In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as py
import seaborn as sns
```

In [2]:

```
d=pd.read_csv(r"C:\Users\user\Downloads\5_Instagram data - 5_Instagram data.csv")
d
```

Out[2]:

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares	Likes	Profile Visits	Follows
0	3920	2586	1028	619	56	98	9	5	162	35	2
1	5394	2727	1838	1174	78	194	7	14	224	48	10
2	4021	2085	1188	0	533	41	11	1	131	62	12
3	4528	2700	621	932	73	172	10	7	213	23	8
4	2518	1704	255	279	37	96	5	4	123	8	0
...	...	...	...	...	...	...	...	...	...	...	...
114	13700	5185	3041	5352	77	573	2	38	373	73	80
115	5731	1923	1368	2266	65	135	4	1	148	20	18

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares	Likes	Profile Visits	Follows
116	4139	1133	1538	1367	33	36	0	1	92	34	10
117	32695	11815	3147	17414	170	1095	2	75	549	148	214
118	36919	13473	4176	16444	2547	653	5	26	443	611	228

119 rows × 13 columns

In [3]:
```
d.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 [4]:
```
d.head()
```

Out[4]:

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares	Likes	Profile Visits	Follows
0	3920	2586	1028	619	56	98	9	5	162	35	2

i
da

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares	Likes	Profile Visits	Follows	
1	5394	2727	1838	1174	78	194	7	14	224	48	10	
2	4021	2085	1188	0	533	41	11	1	131	62	12	
3	4528	2700	621	932	73	172	10	7	213	23	8	
4	2518	1704	255	279	37	96	5	4	123	8	0	

In [5]:
```
d.describe()
```

Out[5]:

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	
count	119.000000	119.000000	119.000000	119.000000	119.000000	119.000000	119.000000	1
mean	5703.991597	2475.789916	1887.512605	1078.100840	171.092437	153.310924	6.663866	
std	4843.780105	1489.386348	1884.361443	2613.026132	289.431031	156.317731	3.544576	
min	1941.000000	1133.000000	116.000000	0.000000	9.000000	22.000000	0.000000	
25%	3467.000000	1945.000000	726.000000	157.500000	38.000000	65.000000	4.000000	
50%	4289.000000	2207.000000	1278.000000	326.000000	74.000000	109.000000	6.000000	
75%	6138.000000	2602.500000	2363.500000	689.500000	196.000000	169.000000	8.000000	
max	36919.000000	13473.000000	11817.000000	17414.000000	2547.000000	1095.000000	19.000000	

In [6]:
```
d.columns
```

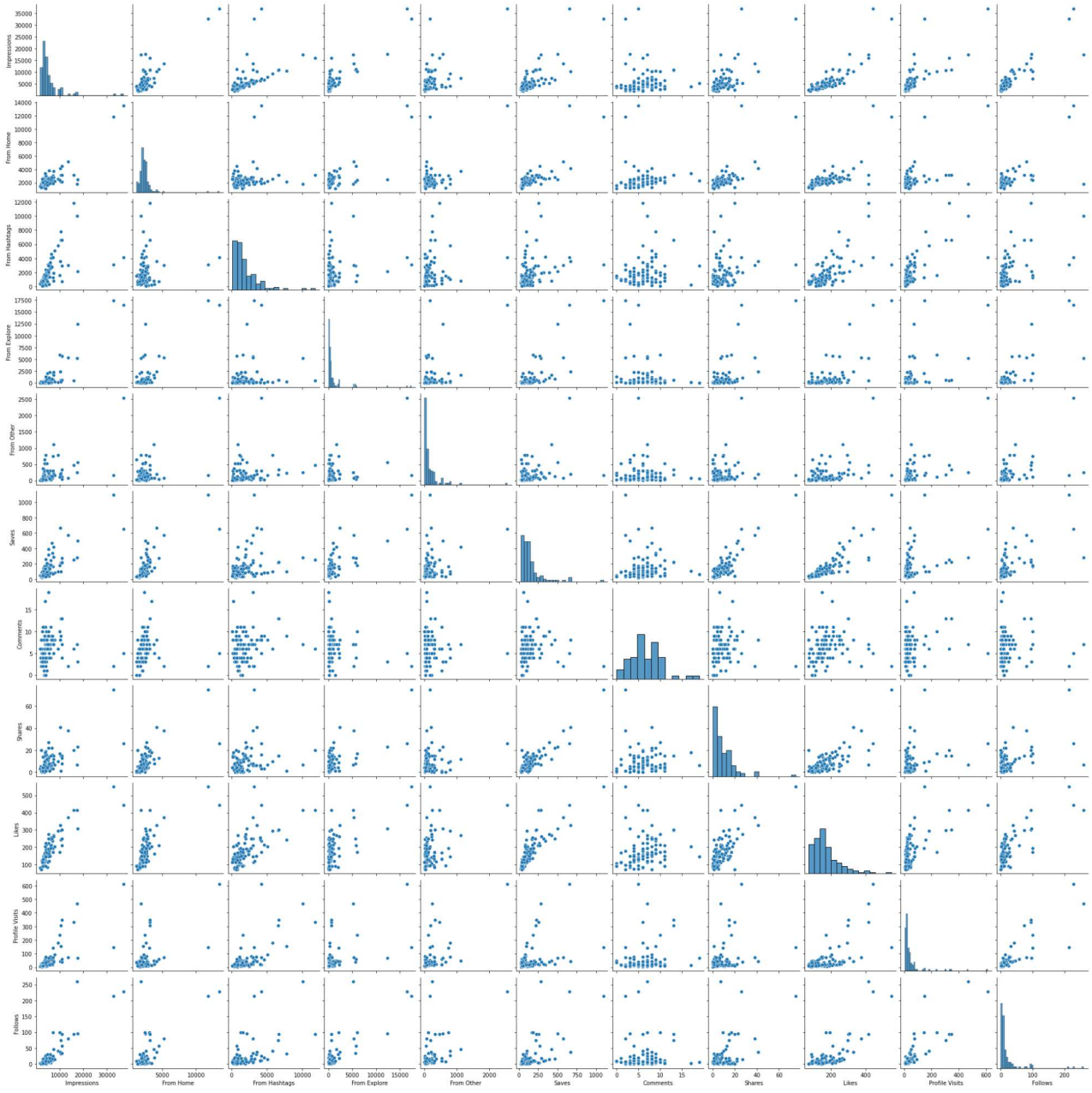
Out[6]:
```
Index(['Impressions', 'From Home', 'From Hashtags', 'From Explore',
       'From Other', 'Saves', 'Comments', 'Shares', 'Likes', 'Profile Visits',
       'Follows', 'Caption', 'Hashtags'],
      dtype='object')
```

In [7]:
```
d.index
```

Out[7]:
```
RangeIndex(start=0, stop=119, step=1)
```

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

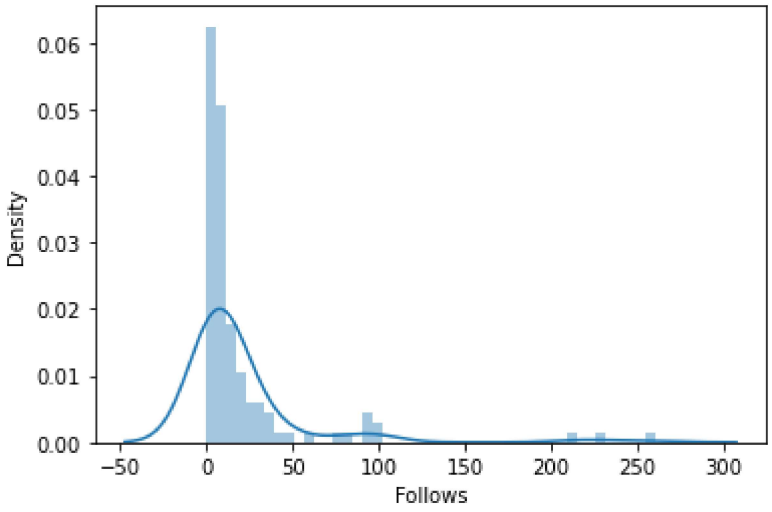
Out[8]:
```
<seaborn.axisgrid.PairGrid at 0x1cc36fc5f40>
```



In [9]:
```
sns.distplot(d['Follows'])
```

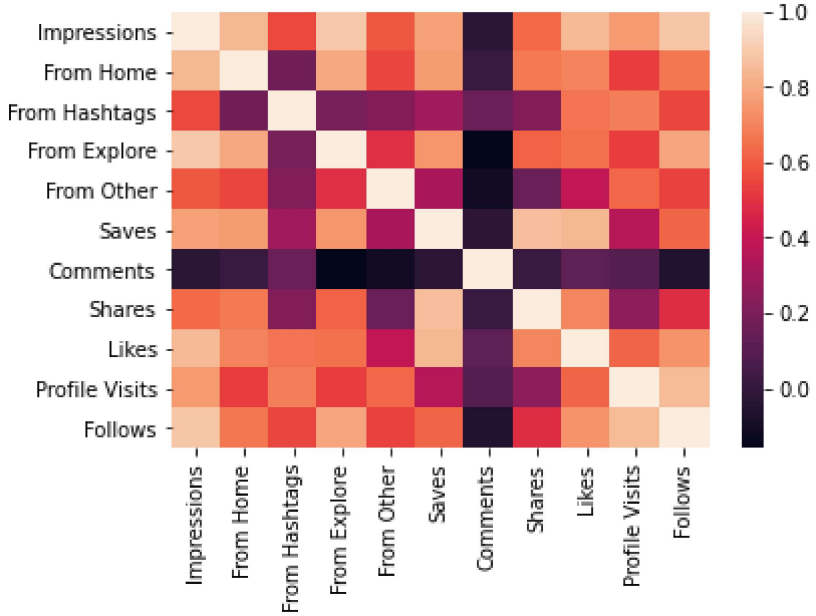
```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[9]:
```
<AxesSubplot:xlabel='Follows', ylabel='Density'>
```



In [10]:
```
d1=d[['Impressions', 'From Home', 'From Hashtags', 'From Explore',
       'From Other', 'Saves', 'Comments', 'Shares', 'Likes', 'Profile Visits',
       'Follows']]
sns.heatmap(d1.corr())
```

Out[10]: <AxesSubplot:>



In [11]:
```
x=d1[['Impressions', 'From Home', 'From Hashtags', 'From Explore',
       'From Other', 'Saves', 'Comments', 'Shares', 'Likes', 'Profile Visits']]
y=d1[ 'Follows']
```

In [12]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3)
```

In [13]:
```
from sklearn.linear_model import LinearRegression
```

In [14]:
```
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[14]:
```
LinearRegression()
```

In [15]:
```
print(lr.intercept_)
```

```
-3.0675269254359385
```

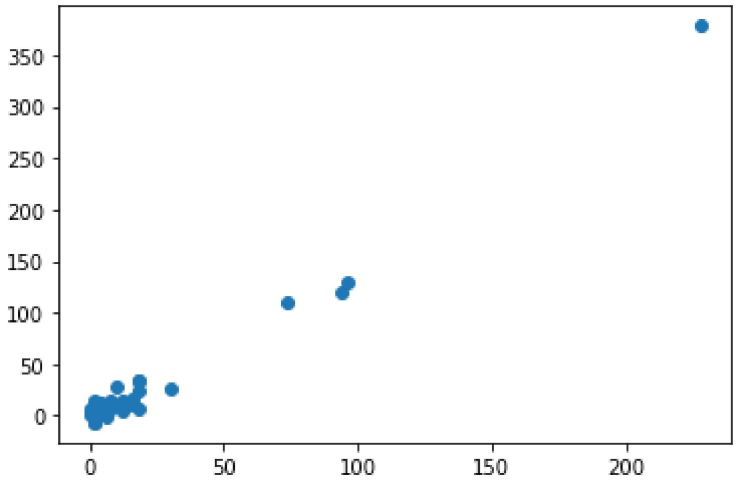
In [16]:
```
coeff =pd.DataFrame(lr.coef_,x.columns,columns=["Co-efficient"])
coeff
```

Out[16]:

	Co-efficient
Impressions	-0.006192
From Home	0.006354
From Hashtags	0.004436
From Explore	0.013517
From Other	0.008848
Saves	0.031303
Comments	-0.936898
Shares	-0.248400
Likes	0.031496
Profile Visits	0.389938

In [17]:
```
prediction =lr.predict(x_test)
py.scatter(y_test,prediction)
```

Out[17]:
```
<matplotlib.collections.PathCollection at 0x1cc3d86d730>
```



In [18]:
```
print(lr.score(x_test,y_test))
```

```
0.5654638437551043
```

In [19]:
```
print(lr.score(x_train,y_train))
```

```
0.9360306332103877
```

In [20]:
```
from sklearn.linear_model import Ridge,Lasso
```

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

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

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

Out[22]:
```
0.5655704247947426
```

In [23]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:5
30: ConvergenceWarning: Objective did not converge. You might want to increase the numbe
r of iterations. Duality gap: 2853.3920857249045, tolerance: 13.446650602409637
  model = cd_fast.enet_coordinate_descent(
```

Out[23]:
```
Lasso(alpha=10)
```

In [27]:
```
la.score(x_test,y_test)
```

Out[27]:
```
0.586021288667657
```

In [29]:
```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:5
30: ConvergenceWarning: Objective did not converge. You might want to increase the numbe
r of iterations. Duality gap: 4300.120255243906, tolerance: 13.446650602409637
  model = cd_fast.enet_coordinate_descent(
```

Out[29]:
```
ElasticNet()
```

In [31]:
```
print(en.coef_)
```

```
[-1.84797225e-03  1.81788884e-03  8.89140756e-05  9.22451420e-03
  4.55372730e-03  3.14591029e-02 -8.49310186e-01 -2.26031582e-01
  2.99832549e-02  3.88581563e-01]
```

In [32]:
```
print(en.intercept_)
```

```
-3.5478762231419694
```

In [33]:
```
print(en.score(x_test,y_test))
```

```
0.5731402511911358
```

In [34]:
```
from sklearn import metrics
```

In [38]:
```
print("Mean Absolute Error:",metrics.mean_absolute_error(y_test,prediction))
```

```
Mean Absolute Error: 11.385196447049093
```

In [39]:
```
print("Mean Squared Error:",metrics.mean_squared_error(y_test,prediction))
```

```
Mean Squared Error: 761.9135505022385
```

In [37]:
```
print("Root Mean Squared Error:",np.sqrt(metrics.mean_squared_error(y_test,prediction))
```

```
Root Mean Squared Error: 27.60278157183146
```

In [ ]:
```

```