In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:
```
df=pd.read_csv(r"C:\Users\user\Downloads\5_Instagram data.csv")
df.fillna(0,inplace=True)
df
```

Out[2]:

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares	Likes	Profile Visits	F
0	3920	2586	1028	619	56	98	9	5	162	35	
1	5394	2727	1838	1174	78	194	7	14	224	48	
2	4021	2085	1188	0	533	41	11	1	131	62	
3	4528	2700	621	932	73	172	10	7	213	23	
4	2518	1704	255	279	37	96	5	4	123	8	
...	...	...	...	...	...	...	...	...	...	...	
114	13700	5185	3041	5352	77	573	2	38	373	73	
115	5731	1923	1368	2266	65	135	4	1	148	20	
116	4139	1133	1538	1367	33	36	0	1	92	34	
117	32695	11815	3147	17414	170	1095	2	75	549	148	

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares	Likes	Profile Visits	F
118	36919	13473	4176	16444	2547	653	5	26	443	611	

119 rows × 13 columns

In [3]:
```
df.head()
```

Out[3]:

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares	Likes	Profile Visits	Foll
0	3920	2586	1028	619	56	98	9	5	162	35	
1	5394	2727	1838	1174	78	194	7	14	224	48	
2	4021	2085	1188	0	533	41	11	1	131	62	
3	4528	2700	621	932	73	172	10	7	213	23	
4	2518	1704	255	279	37	96	5	4	123	8	

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 119 entries, 0 to 118
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   Impressions     119 non-null    int64 
 1   From Home       119 non-null    int64 
 2   From Hashtags   119 non-null    int64 
 3   From Explore    119 non-null    int64 
 4   From Other      119 non-null    int64 
 5   Saves           119 non-null    int64 
 6   Comments        119 non-null    int64 
 7   Shares          119 non-null    int64 
 8   Likes           119 non-null    int64 
 9   Profile Visits  119 non-null    int64 
 10  Follows         119 non-null    int64 
 11  Caption         119 non-null    object
 12  Hashtags        119 non-null    object
dtypes: int64(11), object(2)
memory usage: 12.2+ KB
```

In [5]:
```
import seaborn as sns
```

In [6]:
```
df.describe()
```

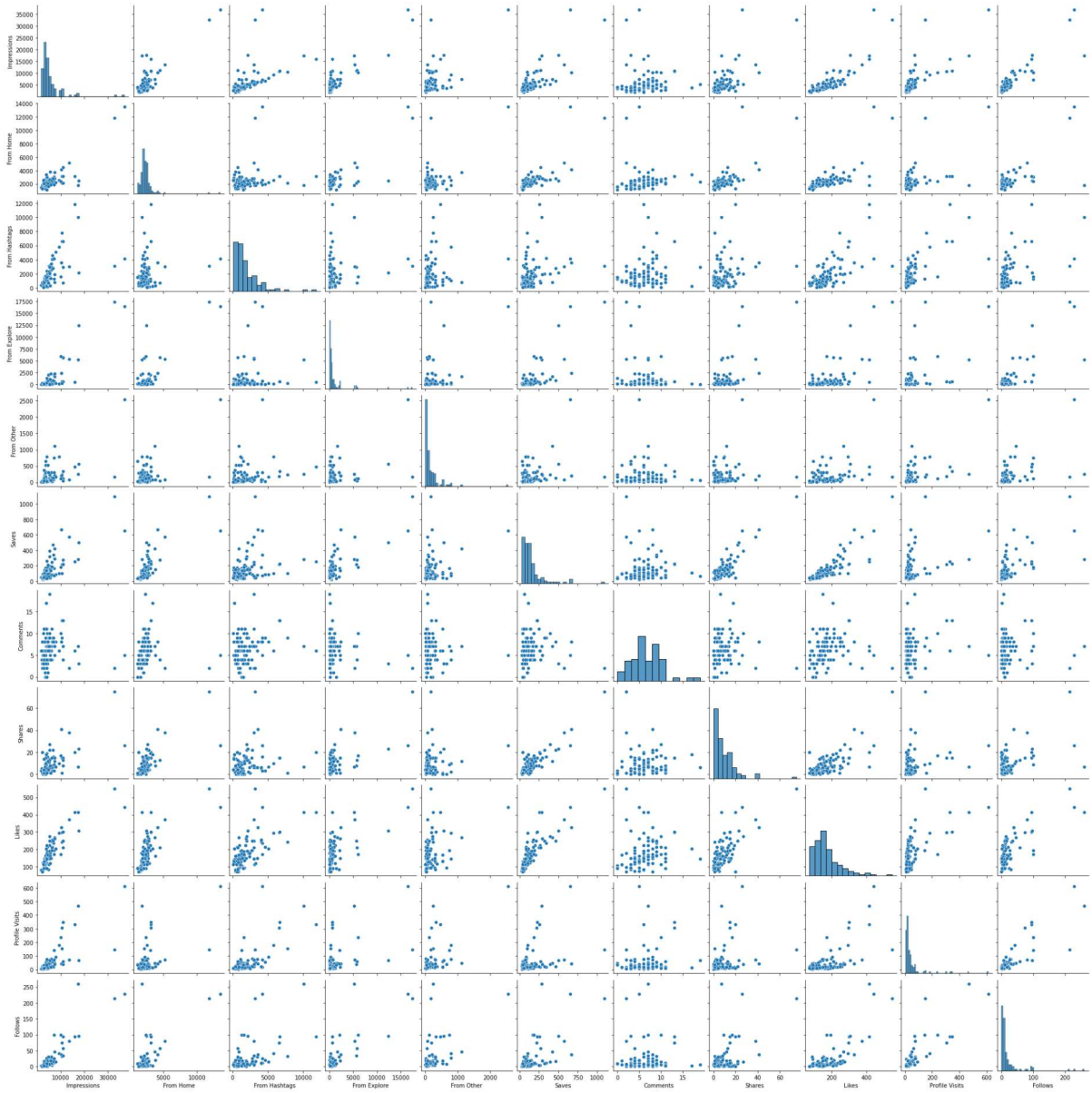
Out[6]:

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comm
count	119.000000	119.000000	119.000000	119.000000	119.000000	119.000000	119.00
mean	5703.991597	2475.789916	1887.512605	1078.100840	171.092437	153.310924	6.66
std	4843.780105	1489.386348	1884.361443	2613.026132	289.431031	156.317731	3.54
min	1941.000000	1133.000000	116.000000	0.000000	9.000000	22.000000	0.00
25%	3467.000000	1945.000000	726.000000	157.500000	38.000000	65.000000	4.00
50%	4289.000000	2207.000000	1278.000000	326.000000	74.000000	109.000000	6.00
75%	6138.000000	2602.500000	2363.500000	689.500000	196.000000	169.000000	8.00
max	36919.000000	13473.000000	11817.000000	17414.000000	2547.000000	1095.000000	19.00

In [7]:
```
df=pd.read_csv("5_Instagram data.csv")
```

In [8]:
```
sns.pairplot(df)
```

Out[8]:
```
<seaborn.axisgrid.PairGrid at 0x26f4590ea90>
```

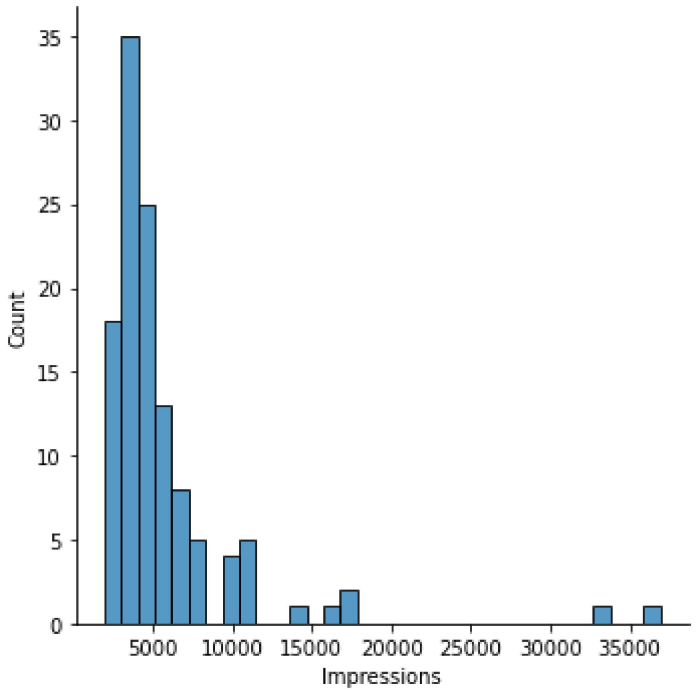


In [9]:
```
df1=df.drop(['Comments'],axis=1)
df1
df1=df1.drop(df1.index[1537:])
df1.isna().sum()
```

Out[9]:
```
Impressions       0
From Home         0
From Hashtags     0
From Explore      0
From Other        0
Saves             0
Shares            0
Likes             0
Profile Visits    0
Follows           0
Caption           0
Hashtags          0
dtype: int64
```

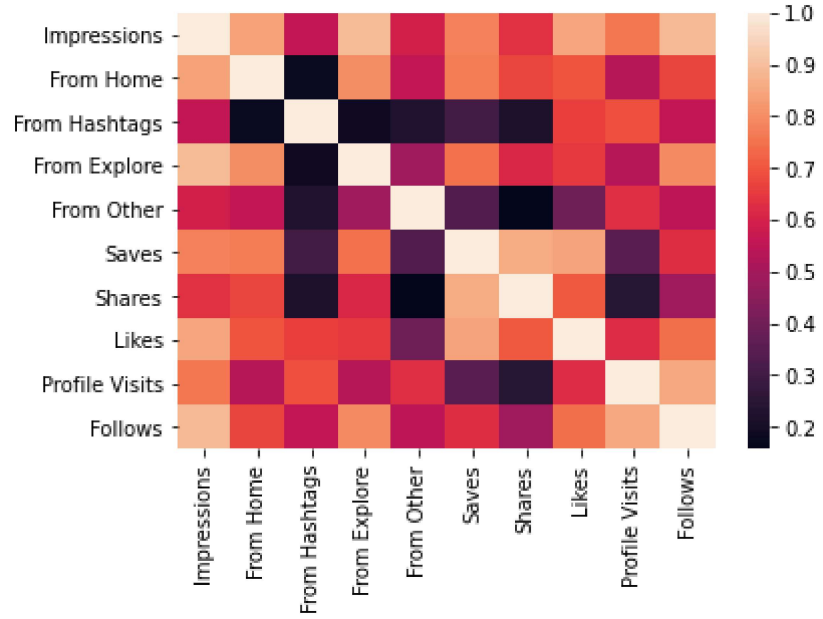
In [10]:
```
sns.displot(df['Impressions'])
```

Out[10]:
```
<seaborn.axisgrid.FacetGrid at 0x26f4a488910>
```



In [11]:
```
sns.heatmap(df1.corr())
```

Out[11]: <AxesSubplot:>



In [12]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [13]:
```
df1.isna().sum()
```

Out[13]:
```
Impressions       0
From Home         0
From Hashtags     0
From Explore      0
From Other        0
Saves             0
Shares            0
Likes             0
Profile Visits    0
Follows           0
Caption           0
Hashtags          0
dtype: int64
```

In [14]:
```
y=df1['Likes']
x=df1.drop(['Caption','Hashtags'],axis=1)
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
print(x_train)
```

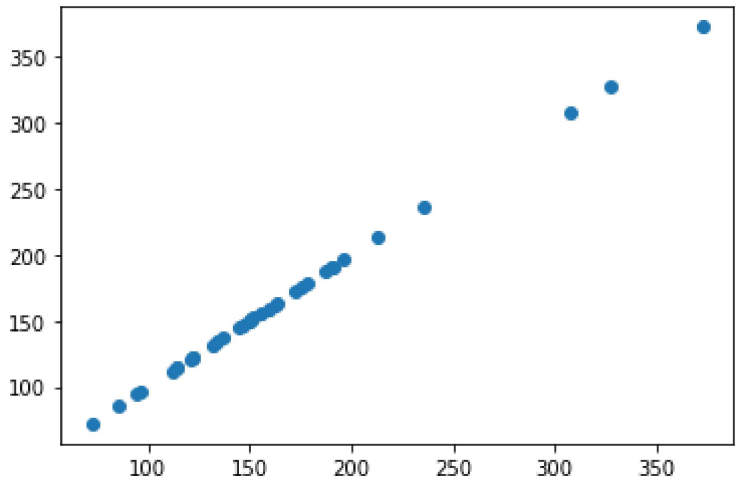
```
    Impressions  From Home  From Hashtags  From Explore  From Other  Saves  \
35         2523       1659            796            29          21     34
72         3606       2509            183           446         300    174
83         4002       3401            278           128          73    111
61         6339       2190           4036            48          27    171
24         4628       2406           1260           861          26    144
..          ...        ...            ...           ...         ...    ...
0          3920       2586           1028           619          56     98
52         2941       1716           1058            84          48     48
7          3541       2071            628           500          60    135
64         7571       3717            841          1716        1115    421
74         6559       2225           4041           158          72    179

    Shares  Likes  Profile Visits  Follows
35       0     86               4        2
72      15    138              17       12
83      18    205              16        2
61       5    248              21       10
24       3    160              10        4
..     ...    ...             ...      ...
0        5    162              35        2
52       1     99              12        4
7        9    124              12        6
64      12    269              50       46
74       6    257              22       12

[83 rows x 10 columns]
```

In [15]:
```
model=LinearRegression()
model.fit(x_train,y_train)
model.intercept_
```

Out[15]:
```
-7.105427357601002e-13
```

In [16]:
```
prediction=model.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[16]:
```
<matplotlib.collections.PathCollection at 0x26f4e0e5670>
```



In [17]:
```
model.score(x_test,y_test)
```

Out[17]:
```
1.0
```

In [18]:
```
from sklearn.linear_model import Ridge,Lasso
```

In [19]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[19]:
```
Ridge(alpha=10)
```

In [20]:
```
rr.score(x_test,y_test)
```

Out[20]:
```
0.9999999935804778
```

In [21]:
```
la =Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[21]:
```
Lasso(alpha=10)
```

In [22]:
```
la.score(x_test,y_test)
```

Out[22]:
```
0.9999845944911459
```

In [23]:
```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
print(en.coef_)
print(en.intercept_)
print(en.predict(x_test))
print(en.score(x_test,y_test))
from sklearn import metrics
print("Mean Absolute Error:",metrics.mean_absolute_error(y_test,prediction))
print("Mean Squared Error:",metrics.mean_squared_error(y_test,prediction))
print("Root Mean Squared Error:",np.sqrt(metrics.mean_squared_error(y_test,prec
```

```
[-2.73324381e-04  3.08785864e-04  3.28143840e-04  2.65893595e-04
  2.82219150e-04  7.65557477e-04  0.00000000e+00  9.97521227e-01
 -0.00000000e+00  0.00000000e+00]
0.1545091620847927
[328.14083722 121.0123474  236.3265904  121.99532989 195.97189891
 114.03007253 162.96346015 189.951583   186.94081757 151.99972609
 113.98532621  72.0612509  149.99184938 132.01725967 122.08085335
 171.97136112  94.05261907 133.95819648 178.02259982 112.04785196
 144.97808057  85.05833179 174.94968083 307.8467465  147.07154204
 190.87793596 372.96778075 149.99398787 137.06604274 155.99257904
 161.93446632  96.07518261 158.96005402 150.96164369 151.02266212
 212.82650758]
0.9999982366421484
Mean Absolute Error: 6.663311877572495e-13
Mean Squared Error: 7.599766755251328e-25
Root Mean Squared Error: 8.717664111016969e-13
```