In [136]:

```
# IMPORT LIBRARIES
import numpy  as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [137]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\USA_Housing.csv")
a
```

Out[137]:

	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	208 Michael 674\nLaur
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06	188 Johns Suite 0 Kathl
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06	9127 Stravenue\nDa W
3	63345.240046	7.188236	5.586729	3.26	34310.242831	1.260617e+06	USS Barnett\
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05	USNS Raymc
...	...	...	...	...	...	...	
4995	60567.944140	7.830362	6.137356	3.46	22837.361035	1.060194e+06	USNS Willia AP 30
4996	78491.275435	6.999135	6.576763	4.02	25616.115489	1.482618e+06	PSC 9 8489\nAPO A
4997	63390.686886	7.250591	4.805081	2.13	33266.145490	1.030730e+06	4215 Trac Suite 076\nJo
4998	68001.331235	5.534388	7.130144	5.44	42625.620156	1.198657e+06	USS Wallace\
4999	65510.581804	5.992305	6.792336	4.07	46501.283803	1.298950e+06	37778 Georg Apt. 509\nE

5000 rows × 7 columns

In [138]:

```
a=a.head(10)
a
```

Out[138]:

	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 Ferr 674\nLaurabu 3
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06	188 Johnson Suite 079\r Kathleen,
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\ AE (
5	80175.754159	4.988408	6.104512	4.04	26748.428425	1.068138e+06	06039 Jennifer Is Apt. 443\nTrac
6	64698.463428	6.025336	8.147760	3.41	60828.249085	1.502056e+06	4759 Daniel S 442\nNguyenburg
7	78394.339278	6.989780	6.620478	2.42	36516.358972	1.573937e+06	972 Viaduct\nLake W TN 17778
8	59927.660813	5.362126	6.393121	2.30	29387.396003	7.988695e+05	USS Gilbert\nFF
9	81885.927184	4.423672	8.167688	6.10	40149.965749	1.545155e+06	Unit 944 0958\nDPO AE 9

In [139]:

```
# to find
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 [140]:

```
# to display summary of statastic
a.describe()
```

Out[140]:

	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 [141]:

```
# to display colum heading
a.columns
```

Out[141]:

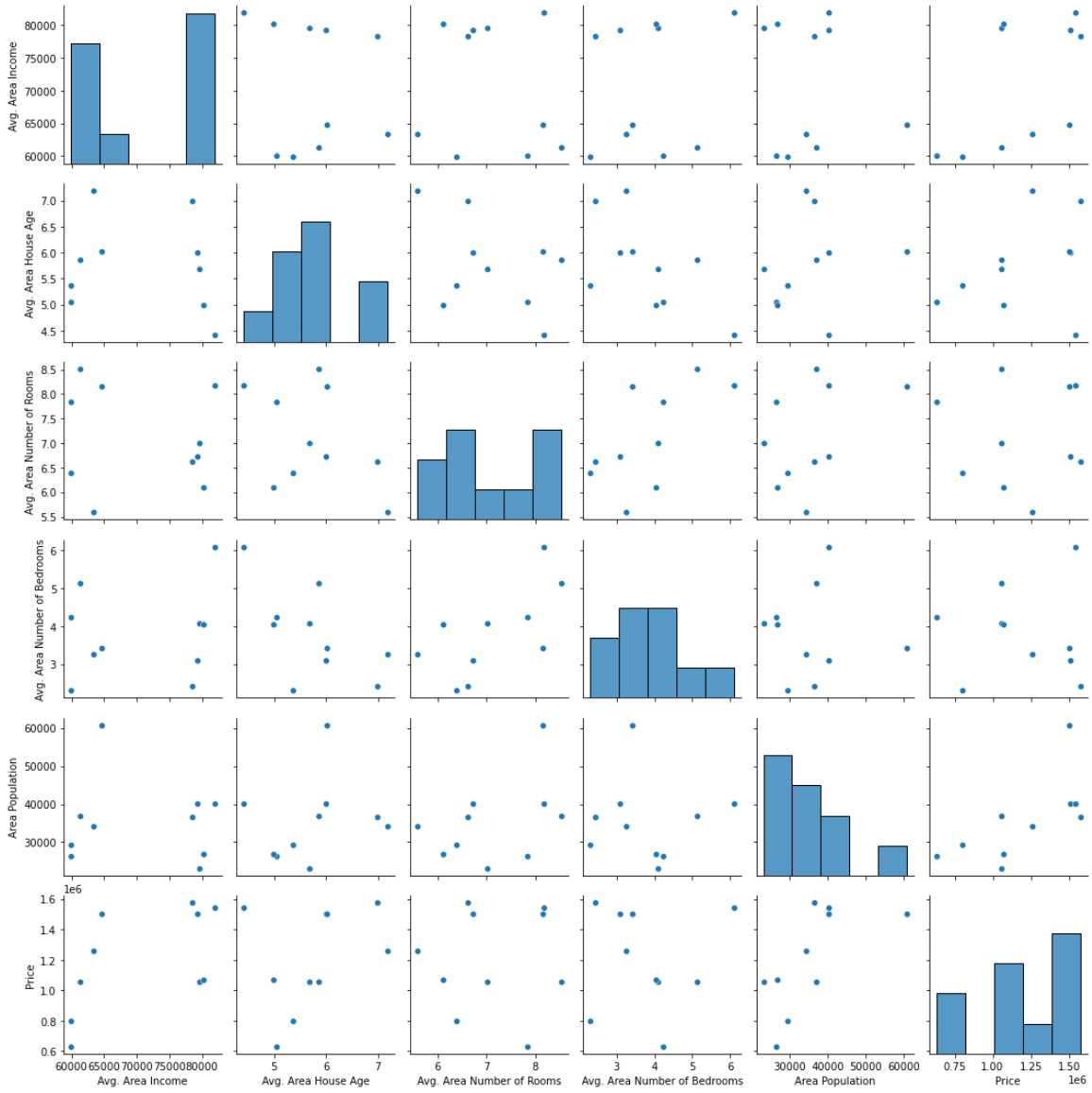
```
Index(['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Roo
ms',
       'Avg. Area Number of Bedrooms', 'Area Population', 'Price', 'Addres
s'],
      dtype='object')
```

In [142]:

```
sns.pairplot(a)
```

Out[142]:

```
<seaborn.axisgrid.PairGrid at 0x243f0b3eb20>
```

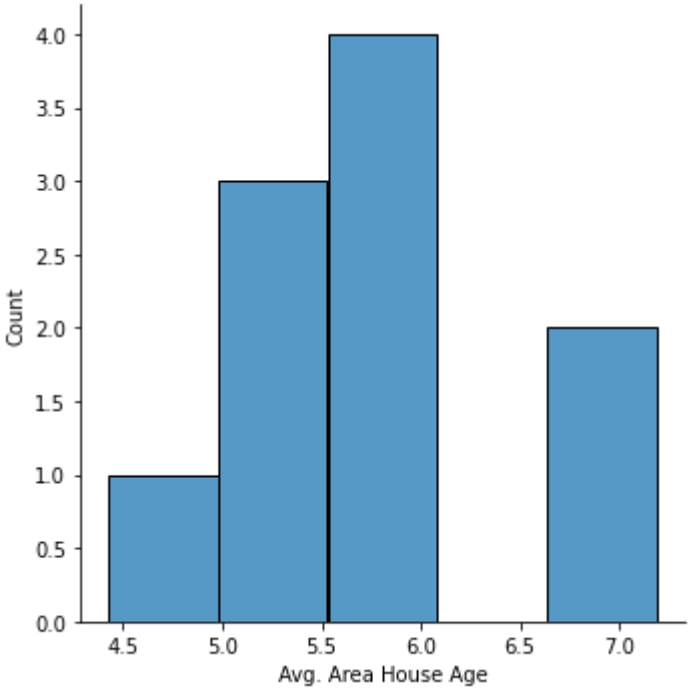


In [143]:

```
sns.displot(a["Avg. Area House Age"])
```

Out[143]:

```
<seaborn.axisgrid.FacetGrid at 0x243f2c21d60>
```



In [144]:

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

Out[144]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population
0	79545.458574	5.682861	7.009188	4.09	23086.800503
1	79248.642455	6.002900	6.730821	3.09	40173.072174
2	61287.067179	5.865890	8.512727	5.13	36882.159400
3	63345.240046	7.188236	5.586729	3.26	34310.242831
4	59982.197226	5.040555	7.839388	4.23	26354.109472
5	80175.754159	4.988408	6.104512	4.04	26748.428425
6	64698.463428	6.025336	8.147760	3.41	60828.249085
7	78394.339278	6.989780	6.620478	2.42	36516.358972
8	59927.660813	5.362126	6.393121	2.30	29387.396003
9	81885.927184	4.423672	8.167688	6.10	40149.965749

In [145]:

```
sns.heatmap(b.corr())
```

Out[145]:

<AxesSubplot:>

In [147]:

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

In [148]:

```
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 [149]:

```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[149]:

```
LinearRegression()
```

In [150]:

```
lr.intercept_
```

Out[150]:

```
5.1514348342607263e-14
```

In [151]:

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

Out[151]:

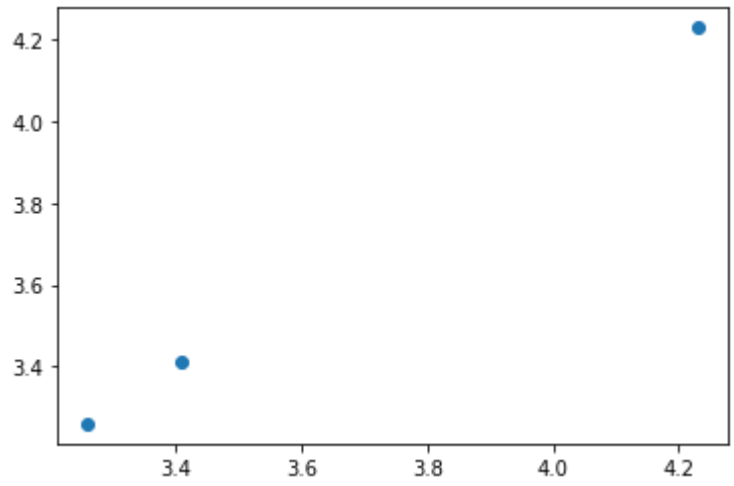
	Co-efficient
Avg. Area Income	9.344822e-20
Avg. Area House Age	3.142619e-16
Avg. Area Number of Rooms	-5.297074e-16
Avg. Area Number of Bedrooms	1.000000e+00
Area Population	-1.651153e-18

In [152]:

```
prediction = lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[152]:

```
<matplotlib.collections.PathCollection at 0x243f31f3610>
```



In [153]:

```
lr.score(x_test,y_test)
```

Out[153]:

```
1.0
```

In [154]:

```
lr.score(x_train,y_train)
```

Out[154]:

```
1.0
```

In [155]:

```
from sklearn.linear_model import Ridge,Lasso
```

In [156]:

```
rr=Ridge(alpha=10)
rr.fit(x_test,y_test)
```

Out[156]:

```
Ridge(alpha=10)
```

In [157]:

```
rr.score(x_test,y_test)
```

Out[157]:

```
0.9999999999930226
```

In [158]:

```
la=Lasso(alpha=10)
la.fit(x_test,y_test)
```

Out[158]:

```
Lasso(alpha=10)
```

In [159]:

```
la.score(x_test,y_test)
```

Out[159]:

```
0.9993347523195334
```

In [160]:

```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
```

Out[160]:

```
ElasticNet()
```

In [161]:

```
en.coef_
```

Out[161]:

```
array([ 1.33700254e-05, -0.00000000e+00,  0.00000000e+00,  5.12123685e-01,
        2.10633304e-05])
```

In [162]:

```
en.intercept_
```

Out[162]:

```
0.19862727830626303
```

In [163]:

```
prediction=en.predict(x_test)
prediction
```

Out[163]:

```
array([3.43776594, 4.09123465, 3.72197928])
```

In [164]:

```
en.score(x_test,y_test)
```

Out[164]:

```
-0.38238127985761383
```

EVALUATION METRICS

In [165]:

```
from sklearn import metrics
```

In [166]:

```
print("Mean Absolute Error:",metrics.mean_absolute_error(y_test,prediction))
```

```
Mean Absolute Error: 0.45567377105552875
```

In [167]:

```
print("Mean Squared Error",metrics.mean_squared_error(y_test,prediction))
```

```
Mean Squared Error 0.2512554775101208
```

In [168]:

```
print("Root Mean Squared Error",np.sqrt(metrics.mean_squared_error(y_test,prediction)))
```

```
Root Mean Squared Error 0.5012539052317905
```

In [ ]:

```
```