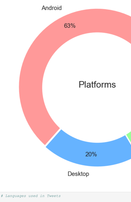
```
### ||| Twitter Analysis ||| ###

# Importing Modules
import tp
import tweepy
import pandas as pd
import numpy as np
import pandas as pd
import matplotlib as mp
import matplotlib.mp
import matplotlib.pyplot as plt
%matplotlib inline

# Plotting Top 10 HashTags

df = pd.read_csv("Dump/Top_HashTags.csv")
sb.set_theme(context = "poster", style="darkgrid")

plt.figure(figsize=(30,15))
g = sb.barplot(data=df,x="HashTags",y="Count_Volume")

plt.title("Top 10 HashTags")
plt.savefig("Top_HashTags.png")
```

```
### END ||| Tweets ||| ###

# Creating Tweets Dataframe
Tweets = pd.read_csv("Dump/Tweets.csv")
Tweets.head()
```

	id	user_id	created_at	lang	retweet_count	favorite_count	source	hashtags
0	1345149580229132025	1245844783935754386	2021-01-01 23:20:25+00:00	en	362	0	Twitter for iPhone	[HappyNewYear2021, happynewyear2021]
1	1345146000092142376	1334507597586583772	2021-01-01 23:06:12+00:00	en	24	0	Twitter for Android	[BHMMTeSaNaPlus, NewYear, DigitalHyundai]
2	1345147663379488386	1334507597586583772	2021-01-01 23:12:10+00:00	en	38	0	Twitter for Android	[DigitalZ]
3	1345147279387410944	1335855435984769784	2021-01-01 23:10:23+00:00	fr	41	0	Twitter for iPhone	[]
4	1345143604207020512	1212464330967668416	2021-01-01 23:02:08+00:00	en	94	0	Twitter for Android	[POB]

```
# Wordcloud of Tweets

df = Tweets
stop = set(wordcloud.STOPWORDS)
text = ""

TwetList = list(df[df.lang.str.contains("en")]["text"])

for x in TwetList:
    x = str(x)
    x.lower()
    x = re.sub(r"(\w+:\/\/\S+)|(?:@[A-Za-z0-9]+)|(?:0-9A-Za-z \t])|(\w+:\/\/\S+)", " ", x).split()
    text+=x+" "

cloud = wordcloud.WordCloud(width = 1920, height = 900, background_color="black", stopwords = stop, min_fo
plt.figure(figsize=(40,15))
plt.axis("off")
plt.imshow(cloud)
plt.savefig("Wordcloud Tweets.png")
```

```
# Platforms used to Tweet

df = Tweets
df = df[df["Android","Desktop","iPhone","Other"]]
size = 0
size.append(len(df[df["source"].str.contains("Twitter for Android")]))
size.append(len(df[df["source"].str.contains("Twitter for iPhone")]))
size.append(len(df[df["source"].str.contains("Twitter for iPhone")]))
size.append(len(df[df.source]))
size = [63,7,9,26]
colors = ["#7FFFD4", "#A52A2A", "#FFc000"]

sb.set_theme(context = "poster", style="darkgrid")
plt.figure(figsize=(15,15))
plt.pie(sizes, colors = colors, labelsize=40, autopct="%1.0f%%", pctdistance=0.85, explode = explode,textpr
plt.title("Platforms")
plt.gca().set_ylabel("")
plt.gca().set_xlabel("Platforms", sp=0, fontsize=30,bar="center")
plt.legend(loc=1)
plt.savefig("Platforms.png")
```




```
# Languages used in Tweets

df = Tweets
df = df.sort_values(counts).to_frame().reset_index()

df.columns = ["language","Tweets"]
df = df.sort_values("Tweets").sortedcorp=False
df = df.sort_values("Tweets").sortedcorp=False
df = df.sort_values("Tweets").sortedcorp=False df[df["Tweets"].sum().ignore_index=True]
df = df[-1]

groupedbylanguage= df.groupby("language").sum().reset_index()

sb.set_theme(context = "poster", style="darkgrid")
plt.figure(figsize=(15,15))
g = sb.barplot(data=df,x="language",y="Tweets")

# x = list(df["Tweets"])
for i in range(len(f)):
    g.text(i,f[i],f[i],ef8[i])

plt.title("Languages Used In Tweets")
plt.savefig("language.png")
```

```
# Tweets Vs. Hour of Day

df = Tweets

Activity = [i[2 for i in range(24)]

for i in df["created_at"]:
    Activity[int(str(i)[11:13])]+=1

sb.set_theme(context = "poster", style="darkgrid")
plt.figure(figsize=(15,15))
x = [str(i) for i in Activity]
y = [Activity[i] for i in Activity]
plt.bar(x,y,color="#FFc000")
for i in range(len(y)):
    plt.text(i-0.25,y[i],str(y[i]),fontsize=21)
plt.xticks(rotation=0,fontsize=21)
plt.ylabel("No. of Tweets",fontsize = 30)
plt.title("Tweets Vs. Hour of Day")
plt.savefig("Tweeting_Hour.png")
```

```
# Sentiment Analysis

df = Tweets
Tweet_Text = df[df.lang.str.contains("en")]["text"]

Positive = 0
Negative = 0
Neutral = 0

for t in Tweet_Text:
    t = re.sub(r"(\w+:\/\/\S+)|(?:@[A-Za-z0-9]+)|(?:0-9A-Za-z \t])|(\w+:\/\/\S+)", " ", t).split()
    Result = tb.TextBlob(Cleaned_Text).sentiment.polarity
    if Result > 0:Positive+=1
    elif Result < 0:Negative+=1
    else: Neutral+=1

labels = ["Positive","Negative","Neutral"]
sizes = [Positive,Negative,Neutral]

explode = (0.10,0.0,0.0)
colors = ["#FFC000","#A52A2A","#4A0E0F"]

sb.set_theme(context = "poster", style="darkgrid")
plt.figure(figsize=(15,15))
plt.pie(sizes, colors = colors, labelsize=40, autopct="%1.0f%%", pctdistance=0.85, explode = explode,textpr
plt.gca().set_ylabel("")
plt.gca().set_xlabel("Sentiment in Analysis", sp=0, fontsize=30,bar="center")
plt.legend(loc=1)
plt.savefig("Sentiment.png")
```

```
### END ||| Tweets ||| ###

# Creating Users Dataframe
Users = pd.read_csv("Dump/Users.csv")
Users.head()
```

	id	name	screen_name	created_at	location	protected	followers_count	friends_count	favourite
0	1245844783935754386	Sohar uddin	suharuddin888	2020-04-13 23:48:06+00:00		False	15	30	
1	1334507597586583772	Vipul	Vipul04122	2020-12-03 20:19:04+00:00	भूई मि	False	3	17	
2	1334507597586583772	Vipul	Vipul04122	2020-12-03 20:19:04+00:00	भूई मि	False	2	17	
3	1335855435984769784	Nikhil Patra	nikhil_patra	2018-06-08 18:13:00+00:00		False	54	34	
4	1212464330967668416	Chukwu	Abel00798437	2020-01-01 20:18:06+00:00	Biafra	False	187	242	

```
# Percentage of Verified Accounts

df = Users

Ver = len(df[df.verified==True])
No = len(df[df.verified==False])

sb.set_theme(context = "poster", style="darkgrid")
plt.figure(figsize=(15,15))
x = [Ver,No]
plt.pie(x,labelsize=[f"Unverified",f"Verified"],explode=(0,0),colors=["#FFC000","#FFFFFF"], autopct="%1.0f%%", ec
plt.savefig("Verified.png")
```

```
# Followers Distribution

df = Users
Followers = list(df["followers_count"])

b = [f"0-250",f"251-1000",f"1001-2500",f"2501-10000",f"+10000"]f=[]

for i in Followers:
    if i<=250:f+=[f"0-250"]raw
    elif i<=1000:f+=[f"251-1000"]raw
    elif i<=2500:f+=[f"1001-2500"]raw
    elif i<=10000:f+=[f"2501-10000"]raw
    else:f+=[f"+10000"]raw

sb.set_theme(context = "poster", style="darkgrid")
plt.figure(figsize=(15,15))
x = [f.count(i) for i in b]
y = [f[i] for i in b]
plt.bar(x,y,color="#ffc000")
for i in range(len(y)):
    plt.text(i-0.25,y[i],str(y[i]),fontsize=20)
plt.xticks(rotation=0,fontsize=21)
plt.ylabel("No. of Accounts",fontsize = 30)
plt.title("Followers Distribution")
plt.savefig("Followers.png")
```

```
# Friends Distribution

df = Users
Friends = list(df["friends_count"])

b = [f"0-250",f"251-1000",f"1001-2500",f"2501-10000",f"+10000"]f=[]

for i in Friends:
    if i<=250:f+=[f"0-250"]raw
    elif i<=1000:f+=[f"251-1000"]raw
    elif i<=2500:f+=[f"1001-2500"]raw
    elif i<=10000:f+=[f"2501-10000"]raw
    else:f+=[f"+10000"]raw

sb.set_theme(context = "poster", style="darkgrid")
plt.figure(figsize=(15,15))
x = [f.count(i) for i in b]
y = [f[i] for i in b]
plt.bar(x,y,color="#ffc000")
for i in range(len(y)):
    plt.text(i-0.25,y[i],str(y[i]),fontsize=20)
plt.xticks(rotation=0,fontsize=21)
plt.ylabel("No. of Accounts",fontsize = 30)
plt.title("Friends Distribution")
plt.savefig("Friends.png")
```



```
### END OF NOTEBOOK ###
```