D12

In [1]:

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

In [2]:

```
df=pd.read_csv(r"C:\Users\user\Downloads\9_bottle.csv").dropna(axis="columns")
df
```

```
C:\ProgramData\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3
165: DtypeWarning: Columns (47,73) have mixed types.Specify dtype option on i
mport 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	RecInd	R_Depth	R_PRES
0	1	1	054.0 056.0	19-4903CR-HY-060-0930-05400560-0000A-3	0	3	0.0	0
1	1	2	054.0 056.0	19-4903CR-HY-060-0930-05400560-0008A-3	8	3	8.0	8
2	1	3	054.0 056.0	19-4903CR-HY-060-0930-05400560-0010A-7	10	7	10.0	10
3	1	4	054.0 056.0	19-4903CR-HY-060-0930-05400560-0019A-3	19	3	19.0	19
4	1	5	054.0 056.0	19-4903CR-HY-060-0930-05400560-0020A-7	20	7	20.0	20
...	...	...	...	...	...	...	...	...
864858	34404	864859	093.4 026.4	20-1611SR-MX-310-2239-09340264-0000A-7	0	7	0.0	0
864859	34404	864860	093.4 026.4	20-1611SR-MX-310-2239-09340264-0002A-3	2	3	2.0	2
864860	34404	864861	093.4 026.4	20-1611SR-MX-310-2239-09340264-0005A-3	5	3	5.0	5
864861	34404	864862	093.4 026.4	20-1611SR-MX-310-2239-09340264-0010A-3	10	3	10.0	10
864862	34404	864863	093.4 026.4	20-1611SR-MX-310-2239-09340264-0015A-3	15	3	15.0	15

864863 rows × 8 columns

In [3]:

```
df.head()
```

Out[3]:

	Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	RecInd	R_Depth	R_PRES
0	1	1	054.0 056.0	19-4903CR-HY-060-0930-05400560-0000A-3	0	3	0.0	0
1	1	2	054.0 056.0	19-4903CR-HY-060-0930-05400560-0008A-3	8	3	8.0	8
2	1	3	054.0 056.0	19-4903CR-HY-060-0930-05400560-0010A-7	10	7	10.0	10
3	1	4	054.0 056.0	19-4903CR-HY-060-0930-05400560-0019A-3	19	3	19.0	19
4	1	5	054.0 056.0	19-4903CR-HY-060-0930-05400560-0020A-7	20	7	20.0	20

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 864863 entries, 0 to 864862
Data columns (total 8 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   RecInd    864863 non-null  int64  
 6   R_Depth   864863 non-null  float64
 7   R_PRES    864863 non-null  int64  
dtypes: float64(1), int64(5), object(2)
memory usage: 52.8+ MB
```

In [5]:

```
df.describe()
```

Out[5]:

	Cst_Cnt	Btl_Cnt	Depthm	RecInd	R_Depth	R_PRE
count	864863.000000	864863.000000	864863.000000	864863.000000	864863.000000	864863.00000
mean	17138.790958	432432.000000	226.831951	4.700273	226.832495	228.39569
std	10240.949817	249664.587267	316.050259	1.877428	316.050007	319.45673
min	1.000000	1.000000	0.000000	3.000000	0.000000	0.00000
25%	8269.000000	216216.500000	46.000000	3.000000	46.000000	46.00000
50%	16848.000000	432432.000000	125.000000	3.000000	125.000000	126.00000
75%	26557.000000	648647.500000	300.000000	7.000000	300.000000	302.00000
max	34404.000000	864863.000000	5351.000000	7.000000	5351.000000	5458.00000

In [6]:

```
df.columns
```

Out[6]:

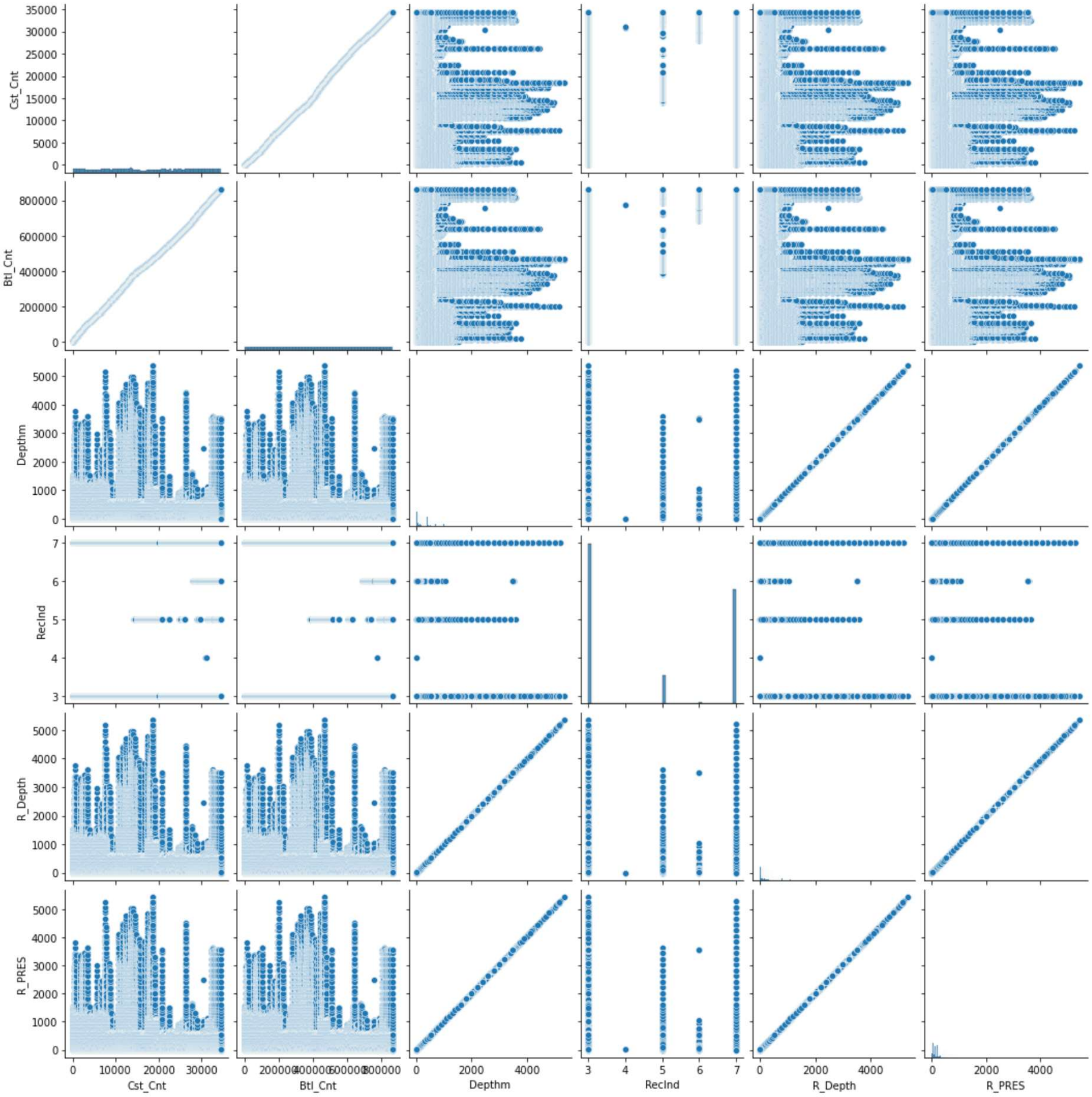
```
Index(['Cst_Cnt', 'Btl_Cnt', 'Sta_ID', 'Depth_ID', 'Depthm', 'RecInd',
       'R_Depth', 'R_PRES'],
      dtype='object')
```

In [7]:

```
sns.pairplot(df)
```

Out[7]:

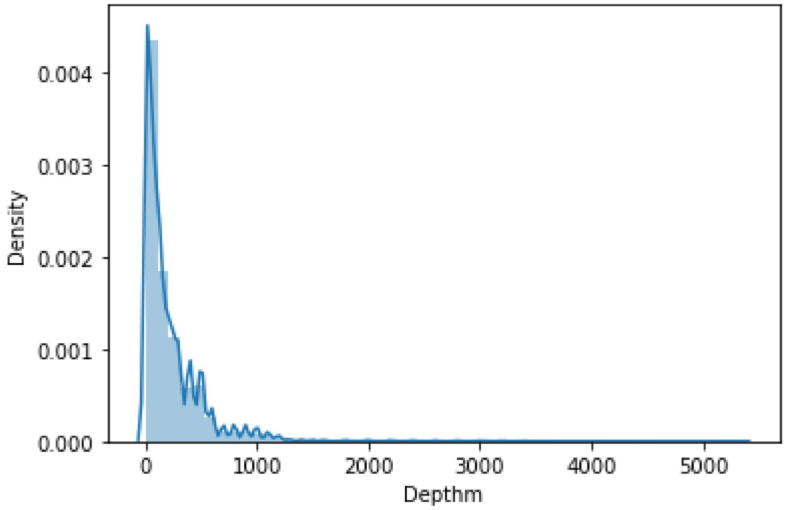
```
<seaborn.axisgrid.PairGrid at 0x26cc481cee0>
```



In [8]:
```
sns.distplot(df["Depthm"])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: Fut
ureWarning: `distplot` is a deprecated function and will be removed in a futu
re version. Please adapt your code to use either `displot` (a figure-level fu
nction with similar flexibility) or `histplot` (an axes-level function for hi
stograms).
  warnings.warn(msg, FutureWarning)
```

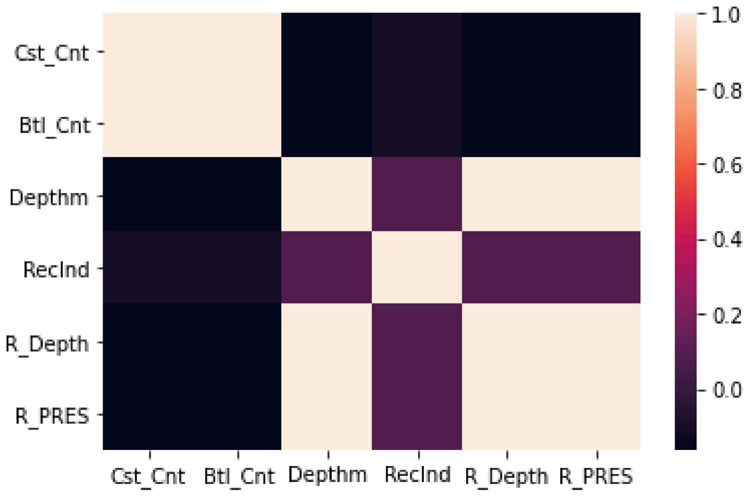
Out[8]:
```
<AxesSubplot:xlabel='Depthm', ylabel='Density'>
```



In [9]:
```
df1=df[['Cst_Cnt', 'Btl_Cnt', 'Sta_ID', 'Depth_ID','Depthm', 'RecInd',
       'R_Depth', 'R_PRES']]
```

In [10]:
```
sns.heatmap(df1.corr())
```

Out[10]:
```
<AxesSubplot:>
```



In [11]:
```
x=df1[['Cst_Cnt','Btl_Cnt','RecInd','R_Depth','R_PRES']]
y=df1["Depthm"]
```

In [12]:
```
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 [13]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[13]:
```
LinearRegression()
```

In [14]:
```
print(lr.intercept_)
```

```
0.002896043046177965
```

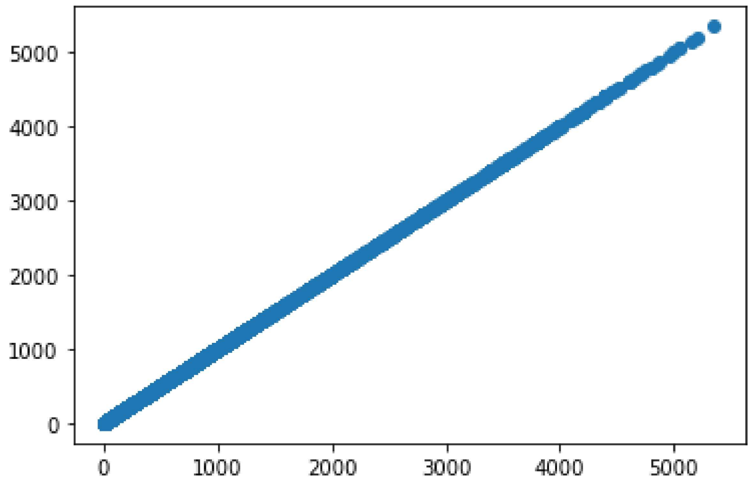
In [15]:
```
coeff = pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[15]:

	Co-efficient
Cst_Cnt	1.683472e-06
Btl_Cnt	-7.264399e-08
RecInd	-2.680345e-04
R_Depth	1.000294e+00
R_PRES	-2.901975e-04

In [16]:
```
prediction=lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[16]:
```
<matplotlib.collections.PathCollection at 0x26c8222de50>
```



In [17]:
```
print(lr.score(x_test,y_test))
```

```
0.999999994635852
```

In [18]:
```
from sklearn.linear_model import Ridge,Lasso
```

```
In [19]: rr=Ridge(alpha=10)
         rr.fit(x_train,y_train)

Out[19]: Ridge(alpha=10)
```

```
In [20]: rr.score(x_test,y_test)

Out[20]: 0.9999999946358353
```

```
In [21]: la=Lasso(alpha=10)
         la.fit(x_train,y_train)

Out[21]: Lasso(alpha=10)
```

```
In [22]: la.score(x_test,y_test)

Out[22]: 0.9999999840083624
```

```
In [24]: from sklearn.linear_model import ElasticNet
         en=ElasticNet()
         en.fit(x_train,y_train)

Out[24]: ElasticNet()
```

```
In [25]: print(en.coef_)

         [-1.67618585e-05  6.87543339e-07 -0.00000000e+00  9.99926598e-01
           6.82191618e-05]
```

```
In [26]: print(en.intercept_)

         -0.009511067960346509
```

```
In [27]: print(en.predict(x_train))

         [149.99488002 399.99057014  29.99229862 ...  49.99234836 249.99697777
          145.99229192]
```

```
In [28]: print(en.score(x_train,y_train))

         0.9999999940422679
```

```
In [29]: from sklearn import metrics
```

```
In [31]: print("Mean Absolytre Error:",metrics.mean_absolute_error(y_test,prediction))

         Mean Absolytre Error: 0.0016168858819957744
```

```
In [32]: print("Mean Square Error:",metrics.mean_squared_error(y_test,prediction))

         Mean Square Error: 0.0005415532691665863
```

In [33]:

```
print("Root Mean Square Error:",np.sqrt(metrics.mean_absolute_error(y_test,pre
```

```
Root Mean Square Error: 0.040210519543967274
```

In [ ]: