In [56]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
df=pd.read_csv(r"C:\Users\G S R  KARTHIK\Downloads\archive (3)\USA_Housing.csv")
print(df)
```

```
      Avg. Area Income  Avg. Area House Age  Avg. Area Number of Rooms
0         79545.458574             5.682861                   7.009188  \
1         79248.642455             6.002900                   6.730821
2         61287.067179             5.865890                   8.512727
3         63345.240046             7.188236                   5.586729
4         59982.197226             5.040555                   7.839388
...                ...                  ...                        ...
4995      60567.944140             7.830362                   6.137356
4996      78491.275435             6.999135                   6.576763
4997      63390.686886             7.250591                   4.805081
4998      68001.331235             5.534388                   7.130144
4999      65510.581804             5.992305                   6.792336

      Avg. Area Number of Bedrooms  Area Population         Price
0                             4.09     23086.800503  1.059034e+06  \
1                             3.09     40173.072174  1.505891e+06
2                             5.13     36882.159400  1.058988e+06
3                             3.26     34310.242831  1.260617e+06
4                             4.23     26354.109472  6.309435e+05
...                            ...              ...           ...
4995                          3.46     22837.361035  1.060194e+06
4996                          4.02     25616.115489  1.482618e+06
4997                          2.13     33266.145490  1.030730e+06
4998                          5.44     42625.620156  1.198657e+06
4999                          4.07     46501.283803  1.298950e+06

                                                Address
0     208 Michael Ferry Apt. 674\nLaurabury, NE 3701...
1     188 Johnson Views Suite 079\nLake Kathleen, CA...
2     9127 Elizabeth Stravenue\nDanieltown, WI 06482...
3                             USS Barnett\nFPO AP 44820
4                            USNS Raymond\nFPO AE 09386
...                                                 ...
4995                   USNS Williams\nFPO AP 30153-7653
4996              PSC 9258, Box 8489\nAPO AA 42991-3352
4997  4215 Tracy Garden Suite 076\nJoshualand, VA 01...
4998                          USS Wallace\nFPO AE 73316
4999  37778 George Ridges Apt. 509\nEast Holly, NV 2...

[5000 rows x 7 columns]
```

In [57]:

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

Out[57]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	Ad
0	79545.458574	5.682861	7.009188	4.09	23086.800503	1.059034e+06	208 Michael Fe 674\nLaurabu
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
5	80175.754159	4.988408	6.104512	4.04	26748.428425	1.068138e+06	06039 Jennifer Apt. 443\nTra
6	64698.463428	6.025336	8.147760	3.41	60828.249085	1.502056e+06	4759 Daniel 442\nNguyenbur
7	78394.339278	6.989780	6.620478	2.42	36516.358972	1.573937e+06	972 Viaduct\nLake V TN 1777
8	59927.660813	5.362126	6.393121	2.30	29387.396003	7.988695e+05	USS Gilbert\nF
9	81885.927184	4.423672	8.167688	6.10	40149.965749	1.545155e+06	Unit 94 0958\nDPO AE

In [58]:

```
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 [59]:

```
df.describe()
```

Out[59]:

	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 [60]:

```
df.columns
```

Out[60]:

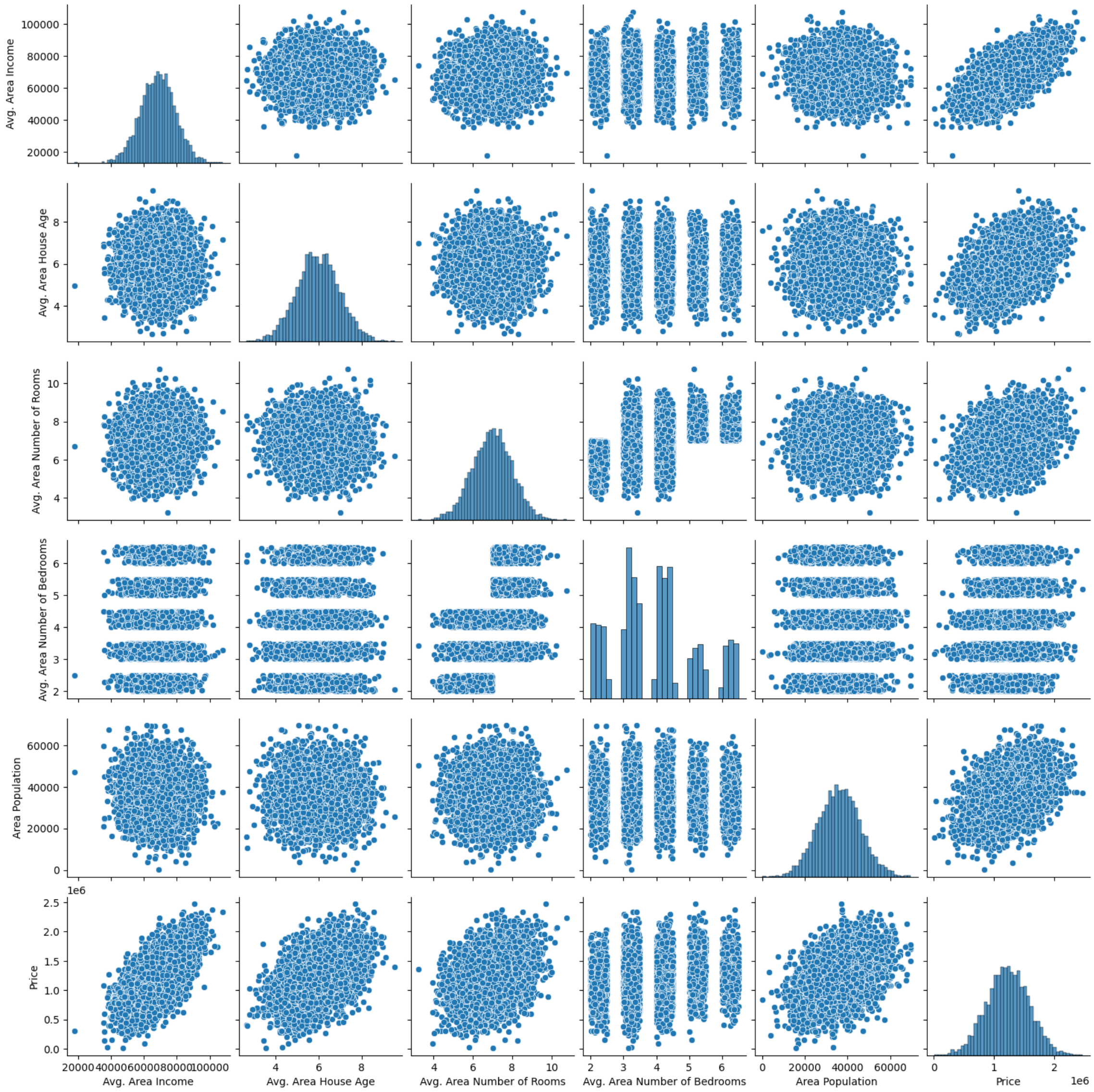
```
Index(['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Roo
ms',
       'Avg. Area Number of Bedrooms', 'Area Population', 'Price', 'Addres
s'],
      dtype='object')
```

In [61]:

```
#EDA
sns.pairplot(df)
```

Out[61]:

```
<seaborn.axisgrid.PairGrid at 0x1ca93dbf610>
```

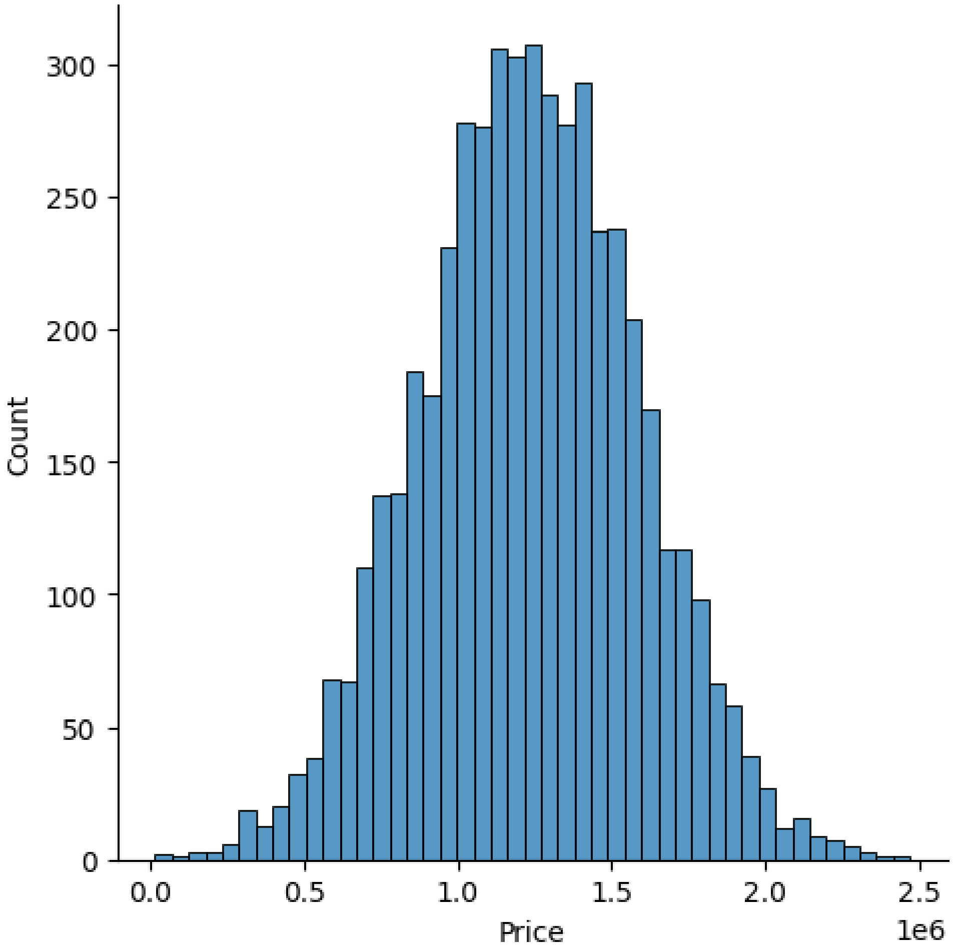


In [62]:

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

Out[62]:

```
<seaborn.axisgrid.FacetGrid at 0x1ca95421e70>
```

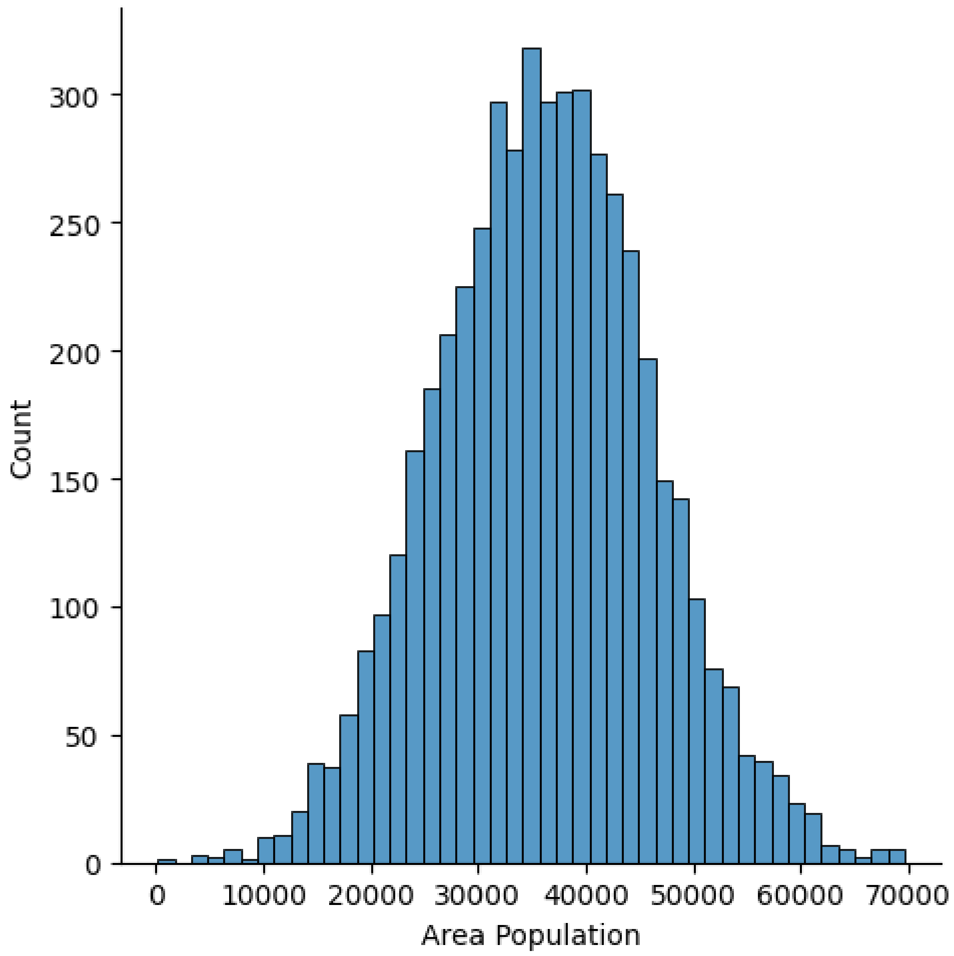


In [63]:

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

Out[63]:

```
<seaborn.axisgrid.FacetGrid at 0x1ca94160a90>
```



In [64]:

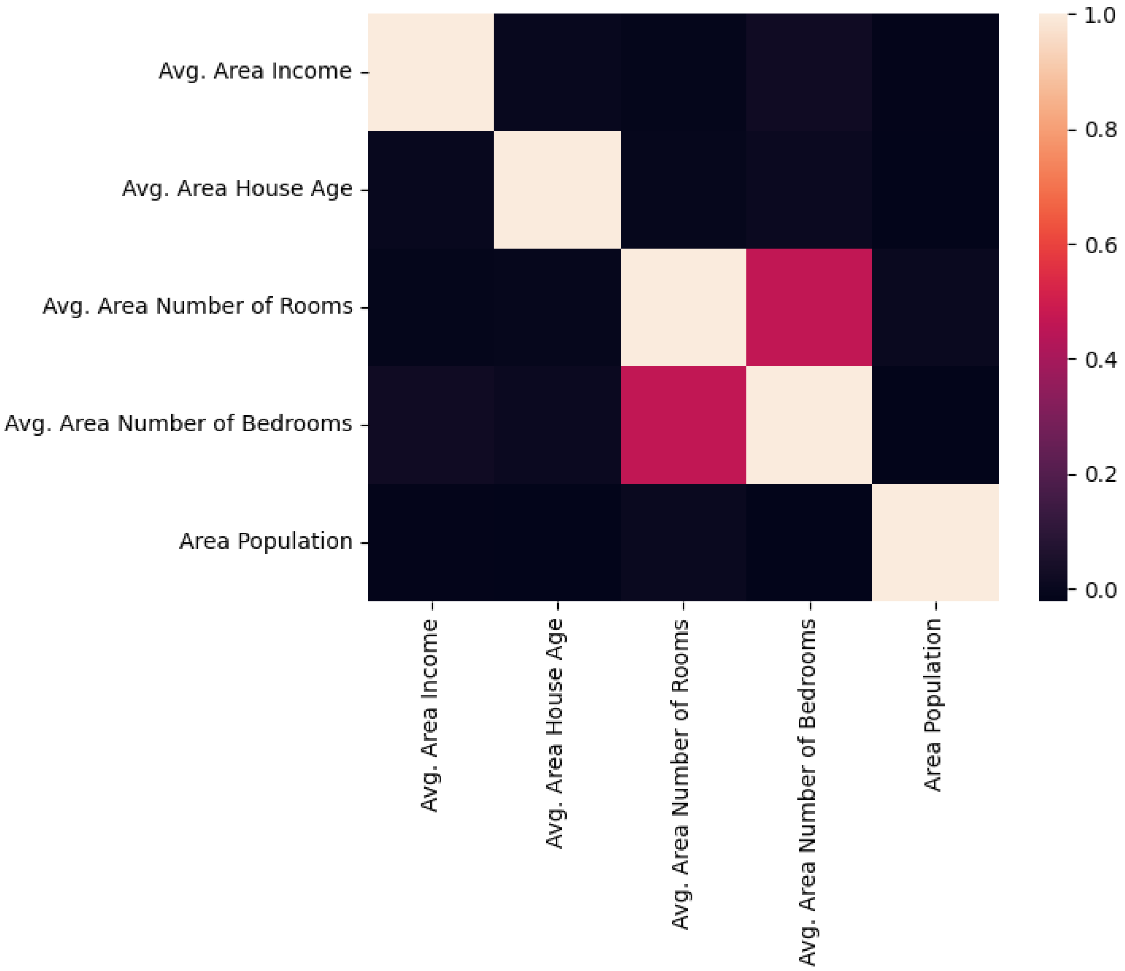
```
Housedf=df[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
       'Avg. Area Number of Bedrooms', 'Area Population']]
```

In [65]:

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

Out[65]:

```
<Axes: >
```

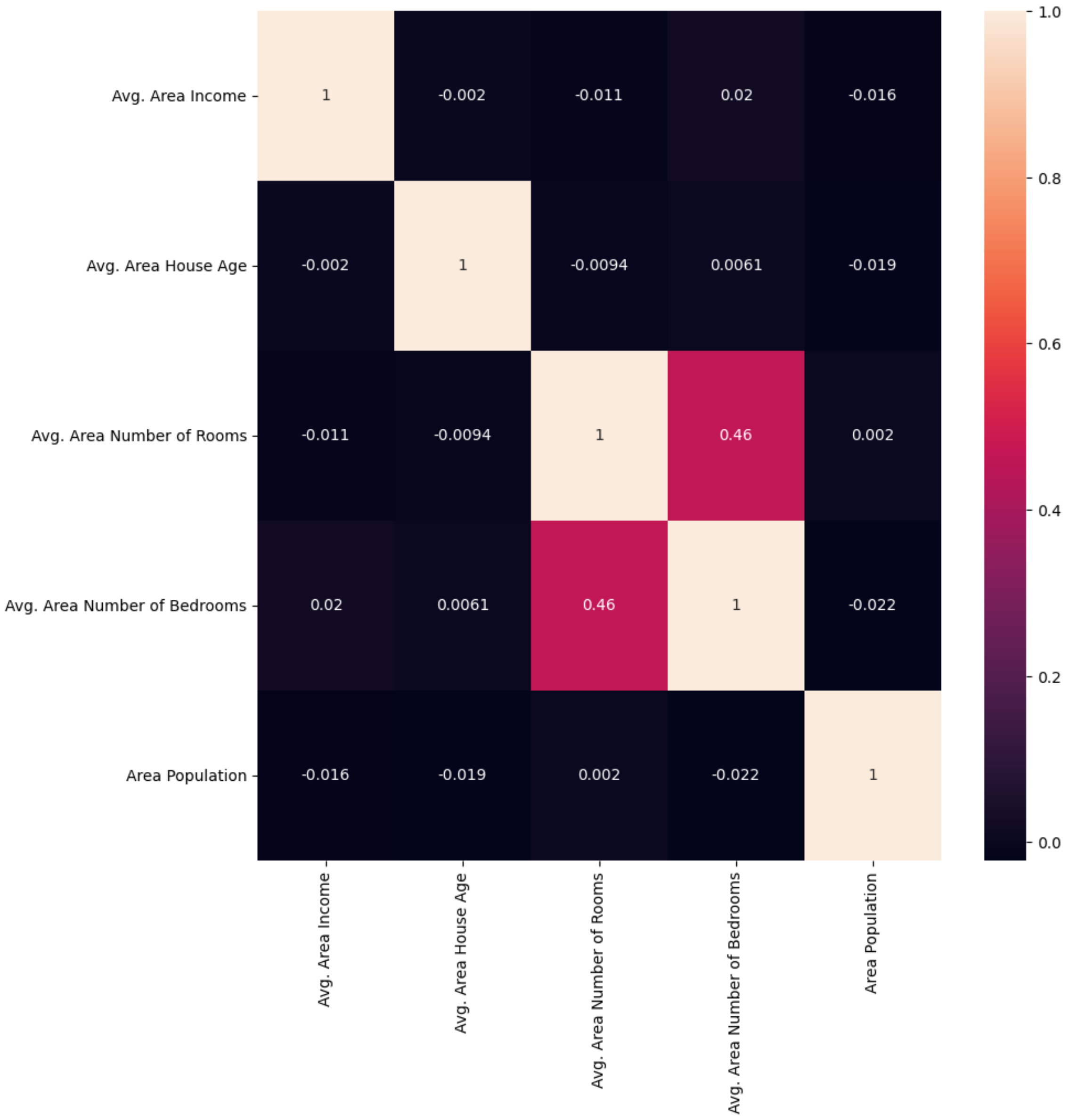


In [66]:

```
#ridge regression
plt.figure(figsize=(10,10))
sns.heatmap(Housedf.corr(),annot=True)
```

Out[66]:

```
<Axes: >
```



In [67]:

```
#Training the model
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 [68]:

```
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)
from sklearn.linear_model import LinearRegression
regr=LinearRegression()
regr.fit(X_train,y_train)
print(regr.intercept_)
```

```
-2641372.6673006266
```

In [69]:

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

Out[69]:

	coefficient
Avg. Area Income	21.617635
Avg. Area House Age	165221.119872
Avg. Area Number of Rooms	121405.376596
Avg. Area Number of Bedrooms	1318.718783
Area Population	15.225196

In [70]:

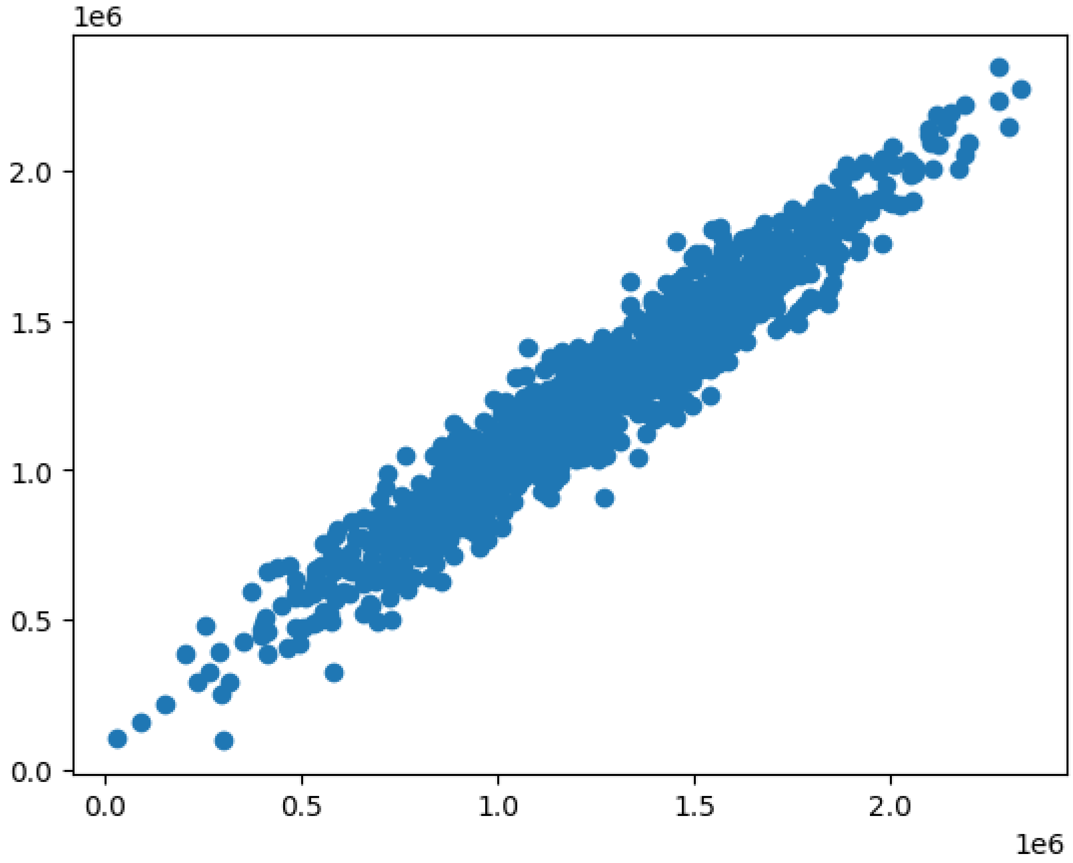
```
predictions=regr.predict(X_test)
```

In [71]:

```
plt.scatter(y_test,predictions)
```

Out[71]:

```
<matplotlib.collections.PathCollection at 0x1ca97b515a0>
```

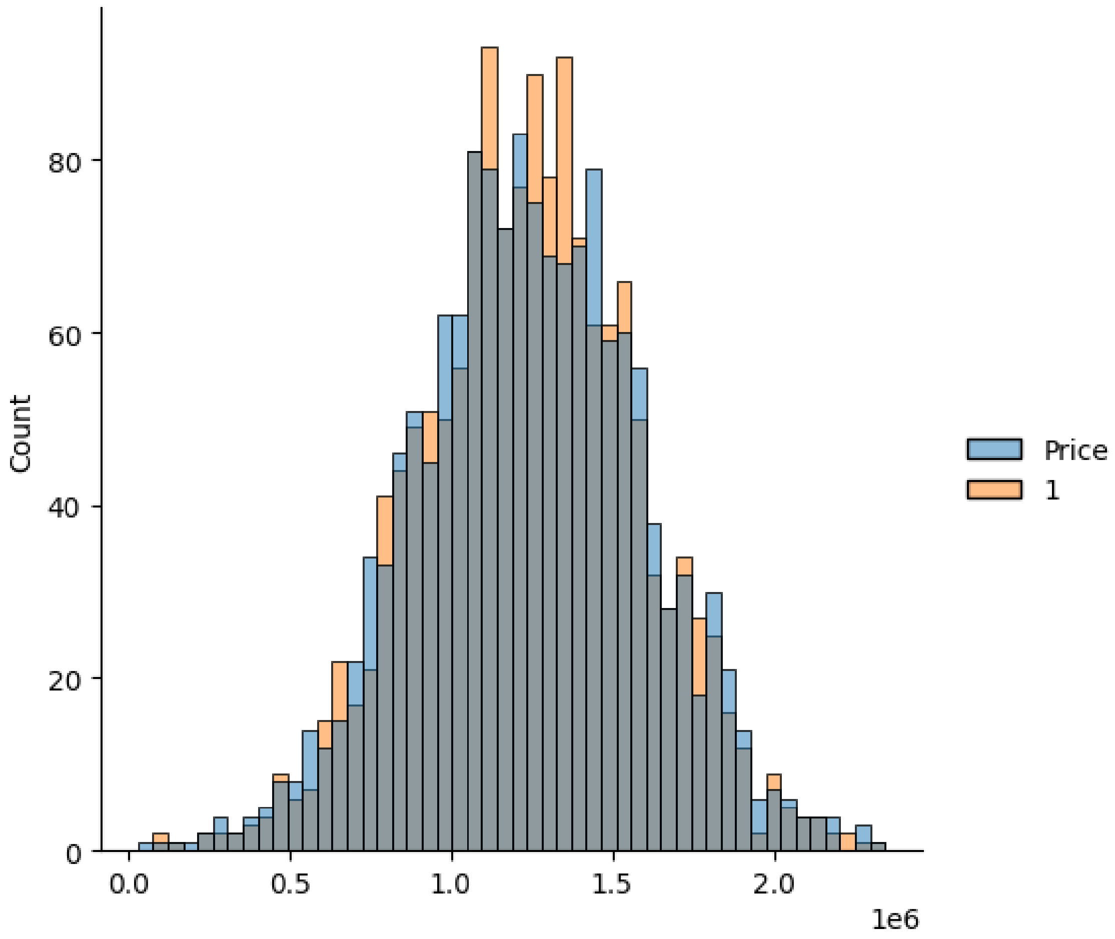


In [72]:

```
sns.displot((y_test,predictions),bins=50)#without semicolon
```

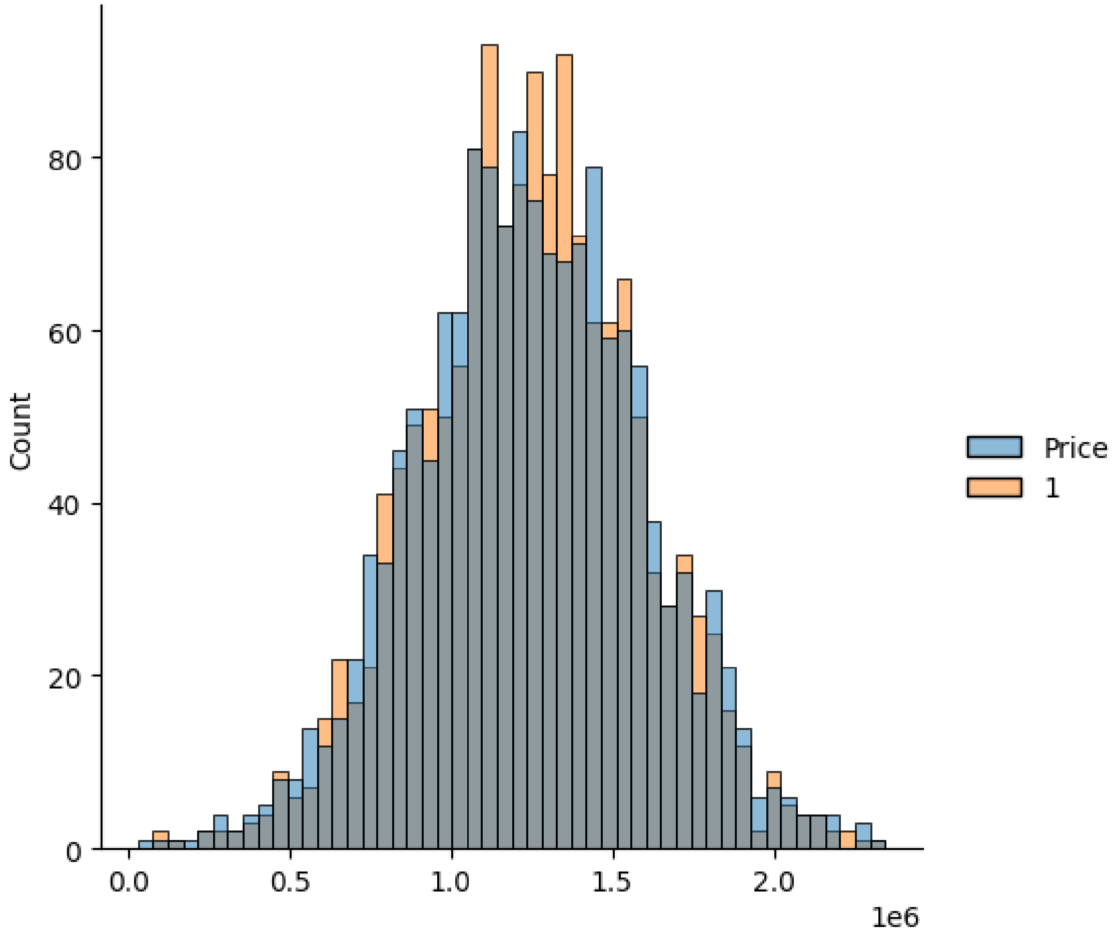
Out[72]:

```
<seaborn.axisgrid.FacetGrid at 0x1ca97b9fac0>
```



In [73]:

```
sns.displot((y_test,predictions),bins=50);#with semicolon
```



In [74]:

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

```
MAE: 81257.5579585557
MSE: 10169125565.89724
MAE: 100842.08231634866
```

In [75]:

```
#accuracy
regr=LinearRegression()
regr.fit(X_train,y_train)
regr.fit(X_train,y_train)
print(regr.score(X_test,y_test))
```

```
0.9185060945363651
```

In [76]:

```
df.fillna(method='ffill',inplace=True)
```

In [77]:

```
x=np.array(df['Avg. Area Income']).reshape(-1,1)
y=np.array(df['Price']).reshape(-1,1)
df.dropna(inplace=True)
```

In [78]:

```
X_train,X_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
regr.fit(X_train,y_train)
regr.fit(X_train,y_train)
```

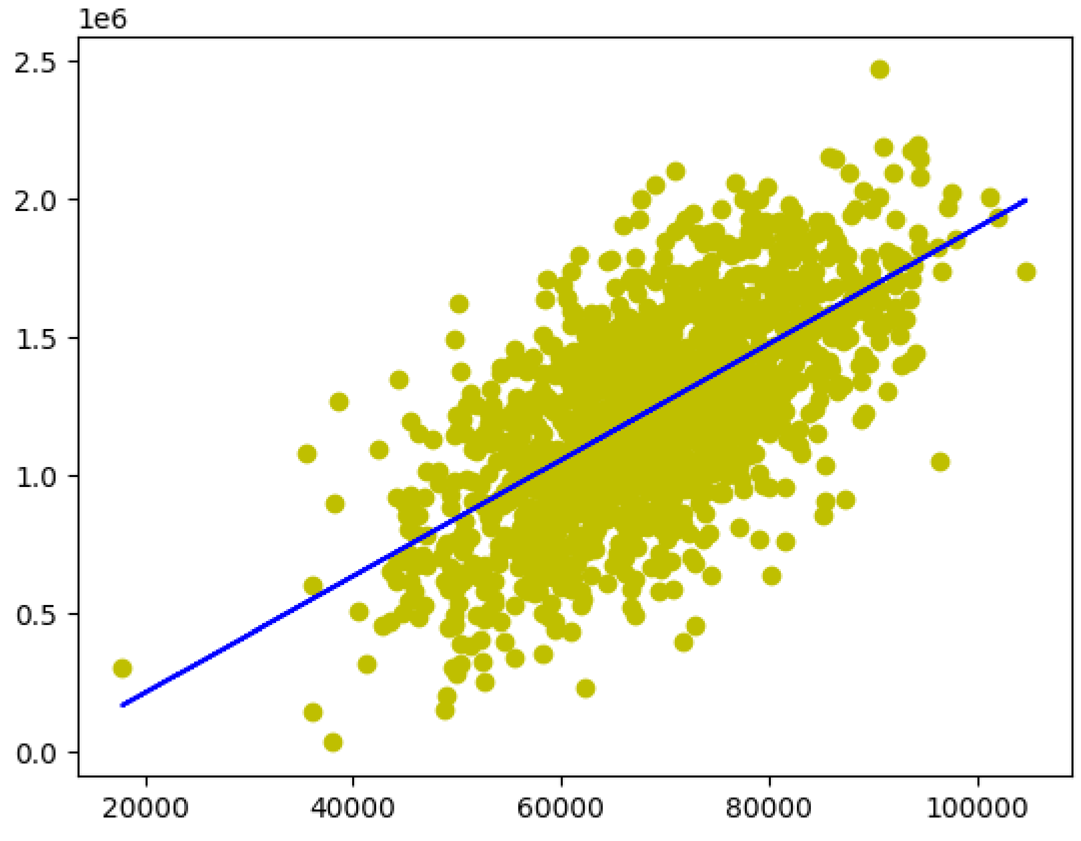
Out[78]:

```
LinearRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [79]:

```
y_pred=lm.predict(X_test)
plt.scatter(X_test,y_test,color='y')
plt.plot(X_test,y_pred,color='b')
plt.show()
```



In [ ]: