

```
In [35]: # import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
In [40]: x=pd.read_csv(r"C:\Users\user\Downloads\10_USA_Housing - 10_USA_Housing.csv")
```

Out[40]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	Addr
0	79545.45857	5.682861	7.009188	4.09	23086.80050	1.059034e+06	208 Michael Ferry , 674\nLaurabury, 370
1	79248.64245	6.002900	6.730821	3.09	40173.07217	1.505891e+06	188 Johnson Vi Suite 079\nL Kathleen, C
2	61287.06718	5.865890	8.512727	5.13	36882.15940	1.058988e+06	9127 Elizak Stravenue\nDanielto WI 0648
3	63345.24005	7.188236	5.586729	3.26	34310.24283	1.260617e+06	USS Barnett\nFPO 44
4	59982.19723	5.040555	7.839388	4.23	26354.10947	6.309435e+05	USNS Raymond\nF AE 09
...	...	...	...	...	...	...	...
4995	60567.94414	7.830362	6.137356	3.46	22837.36103	1.060194e+06	USNS Williams\nF AP 30153-7
4996	78491.27543	6.999135	6.576763	4.02	25616.11549	1.482618e+06	PSC 9258, 8489\nAPO 42991-3
4997	63390.68689	7.250591	4.805081	2.13	33266.14549	1.030730e+06	4215 Tracy Gar Suite 076\nJoshual VA (
4998	68001.33124	5.534388	7.130144	5.44	42625.62016	1.198657e+06	USS Wallace\nFPO 73
4999	65510.58180	5.992305	6.792336	4.07	46501.28380	1.298950e+06	37778 George Rid Apt. 509\nEast H NV

5000 rows × 7 columns

In [41]: `x=x.head(10)`

Out[41]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	Address
0	79545.45857	5.682861	7.009188	4.09	23086.80050	1.059034e+06	208 Michael Ferry Apt. 674\nLaurabury, NE 3701...
1	79248.64245	6.002900	6.730821	3.09	40173.07217	1.505891e+06	188 Johnson Views Suite 079\nLake Kathleen, CA...
2	61287.06718	5.865890	8.512727	5.13	36882.15940	1.058988e+06	9127 Elizabeth Stravenue\nDanielstown, WI 06482...
3	63345.24005	7.188236	5.586729	3.26	34310.24283	1.260617e+06	USS Barnett\nFPO AP 44820
4	59982.19723	5.040555	7.839388	4.23	26354.10947	6.309435e+05	USNS Raymond\nFPO AE 09386
5	80175.75416	4.988408	6.104512	4.04	26748.42842	1.068138e+06	06039 Jennifer Islands Apt. 443\nTracyport, KS...
6	64698.46343	6.025336	8.147760	3.41	60828.24909	1.502056e+06	4759 Daniel Shoals Suite 442\nNguyenburgh, CO ...
7	78394.33928	6.989780	6.620478	2.42	36516.35897	1.573937e+06	972 Joyce Viaduct\nLake William, TN 17778-6483
8	59927.66081	5.362126	6.393121	2.30	29387.39600	7.988695e+05	USS Gilbert\nFPO AA 20957
9	81885.92718	4.423672	8.167688	6.10	40149.96575	1.545155e+06	Unit 9446 Box 0958\nDPO AE 97025

In [42]:

```
<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 [43]:

```
Out[43]: 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 [47]: `d=x[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms']]`

Out[47]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms
0	79545.45857	5.682861	7.009188
1	79248.64245	6.002900	6.730821
2	61287.06718	5.865890	8.512727
3	63345.24005	7.188236	5.586729
4	59982.19723	5.040555	7.839388
5	80175.75416	4.988408	6.104512
6	64698.46343	6.025336	8.147760
7	78394.33928	6.989780	6.620478
8	59927.66081	5.362126	6.393121
9	81885.92718	4.423672	8.167688

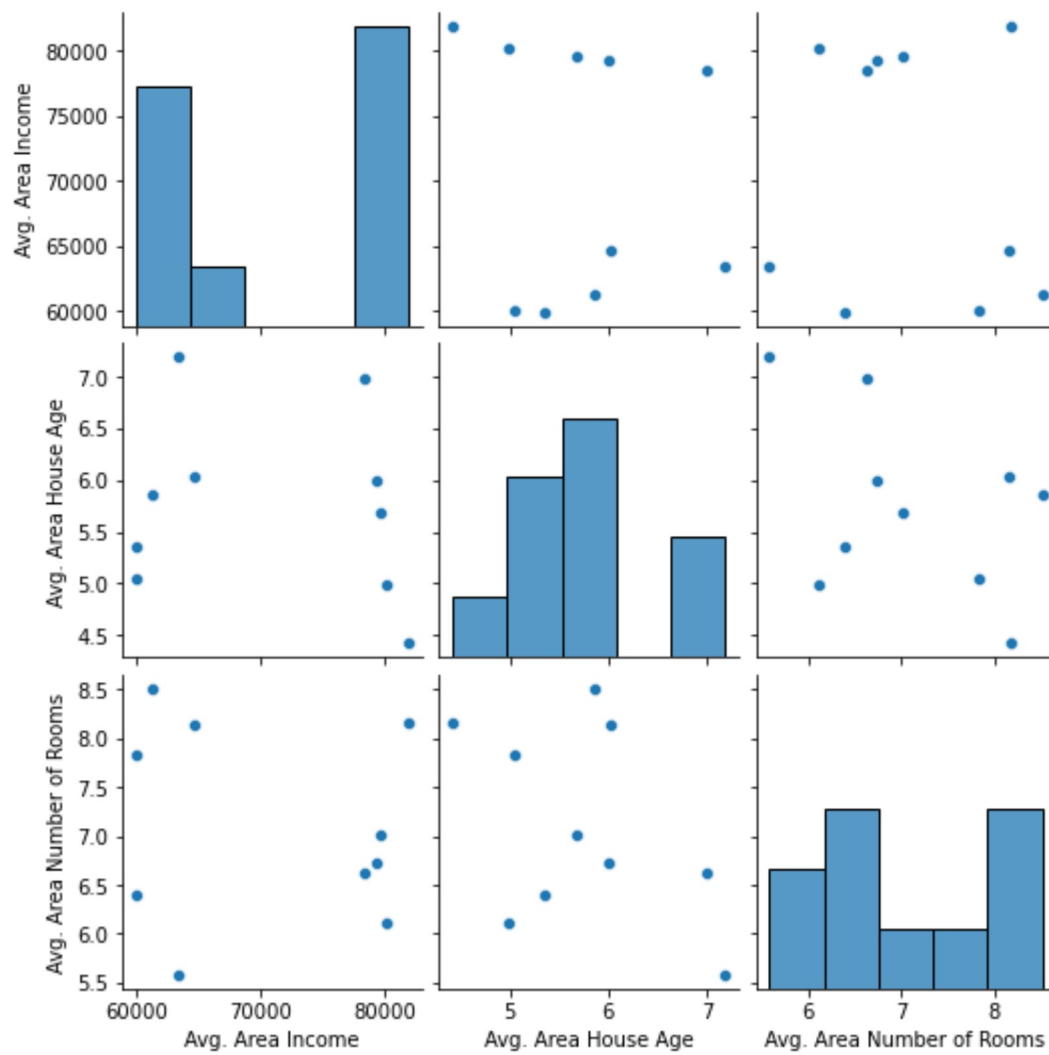
In [48]:

Out[48]:

	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	35443.678260	1.200363e+06
std	9631.232524	0.866393	0.996334	1.177908	10754.822722	3.315477e+05
min	59927.660810	4.423672	5.586729	2.300000	23086.800500	6.309435e+05
25%	61801.610398	5.120947	6.449960	3.132500	27408.170315	1.058999e+06
50%	71546.401355	5.774376	6.870005	3.725000	35413.300900	1.164377e+06
75%	79471.254540	6.019727	8.070667	4.195000	39333.014163	1.504932e+06
max	81885.927180	7.188236	8.512727	6.100000	60828.249090	1.573937e+06

In [49]:

Out[49]: <seaborn.axisgrid.PairGrid at 0x1909a795250>

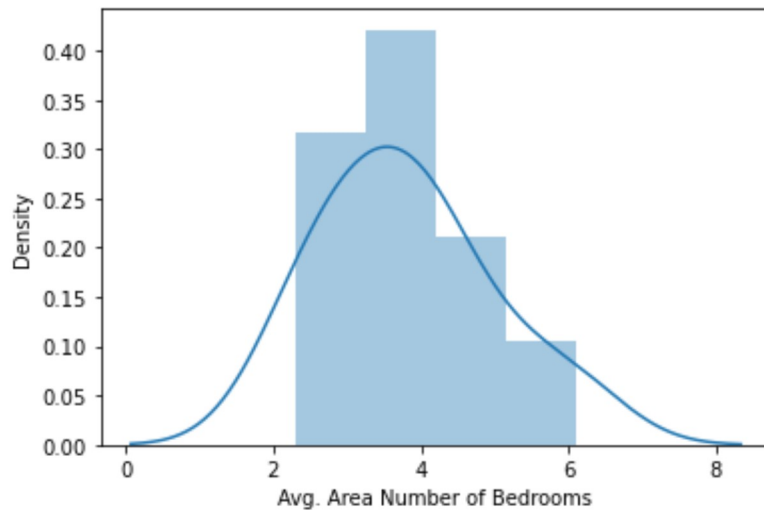


In [50]:

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `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 flexibility) or `histplot` (an axes-level function for histograms).
```

```
warnings.warn(msg, FutureWarning)
```

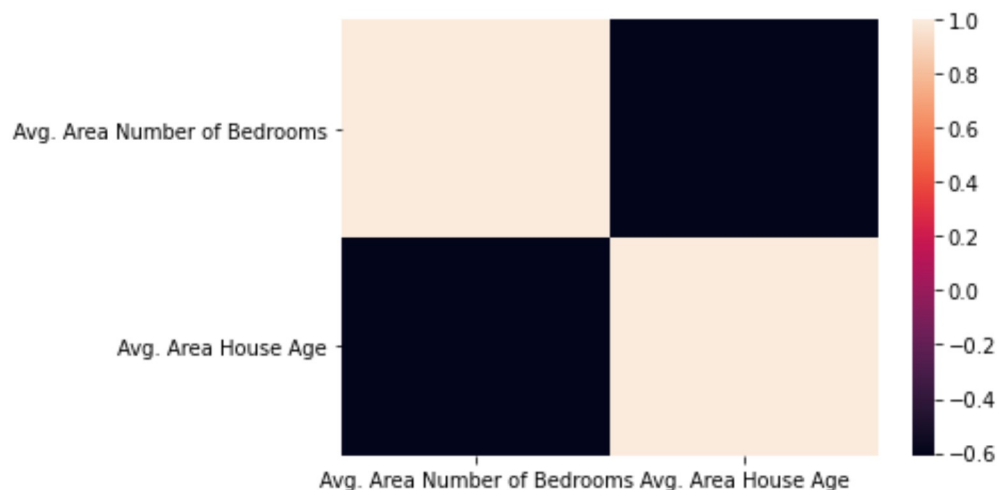
Out[50]: &lt;AxesSubplot:xlabel='Avg. Area Number of Bedrooms', ylabel='Density'&gt;



In [51]:

In [52]:

Out[52]: &lt;AxesSubplot:&gt;

In [54]: `x=x1[['Avg. Area House Age']]`

In [55]: *# to split my dataset into training and test data*

```
from sklearn.model_selection import train_test_split
```

In [56]: **from** sklearn.linear\_model **import** LinearRegression

```
lr=LinearRegression()
```

Out[56]: LinearRegression()

In [57]:

```
8.881784197001252e-16
```

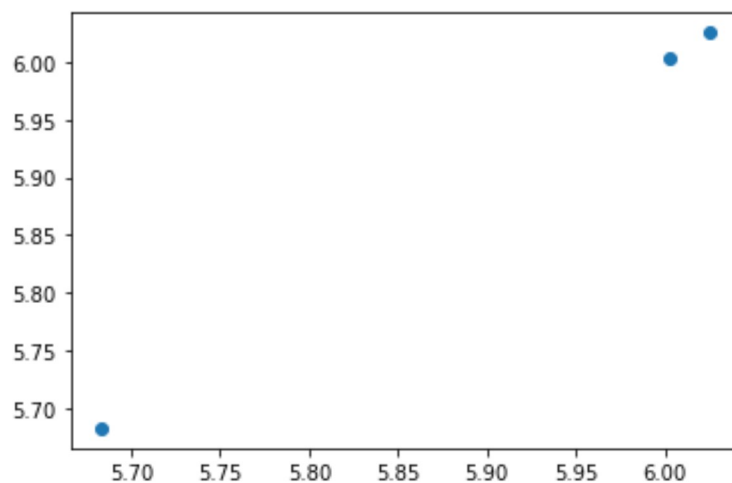
In [58]: `coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])`

Out[58]:

Co-efficient	
Avg. Area House Age	1.0

In [59]: `prediction=lr.predict(x_test)`

Out[59]: <matplotlib.collections.PathCollection at 0x1909c048400>



In [60]:

Out[60]: 1.0

In [61]:

Out[61]: 1.0

In [62]:

```
In [63]: rr=Ridge(alpha=10)
         rr.fit(x_train,y_train)
```

```
Out[63]: -0.01570413285896133
```

```
In [64]: la=Lasso(alpha=10)
```

```
Out[64]: Lasso(alpha=10)
```

```
In [65]:
```

```
Out[65]: -1.7955229079070212
```

```
In [66]: from sklearn.linear_model import ElasticNet
         en=ElasticNet()
```

```
Out[66]: ElasticNet()
```

```
In [67]:
```

```
Out[67]: array([0.30624783])
```

```
In [68]:
```

```
Out[68]: array([5.69065474, 5.79553684, 5.78866583])
```

```
In [69]:
```

```
Out[69]: 3.9502907986717686
```

```
In [70]:
```

```
Out[70]: -0.34546301933684553
```

```
In [71]:
```

```
In [72]:
```

```
Mean Absolute Error 0.0
```

```
In [73]:
```

```
Mean Squared Error 0.0
```

```
In [74]:
```

```
Root Mean Squared Error 0.0
```

```
In [ ]:
```

