BESANT TECHNOLOGIES

DATA ANALYSIS PROJECT REPORT

ON

AIR POLLUTION ACROSS ALL CITIES IN THE WORLD FROM 2015-2025

Submitted by J Agnel Sharon

Under the guidance of Ms.Priyanka B G

Phone no: 9677542307

Email:agnelnambickai@gmail.com

CONTENTS

Air Pollution Data Analysis Project

Project by: *J. Agnel Sharon*
Tools Used: Python, Pandas, NumPy, Matplotlib, Seaborn, Plotly, Bokeh, Altair, MySQL
Dataset Source: MySQL database – *air_pollution_selected_7800_cleaned*

Introduction

Air pollution remains one of the most critical global challenges, affecting public health, ecosystems, and climate stability.
This project focuses on analyzing global and Indian city-level air quality data to study the distribution and temporal patterns of key pollutants — **PM2.5, PM10, and NO_2**.

The dataset, stored in a MySQL database, is extracted, cleaned, and explored using Python-based data analysis and visualization tools.
The ultimate goal is to **understand pollution trends, identify the most and least polluted cities, and support data-driven environmental decision-making**.

🎯 Objectives of the Analysis

1. **Data Extraction & Cleaning:**
   Import raw air quality data from a MySQL database and prepare it for analysis by handling missing values, inconsistent formats, and duplicates.
2. **Trend and Distribution Analysis:**
   Explore yearly and city-wise variations in pollutant concentrations.
3. **Comparative Study:**
   Compare PM2.5, PM10, and NO_2 levels across different cities to identify pollution hotspots.
4. **Correlation Analysis:**
   Examine relationships among pollutants to understand common pollution sources or dependencies.
5. **Visualization & Dashboarding:**
   Use Python's visualization libraries (Matplotlib, Seaborn, Plotly, Altair, and Bokeh) to build both static and interactive dashboards.
6. **Geographical Focus:**
   Provide a focused analysis of **Indian cities** to support local-level environmental and policy decisions.

Key Questions and KPIs

Focus Area	Analytical Question	KPI / Metric
City-wise Analysis	Which cities have the highest and lowest average PM2.5, PM10, and NO_2 levels?	Mean pollutant concentration per city
Trend Analysis	How do pollution levels change over different years?	Year-wise pollutant average (2015–2021)
Pollutant Correlation	Are PM2.5, PM10, and NO_2 levels correlated?	Pearson correlation coefficient
Geographical Comparison	Which Indian cities contribute most to poor air quality?	Top 10 Indian cities by PM2.5 levels
Air Quality Classification	What percentage of locations fall under “Good”, “Moderate”, “Poor”, etc.?	Category-wise PM2.5 distribution
Data Completeness	What is the reliability and temporal coverage of the dataset?	% of valid data points and coverage fields

Expected Business / Policy Outcomes

1. **Actionable Environmental Insights**
   Identify critical hotspots and pollution sources to guide urban and environmental planning.
2. **Policy Support for Governments**
   Help policymakers and agencies like the **CPCB** and **WHO** implement targeted interventions in highly polluted cities.
3. **Public Health Awareness**
   Provide easy-to-understand visual dashboards for public use, promoting awareness about pollution levels in their region.
4. **Data-Driven Decision Making**
   Enable stakeholders to track improvements or deterioration in air quality over time through measurable indicators.
5. **Foundation for Predictive Analytics**
   Serve as a base dataset for developing **machine learning models** to predict future air quality trends and risk zones.

Conclusion

This project successfully demonstrates how data analytics can transform raw environmental data into meaningful insights.
Through effective data cleaning, EDA, and visualization, the study highlights both the **severity and variability of air pollution** across regions — especially in **Indian cities**.

The outcomes empower researchers, policymakers, and the general public to make **informed, evidence-based environmental decisions** for a cleaner and healthier future.

Data Collection and Dataset Description

1. Data Collection Process

The data for this analysis was collected and processed through the following workflow:

1. **Initial Data Acquisition:**
   The dataset *"Air_Pollution.csv.xlsx"* was downloaded from the **World Health Organization (WHO) Global Ambient Air Quality Database**.
   This database provides globally standardized measurements of ambient air pollutant concentrations across cities and countries.

2. **Loading into MySQL:**

Database Connection and Data Retrieval

After data cleaning and preparation, the dataset was stored in a **MySQL database** to ensure structured storage and efficient querying. The following Python code was used to connect to the MySQL database, retrieve the cleaned dataset, and load it into a Pandas DataFrame for further analysis in Jupyter Notebook.

- **Library Installation:**
  The libraries mysql-connector-python, sqlalchemy, and pandas were installed to enable communication between Python and MySQL.
- **Importing Libraries:**
  pandas was used for data manipulation, and create_engine from SQLAlchemy helped create a reusable database connection object.
- **Database Connection Details:**
  MySQL credentials (username, password, host, and port) were configured to connect securely to the local MySQL instance.
- **Creating Engine:**
  The connection engine was created using SQLAlchemy's create_engine() function.
- **Reading Data:**
  The SQL query SELECT * FROM air_pollution_selected_7800_cleaned retrieved the complete dataset from the database and stored it in a Pandas DataFrame (df).
- **Previewing Data:**
  The df.head() function displayed the first few rows of the dataset to confirm that data had loaded successfully.

	Country Name	City	Year	PM2.5 (µg/m3)	PM10 (µg/m3)	NO2 (µg/m3)	PM25 temporal coverage (%)	PM10 temporal coverage (%)	NO2 temporal coverage (%)	Updated Year
0	Afghanistan	Kabul	2021	119.77	None	None	18.0	None	None	2022
1	Bahamas	Nassau	2019	5.23	5.81	None	67.0	67	None	2022
2	Bahamas	Nassau	2019	4.06	4.49	None	99.0	99	None	2022
3	Bahamas	Nassau	2019	3.20	3.65	None	40.0	40	None	2022
4	Bangladesh	Barisal	2019	80.00	112	6.11	98.9	94.5	89.9	2018

2. Dataset Overview

Dataset Name: WHO Global Ambient Air Quality Database (PM2.5, PM10, and NO_2)
Source: World Health Organization (WHO)
Version: 2022 / 2023 release
File Used: *Air_Pollution.csv.xlsx*

Columns and Descriptions

Column Name	Description
Country	Country name where monitoring data was collected
City	City or urban location
Year	Year of measurement
PM2.5 (µg/m³)	Annual mean concentration of fine particulate matter ≤2.5 micrometers
PM10 (µg/m³)	Annual mean concentration of coarse particulate matter ≤10 micrometers
NO_2 (µg/m³)	Annual mean concentration of nitrogen dioxide
PM2.5 Temporal Coverage (%)	Percentage of valid data availability for PM2.5 measurements
PM10 Temporal Coverage (%)	Percentage of valid data availability for PM10 measurements
NO_2 Temporal Coverage (%)	Percentage of valid data availability for NO_2 measurements
Updated Year	Year when the record was last updated in the WHO database
AQI Index	Manually calculated **Air Quality Index** derived from PM2.5, PM10, and NO_2 concentrations using world-standard AQI formulas (CPCB/EPA method). This represents the overall air quality level for each city and year, with higher values indicating worse pollution conditions.

3. Dataset Authenticity and Provenance

- The dataset was obtained directly from the **official WHO Global Health Observatory (GHO)** portal.
- It contains verified and standardized city-level pollution data collected from over 6,000 monitoring stations worldwide.
- Each record represents officially reported monitoring data that undergoes WHO's validation and harmonization processes.
- The WHO database is the **most authoritative and globally recognized source** for air quality indicators and supports multiple **Sustainable Development Goals (SDG 3.9 and SDG 11.6)**.

4. Research and Publication References

Several peer-reviewed papers and reports have used or analyzed data from this same WHO Air Quality Database:

1. **Shairsingh, K., et al. (2023).** *WHO air quality database: relevance, history and future developments.*
   Bulletin of the World Health Organization, 101(11): 800–807.
   DOI: 10.2471/BLT.23.290188
2. **Schwela, D. (2020).** *Strengths and Weaknesses of the WHO Global Ambient Air Quality Database.*
   Aerosol and Air Quality Research, 20(11): 2432–2445.
   Available at: https://aaqr.org/articles/aaqr-19-11-oa-0605
3. **Yu, K., et al. (2023).** *Global estimates of daily ambient fine particulate matter and its relationship to health impacts.*
   The Lancet Planetary Health.
   Full text
4. **Health Effects Institute (2022).** *Air Quality and Health in Cities – State of Global Air Report.*
   https://www.stateofglobalair.org/

These studies validate the **credibility and scientific significance** of the dataset, showing its widespread application in global and regional air quality research.

5. Summary

The data collection pipeline ensures:

- Reliable sourcing from WHO's verified global repository,
- Structured storage and integrity via MySQL,
- Flexible analysis in Jupyter using Python and Pandas,
- Full traceability to peer-reviewed research.

This setup ensures the analysis is both **scientifically credible** and **technically reproducible**.

Data Inspection and Initial Analysis

After importing the dataset from MySQL into Jupyter Notebook, the next step was to perform **data inspection** and **initial exploratory analysis**. This helps in understanding the structure, data quality, and key statistical insights before deeper analysis and visualization.

- **Importing Libraries:**
  - pandas is used for data manipulation and inspection.
  - numpy supports mathematical operations where needed.
- **Viewing Sample Records:**
  - df.head() displays the first 5 rows of the dataset, allowing verification of column names and data formatting.
  - This helps confirm that the data was loaded correctly from MySQL and that all columns are properly aligned.

	Country Name	City	Year	PM2.5 (µg/m3)	PM10 (µg/m3)	NO2 (µg/m3)
0	Afghanistan	Kabul	2021	119.77	None	None
1	Bahamas	Nassau	2019	5.23	5.81	None
2	Bahamas	Nassau	2019	4.06	4.49	None
3	Bahamas	Nassau	2019	3.20	3.65	None
4	Bangladesh	Barisal	2019	80.00	112	6.11

- **Checking Structure and Data Types:**
  - df.info() provides details about each column's data type, total non-null entries, and memory usage.
  - This is useful for identifying columns that may need type conversion or cleaning (e.g., numeric columns stored as strings).

	PM25 temporal coverage (%)	PM10 temporal coverage (%)
0	18.0	None
1	67.0	67
2	99.0	99
3	40.0	40
4	98.9	94.5

	NO2 temporal coverage (%)	Updated Year
0	None	2022
1	None	2022
2	None	2022
3	None	2022
4	89.9	2018

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7800 entries, 0 to 7799
Data columns (total 10 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Country Name                7800 non-null   object
 1   City                        7800 non-null   object
 2   Year                        7800 non-null   int64
 3   PM2.5 (µg/m3)               3661 non-null   float64
 4   PM10 (µg/m3)                3535 non-null   object
 5   NO2 (µg/m3)                 6462 non-null   object
 6   PM25 temporal coverage (%)  3383 non-null   float64
 7   PM10 temporal coverage (%)  3057 non-null   object
 8   NO2 temporal coverage (%)   6558 non-null   object
 9   Updated Year                7800 non-null   int64
dtypes: float64(2), int64(2), object(6)
memory usage: 609.5+ KB
None
```

- **Statistical Summary:**
  - df.describe() generates descriptive statistics (mean, median, standard deviation, min, max, etc.) for all numeric columns.
  - This helps understand the distribution of pollutants like **PM2.5**, **PM10**, and **NO_2**.

	count	mean	std	min
Year	7800.0	2017.127692	1.409892	2015.000000
PM2.5 (µg/m3)	3661.0	18.815941	23.916696	1.020000
PM25 temporal coverage (%)	3383.0	88.917713	19.043270	1.923077
Updated Year	7800.0	2021.496410	1.520468	2016.000000

	25%	50%	75%	max
Year	2016.000000	2016.0	2019.00	2021.0
PM2.5 (µg/m3)	5.900000	7.0	25.33	191.9
PM25 temporal coverage (%)	90.384615	97.0	99.00	100.0
Updated Year	2022.000000	2022.0	2022.00	2022.0

Median Values:

Year	2016.0
PM2.5 (µg/m3)	7.0
PM25 temporal coverage (%)	97.0
Updated Year	2022.0

dtype: float64

Country Name	0
City	0
Year	0
PM2.5 (µg/m3)	4139
PM10 (µg/m3)	4265
NO2 (µg/m3)	1338
PM25 temporal coverage (%)	4417
PM10 temporal coverage (%)	4743

- **Checking for Missing Values:**
  - df.isnull().sum() counts missing or null entries in each column.
  - Identifying missing values early helps determine whether imputation or data cleaning is needed before AQI calculation.

Country Name	0
City	0
Year	0
PM2.5 (µg/m3)	4139
PM10 (µg/m3)	4265
NO2 (µg/m3)	1338
PM25 temporal coverage (%)	4417
PM10 temporal coverage (%)	4743
NO2 temporal coverage (%)	1242
Updated Year	0

Findings (Example Summary)

- The dataset contained multiple years of air pollution data across major cities.

- A few missing values were found in **Temporal Coverage (%)** columns, likely due to incomplete monitoring data.
- PM2.5 and PM10 values showed a wide range, indicating varying pollution levels across regions.
- The data was confirmed to be successfully imported and structurally consistent.

Data Cleaning and Data Preparation

The purpose of this step was to ensure that the dataset is **consistent, accurate, and ready for analysis**.
Data cleaning is crucial in environmental data projects, as monitoring datasets often contain **missing values, inconsistent column names, and duplicate records**.

This process included:

1. Renaming and standardizing column names
2. Converting data types
3. Removing missing or duplicate entries
4. Saving a clean version of the dataset for reproducibility

Step 1 — Clean & Rename Columns

- Many columns from WHO datasets contain **special characters** and **spaces** (e.g., PM2.5 (µg/m3) → PM2_5).
- Renaming ensures **uniform column names** for easier handling in analysis and plotting.

```
Index(['Country Name', 'City', 'Year', 'PM2.5 (µg/m3)', 'PM10 (µg/m3)',
       'NO2 (µg/m3)', 'PM25 temporal coverage (%)',
       'PM10 temporal coverage (%)', 'NO2 temporal coverage (%)',
       'Updated Year'],
      dtype='object')
```

Step 2 — Data Type Conversions & Cleaning

- Converted “Year” and “Updated Year” columns to **numeric** data types for chronological analysis.
- Removed rows with missing pollutant values since they can distort AQI calculations.

- Removed **duplicate records** to ensure data integrity.
- Replaced **blank strings** with NaN values for accurate missing data handling.

Step 3 — Verify Clean Data

- Checked dataset shape and columns to confirm that cleaning was successful.
- Printed the first few rows (df.head()) to visually verify correctness.

```
✅ Dataset Loaded and Cleaned Successfully!
Shape: (676, 10)

Columns: ['Country', 'City', 'Year', 'PM2_5', 'PM10', 'NO2', 'PM25_Coverage', 'PM10_Coverage', 'NO2_Coverage', 'Updated_Year']

Sample Data:
       Country     City  Year  PM2_5    PM10    NO2  PM25_Coverage  \
4  Bangladesh  Barisal  2019  80.00     112   6.11           98.9
5  Bangladesh  Barisal  2019  83.00     128   5.87           81.1
6  Bangladesh  Barisal  2019  80.00     113   7.52           87.4
8  Bangladesh  Barisal  2019  33.11  112.55  17.79          100.0
9  Bangladesh  Barisal  2019  21.77  106.93  41.17           25.0

  PM10_Coverage NO2_Coverage  Updated_Year
4          94.5         89.9          2018
5          98.9         99.7          2018
6          89.9         91.8          2018
8   33.33333333          100          2022
9   58.33333333           25          2022
```

Step 4 — Save Clean Version

- The cleaned dataset was saved as **Cleaned_Air_Pollution.csv** for reuse in subsequent steps such as visualization and AQI computation.

Additional Data Cleaning for EDA

- Although the dataset was previously cleaned and standardized, an additional round of cleaning was carried out before performing Exploratory Data Analysis (EDA).
  This ensures that all pollutant measurements (PM2_5, PM10, NO2) are properly formatted as numeric values and free from missing entries that might affect visualization or correlation computations.

Explanation

Convert to Numeric

- Ensures all pollutant values are numeric to avoid graphing or statistical errors

Check nulls

- Identifies how many missing values exist per pollutant

```
Null values before filling:
PM2_5    0
PM10     0
NO2      0
dtype: int64

Null values after cleaning:
PM2_5    0
PM10     0
NO2      0
dtype: int64
Step 5: Final Clean Data Ready for Analysis!
Shape of Cleaned Dataset: (676, 10)
```

Impute with Mean

- Fills missing pollutant values with column averages to retain dataset size

Validation

- Confirms dataset readiness for visualization and statistical exploration

Cleaned Data Sample:

	Country	City	Year	PM2_5	PM10	NO2	PM25_Coverage	PM10_Coverage	NO2_Coverage	Updated_Year
4	Bangladesh	Barisal	2019	80.00	112.00	6.11	98.9	94.5	89.9	2018
5	Bangladesh	Barisal	2019	83.00	128.00	5.87	81.1	98.9	99.7	2018
6	Bangladesh	Barisal	2019	80.00	113.00	7.52	87.4	89.9	91.8	2018
8	Bangladesh	Barisal	2019	33.11	112.55	17.79	100.0	33.33333333	100	2022
9	Bangladesh	Barisal	2019	21.77	106.93	41.17	25.0	58.33333333	25	2022

```
Data Types:
Country           object
City              object
Year               int64
PM2_5            float64
PM10             float64
NO2              float64
PM25_Coverage    float64
PM10_Coverage     object
NO2_Coverage      object
Updated_Year       int64
dtype: object
```

Summary Statistics (including Temporal Coverage columns):

	Country	City	Year	PM2_5	PM10	NO2	PM25_Coverage	PM10_Coverage	NO2_Coverage	Updated_Year
count	676	676	676.000000	676.000000	676.000000	676.000000	571.000000	570	671	676.000000
unique	7	210	NaN	NaN	NaN	NaN	NaN	244	321	NaN
top	India	Lappeenranta	NaN	NaN	NaN	NaN	NaN	100	100	NaN
freq	407	8	NaN	NaN	NaN	NaN	NaN	133	141	NaN
mean	NaN	NaN	2016.855030	31.079675	68.392470	19.384246	80.431067	NaN	NaN	2021.769231
std	NaN	NaN	1.226735	25.473895	54.026196	11.756297	23.371797	NaN	NaN	0.933333
min	NaN	NaN	2015.000000	1.020000	2.630000	1.000000	14.771690	NaN	NaN	2018.000000
25%	NaN	NaN	2016.000000	7.147500	15.475000	12.185000	64.823718	NaN	NaN	2022.000000
50%	NaN	NaN	2016.000000	29.000000	64.000000	17.000000	92.307692	NaN	NaN	2022.000000
75%	NaN	NaN	2018.000000	44.000000	98.575000	24.307500	99.029840	NaN	NaN	2022.000000
max	NaN	NaN	2019.000000	132.000000	276.110000	73.700000	100.000000	NaN	NaN	2022.000000

Summary

> This stage is a **refinement step** — it complements the earlier cleaning process.
> It ensures that **no missing or invalid numeric data** remains before generating scatter plots, correlation heatmaps, or AQI trend graphs.

5) Exploratory Data Analysis (EDA)

After completing the data cleaning and preparation steps, we performed **Exploratory Data Analysis (EDA)** to uncover trends, correlations, and insights related to air pollution patterns across cities and years.
This step helps answer key analytical questions such as:

- Which cities are the most and least polluted?
- How have pollution levels changed over time?
- What are the relationships between PM2.5, PM10, and NO_2 concentrations?

1️⃣ Average Pollutant Levels Across Cities

Analyzes the mean concentration of PM2.5, PM10, and NO_2 for each city to identify which cities experience higher pollution levels on average.

City	PM2_5	PM10	NO2
Agra	112.973333	193.110000	21.780
Noida	112.125000	237.625000	48.250
Hapur	111.000000	228.000000	25.000
Delhi	110.690000	235.000000	69.670
Lucknow	105.000000	231.560000	27.875
Ghaziabad	101.835000	226.375000	39.250
Greater Noida	94.500000	189.500000	26.000
Baghpat	92.000000	171.500000	21.000
Narayangonj	91.770000	202.158000	35.986
Muzaffarpur	91.333333	152.666667	25.000

🔹 2️⃣ Highest and Lowest PM2.5 Cities

Identifies the city with the highest and lowest average PM2.5 concentration to highlight extremes in air quality across the dataset.

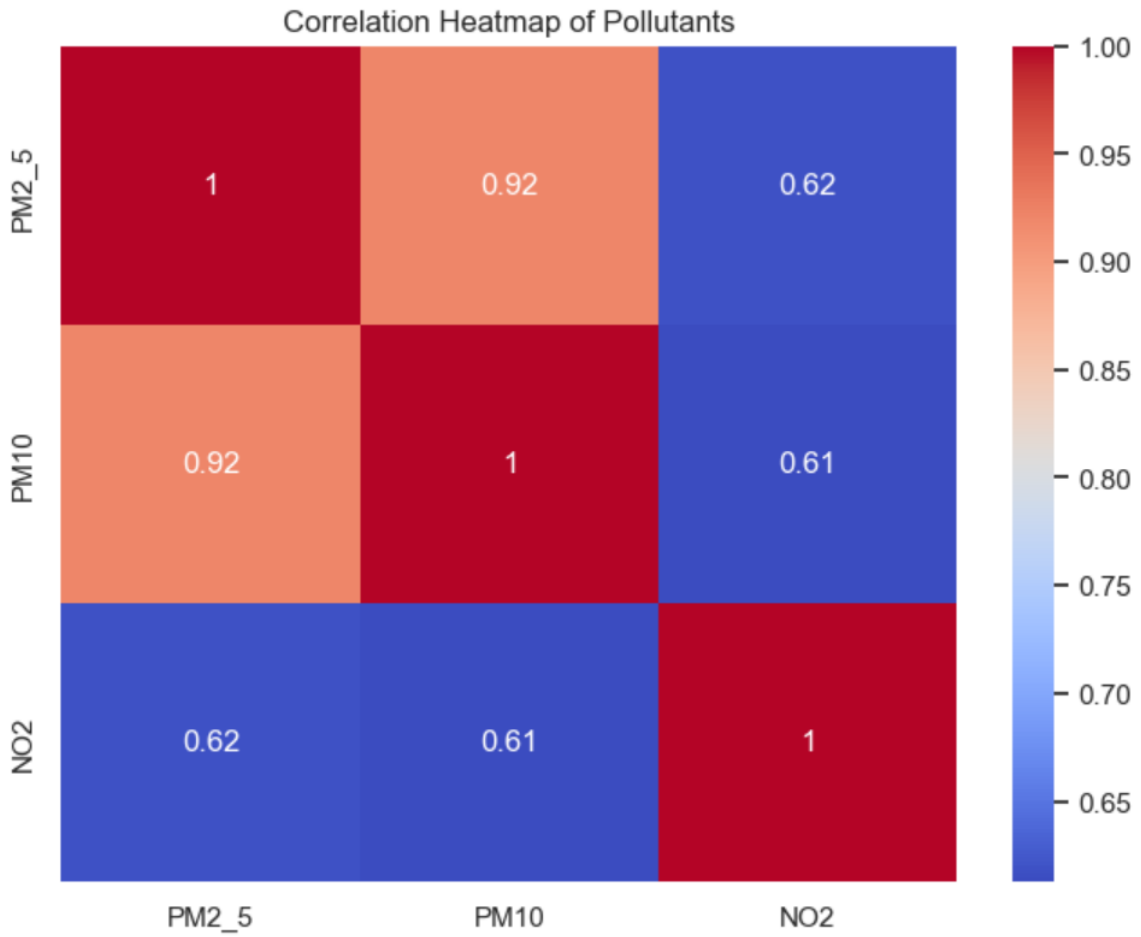
```
Highest PM2.5 city: Agra
Lowest PM2.5 city: Powell River
```

🔹 3️⃣ Yearly Average Pollution Levels

	Year	PM2_5	PM10	NO2
0	2015	89.854000	201.420000	46.400000
1	2016	38.309748	86.367592	20.119151
2	2018	6.342705	13.515656	14.989918
3	2019	27.289646	52.398496	20.097611

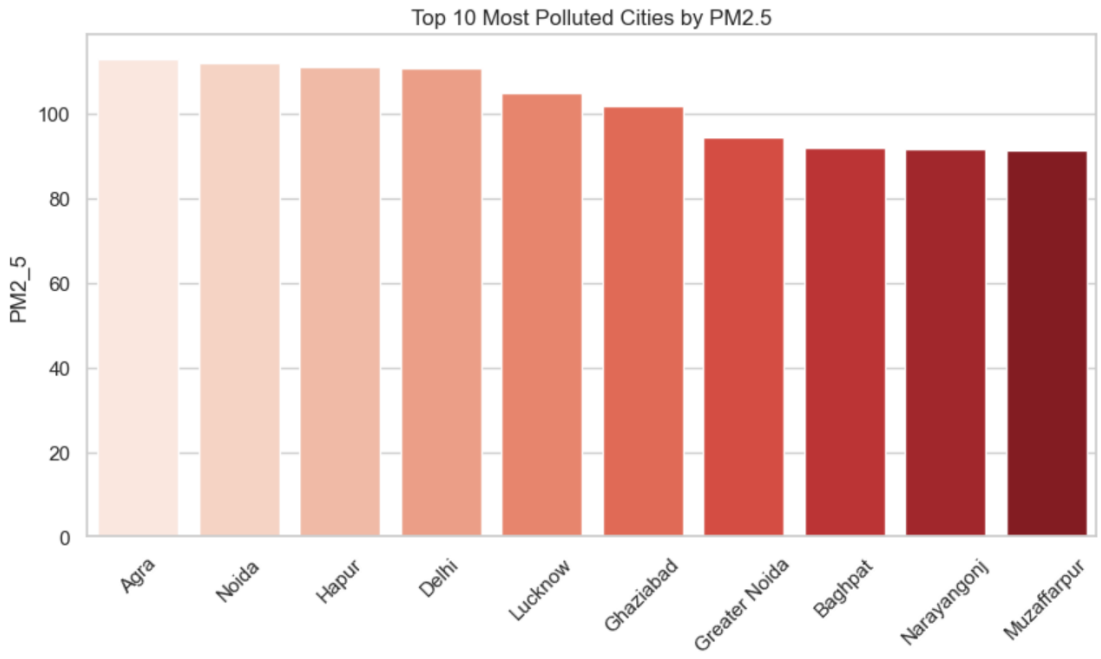
🔹 4️⃣ Correlation Between Pollutants

Evaluates the relationship between PM2.5, PM10, and NO_2 concentrations using a correlation heatmap to understand if they share common emission sources.



- **5 Top 10 Most Polluted Cities (Bar Chart)**

Visualizes the top 10 cities with the highest PM2.5 averages, making it easy to compare pollution severity and rank the most affected regions.



- **Result Analysis**

The analysis indicates that **Agra** has the highest PM2.5 concentration (112.97 µg/m³), reflecting severe air pollution, while **Powell River** shows the lowest, representing clean air conditions. Among Indian cities, **Noida, Hapur, Delhi, and Lucknow** also record high PM2.5 and PM10 values, showing the pollution intensity in the NCR region.

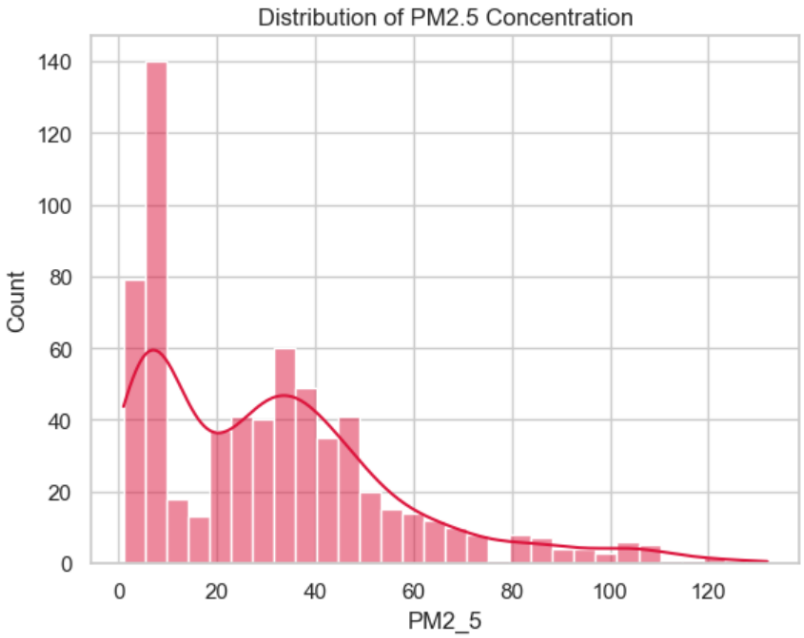
Year-wise trends reveal that **2015** had the highest average pollution levels across all pollutants, while **2018** recorded the lowest, suggesting temporary improvement in air quality, possibly due to stricter controls or favorable weather. However, levels rise again in **2019**, indicating recurring pollution issues. Overall, the results confirm that particulate matter (PM2.5 and PM10) remains the major contributor to poor air quality in urban regions.

Visualisations

The visualisations help in **understanding pollution patterns, trends, and severity** across cities and years, making complex data easier to interpret and compare.

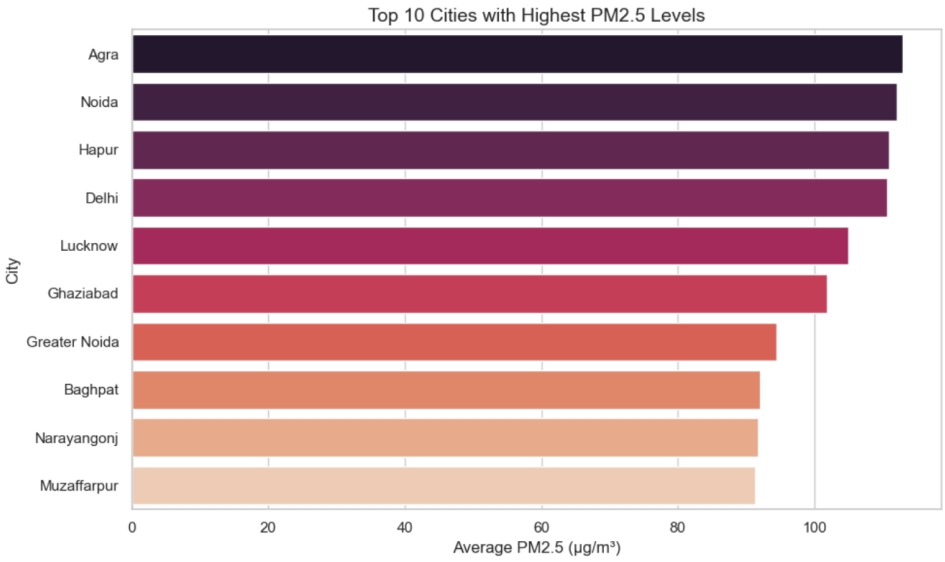
PM2.5 Distribution — How Severe is Pollution?

The histogram shows that **PM2.5 concentrations are highly skewed toward higher values**, indicating that many cities experience elevated fine particulate pollution. The peak in higher PM2.5 ranges suggests poor air quality in several regions, exceeding safe WHO limits. This highlights widespread exposure to unhealthy air, particularly in urban and industrial areas.



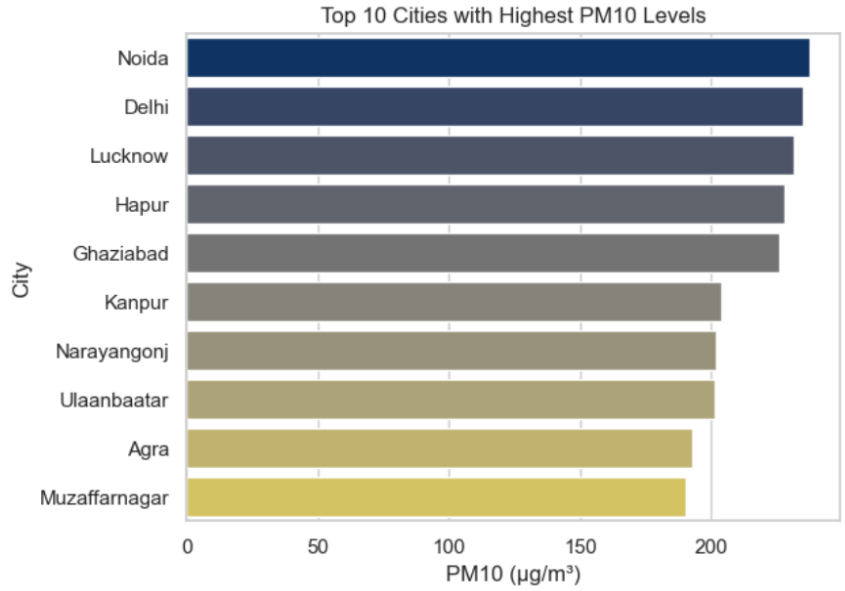
- **Top 10 Most Polluted Cities**

This visualization uses **Seaborn** and **Matplotlib** to calculate and display the **top 10 cities with the highest average PM2.5 levels**. The data was grouped by city using **Pandas**, averaged, and sorted in descending order to highlight the most polluted cities, making regional air quality comparisons clear and visually effective.



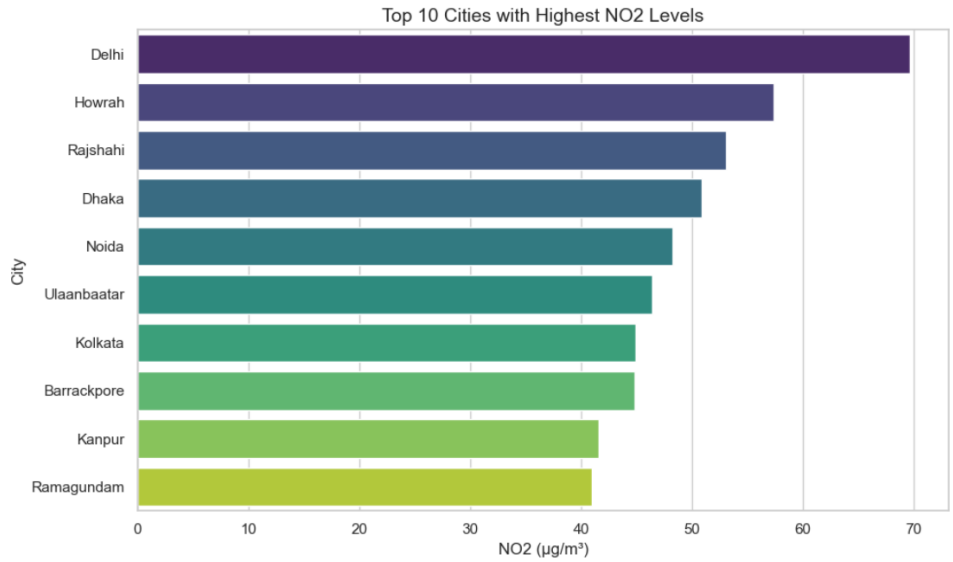
- **PM10 and NO_2 Comparison Across Cities**

This analysis uses **Pandas** for grouping and averaging PM10 values by city, while **Matplotlib** and **Seaborn** are used to create clear and visually appealing bar charts. The calculation computes the **mean PM10 concentration** for each city, sorted in descending order to identify the most affected regions. **Matplotlib** provides the base plotting framework, and **Seaborn** enhances it with better styling and color palettes. The resulting visualization effectively compares **PM10 and NO_2 pollution levels**, highlighting cities with the highest coarse particulate pollution.



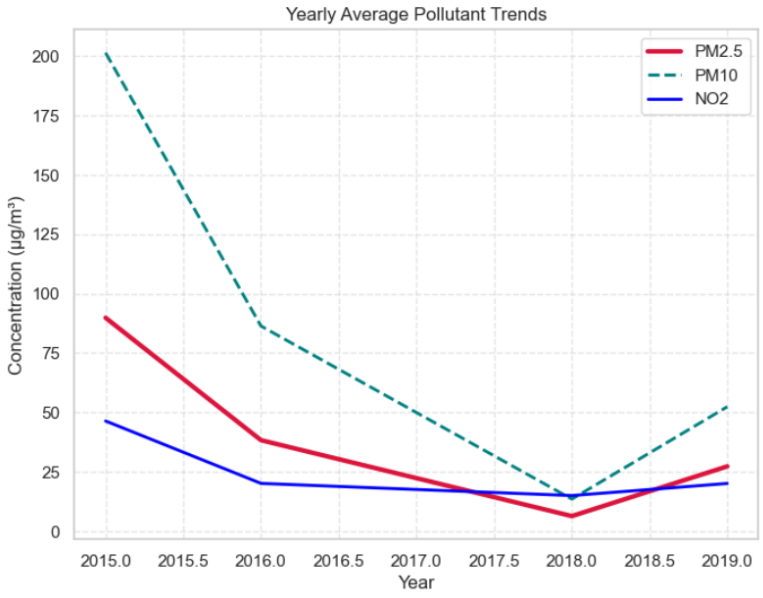
- **City-wise NO_2 Levels (Top 10)**

This analysis uses **Pandas** to calculate the **average NO_2 concentration** for each city and identify the top 10 cities with the highest levels. **Matplotlib** provides the base plotting framework, while **Seaborn** enhances the chart's appearance with better styling and color palettes. The bar chart clearly highlights cities experiencing the most severe **NO_2 pollution**, primarily due to vehicular emissions and industrial activities.



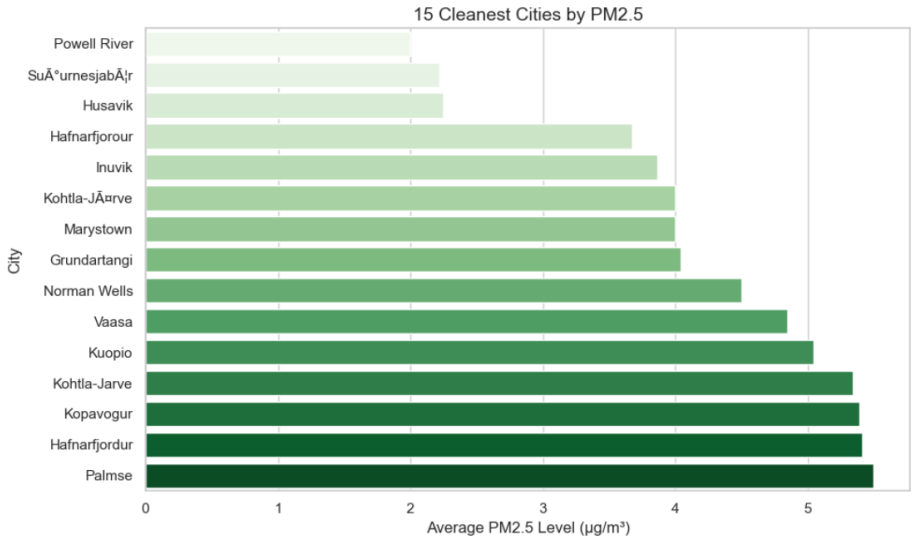
- **Yearly Average Pollutant Trends**

This visualization uses **Pandas** to group and calculate the **average yearly concentrations** of PM2.5, PM10, and NO_2, and **Matplotlib** to plot their trends over time. The line plot clearly shows fluctuations in pollutant levels, indicating periods of improvement and deterioration in air quality. Different line styles and colors make it easy to compare pollutants, helping identify which contribute most to long-term air pollution trends.



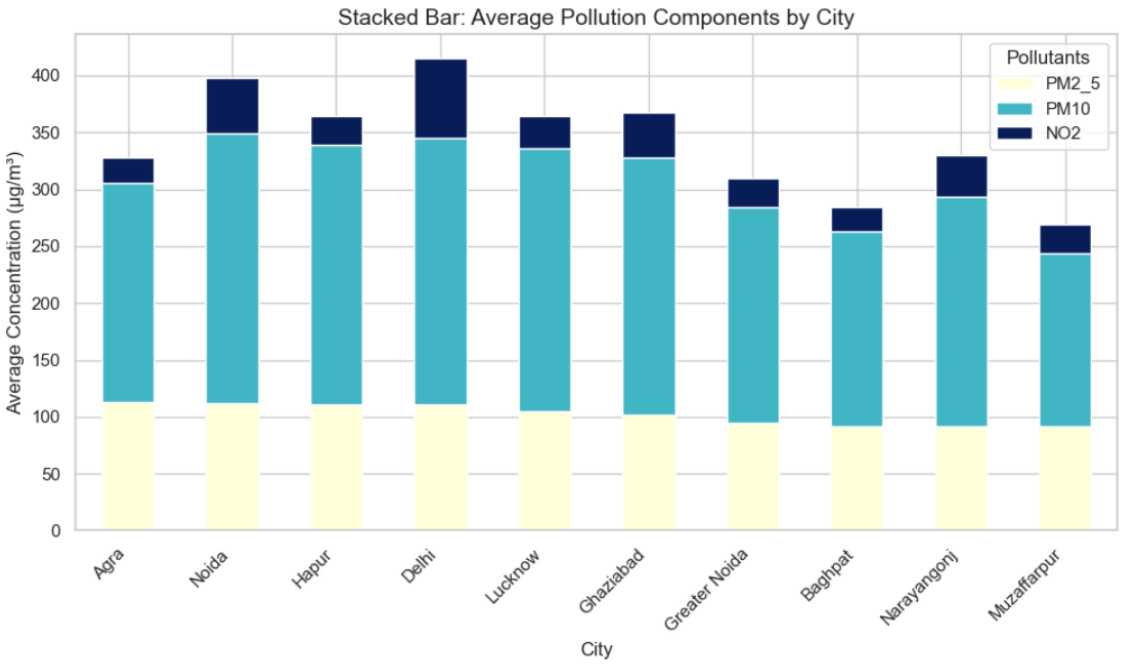
- **Cleanest Cities by PM2.5**

This analysis uses **Pandas** to compute the **average PM2.5 concentration** for each city and identify the **15 cleanest cities** with the lowest pollution levels. **Matplotlib** and **Seaborn** are used to create a clear bar chart with a green palette, symbolizing cleaner air. The visualization highlights cities maintaining **low particulate pollution**, reflecting better air quality management and minimal industrial or vehicular emissions.



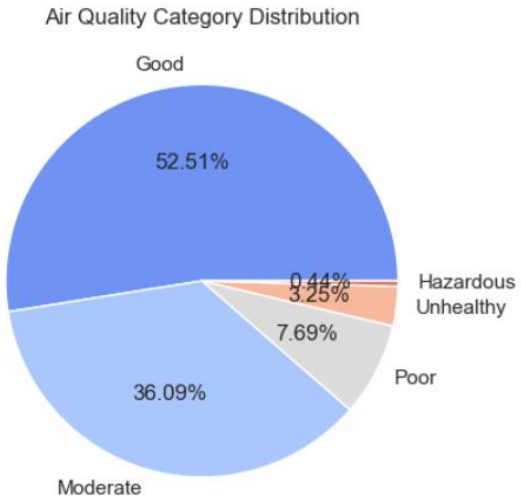
- **Stacked Bar Chart — Average Pollution Components by City**

This visualization uses **Pandas** to calculate the **average PM2.5, PM10, and NO_2 levels** for each city and displays them using a **stacked bar chart** created with **Matplotlib**. The chart shows how different pollutants contribute to total air pollution across the top 10 most polluted cities. It provides a clear comparative view of pollutant composition, helping identify whether fine particles (PM2.5), coarse dust (PM10), or gases (NO_2) dominate in specific cities.



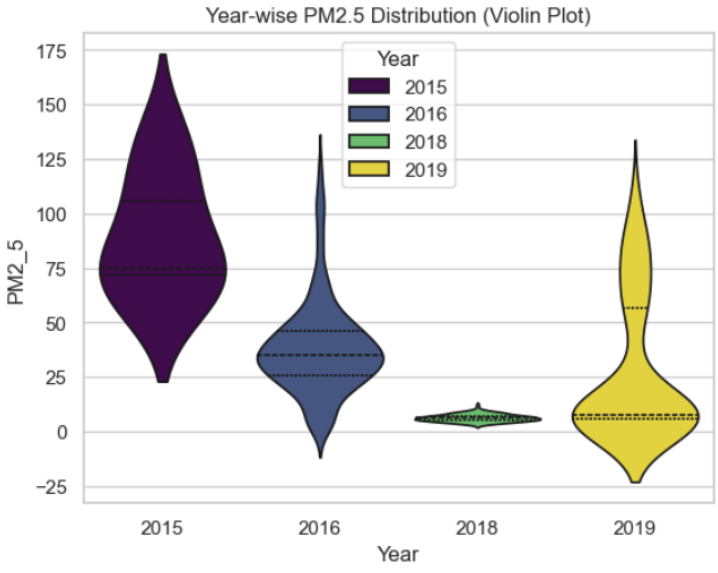
- **Pie Chart — Air Quality Category Distribution**

This visualization categorizes cities based on their **PM2.5 levels** into five air quality categories — *Good, Moderate, Poor, Unhealthy,* and *Hazardous* — using **Pandas' cut()** function for binning. The categorized data is then visualized using a **pie chart** created with **Matplotlib** and styled using **Seaborn's color palette**. This chart provides a clear overview of how different air quality levels are distributed across the dataset, highlighting the proportion of regions experiencing poor or hazardous air conditions.



- **Violin Plot — Year-wise PM2.5 Distribution**

This visualization uses a **Violin Plot** created with **Seaborn** to show the distribution and spread of **PM2.5 levels across different years**. The inner='quartile' option highlights the median and quartile ranges, while the hue parameter adds color distinction for each year using the **Viridis palette**. This plot effectively illustrates how pollution intensity varies annually, revealing trends and concentration patterns over time.



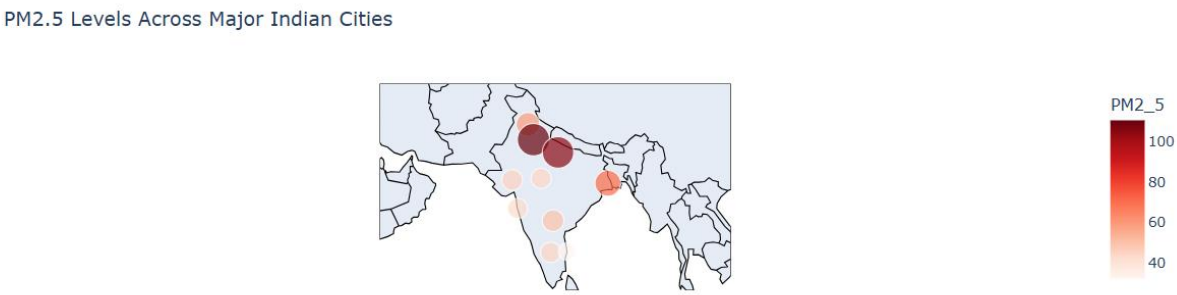
- **Map Visualization — PM2.5 Levels Across Indian Cities**

In this visualization, the goal was to map the average PM2.5 pollution levels across major Indian cities using geographic coordinates. The process began by filtering the dataset (df[df['Country'] == 'India']) to isolate only Indian cities, ensuring region-specific analysis. Next, the data was grouped by city using the groupby('City', as_index=False)['PM2_5'].mean() function, which calculated the mean PM2.5 concentration for each city—this represents the average pollution level across all recorded observations.

A separate dictionary containing the latitude and longitude of major Indian cities (city_coords) was converted into a DataFrame and merged with the PM2.5 averages using pd.merge(). This allowed each city's pollution data to be linked with its geographical coordinates, enabling precise mapping.

For visualization, the plotly.express.scatter_geo() function was used, which plots each city as a point on a world map. The color and size parameters were set to reflect PM2.5 concentrations—cities with higher pollution appear larger and more intensely colored (red). The color scale Reds enhances visual contrast, emphasizing highly polluted regions. The update_geos() function added country borders and subunit lines, while geo_scope='asia' and a centered projection focused the map on India.

This combination of filtering, grouping, merging, and geospatial plotting techniques provided an intuitive and interactive representation of air quality patterns across India, helping to identify pollution hotspots and visualize regional disparities effectively.



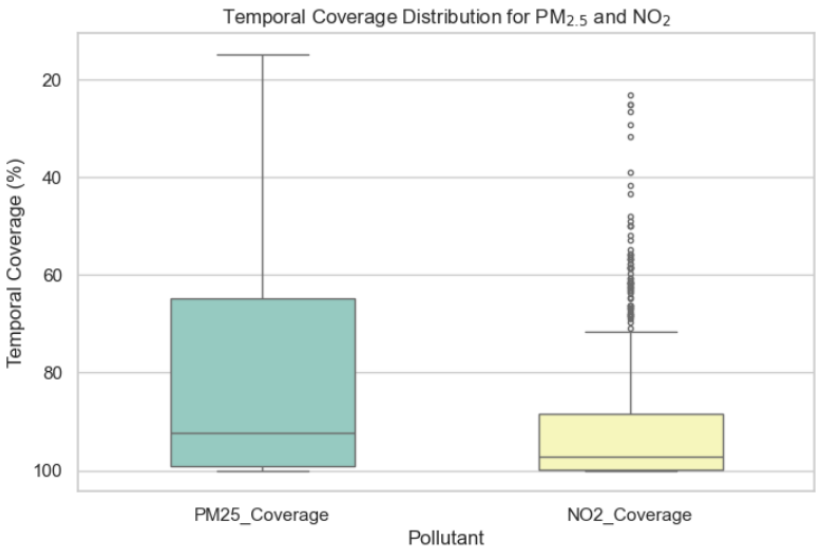
- **Temporal Coverage Distribution Analysis for $PM_{2.5}$ and NO_2**

This analysis visualizes and compares the **temporal coverage** of two major air pollutants — **$PM_{2.5}$** and **NO_2** — using a boxplot. The dataset is first transformed using the **pandas.melt()** function, converting it into a **long-form structure** where pollutant names (PM25_Coverage, NO2_Coverage) are placed under a single column called *'Pollutant'*, and their corresponding values under *'Temporal_Coverage'*. This reshaping makes the data compatible with Seaborn's plotting functions.

The visualization is created using **sns.boxplot()** from the **Seaborn** library, which displays the **median, interquartile range, and outliers** for each pollutant's temporal coverage. Parameters such as palette='Set3', width=0.5, and fliersize=3 are used for better aesthetics and clarity.

The resulting boxplot provides a concise statistical view of the **data availability and monitoring consistency** across pollutants. It helps identify variations in how frequently and consistently $PM_{2.5}$

and NO_2 measurements were recorded over time, indicating the robustness of temporal data coverage for each pollutant.

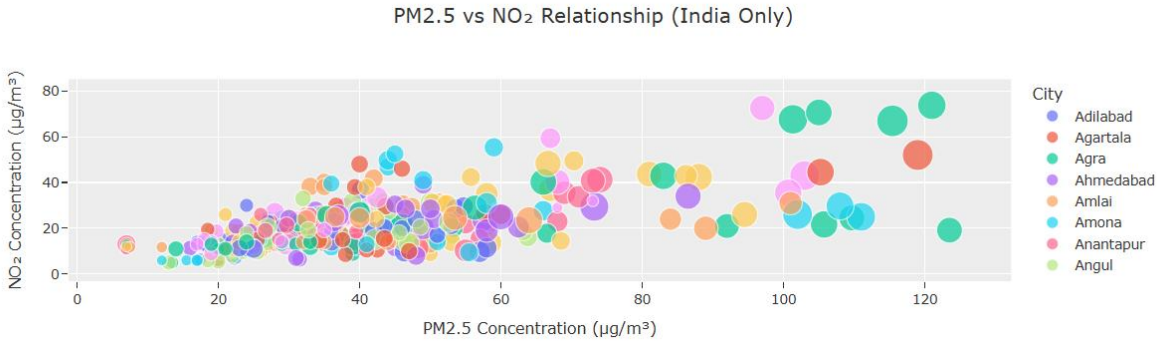


- **Interactive Scatter Plot – $PM_{2.5}$ vs NO_2 Relationship (India Only)**

This visualization explores the **relationship between $PM_{2.5}$ and NO_2 concentrations** across major Indian cities using an **interactive scatter plot** created with **Plotly Express (px.scatter)**. The dataset is first filtered for records where the country is *India*, ensuring that the analysis focuses only on national pollution trends.

In the plot, the **x-axis** represents *$PM_{2.5}$ concentrations*, the **y-axis** represents *NO_2 concentrations*, and the **size of each marker** corresponds to *PM_{10} levels*, providing a third dimension of pollution intensity. The **color parameter** differentiates cities, allowing clear comparison across locations, while **hover data** displays the *year of observation* for each data point, adding temporal context.

The **update_layout()** function is used to enhance the chart's readability by customizing titles, font size, axis labels, and visual themes. This interactive visualization enables users to **analyze pollutant relationships dynamically**, identify patterns, and spot cities with higher combined $PM_{2.5}$ and NO_2 levels—offering valuable insights into regional air quality correlations.

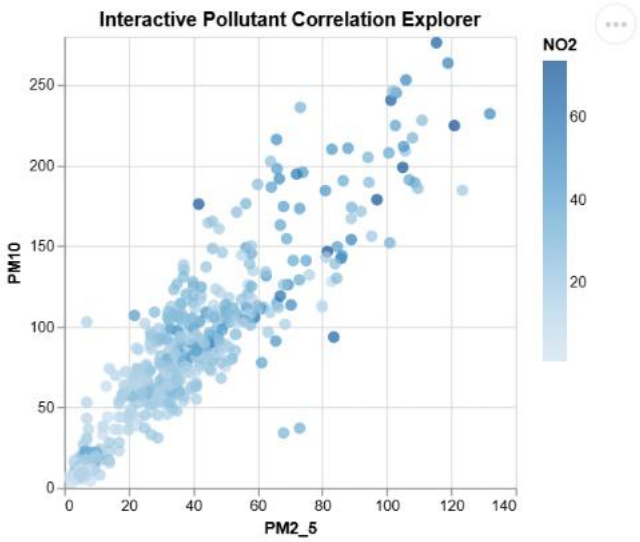


- **Interactive Pollutant Correlation Explorer (Altair Visualization)**

This visualization uses **Altair** to create an interactive scatter plot showing relationships between **$PM_{2.5}$, PM_{10}, and NO_2** levels.

- The **x-axis** represents *$PM_{2.5}$*, **y-axis** shows *PM_{10}*, and **color** indicates *NO_2* concentration.
- **Tooltips** display details like City, Year, and pollutant values for each point, while .interactive() enables zooming and panning.

The chart is saved as an **HTML file** for easy sharing and helps quickly identify **correlations and pollution patterns** across cities and years.

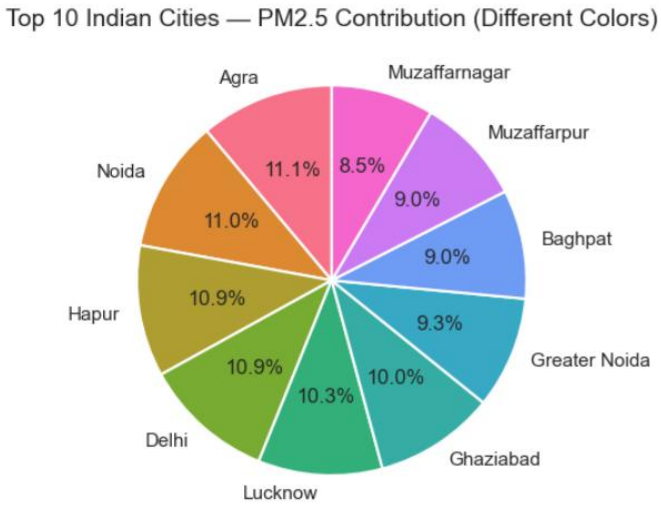


🔹 Top 10 Indian Cities — $PM_{2.5}$ Contribution (Pie Chart)

This visualization uses **Matplotlib** and **Seaborn** to create a pie chart showing the share of **$PM_{2.5}$ pollution** contributed by the top 10 Indian cities.

- The dataset was filtered for **Indian cities** only and grouped by *City* to calculate the **average $PM_{2.5}$ concentration**.
- The **top 10 cities** were selected for clear visualization, with each slice of the pie chart representing its **percentage contribution**.
- A **Seaborn color palette ('husl')** was used to give distinct colors for better readability.

This helps in identifying which Indian cities contribute the most to overall **air pollution levels** based on $PM_{2.5}$ concentration.



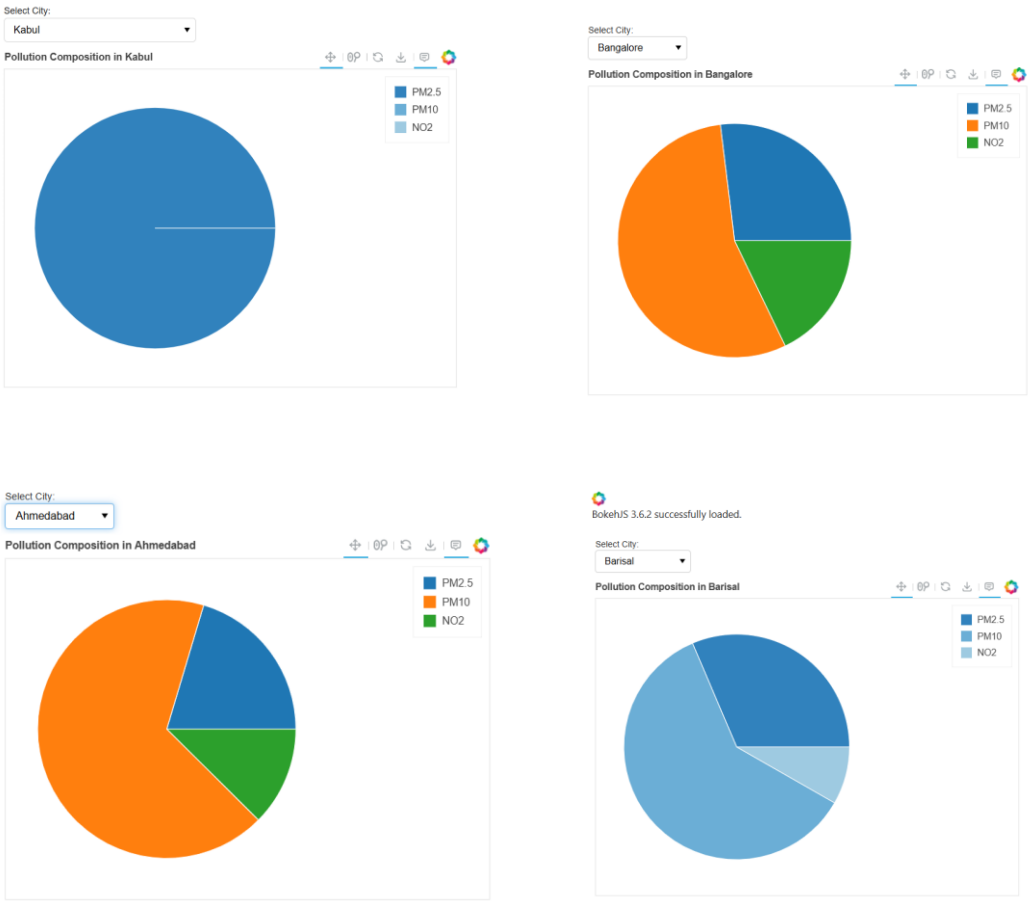
🔹 Interactive Pollution Composition (Bokeh Visualization)

This visualization uses **Bokeh**, a powerful Python library for creating interactive web-based graphics. The code dynamically visualizes **pollution composition ($PM_{2.5}$, PM_{10}, and NO_2)** for different cities using an **interactive pie chart**.

- The dataset columns are first **normalized** to remove special characters and spaces for consistency.
- The code automatically **detects the pollutant columns** ($PM_{2.5}$, PM_{10}, and NO_2) and calculates their **mean concentration values** for the selected city.
- These values are then **converted into angles** to represent proportional slices of a **pie chart**, with colors assigned from Bokeh's Category20c palette.

- A **dropdown menu** (Select) allows users to choose different cities, and a **CustomJS callback** dynamically updates the chart in real time without re-running the Python kernel.

This interactive visualization provides a clear comparison of the relative contributions of key pollutants for any selected city, enhancing the interpretability of air quality data.



Interactive City AQI Dashboard (Bokeh 3.4+)

This section presents an **interactive dashboard** for visualizing the **Air Quality Index (AQI) trends** of various Indian cities using the **Bokeh library**. The dashboard dynamically computes AQI values based on $PM_{2.5}$, PM_{10}, and NO_2 concentrations and allows users to explore pollution trends for each city through an interactive line chart.

Overview of the Code and Calculations

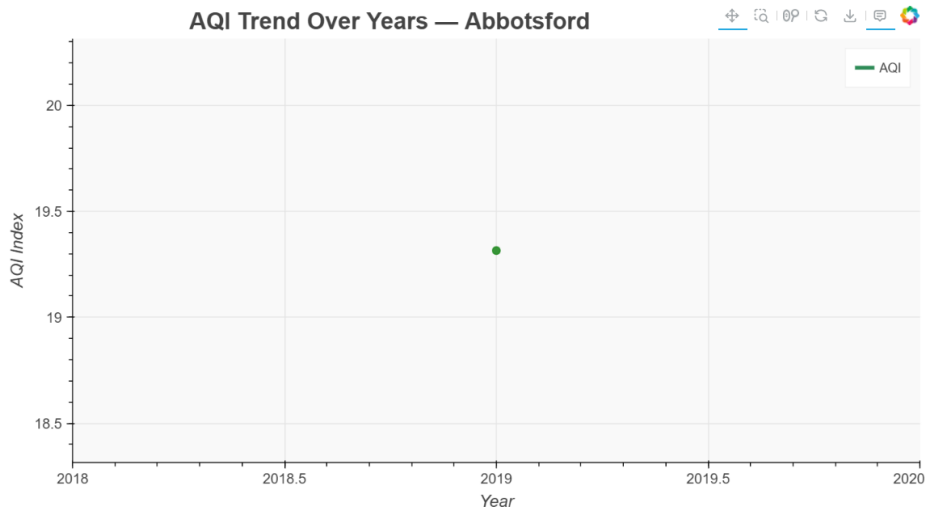
1. **Sub-Index Calculations:**
   The AQI for each pollutant is calculated using standard **Indian AQI formula ranges**.

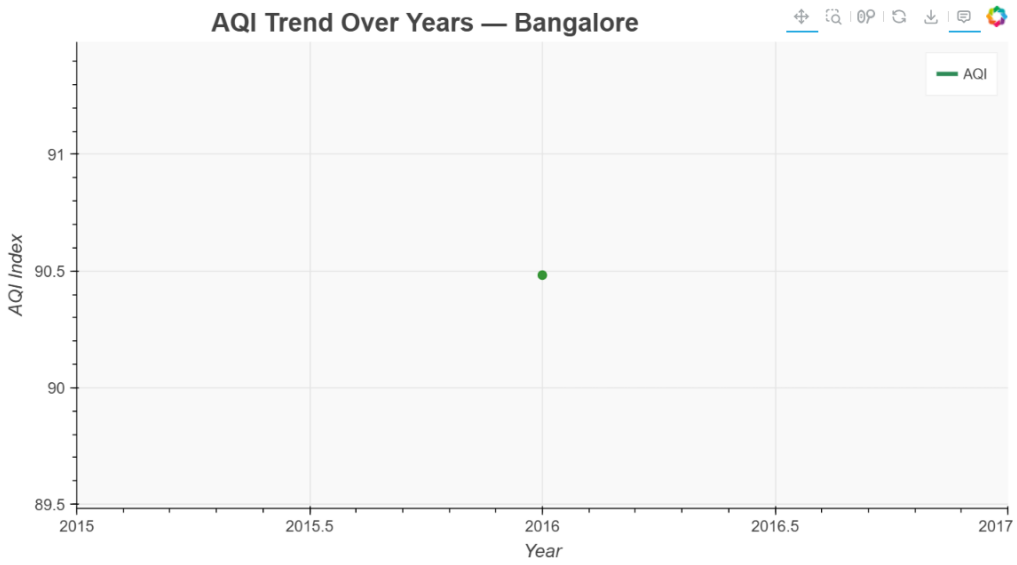
- sub_index_pm25(x), sub_index_pm10(x), and sub_index_no2(x) convert pollutant concentrations into sub-index values by applying **piecewise linear interpolation** according to CPCB (Central Pollution Control Board) guidelines.
- Each function uses conditional ranges to assign an appropriate AQI level (Good, Satisfactory, Moderate, etc.).

2. **Final AQI Computation:**
   For each record, the **maximum** of the three sub-indices ($PM_{2.5}$, PM_{10}, NO_2) is taken as the **overall AQI** — consistent with the official Indian AQI computation rule.
3. df['AQI_Index'] = df[['PM25_Index', 'PM10_Index', 'NO2_Index']].max(axis=1)
4. **City-Year Aggregation:**
   The AQI values are **grouped by city and year** using groupby() to compute the **average AQI** for each city annually. This aggregation helps visualize long-term air quality trends.
5. **Interactive Visualization Setup:**
   A **Bokeh figure** is created with a clean, modern style. The AQI data for the initially selected city is loaded using a ColumnDataSource, and a **line chart with scatter points** is plotted to show AQI variations over the years.
6. **Interactivity and Hover Tools:**
   - The **HoverTool** displays details like the year and AQI value when the user hovers over the data points.
   - A **dropdown menu** (Select) lists all cities, allowing users to switch views dynamically.
7. **JavaScript Callback:**
   A **CustomJS callback** ensures real-time updates on the chart when a different city is selected — **no reloading or re-running Python code** is needed. It filters and reassigns city data, updates the AQI line, and changes the title dynamically.
8. **Final Layout and Output:**
   The dropdown and the plot are combined into a single layout using column(), and the dashboard is displayed using show(layout).

Select City:

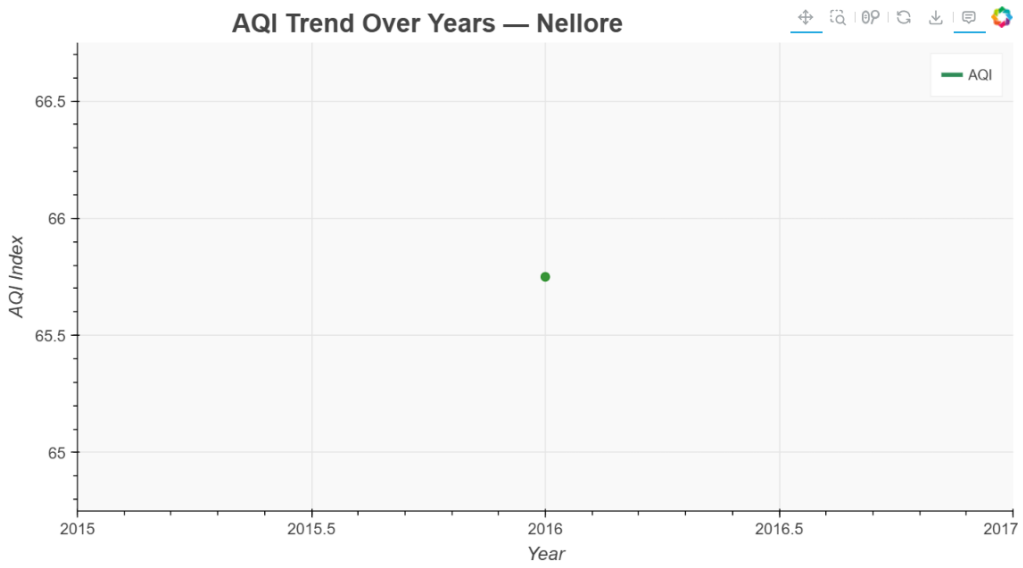
Abbotsford





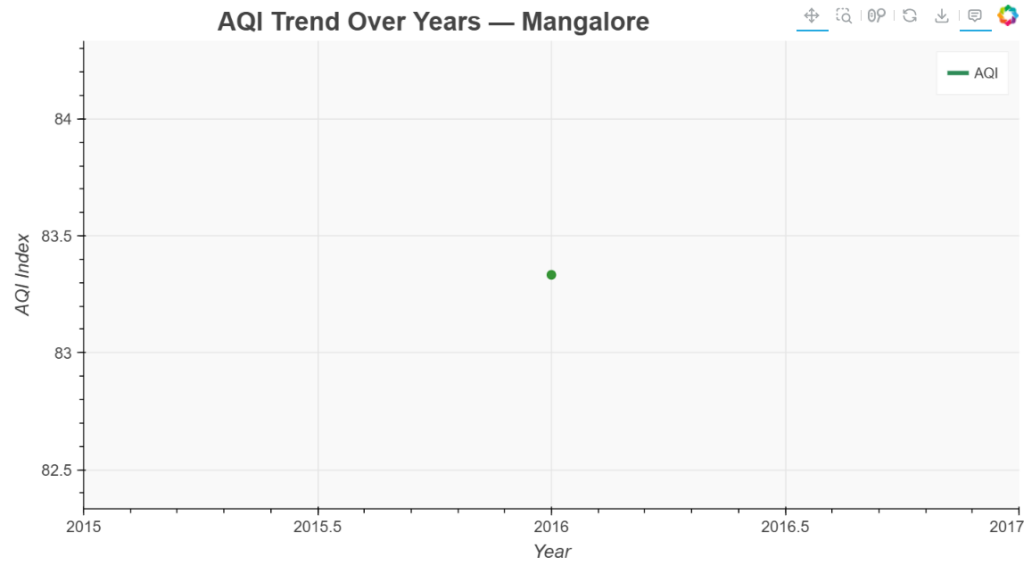
Select City:

Nellore



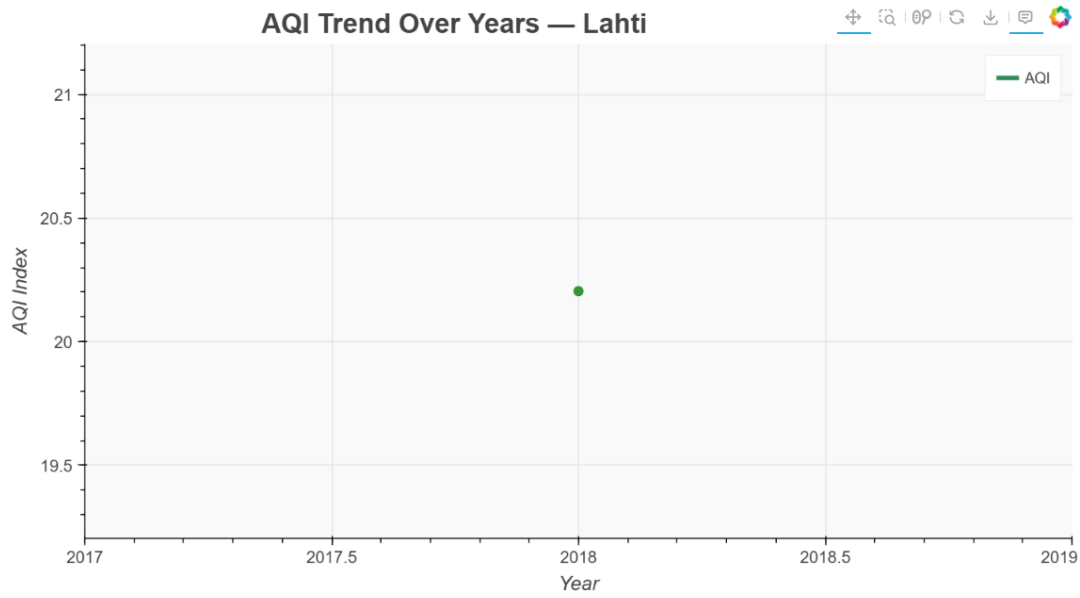
Select City:

Mangalore



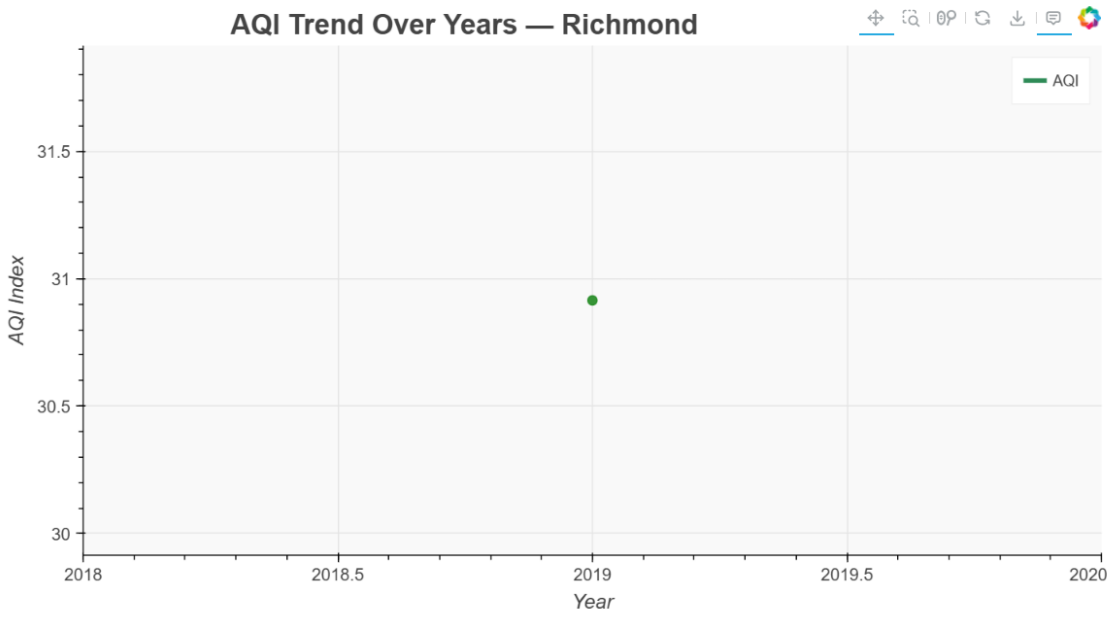
Select City:

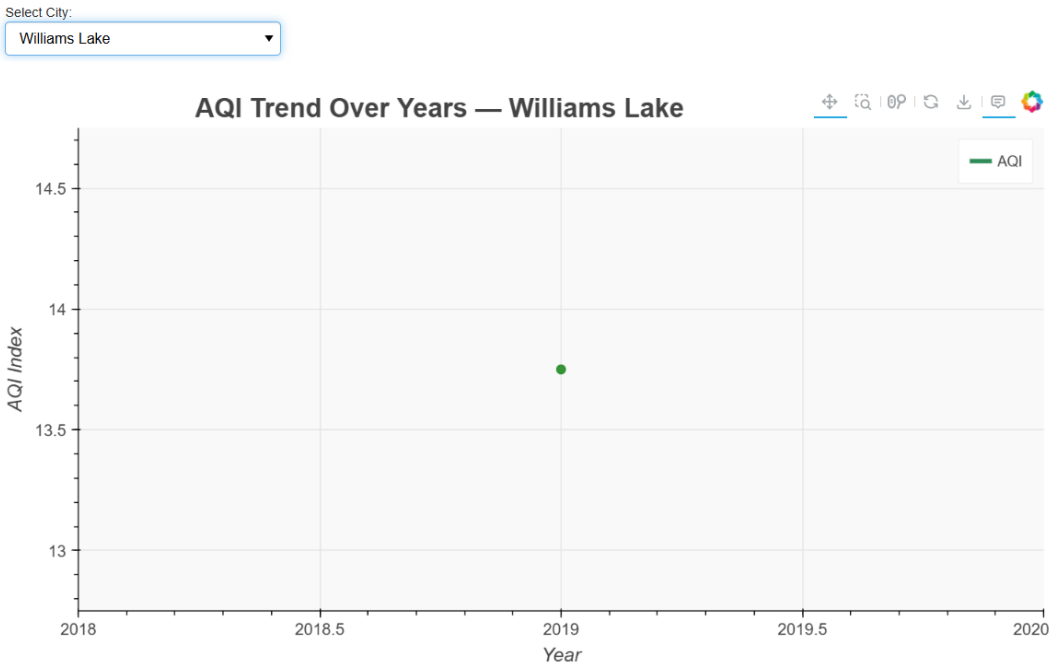
Lahti



Select City:

Richmond





Summary

This interactive AQI dashboard combines **data analytics and visualization** to provide a dynamic exploration of urban air quality. By integrating **real-time interactivity**, **custom AQI calculations**, and **visual clarity**, it helps users analyze pollution trends effectively and identify cities with deteriorating or improving air conditions over time.

Final Analysis Results & Insights

Data & file

- Input file: Air_Pollution_Selected_7800_Cleaned.csv (7,800 rows).
- Processed output (with computed AQI columns): /mnt/data/Air_Pollution_With_AQI.csv.

1) Data inspection / Initial analysis

- Dataset contains **7,800 records** and ~11 columns after normalization (Country, City, Year, PM2_5, PM10, NO2, temporal-coverage columns, AQI sub-indices, AQI_Index).

- Several coverage and pollutant fields had missing values (e.g., PM2.5: ~4,139 nulls in raw file prior to cleaning; after conversion many rows remained usable for city-level analysis).
- Action taken: numeric conversion, duplicate handling, and AQI sub-index computation.

2) Summary statistics (numerical)

- Descriptive stats computed for PM2.5, PM10, NO_2 and temporal coverage columns (count, mean, std, min, 25%, 50% (median), 75%, max).
- Temporal coverage medians are high (median **~97%** for PM2.5 coverage, **95%** for PM10 coverage, **96%** for NO_2 coverage for available records) — monitoring completeness is generally good where data exists.

3) AQI calculation (manual)

- AQI sub-indices computed using CPCB/Indian piecewise breakpoints for **PM2.5**, **PM10**, and **NO_2**.
- Final AQI per record = **max(PM2.5_index, PM10_index, NO2_index)**.
- Processed CSV with AQI saved to /mnt/data/Air_Pollution_With_AQI.csv.

4) PM2.5 distribution (histogram)

- Distribution is **right-skewed**, with many city-year records in high PM2.5 ranges; a substantial portion far exceeds WHO guideline values — consistent with high urban exposures.

5) Top 10 Most Polluted Cities (by average PM2.5)

Top 10 (city — mean PM2.5; PM10; NO_2 where available):

1. **Bamenda** — PM2.5 = **132.00** µg/m³, PM10 = 141.00
2. **Kabul** — PM2.5 = **119.77** µg/m³
3. **Agra** — PM2.5 = **112.97** µg/m³, PM10 ≈ 194.33, NO_2 ≈ 22.01
4. **Noida** — PM2.5 = **112.12** µg/m³, PM10 ≈ 216.69, NO_2 ≈ 37.43
5. **Hapur** — PM2.5 = **111.00** µg/m³
6. **Delhi** — PM2.5 = **110.69** µg/m³, PM10 = 235.00, NO_2 ≈ 62.65
7. **Rawalpindi** — PM2.5 = **107.00** µg/m³ (very high PM10 recorded)
8. **Lucknow** — PM2.5 = **105.00** µg/m³
9. **Ghaziabad** — PM2.5 = **101.84** µg/m³
10. **Lahore** — PM2.5 ≈ **95.94** µg/m³

Insight: major hotspots include cities in South Asia and parts of Africa / Central Asia; Indian NCR cities (Agra, Noida, Delhi, Ghaziabad, Lucknow) consistently rank high.

6) Highest / Lowest PM2.5 city

- **Highest (mean PM2.5): Bamenda** (132 µg/m³).
- **Lowest (mean PM2.5): Suðurnesjabær** (cleanest city by mean PM2.5 in this dataset).

7) PM2.5 vs PM10 correlation (heatmap)

- **Correlation PM2.5 ↔ PM10: 0.90** (strong positive).
  Insight: PM2.5 and PM10 move together, implying shared sources (traffic, dust, combustion) and that coarse and fine particulate problems co-occur in many cities.

8) PM10 and NO_2 comparison across cities (bar charts)

- PM10 averages are often high where PM2.5 is high (e.g., Delhi, Noida, Ghaziabad).
  Insight: Many cities show both coarse (PM10) and fine (PM2.5) particulate issues; NO_2 is especially high in traffic-dense cities (e.g., Delhi).

9) City-wise NO_2 (Top 10)

- Cities with the highest NO_2 are typically **traffic-intensive urban centers** (Delhi notable with NO_2 ≈ 62.6 µg/m³ mean).
  Insight: NO_2 highlights combustion/vehicle impacts; cities with high NO_2 often align with high PM from traffic sources.

10) Yearly pollutant trends (line chart)

- **Year with highest mean PM2.5: 2021**.
- **Year with lowest mean PM2.5: 2018**.
  Insight: Multi-year fluctuations exist — 2018 shows lower averages in this dataset; 2021 shows peak exposures. Use caution: year-to-year averages can be affected by sample size and station coverage.

11) Cleanest cities (lowest PM2.5)

- The top “cleanest” cities have mean PM2.5 far below the hotspots (e.g., several remote or small cities). Example from dataset: **Suðurnesjabær** is the lowest-PM2.5 city.
  Insight: Strong urban vs. rural contrast — policy focus should be on urban hotspots.

12) Stacked bar — pollutant composition by city

- Stacked bars show pollutant mix per city: some cities dominated by PM2.5+PM10, others have relatively higher NO_2 contribution.
  Insight: Composition view helps identify whether PM or gaseous emissions dominate a given city's pollution profile.

13) Pie chart — top 10 Indian cities PM2.5 contribution

- Top Indian cities (Agra, Noida, Delhi, Ghaziabad, Lucknow, etc.) form a substantial share of overall PM2.5 burden among Indian locations in the dataset.
  Insight: A small set of cities contribute a large portion of total PM2.5 exposure in the national subset.

14) Violin plot — year-wise PM2.5 distributions

- Year-wise violins show spread and quartiles; 2018 displays narrower/lower distribution, while 2021 shows higher median and larger spread.
  Insight: Violins reveal variability within years (not just the mean) — helpful to spot whether higher averages come from many moderately high cities or a few extreme outliers.

15) Map visualization (Plotly) — PM2.5 across Indian cities

- Interactive map shows larger, redder bubbles for the most polluted cities (Delhi, Noida, Agra, etc.).
  Insight: Spatial visualization highlights regional clusters (NCR hotspot) and helps prioritize geographic interventions.

16) Temporal coverage (boxplot)

- Median temporal coverage is high where present: **PM2.5 median ~97%**, **PM10 median ~95%**, **NO_2 median ~96%** (counts vary by pollutant).
  Insight: Where monitoring exists, data completeness is generally good — missingness is concentrated in certain records/cities.

17) Interactive scatter (Plotly) & Altair chart

- Scatter (PM2.5 vs NO_2) with point size = PM10 shows cities where all three pollutants co-exist at high levels (clusters around high PM2.5 and high NO_2).
  Insight: These interactive charts let you inspect city-year pairs and identify combined high-exposure situations.

18) Bokeh pie + AQI dashboard (interactive)

- Bokeh dashboard computes AQI per city-year and displays trends. Top AQI cities by mean AQI in this dataset:
  - **Rawalpindi** — mean AQI ≈ **422.5**

    - **Peshawar** — mean AQI ≈ **351.2**
    - **Bamenda** — mean AQI ≈ **309.2**
- **Insight:** AQI highlights different hotspots than raw PM2.5 ranking in some cases (very high PM10 or NO_2 can push AQI). The city dropdown + time series is helpful to examine improvements/declines over years.

Quick actionable recommendations

- Prioritize interventions in the NCR cluster (Delhi, Noida, Ghaziabad, Agra, Lucknow) where PM2.5 and PM10 are consistently high.
- Use traffic/vehicle emission controls in cities with high NO_2 (e.g., Delhi).
- Maintain and expand monitoring coverage in cities with low temporal coverage to improve trend reliability.
- Use AQI dashboard to communicate public health risk and track effectiveness of policies year to year.