

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
data=pd.read_csv('USA_Housing.csv')
data
```

	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\nDanielstown, WI 06482...
3                                     USS Barnett\nFP0 AP 44820
4                                     USNS Raymond\nFP0 AE 09386
...
4995                                     USNS Williams\nFP0 AP 30153-7653
4996                                     PSC 9258, Box 8489\nAP0 AA 42991-3352
4997  4215 Tracy Garden Suite 076\nJoshualand, VA 01...
4998                                     USS Wallace\nFP0 AE 73316
4999  37778 George Ridges Apt. 509\nEast Holly, NV 2...

```

```
[5000 rows x 7 columns]
```

```
data.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
```

```
data.describe()
```

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms
count	5000.000000	5000.000000	5000.000000
mean	68583.108984	5.977222	6.987792
std	10657.991214	0.991456	1.005833
min	17796.631190	2.644304	3.236194
25%	61480.562388	5.322283	6.299250
50%	68804.286404	5.970429	7.002902
75%	75783.338666	6.650808	7.665871
max	107701.748378	9.519088	10.759588

	Avg. Area Number of Bedrooms	Area Population	Price
count	5000.000000	5000.000000	5.000000e+03
mean	3.981330	36163.516039	1.232073e+06
std	1.234137	9925.650114	3.531176e+05
min	2.000000	172.610686	1.593866e+04
25%	3.140000	29403.928702	9.975771e+05
50%	4.050000	36199.406689	1.232669e+06
75%	4.490000	42861.290769	1.471210e+06
max	6.500000	69621.713378	2.469066e+06

data.columns

```
Index(['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
      'Avg. Area Number of Bedrooms', 'Area Population', 'Price',
      'Address'],
      dtype='object')
```

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

x

	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

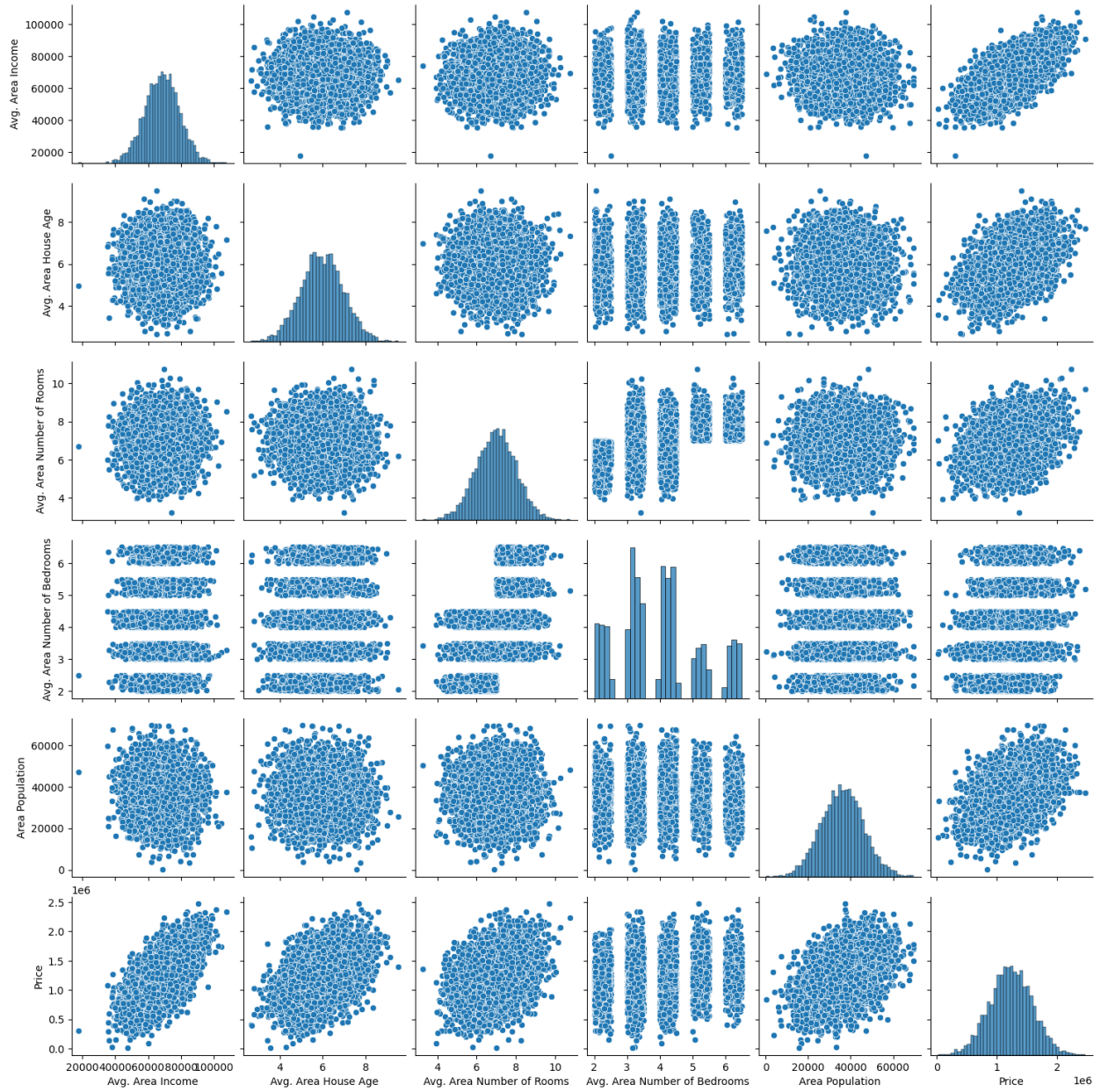
	Avg. Area Number of Bedrooms	Area Population
0	4.09	23086.800503

1	3.09	40173.072174
2	5.13	36882.159400
3	3.26	34310.242831
4	4.23	26354.109472
...	...	...
4995	3.46	22837.361035
4996	4.02	25616.115489
4997	2.13	33266.145490
4998	5.44	42625.620156
4999	4.07	46501.283803

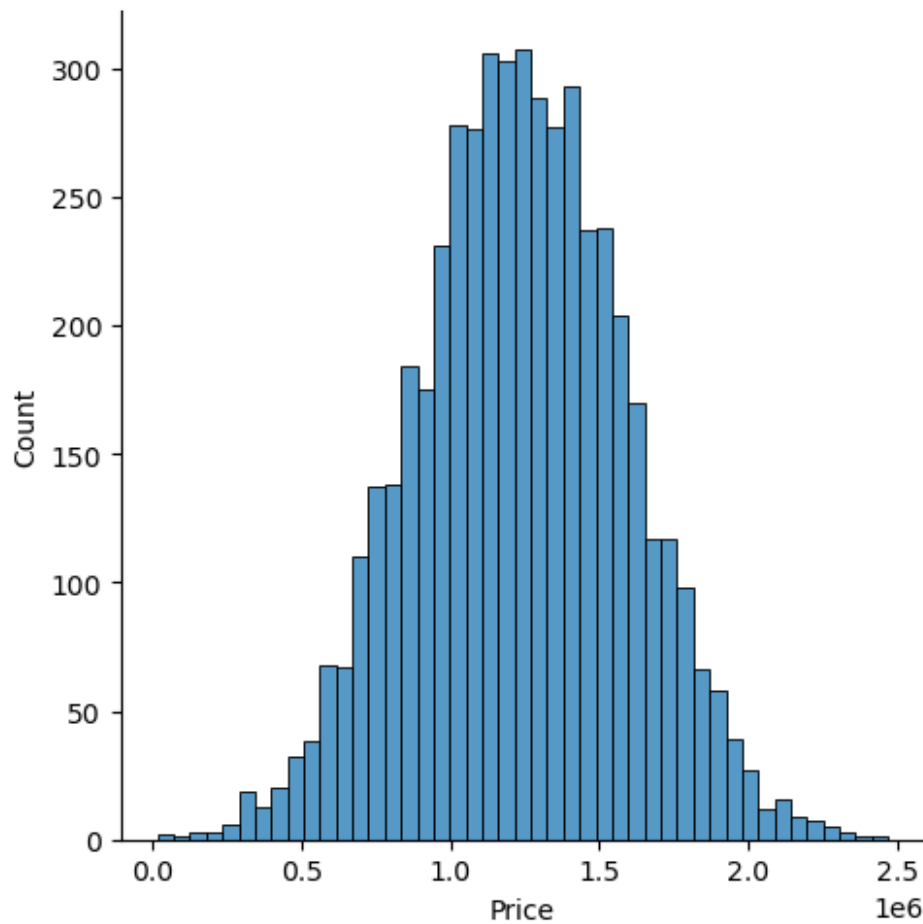
[5000 rows x 5 columns]

sns.pairplot(data)

<seaborn.axisgrid.PairGrid at 0x16f20381e10>



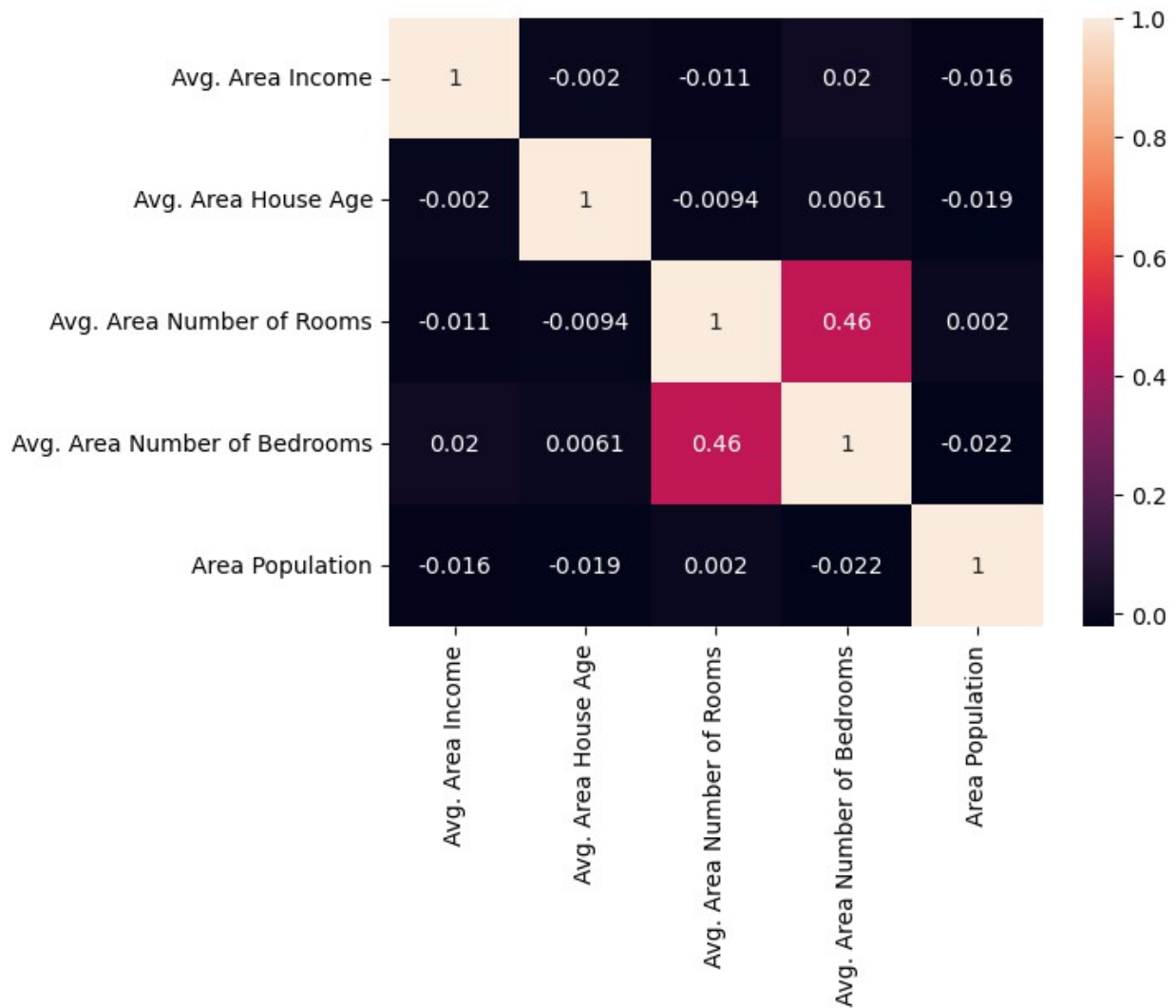
```
y=data['Price']
sns.displot(data['Price'])
<seaborn.axisgrid.FacetGrid at 0x16f220def50>
```



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

```
sns.heatmap(x.corr(), annot=True)
```

```
<Axes: >
```



```

from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(x, y,
test_size=0.4, random_state=10)

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

LinearRegression()
predict=lm.predict(X_test)

import sklearn
from sklearn.linear_model import LinearRegression

```

```

lm = LinearRegression()
model=lm.fit(X_train, Y_train)

Ytrain_pred = lm.predict(X_train)
Ytest_pred = lm.predict(X_test)

dt=pd.DataFrame(Ytrain_pred,Y_train)
dt=pd.DataFrame(Ytest_pred,Y_test)

from sklearn.metrics import mean_squared_error, r2_score
mse = mean_squared_error(Y_test, Ytest_pred)
print(mse)
mse = mean_squared_error(Ytrain_pred,Y_train)
print(mse)

10362287841.88252
10143862954.921873

mse = mean_squared_error(Y_test, Ytest_pred)
print(mse)

10362287841.88252

plt.scatter(Y_train, Ytrain_pred, c='blue', marker='o',
label='Training data')
plt.scatter(Y_test, Ytest_pred, c='lightgreen', marker='s',
label='Test data')

plt.xlabel('True values')
plt.ylabel('Predicted values')
plt.title("True values vs Predicted values")
plt.legend(loc='upper left')

plt.show()

```



