

```

import matplotlib.pyplot as plt
from matplotlib.figure import Figure
import pandas as pd
import seaborn as sns
import numpy as np

df = pd.read_csv('D:/dataset_Facebook.csv', sep=';')

df.head(5)

```



Page total likes	Type	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach	Lifetime Post Total Impressions	Lifetime Engaged Users
------------------	------	----------	------------	--------------	-----------	------	---------------------------	---------------------------------	------------------------

0	139441	Photo	2	12	4	3	0.0	2752	5091	178
1	139441	Status	2	12	3	10	0.0	10460	19057	1457
2	139441	Photo	3	12	3	3	0.0	2413	4373	177
3	139441	Photo	2	12	2	10	1.0	50128	87991	2211
4	139441	Photo	2	12	2	3	0.0	7244	13594	671



```

ts = df.groupby('Type')['share'].sum()
#ts = ts.div(ts.sum()).astype(float).round(2)*100
ts

```

```

Type
Link      284.0
Photo     11461.0
Status    1414.0
Video     365.0
Name: share, dtype: float64

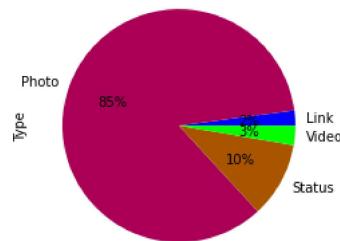
```

```
ts.plot.pie(label='Type', title = " PIE CHART WITH PERCENTAGE", colormap='brg', autopct='%1.0f%')
```

```

<AxesSubplot:title={'center':' PIE CHART WITH PERCENTAGE'}, ylabel='Type'>
PIE CHART WITH PERCENTAGE

```



```

ts = df.groupby('Type')['like'].sum()
#ts = ts.div(ts.sum()).astype(float).round(2)*100
ts

```

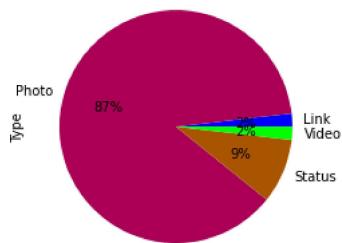
```

Type
Link      1613.0
Photo     77610.0
Status    7952.0
Video     1620.0
Name: like, dtype: float64

```

```
ts.plot.pie(label='Type', title = " PIE CHART WITH PERCENTAGE", colormap='brg', autopct='%1.0f%')
```

```
<AxesSubplot:title={'center':' PIE CHART WITH PERCENTAGE'}, ylabel='Type'>
PIE CHART WITH PERCENTAGE
```



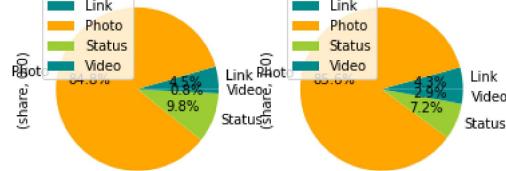
```
ts = pd.pivot_table(df, index=['Type'], columns=['Paid'], values=['share'], aggfunc='count')
ts
```

share		
Paid	0.0	1.0
Type		
Link	16	6
Photo	302	119
Status	35	10
Video	3	4

```
plt.figure();
ts.plot.pie(title='MULTIPLE PIE CHARTS', colors=['darkcyan', 'orange', 'yellowgreen'], autopct='%.1f%%', subplots=True)
```

```
array([<AxesSubplot:ylabel='(share, 0.0)'>,
       <AxesSubplot:ylabel='(share, 1.0)'>], dtype=object)
<Figure size 432x288 with 0 Axes>
```

MULTIPLE PIE CHARTS

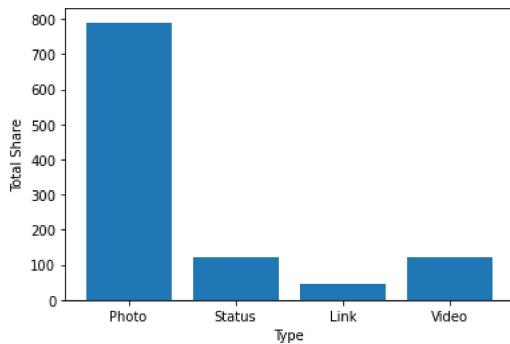


```
x=df['Type']
y=df['share']
```

```
#plotting the data
```

```
plt.bar(x,y)
#plt.title('Tips Data set')
plt.xlabel('Type')
plt.ylabel('Total Share')
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```

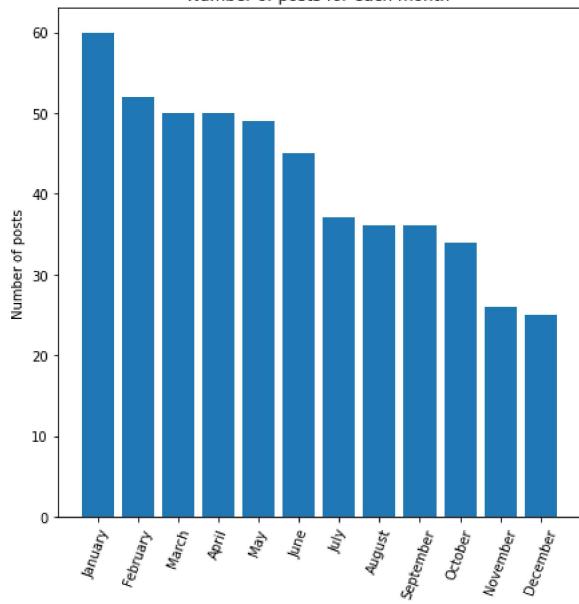


```
plt.figure(figsize=(7,7))

objects = ('January', 'February', 'March', 'April','May','June','July','August','September','October','November','December')
y_pos = np.arange(len(objects))

plt.bar(y_pos, df.loc[:, 'Post Month'].value_counts().values)
plt.xticks(y_pos, objects, rotation=70)
plt.ylabel('Number of posts')
plt.title('Number of posts for each month')

Text(0.5, 1.0, 'Number of posts for each month')
```

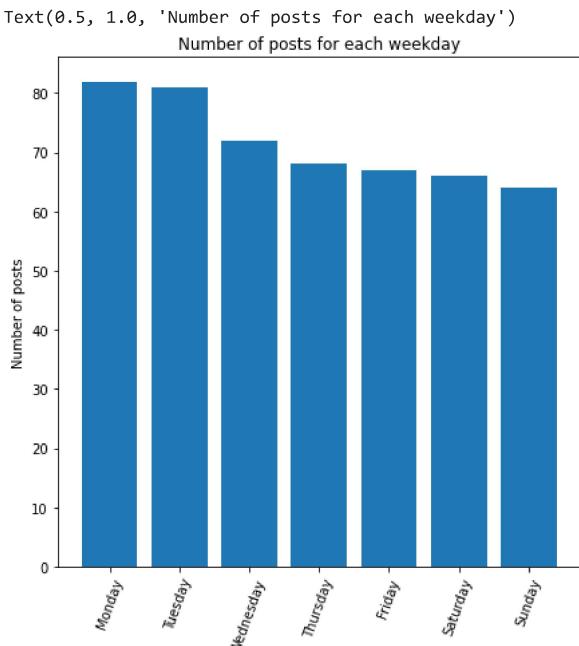


```
plt.figure(figsize=(7,7))

objects = ('Monday', 'Tuesday', 'Wednesday', 'Thursday','Friday','Saturday','Sunday')
y_pos = np.arange(len(objects))

plt.bar(y_pos, df.loc[:, 'Post Weekday'].value_counts().values)
plt.xticks(y_pos, objects, rotation=70)
plt.ylabel('Number of posts')
plt.title('Number of posts for each weekday')

Text(0.5, 1.0, 'Number of posts for each weekday')
```



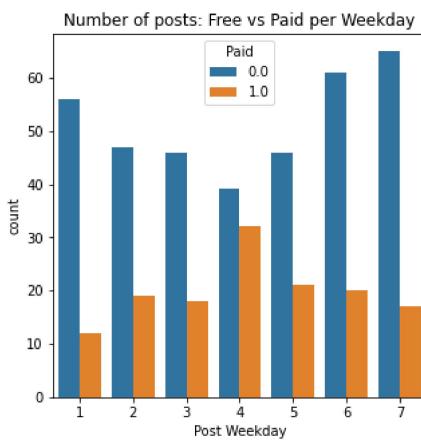
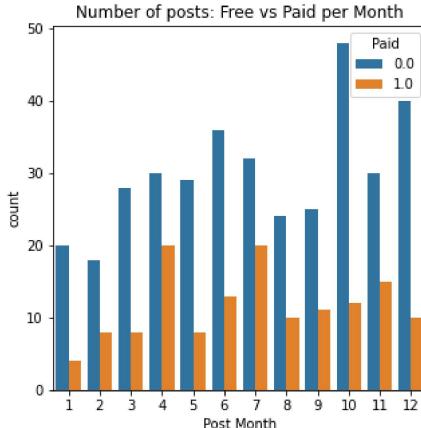
```

plt.figure(figsize=(5,5))
sns.countplot(x='Post Month',hue='Paid',data=df)
plt.title("Number of posts: Free vs Paid per Month")

plt.figure(figsize=(5,5))
sns.countplot(x='Post Weekday',hue='Paid',data=df)
plt.title("Number of posts: Free vs Paid per Weekday")

Text(0.5, 1.0, 'Number of posts: Free vs Paid per Weekday')

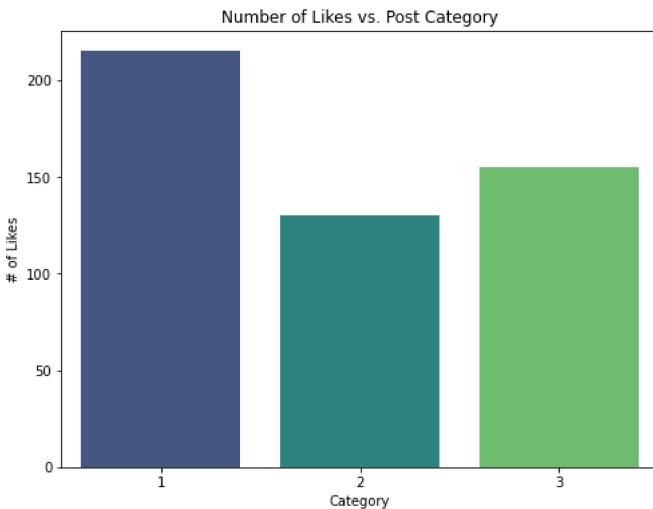
```



```

plt.figure(figsize=(8,6))
sns.countplot(x='Category',data=df,palette='viridis')
plt.ylabel("# of Likes")
plt.title("Number of Likes vs. Post Category")
plt.savefig('catCount.png', bbox_inches='tight')

```

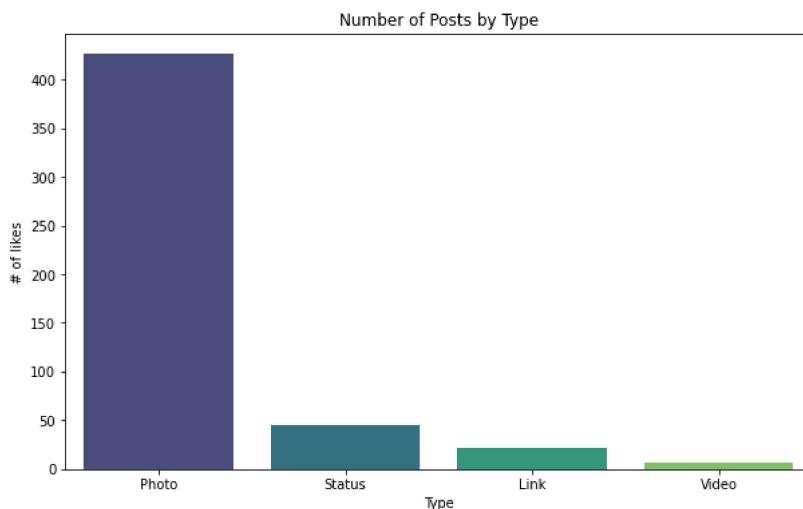


```

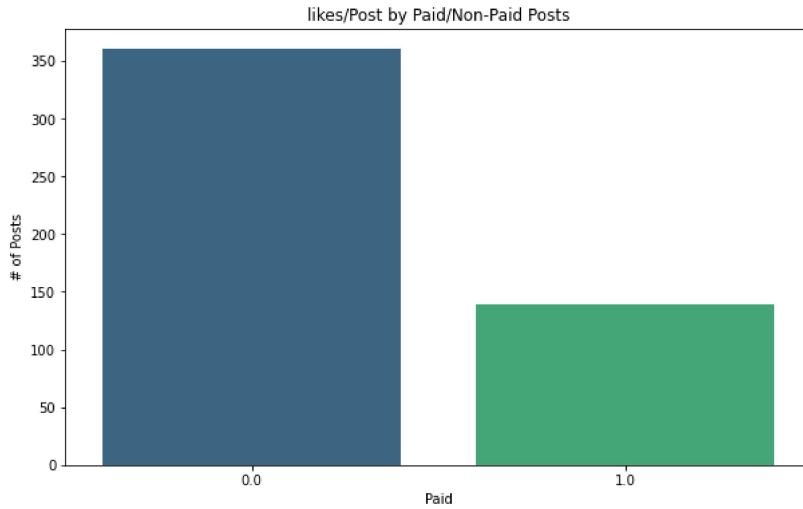
plt.figure(figsize=(10,6))
sns.countplot(x='Type',data=df,palette='viridis')
# plt.ylim(0,750)

```

```
#sns.despine(offset=4,bottom=True)
plt.title("Number of Posts by Type")
plt.ylabel("# of likes")
#plt.legend(loc='upper left')
plt.savefig('typeCount.png', bbox_inches='tight')
```



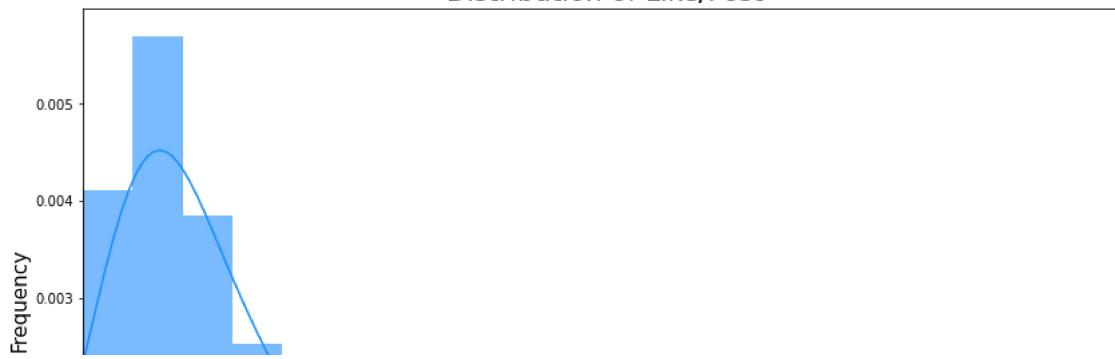
```
plt.figure(figsize=(10,6))
sns.countplot(x='Paid',data=df,palette='viridis')
#sns.despine(offset=4,bottom=True)
plt.title("likes/Post by Paid/Non-Paid Posts")
plt.ylabel("# of Posts")
plt.savefig('paidCount.png', bbox_inches='tight')
```



```
plt.figure(figsize=(14,8))
sns.distplot(df[df['like']<1000]['like'],bins=20,color='dodgerblue',hist_kws={'alpha':0.6})
plt.xlim(0,1000)
plt.xlabel("# of Likes",fontsize=15)
plt.ylabel('Frequency',fontsize=15)
plt.title('Distribution of Like/Post',fontsize=20)
```

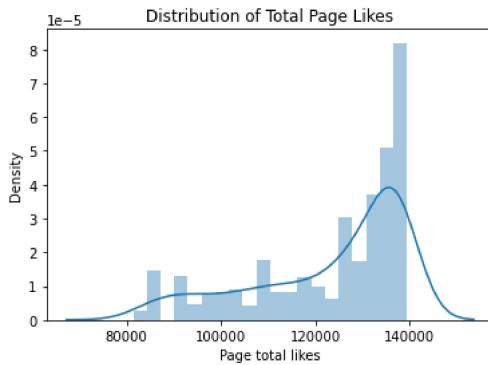
```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function
  warnings.warn(msg, FutureWarning)
Text(0.5, 1.0, 'Distribution of Like/Post')
```

Distribution of Like/Post



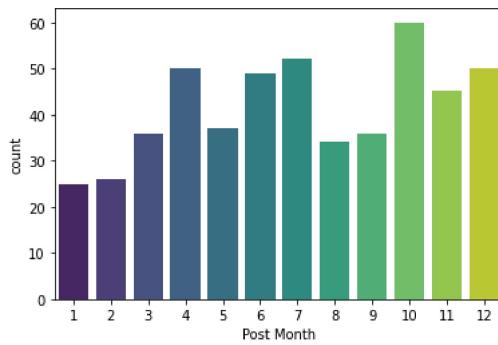
```
sns.distplot(df['Page total likes'], bins=20)
plt.title("Distribution of Total Page Likes")
```

```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function
  warnings.warn(msg, FutureWarning)
Text(0.5, 1.0, 'Distribution of Total Page Likes')
```



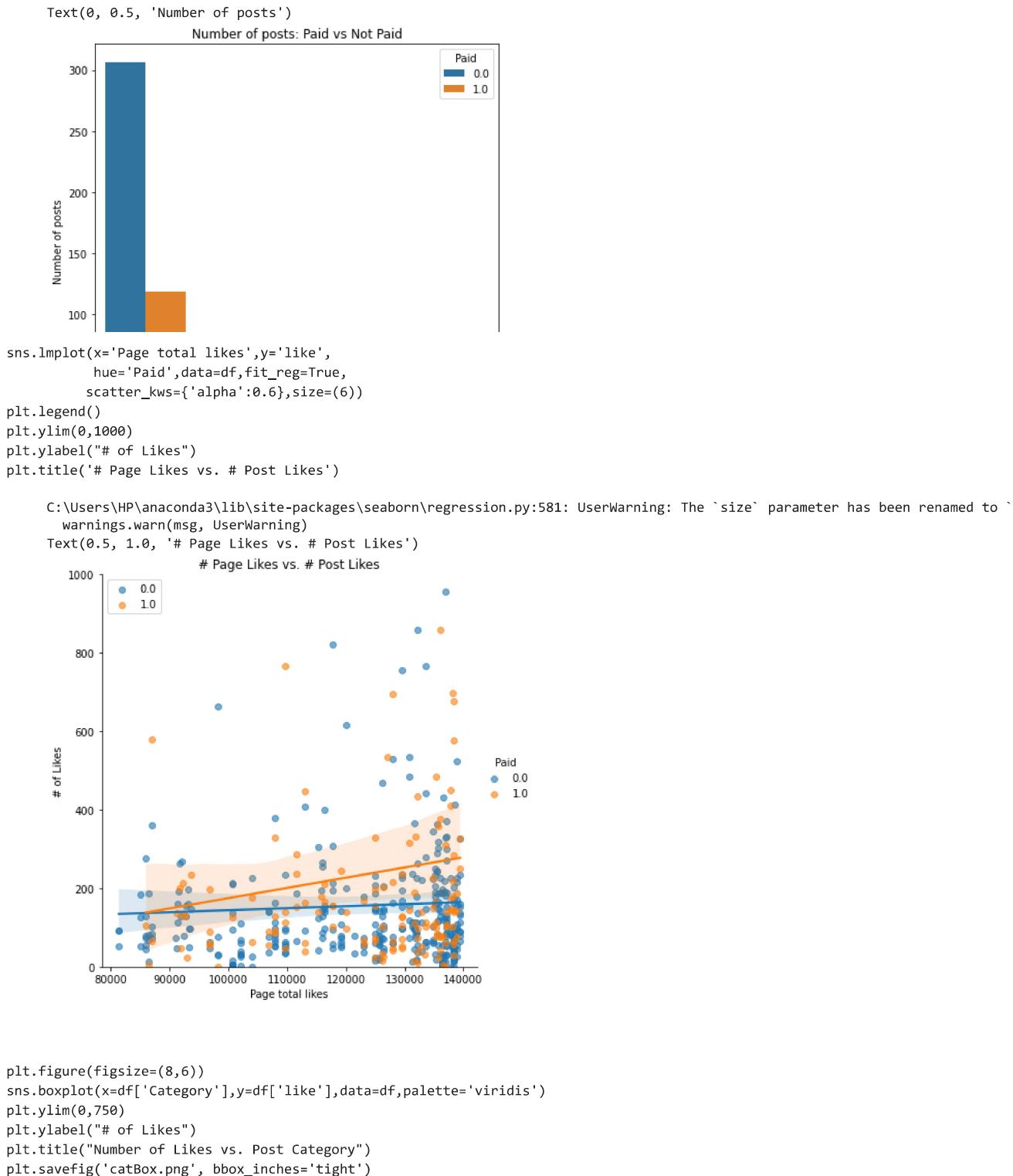
```
sns.countplot(df['Post Month'], palette='viridis')
```

```
C:\Users\HP\anaconda3\lib\site-packages\_decorators.py:36: FutureWarning: Pass the following variable as a keyword
  warnings.warn(
<AxesSubplot:xlabel='Post Month', ylabel='count'>
```



```
plt.figure(figsize=(7,7))
```

```
sns.countplot(x='Type', hue='Paid', data=df)
plt.title("Number of posts: Paid vs Not Paid")
plt.ylabel("Number of posts")
```

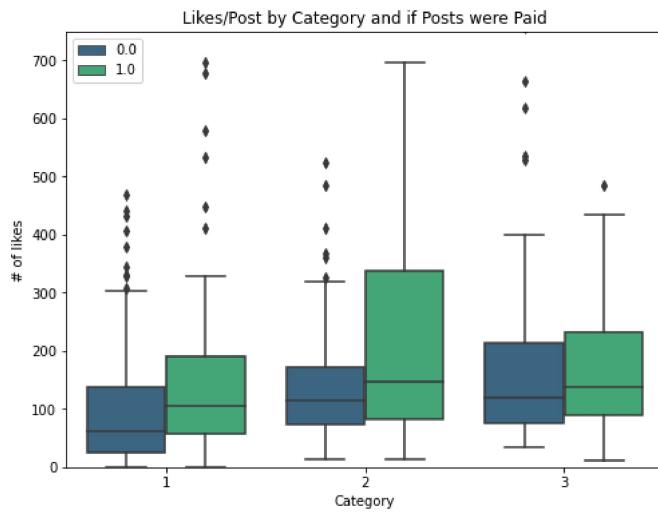


Number of Likes vs. Post Category

```

plt.figure(figsize=(8,6))
sns.boxplot(x='Category',y='like',data=df,hue='Paid',palette='viridis')
plt.ylim(0,750)
plt.ylabel("# of likes")
plt.legend(loc='upper left')
plt.title("Likes/Post by Category and if Posts were Paid")
plt.savefig('catPaidCount.png', bbox_inches='tight')

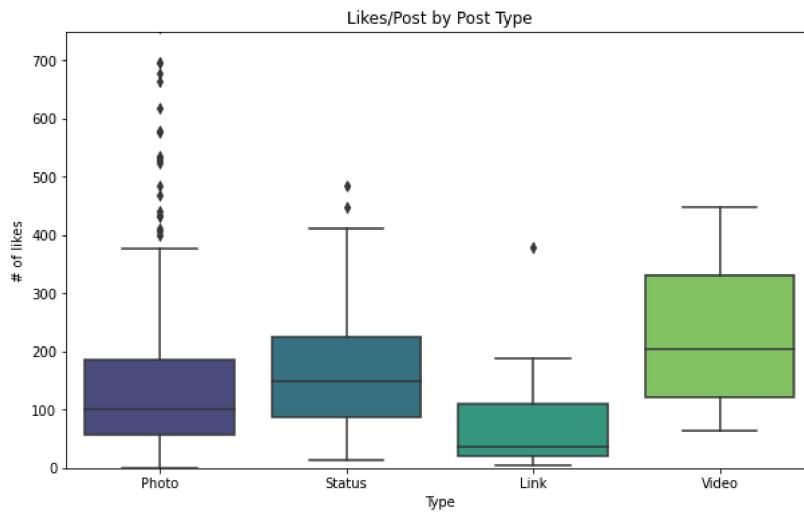
```



```

plt.figure(figsize=(10,6))
sns.boxplot(x='Type',y='like',data=df,palette='viridis')
plt.ylim(0,750)
#sns.despine(offset=4,bottom=True)
plt.title("Likes/Post by Post Type")
plt.ylabel("# of likes")
# plt.legend(loc='upper left')
plt.savefig('typeBox.png', bbox_inches='tight')

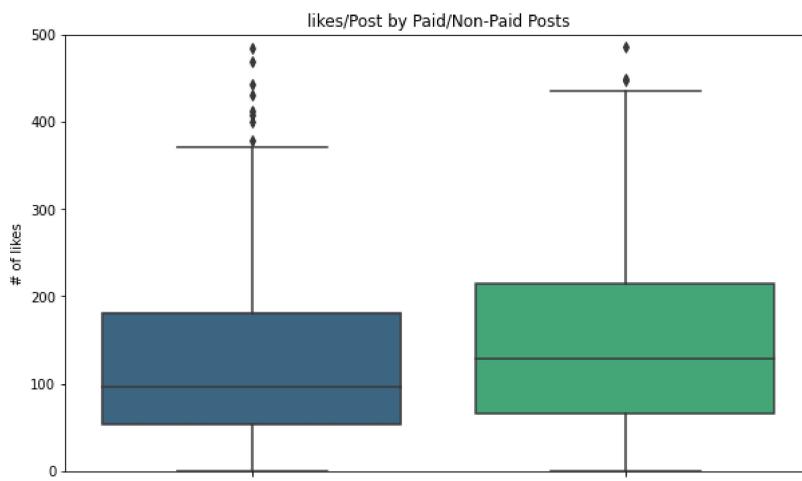
```



```

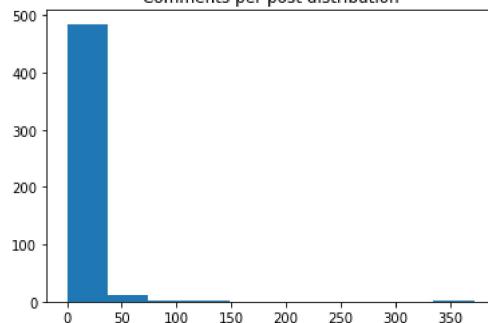
plt.figure(figsize=(10,6))
sns.boxplot(x='Paid',y='like',data=df,palette='viridis')
plt.ylim(0,500)
#sns.despine(offset=4,bottom=True)
plt.title("Likes/Post by Paid/Non-Paid Posts")
plt.ylabel("# of likes")
plt.savefig('paidBox.png', bbox_inches='tight')

```



```
plt.hist(df.loc[:, 'comment'])
plt.title("Comments per post distribution")
```

Text(0.5, 1.0, 'Comments per post distribution')



```
df.isnull().sum()
```

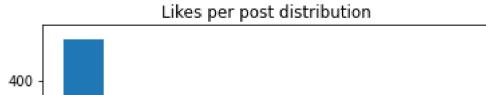
Page total likes	0
Type	0
Category	0
Post Month	0
Post Weekday	0
Post Hour	0
Paid	1
Lifetime Post Total Reach	0
Lifetime Post Total Impressions	0
Lifetime Engaged Users	0
Lifetime Post Consumers	0
Lifetime Post Consumptions	0
Lifetime Post Impressions by people who have liked your Page	0
Lifetime Post reach by people who like your Page	0
Lifetime People who have liked your Page and engaged with your post	0
comment	0
like	1
share	4
Total Interactions	0

dtype: int64

```
df = df.dropna(subset=['Paid'])
```

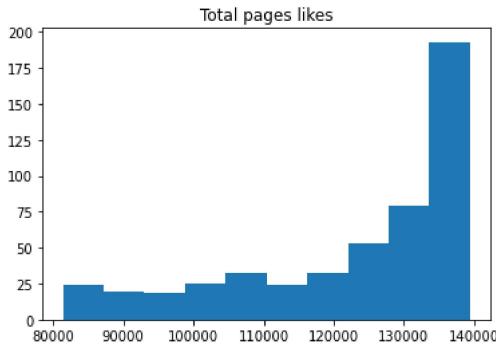
```
plt.hist(df.loc[:, 'like'])
plt.title("Likes per post distribution")
```

```
Text(0.5, 1.0, 'Likes per post distribution')
```



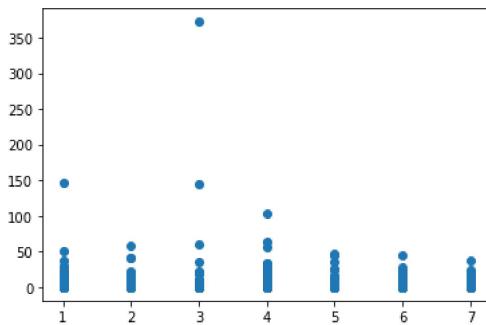
```
plt.hist(df.loc[:, 'Page total likes'])
plt.title("Total pages likes")
```

```
Text(0.5, 1.0, 'Total pages likes')
```



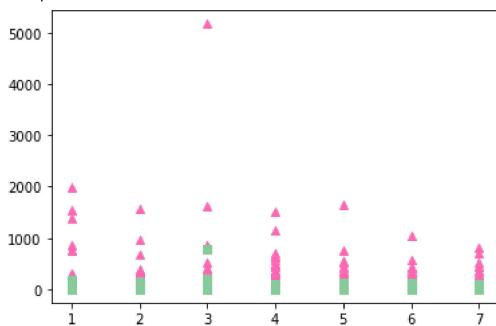
```
x=df['Post Weekday']
y=df['comment']
plt.scatter(x, y)
```

```
<matplotlib.collections.PathCollection at 0x1e63f3ae970>
```

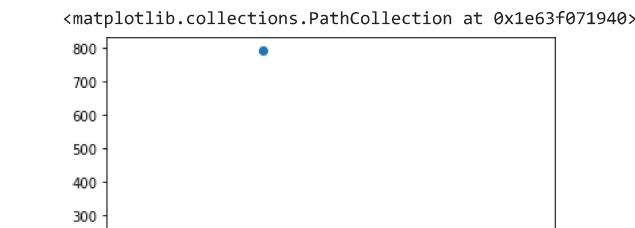


```
x=df['Post Weekday']
y=df['like']
plt.scatter(x, y, color = 'hotpink', marker = "^")
x=df['Post Weekday']
y=df['share']
plt.scatter(x, y, color = '#88c999', marker = "s")
```

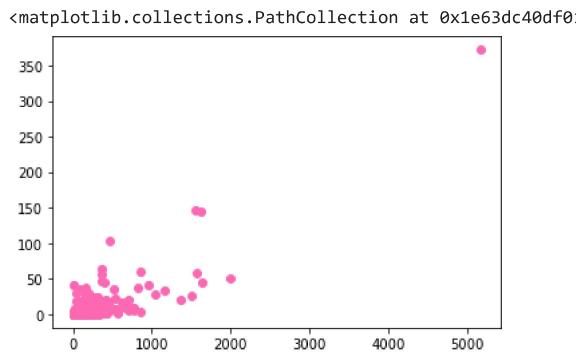
```
<matplotlib.collections.PathCollection at 0x1e63f4181c0>
```



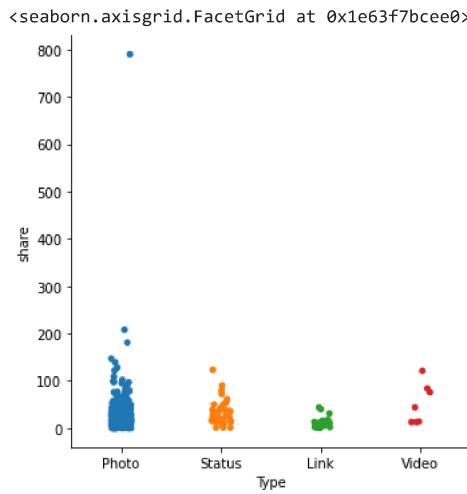
```
x=df['Post Weekday']
y=df['share']
plt.scatter(x, y)
```



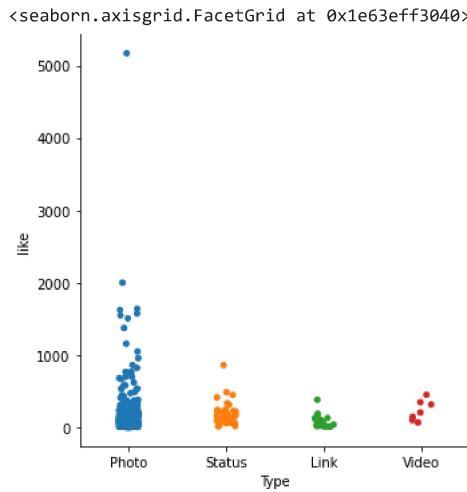
```
x=df['like']
y=df['comment']
plt.scatter(x, y,color = 'hotpink')
```



```
sns.catplot(x='Type',y='share',data=df)
```

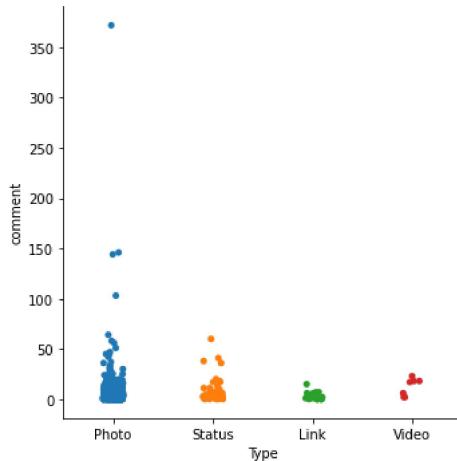


```
sns.catplot(x='Type',y='like',data=df)
```



```
sns.catplot(x='Type',y='comment',data=df)
```

```
<seaborn.axisgrid.FacetGrid at 0x1e63f948ee0>
```



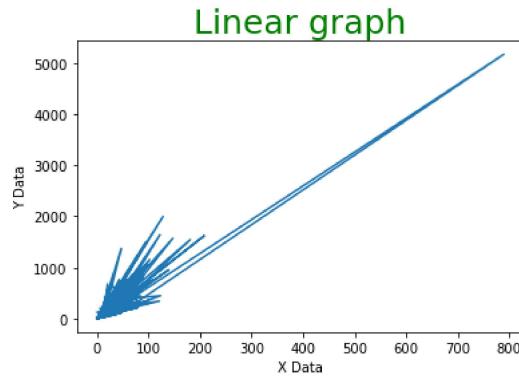
```
# plotting the data
x=df['share']
y=df['like']
plt.plot(x,y)

# add title to the plot
plt.title("Linear graph", fontsize =25, color="green")

# add label to the x-axis and y-axis

plt.xlabel("X Data")
plt.ylabel("Y Data")

Text(0, 0.5, 'Y Data')
```



```
df1=pd.read_csv("D:/AirQuality.csv",encoding='cp1252')
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_10164\3877287812.py:1: DtypeWarning: Columns (0) have mixed types. Specify dtype option on import or set low_memory=False.
```

```
df1=df1.drop(['stn_code','agency', 'location_monitoring_station'],axis=1)
```

```
df1 = df1.dropna(subset=['date'])
```

```
# define columns of importance, which shall be used regularly
```

```
COLS = ['so2','no2', 'rspm', 'spm', 'pm2_5']
```

```
df1.isna().sum()
```

sampling_date	0
state	0
location	0
type	5390
so2	34643
no2	16230
rspm	40219

```
spm           237380
pm2_5         426421
date           0
dtype: int64
```

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 435735 entries, 0 to 435738
Data columns (total 10 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   sampling_date 435735 non-null   object  
 1   state          435735 non-null   object  
 2   location        435735 non-null   object  
 3   type            430345 non-null   object  
 4   so2             401092 non-null   float64 
 5   no2             419505 non-null   float64 
 6   rspm            395516 non-null   float64 
 7   spm              198355 non-null   float64 
 8   pm2_5           9314 non-null    float64 
 9   date            435735 non-null   object  
dtypes: float64(5), object(5)
memory usage: 36.6+ MB
```

```
df1 = df1.dropna(subset=['type'])
df1 = df1.dropna(subset=['so2'])
df1 = df1.dropna(subset=['no2'])
df1 = df1.dropna(subset=['rspm'])
df1 = df1.dropna(subset=['spm'])
df1 = df1.dropna(subset=['pm2_5'])
```

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 0 entries
Data columns (total 10 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   sampling_date 0 non-null     object  
 1   state          0 non-null     object  
 2   location        0 non-null     object  
 3   type            0 non-null     object  
 4   so2             0 non-null     float64 
 5   no2             0 non-null     float64 
 6   rspm            0 non-null     float64 
 7   spm              0 non-null     float64 
 8   pm2_5           0 non-null     float64 
 9   date            0 non-null     object  
dtypes: float64(5), object(5)
memory usage: 0.0+ bytes
```