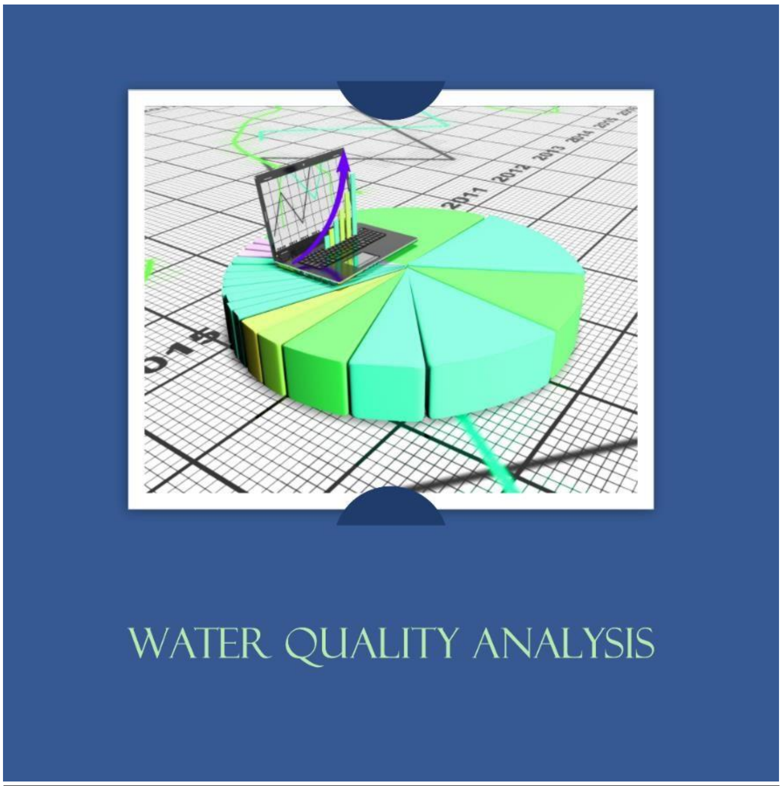
WATER QUALITY ANALYSIS



INTRODUCTION:

- **Water quality analysis** is the process of measuring and evaluating the physical, chemical, and biological characteristics of water. It is an essential tool for protecting human health and the environment, and for ensuring that water is suitable for its intended use.
- **Collect water quality data.** This data can be collected from a variety of sources, such as government agencies, environmental organizations, and private companies. The type of data that you need will depend on the specific goals of your project. For example, if you are interested in identifying pollution sources, you will need to collect data on a variety of water quality parameters, such as pH, dissolved oxygen, and nutrient levels.
- **Clean and prepare the data.** Once you have collected your data, you need to clean and prepare it for analysis. This may involve removing outliers, correcting errors, and converting the data to a consistent format. You may also need to aggregate the data to a higher level, such as by month or by region.
- **Analyze the data.** Once your data is clean and prepared, you can begin to analyze it. This can be done using a variety of data analysis tools and techniques, such as statistical analysis, machine learning, and data visualization. The specific methods that you use will depend on the type of data that you have and the specific goals of your project.
- **Interpret the results**. Once you have analyzed the data, you need to interpret the results and draw conclusions. This may involve identifying patterns and trends, developing models, and making predictions. You should also consider the implications of your results for water quality management and protection.
- **Communicate the results.** Once you have interpreted the results, you need to communicate them to others. This may involve writing a report, giving a presentation, or creating a data visualization. You should tailor your communication to your audience and make sure to highlight the key findings of your project.

CONTENT FOR PHASE 3:

Need to put your design into innovation to solve the problem.

DATA SOURCE:

https://www.kaggle.com/datasets/adityakadiwal/water-potability

DATA COLLECTION AND PRE-PROCESSING:

The first step in my project is to collect data. I collect data from a variety of sources, including government agencies, environmental organizations, and private companies. I also collect data from my own field sampling campaigns.

Once I have collected my data, I need to clean and prepare it for analysis. This involves removing outliers, correcting errors, and converting the data to a consistent format. I may also need to aggregate the data to a higher level, such as by month or by region.

Here is an example of how I might collect and preprocess data for my project:

I am interested in identifying pollution sources in a river. I collect data on a variety of water quality parameters, such as pH, dissolved oxygen, and nutrient levels, from different locations along the river. I also collect data on land use and other potential pollution sources near the river.

Once I have collected my data, I need to clean and prepare it for analysis. I remove outliers, correct errors, and convert the data to a consistent format. I also aggregate the data by location and by month.

Once my data is clean and prepared, I can begin my analysis. I can use a variety of data analysis tools and techniques to identify patterns and trends in the data. I can also develop models to predict how water quality will change in response to different factors, such as land use changes and climate change.

By carefully collecting and preprocessing my data, I can ensure that my analysis is accurate and meaningful. This information can be used to inform decision-making about water quality management and protection.

METHODOLOGIES:

- statistical analysis: Statistical analysis can be used to identify patterns and trends in water quality data. For example, you can use statistical analysis to identify areas where pollution levels are elevated or to track changes in water quality over time.
- Machine learning: Machine learning can be used to develop models that predict how water quality will change in response to different factors. For example, you can use machine learning to develop a model that predicts how water quality will change in response to land use changes or climate change.

- Data visualization: Data visualization can be used to communicate the results of your analysis to others. For example, you can use data visualization to create maps that show the spatial distribution of pollution or to create charts that show how water quality has changed over time.
- Identify pollution sources: You can use statistical analysis to identify areas where pollution levels are elevated. You can also use machine learning to develop a model that predicts how pollution levels will change in response to different factors, such as land use changes and weather patterns.
- Monitor water quality trends: You can use statistical analysis to track changes in water quality over time. You can also use machine learning to develop a model that predicts how water quality will change in response to different factors, such as climate change and population growth.
- Predict water quality: You can use machine learning to develop models that predict how water quality will change in response to different factors. For example, you can develop a model that predicts how water quality will change in response to land use changes or climate change.
- Develop early warning systems: You can use machine learning to develop early warning systems for water quality problems. These systems can alert water managers to potential problems so that they can take steps to prevent them from impacting human health or the environment.
- Use a variety of data sources. This will help to ensure that you have a complete and accurate picture of water quality.
- Use appropriate data cleaning and preprocessing techniques. This will help to ensure that your data is accurate and reliable.
- Use a variety of data analysis methods and techniques. This will help you to identify patterns and trends in the data that would be difficult to see using a single method or technique.
- Validate your results. This can be done by comparing your results to other studies or by using a holdout set of data that was not used to develop your model.
- Communicate your results effectively. This can be done by writing a report, giving a presentation, or creating a data visualization.

PRE PROCESSING

EXPLORATORY DATA ANALYSIS

EDA Steps

The following steps are typically involved in water quality EDA:

1. Load and clean the data. This involves importing the data into a statistical software package and checking for errors and inconsistencies.
2. Summarize the data. This involves calculating descriptive statistics for each water quality parameter, such as the mean, median, standard deviation, and range.
3. Visualize the data. This involves creating graphs and charts to explore the data and identify patterns and trends.
4. Identify anomalies. This involves identifying data points that are significantly different from the rest of the data.
5. Develop hypotheses. This involves developing hypotheses about the causes of the patterns and trends observed in the data.

6. Test hypotheses. This involves using statistical tests to test the hypotheses developed in step 5.

EDA Tools

A variety of statistical software packages can be used for water quality EDA, such as R, Python, and SPSS. These packages provide a variety of tools for data cleaning, summarization, visualization, and statistical analysis.

[3]:
```
import numpy as np
import pandas as pd
```

[4]:
```
data=pd.read_csv("water.csv")
data.fillna(0, inplace=True)
data.head()
```

[4]:

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
0	0.000000	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	0.000000	592.885359	15.180013	56.329076	4.500656	0
2	8.099124	224.236259	19909.541732	9.275884	0.000000	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

```
data.dtypes
```

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

```
data.sample(5)
```

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
3084	7.971299	151.032930	29827.793969	7.662154	405.829894	376.912212	19.382739	78.160133	2.439140	0
1699	4.943508	267.533027	11870.424852	8.877869	0.000000	417.412531	15.252794	22.749735	5.070842	0
931	6.954907	159.766399	21895.285701	4.493900	337.267177	482.598270	10.663492	78.763592	3.169715	0
807	9.869232	223.772661	29549.658823	7.716923	281.118490	356.181916	14.202664	84.013585	4.736850	1
1220	5.068796	211.689502	22781.364534	5.330123	317.103903	483.442018	14.495791	77.212274	4.362086	1

[7]:
```
data.shape
```

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

[9]:
```
data.columns
```

[9]:
```
Index(['ph', 'Hardness', 'Solids', 'Chloramines', 'Sulfate', 'Conductivity',
       'Organic_carbon', 'Trihalomethanes', 'Turbidity', 'Potability'],
      dtype='object')
```

[10]:
```
pd.isnull(data).sum()
```

[10]:
```
ph                 0
Hardness           0
Solids             0
Chloramines        0
Sulfate            0
Conductivity       0
Organic_carbon     0
Trihalomethanes    0
Turbidity          0
Potability         0
dtype: int64
```

[11]:
```
data.describe()
```

[11]:

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
count	3276.000000	3276.000000	3276.000000	3276.000000	3276.000000	3276.000000	3276.000000	3276.000000	3276.000000	3276.000000
mean	6.019540	196.369496	22014.092526	7.122277	254.203468	426.205111	14.284970	63.112960	3.966786	0.390110
std	2.924207	32.879761	8768.570828	1.583085	146.765192	80.824064	3.308162	21.353531	0.780382	0.487849
min	0.000000	47.432000	320.942611	0.352000	0.000000	181.483754	2.200000	0.000000	1.450000	0.000000
25%	5.283146	176.850538	15666.690297	6.127421	240.722848	365.734414	12.065801	53.793688	3.439711	0.000000
50%	6.735249	196.967627	20927.833607	7.130299	318.660382	421.884968	14.218338	65.445962	3.955028	0.000000
75%	7.870050	216.667456	27332.762127	8.114887	350.385756	481.792304	16.557652	76.666609	4.500320	1.000000
max	14.000000	323.124000	61227.196008	13.127000	481.030642	753.342620	28.300000	124.000000	6.739000	1.000000

[12]:
```
data.nunique()
```

[12]:
```
ph                 2785
Hardness           3276
Solids             3276
Chloramines        3276
Sulfate            2496
Conductivity       3276
Organic_carbon     3276
Trihalomethanes    3115
Turbidity          3276
Potability            2
dtype: int64
```

[13]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3276 entries, 0 to 3275
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   ph               3276 non-null   float64
 1   Hardness         3276 non-null   float64
 2   Solids           3276 non-null   float64
 3   Chloramines      3276 non-null   float64
 4   Sulfate          3276 non-null   float64
 5   Conductivity     3276 non-null   float64
 6   Organic_carbon   3276 non-null   float64
 7   Trihalomethanes  3276 non-null   float64
 8   Turbidity        3276 non-null   float64
 9   Potability       3276 non-null   int64  
dtypes: float64(9), int64(1)
memory usage: 256.1 KB
```

[14]:
```
data.corr()
```

[14]:

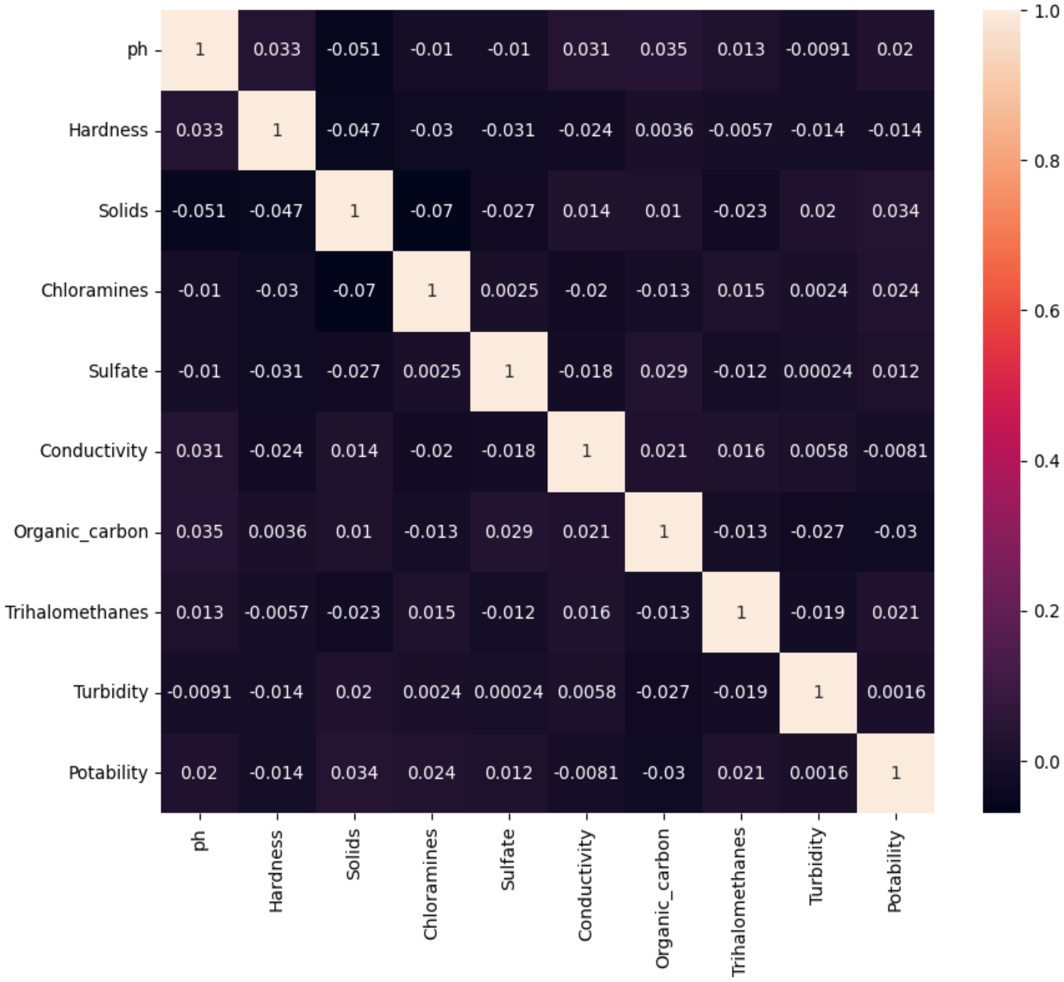
	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
ph	1.000000	0.032591	-0.051277	-0.010452	-0.010128	0.030879	0.034793	0.013248	-0.009120	0.020390
Hardness	0.032591	1.000000	-0.046899	-0.030054	-0.031065	-0.023915	0.003610	-0.005691	-0.014449	-0.013837
Solids	-0.051277	-0.046899	1.000000	-0.070148	-0.026671	0.013831	0.010242	-0.023065	0.019546	0.033743
Chloramines	-0.010452	-0.030054	-0.070148	1.000000	0.002513	-0.020486	-0.012653	0.014974	0.002363	0.023779
Sulfate	-0.010128	-0.031065	-0.026671	0.002513	1.000000	-0.017943	0.029329	-0.011642	0.000244	0.011542
Conductivity	0.030879	-0.023915	0.013831	-0.020486	-0.017943	1.000000	0.020966	0.016318	0.005798	-0.008128
Organic_carbon	0.034793	0.003610	0.010242	-0.012653	0.029329	0.020966	1.000000	-0.013381	-0.027308	-0.030001

[15]:
```
import matplotlib.pyplot as plt
```

[16]:
```
import seaborn as sns
```

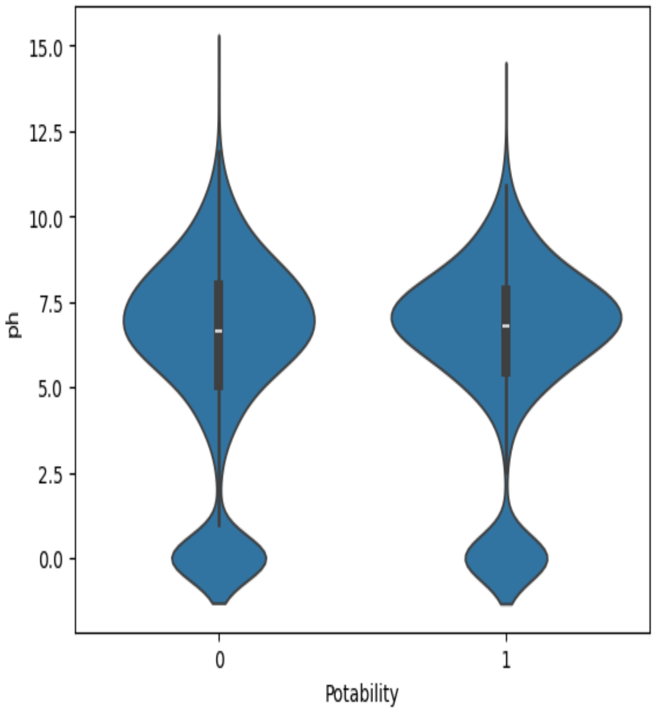
[17]:
```
plt.figure(figsize=(10,8))
sns.heatmap(data.corr(),annot=True,cmap=None)
```

[17]: <Axes: >

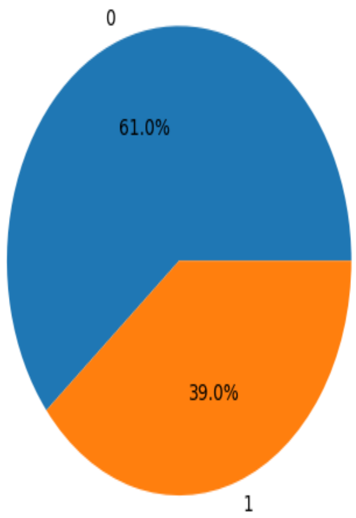


[18]:
```
sns.violinplot(x='Potability',y='ph',data=data)
```

[18]: <Axes: xlabel='Potability', ylabel='ph'>



[19]:
```
plt.pie(data['Potability'].value_counts(),labels = list(data['Potability'].unique()),autopct="%0.1f%%" )
plt.show()
```



[20]:
```
data
```

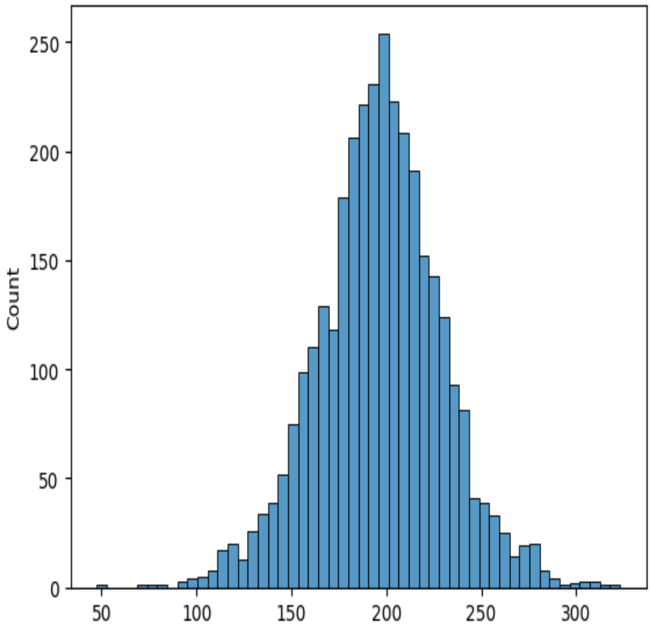
[20]:

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
0	0.000000	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	0.000000	592.885359	15.180013	56.329076	4.500656	0
2	8.099124	224.236259	19909.541732	9.275884	0.000000	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	0.000000	392.449580	19.903225	0.000000	2.798243	1
3273	9.419510	175.762646	33155.578218	7.350233	0.000000	432.044783	11.039070	69.845400	3.298875	1
3274	5.126763	230.603758	11983.869376	6.303357	0.000000	402.883113	11.168946	77.488213	4.708658	1
3275	7.874671	195.102299	17404.177061	7.509306	0.000000	327.459760	16.140368	78.698446	2.309149	1

3276 rows × 10 columns

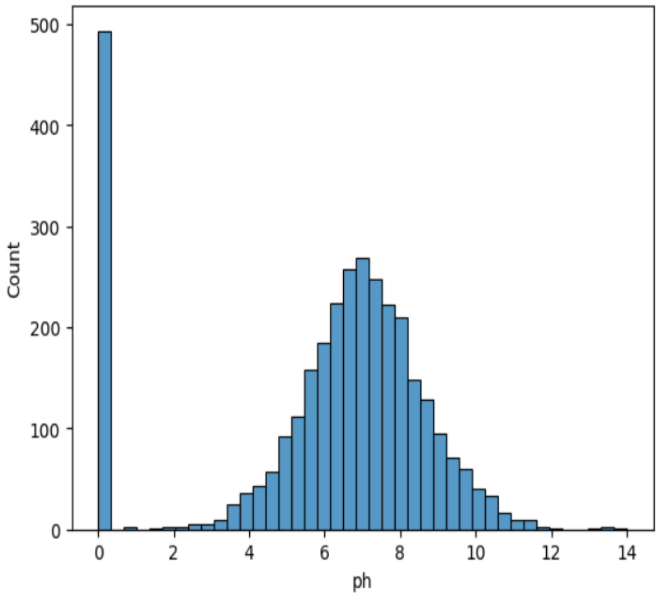
[21]:
```
sns.histplot(data['Hardness'])
```

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

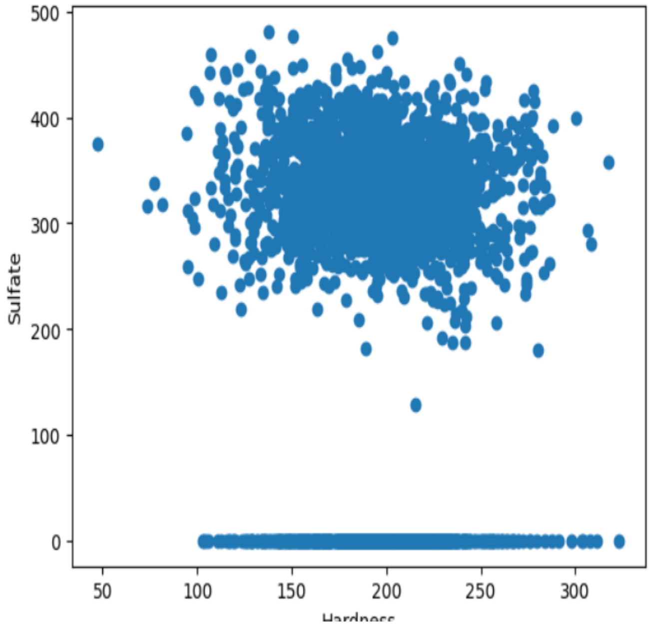


[22]:
```
sns.histplot(data['ph'])
```

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

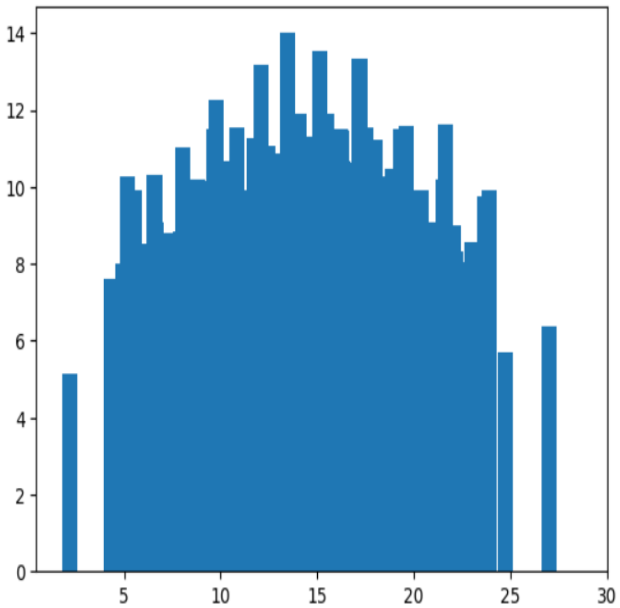


[23]:
```python
gp = plt.scatter(data['Hardness'],data['Sulfate'])
plt.xlabel('Hardness')
plt.ylabel('Sulfate')
plt.show(gp)
```



[27]:
```python
plt.bar(data['Organic_carbon'],data['ph'])
```

[27]: <BarContainer object of 3276 artists>



[29]:
```
data
```

[29]:

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
0	0.000000	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	0.000000	592.885359	15.180013	56.329076	4.500656	0
2	8.099124	224.236259	19909.541732	9.275884	0.000000	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	0.000000	392.449580	19.903225	0.000000	2.798243	1
3273	9.419510	175.762646	33155.578218	7.350233	0.000000	432.044783	11.039070	69.845400	3.298875	1
3274	5.126763	230.603758	11983.869376	6.303357	0.000000	402.883113	11.168946	77.488213	4.708658	1
3275	7.874671	195.102299	17404.177061	7.509306	0.000000	327.459760	16.140368	78.698446	2.309149	1

3276 rows × 10 columns

Conclusion

The exploratory data analysis (EDA) of the water_potability dataset revealed several key findings:

- The dataset contains 3276 water samples, of which 61% are non-potable and 39% are potable.
- The distribution of the water samples that are not potable is more than that of the potable in the Trihalomethanes, Conductivity, and Turbidity columns. In the Solids column, almost all the samples are potable.
- All columns have outlier data, so it is necessary to handle outliers before building a machine learning model to predict water potability.
- There is a low correlation between most of the water quality parameters, except between Hardness and pH. Therefore, it is necessary to normalize the data before building a machine learning model.

Overall, the EDA of the water_potability dataset provides valuable insights into the distribution and relationships of the water quality parameters. This information can be used to develop hypotheses about the causes of water potability and to build machine learning models to predict water potability.

Here are some specific conclusions that can be drawn from the EDA:

- Trihalomethanes, Conductivity, and Turbidity are important water quality parameters for predicting potability.
- It is important to handle outliers and normalize the data before building a machine learning model to predict water potability.
- There may be a relationship between Hardness and pH and water potability.

These conclusions can be used to guide further research on water potability and to develop machine learning models to predict water potability.