whatsapp-chat-data-analysis

System Setup

List of all the python libraries that are required

- numpy
- pandas
- matplotlib
- seaborn
- wordcloud
- emoji

Run the following command to get all the listed python libraries

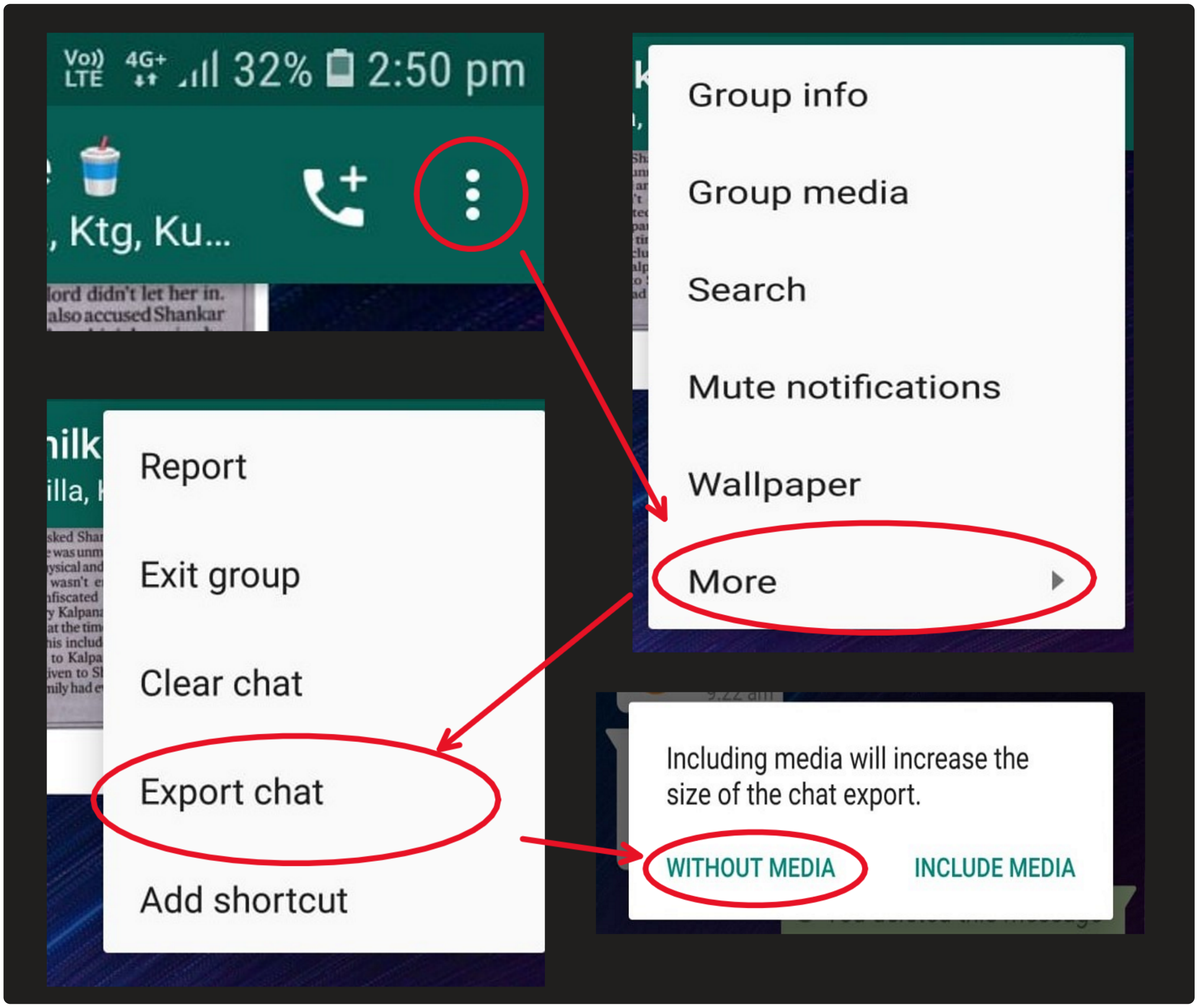
```
pip install numpy pandas matplotlib seaborn wordcloud emoji --upgrade
```

Check whether you have all the required libraries so the cell runs without any errors.

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

How to obtain Whatsapp Chat data

- Open whatsapp
- Open a Group/Inbox
- Click on the 3 dotted options button
- Click on more
- Click on export chat
- Click on without media
- Export via Email/other IM's/....
- Download to your system rename to chat-data.txt and put it in a folder



```
Without media: exports 40k messages
With media: exports 10k messages along with pictures/videos
As im are doing chat data analysis i went with `without media` option
```

Data Preprocessing

- Regex cheatsheet
  - https://www.rexegg.com/regex-quickstart.html
- Regex test - live
  - https://regexr.com/
- Datetime format
  - http://strftime.org/

Use a custom a regex and datatime format by reffering to the above links if you run into empty df or format errors. As the exports from whatsapp are not standardized.

```
def rawToDf(file, key):
    split_formats = {
        '12hr' : '\d{1,2}/\d{1,2}/\d{2,4},\s\d{1,2}:\d{2}\s[APap][mM]\s-\s',
        '24hr' : '\d{1,2}/\d{1,2}/\d{2,4},\s\d{1,2}:\d{2}\s-\s',
        'custom' : ''
    }
    datetime_formats = {
        '12hr' : '%d/%m/%Y, %I:%M %p - ',
        '24hr' : '%d/%m/%Y, %H:%M - ',
        'custom': ''
    }

    with open(file, 'r') as raw_data:
        raw_string = ' '.join(raw_data.read().split('\n')) # converting the list sp
        user_msg = re.split(split_formats[key], raw_string) [1:] # splits at all th
        date_time = re.findall(split_formats[key], raw_string) # finds all the date

        df = pd.DataFrame({'date_time': date_time, 'user_msg': user_msg}) # exporti

    # converting date-time pattern which is of type String to type datetime,
    # format is to be specified for the whole string where the placeholders are ext
    df['date_time'] = pd.to_datetime(df['date_time'], format=datetime_formats[key])

    # split user and msg
    usernames = []
    msgs = []
    for i in df['user_msg']:
        a = re.split('([\w\W]+?):\s', i) # lazy pattern match to first {user_name}:
        if(a[1:]): # user typed messages
            usernames.append(a[1])
            msgs.append(a[2])
        else: # other notifications in the group(eg: someone was added, some left .
            usernames.append("grp_notif")
            msgs.append(a[0])

    # creating new columns
    df['user'] = usernames
    df['msg'] = msgs

    # dropping the old user_msg col.
    df.drop('user_msg', axis=1, inplace=True)

    return df
```

Import data

```
df = rawToDf('chat-data.txt', '12hr')
```

```
df.tail()
```

	date_time	user	msg
39994	2019-07-22 20:42:00	Nikil DB	Konege playing 11...full change aagogiratte
39995	2019-07-22 21:55:00	Sandesh..!!	Aadodha naale
39996	2019-07-22 22:17:00	Sri Hari Colle	<Media omitted>
39997	2019-07-22 22:17:00	Sandesh..!!	Lol
39998	2019-07-22 22:18:00	Sandesh..!!	Always the personal reasons

```
df.shape # no. of msgs
```

```
(39999, 3)
```

```
me = "Prajwal Prashanth"
```

Data Cleaning

```
images = df[df['msg']=="<Media omitted> "] #no. of images, images are represented b
images.shape
```

```
(860, 3)
```

```
df["user"].unique()
```

```
array(['Sandesh..!!', 'Sri Hari Colle', 'Prajwal Prashanth', 'Venkat',
       '+91 98863 53469', 'Nikil DB', 'Ktg', 'Billa', 'manish lakshman',
       'Kushal Ramakanth', 'Keshava', 'Abhishek Dharani', 'grp_notif',
       'Srinidhi Nie', 'Kranti Jio', 'Prajwal Kaaadi'], dtype=object)
```

```
grp_notif = df[df['user']=="grp_notif"] #no. of grp notifications
grp_notif.shape
```

```
(41, 3)
```

```
df.drop(images.index, inplace=True) #removing images
df.drop(grp_notif.index, inplace=True) #removing grp_notif
```

```
df.tail()
```

	date_time	user	msg

39993	2019-07-22 20:34:00	Venkat	Neen bandhilla antha kudililla
39994	2019-07-22 20:42:00	Nikil DB	Konege playing 11...full change aagogiratte
39995	2019-07-22 21:55:00	Sandesh..!!	Aadodha naale
39997	2019-07-22 22:17:00	Sandesh..!!	Lol
39998	2019-07-22 22:18:00	Sandesh..!!	Always the personal reasons

```
df.reset_index(inplace=True, drop=True)
df.shape
```

```
(39098, 3)
```

Lets Discuss on what do we want to get out of this data

```
* Is raw data enough to get that insight?
* if not what can be possible way to get that insight?
* Whats the use of that insight?
```

Questions from the audience

Q 1) Who is the most active member of the group. Who is the least active?

```
df.groupby("user")["msg"].count().sort_values(ascending=False)
```

```
user
Sandesh..!!          9257
Sri Hari Colle       9138
Venkat               5259
Nikil DB             4977
Prajwal Prashanth    4383
Billa                1762
Ktg                  1436
manish lakshman      1297
Abhishek Dharani      587
Kushal Ramakanth      342
Prajwal Kaaadi        191
Kranti Jio            182
Srinidhi Nie          103
Keshava                94
+91 98863 53469        90
Name: msg, dtype: int64
```

Q 2) Count of all the emojis that i have used?

```python
emoji_ctr = Counter()
emojis_list = map(lambda x: ''.join(x.split()), emoji.UNICODE_EMOJI.keys())
r = re.compile('|'.join(re.escape(p) for p in emojis_list))
for idx, row in df.iterrows():
    if row["user"] == me:
        emojis_found = r.findall(row["msg"])
        for emoji_found in emojis_found:
            emoji_ctr[emoji_found] += 1
```

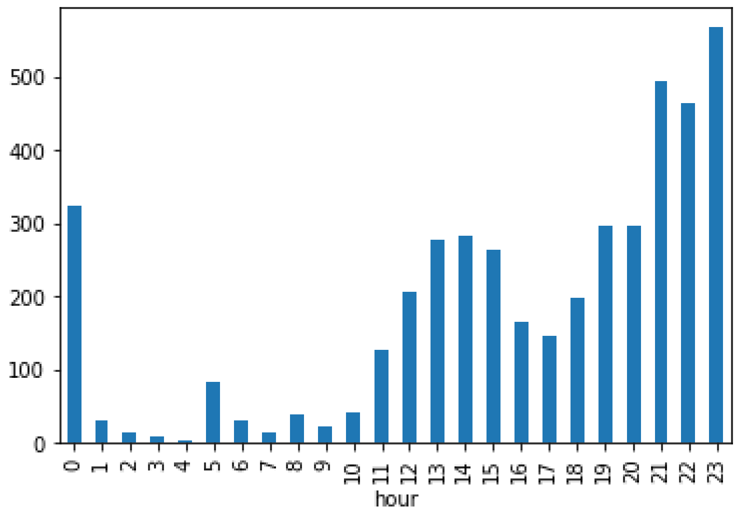
```python
for item in emoji_ctr.most_common(10):
    print(item[0] + " - " + str(item[1]))
```

```
😂 - 74
🏻 - 30
😢 - 22
✌ - 18
👎 - 18
👍 - 15
😶 - 4
😭 - 3
🏼 - 3
😅 - 2
```

Q 3) What can my activity say about my sleep cycle?

```python
df['hour'] = df['date_time'].apply(lambda x: x.hour)
df[df['user']==me].groupby(['hour']).size().sort_index()
.plot(x="hour", kind='bar')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8631472310>
```



Q 4)

What is the difference in Weekend vs Weekday usage pattern?

How many words do I type on average on weekday vs weekend?

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DatetimeIndex.weekday.html

```
df['weekday'] = df['date_time'].apply(lambda x: x.day_name()) # can use day_name or
```

```
df['is_weekend'] = df.weekday.isin(['Sunday', 'Saturday'])
```

```
msgs_per_user = df['user'].value_counts(sort=True)
msgs_per_user
```

```
Sandesh..!!          9257
Sri Hari Colle       9138
Venkat               5259
Nikil DB             4977
Prajwal Prashanth    4383
Billa                1762
Ktg                  1436
manish lakshman      1297
Abhishek Dharani      587
Kushal Ramakanth      342
Prajwal Kaaadi        191
Kranti Jio            182
Srinidhi Nie          103
Keshava                94
+91 98863 53469        90
Name: user, dtype: int64
```

```
top5_users = msgs_per_user.index.tolist()[:5]
top5_users
```

```
['Sandesh..!!', 'Sri Hari Colle', 'Venkat', 'Nikil DB', 'Prajwal Prashanth']
```

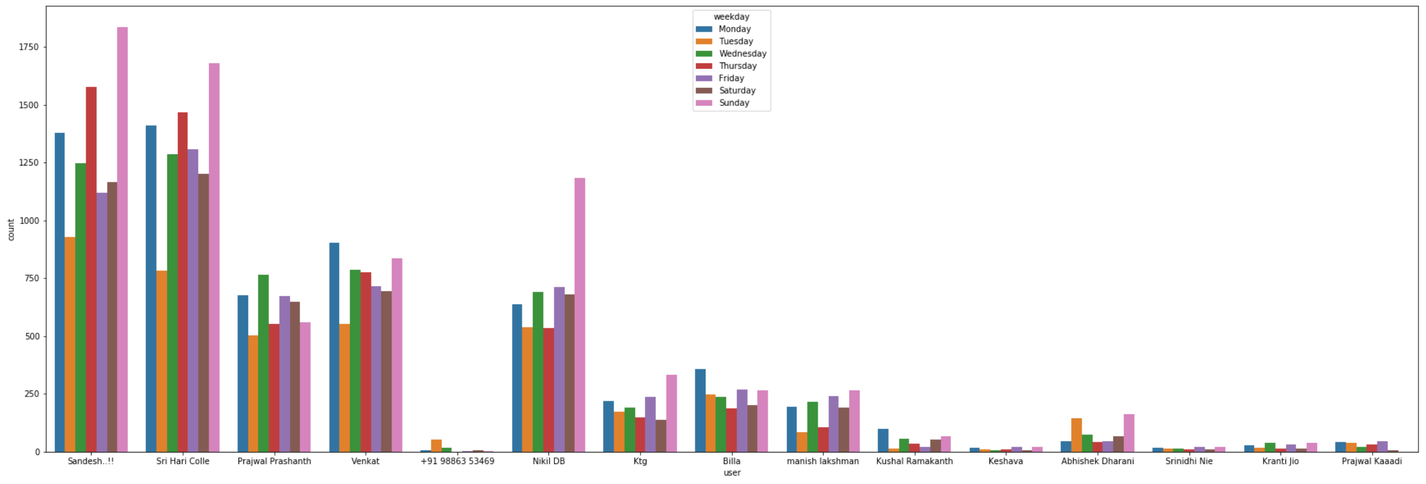
```
df_top5 = df.copy()
df_top5 = df_top5[df_top5.user.isin(top5_users)]
df_top5.head()
```

	date_time	user	msg	weekday	is_weekend	hour
0	2018-05-14 21:07:00	Sandesh..!!	Lo	Monday	False	21

0	2018-05-14 21:07:00	Sandesh..!!	Lo	Monday	False	21
1	2018-05-14 21:07:00	Sandesh..!!	Inna srh and re melidhe	Monday	False	21
2	2018-05-14 21:07:00	Sandesh..!!	Loude	Monday	False	21
3	2018-05-14 21:08:00	Sri Hari Colle	Run rate maga key	Monday	False	21
4	2018-05-14 21:08:00	Prajwal Prashanth	90 ge all out madbeku	Monday	False	21

```
plt.figure(figsize=(30,10))
sns.countplot(x="user", hue="weekday", data=df)
```

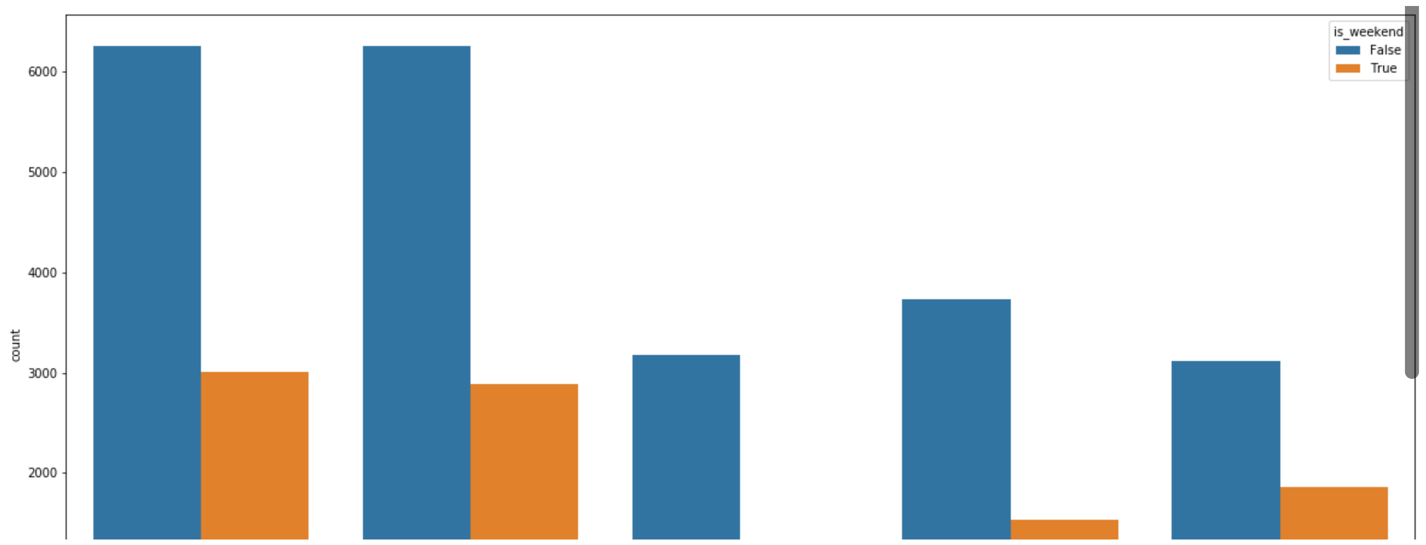
```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff51e2a8190>
```



```
df_top5['is_weekend'] = df_top5.weekday.isin(['Sunday', 'Saturday'])
```

```
plt.figure(figsize=(20,10))
sns.countplot(x="user", hue="is_weekend", data=df_top5)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff51c7cd610>
```



```python
def word_count(val):
    return len(val.split())
```

```python
df['no_of_words'] = df['msg'].apply(word_count)
```

```python
df_top5['no_of_words'] = df_top5['msg'].apply(word_count)
```

```python
total_words_weekday = df[df['is_weekend']==False]['no_of_words'].sum()
total_words_weekday
```

```
91889
```

```python
total_words_weekend = df[df['is_weekend']]['no_of_words'].sum()
total_words_weekend
```

```
41129
```

```python
total_words_weekday/5 # average words on a weekday
```

```
18377.8
```

```python
total_words_weekend/2 # average words on a weekend
```

```
20564.5
```

```python
df.groupby('user')['no_of_words'].sum().sort_values(ascending=False)
```

```
user
Sandesh..!!          32234
Sri Hari Colle       27111
Venkat               20728
```

```
Prajwal Prashanth    17724
Nikil DB             16901
Billa                 4852
manish lakshman       4203
Ktg                   3701
Abhishek Dharani      2001
Kushal Ramakanth      1331
Prajwal Kaaadi         764
Kranti Jio             516
+91 98863 53469        447
Srinidhi Nie           287
Keshava                218
Name: no_of_words, dtype: int64
```

```
(df_top5.groupby('user')['no_of_words'].sum()/df_top5.groupby('user').size()).sort_v
```

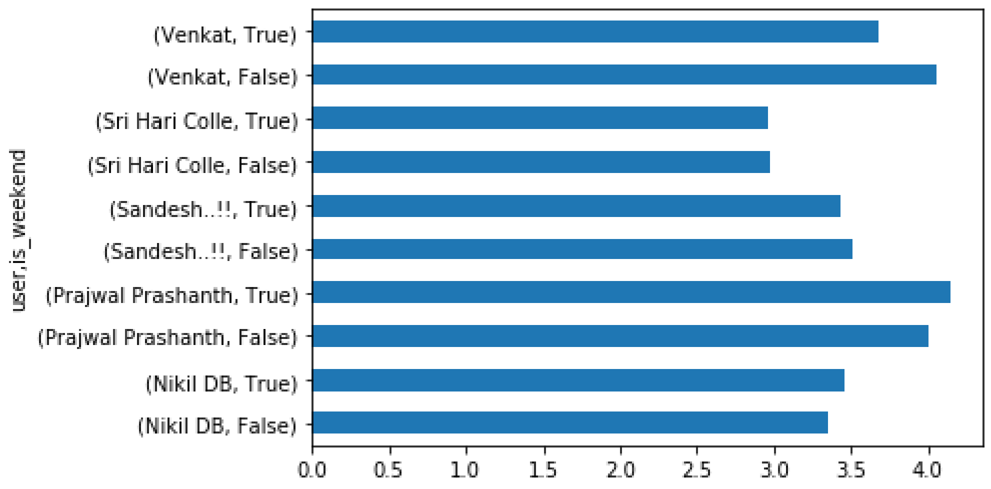
```
user
Prajwal Prashanth    4.043806
Venkat               3.941434
Sandesh..!!          3.482122
Nikil DB             3.395821
Sri Hari Colle       2.966842
dtype: float64
```

```
wordPerMsg_weekday_vs_weekend = (df_top5.groupby(['user', 'is_weekend'])['no_of_word
wordPerMsg_weekday_vs_weekend
```

```
user               is_weekend
Nikil DB           False          3.359782
                   True           3.456009
Prajwal Prashanth  False          4.004094
                   True           4.148179
Sandesh..!!        False          3.507355
                   True           3.429570
Sri Hari Colle     False          2.969789
                   True           2.960444
Venkat             False          4.049866
                   True           3.676913
dtype: float64
```

```
wordPerMsg_weekday_vs_weekend.plot(kind='barh')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff51c51b710>
```



Q 5)

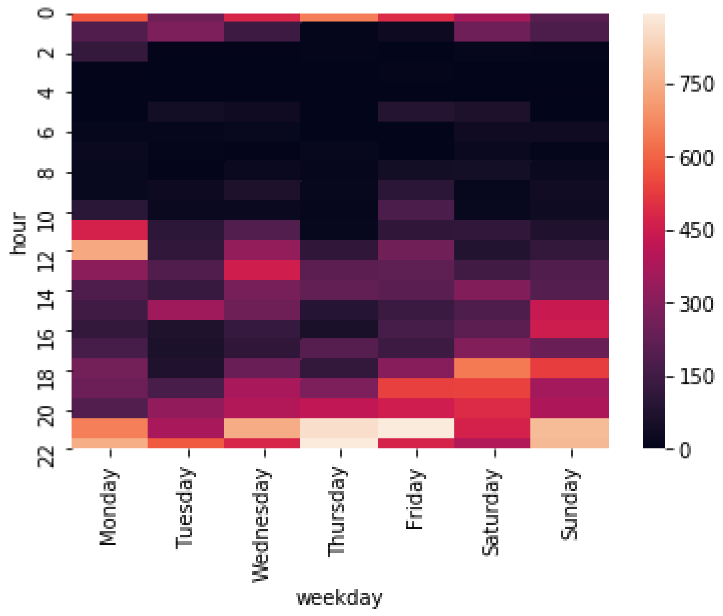
Most Usage - Time of Day

```
x = df.groupby(['hour', 'weekday'])['msg'].size().reset_index()
x2 = x.pivot("hour", 'weekday', 'msg')
x2.head()
```

weekday	Friday	Monday	Saturday	Sunday	Thursday	Tuesday	Wednesday
hour							
0	494.0	578.0	367.0	206.0	650.0	248.0	478.0
1	30.0	188.0	253.0	181.0	9.0	286.0	144.0
2	3.0	124.0	13.0	7.0	8.0	5.0	6.0
3	8.0	5.0	NaN	NaN	1.0	1.0	1.0
4	1.0	2.0	NaN	5.0	1.0	2.0	1.0

```
days = ["Monday", 'Tuesday', "Wednesday", "Thursday", "Friday", "Saturday", "Sunday'
sns.heatmap(x2[days].fillna(0), robust=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff51c4afe10>
```



Q 6)

In any group, do I have any inclination towards responding to someone?

```
my_msgs_index = np.array(df[df['user']==me].index)
print(my_msgs_index, my_msgs_index.shape)
```

```
[    4     5    11 ... 39073 39076 39077] (4383,)
```

```
prev_msgs_index = my_msgs_index - 1
print(prev_msgs_index, prev_msgs_index.shape)
```

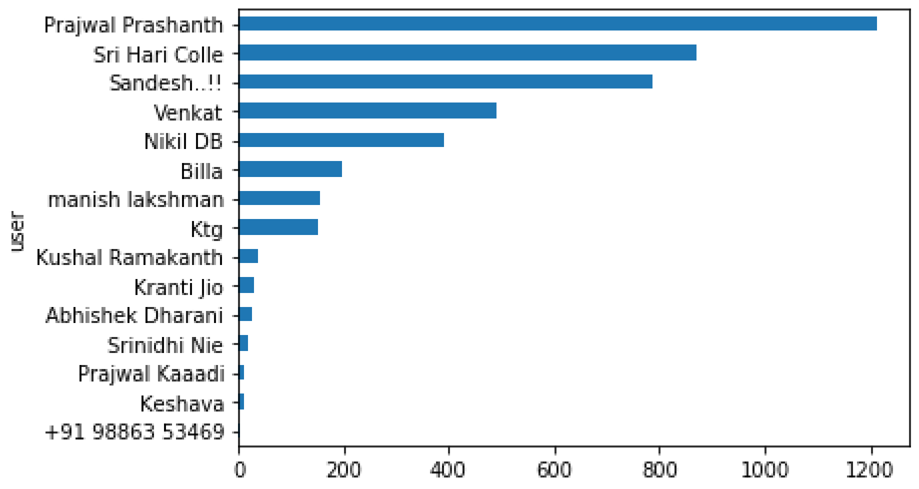
```
[    3     4    10 ... 39072 39075 39076] (4383,)
```

```
df_replies = df.iloc[prev_msgs_index].copy()
df_replies.shape
```

```
(4383, 7)
```

```
df_replies.groupby(["user"])["msg"].size().sort_values().plot(kind='barh')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff51c3bda10>
```

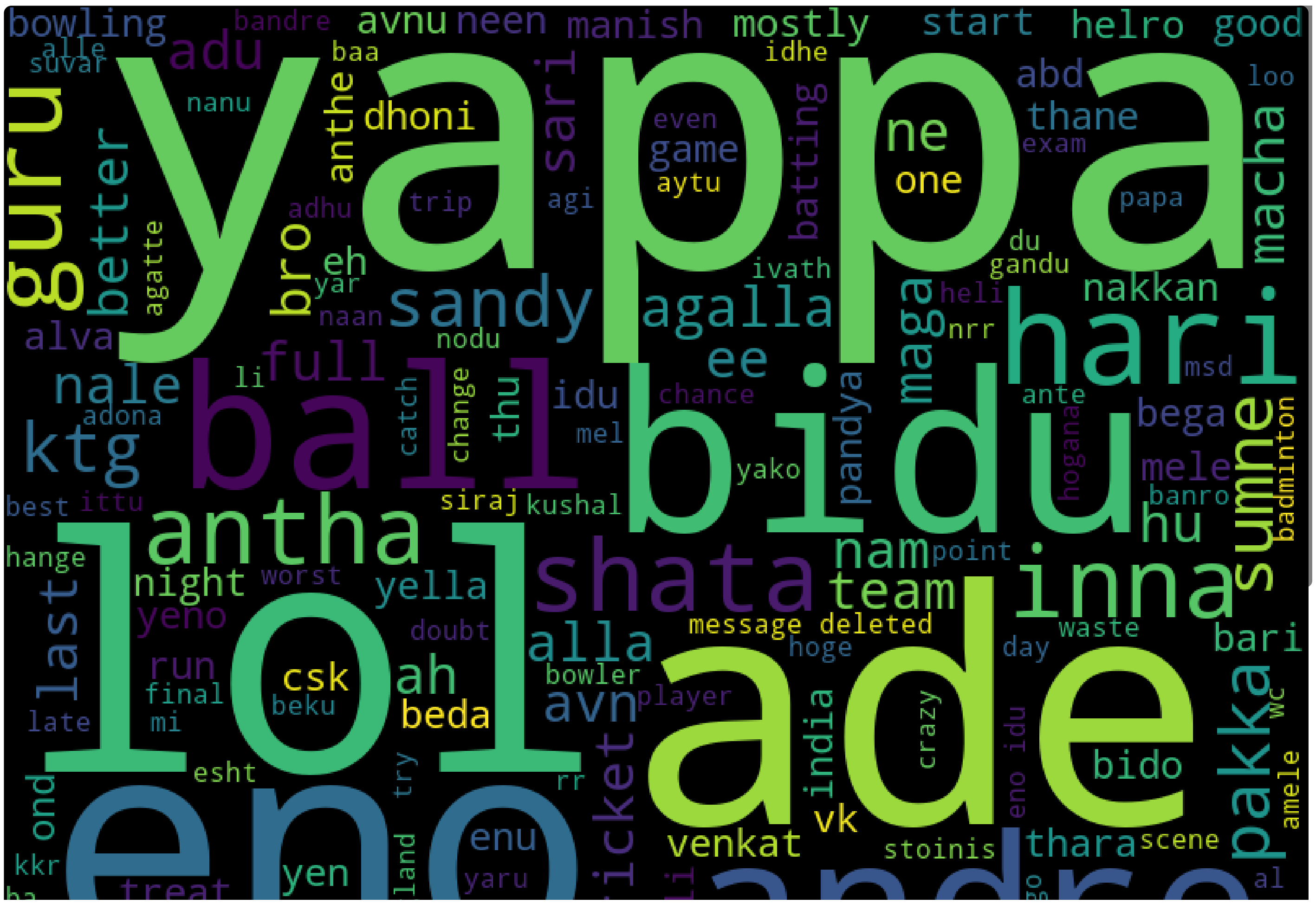


Q 7)

Which are the most common words?

```
comment_words = ' '
stopwords = STOPWORDS.update(['lo', 'ge', 'Lo', 'illa', 'yea', 'ella', 'en', 'na',

for val in df.msg.values:
    val = str(val)
    tokens = val.split()

    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    for words in tokens:
        comment_words = comment_words + words + ' '


wordcloud = WordCloud(width = 800, height = 800,
                background_color ='black',
                stopwords = stopwords,
                min_font_size = 10).generate(comment_words)
```

```
wordcloud.to_image()
```


yappa
bidu
lol
ade
ball
eno
hari
antha
inna
shata
guru
sandy
agalla
ktg
nale
better
bowling
adu
avnu
neen
manish
mostly
start
helro
good
dhoni
ne
one
game
thane
abd
macha
sumne
pakka
team
nam
alla
avn
csk
ah
beda
bido
venkat
vk
yella
night
last
run
full
ee
maga
nakkan
pandya
batting
sari
anthe
bro
eh
alva
bega
mele
hu
bari
india
enu
yen
thara
scene
stoinis
message deleted
treat
yeno
idu
thu
change
catch
siraj
kushal
point
waste
day
hoge
bowler
player
doubt
final
beku
late
mi
esht
ond
kkr
rr
yaru
land
try
crazy
eno idu
amele
wc
badminton
banro
hogana
ante
msd
nrr
heli
gandu
du
ivath
papa
exam
loo
aytu
even
agi
trip
adhu
yar
naan
nodu
chance
mel
yako
li
adona
best
ittu
hange
worst
agatte
nanu
suvar
alle
bandre
baa
idhe
al