**AIR QUALITY DATA**

* + - **Name – Muskan Singh**
    - **Reg – no –**
    - **Roll-no- 28**
    - **Section – K23ED**
    - In This project I have covered almost every point of python libraries including NumPy pandas mat plot and seaborn
    - The Website from which I have taken this dataset is -- https://www.kaggle.com/datasets/rohanrao/air-quality-data-in-india
    - This project is based on Electric Vehicle Population Data

LINKDIN POST LINK:

<https://www.linkedin.com/posts/muskaan-singh-ba5a96297_python-dataanalytics-seaborn-activity-7316755877061873664-GiZL?utm_source=social_share_send&utm_medium=android_app&rcm=ACoAAEfisC0BHeR0lDEHxi60I4SfVNFS5SyQT3Q&utm_campaign=whatsapp>

GITHUB POST LINK:

**📝 Air Quality Data in India (2015–2020) – Easy Explanation**

**📌 Overview**

This dataset contains air pollution readings collected from various cities across India over a span of five years (2015 to 2020). The data was made available by the **Central Pollution Control Board (CPCB)** and includes measurements for different pollutants.

**📂 What’s in the dataset?**

Each row in the dataset represents a **recorded observation** of a particular pollutant at a specific **location and time**. Here are the key columns:

* **Date** and **Time**: When the reading was taken
* **City** and **Location**: Where the reading was recorded
* **Pollutant**: Type of pollutant (e.g., PM2.5, PM10, NO₂, CO, SO₂, O₃)
* **Pollutant concentration**: The amount of pollutant present in the air (usually in µg/m³ or mg/m³)
* **AQI Bucket**: Category of air quality (e.g., Good, Satisfactory, Poor, etc.)

**🔍 Why is this data important?**

* Helps monitor the air quality over time
* Identifies highly polluted cities or locations
* Tracks which pollutants are most dangerous or common
* Supports the government and environmental bodies in taking action for cleaner air

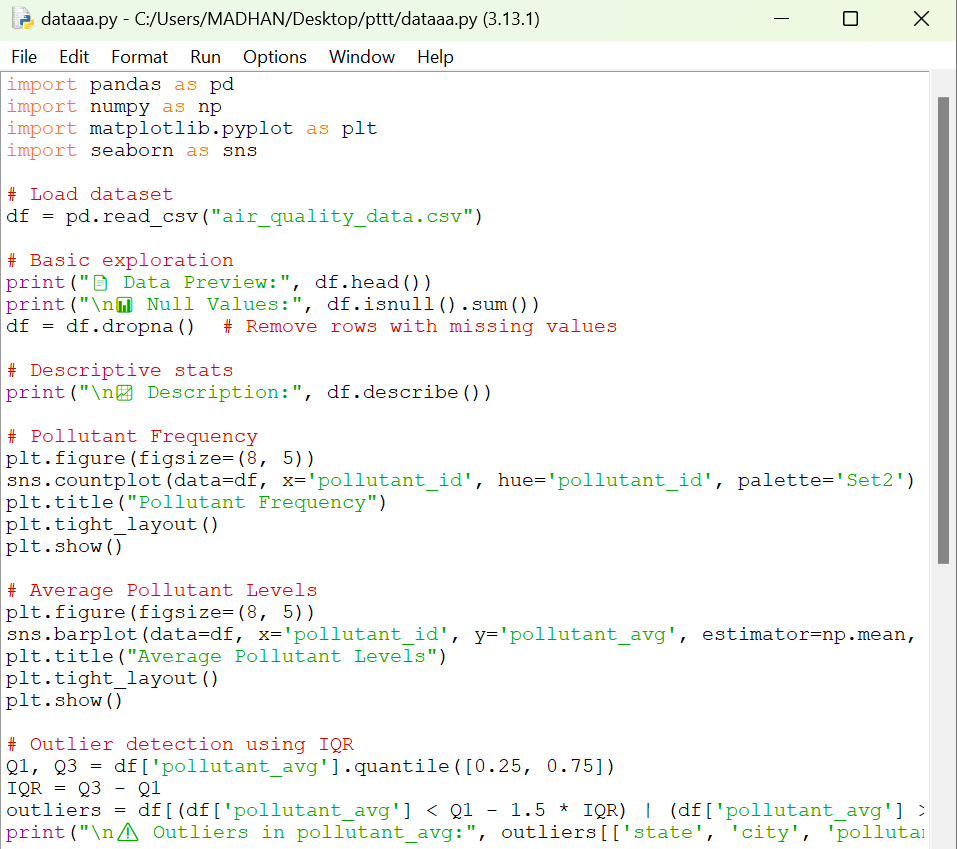
**🔬 What can we analyze?**

Using Python and libraries like Pandas, Matplotlib, and Seaborn, we can:

* Visualize how pollutant levels vary across cities and years
* Find which pollutants are most common or most harmful
* Detect outliers using statistical methods like the **Interquartile Range (IQR)**
* Track **trends over time** using line plots
* Compare air quality between different geographic regions

**✅ Conclusion**

This dataset is a powerful resource for understanding the state of air quality in India. With proper analysis, it reveals pollution trends, identifies problem areas, and encourages better environmental policy and awareness. As air pollution continues to affect health and climate, such data-driven insights can play a crucial role in making our cities cleaner and safer.

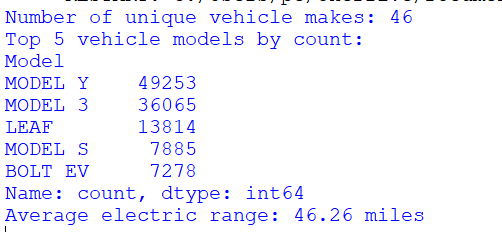
1. Importing the warnings and python libraries in idle python   
 

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.



# HISTOGRAM

creating a histogram based on crime data

Ans – The output of the code is

A screenshot of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

1. creating a Count plot of POLLUTANT Type Distribution

Ans – The output of the code is A screenshot of a computer screen

AI-generated content may be incorrect.

# Seaborn python libraries

1. Creating a bar chat of POLLUTANT AVERAGE

Ans – The output of the code is

A screenshot of a computer screen

AI-generated content may be incorrect.

Ans – The output of the code is

A screenshot of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

BELOW I HAVE WRITTEN ALL THE CODE WHICH I HAVE WRITE IN IDLE PYTHON   
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

# Load dataset

df = pd.read\_csv("air\_quality\_data.csv")

# Basic exploration

print("📄 Data Preview:", df.head())

print("\n📊 Null Values:", df.isnull().sum())

df = df.dropna() # Remove rows with missing values

# Descriptive stats

print("\n📈 Description:", df.describe())

# Pollutant Frequency

plt.figure(figsize=(8, 5))

sns.countplot(data=df, x='pollutant\_id', hue='pollutant\_id', palette='Set2')

plt.title("Pollutant Frequency")

plt.tight\_layout()

plt.show()

# Outlier detection using IQR

Q1, Q3 = df['pollutant\_avg'].quantile([0.25, 0.75])

IQR = Q3 - Q1

outliers = df[(df['pollutant\_avg'] < Q1 - 1.5 \* IQR) | (df['pollutant\_avg'] > Q3 + 1.5 \* IQR)]

print("\n⚠️ Outliers in pollutant\_avg:", outliers[['state', 'city', 'pollutant\_id', 'pollutant\_avg']])

# Histogram of Pollutant Average

plt.figure(figsize=(8, 5))

sns.histplot(df['pollutant\_avg'], bins=15, kde=True, color='skyblue', edgecolor='black')

plt.title("Distribution of Pollutant Average")

plt.tight\_layout()

plt.show()

# Lineplot of Pollutant Levels Over Time

df['last\_update'] = pd.to\_datetime(df['last\_update'])

df = df.sort\_values('last\_update')

plt.figure(figsize=(12, 6))

sns.lineplot(data=df, x='last\_update', y='pollutant\_avg', hue='pollutant\_id', marker='o')

plt.title("Pollutant Trends Over Time")

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()

# Scatterplot of Pollutant Average vs Latitude

plt.figure(figsize=(10, 6))

sns.scatterplot(data=df, x='latitude', y='pollutant\_avg', hue='pollutant\_id', palette='viridis')

plt.title("Pollutant Average vs Latitude")

plt.tight\_layout()

plt.show()

# Correlation Heatmap

numerical\_df = df[['pollutant\_min', 'pollutant\_max', 'pollutant\_avg', 'latitude', 'longitude']]

corr\_matrix = numerical\_df.corr()

plt.figure(figsize=(10, 6))

sns.heatmap(corr\_matrix, annot=True, cmap='coolwarm')

plt.title("Correlation Heatmap")

plt.tight\_layout()

plt.show()