

Data Collection

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

```
In [2]: data = pd.read_csv(r"C:\Users\user\Downloads\9_bottle.csv")
data
```

```
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	O2ml_L	STheta	O2Sa
0	1	1	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0000A-3	0	10.500	33.4400	NaN	25.64900	Na
1	1	2	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0008A-3	8	10.460	33.4400	NaN	25.65600	Na
2	1	3	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0010A-7	10	10.460	33.4370	NaN	25.65400	Na
3	1	4	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0019A-3	19	10.450	33.4200	NaN	25.64300	Na
4	1	5	054.0 056.0	19- 4903CR- HY-060- 0930- 05400560- 0020A-7	20	10.450	33.4210	NaN	25.64300	Na
...
864858	34404	864859	093.4 026.4	20- 1611SR- MX-310- 2239- 09340264- 0000A-7	0	18.744	33.4083	5.805	23.87055	108.7
864859	34404	864860	093.4 026.4	20- 1611SR- MX-310- 2239- 09340264- 0002A-3	2	18.744	33.4083	5.805	23.87072	108.7
864860	34404	864861	093.4 026.4	20- 1611SR- MX-310- 2239- 09340264- 0005A-3	5	18.692	33.4150	5.796	23.88911	108.4
864861	34404	864862	093.4 026.4	20- 1611SR- MX-310- 2239- 09340264- 0010A-3	10	18.161	33.4062	5.816	24.01426	107.7

	Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Salnty	O2ml_L	STheta	O2Sat
864862	34404	864863	093.4 026.4	20- 1611SR- MX-310- 2239- 09340264- 0015A-3	15	17.533	33.3880	5.774	24.15297	105.6

864863 rows × 74 columns

```
In [3]: data1 = data[0:5000]  
data1
```

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	...
...
4995	165	4996	092.0 098.0	19-4904NS-HY-102-1342-09200980-0099A-3	99	11.41	33.440	5.42	25.490	87.6	...
4996	165	4997	092.0 098.0	19-4904NS-HY-102-1342-09200980-0100A-7	100	11.36	33.444	5.39	25.502	87.0	...
4997	165	4998	092.0 098.0	19-4904NS-HY-102-1342-09200980-0125A-7	125	10.16	33.555	4.59	25.800	72.2	...
4998	165	4999	092.0 098.0	19-4904NS-HY-102-1342-09200980-0149A-3	149	9.24	33.680	3.78	26.049	58.3	...

	Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Salnty	O2ml_L	STheta	O2Sat	...
4999	165	5000	092.0 098.0	19-4904NS-HY-102-1342-09200980-0150A-7	150	9.22	33.682	3.76	26.054	58.0	...

5000 rows × 74 columns

In [4]: `data1.info()`


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

```
RangeIndex: 5000 entries, 0 to 4999
```

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

#	Column	Non-Null Count	Dtype
0	Cst_Cnt	5000 non-null	int64
1	Btl_Cnt	5000 non-null	int64
2	Sta_ID	5000 non-null	object
3	Depth_ID	5000 non-null	object
4	Depthm	5000 non-null	int64
5	T_degC	4980 non-null	float64
6	Salnty	4848 non-null	float64
7	O2ml_L	2810 non-null	float64
8	STheta	4833 non-null	float64
9	O2Sat	2713 non-null	float64
10	Oxy_μmol/Kg	2713 non-null	float64
11	BtlNum	0 non-null	float64
12	RecInd	5000 non-null	int64
13	T_prec	4980 non-null	float64
14	T_qual	51 non-null	float64
15	S_prec	4848 non-null	float64
16	S_qual	238 non-null	float64
17	P_qual	5000 non-null	float64
18	O_qual	2193 non-null	float64
19	SThtaq	281 non-null	float64
20	O2Satq	2356 non-null	float64
21	ChlorA	0 non-null	float64
22	Chlqua	5000 non-null	float64
23	Phaeop	0 non-null	float64
24	Phaqua	5000 non-null	float64
25	PO4uM	1046 non-null	float64
26	PO4q	3954 non-null	float64
27	SiO3uM	0 non-null	float64
28	SiO3qu	5000 non-null	float64
29	NO2uM	0 non-null	float64
30	NO2q	5000 non-null	float64
31	NO3uM	0 non-null	float64
32	NO3q	5000 non-null	float64
33	NH3uM	0 non-null	float64
34	NH3q	5000 non-null	float64
35	C14As1	0 non-null	float64
36	C14A1p	0 non-null	float64
37	C14A1q	5000 non-null	float64
38	C14As2	0 non-null	float64
39	C14A2p	0 non-null	float64
40	C14A2q	5000 non-null	float64
41	DarkAs	0 non-null	float64
42	DarkAp	0 non-null	float64
43	DarkAq	5000 non-null	float64
44	MeanAs	0 non-null	float64
45	MeanAp	0 non-null	float64
46	MeanAq	5000 non-null	float64
47	IncTim	0 non-null	object
48	LightP	0 non-null	float64
49	R_Depth	5000 non-null	float64
50	R_TEMP	4980 non-null	float64
51	R_POTEMP	4775 non-null	float64

```

52  R_SALINITY          4848 non-null    float64
53  R_SIGMA             4719 non-null    float64
54  R_SVA               4719 non-null    float64
55  R_DYNHT             4786 non-null    float64
56  R_O2                2810 non-null    float64
57  R_O2Sat             2690 non-null    float64
58  R_SIO3              0 non-null       float64
59  R_PO4               1046 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              5000 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: 2.8+ MB

```

In [5]: data1.columns

```

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',
              'SThetaq', '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 = data1[['Cst_Cnt', 'Btl_Cnt', 'Depthm', 'T_degC', 'Salnty']]
df1
```

Out[6]:

	Cst_Cnt	Btl_Cnt	Depthm	T_degC	Salnty
0	1	1	0	10.50	33.440
1	1	2	8	10.46	33.440
2	1	3	10	10.46	33.437
3	1	4	19	10.45	33.420
4	1	5	20	10.45	33.421
...
4995	165	4996	99	11.41	33.440
4996	165	4997	100	11.36	33.444
4997	165	4998	125	10.16	33.555
4998	165	4999	149	9.24	33.680
4999	165	5000	150	9.22	33.682

5000 rows × 5 columns

```
In [7]: df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Cst_Cnt     5000 non-null   int64
1   Btl_Cnt     5000 non-null   int64
2   Depthm      5000 non-null   int64
3   T_degC      4980 non-null   float64
4   Salnty      4848 non-null   float64
dtypes: float64(2), int64(3)
memory usage: 195.4 KB
```

```
In [8]: df1.isna().sum()
```

```
Out[8]: Cst_Cnt      0
Btl_Cnt      0
Depthm       0
T_degC       20
Salnty      152
dtype: int64
```

```
In [9]: df2 = df1.fillna(value=30)
```

```
In [10]: df2.isna().sum()
```

```
Out[10]: Cst_Cnt      0  
Btl_Cnt      0  
Depthm      0  
T_degC      0  
Salnty      0  
dtype: int64
```

```
In [11]: df2.describe()
```

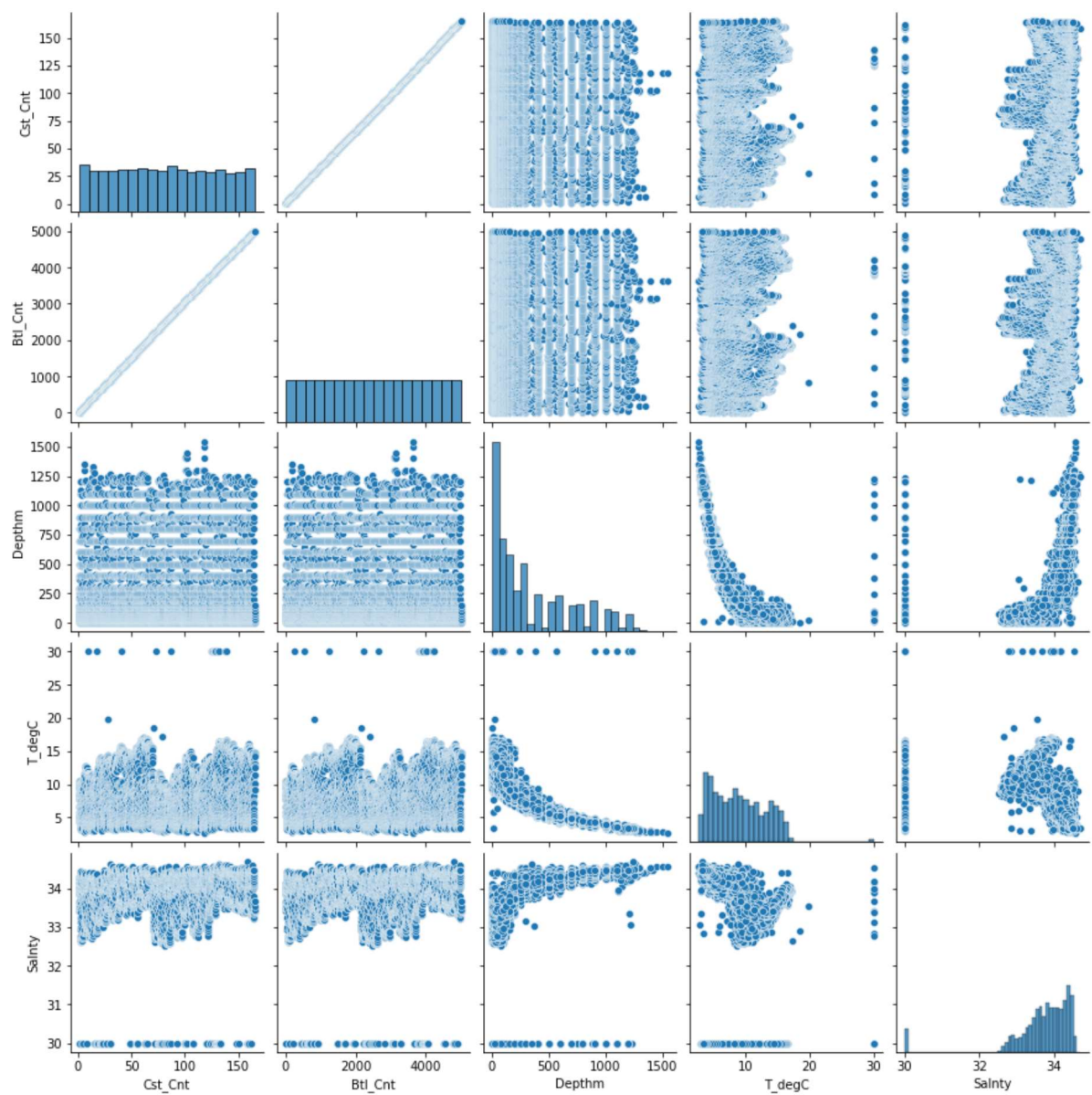
```
Out[11]:
```

	Cst_Cnt	Btl_Cnt	Depthm	T_degC	Salnty
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000
mean	82.129400	2500.500000	347.985200	9.058438	33.711692
std	47.348975	1443.520003	358.279702	4.122914	0.818492
min	1.000000	1.000000	0.000000	2.700000	30.000000
25%	41.000000	1250.750000	55.000000	5.400000	33.460000
50%	82.000000	2500.500000	200.000000	8.630000	33.863500
75%	123.000000	3750.250000	600.000000	12.230000	34.250000
max	165.000000	5000.000000	1547.000000	30.000000	34.700000

EDA and Visualization

```
In [12]: sns.pairplot(df2)
```

```
Out[12]: <seaborn.axisgrid.PairGrid at 0x2528bb2ba00>
```

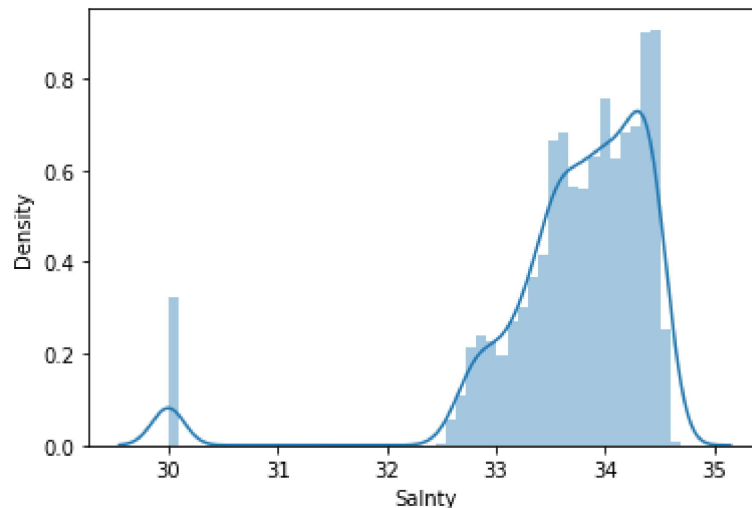


```
In [13]: sns.distplot(df2["Salnty"])
```

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

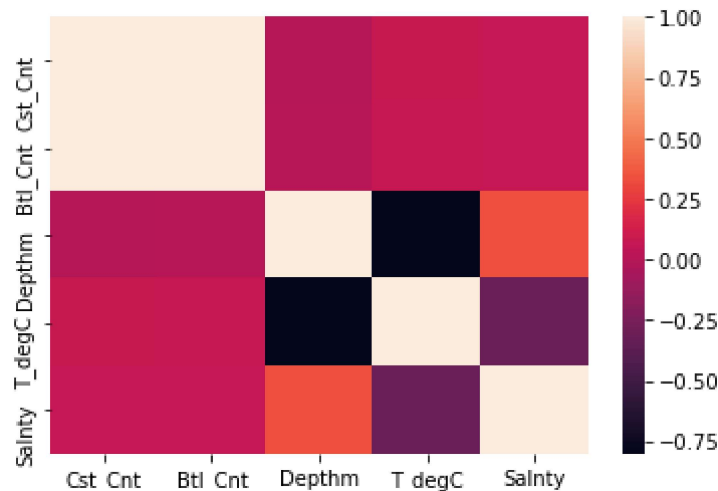
```
warnings.warn(msg, FutureWarning)
```

```
Out[13]: <AxesSubplot:xlabel='Salnty', ylabel='Density'>
```



```
In [14]: sns.heatmap(df2.corr())
```

```
Out[14]: <AxesSubplot:>
```



Linear Regression

```
In [15]: x = df2[['Cst_Cnt', 'Btl_Cnt', 'Depthm', 'T_degC']]
y = df2["Salnty"]
```

```
In [16]: from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.5)
```

```
In [17]: from sklearn.linear_model import LinearRegression

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

Out[17]: LinearRegression()

```
In [18]: print(lr.intercept_)

33.904138254630645
```

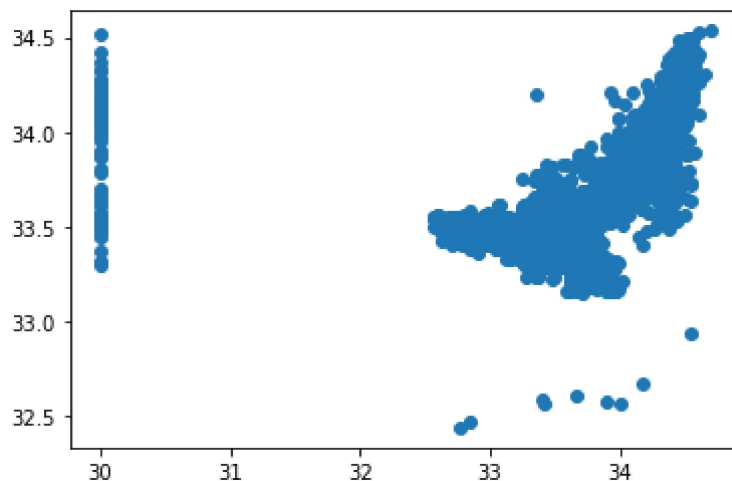
```
In [19]: coeff = pd.DataFrame(lr.coef_,x.columns,columns=["Co-efficient"])
coeff
```

Out[19]:

	Co-efficient
Cst_Cnt	0.161904
Btl_Cnt	-0.005263
Depthm	0.000413
T_degC	-0.053049

```
In [20]: prediction = lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[20]: <matplotlib.collections.PathCollection at 0x2528dcc57f0>



```
In [21]: print(lr.score(x_test,y_test))

0.1016214780853264
```

```
In [22]: lr.score(x_train,y_train)
```

Out[22]: 0.13883233817465412

```
In [24]: from sklearn.linear_model import Ridge,Lasso
```

```
In [25]: rr = Ridge(alpha=10)
rr.fit(x_train,y_train)
rr.score(x_test,y_test)
```

```
Out[25]: 0.10190672007506918
```

```
In [26]: rr.score(x_train,y_train)
```

```
Out[26]: 0.13882912883205079
```

```
In [27]: ls = Lasso(alpha=10)
ls.fit(x_train,y_train)
ls.score(x_test,y_test)
```

```
Out[27]: 0.11789109389070662
```

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

```
Out[28]: 0.11161921440472855
```

```
In [ ]:
```