USA HOUSING PRICE PREDICTION

```
1) import pandas as pd


2) # Load the dataset

file_path = "/content/USA_Housing (1).csv"
df = pd.read_csv(file_path)

# Display basic information about the dataset
df.info(), df.head()
```

OUTPUT

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 7 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Avg. Area Income              5000 non-null   float64
 1   Avg. Area House Age           5000 non-null   float64
 2   Avg. Area Number of Rooms     5000 non-null   float64
 3   Avg. Area Number of Bedrooms  5000 non-null   float64
 4   Area Population               5000 non-null   float64
 5   Price                         5000 non-null   float64
 6   Address                       5000 non-null   object
dtypes: float64(6), object(1)
memory usage: 273.6+ KB
(None,
    Avg. Area Income  Avg. Area House Age  Avg. Area Number of
Rooms  \
 0      79545.458574             5.682861                   7.009188
 1      79248.642455             6.002900                   6.730821
 2      61287.067179             5.865890                   8.512727
 3      63345.240046             7.188236                   5.586729
 4     59982.197226          5.040555               7.839388

    Avg. Area Number of Bedrooms  Area Population         Price  \
 0                          4.09     23086.800503  1.059034e+06
 1                          3.09     40173.072174  1.505891e+06
 2                          5.13     36882.159400  1.058988e+06
 3                          3.26     34310.242831  1.260617e+06
 4                          4.23     26354.109472  6.309435e+05

                                              Address
 0  208 Michael Ferry Apt. 674\nLaurabury, NE 3701...
 1  188 Johnson Views Suite 079\nLake Kathleen, CA...
 2  9127 Elizabeth Stravenue\nDanieltown, WI 06482...
 3                          USS Barnett\nFPO AP 44820
 4                         USNS Raymond\nFPO AE 09386  )
```

```
3) # Check for duplicate rows
duplicate_rows = df.duplicated().sum()
```

```
4) # Check for outliers using summary statistics
summary_stats = df.describe()

duplicate_rows, summary_stats
```

OUTPUT

```
(0,
       Avg. Area Income  Avg. Area House Age  Avg. Area Number of Rooms  \
 count       5000.000000          5000.000000                5000.000000
 mean       68583.108984             5.977222                   6.987792
 std        10657.991214             0.991456                   1.005833
 min        17796.631190             2.644304                   3.236194
 25%        61480.562388             5.322283                   6.299250
 50%      68804.286404          5.970429              7.002902
 75%      75783.338666          6.650808              7.665871
 max     107701.748378          9.519088             10.759588

       Avg. Area Number of Bedrooms  Area Population         Price
 count                   5000.000000      5000.000000  5.000000e+03
 mean                       3.981330     36163.516039  1.232073e+06
 std                        1.234137      9925.650114  3.531176e+05
 min                        2.000000       172.610686  1.593866e+04
 25%                        3.140000     29403.928702  9.975771e+05
 50%                        4.050000     36199.406689  1.232669e+06
 75%                        4.490000     42861.290769  1.471210e+06
 max                        6.500000     69621.713378  2.469066e+06  )
```

5) # Display basic info and first few rows

```
df.info(), df.head()
```

OUTPUT

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 7 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Avg. Area Income              5000 non-null
float64
 1   Avg. Area House Age           5000 non-null
float64
 2   Avg. Area Number of Rooms     5000 non-null
float64
 3   Avg. Area Number of Bedrooms  5000 non-null   float64
 4   Area Population               5000 non-null
float64
 5   Price                         5000 non-null
float64
 6   Address                       5000 non-null
object
dtypes: float64(6), object(1)
```

```
memory usage: 273.6+ KB
(None,
    Avg. Area Income  Avg. Area House Age  Avg. Area Number of
Rooms  \
 0      79545.458574            5.682861                 7.009188
 1      79248.642455            6.002900                 6.730821
 2      61287.067179            5.865890                 8.512727
 3      63345.240046            7.188236                 5.586729
 4      59982.197226            5.040555                 7.839388

    Avg. Area Number of Bedrooms                    Area Population
Price  \
 0                        4.09                          23086.800503
1.059034e+06
 1                        3.09                           40173.072174
1.505891e+06
 2                        5.13
36882.159400         1.058988e+06
 3                        3.26
34310.242831         1.260617e+06
 4                        4.23                           26354.109472
6.309435e+05

                                            Address
 0  208 Michael Ferry Apt. 674\nLaurabury, NE 3701...
 1  188 Johnson Views Suite 079\nLake Kathleen, CA...
 2  9127 Elizabeth Stravenue\nDanieltown, WI 06482...
 3                         USS Barnett\nFPO AP 44820
 4                        USNS Raymond\nFPO AE 09386  )
```

```
6) import seaborn as sns

import matplotlib.pyplot as plt
```

```
7) # Drop Address column as it's not useful for analysis

df_cleaned = df.drop(columns=['Address'])
```

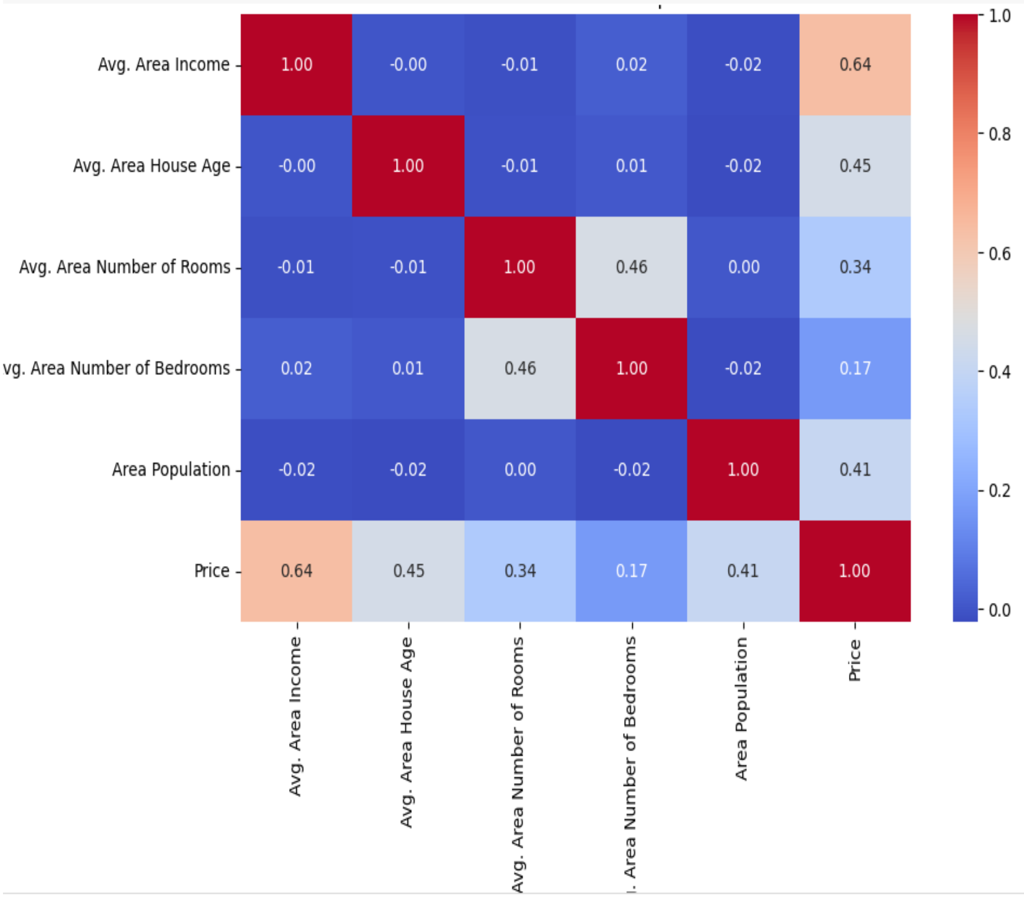
```
8) # Summary statistics
summary_stats = df_cleaned.describe()
```

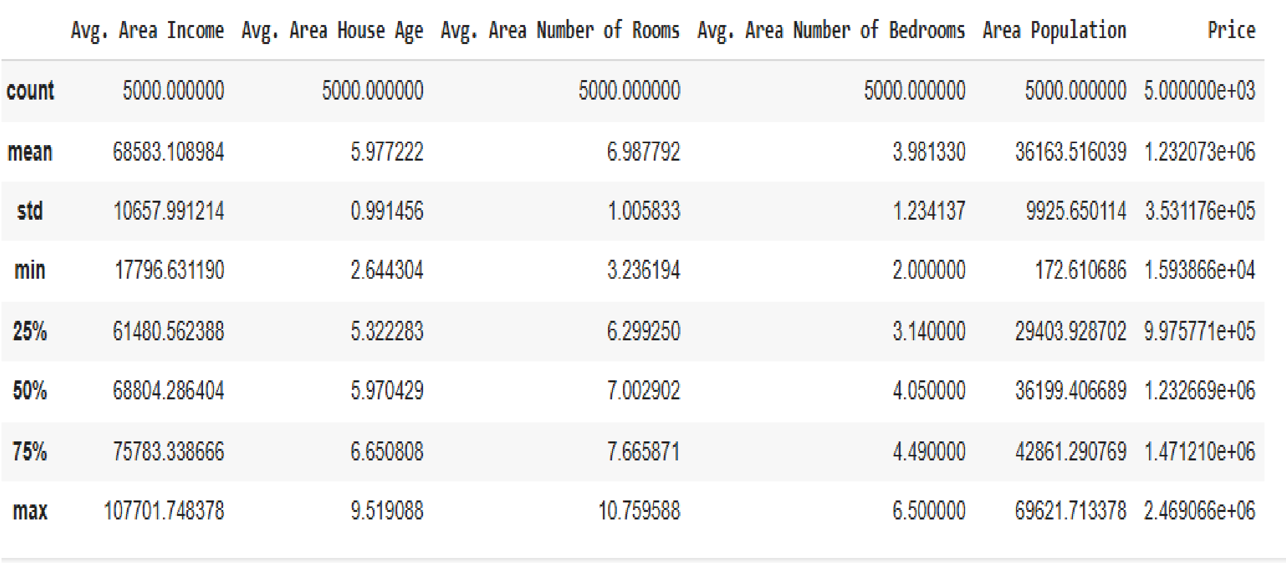
```
9) # Correlation matrix
correlation_matrix = df_cleaned.corr()
```

```
10) # Visualization - Correlation Heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm",
fmt=".2f")
plt.title("Correlation Heatmap")
plt.show()

summary_stats
```

```
mmary_stats
```





	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5.000000e+03
mean	68583.108984	5.977222	6.987792	3.981330	36163.516039	1.232073e+06
std	10657.991214	0.991456	1.005833	1.234137	9925.650114	3.531176e+05
min	17796.631190	2.644304	3.236194	2.000000	172.610686	1.593866e+04
25%	61480.562388	5.322283	6.299250	3.140000	29403.928702	9.975771e+05
50%	68804.286404	5.970429	7.002902	4.050000	36199.406689	1.232669e+06
75%	75783.338666	6.650808	7.665871	4.490000	42861.290769	1.471210e+06
max	107701.748378	9.519088	10.759588	6.500000	69621.713378	2.469066e+06

11) # Display basic information and first few rows

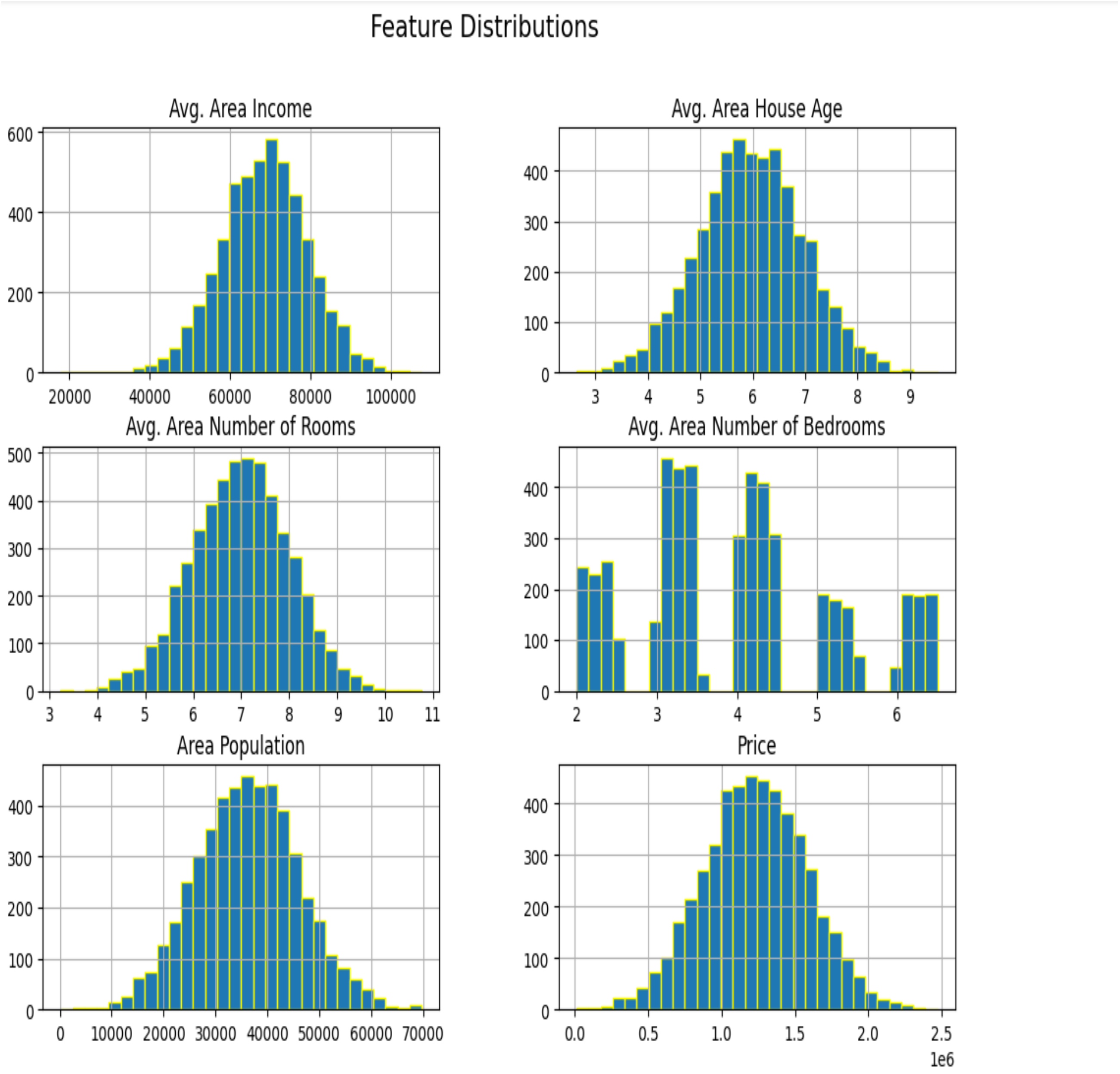
```
df.info(), df.head()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 7 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Avg. Area Income              5000 non-null   float64
 1   Avg. Area House Age           5000 non-null   float64
 2   Avg. Area Number of Rooms     5000 non-null   float64
 3   Avg. Area Number of Bedrooms  5000 non-null   float64
 4   Area Population               5000 non-null   float64
 5   Price                         5000 non-null   float64
 6   Address                       5000 non-null   object
dtypes: float64(6), object(1)
memory usage: 273.6+ KB
(None,
   Avg. Area Income  Avg. Area House Age  Avg. Area Number of
Rooms  \
 0      79545.458574             5.682861                   7.009188
 1      79248.642455             6.002900                   6.730821
 2      61287.067179             5.865890                   8.512727
 3      63345.240046             7.188236                   5.586729
 4      59982.197226             5.040555                   7.839388

   Avg. Area Number of Bedrooms  Area Population         Price  \
 0                          4.09     23086.800503  1.059034e+06
 1                          3.09     40173.072174  1.505891e+06
 2                          5.13     36882.159400  1.058988e+06
 3                          3.26     34310.242831  1.260617e+06
 4                          4.23     26354.109472  6.309435e+05
```

```
                                             Address
0  208 Michael Ferry Apt. 674\nLaurabury, NE 3701...
1  188 Johnson Views Suite 079\nLake Kathleen, CA...
2  9127 Elizabeth Stravenue\nDanieltown, WI 06482...
3                         USS Barnett\nFPO AP 44820
4                        USNS Raymond\nFPO AE 09386  )
```

```
12) #Plot histograms for feature distributions

df.hist(figsize=(12, 8), bins=30, edgecolor='yellow')
plt.suptitle("Feature Distributions", fontsize=16)
plt.show()
```



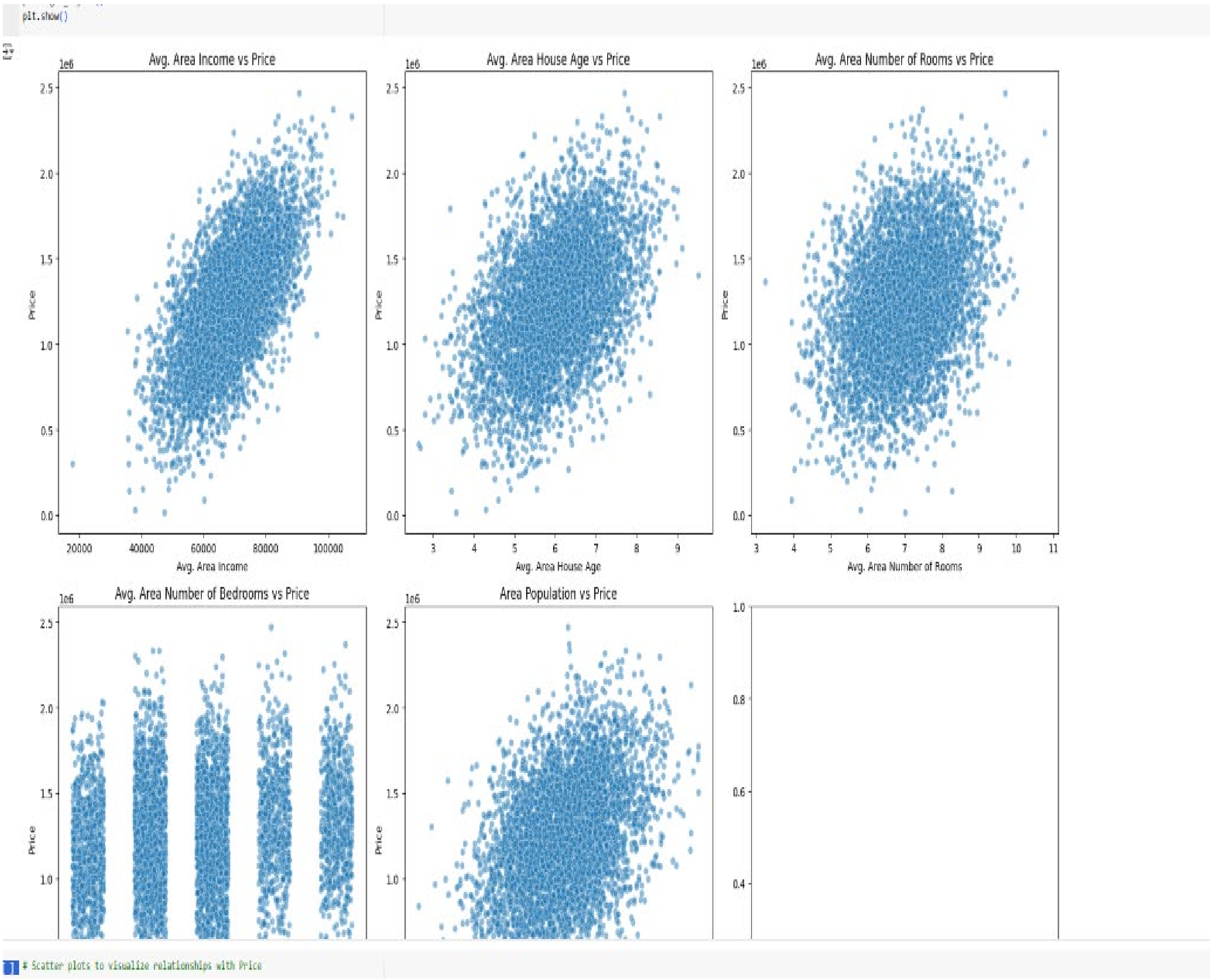
```
13) # Scatter plots to check relationships with house prices

fig, axes = plt.subplots(2, 3, figsize=(18, 12))
features = ["Avg. Area Income", "Avg. Area House Age", "Avg. Area Number of Rooms",
          "Avg. Area Number of Bedrooms", "Area Population"]

for ax, feature in zip(axes.flat, features):
   sns.scatterplot(x=df[feature], y=df["Price"], ax=ax, alpha=0.5)
```

```
    ax.set_title(f"{feature} vs Price")

plt.tight_layout()
plt.show()
```



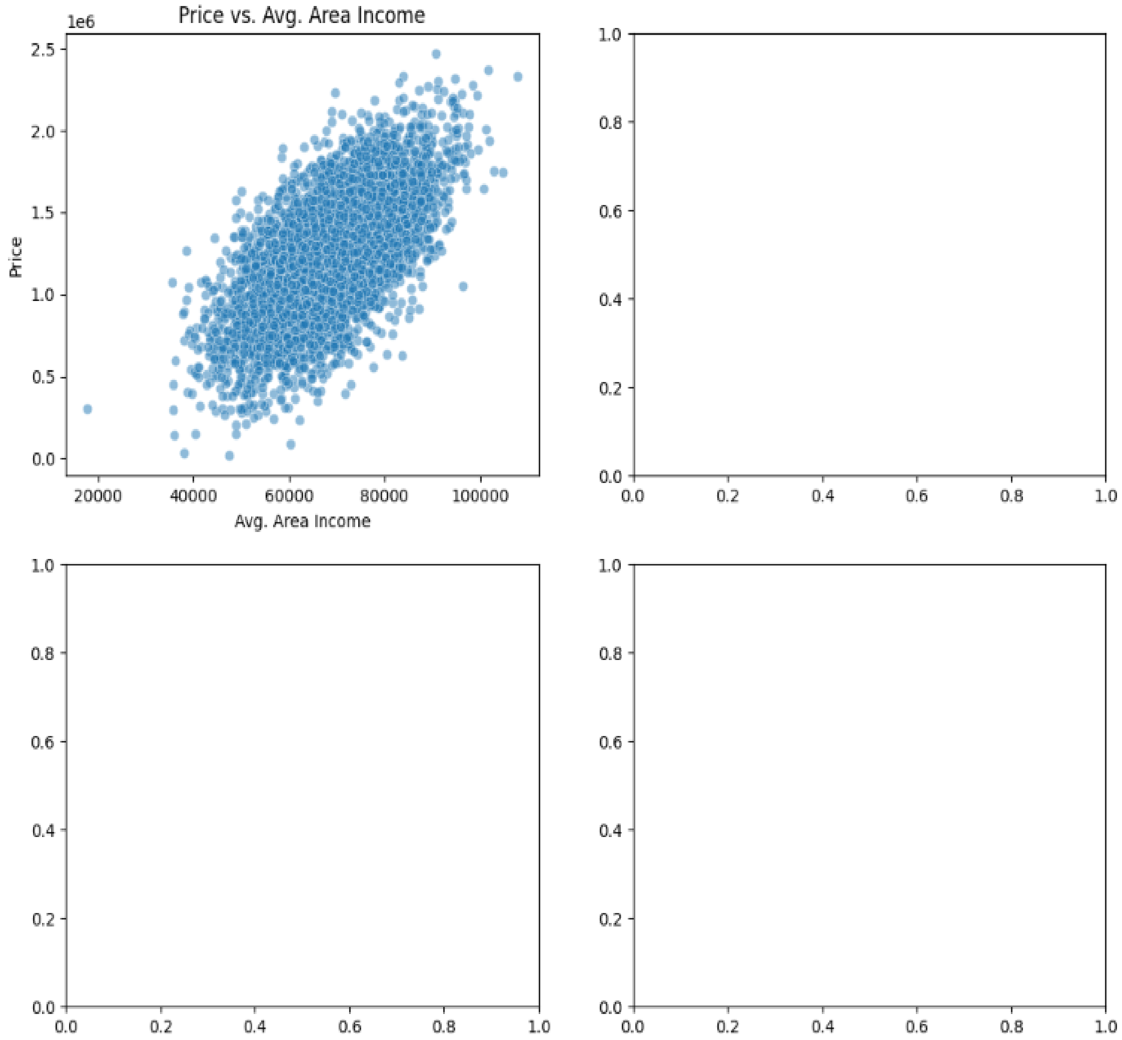
```
plt.show()
```

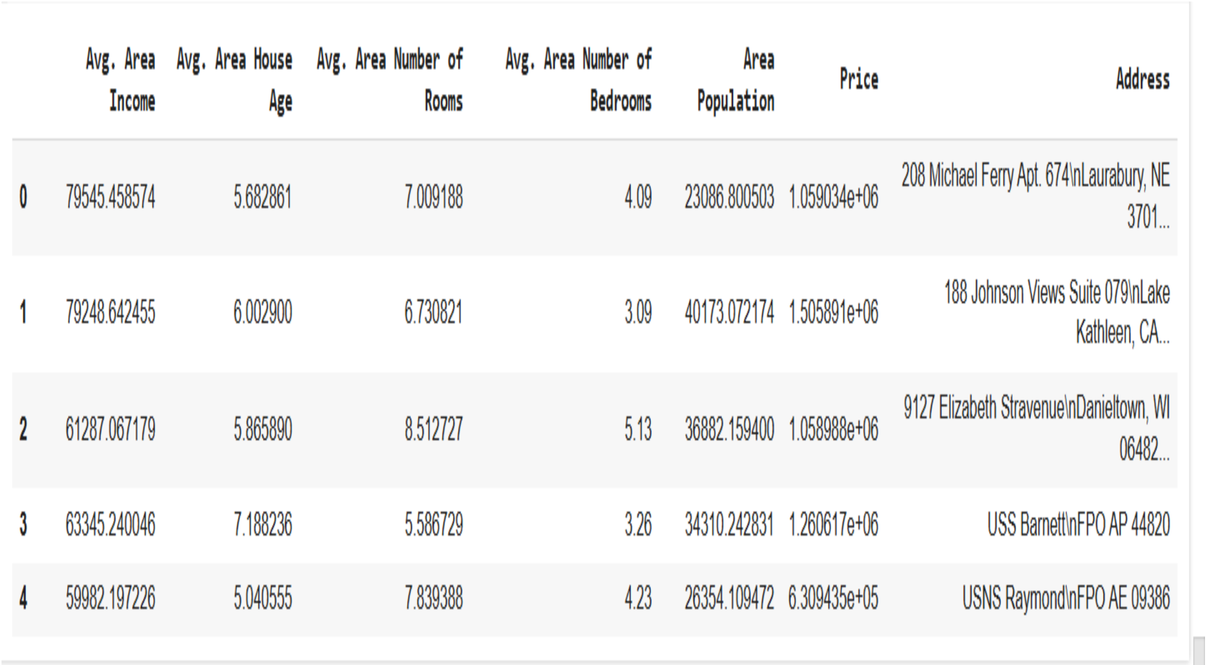
```
# Scatter plots to visualize relationships with Price
```

```
14) # Scatter plots to visualize relationships with Price

fig, axes = plt.subplots(2, 2, figsize=(12, 10))
sns.scatterplot(x=df["Avg. Area Income"], y=df["Price"], ax=axes[0,
0], alpha=0.5)
axes[0, 0].set_title("Price vs. Avg. Area Income")
```

```
Text(0.5, 1.0, 'Price vs. Avg. Area Income')
```





	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

16) df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 7 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   Avg. Area Income              5000 non-null   float64
 1   Avg. Area House Age           5000 non-null   float64
 2   Avg. Area Number of Rooms     5000 non-null   float64
 3   Avg. Area Number of Bedrooms  5000 non-null   float64
 4   Area Population               5000 non-null   float64
 5   Price                         5000 non-null   float64
 6   Address                       5000 non-null   object 
dtypes: float64(6), object(1)
memory usage: 273.6+ KB
```

17) df.isna().sum()

```
                                   0
            Avg. Area Income       0
          Avg. Area House Age      0
      Avg. Area Number of Rooms    0
    Avg. Area Number of Bedrooms   0
             Area Population       0
                  Price            0
                 Address           0

dtype: int64
```

18)
```
#checking column names
df.columns
```

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

19)
```
df['Avg. Area Number of Bedrooms'].nunique()
```

```
255
```

20)
```
df['Avg. Area Number of Rooms'].nunique()
```

```
5000
```

21)

```
df['Avg. Area House Age'].nunique()
```

```
5000
```

22)

```
df['Avg. Area Income'].nunique()
```

```
5000
```

23)

```
df["Area Population"].nunique()
```

```
5000
```