ harinisrinivasan20059
+91 90950 13284
~ RithikRaja
+91 93604 23050
+91 82207 47701
+91 93441 32869
+91 86086 92727
+91 6380 642 787
+91 88387 10602
~ Justin
+91 93635 72386
+91 6382 995 429
+91 79049 11880
+91 72991 77613
+91 97898 71907
~ KAAVIYA K
~ 🍷🌹
~ Dhanushreddy
~ Dharaneesh
~ Varshaa
~ Namrutha ❤️
~ Gopi
~ Lavanya.I
~ Vishal ❤️
~ Ashwin Balaaji Jagadish
+91 83000 69029
~ Ajis Roshan
~ Abhishek Deenan
+91 80721 11140
+91 86676 57754
+91 73389 63256
+91 98402 45291
+91 80569 25409
~ Ram Prasath
~ 🐜
~ Ponvizhi
~ Vishnu Siva
+91 90031 97480
+91 93632 23878

+91 84280 18381
+91 90435 09447
~ Malavika
+91 93608 92557
+91 6374 241 280
~ Abishek
+91 94877 69772
~ Yathu
+91 85208 42026
+91 93427 17121
~ Alwin.A
~ BASKARAN . V
~ ❤️100
~ U.S.Prasath
+91 97911 27279
~ ~lalithkumar
~ DINESH..👉
~ Surya Narayanan
~ Harish Depur
~ shanthy
+91 84899 50717
~ B|ithesome..° ✨
+91 95000 21804
+91 79049 05742
+91 96771 12187
+91 83102 39325
+91 98403 49253
+91 73050 07698
+91 73584 80747
~ Sharmada
~ Meganathan
~ Kavindar
~ KISHAN JAI SOORYA N
+91 73387 98112
+91 6382 859 823
+91 97907 99570
~ Prasanth
+91 80151 84882
~ Angel Antony 😊
+91 74492 50726
~ Sam👉
~ Dharshy
~ Sreecharan
~ Hac2how
~ ❤️Partha ❤️
~ Shashwanth
~ S Sanjay
~ M.A.Tarun
~ Subha
~ Santhosh👉
~ Hemanthkumar
~ Aravindharaj
~ 🍷 KARUN ROHITH 🍷
~ Rahul.M.J
~ Yugan Sakthi
+91 88389 31951

~ Sam
~ Vishnu
+91 91 76559 155
+91 96002 92350
~ sam
~ S H Y A M
~ Mithra Shankar
+91 90439 10228
~ SWATHI 🐼
+91 90259 86953
+91 97899 29148
~ Tharun Karthick T
+91 76395 36789
~ ROHITH
~ Babu R
+91 84382 18687
~ Ranjani ✧
~ Rithik
+91 96005 72932
~ *abz*
+91 94432 49530
~ ShankariDeenadayalan 🐱
~ Ushaasree
+91 96290 96790
~ Glee
+91 94874 54745
+91 74183 56711
~ Kanishh ✧ !!
~ Hari Haran
~ SHINCHAN
+91 97511 59410
+91 76048 30742
~
+91 80561 95875
~ ~Akash.P
+91 88073 31644
+91 80159 62208
~ ~Hari~ ❤️
~ **Benfadesan.f**
+91 99628 02851
+91 70924 17471
~ LOGAN SW
~ Ameeshab 🐼 ❤️
+91 96265 20880
+91 79047 06408
+91 99620 97227
~ Vickie 🐼 🐱
+91 79044 69214
+91 98401 10071
~ D
~ ★Abishek★
~ GIRIDHARAN.S
+91 95973 74955
+91 97875 74976
~ Aathif PM
+91 6374 433 839

~ Aadhitya
~ bharathi suresh
+91 98943 14946
+91 72001 23883
+91 73052 91852
+91 94433 39967
~ LavanyaIT💖
~ 😊
+91 74180 98810
+91 99621 46336
+91 76959 04608
+91 81244 73373
+91 85083 18039
~ Anees
~ Joshwa
+91 70104 45374
+91 91 76334 828
+91 80155 33580
~ Lokesh Kumar S
+91 89395 93779
+91 73733 39882
~ Anandha Krishnan
+91 80151 91590
~ Regular Guy
+91 6369 858 529
+91 78128 85051
+91 6383 166 871
~ Sanjith
~ Lucky
~ HARISH RAJ
~ Sangeetha Chandrasekaran
~ Sruthika
~ **yazhini Ramesh**💖
~ Sreevarshan :)
~ Sridar
~ Logith
~ Ms😊
~ Dharun
+91 95431 90629
~ Kabilan
+91 73056 25882
~ Jayaseelan.U
+91 93631 67941
~ Sandhiya
~ Gautham
~ Aksharaa
+91 6383 487 321
~ prithiviraj
Saranya Sivanandham
~ lil ambitious 📷
~ Heshinth
~ ...Krishnan....👤...D..
+91 89254 44655
+91 70104 38570
~ Br..
+91 73580 14159

~ Jayachandran ✨
~ Vishal 😊🌍
+91 89250 89887
~ ❤️ Arjuna ❤️
~ Maneeshika Jayakumar ✨
~ L
+91 89391 57959
+91 95147 15780
+91 73959 02664
~ Athi 💎
+91 87787 55684
+91 90250 66782
+91 98401 09586
+91 93453 08019
+91 6381 998 081
+91 86678 44756
+91 6383 126 136
+91 94869 33839
+91 93614 51573
+91 91 5045 040 1
~ Jebisha 🌸
~ Mr .Sai 😊
+91 93426 83811
~ Maria Betsy
+91 6385 721 237
+91 75581 07783
+91 96293 20388
+91 6380 103 115
+91 97901 51293
+91 90255 18163
+91 99762 31117
~ TUVI
~ Sivaneshwaran
+91 98946 32065
~ Pooja Murugan ❤️
+91 91 5072 134 5
+91 87789 55711
~ Akshaya 😊
~ Murshitha
~ Navya
+91 90438 32941
~ Prithika Vaishnavi
+91 93445 04699
~ Manjhari.S
+91 93927 81901
+91 95149 30663
~ Sudhanthira Sennakesavan
~ k.mahe_06
+91 97902 71779
+91 88076 36414
+91 94446 68085
+91 95245 91668
+91 98409 83043
+91 82201 85904
+91 86105 20241
+91 88256 32958

~ 🤪~shaluma MK~ 🤪
~ Dharshu
~ Nithiyanandham
+91 86104 63717
~ Gowtham
~ Deepak
~ ANJALI
~ Sanjay
~ **VENKAT**❤️ 🤪
+91 97899 11358
+91 91 76045 796
~ DHEERAJ PABOLU ™
~ Manoj kumar S
+91 88384 59406
~ 🤪MR🤪
~ Varun~
~ ❤️
~ Kaushik💞
~ majith
~ purusothaman 🤪
~ yokesh
~ Praveen S
~ Aswin
~ **LOKEE**
~ Harish N
~ Harikaran.š
~ Sandeep
~ Balagovind
~ Surendran🤪
~ Ashwin
~ Aravind🤪
~ Saran
~ Prem Kumar.P
~ vishal
~ Day.owl_66
~ Sanjay Narayanan
~ Lokesh Ravichandran
~ sai.... ❤️
~ NIRMAL B 🤪
~ R.P.L
~ Tarun Raj D
~ Roshan
~ S. Midun
+91 93609 92351
~ Shanmugam
+91 99402 65356
~ Suresh.__.vr.__.46
~ SAI SRIRAM
~ Lobo
~ Pooja gnanam
~ Blaze
+91 95858 98998
~ MUKESH KUMAR S
+91 98407 85623
+91 91 5027 653 5
+91 95054 86171

~ 📶
+91 79047 26066
~ Kabilesh
~ Uday
+91 6379 498 477
+91 91 76060 439
+91 90806 19919
~ Alhamdulillah
~ Lohith
+91 6379 291 049
+91 99621 87961
~ ⚡ Jashu ⚡
+91 93617 62437
+91 73054 71244
~ rsuchi2979@gmail.com
+91 97505 57565
~ P.U.R.N.I 🌙
+91 93452 30210
+91 98415 49235
+91 73959 66556
+91 90800 18747
~ Aravindhan
~ Nitesh
+91 97878 79025
~ **Subhishha**
~ PradeepXD
+91 95976 74125
+91 93600 41259
+91 95663 01728
+91 80741 03473
~ TEJESH
+91 93618 13253
+91 86681 66606
+91 70105 47753
+91 78068 69941
+91 70101 78489
+91 77080 39828
+91 6379 564 673
~ K...R...
~ Elanthari Paiyan
+91 93610 85812
+91 73054 37637
~ Pooja
~ ~mythreya
~ Hariesh S
+91 93424 41108
+91 81487 82910
~ 😊
~ udhaya
~ Abhijith
~ Gunamadhy Raja
~ ❤️
+91 6374 769 955
~ nithish
+91 6380 885 562
~ EVIL DAD

+91 82481 15230
~ Monika P
~ ~Rlkv~
+91 73388 97287
+91 73974 38415
+91 93986 92641
+91 70108 48477
+91 89395 89292
~ ♥
~ Gayathri
+91 86808 93072
+91 95141 80804
~ Jyoshnavi
~ Shrinidhi Vijaykumar
~ Divyadharshini 📞💎📞 Jeevan
~ Aswinraaj
~ sudha
~ **Balaji B**
~ Likki
~ Likhitha
~ Swathi
~ ADIN GERMANO
~ Ts Reddy !!
~ Yashika
+91 93424 89248
+91 97012 81556
+91 91 5988 435 3
~ Kirthick
+91 6380 653 660
+91 93633 11857
+91 83007 16732
+91 72990 74299
+91 6369 218 864
~ Harshini.V
+91 6369 573 120
+91 89038 81231
+91 99940 92100
+91 89392 45798
+91 72003 77580
+91 73972 35544
+91 6382 033 594
+91 99441 40269
~ Ram Prajith
+91 73054 82448
+91 87780 05185
+91 99409 65722
+91 82206 62728
+91 90472 63093
+91 74184 72778
+91 93456 82144
~ Fahim@ ™
+91 93422 86791
+91 6382 152 878
+91 80560 73971
+91 70105 77232
+91 93610 85806

```
In [12]: ## Printing total unique users in data  
len(unique_values)
```

```
Out[12]: 1008
```

```
In [13]: ## Getting count of messages sent by user  
count = df['Username/Number'].value_counts()  
print(count)
```

```
Username/Number  
ThiyagaB          1218  
Koushik PEC       436  
~ ^_^            312  
~ Shankar Sai     192  
~ Fadhil          146  
...  
~ Jayasri         1  
+91 93613 21901   1  
~ PVVishal        1  
~ Vasundaraa      1  
+91 93610 85806   1  
Name: count, Length: 1007, dtype: int64
```

```
In [14]: ## Getting the unique for no. of times messages sent by a unique user  
count.unique()
```

```
Out[14]: array([1218, 436, 312, 192, 146, 145, 111, 87, 84, 72, 61,  
                55, 53, 49, 47, 45, 44, 42, 41, 38, 35, 32,  
                28, 26, 25, 24, 23, 22, 21, 20, 19, 18, 17,  
                16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6,  
                5, 4, 3, 2, 1], dtype=int64)
```

```
In [15]: import warnings  
warnings.filterwarnings("ignore", category=UserWarning)
```

Percentage of Messages sent by User

```
In [16]: import matplotlib.pyplot as plt  
import pandas as pd  
  
# Calculate the total message count  
total_messages = df['Username/Number'].count()  
  
# Calculate the message count and percentage for each user  
message_counts = df['Username/Number'].value_counts().reset_index()  
message_counts.columns = ['Username/Number', 'Count']  
message_counts['Percentage'] = (message_counts['Count'] / total_messages) * 100  
  
# Sort the message_counts DataFrame by Count in descending order  
message_counts = message_counts.sort_values(by='Count', ascending=False)  
  
# Threshold for aggregating users with less than 2% into 'Others'  
threshold = 2
```

```
# Create a new DataFrame to aggregate users with less than the threshold
top_users = message_counts[message_counts['Percentage'] >= threshold]
other_users = message_counts[message_counts['Percentage'] < threshold]

# Aggregate users with less than the threshold into 'Others'
other_users_count = other_users['Count'].sum()
top_users.loc[len(top_users)] = ['Others', other_users_count, other_users['Percentage'].sum()]

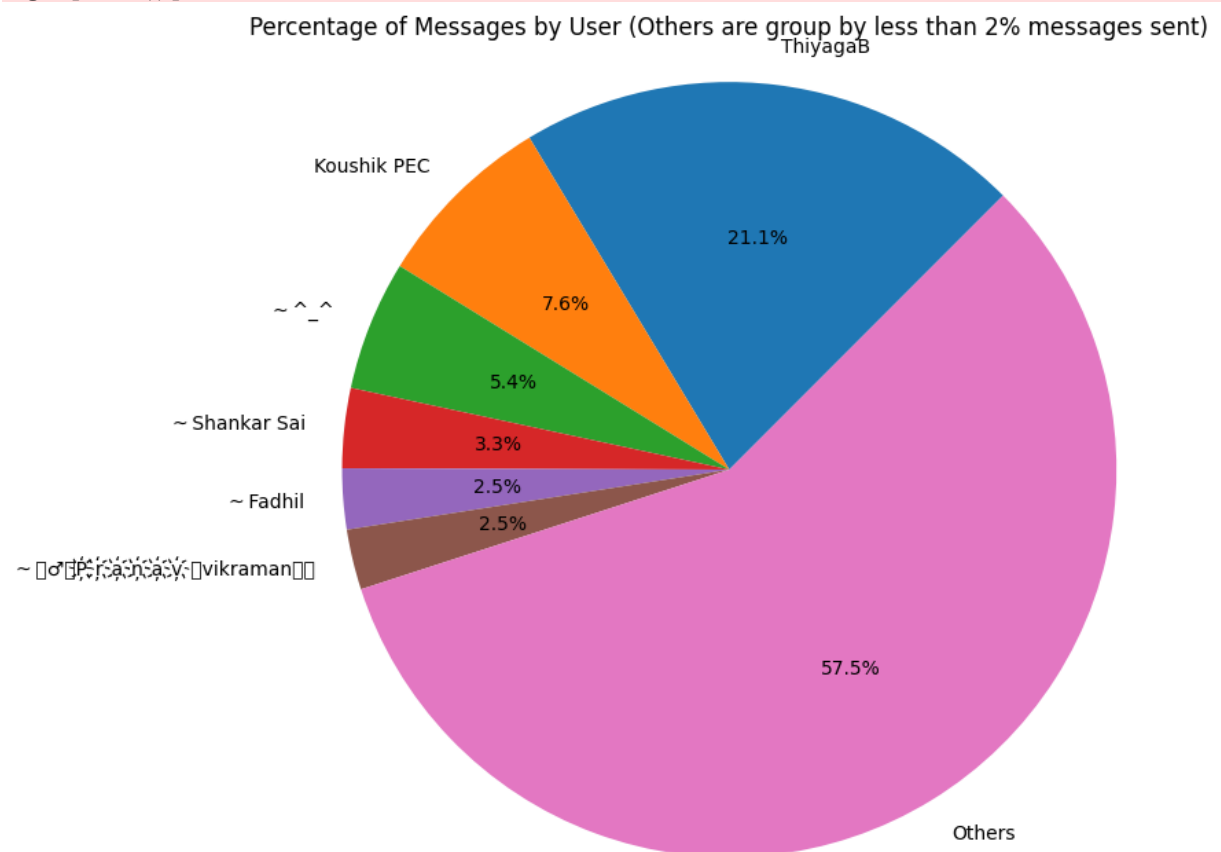
# Create a pie chart with labels at 45-degree angle
plt.figure(figsize=(8, 8))
plt.pie(top_users['Count'], labels=top_users['Username/Number'], autopct='%1.1f%%',
plt.axis('equal')
plt.title('Percentage of Messages by User (Others are group by less than 2% message
plt.show()
```

C:\Users\imyad\AppData\Local\Temp\ipykernel_18512\4095847584.py:24: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
top_users.loc[len(top_users)] = ['Others', other_users_count, other_users['Percentage'].sum()]
```



There was no null / missing values but the empty records was considered as nulls


```
Out[20]: 0      11:02:30 AM
1      11:02:30 AM
2      11:03:11 AM
3      11:06:18 AM
4      11:08:21 AM
...
10023   1:32:54 PM
10024   1:57:53 PM
10025   1:58:48 PM
10026   1:59:34 PM
10027   2:19:39 PM
Name: Time, Length: 5763, dtype: object
```

Transforming the Time Feature suitable for Time Series Analysis

```
In [21]: # Remove "AM" and "PM" from the 'Time' column if present
df['Time'] = df['Time'].str.replace(r' AM| PM', '', regex=True)

# Convert the 'Date' column to datetime format
df['Date'] = pd.to_datetime(df['Date'], format='%Y-%m-%d')

# Convert the 'Time' column to datetime format (as time only)
df['Time'] = pd.to_datetime(df['Time'], format='%I:%M:%S', errors='coerce').dt.time

# Create a new datetime column by combining 'Date' and 'Time'
df['Datetime'] = pd.to_datetime(df['Date'].astype(str) + ' ' + df['Time'].astype(str))

# Drop the original 'Date' and 'Time' columns if you don't need them
df.drop(columns=['Date', 'Time'], inplace=True)

# Set 'Datetime' as the index (important for time series analysis)
df.set_index('Datetime', inplace=True)

In [22]: ## Displaying head after conversion
df.head()
```

Out[22]:

	Username/Number	Message
Datetime		
2023-04-28 11:02:30	CodeSapiens - Code 'Seivom'	CodeSapiens - Code 'Seivom': Messages and cal...
2023-04-28 11:02:30	Product Manager	Product Manager: Product Manager created this...
2023-04-28 11:03:11	CodeSapiens - Code 'Seivom'	CodeSapiens - Code 'Seivom': Product Manager ...
2023-04-28 11:06:18	CodeSapiens - Code 'Seivom'	CodeSapiens - Code 'Seivom': You changed this...
2023-04-28 11:08:21	CodeSapiens - Code 'Seivom'	CodeSapiens - Code 'Seivom': You changed the ...

In [23]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 5763 entries, 2023-04-28 11:02:30 to 2023-10-01 02:19:39
Data columns (total 2 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Username/Number  5763 non-null   object
1   Message          5763 non-null   object
dtypes: object(2)
memory usage: 135.1+ KB
```

Finding the Day with maximum number of messages

```
In [24]: # Group by date and count the number of messages for each date
daily_message_counts = df.groupby(df.index.date)['Username/Number'].count()

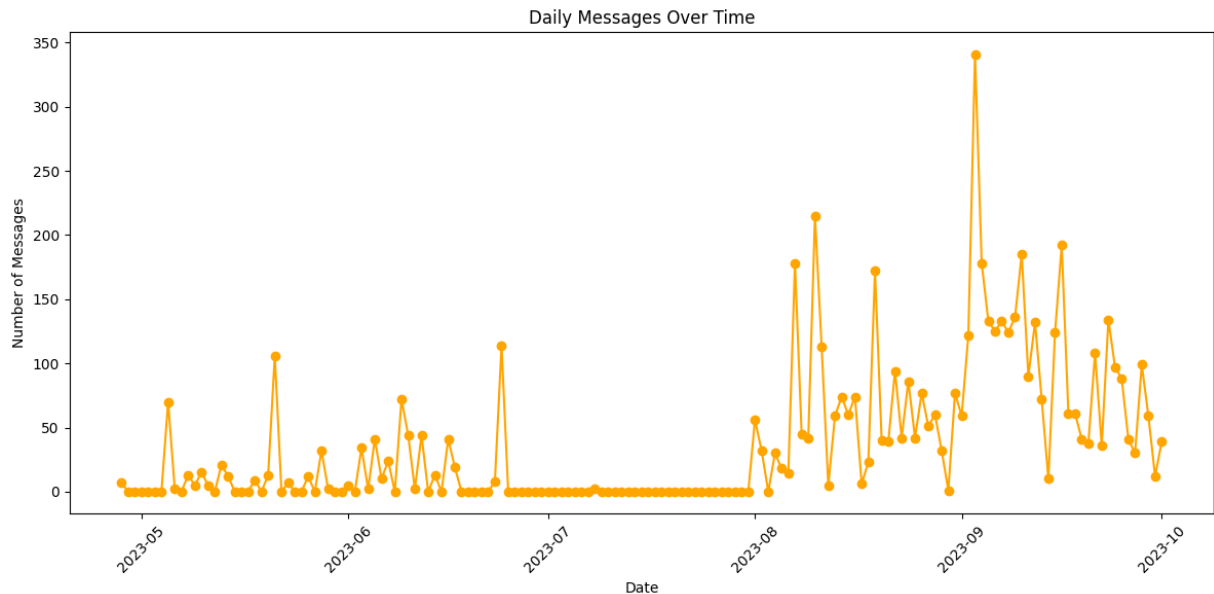
# Find the date with the most messages
most_messages_date = daily_message_counts.idxmax()
most_messages_count = daily_message_counts.max()

print(f"The day with the most messages is {most_messages_date} with {most_messages_
```

The day with the most messages is 2023-09-03 with 341 messages.

```
In [25]: # Resample the data to group by day and count the number of messages for each day
daily_message_counts = df.resample('D').count()

# Create a plot using Seaborn
plt.figure(figsize=(12, 6))
plt.plot(daily_message_counts.index, daily_message_counts['Message'], marker='o', c
plt.xlabel('Date')
plt.ylabel('Number of Messages')
plt.title('Daily Messages Over Time')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



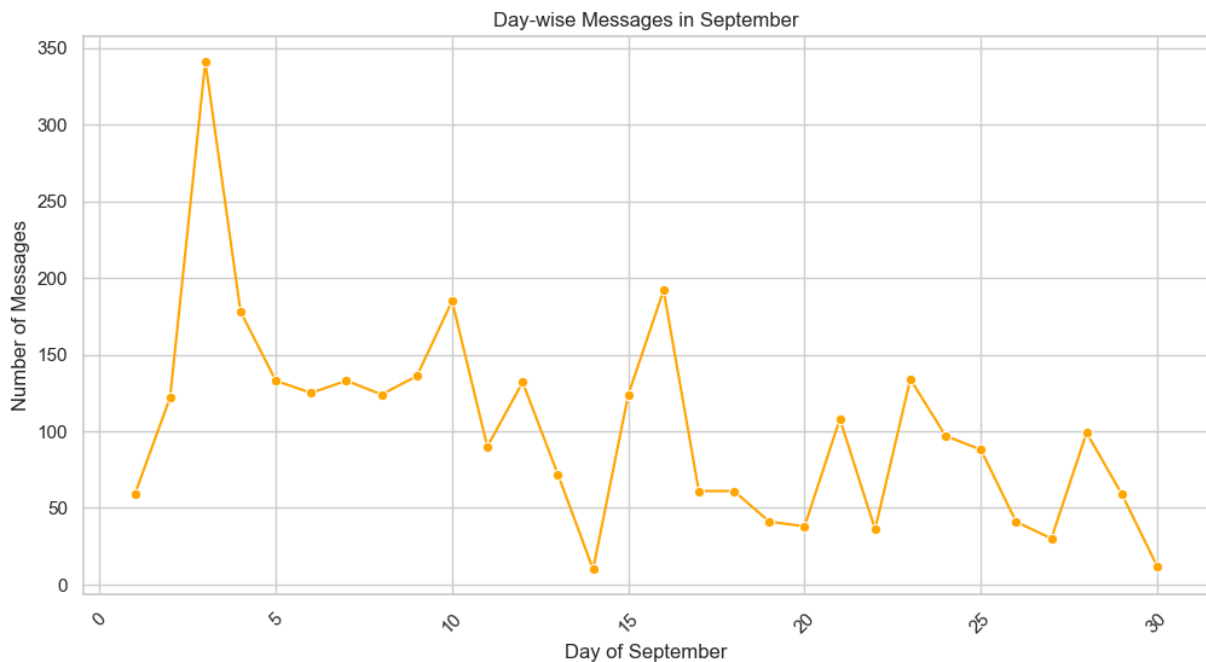
Since the Most messages was in September We analyze

Number of messages sent day wise in Setptember

```
In [26]: # Filter the DataFrame to include only data for September
september_data = df[(df.index.month == 9)]

# Resample the data to group by day and count the number of messages for each day
daily_message_counts = september_data.resample('D').count()

# Create a plot using Seaborn
plt.figure(figsize=(12, 6))
sns.set(style="whitegrid")
sns.lineplot(data=daily_message_counts, x=daily_message_counts.index.day, y='Message')
plt.xlabel('Day of September')
plt.ylabel('Number of Messages')
plt.title('Day-wise Messages in September')
plt.xticks(rotation=45)
plt.show()
```



The day with most messages was 2023-09-03 with 341 messages.

So it would be better for displaying the messages on that day

```
In [27]: import pandas as pd
import matplotlib.pyplot as plt
from wordcloud import WordCloud, STOPWORDS

# Filter the DataFrame for messages on the day with the most messages (2023-09-03)
date_with_most_messages = '2023-09-03'
filtered_df = df[df.index.date == pd.to_datetime(date_with_most_messages).date()]

# Extract the message text from the filtered DataFrame
messages = filtered_df['Message']

# Combine all messages into a single string
message_text = ' '.join(messages)

# Create a WordCloud object without specifying a font
wordcloud = WordCloud(
    font_path='Poppins/Poppins-Medium.ttf',
    width=800,
    height=400,
    background_color='white',
    stopwords=STOPWORDS
).generate(message_text)

# Display the WordCloud
plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation='bilinear')
```



```
plt.show()
```

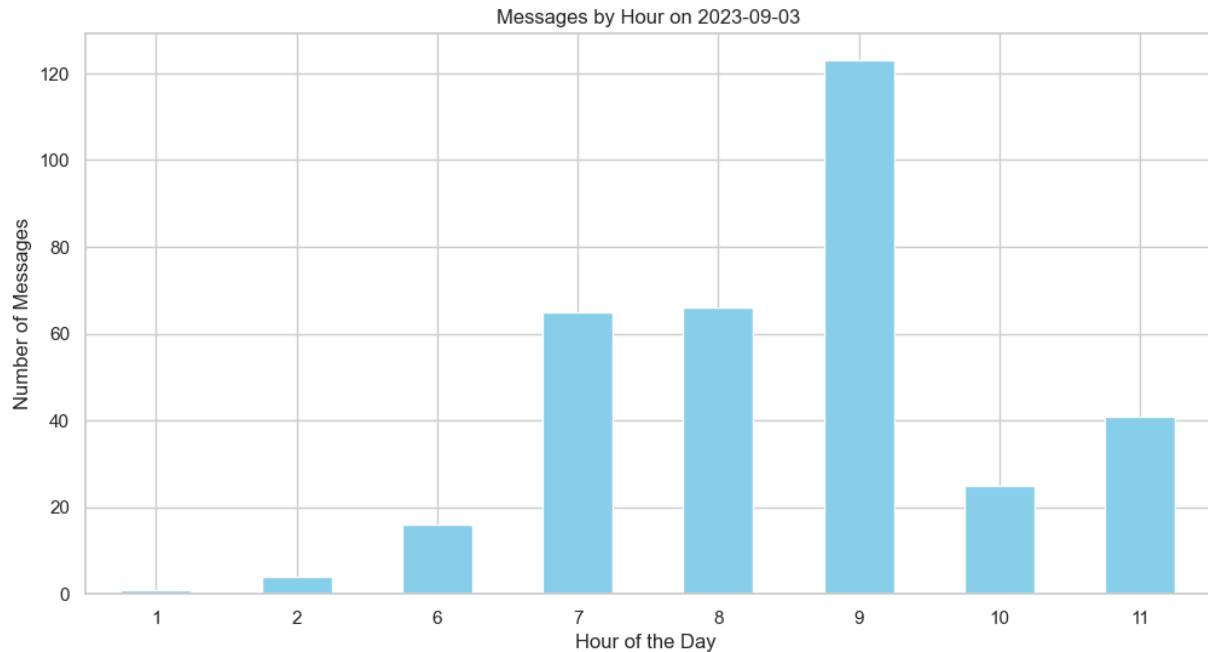
C:\Users\imyad\AppData\Local\Temp\ipykernel_18512\404231117.py:6: SettingWithCopyWarning:

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-a-copy

```
filtered_df['Hour'] = filtered_df.index.hour
```



Analyzing the day with No messages

```
In [30]: # Find the minimum and maximum dates in your DataFrame
min_date = df.index.min()
max_date = df.index.max()

# Create a list of all days within the date range of your DataFrame
all_days = pd.date_range(min_date, max_date).date

# Get the days with messages from the DataFrame's index
days_with_messages = df.index.date

# Find the days with no messages
days_with_no_messages = [day for day in all_days if day not in days_with_messages]

print("Days with no messages:")
print(days_with_no_messages)
```

Days with no messages:

```
[datetime.date(2023, 4, 29), datetime.date(2023, 4, 30), datetime.date(2023, 5, 1),
datetime.date(2023, 5, 2), datetime.date(2023, 5, 3), datetime.date(2023, 5, 4), dat
etime.date(2023, 5, 7), datetime.date(2023, 5, 12), datetime.date(2023, 5, 15), date
time.date(2023, 5, 16), datetime.date(2023, 5, 17), datetime.date(2023, 5, 19), date
time.date(2023, 5, 22), datetime.date(2023, 5, 24), datetime.date(2023, 5, 25), date
time.date(2023, 5, 27), datetime.date(2023, 5, 30), datetime.date(2023, 5, 31), date
time.date(2023, 6, 2), datetime.date(2023, 6, 8), datetime.date(2023, 6, 13), dateti
me.date(2023, 6, 15), datetime.date(2023, 6, 18), datetime.date(2023, 6, 19), dateti
me.date(2023, 6, 20), datetime.date(2023, 6, 21), datetime.date(2023, 6, 22), dateti
me.date(2023, 6, 25), datetime.date(2023, 6, 26), datetime.date(2023, 6, 27), dateti
me.date(2023, 6, 28), datetime.date(2023, 6, 29), datetime.date(2023, 6, 30), dateti
me.date(2023, 7, 1), datetime.date(2023, 7, 2), datetime.date(2023, 7, 3), datetime.
date(2023, 7, 4), datetime.date(2023, 7, 5), datetime.date(2023, 7, 6), datetime.dat
e(2023, 7, 7), datetime.date(2023, 7, 9), datetime.date(2023, 7, 10), datetime.date
(2023, 7, 11), datetime.date(2023, 7, 12), datetime.date(2023, 7, 13), datetime.date
(2023, 7, 14), datetime.date(2023, 7, 15), datetime.date(2023, 7, 16), datetime.date
(2023, 7, 17), datetime.date(2023, 7, 18), datetime.date(2023, 7, 19), datetime.date
(2023, 7, 20), datetime.date(2023, 7, 21), datetime.date(2023, 7, 22), datetime.date
(2023, 7, 23), datetime.date(2023, 7, 24), datetime.date(2023, 7, 25), datetime.date
(2023, 7, 26), datetime.date(2023, 7, 27), datetime.date(2023, 7, 28), datetime.date
(2023, 7, 29), datetime.date(2023, 7, 30), datetime.date(2023, 7, 31), datetime.date
(2023, 8, 3)]
```

Displaying what we inferred before as graph

```
In [31]: # Convert the List to a Pandas DataFrame
df_no_messages = pd.DataFrame({'Date': days_with_no_messages})

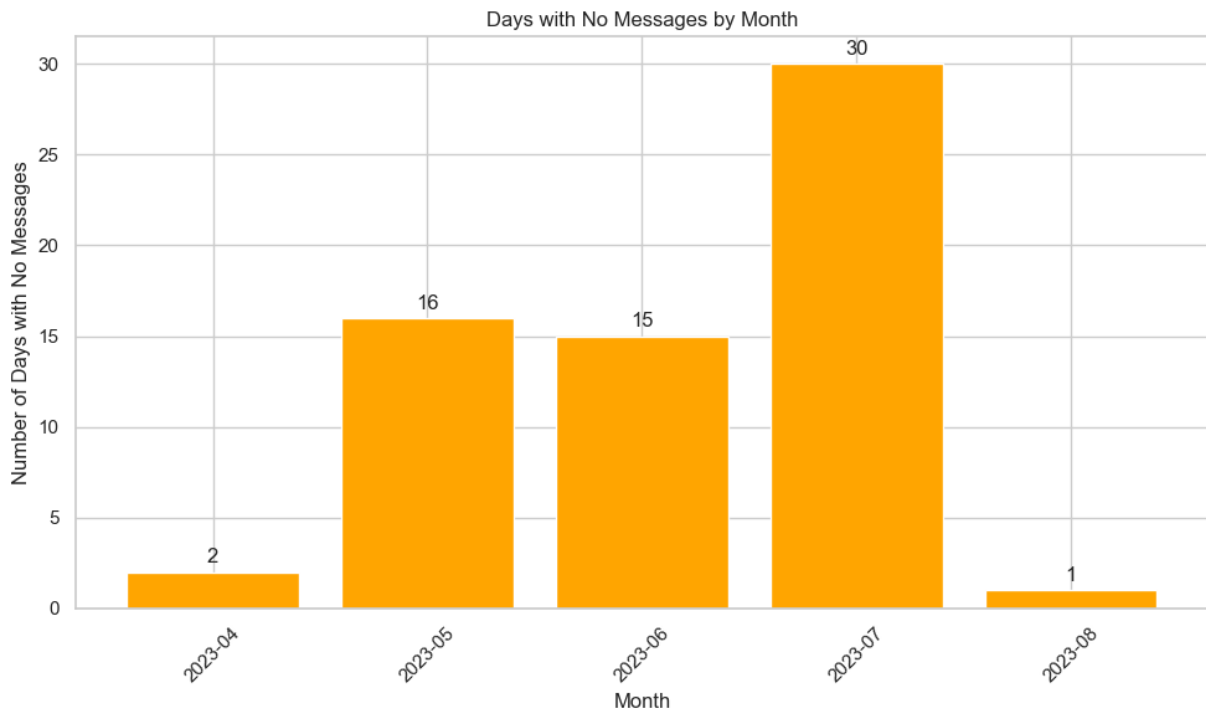
# Extract month and year from each date
df_no_messages['Month'] = df_no_messages['Date'].apply(lambda x: x.strftime('%Y-%m'))

# Group by month and count occurrences
monthly_counts = df_no_messages['Month'].value_counts().sort_index()

# Create a bar chart
plt.figure(figsize=(10, 6))
bars = plt.bar(monthly_counts.index, monthly_counts.values, color='orange')
plt.xticks(rotation=45)
plt.xlabel('Month')
plt.ylabel('Number of Days with No Messages')
plt.title('Days with No Messages by Month')

# Add count annotations on top of the bars
for bar in bars:
    height = bar.get_height()
    plt.annotate(f'{height}', xy=(bar.get_x() + bar.get_width() / 2, height),
                xytext=(0, 3), textcoords='offset points',
                ha='center', va='bottom')

plt.tight_layout()
plt.show()
```

In []:

```
In [32]: # Group the data by month and count messages
monthly_counts = df.resample('M').count()['Message']

# Calculate the cumulative sum
cumulative_sum = monthly_counts.cumsum()

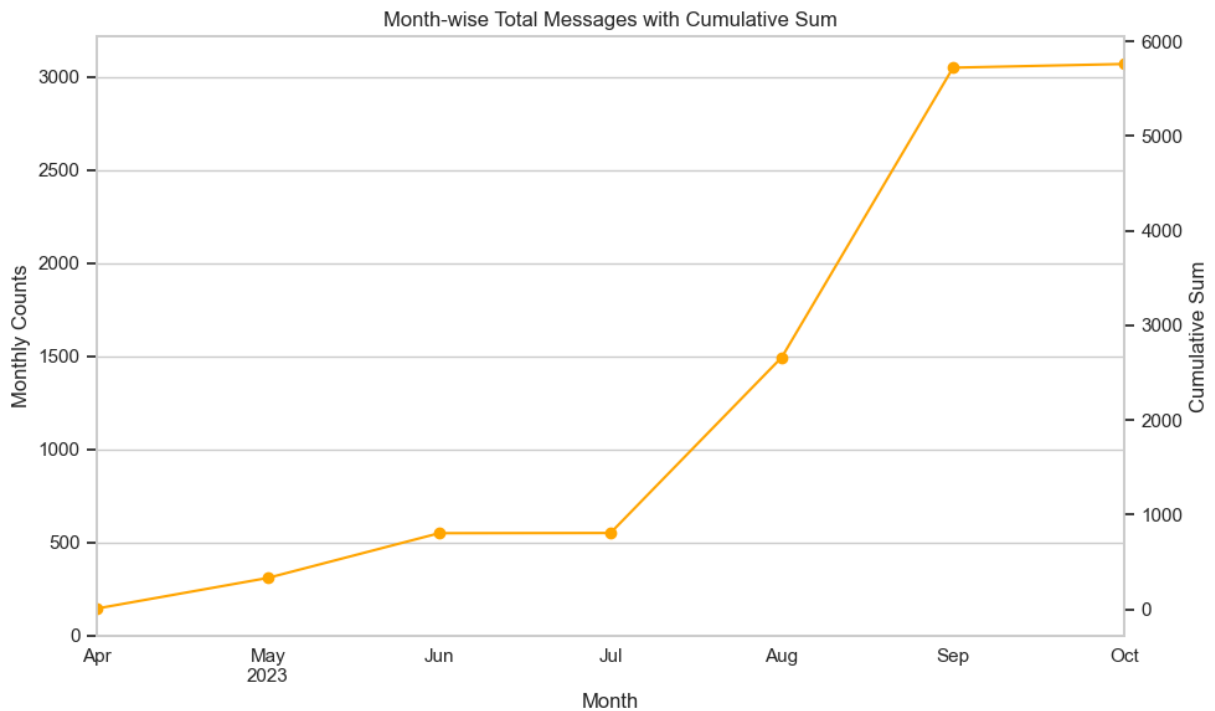
In [33]: # Create a figure and axis for the plot
fig, ax = plt.subplots(figsize=(10, 6))

# Plot the monthly counts
monthly_counts.plot(kind='bar', ax=ax, label='Monthly Counts', color='skyblue', alp

# Plot the cumulative sum as a line
cumulative_sum.plot(secondary_y=True, ax=ax, marker='o', color='orange', label='Cum

# Set Labels and Legend
ax.set_xlabel('Month')
ax.set_ylabel('Monthly Counts')
ax.right_ax.set_ylabel('Cumulative Sum')

# Display the plot
plt.title('Month-wise Total Messages with Cumulative Sum')
plt.tight_layout()
plt.show()
```



Making the Feature 'Message' meaningful

```
In [34]: ## Displaying the message feature
df['Message']
```

```
Out[34]: Datetime
2023-04-28 11:02:30    CodeSapiens - Code 'Seivom': Messages and cal...
2023-04-28 11:02:30    Product Manager: Product Manager created this...
2023-04-28 11:03:11    CodeSapiens - Code 'Seivom': Product Manager ...
2023-04-28 11:06:18    CodeSapiens - Code 'Seivom': You changed this...
2023-04-28 11:08:21    CodeSapiens - Code 'Seivom': You changed the ...
...
2023-10-01 01:32:54    +91 99441 40269: +91 99441 40269 left
2023-10-01 01:57:53    ~ Mr.Coder: enakum athala pudikathu sir..so pr...
2023-10-01 01:58:48    ~ Mr.Coder: kandipaaga sir
2023-10-01 01:59:34    ~ Mr.Coder: fine sir
2023-10-01 02:19:39    +91 93610 85806: +91 93610 85806 joined u...
Name: Message, Length: 5763, dtype: object
```

```
In [35]: # Extract data after ':' in all records of the 'Message' column
df['Message'] = df['Message'].str.split(':', 1).str[-1]
```

Finding Number of Unique Words Used

```
In [36]: from collections import Counter

messages = df['Message']

# Number of Unique Words Used
unique_words = set(" ".join(messages).split())
```

```
num_unique_words = len(unique_words)
print(f"Number of Unique Words Used: {num_unique_words}")
```

Number of Unique Words Used: 11727

Finding the most commonly used word

```
In [37]: # Most Commonly Used Word
word_counts = Counter(" ".join(messages).split())
most_common_word, most_common_word_count = word_counts.most_common(1)[0]
print(f"Most Commonly Used Word: '{most_common_word}' (Used {most_common_word_count})")
```

Most Commonly Used Word: 'this' (Used 1197 times)

```
In [38]: from collections import Counter

# Combine all messages into a single string
message_text = ' '.join(messages)

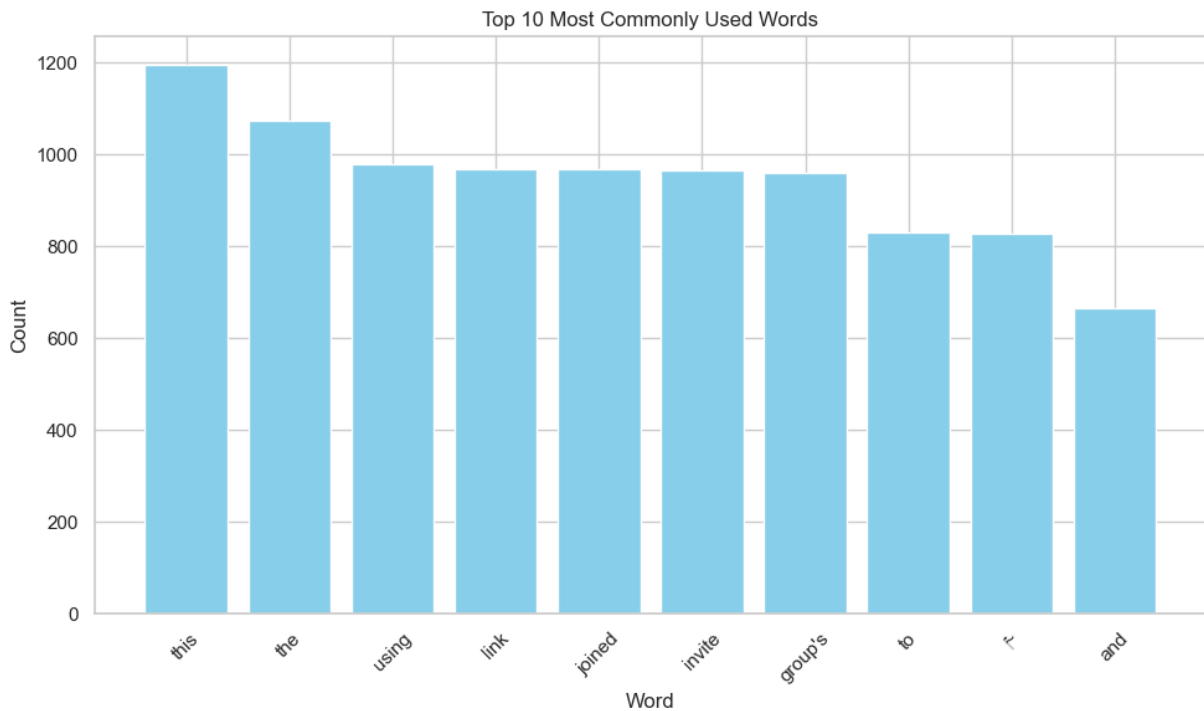
# Tokenize the message text (split it into words)
words = message_text.split()

# Count the frequency of each word
word_counts = Counter(words)

# Find the top 10 most commonly used words
top_10_words = word_counts.most_common(10)

# Create a DataFrame for the top 10 words
df_top_words = pd.DataFrame(top_10_words, columns=['Word', 'Count'])

# Create a bar plot
plt.figure(figsize=(10, 6))
plt.bar(df_top_words['Word'], df_top_words['Count'], color='skyblue')
plt.xlabel('Word')
plt.ylabel('Count')
plt.title('Top 10 Most Commonly Used Words')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



Finding the longest message sent and the user who sent it

```
In [39]: # Calculate the length of each message
df['Message_Length'] = df['Message'].str.len()

# Find the index of the row with the longest message
index_of_longest_message = df['Message_Length'].idxmax()

# Get the username and message text of the longest message
longest_message_username = df.loc[index_of_longest_message, 'Username/Number']
longest_message_text = df.loc[index_of_longest_message, 'Message']

print(f"Username of Longest Message: {longest_message_username}")
print(f"Longest Message: {longest_message_text}")
```

Username of Longest Message: ~ Pharaoh Suriya

Longest Message: If you ask me, I do think it's one of the basic results of evolution at play. And humans especially have knowledge that gets passed on beyond generations other than through genetics by something called culture. This is an extremely powerful tool as the only thing that a living thing is concerned with seems to be the survival of its kind and so will be willing to do anything to achieve that and that thing is generally termed intelligence. So when we see something that we know is going to make us live longer or help us reproduce, it checks in for things necessary to survive by our genetic material whose only purpose is to continue existing. So, we tend to do such things and keep it in higher regard. For example, ancient humans saw forest fire and how animals that got cooked in them tasted and how they were better because they won't cause much diseases as the bacteria will be destroyed by heat and that became an incentive to learn how to create a fire, similarly with other things, everything just ensures one's survival. So in today's world money and fame can both keep you above others of our species which means better survival, this is what motivates every human to do things like these. Coming to Tamil Nadu I guess we are somewhat too stuck to culture though I'm not sure why but I do have guesses, maybe because our education system is still the very system created by the British to create slaves, like you aren't programmed to ask questions or be creative and both of these are discouraged in our society and therefore we tend to be extremely not adventurous and do things that are already established to ensure our survival, so if it comes to sports you go with cricket because it's established, similarly with IITs like I was in that rabbit hole for most of my schooling, like Sundar Pichai was enough to advertise for that and I think the same applies to chess, we have become lazy and less adventurous due to our society and education system that discourage us from having creativity we just do what we know will help us survive and in that way as people have said earlier it's no surprise that chess is a cultural thing and few big names in the industry is enough incentives to take that as a livelihood, if you ask me, L for the education system and a bigger L for the society, one advice would be to please use more creativity don't listen to toxic people that say being adventurous is stupid. Thank you 😊

Displaying the Top 10 Longest messages

```
In [40]: import pandas as pd
import matplotlib.pyplot as plt

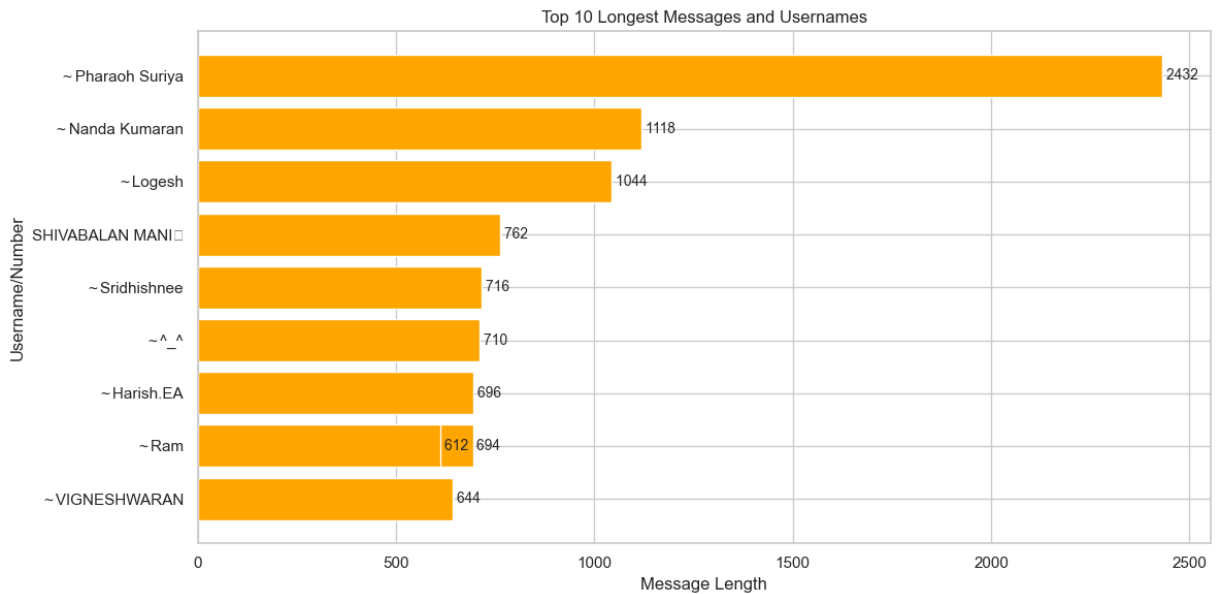
# Sort the DataFrame by message length in descending order
df_sorted = df.sort_values(by='Message_Length', ascending=False)

# Select the top 10 longest messages and usernames
top_10_longest = df_sorted.head(10)

# Create a bar chart to display the top 10 longest messages and usernames
plt.figure(figsize=(12, 6))
bars = plt.barh(top_10_longest['Username/Number'], top_10_longest['Message_Length'])
plt.xlabel('Message Length')
plt.ylabel('Username/Number')
plt.title('Top 10 Longest Messages and Usernames')
plt.gca().invert_yaxis() # Invert the y-axis to display the longest message at the top

# Add message lengths on top of the bars
for bar, length in zip(bars, top_10_longest['Message_Length']):
    plt.text(bar.get_width() + 10, bar.get_y() + bar.get_height()/2, f'{length} ',
```

```
plt.tight_layout()
plt.show()
```



Analyzing the links shared in chat

```
In [49]: import re

# Define a regular expression pattern to match URLs
url_pattern = r'https?://\S+|www\.\S+'

# Function to extract URLs from a text
def extract_urls(text):
    return re.findall(url_pattern, text)

# Apply the function to extract URLs from the 'Message' column
df['Links'] = df['Message'].apply(extract_urls)

# Create a new DataFrame to store the links alone
links_df = pd.DataFrame({'Links': df['Links'].explode().dropna()})

links_df.head()
```

Out[49]:

Links

Datetime

2023-05-05 03:58:15	https://chat.whatsapp.com/Hk6Nqyvub6P49LuRmu8PqB
2023-05-05 06:25:01	https://youtu.be/zDAYZU4A3w0
2023-05-08 10:34:21	https://takkunu-dfd27.web.app/
2023-05-11 07:08:45	https://www.geeksforgeeks.org/complete-roadmap...
2023-05-26 09:42:16	https://youtu.be/8J-V3J3CBes

```
In [53]: # Assuming you have the 'Links' column in the 'links_df' DataFrame
# Extract the part of the links after '//' and before '/'
links_df['Extracted Links'] = links_df['Links'].str.extract(r'//(.*)/')

links_df.head()
```

```
Out[53]:
```

	Links	Extracted Links
Datetime		
2023-05-05 03:58:15	https://chat.whatsapp.com/Hk6NqyvuB6P49LuRmu8PqB	chat.whatsapp.com
2023-05-05 06:25:01	https://youtu.be/zDAYZU4A3w0	youtu.be
2023-05-08 10:34:21	https://takkunu-dfd27.web.app/	takkunu-dfd27.web.app
2023-05-11 07:08:45	https://www.geeksforgeeks.org/complete-roadmap...	www.geeksforgeeks.org
2023-05-26 09:42:16	https://youtu.be/8J-V3J3CBes	youtu.be

```
In [54]: links_df['Extracted Links'].value_counts()
```

```

Out[54]: Extracted Links
youtube.com          13
instagram.com        10
www.youtube.com      5
youtu.be             5
www.instagram.com    5
chat.whatsapp.com    4
www.indiatoday.in    2
internshala.com      2
www.producthunt.com  1
medium.com           1
x.com                1
ig.me                1
spotify.link         1
play.google.com      1
www.linkedin.com     1
icml.cc              1
vtopcc6.vit.ac.in   1
www.collegeevents.info 1
devdocs.io           1
www.knowafest.com    1
towardsdatascience.com 1
forms.gle            1
t.me                 1
hnm.lumoscit.in     1
about.fb.com         1
pravartak.org.in    1
www.businesstoday.in 1
www.cloudskillsboost.google 1
www.apple.com        1
learndigital.withgoogle.com 1
www.geeksforgeeks.org 1
takkunu-dfd27.web.app 1
event.webinarjam.com 1
Name: count, dtype: int64

```

```

In [58]: import pandas as pd
import matplotlib.pyplot as plt

# Assuming you have the 'Extracted Links' column in the 'links_df' DataFrame
# Count the occurrences of each extracted link
link_counts = links_df['Extracted Links'].value_counts()

# Select the top 10 most common extracted links
top_10_links = link_counts.head(10)

# Create a bar chart to visualize the top 10 extracted links
plt.figure(figsize=(12, 6))
ax = top_10_links.plot(kind='bar', color='red')
plt.xlabel('Extracted Links')
plt.ylabel('Count')
plt.title('Top 10 Most Common Extracted Links')
plt.xticks(rotation=45)

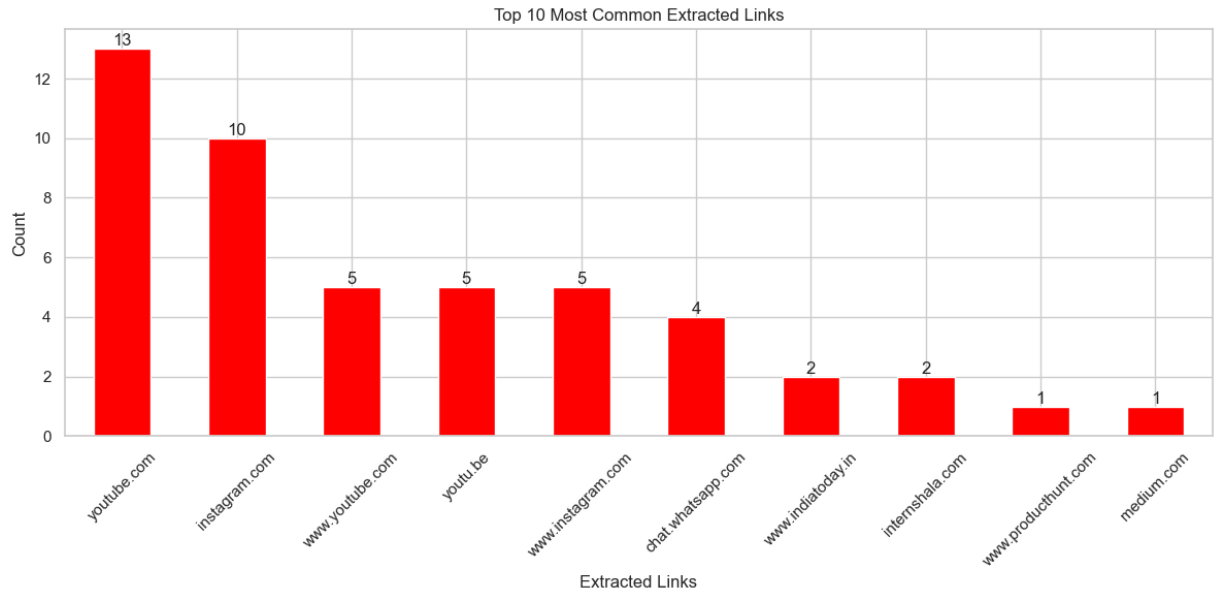
# Display the count on top of each bar
for i, count in enumerate(top_10_links):

```



```
ax.text(i, count, str(count), ha='center', va='bottom')

plt.tight_layout()
plt.show()
```



**ANDDD Here Comes the interesting part....
(Sentiment Analysis on each message)**

```
In [43]: from textblob import TextBlob

# Create a function to analyze sentiment
def analyze_sentiment(text):
    analysis = TextBlob(text)
    if analysis.sentiment.polarity > 0:
        return "Positive"
    elif analysis.sentiment.polarity == 0:
        return "Neutral"
    else:
        return "Negative"

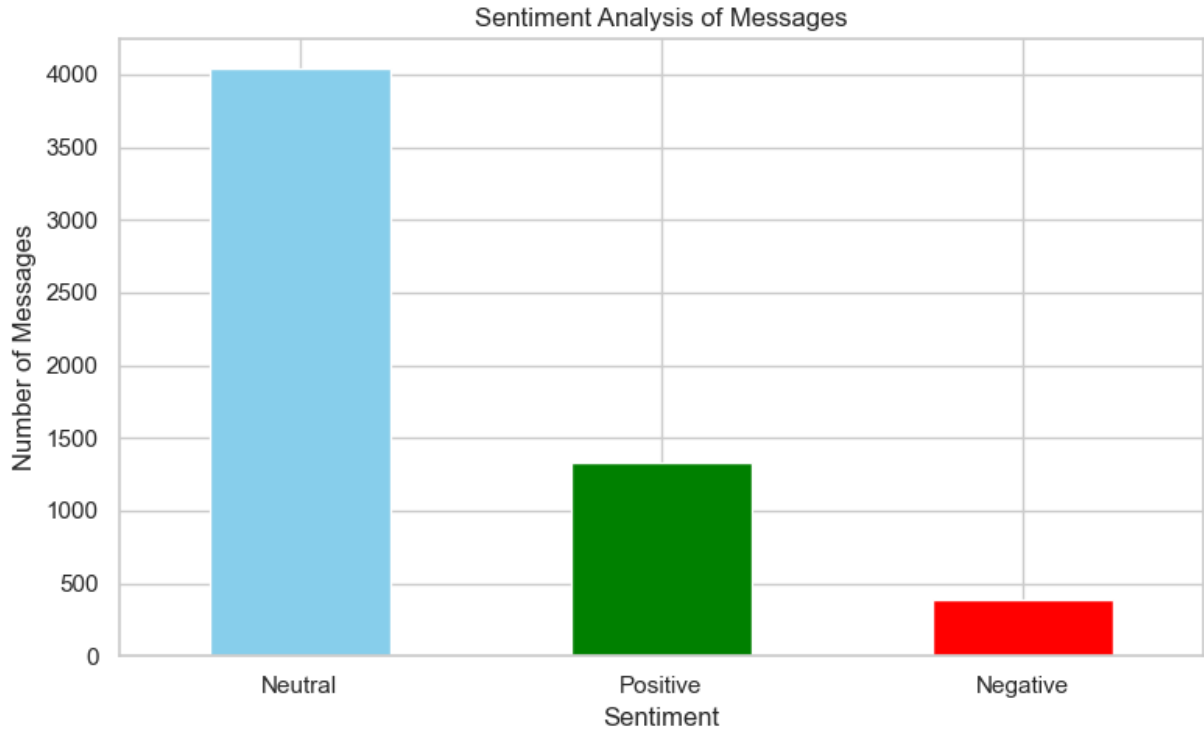
# Apply the sentiment analysis function to your DataFrame
df['Sentiment'] = df['Message'].apply(analyze_sentiment)

# Count the number of messages in each sentiment category
sentiment_counts = df['Sentiment'].value_counts()

# Define custom colors for each sentiment category
colors = {'Positive': 'green', 'Neutral': 'skyblue', 'Negative': 'red'}

# Create a bar chart with custom colors
plt.figure(figsize=(8, 5))
sentiment_counts.plot(kind='bar', color=[colors[sentiment] for sentiment in sentiment_counts.index])
plt.xlabel('Sentiment')
plt.ylabel('Number of Messages')
plt.title('Sentiment Analysis of Messages')
plt.xticks(rotation=0)
```

```
plt.tight_layout()
plt.show()
```



In [44]: `df.head()`

Out[44]:

	Username/Number	Message	Message_Length	Links	Sentiment
Datetime					
2023-04-28 11:02:30	CodeSapiens - Code 'Seivom'	Messages and calls are end-to-end encrypted. ...	121	[]	Neutral
2023-04-28 11:02:30	Product Manager	Product Manager created this group	35	[]	Neutral
2023-04-28 11:03:11	CodeSapiens - Code 'Seivom'	Product Manager added you	26	[]	Neutral
2023-04-28 11:06:18	CodeSapiens - Code 'Seivom'	You changed this group's icon	30	[]	Neutral
2023-04-28 11:08:21	CodeSapiens - Code 'Seivom'	You changed the group name to "CodeSapiens - ...	57	[]	Neutral

In [45]:

```
# Group messages by 'Username' and 'Sentiment', then count the number of messages i
sentiment_counts = df.groupby(['Username/Number', 'Sentiment']).size().unstack(fill

# Calculate the total positive, negative, and neutral messages for each user
sentiment_counts['Total Positive'] = sentiment_counts['Positive']
```

```

sentiment_counts['Total Negative'] = sentiment_counts['Negative']
sentiment_counts['Total Neutral'] = sentiment_counts['Neutral']

# Sort users by total positive, negative, and neutral messages and select the top 10
top_10_positive_users = sentiment_counts['Total Positive'].sort_values(ascending=False)
top_10_negative_users = sentiment_counts['Total Negative'].sort_values(ascending=False)
top_10_neutral_users = sentiment_counts['Total Neutral'].sort_values(ascending=False)

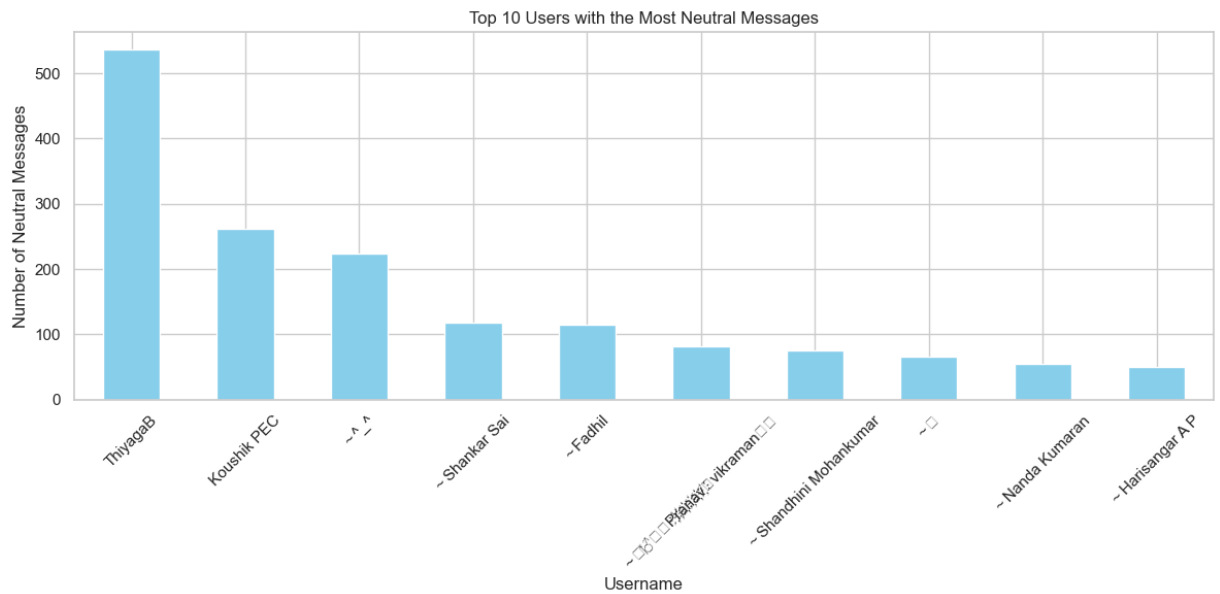
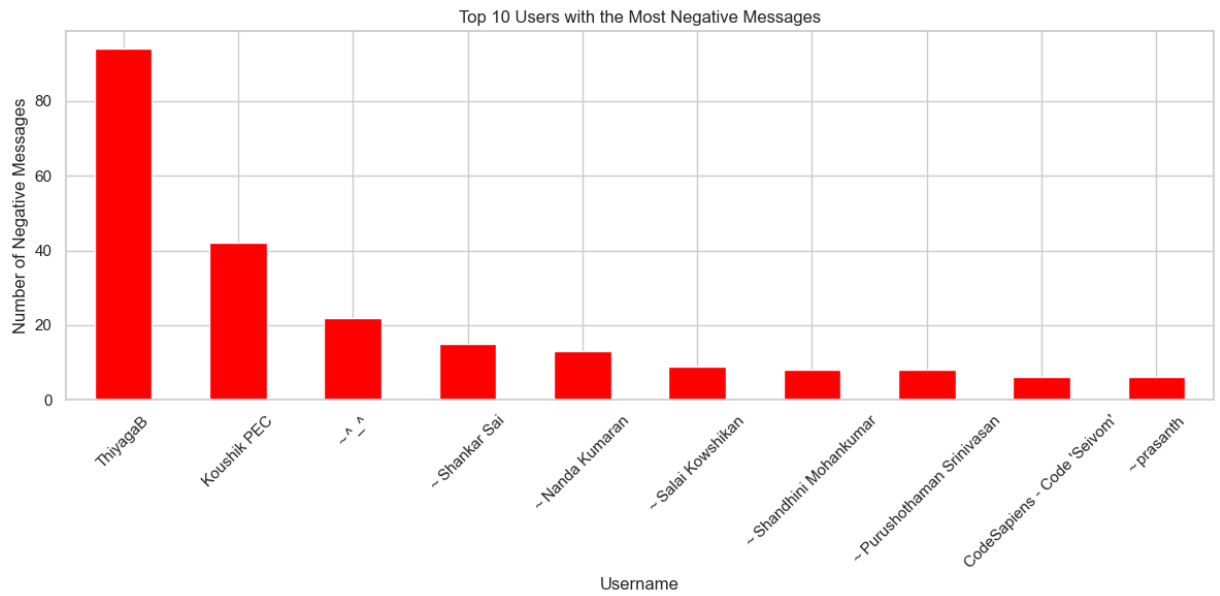
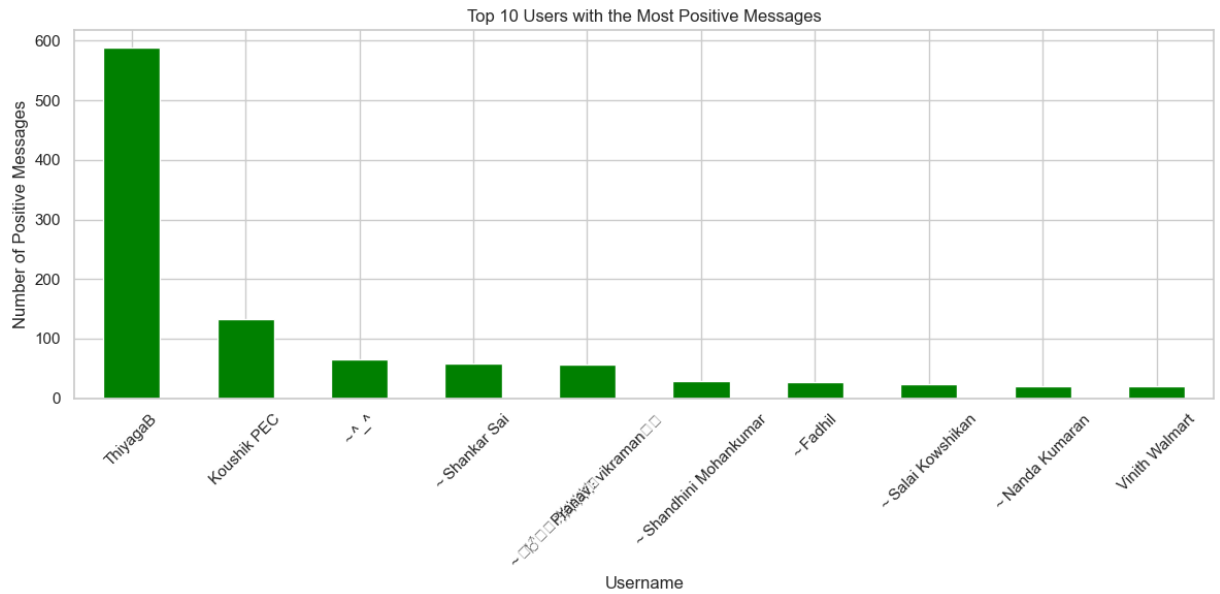
# Create separate bar charts for the top 10 positive, negative, and neutral users

# Bar chart for top 10 users with the most Positive messages
plt.figure(figsize=(12, 6))
top_10_positive_users.plot(kind='bar', color='green')
plt.xlabel('Username')
plt.ylabel('Number of Positive Messages')
plt.title('Top 10 Users with the Most Positive Messages')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

# Bar chart for top 10 users with the most Negative messages
plt.figure(figsize=(12, 6))
top_10_negative_users.plot(kind='bar', color='red')
plt.xlabel('Username')
plt.ylabel('Number of Negative Messages')
plt.title('Top 10 Users with the Most Negative Messages')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

# Bar chart for top 10 users with the most Neutral messages
plt.figure(figsize=(12, 6))
top_10_neutral_users.plot(kind='bar', color='skyblue')
plt.xlabel('Username')
plt.ylabel('Number of Neutral Messages')
plt.title('Top 10 Users with the Most Neutral Messages')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

```



Label Encoding the Sentiment data to numeric

```
In [46]: # Create a mapping dictionary for sentiment labels
sentiment_mapping = {'Positive': 1, 'Negative': 0, 'Neutral': 2}

# Map the 'Sentiment' column to numeric values using the mapping dictionary
df['Sentiment_Numeric'] = df['Sentiment'].map(sentiment_mapping)

# Now, df['Sentiment_Numeric'] will contain numeric labels (1 for Positive, 0 for N
```

```
In [47]: df.head()
```

```
Out[47]:
```

	Username/Number	Message	Message_Length	Links	Sentiment	Sentiment
	Datetime					
	2023-04-28 11:02:30	CodeSapiens - Code 'Seivom'	Messages and calls are end-to-end encrypted. ...	121	[]	Neutral
	2023-04-28 11:02:30	Product Manager	Product Manager created this group	35	[]	Neutral
	2023-04-28 11:03:11	CodeSapiens - Code 'Seivom'	Product Manager added you	26	[]	Neutral
	2023-04-28 11:06:18	CodeSapiens - Code 'Seivom'	You changed this group's icon	30	[]	Neutral
	2023-04-28 11:08:21	CodeSapiens - Code 'Seivom'	You changed the group name to "CodeSapiens - ...	57	[]	Neutral

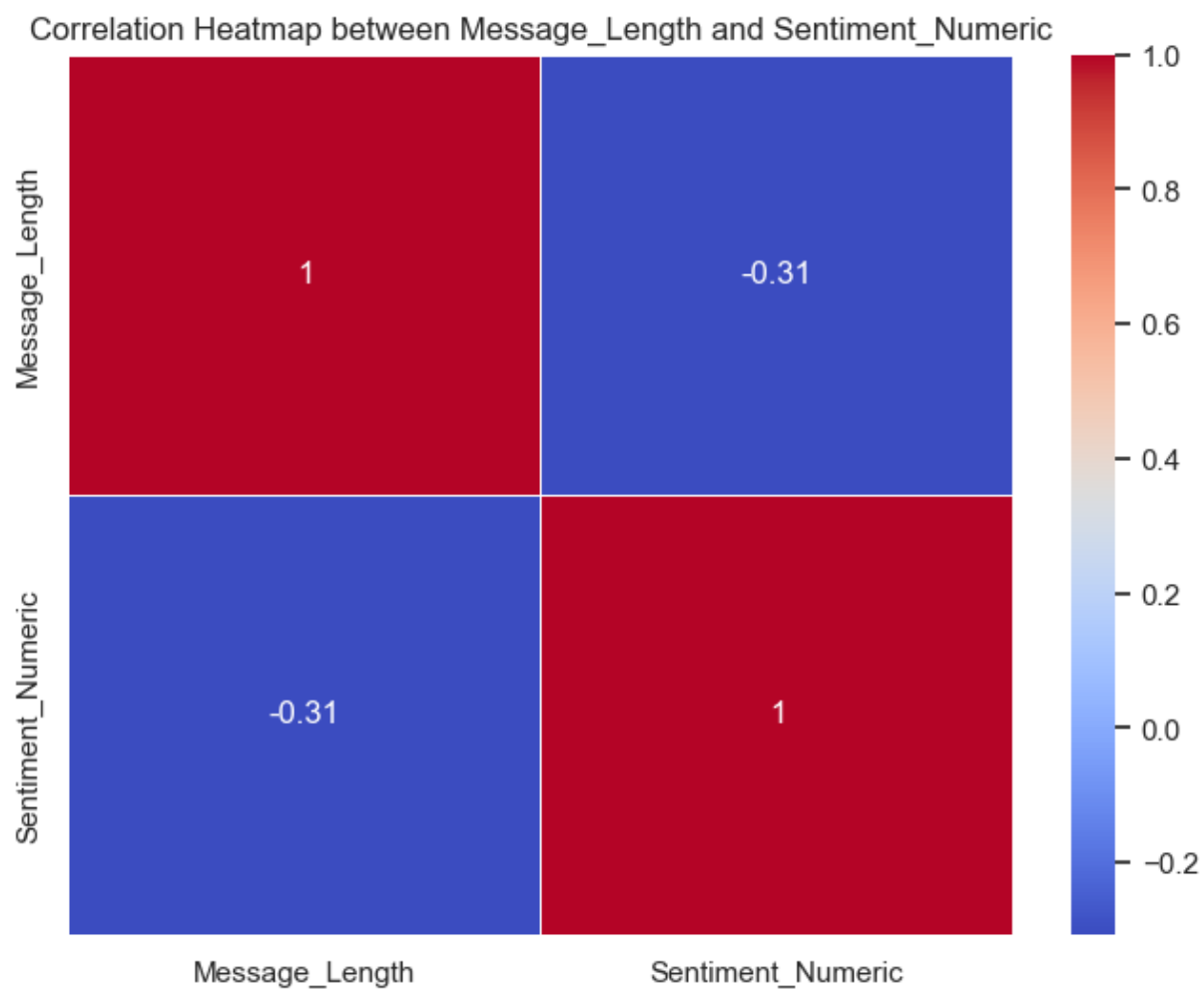
Analyzing if Message length and sentiment has correlation

```
In [48]: import seaborn as sns
import matplotlib.pyplot as plt

# Calculate the correlation matrix
correlation_matrix = df[['Message_Length', 'Sentiment_Numeric']].corr()

# Create a heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', linewidths=.5)
```

```
plt.title('Correlation Heatmap between Message_Length and Sentiment_Numeric')
plt.show()
```



In []:

In []:

In []:

In []:

In []:

In []: