Name of the Project : Water quality parameters

Phase – 4

Module 7: Data Exploration & Visualization

When exploring and visualizing water quality data, various parameters are crucial for understanding and conveying the quality of water sources. Visualization techniques help in identifying trends, anomalies, and patterns in the data. Here are some key water quality parameters in the context of data exploration and visualization:

Time Series Data: Water quality data often varies over time. Visualizing parameters like temperature, pH, dissolved oxygen, and pollutant concentrations over time helps in understanding seasonal variations and long-term trends.

Spatial Data: Consider using maps and geographic information system (GIS) tools to visualize water quality parameters across different locations. This can reveal spatial patterns and the impact of geographic features on water quality.

Box Plots and Histograms: These graphical representations can help show the distribution of water quality parameters. For example, a box plot can reveal the median, quartiles, and potential outliers in a dataset.

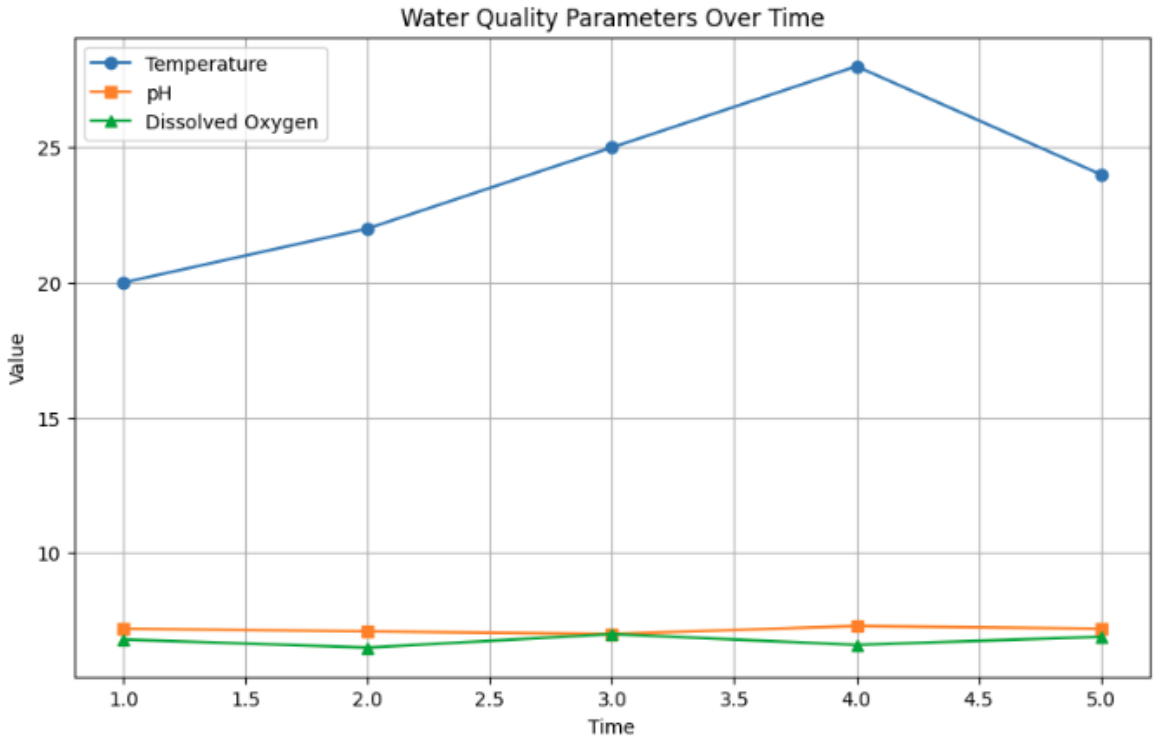
Scatter Plots: Scatter plots can be used to explore relationships between different water quality parameters. For instance, you can examine the correlation between temperature and dissolved oxygen or nutrient levels and algal blooms.

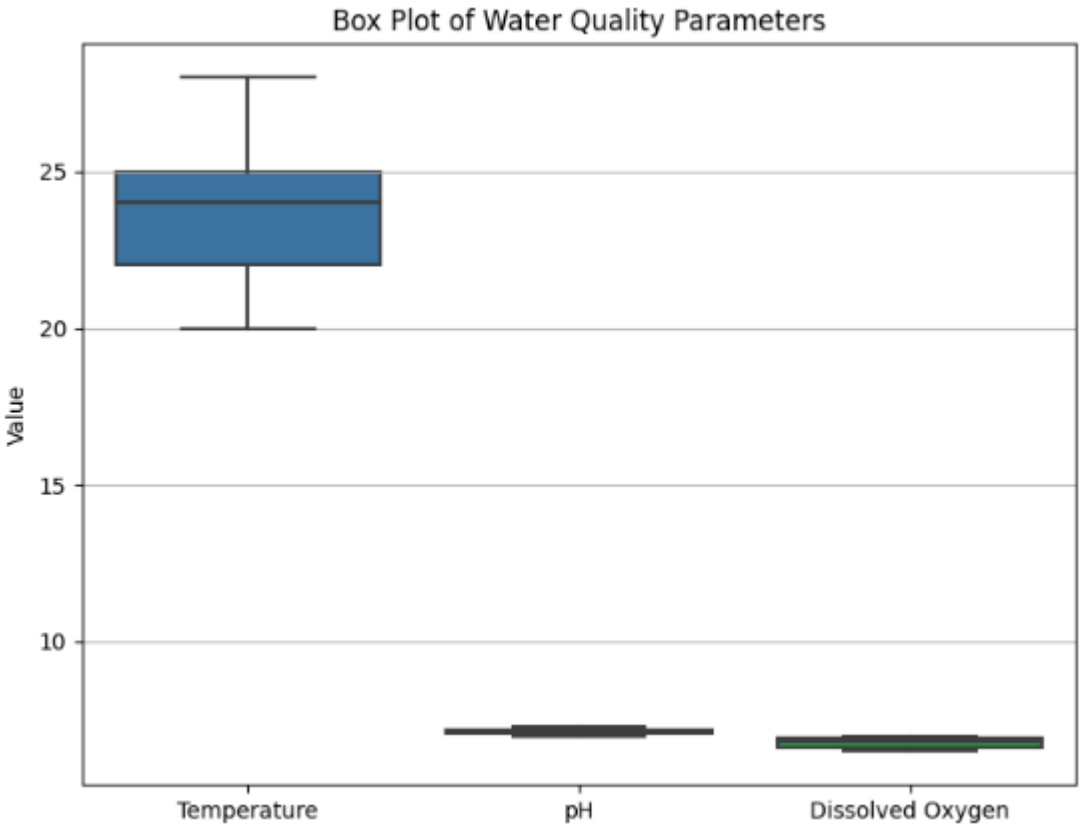
3D Plots: If you have data on water quality parameters in three dimensions (e.g., time, location, and a parameter's value), 3D plots or scatterplots with three axes can be used to visualize the data.

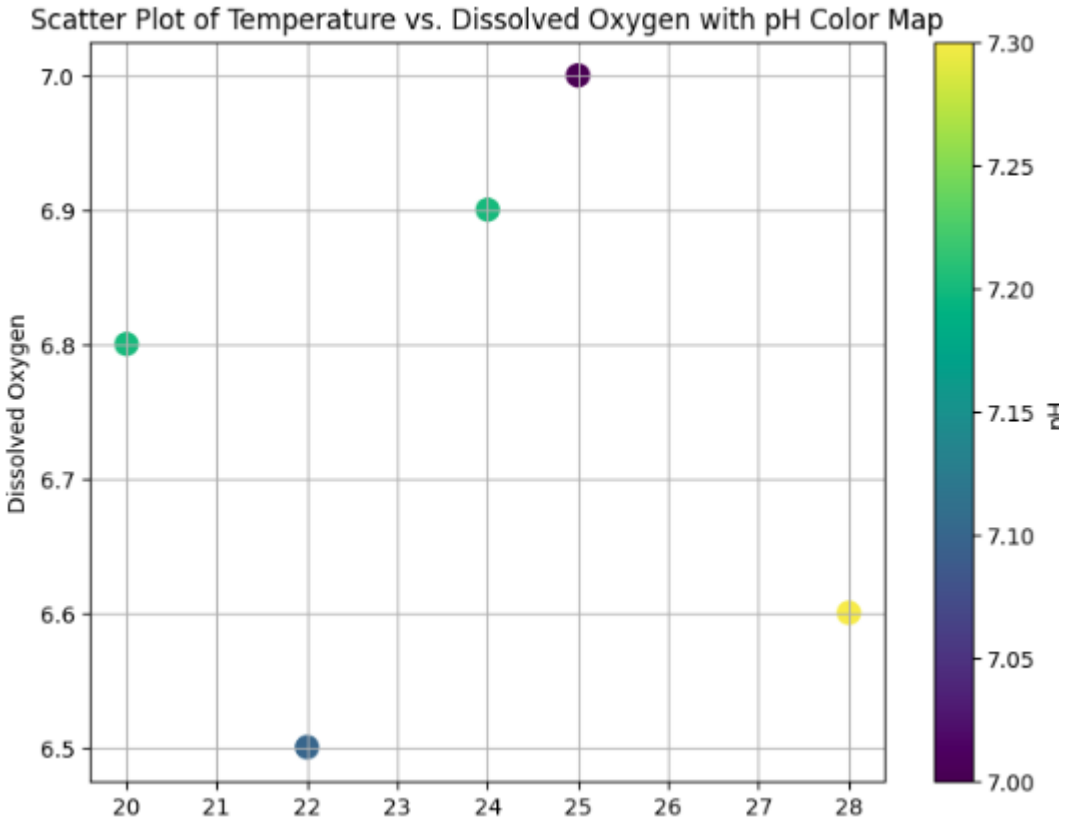
```
pip install matplotlib seaborn
```

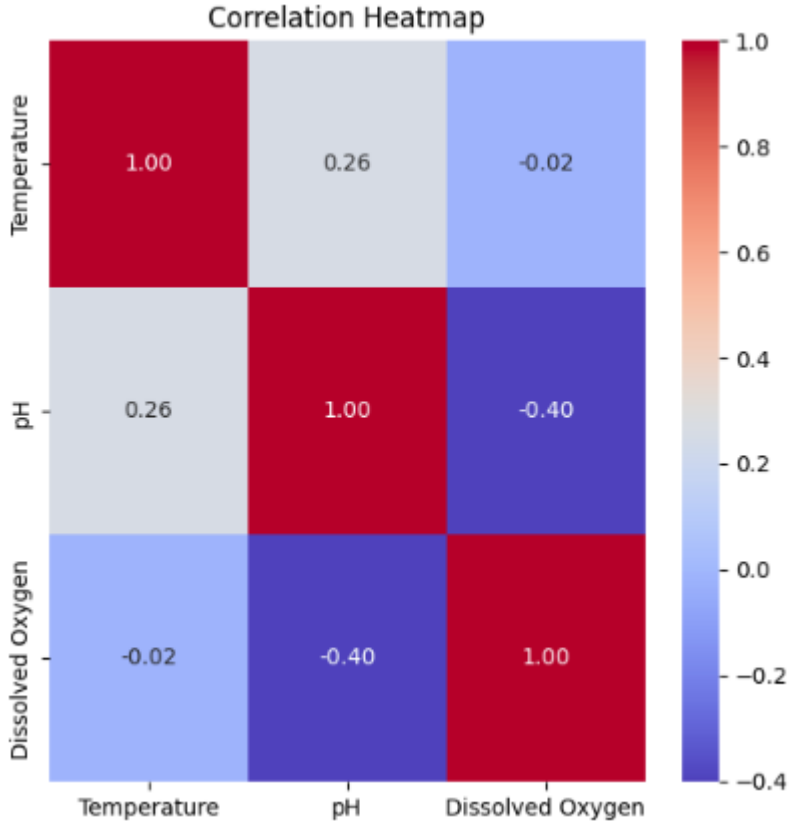
```
import matplotlib.pyplot as plt
import seaborn as sns

# Sample data
time = [1, 2, 3, 4, 5]
temperature = [20, 22, 25, 28, 24]
pH = [7.2, 7.1, 7.0, 7.3, 7.2]
```

```
dissolved_oxygen = [6.8, 6.5, 7.0, 6.6, 6.9]

# Time Series Plot
plt.figure(figsize=(10, 6))
plt.plot(time, temperature, label='Temperature', marker='o')
plt.plot(time, pH, label='pH', marker='s')
plt.plot(time, dissolved_oxygen, label='Dissolved Oxygen', marker='^')
plt.xlabel('Time')
plt.ylabel('Value')
plt.title('Water Quality Parameters Over Time')
plt.legend()
plt.grid(True)
plt.show()

# Box Plot
plt.figure(figsize=(8, 6))
sns.boxplot(data=[temperature, pH, dissolved_oxygen], orient='v')
plt.xticks([0, 1, 2], ['Temperature', 'pH', 'Dissolved Oxygen'])
plt.ylabel('Value')
plt.title('Box Plot of Water Quality Parameters')
plt.grid(axis='y')
plt.show()

# Scatter Plot
plt.figure(figsize=(8, 6))
plt.scatter(temperature, dissolved_oxygen, c=pH, cmap='viridis', s=100)
plt.colorbar(label='pH')
plt.xlabel('Temperature')
plt.ylabel('Dissolved Oxygen')
plt.title('Scatter Plot of Temperature vs. Dissolved Oxygen with pH
Color Map')
plt.grid(True)
plt.show()

# Heatmap
import numpy as np
data = np.array([temperature, pH, dissolved_oxygen])
correlation_matrix = np.corrcoef(data)
plt.figure(figsize=(6, 6))
sns.heatmap(correlation_matrix, annot=True, fmt=".2f", cmap='coolwarm',
xticklabels=['Temperature', 'pH', 'Dissolved Oxygen'],
yticklabels=['Temperature', 'pH', 'Dissolved Oxygen'])
plt.title('Correlation Heatmap')
plt.show()
```


Water Quality Parameters Over Time
Temperature
pH
Dissolved Oxygen
Value
25
20
15
10
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Time


Box Plot of Water Quality Parameters
Value
25
20
15
10
Temperature
pH
Dissolved Oxygen





Module 8 : Supervised Learning - Regression

In the context of supervised learning regression for water quality analysis, you

can use water quality parameters as input features to predict a specific water quality metric. Here's how you can use these parameters in a regression analysis.

```
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt

# Sample data
temperature = [20, 22, 25, 28, 24]
pH = [7.2, 7.1, 7.0, 7.3, 7.2]
suspended_solids = [10, 15, 12, 8, 11]
dissolved_oxygen = [6.8, 6.5, 7.0, 6.6, 6.9]

# Prepare data
X = np.array([temperature, pH, suspended_solids]).T  # Water quality
parameters
y = np.array(dissolved_oxygen)  # Target variable

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Create and train a linear regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"Mean Squared Error: {mse:.2f}")
print(f"R-squared: {r2:.2f}")

# Visualize the model's predictions
plt.scatter(y_test, y_pred)
plt.xlabel("Actual Dissolved Oxygen")
plt.ylabel("Predicted Dissolved Oxygen")
plt.title("Actual vs. Predicted Dissolved Oxygen")
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_regression.py:918: UndefinedMetricWarning: R^2 score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
Mean Squared Error: 0.64
R-squared: nan
```