

Is Delhi's air killing productivity & lives silently?

This project analyzes Delhi's air quality using Python, SQL, Excel, and Power BI.

Paras Kumar

A Glimpse into the Analysis Workflow

This project rigorously explores Delhi's air quality, moving from raw data to actionable insights.

1

Data Acquisition

Gathering raw air quality data from reliable sources.

2

Data Cleaning

Preparing and refining data for accuracy and consistency.

3

Insight Extraction

Utilizing powerful query languages to uncover hidden trends.

4

Visualization (in Progress)

Creating interactive dashboards for impactful presentation.

Is Delhi's Air Really That Bad?

Let the data speak.

By Paras Kumar

The Problem: A Daily Reality in Delhi

Air pollution in Delhi isn't just a headline—it's a harsh daily reality. This analysis dives into data to uncover patterns, severity, and insights behind the city's ongoing air quality crisis.

My Tech Stack:

The Tech Stack I used for this project is mentioned below.

Having different tools for different purposes playing a crucial role in data cleaning, optimization and querying



- **Python:** Used for cleaning, analysing and modelling the data using pandas library
- **SQL:** Employed for efficient data extraction, querying specific trends, and categorizing pollution levels within the dataset.
- **Excel:** Used for cross-verification of aggregated data and performing quick sanity checks on initial findings.

Step 1: Cleaning the Mess

Since, I had a abundance of data ranging from all the cities in India, the primary step was to segregate and filter the city only to "DELHI"

WHAT I DID?

Like any dataset, this dataset comprised of null, empty values. Subsequently, The primary motive was to clean it.

Data Cleaning Process:

- Applied `dropna()` to remove any inconsistency by removing the missing values
- There existed potential outliers that skewed the data, so i cleaned it up.

```
#handling the missing values using fillna() and dropna()
import pandas as pd
cleaned_delhi_aqi = pd.read_csv("Delhi_aqi_project/city_day.csv/03.delhi_aqi_datetime_converted.csv")

# print(cleaned_delhi_aqi)

#to find the sum of null values in each column #####
# print(cleaned_delhi_aqi.isnull().sum())

#to see if a specific column is null
# null_values_delhi_aqi = cleaned_delhi_aqi[cleaned_delhi_aqi['AQI'].isnull()]
# print(null_values_delhi_aqi)

#dropping the null values
delhi_aqi_no_nulls= cleaned_delhi_aqi.dropna()
print(delhi_aqi_no_nulls)

#to find total null values in the entire dataframe:
# print(delhi_aqi.isnull().sum().sum())

delhi_aqi_no_nulls.to_csv('delhi_aqi_cleaned.csv')
```

Step 2: Digging with SQL

SQL helped me in structuring and querying the data.

Below are the queries and results of the top 5 polluted and cleanest days in Delhi

```
#DAYS|YEAR|MONTH|AQI|RANKING!  
  
SELECT  
    *  
FROM  
    aqi_data;  
  
select  
    AQI,  
    day(date) as day,  
    year(date) as year,  
    monthname(date) as month  
FROM  
    (select  
        date,  
        aqi,  
        rank() over(partition by year(date),monthname(date))  
        order by aqi desc  
        as ranked_aqi  
    FROM  
        aqi_data  
    ) as ranked_table  
where ranked_aqi <=5  
limit 5;
```

```
#9) Which were the top 5 cleanest days i
```

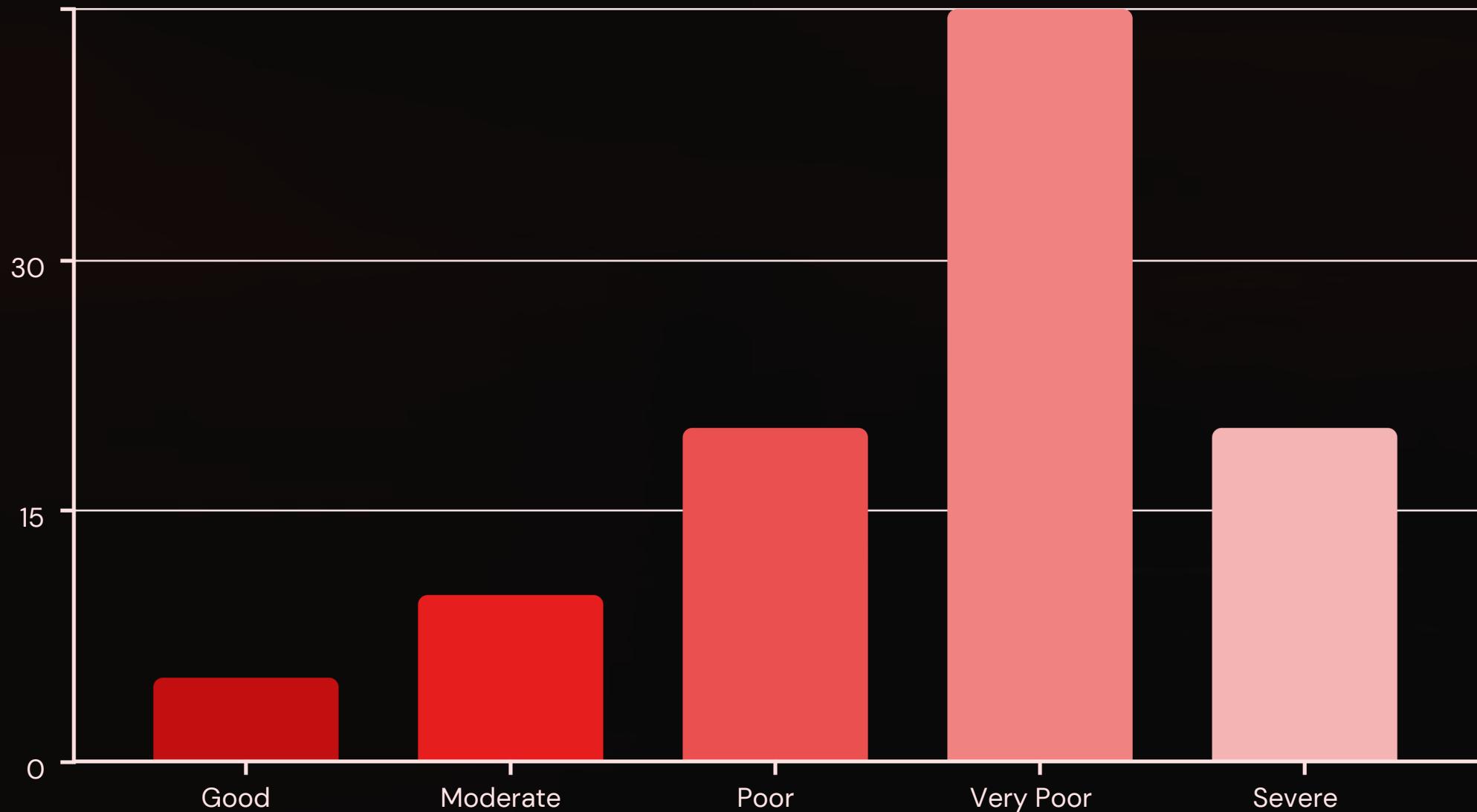
```
#DAYS|AQI|RANKIING
```

```
SELECT  
    day(date),  
    AQI  
FROM(  
    SELECT  
        date,  
        AQI,  
        RANK( ) OVER(PARTITION BY day(date)  
        ORDER BY aqi ASC) as ranked_aqi  
    FROM aqi_data  
    ) as ranked_table  
where ranked_aqi <=5  
limit 5;
```

	AQI	day	year	month
▶	406	19	2015	April
	392	20	2015	April
	391	21	2015	April
	383	24	2015	April
	378	18	2015	April

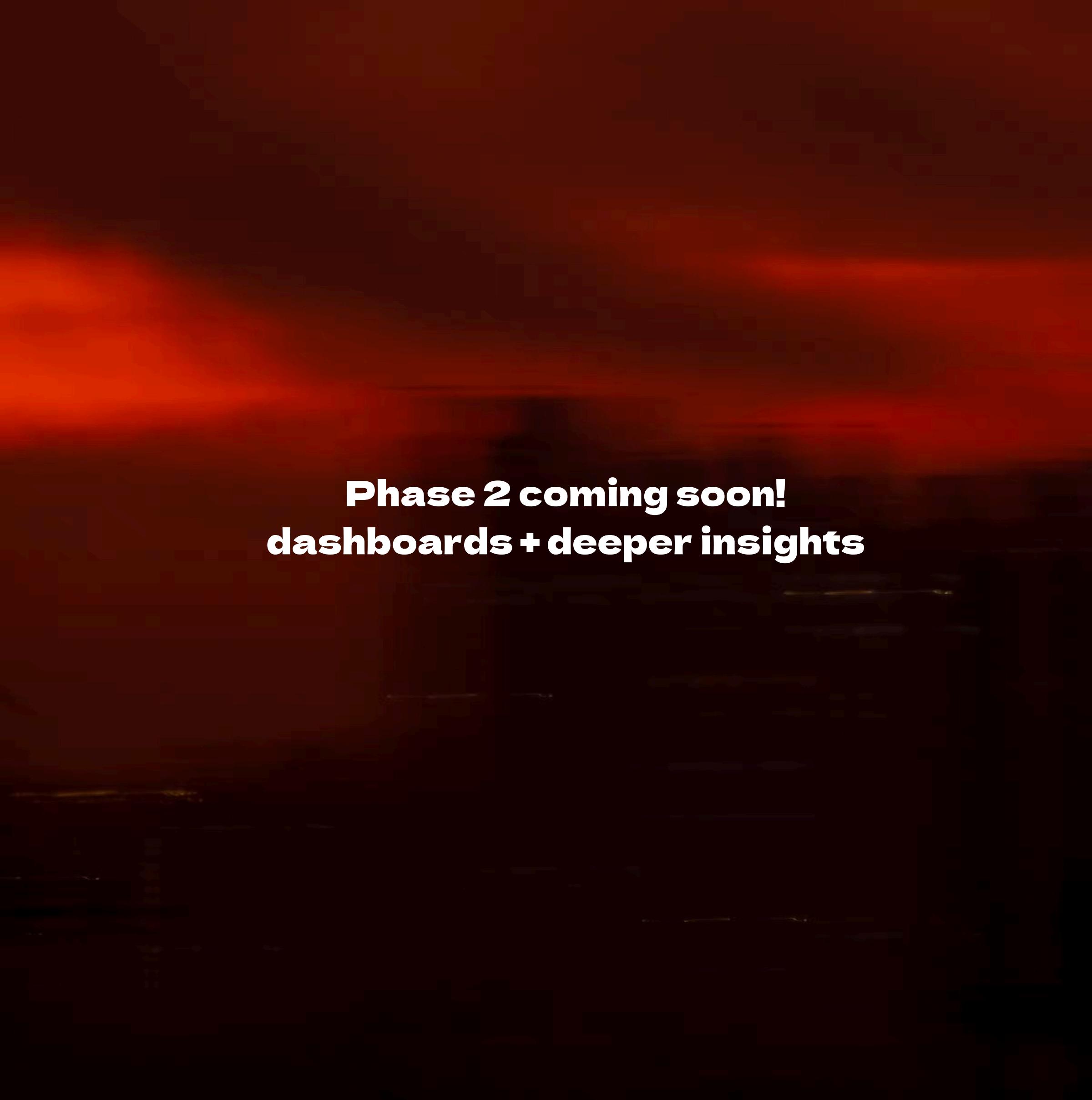
	day	year	monthname	AQI
▶	1	2015	April	309
	1	2018	April	217
	1	2019	April	153
	1	2020	April	80
	1	2018	August	155

Step 3: What the Data Told Me



Key Findings: Disturbing Patterns Emerge

- AQI worsening post-2018
- Nov-Jan most hazardous
- **High PM2.5 in working zones**
- **PM2.5 levels show long-term urban health threats**



Phase 2 coming soon!
dashboards + deeper insights