Problem statement

A real estate agent want to help to predict the house price for regions in the USA. He gave the data set to work on and I decided to use the Linear Regression Model.

Data Collection

the dataset contains 7 columns and 5000 rows in the CSV extension. The data contains the following coloumns: 'Avg.Area Income'-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;'Price'-Price that the house sold at;'Address'-Address of the houses.

In [2]:

```
# importing th libraries

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

In [4]:

```python
# reading the file
df=pd.read_csv(r"C:\Users\yoshitha lakshmi\OneDrive\Desktop\python\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	79545.45857	5.682861	7.009188	4.09	23086.80050	1.059034e+06	208 Michael 674\nLaur
1	79248.64245	6.002900	6.730821	3.09	40173.07217	1.505891e+06	188 Johns Suite (Kathl
2	61287.06718	5.865890	8.512727	5.13	36882.15940	1.058988e+06	9127 Stravenue\nD W
3	63345.24005	7.188236	5.586729	3.26	34310.24283	1.260617e+06	USS Barnett
4	59982.19723	5.040555	7.839388	4.23	26354.10947	6.309435e+05	USNS Raym
...	...	...	...	...	...	...	...
4995	60567.94414	7.830362	6.137356	3.46	22837.36103	1.060194e+06	USNS Willia AP 30
4996	78491.27543	6.999135	6.576763	4.02	25616.11549	1.482618e+06	PSC 8489\nAPO
4997	63390.68689	7.250591	4.805081	2.13	33266.14549	1.030730e+06	4215 Trac Suite 076\nJo
4998	68001.33124	5.534388	7.130144	5.44	42625.62016	1.198657e+06	USS Wallace
4999	65510.58180	5.992305	6.792336	4.07	46501.28380	1.298950e+06	37778 Geor Apt. 509\nE

5000 rows × 7 columns

In [3]:

```
df.head()
```

Out[3]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	Ad
0	79545.45857	5.682861	7.009188	4.09	23086.80050	1.059034e+06	208 Michael Fer 674\nLaurabu 3
1	79248.64245	6.002900	6.730821	3.09	40173.07217	1.505891e+06	188 Johnson Suite 079\ Kathleen
2	61287.06718	5.865890	8.512727	5.13	36882.15940	1.058988e+06	9127 Eliz Stravenue\nDanie WI 06
3	63345.24005	7.188236	5.586729	3.26	34310.24283	1.260617e+06	USS Barnett\nFF
4	59982.19723	5.040555	7.839388	4.23	26354.10947	6.309435e+05	USNS Raymond\ 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%	61480.562390	5.322283	6.299250	3.140000	29403.928700	9.975771e+05
50%	68804.286405	5.970429	7.002902	4.050000	36199.406690	1.232669e+06
75%	75783.338665	6.650808	7.665871	4.490000	42861.290770	1.471210e+06
max	107701.748400	9.519088	10.759588	6.500000	69621.713380	2.469066e+06

In [8]:

```
df.columns
```

Out[8]:

```
Index(['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Ro
oms',
       'Avg. Area Number of Bedrooms', 'Area Population', 'Price', 'Addre
ss'],
      dtype='object')
```

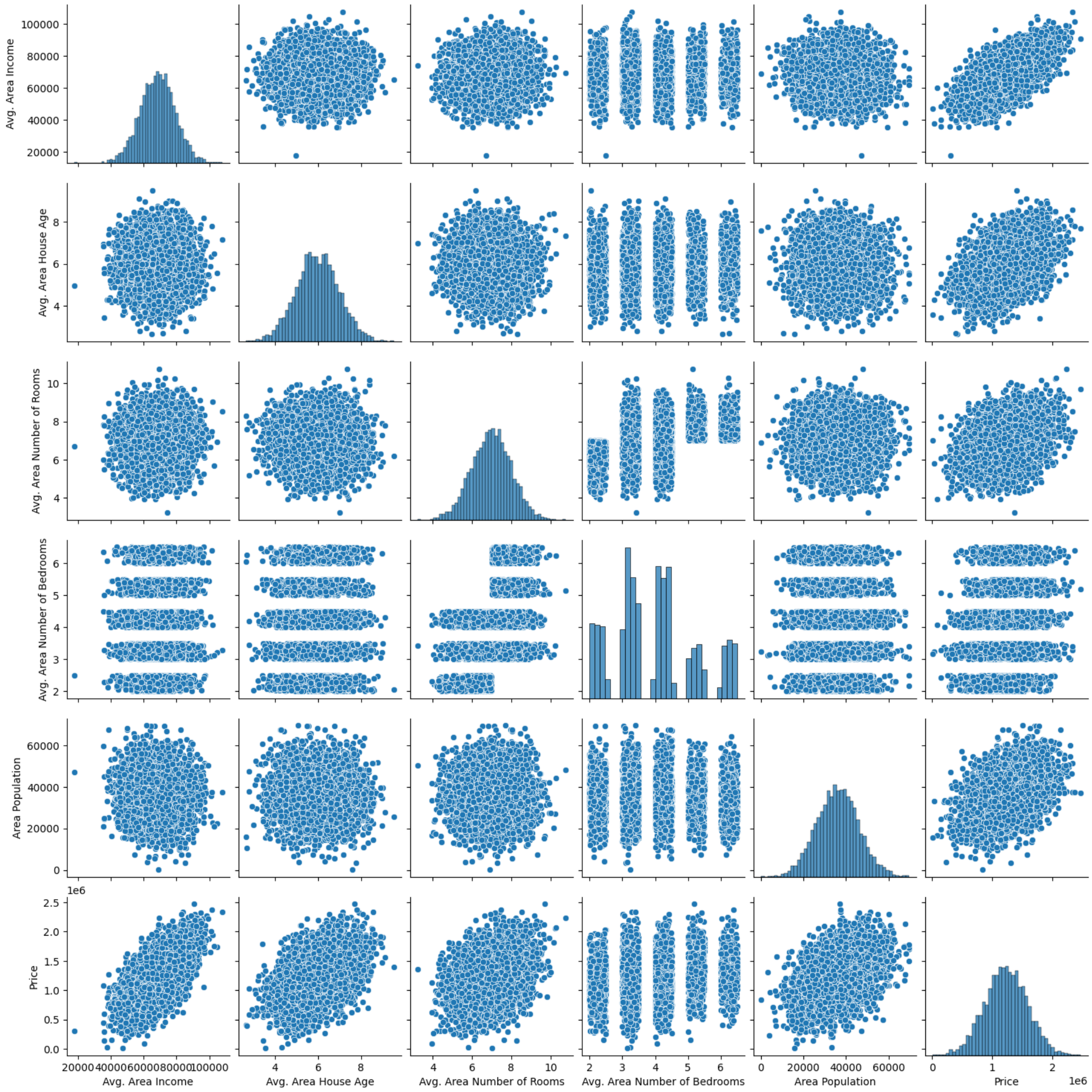
Exploratory Data Analysis

In [5]:

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

Out[5]:

```
<seaborn.axisgrid.PairGrid at 0x204c4b497b0>
```

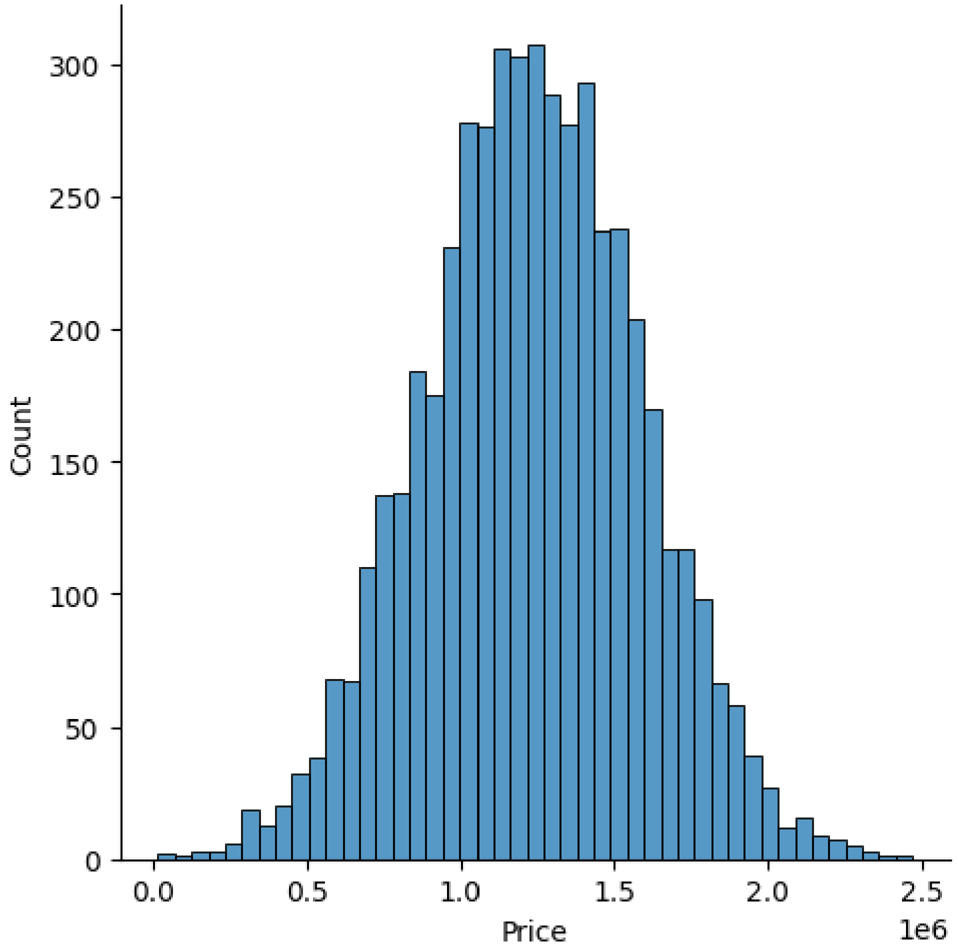


In [8]:

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

Out[8]:

```
<seaborn.axisgrid.FacetGrid at 0x204c900ac80>
```

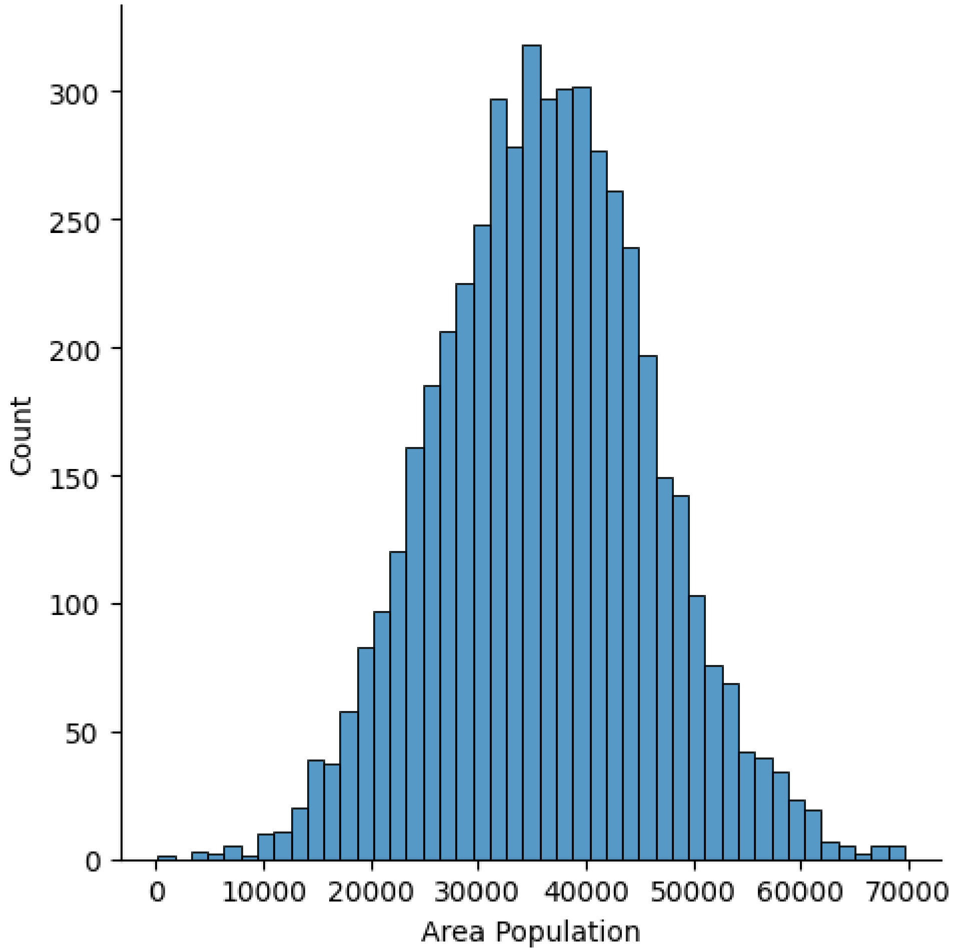


In [9]:

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

Out[9]:

```
<seaborn.axisgrid.FacetGrid at 0x204a4d98580>
```



In [11]:

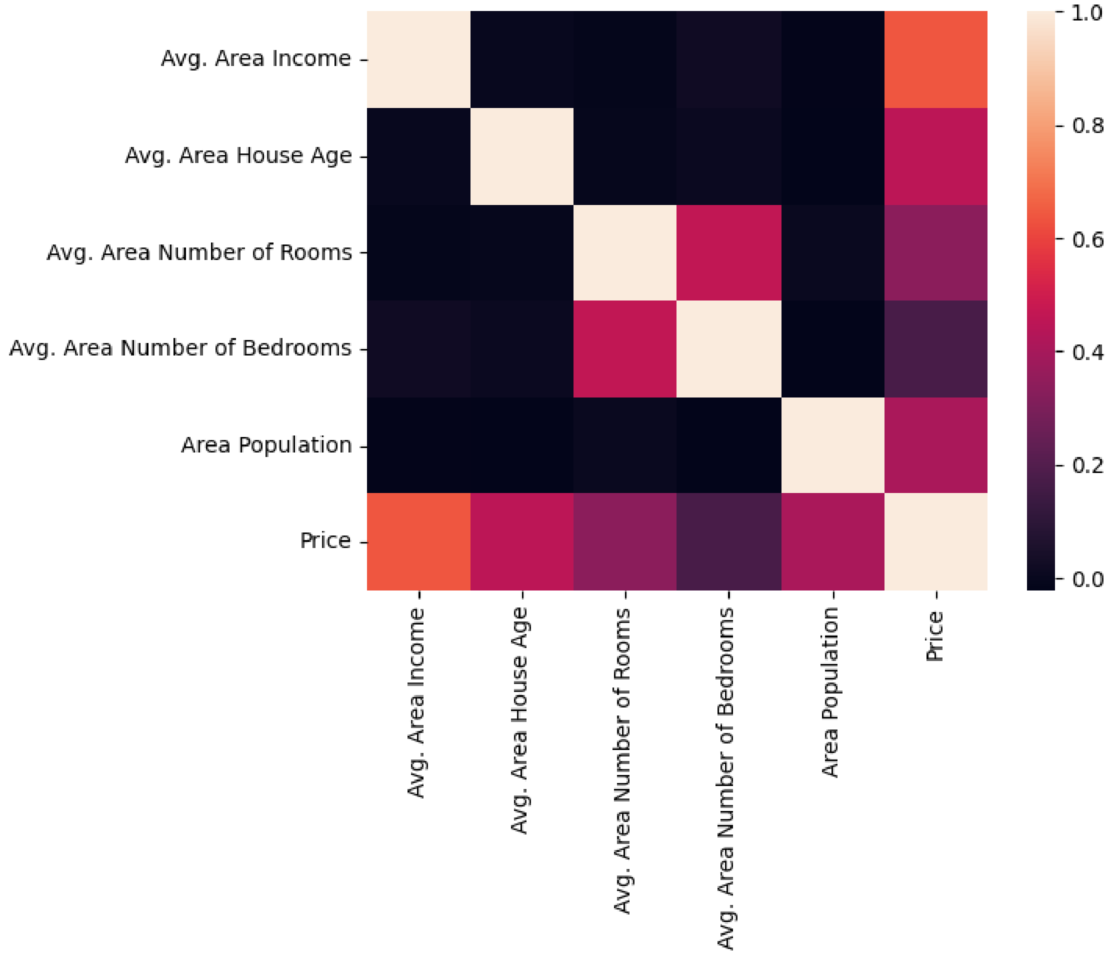
```
Housedf = df[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms
       'Avg. Area Number of Bedrooms', 'Area Population', 'Price']]
```

In [12]:

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

Out[12]:

```
<Axes: >
```



To Train the model

We are going to train the Linear Regression model.We need to first split up our data into X list that contain the features to train on,and a Y list with the target variable,in this case, the price column. We will ignore the Address column because it only has text which is not useful for linear regression modeling.

In [16]:

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

In [19]:

```
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 [20]:

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

Out[20]:

```
▾ LinearRegression
LinearRegression()
```

In [21]:

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

```
-2641372.667264207
```

In [22]:

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

Out[22]:

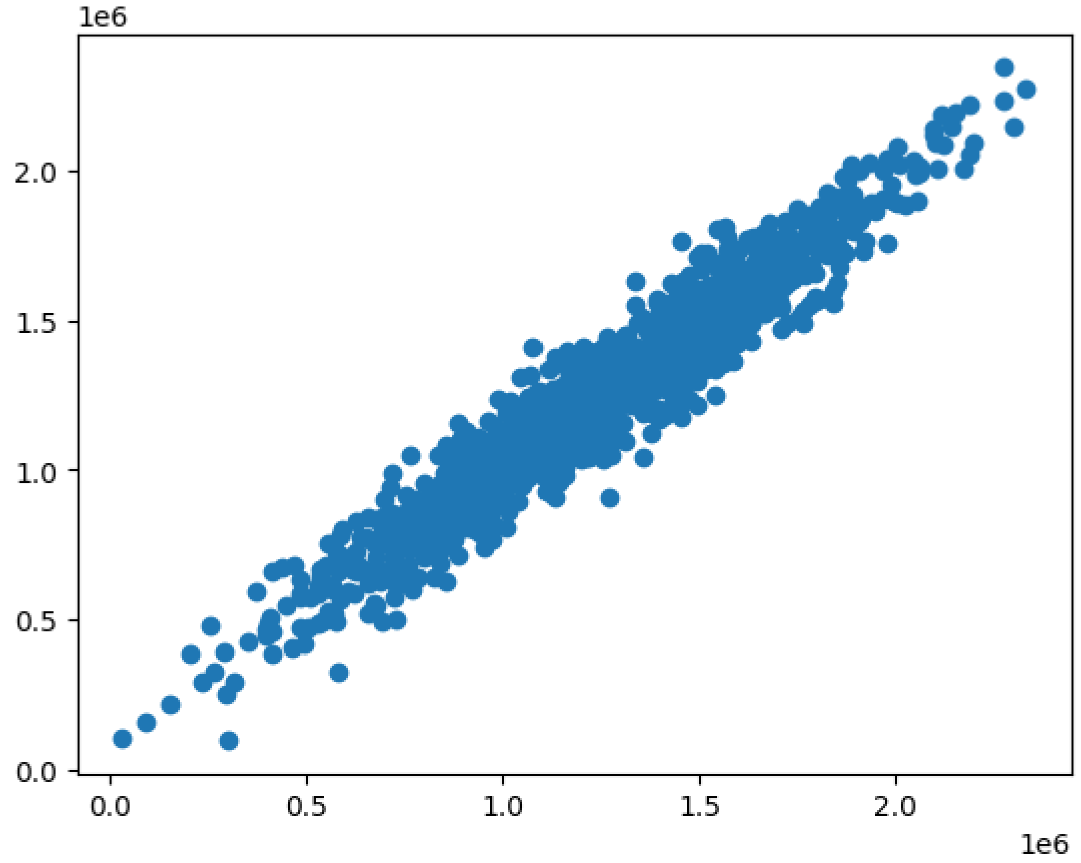
	coefficient
Avg. Area Income	21.617635
Avg. Area House Age	165221.119872
Avg. Area Number of Rooms	121405.376595
Avg. Area Number of Bedrooms	1318.718781
Area Population	15.225196

In [23]:

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

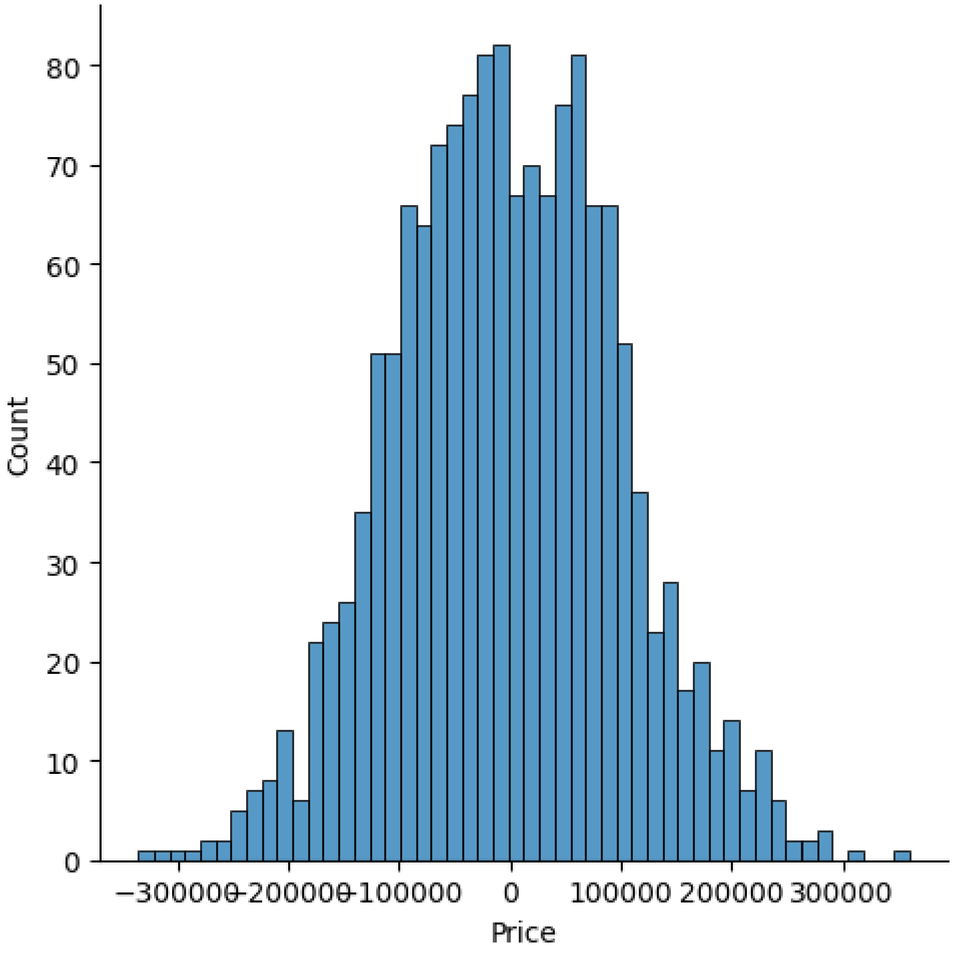
Out[23]:

```
<matplotlib.collections.PathCollection at 0x204b3245db0>
```



In [26]:

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



In [27]:

```
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: 81257.55794597675
MSE: 10169125565.18005
RMSE: 100842.08231279266
```

In [ ]: