

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [15]: df=pd.read_csv("10_USA_Housing.csv")
df
```

Out[15]:

	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\nLaurabur 3
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06	188 Johnson \\\nSuite 079\nr Kathleen,
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06	9127 Eliz Stravenue\nDaniel WI 06
3	63345.240046	7.188236	5.586729	3.26	34310.242831	1.260617e+06	USS Barnett\nFP 4
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05	USNS Raymond\nAE C
...	
4995	60567.944140	7.830362	6.137356	3.46	22837.361035	1.060194e+06	USNS Williams\nAP 30153-
4996	78491.275435	6.999135	6.576763	4.02	25616.115489	1.482618e+06	PSC 9258 8489\nAPO AA 4:
4997	63390.686886	7.250591	4.805081	2.13	33266.145490	1.030730e+06	4215 Tracy G. Suite 076\nJoshua VA
4998	68001.331235	5.534388	7.130144	5.44	42625.620156	1.198657e+06	USS Wallace\nFP 7
4999	65510.581804	5.992305	6.792336	4.07	46501.283803	1.298950e+06	37778 George R Apt. 509\nEast N

5000 rows × 7 columns

In [3]: 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
```

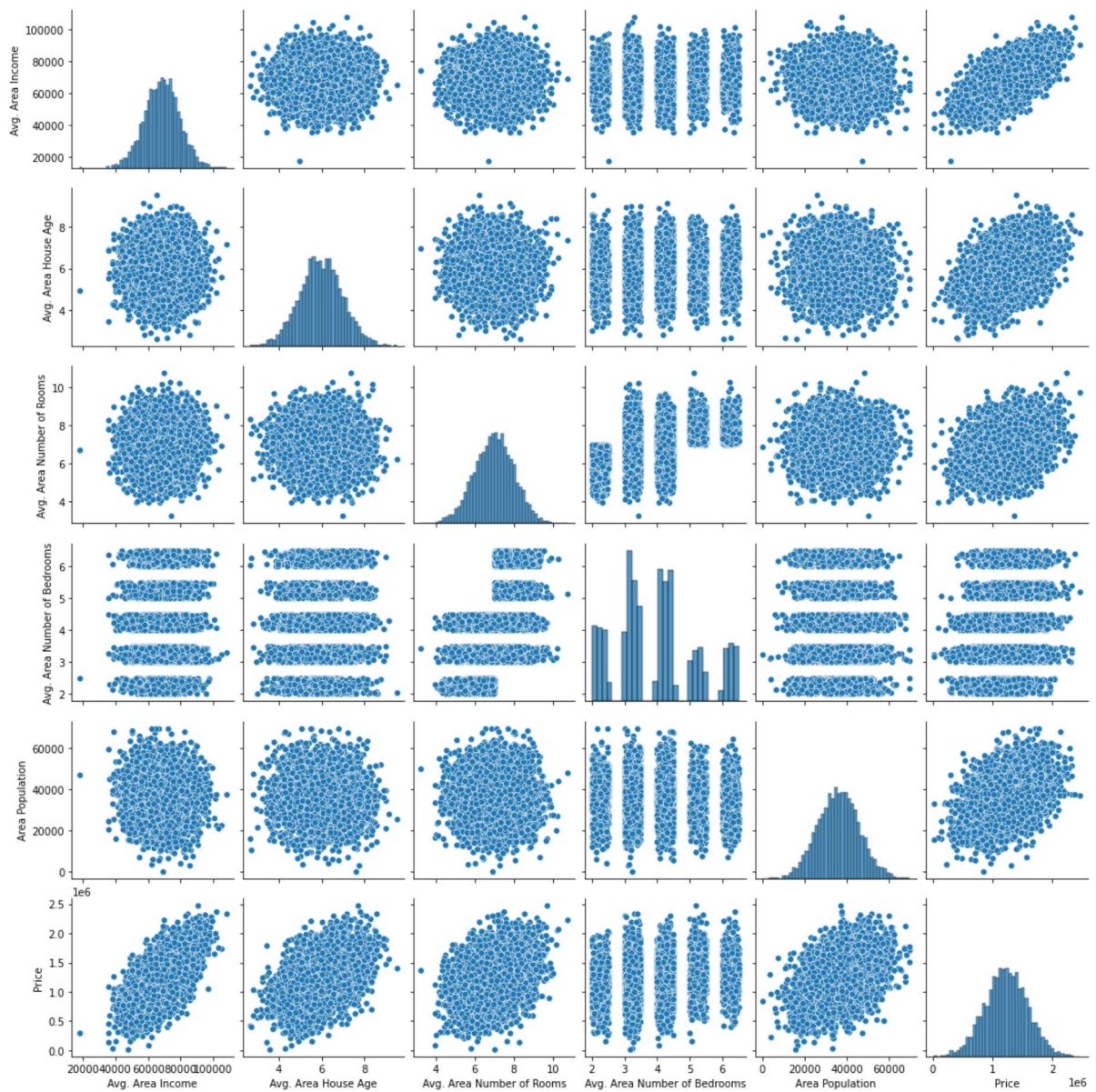
In [4]: df.describe()

Out[4]:

	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

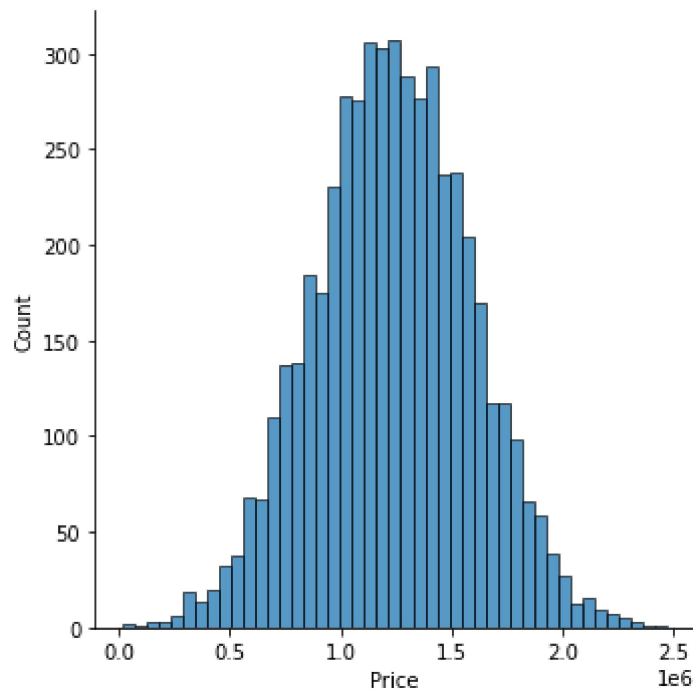
```
In [5]: sns.pairplot(df)
```

```
Out[5]: <seaborn.axisgrid.PairGrid at 0x19fc66de3a0>
```



```
In [6]: sns.displot(df['Price'])
```

```
Out[6]: <seaborn.axisgrid.FacetGrid at 0x19fcc7f1130>
```



```
In [16]: df1=df.drop(['Address'],axis=1)
df1
```

```
Out[16]:
```

	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
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06
3	63345.240046	7.188236	5.586729	3.26	34310.242831	1.260617e+06
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05
...
4995	60567.944140	7.830362	6.137356	3.46	22837.361035	1.060194e+06
4996	78491.275435	6.999135	6.576763	4.02	25616.115489	1.482618e+06
4997	63390.686886	7.250591	4.805081	2.13	33266.145490	1.030730e+06
4998	68001.331235	5.534388	7.130144	5.44	42625.620156	1.198657e+06
4999	65510.581804	5.992305	6.792336	4.07	46501.283803	1.298950e+06

5000 rows × 6 columns

```
In [8]: sns.heatmap(df1.corr())
```

```
Out[8]: <AxesSubplot:>
```



```
In [9]: from sklearn.model_selection import train_test_split  
from sklearn.linear_model import LinearRegression
```

```
In [53]: y=df['Price']
x=df1.drop(['Price'],axis=1)
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
print(x_train)
```

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	\
4300	78358.948450	5.258160	8.102096	
3408	65417.891826	6.222731	7.701180	
1705	65018.541208	5.981193	6.449330	
348	61881.163717	6.293585	6.048295	
1802	61475.255362	6.578331	7.656512	
...	
3068	70217.780346	5.576018	6.431351	
679	84364.707328	4.698058	7.107708	
1698	65543.338541	3.945932	7.424297	
2664	65885.135759	7.652591	6.196093	
3295	54649.236026	5.763118	7.867808	

	Avg. Area Number of Bedrooms	Area Population
4300	6.19	40591.325222
3408	5.25	39446.674858
1705	3.20	32580.780183
348	2.31	21397.448601
1802	6.37	31437.514861
...
3068	4.10	38779.346013
679	3.46	50521.104235
1698	6.38	28939.038840
2664	4.02	34100.916771
3295	4.10	35497.432191

[3500 rows x 5 columns]

```
In [54]: model=LinearRegression()
model.fit(x_train,y_train)
model.intercept_
```

Out[54]: -2646608.015189554

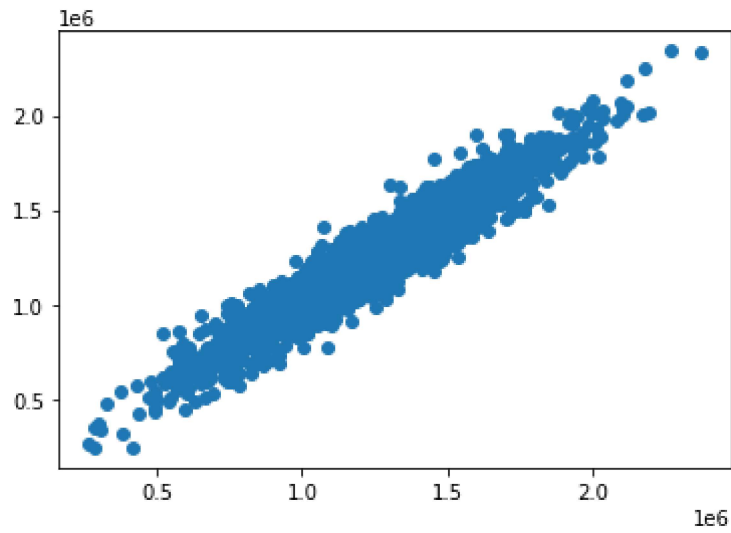
```
In [58]: coeff=pd.DataFrame(model.coef_,x.columns,columns=["Coefficient"])
coeff
```

Out[58]:

	Coefficient
Avg. Area Income	21.648342
Avg. Area House Age	166558.015319
Avg. Area Number of Rooms	119812.907738
Avg. Area Number of Bedrooms	1997.031582
Area Population	15.271302

```
In [60]: prediction=model.predict(x_test)  
plt.scatter(y_test,prediction)
```

Out[60]: <matplotlib.collections.PathCollection at 0x19fce9fcac0>



```
In [62]: model.score(x_test,y_test)
```

Out[62]: 0.9107734856921565