Project Report For CS661: BIG DATA VISUAL ANALYTICS
2023-2024 Semester II

# Visual Analytics Of Air Pollutants And Pollution-related Health Impact In India

Team members:-
Abhishek Srivastava, Anay Sinhal, Arth Banka, Happy Pachori, Pankaj Nirmal, Riktesh Singh, Saransh Shivhare, Tanmay Chhapola, Vaibhav Nalotia

## 1. Introduction

Air pollution stands as a formidable health menace in India, propelled by the rapid pace of urbanization and industrial expansion. Recognizing the urgent need for intervention, we propose a dynamic solution through a visual analytics project aimed at dissecting air quality data and unraveling its profound impact on respiratory health. Leveraging a robust dataset spanning from 2021 to 2023 sourced from Kaggle, encompassing pollutants such as ozone, CO, SO2, NO2, PM10, and PM2.5, our initiative embarks on a comprehensive exploration.

At its core, our project delves into three pivotal dimensions: temporal trends, spatial distribution, and health ramifications. By scrutinizing temporal shifts, we aim to elucidate how air quality undergoes metamorphosis over time, shedding light on emerging patterns and identifying critical junctures. Through spatial analysis, we endeavor to map the intricate tapestry of pollution across diverse regions, delineating hotspots and assessing disparities. Crucially, we aspire to unravel the tangible health impacts inflicted by air pollution, particularly on respiratory well-being, underscoring the gravity of the issue.

Our ultimate goal is to craft an immersive and user-friendly web-based visualization platform, empowering stakeholders to navigate and interpret AQI data with precision and ease. By illuminating the multifaceted facets of air quality dynamics in India, our endeavor strives to underscore the imperative of concerted efforts to combat air pollution, safeguarding public health, and fostering a sustainable future.

## 2. Tasks

### (a) Visualization of pollutants

- Comparing data across different states:
  - Essential to understand spatial variations in pollutant levels.
  - Identify regions with higher pollution burdens.
  - Use visualizations such as choropleth maps or bar charts.
  - Enable policymakers and stakeholders to prioritize interventions effectively.
- Comparing pollutant data across different timelines:
  - Crucial for detecting temporal trends and seasonal variations in air quality.
  - Utilize time series plots and Parallel Coordinate plot.
  - Visualize how pollutant concentrations fluctuate over days, months, or years.
  - Assess the effectiveness of interventions and track progress towards air quality goals.
- Combining analysis of multiple pollutants:
  - Essential for understanding the overall composition of air pollution and its health implications.
  - Use visualizations that overlay multiple pollutants on the same graph or display their correlations.
  - Examples include scatter plots or correlation matrices.
  - Identify common sources or synergistic effects between different pollutants.

### (b) Trend analysis and comparison

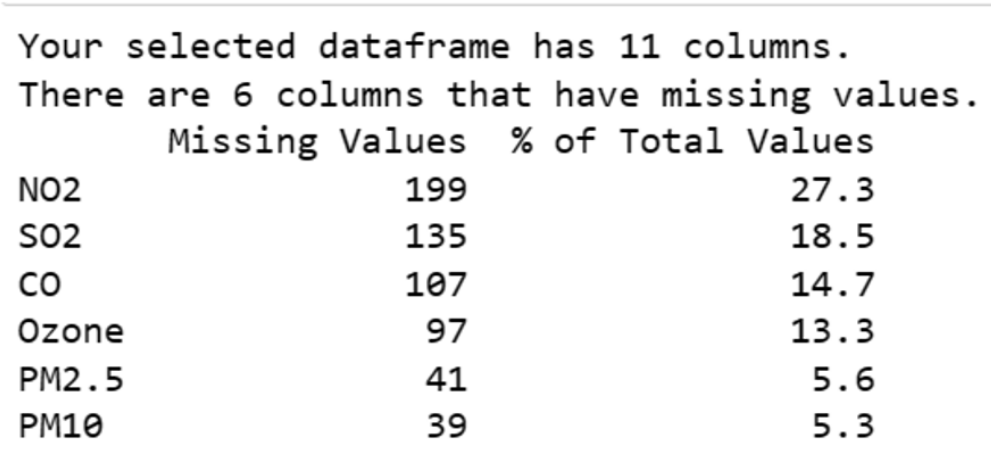
- Comparison of pollutants at different times:
  - Analyze how pollutant levels change over various time intervals (e.g., days, months, years).
  - Detect seasonal variations, long-term trends, and the effectiveness of pollution control measures.
- Comparison of pollutant levels across different states:
  - Understand spatial variations in air quality.
  - Identify regions with higher pollution burdens.
  - Use visualizations such as line graphs or stacked bar charts.
- Relative comparisons:
  - Compare pollutant levels relative to a reference point or baseline.
  - Assess the magnitude of changes and progress towards air quality goals.
- Frequency analysis:
  - Examine the occurrence and distribution of pollutant concentrations over time.

  - Gain insights into patterns of pollution episodes and their potential causes.
- Time series analysis techniques:
  - Utilize decomposition, smoothing, and forecasting methods.
  - Extract meaningful patterns from temporal data.
  - Make informed decisions about pollution management strategies.

# 3. Solution

## Data Preprocessing:

The data was first converted from xlxs format to the csv format for data analysis. The pandas library was used to load the dataset for analysis. We proceeded with the analysis of the dataset by checking for its completeness. Finding the number or percentage of missing values in each column of the dataset is the first stage. This will provide insight into the distribution of values that are missing.



```
Your selected dataframe has 11 columns.
There are 6 columns that have missing values.
       Missing Values  % of Total Values
NO2               199               27.3
SO2               135               18.5
CO                107               14.7
Ozone              97               13.3
PM2.5              41                5.6
PM10               39                5.3
```

Figure 1: A sample table for showing the data completeness of pollutants in Jammu and Kashmir

The dataset had to be pre-processed differently for the data of all the states. This was done because analysis showed that different states had different percentages of missing values. The reasons for missing data also seemed to differ across states. This is consistent with real-world logic as well, given that each state will differ in how they collect pollutant data, some states will prioritise this task more, some states will have higher expenditure and better equipment for collecting pollutant data.
We utilised the Missingno library to evaluate the data's missingness visually. It is a

package for missing value graphical analysis. The missingness data for each state was also visualised graphically through a bar graph as shown below.

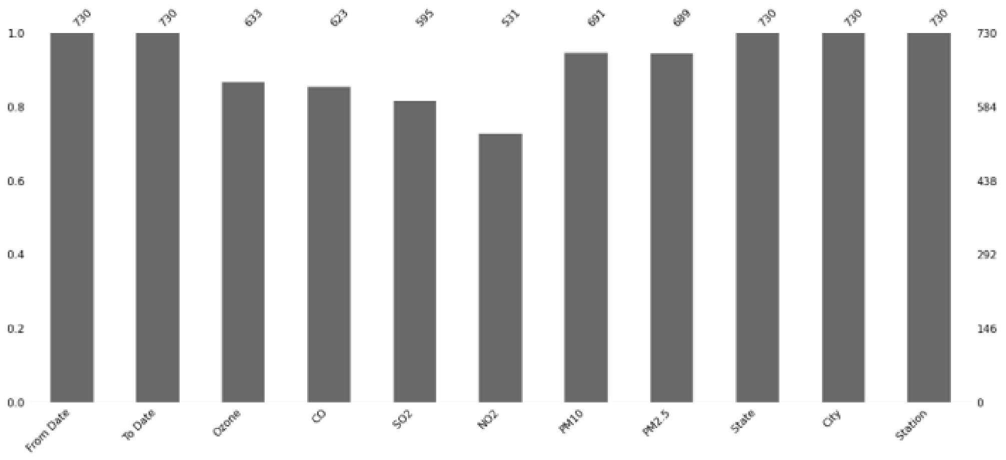


Figure 2: A bar graph for showing the data completeness of pollutants in Jammu and Kashmir

We then utilized the library to identify potentially correlated features in the dataset. we employed heatmaps, matrix plots, and dendrograms to preprocess and identify these correlations in our air pollutant data.

## Heatmap Analysis:

We utilized the missingno.heatmap() function to create a visual representation of missing values across all features (pollutants) in our air quality data. This heatmap provided insights into:

Identifying groups of pollutants with similar missingness patterns, suggesting potential causes for missing data (e.g., instrument malfunction affecting multiple pollutants).

Focusing our data cleaning efforts on specific features or combinations of features that exhibited high missingness correlations.

Blue squares indicate strong positive correlations, meaning if one pollutant has missing values, the other is likely missing values as well.

White squares indicate negative correlations, suggesting missingness in one pollutant might not necessarily correspond to missingness in another.

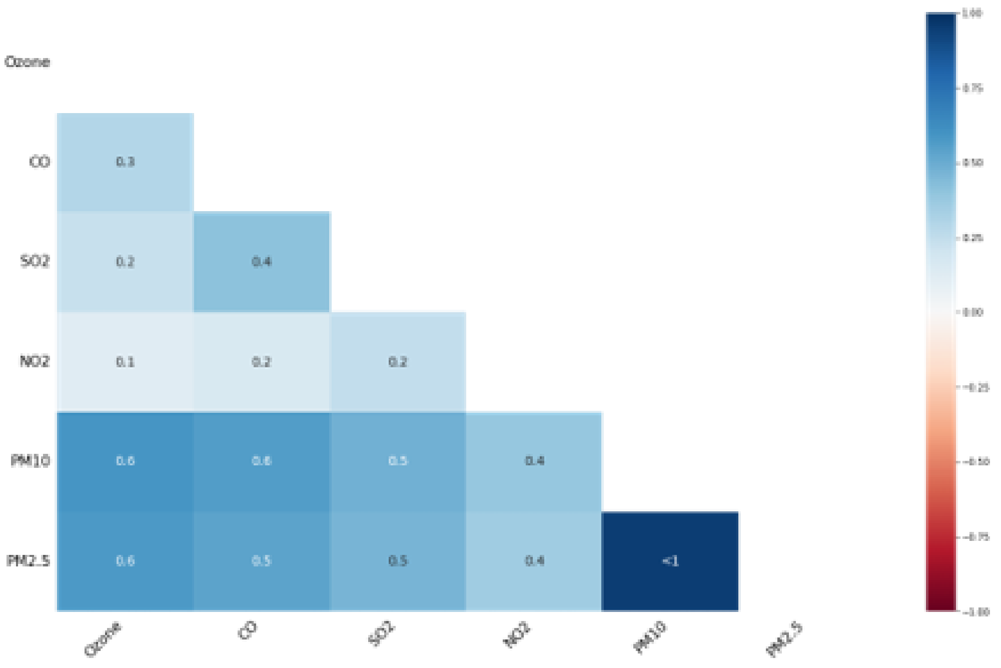


Figure 3: A heatmap for showing the data missingness correlations of pollutants in Jammu and Kashmir. Here we see high correlation for PM 2.5 and PM 10. This makes real-world logical sense as well since they are likely to be detected by similar equipment. So if data is not collected then it won't be collected for either of them.

## Dendrogram Analysis:

A dendrogram visually represents the hierarchical clustering of features based on their missingness patterns.

Features with similar missingness patterns are grouped closer together in the dendrogram.

The dendrogram can be used to:

Identify distinct clusters of features with similar missingness behavior, potentially revealing underlying factors influencing missingness (e.g., location-specific issues).

Support the findings from the heatmap by visually confirming groups of features with correlated missingness.

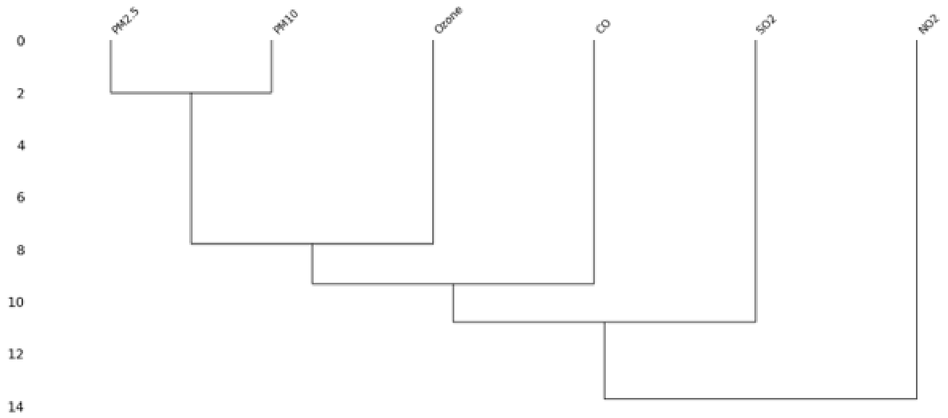


Figure 4: A dendrogram for showing the data missingness correlations of pollutants in Jammu and Kashmir. Here we see high correlation for PM 2.5 and PM 10. This is consistent with the heatmap and confirms its findings.

## Matrix Analysis:

The matrix plot complements the heatmap by displaying the actual percentage of missing values for each pollutant and the overall dataset.

This helps us quickly identify features with a high proportion of missing data.

We utilized the matrix plot for:

Prioritizing data cleaning for features with significantly higher missingness compared to others.

Deciding on appropriate strategies for handling missing values (e.g., imputation vs. removal) based on the severity of missingness.

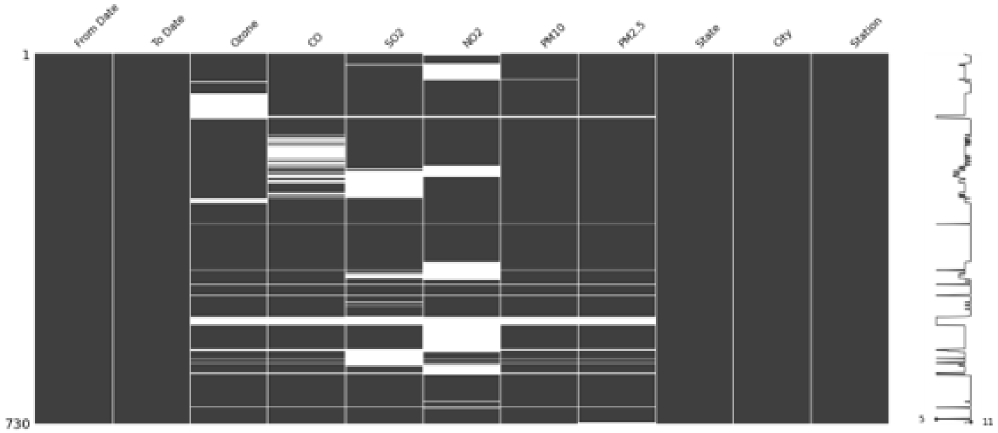


Figure 5: A matrix plot for showing the data missingness of pollutants in Jammu and Kashmir.

The following can be reasons for data missingness: Missing Completely at Random (MCAR): Imagine flipping an unbiased coin to decide whether to record a data point. Heads, you record it, tails, you skip it. In MCAR, the missing values occur completely by chance and have no relationship to the actual pollutant levels or any other information in the dataset. This is the ideal scenario, but it's not always realistic.

Missing at Random (MAR): This is more common. Here, the missing values aren't entirely random, but they can be explained by existing data. For example, a sensor malfunction might cause missing data points for a specific pollutant on a particular day. However, if we know the sensor malfunctioned, we can potentially account for it using the data from other functioning sensors or weather conditions.

Missing Not at Random (MNAR): This is the trickiest situation. MNAR occurs when the missing data points are themselves informative. For instance, air quality monitors might be less likely to record extremely high pollution levels due to safety protocols shutting them down. In this case, the missing data itself tells us something about the pollution levels (likely very high!), but we don't have the actual values. MNAR is challenging to handle because the missing information can bias our analysis.

We have used our analysis to identify reasons for missingness and taking into account other factors like how much data in a series is missing and how many features are missing simultaneously in a given row; we have used various techniques to handle the missing data.

We employed several imputation techniques to fill these gaps:

Linear Interpolation: This method estimates missing values by assuming a linear relationship between existing data points. It essentially draws a straight line between known values before and after the missing point to predict its likely value. WE used this method very rarely for extremely small gaps in data.

K-Nearest Neighbors (KNN) Imputation: We used this approach for dealing with medium sized chunks of missing data values across multiple features. This approach identifies the k data points (neighbors) most similar to the one with a missing value. The missing value is then imputed based on the average value of those neighbors' corresponding feature or by weighting the neighbors' values based on their distance to the point with missing data. KNN imputation is useful when the relationship between features might not be strictly linear.

Multivariate Imputation by Chained Equations (MICE): We used this technique largely when there were large chunks of missing data. This is a more sophisticated technique that tackles missing values across multiple features simultaneously. It works by creating a series of regression models, each predicting one feature with missing values based on the other features (both with and without missing values) in the dataset. MICE iterates through these models, refining the imputed values with each pass. By considering relationships between features, MICE can lead to more accurate imputations compared to simpler methods.

By employing these techniques, we were able to minimize the impact of missing data and create a more complete picture of air quality patterns.
Beyond these corrections, we also standardized the data format of the collection of the data since that made it easier to sort data according to date and identify correlations between data missingness and date. For specific plots, we have also used the data to categorize pollution levels as 'good, 'moderate,' and 'poor' and engineered relevant new features as necessary.

## Creating the Dashboard and Integrating

We've built an interactive pollution data analysis dashboard using JavaScript and the D3.js library. D3.js is a powerful tool for creating dynamic and interactive data visualizations on the web. Here are four key reasons why we chose D3.js for our project:

1. **Interactive Visualizations:** D3.js allows us to create highly interactive visualizations, making it easier for users to explore and understand the data. With features like tooltips, zooming, and panning, users can interact with the data in real-time, gaining deeper insights into air quality dynamics.
2. **Web-Based Dashboards:** By leveraging D3.js, we've developed a web-based dashboard that provides a user-friendly interface for accessing and analyzing pollution data. This allows users to access the dashboard from any web browser, making it convenient and accessible.
3. **JavaScript Integration:** Since D3.js is a JavaScript library, it seamlessly integrates with other JavaScript frameworks and libraries, allowing for flexible and efficient development. This also enables us to incorporate additional features and functionalities into the dashboard as needed.
4. **Customizability and Flexibility:** D3.js offers extensive customization options, allowing us to tailor the visualizations and dashboard layout to meet specific project requirements. From custom color schemes to advanced data manipulation techniques, D3.js provides the flexibility needed to create a robust and adaptable dashboard for pollution data analysis.

Overall, by harnessing the capabilities of D3.js, we've created an interactive and user-friendly dashboard that empowers users to explore and analyze pollution data with ease, ultimately supporting informed decision-making and action for better environmental health outcomes.

## • Languages and Libraries used

1. **Python & JavaScript**
2. **D3.js**
3. **Pandas**

# 4. Results

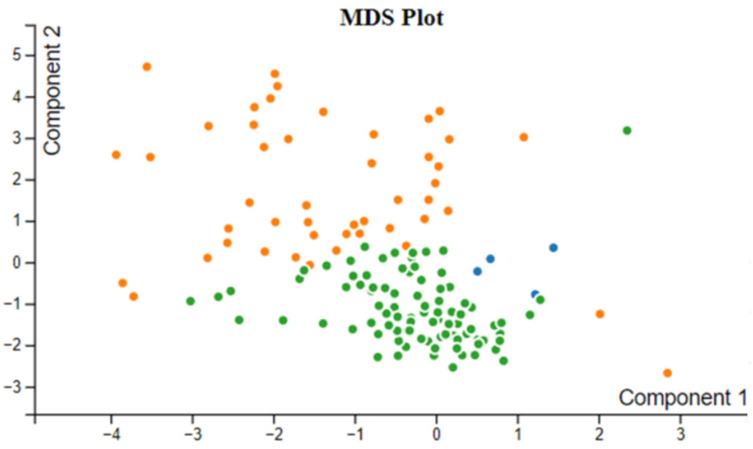
- **Homepage**

  The project homepage offers a comprehensive overview of air quality dynamics through interactive visualizations:

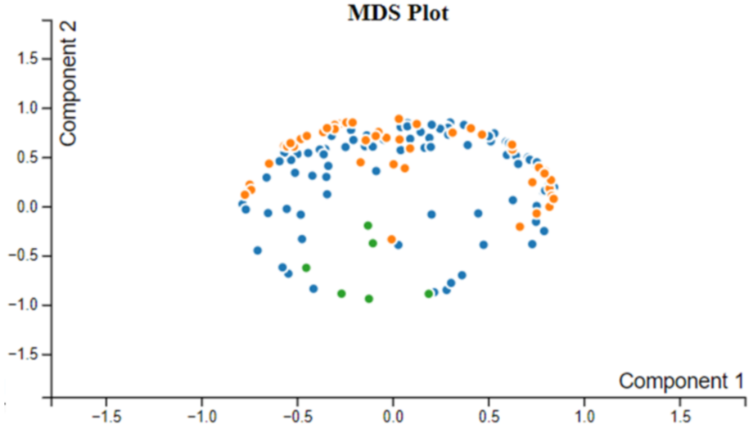
  - Pollution Index Graphs: Explore pollution indexes across all states and times for various pollutants, providing insights into spatial and temporal variations.
  - Time Series Analysis: Delve into detailed trend analysis of pollution data over time, identifying seasonal patterns and long-term trends.
  - Comparison Across Timelines: Compare pollution data across different years, enabling assessment of the impact of interventions and seasonal variations.
  - Charts for Deaths: Access charts depicting the correlation between air pollution and mortality rates, highlighting the health implications of poor air quality.

- **Multidimensional Scaling (MDS)**

  Multidimensional Scaling (MDS), pioneered by Kruskal, Shepard, and Torgerson, is a pivotal method in data science for visualizing and comparing similarities and dissimilarities in high-dimensional data. In geostatistics, MDS aids in comprehending multivariate data by reducing its dimensionality, facilitating the observation of patterns, gradients, and clusters. MDS with Euclidean distance depicts a third cluster in the plot, potentially an outlier due to lower values for component1. Meanwhile, MDS with Correlation distance uncovers two possibly overlapping clusters, indicating similar pollution profiles among monitoring stations in India. The distorted circle shape hints at potential weak or non-linear relationships between pollution measurements, despite employing correlation as the distance metric.



Euclidian Distance



Correlation Distance

- **Dimensionality Reduction Techniques**

  We have included two dimensionality Reduction Techniques i.e. PCA and t-SNE into Air Quality Index Dashboard.

  **PCA** is a widely-used linear technique that efficiently reduces the dimensionality of high-dimensional data while preserving as much variance as possible. It identifies the most significant patterns in the data by projecting it onto orthogonal principal components.

  **t-SNE** is a nonlinear technique that preserves the local and global structure of high-dimensional data in low-dimensional embeddings. It is particularly effective at capturing complex relationships and revealing underlying patterns that may not be apparent in the original data space.

  **Insights from the Plot:** By analyzing this plot, you can gain insights into the dimensionality of the air quality data. The number of principal components needed to explain a sufficient amount of variance (indicated by the elbow point) can help determine the underlying structure of the data.

  For instance, if a relatively small number of PCs (e.g., 2 or 3) can explain a high percentage of the variance (e.g., over 90%), it suggests that the relationships between SO2, O3, NO2, and CO can be effectively captured with a lower-dimensional representation. This could be helpful for further analysis or visualization purposes.

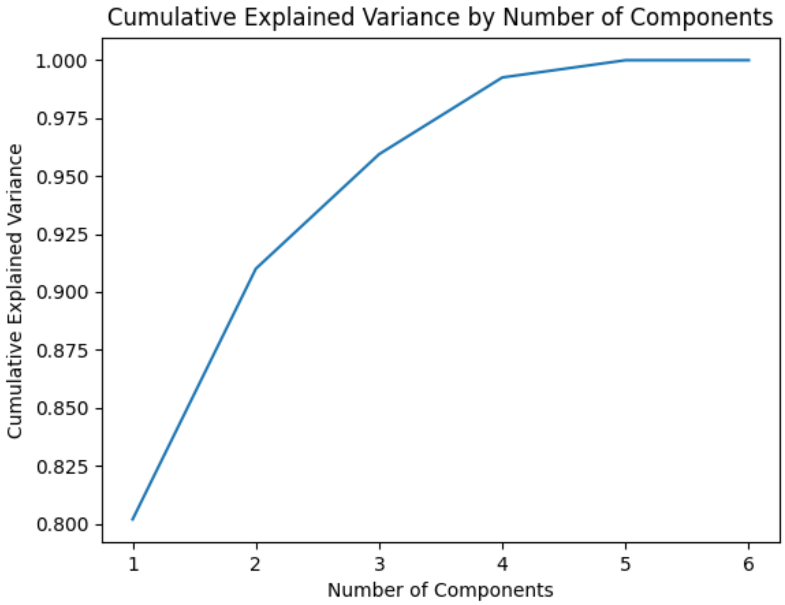


Figure 6: Cumulative Variance wrt number of components in PCA

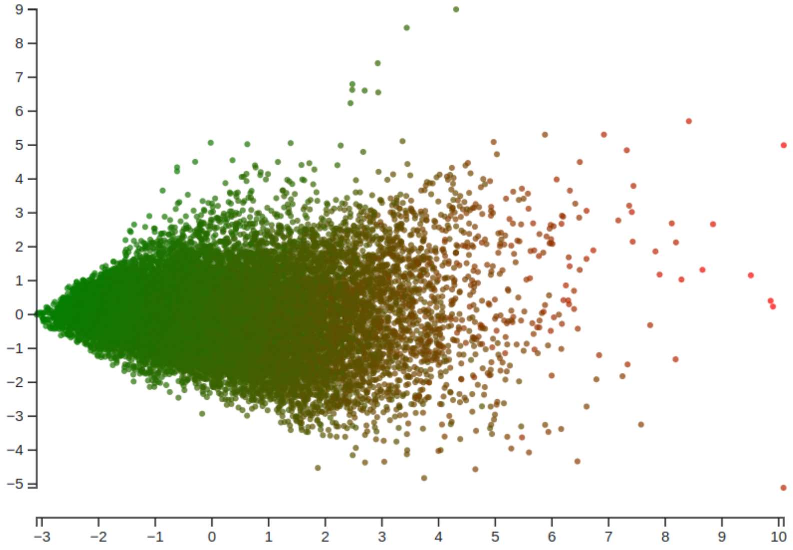


Figure 7: PCA Plot from pollutants data of Madhya Pradesh

This Principal Component Analysis (PCA) plot depicts the variation of air quality data across different locations. Each dot represents a measurement, with colours indicating the Air Quality Index (AQI), Green dot for favourable air quality, and Red for unfavourable air quality. Dates with similar air quality conditions tend to cluster together in the plot. The direction and length of the axes represent the factors that most influence the variation in air quality. A general pattern is observed over the data of all states, as we move away from the vertical component the air quality starts deteriorating.

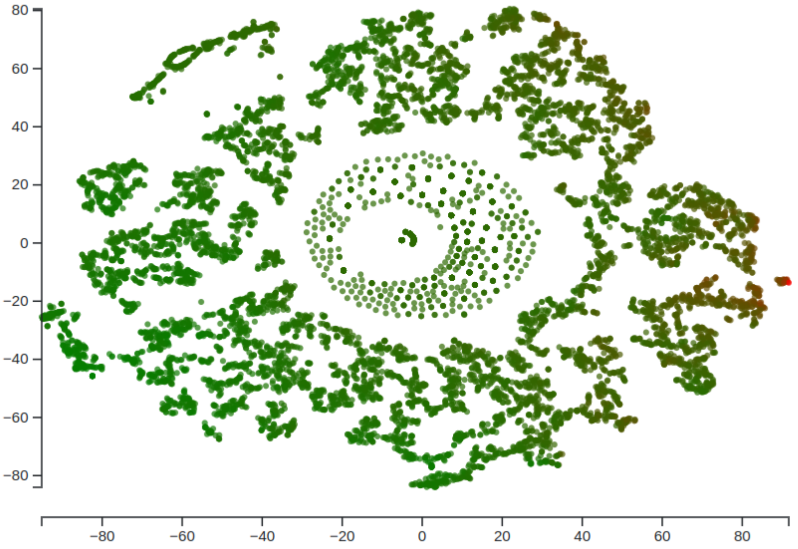
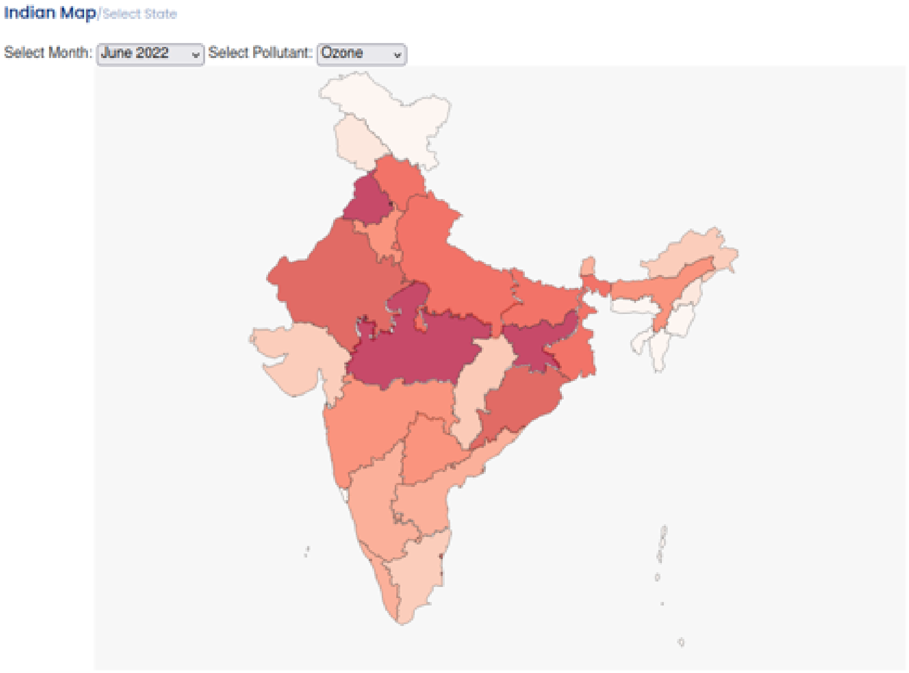


Figure 8: t-SNE Plot from pollutants data of Madhya Pradesh

There are a few data points scattered far from the main clusters. These outliers could represent dates with unique air quality conditions or potentially indicate errors in the data collection process. Further investigation into these outliers might be necessary. While it's difficult to say definitively from the t-SNE plot alone, the spread of the data points suggests that no single pollutant is solely responsible for air quality variations. There are likely complex interactions between the four pollutants measured (ozone, SO2, NO2, and CO) that influence the overall AQI.

- **Visualization of pollutants on Indian Map**

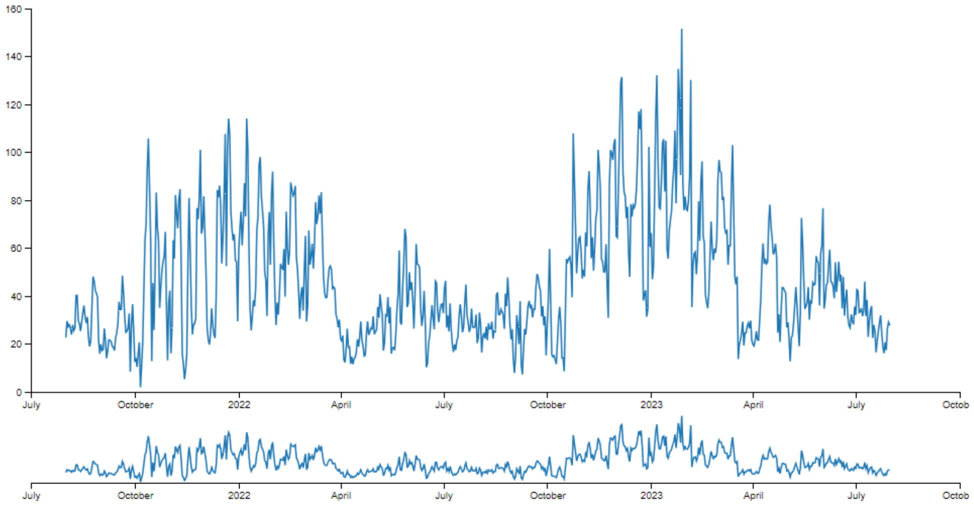
  The dataset was preprocessed to generate pollutant-wise and month-wise average JSON data. An interactive map was developed to visualize air pollution levels across Indian states based on different pollutants. Upon selecting a state and time, the map dynamically updates to display color-coded state boundaries indicating pollution levels. Further, interactive plots including time series, parallel coordinates, and PCA plots are displayed for the selected state, showcasing the integrated plot's responsiveness. Additionally, clicking on any state on the map reveals the state name and the particular pollutant value for the selected month and pollutant.



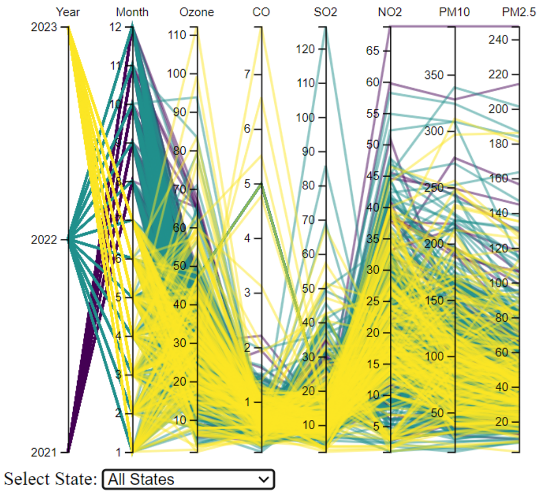
Select Month: June 2022 Select Pollutant: Ozone

- **Time-series Analysis**

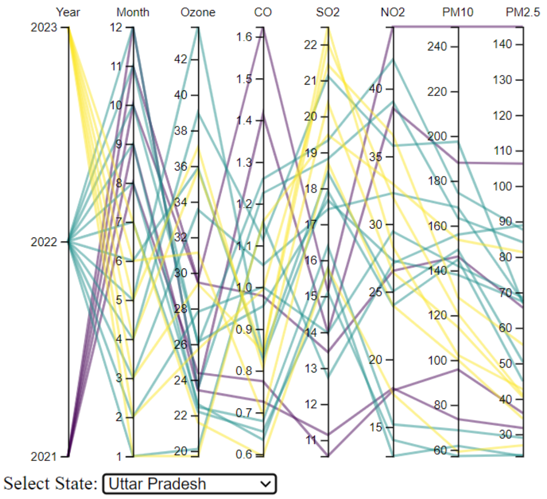
  The time series analysis graph of pollution data provides insights into temporal trends and patterns. By visualizing pollutant levels over time, it allows for the identification of seasonal variations, long-term trends, and the effectiveness of pollution control measures. Users can explore how pollution levels fluctuate across different time intervals, aiding in informed decision-making and policy planning for mitigating air pollution's adverse effects on public health and the environment.



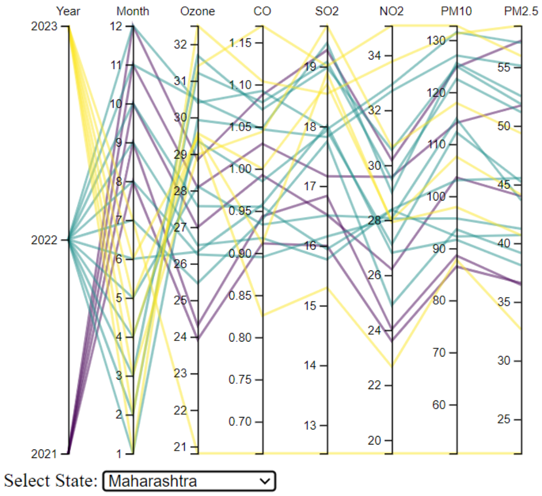
- **Pollution index comparison**
  A parallel coordinate plot visually represents multivariate data, where each observation is depicted as a line crossing parallel axes corresponding to specific variables. In our case, it displays pollutant levels in a selected state for a given year and month. This method efficiently illustrates relationships between multiple variables in a 2D space, offering insights into trends and correlation between the variables. Initially, the plot shows combined data from all states, and upon selection of a state, updating to focus only on the chosen state's pollutant levels. Distinct relationships between variables are depicted through converging, diverging, or parallel lines within the plot.



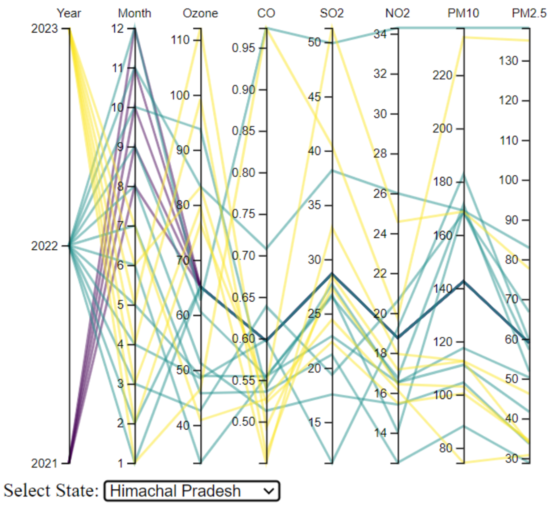
Select State: All States



Select State: Uttar Pradesh



Select State: Maharashtra



Select State: Himachal Pradesh

- ### Distribution of deaths

  Utilizing D3.js, a straightforward bar chart was constructed to illustrate the distribution of deaths attributed to air pollution in India spanning three decades. Scale functions and alignments were applied to accurately position the bars within the SVG container, ensuring clarity and readability of the visualization. This visualization offers a concise yet informative representation of the impact of air pollution on mortality rates over time, aiding in raising awareness and informing public health policies for mitigating its adverse effects.

# 5. Conclusion

In summary, our project focused on analyzing air pollution data in India from 2021 to 2023. Through detailed data preprocessing and the development of an interactive dashboard using D3.js and JavaScript, we aimed to provide a user-friendly platform for exploring and understanding air quality trends.

Our visualizations, including pollution index graphs, time series analyses, and interactive maps, highlighted important patterns in air pollution across different regions and over time. These insights can help inform policy decisions aimed at improving air quality and public health.

By presenting our findings in an accessible and interactive format, we hope to contribute to the ongoing efforts to combat air pollution and promote environmental sustainability in India.

**Link to source code** Github Repository

## References

- D3 documentation
- Using T-SNE in Python to Visualize High-Dimensional Data Sets
- Kaggle Dataset-1
- Kaggle Dataset-2