Regressão_Linear

April 29, 2025

0.1 Regressão Linear

- **Regressão Linear Simples**: um atributo independente e um atributo dependente.
- **Regressão Linear Univariada**: um atributo dependente e um conjunto de atributos independentes
- **Regressão Multivariada**: mais de um atributo dependente.

0.1.1 Regressão Linear Simples

- Y - variável dependente
- X - variável independente
- β_0 - a interceptação
- β_1 - a inclinação

Equação da reta para a RLS

$e = \beta_0 + \beta_1 * X$

[3]:

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

[4]:

```
# Valores de para x de 0 até 19, totalizando 20 pontos
x = np.arange(0,19)
# Matriz para multiplicação dos dados
A=np.array([x, np.ones(19)])
# conjunto de alvos para regressão
y = [22, 24, 23, 20, 23, 30, 22, 24, 24, 20, 21, 30, 24, 20, 28, 20, 22, 24, 27]
```

[5]:

```
x
```

[5]:

```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       17, 18])
```

[6]:

```
print(A)
```

```
[[ 0.  1.  2.  3.  4.  5.  6.  7.  8.  9. 10. 11. 12. 13. 14. 15. 16. 17.
  18.]
 [ 1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.  1.
   1.]]
```

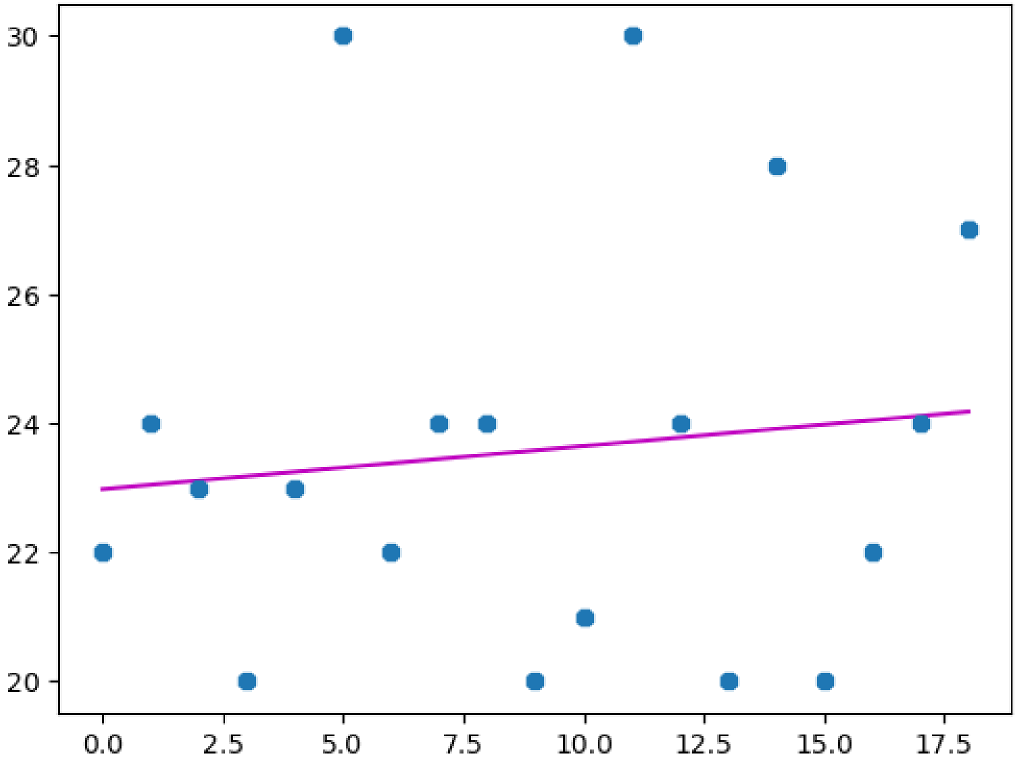
[7]:
```
w=np.linalg.lstsq(A.T,y)[0]
```

```
/tmp/ipykernel_2595866/1867454896.py:1: FutureWarning: `rcond` parameter will
change to the default of machine precision times ``max(M, N)`` where M and N are
the input matrix dimensions.
To use the future default and silence this warning we advise to pass
`rcond=None`, to keep using the old, explicitly pass `rcond=-1`.
  w=np.linalg.lstsq(A.T,y)[0]
```

[8]:
```
# na posição [0] temos o o valor de beta 1 e na posição 1 o valor de beta 0
print(w)
```

```
[ 0.06666667 22.97894737]
```

[9]:
```
linha = w[0]*x+w[1]
plt.plot(x, linha,'m-')
plt.plot(x, y, '8')
plt.show()
```



[10]:
```
p19=w[0]*19+w[1]
p19
```

[10]:
```
24.2456140350877
```

```
[ ]:
```

```
[11]: import seaborn as sns
```

```
[12]: casas = pd.read_csv('USA_Housing.csv')
```

0.2 Análise estátistica do quadro de dados

```
[13]: casas.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
```

```
[14]: casas.head()
```

```
[14]:    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
```

[15]:
```
casas.tail()
```

[15]:
```
      Avg. Area Income  Avg. Area House Age  Avg. Area Number of Rooms  \
4995      60567.944140             7.830362                   6.137356
4996      78491.275435             6.999135                   6.576763
4997      63390.686886             7.250591                   4.805081
4998      68001.331235             5.534388                   7.130144
4999      65510.581804             5.992305                   6.792336

      Avg. Area Number of Bedrooms  Area Population         Price  \
4995                          3.46     22837.361035  1.060194e+06
4996                          4.02     25616.115489  1.482618e+06
4997                          2.13     33266.145490  1.030730e+06
4998                          5.44     42625.620156  1.198657e+06
4999                          4.07     46501.283803  1.298950e+06

                                                Address
4995                   USNS Williams\nFPO AP 30153-7653
4996              PSC 9258, Box 8489\nAPO AA 42991-3352
4997  4215 Tracy Garden Suite 076\nJoshualand, VA 01…
4998                          USS Wallace\nFPO AE 73316
4999  37778 George Ridges Apt. 509\nEast Holly, NV 2…
```

[16]:
```
casas.describe().transpose()
```

[16]:
```
                               count          mean            std  \
Avg. Area Income              5000.0  6.858311e+04   10657.991214
Avg. Area House Age           5000.0  5.977222e+00       0.991456
Avg. Area Number of Rooms     5000.0  6.987792e+00       1.005833
Avg. Area Number of Bedrooms  5000.0  3.981330e+00       1.234137
Area Population               5000.0  3.616352e+04    9925.650114
Price                         5000.0  1.232073e+06  353117.626581

                                       min            25%           50%  \
Avg. Area Income              17796.631190   61480.562388  6.880429e+04
Avg. Area House Age               2.644304       5.322283  5.970429e+00
Avg. Area Number of Rooms         3.236194       6.299250  7.002902e+00
Avg. Area Number of Bedrooms      2.000000       3.140000  4.050000e+00
Area Population                 172.610686   29403.928702  3.619941e+04
Price                         15938.657923  997577.135049  1.232669e+06

                                       75%           max
Avg. Area Income              7.578334e+04  1.077017e+05
Avg. Area House Age           6.650808e+00  9.519088e+00
Avg. Area Number of Rooms     7.665871e+00  1.075959e+01
Avg. Area Number of Bedrooms  4.490000e+00  6.500000e+00
Area Population               4.286129e+04  6.962171e+04
```

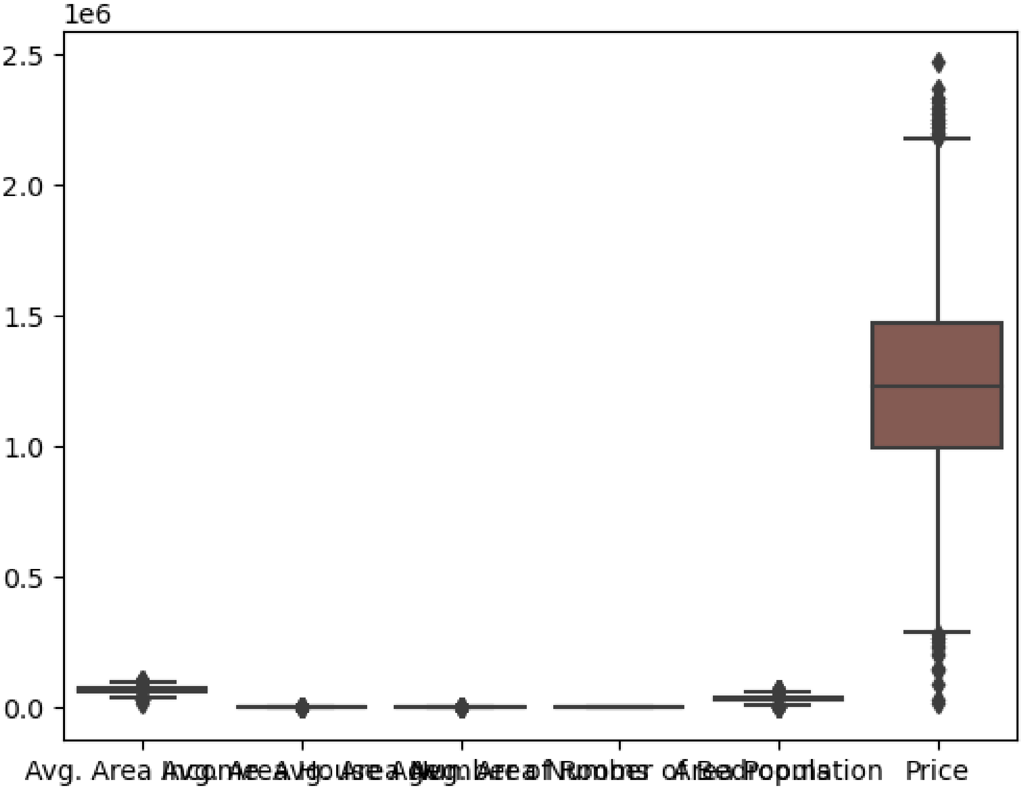
```
Price                               1.471210e+06  2.469066e+06
```

0.3 Análise Bivariada dos dados

- Boxplot
- gráfico de barras
- Pairplot
- Heatmap correlação dos dados

[17]:
```
sns.boxplot(casas)
```

[17]:
```
<AxesSubplot: >
```



[18]:
```
Q1=casas.quantile(0.25, numeric_only=True)
Q3=casas.quantile(0.75, numeric_only=True)
IRQ = Q3-Q1
```

[19]:
```
print(Q1)
```

```
Avg. Area Income                61480.562388
Avg. Area House Age                 5.322283
Avg. Area Number of Rooms           6.299250
Avg. Area Number of Bedrooms         3.140000
Area Population                  29403.928702
Price                           997577.135049
Name: 0.25, dtype: float64
```

[20]:
```
print(Q3)
```

```
Avg. Area Income                7.578334e+04
Avg. Area House Age             6.650808e+00
Avg. Area Number of Rooms       7.665871e+00
Avg. Area Number of Bedrooms    4.490000e+00
Area Population                 4.286129e+04
Price                           1.471210e+06
Name: 0.75, dtype: float64
```

[21]:
```
print(IRQ)
```

```
Avg. Area Income                 14302.776278
Avg. Area House Age                  1.328525
Avg. Area Number of Rooms            1.366621
Avg. Area Number of Bedrooms         1.350000
Area Population                  13457.362067
Price                           473633.069163
dtype: float64
```

[22]:
```
contador = casas[(casas<(Q1-1.5*IRQ)) | (casas > (Q3+1.5*IRQ))].count()

df_contagem = pd.DataFrame(contador, columns=['contagem de outliers'])
```

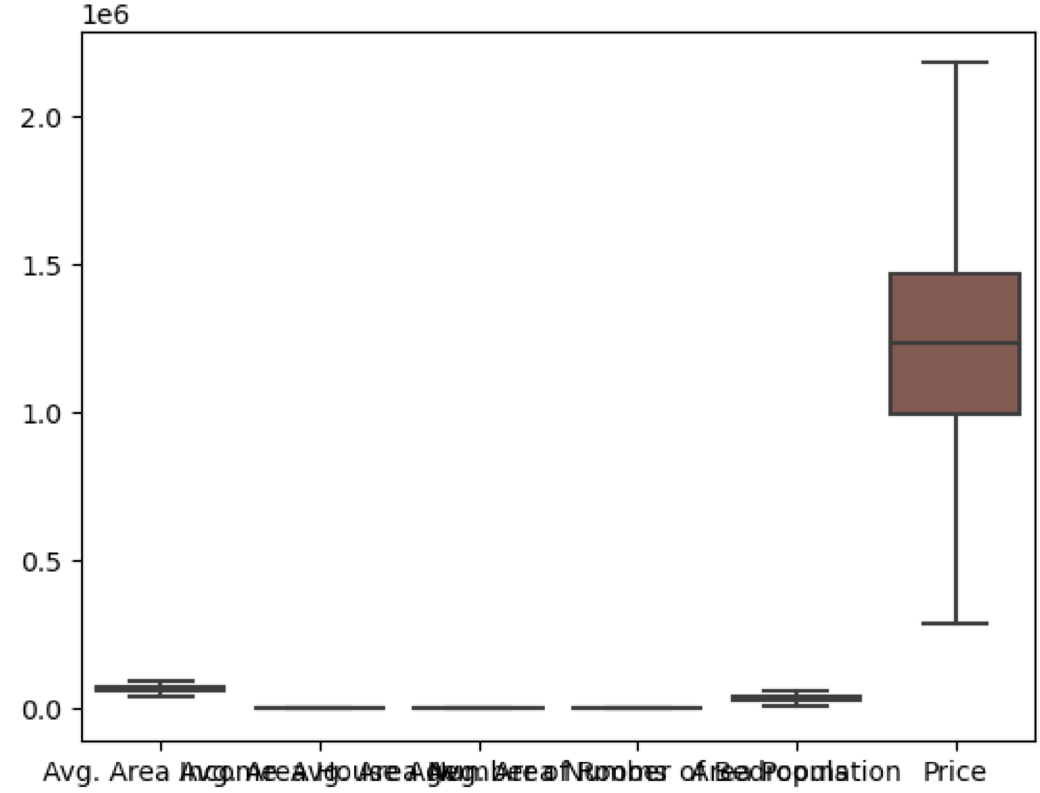
```
/tmp/ipykernel_2595866/1575016419.py:1: FutureWarning: Automatic reindexing on
DataFrame vs Series comparisons is deprecated and will raise ValueError in a
future version. Do `left, right = left.align(right, axis=1, copy=False)` before
e.g. `left == right`
  contador = casas[(casas<(Q1-1.5*IRQ)) | (casas > (Q3+1.5*IRQ))].count()
```

[23]:
```
df_contagem
```

[23]:
```
                              contagem de outliers
Avg. Area Income                                32
Avg. Area House Age                             25
Avg. Area Number of Rooms                       24
Avg. Area Number of Bedrooms                     0
Area Population                                 30
Price                                           35
Address                                          0
```

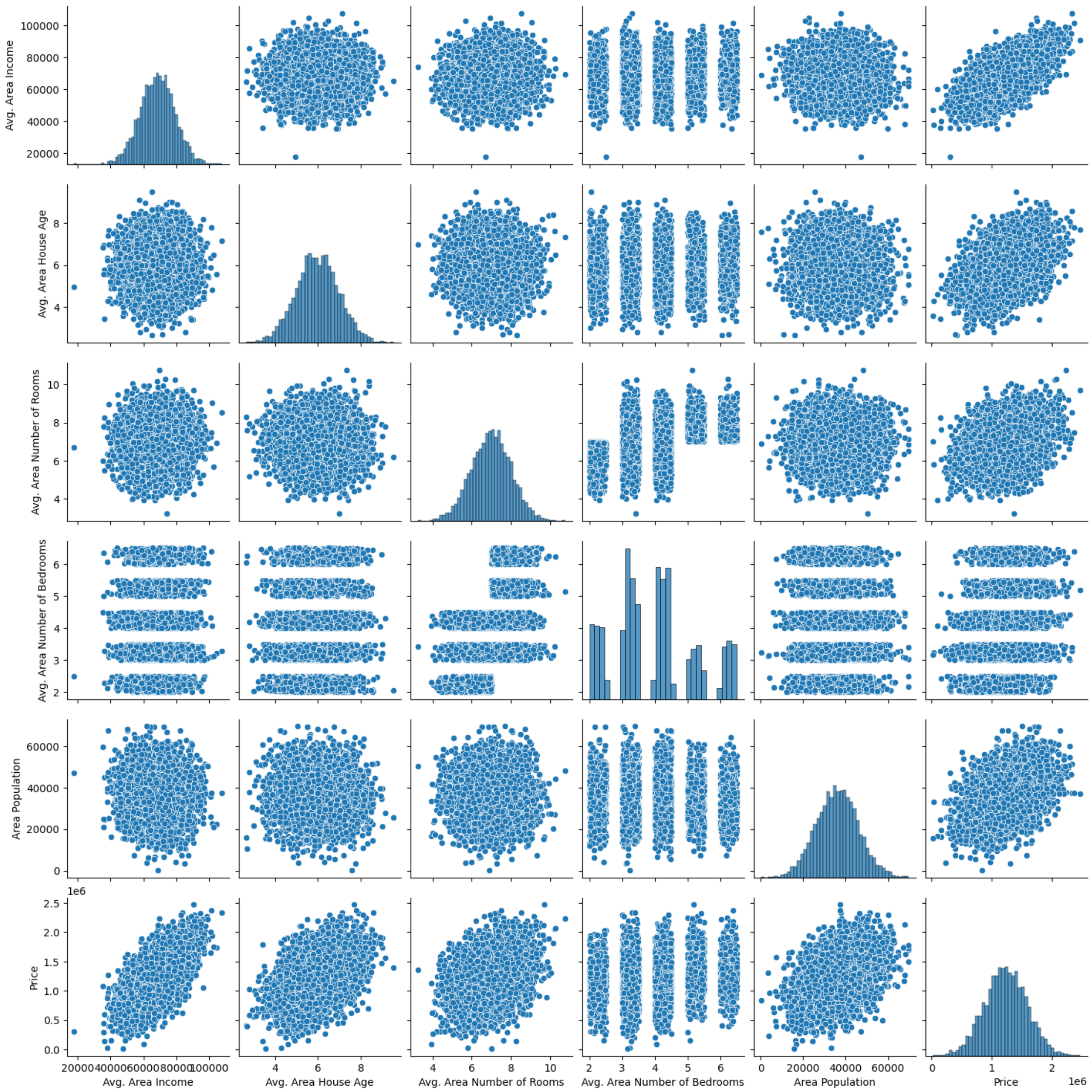
[24]:
```
sns.boxplot(casas, showfliers=False)
```

[24]:
```
<AxesSubplot: >
```



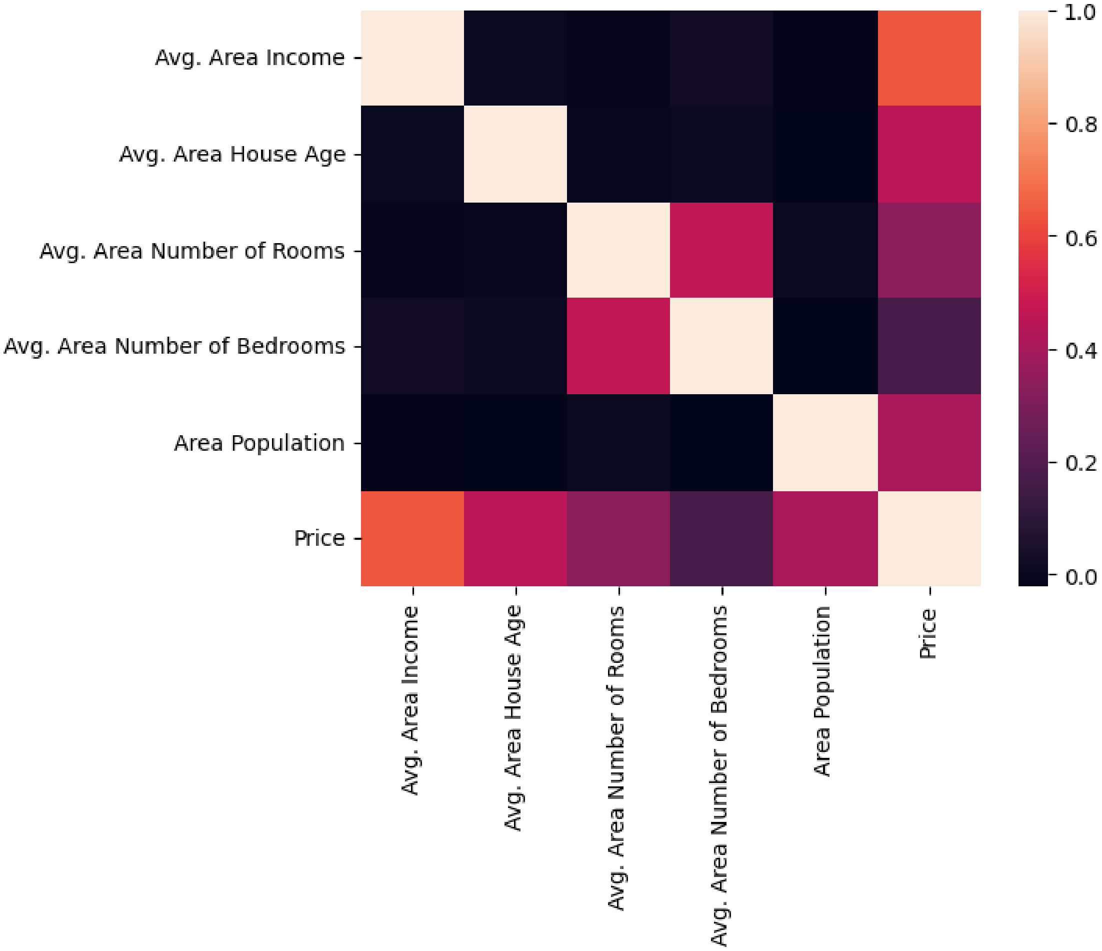
```
[25]: sns.pairplot(casas)
```

```
[25]: <seaborn.axisgrid.PairGrid at 0x7f36990ad250>
```



[26]:
```
sns.heatmap(casas.corr(numeric_only = True))
```

[26]:
```
<AxesSubplot: >
```



[27]:
```
casas. head(2)
```

[27]:
```
   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

   Avg. Area Number of Bedrooms  Area Population         Price  \
0                          4.09     23086.800503  1.059034e+06
1                          3.09     40173.072174  1.505891e+06

                                             Address
0  208 Michael Ferry Apt. 674\nLaurabury, NE 3701…
1  188 Johnson Views Suite 079\nLake Kathleen, CA…
```

[28]:
```
X = casas[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of␣
↪Rooms',
               'Avg. Area Number of Bedrooms', 'Area Population']]
```

```python
Y = casas['Price']
```

[29]:

```python
from sklearn.model_selection import train_test_split

X_treino, X_teste, y_treino, y_teste = train_test_split(X, Y, train_size=0.7,␣
 ↪random_state=50)
```

[30]:

```python
from sklearn.linear_model import LinearRegression
```

[31]:

```python
LR = LinearRegression()
```

[32]:

```python
LR.fit(X_treino, y_treino)
```

[32]:

```
LinearRegression()
```

[33]:

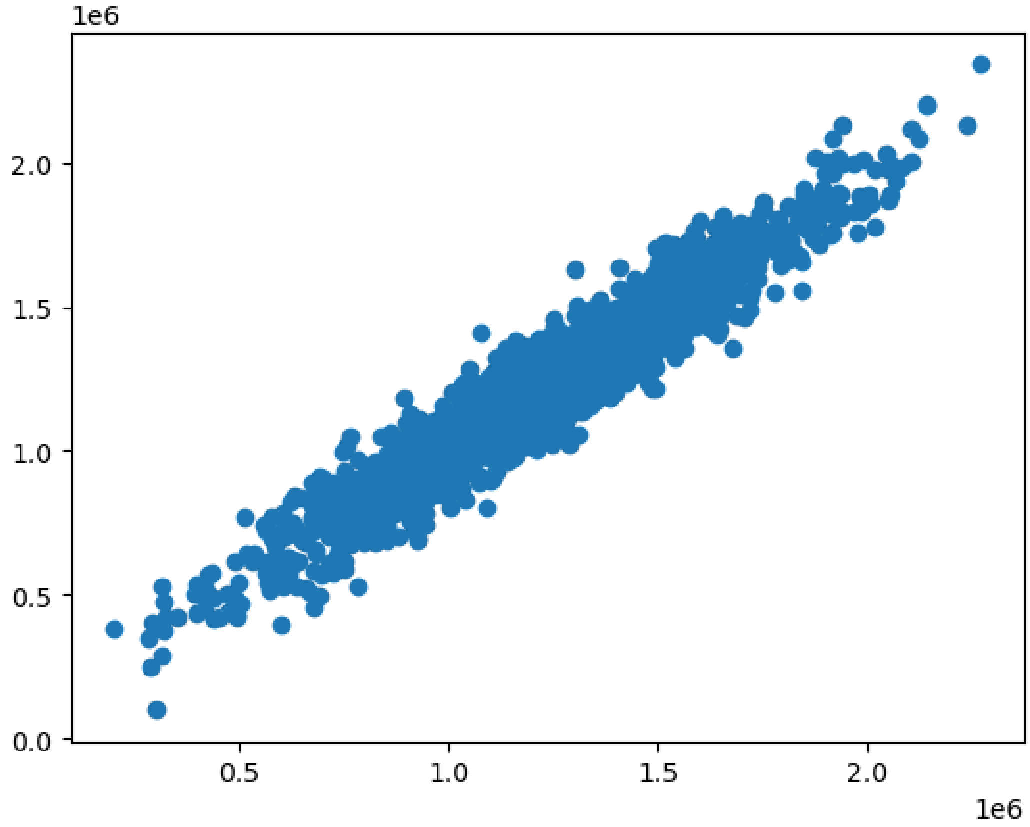
```python
predição = LR.predict(X_teste)
```

[34]:

```python
plt.scatter(y_teste, predição)
```

[34]:

```
<matplotlib.collections.PathCollection at 0x7f372c6e44d0>
```



[35]:

```python
from sklearn.metrics import mean_absolute_error,mean_squared_error

mae = mean_absolute_error(y_true=y_teste,y_pred=predição)
#squared True returns MSE value, False returns RMSE value.
mse = mean_squared_error(y_true=y_teste,y_pred=predição) #default=True
rmse = mean_squared_error(y_true=y_teste,y_pred=predição,squared=False)

print("MAE:",mae)
print("MSE:",mse)
print("RMSE:",rmse)
```

```
MAE: 80728.93384538879
MSE: 10077066685.864893
RMSE: 100384.59386711137
```

0.4 Exercício

Crie e treine um modelo de regressão linear para prever a quantidade de banheiros que corresponde à variável alvo (Avg. Area Number of Bedrooms), utilizando a quantidade de quartos presentes nas casas (Avg. Area Number of Rooms) da base de dados USA_Housing.csv.

[ ]:

```
```