Problem Statement

Linear Regression

Import Libraries

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
a=pd.read_csv("bottle.csv")
a
```

```
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[2]:

	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	..

	Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Salnty	O2ml_L	STheta	O2Sat	..
...	...	...	...	...	...	...	...	...	...	...	..
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	..
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

To display top 10 rows

In [10]:
```
c=a.head(15)
c
```

Out[10]:

	Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Salnty	O2ml_L	STheta	O2Sat	...	R_PH
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-	8	10.46	33.440	NaN	25.656	NaN	...	

	Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Salnty	O2ml_L	STheta	O2Sat	...	R_PH
				0930-05400560-0008A-3								
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	...	
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-	78	9.83	33.510	NaN	25.819	NaN	...	

	Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Salnty	O2ml_L	STheta	O2Sat	...	R_PH
				0930-05400560-0078A-3								
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 rows × 74 columns

To find Missing values

In [15]:
```
c.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15 entries, 0 to 14
Data columns (total 74 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Cst_Cnt              15 non-null     int64
 1   Btl_Cnt              15 non-null     int64
 2   Sta_ID               15 non-null     object
 3   Depth_ID             15 non-null     object
 4   Depthm               15 non-null     int64
 5   T_degC               15 non-null     float64
 6   Salnty               15 non-null     float64
 7   O2ml_L               0 non-null      float64
 8   STheta               15 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               15 non-null     int64
 13  T_prec               15 non-null     float64
 14  T_qual               0 non-null      float64
 15  S_prec               15 non-null     float64
```

```
 16  S_qual               0 non-null      float64
 17  P_qual               15 non-null     float64
 18  O_qual               15 non-null     float64
 19  SThtaq               0 non-null      float64
 20  O2Satq               15 non-null     float64
 21  ChlorA               0 non-null      float64
 22  Chlqua               15 non-null     float64
 23  Phaeop               0 non-null      float64
 24  Phaqua               15 non-null     float64
 25  PO4uM                0 non-null      float64
 26  PO4q                 15 non-null     float64
 27  SiO3uM               0 non-null      float64
 28  SiO3qu               15 non-null     float64
 29  NO2uM                0 non-null      float64
 30  NO2q                 15 non-null     float64
 31  NO3uM                0 non-null      float64
 32  NO3q                 15 non-null     float64
 33  NH3uM                0 non-null      float64
 34  NH3q                 15 non-null     float64
 35  C14As1               0 non-null      float64
 36  C14A1p               0 non-null      float64
 37  C14A1q               15 non-null     float64
 38  C14As2               0 non-null      float64
 39  C14A2p               0 non-null      float64
 40  C14A2q               15 non-null     float64
 41  DarkAs               0 non-null      float64
 42  DarkAp               0 non-null      float64
 43  DarkAq               15 non-null     float64
 44  MeanAs               0 non-null      float64
 45  MeanAp               0 non-null      float64
 46  MeanAq               15 non-null     float64
 47  IncTim               0 non-null      object
 48  LightP               0 non-null      float64
 49  R_Depth              15 non-null     float64
 50  R_TEMP               15 non-null     float64
 51  R_POTEMP             15 non-null     float64
 52  R_SALINITY           15 non-null     float64
 53  R_SIGMA              15 non-null     float64
 54  R_SVA                15 non-null     float64
 55  R_DYNHT              15 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               15 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: 8.8+ KB
```

To display summary of statistics

In [11]:

```
a.describe()
```

Out[11]:

	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

To display column heading

In [12]:
```
a.columns
```

Out[12]:
```
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')
```

Pairplot

In [18]:
```
s=a.dropna(axis=1)
s
```

Out[18]:

	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
...	...	...	...	...	...	...	...	...

	Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	RecInd	R_Depth	R_PRES
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

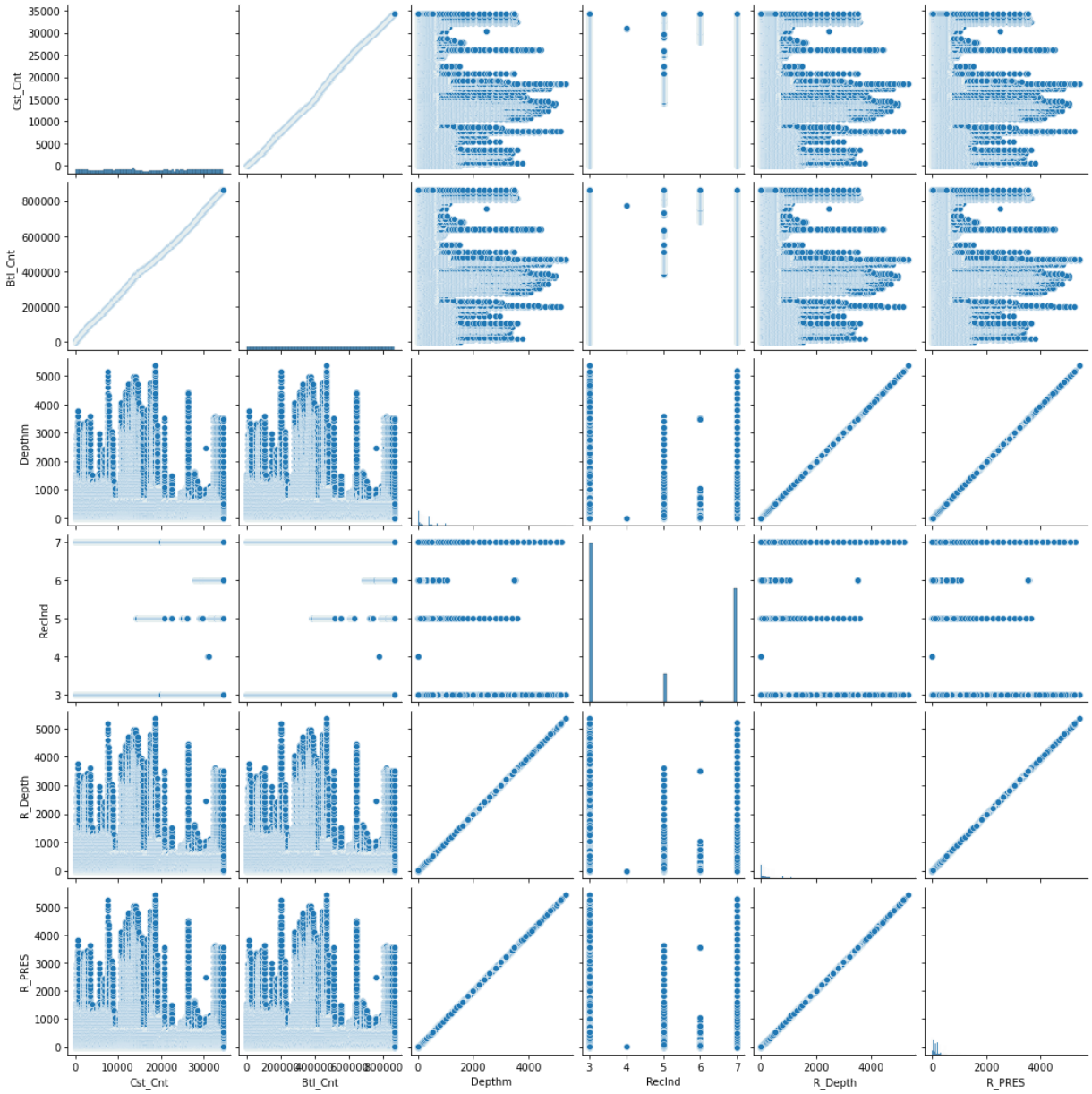
864863 rows × 8 columns

In [28]:
```
s.columns
```

Out[28]:
```
Index(['Cst_Cnt', 'Btl_Cnt', 'Sta_ID', 'Depth_ID', 'Depthm', 'RecInd',
       'R_Depth', 'R_PRES'],
      dtype='object')
```

In [19]:
```
sns.pairplot(s)
```

Out[19]:
```
<seaborn.axisgrid.PairGrid at 0x1547c2b2a30>
```

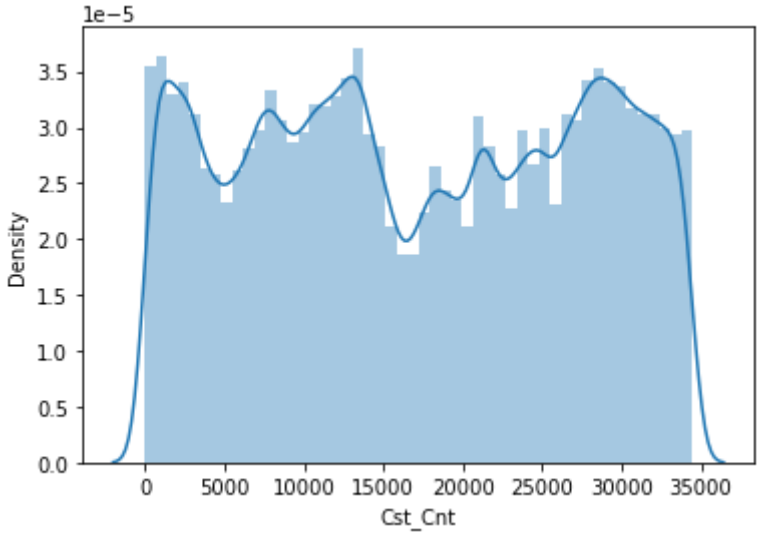


Distribution Plot

In [20]:
```
sns.distplot(s['Cst_Cnt'])
```

```
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)
```

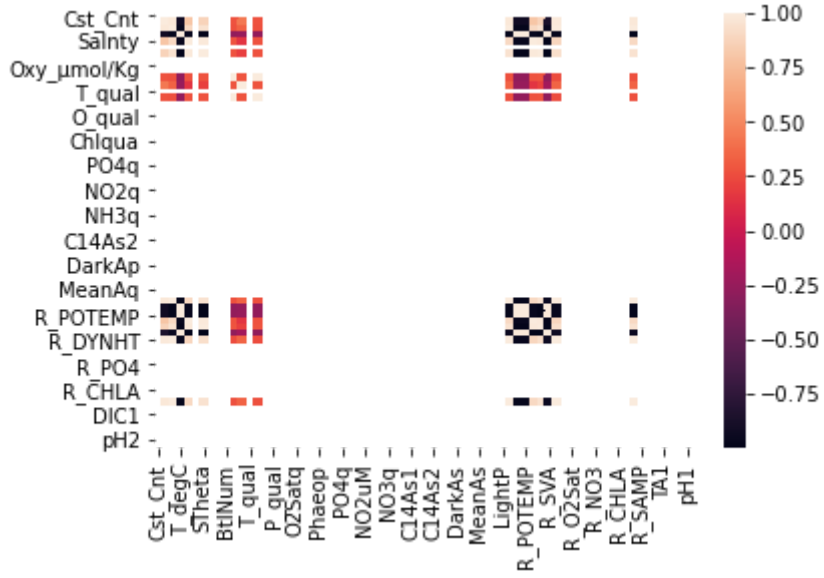
Out[20]: <AxesSubplot:xlabel='Cst_Cnt', ylabel='Density'>



Correlation

In [21]:
```
b=c[['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']]
sns.heatmap(.corr())
```

Out[21]: <AxesSubplot:>



Train the model - Model Building

We are going to train linear regression model: We need to split out data into 2 variables x,y where x is independant and y is dependant on x(output). We could ignore address column as it is not required for our model.

In [32]:
```
g=s[['Cst_Cnt', 'Btl_Cnt','Depthm','RecInd','R_Depth']]
h=s['R_PRES']
```

To split dataset into training end test

In [34]:
```
from sklearn.model_selection import train_test_split
g_train,g_test,h_train,h_test=train_test_split(g,h,test_size=0.5)
```

To run the model

In [35]:
```
from sklearn.linear_model import LinearRegression
```

In [36]:
```
lr=LinearRegression()
lr.fit(g_train,h_train)
```

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

In [37]:
```
print(lr.intercept_)
```

```
-1.0672657730599155
```

Coeffecient

In [38]:
```
coeff=pd.DataFrame(lr.coef_,g.columns,columns=['Co-effecient'])
coeff
```

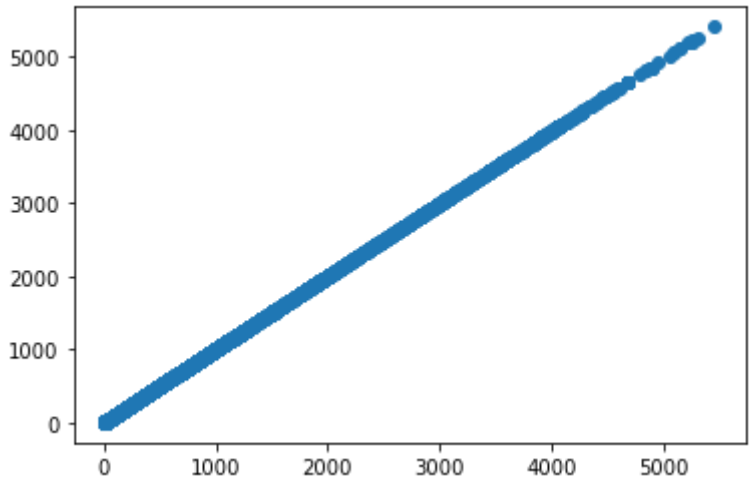
Out[38]:

	Co-effecient
Cst_Cnt	-0.000169
Btl_Cnt	0.000007
Depthm	-0.606615
RecInd	-0.019260
R_Depth	1.617523

Best Fit line

In [39]:
```
prediction=lr.predict(g_test)
plt.scatter(h_test,prediction)
```

Out[39]:
```
<matplotlib.collections.PathCollection at 0x154799d4580>
```



To find score

In [40]:
```
print(lr.score(g_test,h_test))
```

```
0.9999882417296662
```

In [ ]:
```

```