

In [1]:

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

In [2]:

```
df=pd.read_csv(r'C:\Users\user\Downloads\5_Instagram data.csv')
df
```

Out[2]:

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares	Likes	Profile Visits	Follows	Caption
0	3920	2586	1028	619	56	98	9	5	162	35	2	Here are some of the most important data visualizations.
1	5394	2727	1838	1174	78	194	7	14	224	48	10	Here are some of the best data science projects.
2	4021	2085	1188	0	533	41	11	1	131	62	12	Learn how to train a machine learning model and.
3	4528	2700	621	932	73	172	10	7	213	23	8	Here are some of the best Python programs to learn.
4	2518	1704	255	279	37	96	5	4	123	8	0	Plotting data with annotations while visualizing your data.
...	...	...	...	...	...	...	...	...	...	...	...	...
114	13700	5185	3041	5352	77	573	2	38	373	73	80	Here are some of the best data science certifications.
115	5731	1923	1368	2266	65	135	4	1	148	20	18	Clustering is a machine learning technique used for data analysis.
116	4139	1133	1538	1367	33	36	0	1	92	34	10	Clustering music genres is a task called task clustering.
117	32695	11815	3147	17414	170	1095	2	75	549	148	214	Here are some of the best data science certifications.
118	36919	13473	4176	16444	2547	653	5	26	443	611	228	175 Python Projects with Source Code solved and explained.

119 rows × 13 columns



In [3]:

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

Out[3]:

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares	Likes	Profile Visits	Follows	Caption
0	3920	2586	1028	619	56	98	9	5	162	35	2	Here are some of the most important data visua...
1	5394	2727	1838	1174	78	194	7	14	224	48	10	Here are some of the best data science project...
2	4021	2085	1188	0	533	41	11	1	131	62	12	Learn how to train a machine learning model an...
3	4528	2700	621	932	73	172	10	7	213	23	8	Here's how you can write a Python program to d...
4	2518	1704	255	279	37	96	5	4	123	8	0	Plotting annotations while visualizing your da...
5	3884	2046	1214	329	43	74	7	10	144	9	2	Here are some of the most important soft skill...
6	2621	1543	599	333	25	22	5	1	76	26	0	Learn how to analyze a candlestick chart as a ...
7	3541	2071	628	500	60	135	4	9	124	12	6	Here are some of the best books that you can f...
8	3749	2384	857	248	49	155	6	8	159	36	4	Here are some of the best data analysis projec...
9	4115	2609	1104	178	46	122	6	3	191	31	6	Here are two best ways to count the number of ...



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]:

```
df.describe()
```

Out[5]:

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares
count	119.000000	119.000000	119.000000	119.000000	119.000000	119.000000	119.000000	119.000000
mean	5703.991597	2475.789916	1887.512605	1078.100840	171.092437	153.310924	6.663866	9.361345
std	4843.780105	1489.386348	1884.361443	2613.026132	289.431031	156.317731	3.544576	10.089205
min	1941.000000	1133.000000	116.000000	0.000000	9.000000	22.000000	0.000000	0.000000
25%	3467.000000	1945.000000	726.000000	157.500000	38.000000	65.000000	4.000000	3.000000
50%	4289.000000	2207.000000	1278.000000	326.000000	74.000000	109.000000	6.000000	6.000000
75%	6138.000000	2602.500000	2363.500000	689.500000	196.000000	169.000000	8.000000	13.500000
max	36919.000000	13473.000000	11817.000000	17414.000000	2547.000000	1095.000000	19.000000	75.000000

In [6]:

```
df.columns
```

Out[6]:

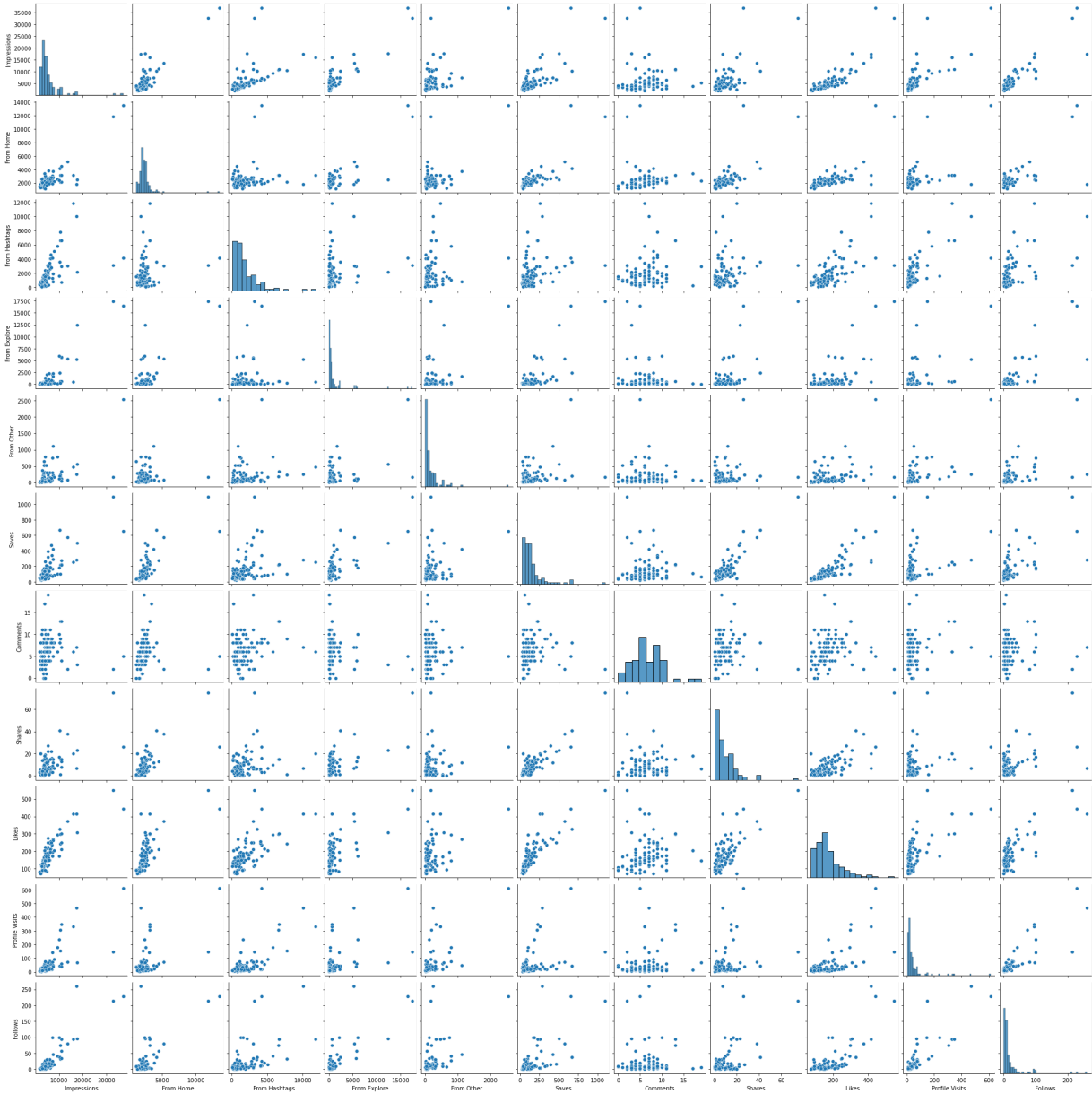
```
Index(['Impressions', 'From Home', 'From Hashtags', 'From Explore',
      'From Other', 'Saves', 'Comments', 'Shares', 'Likes', 'Profile Visits',
      'Follows', 'Caption', 'Hashtags'],
      dtype='object')
```

In [7]:

```
sns.pairplot(df)
```

Out[7]:

<seaborn.axisgrid.PairGrid at 0x18a65e33f70>



In [8]:

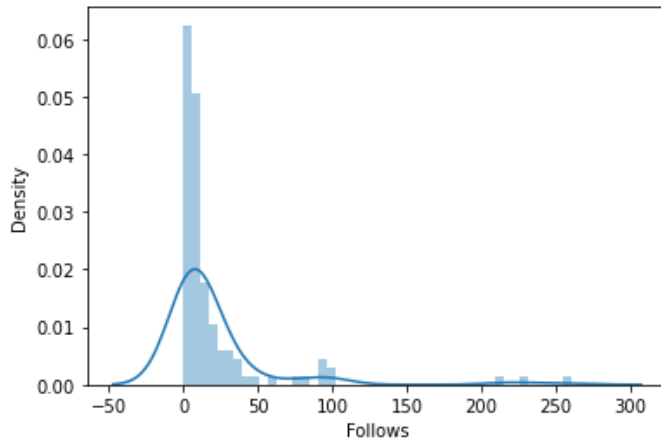
```
sns.distplot(df['Follows'])
```

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 adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

warnings.warn(msg, FutureWarning)

Out[8]:

```
<AxesSubplot:xlabel='Follows', ylabel='Density'>
```

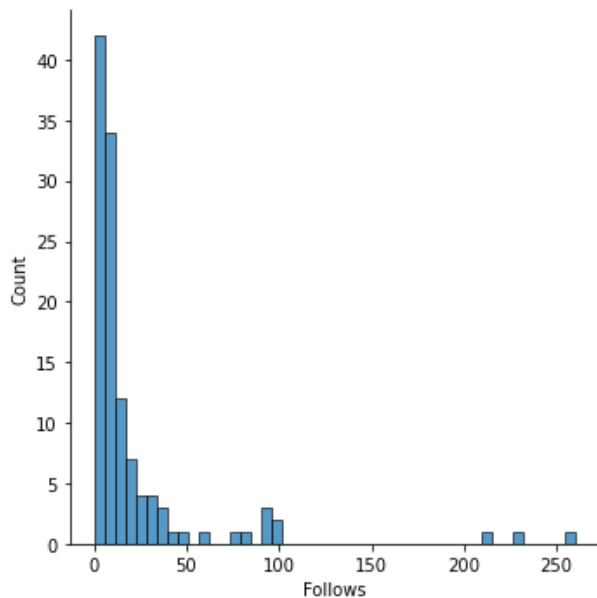


In [9]:

```
sns.displot(df["Follows"])
```

Out[9]:

```
<seaborn.axisgrid.FacetGrid at 0x18a6abf2fa0>
```



In [10]:

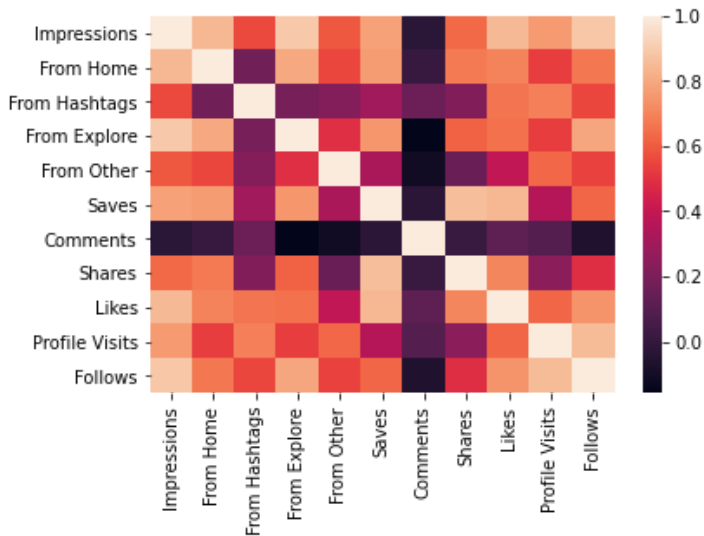
```
df1=df[['Impressions', 'From Home', 'From Hashtags', 'From Explore',  
        'From Other', 'Saves', 'Comments', 'Shares', 'Likes', 'Profile Visits',  
        'Follows', 'Caption', 'Hashtags']]
```

In [11]:

```
sns.heatmap(df1.corr())
```

Out[11]:

&lt;AxesSubplot:&gt;



In [12]:

```
x=df1[['Impressions', 'From Home', 'From Hashtags', 'From Explore',
        'From Other', 'Saves', 'Comments', 'Shares', 'Likes', 'Profile Visits']]
y=df1[['Follows']]
```

In [13]:

```
from sklearn.model_selection import train_test_split
```

In [14]:

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3)
```

In [15]:

```
from sklearn.linear_model import LinearRegression
```

```
lr=LinearRegression()
lr.fit(x_train,y_train)#ValueError: Input contains NaN, infinity or a value too large for dtype('float64')
```

Out[15]:

```
LinearRegression()
```

In [16]:

```
print(lr.intercept_)
```

```
[-8.87917348]
```

In [17]:

```
coef= pd.DataFrame(lr.coef_)
coef
```

Out[17]:

	0	1	2	3	4	5	6	7	8	9
0	-0.000298	0.00864	0.001038	0.006009	0.017943	0.022218	-0.610022	0.03542	-0.088952	0.289857

In [18]:

```
print(lr.score(x_test,y_test))
```

0.6276565204080369

In [19]:

```
prediction = lr.predict(x_test)  
plt.scatter(y_test,prediction)
```

Out[19]:

<matplotlib.collections.PathCollection at 0x18a6c7bb940>

