Exploratory Analysis On IPL2022 Tweets

Exploratory Data Analysis(EDA) is the porcess of exploring, investigating and gathering insight from data using statistical meassures and visualizations. The objective of EDA is to develop and understanding of data, by uncocovering trends, relationships and patterns. EDA is both a science and an art. On the one hand it requires the knowledge of statistics, visualization techniques and data analysis tools like Numpy, Pandas, Seaborn etc. On the other hand, it requires asking interesting questions to guide the investigations and interpreting numbers and figures to generate useful insights.

In this project, I have selected an IPL 2022 tweets dataset from kaggle to explore and analyze the sites. We'll use the the python libraries pandas, matplotlib, seaborn and plotly to do exploratory data analysis on the dataset.

1 Downloading a dataset from kaggle an online source.

2 Data preparation and cleaning with pandas.

3 Open-ended exploratory analysis and visualization.

4 Asking and answering interesting questions.

5 Summarizing inferences and drawing conclusions

This data consists of the tweets with the trending #ipl2022 hashtags made by the fans of cricket

The data is extracted using TwitterAPI and a python script! https://www.kaggle.com/kaushiksuresh147/twitter-data-extraction-for-ipl2020. The data will be updated on a daily basis.The data consists of 3 years of tweets made by fans during the IPL seasons 2020, 2021, and 2022

Here in EDA analysis we are going to select the data for the year 2022. The data consists of 13 columns and 574,664 rows. The columns are-

- 'user_name'
- 'user_location'
- 'user_description'
- 'user_created'
- 'user_followers'
- 'user_friends'
- 'user_favourites'
- 'user_verified'
- 'date'
- 'text'
- 'source'
- 'hastags' -'is_retweets'

How to run the code

The easiest way to start executing the code is to click the Run button at the top of this page and select Run on Binder. You can also select "Run on Colab" or "Run on Kaggle", but you'll need to create an account on Google

Colab or Kaggle to use these platforms. You can make changes and save your own version of the notebook to Jovian by executing the following cells.

Since the selected dataset contains 5+ million rows of data, I have selected "Gogle Colab" to execute the code for faster response.

When you are commiting the notebook to Jovian for the first time in "Colab" it will ask for API key which will be found in your Jovian account getstarted section.

Installing the required packages

In this project, we'll use data analysis tools like Numpy, Pandas and visulization tools like matplotlib, seaborn, plotly and folium.

let's install the required libraries and import them.

```
print('Hello World')
```

```
Hello World
```

```
!pip install numpy==1.24.1 pandas==1.1.5 wordcloud jovian opendatasets matplotlib seabo
```

```
import jovian
```

Downloading a dataset from kaggle an online source

```
import opendatasets as od
```

```
dataset_url = 'https://www.kaggle.com/datasets/kaushiksuresh147/ipl2020-tweets'
```

```

```

```
od.download(dataset_url)
```

```
Skipping, found downloaded files in "./ipl2020-tweets" (use force=True to force
download)
```

```
import os
```

```
data_dir ='ipl2020-tweets'
```

```
os.listdir(data_dir)
```

```
['IPL2020_Tweets.csv', 'IPL_2022_tweets.csv', 'IPL_2021_tweets.csv']
```

Here we have three data of ipl(year 2021, year 2021 and year 2022) are availble but we are selectiing the data of year 2022 because it is latest data and want to work on latest raw data

Data preparation and cleaning with pandas

- load the file using pandas
- look and observe the information about data which is in coloumns and rows
- fix or clean any missing or incorrect values

```
import pandas as pd
```

```
survey_raw_df = pd.read_csv(data_dir + '/IPL_2022_tweets.csv')
```

```
/usr/local/lib/python3.8/dist-packages/IPython/core/interactiveshell.py:3057:
DtypeWarning: Columns (5,6,7,12) have mixed types.Specify dtype option on import or set
low_memory=False.
  has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
```

```
survey_raw_df
```

	user_name	user_location	user_description	user_created	user_followers	user_friends
0	World Cricket Baba	india	! Links Baba:\nhttps://t.co/L3bxQ4jPHK\nhttps:...	2022-04-13 06:34:29	1076.0	63
1	The Times Of India	New Delhi	News. Views. Analysis. Conversations. India's ...	2010-04-19 10:50:15	14429584.0	457
2	World Cricket Baba	india	! Links Baba:\nhttps://t.co/L3bxQ4jPHK\nhttps:...	2022-04-13 06:34:29	1076.0	63
3	Social Animal	India	I'm here to avoid my friends on Facebook.	2013-10-15 04:34:14	124.0	502
4	World Cricket Baba	india	! Links Baba:\nhttps://t.co/L3bxQ4jPHK\nhttps:...	2022-04-13 06:34:29	1076.0	63
...	...	...	...	...	...	...
574659	Rohit Sharma FC	NaN	Hii This is an Die Heart Fan Club of Rohit Sharma	2021-08-02 04:05:06	3.0	11
574660	Sanket Pandey	India	Proud to be an indian\n-Jay hind.\n-Vande matr...	2017-01-11 13:44:24	14.0	333

	user_name	user_location	user_description	user_created	user_followers	user_friends
574661	InsideSport	New Delhi, India	Official website of InsideSport - India's prem...	2017-01-21 11:03:22	5654.0	759
574662	Deepanshu Speaks	India	Offical Twitter handle of YouTube channel #Dee...	2021-06-17 14:44:22	11.0	42
574663	Aakash Srivastava	Noida, India	Writer with @SportsTiger	2016-09-10 08:31:12	103.0	2022

574664 rows × 13 columns

- we are checking it's column and it's name by column method

```
survey_raw_df.columns
```

```
Index(['user_name', 'user_location', 'user_description', 'user_created',
       'user_followers', 'user_friends', 'user_favourites', 'user_verified',
       'date', 'text', 'hashtags', 'source', 'is_retweet'],
      dtype='object')
```

```

```

- let's check it has how many total rows and column in dataframe

```
survey_raw_df.shape
```

```
(574664, 13)
```

- By using info() method we will find out how many non null and its Dtype . if it has not proper Dtype accourding to the data filled in the columns then will change it in next process in data cleaning

```
survey_raw_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 574664 entries, 0 to 574663
Data columns (total 13 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   user_name         574664 non-null  object 
 1   user_location     384107 non-null  object 
 2   user_description  492538 non-null  object 
 3   user_created      574655 non-null  object 
 4   user_followers    574649 non-null  float64
 5   user_friends      574649 non-null  object 
 6   user_favourites   574649 non-null  object 
 7   user_verified     574649 non-null  object 
```

```
 8   date              574649 non-null  object
 9   text              574649 non-null  object
 10  hashtags          574620 non-null  object
 11  source            574640 non-null  object
 12  is_retweet        574640 non-null  object
dtypes: float64(1), object(12)
memory usage: 57.0+ MB
```

Exploratory Analysis

Here we will apply describe() method to check it statical values

```
survey_raw_df.describe()
```

	user_followers
count	5.746490e+05
mean	1.120746e+05
std	7.588829e+05
min	0.000000e+00
25%	3.900000e+01
50%	2.030000e+02
75%	1.142000e+03
max	2.011137e+07

- In above describe method is giving result only for user_followers because it has float dtype and other coloum is object. But when we observe the survey_raw_df carefully we find that user_friends and user_favourite is int dtype but it is mentioned object type so we need to change it from object to int type.
- Some observations like user_created and date columns are mentioned object dtype but it should be date type data. Therefore we need to change it from object to datetimestamp

We don't want to disturb survey raw data therfore we are creating copy of raw data of survey for analysis and let's give its name ipl_2022_df

```
ipl_2022_df = survey_raw_df.copy()
```

```
ipl_2022_df
```

	user_name	user_location	user_description	user_created	user_followers	user_friends
0	World Cricket Baba	india	! Links Baba:\nhttps://t.co/L3bxQ4jPHK\nhttps:...	2022-04-13 06:34:29	1076.0	63
1	The Times Of India	New Delhi	News. Views. Analysis. Conversations. India's ...	2010-04-19 10:50:15	14429584.0	457
2	World Cricket Baba	india	! Links Baba:\nhttps://t.co/L3bxQ4jPHK\nhttps:...	2022-04-13 06:34:29	1076.0	63
3	Social Animal	India	I'm here to avoid my friends on Facebook.	2013-10-15 04:34:14	124.0	502
4	World Cricket Baba	india	! Links Baba:\nhttps://t.co/L3bxQ4jPHK\nhttps:...	2022-04-13 06:34:29	1076.0	63
...	...	...	...	...	...	...
574659	Rohit Sharma FC	NaN	Hii This is an Die Heart Fan Club of Rohit Sharma	2021-08-02 04:05:06	3.0	11

	user_name	user_location	user_description	user_created	user_followers	user_friends
574660	Sanket Pandey	India	Proud to be an indian\n-Jay hind.\n-Vande matr...	2017-01-11 13:44:24	14.0	333
574661	InsideSport	New Delhi, India	Official website of InsideSport - India's prem...	2017-01-21 11:03:22	5654.0	759
574662	Deepanshu Speaks	India	Offical Twitter handle of YouTube channel #Dee...	2021-06-17 14:44:22	11.0	42
574663	Aakash Srivastava	Noida, India	Writer with @SportsTiger	2016-09-10 08:31:12	103.0	2022

574664 rows × 13 columns

copy of survey raw data i.e. ipl_2022_df, it has samen number of rows and it's columns. let's check its rows and column using by shape methods

```
ipl_2022_df.shape
```

```
(574664, 13)
```

```
ipl_2022_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 574664 entries, 0 to 574663
Data columns (total 13 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   user_name         574664 non-null  object 
 1   user_location     384107 non-null  object 
 2   user_description  492538 non-null  object 
 3   user_created      574655 non-null  object 
 4   user_followers    574649 non-null  float64
 5   user_friends      574649 non-null  object 
 6   user_favourites   574649 non-null  object 
 7   user_verified     574649 non-null  object 
 8   date              574649 non-null  object 
 9   text              574649 non-null  object 
 10  hashtags          574620 non-null  object 
 11  source            574640 non-null  object 
 12  is_retweet        574640 non-null  object 
dtypes: float64(1), object(12)
memory usage: 57.0+ MB
```

above using info() method, we found that

- column date and user_created is object dtype but when we observe it in dataframe it is combination of date and time. it means it should be datetime dtype
- Similiarly coumns user_friends, user_favourites and user_followers are also mentioned object dtype but it is float dtype.
- column named is user_verified is also mentioned here object but when we observe the data it has two value yes or no. It means it should be boolean dtype

DataFrame's each column know it dtype by applying unique() method

```
ipl_2022_df['date'].unique()
```

```
array(['2022-06-20 22:00:03', '2022-06-20 21:30:00',
       '2022-06-20 20:00:24', ..., '2022-01-06 12:59:46',
       '2022-01-06 12:52:12', '2022-01-06 12:51:05'], dtype=object)
```

```
ipl_2022_df['user_created'].unique()
```

```
array(['2022-04-13 06:34:29', '2010-04-19 10:50:15',
       '2013-10-15 04:34:14', ..., '2022-01-06 08:48:51',
       '2021-09-24 01:22:04', '2017-01-11 13:44:24'], dtype=object)
```

```
ipl_2022_df['user_friends'].unique()
```

```
array([63.0, 457.0, 502.0, ..., 9722.0, 3435.0, 3912.0], dtype=object)
```

```
ipl_2022_df['user_verified'].unique()
```

```
array([False, True, nan, 'False', 'True',
       'Rinku singh is looking so confident &amp;amp; promising. Back him @KKRiders.
#KKRvsRR #IPL2022 #KKR https://t.co/EmxqaIXdyQ',
       'Witnessing some of the worst fielding and catch drops in #IPL2022. Something
abnormal.',
       '#IPL2022 Drinking Water Shortage in Wankhede today 22/4/2022. Outside Water not
allowed. Inside water not available. 1/2 https://t.co/AY1NKsT2uy'],
      dtype=object)
```

```
ipl_2022_df['user_favourites'].unique()
```

```
array([699.0, 6.0, 2675.0, ..., 103180.0, 14851.0, 51926.0], dtype=object)
```

```
ipl_2022_df['user_followers'].unique()
```

```
array([1.0760000e+03, 1.4429584e+07, 1.2400000e+02, ..., 7.5450000e+03,
       3.5306400e+05, 4.0615000e+04])
```

DataFrame columns like user_friends, user_favourites and user_followers are object type in DataFrame. now we are cleaning these column using pd.to_numeric method to change it's type for float dtype

```
ipl_2022_df['userFriends']= pd.to_numeric(ipl_2022_df.user_friends, errors= 'coerce')
ipl_2022_df['userFavourites']= pd.to_numeric(ipl_2022_df.user_favourites, errors= 'coer
ipl_2022_df['userFollowers']= pd.to_numeric(ipl_2022_df.user_followers, errors ='coerce
```

Column date and user_created is object type in raw data but it is date and time related data thererore it needs to change datetime from object dtype

```
ipl_2022_df['Date']= pd.to_datetime(ipl_2022_df.date, errors='coerce')
ipl_2022_df['userCreated']= pd.to_datetime(ipl_2022_df.user_created, errors = 'coerce')
```

column user_verified has to value yes and no therefore it should be also changed boolean type from object dtype

```
ipl_2022_df['User_Verified']= ipl_2022_df['user_verified'].astype(bool)
```

```

```

Now let's check ipl_2022_df. Here new columns has been added which was intially object dtype and after cleaning it has changed it's proper dtype

```
ipl_2022_df
```

	user_name	user_location	user_description	user_created	user_followers	user_friends
0	World Cricket Baba	india	! Links Baba:\nhttps://t.co/L3bxQ4jPHK\nhttps:...	2022-04-13 06:34:29	1076.0	63
1	The Times Of India	New Delhi	News. Views. Analysis. Conversations. India's ...	2010-04-19 10:50:15	14429584.0	457
2	World Cricket Baba	india	! Links Baba:\nhttps://t.co/L3bxQ4jPHK\nhttps:...	2022-04-13 06:34:29	1076.0	63
3	Social Animal	India	I'm here to avoid my friends on Facebook.	2013-10-15 04:34:14	124.0	502
4	World Cricket Baba	india	! Links Baba:\nhttps://t.co/L3bxQ4jPHK\nhttps:...	2022-04-13 06:34:29	1076.0	63
...	...	...	...	...	...	...
574659	Rohit Sharma FC	NaN	Hii This is an Die Heart Fan Club of Rohit Sharma	2021-08-02 04:05:06	3.0	11

	user_name	user_location	user_description	user_created	user_followers	user_friends
574660	Sanket Pandey	India	Proud to be an indian\n-Jay hind.\n-Vande matr...	2017-01-11 13:44:24	14.0	333
574661	InsideSport	New Delhi, India	Official website of InsideSport - India's prem...	2017-01-21 11:03:22	5654.0	759
574662	Deepanshu Speaks	India	Offical Twitter handle of YouTube channel #Dee...	2021-06-17 14:44:22	11.0	42
574663	Aakash Srivastava	Noida, India	Writer with @SportsTiger	2016-09-10 08:31:12	103.0	2022

574664 rows × 19 columns

After cleaning when we are applying describe() method we are getting statatical value which we have changed object dtype to it's proper dtype

```
ipl_2022_df.describe()
```

	user_followers	userFriends	userFavourites	userFollowers
count	5.746490e+05	574646.000000	5.746460e+05	5.746490e+05
mean	1.120746e+05	912.186482	1.929083e+04	1.120746e+05
std	7.588829e+05	2060.280243	4.538694e+04	7.588829e+05
min	0.000000e+00	0.000000	0.000000e+00	0.000000e+00
25%	3.900000e+01	104.000000	4.650000e+02	3.900000e+01
50%	2.030000e+02	364.000000	3.439000e+03	2.030000e+02
75%	1.142000e+03	1074.000000	1.606550e+04	1.142000e+03
max	2.011137e+07	350197.000000	1.236671e+06	2.011137e+07

Now we can see the columns and it Dtype where it has changed it's proper Dtype. we can check userFriends, userFollowers, userFavourites, Date, user_created and user_verified in below info() method

```
ipl_2022_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 574664 entries, 0 to 574663
Data columns (total 19 columns):
 #   Column            Non-Null Count   Dtype 
---  ------            --------------   ----- 
 0   user_name         574664 non-null  object
 1   user_location     384107 non-null  object
 2   user_description  492538 non-null  object
 3   user_created      574655 non-null  object
```

```
 4   user_followers    574649 non-null  float64
 5   user_friends      574649 non-null  object
 6   user_favourites   574649 non-null  object
 7   user_verified     574649 non-null  object
 8   date              574649 non-null  object
 9   text              574649 non-null  object
 10  hashtags          574620 non-null  object
 11  source            574640 non-null  object
 12  is_retweet        574640 non-null  object
 13  userFriends       574646 non-null  float64
 14  userFavourites    574646 non-null  float64
 15  userFollowers     574649 non-null  float64
 16  Date              574646 non-null  datetime64[ns]
 17  userCreated       574646 non-null  datetime64[ns]
 18  User_Verified     574664 non-null  bool
dtypes: bool(1), datetime64[ns](2), float64(4), object(12)
memory usage: 79.5+ MB
```

Find out the Null values in the columns of IPL2022 DataFrame

```
ipl_2022_df.isna().sum()
```

```
user_name              0
user_location       190557
user_description     82126
user_created             9
user_followers          15
user_friends            15
user_favourites         15
user_verified           15
date                    15
text                    15
hashtags                44
source                  24
is_retweet              24
userFriends             18
userFavourites          18
userFollowers           15
Date                    18
userCreated             18
User_Verified            0
dtype: int64
```

```
ipl_2022_df.isna().sum().sort_values(ascending=False)
```

```
user_location       190557
user_description     82126
```

```
hashtags                44
is_retweet              24
source                  24
userFavourites          18
userFriends             18
Date                    18
userCreated             18
userFollowers           15
date                    15
user_verified           15
user_favourites         15
user_friends            15
user_followers          15
text                    15
user_created             9
User_Verified            0
user_name                0
dtype: int64
```

```
len(ipl_2022_df)
```

```
574664
```

percentage of missing values in per columns

```
missing_percentage = (ipl_2022_df.isna().sum().sort_values(ascending=False) / len(ipl_2
missing_percentage
```

```
user_location       33.159725
user_description    14.291134
hashtags             0.007657
is_retweet           0.004176
source               0.004176
userFavourites       0.003132
userFriends          0.003132
Date                 0.003132
userCreated          0.003132
userFollowers        0.002610
date                 0.002610
user_verified        0.002610
user_favourites      0.002610
user_friends         0.002610
user_followers       0.002610
text                 0.002610
user_created         0.001566
User_Verified        0.000000
user_name            0.000000
dtype: float64
```

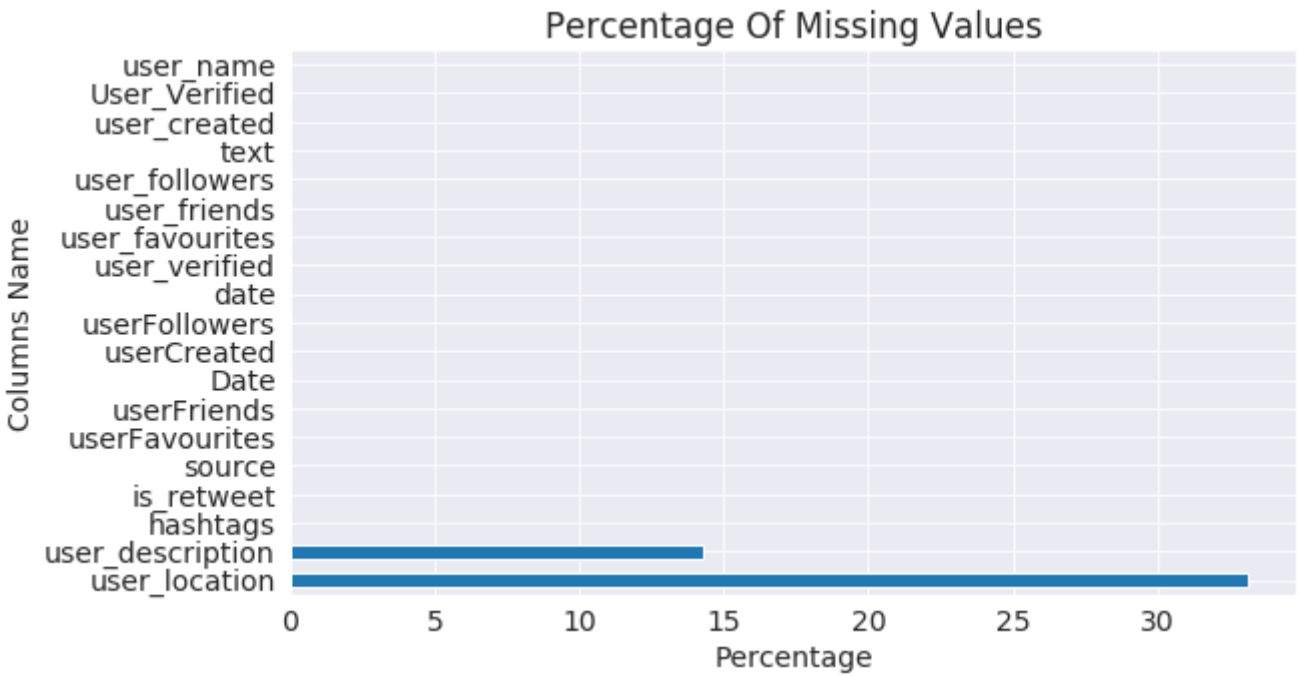
Visualization of percentage of Missing value

```
pip install matplotlib==3.1.3
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting matplotlib==3.1.3
  Using cached matplotlib-3.1.3-cp38-cp38-manylinux1_x86_64.whl (13.1 MB)
Requirement already satisfied: numpy>=1.11 in /usr/local/lib/python3.8/dist-packages (from matplotlib==3.1.3) (1.24.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.8/dist-packages (from matplotlib==3.1.3) (0.11.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib==3.1.3) (3.0.9)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib==3.1.3) (2.8.2)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib==3.1.3) (1.4.4)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.8/dist-packages (from python-dateutil>=2.1->matplotlib==3.1.3) (1.15.0)
Installing collected packages: matplotlib
  Attempting uninstall: matplotlib
    Found existing installation: matplotlib 3.6.2
    Uninstalling matplotlib-3.6.2:
      Successfully uninstalled matplotlib-3.6.2
Successfully installed matplotlib-3.1.3
```

```
import matplotlib.pyplot as plt
```

```
plt.title('Percentage Of Missing Values')
plt.xlabel('Percentage')
plt.ylabel('Columns Name')
missing_percentage.plot(kind= 'barh');
```



Above Graph indicates us that use location has more than 30 % missing value or we can say that more than 30% is null values. After that use desctiption is about(or near) 15% missing value.

```
ipl_2022_df.columns
```

```
Index(['user_name', 'user_location', 'user_description', 'user_created',
       'user_followers', 'user_friends', 'user_favourites', 'user_verified',
       'date', 'text', 'hashtags', 'source', 'is_retweet', 'userFriends',
       'userFavourites', 'userFollowers', 'Date', 'userCreated',
       'User_Verified'],
      dtype='object')
```

```
ipl_2022_df.dropna(subset=['hashtags', 'is_retweet', 'source', 'user_favourites', 'user
```

	user_name	user_location	user_description	user_created	user_followers	user_friends
0	World Cricket Baba	india	! Links Baba:\nhttps://t.co/L3bxQ4jPHK\nhttps:...	2022-04-13 06:34:29	1076.0	63
1	The Times Of India	New Delhi	News. Views. Analysis. Conversations. India's ...	2010-04-19 10:50:15	14429584.0	457
2	World Cricket Baba	india	! Links Baba:\nhttps://t.co/L3bxQ4jPHK\nhttps:...	2022-04-13 06:34:29	1076.0	63
3	Social Animal	India	I'm here to avoid my friends on Facebook.	2013-10-15 04:34:14	124.0	502
4	World Cricket Baba	india	! Links Baba:\nhttps://t.co/L3bxQ4jPHK\nhttps:...	2022-04-13 06:34:29	1076.0	63
...	...	...	...	...	...	...

	user_name	user_location	user_description	user_created	user_followers	user_friends
574659	Rohit Sharma FC	NaN	Hii This is an Die Heart Fan Club of Rohit Sharma	2021-08-02 04:05:06	3.0	11
574660	Sanket Pandey	India	Proud to be an indian\n-Jay hind.\n-Vande matr...	2017-01-11 13:44:24	14.0	333
574661	InsideSport	New Delhi, India	Official website of InsideSport - India's prem...	2017-01-21 11:03:22	5654.0	759
574662	Deepanshu Speaks	India	Offical Twitter handle of YouTube channel #Dee...	2021-06-17 14:44:22	11.0	42
574663	Aakash Srivastava	Noida, India	Writer with @SportsTiger	2016-09-10 08:31:12	103.0	2022

574617 rows × 19 columns

After removing the Null values from the above selected columns now the row number become 574617 intally it was574664 before the applying dropna() method

```
ipl_2022_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 574664 entries, 0 to 574663
Data columns (total 19 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   user_name         574664 non-null  object 
 1   user_location     384107 non-null  object 
 2   user_description  492538 non-null  object 
 3   user_created      574655 non-null  object 
 4   user_followers    574649 non-null  float64
 5   user_friends      574649 non-null  object 
 6   user_favourites   574649 non-null  object 
 7   user_verified     574649 non-null  object 
 8   date              574649 non-null  object 
 9   text              574649 non-null  object 
 10  hashtags          574620 non-null  object 
 11  source            574640 non-null  object 
 12  is_retweet        574640 non-null  object 
 13  userFriends       574646 non-null  float64
 14  userFavourites    574646 non-null  float64
 15  userFollowers     574649 non-null  float64
```

```
 16  Date              574646 non-null  datetime64[ns]
 17  userCreated       574646 non-null  datetime64[ns]
 18  User_Verified     574664 non-null  bool
dtypes: bool(1), datetime64[ns](2), float64(4), object(12)
memory usage: 79.5+ MB
```

- if we want to check any particular name or value in column then we can use in method and find that value. if it is contain with that value then it will give result True or it will throw False

```
pip install seaborn==0.12.2
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: seaborn==0.12.2 in /usr/local/lib/python3.8/dist-packages (0.12.2)
Requirement already satisfied: matplotlib!=3.6.1,>=3.1 in /usr/local/lib/python3.8/dist-packages (from seaborn==0.12.2) (3.1.3)
Requirement already satisfied: pandas>=0.25 in /usr/local/lib/python3.8/dist-packages (from seaborn==0.12.2) (1.1.5)
Requirement already satisfied: numpy!=1.24.0,>=1.17 in /usr/local/lib/python3.8/dist-packages (from seaborn==0.12.2) (1.24.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.8/dist-packages (from matplotlib!=3.6.1,>=3.1->seaborn==0.12.2) (0.11.0)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib!=3.6.1,>=3.1->seaborn==0.12.2) (2.8.2)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib!=3.6.1,>=3.1->seaborn==0.12.2) (3.0.9)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib!=3.6.1,>=3.1->seaborn==0.12.2) (1.4.4)
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.8/dist-packages (from pandas>=0.25->seaborn==0.12.2) (2022.7)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.8/dist-packages (from python-dateutil>=2.1->matplotlib!=3.6.1,>=3.1->seaborn==0.12.2) (1.15.0)
```

```
import seaborn as sns
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline
```

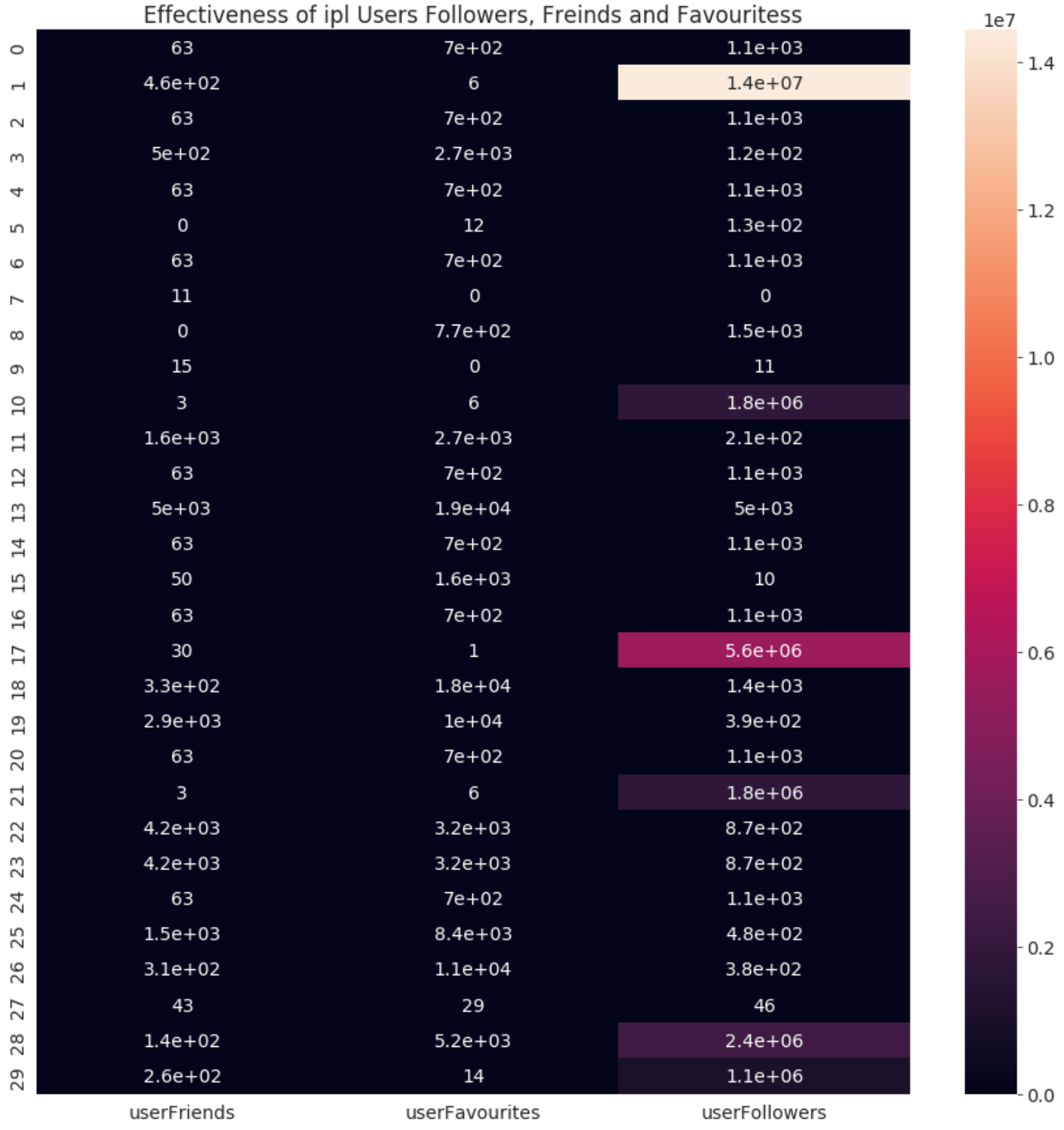
```
sns.set_style('darkgrid')
matplotlib.rcParams['font.size'] = 14
matplotlib.rcParams['figure.figsize'] = (9, 5)
```

visualise the effectiveness of userFriends, userFavourites and userFollowers

```
heat_df=ipl_2022_df[['userFriends','userFavourites','userFollowers']].head(30)
heat_df
```

	userFriends	userFavourites	userFollowers
0	63.0	699.0	1076.0
1	457.0	6.0	14429584.0
2	63.0	699.0	1076.0
3	502.0	2675.0	124.0
4	63.0	699.0	1076.0
5	0.0	12.0	132.0
6	63.0	699.0	1076.0
7	11.0	0.0	0.0
8	0.0	768.0	1547.0
9	15.0	0.0	11.0
10	3.0	6.0	1766764.0
11	1608.0	2677.0	210.0
12	63.0	699.0	1076.0
13	5049.0	19036.0	5042.0
14	63.0	699.0	1076.0
15	50.0	1573.0	10.0
16	63.0	699.0	1076.0
17	30.0	1.0	5601715.0
18	332.0	17641.0	1405.0
19	2916.0	10202.0	389.0
20	63.0	699.0	1076.0
21	3.0	6.0	1766764.0
22	4165.0	3174.0	872.0
23	4165.0	3174.0	872.0
24	63.0	699.0	1076.0
25	1466.0	8427.0	477.0
26	314.0	11039.0	376.0
27	43.0	29.0	46.0
28	141.0	5190.0	2386087.0
29	262.0	14.0	1058647.0

```
plt.figure(figsize=(15,15))
plt.title('Effectiveness of ipl Users Followers, Freinds and Favouritess')
sns.heatmap(heat_df, annot=True);
```



In heatmap it is clear that userFollowers are more effective as compared to the userFriends and userFavourites.Rather it should also mention that few favourites are more in tweeting.

Open-ended exploratory analysis and visualization

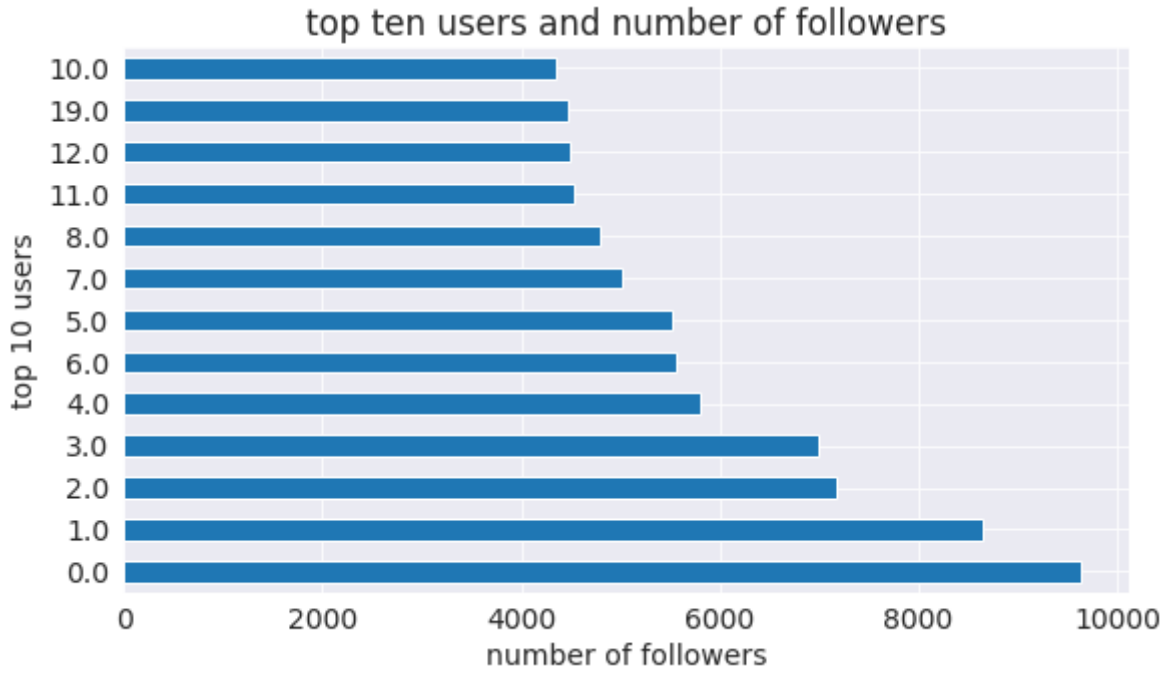
user-followers

```
ipl_2022_df.user_followers.value_counts()
```

```
0.0          9643
1.0          8636
2.0          7175
3.0          6991
```

```
4.0          5811
             ...
10024.0         1
21469.0         1
3232.0          1
21463.0         1
5867562.0       1
Name: user_followers, Length: 19207, dtype: int64
```

```
plt.title('top ten users and number of followers')
plt.xlabel('number of followers')
plt.ylabel('top 10 users')
ipl_2022_df.user_followers.value_counts()[:10].plot(kind='barh');
```



Above graph shows Top 10 tweets followers. The maximum followers number is near 10000 i.e exact value is 9643.we can find the value from the list of series and also can see in graph

source

```
sources= ipl_2022_df.source[:100]
sources
```

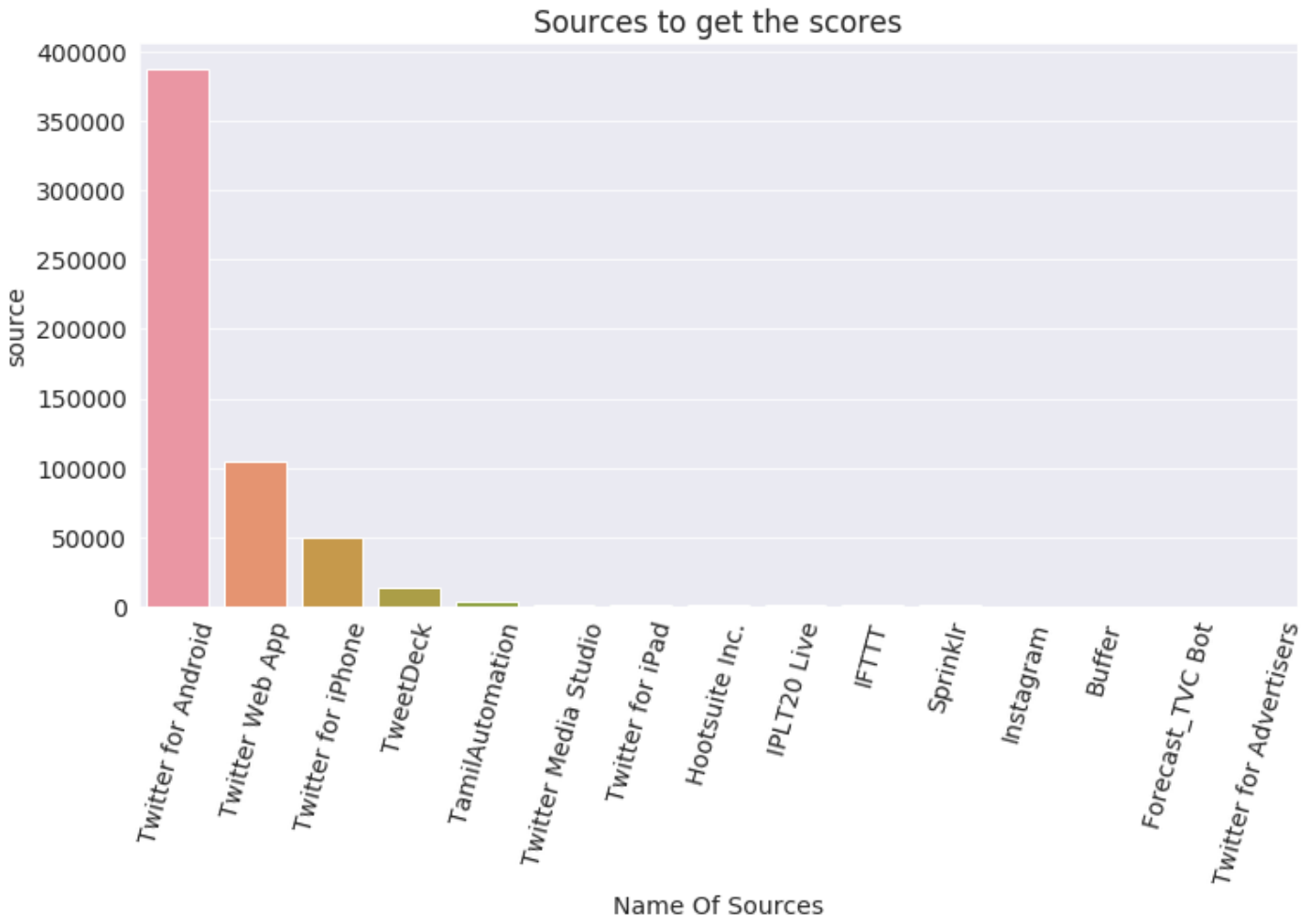
```
0                Postify1
1         Twitter Web App
2                Postify1
3     Twitter for Android
4                Postify1
              ...
95    Twitter for Android
96               Postify1
97    Twitter for Android
98    Twitter for Android
```

```
99    Twitter for Android
Name: source, Length: 100, dtype: object
```

```
ipl_source= ipl_2022_df['source'].value_counts().head(15)
ipl_source
```

```
Twitter for Android        386808
Twitter Web App            105127
Twitter for iPhone          49438
TweetDeck                   13768
TamilAutomation              3368
Twitter Media Studio         1936
Twitter for iPad             1731
Hootsuite Inc.               1183
IPLT20 Live                  1178
IFTTT                        1051
Sprinklr                     1040
Instagram                     687
Buffer                        677
Forecast_TVC Bot              584
Twitter for Advertisers       560
Name: source, dtype: int64
```

```
plt.figure(figsize=(12,6))
plt.title('Sources to get the scores')
plt.xticks(rotation=75)
plt.xlabel('Name Of Sources')
sns.barplot(x=ipl_source.index, y=ipl_source);
```



The above graph shows that most of the tweets has done from the Android phone and it's value is much more than the other sources like from the web app or iphone. As compared to the web app or ipnone, The tweeting from the android are almost four times or we can say that it has 400 percentage as compared to the tweets from the web app

```
from wordcloud import WordCloud
```

user_friends

```
ipl_2022_df['user_friends'].head(50)
```

```
0        63
1       457
2        63
3       502
4        63
5         0
6        63
7        11
8         0
9        15
10        3
11     1608
12       63
13     5049
```

```
14      63
15      50
16      63
17      30
18     332
19    2916
20      63
21       3
22    4165
23    4165
24      63
25    1466
26     314
27      43
28     141
29     262
30      63
31     504
32     262
33      63
34     262
35    5000
36       1
37      63
38      65
39      63
40     262
41       0
42    5004
43     815
44      63
45      43
46     457
47       0
48       5
49     262
Name: user_friends, dtype: object
```

```
import plotly.express as px

fig= px.histogram(ipl_2022_df['user_friends'].head(5000), x='user_friends', marginal='b
fig.update_layout(
    autosize=False,
    width=1000,
    height=500,)
fig.show();
```

User friends between zero to hundreds are tweeting more than 2000 and the value shows 2135 in the graph. It is showing something different than the other ranges of distribution.

user_name

list of user name on the basis of number of tweets

```
ipl_user= ipl_2022_df['user_name'].value_counts().head(30)
ipl_user
```

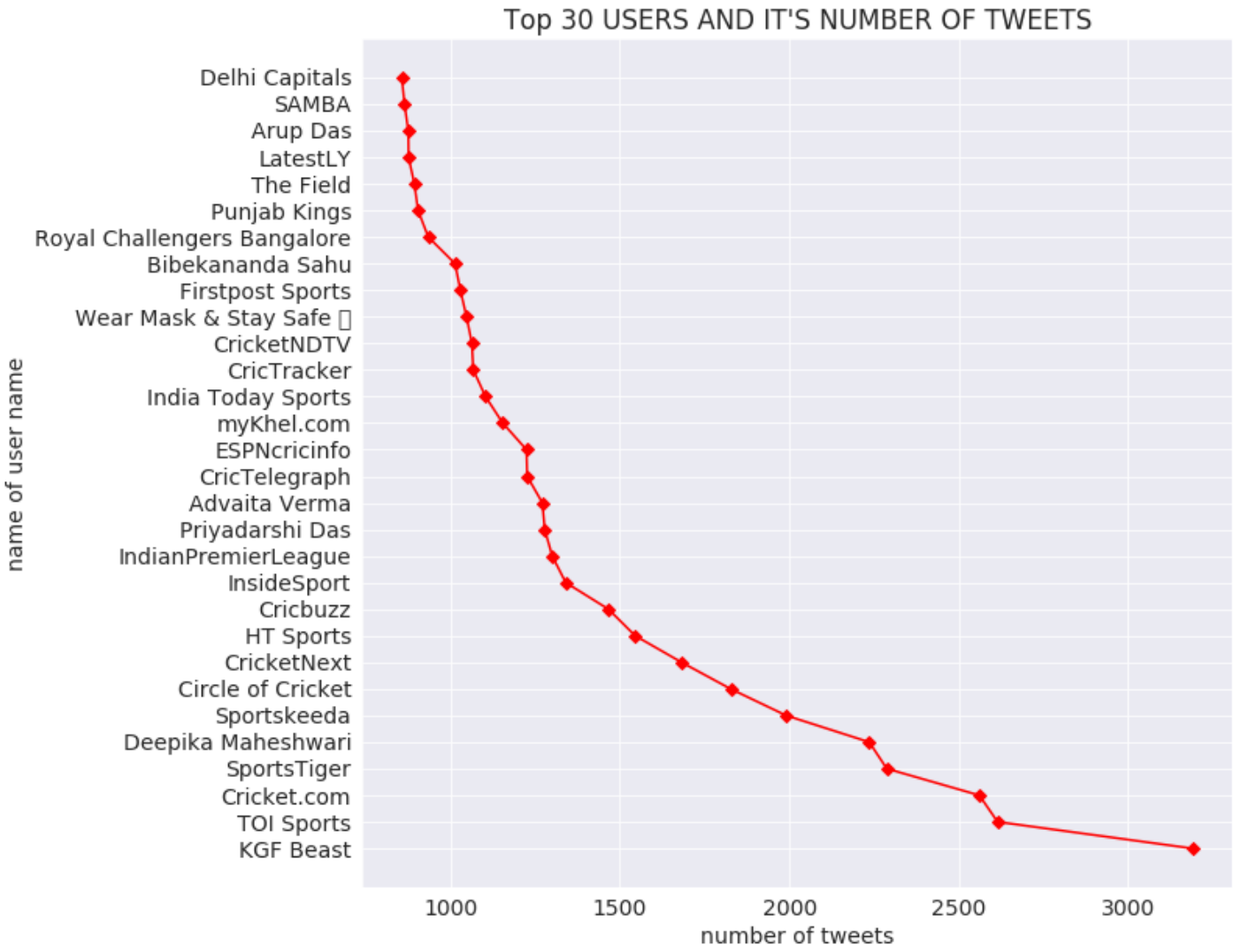
```
KGF Beast                      3191
TOI Sports                     2615
Cricket.com                    2561
SportsTiger                    2288
Deepika Maheshwari             2236
Sportskeeda                    1991
Circle of Cricket              1827
CricketNext                    1682
HT Sports                      1545
Cricbuzz                       1466
InsideSport                    1340
IndianPremierLeague            1300
Priyadarshi Das                1278
Advaita Verma                  1271
CricTelegraph                  1226
ESPNcricinfo                   1224
myKhel.com                     1154
India Today Sports             1103
CricTracker                    1067
CricketNDTV                    1063
Wear Mask & Stay Safe □        1047
Firstpost Sports               1029
Bibekananda Sahu               1013
Royal Challengers Bangalore     934
Punjab Kings                    904
The Field                       893
LatestLY                        876
Arup Das                        875
SAMBA                           864
Delhi Capitals                  856
Name: user_name, dtype: int64
```

```
wordcloud2 = WordCloud().generate_from_frequencies(ipl_user)
# Generate plot
plt.figure(figsize=(15,10))
plt.imshow(wordcloud2)
plt.axis("off")
plt.show()
```

with the help of wordcloud it is clear that username TOI, KGF BEAST and Cricket.com are tweeting more and it is high in numbers

```
plt.figure(figsize=(10,10))
plt.title("Top 30 USERS AND IT'S NUMBER OF TWEETS")
plt.xlabel('number of tweets')
plt.ylabel('name of user name')
plt.plot( ipl_user, ipl_user.index, 'r-', marker='D');
```

```
/usr/local/lib/python3.8/dist-packages/matplotlib/backends/backend_agg.py:211:
RuntimeWarning:

Glyph 128591 missing from current font.

/usr/local/lib/python3.8/dist-packages/matplotlib/backends/backend_agg.py:180:
RuntimeWarning:

Glyph 128591 missing from current font.
```



Above graph is between user name and it tweets. Graph shows that KGF Beast has tweeted more and graph is above the 3000 point and list shows that KGF Beast tweet 3191. second tweeted is TOI Sport which it value lies near 2600 aroung and when we observe list of series of username then found that it's value i s 2615. In the list of top 30 username the less tweets found in Delhi Capitals which lies near 500 to 800 values. When we check it value in list of series of username then we find that it has exactly value is 856 tweets

userFavourites

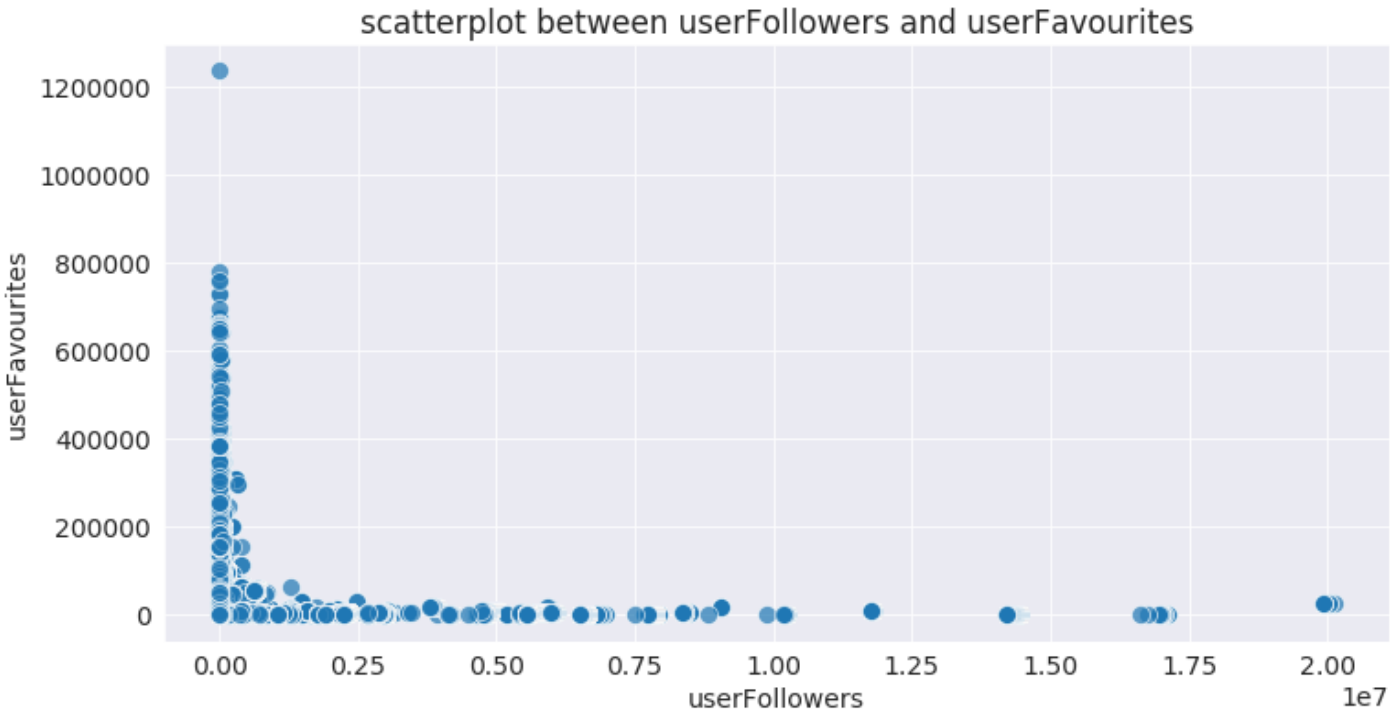
```
ipl_2022_df.userFollowers
```

```
0             1076.0
1         14429584.0
2             1076.0
3              124.0
4             1076.0
              ...
574659           3.0
574660          14.0
574661        5654.0
574662          11.0
574663         103.0
Name: userFollowers, Length: 574664, dtype: float64
```

```
ipl_2022_df.userFavourites
```

```
0          699.0
1            6.0
2          699.0
3         2675.0
4          699.0
           ...
574659     219.0
574660    3844.0
574661    7723.0
574662    1479.0
574663    1557.0
Name: userFavourites, Length: 574664, dtype: float64
```

```
plt.figure(figsize=(12,6))
plt.title('scatterplot between userFollowers and userFavourites')
sns.scatterplot( data =ipl_2022_df, x='userFollowers', y='userFavourites', alpha= 0.7,
```



In this graph it is clear that most of the user favourite and user followers are lies betweeen 0 to (0.25 x 1e7) .Number of top user favourites scoring upto 8000 which are near 0 and (0.1.25 x 1e7)

List out the columns according to the memory occupy during dataframe formation

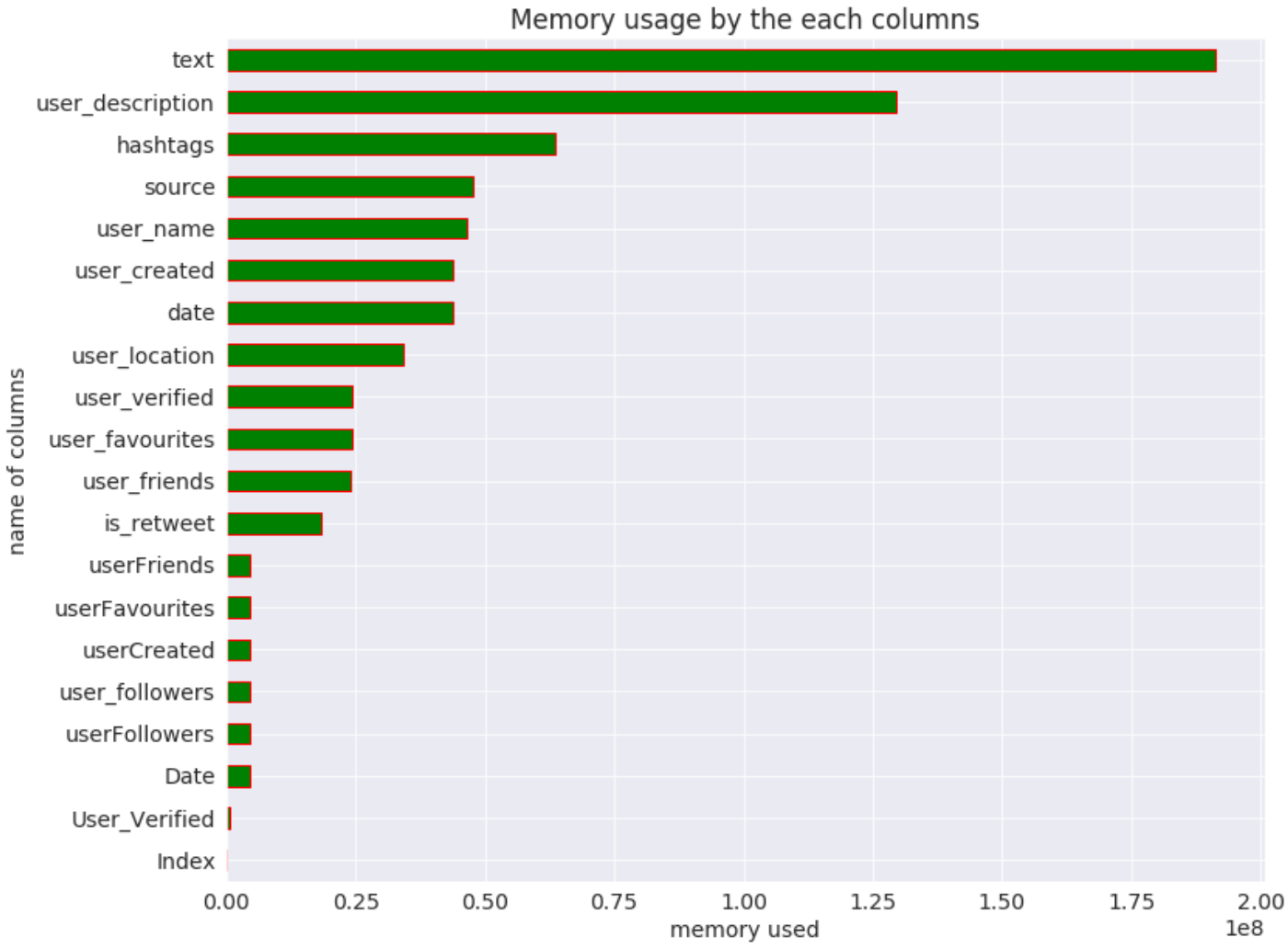
```
ipl_memory_usage= ipl_2022_df.memory_usage(deep=True)
ipl_memory_usage.sort_values(ascending=False)
```

```
text                191140543
user_description    129549121
hashtags             63420436
```

```
source               47781101
user_name            46327434
user_created         43673936
date                 43673798
user_location        34269258
user_verified        24389907
user_favourites      24209411
user_friends         24031010
is_retweet           18389248
userFriends           4597312
userFavourites        4597312
userCreated           4597312
user_followers        4597312
userFollowers         4597312
Date                  4597312
User_Verified          574664
Index                     128
dtype: int64
```

Visualize the memory usage on horizental bar chart

```
plt.figure(figsize=(12,10))
plt.title('Memory usage by the each columns')
plt.xlabel('memory used')
plt.ylabel('name of columns')
ipl_memory_usage.sort_values().plot(kind= 'barh',edgecolor='red', color='green');
```



From the above graph we can find that text are occupying more memoory and that is lies near . Then after user description is occupying the memory and its value lies near 1.25 x 10

working on Date

Extract date, time and hour from the datetime Dtype column

```
ipl_2022_df.Date
```

```
0         2022-06-20 22:00:03
1         2022-06-20 21:30:00
2         2022-06-20 20:00:24
3         2022-06-20 19:10:00
4         2022-06-20 19:00:18
                  ...
574659    2022-01-06 13:05:44
574660    2022-01-06 13:04:34
574661    2022-01-06 12:59:46
574662    2022-01-06 12:52:12
574663    2022-01-06 12:51:05
Name: Date, Length: 574664, dtype: datetime64[ns]
```

Date Extracted

```
ipl_2022_df.Date.dt.date
```

```
0         2022-06-20
1         2022-06-20
2         2022-06-20
3         2022-06-20
4         2022-06-20
             ...
574659    2022-01-06
574660    2022-01-06
574661    2022-01-06
574662    2022-01-06
574663    2022-01-06
Name: Date, Length: 574664, dtype: object
```

Time Extracted

```
ipl_2022_df.Date.dt.time
```

```
0         22:00:03
1         21:30:00
2         20:00:24
3         19:10:00
4         19:00:18
            ...
574659    13:05:44
574660    13:04:34
574661    12:59:46
574662    12:52:12
574663    12:51:05
Name: Date, Length: 574664, dtype: object
```

Hour Extracted

```
ipl_2022_df.Date.dt.hour
```

```
0         22.0
1         21.0
2         20.0
3         19.0
4         19.0
          ...
574659    13.0
574660    13.0
574661    12.0
574662    12.0
574663    12.0
Name: Date, Length: 574664, dtype: float64
```

Asking and answering interesting questions.

1. What are the maximaum user_followers, user_friends and user_favourites ?
2. which date and time of user_created when tweerer id created who tweeted first and last during the ipl2022 ?
3. what are the percentage of user_followers having more than a million followers?
4. Which users are tweeting first and last time during ipl 2022 ?
5. Find out maximum and minimum tweets on a particular dates during ipl 2022 ?
6. how many locations where it were tweeting from Delhi, pakistan and bangladesh during ipl 2022 ?
7. which time interval user are more likely to tweeting during 24 hour ?
8. which day user are more tweeting in the week ?

QUESTION 1 What are the maximaum user_followers, user_friends and user_favourites ?

```
ipl_2022_df.userFollowers.max()
```

```
20111374.0
```

maximum user followers are 20111374 during ipl 2022

```
ipl_2022_df.userFriends.max()
```

```
350197.0
```

maximum user friends are 350197 during ipl 2022

```
ipl_2022_df.userFavourites.max()
```

```
1236671.0
```

Maximum user favourite are 1236671 during ipl 2022

QUESTION 2 Which date and time of user_created when tweerer id created who tweeted first and last during the ipl2022 ?

```
ipl_2022_df['userCreated']
```

```
0         2022-04-13 06:34:29
1         2010-04-19 10:50:15
2         2022-04-13 06:34:29
3         2013-10-15 04:34:14
4         2022-04-13 06:34:29
                  ...
574659    2021-08-02 04:05:06
574660    2017-01-11 13:44:24
574661    2017-01-21 11:03:22
```

```
574662   2021-06-17 14:44:22
574663   2016-09-10 08:31:12
Name: userCreated, Length: 574664, dtype: datetime64[ns]
```

- Details of date and time of user created who tweeted first during ipl2022

```
ipl_2022_df['userCreated'][0]
```

```
Timestamp('2022-04-13 06:34:29')
```

User who tweeted first has created id at 6.34 am in the morning and date was 13th April 2022

- Details of date and time of user created who tweeted last during ipl2022

```
ipl_2022_df['userCreated'][-1:]
```

```
574663   2016-09-10 08:31:12
Name: userCreated, dtype: datetime64[ns]
```

user who has tweeted last has created id at 8.31 am in the morning and date was 10th september2016

QUESTION 3 what are the percentage of user_followers having more than a million followers ?

```
more_than_one_million_followers= ipl_2022_df.userFollowers[ipl_2022_df.userFollowers >=
more_than_one_million_followers.sort_values(ascending=False)
```

```
3121      20111374.0
102649    20009365.0
94735     20009323.0
487660    19905296.0
463621    19905251.0
             ...
328468      100424.0
331425      100423.0
457138      100047.0
457364      100041.0
418021      100040.0
Name: userFollowers, Length: 30494, dtype: float64
```

```
fig= px.histogram(more_than_one_million_followers, x=more_than_one_million_followers.sc
fig.update_layout(
    autosize=False,
    width=1000,
    height=500),
fig.show();
```

Above histogram graph shows that range between one million to two million the follwers are 9378. Then after tweets follwer between 2 million to 3 million is 1867. It means drastically decreasing the followers. moreover it has

observed the followers between 1 million to 1.1 million increasing again and the follower goes us to 2784. In box plot show that the average followers are 864.748 million

```
percentage_of_one_million_followers= (len(more_than_one_million_followers)/ len(ipl_202
percentage_of_one_million_followers
```

```
5.306405134130553
```

the percentage of more than one million followers are 5.31 %

```
below_than_one_million_followers= ipl_2022_df.userFollowers[ipl_2022_df.userFollowers <
below_than_one_million_followers
```

```
0         1076.0
2         1076.0
3          124.0
4         1076.0
5          132.0
           ...
574659       3.0
574660      14.0
574661    5654.0
574662      11.0
574663     103.0
Name: userFollowers, Length: 544155, dtype: float64
```

```
percentage_of_below_than_one_million_followers=(len(below_than_one_million_followers)/
percentage_of_below_than_one_million_followers
```

```
94.69098464494034
```

the percentage of below than one million followers are 94.69 %

QUESTION 4 Which users are tweeting first and last time during ipl 2022 ?

```
ipl_2022_df.Date
```

```
0        2022-06-20 22:00:03
1        2022-06-20 21:30:00
2        2022-06-20 20:00:24
3        2022-06-20 19:10:00
4        2022-06-20 19:00:18
                 ...
574659   2022-01-06 13:05:44
574660   2022-01-06 13:04:34
574661   2022-01-06 12:59:46
574662   2022-01-06 12:52:12
574663   2022-01-06 12:51:05
Name: Date, Length: 574664, dtype: datetime64[ns]
```

```
tweeting_first= ipl_2022_df.Date.min()
tweeting_first
```

```
Timestamp('2022-01-06 12:51:05')
```

First tweets during ipl2022 was on 6th of January 2022 and time when it was tweeted at 12.15 am

```
tweeting_last= ipl_2022_df.Date.max()
tweeting_last
```

```
Timestamp('2022-06-20 22:00:03')
```

The last tweets during the ipl 2022 was on 20th of june 2022 and it's timing was 10 pm

QUESTION 5 Find out maximum and minimum tweets on a particular dates during ipl 2022 ?

```
number_of_tweets=ipl_2022_df.Date.dt.date.value_counts()
number_of_tweets
```

```
2022-03-31    25341
2022-04-22    22672
2022-04-12    22031
2022-04-21    21519
2022-04-30    21296
              ...
2022-06-18       55
2022-05-08       49
2022-06-11       45
2022-05-06       15
2022-05-01        2
Name: Date, Length: 123, dtype: int64
```

```
maximum_tweets_date= number_of_tweets.iloc[0:1]
maximum_tweets_date
```

```
2022-03-31    25341
Name: Date, dtype: int64
```

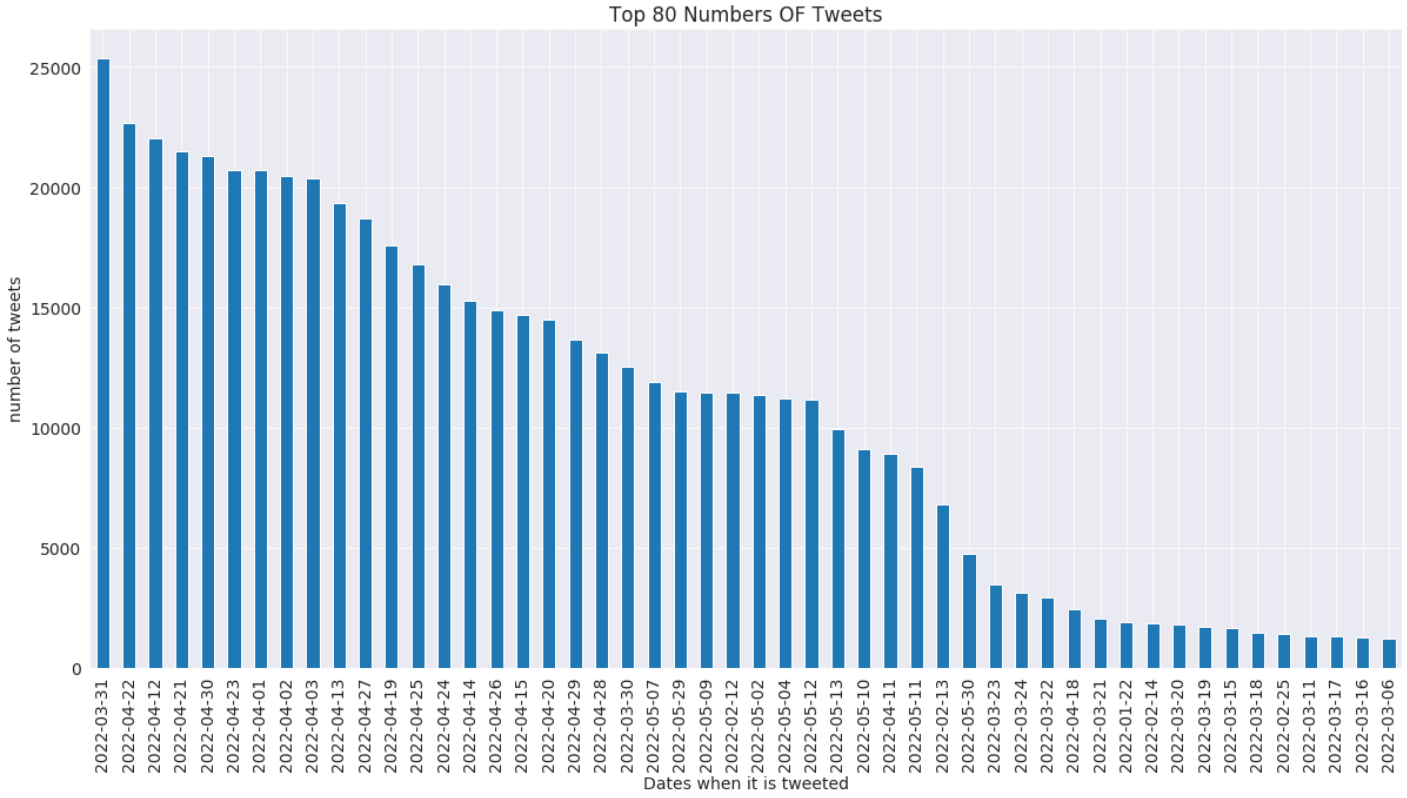
Hence the result is maximum tweets is 25341 on 31 march 2022 during the ipl 2022

```
minimum_tweets_date= number_of_tweets.iloc[-1:]
minimum_tweets_date
```

```
2022-05-01    2
Name: Date, dtype: int64
```

Here it is clear that minimum tweets is only 2 on first May of 2022 during the ipl 2022

```
plt.figure(figsize=(20,10))
plt.title('Top 80 Numbers OF Tweets')
plt.xlabel('Dates when it is tweeted')
plt.ylabel('number of tweets')
number_of_tweets.head(50).plot(kind='bar');
```



From the above visualization it is showing the maximum tweets are above 25000 k. Here it should be mention that top 50 Dates has chosen when highest tweets are tweeted during the ipl 2022

QUESTION 6 How many locations were tweeted from Delhi, pakistan and bangladesh during ipl 2022

```
'Delhi' in ipl_2022_df.user_location.values
```

```
True
```

```
ipl_2022_df[ipl_2022_df.user_location == 'Delhi']['user_location'].value_counts()
```

```
Delhi    2628
Name: user_location, dtype: int64
```

In Delhi, it has 2628 locations where it were tweeted duing ipl2022

```
'pakistan' in ipl_2022_df.user_location.values
```

```
True
```

```
ipl_2022_df[ipl_2022_df.user_location == 'pakistan']['user_location'].value_counts()
```

```
pakistan    10
Name: user_location, dtype: int64
```

In Pakistan, it has 10 locations where it were tweeted duing ipl2022

```
'Bangladesh' in ipl_2022_df.user_location.values
```

```
True
```

```
ipl_2022_df[ipl_2022_df.user_location == 'Bangladesh']['user_location'].value_counts()
```

```
Bangladesh    217
Name: user_location, dtype: int64
```

In Bangladesh, it has 217 locations where it were tweeted duing ipl2022

```
user_locations= ipl_2022_df.user_location.value_counts().head(30)
user_locations
```

```
India                   51775
Mumbai, India           22524
New Delhi, India        21879
Kolkata, India           9554
Bengaluru, India         6250
Hyderabad, India         5574
Chennai, India           5233
Mumbai                   4278
Jaipur, India            3866
Indore, India            3748
Pune, India              3694
Delhi, India             3153
Kolkata                  3152
Noida, India             2965
Ahmadabad City, India    2931
New Delhi                2777
Gujarat, India           2770
india                    2767
Delhi                    2628
Lucknow, India           2534
India □□                 2373
Jaipur, Rajasthan        2334
West Bengal, India       2137
Chennai                  1979
Maharashtra, India       1831
Uttar Pradesh, India     1783
Gurgaon, India           1759
Rajasthan, India         1755
```

```
INDIA                    1678
भारत                     1635
Name: user_location, dtype: int64
```

```
from wordcloud import WordCloud
```

```
wordcloud2 = WordCloud().generate_from_frequencies(user_locations)
# Generate plot
plt.figure(figsize=(15,10))
plt.imshow(wordcloud2)
plt.axis("off")
plt.show()
```

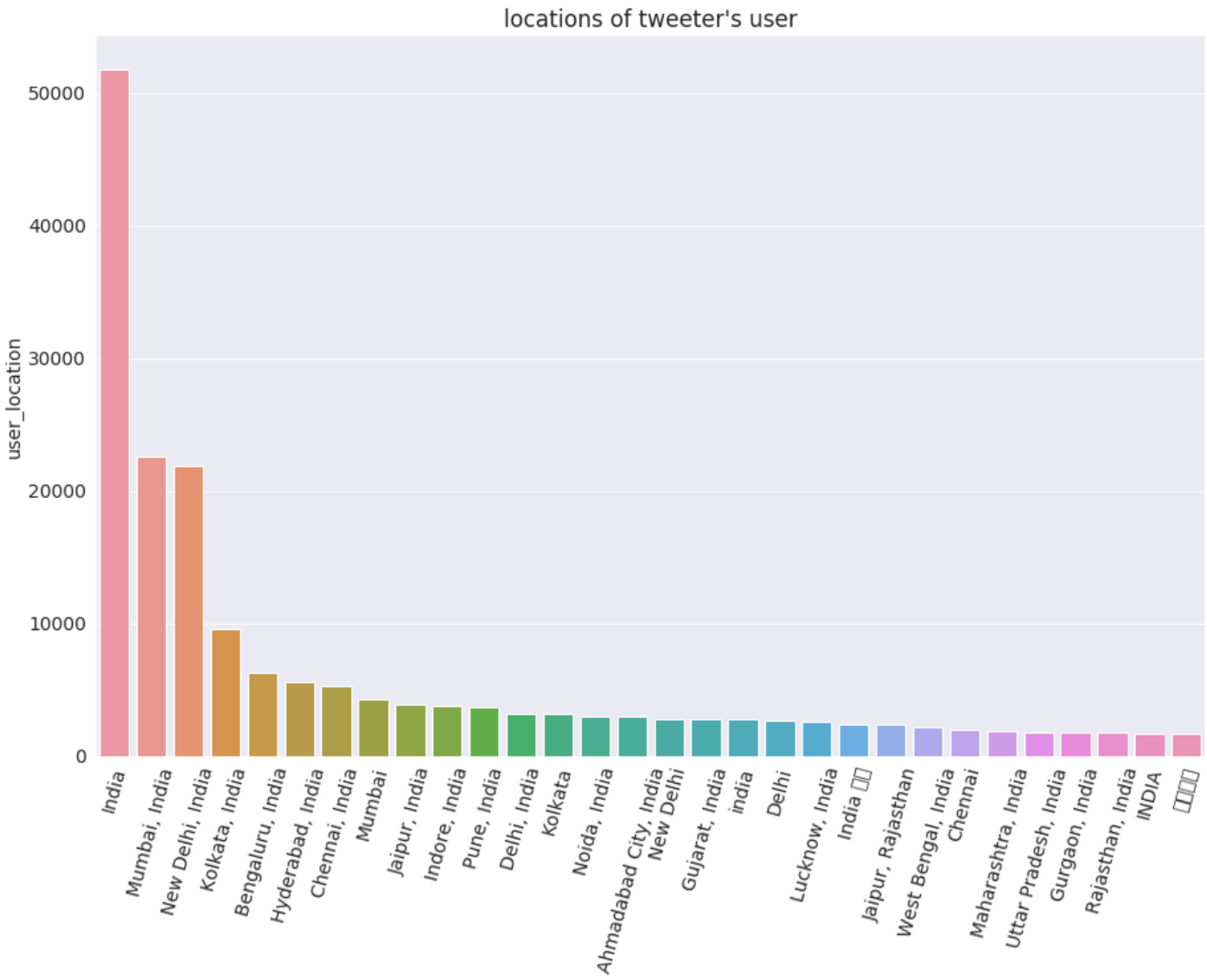
from the above cloud tweeting from india is much more and then after Mumbai, Kolkata and New Delhi is acquriring to tweets alot as compared to the other cities of india

```
plt.figure(figsize=(15,10))
plt.title("locations of tweeter's user")
plt.xticks(rotation=75)
plt.ylabel('number of tweets from that place')
sns.barplot(y= user_locations, x=user_locations.index);
```

```
/usr/local/lib/python3.8/dist-packages/matplotlib/backends/backend_agg.py:211:
RuntimeWarning:

Glyph 127470 missing from current font.

/usr/local/lib/python3.8/dist-packages/matplotlib/backends/backend_agg.py:211:
RuntimeWarning:

Glyph 127475 missing from current font.
```

```
/usr/local/lib/python3.8/dist-packages/matplotlib/backends/backend_agg.py:211:
RuntimeWarning:

Glyph 2349 missing from current font.

/usr/local/lib/python3.8/dist-packages/matplotlib/backends/backend_agg.py:211:
RuntimeWarning:

Glyph 2366 missing from current font.

/usr/local/lib/python3.8/dist-packages/matplotlib/backends/backend_agg.py:211:
RuntimeWarning:

Glyph 2352 missing from current font.

/usr/local/lib/python3.8/dist-packages/matplotlib/backends/backend_agg.py:211:
RuntimeWarning:

Glyph 2340 missing from current font.

/usr/local/lib/python3.8/dist-packages/matplotlib/backends/backend_agg.py:180:
RuntimeWarning:

Glyph 127470 missing from current font.

/usr/local/lib/python3.8/dist-packages/matplotlib/backends/backend_agg.py:180:
RuntimeWarning:

Glyph 127475 missing from current font.

/usr/local/lib/python3.8/dist-packages/matplotlib/backends/backend_agg.py:180:
RuntimeWarning:

Glyph 2349 missing from current font.

/usr/local/lib/python3.8/dist-packages/matplotlib/backends/backend_agg.py:180:
RuntimeWarning:

Glyph 2366 missing from current font.

/usr/local/lib/python3.8/dist-packages/matplotlib/backends/backend_agg.py:180:
RuntimeWarning:
```

```
Glyph 2352 missing from current font.

/usr/local/lib/python3.8/dist-packages/matplotlib/backends/backend_agg.py:180:
RuntimeWarning:

Glyph 2340 missing from current font.
```



from the above graph of user locations shows that maximum tweets are from the india itself. Graph clearly show that rest part of the india's tweets are much higher than the metroplolitan city but it is also clear thet the metro cities and big city like Kolkata,Bangluru, chennai and Hydrabad has sufficient amount has tweeted. it has smart value of tweets as compared to the other city of india

QUESTION 7 which time interval user are more likely to tweeting during 24 hour ?

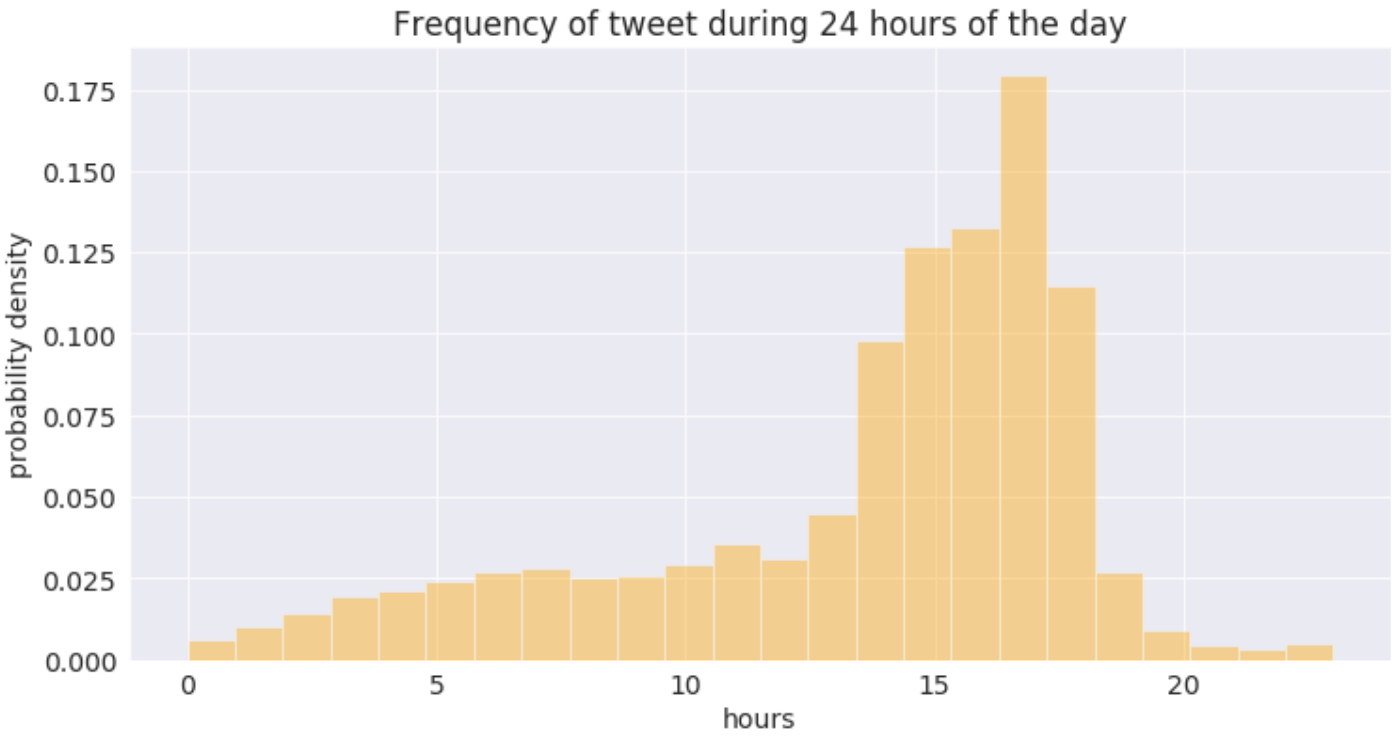
```
ipl_2022_df.Date.dt.hour
```

```
0         22.0
```

```
1         21.0
2         20.0
3         19.0
4         19.0
          ...
574659    13.0
574660    13.0
574661    12.0
574662    12.0
574663    12.0
Name: Date, Length: 574664, dtype: float64
```

```
plt.figure(figsize=(12,6))
plt.title('Frequency of tweet during 24 hours of the day')
fig= sns.distplot(ipl_2022_df.Date.dt.hour, bins=24,kde=False,color='orange', norm_hist
fig.set_xlabel('hours');
fig.set_ylabel('probability density');
```

```
<ipython-input-102-3509c6b1e59e>:3: UserWarning:


`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751
```



From the above graph it is clear that from 14 hours to 18 hours tweeting was higher than the other time. So it is aspected that most of the matched was daynight match which stared at around 2 pm and continued up to 6 pm

QUESTION 8 Which day user are more tweeting in the week

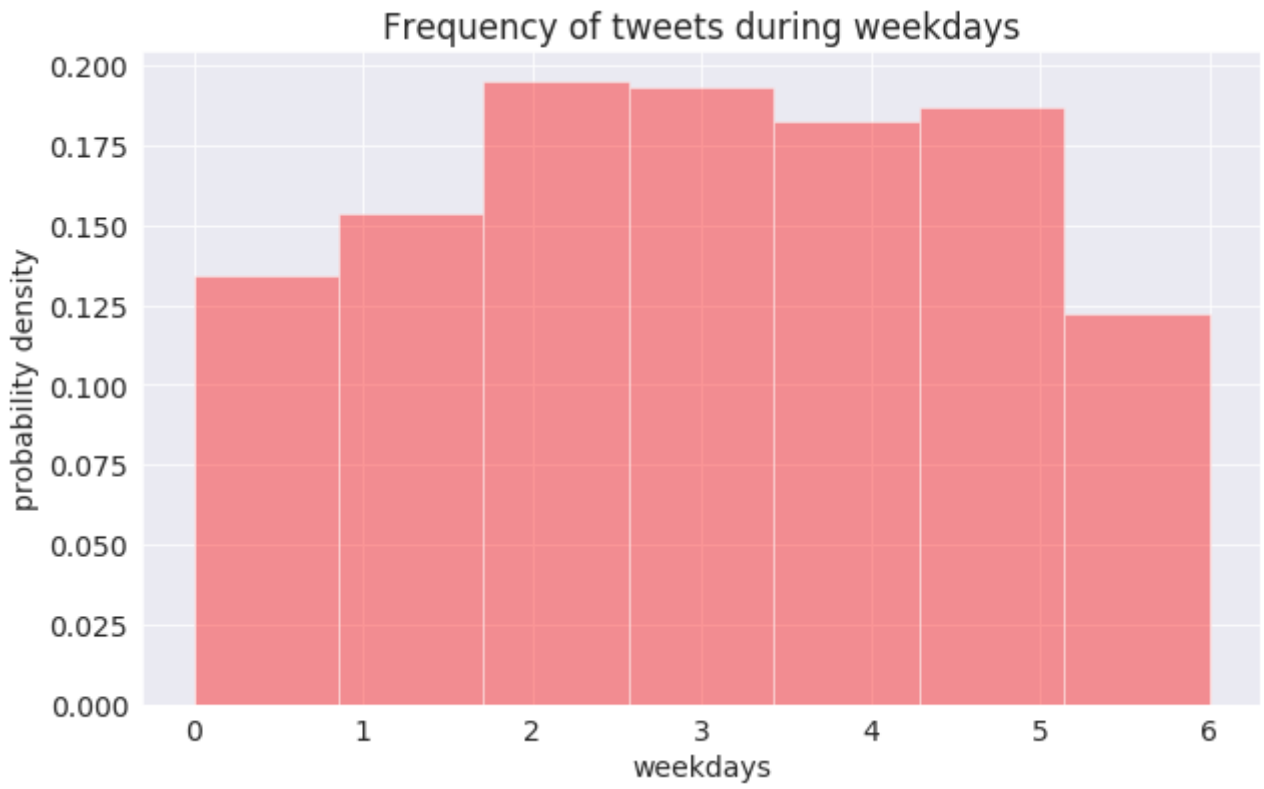
```
ipl_2022_df.Date.dt.dayofweek
```

```
0         0.0
1         0.0
2         0.0
3         0.0
4         0.0
         ...
574659    3.0
574660    3.0
574661    3.0
574662    3.0
574663    3.0
Name: Date, Length: 574664, dtype: float64
```

```
plt.figure(figsize=(10,6))
plt.title('Frequency of tweets during weekdays')
fig= sns.distplot(ipl_2022_df.Date.dt.dayofweek, bins=7,kde=False, color= 'red', norm_h
fig.set_ylabel('probability density')
fig.set_xlabel('weekdays');
```

```
<ipython-input-104-73c86b4c0295>:3: UserWarning:
```

```
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751
```



From the above graph of week days, it is clear that from Tuesday and Wednesday tweeting is more and after Thursday and Friday it is slightly less but Saturday and Sunday is less tweeting as compared to the other days

Summarizing inferences and drawing conclusions

###Summery

- In this project i have learnt various method of Pandas DataFrame, i worked on datetime Dtype, which was interesting for me because i have extracted date, time, hour.
- Change the object Dtype to the required Dtype were also very learning method because after changing it gave us very different statical result when we applied describe method and seen it' changed Dtype in .info() method too.
- Intersting insightes:- During analysis it was interesting that indians are more interested for ipl as compared to the other countries. Apart from that Mumbai, Bangluru , Chennai, Hydrabad has massive tweets as compared to the other cities of india. In overall india's tweets are 400 % higher than the tweets from the Mumbai.

Conclusion and Future Work

- I observed one very interesting behavour of tweeters were very active during the match was playing . They were tweeting much more(from 2 pm to the 7 pm) as compared to the rest of day times.
- I want to work on the cricket world cup score data where dataset are include with the runs of batsman, strike rate, bowler's wickets, maiden overs, venues and cricket stadium where more frequency of tournaments has organised.

Reference

Followed the documention of pandas DataFrame, used many method for cleaning and removing as well as for sorting the values https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html

Followed the documention of plotly express, seaborn, and matplotlib for visualization purpose

https://plotly.github.io/plotly.py-docs/generated/plotly.express.html

https://matplotlib.org/stable/plot_types/index.html

https://seaborn.pydata.org/

*Apart from the documention page of pandas, plotly , seaborn and matplotlib, google search engine, stack overflow and W3 School were very helpful for me to analysis the exploratory data.