Phase 3: Development Part 1

1. Data Acquisition:

- 1.1. Identify data sources: Look for governmental databases, research institutions,
- Monitoring stations, satellite data, etc.
- 1.2. Data download or retrieval: Use APIs, web scraping tools, or direct downloads to gather the required data.
- 1.3. Store the raw data securely, preferably in a version-controlled environment to track changes.

2. Data Preprocessing:

2.1. Data Cleaning:

- Identify and handle missing values: use imputation, deletion, or other suitable methods.
- Remove or correct any outliers or anomalies.

2.2. Data Transformation:

- Standardize units (e.g., converting all measurements to µg/m^3).
- Normalize or scale data if required.

2.3. Data Integration:

- Merge or join datasets from different sources to create a comprehensive dataset.
- Ensure temporal and spatial alignment of records.

2.4. Feature Engineering:

- Create derived features that might be beneficial for analysis (e.g., rolling averages for air quality metrics).

3. Exploratory Data Analysis (EDA):

3.1. Descriptive statistics:

- Compute means, medians, standard deviations, percentiles, etc., to get an overview of the data distribution.

3.2. Correlation analysis:

- Identify relationships between different air pollutants and other relevant variables.

3.3. Temporal trends:

- Examine how air quality metrics change over time, considering daily, monthly, seasonal, or yearly patterns.

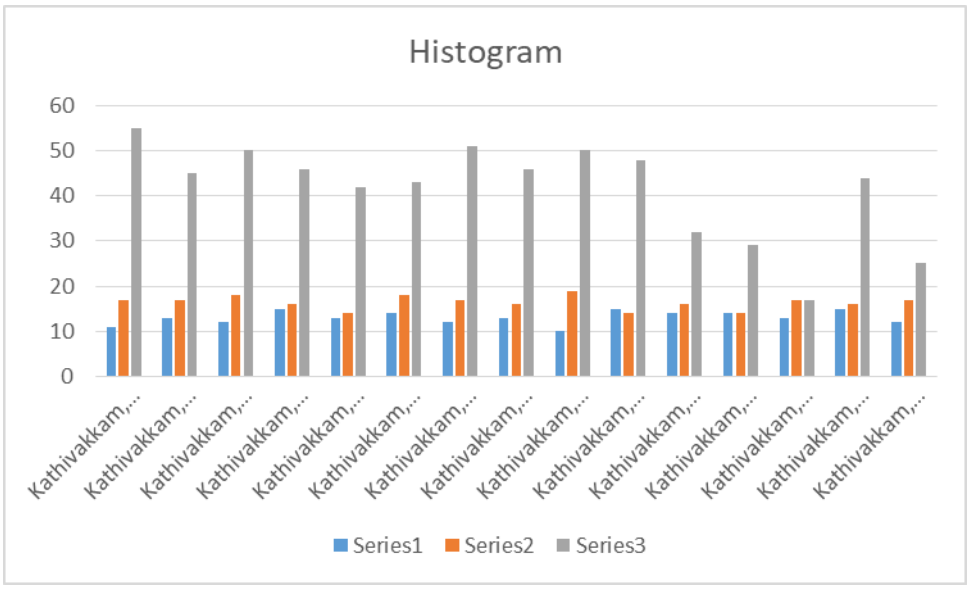
4. Visualization:

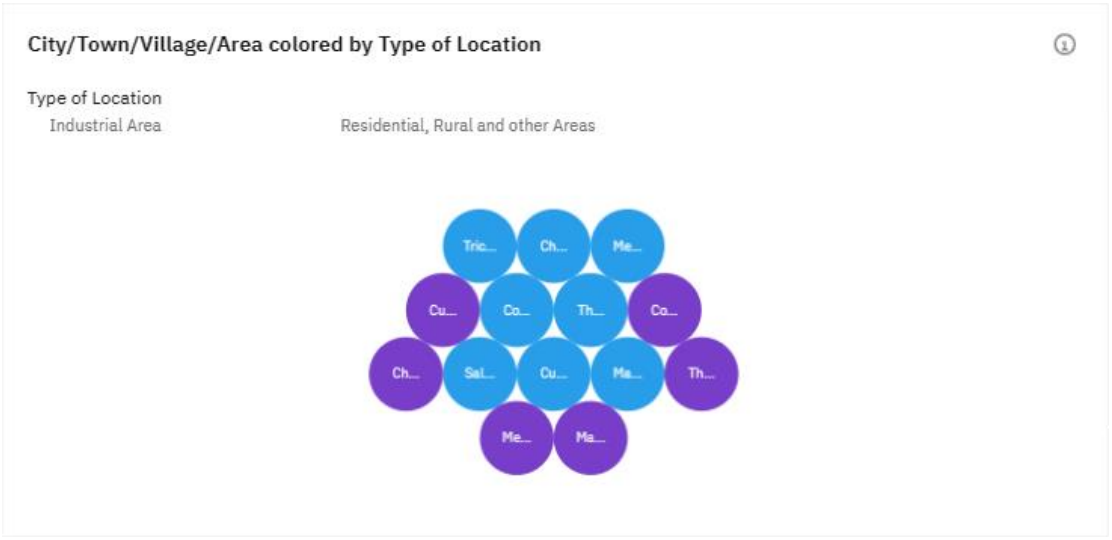
- 4.1. Time series plots: Visualize changes in air quality metrics over time.
- 4.2. Heat maps: Represent correlations or geographical concentration levels of pollutants.
- 4.3. Scatter plots or pair plots: Understand pairwise relationships between variables.
- 4.4. Geographic visualizations: Using GIS tools, represent spatial patterns of air quality on maps to identify pollution hotspots.

Program:

```
import pandas as pd

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

if 'date_column_name' in data.columns:

data['date_column_name'] = pd.to_datetime(data['date_column_name'])

data = pd.get_dummies(data)

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

data[['column_to_scale1', 'column_to_scale2']] =
scaler.fit_transform(data[['column_to_scale1', 'column_to_scale2']])

import matplotlib.pyplot as plt

plt.hist(data['Location of Monitoring Station'], bins=20)

plt.pie(data['SO2'])

plt.show()
```

Output:


Histogram
60
50
40
30
20
10
0
Kathivakkam,...
Series1
Series2
Series3


City/Town/Village/Area colored by Type of Location
Type of Location
Industrial Area
Residential, Rural and other Areas


Agency colored by City/Town/Village/Area
City/Town/Village/Area
Chennai
Coimbatore
Cuddalore
Madurai
Mettur
Salem
Thoothukudi
Trichy

City/Town/Village/Area colored by Stn Code

Stn Code

38 71 72 159 160 161 237 238 239 240 306 307 308 309 366 375
759 760 761 762 763 764 765 766 767 769 770 771 772 773