# Ivan David Alamu

# 10201100076

In [54]:
```python
import pandas as pd
import numpy as np
```

In [2]:
```python
data_water= pd.read_csv('water_potability.csv')
```

In [3]:
```python
data_water
```

Out[3]:

| | ph | Hardness | Solids | Chloramines | Sulfate | Conductivity | Organic_carbon | Trihalomethanes | Turbidity | Potability |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | NaN | 204.890455 | 20791.318981 | 7.300212 | 368.516441 | 564.308654 | 10.379783 | 86.990970 | 2.963135 | 0 |
| **1** | 3.716080 | 129.422921 | 18630.057858 | 6.635246 | NaN | 592.885359 | 15.180013 | 56.329076 | 4.500656 | 0 |
| **2** | 8.099124 | 224.236259 | 19909.541732 | 9.275884 | NaN | 418.606213 | 16.868637 | 66.420093 | 3.055934 | 0 |
| **3** | 8.316766 | 214.373394 | 22018.417441 | 8.059332 | 356.886136 | 363.266516 | 18.436524 | 100.341674 | 4.628771 | 0 |
| **4** | 9.092223 | 181.101509 | 17978.986339 | 6.546600 | 310.135738 | 398.410813 | 11.558279 | 31.997993 | 4.075075 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **3271** | 4.668102 | 193.681735 | 47580.991603 | 7.166639 | 359.948574 | 526.424171 | 13.894419 | 66.687695 | 4.435821 | 1 |
| **3272** | 7.808856 | 193.553212 | 17329.802160 | 8.061362 | NaN | 392.449580 | 19.903225 | NaN | 2.798243 | 1 |
| **3273** | 9.419510 | 175.762646 | 33155.578218 | 7.350233 | NaN | 432.044783 | 11.039070 | 69.845400 | 3.298875 | 1 |
| **3274** | 5.126763 | 230.603758 | 11983.869376 | 6.303357 | NaN | 402.883113 | 11.168946 | 77.488213 | 4.708658 | 1 |
| **3275** | 7.874671 | 195.102299 | 17404.177061 | 7.509306 | NaN | 327.459760 | 16.140368 | 78.698446 | 2.309149 | 1 |

3276 rows × 10 columns

In [4]:
```python
data_water.head()
```

Out[4]:

| | ph | Hardness | Solids | Chloramines | Sulfate | Conductivity | Organic_carbon | Trihalomethanes | Turbidity | Potability |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | NaN | 204.890455 | 20791.318981 | 7.300212 | 368.516441 | 564.308654 | 10.379783 | 86.990970 | 2.963135 | 0 |
| **1** | 3.716080 | 129.422921 | 18630.057858 | 6.635246 | NaN | 592.885359 | 15.180013 | 56.329076 | 4.500656 | 0 |
| **2** | 8.099124 | 224.236259 | 19909.541732 | 9.275884 | NaN | 418.606213 | 16.868637 | 66.420093 | 3.055934 | 0 |
| **3** | 8.316766 | 214.373394 | 22018.417441 | 8.059332 | 356.886136 | 363.266516 | 18.436524 | 100.341674 | 4.628771 | 0 |
| **4** | 9.092223 | 181.101509 | 17978.986339 | 6.546600 | 310.135738 | 398.410813 | 11.558279 | 31.997993 | 4.075075 | 0 |

In [7]:
```python
data_water.shape
```

Out[7]:
```
(3276, 10)
```

# 1. Implement all necessary data preprocessing on this data set

# Check for missing value

In [9]:
```python
data_water.isnull().sum()
```

Out[9]:
```
ph                 491
Hardness             0
Solids               0
Chloramines          0
Sulfate            781
Conductivity         0
Organic_carbon       0
Trihalomethanes    162
Turbidity            0
Potability           0
dtype: int64
```

In [10]:
```python
data_water.isnull().sum().sum()
```

Out[10]:
```
1434
```

So in the given dataset we have 1434 missing values

# Filling missing value

In this part we fill the missing value the 0

In [11]:
```python
 #filling the missing value with value= 0
    fill_missing_value= data_water.fillna(value=0)
```

In [14]:
```python
fill_missing_value.isnull().sum().sum()
```

Out[14]:
```
0
```

All the missing value have been replace by the value O by using this filling method we can see that we don't have any missing value in the dataset

In [16]:
```python
#fill the missing with previous value

fill_missing_value_previous= data_water.fillna(method='pad')
fill_missing_value_previous
```

Out[16]:

| | ph | Hardness | Solids | Chloramines | Sulfate | Conductivity | Organic_carbon | Trihalomethanes | Turbidity | Potability |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | NaN | 204.890455 | 20791.318981 | 7.300212 | 368.516441 | 564.308654 | 10.379783 | 86.990970 | 2.963135 | 0 |
| **1** | 3.716080 | 129.422921 | 18630.057858 | 6.635246 | 368.516441 | 592.885359 | 15.180013 | 56.329076 | 4.500656 | 0 |
| **2** | 8.099124 | 224.236259 | 19909.541732 | 9.275884 | 368.516441 | 418.606213 | 16.868637 | 66.420093 | 3.055934 | 0 |
| **3** | 8.316766 | 214.373394 | 22018.417441 | 8.059332 | 356.886136 | 363.266516 | 18.436524 | 100.341674 | 4.628771 | 0 |
| **4** | 9.092223 | 181.101509 | 17978.986339 | 6.546600 | 310.135738 | 398.410813 | 11.558279 | 31.997993 | 4.075075 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **3271** | 4.668102 | 193.681735 | 47580.991603 | 7.166639 | 359.948574 | 526.424171 | 13.894419 | 66.687695 | 4.435821 | 1 |
| **3272** | 7.808856 | 193.553212 | 17329.802160 | 8.061362 | 359.948574 | 392.449580 | 19.903225 | 66.687695 | 2.798243 | 1 |
| **3273** | 9.419510 | 175.762646 | 33155.578218 | 7.350233 | 359.948574 | 432.044783 | 11.039070 | 69.845400 | 3.298875 | 1 |
| **3274** | 5.126763 | 230.603758 | 11983.869376 | 6.303357 | 359.948574 | 402.883113 | 11.168946 | 77.488213 | 4.708658 | 1 |
| **3275** | 7.874671 | 195.102299 | 17404.177061 | 7.509306 | 359.948574 | 327.459760 | 16.140368 | 78.698446 | 2.309149 | 1 |

3276 rows × 10 columns

In [17]:
```python
fill_missing_value_previous.head()
```

Out[17]:

| | ph | Hardness | Solids | Chloramines | Sulfate | Conductivity | Organic_carbon | Trihalomethanes | Turbidity | Potability |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | NaN | 204.890455 | 20791.318981 | 7.300212 | 368.516441 | 564.308654 | 10.379783 | 86.990970 | 2.963135 | 0 |
| **1** | 3.716080 | 129.422921 | 18630.057858 | 6.635246 | 368.516441 | 592.885359 | 15.180013 | 56.329076 | 4.500656 | 0 |
| **2** | 8.099124 | 224.236259 | 19909.541732 | 9.275884 | 368.516441 | 418.606213 | 16.868637 | 66.420093 | 3.055934 | 0 |
| **3** | 8.316766 | 214.373394 | 22018.417441 | 8.059332 | 356.886136 | 363.266516 | 18.436524 | 100.341674 | 4.628771 | 0 |
| **4** | 9.092223 | 181.101509 | 17978.986339 | 6.546600 | 310.135738 | 398.410813 | 11.558279 | 31.997993 | 4.075075 | 0 |

In [19]:
```python
fill_missing_value_previous.isnull().sum()
```

Out[19]:
```
ph                 1
Hardness           0
Solids             0
Chloramines        0
Sulfate            0
Conductivity       0
Organic_carbon     0
Trihalomethanes    0
Turbidity          0
Potability         0
dtype: int64
```

we notice that when we try to fill the missing value with previous value we still have 1 missing value so we will try another way to fill the missing value

# remove missing value

In [24]:
```python
remove_missing_value= data_water.dropna()
remove_missing_value.isnull().sum()
```

Out[24]:
```
ph                 0
Hardness           0
Solids             0
Chloramines        0
Sulfate            0
Conductivity       0
Organic_carbon     0
Trihalomethanes    0
Turbidity          0
Potability         0
dtype: int64
```

In [26]:
```python
remove_missing_value.isnull().sum().sum()
```

Out[26]:
```
0
```

by using the methode dropna() we remove all the missing value in the dataset.

# 2.Define an outlier and give its importance in data analysis.

An outlier is an observation, data point, or value within a dataset that significantly deviates from the majority of the data. It is an extreme value that is either much larger or much smaller than most other data points. Outliers can be univariate (outliers in a single variable) or multivariate (outliers in multiple variables simultaneously).

Importance of Outliers in Data Analysis:

1.Error Detection and Data Quality Assurance: Outliers often signify errors or anomalies in the data, such as data entry mistakes or measurement errors. Identifying and addressing outliers is crucial for maintaining data quality and ensuring the reliability and accuracy of analysis results.

2.Impact on Statistical Measures: Outliers can significantly influence summary statistics like the mean and standard deviation. They can skew these measures, potentially leading to incorrect conclusions about the central tendencies and variability of the dataset. Recognizing and handling outliers is vital for robust and meaningful statistical analysis.

3.Insight Generation and Anomaly Detection: Outliers can provide valuable insights into the data. They may represent rare events, exceptional cases, or unusual patterns that might otherwise go unnoticed. In some cases, outliers are precisely what analysts are interested in, as they can point to critical observations or anomalies in the data that require special attention or investigation.

# 3. Explain three common ways of detecting outliers in the given dataset.

1.Z-Score Method:

Calculate the Z-score for each data point, which measures how many standard deviations it is from the mean. Data points with high absolute Z-scores (e.g., greater than 2 or 3) are considered outliers.

2.IQR (Interquartile Range) Method:

Calculate the interquartile range (IQR) as the range between the first quartile (Q1) and the third quartile (Q3). Data points beyond Q1 - 1.5 *IQR or Q3 + 1.5* IQR are considered outliers.

3.Visual Inspection:

Create visualizations like box plots, scatter plots, or histograms to identify data points that lie far from the bulk of the data. Outliers are data points that appear as individual points far from the central data cluster.

In [ ]: