In [2]:
```
#linear regression
```

In [1]:
```
import numpy as np
```

In [2]:
```
import pandas as pd
```

In [3]:
```
import matplotlib.pyplot as plt
```

In [6]:
```
import seaborn as sns
%matplotlib inline
```

In [ ]:

In [8]:
```
cd
```

```
C:\Users\758449
```

In [8]:
```
cd \Users\758449\Downloads
```

```
C:\Users\758449\Downloads
```

In [14]:
```
data = pd.read_csv('real_estate_price_size.csv')
```

In [15]:
```
data
```

Out[15]:

	price	size
0	234314.144	643.09
1	228581.528	656.22
2	281626.336	487.29
3	401255.608	1504.75
4	458674.256	1275.46
...	...	...
95	252460.400	549.80
96	310522.592	1037.44
97	383635.568	1504.75
98	225145.248	648.29
99	274922.856	705.29

100 rows × 2 columns

In [16]:
```
data.head()
```

Out[16]:

	price	size
0	234314.144	643.09
1	228581.528	656.22
2	281626.336	487.29
3	401255.608	1504.75
4	458674.256	1275.46

In [18]:
```
data.dtypes
```

Out[18]:
```
price    float64
size     float64
dtype: object
```

In [22]:
```
data.describe()
```

Out[22]:

	price	size
count	100.000000	100.000000
mean	292289.470160	853.024200
std	77051.727525	297.941951
min	154282.128000	479.750000
25%	234280.148000	643.330000
50%	280590.716000	696.405000
75%	335723.696000	1029.322500
max	500681.128000	1842.510000

In [23]:
```
data.isnull()
```

Out[23]:

	price	size
0	False	False
1	False	False
2	False	False
3	False	False
4	False	False
...	...	...
95	False	False
96	False	False
97	False	False
98	False	False
99	False	False

100 rows × 2 columns

In [37]:
```
x = 'size'
```

In [38]:
```
y = 'price'
```

In [39]:
```
data
```

Out[39]:

	price	size
0	234314.144	643.09
1	228581.528	656.22
2	281626.336	487.29
3	401255.608	1504.75
4	458674.256	1275.46
...	...	...
95	252460.400	549.80
96	310522.592	1037.44
97	383635.568	1504.75
98	225145.248	648.29
99	274922.856	705.29

100 rows × 2 columns

In [42]:
```
X = data['size'].values
```

In [45]:
```
Y= data['price'].values
```

In [31]:
```
from sklearn.model_selection import train_test_split
```

In [123]:
```
X_train , X_test,Y_train, Y_test= train_test_split(X, Y, test_size=0.44, random_state=42)
```

In [ ]:

In [124]:
```
X_train , X_test,Y_train, Y_test
```

Out[124]:
```
(array([[1334.1 ],
        [1027.76],
        [1122.34],
        [1103.3 ],
        [1205.62],
        [1009.25],
        [ 525.81],
        [1504.75],
        [ 623.94],
        [ 685.48],
        [ 682.26],
        [ 682.26],
        [ 570.89],
        [ 685.48],
        [ 823.21],
        [1236.93],
        [1032.06],
        [ 705.29],
        [ 647.5 ],
        [1334.1 ],
        [ 647.5 ],
        [ 549.8 ],
        [ 633.19],
        [ 685.48],
        [ 698.5 ],
        [1504.75],
        [1188.62],
        [ 682.26],
        [ 827.84],
        [ 633.19],
        [ 648.29],
        [1283.85],
        [ 685.48],
        [ 597.9 ],
        [ 698.29],
        [ 569.17],
        [1021.95],
        [ 685.48],
        [ 570.25],
        [1496.36],
        [ 656.22],
        [1021.95],
        [ 549.69],
        [ 487.29],
        [ 518.38],
        [1028.41],
        [1071.55],
        [ 685.48],
        [ 479.75],
        [ 681.07],
        [ 620.71],
        [ 828.16],
        [ 643.41],
        [1300.96],
        [ 694.52],
        [1021.95]]),
 array([[ 643.09],
        [ 727.88],
        [1021.95],
        [ 698.29],
        [1060.36],
        [ 698.29],
        [1207.45],
        [ 681.07],
        [1060.36],
        [ 643.09],
        [ 681.07],
        [1010.33],
        [ 549.8 ],
        [ 525.81],
        [ 694.52],
        [1275.46],
        [1183.46],
        [1334.1 ],
        [ 694.52],
        [ 681.07],
        [1508.84],
        [ 601.66],
        [ 570.89],
        [ 823.21],
        [1496.36],
        [1379.72],
        [1021.95],
        [1037.44],
        [ 694.52],
        [ 656.22],
        [1842.51],
        [ 698.29],
        [1009.25],
        [ 681.07],
        [ 698.29],
        [ 575.19],
        [1009.25],
        [ 827.09],
        [ 622.97],
        [ 690.54],
        [ 617.05],
        [ 857.54],
        [ 620.82],
        [ 549.8 ]]),
 array([[406852.304],
        [299416.976],
        [408637.816],
        [362519.72 ],
        [359674.44 ],
        [418753.008],
        [168047.264],
        [401255.608],
        [234178.16 ],
        [292965.216],
        [331101.344],
        [251560.04 ],
        [265129.064],
        [302393.384],
        [269225.92 ],
        [354512.112],
        [320345.52 ],
        [274922.856],
        [255629.16 ],
        [414682.648],
        [225656.12 ],
        [271726.752],
        [204302.976],
        [294582.944],
        [263311.696],
        [383635.568],
        [342988.456],
        [217468.224],
        [330677.128],
        [257828.416],
        [225145.248],
        [395242.096],
        [286161.6  ],
        [207742.248],
        [262477.856],
        [251332.592],
        [334938.872],
        [303597.216],
        [233493.208],
        [368988.432],
        [228581.528],
        [258637.008],
        [171795.24 ],
        [281626.336],
        [183459.488],
        [327252.112],
        [338078.168],
        [228313.024],
        [154282.128],
        [190909.056],
        [268125.08 ],
        [251188.824],
        [181587.576],
        [444192.008],
        [298170.88 ],
        [393069.76 ]]),
 array([[282683.544],
        [269523.056],
        [276875.632],
        [300061.48 ],
        [293044.496],
        [245747.2  ],
        [412569.472],
        [180307.216],
        [279555.096],
        [234314.144],
        [225451.984],
        [301635.728],
        [211724.096],
        [191486.896],
        [251140.656],
        [458674.256],
        [382120.152],
        [365863.936],
        [215472.104],
        [225452.32 ],
        [500681.128],
        [211904.536],
        [271793.312],
        [287350.   ],
        [454512.76 ],
        [440201.616],
        [310045.712],
        [310522.592],
        [218630.608],
        [298926.496],
        [494778.992],
        [201778.048],
        [376253.808],
        [297760.44 ],
        [266684.248],
        [245050.28 ],
        [355251.2  ],
        [304587.272],
        [302000.92 ],
        [248337.6  ],
        [262423.504],
        [285223.176],
        [175716.48 ],
        [252460.4  ]]))
```

In [ ]:

In [ ]:

In [ ]:

In [53]:
```
from sklearn.linear_model import LinearRegression
```

In [81]:
```
reg = LinearRegression()
```

In [125]:
```
X_train.reshape(-1,1)
```

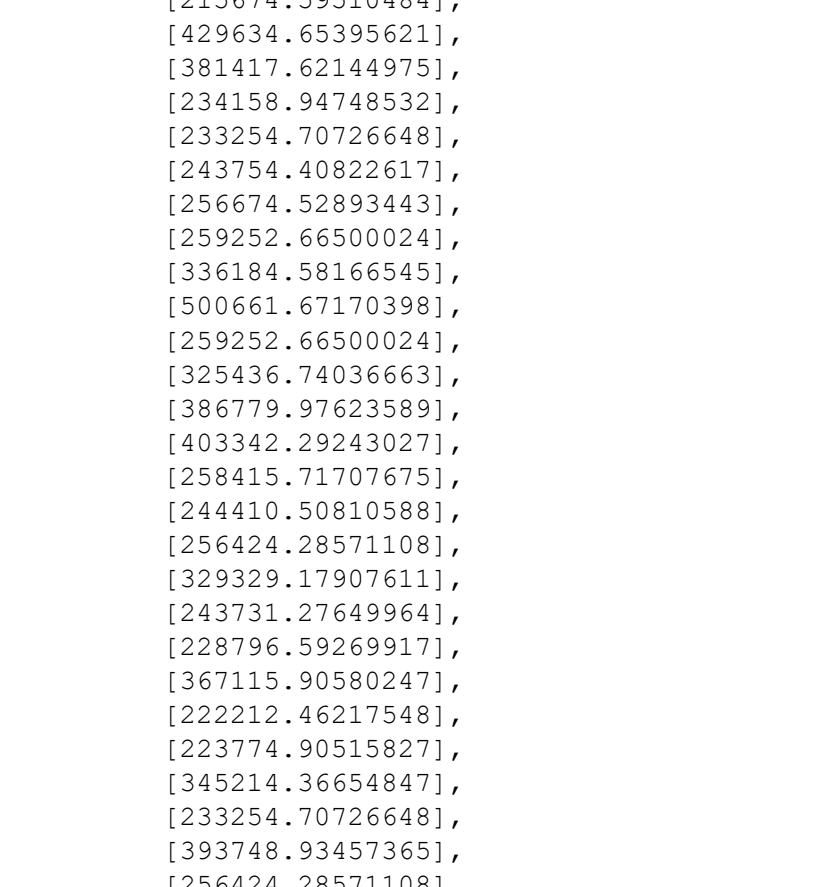
Out[125]:
```
array([[1334.1 ],
       [1027.76],
       [1122.34],
       [1103.3 ],
       [1205.62],
       [1009.25],
       [ 525.81],
       [1504.75],
       [ 623.94],
       [ 685.48],
       [ 682.26],
       [ 682.26],
       [ 570.89],
       [ 685.48],
       [ 823.21],
       [1236.93],
       [1032.06],
       [ 705.29],
       [ 647.5 ],
       [1334.1 ],
       [ 647.5 ],
       [ 549.8 ],
       [ 633.19],
       [ 685.48],
       [ 698.5 ],
       [1504.75],
       [1188.62],
       [ 682.26],
       [ 827.84],
       [ 633.19],
       [ 648.29],
       [1283.85],
       [ 685.48],
       [ 597.9 ],
       [ 698.29],
       [ 569.17],
       [1021.95],
       [ 685.48],
       [ 570.25],
       [1496.36],
       [ 656.22],
       [1021.95],
       [ 549.69],
       [ 487.29],
       [ 518.38],
       [1028.41],
       [1071.55],
       [ 685.48],
       [ 479.75],
       [ 681.07],
       [ 620.71],
       [ 828.16],
       [ 643.41],
       [1300.96],
       [ 694.52],
       [1021.95]])
```

In [ ]:

In [ ]:

In [110]:
```
plt.scatter(X_train,Y_train)
```

Out[110]:
```
<matplotlib.collections.PathCollection at 0x1cb96877070>
```



In [128]:
```
reg.fit(X_train,Y_train)
```

Out[128]:
```
LinearRegression()
```

In [132]:
```
plt.show()
```

In [134]:
```
Y_pred = reg.predict(X)
```

In [ ]:

In [136]:
```
Y_pred
```

Out[136]:
```
array([[248437.53140607],
       [251198.61839987],
       [215674.59510484],
       [429634.65395621],
       [381417.62144975],
       [234158.94748532],
       [233254.70726648],
       [243754.40822617],
       [256674.52893443],
       [259252.66500024],
       [336184.58166545],
       [500661.67170398],
       [259252.66500024],
       [325436.74036663],
       [386779.97623589],
       [403342.29243027],
       [258415.71707675],
       [244410.50810588],
       [256424.28571108],
       [329329.17907611],
       [243731.27649964],
       [228796.59269917],
       [367115.90580247],
       [222212.46217548],
       [223774.90515827],
       [345214.36654847],
       [233254.70726648],
       [393748.93457365],
       [256424.28571108],
       [427870.33408736],
       [325663.85186346],
       [256424.28571108],
       [238934.59757133],
       [223774.90515827],
       [293533.88371542],
       [244206.52833559],
       [286314.68215431],
       [233120.12267576],
       [257351.65765645],
       [260045.4523549 ],
       [328107.40333856],
       [256674.52893443],
       [286314.68215431],
       [393748.93457365],
       [336184.58166545],
       [260045.4523549 ],
       [246355.67601851],
       [260045.4523549 ],
       [246355.67601851],
       [242961.62087151],
       [249364.90335144],
       [328107.40333856],
       [328107.40333856],
       [266267.88679105],
       [249364.90335144],
       [430494.73360623],
       [330233.41929495],
       [383181.9413186 ],
       [287288.31755274],
       [232893.01117894],
       [287355.60984809],
       [260089.61292372],
       [366731.0779884 ],
       [328107.40333856],
       [257351.65765645],
       [287130.6012355 ],
       [325436.74036663],
       [228819.7244257 ],
       [257351.65765645],
       [427870.33408736],
       [328107.40333856],
       [248504.82370142],
       [251198.61839987],
       [228819.7244257 ],
       [257351.65765645],
       [257351.65765645],
       [362071.08653503],
       [393748.93457365],
       [256674.52893443],
       [363156.17479764],
       [256424.28571108],
       [349218.25812212],
       [256424.28571108],
       [248437.53140607],
       [257351.65765645],
       [325436.74036663],
       [214089.02039553],
       [329465.86655105],
       [239725.28204176],
       [373315.20851209],
       [259252.66500024],
       [338537.70911866],
       [259252.66500024],
       [260045.4523549 ],
       [260045.4523549 ],
       [228819.7244257 ],
       [331364.77101061],
       [429634.65395621],
       [249531.03120559],
       [261517.4713158 ]])
```

In [138]:
```
reg.intercept_
```

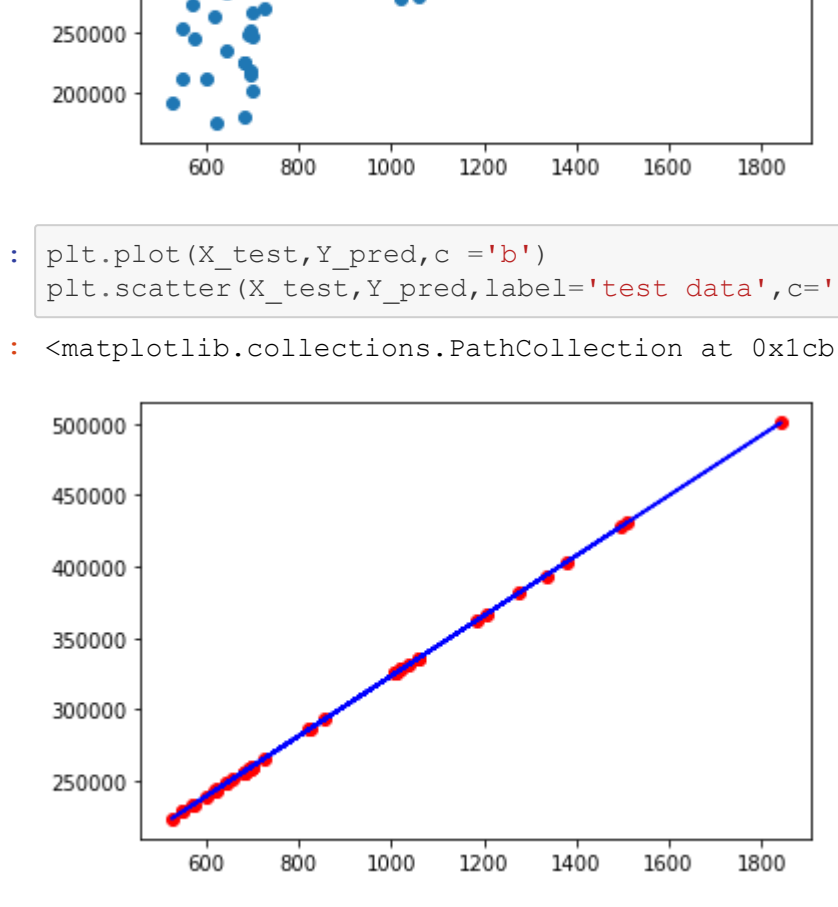
Out[138]:
```
array([113203.14946798])
```

In [142]:
```
reg.coef_
```

Out[142]:
```
array([[210.28842299]])
```

In [150]:
```
plt.scatter(X_train,Y_train, cmap='red')
```

Out[150]:
```
<matplotlib.collections.PathCollection at 0x1cb969d47c0>
```



In [148]:
```
r2_score =reg.score(X_train,Y_train)
```

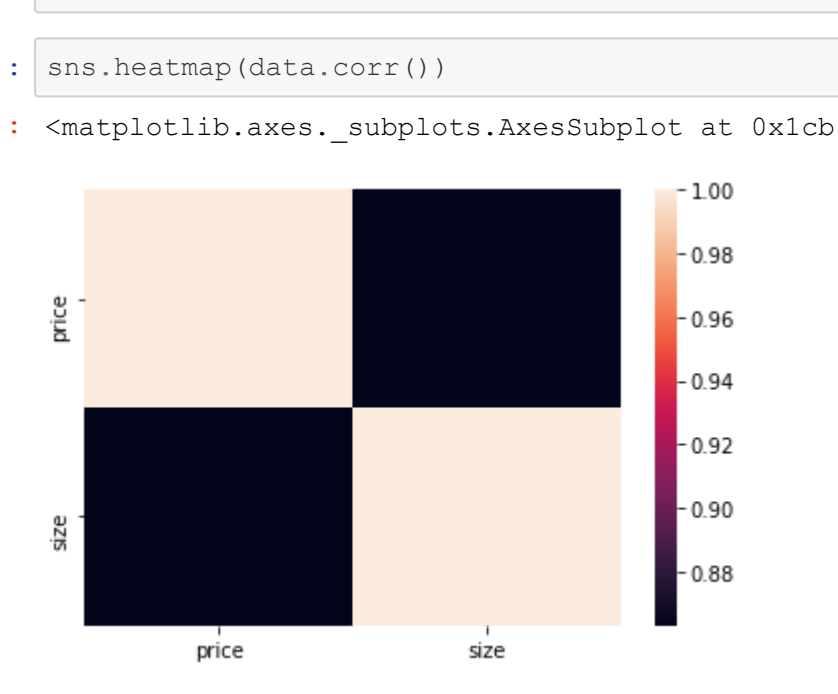
In [152]:
```
print(r2_score)
```

```
0.7011041187028517
```

In [156]:
```
Y_pred=reg.predict(X_test)
```

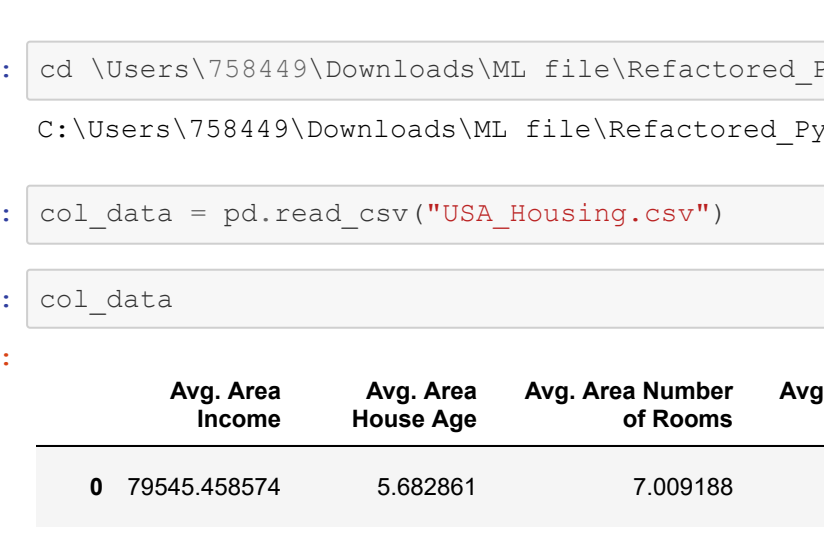
In [158]:
```
plt.scatter(X_test,Y_test)
```

Out[158]:
```
<matplotlib.collections.PathCollection at 0x1cb96a98340>
```



In [168]:
```
plt.plot(X_test,Y_pred,c ='b')
plt.scatter(X_test,Y_pred,label='test data',c='r')
```

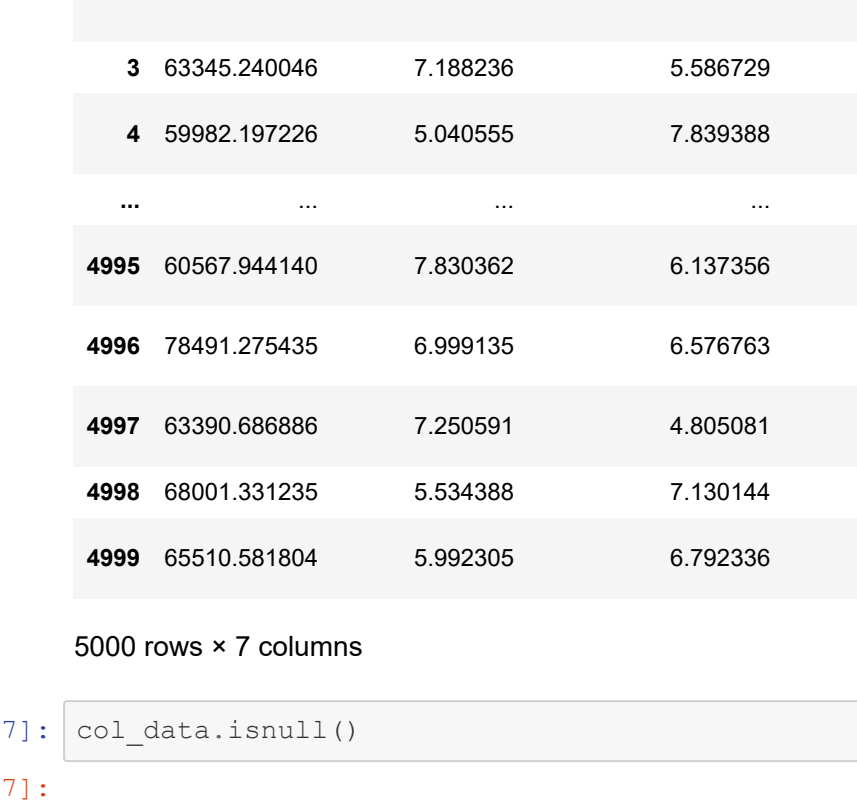
Out[168]:
```
<matplotlib.collections.PathCollection at 0x1cb97c8b910>
```



In [174]:
```
from sklearn.metrics import mean_squared_error
```

In [181]:
```
sns.heatmap(data.corr())
```

Out[181]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1cb97e789a0>
```



In [10]:
```
cd \Users\758449\Downloads\ML file\Refactored_Py_DS_ML_Bootcamp-master\11-Linear-Regression
```

```
C:\Users\758449\Downloads\ML file\Refactored_Py_DS_ML_Bootcamp-master\11-Linear-Regression
```

In [11]:
```
col_data = pd.read_csv("USA_Housing.csv")
```

In [13]:
```
col_data
```

Out[13]:

	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
...	...	...	...	...	...	...	...
4995	60567.944140	7.830362	6.137356	3.46	22837.361035	1.060194e+06	USNS Williams\nFPO AP 30153-7653
4996	78491.275435	6.999135	6.576763	4.02	25616.115489	1.482618e+06	PSC 9258, Box 8489\nAPO AA 42991-3352
4997	63390.686886	7.250591	4.805081	2.13	33266.145490	1.030730e+06	4215 Tracy Garden Suite 076\nJoshualand, VA 01...
4998	68001.331235	5.534388	7.130144	5.44	42625.620156	1.198657e+06	USS Wallace\nFPO AE 73316
4999	65510.581804	5.992305	6.792336	4.07	46501.283803	1.298950e+06	37778 George Ridges Apt. 509\nEast Holly, NV 2...

5000 rows × 7 columns

In [17]:
```
col_data.isnull()
```

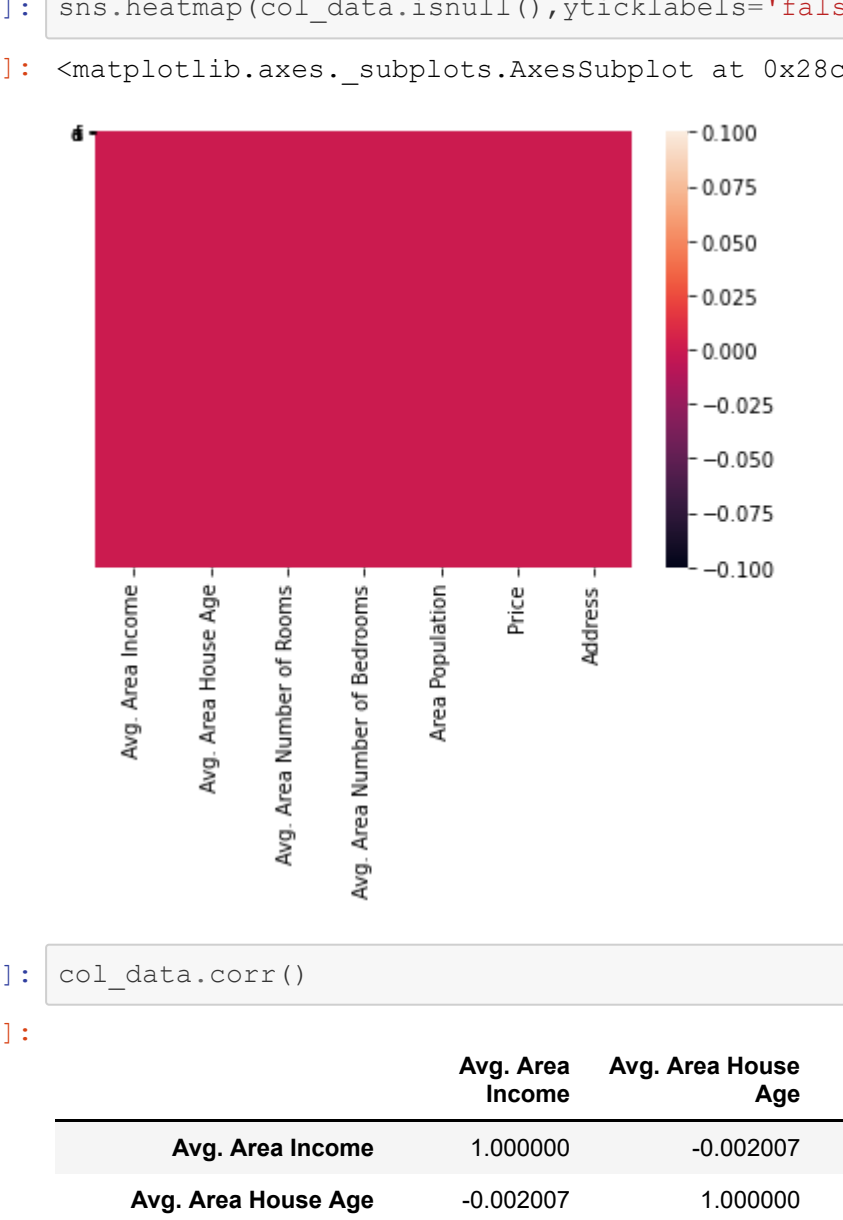
Out[17]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	Address
0	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...
4995	False	False	False	False	False	False	False
4996	False	False	False	False	False	False	False
4997	False	False	False	False	False	False	False
4998	False	False	False	False	False	False	False
4999	False	False	False	False	False	False	False

5000 rows × 7 columns

In [26]:
```
sns.heatmap(col_data.isnull(),yticklabels='false')
```

Out[26]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x28c262a6760>
```



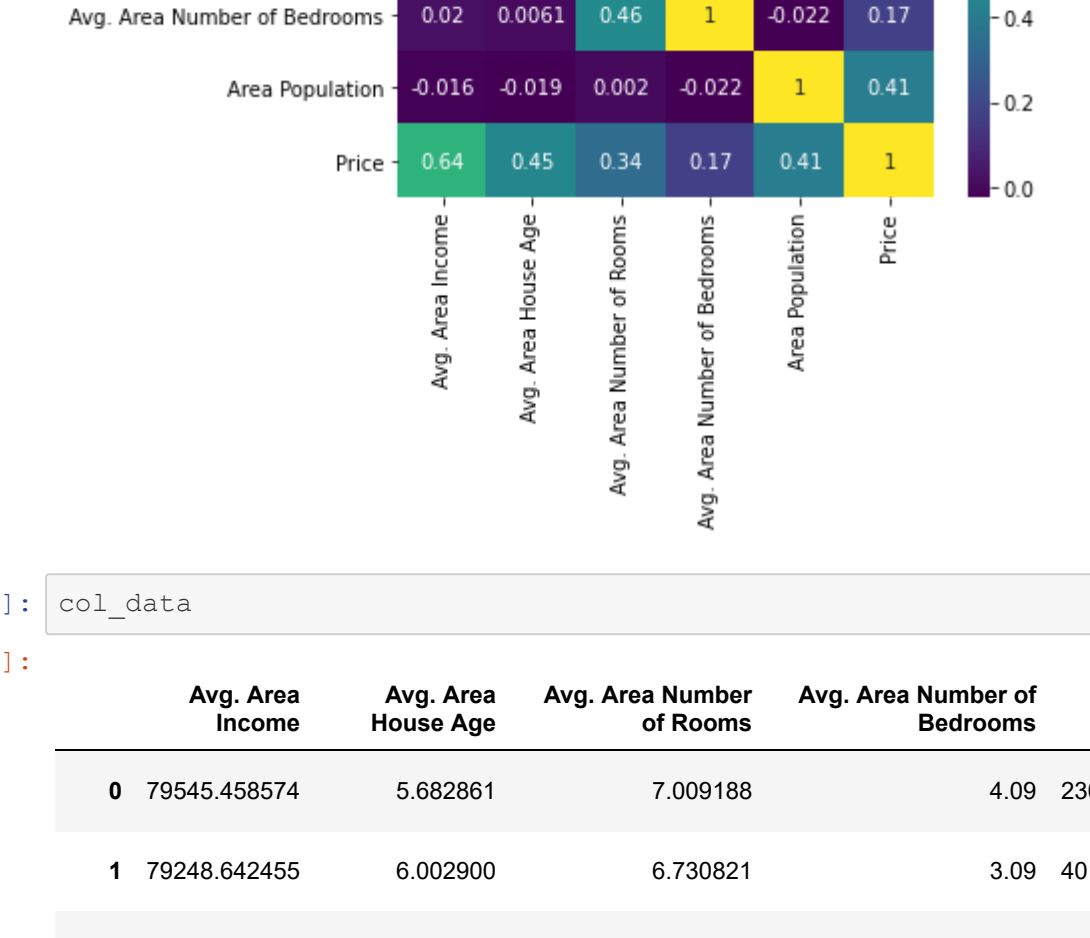
In [29]:
```
col_data.corr()
```

Out[29]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price
Avg. Area Income	1.000000	-0.002007	-0.011032	0.019788	-0.016234	0.639734
Avg. Area House Age	-0.002007	1.000000	-0.009428	0.006149	-0.018743	0.452543
Avg. Area Number of Rooms	-0.011032	-0.009428	1.000000	0.462695	0.002040	0.335664
Avg. Area Number of Bedrooms	0.019788	0.006149	0.462695	1.000000	-0.022168	0.171071
Area Population	-0.016234	-0.018743	0.002040	-0.022168	1.000000	0.408556
Price	0.639734	0.452543	0.335664	0.171071	0.408556	1.000000

In [39]:
```
sns.heatmap(col_data.corr(),annot=True,cmap='viridis')
```

Out[39]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x28c27927c70>
```



In [47]:
```
col_data
```

Out[47]:

	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
...	...	...	...	...	...	...	...
4995	60567.944140	7.830362	6.137356	3.46	22837.361035	1.060194e+06	USNS Williams\nFPO AP 30153-7653
4996	78491.275435	6.999135	6.576763	4.02	25616.115489	1.482618e+06	PSC 9258, Box 8489\nAPO AA 42991-3352
4997	63390.686886	7.250591	4.805081	2.13	33266.145490	1.030730e+06	4215 Tracy Garden Suite 076\nJoshualand, VA 01...
4998	68001.331235	5.534388	7.130144	5.44	42625.620156	1.198657e+06	USS Wallace\nFPO AE 73316
4999	65510.581804	5.992305	6.792336	4.07	46501.283803	1.298950e+06	37778 George Ridges Apt. 509\nEast Holly, NV 2...

5000 rows × 7 columns

In [54]:
```
data =col_data.drop(['Address'],axis=1)
```

In [55]: `data`

Out[55]:

	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 [75]: `X_= data.iloc[:,[0,1,2,3,4]]`

In [76]: `X_axis`

Out[76]:

	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
...	...	...	...	...	...
4995	60567.944140	7.830362	6.137356	3.46	22837.361035
4996	78491.275435	6.999135	6.576763	4.02	25616.115489
4997	63390.686886	7.250591	4.805081	2.13	33266.145490
4998	68001.331235	5.534388	7.130144	5.44	42625.620156
4999	65510.581804	5.992305	6.792336	4.07	46501.283803

5000 rows × 5 columns

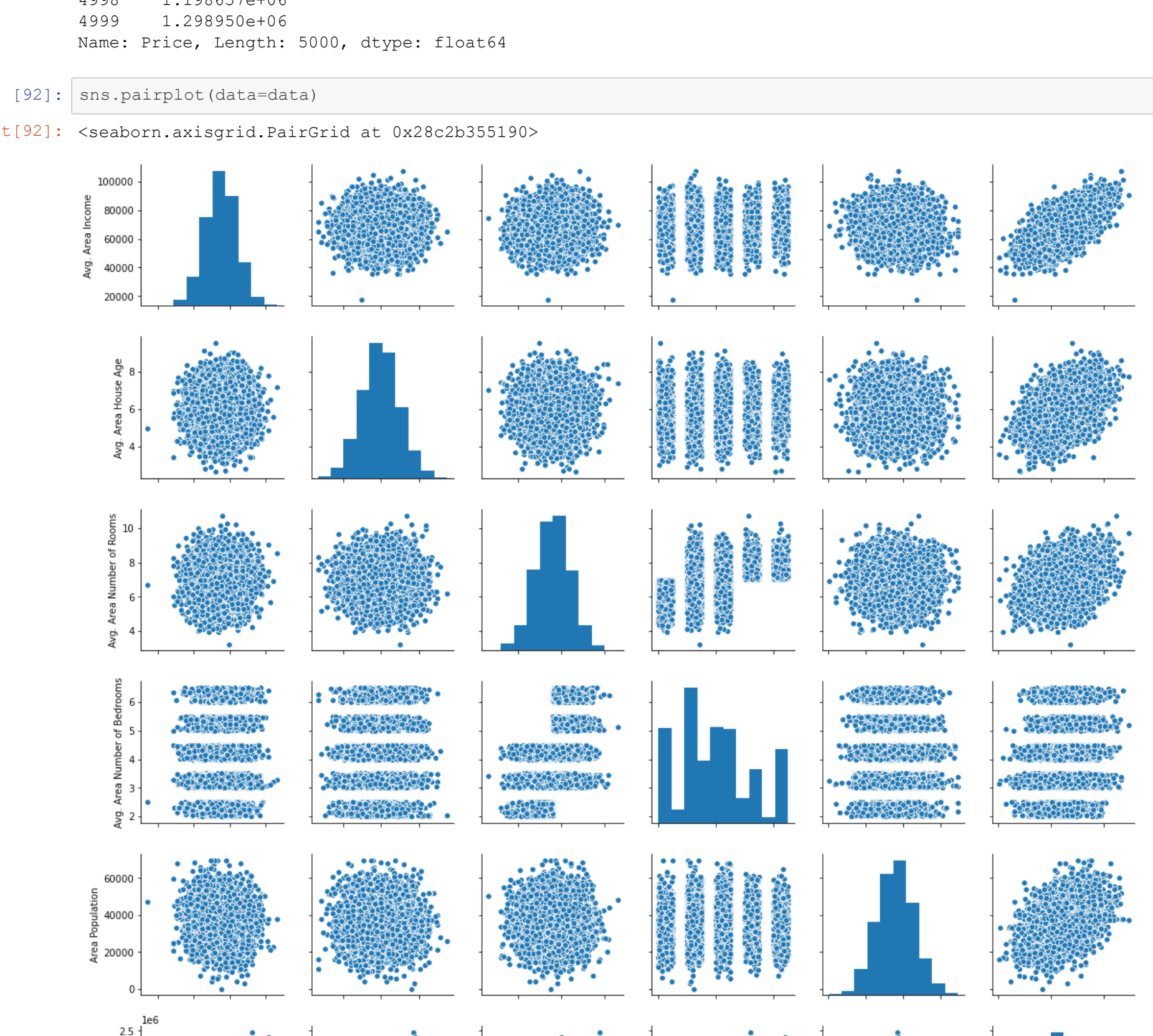
In [85]: `Y_axis =data['Price']`

In [90]: `Y_axis`

Out[90]:
```
0       1.059034e+06
1       1.505891e+06
2       1.058988e+06
3       1.260617e+06
4       6.309435e+05
            ...
4995    1.060194e+06
4996    1.482618e+06
4997    1.030730e+06
4998    1.198657e+06
4999    1.298950e+06
Name: Price, Length: 5000, dtype: float64
```

In [92]: `sns.pairplot(data=data)`

Out[92]: `<seaborn.axisgrid.PairGrid at 0x28c2b355190>`



In [ ]:

In [83]: `from sklearn.model_selection import train_test_split`

In [94]: `X_train, X_test, y_train, y_test = train_test_split( X_axis, Y_axis, test_size=0.33, random_state=42)`

In [98]: `from sklearn.metrics import mean_squared_error`

In [100]: `X_train, X_test, y_train, y_test`

Out[100]:
```
(      Avg. Area Income  Avg. Area House Age  Avg. Area Number of Rooms  \
 1522      67716.748222             6.431119                   7.753989
 835       78095.352630             6.837193                   7.119862
 358       55577.071754             6.533916                   7.162536
 138       63876.059910             7.080219                   7.187433
 299       52188.118727             5.636892                   5.807401
 ...                ...                  ...                        ...
 4426      76223.561256             6.371627                   5.342217
 466       56685.014442             6.958045                   7.502115
 3092      66195.337714             6.507971                   6.611861
 3772      58694.315017             7.394768                   9.269453
 860       61162.580254             5.896316                   7.880521

       Avg. Area Number of Bedrooms  Area Population
 1522                          5.28     40562.188948
 835                           3.00     32346.434992
 358                           3.38     19560.012414
 138                           3.37     38052.876443
 299                           4.03     42495.754677
 ...                            ...              ...
 4426                          2.42     30165.337445
 466                           3.38     43322.166854
 3092                          3.14     37288.923574
 3772                          4.32     49960.977236
 860                           6.04     36033.701431

 [3350 rows x 5 columns],
       Avg. Area Income  Avg. Area House Age  Avg. Area Number of Rooms  \
 1501      61907.593345             7.017838                   6.440256
 2586      57160.202243             6.893260                   6.921532
 2653      70190.796445             6.745054                   6.662567
 1055      69316.796889             6.300409                   7.873576
 705       72991.481649             3.412866                   6.494081
 ...                ...                  ...                        ...
 908       80000.764172             5.612802                   5.910013
 2114      68450.343795             6.829171                   6.459429
 3896      58318.861500             6.643226                   7.028824
 1627      78866.665247             6.331005                   6.834236
 2873      57471.943168             5.970953                   7.359195

       Avg. Area Number of Bedrooms  Area Population
 1501                          3.25     43828.947207
 2586                          3.13     43467.147035
 2653                          2.01     29215.136112
 1055                          4.28     24448.211461
 705                           2.48     50626.495426
 ...                            ...              ...
 908                           3.17     42245.446401
 2114                          2.38     32147.638435
 3896                          4.29     29491.635273
 1627                          3.44     39816.749113
 2873                          5.37     35822.601484

 [1650 rows x 5 columns],
 1522    1.535782e+06
 835     1.538923e+06
 358     8.180579e+05
 138     1.388530e+06
 299     7.838084e+05
             ...
 4426    1.023944e+06
 466     1.223101e+06
 3092    1.318559e+06
 3772    1.370674e+06
 860     1.060898e+06
 Name: Price, Length: 3350, dtype: float64,
 1501    1.339096e+06
 2586    1.251794e+06
 2653    1.340039e+06
 1055    1.431508e+06
 705     1.042374e+06
             ...
 908     1.383967e+06
 2114    1.203090e+06
 3896    8.592087e+05
 1627    1.509423e+06
 2873    9.987020e+05
 Name: Price, Length: 1650, dtype: float64)
```

In [104]: `from sklearn.linear_model import LinearRegression`

In [108]: `from sklearn.linear_model import LinearRegression`

In [113]: `reg = LinearRegression()`

In [124]: `reg.fit(X_train,y_train)`

Out[124]: `LinearRegression()`

In [127]: `y_pred = reg.predict(X_test)`

In [128]: `y_pred`

Out[128]:
```
array([1310035.96931578, 1238811.85458543, 1245265.83079141, ...,
       1024496.96962118, 1548581.80349715, 1032639.16178956])
```

In [ ]:

In [132]: `reg.coef_`

Out[132]:
```
array([2.15898874e+01, 1.66102501e+05, 1.19895936e+05, 1.90107101e+03,
       1.52315025e+01])
```

In [137]: `cdf =pd.DataFrame(reg.coef_,X_axis.columns,columns=['coeff'])`

In [139]: `cdf`

Out[139]:

	coeff
Avg. Area Income	21.589887
Avg. Area House Age	166102.501246
Avg. Area Number of Rooms	119895.936402
Avg. Area Number of Bedrooms	1901.071012
Area Population	15.231503

In [1]: `sns.distplot(cdf,rug='False',hist='True',bins=5)`

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-6658b50f8b8f> in <module>
----> 1 sns.distplot(cdf,rug='False',hist='True',bins=5)

NameError: name 'sns' is not defined
```

In [2]: `pip install pandas-profiling`

```
Collecting pandas-profilingNote: you may need to restart the kernel to use updated packages.

  WARNING: Retrying (Retry(total=4, connect=None, read=None, redirect=None, status=None)) after conne
ction broken by 'SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate v
erify failed: unable to get local issuer certificate (_ssl.c:1108)'))': /packages/dd/12/e2870750c5320
116efe7bebd4ae1709cd7e35e3bc23ac8039864b05b9497/pandas_profiling-2.11.0-py2.py3-none-any.whl
  WARNING: Retrying (Retry(total=3, connect=None, read=None, redirect=None, status=None)) after conne
ction broken by 'SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate v
erify failed: unable to get local issuer certificate (_ssl.c:1108)'))': /packages/dd/12/e2870750c5320
116efe7bebd4ae1709cd7e35e3bc23ac8039864b05b9497/pandas_profiling-2.11.0-py2.py3-none-any.whl
  WARNING: Retrying (Retry(total=2, connect=None, read=None, redirect=None, status=None)) after conne
ction broken by 'SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate v
erify failed: unable to get local issuer certificate (_ssl.c:1108)'))': /packages/dd/12/e2870750c5320
116efe7bebd4ae1709cd7e35e3bc23ac8039864b05b9497/pandas_profiling-2.11.0-py2.py3-none-any.whl
  WARNING: Retrying (Retry(total=1, connect=None, read=None, redirect=None, status=None)) after conne
ction broken by 'SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate v
erify failed: unable to get local issuer certificate (_ssl.c:1108)'))': /packages/dd/12/e2870750c5320
116efe7bebd4ae1709cd7e35e3bc23ac8039864b05b9497/pandas_profiling-2.11.0-py2.py3-none-any.whl
  WARNING: Retrying (Retry(total=0, connect=None, read=None, redirect=None, status=None)) after conne
ction broken by 'SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate v
erify failed: unable to get local issuer certificate (_ssl.c:1108)'))': /packages/dd/12/e2870750c5320
116efe7bebd4ae1709cd7e35e3bc23ac8039864b05b9497/pandas_profiling-2.11.0-py2.py3-none-any.whl
ERROR: Could not install packages due to an EnvironmentError: HTTPSConnectionPool(host='files.pythonh
osted.org', port=443): Max retries exceeded with url: /packages/dd/12/e2870750c5320116efe7bebd4ae1709
cd7e35e3bc23ac8039864b05b9497/pandas_profiling-2.11.0-py2.py3-none-any.whl (Caused by SSLError(SSLCer
tVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: unable to get loca
l issuer certificate (_ssl.c:1108)')))
```

In [3]: `!pip install -U pandas-profiling`

```
Collecting pandas-profiling
  WARNING: Retrying (Retry(total=4, connect=None, read=None, redirect=None, status=None)) after conne
ction broken by 'SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate v
erify failed: unable to get local issuer certificate (_ssl.c:1108)'))': /packages/dd/12/e2870750c5320
116efe7bebd4ae1709cd7e35e3bc23ac8039864b05b9497/pandas_profiling-2.11.0-py2.py3-none-any.whl
  WARNING: Retrying (Retry(total=3, connect=None, read=None, redirect=None, status=None)) after conne
ction broken by 'SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate v
erify failed: unable to get local issuer certificate (_ssl.c:1108)'))': /packages/dd/12/e2870750c5320
116efe7bebd4ae1709cd7e35e3bc23ac8039864b05b9497/pandas_profiling-2.11.0-py2.py3-none-any.whl
  WARNING: Retrying (Retry(total=2, connect=None, read=None, redirect=None, status=None)) after conne
ction broken by 'SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate v
erify failed: unable to get local issuer certificate (_ssl.c:1108)'))': /packages/dd/12/e2870750c5320
116efe7bebd4ae1709cd7e35e3bc23ac8039864b05b9497/pandas_profiling-2.11.0-py2.py3-none-any.whl
  WARNING: Retrying (Retry(total=1, connect=None, read=None, redirect=None, status=None)) after conne
ction broken by 'SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate v
erify failed: unable to get local issuer certificate (_ssl.c:1108)'))': /packages/dd/12/e2870750c5320
116efe7bebd4ae1709cd7e35e3bc23ac8039864b05b9497/pandas_profiling-2.11.0-py2.py3-none-any.whl
  WARNING: Retrying (Retry(total=0, connect=None, read=None, redirect=None, status=None)) after conne
ction broken by 'SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate v
erify failed: unable to get local issuer certificate (_ssl.c:1108)'))': /packages/dd/12/e2870750c5320
116efe7bebd4ae1709cd7e35e3bc23ac8039864b05b9497/pandas_profiling-2.11.0-py2.py3-none-any.whl
ERROR: Could not install packages due to an EnvironmentError: HTTPSConnectionPool(host='files.pythonh
osted.org', port=443): Max retries exceeded with url: /packages/dd/12/e2870750c5320116efe7bebd4ae1709
cd7e35e3bc23ac8039864b05b9497/pandas_profiling-2.11.0-py2.py3-none-any.whl (Caused by SSLError(SSLCer
tVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: unable to get loca
l issuer certificate (_ssl.c:1108)')))
```