In [ ]:

```

```

In [87]:

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

In [89]:

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

Out[89]:

	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 C 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 8489\nAPO
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 [90]:

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

Out[90]:

	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\ 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
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 2
9	81885.927184	4.423672	8.167688	6.10	40149.965749	1.545155e+06	Unit 944 0958\nDPO AE

In [91]:

```
# 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 [92]:

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

Out[92]:

	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 [93]:

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

Out[93]:

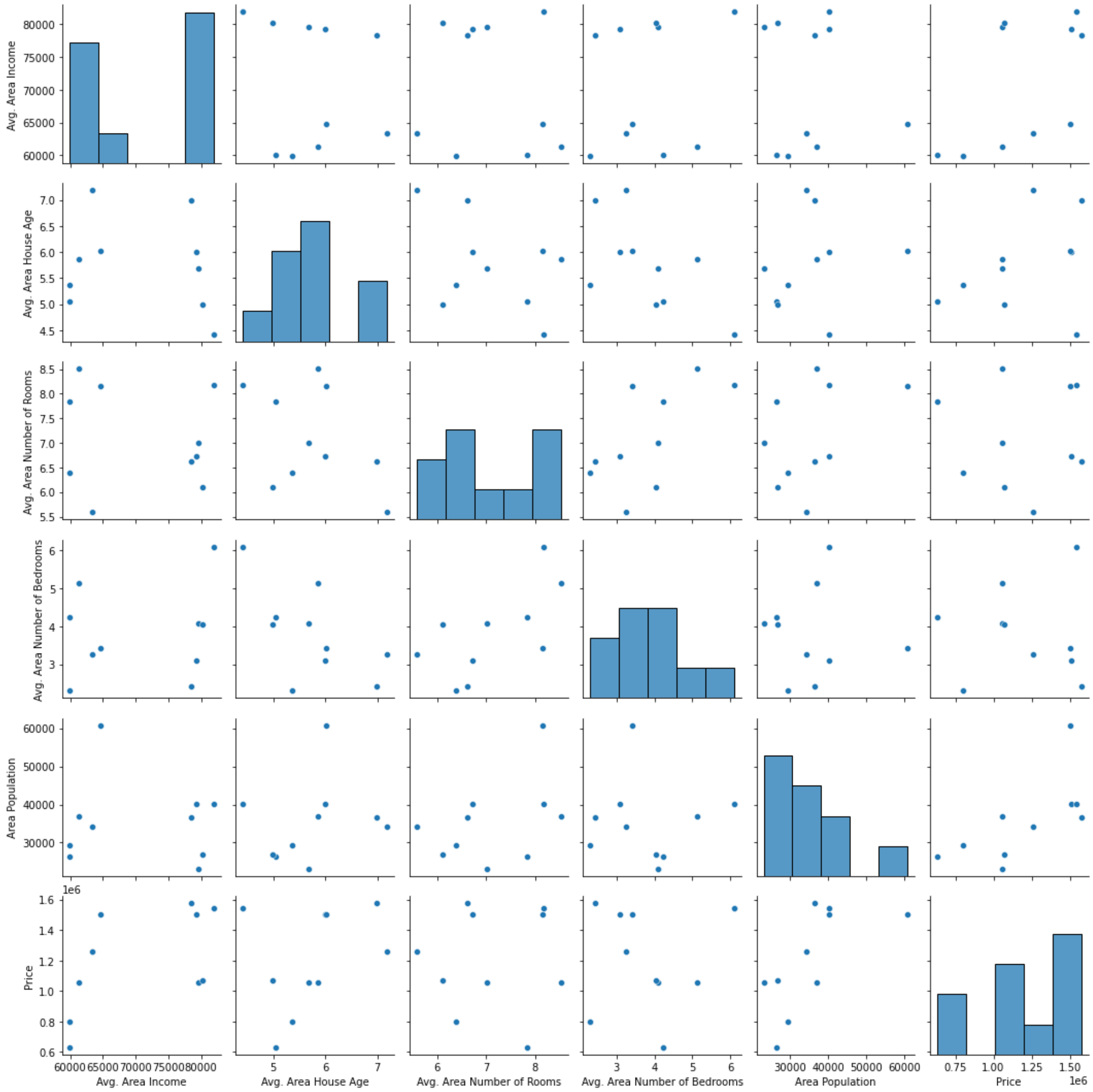
```
Index(['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Roo
ms',
       'Avg. Area Number of Bedrooms', 'Area Population', 'Price', 'Addres
s'],
      dtype='object')
```

In [94]:

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

Out[94]:

```
<seaborn.axisgrid.PairGrid at 0x20b062ec130>
```

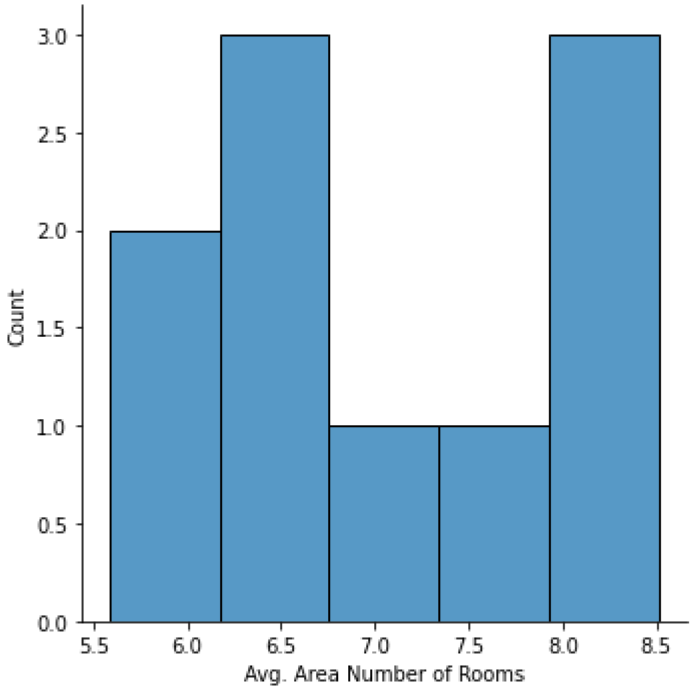


In [95]:

```
sns.displot(a["Avg. Area Number of Rooms"])
```

Out[95]:

```
<seaborn.axisgrid.FacetGrid at 0x20b06f4f280>
```



In [96]:

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

Out[96]:

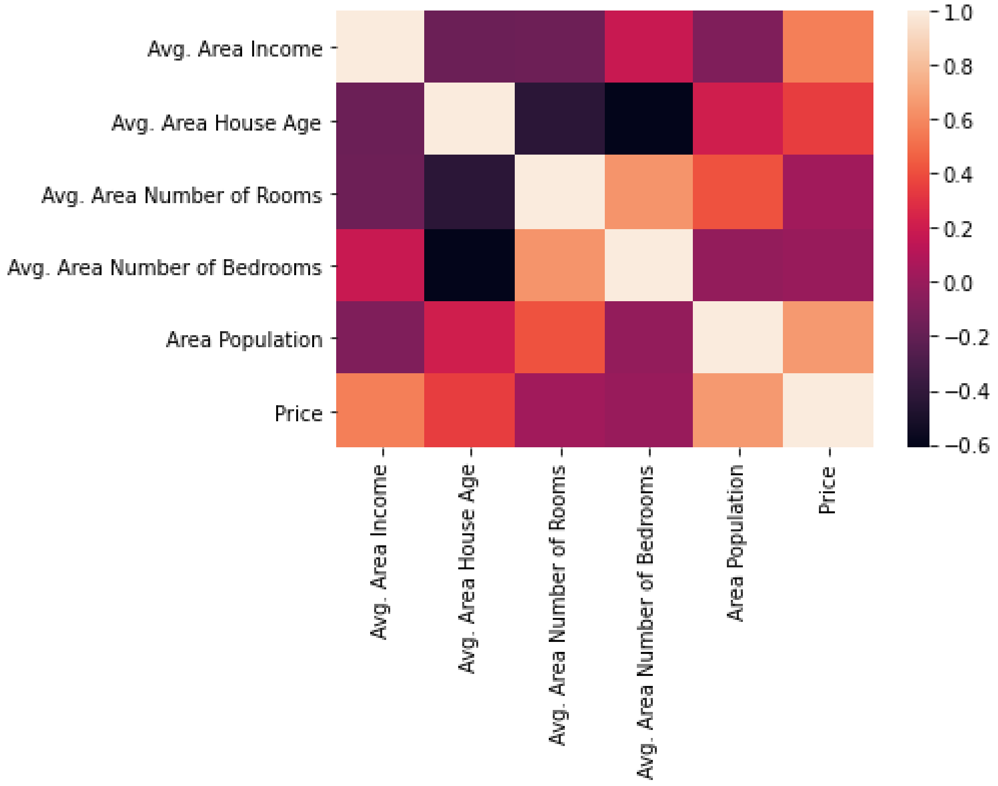
	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price
0	79545.458574	5.682861	7.009188	4.09	23086.800503	1.059034e+06
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06
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
5	80175.754159	4.988408	6.104512	4.04	26748.428425	1.068138e+06
6	64698.463428	6.025336	8.147760	3.41	60828.249085	1.502056e+06
7	78394.339278	6.989780	6.620478	2.42	36516.358972	1.573937e+06
8	59927.660813	5.362126	6.393121	2.30	29387.396003	7.988695e+05
9	81885.927184	4.423672	8.167688	6.10	40149.965749	1.545155e+06

In [97]:

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

Out[97]:

```
<AxesSubplot:>
```



In [99]:

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

In [100]:

```
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 [101]:

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

Out[101]:

```
LinearRegression()
```

In [102]:

```
lr.intercept_
```

Out[102]:

```
-28.373102728365442
```

In [103]:

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

Out[103]:

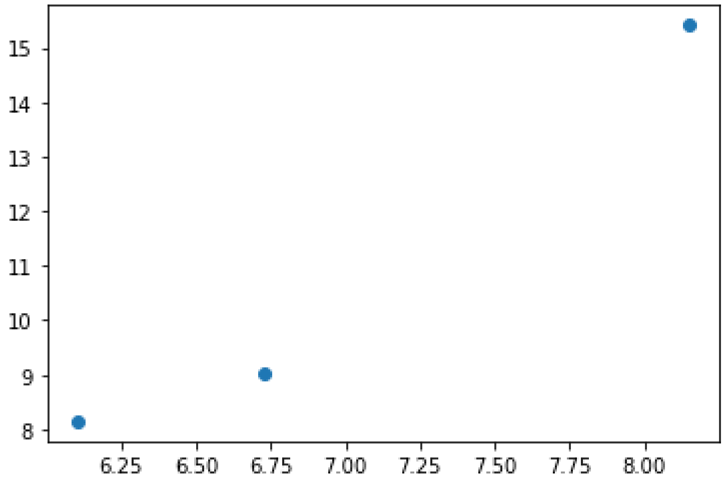
	Co-efficient
Avg. Area Income	0.000385
Avg. Area House Age	1.465367
Avg. Area Number of Bedrooms	0.379008
Area Population	0.000570
Price	-0.000017

In [104]:

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

Out[104]:

```
<matplotlib.collections.PathCollection at 0x20b08710610>
```



In [105]:

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

Out[105]:

```
-27.390562146147364
```

In [106]:

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

Out[106]:

```
0.9999989995709274
```

In [107]:

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

In [108]:

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

Out[108]:

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

In [109]:

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

Out[109]:

```
0.9999999999999996
```

In [110]:

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

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_coordinat
e_descent.py:530: ConvergenceWarning: Objective did not converge. You migh
t want to increase the number of iterations. Duality gap: 0.00043622210813
88731, tolerance: 0.0002191612078186376
  model = cd_fast.enet_coordinate_descent(
```

Out[110]:

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

In [111]:

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

Out[111]:

```
0.9999976321927122
```

In [ ]:

```

```