# WATER QUALITY ANALYSIS

**PHASE 5:**

**PROJECT'S OBJECTIVE:**

To analyze and monitor water quality data using IBM Cognos in order to ensure the safety and sustainability of water resources, identify trends, anomalies, and potential issues, and provide actionable insights to improve water quality management.

In this project, you would use IBM Cognos to perform various data analytics tasks, such as data exploration, visualization, and predictive modeling, to achieve the following specific goals:

**1.Data Collection:**
Gather and integrate relevant water quality data from various sources, such as sensors, monitoring stations, and historical records.

**2. Data Exploration and Visualization:**
Use IBM Cognos to explore the water quality data, identify patterns, and visualize key parameters over time. Create informative dashboards and reports for better data understanding.

**3. Anomaly Detection:**
Implement anomaly detection algorithms to identify unusual water quality events or deviations from expected norms, which might indicate contamination or issues with water sources.

**4. Trend Analysis:**
Analyze historical data to identify long-term trends in water quality, such as seasonal variations or gradual deterioration.

**5. Predictive Modeling:**
Build predictive models to forecast future water quality based on historical data and external factors, enabling proactive management and decision-making.

**6. Quality Assessment:**
Use appropriate metrics and quality assessment techniques to evaluate the reliability and accuracy of the water quality data, ensuring data integrity.

**7. Alerting and Reporting:**
Set up automated alerts or notifications in Cognos to notify relevant authorities or stakeholders when water quality parameters fall outside acceptable ranges.

**8. Recommendations:**
Provide actionable insights and recommendations based on data analysis to improve water quality management practices and decision-making.

**9. Documentation and Reporting:**
Create comprehensive reports and documents that summarize the analysis, findings, and recommendations for water quality management.

The objective of this project is to use IBM Cognos as a powerful tool for water quality data analytics, helping ensure the safety and sustainability of water resources, and enabling timely responses to potential water quality issues. It contributes to better resource management and the protection of public health and the environment.

**DESIGN THINKING PROCESS:**

Here's a simplified design thinking process for water quality analysis in data analytics with Cognos:

**Empathize:**

- Begin by understanding the needs and challenges of stakeholders involved in water quality analysis. This includes scientists, environmentalists, regulators, and the public.
- Conduct interviews, surveys, and observations to gather insights into their goals, pain points, and data analysis requirements.

**Define:**
Define the specific problem or goal you want to address with water quality analysis, such as detecting contamination or predicting water quality trends.Create a clear problem statement and success criteria for the project.

**Ideate:**
Explore different analysis techniques, including statistical methods and machine learning algorithms that can be applied using IBM Cognos.

**Prototype:**
Build prototype data analysis models and dashboards using IBM Cognos. These prototypes should visualize water quality data and provide insights into the defined problem.

**Test:**
Gather feedback from stakeholders and end-users on the prototypes. Ensure that the data analysis and visualizations meet their needs and are user-friendly.

**Iterate:**
Use feedback to refine and improve the data analysis models and visualizations.
Continue to iterate and test until the solution effectively addresses the water quality analysis problem.

**Implement:**
Deploy the finalized data analytics solution within your organization, integrating it with IBM Cognos for regular data collection, analysis, and reporting.

**Evaluate:**
Continuously monitor the performance and impact of the solution on water quality analysis.
Make adjustments as needed based on ongoing evaluations.

**Communicate:**
Share the findings and insights from water quality analysis with relevant stakeholders, including decision-makers and the public, using clear and informative reports and dashboards created in IBM Cognos.

**Scale:**
If successful, consider scaling up the solution to address broader water quality issues or expand its application to different regions or water sources.

**DEVELOPMENT PHASE 1:**

**Data Preprocessing and Cleaning:**

Clean the collected data to ensure its quality and accuracy.

```
#importing data set
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
main_dat = pd.read_csv("water_potability.csv")
ks = main_dat.copy()

#copy of original data set
ks.head()
```

**OUTPUT:**

SECTION-1 (Data preprocessing)

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

```
[2]: main_dat = pd.read_csv("water_potability.csv")
     ks = main_dat.copy() #copy of original data set
```

```
[3]: ks.head()
```

[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 |

ks.sample(5)
ks.shape
ks.columns

**OUTPUT:**

```
[4]: ks.sample(5)
```

[4]:

| | ph | Hardness | Solids | Chloramines | Sulfate | Conductivity | Organic_carbon | Trihalomethanes | Turbidity | Potability |
|---|---|---|---|---|---|---|---|---|---|---|
| **1018** | 6.013161 | 218.843256 | 21573.747571 | 9.295852 | 321.168313 | 444.276635 | 14.744347 | 62.443239 | 3.455623 | 0 |
| **248** | 6.581878 | 272.982745 | 37169.444404 | 8.114731 | 416.083481 | 351.476839 | 15.129334 | 79.261026 | 4.201663 | 0 |
| **1998** | 7.544306 | 211.051146 | 34359.400797 | 8.166793 | 365.812313 | 447.520655 | 18.553478 | 60.162746 | 3.714096 | 1 |
| **2227** | NaN | 159.832881 | 23917.190146 | 6.781576 | 369.223852 | 472.927194 | 13.891834 | 85.758645 | 2.857687 | 0 |
| **2484** | 6.653650 | 172.584512 | 34816.444538 | 8.289307 | 293.611048 | 389.471149 | 15.872474 | 67.976869 | 4.871406 | 0 |

```
[5]: ks.shape
```

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

```
[6]: ks.columns
```

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

pd.isnull(ks).sum()
ks.dropna(inplace=True)
pd.isnull(ks).sum()

**OUTPUT:**

```
[7]: pd.isnull(ks).sum()
```

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

```
[8]: ks.dropna(inplace=True)
     pd.isnull(ks).sum()
```

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

ks.describe()
ks.nunique()

**OUTPUT:**

```
[9]: ks.describe()
```

[9]:

| | ph | Hardness | Solids | Chloramines | Sulfate | Conductivity | Organic_carbon | Trihalomethanes | Turbidity | Potability |
|---|---|---|---|---|---|---|---|---|---|---|
| count | 2011.000000 | 2011.000000 | 2011.000000 | 2011.000000 | 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 | 14.357709 | 66.400859 | 3.969729 | 0.403282 |
| std | 1.573337 | 32.635085 | 8642.239815 | 1.584820 | 41.205172 | 80.712572 | 3.324959 | 16.077109 | 0.780346 | 0.490678 |
| min | 0.227499 | 73.492234 | 320.942611 | 1.390871 | 129.000000 | 201.619737 | 2.200000 | 8.577013 | 1.450000 | 0.000000 |
| 25% | 6.089723 | 176.744938 | 15615.665390 | 6.138895 | 307.632511 | 366.680307 | 12.124105 | 55.952664 | 3.442915 | 0.000000 |
| 50% | 7.027297 | 197.191839 | 20933.512750 | 7.143907 | 332.232177 | 423.455906 | 14.322019 | 66.542198 | 3.968177 | 0.000000 |
| 75% | 8.052969 | 216.441070 | 27182.587067 | 8.109726 | 359.330555 | 482.373169 | 16.683049 | 77.291925 | 4.514175 | 1.000000 |
| max | 14.000000 | 317.338124 | 56488.672413 | 13.127000 | 481.030642 | 753.342620 | 27.006707 | 124.000000 | 6.494749 | 1.000000 |

```
[10]: ks.nunique()
```

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

ks.info()
ks.dtypes

**OUTPUT:**

```
[11]: ks.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
```

```
[12]: ks.dtypes
```

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

#finding the correlation
ks.corr()

**OUTPUT:**

```
[13]: #finding the correlation
      ks.corr()
```

[13]:

| | ph | Hardness | Solids | Chloramines | Sulfate | Conductivity | Organic_carbon | Trihalomethanes | Turbidity | Potability |
|---|---|---|---|---|---|---|---|---|---|---|
| **ph** | 1.000000 | 0.108948 | -0.087615 | -0.024768 | 0.010524 | 0.014128 | 0.028375 | 0.018278 | -0.035849 | 0.014530 |
| **Hardness** | 0.108948 | 1.000000 | -0.053269 | -0.022685 | -0.108521 | 0.011731 | 0.013224 | -0.015400 | -0.034831 | -0.001505 |
| **Solids** | -0.087615 | -0.053269 | 1.000000 | -0.051789 | -0.162769 | -0.005198 | -0.005484 | -0.015668 | 0.019409 | 0.040674 |
| **Chloramines** | -0.024768 | -0.022685 | -0.051789 | 1.000000 | 0.006254 | -0.028277 | -0.023808 | 0.014990 | 0.013137 | 0.020784 |
| **Sulfate** | 0.010524 | -0.108521 | -0.162769 | 0.006254 | 1.000000 | -0.016192 | 0.026776 | -0.023347 | -0.009934 | -0.015303 |
| **Conductivity** | 0.014128 | 0.011731 | -0.005198 | -0.028277 | -0.016192 | 1.000000 | 0.015647 | 0.004888 | 0.012495 | -0.015496 |
| **Organic_carbon** | 0.028375 | 0.013224 | -0.005484 | -0.023808 | 0.026776 | 0.015647 | 1.000000 | -0.005667 | -0.015428 | -0.015567 |
| **Trihalomethanes** | 0.018278 | -0.015400 | -0.015668 | 0.014990 | -0.023347 | 0.004888 | -0.005667 | 1.000000 | -0.020497 | 0.009244 |
| **Turbidity** | -0.035849 | -0.034831 | 0.019409 | 0.013137 | -0.009934 | 0.012495 | -0.015428 | -0.020497 | 1.000000 | 0.022682 |
| **Potability** | 0.014530 | -0.001505 | 0.040674 | 0.020784 | -0.015303 | -0.015496 | -0.015567 | 0.009244 | 0.022682 | 1.000000 |

```
[ ]:
```

**DEVELOPMENT PHASE 2:**

**DATA ANALYSIS**:

- Import and prepare your water quality data in IBM Cognos for analysis.
- Explore and clean the dataset to handle missing values and outliers.
- Calculate relevant statistical measures to gain insights into the data.

**MODEL BUILDING**:

- Depending on your goals, you can build various models, such as regression models to predict water quality parameters or classification models to detect anomalies.
- Use Cognos for data modeling and to create and train machine learning models.

**EVALUATION:**

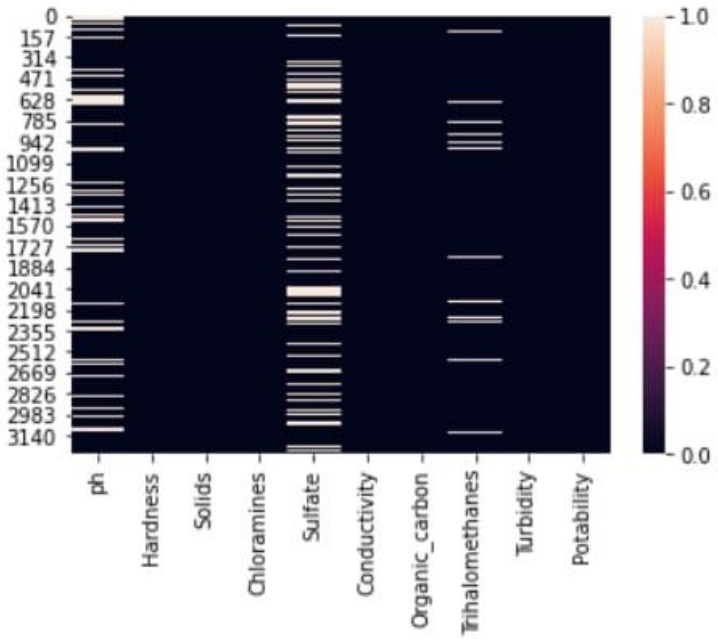
- Assess the performance of your models by using metrics like RMSE (Root Mean Square Error), R-squared, or classification accuracy, depending on the model type.
- Evaluate the models' robustness and generalization to ensure their reliability.

**VISUALIZATION:**

- Create visualizations in IBM Cognos to represent your data and model results effectively.
- Use charts, graphs, and dashboards to present water quality trends, correlations, and model predictions.

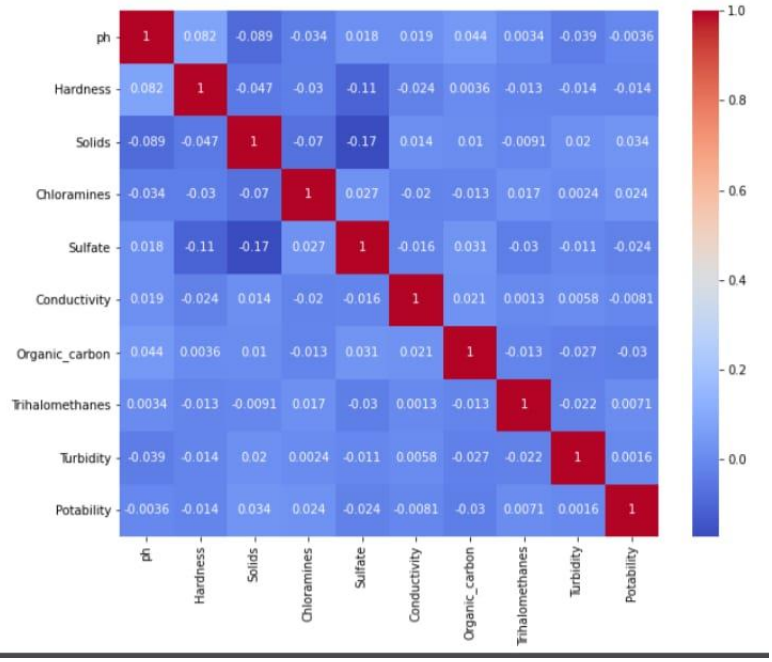
sns.heatmap(df.isnull())

**OUTPUT:**



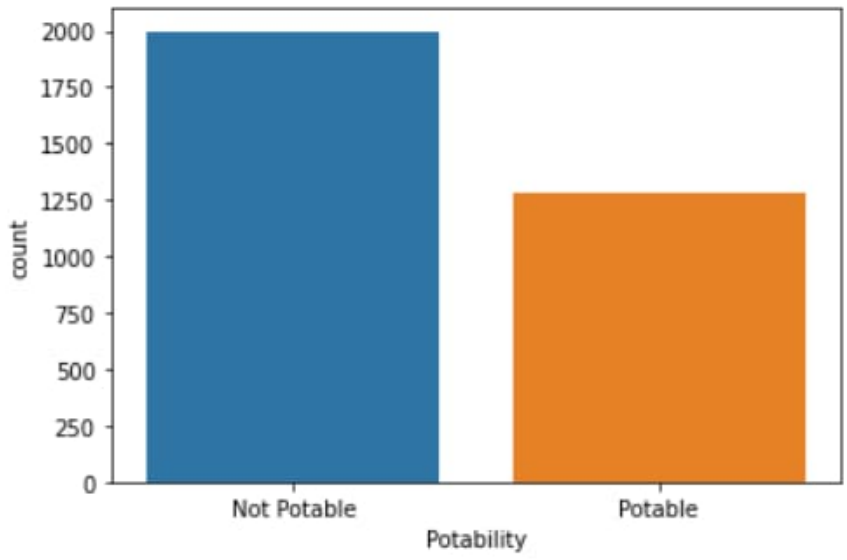
```
plt.figure(figsize=(10, 8))
sns.heatmap(df.corr(), annot= True, cmap='coolwarm')
```

**OUTPUT:**



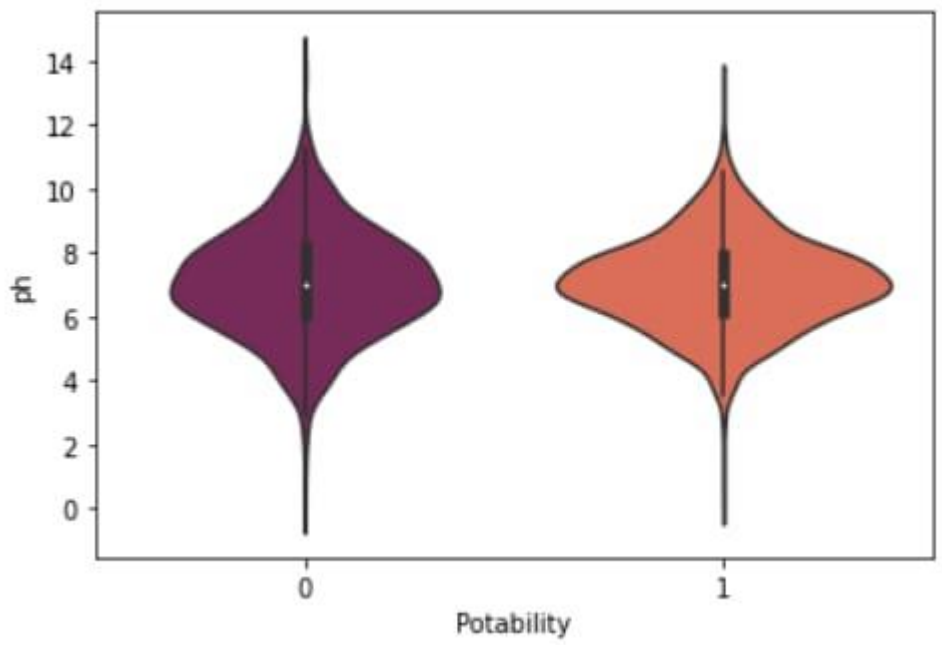

```
ax = sns.countplot(x = "Potability",data= df, saturation=0.8)
plt.xticks(ticks=[0, 1], labels = ["Not Potable", "Potable"])
plt.show()
```

**OUTPUT:**



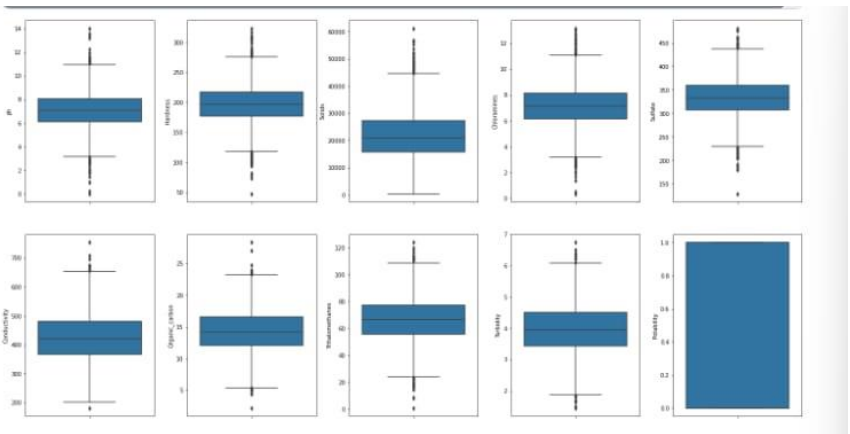
```
sns.violinplot(x='Potability', y='ph', data=df, palette='rocket')
```

**OUTPUT:**



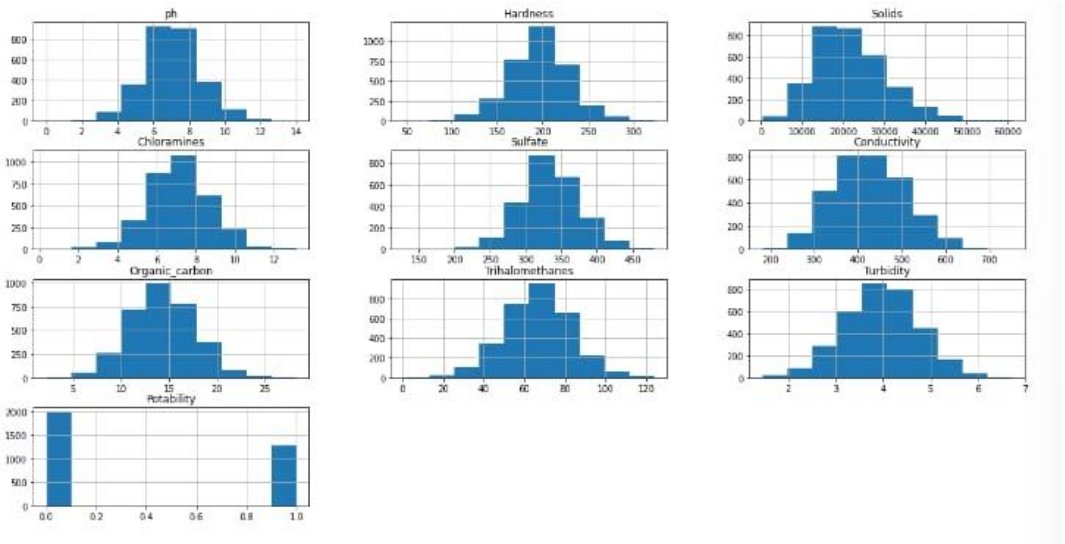

```
fig, ax = plt.subplots(ncols = 5, nrows = 2, figsize = (20, 10))
index = 0
ax = ax.flatten()
for col, value in df.items():
    sns.boxplot(y=col, data=df, ax=ax[index])
    index += 1
plt.tight_layout(pad = 0.5, w_pad=0.7, h_pad=5.0)
```

**OUTPUT:**



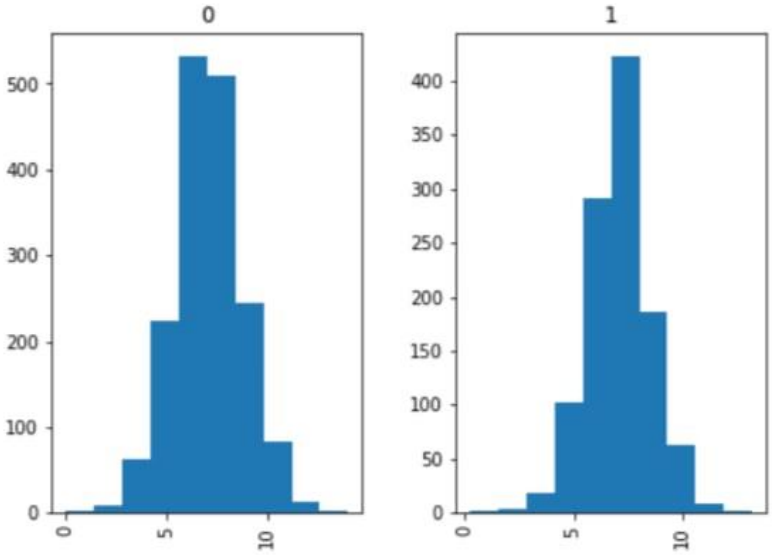
```
plt.rcParams['figure.figsize'] = [20,10]

df.hist()

plt.show()
```

**OUTPUT:**



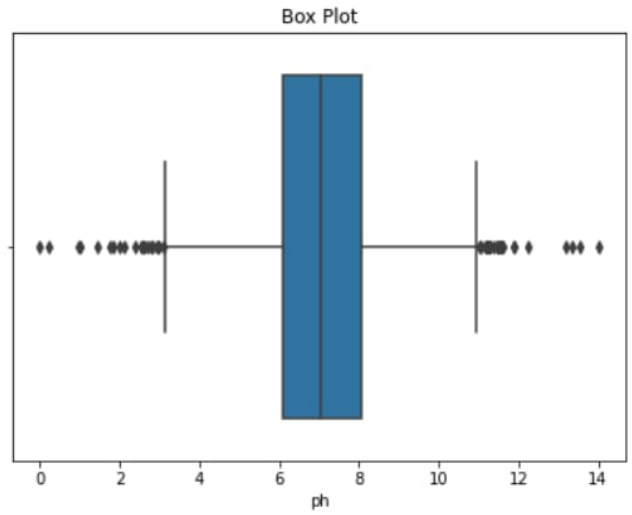

```
plt.rcParams['figure.figsize'] = [7,5]

sns.distplot(df['Potability'])
```

**OUTPUT:**



```
# Individual box plot for each feature
def Box(df):
    plt.title("Box Plot")
    sns.boxplot(df)
    plt.show()
Box(df['ph'])
```

**OUTPUT:**




```
fig = px.box(df, x="Potability", y="ph", color="Potability", width=800, height=400)
fig.show()
```

**OUTPUT:**

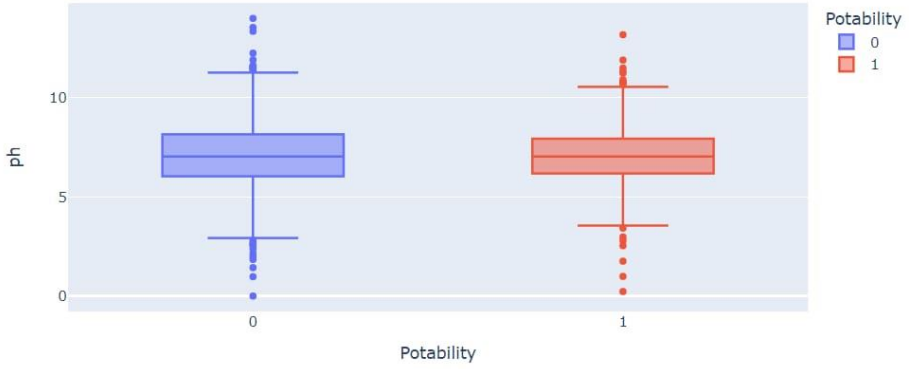


fig = px.histogram (df, x = "Sulfate",  facet_row = "Potability",  template = 'plotly_dark')

fig.show ()

**OUTPUT:**

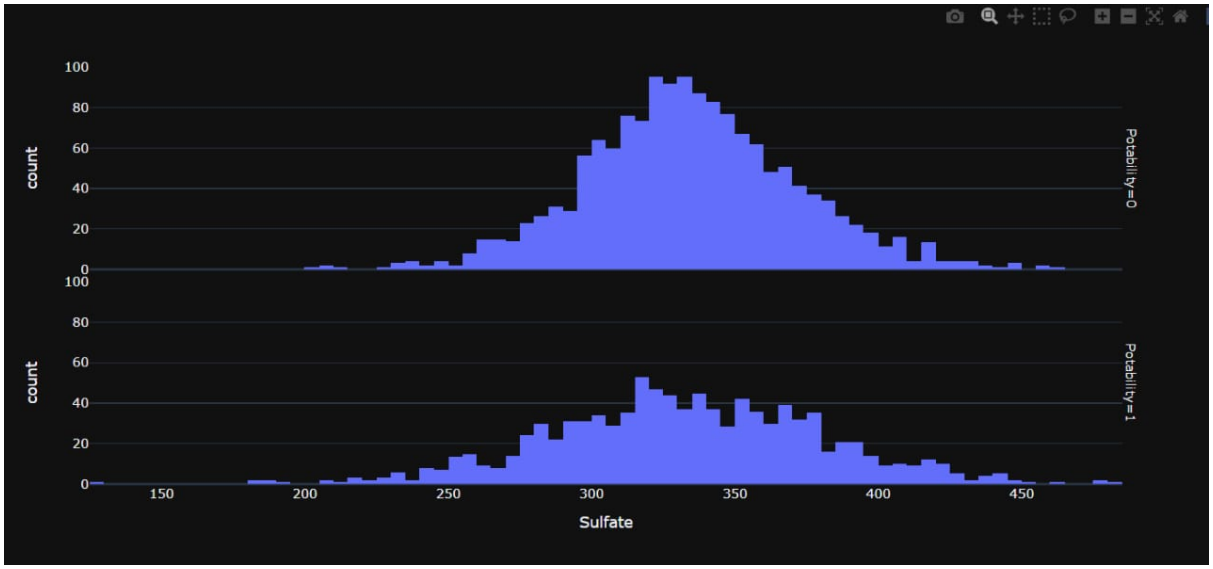
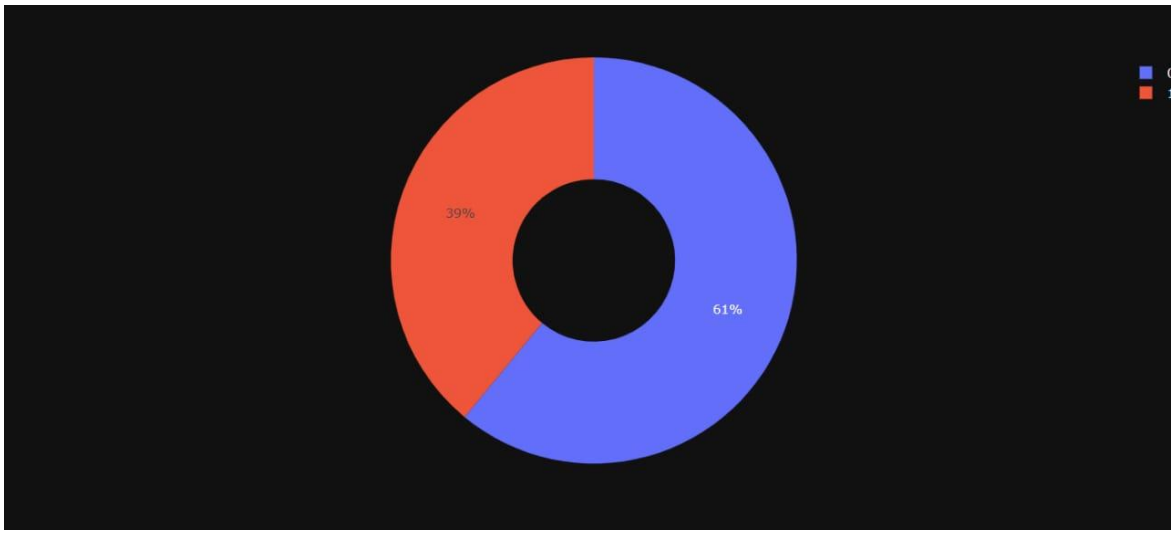



fig =  px.pie (df, names = "Potability", hole = 0.4, template = "plotly_dark")

fig.show ()

**OUTPUT:**



## ANALYSIS OBJECTIVES:

**Objective 1:** Assess Potability.
Determine if the water meets regulatory standards for drinking, ensuring it is safe for consumption.

**Objective 2:** Identify Deviations from Standards.
Detect and flag any instances where water quality parameters deviate from established standards or guidelines.

**Objective 3:** Understand Parameter Relationships.
Investigate the interdependencies among water quality parameters (e.g., pH, Hardness, Solids) to gain insights into their impact on water potability and each other.

## DATA COLLECTION PROCESS:

The first step is to collect water quality data that includes relevant parameters such as pH, Hardness, Solids, etc. The data must be sourced from reliable and representative samples of water sources to support accurate analysis.

- Water quality data is collected from various sources, including rivers, lakes, reservoirs,groundwater,and treatment facilities.

- Data may include measurements of physical, chemical, and biological parameters,such as suchtemperature,PH,turbidity, dissolved oxygen, nutrients, heavy metals, and microbial contaminants.

- Sensors, monitoring stations, and sampling methods are used to collect data over time, providing adetailed picture of water quality dynamics.

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

## DATA VISUALIZATION STRATEGY USING IBM COGNOS:

Visualization is a crucial aspect of conveying insights effectively to various stakeholders. The strategy involves:

**Parameter Distributions Visualization:**
Employ histograms, box plots, or density plots to visualize parameter distributions, providing an understanding of typical values and variability.

**Correlations Visualization:**
Create correlation matrices and heatmaps to visually represent the relationships between different parameters, aiding in the identification of strong correlations and their influence on water quality.

**Potability Assessment Visualization:**
Utilize bar charts or pie charts to depict the proportion of water samples meeting potability standards versus those that do not, offering a clear overview of water quality in the dataset.

**Geospatial Visualization:**
If geographic data is available, create maps that display water source locations and color-code them based on their potability status, helping identify geographical patterns.

**Time Series Analysis:**
If the data includes timestamps, generate time series plots to visualize temporal trends and changes in water quality.

**PYTHON CODE INTEGERATION:**

**Data Preprocessing and Cleaning:**

Clean the collected data to ensure its quality and accuracy.

```
#importing data set
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
main_dat = pd.read_csv("water_potability.csv")
ks = main_dat.copy()

#copy of original data set
ks.head()
ks.sample(5)
ks.shape
ks.columns
pd.isnull(ks).sum()
ks.dropna(inplace=True)
pd.isnull(ks).sum()
ks.describe()
ks.nunique()
ks.info()
ks.dtypes

#finding the correlation
ks.corr()
```

**Visualization:**

```
ax = sns.countplot(x = "Potability",data= df, saturation=0.8)
plt.xticks(ticks=[0, 1], labels = ["Not Potable", "Potable"])
plt.show()

sns.violinplot(x='Potability', y='ph', data=df, palette='rocket')

fig, ax = plt.subplots(ncols = 5, nrows = 2, figsize = (20, 10))
index = 0
ax = ax.flatten()
```

```
for col, value in df.items():
    sns.boxplot(y=col, data=df, ax=ax[index])
    index += 1
plt.tight_layout(pad = 0.5, w_pad=0.7, h_pad=5.0)

plt.rcParams['figure.figsize'] = [20,10]
df.hist()
plt.show()

plt.rcParams['figure.figsize'] = [7,5]
sns.distplot(df['Potability'])

# Individual box plot for each feature
def Box(df):
    plt.title("Box Plot")
    sns.boxplot(df)
    plt.show()
Box(df['ph'])

fig = px.box(df, x="Potability", y="ph", color="Potability", width=800, height=400)
fig.show()

fig = px.histogram (df, x = "Sulfate",  facet_row = "Potability",  template = 'plotly_dark')
fig.show ()

fig =  px.pie (df, names = "Potability", hole = 0.4, template = "plotly_dark")
fig.show ()
```

**Explain how the insights from the analysis can help website owners improve user experience.**

Improving the user experience on a website related to water quality analysis using data analytics with Cognos involves leveraging the insights gained from the analysis to make data-driven decisions and enhancements. Here's how insights can help website owners enhance the user experience:

**1. Tailored Information:**

- Analyze the data to understand the specific interests and needs of different user segments (e.g., researchers, policymakers, citizens).
- Tailor the website's content and resources to provide relevant information to each user group. For example, provide detailed technical data for researchers while offering simplified water quality reports for the general public.

**2. Interactive Data Visualization:**
- Use insights to identify key water quality indicators and trends that are most relevant to users.
- Develop interactive data visualization dashboards in Cognos that allow users to explore and understand water quality data easily.
- Implement user-friendly charts, maps, and graphs that make complex data more accessible and engaging.

**3. Real-Time Updates:**
- Leverage insights about the frequency and patterns of user visits to provide real-time or periodic updates on water quality conditions.
- Implement automated alerts and notifications based on specific triggers, such as water quality thresholds or events.

**4. Data Transparency:**
- Use the analysis to determine which water quality parameters are of the greatest concern to users and regulators.
- Ensure transparency by providing detailed information about data sources, collection methods, and data quality.
- Allow users to access and download raw or processed data for their analysis, promoting trust and credibility.

**5. Educational Resources:**
- Identify areas where users may lack knowledge or have misconceptions about water quality issues.
- Develop educational resources, such as articles, videos, or infographics, to help users understand the data and its implications.
- Include explanations of technical terms and jargon to make the content more accessible.

**6. User Feedback:**
- Utilize feedback mechanisms on the website to gather input from users about their experience and needs.
- Act on user feedback to make continuous improvements, such as adjusting website navigation, content organization, and the usability of data tools.

**7. Mobile Optimization:**
- Consider insights regarding user device preferences and access patterns.
- Optimize the website and data visualizations for mobile devices to ensure a seamless experience for users on smartphones and tablets.

**8. User-Centric Design:**
- Apply insights to inform website design decisions, including layout, color schemes, and fonts.
- Prioritize a clean and user-centric design that facilitates easy navigation and information retrieval.

**9. Community Engagement:**
- Encourage user engagement through community forums, discussions, or citizen science projects based on insights into user interests.
- Foster a sense of community and involvement in water quality monitoring efforts.

**10. Accessibility:**

- Ensure that the website and data visualizations are accessible to users with disabilities, complying with accessibility standards.
- Make data and insights available to a broad audience, including those with different needs.

By continuously analyzing data and seeking insights, website owners can tailor their water quality analysis website to meet the diverse needs of users, ultimately improving user experience, promoting data literacy, and fostering greater engagement with water quality information and issues.

**Project Conclusion:**

In conclusion, data analytics is an indispensable tool in the field of water quality analysis, enabling us to gain valuable insights into the health and safety of water sources. By harnessing the power of data analytics, we can make informed decisions that impact public health, environmental sustainability, and resource management.