In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [215]:
```
a=pd.read_csv(r"C:\Users\user\Downloads\10_USA_Housing.csv")
a
```

Out[215]:

	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 Ferry Apt. 674\nLaurabury, NE 3701...
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06	188 Johnson Views Suite 079\nLake Kathleen, CA...
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06	9127 Elizabeth Stravenue\nDanieltown, WI 06482...
3	63345.240046	7.188236	5.586729	3.26	34310.242831	1.260617e+06	USS Barnett\nFPO AP 44820
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05	USNS Raymond\nFPO AE 09386

In [216]:
```
a=a.head(10)
a
```

Out[216]:

	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 Ferry Apt. 674\nLaurabury, NE 3701...
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06	188 Johnson Views Suite 079\nLake Kathleen, CA...
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06	9127 Elizabeth Stravenue\nDanieltown, WI 06482...
3	63345.240046	7.188236	5.586729	3.26	34310.242831	1.260617e+06	USS Barnett\nFPO AP 44820
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05	USNS Raymond\nFPO AE 09386
5	80175.754159	4.988408	6.104512	4.04	26748.428425	1.068138e+06	06039 Jennifer Islands Apt. 443\nTracyport, KS...
6	64698.463428	6.025336	8.147760	3.41	60828.249085	1.502056e+06	4759 Daniel Shoals Suite 442\nNguyenburgh, CO ...
7	78394.339278	6.989780	6.620478	2.42	36516.358972	1.573937e+06	972 Joyce Viaduct\nLake William, TN 17778-6483
8	59927.660813	5.362126	6.393121	2.30	29387.396003	7.988695e+05	USS Gilbert\nFPO AA 20957
9	81885.927184	4.423672	8.167688	6.10	40149.965749	1.545155e+06	Unit 9446 Box 0958\nDPO AE 97025

In [217]:
```
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 7 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   Avg. Area Income             10 non-null     float64
 1   Avg. Area House Age          10 non-null     float64
 2   Avg. Area Number of Rooms    10 non-null     float64
 3   Avg. Area Number of Bedrooms  10 non-null     float64
 4   Area Population              10 non-null     float64
 5   Price                        10 non-null     float64
 6   Address                      10 non-null     object 
dtypes: float64(6), object(1)
memory usage: 688.0+ bytes
```

In [218]:
```
a.columns
```

Out[218]:
```
Index(['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
       'Avg. Area Number of Bedrooms', 'Area Population', 'Price', 'Address'],
      dtype='object')
```

In [219]:
```
a.head()
```

Out[219]:

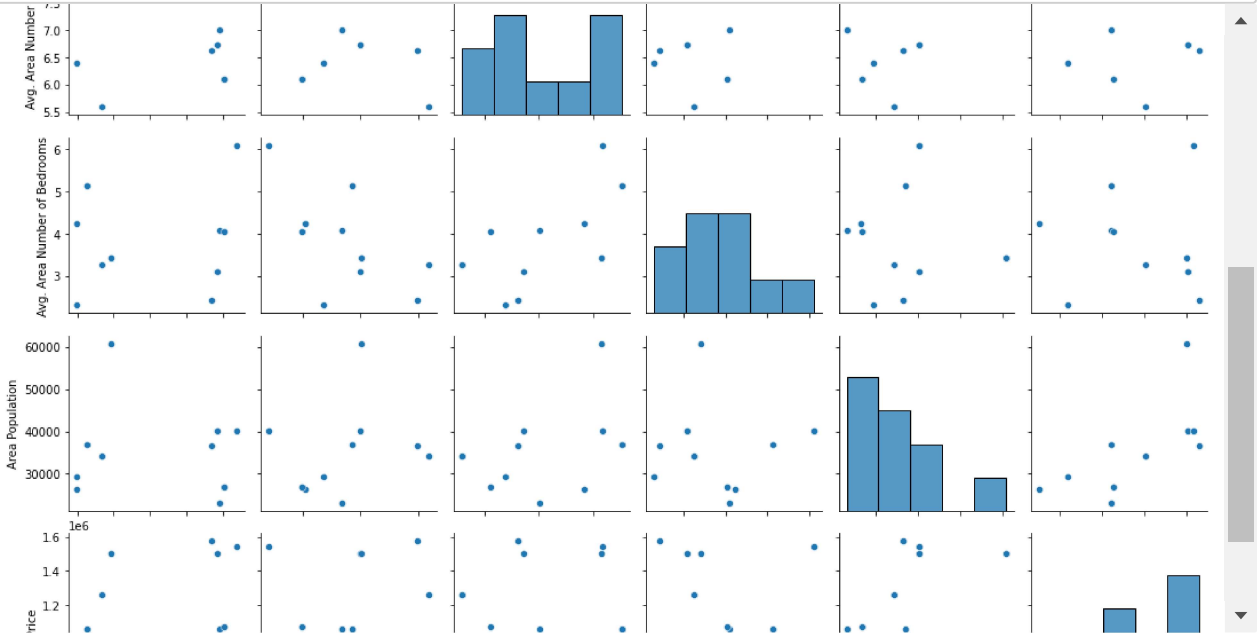
	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	Address
0	79545.458574	5.682861	7.009188	4.09	23086.800503	1.059034e+06	208 Michael Ferry Apt. 674\nLaurabury, NE 3701...
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06	188 Johnson Views Suite 079\nLake Kathleen, CA...
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06	9127 Elizabeth Stravenue\nDanieltown, WI 06482...
3	63345.240046	7.188236	5.586729	3.26	34310.242831	1.260617e+06	USS Barnett\nFPO AP 44820
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05	USNS Raymond\nFPO AE 09386

In [220]:
```
a.describe()
```

Out[220]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price
count	10.000000	10.000000	10.000000	10.000000	10.000000	1.000000e+01
mean	70849.075034	5.756976	7.111241	3.807000	35443.678261	1.200363e+06
std	9631.232526	0.866393	0.996334	1.177908	10754.822720	3.315477e+05
min	59927.660813	4.423672	5.586729	2.300000	23086.800503	6.309435e+05
25%	61801.610396	5.120947	6.449960	3.132500	27408.170319	1.058999e+06
50%	71546.401353	5.774376	6.870005	3.725000	35413.300902	1.164377e+06
75%	79471.254544	6.019727	8.070667	4.195000	39333.014162	1.504932e+06
max	81885.927184	7.188236	8.512727	6.100000	60828.249085	1.573937e+06

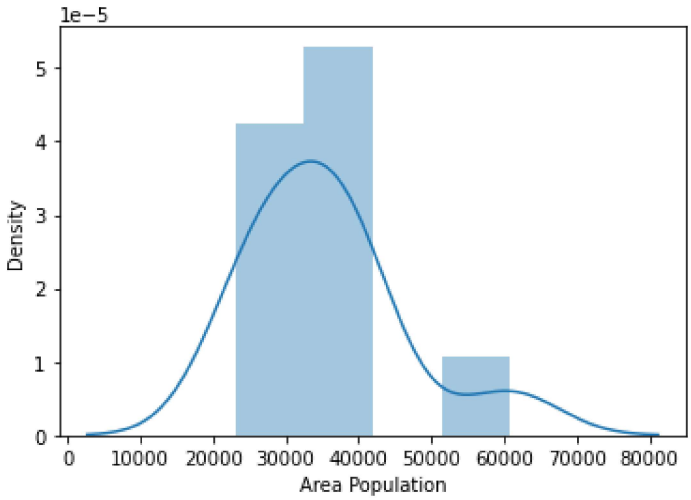
In [221]:
```
sns.pairplot(a)
```



In [222]:
```
sns.distplot(a['Area Population'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarnin
g: `distplot` is a deprecated function and will be removed in a future version. Please
adapt your code to use either `displot` (a figure-level function with similar flexibil
ity) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[222]: <AxesSubplot:xlabel='Area Population', ylabel='Density'>



In [225]:
```
x1=a[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms']]
```

In [226]:
```
sns.heatmap(x1.corr())
```

Out[226]: <AxesSubplot:>

In [227]:
```
x=a[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms']]
y=a['Area Population']
```

In [228]:
```
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)
```

In [229]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[229]: LinearRegression()

In [230]:
```
print(lr.intercept_)
```

```
-26194.79728199637
```

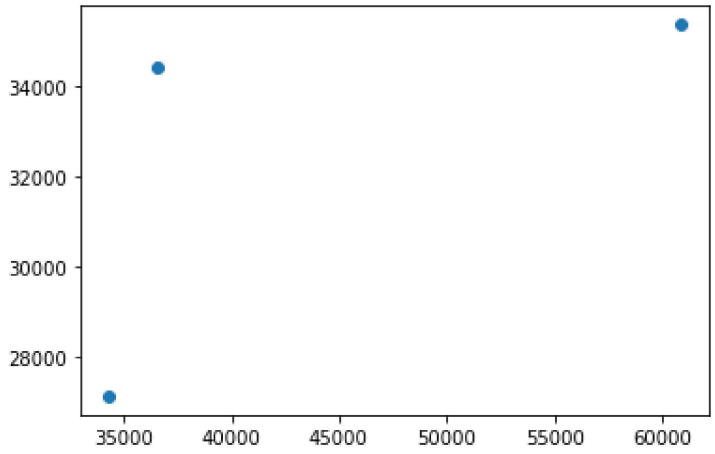
In [231]:
```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[231]:

	Co-efficient
Avg. Area Income	0.232011
Avg. Area House Age	2197.794025
Avg. Area Number of Rooms	4088.861223

In [232]:
```
prediction=lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[232]:
```
<matplotlib.collections.PathCollection at 0x22eb65305e0>
```



In [233]:
```
print(lr.score(x_test,y_test))
```

```
-0.6249463946990192
```

In [234]:
```
from sklearn.linear_model import Ridge,Lasso
```

In [235]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[235]:
```
Ridge(alpha=10)
```

In [236]:
```
rr.score(x_test,y_test)
```

Out[236]:
```
-1.0172030813258703
```

In [237]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[237]:
```
Lasso(alpha=10)
```

In [238]:
```
la.score(x_test,y_test)
```

Out[238]:
```
-0.6319357363018048
```

In [239]:
```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
```

Out[239]:
```
ElasticNet()
```

In [240]:
```
print(en.coef_)
```

```
[1.71664124e-01 5.57075917e+02 2.24794401e+03]
```

In [241]:
```
print(en.intercept_)
```

```
240.2385333075581
```

In [242]:
```
print(en.predict(x_test))
```

```
[32474.03593588 27677.39013338 33018.92046175]
```

In [243]:
```
en.score(x_test,y_test)
```

Out[243]:
```
-0.9251830959319642
```

In [244]:
```
from sklearn import metrics
```

In [245]:
```
print("Mean Absolute Error",metrics.mean_absolute_error(y_test,prediction))
```

```
Mean Absolute Error 11570.644783813244
```

In [246]:
```
print("Mean Squared Error",metrics.mean_squared_error(y_test,prediction))
```

```
Mean Squared Error 234559135.26284567
```

In [247]:
```
print(" Root Mean Squared Error",np.sqrt(metrics.mean_squared_error(y_test,prediction))
```

```
 Root Mean Squared Error 15315.323544177762
```