Exploratory Data Analysis (EDA) Report – Air Quality

Introduction

Exploratory Data Analysis (EDA) plays a vital role in understanding environmental datasets, especially air quality data that directly impacts public health and policy decisions. This project focuses on performing EDA on the **qualityair.csv** dataset to analyze air pollution levels across different locations in India. The analysis helps identify pollution patterns, variations among cities, and relationships between different pollutants.

Dataset Description

Dataset Name: qualityair.csv

The dataset contains air quality information collected from various monitoring stations across India.

Variables in the Dataset

1. **Numerical Variables**
    - Minimum pollutant concentration
    - Maximum pollutant concentration
    - Average pollutant concentration
2. **Categorical Variables**
    - Country
    - State
    - City
    - Station
    - Pollutant type
3. **Geographical Variables**
    - Latitude
    - Longitude

Objectives of the Study

- To analyze air quality across different Indian cities and stations
- To understand the distribution and variability of major air pollutants
- To identify patterns, trends, and anomalies in pollutant concentrations
- To support environmental monitoring and data-driven decision-making

---

Procedure: Exploratory Data Analysis

1: Understanding the Problem and the Data

In this step, the problem statement is clearly defined and the nature of the dataset is understood. The goal is to analyze air quality data to identify pollution levels, trends, and variations across different locations and pollutants. Understanding the dataset structure, variables, and objectives ensures a focused and meaningful analysis.

2: Importing and Inspecting the Data

The dataset is imported into the Python environment using Pandas. Initial inspection is performed to understand the data layout and quality.

- read_csv() is used to load the dataset
- head() displays sample records
- info() provides details about data types and missing values

This step helps identify potential data issues before further processing.

3: Data Cleaning and Preparation

This step focuses on improving data quality by handling missing values, removing duplicates, and preparing the data for analysis.

- Missing numerical values are handled using mean imputation
- Duplicate records are removed to avoid biased results
- Data is checked for consistency and accuracy

A clean and well-prepared dataset leads to reliable analysis and insights.

---

1. Understanding the Problem and the Data

The first step involves clearly defining the problem and understanding the dataset structure. The objective is to evaluate air pollution levels and compare pollutant concentrations across multiple locations.

---

2. Importing and Inspecting the Data

The dataset is loaded into a Pandas DataFrame using the read_csv() function. Initial exploration is done using:

- head() to view sample records
- info() to understand data types, non-null values, and dataset size

This step helps identify inconsistencies, missing values, and data quality issues early in the analysis.

```
(kaviya) PS C:\Users\Kaviya S\OneDrive\Desktop\dscience> & "C:/Users/Kaviya S/OneDrive/Desktop/dscience/kaviya/Scripts/python.exe"
/Users/Kaviya S/OneDrive/Desktop/dscience/main.py"
First 5 rows:
   country  state     city                       station  ... pollutant_id  pollutant_min  pollutant_max pollutant_avg
0   India  Bihar    Buxar     Charitra Van, Buxar - BSPCB  ...          SO2            1.0           12.0           6.0
1   India  Bihar  Chhapra  Darshan Nagar, Chhapra - BSPCB  ...        PM2.5           15.0          305.0         152.0
2   India  Bihar  Chhapra  Darshan Nagar, Chhapra - BSPCB  ...          NH3            5.0           12.0           8.0
3   India  Bihar  Chhapra  Darshan Nagar, Chhapra - BSPCB  ...           CO           45.0           54.0          51.0
4   India  Bihar  Chhapra  Darshan Nagar, Chhapra - BSPCB  ...        OZONE           16.0           18.0          17.0

[5 rows x 11 columns]
Last 5 rows:
     country               state             city  ... pollutant_min pollutant_max  pollutant_avg
234   India               Bihar             Gaya  ...           4.0           7.0            5.0
235   India               Bihar             Gaya  ...           NaN           NaN            NaN
236   India  Andaman and Nicobar  Sri Vijaya Puram  ...          24.0          37.0           28.0
237   India      Andhra_Pradesh         Anantapur  ...          20.0          27.0           22.0
238   India      Andhra_Pradesh     Visakhapatnam  ...         160.0         417.0          251.0

[5 rows x 11 columns]
Data Info:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 239 entries, 0 to 238
Data columns (total 11 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   country        239 non-null    object
 1   state          239 non-null    object
 2   city           239 non-null    object
 3   station        239 non-null    object
 4   last_update    239 non-null    object
 5   latitude       239 non-null    float64
 6   longitude      239 non-null    float64
 7   pollutant_id   239 non-null    object
 8   pollutant_min  223 non-null    float64
 9   pollutant_max  223 non-null    float64
 10  pollutant_avg  223 non-null    float64
dtypes: float64(5), object(6)
memory usage: 20.7+ KB
None
                                                                                                    Ln 20, Col 13    Spaces
```

---

3. Handling Missing Values and Duplicates

Detection of Missing Values

- Missing values are identified using isnull().sum().

Handling Missing Values

- Numerical missing values are treated using **mean imputation** to preserve the overall distribution of the data.

Removing Duplicates

- Duplicate rows are identified and removed using drop_duplicates() to ensure data accuracy and integrity.

```
None
Data Description:
          latitude   longitude  pollutant_min  pollutant_max  pollutant_avg
count  239.000000  239.000000     223.000000     223.000000     223.000000
mean    21.874877   85.557892      31.452915      81.345291      51.762332
std      5.198458    4.950717      39.013468     100.188705      58.200198
min     11.654054   77.593027       1.000000       1.000000       1.000000
25%     16.515083   80.649110       5.500000      12.000000       9.000000
50%     24.828270   84.982348      17.000000      39.000000      28.000000
75%     26.079724   91.439057      42.500000     108.000000      80.000000
max     27.103358   94.636574     272.000000     435.000000     319.000000
Missing values:
 country           0
state              0
city               0
station            0
last_update        0
latitude           0
longitude          0
pollutant_id       0
pollutant_min     16
pollutant_max     16
pollutant_avg     16
dtype: int64

Missing values (isnull) after dropna and duplicates removed:
country          0
state            0
city             0
station          0
last_update      0
latitude         0
longitude        0
pollutant_id     0
pollutant_min    0
pollutant_max    0
pollutant_avg    0
dtype: int64
```

---

4. Exploring Data Characteristics

Descriptive statistics are calculated to understand the behavior of numerical variables. Key statistical measures include:

- **Mean:** Average pollutant concentration
- **Median:** Central tendency of the data
- **Standard Deviation:** Degree of dispersion
- **Minimum and Maximum:** Extreme pollutant values

These statistics help compare pollution intensity across different cities and monitoring stations.

```
DESCRIPTIVE STATISTICS

Column Name: latitude
Mean : 21.91481285654709
Median : 25.251013
Mode : 13.20488
Standard Deviation : 5.258030433450641
Variance : 27.64688403909314
Minimum : 11.654054
Maximum : 27.103358
----------------------------------------
Column Name: longitude
Mean : 85.69720734955156
Median : 84.992416
Mode : 78.824187
Standard Deviation : 5.017545421495591
Variance : 25.17576205677136
Minimum : 77.593027
Maximum : 94.636574
----------------------------------------
Column Name: pollutant_min
Mean : 31.45291479820628
Median : 17.0
Mode : 1.0
Standard Deviation : 39.013468199137854
Variance : 1522.0507009251405
Minimum : 1.0
Maximum : 272.0
----------------------------------------
Column Name: pollutant_max
Mean : 81.34529147982063
Median : 39.0
Mode : 7.0
Standard Deviation : 100.18870511756991
Variance : 10037.77663313538
Minimum : 1.0
Maximum : 435.0
----------------------------------------
```

5. Data Transformation

To make fair comparisons between pollutants measured on different scales, data transformation techniques are applied.

Min–Max Normalization

- Scales numerical features to a range between 0 and 1
- Useful for visualization and comparison

$$x_{\text{norm}} = \frac{x - \min(x)}{\max(x) - \min(x)}$$

```
Transforming Data:

Normalization applied to numeric colums
```

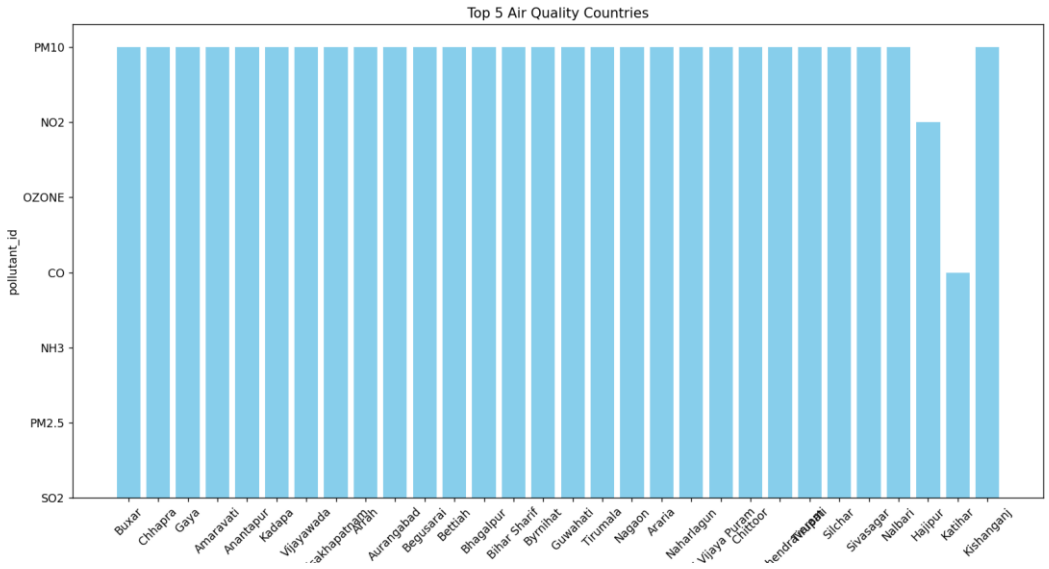
---

6. Data Visualization

Visualization techniques are used to convert numerical data into meaningful graphical representations.

Univariate Analysis

- **Histogram:** Shows the distribution of pollutant concentrations



Multivariate Analysis

- **Bar Chart:** Compares average pollutant levels across cities

---

7. Advanced Data Visualization Techniques

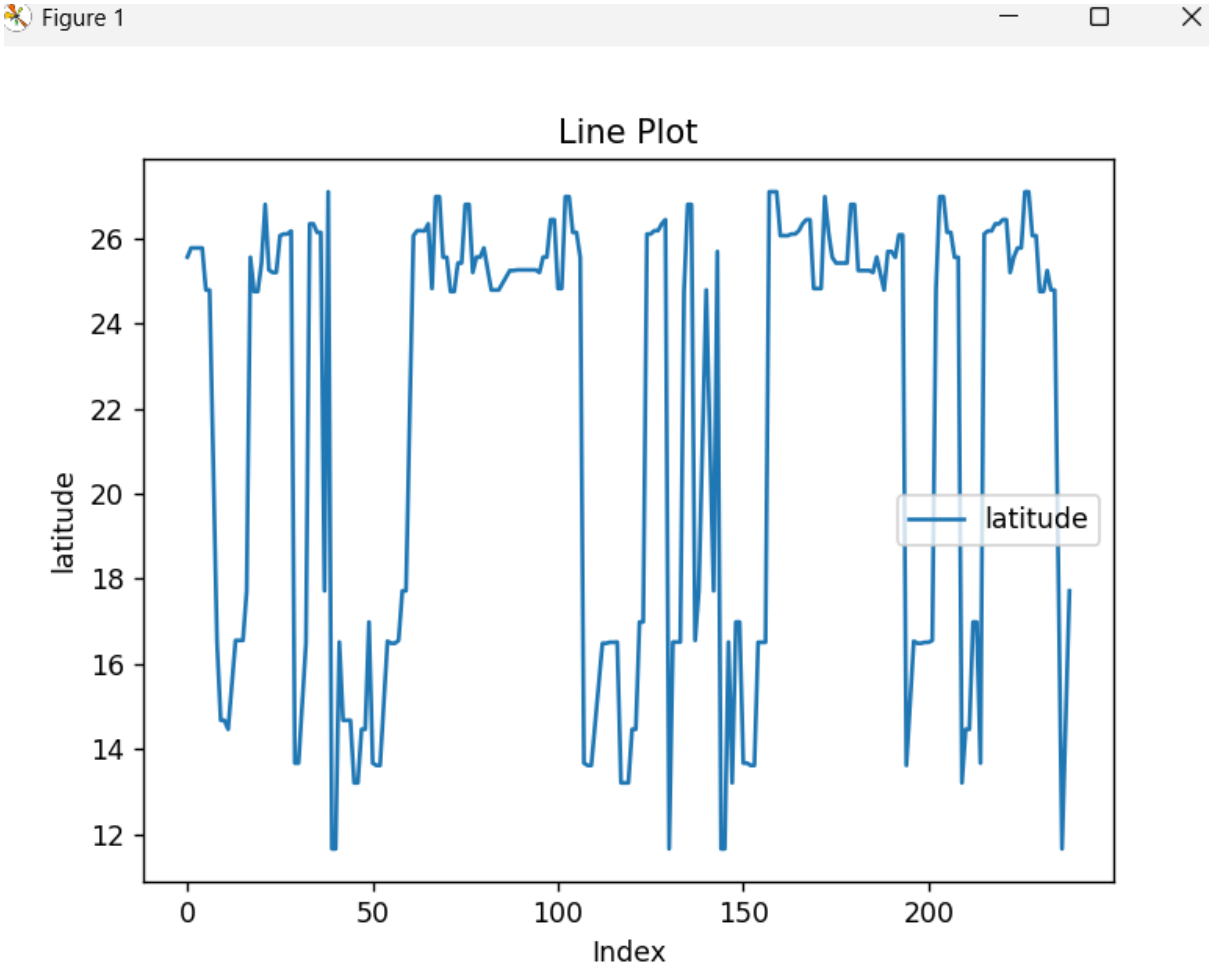
To gain deeper insights into air quality patterns, multiple visualization techniques are used. Each plot highlights different aspects of the dataset and helps in better interpretation of pollutant behavior.

7.1 Line Plot

Line plots are used to observe trends and changes in pollutant concentrations over time or across ordered locations. They help identify increasing or decreasing pollution levels and seasonal or spatial variations.

Purpose:

- Track pollution trends
- Compare changes across cities or pollutants

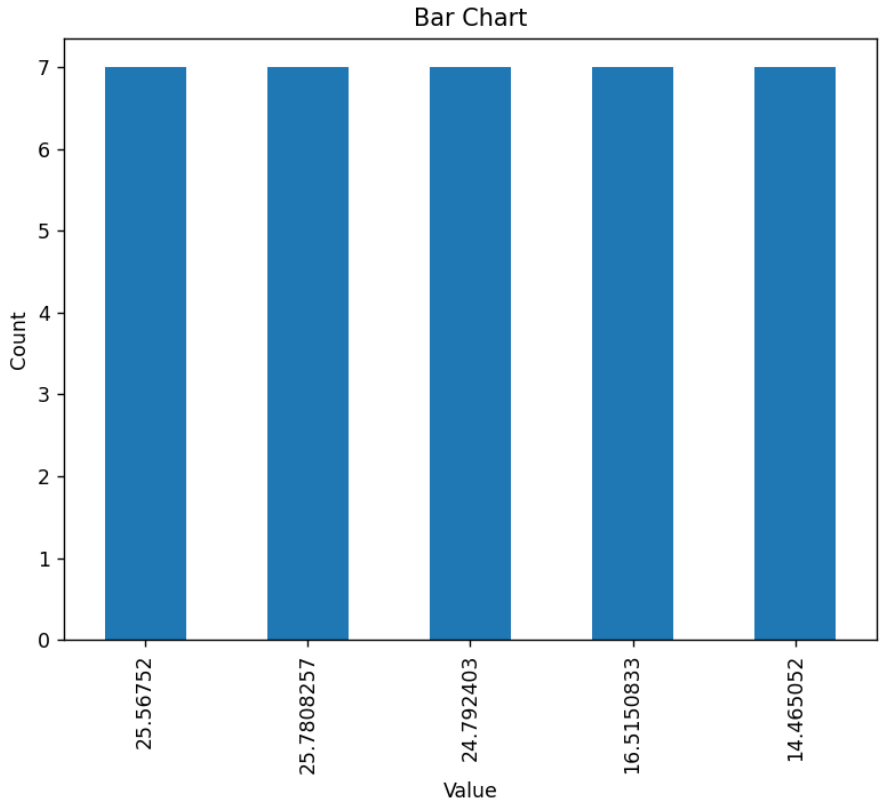


7.2 Bar Chart

Bar charts are effective for comparing average pollutant concentrations across different cities, states, or pollutant types. Each bar represents the magnitude of pollution for a specific category.

Purpose:

- Compare pollution levels between locations
- Identify highly polluted cities or regions

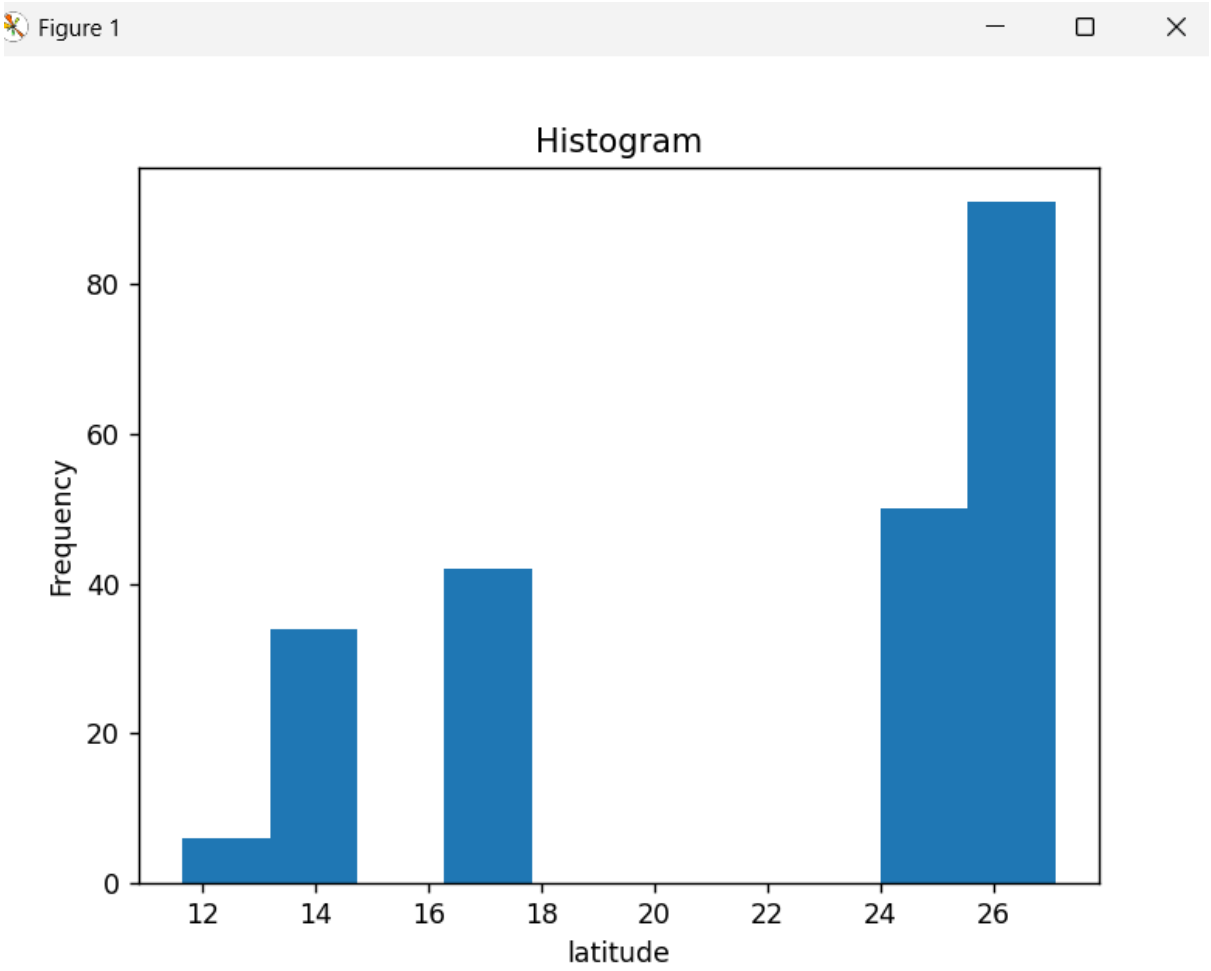


7.3 Histogram

Histograms display the frequency distribution of pollutant concentrations. They show how often values occur within specific ranges and help understand data skewness and spread.

Purpose:

- Understand data distribution
- Identify normal or skewed patterns



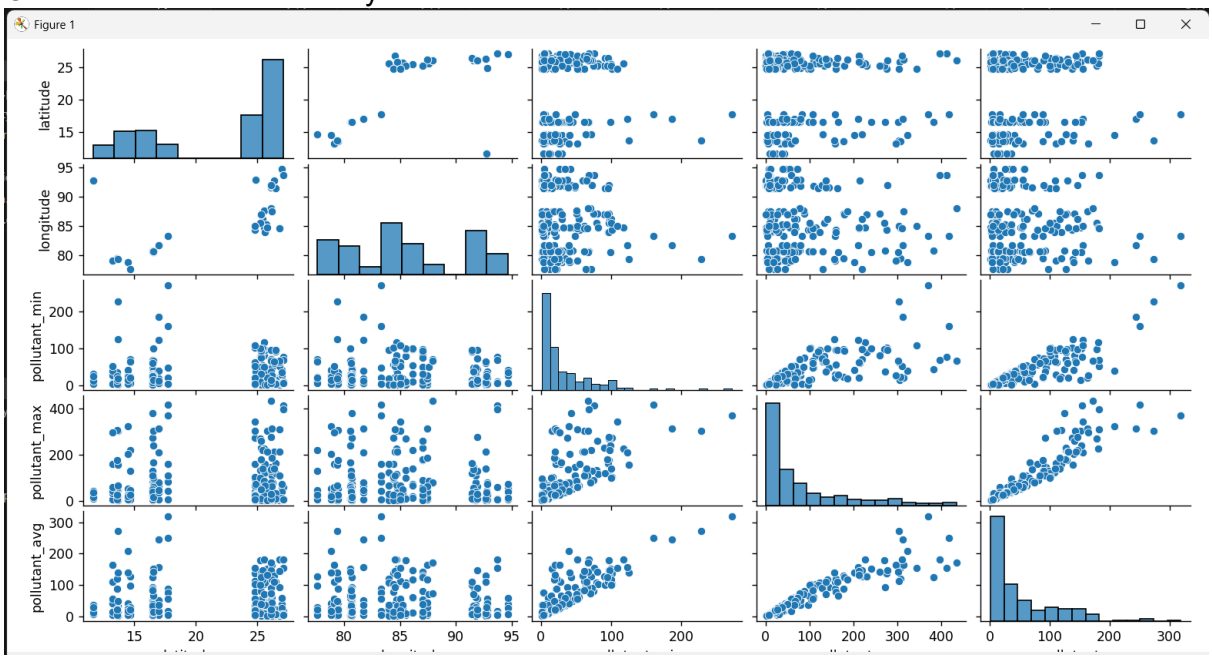
7.4 Pair Plot

Pair plots visualize pairwise relationships between multiple numerical variables in a dataset. They include scatter plots for variable combinations and histograms along the diagonal.

Purpose:

- Identify relationships between pollutants
- Detect clusters and patterns

- Observe correlations visually

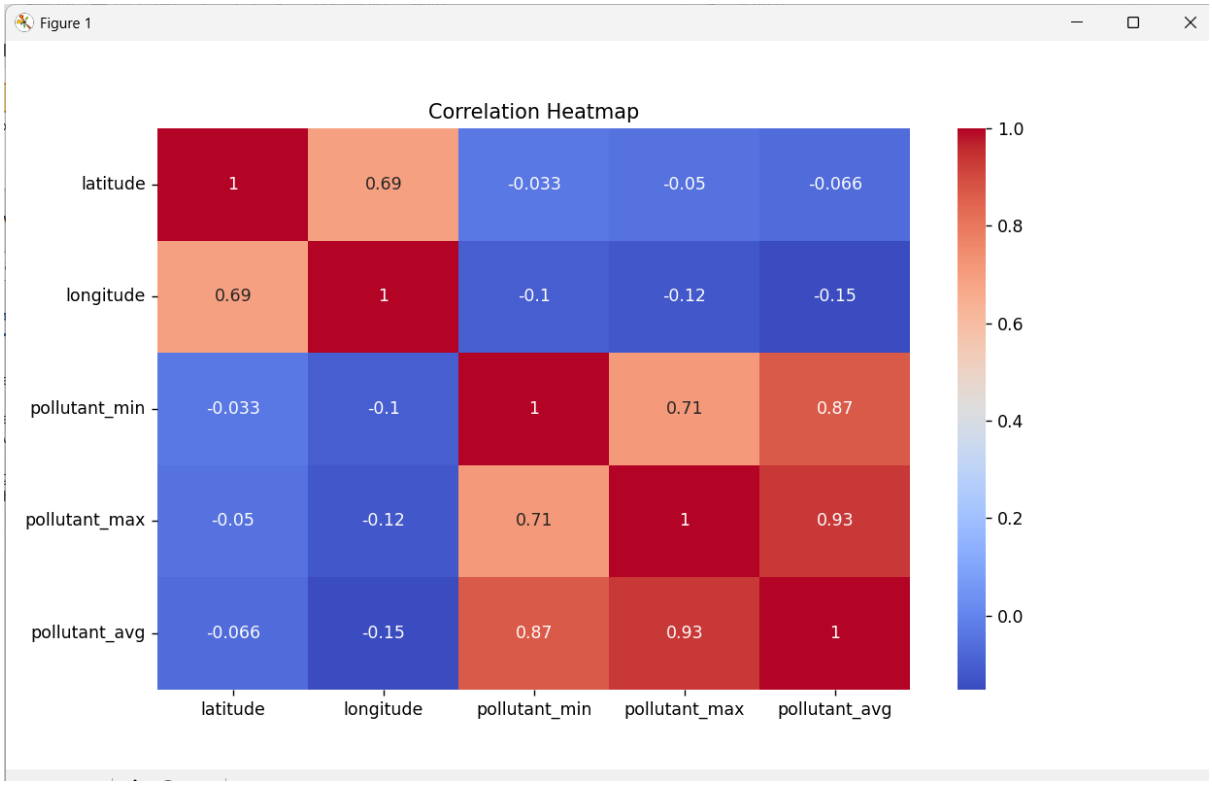


7.5 Correlation Heatmap

A correlation heatmap represents correlation coefficients between numerical variables using color gradients. Strong positive and negative correlations are easily identified.

Purpose:

- Measure strength and direction of relationships between pollutants
- Support feature selection for modelling

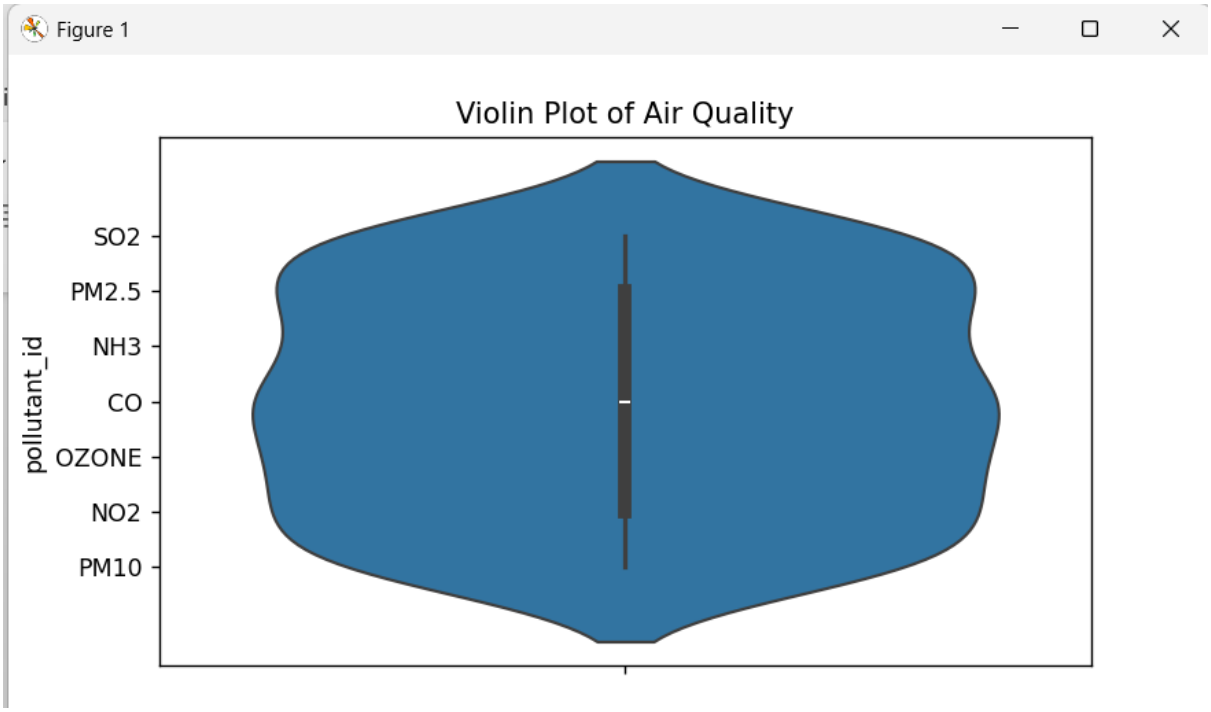


7.6 Violin Plot

A violin plot combines the features of a box plot and a density plot. It shows the distribution shape of pollutant concentrations along with summary statistics such as median and spread.

Purpose:

- Visualize data distribution and density
- Compare pollutant concentration distributions across cities or pollutant types
- Identify skewness and variability more clearly than box plots



---

8. Handling Outliers

```
/Users/Kaviya S/OneDrive/Desktop/dscience/main.py"

outlier in the dataset

     latitude  longitude  pollutant_min  pollutant_max  pollutant_avg
0         NaN        NaN            NaN            NaN            NaN
1         NaN        NaN            NaN          305.0            NaN
2         NaN        NaN            NaN            NaN            NaN
3         NaN        NaN            NaN            NaN            NaN
4         NaN        NaN            NaN            NaN            NaN
..        ...        ...            ...            ...            ...
233       NaN        NaN            NaN            NaN            NaN
238       NaN        NaN          160.0          417.0          251.0
```

8.1 Communicating Findings and Insights

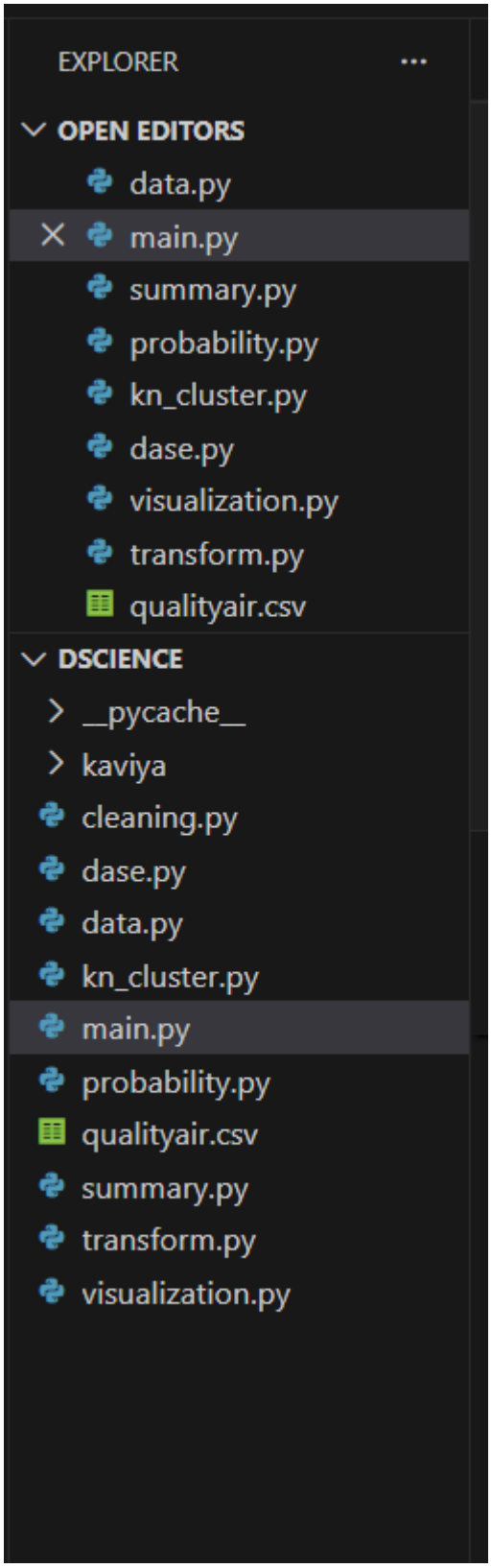
The final step focuses on presenting insights in a clear and understandable manner.

- **Comparative Analysis:** City-wise and pollutant-wise comparison
- **Variability Analysis:** Understanding range and dispersion of pollutants
- **Visual Summaries:** Charts and plots highlighting high-pollution regions

---

Conclusion

This project provides a comprehensive Exploratory Data Analysis of the **qualityair.csv** dataset. Through systematic data inspection, cleaning, transformation, statistical analysis, and visualization, the study reveals important patterns in air pollutant concentrations across Indian locations. The insights gained from this analysis support effective environmental monitoring and enable informed decision-making for air quality management.


EXPLORER
OPEN EDITORS
data.py
main.py
summary.py
probability.py
kn_cluster.py
dase.py
visualization.py
transform.py
qualityair.csv
DSCIENCE
__pycache__
kaviya
cleaning.py
dase.py
data.py
kn_cluster.py
main.py
probability.py
qualityair.csv
summary.py
transform.py
visualization.py