

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

```
HouseDF=pd.read_csv('USA_Housing.csv')
```

```
HouseDF.head(10)
```

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	Address
0	79545.458574	5.682861	7.009188	4.09	23086.800503	1.059034e+06	208 Michael Ferr 674\nLaurabui 3
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06	188 Johnson Suite 079\nKathleen,
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06	9127 Eliz Stravenue\nDanie WI 06
3	63345.240046	7.188236	5.586729	3.26	34310.242831	1.260617e+06	USS Barnett\nFF
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05	USNS Raymond\nAE (
5	80175.754159	4.988408	6.104512	4.04	26748.428425	1.068138e+06	06039 Jennifer Is Apt. 443\nTrac

```
HouseDF.tail(10)
```

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	Address
4990	52723.876555	5.452237	8.124571	6.39	14802.088438	4.795006e+05	86727 Kell Plaza\nLaki Veronica, IL 0447
4991	74102.191890	5.657841	7.683993	3.13	24041.270592	1.263721e+06	2871 Johi Lodge\nAmychestei GU 61734-559
4992	87499.125743	6.403473	4.836091	4.02	40815.199679	1.568701e+06	Unit 2096 Bo 9559\nDPO At 80983-879
4993	69639.140896	5.007510	7.778375	6.05	54056.128430	1.381831e+06	5259 Davi Causeway Apl 975\nSouth Alexstad
4994	73060.846226	5.293682	6.312253	4.16	22695.695480	9.053549e+05	5224 Lam Passage\nNancystad GA 1657
4995	60567.944140	7.830362	6.137356	3.46	22837.361035	1.060194e+06	USNS Williams\nFPC AP 30153-765 DSC 0258 B

```
HouseDF.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
```

```
HouseDF.describe()
```

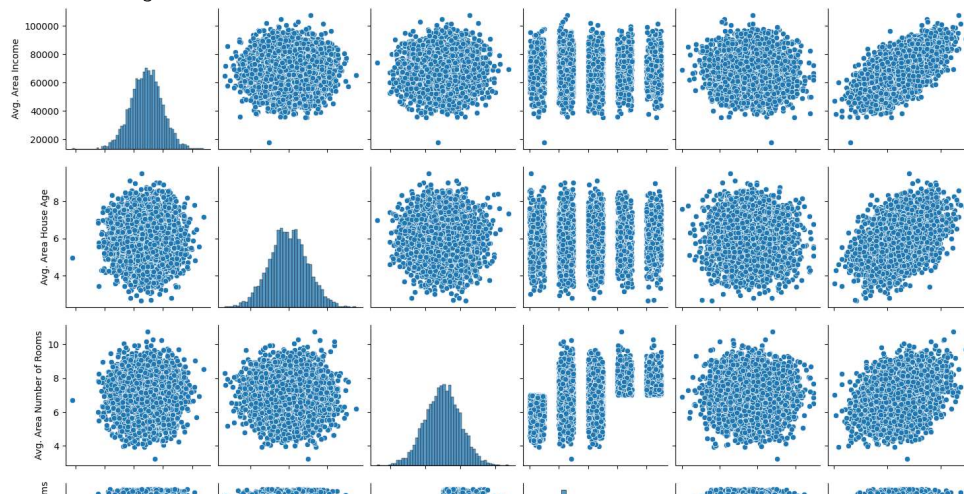
	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5.000000e+03
mean	68583.108984	5.977222	6.987792	3.981330	36163.516039	1.232073e+06
std	10657.991214	0.991456	1.005833	1.234137	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

```
HouseDF.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')
```

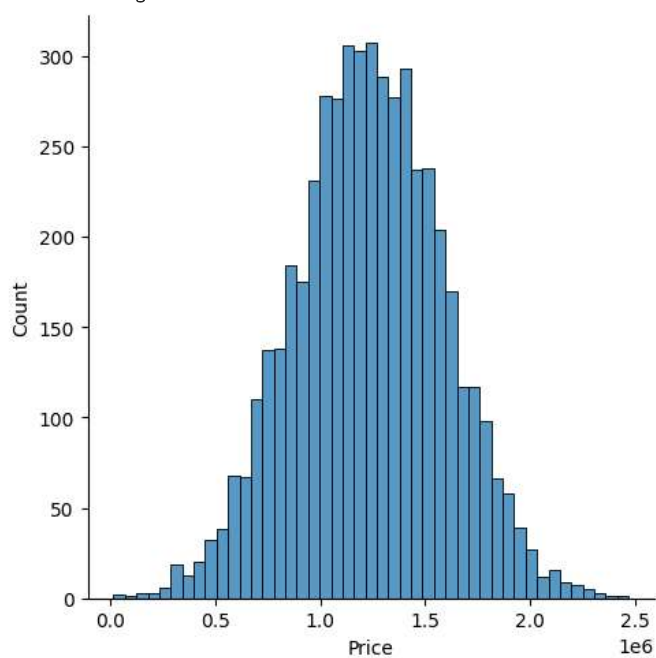
```
sns.pairplot(HouseDF)
```

```
<seaborn.axisgrid.PairGrid at 0x7f18b15fedf0>
```



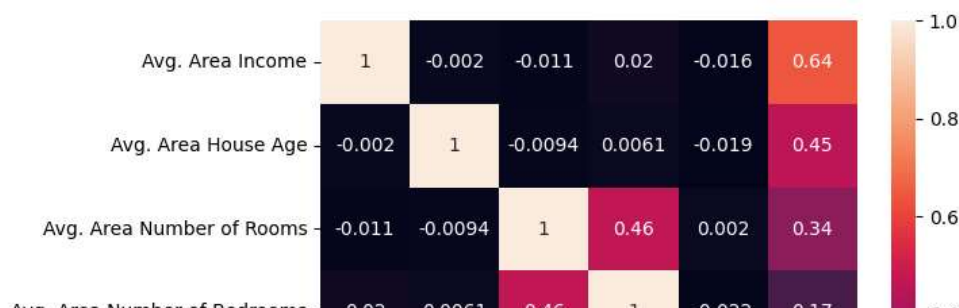
```
sns.displot(HouseDF[ 'Price' ])
```

```
<seaborn.axisgrid.FacetGrid at 0x7f18d01db820>
```



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

```
<ipython-input-18-8cf50b268fbb>:1: FutureWarning: The default value of numeric_only in DataFrames.heatmap(HouseDF.corr(),annot=True)
<Axes: >
```



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

```
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=101)
```

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

```
lm=LinearRegression()
```

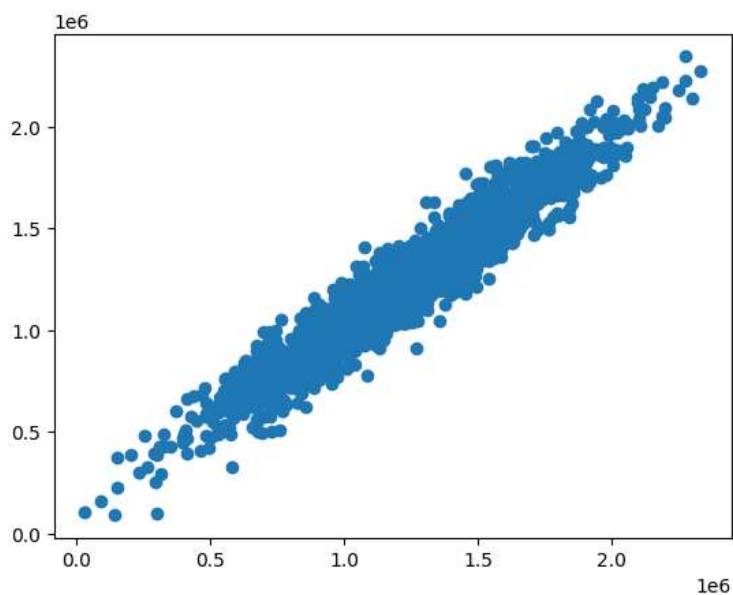
```
lm.fit(X_train,y_train)
```

```
LinearRegression
LinearRegression()
```

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

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

```
<matplotlib.collections.PathCollection at 0x7f187b9c3520>
```



Double-click (or enter) to edit

```
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_absolute_error(y_test,predictions)))
```

```
MAE: 82288.22251914942  
MSE: 10460958907.208977  
RMSE: 286.8592381624643
```

