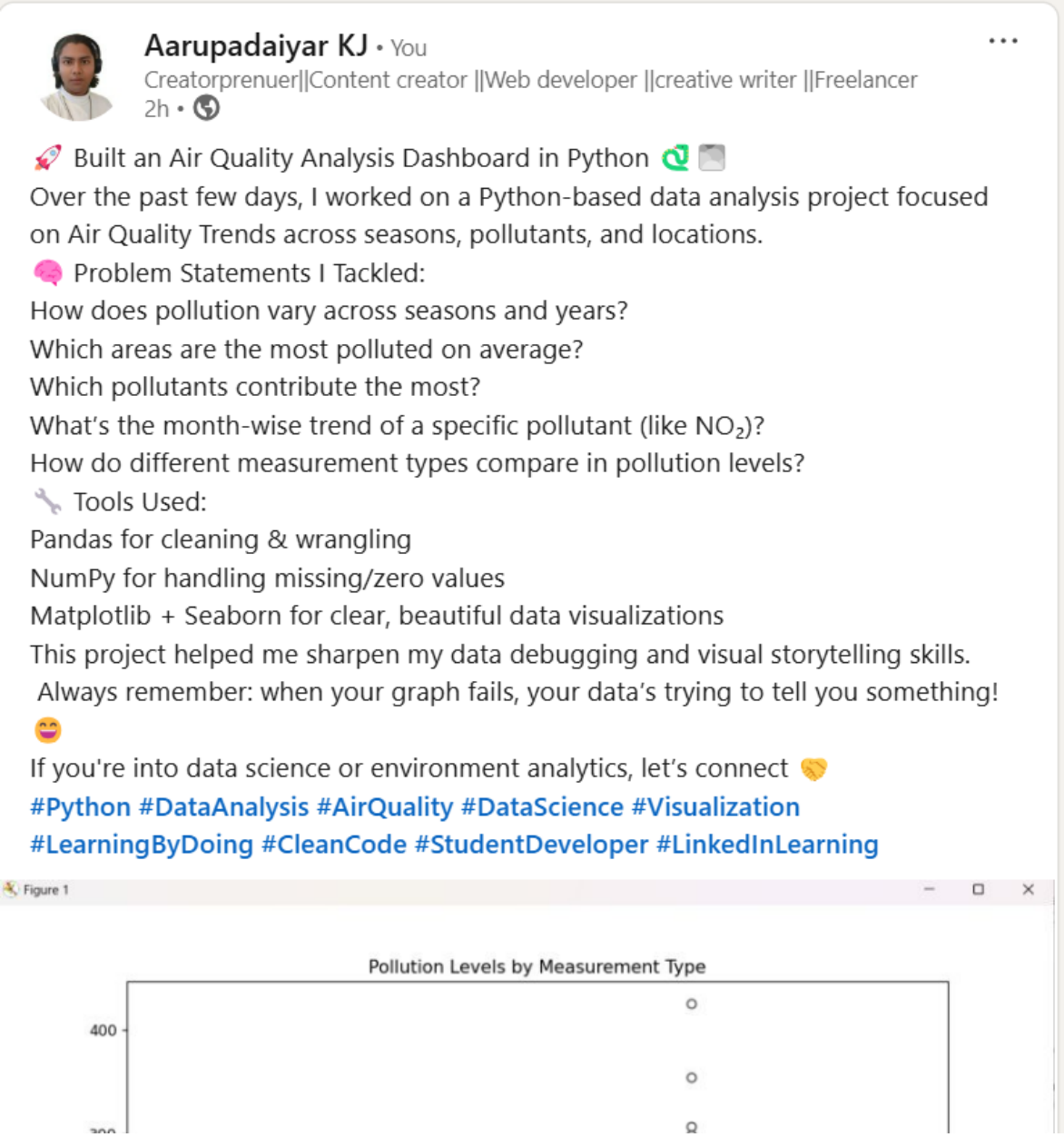
Linkedin Engagement report –12307770



Aarupadaiyar KJ • You
Creatorprenuer||Content creator ||Web developer ||creative writer ||Freelancer
2h •

🚀 Built an Air Quality Analysis Dashboard in Python

Over the past few days, I worked on a Python-based data analysis project focused on Air Quality Trends across seasons, pollutants, and locations.

🧠 Problem Statements I Tackled:

How does pollution vary across seasons and years?

Which areas are the most polluted on average?

Which pollutants contribute the most?

What's the month-wise trend of a specific pollutant (like NO_2)?

How do different measurement types compare in pollution levels?

🔧 Tools Used:

Pandas for cleaning & wrangling

NumPy for handling missing/zero values

Matplotlib + Seaborn for clear, beautiful data visualizations

This project helped me sharpen my data debugging and visual storytelling skills.

Always remember: when your graph fails, your data's trying to tell you something! 😄

If you're into data science or environment analytics, let's connect 🤝

#Python #DataAnalysis #AirQuality #DataScience #Visualization #LearningByDoing #CleanCode #StudentDeveloper #LinkedInLearning

You and 13 others 5 comments

Reactions

Like Comment Repost Send

186 impressions View analytics

Add a comment...

Most relevant

Kundan Kumar • 1st 1h •••

COMPUTER SCIENCE STUDENT AT LOVELY PROFESSIONAL UNIVERSITY

Looks great

Like | Reply

Ravada Sashank • 1st 2h •••

"Aspiring Software Engineer | Computer Science Student at LPU | Passionat...

Very well done, Aarupadaiyar

Like · 1 | Reply

YASHASHVI PAL • 2nd 1h •••

Pursuing Bachelors in technology (computer science and engineering) fro...

6943616/

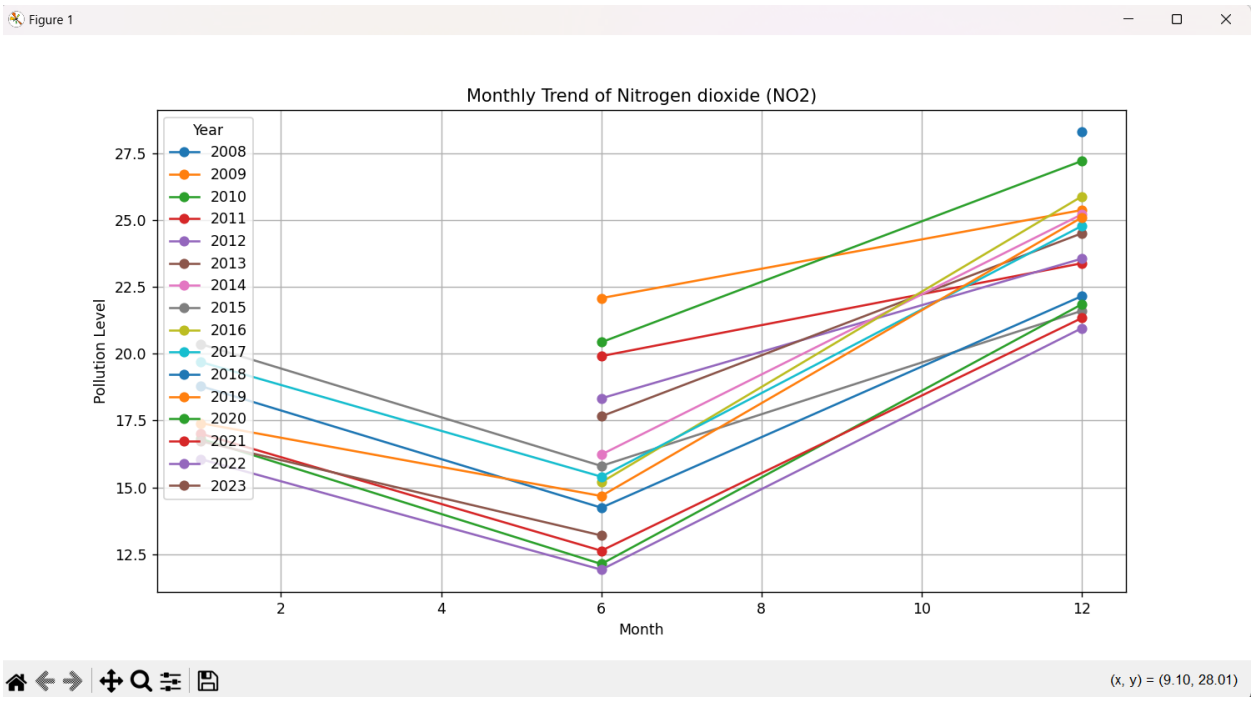


Figure 1
Monthly Trend of Nitrogen dioxide (NO2)
Year
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
Pollution Level
27.5
25.0
22.5
20.0
17.5
15.0
12.5
2
4
6
8
10
12
Month
(x, y) = (9.10, 28.01)

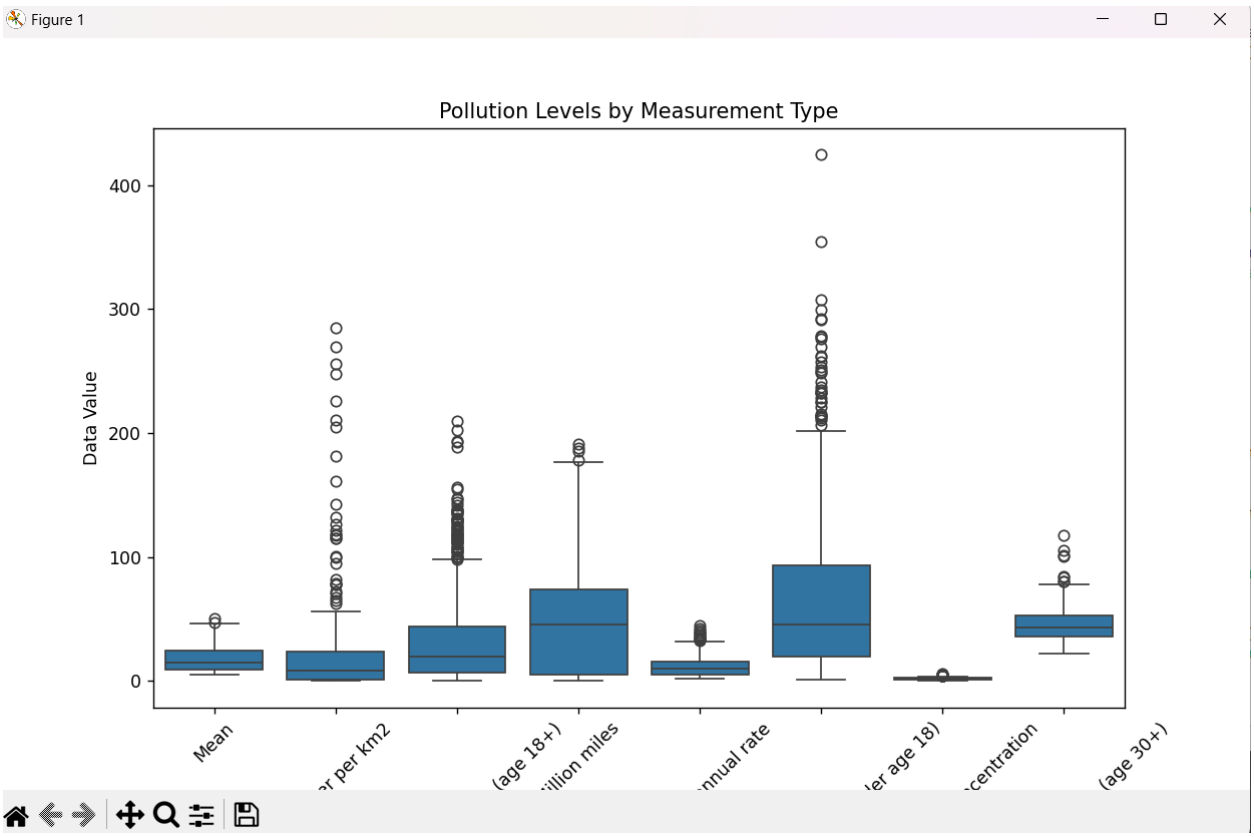


Figure 1
Pollution Levels by Measurement Type
Data Value
400
300
200
100
0
Mean
(age 18+)
(age 30+)

```
"""Problem Statement 1: Seasonal Pollution Patterns
"How do pollution levels vary across different seasons and years?"
 Problem Statement 2: Top 10 Most Polluted Areas
"Which areas recorded the highest average pollution levels?"
Problem Statement 3: Pollution Comparison by Pollutant Type
"Which pollutant types contribute the most to overall pollution?"
Problem Statement 4: Monthly Trend of a Specific Pollutant
"How does the level of a specific pollutant (e.g., NO2 or PM2.5) change over time?"
Problem Statement 5: Measure-wise Contribution
"How do different measurement types (like concentration vs percentage) compare in value?"
"""
"""
❌ Problem Type         📄 Details
1. Missing Values       Empty cells in columns like Data Value, Start_Date
2. Invalid Formats      Date stored as text, inconsistent types
3. Duplicate Rows       Same row repeated multiple times
4. Zero Values          Pollution levels = 0 → may need special handling
5. Inconsistent Text    Different spelling/casing (e.g., "NO2", "no2")
6. Irrelevant Columns   Columns like IDs, Geo Codes, Messages
7. Outliers (Optional)  Extremely high/low values that distort analysis
"""
import pandas as pd       # For data handling
import numpy as np        # For numerical calculations
import matplotlib.pyplot as plt   # For plotting
import seaborn as sns     # For advanced plots
df = pd.read_csv("Air_Quality.csv")
df.head()
# Checking missing value and print"
print("Missing values before cleaning:\n", df.isnull().sum())
#Replace 0
df['Data Value'] = df['Data Value'].replace(0, np.nan)
#Fill the blank with mean
df['Data Value'].fillna(df['Data Value'].mean(), inplace=True)
# Convert date to datetime
df['Start_Date'] = pd.to_datetime(df['Start_Date'], errors='coerce')
# Drop irrelevant columns
irrelevant = ['Message', 'Geo Code', 'Unnamed: 0']
df.drop(columns=[col for col in irrelevant if col in df.columns], inplace=True)
#  Add Season column
def get_season(month):
    if month in [12, 1, 2]: return 'Winter'
    elif month in [3, 4, 5]: return 'Spring'
    elif month in [6, 7, 8]: return 'Summer'
    else: return 'Autumn'

df['Season'] = df['Start_Date'].dt.month.apply(get_season)
df['Year'] = df['Start_Date'].dt.year
df['Month'] = df['Start_Date'].dt.month

# ------------------------------
```

```python
# Problem Statement 1:
# Seasonal Pollution Patterns
# --------------------------------
plt.figure(figsize=(10,6))
sns.boxplot(data=df, x='Season', y='Data Value')
plt.title("Seasonal Pollution Patterns")
plt.ylabel("Pollution Level")
plt.show()

# --------------------------------
# Problem Statement 2:
# Top 10 Most Polluted Areas
# --------------------------------
top_areas = df.groupby('Geo Place Name')['Data Value'].mean().sort_values(ascending=False).head(10)
plt.figure(figsize=(10,6))
sns.barplot(x=top_areas.values, y=top_areas.index, palette="Reds_r")
plt.title("Top 10 Most Polluted Areas")
plt.xlabel("Average Pollution Level")
plt.show()

# --------------------------------
# Problem Statement 3:
# Pollution by Pollutant Type
# --------------------------------
pollutants = df.groupby('Name')['Data Value'].mean().sort_values(ascending=False)
plt.figure(figsize=(10,6))
sns.barplot(x=pollutants.index, y=pollutants.values, palette="mako")
plt.title("Average Pollution by Pollutant Type")
plt.ylabel("Avg Value")
plt.xticks(rotation=45)
plt.show()



# --------------------------------
# Problem Statement 4:
# Monthly Trend of a Specific Pollutant (e.g. NO2)
# --------------------------------
pollutant_to_check = 'Nitrogen dioxide (NO2)'
filtered = df[df['Name'] == pollutant_to_check]

if filtered.empty:
    print(f"No data found for {pollutant_to_check}")
else:
    monthly_trend = filtered.groupby(['Year', 'Month'])['Data Value'].mean().unstack()
    monthly_trend.T.plot(figsize=(12,6), marker='o')
    plt.title(f"Monthly Trend of {pollutant_to_check}")
    plt.ylabel("Pollution Level")
    plt.xlabel("Month")
    plt.grid()
```

```
plt.figure(figsize=(10,6))
sns.barplot(x=pollutants.index, y=pollutants.values, palette="mako")
plt.title("Average Pollution by Pollutant Type")
plt.ylabel("Avg Value")
plt.xticks(rotation=45)
plt.show()



# --------------------------------
# Problem Statement 4:
# Monthly Trend of a Specific Pollutant (e.g. NO2)
# --------------------------------
pollutant_to_check = 'Nitrogen dioxide (NO2)'
filtered = df[df['Name'] == pollutant_to_check]

if filtered.empty:
    print(f"No data found for {pollutant_to_check}")
else:
    monthly_trend = filtered.groupby(['Year', 'Month'])['Data Value'].mean().unstack()
    monthly_trend.T.plot(figsize=(12,6), marker='o')
    plt.title(f"Monthly Trend of {pollutant_to_check}")
    plt.ylabel("Pollution Level")
    plt.xlabel("Month")
    plt.grid()
    plt.show()



# --------------------------------
# Problem Statement 5:
# Measure-wise Contribution
# --------------------------------
plt.figure(figsize=(10,6))
sns.boxplot(data=df, x='Measure', y='Data Value')
plt.title("Pollution Levels by Measurement Type")
plt.xticks(rotation=45)
plt.ylabel("Data Value")
plt.show()
```

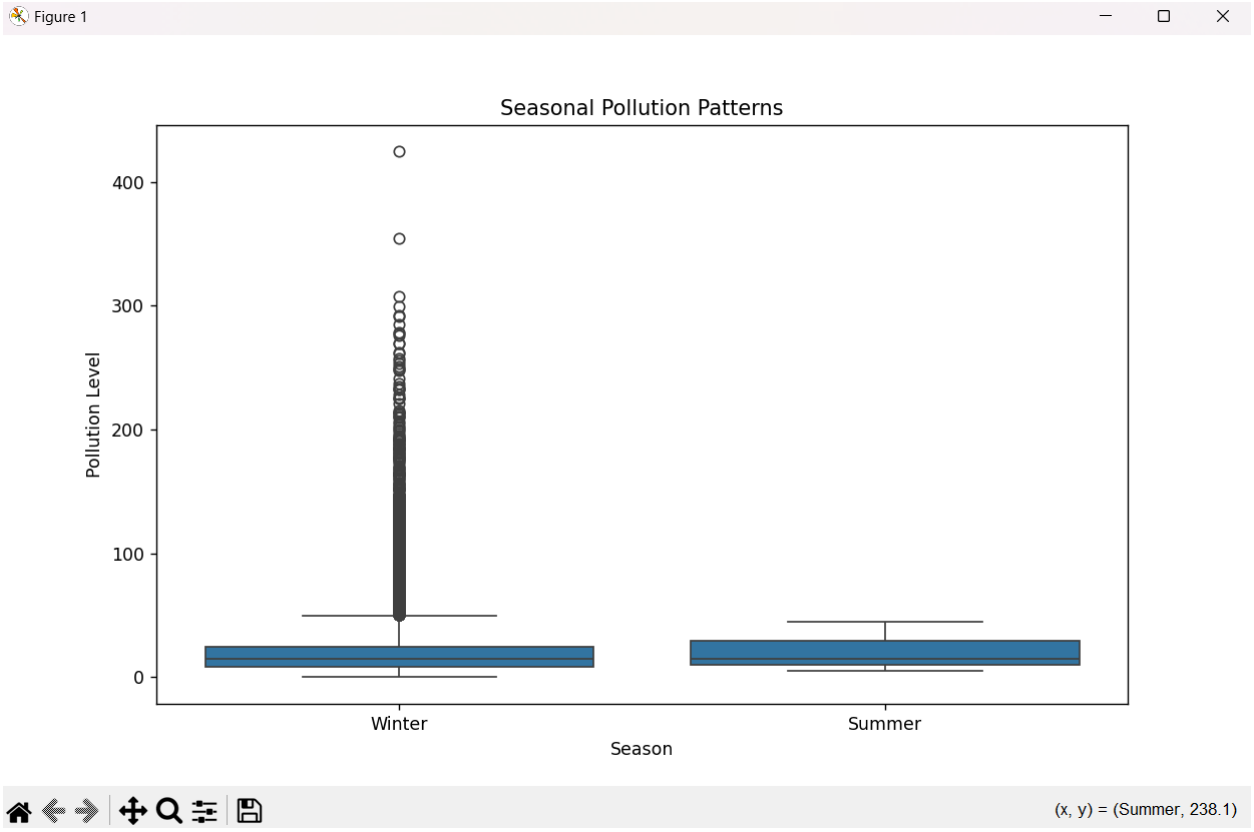


Figure 1
Seasonal Pollution Patterns
Pollution Level
400
300
200
100
0
Winter
Summer
Season
(x, y) = (Summer, 238.1)

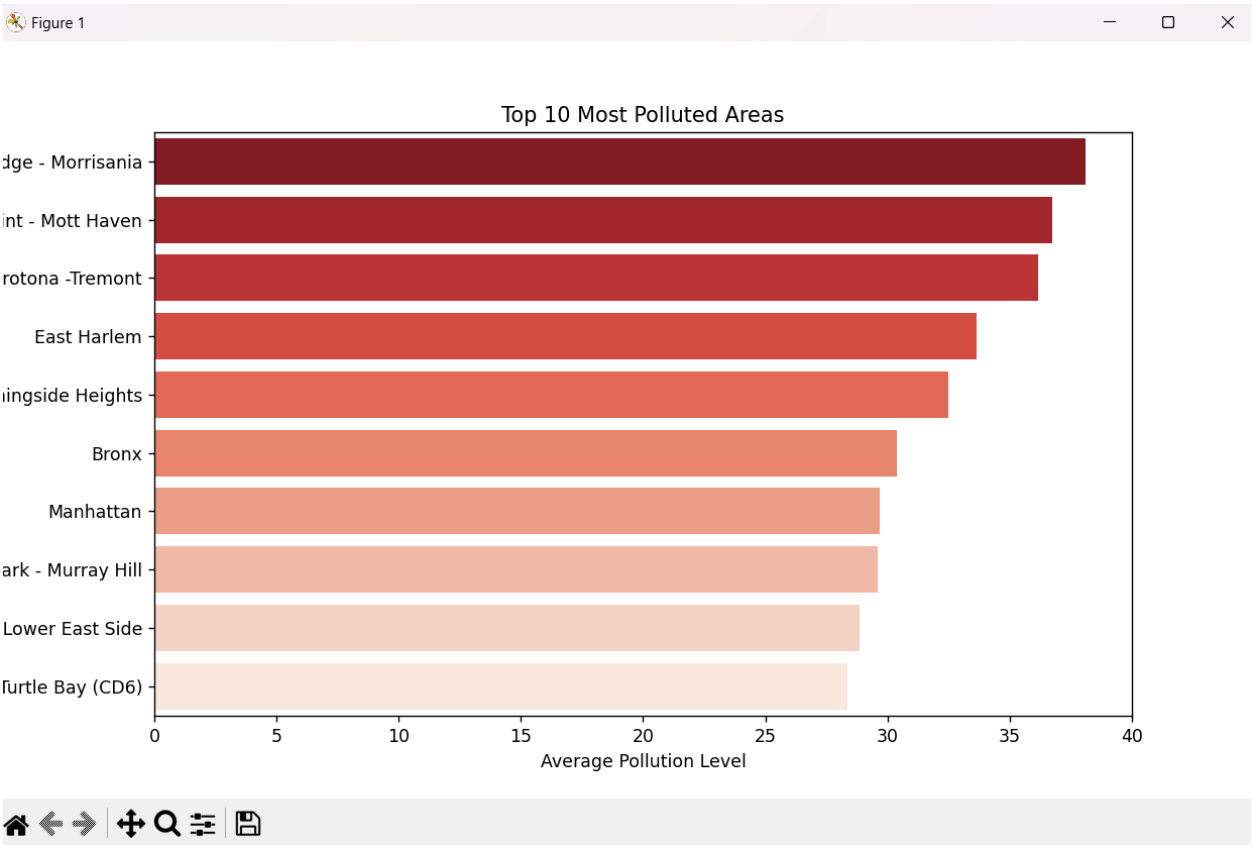


Figure 1
Top 10 Most Polluted Areas
dge - Morrisania
int - Mott Haven
rotona -Tremont
East Harlem
ingside Heights
Bronx
Manhattan
ark - Murray Hill
Lower East Side
Turtle Bay (CD6)
0
5
10
15
20
25
30
35
40
Average Pollution Level