

In [1]:

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

In [2]:

```
df1=pd.read_csv(r'C:\Users\user\Downloads\9_bottle.csv')  
df1
```

```
C:\ProgramData\Anaconda3\lib\site-packages\IPython\core\interactiveshell.p  
y:3165: DtypeWarning: Columns (47,73) have mixed types.Specify dtype optio  
n on import or set low_memory=False.  
    has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
```

Out[2]:

Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Salnty	O2ml_L	STheta	(
0	1	1	054.0 056.0 19-4903CR-HY-060-0930-05400560-0000A-3	0	10.500	33.4400	NaN	25.64900	
1	1	2	054.0 056.0 19-4903CR-HY-060-0930-05400560-0008A-3	8	10.460	33.4400	NaN	25.65600	
2	1	3	054.0 056.0 19-4903CR-HY-060-0930-05400560-0010A-7	10	10.460	33.4370	NaN	25.65400	
3	1	4	054.0 056.0 19-4903CR-HY-060-0930-05400560-0019A-3	19	10.450	33.4200	NaN	25.64300	
4	1	5	054.0 056.0 19-4903CR-HY-060-0930-05400560-0020A-7	20	10.450	33.4210	NaN	25.64300	
...	
864858	34404	864859	093.4 026.4 20-1611SR-MX-310-2239-09340264-0000A-7	0	18.744	33.4083	5.805	23.87055	1
864859	34404	864860	093.4 026.4 20-1611SR-MX-310-2239-09340264-0002A-3	2	18.744	33.4083	5.805	23.87072	1
864860	34404	864861	093.4 026.4 20-1611SR-MX-310-2239-09340264-0005A-3	5	18.692	33.4150	5.796	23.88911	1
864861	34404	864862	093.4 026.4 20-1611SR-MX-310-2239-09340264-0010A-3	10	18.161	33.4062	5.816	24.01426	1

In [3]: Cst_Cnt Btl_Cnt Sta_ID Depth_ID Depthm T_degC Salnty O2ml_L STheta (

df=df1.head(20)

df				20- 1611SR- MX-310- 2239- 09340264- 0015A-3						
864862	34404	864863	093.4 026.4		15	17.533	33.3880	5.774	24.15297	1

864863 rows × 74 columns

Out[3]:

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	10.50	33.440	NaN	25.649	NaN
1	1	2	054.0 056.0 19-4903CR-HY-060-0930-05400560-0008A-3	8	10.46	33.440	NaN	25.656	NaN
2	1	3	054.0 056.0 19-4903CR-HY-060-0930-05400560-0010A-7	10	10.46	33.437	NaN	25.654	NaN
3	1	4	054.0 056.0 19-4903CR-HY-060-0930-05400560-0019A-3	19	10.45	33.420	NaN	25.643	NaN
4	1	5	054.0 056.0 19-4903CR-HY-060-0930-05400560-0020A-7	20	10.45	33.421	NaN	25.643	NaN
5	1	6	054.0 056.0 19-4903CR-HY-060-0930-05400560-0030A-7	30	10.45	33.431	NaN	25.651	NaN
6	1	7	054.0 056.0 19-4903CR-HY-060-0930-05400560-0039A-3	39	10.45	33.440	NaN	25.658	NaN
7	1	8	054.0 056.0 19-4903CR-HY-060-0930-05400560-0050A-7	50	10.24	33.424	NaN	25.682	NaN
8	1	9	054.0 056.0 19-4903CR-HY-060-0930-05400560-0058A-3	58	10.06	33.420	NaN	25.710	NaN

Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Salnty	O2ml_L	STheta	O2Sat	
9	1	10	054.0 056.0 19-4903CR-HY-060-0930-05400560-0075A-7	75	9.86	33.494	NaN	25.801	NaN	
10	1	11	054.0 056.0 19-4903CR-HY-060-0930-05400560-0078A-3	78	9.83	33.510	NaN	25.819	NaN	
11	1	12	054.0 056.0 19-4903CR-HY-060-0930-05400560-0100A-7	100	9.67	33.580	NaN	25.900	NaN	
12	1	13	054.0 056.0 19-4903CR-HY-060-0930-05400560-0117A-3	117	9.50	33.640	NaN	25.975	NaN	
13	1	14	054.0 056.0 19-4903CR-HY-060-0930-05400560-0125A-7	125	9.32	33.689	NaN	26.043	NaN	
14	1	15	054.0 056.0 19-4903CR-HY-060-0930-05400560-0150A-7	150	8.76	33.847	NaN	26.256	NaN	
15	1	16	054.0 056.0 19-4903CR-HY-060-0930-05400560-0152A-3	152	8.71	33.860	NaN	26.274	NaN	
16	1	17	054.0 056.0 19-4903CR-HY-060-0930-05400560-0200A-7	200	8.53	33.876	NaN	26.315	NaN	
17	1	18	054.0 056.0 19-4903CR-HY-060-0930-05400560-0221A-3	221	8.45	NaN	NaN	NaN	NaN	
18	1	19	054.0 056.0 19-4903CR-HY-060-0930-05400560-0250A-7	250	8.26	33.926	NaN	26.396	NaN	

In [4]:

```
Cst_Cnt  Btl_Cnt  Sta_ID  Depth_ID  Depthm  T_degC  Salnty  O2ml_L  STheta  O2Sat  .  
df.info()
```

				19- 4903CR-						
19	1	20	054.0 056.0	HY-060- 0930- 05400560- 0285A-3	285	7.96	33.980	NaN	26.483	NaN

20 rows × 74 columns

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 20 entries, 0 to 19
```

```
Data columns (total 74 columns):
```

#	Column	Non-Null Count	Dtype
0	Cst_Cnt	20 non-null	int64
1	Btl_Cnt	20 non-null	int64
2	Sta_ID	20 non-null	object
3	Depth_ID	20 non-null	object
4	Depthm	20 non-null	int64
5	T_degC	20 non-null	float64
6	Salnty	19 non-null	float64
7	O2ml_L	0 non-null	float64
8	STheta	19 non-null	float64
9	O2Sat	0 non-null	float64
10	Oxy_μmol/Kg	0 non-null	float64
11	BtlNum	0 non-null	float64
12	RecInd	20 non-null	int64
13	T_prec	20 non-null	float64
14	T_qual	0 non-null	float64
15	S_prec	19 non-null	float64
16	S_qual	1 non-null	float64
17	P_qual	20 non-null	float64
18	O_qual	20 non-null	float64
19	SThtaq	1 non-null	float64
20	O2Satq	20 non-null	float64
21	ChlorA	0 non-null	float64
22	Chlqua	20 non-null	float64
23	Phaeop	0 non-null	float64
24	Phaqua	20 non-null	float64
25	PO4uM	0 non-null	float64
26	PO4q	20 non-null	float64
27	SiO3uM	0 non-null	float64
28	SiO3qu	20 non-null	float64
29	NO2uM	0 non-null	float64
30	NO2q	20 non-null	float64
31	NO3uM	0 non-null	float64
32	NO3q	20 non-null	float64
33	NH3uM	0 non-null	float64
34	NH3q	20 non-null	float64
35	C14As1	0 non-null	float64
36	C14A1p	0 non-null	float64
37	C14A1q	20 non-null	float64
38	C14As2	0 non-null	float64
39	C14A2p	0 non-null	float64
40	C14A2q	20 non-null	float64
41	DarkAs	0 non-null	float64
42	DarkAp	0 non-null	float64
43	DarkAq	20 non-null	float64
44	MeanAs	0 non-null	float64
45	MeanAp	0 non-null	float64
46	MeanAq	20 non-null	float64
47	IncTim	0 non-null	object
48	LightP	0 non-null	float64
49	R_Depth	20 non-null	float64
50	R_TEMP	20 non-null	float64
51	R_POTEMP	20 non-null	float64
52	R_SALINITY	19 non-null	float64
53	R_SIGMA	19 non-null	float64
54	R_SVA	19 non-null	float64
55	R_DYNHT	20 non-null	float64


```
56  R_O2                0 non-null    float64
57  R_O2Sat             0 non-null    float64
58  R_SIO3              0 non-null    float64
59  R_PO4               0 non-null    float64
60  R_NO3               0 non-null    float64
61  R_NO2               0 non-null    float64
62  R_NH4               0 non-null    float64
63  R_CHLA              0 non-null    float64
64  R_PHAEO             0 non-null    float64
65  R_PRES              20 non-null   int64
66  R_SAMP              0 non-null    float64
67  DIC1                0 non-null    float64
68  DIC2                0 non-null    float64
69  TA1                 0 non-null    float64
70  TA2                 0 non-null    float64
71  pH2                 0 non-null    float64
72  pH1                 0 non-null    float64
73  DIC Quality Comment 0 non-null    object
```

dtypes: float64(65), int64(5), object(4)
memory usage: 11.7+ KB

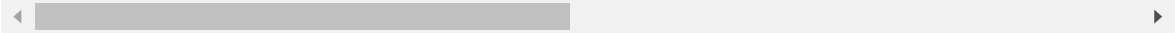
In [5]:

```
df.describe()
```

Out[5]:

	Cst_Cnt	Btl_Cnt	Depthm	T_degC	Salnty	O2ml_L	STheta	O2Sat	Oxy_
count	20.0	20.00000	20.000000	20.000000	19.000000	0.0	19.000000	0.0	
mean	1.0	10.50000	99.350000	9.618500	33.593421	NaN	25.905684	NaN	
std	0.0	5.91608	86.143867	0.872313	0.203257	NaN	0.296826	NaN	
min	1.0	1.00000	0.000000	7.960000	33.420000	NaN	25.643000	NaN	
25%	1.0	5.75000	27.500000	8.747500	33.434000	NaN	25.655000	NaN	
50%	1.0	10.50000	76.500000	9.845000	33.494000	NaN	25.801000	NaN	
75%	1.0	15.25000	150.500000	10.450000	33.768000	NaN	26.149500	NaN	
max	1.0	20.00000	285.000000	10.500000	33.980000	NaN	26.483000	NaN	

8 rows × 10 columns



In [6]:

df.columns

Out[6]:

```
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',
      'SThta', 'O2Satq', 'ChlorA', 'Chlqua', 'Phaeop', 'Phaqua', 'P04u',
      'P04q', 'SiO3uM', 'SiO3qu', 'NO2uM', 'NO2q', 'NO3uM', 'NO3q', 'NH3u',
      'NH3q', 'C14As1', 'C14A1p', 'C14A1q', 'C14As2', 'C14A2p', 'C14A2q',
      'DarkAs', 'DarkAp', 'DarkAq', 'MeanAs', 'MeanAp', 'MeanAq', 'IncTi',
      '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 [8]:

```
df2=df.dropna(axis=1)  
df2
```

Out[8]:

	Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Reclnd	T_prec	P_qual	O_qual
0	1	1	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0000A-3	0	10.50	3	1.0	9.0	9.0
1	1	2	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0008A-3	8	10.46	3	2.0	9.0	9.0
2	1	3	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0010A-7	10	10.46	7	2.0	9.0	9.0
3	1	4	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0019A-3	19	10.45	3	2.0	9.0	9.0
4	1	5	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0020A-7	20	10.45	7	2.0	9.0	9.0
5	1	6	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0030A-7	30	10.45	7	2.0	9.0	9.0
6	1	7	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0039A-3	39	10.45	3	2.0	9.0	9.0
7	1	8	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0050A-7	50	10.24	7	2.0	9.0	9.0
8	1	9	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0058A-3	58	10.06	3	2.0	9.0	9.0
9	1	10	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0075A-7	75	9.86	7	2.0	9.0	9.0

	Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Reclnd	T_prec	P_qual	O_qual
10	1	11	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0078A-3	78	9.83	3	2.0	9.0	9.0
11	1	12	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0100A-7	100	9.67	7	2.0	9.0	9.0
12	1	13	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0117A-3	117	9.50	3	2.0	9.0	9.0
13	1	14	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0125A-7	125	9.32	7	2.0	9.0	9.0
14	1	15	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0150A-7	150	8.76	7	2.0	9.0	9.0
15	1	16	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0152A-3	152	8.71	3	2.0	9.0	9.0
16	1	17	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0200A-7	200	8.53	7	2.0	9.0	9.0
17	1	18	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0221A-3	221	8.45	3	2.0	9.0	9.0
18	1	19	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0250A-7	250	8.26	7	2.0	9.0	9.0
19	1	20	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0285A-3	285	7.96	3	2.0	9.0	9.0

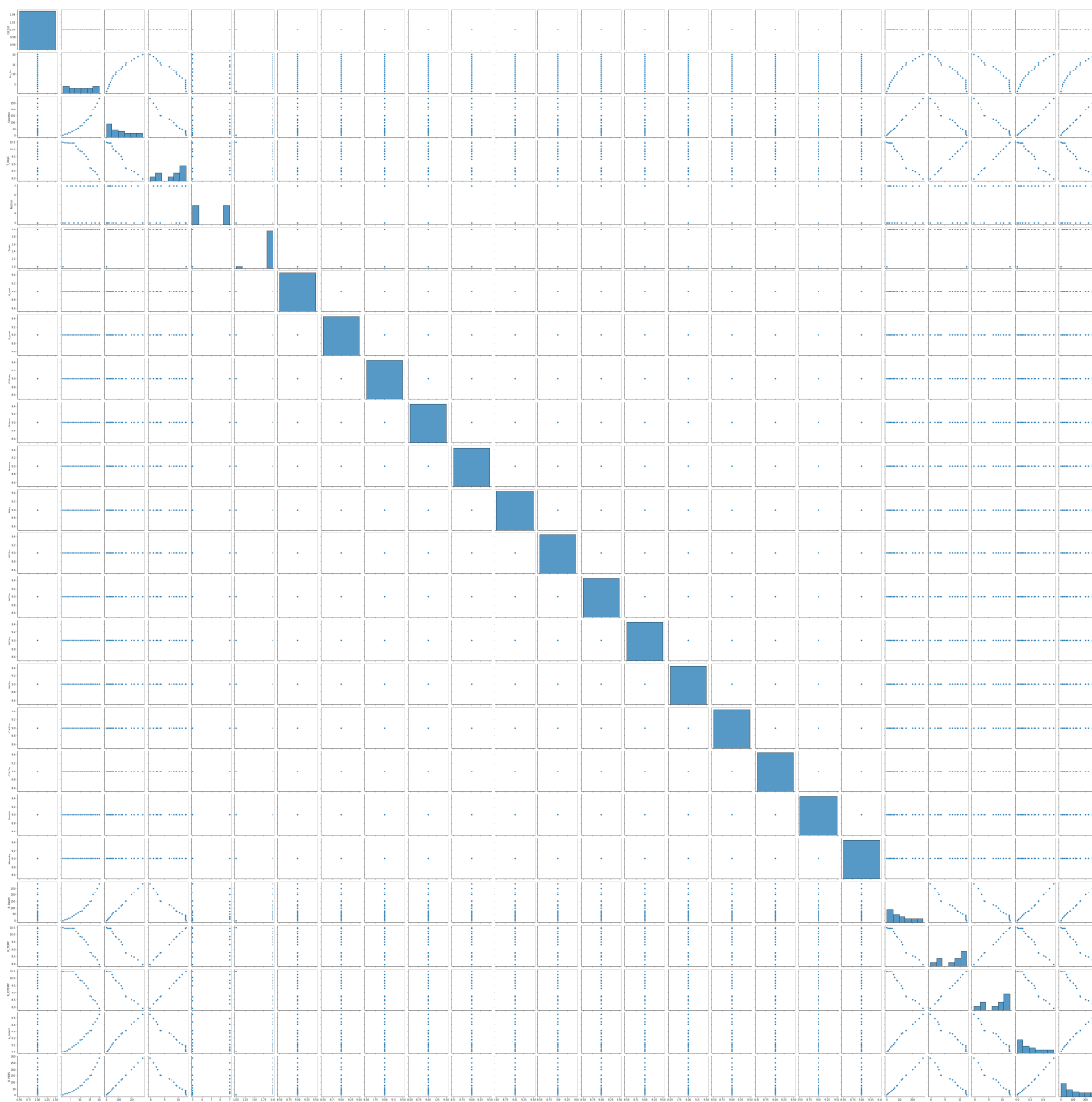
20 rows × 27 columns

In [9]:

```
sns.pairplot(df2)
```

Out[9]:

```
<seaborn.axisgrid.PairGrid at 0x1ceb37dcfa0>
```



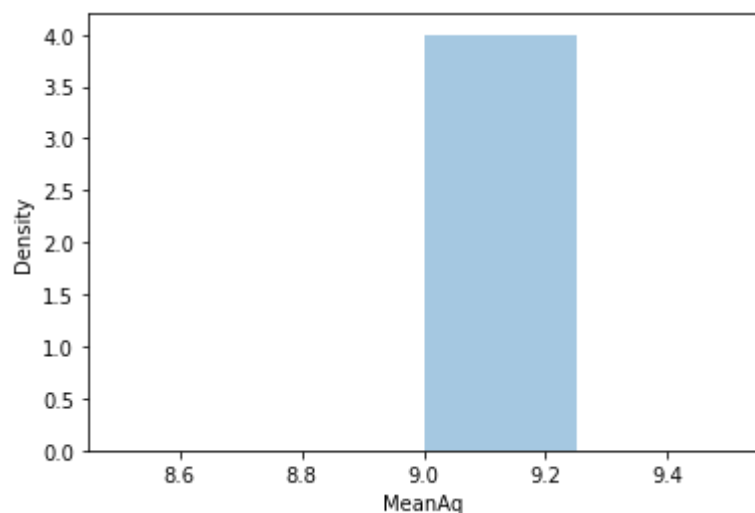
In [11]:

```
sns.distplot(df['MeanAq'])
```

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:306: UserWarning: Dataset has 0 variance; skipping density estimate.
warnings.warn(msg, UserWarning)

Out[11]:

<AxesSubplot:xlabel='MeanAq', ylabel='Density'>

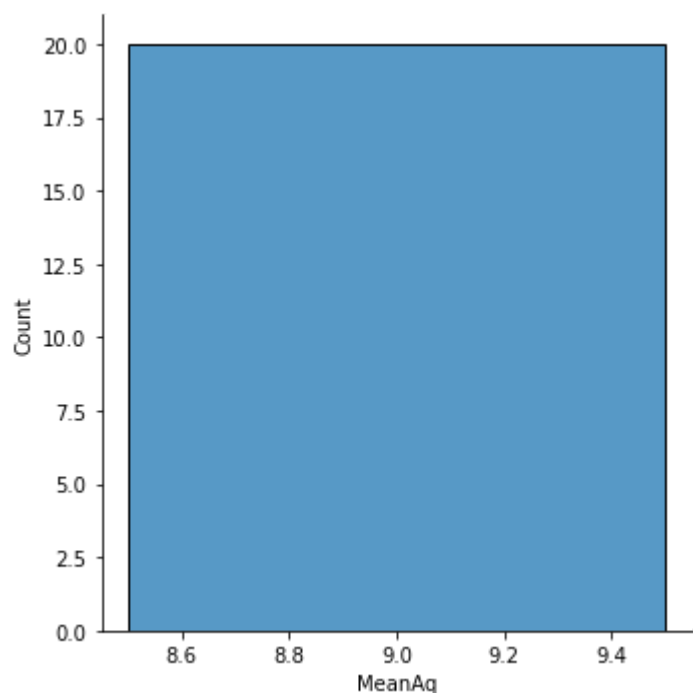


In [12]:

```
sns.displot(df["MeanAq"])
```

Out[12]:

<seaborn.axisgrid.FacetGrid at 0x1cea2089dc0>



In [13]:

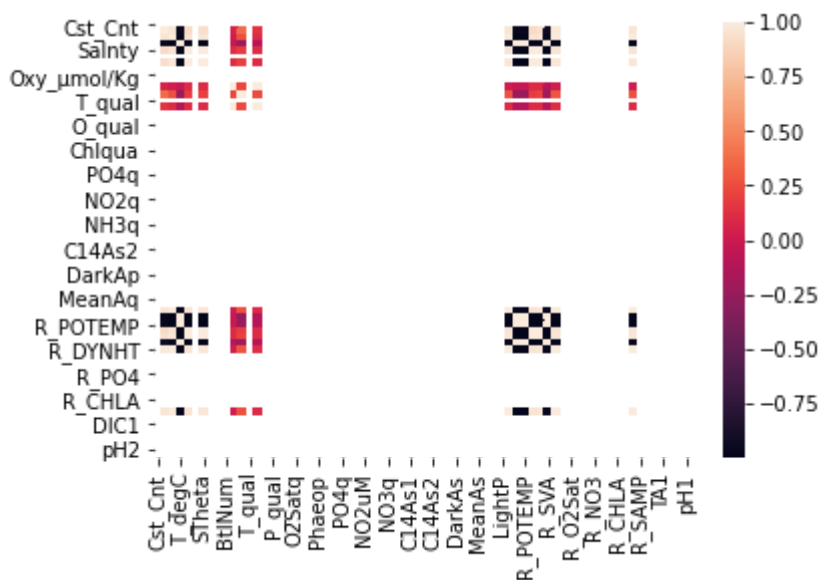
```
df1=df[['Cst_Cnt', 'Btl_Cnt', 'Sta_ID', 'Depth_ID', 'Depthm', 'T_degC',
        'Salnty', 'O2ml_L', 'STheta', 'O2Sat', 'Oxy_μmol/Kg', 'BtlNum',
        'RecInd', 'T_prec', 'T_qual', 'S_prec', 'S_qual', 'P_qual', 'O_qual',
        'SThta', 'O2Satq', 'ChlorA', 'Chlqua', 'Phaeop', 'Phaqua', 'PO4uM',
        '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']]
```

In [14]:

```
sns.heatmap(df1.corr())
```

Out[14]:

<AxesSubplot:>



In [15]:

```
df2=df.dropna(axis=1)  
df2
```

Out[15]:

	Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Reclnd	T_prec	P_qual	O_qual
0	1	1	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0000A-3	0	10.50	3	1.0	9.0	9.0
1	1	2	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0008A-3	8	10.46	3	2.0	9.0	9.0
2	1	3	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0010A-7	10	10.46	7	2.0	9.0	9.0
3	1	4	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0019A-3	19	10.45	3	2.0	9.0	9.0
4	1	5	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0020A-7	20	10.45	7	2.0	9.0	9.0
5	1	6	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0030A-7	30	10.45	7	2.0	9.0	9.0
6	1	7	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0039A-3	39	10.45	3	2.0	9.0	9.0
7	1	8	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0050A-7	50	10.24	7	2.0	9.0	9.0
8	1	9	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0058A-3	58	10.06	3	2.0	9.0	9.0
9	1	10	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0075A-7	75	9.86	7	2.0	9.0	9.0

	Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Reclnd	T_prec	P_qual	O_qual
10	1	11	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0078A-3	78	9.83	3	2.0	9.0	9.0
11	1	12	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0100A-7	100	9.67	7	2.0	9.0	9.0
12	1	13	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0117A-3	117	9.50	3	2.0	9.0	9.0
13	1	14	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0125A-7	125	9.32	7	2.0	9.0	9.0
14	1	15	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0150A-7	150	8.76	7	2.0	9.0	9.0
15	1	16	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0152A-3	152	8.71	3	2.0	9.0	9.0
16	1	17	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0200A-7	200	8.53	7	2.0	9.0	9.0
17	1	18	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0221A-3	221	8.45	3	2.0	9.0	9.0
18	1	19	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0250A-7	250	8.26	7	2.0	9.0	9.0
19	1	20	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0285A-3	285	7.96	3	2.0	9.0	9.0

20 rows × 27 columns

In [16]:

```
x=df1[['Cst_Cnt', 'RecInd', 'Depthm', 'R_PRES']]
y=df1[['Depthm']]
```

In [17]:

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

In [18]:

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3)
```

In [19]:

```
from sklearn.linear_model import LinearRegression

lr=LinearRegression()
lr.fit(x_train,y_train)#ValueError: Input contains NaN, infinity or a value too large for
```

Out[19]:

```
LinearRegression()
```

In [20]:

```
print(lr.intercept_)
```

```
[5.25801624e-13]
```

In [21]:

```
coef= pd.DataFrame(lr.coef_)
coef
```

Out[21]:

	0	1	2	3
0	0.0	-1.084755e-13	1.0	-4.487165e-14

In [22]:

```
print(lr.score(x_test,y_test))
```

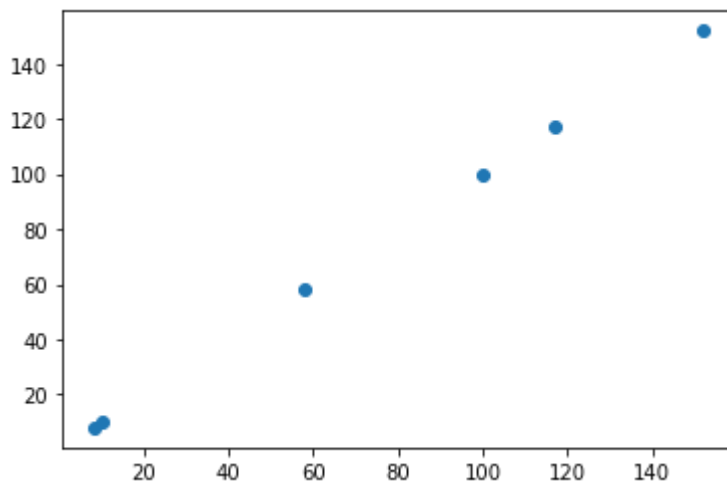
```
1.0
```

In [23]:

```
prediction = lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[23]:

<matplotlib.collections.PathCollection at 0x1cea3c1dc70>



In [24]:

```
lr.score(x_test,y_test)
```

Out[24]:

1.0

In [25]:

```
lr.score(x_train,y_train)
```

Out[25]:

1.0

In [26]:

```
from sklearn.linear_model import Ridge,Lasso
```

In [27]:

```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[27]:

Ridge(alpha=10)

In [28]:

```
rr.score(x_test,y_test)
```

Out[28]:

0.999992993908116

In [29]:

```
la=Lasso(alpha=10)  
la.fit(x_train,y_train)
```

Out[29]:

Lasso(alpha=10)

In [30]:

```
la.score(x_test,y_test)
```

Out[30]:

0.9999979994225565