Author: Karthik Sharma

Title: Analyzing the whatsApp group chat using pandas and Matplotlib

Importing required Libraries

In [239]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import re
import seaborn as sns
import emoji
import collections
from datetime import datetime
import calendar
from wordcloud import WordCloud, STOPWORDS
```

Data Preparation to fetch the date, time, author and message to create a dataframe

In [240]:

```
def startsWithDate(line):
    pattern='^([0-9]{2})(\/)([0-9]{2})(\/)([0-9]{2})'
    check=re.match(pattern,line)
    if check:
        return True
    else:
        return False

def checkAuthor(data):
    data=data.split(':')
    if(len(data)>1):
        return True
    else:
        return False
```

In [241]:

```
record_data=[]
file =open('WhatsApp Chat with IPL.txt',encoding="utf-8")
file.readline()
message=[]
while True:
    line=file.readline()
    date=None
    time=None
    author=None
    if not line:
        break
    line=line.strip()
    if(startsWithDate(line)):
        date=line.split(',')[0]
        data=" ".join(line.split(',')[1:]).strip()
        time=data.split('-')[0].strip()
        data=" ".join(data.split('-')[1:]).strip()

        if checkAuthor(data):
            message=[]
            author=data.split(':')[0].strip()
            message.append("".join(data.split(':')[1:]).strip())
            record_data.append([date,time,author,message])
        else:
            message=[]
            message.append(data)
            record_data.append([date,time,author,message])
    else:
        message.append(line)
```

Dataframe creation

In [242]:

```
data=pd.DataFrame(record_data,columns=['Date','Time','Author','Message'])
data['Message']=data['Message'].apply(lambda x: " ".join(x))
data.dropna()
data.head()
```

Out[242]:

	Date	Time	Author	Message
0	27/01/2018	09:59	None	You created group "IPL 2019"
1	16/03/2019	20:59	None	Sai added you
2	16/03/2019	20:59	None	You changed the group description
3	16/03/2019	21:00	Singu	✓
4	16/03/2019	21:02	None	Mani changed the group description

Now lets group the data to find more stats

In [243]:

```
def emoji_check(data):
    emoji_list=[]
    for word in data:
        if any (char in emoji.UNICODE_EMOJI for char in word):
            emoji_list.append(word)
    return emoji_list
total_messages=data.shape[0]
total_media=data[data['Message'].apply(lambda x: '<Media omitted>' in x)].shape[0]
data['Emoji']=data['Message'].apply(emoji_check)
total_emoji=sum(data['Emoji'].str.len())

print("Group Wise Stats")
print("Total Messages",total_messages)
print("Total Media",total_media)
print("Total Emoji",total_emoji)
```

```
Group Wise Stats
Total Messages 14154
Total Media 1863
Total Emoji 5588
```

Gather word count and letter count

In [248]:

```
data['Letters']=data['Message'].apply(lambda x: len(x))
data['Words']=data['Message'].apply(lambda x: len(x.split(" ")))
data.head(10)
```

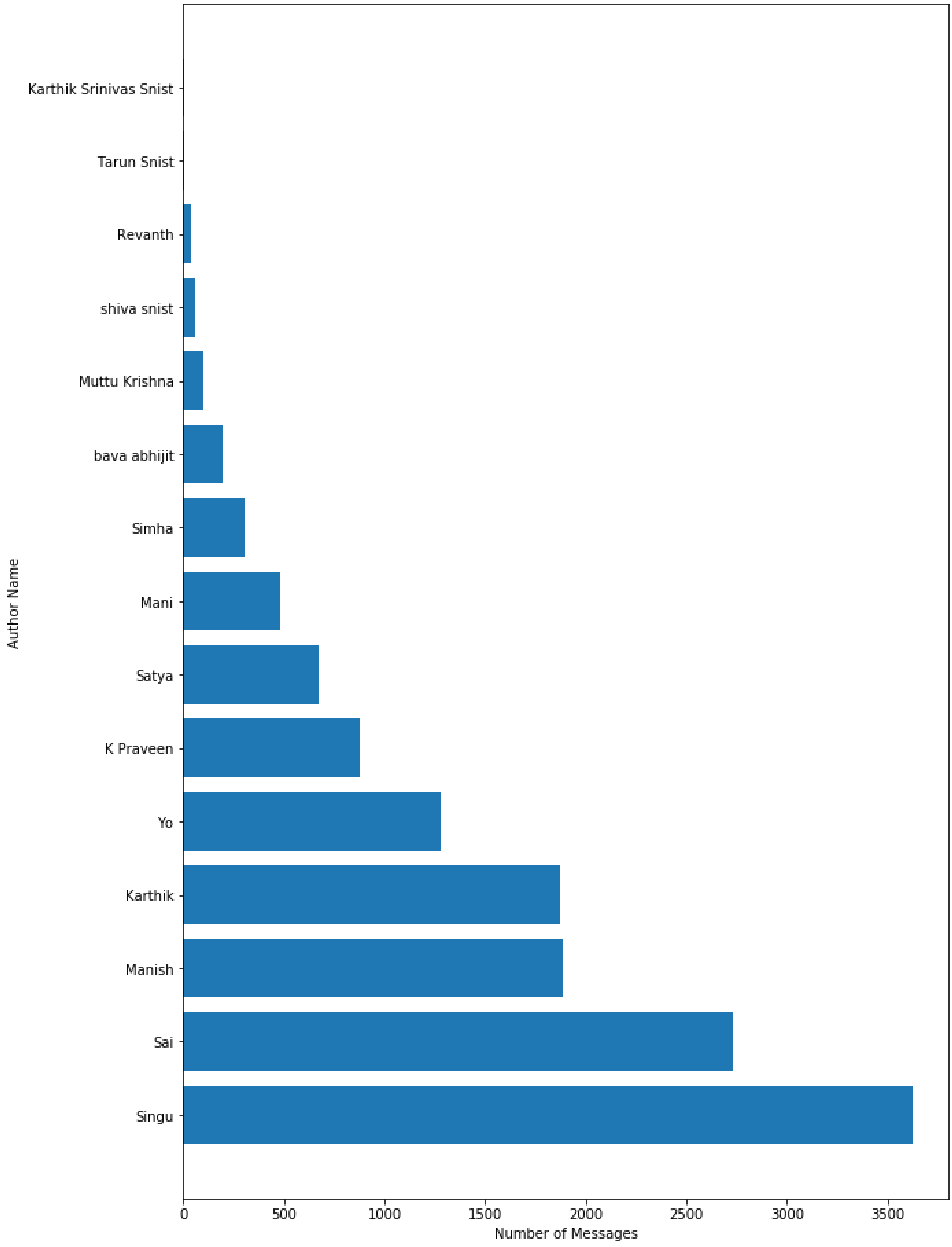
Out[248]:

	Date	Time	Author	Message	Emoji	Letters	Words
0	27/01/2018	09:59	None	You created group "IPL 2019"	[]	28	5
1	16/03/2019	20:59	None	Sai added you	[]	13	3
2	16/03/2019	20:59	None	You changed the group description	[]	33	5
3	16/03/2019	21:00	Singu	✓	[✓]	1	1
4	16/03/2019	21:02	None	Mani changed the group description	[]	34	5
5	16/03/2019	21:02	Mani	😅	[😅]	1	1
6	16/03/2019	21:03	None	Singu changed the group description	[]	35	5
7	16/03/2019	21:09	Karthik	<Media omitted>	[]	15	2
8	16/03/2019	21:13	None	You changed the group description	[]	33	5
9	16/03/2019	21:14	Karthik	Arey sunrisers jersey marindi	[]	29	4

Authors message count

In [253]:

```
authors_data=data.groupby('Author').count()['Message'].sort_values(ascending=False)
plt.figure(figsize=(10,16))
plt.barh(authors_data.index,list(authors_data))
plt.xlabel('Number of Messages')
plt.ylabel('Author Name')
plt.show()
```



Author wise stats

In [230]:

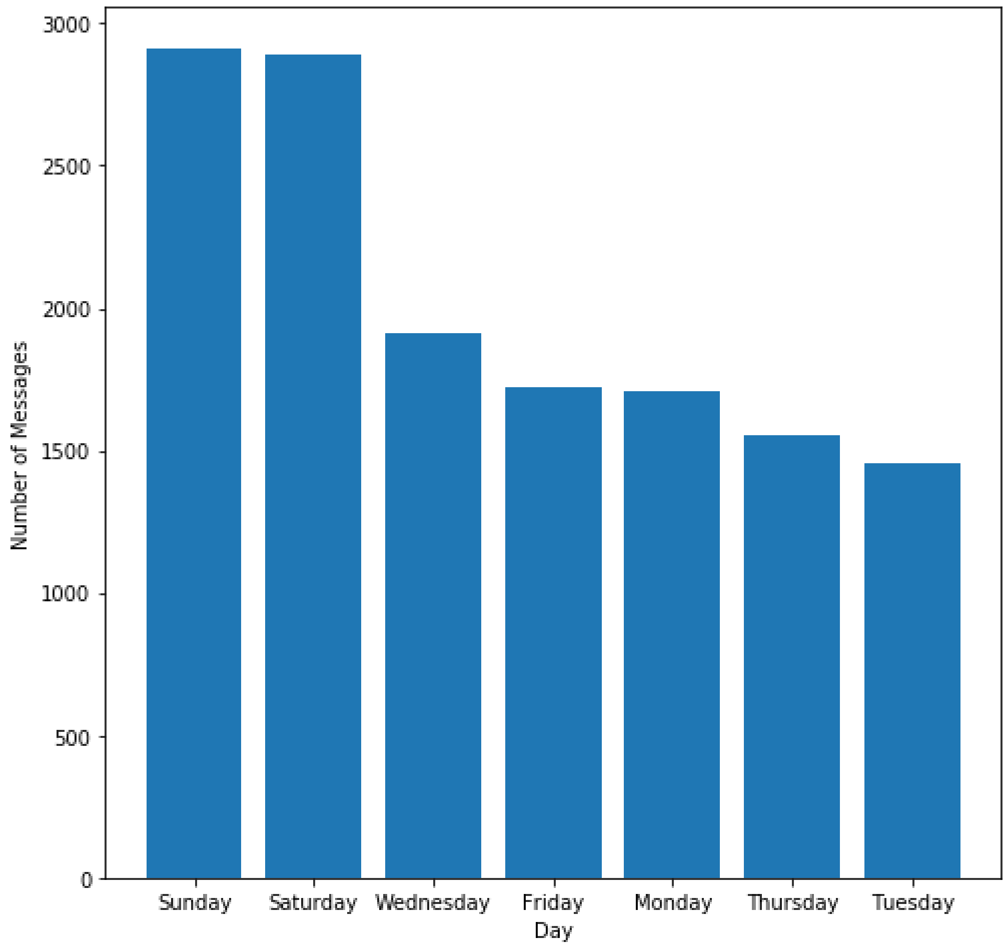
```python
author=list(data['Author'].unique())
#Ignoring None values
author=[aut for aut in author if aut is not None]
#Gathering Stats
for index in range(len(author)):
    new_data=data[data['Author']==author[index]]
    print("Stats related to "+author[index]+" -\n")
    num_messages=new_data.shape[0]
    print("Total Number of messages ",num_messages)
    avg_words_per_message=np.round(np.average(new_data['Words']),2)
    print("Average words per message ",avg_words_per_message)
    emoji_count=sum(new_data['Emoji'].str.len())
    print("Emoji Count ",emoji_count)
    media_count=new_data[new_data['Message'].apply(lambda x: '<Media omitted>' in x)].shape
    print("Media Count ",media_count)
    emoji_list=list([a for b in new_data['Emoji'] for a in b])
    emoji_dict=dict(collections.Counter(emoji_list))
    emoji_dict=sorted(emoji_dict.items(),key=lambda x:x[1], reverse=True)
    emoji_df=pd.DataFrame(emoji_dict,columns=['Emoji','Count'])
    if len(emoji_df) >0:
        common_emoji=emoji_df.iloc[0]['Emoji']
    else:
        common_emoji=None
    print("Most used Emoji ",common_emoji)
    print(" ")
    print("-----------------")
```

```
Most used Emoji

-----------------
Stats related to Simha -

Total Number of messages  301
Average words per message  6.25
Emoji Count  70
Media Count  16
Most used Emoji  🔥

-----------------
Stats related to Sai -

Total Number of messages  2732
Average words per message  4.5
Emoji Count  288
Media Count  623
Most used Emoji  😂
```

Analysis based on day

In [231]:

```
def day_check(text):
    day=datetime.strptime(text, '%d/%m/%Y').weekday()
    return calendar.day_name[day]

data['Day']=data['Date'].apply(day_check)
day_count=data.groupby('Day').count()['Message'].sort_values(ascending=False)
plt.figure(figsize=(8,8))
plt.bar(day_count.index,day_count)
plt.xlabel('Day')
plt.ylabel('Number of Messages')
plt.show()
```

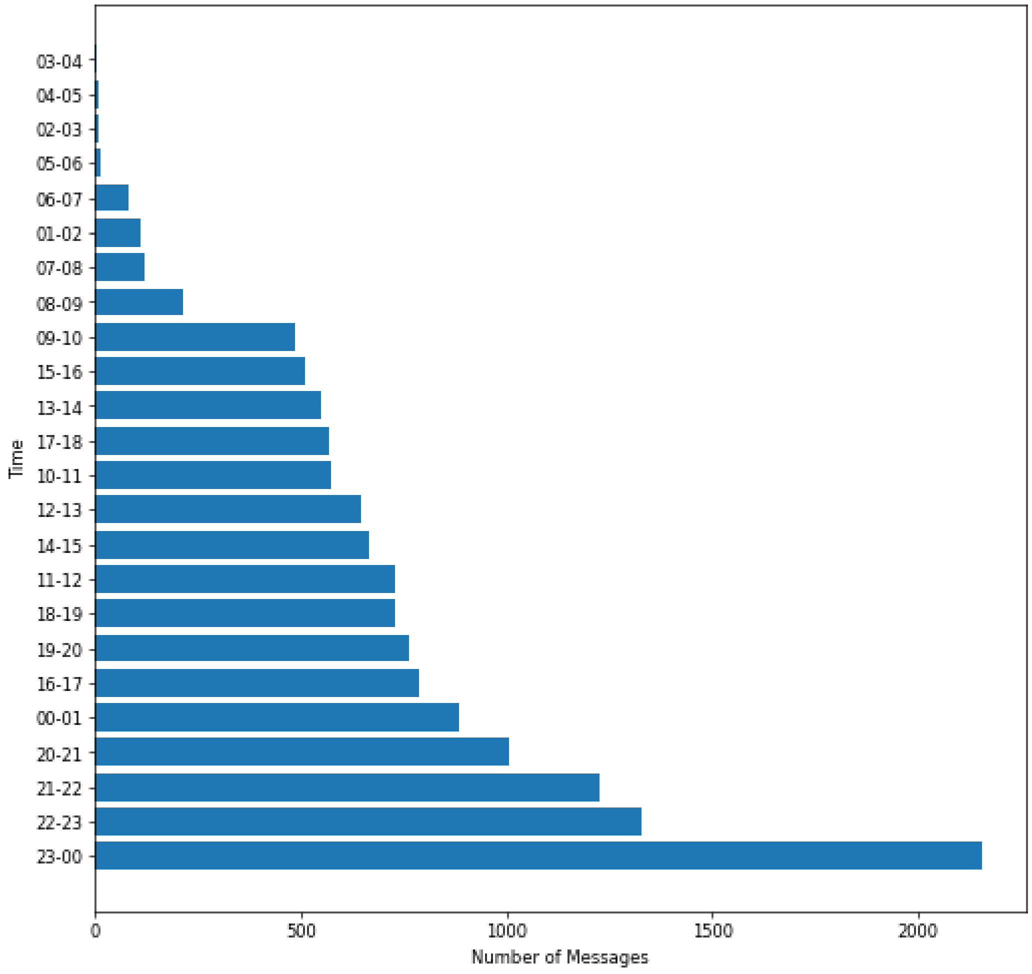


The above bar graph shows that most of the conversation happened during weekend

Analysis Based on Time

In [232]:

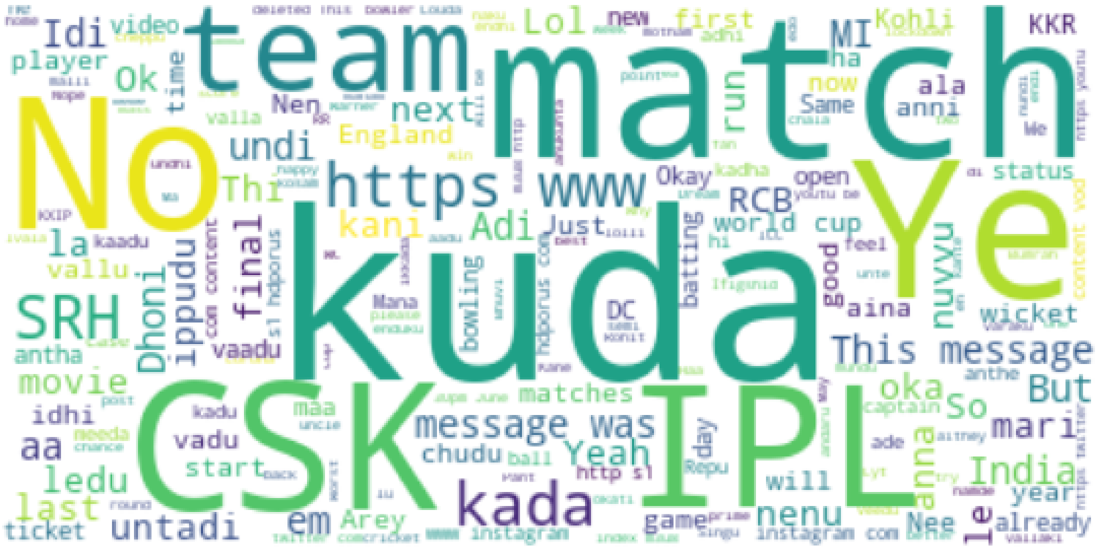
```
def time_check(text):
    time=datetime.strptime(text,'%d/%m/%Y %H:%M')
    hour=time.strftime('%H')
    return hour
def convert_interval(text):
    if str(text).startswith('0'):
        if(int(text[1])+1==10):
            interval=text+'-'+str(int(text[1])+1)
        else:
            interval=text+'-0'+str(int(text[1])+1)
    elif(text=='23'):
        interval=text+'-00'
    else:
        interval=text+'-'+str(int(text)+1)
    return interval

data['Time_Interval'] = (data['Date'] +" "+ data['Time']).apply(time_check)
data['Time_Interval'] = data['Time_Interval'].apply(convert_interval)

time_based_data=data.groupby('Time_Interval').count()['Message'].sort_values(ascending=Fals

plt.figure(figsize=(10,10),dpi=60)
plt.barh(time_based_data.index,time_based_data)
plt.xlabel('Number of Messages')
plt.ylabel('Time')
plt.show()
```



Word Cloud

In [236]:

```
new_data=data[data['Message'].apply(lambda x: '<Media omitted>' not in x)]
total_message=" ".join(list(new_data['Message']))
stopwords = set(STOPWORDS)
stopwords.update(['ki','ra','emo','ee','ga','inka','ah','na','tho','lo','deleted','was','ar
wordcloud = WordCloud(stopwords=stopwords, background_color='white').generate(total_message
plt.figure(figsize=(10,10))
plt.imshow(wordcloud,interpolation='bilinear')
plt.axis('off')
plt.show()
```



Hope you liked it !! See you again with another topic