

# Water Quality Analysis

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

```
In [3]: df=pd.read_csv("water_potability.csv")
```

```
In [4]: df.head(5)
```

```
Out[4]:      ph  Hardness  Solids  Chloramines  Sulfate  Conductivity  Organic_c  
0       NaN  204.890455  20791.318981    7.300212  368.516441  564.308654  10.1  
1  3.716080  129.422921  18630.057858    6.635246  NaN  592.885359  15.1  
2  8.099124  224.236259  19909.541732    9.275884  NaN  418.606213  16.1  
3  8.316766  214.373394  22018.417441    8.059332  356.886136  363.266516  18.1  
4  9.092223  181.101509  17978.986339    6.546600  310.135738  398.410813  11.1
```

```
In [5]: df.shape
```

```
Out[5]: (3276, 10)
```

```
In [6]: df.columns
```

```
Out[6]: Index(['ph', 'Hardness', 'Solids', 'Chloramines', 'Sulfate', 'Conductivity',  
              'Organic_carbon', 'Trihalomethanes', 'Turbidity', 'Potability'],  
              dtype='object')
```

```
In [8]: df.isnull().sum()
```

```
Out[8]: ph          491  
Hardness        0  
Solids          0  
Chloramines      0  
Sulfate          781  
Conductivity      0  
Organic_carbon     0  
Trihalomethanes   162  
Turbidity          0  
Potability         0  
dtype: int64
```

```
In [9]: df.dropna(inplace=True)
```

```
In [11]: df.isnull().sum()
```

```
Out[11]: ph          0  
Hardness      0  
Solids        0  
Chloramines    0  
Sulfate        0  
Conductivity   0  
Organic_carbon 0  
Trihalomethanes 0  
Turbidity      0  
Potability     0  
dtype: int64
```

```
In [12]: df.describe()
```

```
Out[12]:      ph  Hardness  Solids  Chloramines  Sulfate  Conductivity  
count  2011.000000  2011.000000  2011.000000  2011.000000  2011.000000  2011.000000  
mean   7.085990   195.968072  21917.441374   7.134338   333.224672   426.526409  
std    1.573337   32.635085   8642.239815   1.584820   41.205172   80.712572  
min    0.227499   73.492234   320.942611   1.390871   129.000000  201.619737  
25%   6.089723   176.744938  15615.665390   6.138895   307.632511  366.680307  
50%   7.027297   197.191839  20933.512750   7.143907   332.232177  423.455906  
75%   8.052969   216.441070  27182.587067   8.109726   359.330555  482.373169  
max   14.000000  317.338124  56488.672413  13.127000  481.030642  753.342620
```

```
In [14]: df.nunique()
```

```
Out[14]: ph            2011  
Hardness        2011  
Solids          2011  
Chloramines     2011  
Sulfate          2011  
Conductivity    2011  
Organic_carbon  2011  
Trihalomethanes 2011  
Turbidity        2011  
Potability       2  
dtype: int64
```

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

```
<class 'pandas.core.frame.DataFrame'>
Index: 2011 entries, 3 to 3271
Data columns (total 10 columns):
 #   Column            Non-Null Count  Dtype  
--- 
 0   ph                2011 non-null    float64
 1   Hardness          2011 non-null    float64
 2   Solids            2011 non-null    float64
 3   Chloramines       2011 non-null    float64
 4   Sulfate           2011 non-null    float64
 5   Conductivity      2011 non-null    float64
 6   Organic_carbon    2011 non-null    float64
 7   Trihalomethanes  2011 non-null    float64
 8   Turbidity         2011 non-null    float64
 9   Potability        2011 non-null    int64  
dtypes: float64(9), int64(1)
memory usage: 172.8 KB
```

In [17]: `df.dtypes`

```
Out[17]: ph                  float64
          Hardness            float64
          Solids              float64
          Chloramines         float64
          Sulfate             float64
          Conductivity        float64
          Organic_carbon      float64
          Trihalomethanes     float64
          Turbidity            float64
          Potability           int64  
dtype: object
```

## correlations

In [45]: `df.corr`

```

Out[45]: <bound method DataFrame.corr of
          amines      Sulfate  \
          3    8.316766  214.373394  22018.417441    8.059332  356.886136
          4    9.092223  181.101509  17978.986339   6.546600  310.135738
          5    5.584087  188.313324  28748.687739   7.544869  326.678363
          6   10.223862  248.071735  28749.716544   7.513408  393.663396
          7    8.635849  203.361523  13672.091764   4.563009  303.309771
          ...
          ...
          ...
          ...
          3267  8.989900  215.047358  15921.412018   6.297312  312.931022
          3268  6.702547  207.321086  17246.920347   7.708117  304.510230
          3269  11.491011  94.812545  37188.826022   9.263166  258.930600
          3270  6.069616  186.659040  26138.780191   7.747547  345.700257
          3271  4.668102  193.681735  47580.991603   7.166639  359.948574

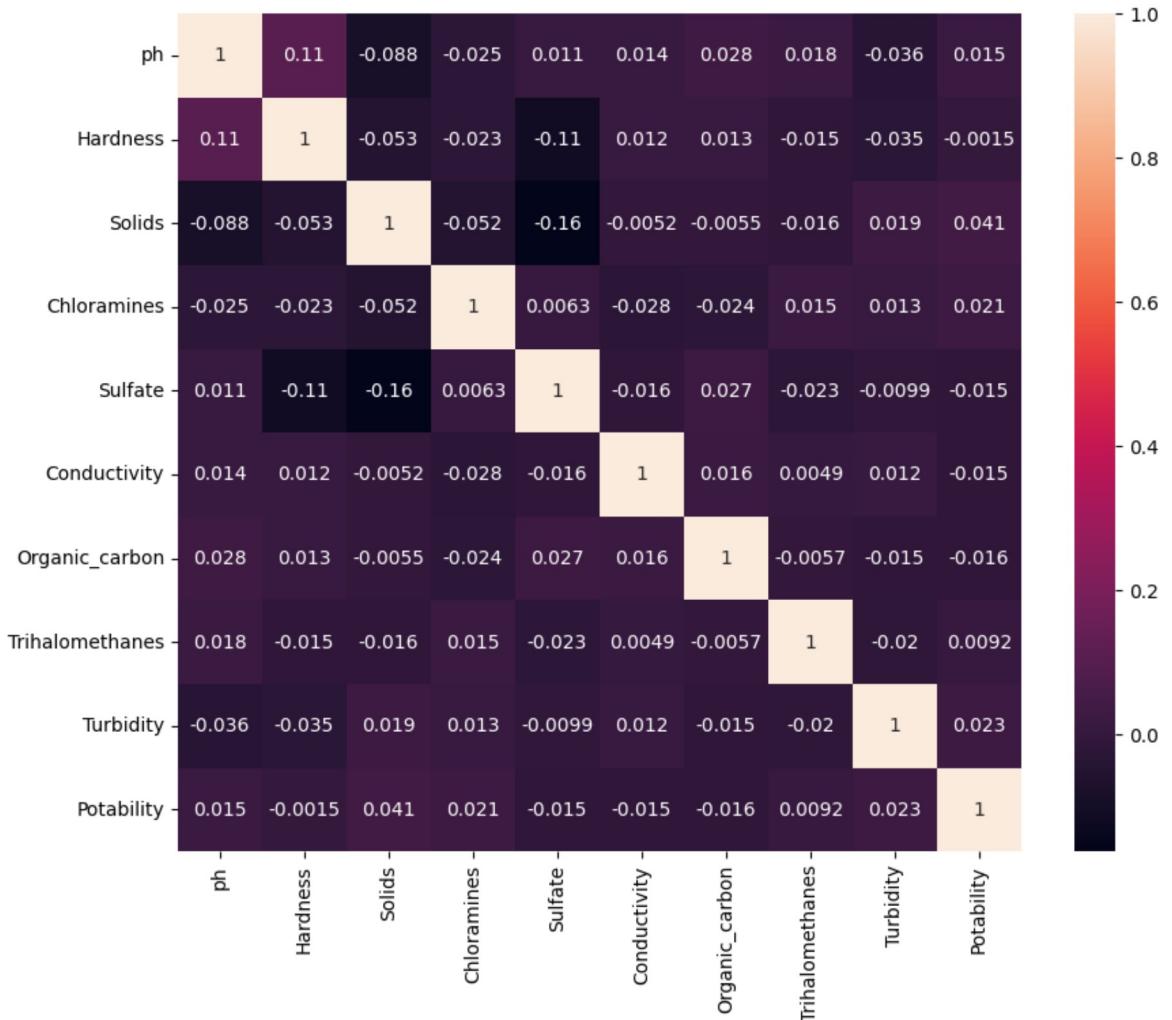
          Conductivity  Organic_carbon  Trihalomethanes  Turbidity  Potability
          3    363.266516        18.436524       100.341674    4.628771        0
          4    398.410813        11.558279       31.997993    4.075075        0
          5    280.467916        8.399735       54.917862    2.559708        0
          6    283.651634       13.789695       84.603556    2.672989        0
          7    474.607645       12.363817       62.798309    4.401425        0
          ...
          ...
          ...
          ...
          3267  390.410231        9.899115       55.069304    4.613843        1
          3268  329.266002       16.217303       28.878601    3.442983        1
          3269  439.893618       16.172755       41.558501    4.369264        1
          3270  415.886955       12.067620       60.419921    3.669712        1
          3271  526.424171       13.894419       66.687695    4.435821        1

```

[2011 rows x 10 columns]>

```
In [32]: import matplotlib.pyplot as plt  
        import seaborn as sns
```

```
In [33]: df['Potability'].value_counts()
```

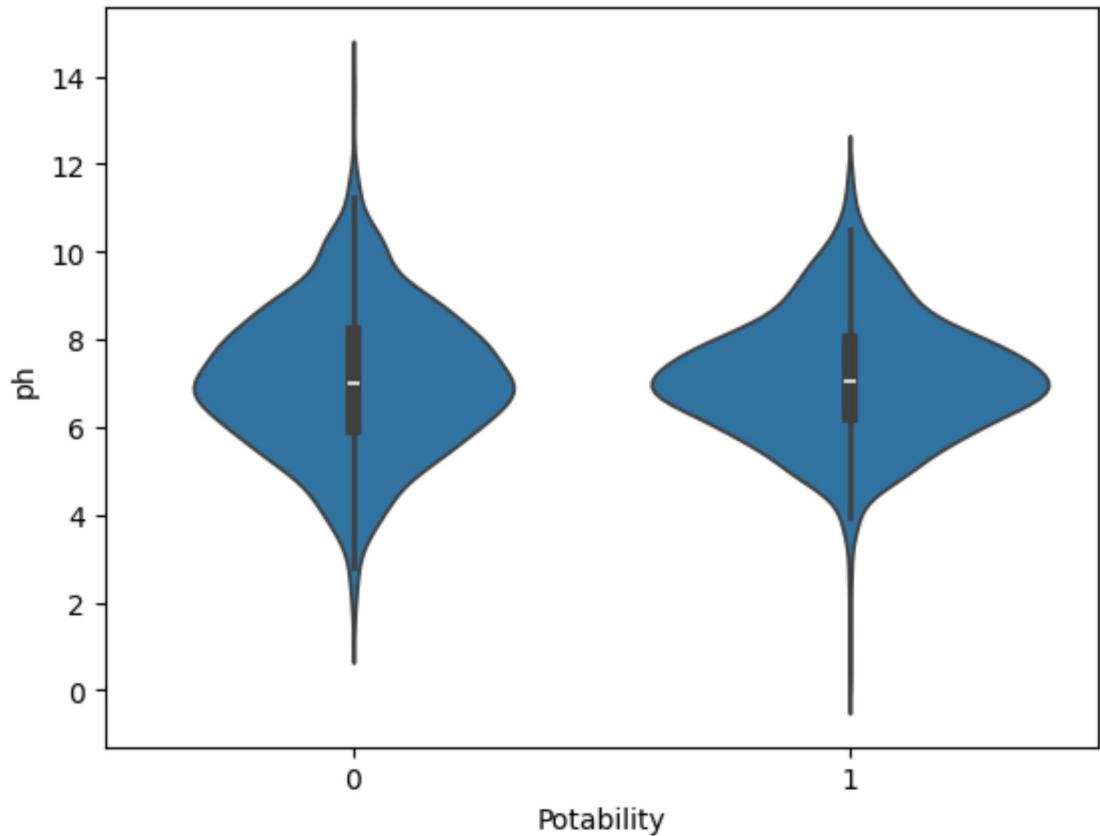


```
In [36]: df['Potability'].value_counts()
```

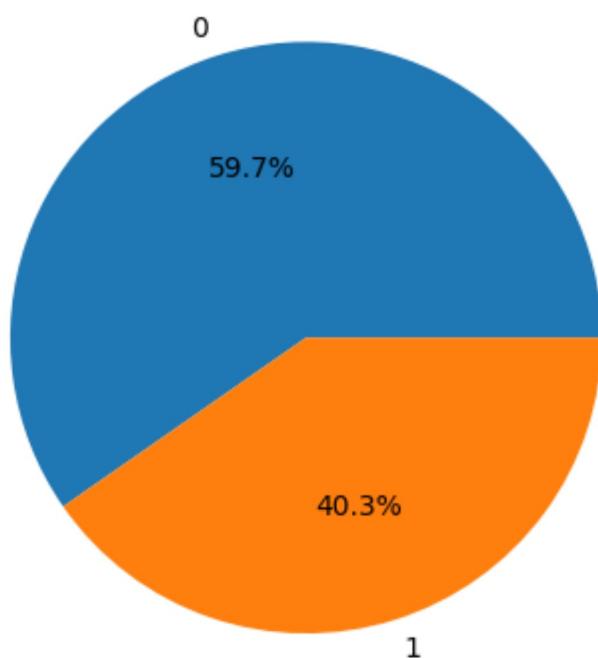
```
Out[36]: Potability
0    1200
1     811
Name: count, dtype: int64
```

```
In [37]: sns.violinplot(x='Potability',y='ph',data=df)
```

```
Out[37]: <Axes: xlabel='Potability', ylabel='ph'>
```



```
In [38]: plt.pie(df['Potability'].value_counts(), labels = list(df['Potability'].unique()))
plt.show()
```



```
In [39]: df
```

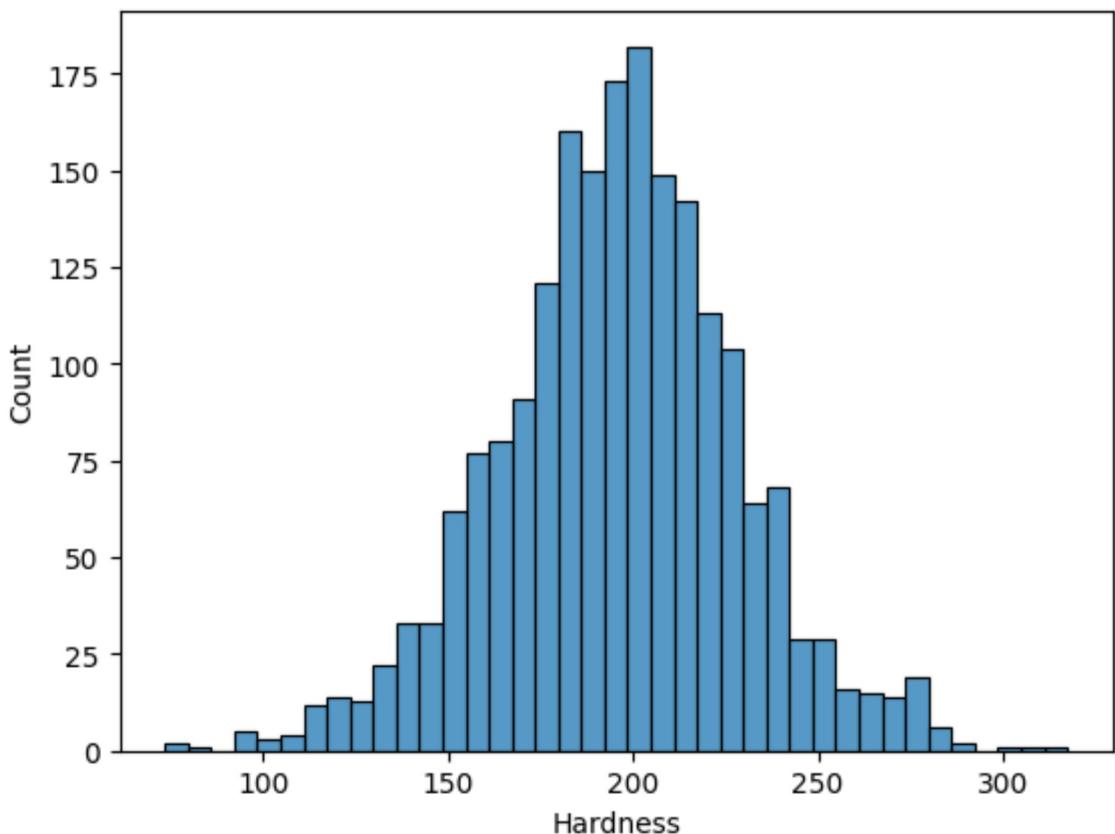
Out[39]:

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic
3	8.316766	214.373394	22018.417441	8.059332	356.886136	363.266516	
4	9.092223	181.101509	17978.986339	6.546600	310.135738	398.410813	
5	5.584087	188.313324	28748.687739	7.544869	326.678363	280.467916	
6	10.223862	248.071735	28749.716544	7.513408	393.663396	283.651634	
7	8.635849	203.361523	13672.091764	4.563009	303.309771	474.607645	
...	...	...	...	...	...	...	...
3267	8.989900	215.047358	15921.412018	6.297312	312.931022	390.410231	
3268	6.702547	207.321086	17246.920347	7.708117	304.510230	329.266002	
3269	11.491011	94.812545	37188.826022	9.263166	258.930600	439.893618	
3270	6.069616	186.659040	26138.780191	7.747547	345.700257	415.886955	
3271	4.668102	193.681735	47580.991603	7.166639	359.948574	526.424171	

2011 rows × 10 columns

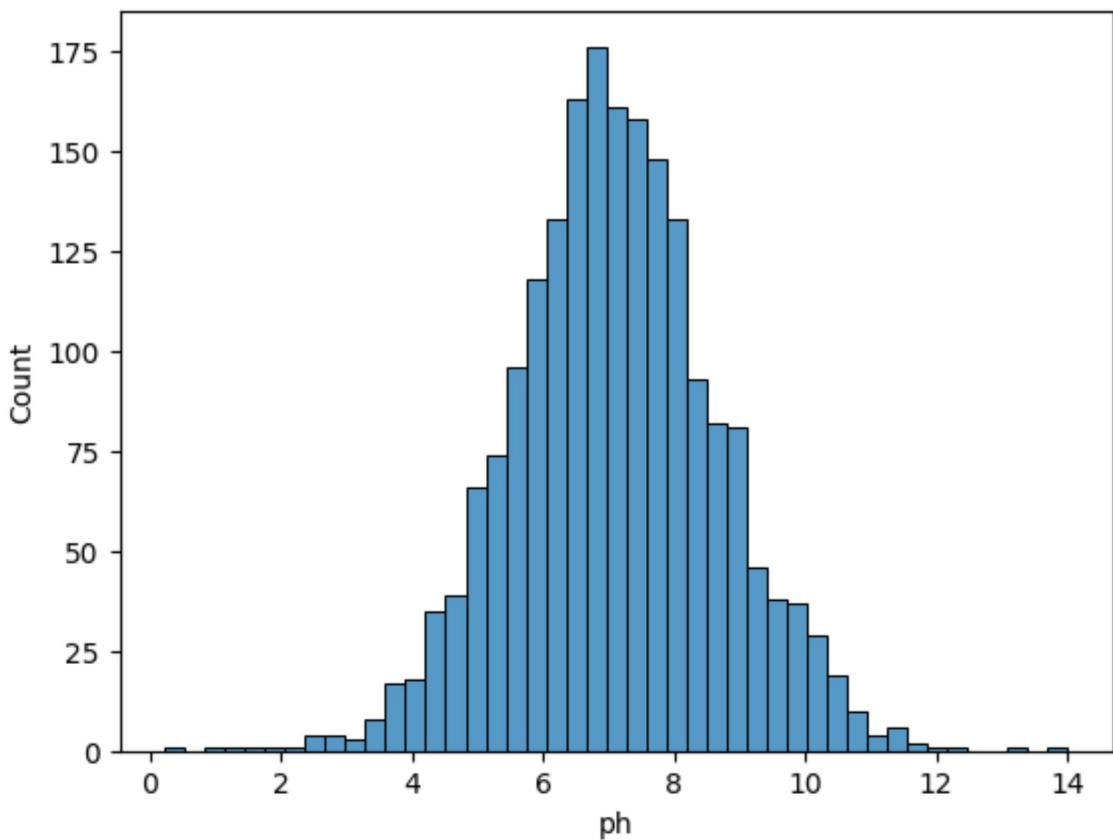
In [40]: `sns.histplot(df['Hardness'])`

Out[40]: <Axes: xlabel='Hardness', ylabel='Count'>

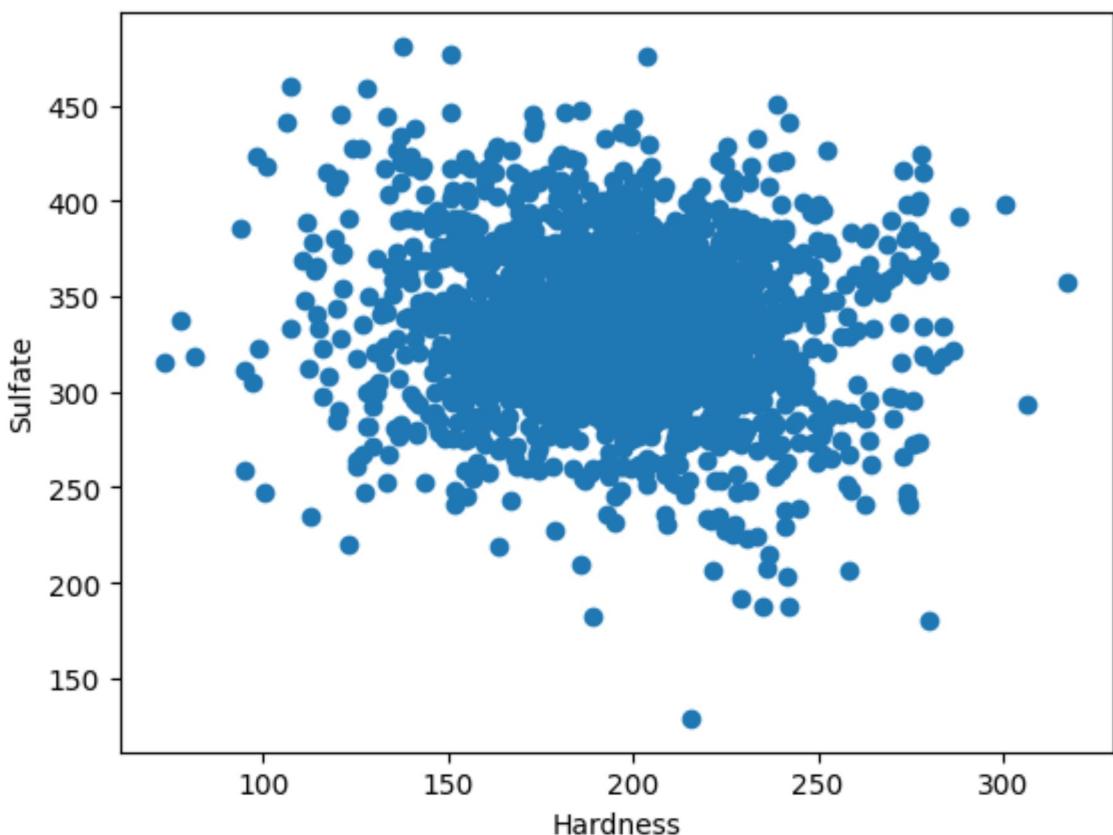


In [41]: `sns.histplot(df['ph'])`

Out[41]: <Axes: xlabel='ph', ylabel='Count'>

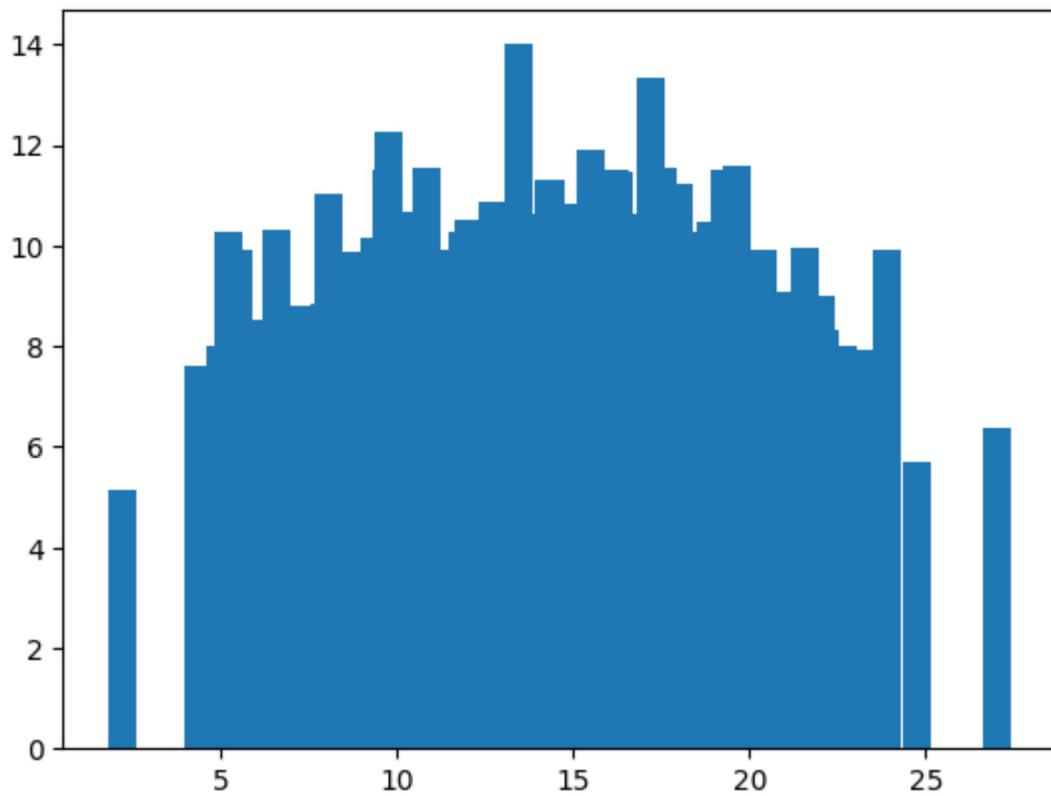


```
In [42]: gp = plt.scatter(df['Hardness'],df['Sulfate'])
plt.xlabel('Hardness')
plt.ylabel('Sulfate')
plt.show(gp)
```



```
In [43]: plt.bar(df['Organic_carbon'],df['ph'])
```

Out[43]: <BarContainer object of 2011 artists>



In [44]: df

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon
3	8.316766	214.373394	22018.417441	8.059332	356.886136	363.266516	1.0
4	9.092223	181.101509	17978.986339	6.546600	310.135738	398.410813	1.0
5	5.584087	188.313324	28748.687739	7.544869	326.678363	280.467916	1.0
6	10.223862	248.071735	28749.716544	7.513408	393.663396	283.651634	1.0
7	8.635849	203.361523	13672.091764	4.563009	303.309771	474.607645	1.0
...	...	...	...	...	...	...	...
3267	8.989900	215.047358	15921.412018	6.297312	312.931022	390.410231	1.0
3268	6.702547	207.321086	17246.920347	7.708117	304.510230	329.266002	1.0
3269	11.491011	94.812545	37188.826022	9.263166	258.930600	439.893618	1.0
3270	6.069616	186.659040	26138.780191	7.747547	345.700257	415.886955	1.0
3271	4.668102	193.681735	47580.991603	7.166639	359.948574	526.424171	1.0

2011 rows × 10 columns

In [ ]: