LinearRegression

In [1]:

```
import numpy as np
import pandas as pd
```

data collection

In [2]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as pp
import seaborn as sb
```

In [3]:

```
df = pd.read_csv(r"C:\Users\user\Desktop\10_USA_Housing.csv")
df
```

Out[3]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	Addres
0	79545.458574	5.682861	7.009188	4.09	23086.800503	1.059034e+06	208 Michael Ferry Ap 674\nLaurabury, N 3701
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06	188 Johnson View Suite 079\nLak Kathleen, CA
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06	9127 Elizabet Stravenue\nDanieltow WI 06482
3	63345.240046	7.188236	5.586729	3.26	34310.242831	1.260617e+06	USS Barnett\nFPO A 4482
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05	USNS Raymond\nFP AE 0938
...	...	...	...	...	...	...	
4995	60567.944140	7.830362	6.137356	3.46	22837.361035	1.060194e+06	USNS Williams\nFP AP 30153-765
4996	78491.275435	6.999135	6.576763	4.02	25616.115489	1.482618e+06	PSC 9258, Bo 8489\nAPO AA 4299 335
4997	63390.686886	7.250591	4.805081	2.13	33266.145490	1.030730e+06	4215 Tracy Garde Suite 076\nJoshualan VA 01
4998	68001.331235	5.534388	7.130144	5.44	42625.620156	1.198657e+06	USS Wallace\nFPO A 7331
4999	65510.581804	5.992305	6.792336	4.07	46501.283803	1.298950e+06	37778 George Ridge Apt. 509\nEast Holl NV 2

5000 rows × 7 columns

first 10 rows

In [4]:

```
df.head(10)
```

Out[4]:

	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
5	80175.754159	4.988408	6.104512	4.04	26748.428425	1.068138e+06	06039 Jennifer Islands Apt. 443\nTracyport, KS...
6	64698.463428	6.025336	8.147760	3.41	60828.249085	1.502056e+06	4759 Daniel Shoals Suite 442\nNguyenburgh, CO ...
7	78394.339278	6.989780	6.620478	2.42	36516.358972	1.573937e+06	972 Joyce Viaduct\nLake William, TN 17778-6483
8	59927.660813	5.362126	6.393121	2.30	29387.396003	7.988695e+05	USS Gilbert\nFPO AA 20957
9	81885.927184	4.423672	8.167688	6.10	40149.965749	1.545155e+06	Unit 9446 Box 0958\nDPO AE 97025

data cleaning

In [6]:

```
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
```

In [7]:

```
df.describe()
```

Out[7]:

	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

In [9]:

```
df.columns
```

Out[9]:

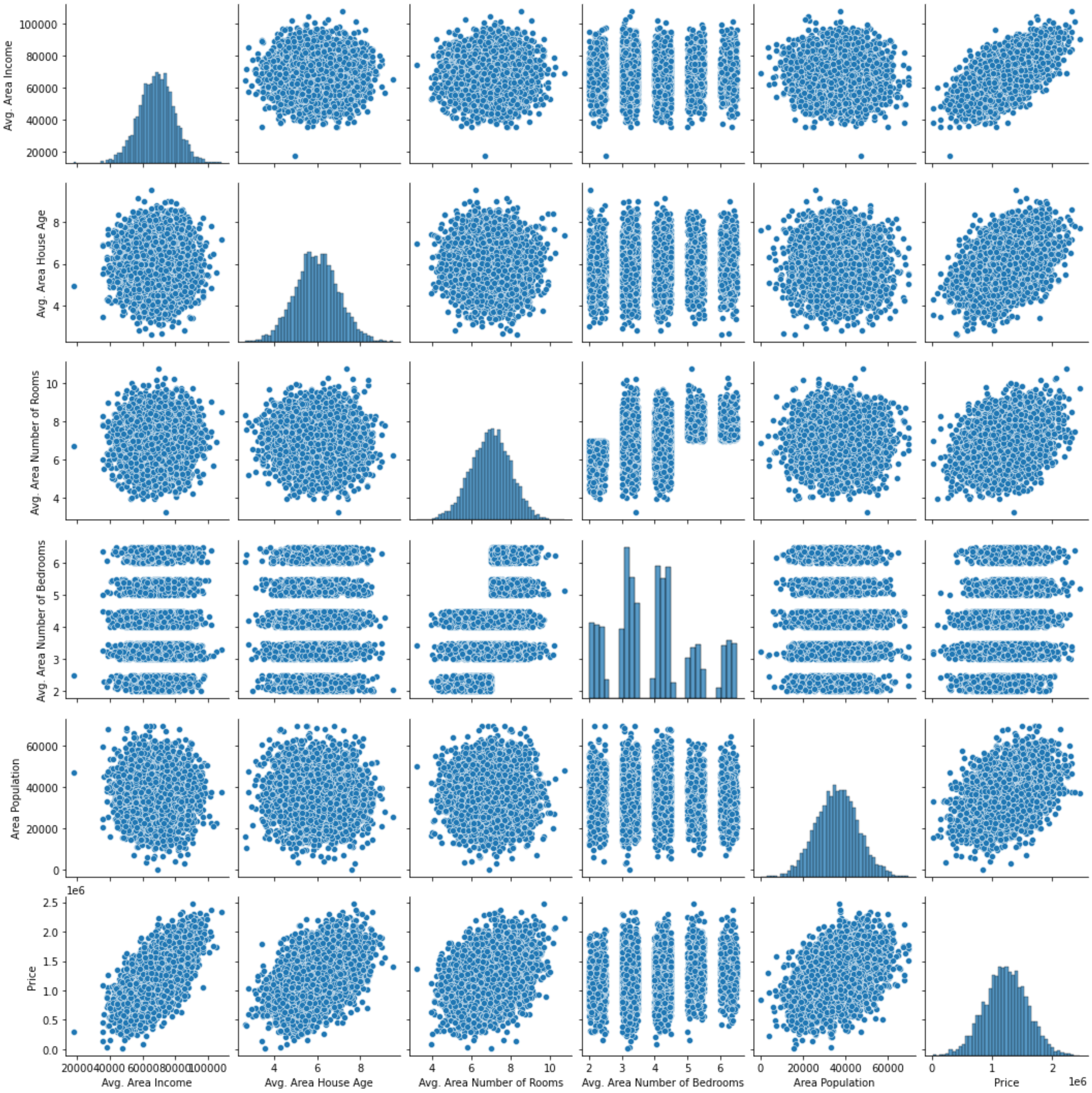
```
Index(['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
       'Avg. Area Number of Bedrooms', 'Area Population', 'Price', 'Address'],
      dtype='object')
```

In [10]:

```
sb.pairplot(df)
```

Out[10]:

```
<seaborn.axisgrid.PairGrid at 0x2590edcc580>
```

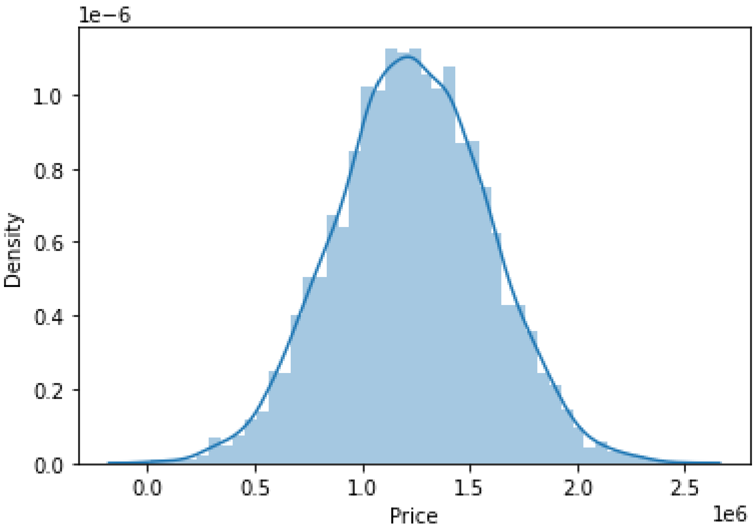


In [16]:

```
sb.distplot(df["Price"])
```

Out[16]:

```
<AxesSubplot:xlabel='Price', ylabel='Density'>
```



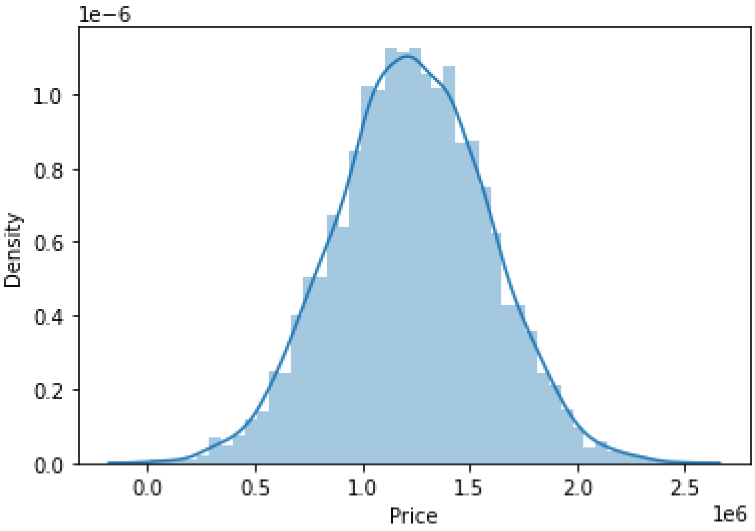
In [17]:

```
sb.distplot(df["Price"])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarn
ing: `distplot` is a deprecated function and will be removed in a future version. Pl
ease adapt your code to use either `displot` (a figure-level function with similar f
lexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[17]:

```
<AxesSubplot:xlabel='Price', ylabel='Density'>
```



In [20]:

```
df1=df[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
       'Avg. Area Number of Bedrooms', 'Area Population', 'Price', 'Address']]
df1
```

Out[20]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	Addres
0	79545.458574	5.682861	7.009188	4.09	23086.800503	1.059034e+06	208 Michael Ferry Ap 674\nLaurabury, N 3701
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06	188 Johnson View Suite 079\nLak Kathleen, CA
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06	9127 Elizabet Stravenue\nDanieltow WI 06482
3	63345.240046	7.188236	5.586729	3.26	34310.242831	1.260617e+06	USS Barnett\nFPO A 4482
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05	USNS Raymond\nFP AE 0938
...	...	...	...	...	...	...	
4995	60567.944140	7.830362	6.137356	3.46	22837.361035	1.060194e+06	USNS Williams\nFP AP 30153-765
4996	78491.275435	6.999135	6.576763	4.02	25616.115489	1.482618e+06	PSC 9258, Bo 8489\nAPO AA 4299 335
4997	63390.686886	7.250591	4.805081	2.13	33266.145490	1.030730e+06	4215 Tracy Garde Suite 076\nJoshualan VA 01

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	Addre
4998	68001.331235	5.534388	7.130144	5.44	42625.620156	1.198657e+06	USS Wallace\nFPO A 7331
4999	65510.581804	5.992305	6.792336	4.07	46501.283803	1.298950e+06	37778 George Ridge Apt. 509\nEast Holl NV 2

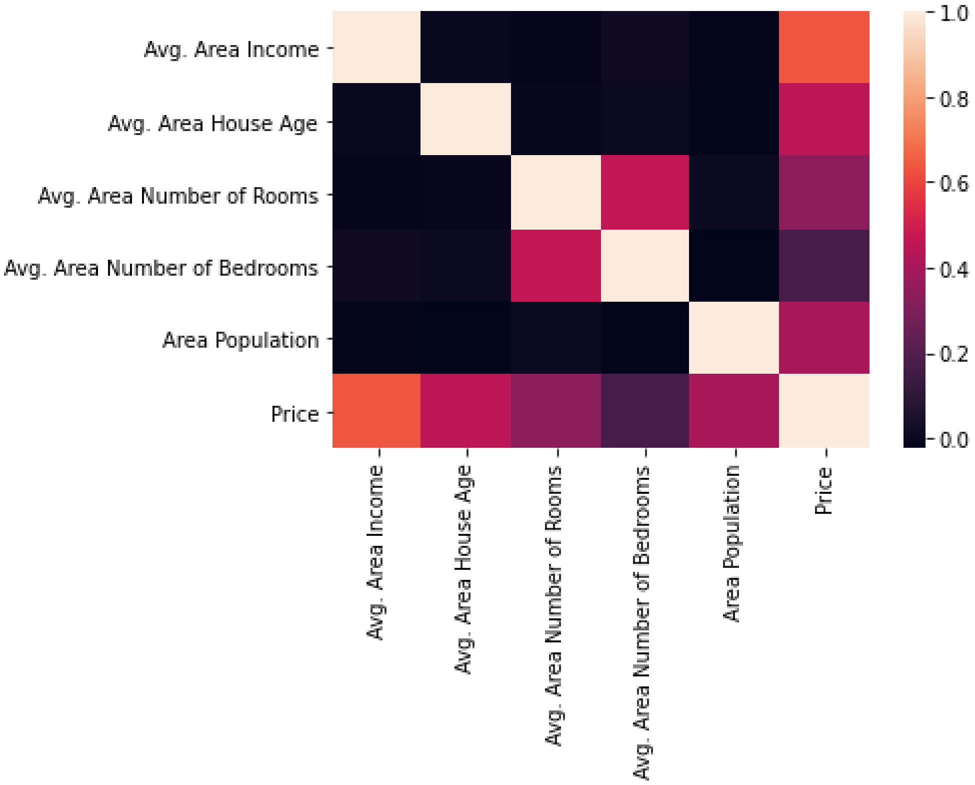
5000 rows × 7 columns

In [21]:

```
sb.heatmap(df1.corr())
```

Out[21]:

```
<AxesSubplot:>
```



model building

In [34]:

```
x = df1[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
       'Avg. Area Number of Bedrooms', 'Area Population', 'Price']]
y = df1['Price']
```

In [35]:

```
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 [36]:

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

```
lr = LinearRegression()
lr.fit(x_train,y_train)
```

Out[36]:
```
LinearRegression()
```

In [38]:
```
print(lr.intercept_)
```

```
-2.3283064365386963e-10
```

In [42]:
```
coef = pd.DataFrame(lr.coef_,x.columns,columns=['Co_efficient'])
coef
```

Out[42]:

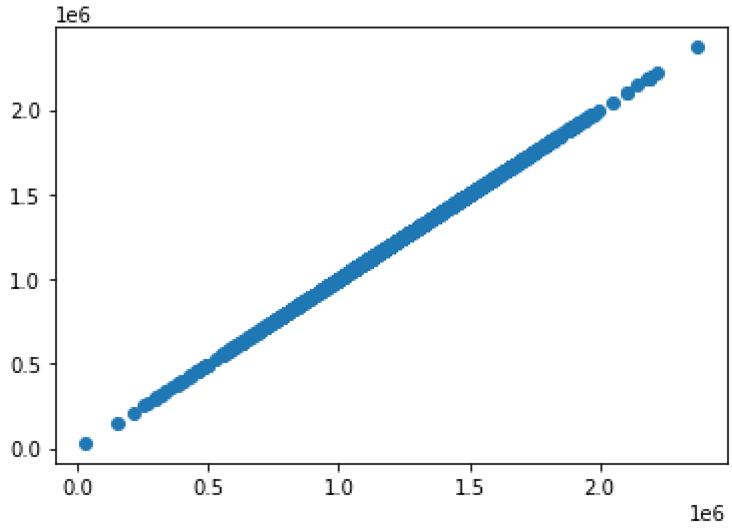
	Co_efficient
Avg. Area Income	3.184052e-15
Avg. Area House Age	-5.061788e-11
Avg. Area Number of Rooms	7.996921e-11
Avg. Area Number of Bedrooms	1.311991e-12
Area Population	9.261990e-15
Price	1.000000e+00

In [43]:
```
print(lr.score(x_test,y_test))
```

```
1.0
```

In [44]:
```
prediction = lr.predict(x_test)
pp.scatter(y_test,prediction)
```

Out[44]:
```
<matplotlib.collections.PathCollection at 0x25913950f40>
```



In [ ]:

In [1]:

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

In [9]:

```
import seaborn as sb
```

In [10]:

```
df = pd.read_csv(r"C:\Users\user\Desktop\9_bottle.csv")
df
```

```
C:\ProgramData\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3165: Dt
ypeWarning: Columns (47,73) have mixed types.Specify dtype option on import or set l
ow_memory=False.
  has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
```

Out[10]:

	Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Salnty	O2ml_L	STheta	O2Sat	..
0	1	1	054.0 056.0	19-4903CR-HY-060-0930-05400560-0000A-3	0	10.500	33.4400	NaN	25.64900	NaN	..
1	1	2	054.0 056.0	19-4903CR-HY-060-0930-05400560-0008A-3	8	10.460	33.4400	NaN	25.65600	NaN	..
2	1	3	054.0 056.0	19-4903CR-HY-060-0930-05400560-0010A-7	10	10.460	33.4370	NaN	25.65400	NaN	..
3	1	4	054.0 056.0	19-4903CR-HY-060-0930-05400560-0019A-3	19	10.450	33.4200	NaN	25.64300	NaN	..
4	1	5	054.0 056.0	19-4903CR-HY-060-0930-05400560-0020A-7	20	10.450	33.4210	NaN	25.64300	NaN	..
...	...	...	...	...	...	...	...	...	...	...	..
864858	34404	864859	093.4 026.4	20-1611SR-MX-310-2239-09340264-0000A-7	0	18.744	33.4083	5.805	23.87055	108.74	..

	Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Salnty	O2ml_L	STheta	O2Sat	..
864859	34404	864860	093.4 026.4	20-1611SR-MX-310-2239-09340264-0002A-3	2	18.744	33.4083	5.805	23.87072	108.74	..
864860	34404	864861	093.4 026.4	20-1611SR-MX-310-2239-09340264-0005A-3	5	18.692	33.4150	5.796	23.88911	108.46	..
864861	34404	864862	093.4 026.4	20-1611SR-MX-310-2239-09340264-0010A-3	10	18.161	33.4062	5.816	24.01426	107.74	..
864862	34404	864863	093.4 026.4	20-1611SR-MX-310-2239-09340264-0015A-3	15	17.533	33.3880	5.774	24.15297	105.66	..

864863 rows × 74 columns

In [11]:

```
df.head(10)
```

Out[11]:

	Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Salnty	O2ml_L	STheta	O2Sat	...	R_PHA
0	1	1	054.0 056.0	19-4903CR-HY-060-0930-05400560-0000A-3	0	10.50	33.440	NaN	25.649	NaN	...	N
1	1	2	054.0 056.0	19-4903CR-HY-060-0930-05400560-0008A-3	8	10.46	33.440	NaN	25.656	NaN	...	N
2	1	3	054.0 056.0	19-4903CR-HY-060-0930-05400560-0010A-7	10	10.46	33.437	NaN	25.654	NaN	...	N
3	1	4	054.0 056.0	19-4903CR-	19	10.45	33.420	NaN	25.643	NaN	...	N

	Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Salnty	O2ml_L	STheta	O2Sat	...	R_PHA
				HY-060-0930-05400560-0019A-3								
4	1	5	054.0 056.0	19-4903CR-HY-060-0930-05400560-0020A-7	20	10.45	33.421	NaN	25.643	NaN	...	N
5	1	6	054.0 056.0	19-4903CR-HY-060-0930-05400560-0030A-7	30	10.45	33.431	NaN	25.651	NaN	...	N
6	1	7	054.0 056.0	19-4903CR-HY-060-0930-05400560-0039A-3	39	10.45	33.440	NaN	25.658	NaN	...	N
7	1	8	054.0 056.0	19-4903CR-HY-060-0930-05400560-0050A-7	50	10.24	33.424	NaN	25.682	NaN	...	N
8	1	9	054.0 056.0	19-4903CR-HY-060-0930-05400560-0058A-3	58	10.06	33.420	NaN	25.710	NaN	...	N
9	1	10	054.0 056.0	19-4903CR-HY-060-0930-05400560-0075A-7	75	9.86	33.494	NaN	25.801	NaN	...	N

10 rows × 74 columns

In [12]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 864863 entries, 0 to 864862
Data columns (total 74 columns):
 #   Column                Non-Null Count   Dtype  
---  ------                --------------   -----  
 0   Cst_Cnt               864863 non-null  int64  
 1   Btl_Cnt               864863 non-null  int64  
 2   Sta_ID                864863 non-null  object 
 3   Depth_ID              864863 non-null  object 
```

```
4   Depthm               864863 non-null  int64  
5   T_degC               853900 non-null  float64
6   Salnty               817509 non-null  float64
7   O2ml_L               696201 non-null  float64
8   STheta               812174 non-null  float64
9   O2Sat                661274 non-null  float64
10  Oxy_µmol/Kg          661268 non-null  float64
11  BtlNum               118667 non-null  float64
12  RecInd               864863 non-null  int64  
13  T_prec               853900 non-null  float64
14  T_qual               23127 non-null   float64
15  S_prec               817509 non-null  float64
16  S_qual               74914 non-null   float64
17  P_qual               673755 non-null  float64
18  O_qual               184676 non-null  float64
19  SThtaq               65823 non-null   float64
20  O2Satq               217797 non-null  float64
21  ChlorA               225272 non-null  float64
22  Chlqua               639166 non-null  float64
23  Phaeop               225271 non-null  float64
24  Phaqua               639170 non-null  float64
25  PO4uM                413317 non-null  float64
26  PO4q                 451786 non-null  float64
27  SiO3uM               354091 non-null  float64
28  SiO3qu               510866 non-null  float64
29  NO2uM                337576 non-null  float64
30  NO2q                 529474 non-null  float64
31  NO3uM                337403 non-null  float64
32  NO3q                 529933 non-null  float64
33  NH3uM                64962 non-null   float64
34  NH3q                 808299 non-null  float64
35  C14As1               14432 non-null   float64
36  C14A1p               12760 non-null   float64
37  C14A1q               848605 non-null  float64
38  C14As2               14414 non-null   float64
39  C14A2p               12742 non-null   float64
40  C14A2q               848623 non-null  float64
41  DarkAs               22649 non-null   float64
42  DarkAp               20457 non-null   float64
43  DarkAq               840440 non-null  float64
44  MeanAs               22650 non-null   float64
45  MeanAp               20457 non-null   float64
46  MeanAq               840439 non-null  float64
47  IncTim               14437 non-null   object 
48  LightP               18651 non-null   float64
49  R_Depth              864863 non-null  float64
50  R_TEMP               853900 non-null  float64
51  R_POTEMP             818816 non-null  float64
52  R_SALINITY           817509 non-null  float64
53  R_SIGMA              812007 non-null  float64
54  R_SVA                812092 non-null  float64
55  R_DYNHT              818206 non-null  float64
56  R_O2                 696201 non-null  float64
57  R_O2Sat              666448 non-null  float64
58  R_SIO3               354099 non-null  float64
59  R_PO4                413325 non-null  float64
60  R_NO3                337411 non-null  float64
61  R_NO2                337584 non-null  float64
62  R_NH4                64982 non-null   float64
63  R_CHLA               225276 non-null  float64
64  R_PHAEO              225275 non-null  float64
65  R_PRES               864863 non-null  int64  
66  R_SAMP               122006 non-null  float64
67  DIC1                 1999 non-null    float64
68  DIC2                 224 non-null     float64
69  TA1                  2084 non-null    float64
70  TA2                  234 non-null     float64
71  pH2                  10 non-null      float64
72  pH1                  84 non-null      float64
```

```
 73  DIC Quality Comment  55 non-null      object
dtypes: float64(65), int64(5), object(4)
memory usage: 488.3+ MB
```

In [13]:

```
df.describe()
```

Out[13]:

	Cst_Cnt	Btl_Cnt	Depthm	T_degC	Salnty	O2ml_L	
count	864863.000000	864863.000000	864863.000000	853900.000000	817509.000000	696201.000000	8
mean	17138.790958	432432.000000	226.831951	10.799677	33.840350	3.392468	
std	10240.949817	249664.587267	316.050259	4.243825	0.461843	2.073256	
min	1.000000	1.000000	0.000000	1.440000	28.431000	-0.010000	
25%	8269.000000	216216.500000	46.000000	7.680000	33.488000	1.360000	
50%	16848.000000	432432.000000	125.000000	10.060000	33.863000	3.440000	
75%	26557.000000	648647.500000	300.000000	13.880000	34.196900	5.500000	
max	34404.000000	864863.000000	5351.000000	31.140000	37.034000	11.130000	

8 rows × 70 columns

In [14]:

```
df.columns
```

Out[14]:

```
Index(['Cst_Cnt', 'Btl_Cnt', 'Sta_ID', 'Depth_ID', 'Depthm', 'T_degC',
       'Salnty', 'O2ml_L', 'STheta', 'O2Sat', 'Oxy_µmol/Kg', 'BtlNum',
       'RecInd', 'T_prec', 'T_qual', 'S_prec', 'S_qual', 'P_qual', 'O_qual',
       'SThtaq', 'O2Satq', 'ChlorA', 'Chlqua', 'Phaeop', 'Phaqua', 'PO4uM',
       'PO4q', 'SiO3uM', 'SiO3qu', 'NO2uM', 'NO2q', 'NO3uM', 'NO3q', 'NH3uM',
       'NH3q', 'C14As1', 'C14A1p', 'C14A1q', 'C14As2', 'C14A2p', 'C14A2q',
       'DarkAs', 'DarkAp', 'DarkAq', 'MeanAs', 'MeanAp', 'MeanAq', 'IncTim',
       'LightP', 'R_Depth', 'R_TEMP', 'R_POTEMP', 'R_SALINITY', 'R_SIGMA',
       'R_SVA', 'R_DYNHT', 'R_O2', 'R_O2Sat', 'R_SIO3', 'R_PO4', 'R_NO3',
       'R_NO2', 'R_NH4', 'R_CHLA', 'R_PHAEO', 'R_PRES', 'R_SAMP', 'DIC1',
       'DIC2', 'TA1', 'TA2', 'pH2', 'pH1', 'DIC Quality Comment'],
      dtype='object')
```

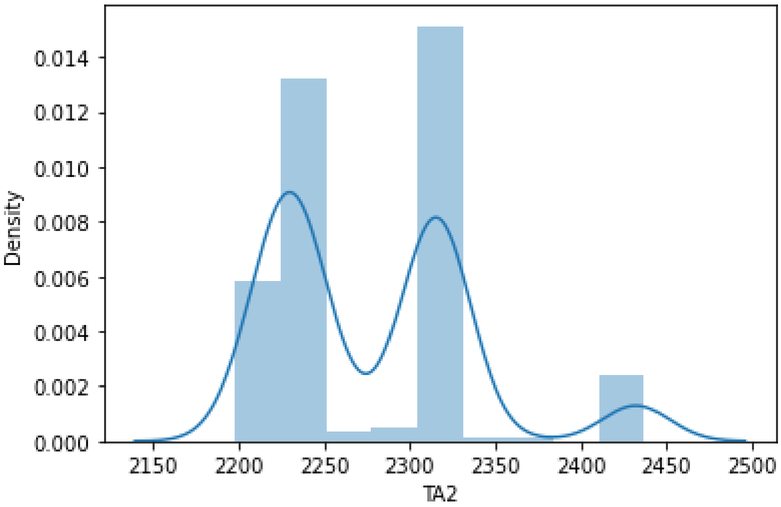
In [15]:

```
sb.distplot(df["TA2"])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarn
ing: `distplot` is a deprecated function and will be removed in a future version. Pl
ease adapt your code to use either `displot` (a figure-level function with similar f
lexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

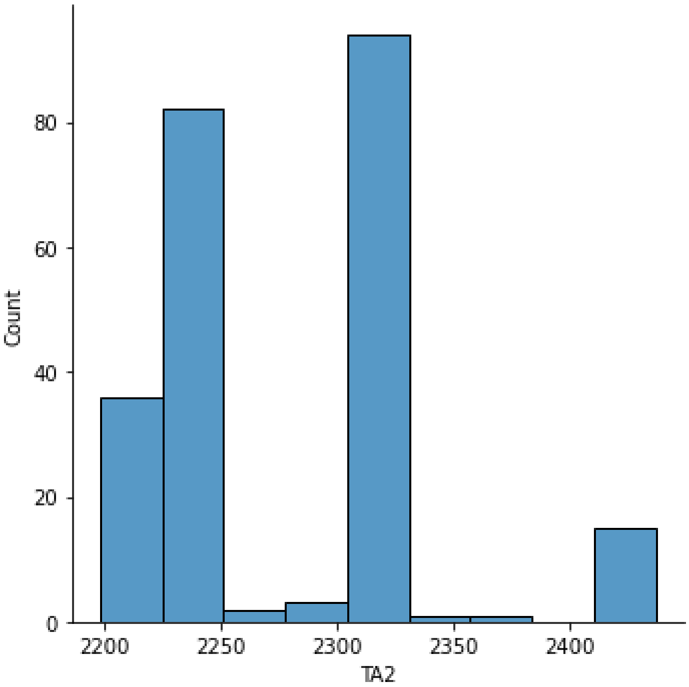
Out[15]:

```
<AxesSubplot:xlabel='TA2', ylabel='Density'>
```



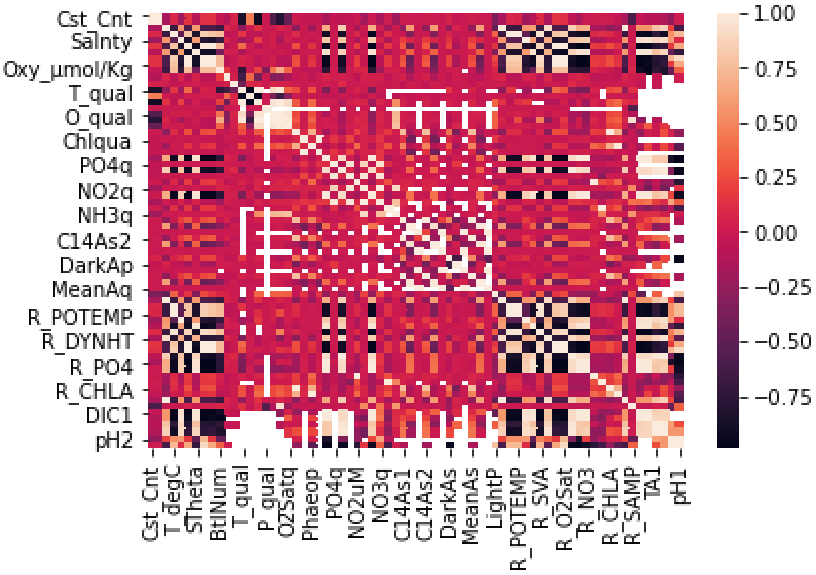
```
In [17]: sb.displot(df["TA2"])
```

```
Out[17]: <seaborn.axisgrid.FacetGrid at 0x21218412160>
```



```
In [19]: sb.heatmap(df.corr())
```

```
Out[19]: <AxesSubplot:>
```


Cst_Cnt
Salnty
Oxy_µmol/Kg
T_qual
O_qual
Chlqua
PO4q
NO2q
NH3q
C14As2
DarkAp
MeanAq
R_POTEMP
R_DYNHT
R_PO4
R_CHLA
DIC1
pH2
Cst_Cnt
T_degC
STheta
BtlNum
T_qual
P_qual
O2Satq
Phaeop
PO4q
NO2uM
NO3q
C14As1
C14As2
DarkAs
MeanAs
LightP
R_POTEMP
R_SVA
R_O2Sat
R_NO3
R_CHLA
R_SAMP
TA1
pH1
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75

```
In [ ]:
```