Deena 20104016

In [1]:

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

In [2]:

```
df = pd.read_csv("9_bottle.csv")
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	T_degC	Salnty	O2n
0	1	1	054.0 056.0	19-4903CR-HY-060-0930-05400560-0000A-3	0	10.500	33.4400	
1	1	2	054.0 056.0	19-4903CR-HY-060-0930-05400560-0008A-3	8	10.460	33.4400	
2	1	3	054.0 056.0	19-4903CR-HY-060-0930-05400560-0010A-7	10	10.460	33.4370	
3	1	4	054.0 056.0	19-4903CR-HY-060-0930-05400560-0019A-3	19	10.450	33.4200	
4	1	5	054.0	19-4903CR-	20	10.450	33.4210	

In [3]:

```
df.head()
```

Out[3]:

	Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Salnty	O2ml_L	S
0	1	1	054.0 056.0	19-4903CR-HY-060-0930-05400560-0000A-3	0	10.50	33.440	NaN	2
1	1	2	054.0 056.0	19-4903CR-HY-060-0930-05400560-0008A-3	8	10.46	33.440	NaN	2
2	1	3	054.0 056.0	19-4903CR-HY-060-0930-05400560-0010A-7	10	10.46	33.437	NaN	2
3	1	4	054.0 056.0	19-4903CR-HY-060-0930-05400560-0019A-3	19	10.45	33.420	NaN	2
4	1	5	054.0 056.0	19-4903CR-HY-060-0930-05400560-0020A-7	20	10.45	33.421	NaN	2

5 rows × 74 columns

In [4]:

```
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 [5]:

Out[5]:

```
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 [6]:
```
df1=df.dropna(axis=1)
df1
```

Out[6]:

	Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	RecInd	R_Depth	R_P
0	1	1	054.0 056.0	19-4903CR-HY-060-0930-05400560-0000A-3	0	3	0.0	
1	1	2	054.0 056.0	19-4903CR-HY-060-0930-05400560-0008A-3	8	3	8.0	
2	1	3	054.0 056.0	19-4903CR-HY-060-0930-05400560-0010A-7	10	7	10.0	
3	1	4	054.0 056.0	19-4903CR-HY-060-0930-05400560-0019A-3	19	3	19.0	
4	1	5	054.0 056.0	19-4903CR-HY-060-0930-05400560-0020A-7	20	7	20.0	
...	...	...	...	...	...	...	...	
864858	34404	864859	093.4 026.4	20-1611SR-MX-310-2239-09340264-0000A-7	0	7	0.0	
864859	34404	864860	093.4 026.4	20-1611SR-MX-310-2239-09340264-0002A-3	2	3	2.0	
864860	34404	864861	093.4 026.4	20-1611SR-MX-310-2239-09340264-0005A-3	5	3	5.0	
864861	34404	864862	093.4 026.4	20-1611SR-MX-310-2239-09340264-0010A-3	10	3	10.0	
864862	34404	864863	093.4 026.4	20-1611SR-MX-310-2239-09340264-0015A-3	15	3	15.0	

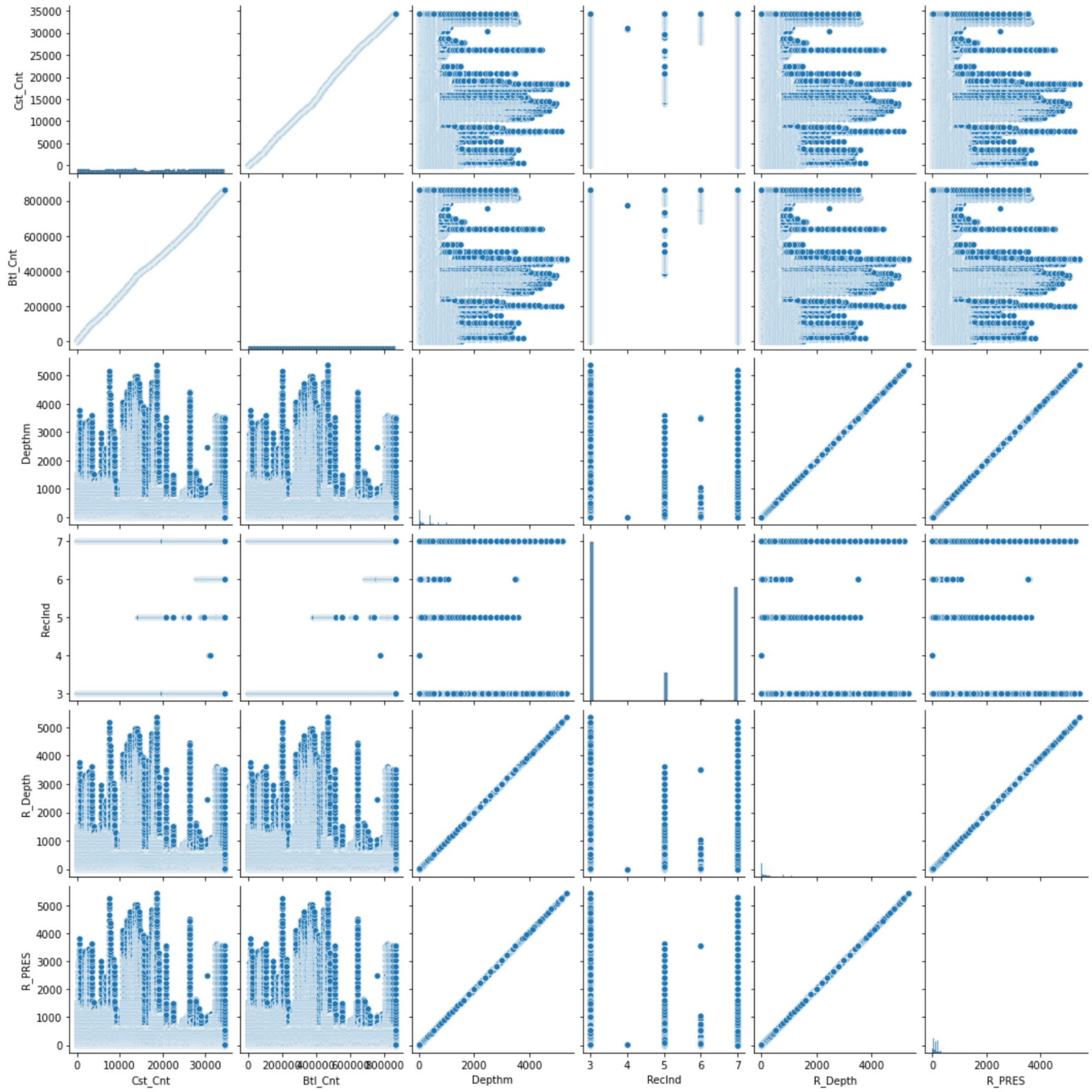
864863 rows × 8 columns

In [7]:
```
df1.columns
```

Out[7]:
```
Index(['Cst_Cnt', 'Btl_Cnt', 'Sta_ID', 'Depth_ID', 'Depthm', 'RecInd',
       'R_Depth', 'R_PRES'],
      dtype='object')
```

In [8]:
```
sns.pairplot(df1)
```

Out[8]:
```
<seaborn.axisgrid.PairGrid at 0x1ec00031130>
```



In [9]:

```
df1.describe()
```

Out[9]:

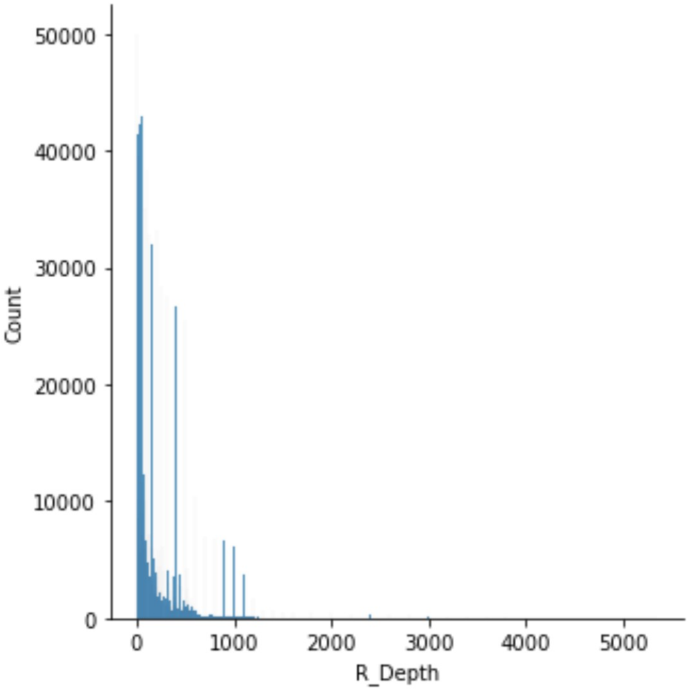
	Cst_Cnt	Btl_Cnt	Depthm	RecInd	R_Depth	R_PRE
count	864863.000000	864863.000000	864863.000000	864863.000000	864863.000000	864863.00000
mean	17138.790958	432432.000000	226.831951	4.700273	226.832495	228.3956
std	10240.949817	249664.587267	316.050259	1.877428	316.050007	319.4567
min	1.000000	1.000000	0.000000	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 [10]:

```
sns.displot(df1['R_Depth'])
```

Out[10]:

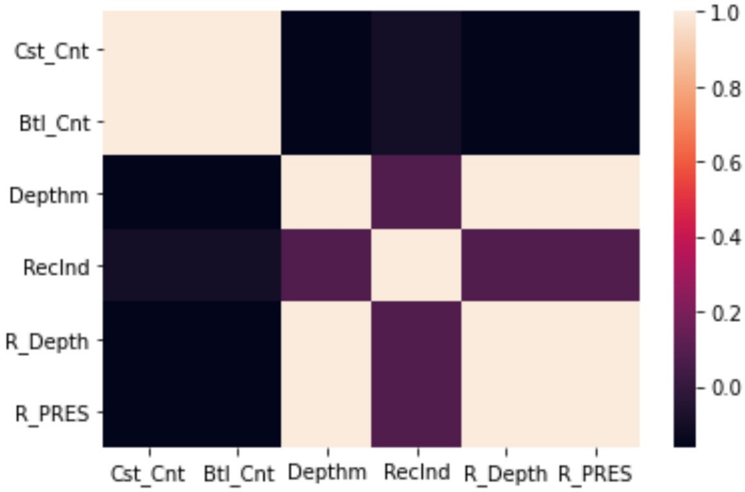
```
<seaborn.axisgrid.FacetGrid at 0x1ec000312b0>
```



In [11]:

Out[11]:

```
<AxesSubplot:>
```



In [12]:

```python
x = df1[['Cst_Cnt', 'Btl_Cnt', 'Depthm', 'RecInd','R_Depth']]
```

In [13]:

```python
# to split my dataset into training and test data
from sklearn.model_selection import train_test_split
```

In [14]:

```python
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
```

Out[14]:

```
LinearRegression()
```

In [15]:

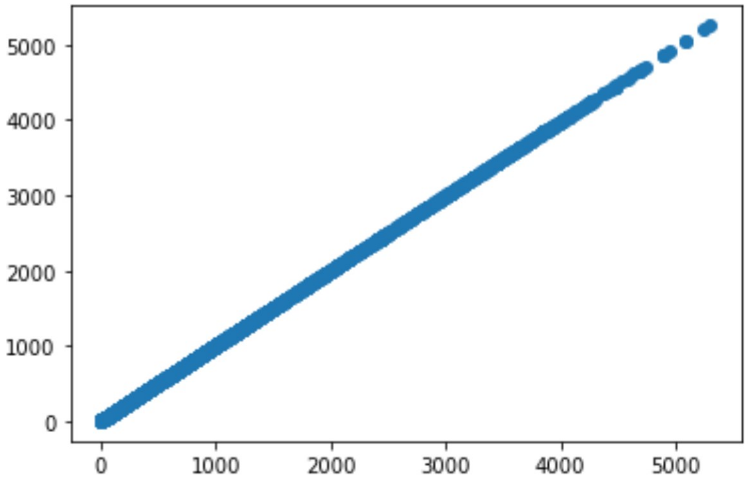
```python
coeff = pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
```

Out[15]:

	Co-efficient
Cst_Cnt	-0.000164
Btl_Cnt	0.000007
Depthm	-0.698206
RecInd	-0.018374
R_Depth	1.709006

```
In [16]: prediction= lr.predict(x_test)
```

```
Out[16]: <matplotlib.collections.PathCollection at 0x1ec09aeee50>
```



```
In [17]:
```

```
Out[17]: 0.9999881620270429
```

LASSO AND RIDGE

```
In [18]:
```

```
In [19]: rr=Ridge(alpha=10)
```

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

```
In [20]:
```

```
Out[20]: 0.999988162028207
```

```
In [21]: la=Lasso(alpha=10)
```

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

```
In [22]:
```

```
Out[22]: 0.9999880748068196
```