In [1]:

```
"""1.Perform the following operations using Python on the Facebook metrics data sets
a. Create data subsets
b. Merge Data
c. Sort Data
d. Transposing Data
e. Shape and reshape Data"""
```

Out[1]:

```
'1.Perform the following operations using Python on the Facebook metri
cs data sets\na. Create data subsets\nb. Merge Data\nc. Sort Data\nd.
Transposing Data\ne. Shape and reshape Data'
```

In [172]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
#import warnings
#%matplotlib inline
```

In [173]:

```
df = pd.read_csv('./Part-B_DSBD/Assignment 1/dataset_Facebook.csv',delimiter=';')
```

In [8]:

```
df.describe()
```

Out[8]:

	Page total likes	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach
count	500.000000	500.000000	500.000000	500.000000	500.000000	499.000000	500.00000
mean	123194.176000	1.880000	7.038000	4.150000	7.840000	0.278557	13903.36000
std	16272.813214	0.852675	3.307936	2.030701	4.368589	0.448739	22740.78789
min	81370.000000	1.000000	1.000000	1.000000	1.000000	0.000000	238.00000
25%	112676.000000	1.000000	4.000000	2.000000	3.000000	0.000000	3315.00000
50%	129600.000000	2.000000	7.000000	4.000000	9.000000	0.000000	5281.00000
75%	136393.000000	3.000000	10.000000	6.000000	11.000000	1.000000	13168.00000
max	139441.000000	3.000000	12.000000	7.000000	23.000000	1.000000	180480.00000

In [9]:

```
df.shape
```

Out[9]:

```
(500, 19)
```

In [10]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 19 columns):
 #   Column
Non-Null Count  Dtype
---  ------
--------------  -----
 0   Page total likes
500 non-null    int64
 1   Type
500 non-null    object
 2   Category
500 non-null    int64
 3   Post Month
500 non-null    int64
 4   Post Weekday
500 non-null    int64
 5   Post Hour
500 non-null    int64
 6   Paid
499 non-null    float64
 7   Lifetime Post Total Reach
500 non-null    int64
 8   Lifetime Post Total Impressions
500 non-null    int64
 9   Lifetime Engaged Users
500 non-null    int64
 10  Lifetime Post Consumers
500 non-null    int64
 11  Lifetime Post Consumptions
500 non-null    int64
 12  Lifetime Post Impressions by people who have liked your Page
500 non-null    int64
 13  Lifetime Post reach by people who like your Page
500 non-null    int64
 14  Lifetime People who have liked your Page and engaged with your po
st  500 non-null    int64
 15  comment
500 non-null    int64
 16  like
499 non-null    float64
 17  share
496 non-null    float64
 18  Total Interactions
500 non-null    int64
dtypes: float64(3), int64(15), object(1)
memory usage: 74.3+ KB
```

In [11]:

```
df.columns
```

Out[11]:

```
Index(['Page total likes', 'Type', 'Category', 'Post Month', 'Post Wee
kday',
       'Post Hour', 'Paid', 'Lifetime Post Total Reach',
       'Lifetime Post Total Impressions', 'Lifetime Engaged Users',
       'Lifetime Post Consumers', 'Lifetime Post Consumptions',
       'Lifetime Post Impressions by people who have liked your Page',
       'Lifetime Post reach by people who like your Page',
       'Lifetime People who have liked your Page and engaged with your
post',
       'comment', 'like', 'share', 'Total Interactions'],
      dtype='object')
```

In [12]:

```
df.head(10)
```

Out[12]:

	Page total likes	Type	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach	Lifetime Post Total Impressions	Lifetime Engaged Users	Co
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	
5	139441	Status	2	12	1	9	0.0	10472	20849	1191	
6	139441	Photo	3	12	1	3	1.0	11692	19479	481	
7	139441	Photo	3	12	7	9	1.0	13720	24137	537	
8	139441	Status	2	12	7	3	0.0	11844	22538	1530	
9	139441	Photo	3	12	6	10	0.0	4694	8668	280	

In [14]:

```
subset1=df[10:40]
subset1
```

Out[14]:

	Page total likes	Type	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach	Lifetime Post Total Impressions	Lifetime Engaged Users	C
10	139441	Status	2	12	5	10	0.0	21744	42334	4258	
11	139441	Photo	2	12	5	10	0.0	3112	5590	208	
12	139441	Photo	2	12	5	10	0.0	2847	5133	193	
13	139441	Photo	2	12	5	3	0.0	2549	4896	249	
14	138414	Photo	2	12	4	5	1.0	22784	39941	887	
15	138414	Status	2	12	3	10	0.0	10060	19680	1264	
16	138414	Photo	3	12	3	3	0.0	1722	2981	163	
17	138414	Photo	1	12	2	12	1.0	53264	111785	1706	
18	138414	Status	3	12	2	3	0.0	3930	7509	130	
19	138414	Photo	3	12	1	11	0.0	1591	2825	121	
20	138414	Photo	2	12	1	3	0.0	2848	5066	200	
21	138414	Photo	1	12	7	10	0.0	1384	2467	15	
22	138414	Link	1	12	7	10	0.0	3454	6853	118	
23	138414	Photo	3	12	7	3	0.0	2723	4888	176	
24	138414	Status	2	12	6	10	0.0	8488	15294	1341	
25	138458	Status	2	12	6	3	0.0	8284	15104	1521	
26	138458	Status	2	12	5	11	0.0	19552	34143	2806	
27	138458	Photo	3	12	5	3	0.0	2478	4306	212	
28	138895	Photo	2	12	5	3	0.0	9560	18264	973	
29	138895	Video	1	12	4	11	1.0	36208	61262	1141	
30	138895	Photo	2	12	4	2	0.0	4940	9390	385	
31	138895	Photo	2	12	3	10	0.0	1683	2929	192	
32	138895	Photo	3	12	3	3	0.0	5280	9578	368	
33	138895	Photo	3	12	2	9	0.0	3002	5318	268	
34	138895	Photo	1	12	2	3	0.0	3766	7149	298	
35	138895	Photo	2	12	1	11	0.0	4512	7808	423	
36	138895	Photo	3	12	1	3	0.0	2690	4628	252	

	Page total likes	Type	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach	Lifetime Post Total Impressions	Lifetime Engaged Users	C
37	138895	Photo	1	12	7	10	1.0	19800	28663	479	
38	138895	Status	2	12	7	9	0.0	17576	33058	5352	
39	138895	Photo	1	12	7	3	0.0	3290	6085	306	

In [16]:

```
df.iloc[2:5] # row no 2,3 & 4 extracted
#df.iloc[:5]  # First 5 rows
```

Out[16]:

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

In [17]:

```
df.iloc[:3,0:4] # first 3 rows for column 1,2,3
```

Out[17]:

	Page total likes	Type	Category	Post Month
0	139441	Photo	2	12
1	139441	Status	2	12
2	139441	Photo	3	12

In [18]:

```
df.iloc[[2,5,6,8,1]] # extract row 2,5,6
```

Out[18]:

	Page total likes	Type	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach	Lifetime Post Total Impressions	Lifetime Engaged Users	Co
2	139441	Photo	3	12	3	3	0.0	2413	4373	177	
5	139441	Status	2	12	1	9	0.0	10472	20849	1191	
6	139441	Photo	3	12	1	3	1.0	11692	19479	481	
8	139441	Status	2	12	7	3	0.0	11844	22538	1530	
1	139441	Status	2	12	3	10	0.0	10460	19057	1457	

In [19]:

```
df.iloc[[2,5,6,8] , [1,2]] # extract row 2,5,6 n Columns 2,3 (Type, Category)
```

Out[19]:

	Type	Category
2	Photo	3
5	Status	2
6	Photo	3
8	Status	2

In [20]:

```
df.iloc[ : ,1]
```

Out[20]:

```
0        Photo
1       Status
2        Photo
3        Photo
4        Photo
        ...
495      Photo
496      Photo
497      Photo
498      Photo
499      Photo
Name: Type, Length: 500, dtype: object
```

In [21]:

```
df.loc[ : ,'Type']
```

Out[21]:

```
0        Photo
1       Status
2        Photo
3        Photo
4        Photo
        ...
495      Photo
496      Photo
497      Photo
498      Photo
499      Photo
Name: Type, Length: 500, dtype: object
```

In [174]:

```
dtadf = df.loc[ : ,'Type']
```

In [177]:

```
pd.DataFrame(dtadf)
```

Out[177]:

	Type
0	Photo
1	Status
2	Photo
3	Photo
4	Photo
...	...
495	Photo
496	Photo
497	Photo
498	Photo
499	Photo

500 rows × 1 columns

In [23]:

```
df2col =df.loc[[2,4,5,8],['Type','Category']]
df2col
```

Out[23]:

	Type	Category
2	Photo	3
4	Photo	2
5	Status	2
8	Status	2

In [24]:

```
df.loc[df['Category']==2]
```

Out[24]:

	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
3	139441	Photo	2	12	2	10	1.0	50128	87991	2211
4	139441	Photo	2	12	2	3	0.0	7244	13594	671
5	139441	Status	2	12	1	9	0.0	10472	20849	1191
...	...	...	...	...	...	...	...	...	...	...
467	91758	Photo	2	2	5	3	1.0	15880	51571	1188
480	86909	Photo	2	1	4	11	0.0	11484	20696	1762
483	86491	Photo	2	1	3	3	0.0	5526	8779	1096
496	81370	Photo	2	1	5	8	0.0	3480	6229	537
499	81370	Photo	2	1	4	4	NaN	4188	7292	564

130 rows × 19 columns

In [25]:

```
df[df['Lifetime Engaged Users']<500]
```

Out[25]:

	Page total likes	Type	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach	Lifetime Post Total Impressions	Lifetime Engaged Users	C
0	139441	Photo	2	12	4	3	0.0	2752	5091	178	
2	139441	Photo	3	12	3	3	0.0	2413	4373	177	
6	139441	Photo	3	12	1	3	1.0	11692	19479	481	
9	139441	Photo	3	12	6	10	0.0	4694	8668	280	
11	139441	Photo	2	12	5	10	0.0	3112	5590	208	
...	...	...	...	...	...	...	...	...	...	...	
438	98195	Link	2	3	6	6	0.0	5730	10083	103	
441	98195	Photo	1	3	5	4	1.0	1845	2670	9	
470	91437	Link	1	2	3	13	0.0	9356	14986	448	
481	86491	Link	1	1	4	4	1.0	4938	7910	66	
485	86491	Link	1	1	2	2	0.0	5168	8371	66	

168 rows × 19 columns

In [26]:

```
df[(df['Lifetime Post Consumptions']<1000 ) & (df['Paid']==1.0)]
```

Out[26]:

	Page total likes	Type	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach	Lifetime Post Total Impressions	Lifetime Engaged Users	(
6	139441	Photo	3	12	1	3	1.0	11692	19479	481	
7	139441	Photo	3	12	7	9	1.0	13720	24137	537	
14	138414	Photo	2	12	4	5	1.0	22784	39941	887	
37	138895	Photo	1	12	7	10	1.0	19800	28663	479	
41	138895	Link	1	12	6	3	1.0	18480	28438	517	
...	...	...	...	...	...	...	...	...	...	...	
412	106928	Photo	1	4	4	3	1.0	5290	8132	612	
413	104070	Photo	1	3	5	14	1.0	3696	5824	517	
441	98195	Photo	1	3	5	4	1.0	1845	2670	9	
445	96749	Photo	1	3	1	13	1.0	4412	8070	711	
481	86491	Link	1	1	4	4	1.0	4938	7910	66	

79 rows × 19 columns

Making DataFrames from the Dtaset

In [27]:

```
xd = df.loc[[0,1,2,3],['Type','Category']]
dfx = pd.DataFrame(xd)
dfx
# dfx = dfx.set_index('Type')
# dfx
```

Out[27]:

	Type	Category
0	Photo	2
1	Status	2
2	Photo	3
3	Photo	2

In [28]:

```
dfy = pd.DataFrame(df.loc[[0,1,2,3],['Post Weekday','Post Hour'] ])
dfy
```

Out[28]:

	Post Weekday	Post Hour
0	4	3
1	3	10
2	3	3
3	2	10

In [29]:

```
dfz = pd.DataFrame(df.loc[[0,1,2,3],['comment','like']])
dfz
```

Out[29]:

	comment	like
0	4	79.0
1	5	130.0
2	0	66.0
3	58	1572.0

In [32]:

```
pd.concat([dfx,dfy,dfz], axis=1)
```

Out[32]:

	Type	Category	Post Weekday	Post Hour	comment	like
0	Photo	2	4	3	4	79.0
1	Status	2	3	10	5	130.0
2	Photo	3	3	3	0	66.0
3	Photo	2	2	10	58	1572.0

In [33]:

```
pd.concat([dfx,dfy,dfz], axis=0)
```

Out[33]:

	Type	Category	Post Weekday	Post Hour	comment	like
0	Photo	2.0	NaN	NaN	NaN	NaN
1	Status	2.0	NaN	NaN	NaN	NaN
2	Photo	3.0	NaN	NaN	NaN	NaN
3	Photo	2.0	NaN	NaN	NaN	NaN
0	NaN	NaN	4.0	3.0	NaN	NaN
1	NaN	NaN	3.0	10.0	NaN	NaN
2	NaN	NaN	3.0	3.0	NaN	NaN
3	NaN	NaN	2.0	10.0	NaN	NaN
0	NaN	NaN	NaN	NaN	4.0	79.0
1	NaN	NaN	NaN	NaN	5.0	130.0
2	NaN	NaN	NaN	NaN	0.0	66.0
3	NaN	NaN	NaN	NaN	58.0	1572.0

Create two dataframes for facebook dataset with common column

In [34]:

```
# create two dataframes for facebook dataset with common column
ldataset=df.iloc[:,0:10]
rdataset=df.iloc[:,9:20]
ldataset.head(5)
```

Out[34]:

	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

In [35]:

```
rdataset.head(5)
```

Out[35]:

	Lifetime Engaged Users	Lifetime Post Consumers	Lifetime Post Consumptions	Lifetime Post Impressions by people who have liked your Page	Lifetime Post reach by people who like your Page	Lifetime People who have liked your Page and engaged with your post	comment	like	sha
0	178	109	159	3078	1640	119	4	79.0	17
1	1457	1361	1674	11710	6112	1108	5	130.0	29
2	177	113	154	2812	1503	132	0	66.0	14
3	2211	790	1119	61027	32048	1386	58	1572.0	147
4	671	410	580	6228	3200	396	19	325.0	49

In [38]:

```
merge_result=pd.merge(ldataset,rdataset, on='Lifetime Engaged Users')
merge_result
#merge_result.head(5)
```

Out[38]:

	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
...	...	...	...	...	...	...	...	...	...	...
701	85979	Photo	3	1	1	2	0.0	4908	7491	957
702	85979	Photo	3	1	7	2	0.0	4800	7754	975
703	85979	Photo	3	1	6	3	1.0	6184	10228	956
704	85093	Photo	3	1	7	10	0.0	5400	9218	810
705	85093	Photo	3	1	7	2	0.0	4684	7536	733

706 rows × 19 columns

merge_result.shape

In [39]:

```
merge_result.shape
```

Out[39]:

```
(706, 19)
```

In [42]:

```
merge_result1=pd.merge(ldataset,rdataset[['Lifetime Engaged Users','Lifetime Post Co
merge_result1
```

Out[42]:

	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
...	...	...	...	...	...	...	...	...	...	...
701	85979	Photo	3	1	1	2	0.0	4908	7491	957
702	85979	Photo	3	1	7	2	0.0	4800	7754	975
703	85979	Photo	3	1	6	3	1.0	6184	10228	956
704	85093	Photo	3	1	7	10	0.0	5400	9218	810
705	85093	Photo	3	1	7	2	0.0	4684	7536	733

706 rows × 11 columns

In [44]:

```
common_values=ldataset['Lifetime Engaged Users'].isin(rdataset['Lifetime Engaged Use
common_values
```

Out[44]:

```
True    500
Name: Lifetime Engaged Users, dtype: int64
```

rightjoin=pd.merge(ldataset,rdataset[['Lifetime Engaged Users','Lifetime Post Consumers']],how='right')

innerjoin=pd.merge(ldataset,rdataset[['Lifetime Engaged Users','Lifetime Post Consumers']],how='inner')

outerjoin=pd.merge(ldataset,rdataset[['Lifetime Engaged Users','Lifetime Post Consumers']],how='outer')

In [47]:

```
leftjoin=pd.merge(ldataset,rdataset[['Lifetime Engaged Users','Lifetime Post Consume
leftjoin.head()
```

Out[47]:

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

In [48]:

```
rightjoin=pd.merge(ldataset,rdataset[['Lifetime Engaged Users','Lifetime Post Consum
rightjoin.head()
```

Out[48]:

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

Sort

In [49]:

```
sd=df.sort_values('Lifetime Post Total Reach')
```

In [50]:

```
sd.head(10)
```

Out[50]:

	Page total likes	Type	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach	Lifetime Post Total Impressions	Lifetime Engaged Users	C
422	102112	Photo	1	3	1	19	0.0	238	570	143	
426	100732	Photo	1	3	7	18	0.0	391	746	131	
432	100732	Photo	1	3	6	17	0.0	452	726	186	
120	136393	Photo	1	10	7	9	0.0	584	1029	273	
123	136393	Photo	1	10	7	7	0.0	617	1071	229	
122	136393	Photo	1	10	7	8	0.0	619	1096	257	
128	136393	Photo	1	10	6	13	0.0	645	1117	195	
133	136393	Photo	1	10	6	8	0.0	652	1331	214	
132	136393	Photo	1	10	6	9	0.0	659	1158	199	
125	136393	Photo	1	10	7	5	0.0	677	1240	236	

In [63]:

```
df[df['Lifetime Post Total Reach']<500]
```

Out[63]:

	Page total likes	Type	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach	Lifetime Post Total Impressions	Lifetime Engaged Users	C
422	102112	Photo	1	3	1	19	0.0	238	570	143	
426	100732	Photo	1	3	7	18	0.0	391	746	131	
432	100732	Photo	1	3	6	17	0.0	452	726	186	

In [62]:

```
ab = df['Lifetime Post Total Reach']<300
dxf =pd.DataFrame(ab)
dxf
```

Out[62]:

	Lifetime Post Total Reach
0	False
1	False
2	False
3	False
4	False
...	...
495	False
496	False
497	False
498	False
499	False

500 rows × 1 columns

In [66]:

```
sd1=df.sort_values('Lifetime Post Total Reach',ascending=False)
sd1.head(5)
```

Out[66]:

	Page total likes	Type	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach	Lifetime Post Total Impressions	Lifetime Engaged Users	C
244	130791	Photo	2	7	3	5	1.0	180480	319133	8072	
464	92079	Photo	1	2	6	13	0.0	158208	453213	2482	
463	92186	Photo	3	2	7	2	1.0	153536	497910	1713	
277	126424	Video	1	6	2	13	0.0	139008	277100	1779	
380	111620	Photo	1	4	7	14	0.0	128064	251269	1539	

In [78]:

```
sd1.sort_values('Page total likes',inplace=False ,ascending=False)
```

Out[78]:

	Page total likes	Type	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach	Lifetime Post Total Impressions	Lifetime Engaged Users
8	139441	Status	2	12	7	3	0.0	11844	22538	1530
1	139441	Status	2	12	3	10	0.0	10460	19057	1457
11	139441	Photo	2	12	5	10	0.0	3112	5590	208
12	139441	Photo	2	12	5	10	0.0	2847	5133	193
10	139441	Status	2	12	5	10	0.0	21744	42334	4258
...	...	...	...	...	...	...	...	...	...	...
495	85093	Photo	3	1	7	2	0.0	4684	7536	733
499	81370	Photo	2	1	4	4	NaN	4188	7292	564
498	81370	Photo	3	1	4	11	0.0	4156	7564	626
497	81370	Photo	1	1	5	2	0.0	3778	7216	625
496	81370	Photo	2	1	5	8	0.0	3480	6229	537

500 rows × 19 columns

In [87]:

```
abmin = df['Page total likes'].min()
abmin
```

Out[87]:

```
81370
```

In [92]:

```
mindf = df[df['Page total likes']== 81370]
#mindf = df[df['Page total likes']== abmin]
mindf.head(10)
```

Out[92]:

	Page total likes	Type	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach	Lifetime Post Total Impressions	Lifetime Engaged Users	C
496	81370	Photo	2	1	5	8	0.0	3480	6229	537	
497	81370	Photo	1	1	5	2	0.0	3778	7216	625	
498	81370	Photo	3	1	4	11	0.0	4156	7564	626	
499	81370	Photo	2	1	4	4	NaN	4188	7292	564	

In [93]:

```
abmax = df['Page total likes'].max()
abmax
```

Out[93]:

```
139441
```

In [99]:

```
maxdf = df[df['Page total likes'] == 139441]
#maxdf = df[df['Page total likes'] == abmax]
maxdf
```

Out[99]:

	Page total likes	Type	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach	Lifetime Post Total Impressions	Lifetime Engaged Users	C
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	
5	139441	Status	2	12	1	9	0.0	10472	20849	1191	
6	139441	Photo	3	12	1	3	1.0	11692	19479	481	
7	139441	Photo	3	12	7	9	1.0	13720	24137	537	
8	139441	Status	2	12	7	3	0.0	11844	22538	1530	
9	139441	Photo	3	12	6	10	0.0	4694	8668	280	
10	139441	Status	2	12	5	10	0.0	21744	42334	4258	
11	139441	Photo	2	12	5	10	0.0	3112	5590	208	
12	139441	Photo	2	12	5	10	0.0	2847	5133	193	
13	139441	Photo	2	12	5	3	0.0	2549	4896	249	

Transform T

In [100]:

```
transdf=df.T
print(transdf)
```

```
                                                    0       1
2    \
Page total likes                                139441  139441  13
9441
Type                                             Photo  Status   P
hoto
Category                                             2       2
3
Post Month                                          12      12
12
Post Weekday                                         4       3
3
Post Hour                                            3      10
3
Paid                                               0.0     0.0
0.0
Lifetime Post Total Reach                         2752   10460
2413
Lifetime Post Total Impressions                   5091   19057
```

In [101]:

```
transform = df.T
#tdf = pd.DataFrame(transform)
#tdf
transform
```

Out[101]:

	0	1	2	3	4	5	6	7	8	
Page total likes	139441	139441	139441	139441	139441	139441	139441	139441	139441	139
Type	Photo	Status	Photo	Photo	Photo	Status	Photo	Photo	Status	Ph
Category	2	2	3	2	2	2	3	3	2	
Post Month	12	12	12	12	12	12	12	12	12	
Post Weekday	4	3	3	2	2	1	1	7	7	
Post Hour	3	10	3	10	3	9	3	9	3	
Paid	0.0	0.0	0.0	1.0	0.0	0.0	1.0	1.0	0.0	
Lifetime Post Total Reach	2752	10460	2413	50128	7244	10472	11692	13720	11844	4(
Lifetime Post Total Impressions	5091	19057	4373	87991	13594	20849	19479	24137	22538	8(
Lifetime Engaged Users	178	1457	177	2211	671	1191	481	537	1530	2
Lifetime Post Consumers	109	1361	113	790	410	1073	265	232	1407	1
Lifetime Post Consumptions	159	1674	154	1119	580	1389	364	305	1692	2
Lifetime Post Impressions by people who have liked your Page	3078	11710	2812	61027	6228	16034	15432	19728	15220	4:
Lifetime Post reach by people who like your Page	1640	6112	1503	32048	3200	7852	9328	11056	7912	2:
Lifetime People who have liked your Page and engaged with your post	119	1108	132	1386	396	1016	379	422	1250	1
comment	4	5	0	58	19	1	3	0	0	
like	79.0	130.0	66.0	1572.0	325.0	152.0	249.0	325.0	161.0	11
share	17.0	29.0	14.0	147.0	49.0	33.0	27.0	14.0	31.0	2
Total Interactions	100	164	80	1777	393	186	279	339	192	1

19 rows × 500 columns

Making Array from the dataFrames

In [102]:

```
import numpy as np
dfshape = np.array(pd.DataFrame(df.loc[[0,1,2,3],['Type','Category']]))
dfshape
```

Out[102]:

```
array([['Photo', 2],
       ['Status', 2],
       ['Photo', 3],
       ['Photo', 2]], dtype=object)
```

In [103]:

```
dfshape.shape
```

Out[103]:

```
(4, 2)
```

In [104]:

```
dfres =dfshape.reshape(2,4)
dfres
```

Out[104]:

```
array([['Photo', 2, 'Status', 2],
       ['Photo', 3, 'Photo', 2]], dtype=object)
```

In [105]:

```
print(dfres.shape)
```

```
(2, 4)
```

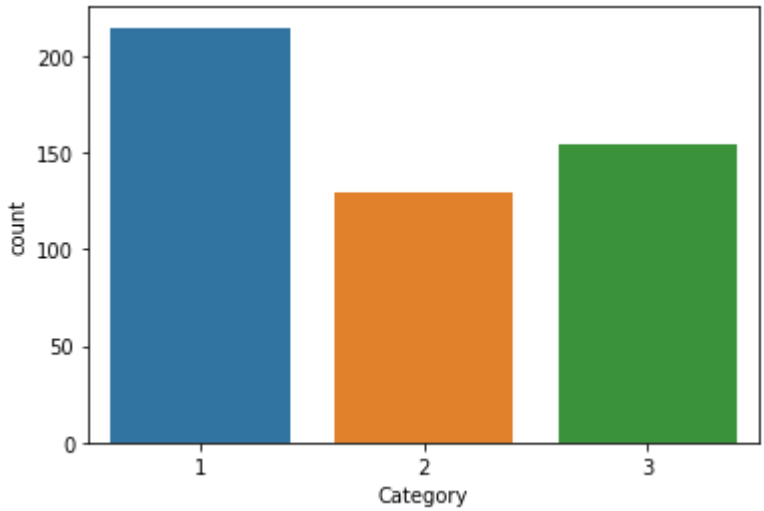
Visualisation

In [106]:

```
sns.countplot( x='Category', data=df)
```

Out[106]:

```
<AxesSubplot:xlabel='Category', ylabel='count'>
```

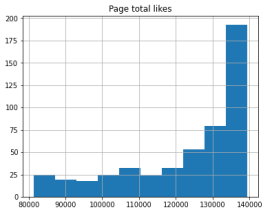


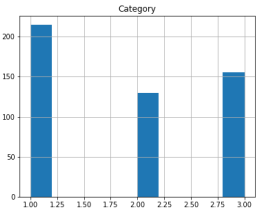
In [107]:

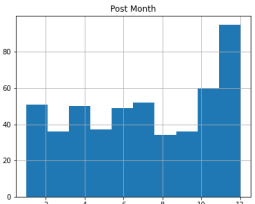
```
# plotting variation graphs for each property
df.hist(figsize = (30,30))
```

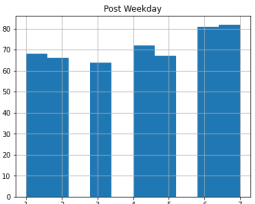
Out[107]:

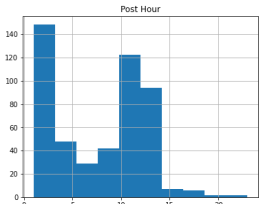
```
array([[<AxesSubplot:title={'center':'Page total likes'}>,
        <AxesSubplot:title={'center':'Category'}>,
        <AxesSubplot:title={'center':'Post Month'}>,
        <AxesSubplot:title={'center':'Post Weekday'}>],
       [<AxesSubplot:title={'center':'Post Hour'}>,
        <AxesSubplot:title={'center':'Paid'}>,
        <AxesSubplot:title={'center':'Lifetime Post Total Reach'}>,
        <AxesSubplot:title={'center':'Lifetime Post Total Impression
s'}>],
       [<AxesSubplot:title={'center':'Lifetime Engaged Users'}>,
        <AxesSubplot:title={'center':'Lifetime Post Consumers'}>,
        <AxesSubplot:title={'center':'Lifetime Post Consumptions'}>,
        <AxesSubplot:title={'center':'Lifetime Post Impressions by peo
ple who have liked your Page'}>],
       [<AxesSubplot:title={'center':'Lifetime Post reach by people wh
o like your Page'}>,
        <AxesSubplot:title={'center':'Lifetime People who have liked y
our Page and engaged with your post'}>,
        <AxesSubplot:title={'center':'comment'}>,
        <AxesSubplot:title={'center':'like'}>],
       [<AxesSubplot:title={'center':'share'}>,
        <AxesSubplot:title={'center':'Total Interactions'}>,
        <AxesSubplot:>, <AxesSubplot:>]], dtype=object)
```

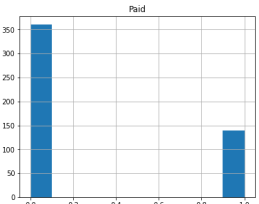

Page total likes

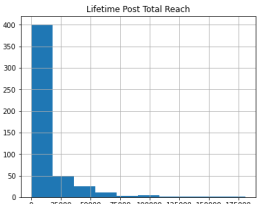

Category

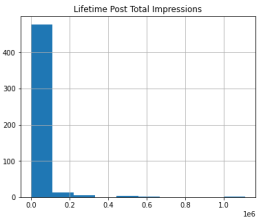

Post Month

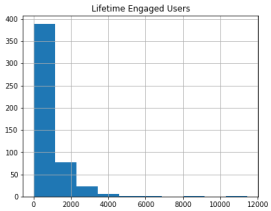

Post Weekday

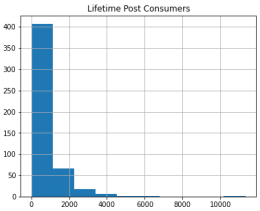

Post Hour

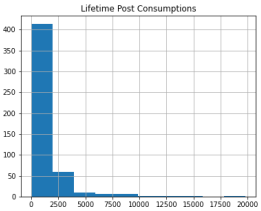

Paid

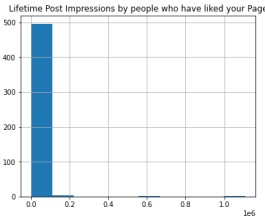

Lifetime Post Total Reach

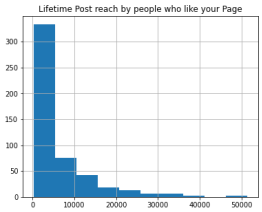

Lifetime Post Total Impressions
1e6

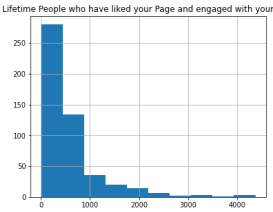

Lifetime Engaged Users

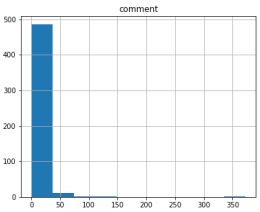

Lifetime Post Consumers

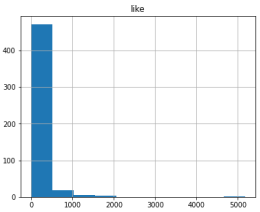

Lifetime Post Consumptions

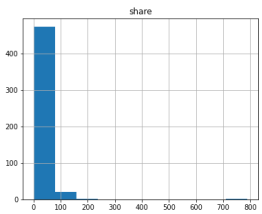

Lifetime Post Impressions by people who have liked your Page
1e6

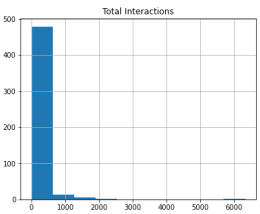

Lifetime Post reach by people who like your Page


Lifetime People who have liked your Page and engaged with your post


comment


like


share


Total Interactions

In [108]:

```
# Shows the correlation Matrix of all the columns
df.corr()
```

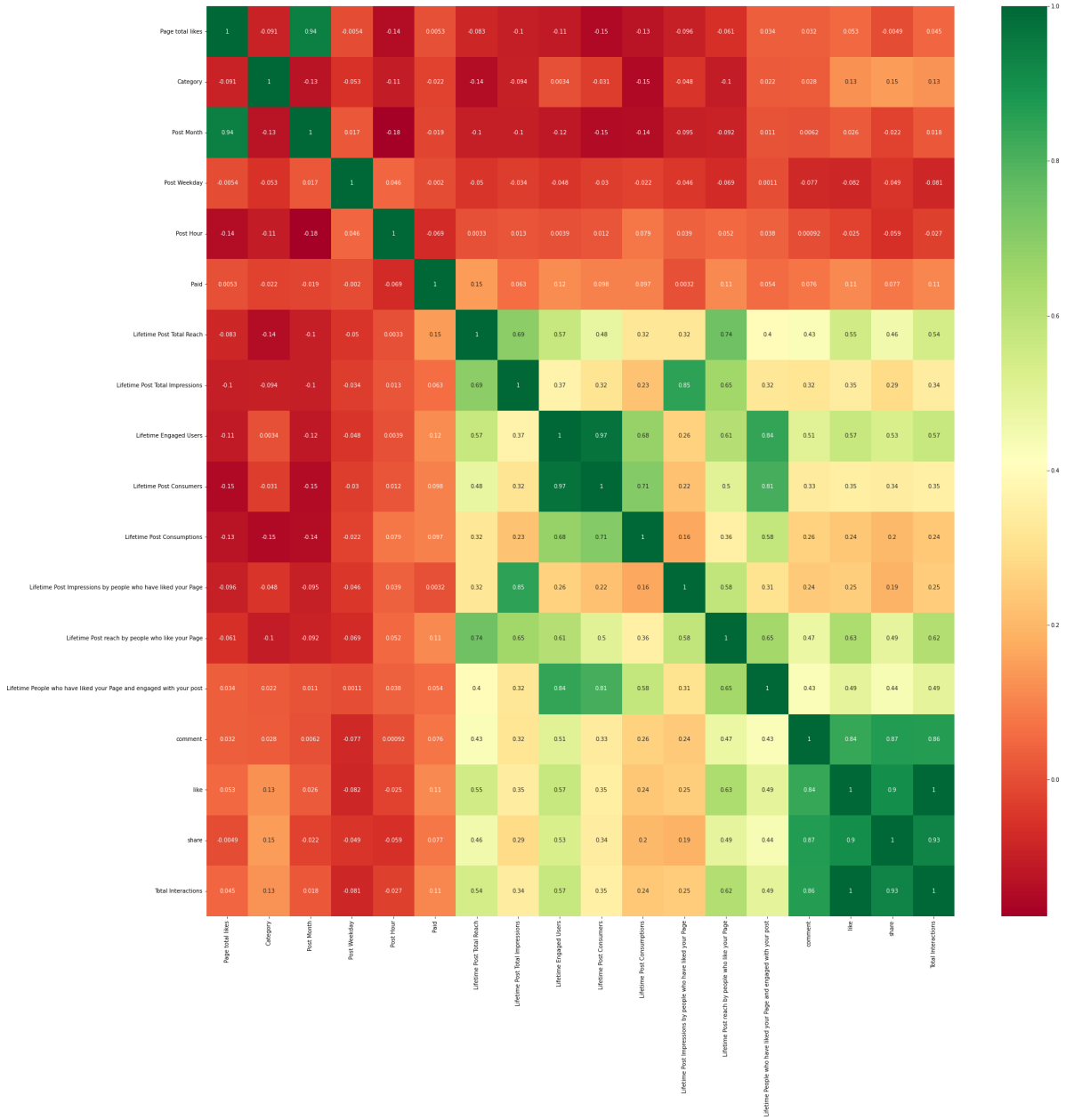
Out[108]:

	Page total likes	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach	L Pos Impre
Page total likes	1.000000	-0.091142	0.941192	-0.005401	-0.143807	0.005341	-0.083245	-0.
Category	-0.091142	1.000000	-0.127690	-0.053239	-0.107383	-0.022474	-0.142073	-0.
Post Month	0.941192	-0.127690	1.000000	0.017050	-0.176390	-0.018934	-0.102506	-0.
Post Weekday	-0.005401	-0.053239	0.017050	1.000000	0.045857	-0.001963	-0.050155	-0.
Post Hour	-0.143807	-0.107383	-0.176390	0.045857	1.000000	-0.069464	0.003338	0.
Paid	0.005341	-0.022474	-0.018934	-0.001963	-0.069464	1.000000	0.146631	0.
Lifetime Post Total Reach	-0.083245	-0.142073	-0.102506	-0.050155	0.003338	0.146631	1.000000	0.
Lifetime Post Total Impressions	-0.102540	-0.094368	-0.101616	-0.033674	0.012747	0.062564	0.694926	1.
Lifetime Engaged Users	-0.111922	0.003392	-0.115898	-0.048382	0.003879	0.117014	0.570629	0.
Lifetime Post Consumers	-0.149129	-0.031172	-0.147083	-0.029602	0.012222	0.097679	0.477908	0.
Lifetime Post Consumptions	-0.128240	-0.149443	-0.142829	-0.021565	0.078759	0.097462	0.324362	0.
Lifetime Post Impressions by people who have liked your Page	-0.096109	-0.047803	-0.094624	-0.046442	0.038892	0.003211	0.322254	0.
Lifetime Post reach by people who like your Page	-0.060516	-0.104456	-0.092012	-0.068741	0.052412	0.110043	0.743053	0.
Lifetime People who have liked your Page and engaged with your post	0.033699	0.021569	0.010956	0.001144	0.038011	0.054163	0.400756	0.
comment	0.031891	0.027842	0.006174	-0.077209	0.000922	0.075761	0.427155	0.
like	0.053276	0.126786	0.025633	-0.082322	-0.024523	0.110694	0.545185	0.

	Page total likes	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach	L Pos Impre
share	-0.004859	0.149211	-0.021859	-0.048713	-0.058680	0.076821	0.456312	0.
Total Interactions	0.045231	0.127307	0.018362	-0.081049	-0.027421	0.107739	0.538597	0.

In [109]:

```
# %matplotlib inline
corrmat = df.corr()
top_corr_features = corrmat.index
plt.figure(figsize=(30,30))
#plot heat map
g=sns.heatmap(df[top_corr_features].corr(),annot=True,cmap="RdYlGn")
```



Assignment 2

Visualize the data using Python libraries matplotlib,seaborn by plotting the following graphs on titanic dataset.

g.Bar-chart

h.Histogram

i.Pie-Chart

j.Scatter-Plot

k.Box-Plot for different features in dataset.

l.Compare different groups in given dataset

In [80]:

```
import pandas as pd
import matplotlib.pyplot as plt
# %matplotlib inline
import seaborn as sb
import numpy as np
```

In [3]:

```
df = pd.read_csv("/Users/prawate/Desktop/JupyterNotebook/Sem6/sample 4/titanic_datas
```

In [4]:

```
df
```

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500
...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500

891 rows × 12 columns

In [6]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

In [8]:

```
df.columns
```

Out[8]:

```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'Sib
Sp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

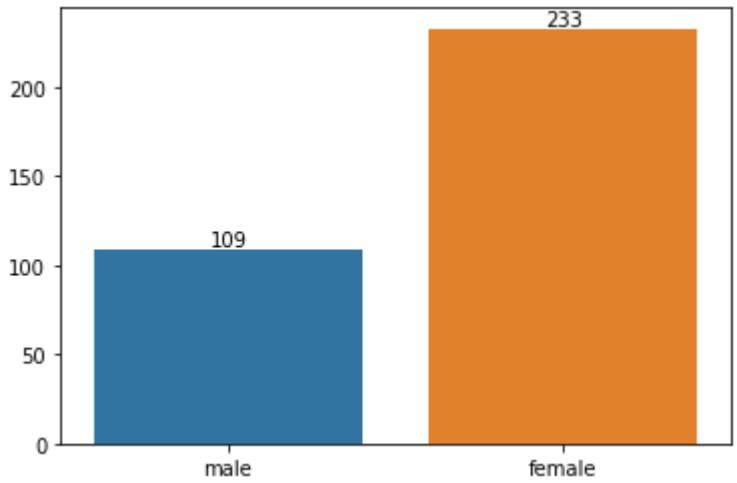
In [39]:

```
male = df[df["Sex"] == 'male']
female = df[df["Sex"] == "female"]
survived_male = df[ (df["Survived"] == 1) & (df["Sex"] == 'male')]
survived_female = df[ (df["Survived"] == 1) & (df["Sex"] == 'female')]


#seaborn == sb
seabar = sb.barplot( x = ['male','female'], y= [len(survived_male) , len(survived_fe
seabar.bar_label(seabar.containers[0])
```

Out[39]:

```
[Text(0, 0, '109'), Text(0, 0, '233')]
```



In [36]:

```
len(survived_male)
len(survived_female)
#survived_female
```

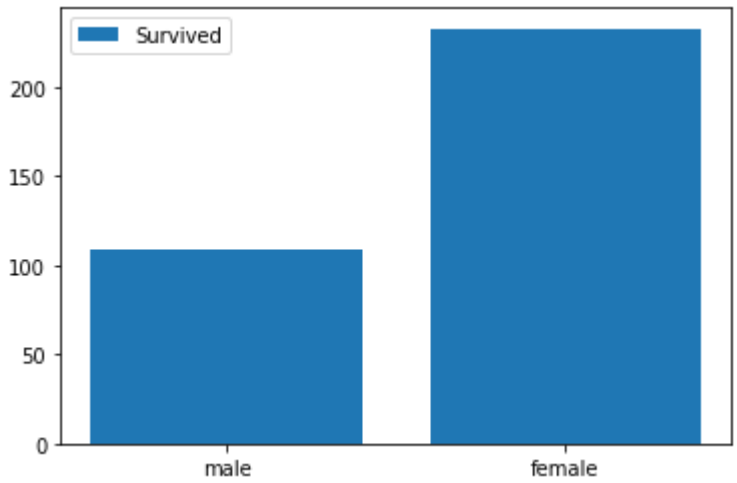
Out[36]:

```
233
```

In [77]:

```
sex = ['male', 'female']
height = [ len(survived_male) , len(survived_female)]
#height

plt.bar(sex , height, label="Survived")     #plt.bar(sex , height, color="green")
plt.legend()
# plt.legend(['Survived','Good'], loc ="upper left")
plt.show()
```



In [78]:

```
df
```

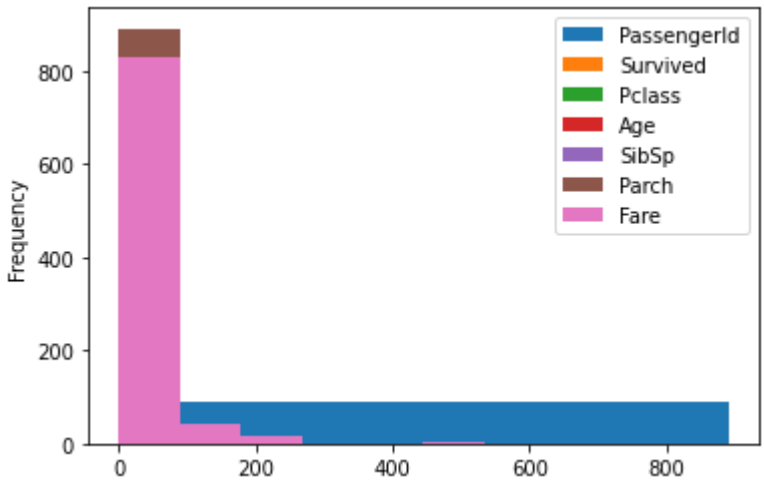
Out[78]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500
...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500

891 rows × 12 columns

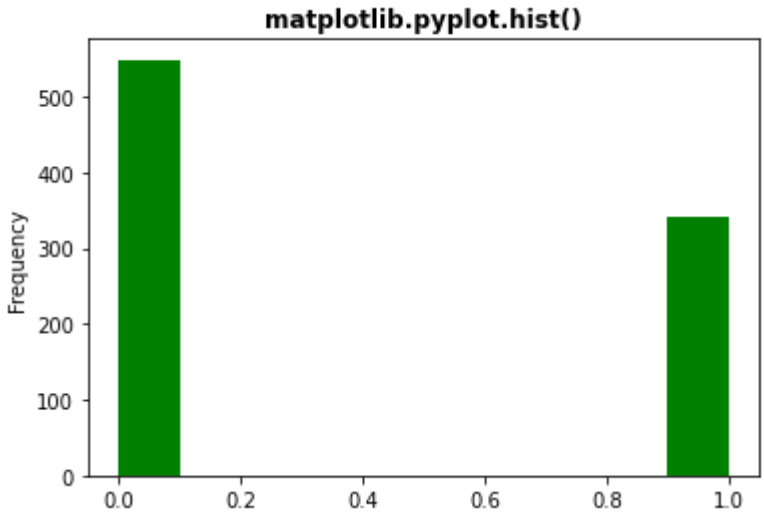
In [82]:

```
df.plot.hist()
plt.show()
```



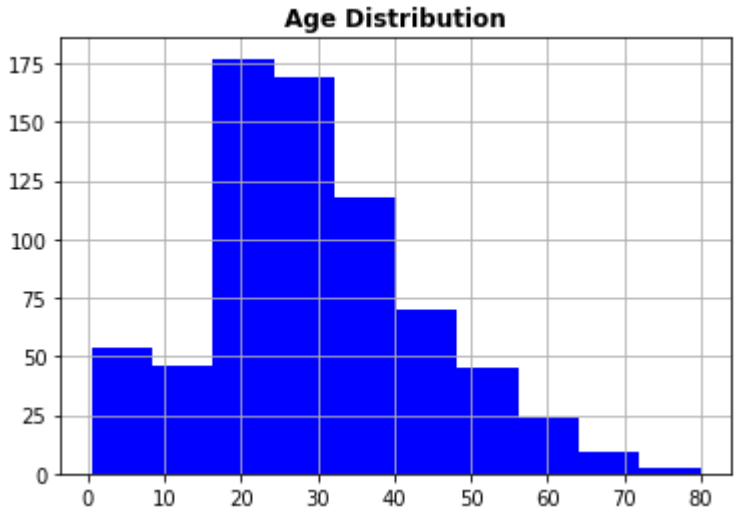
In [109]:

```
survive =df['Survived'].plot.hist( color = "green")
plt.title('matplotlib.pyplot.hist()', fontweight ="bold")
plt.show()
```



In [112]:

```
df['Age'].hist(color = "blue").plot(kind='bar')
plt.title("Age Distribution", fontweight ="bold")
plt.show()
```

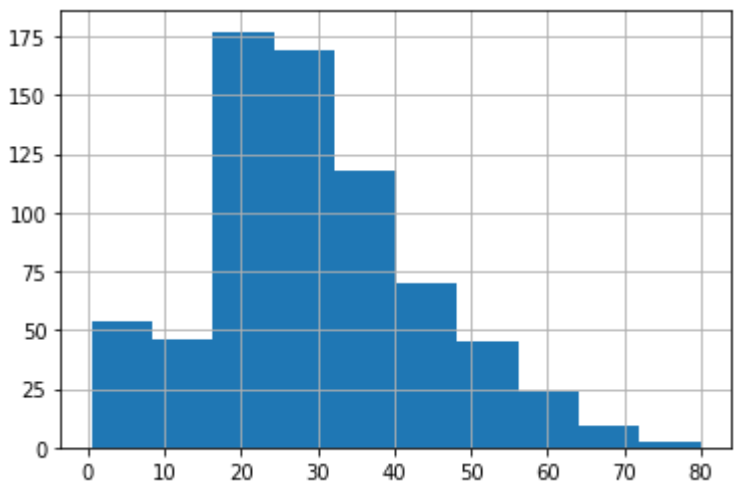


In [159]:

```
df['Age'].hist().plot()
```

Out[159]:

```
[]
```

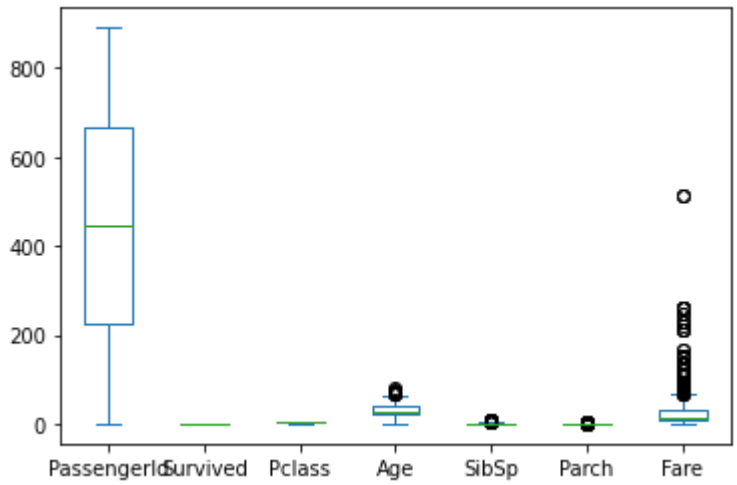


In [168]:

```
# df.plot(kind='box', subplots=False)
df.plot(kind='box')
```

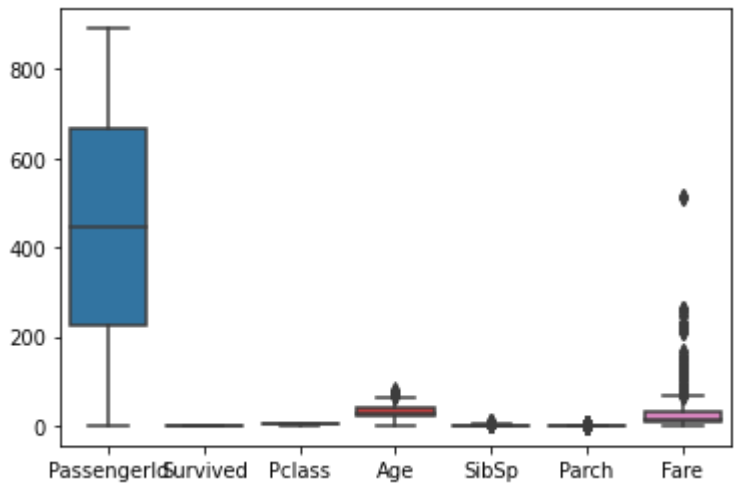
Out[168]:

<AxesSubplot:>



In [160]:

```
sb.boxplot( data = df , notch= False)
plt.show()
```



In [124]:

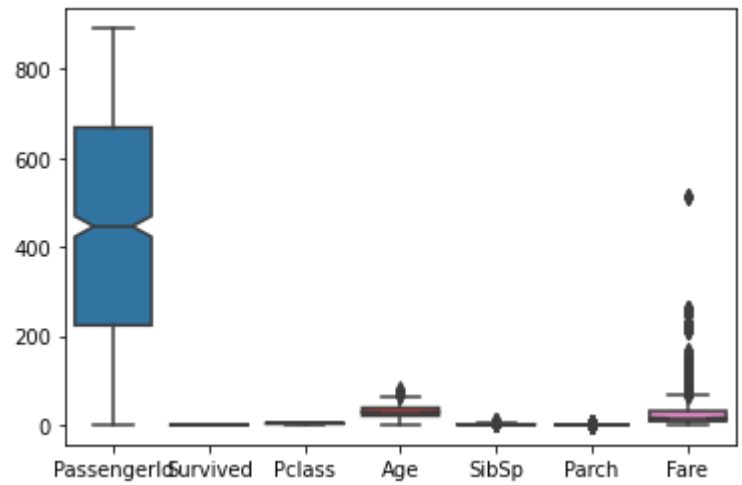
```
df['PassengerId'].describe()
```

Out[124]:

```
count    891.000000
mean     446.000000
std      257.353842
min        1.000000
25%      223.500000
50%      446.000000
75%      668.500000
max      891.000000
Name: PassengerId, dtype: float64
```

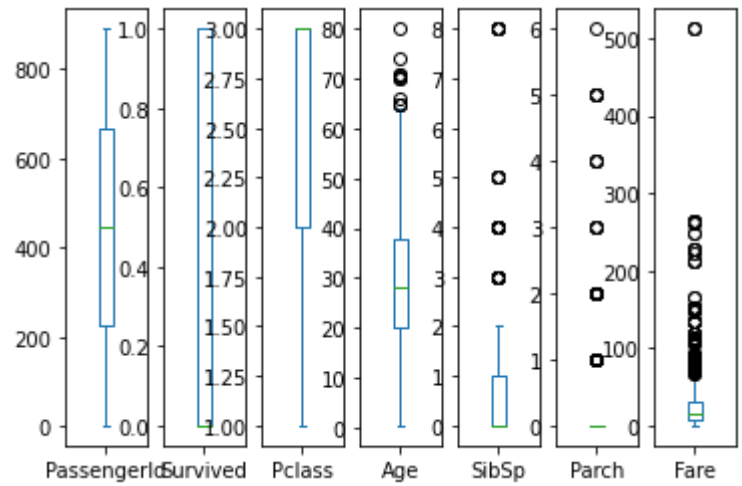
In [125]:

```
sb.boxplot( data = df , notch= True)
plt.show()
```



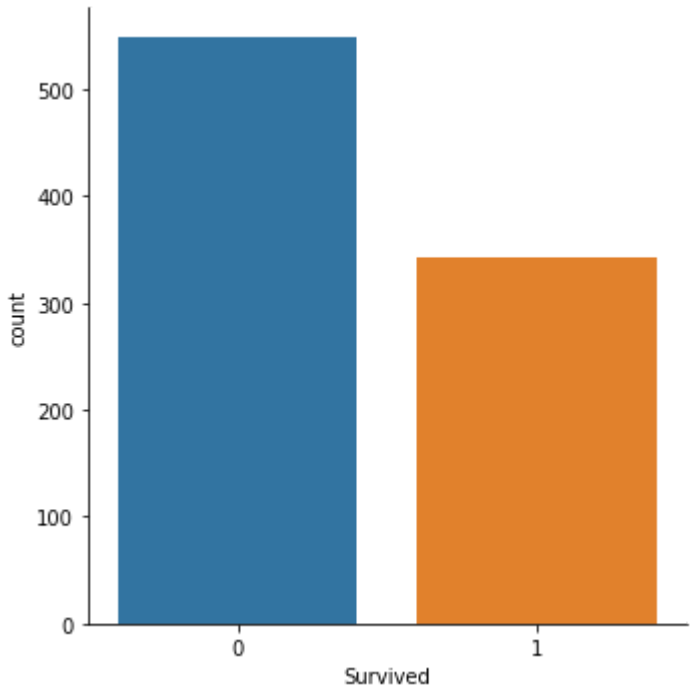
In [129]:

```
df.plot(kind='box', subplots=True)
plt.show()
```



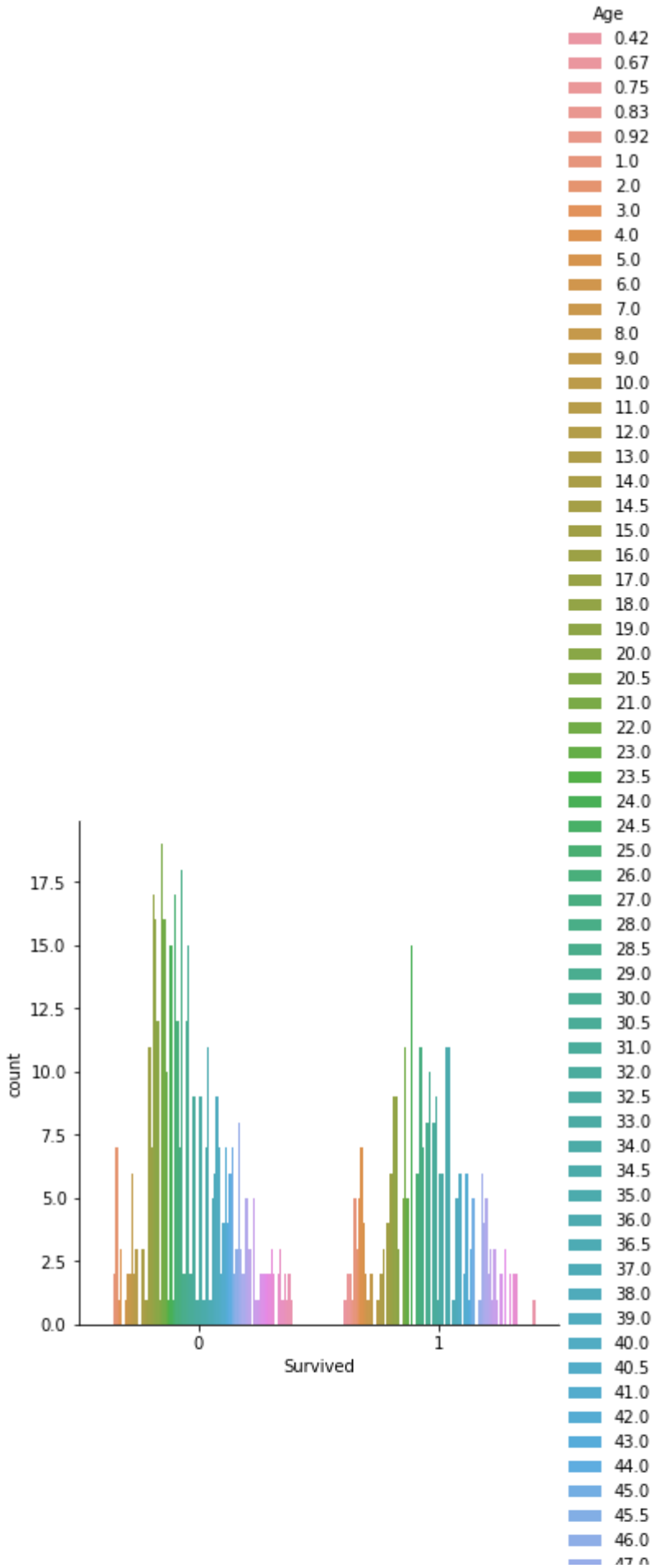
In [133]:

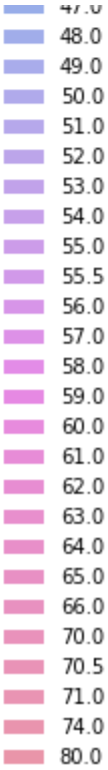
```
sb.catplot(x = 'Survived', kind='count', data = df)
plt.show()
```



In [134]:

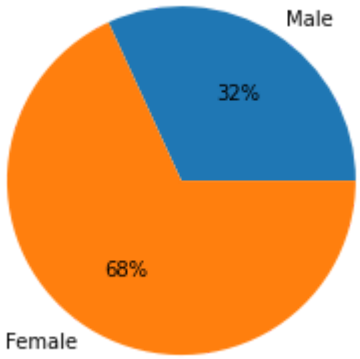
```
sb.catplot(x='Survived', hue='Age', kind='count', data=df)
plt.show()
```





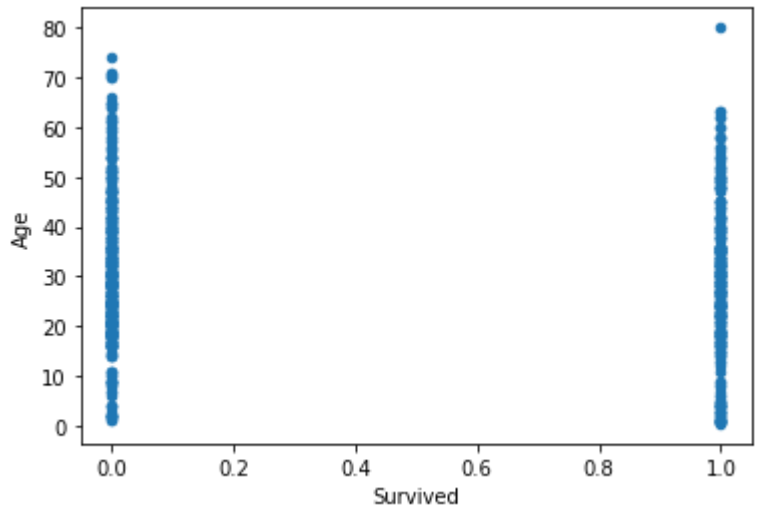
In [152]:

```
plt.pie([len(survived_male), len(survived_female)], labels=['Male', 'Female'], autop
plt.show()
```



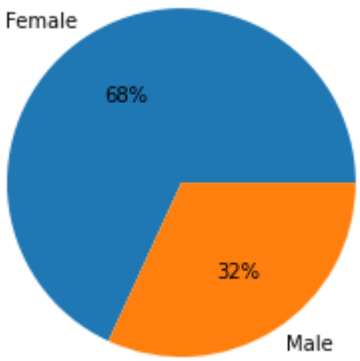
In [153]:

```
df.plot(x='Survived', y='Age', kind='scatter')
plt.show()
```



In [155]:

```
plt.pie([len(survived_female), len(survived_male)], labels=['Female', 'Male'], autop
plt.show()
```



In [169]:

```
df.plot(subplots=True, figsize=(16, 10))
plt.show()
```