PROBLEM STATEMENT:

A real state agents wants help to predict the house price for region in the USA, He gave you the dataset to work on and you decided to use the LinearRegression Model.Create a model that will help him to Estimate of what the house Would Sell for.....

DATA COLLECTION:-

The data set contains 7 columns and 5000 rows with .CSV Extension.

The data contains the following columns: 'Avg.AreaIncome'-Avg.The Income of the house holder of the city house is located,'Avg.Area House Age'-Avg.Age of Houses in the same city.'Avg-Area Number of Rooms'-Avg.Number of Rooms for Houses in the same city:'Avg.Area Number of Bedrooms'-Avg Number of Bedrooms for Houses in the Same city:'Area population'-population of the city:'price'-price that the house sold at;'Address'-Address of the houses;

In [3]:

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

In [4]:

```
df=pd.read_csv(r"C:\Users\monim\Downloads\USA_Housing.csv")
df
```

Out[4]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	
0	79545.458574	5.682861	7.009188	4.09	23086.800503	1.059034e+06	208 Michae 674\nLau
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06	188 Johr Suite Kath
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06	912 Stravenue\nD V
3	63345.240046	7.188236	5.586729	3.26	34310.242831	1.260617e+06	USS Barnet
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05	USNS Raym
...	...	...	...	...	...	...	
4995	60567.944140	7.830362	6.137356	3.46	22837.361035	1.060194e+06	USNS Willi AP 3
4996	78491.275435	6.999135	6.576763	4.02	25616.115489	1.482618e+06	PSC 8489\nAPO
4997	63390.686886	7.250591	4.805081	2.13	33266.145490	1.030730e+06	4215 Tra Suite 076\nJ
4998	68001.331235	5.534388	7.130144	5.44	42625.620156	1.198657e+06	USS Wallace
4999	65510.581804	5.992305	6.792336	4.07	46501.283803	1.298950e+06	37778 Geo Apt. 509\n

5000 rows × 7 columns

In [5]:

```
df.head()
```

Out[5]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	A
0	79545.458574	5.682861	7.009188	4.09	23086.800503	1.059034e+06	208 Michael Fe 674\nLaurab
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06	188 Johnsor Suite 079 Kathleer
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06	9127 Eli Stravenue\nDani WI 0
3	63345.240046	7.188236	5.586729	3.26	34310.242831	1.260617e+06	USS Barnett\nF
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05	USNS Raymonc AE

In [6]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 7 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   Avg. Area Income             5000 non-null   float64
 1   Avg. Area House Age          5000 non-null   float64
 2   Avg. Area Number of Rooms    5000 non-null   float64
 3   Avg. Area Number of Bedrooms 5000 non-null   float64
 4   Area Population              5000 non-null   float64
 5   Price                        5000 non-null   float64
 6   Address                      5000 non-null   object 
dtypes: float64(6), object(1)
memory usage: 273.6+ KB
```

In [7]:

```
df.describe()
```

Out[7]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5.000000e+03
mean	68583.108984	5.977222	6.987792	3.981330	36163.516039	1.232073e+06
std	10657.991214	0.991456	1.005833	1.234137	9925.650114	3.531176e+05
min	17796.631190	2.644304	3.236194	2.000000	172.610686	1.593866e+04
25%	61480.562388	5.322283	6.299250	3.140000	29403.928702	9.975771e+05
50%	68804.286404	5.970429	7.002902	4.050000	36199.406689	1.232669e+06
75%	75783.338666	6.650808	7.665871	4.490000	42861.290769	1.471210e+06
max	107701.748378	9.519088	10.759588	6.500000	69621.713378	2.469066e+06

In [9]:

```
df.columns
```

Out[9]:

```
Index(['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Roo
ms',
       'Avg. Area Number of Bedrooms', 'Area Population', 'Price', 'Addres
s'],
      dtype='object')
```

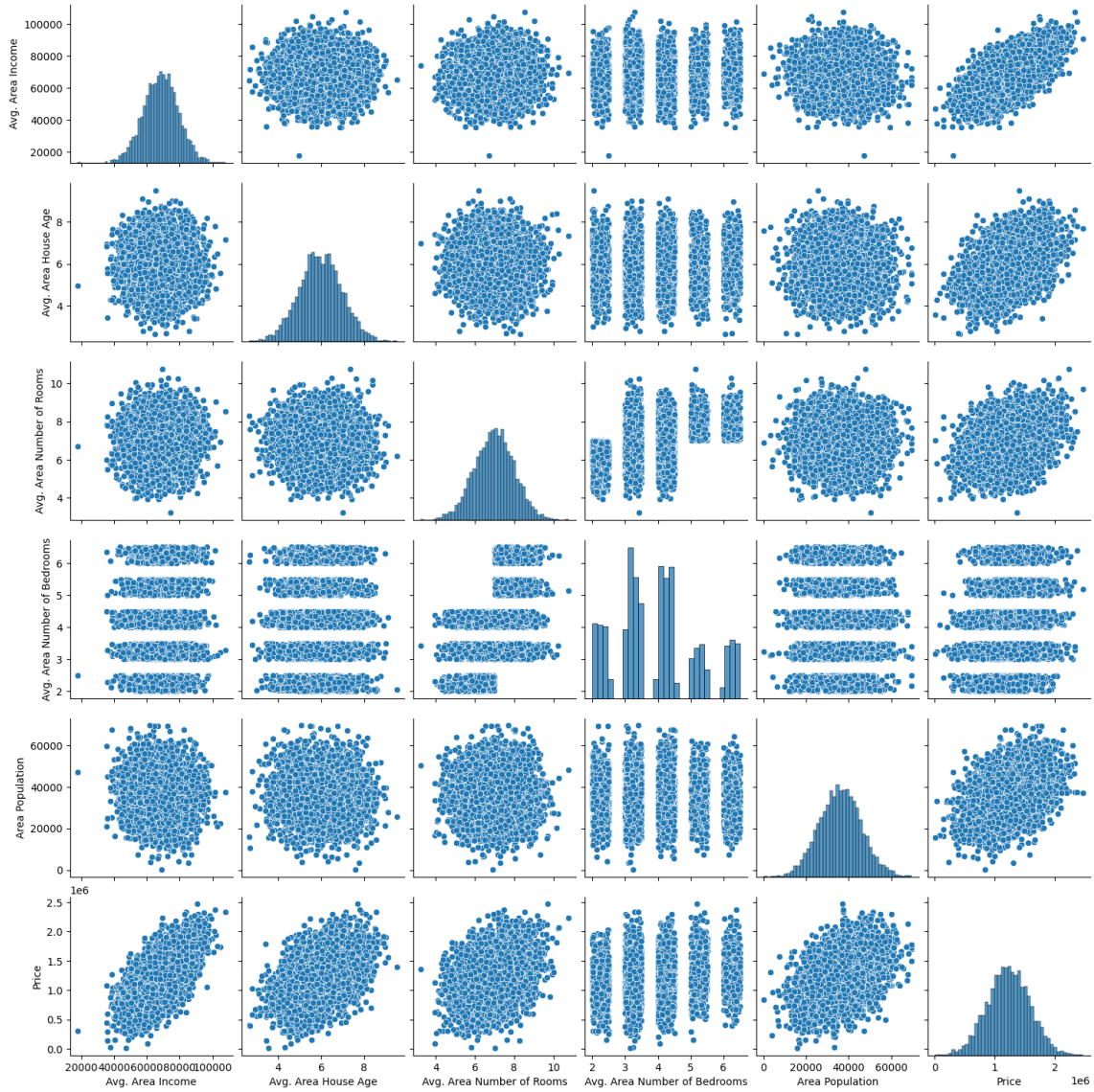
EXPLORATORY DATA ANALYSIS

In [10]:

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

Out[10]:

```
<seaborn.axisgrid.PairGrid at 0x2264e6d1e10>
```

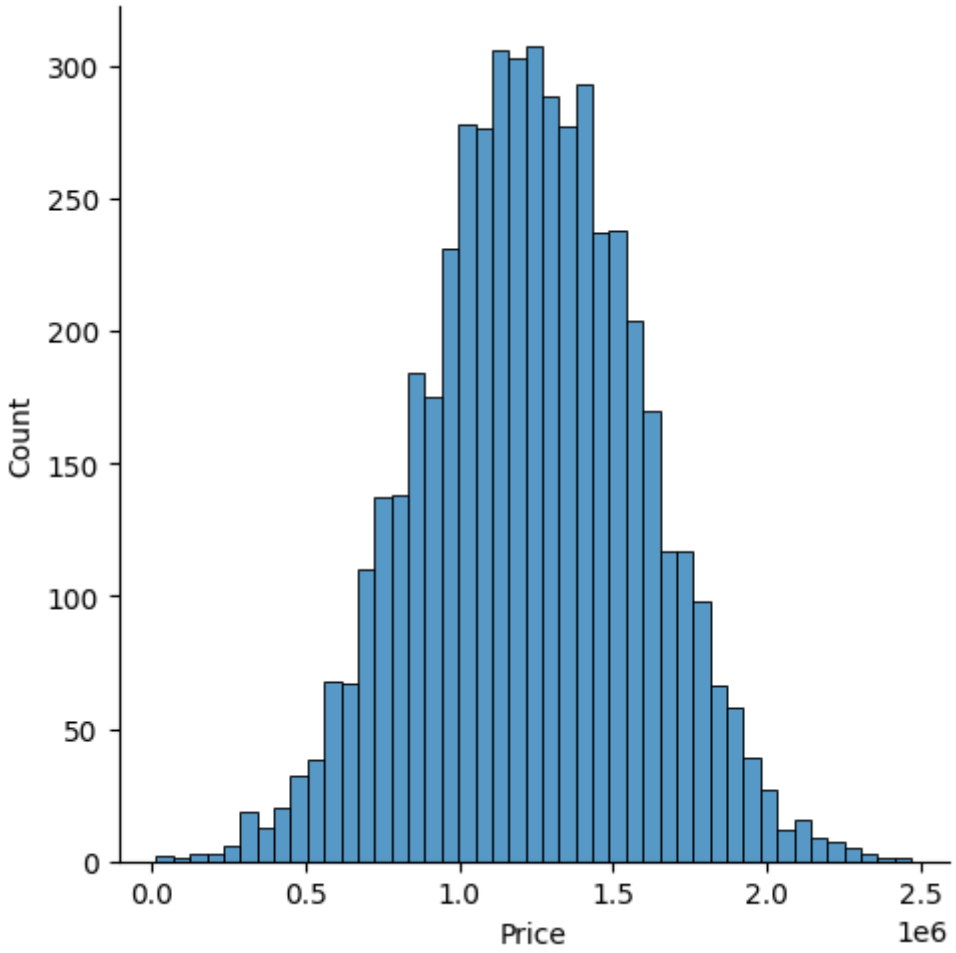


In [12]:

```
sns.displot(df['Price'])
```

Out[12]:

```
<seaborn.axisgrid.FacetGrid at 0x22655c95ab0>
```

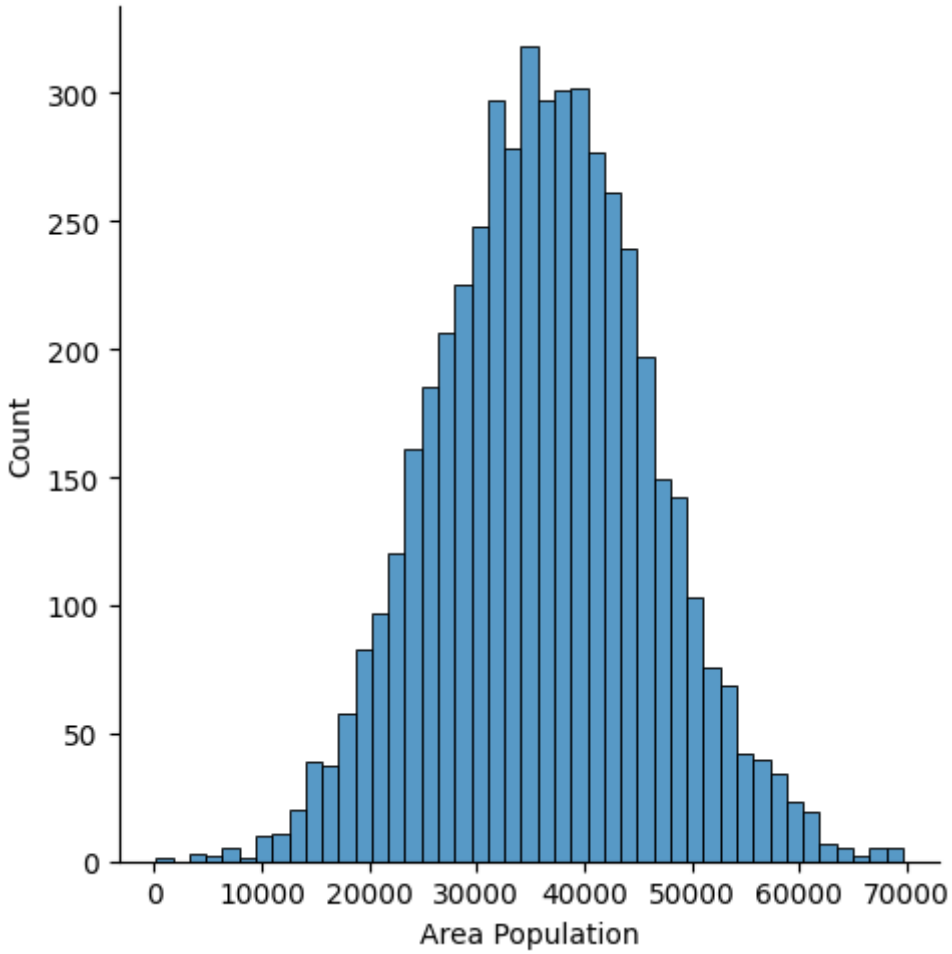


In [13]:

```
sns.displot(df['Area Population'])
```

Out[13]:

```
<seaborn.axisgrid.FacetGrid at 0x22655c76f50>
```



In [16]:

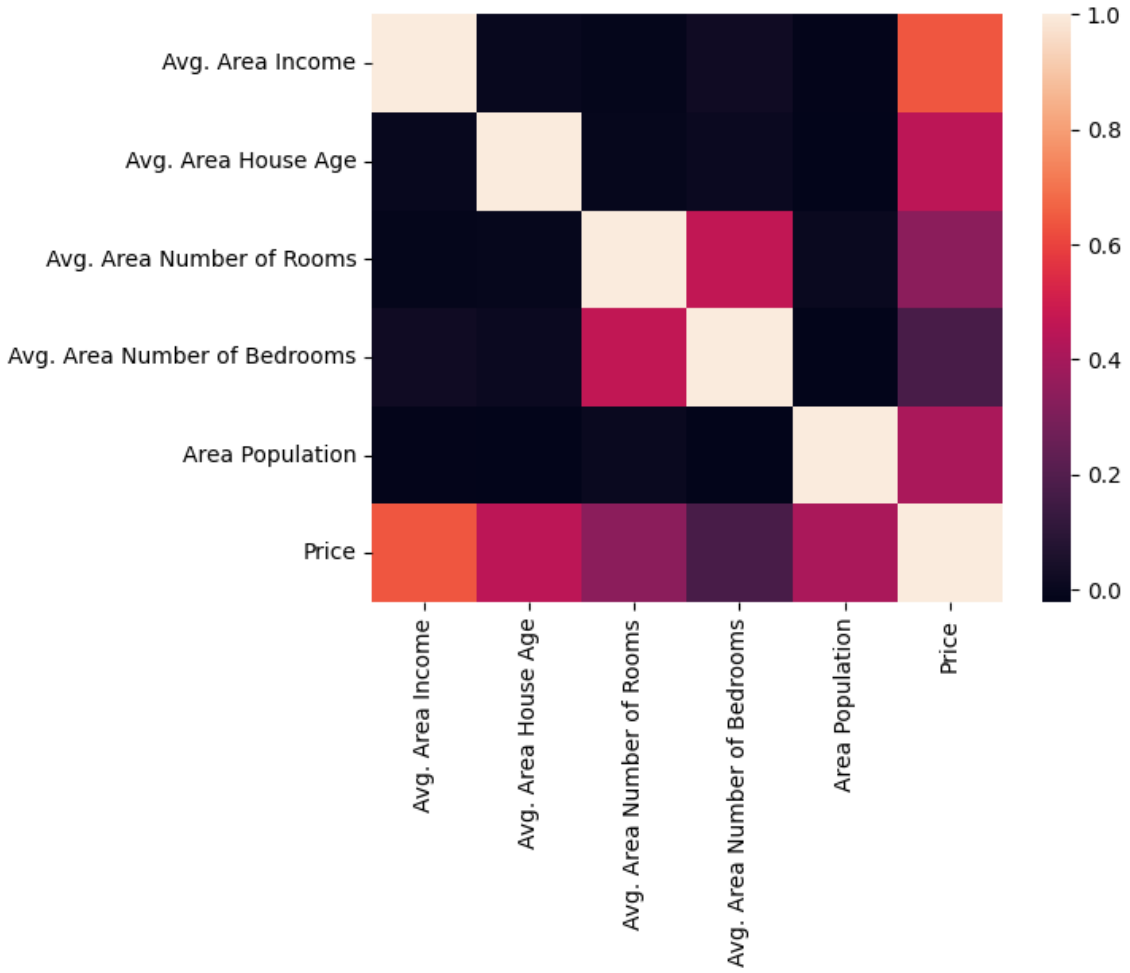
```
Housedf=df[['Avg. Area Income', 'Avg. Area House Age',  'Avg. Area Number of Rooms',
            'Area Population',  'Price']]
```

In [17]:

```
sns.heatmap(Housedf.corr())
```

Out[17]:

```
<Axes: >
```



TO TRAIN THE MODEL

we are going to train Linear Regression Model.we need to first split up our data into a x list that contains the features to train on,and a y list with the target variables, in this case, the price column we will ignore the address column because it only has text which is not useful for linear regression modelling

In [26]:

```
x=Housedf=df[['Avg. Area Income',   'Avg. Area House Age',  'Avg. Area Number of Rooms',
            'Area Population',  'Price']]
y=df['Price']
```

In [27]:

```
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,random_state=101)
```

In [28]:

```
from sklearn.linear_model import LinearRegression
lm=LinearRegression()
lm.fit(x_train,y_train)
```

Out[28]:

```
LinearRegression
LinearRegression()
```

In [30]:

```
print(lm.intercept_)
```

```
-9.313225746154785e-10
```

In [34]:

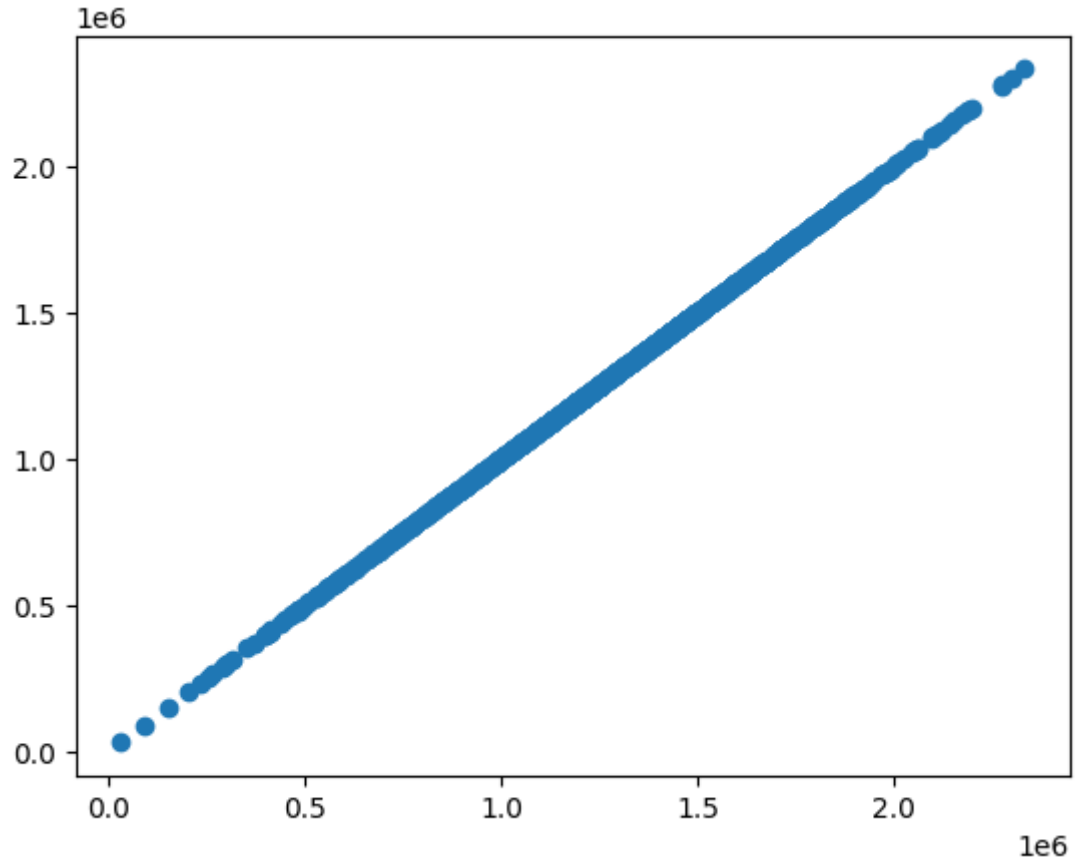
```
coeff_df=pd.DataFrame(lm.coef_,x.columns,columns=['coefficient'])
```

In [35]:

```
predictions=lm.predict(x_test)
plt.scatter(y_test,predictions)
```

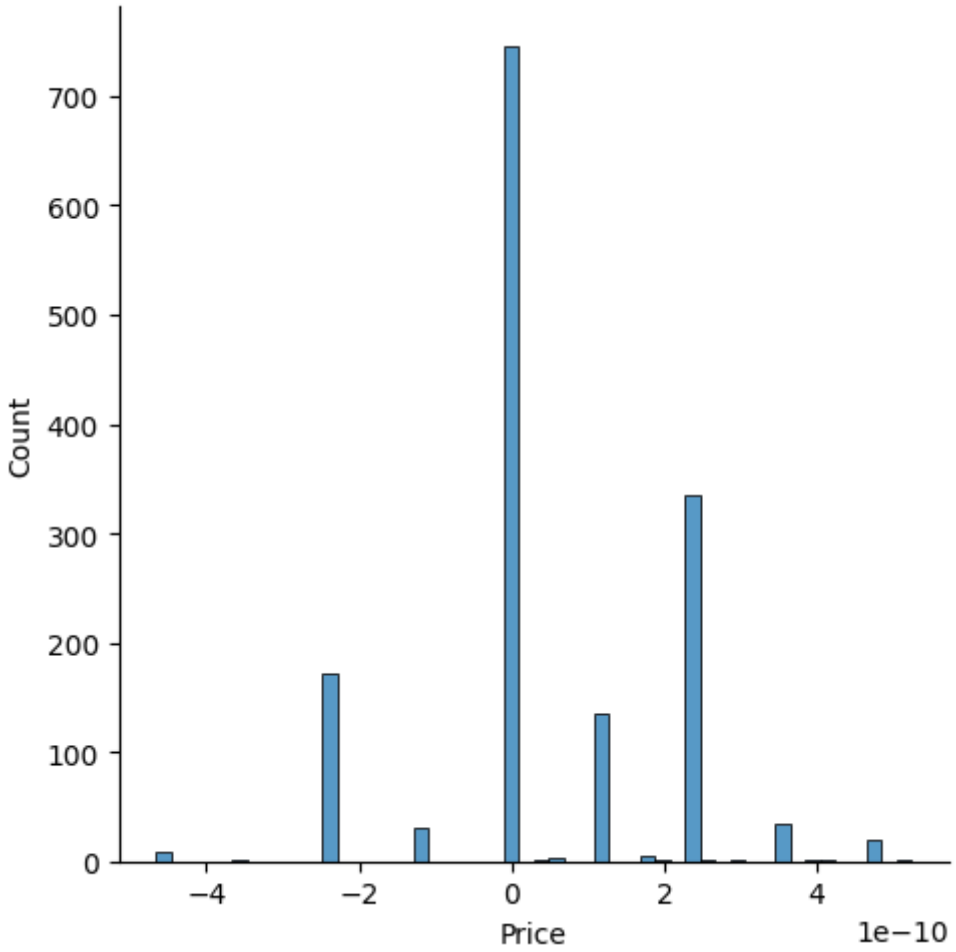
Out[35]:

```
<matplotlib.collections.PathCollection at 0x226586b3100>
```



In [36]:

```
sns.displot((y_test-predictions),bins=50);
```



In [37]:

```
from sklearn import metrics
print('MAE:',metrics.mean_absolute_error(y_test,predictions))
print('MSE:',metrics.mean_squared_error(y_test,predictions))
print('RMSE:',np.sqrt(metrics.mean_squared_error(y_test,predictions)))
```

```
MAE: 1.1111842468380928e-10
MSE: 2.7542970345868087e-20
RMSE: 1.6596074941343234e-10
```

In [ ]:

```

```