AIR QUALITY ANALYSIS IN TAMILNADU

AIRQUALITY ANALYSIS

Air quality analysis is the process of evaluating and accessing the quality of the air in a specific location or region. It involves measuring and analyzing various components and pollutants in the atmosphere to determine the level of air pollution and its potential impact on human health the environment and overall well -being.

DATA SOURCE

https://tn.data.gov.in/resource/location-wise-daily-ambient-air-quality-tamil-nadu-year-2014

Data Collection: Gather air quality data for Tamil Nadu. This data can often be obtained from government agencies or environmental organizations. Ensure that the data includes relevant parameters like PM2.5, PM10, CO, SO2, and NO2 levels over a specific period.

Data Preprocessing: Clean the data, handle missing values, and format it for analysis. You might need to convert date and time fields into a usable format.

Data Analysis: Perform various analyses to understand the air quality. Some common analyses include:

Descriptive statistics to get an overview of the data.

Time-series analysis to identify trends and seasonal patterns.

Correlation analysis to understand relationships between different pollutants.

Spatial analysis to visualize air quality across different regions of Tamil Nadu.

Data Visualization: Visualize the data using IBM Cognos or other data visualization tools like Python's Matplotlib, Seaborn, or Plotly. Create various types of charts and graphs to communicate your findings effectively

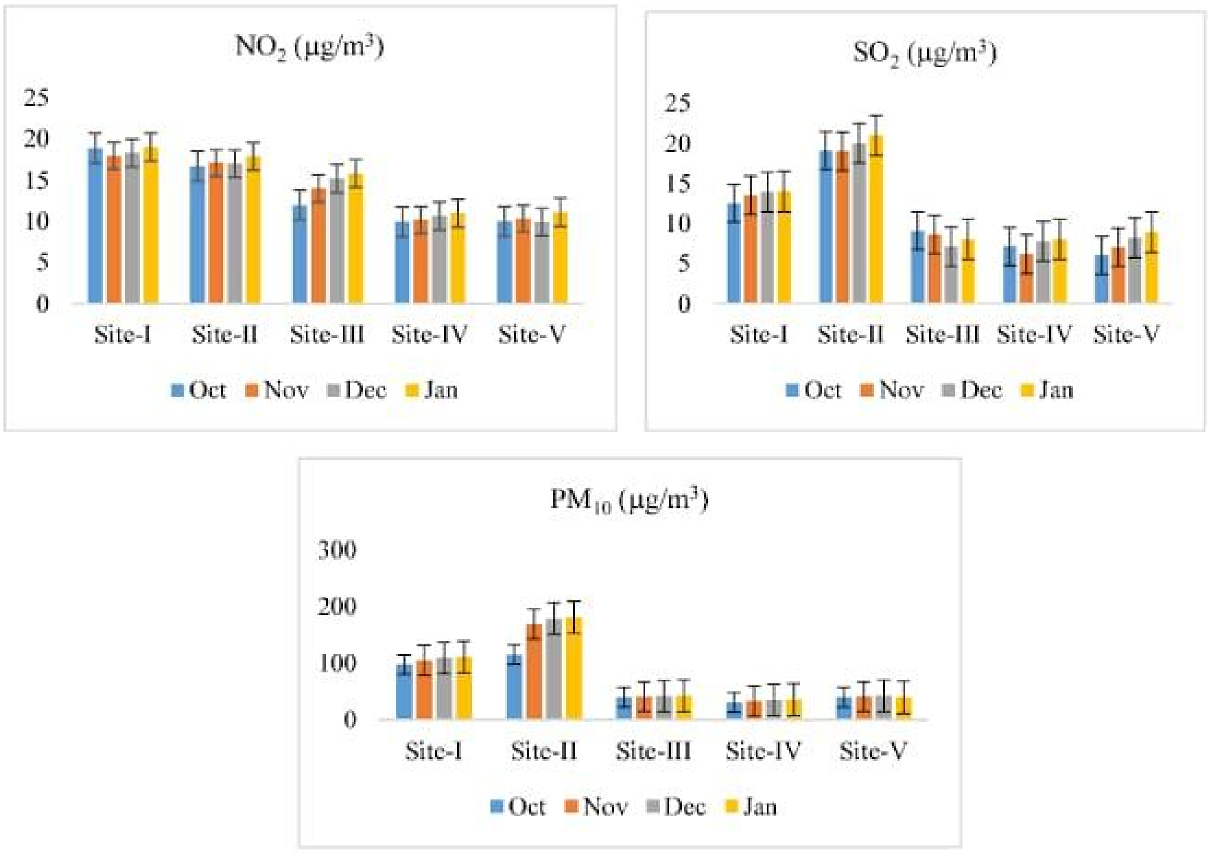


Figure 1. Monthly variation in NO, SO and PM at different sites

DESCRIPTION:

It involves the systematic process of creating and refining models, collecting data, and conducting research to assess and monitor the quality of the air in a specific area. This phase is critical for understanding the sources of air pollution, predicting future air quality, and implementing strategies to improve it.

SUMMARYAND REPORT:

Summarize your findings and conclusions based on analysis and visualisation

Provide recommendations or insights into areas in need of air quality analysis

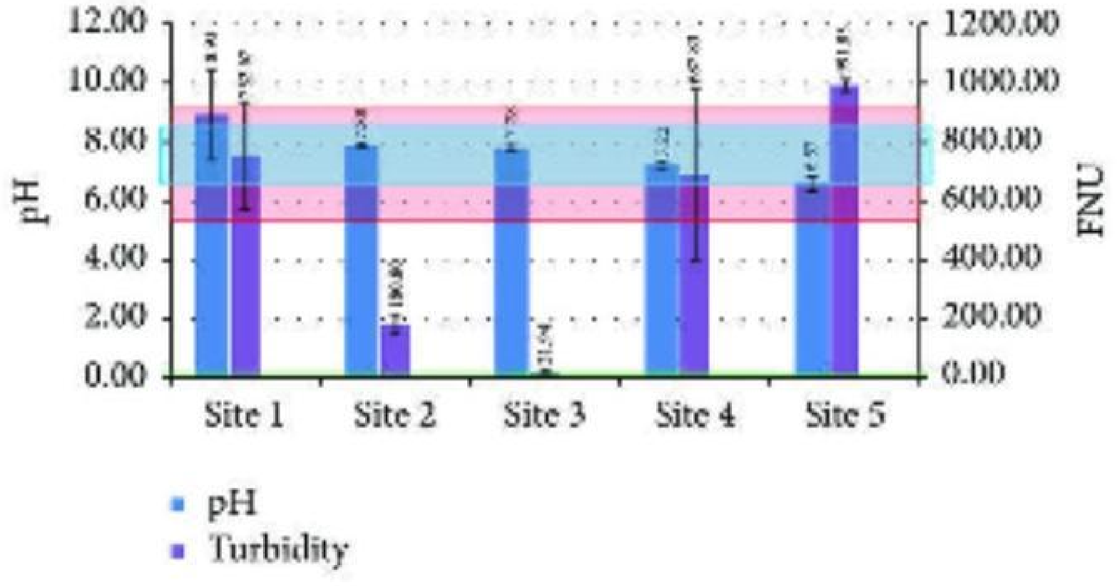
CODING:

```
# Import necessary libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns
```

```
from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean_squared_error, r2_score


# Load your air quality dataset

data = pd.read_csv("cpcb_dly_aq_tamil_nadu-2014.csv")


# Check the column names to ensure they are correct

print(data.columns)


# Data preprocessing

# Remove non-numeric columns and select relevant features

# Assuming your dataset includes features like 'Temperature', 'Humidity', 'Wind Speed'

# and the target variable is 'RSPM/PM10' or another correct target column name


# List of non-numeric columns to be removed

non_numeric_columns = ['Sampling Date', 'State', 'City/Town/Village/Area', 'Location of Monitoring Station', 'Agency', 'Type of Location']


# Remove non-numeric columns

data = data.drop(columns=non_numeric_columns)


# Make sure the column names match those in your dataset

X = data[['Stn Code', 'SO2', 'NO2', 'PM 2.5']]

y = data['RSPM/PM10']  # Replace 'RSPM/PM10' with the correct target column name


# 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)


# Model building (Random Forest Regression as an example)

model = RandomForestRegressor(n_estimators=100, random_state=42)
```

```
model.fit(X_train, y_train)

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

# Model evaluation
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse}")
print(f"R-squared (R2) Score: {r2}")

# Visualize the results
plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_pred, alpha=0.5)
plt.xlabel("Actual RSPM/PM10")
plt.ylabel("Predicted RSPM/PM10")
plt.title("Actual vs. Predicted RSPM/PM10")
plt.grid(True)

# Add a regression line to the scatter plot
sns.regplot(y_test, y_pred, scatter=False, color='red')

plt.show()
```

OUTPUT:



12.00
10.00
8.00
6.00
4.00
2.00
0.00
pH
1200.00
1000.00
800.00
600.00
400.00
200.00
0.00
FNU
Site 1
Site 2
Site 3
Site 4
Site 5
pH
Turbidity