Exploratory Data Analysis

About the dataset

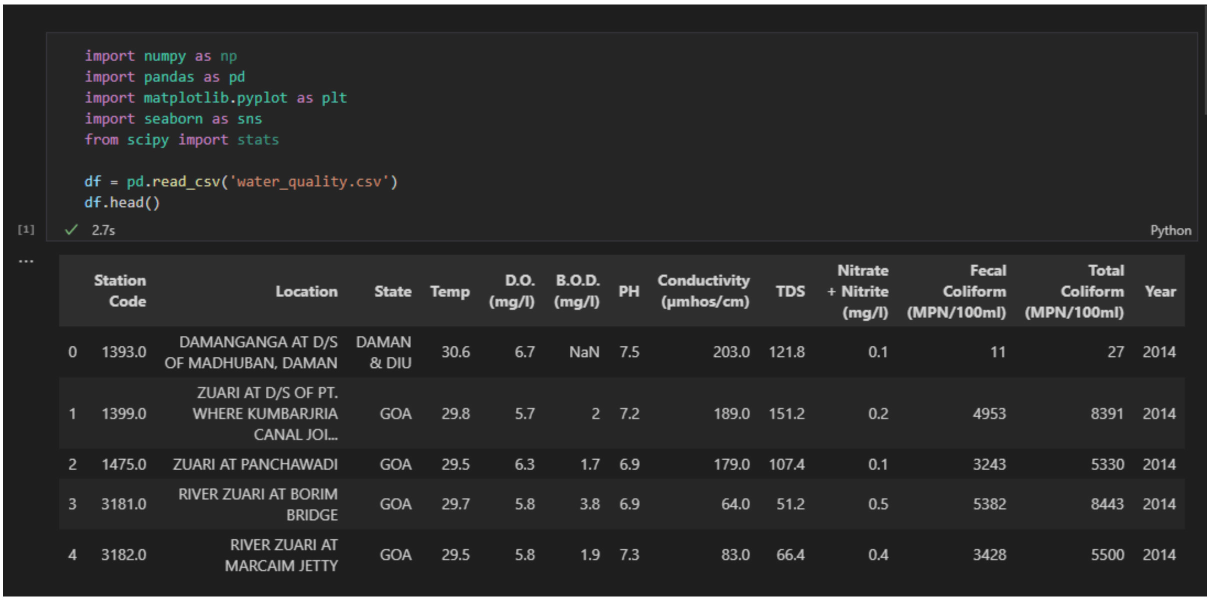
- Unsafe water kills more people each year than war and all other forms of violence combined. The widespread problem of water pollution is jeopardizing our health.
- Water pollution, as we all know, is its contamination by release of toxic substances, detrimental to living organism.
- Water quality refers to the chemical physical, and biological characteristics of water based on standards of its usage. This open-to-use database contains water sample quality measures from various water body stations across India.

Features

- Station Code – Codes assigned to each water body monitoring stations
- Location – Location of the station
- State – A territory within a country (here, India).
- Temp – Temperature in Celsius.
- D.O. (mg/l) – Dissolved Oxygen (DO) is the amount of oxygen present in the water. Desired level of D.O. in water sample is – 6.5-8 mg/L or 80-120%
- B.O.D. (mg/l) – Biochemical Oxygen Demand is amount of oxygen consumed by microorganisms to decompose organic matter under aerobic conditions.
- PH – PH value of the water.
- Conductivity (µmho/cm) – Conductivity of the water measured in µS/cm where S is Siemens.
- TDS – Total Dissolved Solids i.e., total concentration of dissolved substances in water. Desirable limit of TDS is 500mg/L or 1000mg/L
- Nitrate + Nitrite (mg/l) – Presence of Nitrate and Nitrite in water, which is highly toxic.
- Fecal Coliform (MPN/100ml) – Fecal Coliform is an anaerobic rod-shaped bacterium generally originate in intestine of war-blooded animals. MPN/100ml is the Most Probable Number in a sample of 100ml.
- Total Coliform (MPN/100ml) – Total Coliform presence in water. Coliform bacteria are unlikely to cause illness (except few) but their presence in water indicates that disease-causing organisms (pathogens) could be present.
- Year – Year of sampling.

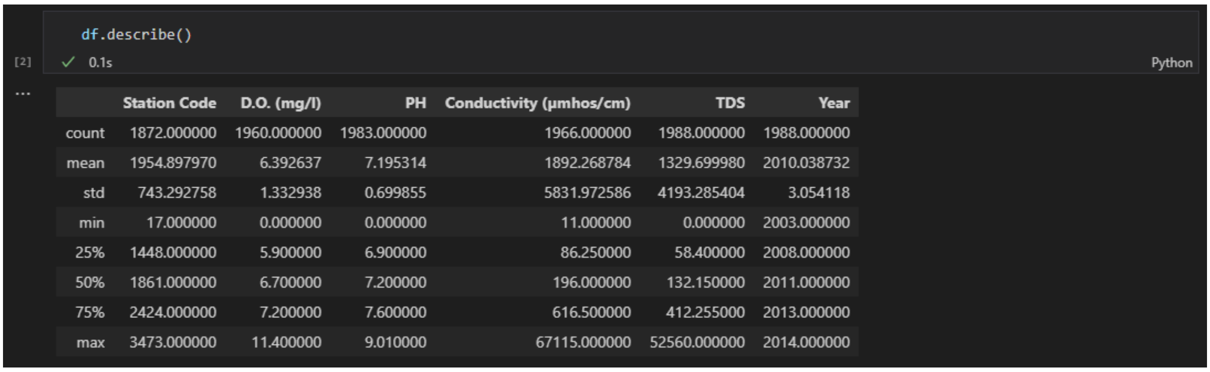
Data Exploration

Importing essential packages and reading dataset



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

df = pd.read_csv('water_quality.csv')
df.head()
```

	Station Code	Location	State	Temp	D.O. (mg/l)	B.O.D. (mg/l)	PH	Conductivity (μmhos/cm)	TDS	Nitrate + Nitrite (mg/l)	Fecal Coliform (MPN/100ml)	Total Coliform (MPN/100ml)	Year
0	1393.0	DAMANGANGA AT D/S OF MADHUBAN, DAMAN	DAMAN & DIU	30.6	6.7	NaN	7.5	203.0	121.8	0.1	11	27	2014
1	1399.0	ZUARI AT D/S OF PT. WHERE KUMBARJRIA CANAL JOI...	GOA	29.8	5.7	2	7.2	189.0	151.2	0.2	4953	8391	2014
2	1475.0	ZUARI AT PANCHAWADI	GOA	29.5	6.3	1.7	6.9	179.0	107.4	0.1	3243	5330	2014
3	3181.0	RIVER ZUARI AT BORIM BRIDGE	GOA	29.7	5.8	3.8	6.9	64.0	51.2	0.5	5382	8443	2014
4	3182.0	RIVER ZUARI AT MARCAIM JETTY	GOA	29.5	5.8	1.9	7.3	83.0	66.4	0.4	3428	5500	2014



```
df.describe()
```

	Station Code	D.O. (mg/l)	PH	Conductivity (μmhos/cm)	TDS	Year
count	1872.000000	1960.000000	1983.000000	1966.000000	1988.000000	1988.000000
mean	1954.897970	6.392637	7.195314	1892.268784	1329.699980	2010.038732
std	743.292758	1.332938	0.699855	5831.972586	4193.285404	3.054118
min	17.000000	0.000000	0.000000	11.000000	0.000000	2003.000000
25%	1448.000000	5.900000	6.900000	86.250000	58.400000	2008.000000
50%	1861.000000	6.700000	7.200000	196.000000	132.150000	2011.000000
75%	2424.000000	7.200000	7.600000	616.500000	412.255000	2013.000000
max	3473.000000	11.400000	9.010000	67115.000000	52560.000000	2014.000000

Observation:

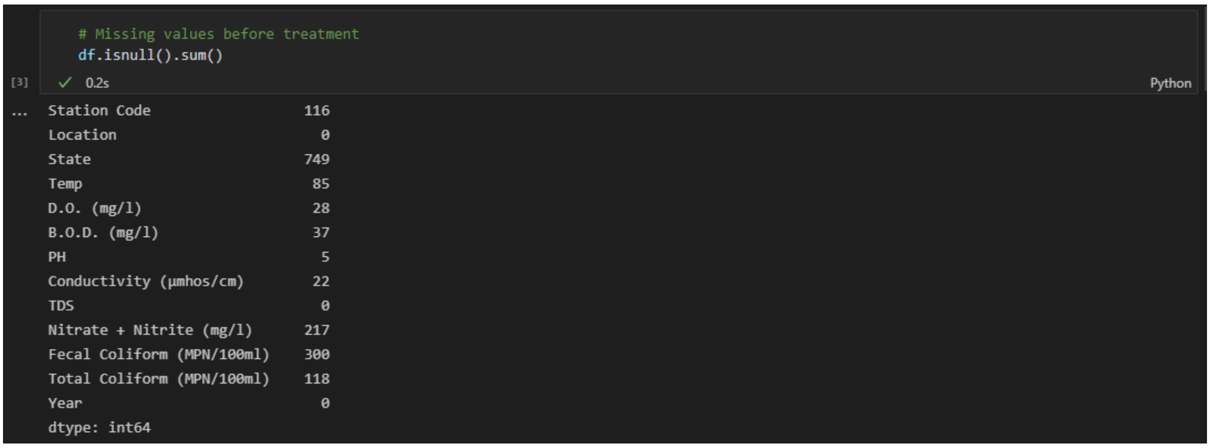
- The data is collected between year 2003 and 2014.
- The mean PH is 7.1 which is 'Neutral' on PH scale, the PH value of the distilled water. Also, the Interquartile range is 0.7 as Q1 = 6.9 and Q3 = 7.6 which is desired range of PH for drinking water.
- The shape of the dataset is 1988x13, i.e., 13 features as mentioned above. A large amount of data is missing.
- Describe method didn't include features such as Temp, B.O.D, Nitrate & Nitrite concentration, and presence of coliform, which are expected to be numeric (float) type, this needs to be taken care.

Plan:

- We need to convert the dtype of each, expected numeric, feature to float.
- Handle missing values.
- Using Correlation Matrix/Heatmap for determining the relationship between each feature and dropping correlated feature to avoid multicollinearity problem.

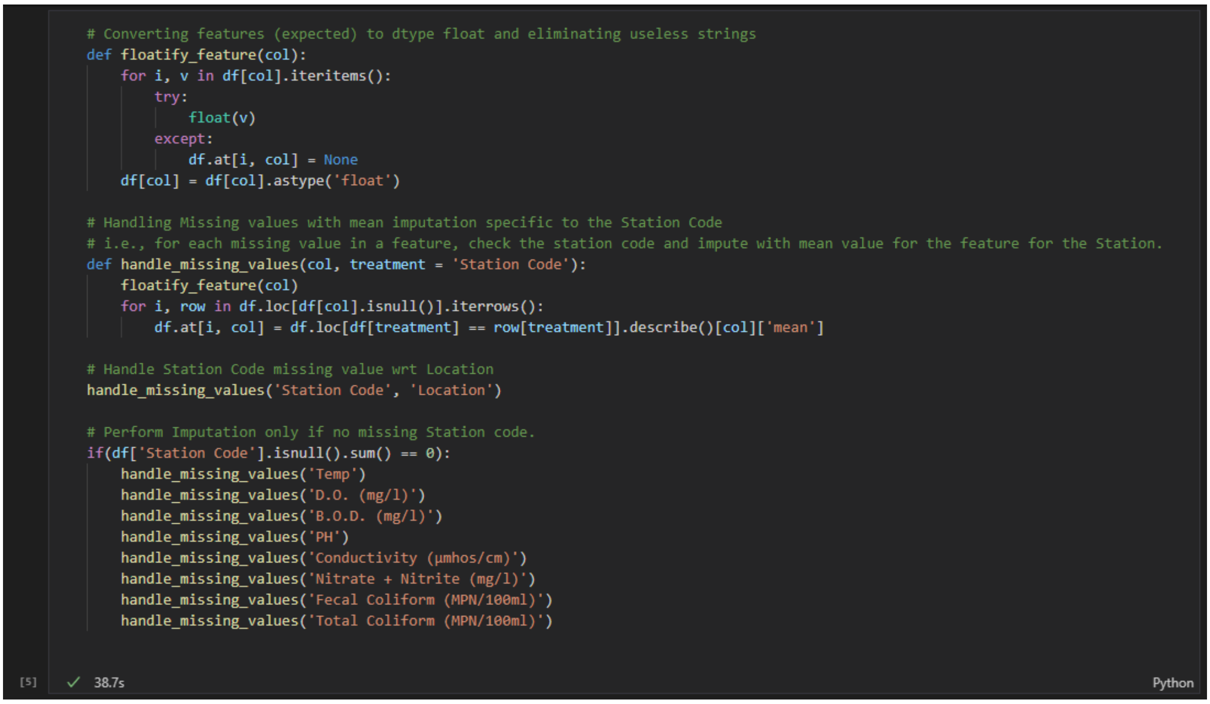
Data Cleaning and Feature Engineering

Missing values and converting data type of expected numeric features to float.

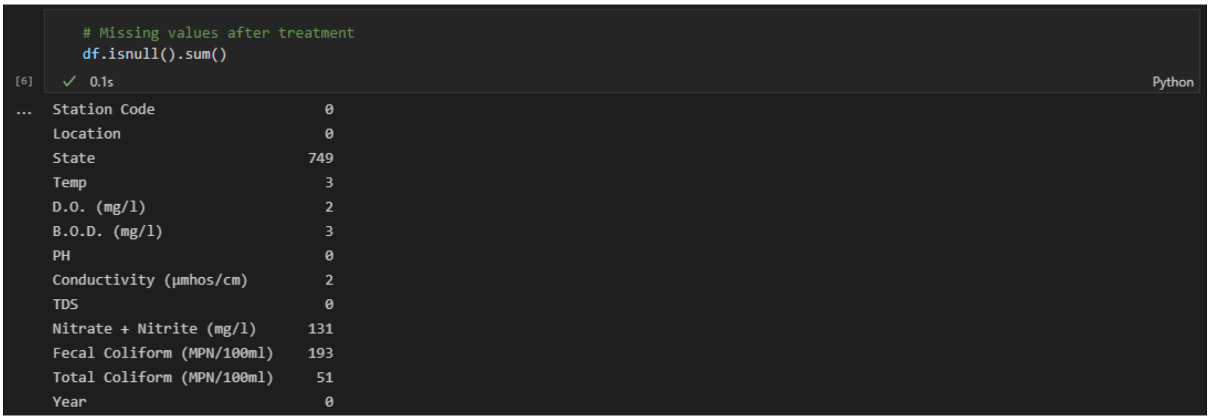


```python
# Missing values before treatment
df.isnull().sum()
```

```
...  Station Code                  116
     Location                        0
     State                         749
     Temp                           85
     D.O. (mg/l)                    28
     B.O.D. (mg/l)                  37
     PH                              5
     Conductivity (µmhos/cm)        22
     TDS                             0
     Nitrate + Nitrite (mg/l)      217
     Fecal Coliform (MPN/100ml)    300
     Total Coliform (MPN/100ml)    118
     Year                            0
     dtype: int64
```

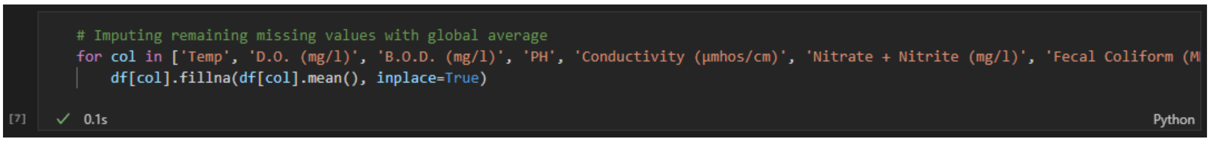


```python
# Converting features (expected) to dtype float and eliminating useless strings
def floatify_feature(col):
    for i, v in df[col].iteritems():
        try:
            float(v)
        except:
            df.at[i, col] = None
    df[col] = df[col].astype('float')

# Handling Missing values with mean imputation specific to the Station Code
# i.e., for each missing value in a feature, check the station code and impute with mean value for the feature for the Station.
def handle_missing_values(col, treatment = 'Station Code'):
    floatify_feature(col)
    for i, row in df.loc[df[col].isnull()].iterrows():
        df.at[i, col] = df.loc[df[treatment] == row[treatment]].describe()[col]['mean']

# Handle Station Code missing value wrt Location
handle_missing_values('Station Code', 'Location')

# Perform Imputation only if no missing Station code.
if(df['Station Code'].isnull().sum() == 0):
    handle_missing_values('Temp')
    handle_missing_values('D.O. (mg/l)')
    handle_missing_values('B.O.D. (mg/l)')
    handle_missing_values('PH')
    handle_missing_values('Conductivity (µmhos/cm)')
    handle_missing_values('Nitrate + Nitrite (mg/l)')
    handle_missing_values('Fecal Coliform (MPN/100ml)')
    handle_missing_values('Total Coliform (MPN/100ml)')
```



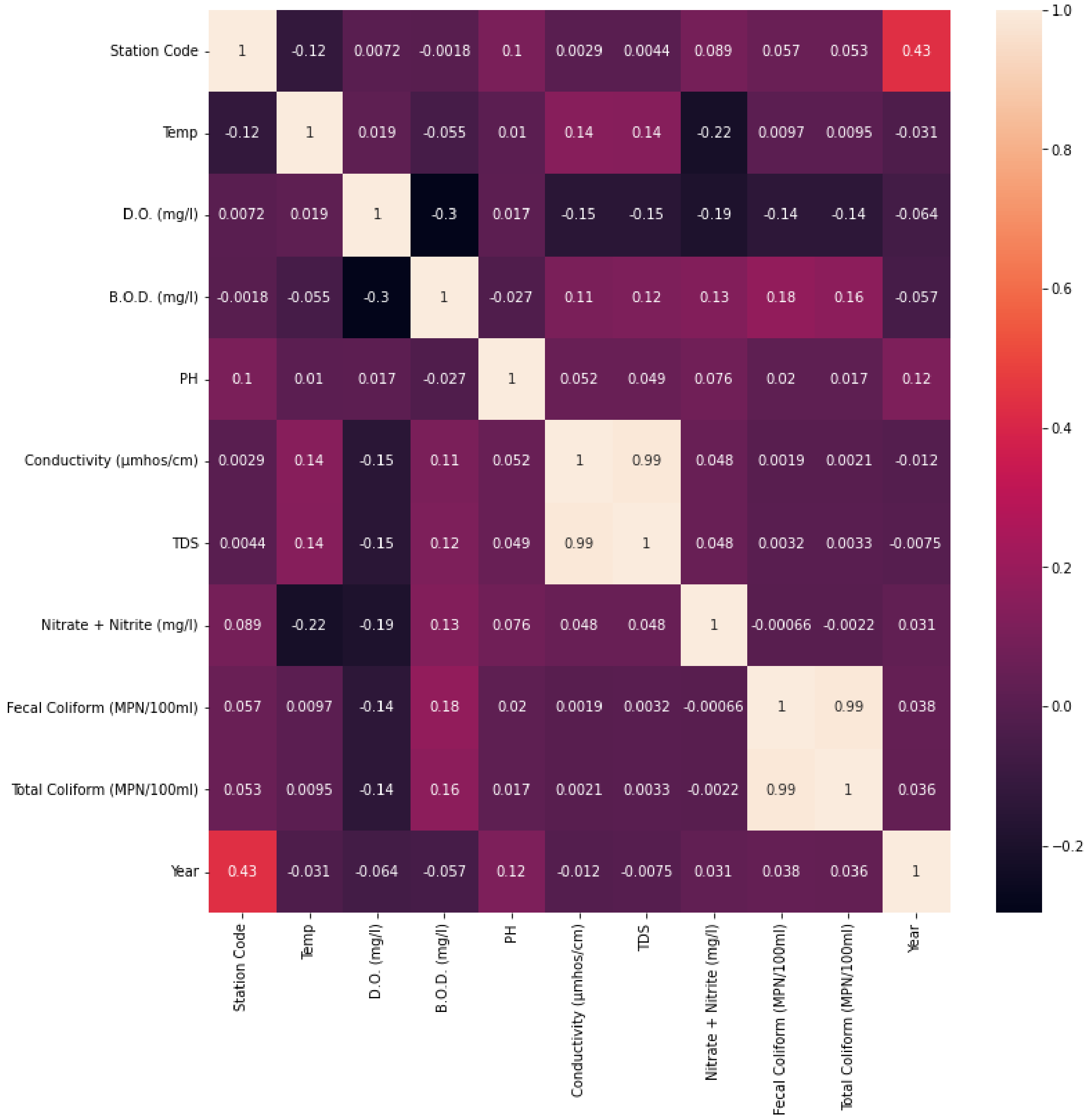
```python
# Missing values after treatment
df.isnull().sum()
```

```
...  Station Code                    0
     Location                        0
     State                         749
     Temp                            3
     D.O. (mg/l)                     2
     B.O.D. (mg/l)                   3
     PH                              0
     Conductivity (µmhos/cm)         2
     TDS                             0
     Nitrate + Nitrite (mg/l)      131
     Fecal Coliform (MPN/100ml)    193
     Total Coliform (MPN/100ml)     51
     Year                            0
```



```python
# Imputing remaining missing values with global average
for col in ['Temp', 'D.O. (mg/l)', 'B.O.D. (mg/l)', 'PH', 'Conductivity (µmhos/cm)', 'Nitrate + Nitrite (mg/l)', 'Fecal Coliform (M
    df[col].fillna(df[col].mean(), inplace=True)
```

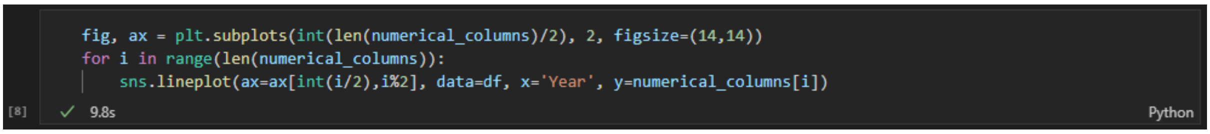
Heatmap using Correlation Matrix



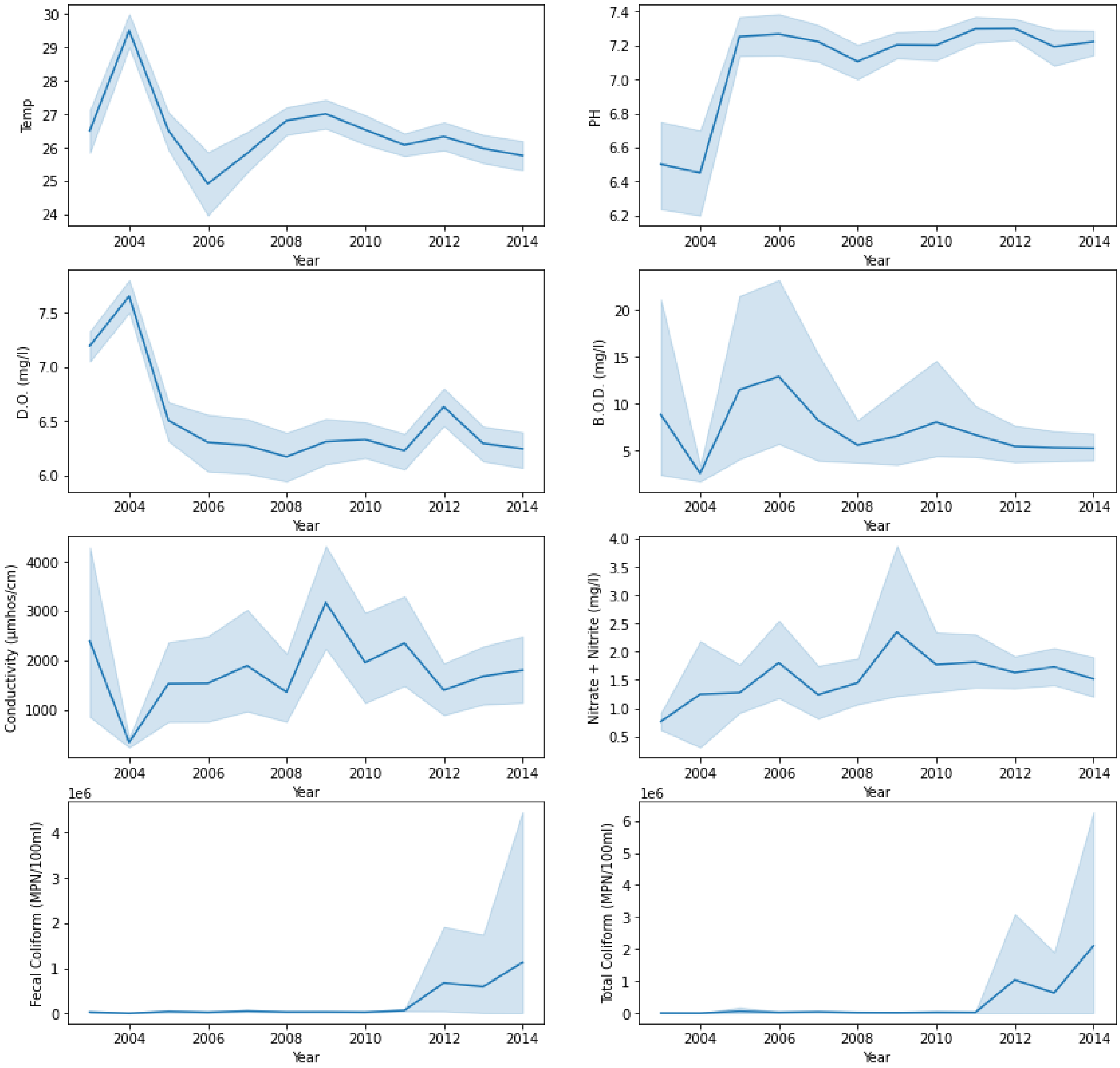
Observation:

- The correlation matrix shows a relationship between Conductivity and the TDS.
- The heatmap also shows that the Fecal Coliform and the Total Coliform presence is water are highly correlated, as expected.
- DO and BDO shows an inverse relationship with each other and also with every other feature comparatively (trivariant analysis)
- Thus, we can drop BDO, TDS, and Fecal Coliform to avoid multicollinearity problem. We can also drop Location and State as Station Code serves the purpose.

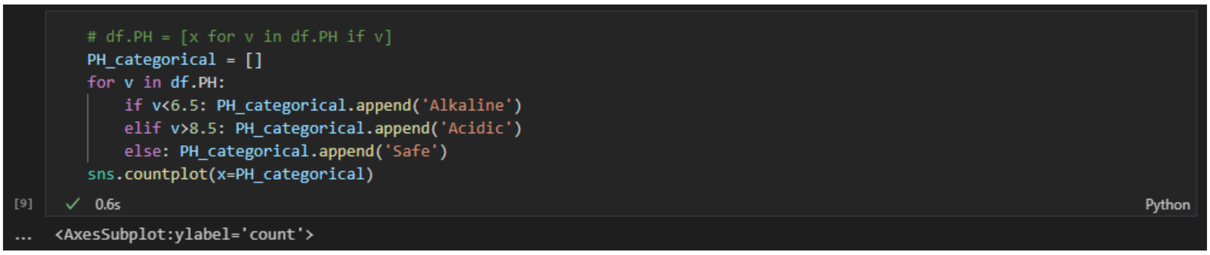
Aggregated Line Graphs showing change in values like PH, DO, Conductivity over the years.



```
fig, ax = plt.subplots(int(len(numerical_columns)/2), 2, figsize=(14,14))
for i in range(len(numerical_columns)):
    sns.lineplot(ax=ax[int(i/2),i%2], data=df, x='Year', y=numerical_columns[i])
```

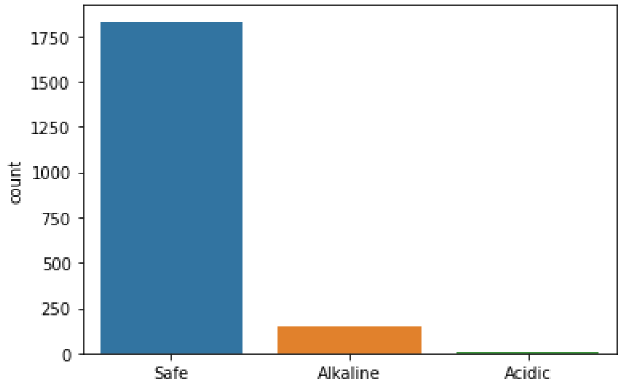


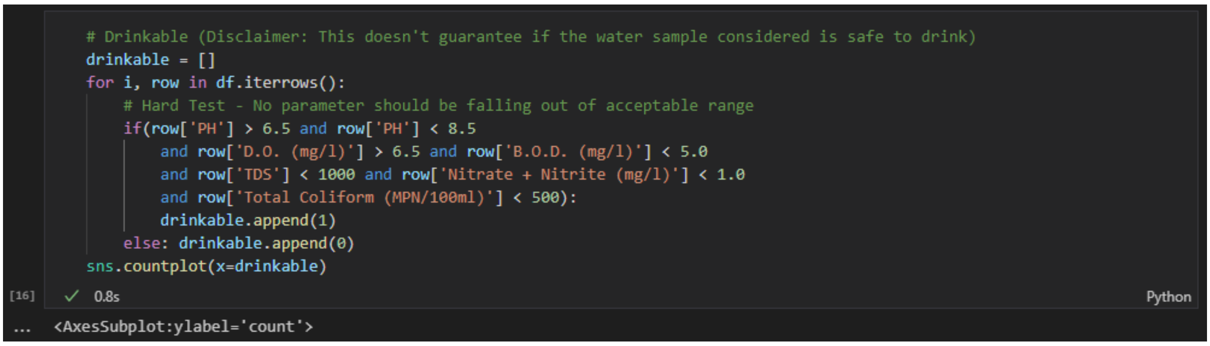
PH binned as Alkaline, Acidic & Safe (based on acceptable levels of pH for drinking water.



```python
# df.PH = [x for v in df.PH if v]
PH_categorical = []
for v in df.PH:
    if v<6.5: PH_categorical.append('Alkaline')
    elif v>8.5: PH_categorical.append('Acidic')
    else: PH_categorical.append('Safe')
sns.countplot(x=PH_categorical)
```

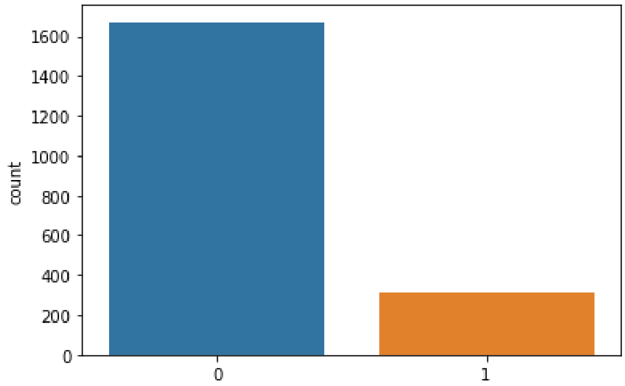
```
<AxesSubplot:ylabel='count'>
```





```python
# Drinkable (Disclaimer: This doesn't guarantee if the water sample considered is safe to drink)
drinkable = []
for i, row in df.iterrows():
    # Hard Test - No parameter should be falling out of acceptable range
    if(row['PH'] > 6.5 and row['PH'] < 8.5
        and row['D.O. (mg/l)'] > 6.5 and row['B.O.D. (mg/l)'] < 5.0
        and row['TDS'] < 1000 and row['Nitrate + Nitrite (mg/l)'] < 1.0
        and row['Total Coliform (MPN/100ml)'] < 500):
        drinkable.append(1)
    else: drinkable.append(0)
sns.countplot(x=drinkable)
```

```
<AxesSubplot:ylabel='count'>
```



Observation:

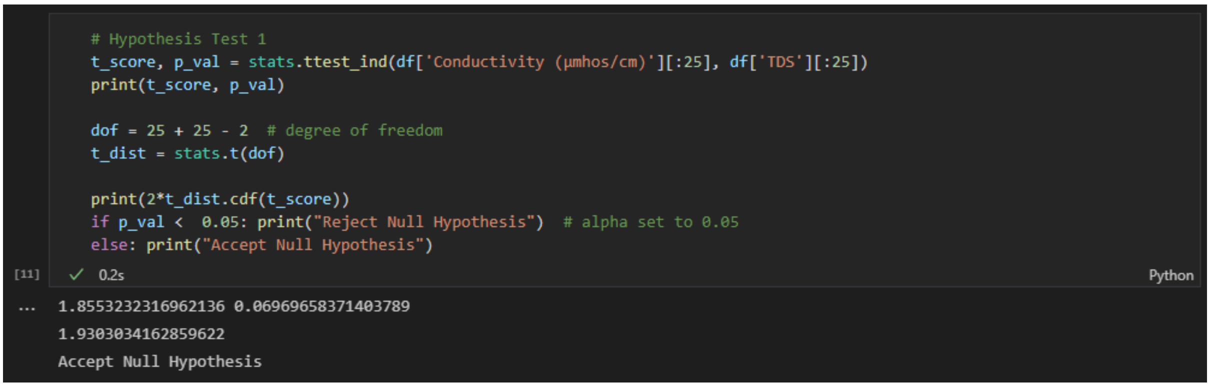
- Most pH values lie in the desired range.
- The knowledge can be improved or rather corrected with counting pH for same stations as one by the most recent year of testing.
- Though, most water bodies have a safe pH but are not really drinkable on considering other parameters like presence of Coliform or TDS.

Hypothesis Testing

Hypothesis Test 1:

Hypothesis 1 (H_0): There is a relationship between Conductivity and TDS as observed.

Alternate Hypothesis (Ha): There is no relation between Conductivity and TDS.



```
# Hypothesis Test 1
t_score, p_val = stats.ttest_ind(df['Conductivity (µmhos/cm)'][:25], df['TDS'][:25])
print(t_score, p_val)

dof = 25 + 25 - 2  # degree of freedom
t_dist = stats.t(dof)

print(2*t_dist.cdf(t_score))
if p_val <  0.05: print("Reject Null Hypothesis")  # alpha set to 0.05
else: print("Accept Null Hypothesis")
```

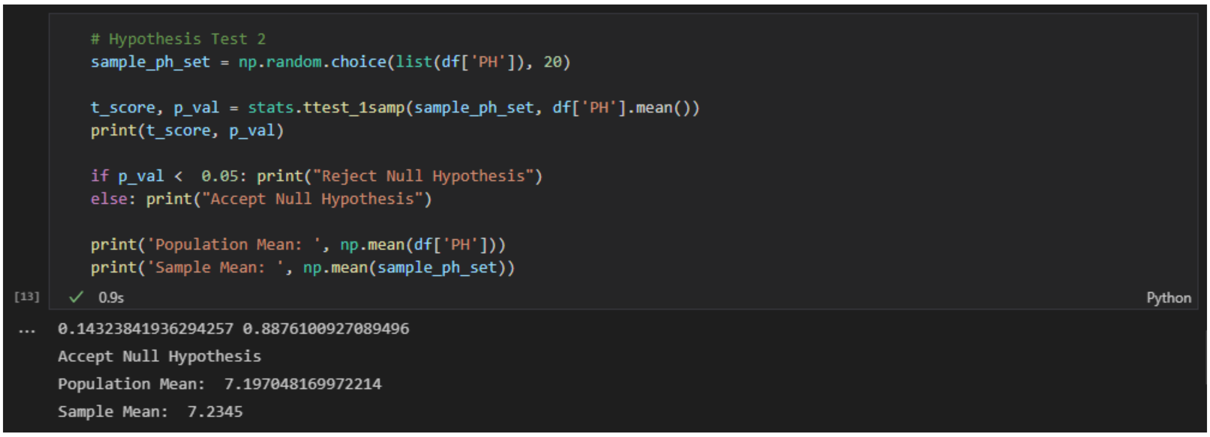
```
... 1.8553232316962136 0.06969658371403789
    1.9303034162859622
    Accept Null Hypothesis
```

Result: As p-value is more than 0.05, we accept the null hypothesis

Hypothesis Test 2:

Hypothesis 2 (H_0): The selected samples of PH from dataset, the sample mean is nearly equal to the mean of the population (the complete dataset).

Alternate Hypothesis (Ha): The sample & population mean are not equal, infact vary a lot.



```
# Hypothesis Test 2
sample_ph_set = np.random.choice(list(df['PH']), 20)

t_score, p_val = stats.ttest_1samp(sample_ph_set, df['PH'].mean())
print(t_score, p_val)

if p_val <  0.05: print("Reject Null Hypothesis")
else: print("Accept Null Hypothesis")

print('Population Mean: ', np.mean(df['PH']))
print('Sample Mean: ', np.mean(sample_ph_set))
```

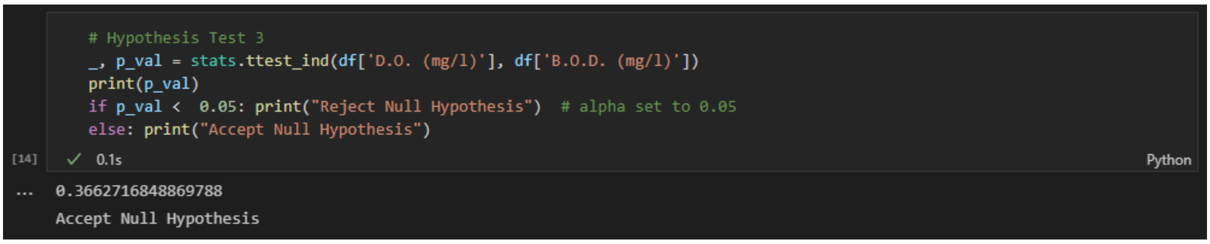
```
... 0.14323841936294257 0.8876100927089496
    Accept Null Hypothesis
    Population Mean:  7.197048169972214
    Sample Mean:  7.2345
```

Result: As p-value is more than 0.05, we accept the null hypothesis.
As we can see that the difference in mean of the sample and the population is very low.

Hypothesis Test 3:

Hypothesis 1 (H_0): D.O. and B.D.O. are correlated.

Alternate Hypothesis (Ha): There is no relation between Conductivity and TDS.



```
# Hypothesis Test 3
_, p_val = stats.ttest_ind(df['D.O. (mg/l)'], df['B.O.D. (mg/l)'])
print(p_val)
if p_val <  0.05: print("Reject Null Hypothesis")  # alpha set to 0.05
else: print("Accept Null Hypothesis")
```

```
... 0.3662716848869788
    Accept Null Hypothesis
```

Result: As p-value is more than 0.05, we accept the null hypothesis

Future Scope

- We can drop BDO, TDS, and Fecal Coliform to avoid multicollinearity problem as the part of Feature Selection process. We can also drop Location and State as Station Code serves the purpose.
- For binning the pH values as categories – Alkaline, Safe and Acidic, we can improve or rather correct it with considering a single pH for same stations as the most recent value by year of testing.
- We can try various classifiers to predict PH class of the water sample using the other given data.
- Outliers and imbalanced data problem need to be tackled.

Conclusion

We analysed the water quality dataset, gaining insights and understanding the importance of water quality. Similar analysis can be performed on such upcoming test results and quality check from more and different water bodies as well.
These values vary daily and are greatly impacted by natural and/or man-made phenomenon. Here, we also determined if water is drinkable. Though, such outcomes do not affirm the reality accurately, but we perceive it as a baseline result and thus, gain an idea about it.

Thank you,
Dilip Jain 😊