Project Writeup

Pollution Patrol

- The names of all team members, along with a brief overview of how each person contributed

Conor Fogarty and Jonathan Abraham. We both pair programmed the python script together, Conor tends to be stronger in Python and has more API experience so he handled a lot of the scripting when Jonathan tends to be strong in Data Analysis/SQL so he handled writing SQL queries once the API had be built to populate the SQlite database. We worked together to create the data analysis and visualization functions. We then both worked together on writing this report, with each person focusing on writing about what they worked on in the creation of the program.

- The goals of your project – what did you hope to learn from your analysis?

We set out with many goals to achieve while working on this project over the past couple of weeks. The main goal of this project is to study air quality in NYC and Boston by focusing on two air pollutants: PM2.5 and ozone (O_3). To analyze these pollutants and their effect on air quality, we used real data from the OpenAQ API to identify trends in pollution levels, find differences between urban locations, and understand the relationship between pollution and human activity.

The first goal of this project focuses on understanding PM2.5 trends over periods of time. PM2.5 is a type of air pollution made up of tiny particles so small they cannot be seen. The issue with this pollutant is that it can get into people's lungs, which is harmful when inhaled. Some causes of PM2.5 include vehicles, factories, and wildfires. This project analyzes daily PM2.5 trends in New York (Manhattan, Brooklyn, and Queens) and Boston (Chinatown, Kenmore, and Roxbury) to find when pollution levels are at their highest and lowest. Instead of checking pollution levels every hour, we decided to calculate the average for each 24-hour period and update it over time. This method depicts the general trend of how pollution changes over time in each city and in comparison to each other.

The second goal of this project focuses on comparing ozone levels between different cities. Ozone, also known as O_3, is a gas in the air that can be good or bad depending on its location. Ozone can be beneficial when high up in the sky, as it protects humans from the sun's harmful rays. However, near the ground, it can be harmful because it can make breathing difficult. This gas forms when pollutants interact with sunlight, which is a concern in urban areas. This project takes the average daily ozone levels in Manhattan and Boston, finding the similarities and differences to understand how factors like location, sunlight, and traffic affect these ozone levels.

The third goal of this project focuses on retrieving the highest and lowest daily averages of PM2.5 levels from each city over the month. This part of the project aims to retrieve the highest and lowest daily averages of PM2.5 levels from all the different cities over the month. We aim to find the specific days where air quality was either at its best or its worst. The purpose

of this analysis is to analyze how pollution changes over time and find any patterns or events that may be the cause of these high and lows.

- A description of the data set, including any preprocessing you did to get the data into a usable format

The data we used for this project is from OpenAQ (explore.openaq.org) and spans the period from November 16, 2024 until December 16th, 2024. This website enables anyone to look at air pollution data from monitoring stations all around the globe. There are several types of information this open data source provides the location of stations,and a wide variety of air quality measurements such as ambient temperature, PM2.5 levels, o3 levels and other air quality measurements depending on each station's measurement tools. For our analysis, we focused on the two pollutants PM2.5 and Ozone (O_3) and collected this data from New York(Manhattan: 625, Brooklyn: 648, and Queens: 631) and Boston (Chinatown: 384, Kenmore: 521, Roxbury: 448, Roxbury, Fort Hill: 2117520, and Von Hillern: 452). These two cities were chosen to compare pollution levels in urban areas and find patterns in their air quality.

The dataset provided many types of fields, such as pollutant (parameter), its measured value, units of measurement (µg/m³ for PM2.5, ppb for O_3), timestamps showing when the measurements were taken, and the exact latitude and longitude coordinates of the monitoring stations. This data was collected for a month (30 full days, from 11-16-2024 to 12-16-2024) to ensure there was enough information to observe any trends. In order to retrieve the data, we had to make a total of 8 calls to 8 different api endpoints from OpenAQ API to get each of the 8 station locations individual data. Once the data was collected, it was parsed and inserted into a SQLite database named **air_quality.db**, utilizing a table called locations to save location id, city, name and coordinate information and a table titled “measurements” was created to store important information such as pollutant type, location IDs, values and units, timestamps, and coordinates for locations.

To create a robust and structurally integral dataset, we initially selected three monitoring points from New York and Boston. We originally chose one data point from each city, however, we ran into data sourcing issues and felt that the data was not diverse enough. We then decided to go with three data points per city, these points were purposely chosen to form a triangle around as much of the city as possible to provide the most inclusive dataset for each city. This approach helped ensure that the dataset was inclusive, collecting more air quality data from different parts of each city. We then ran into issues with Boston, which we go into more detail in in the data issues portion of this report, and added two more data source locations to Boston, totalling five locations in Boston and three in New York.

- A short writeup for each task completed, summarizing the techniques you used, as well as any conclusions you were able to draw

Task 1 and Task 2: Data Collection (API Requests) and Storage(SQlite)

For this task, we created a SQLite database named **air_quality.db** by creating a Python class called **AirQualityDB**. In this class, we implemented the **create_tables** function, which created a locations table to save location data such as city name, latitude, and longitude, and a measurements table to store parameters (pollutant type) with value and unit qualifiers, timestamp, latitude, and longitude. Additionally, we implemented the **insert_location** and **insert_measurements** functions to populate the database tables created using the **create_tables** function. Finally, we created the **fetch_measurements** function to fetch data directly from OpenAQ's API at {https://api.openaq.org/v2/measurements} and properly structure the response so it could be inserted into the tables we made prior. Our fetch_measurements function, when initialized by our main method, submits get requests to 8 different api endpoints (5 Boston station locations and 3 New York Station Locations)and parses them to be placed into the SQlite database.

Task 3: PM2.5 Trends Analysis

For the PM2.5 trend analysis, we created two analytical functions and one visualization function. The analytical functions, **get_pm25_extremes** and **get_city_pm25_trends**, both generate a dataframe using Pandas. Within each of these functions, we defined a variable query, which contains a SQL query to retrieve the required analytical information. We then used the **pandas.read_sql_query** function on the SQLite database, leveraging self.conn to connect to the database and passing the query along with parameters to filter and retrieve the appropriate data.

The final function, **plot_city_pm25_trends**, calls the **get_city_pm25_trends** function with the specified parameter (city). It then plots a figure to visually represent the data retrieved from g**et_city_pm25_trends**. This function uses a for loop to iterate over each location provided in the data, plotting both the raw average PM2.5 values and a rolling average, which smooths the data over time. The resulting figure is saved as a PNG file to represent the trends in PM2.5 levels.

Task 4: Ozone Level Comparison

For the Ozone Level Comparison, we created two analytical functions and one visualization function. Similar to the PM2.5 trend analysis, both analytical functions, **compare_ozone_levels** and **get_city_ozone_levels**, first generated a Pandas dataframe. Within these functions, we created a variable to store the desired SQL query and used the pandas.read_sql_query function to connect to the SQLite database. The query and parameters were used to filter and retrieve the specific information needed from the database. For the visualization, the **plot_cities_ozone_comparison** function called the **compare_ozone_levels**

function with two city parameters, **city1** and **city2**. It then plotted the ozone level data for both cities on a shared figure, allowing for a direct visual comparison of daily average ozone trends between the two cities.

Task 5: Extreme PM2.5 Days(Daily)

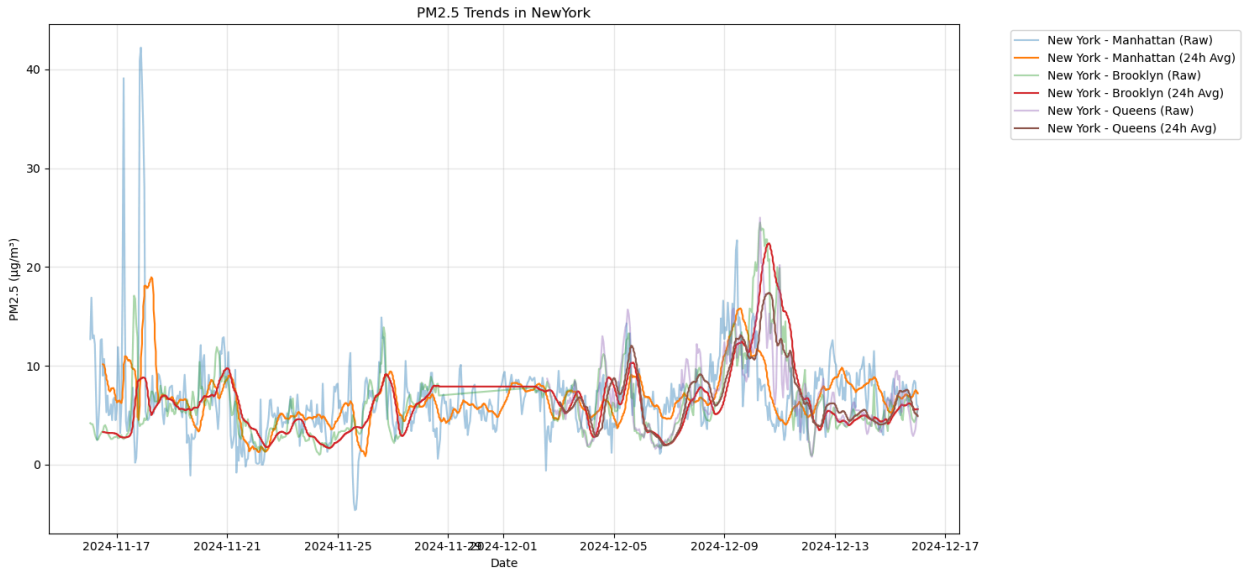
Within the main function, we implemented the AirQualityAnalyzer class, which contains all four analytical functions (two for PM2.5 analysis and two for ozone analysis) along with the two visualization functions. We instantiated this class in a variable called **analyzer**. To analyze PM2.5 extremes, we used the previously created function **get_pm25_extremes** by calling **analyzer.get_pm25_extremes**. This function retrieved the necessary data, and we printed out the five highest and five lowest PM2.5 days recorded in each city over the past 30 days

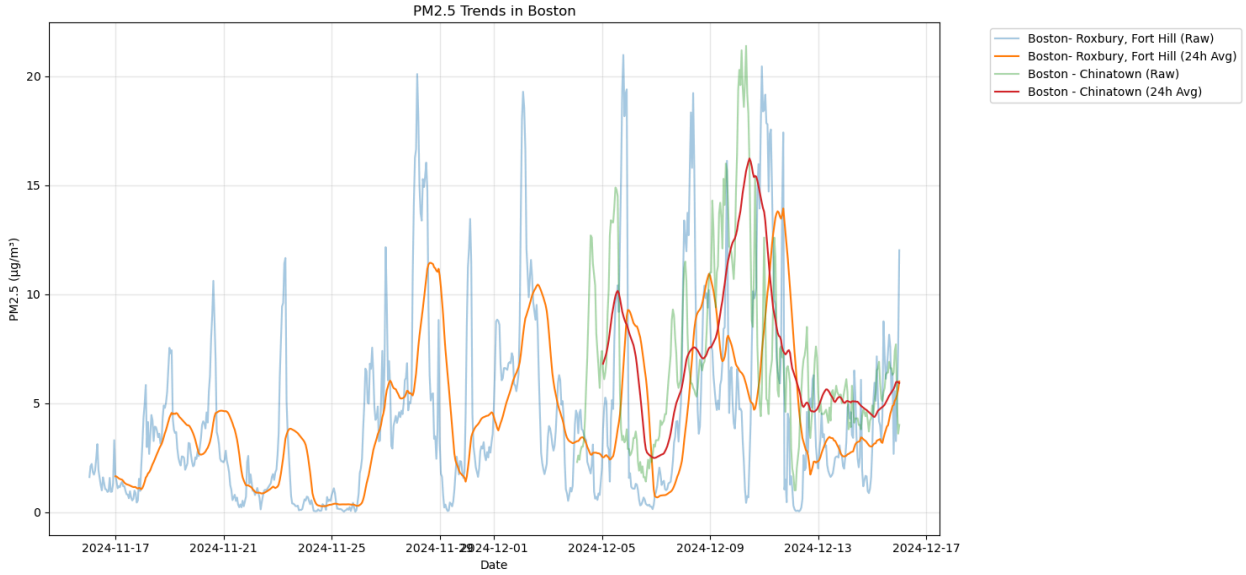
● Description of challenges you encountered when working with the data, and how you were able to overcome them (or not!)

Boston required two more monitoring points compared to New York due to issues with measurements (the measurement of PM2.5) not being recorded or machine malfunctions even though OpenAQ listed them as recording PM2.5. Only Roxbury-Fort Hill, and Chinatown successfully recorded PM2.5 levels. Initially, all three original stations in Boston(Chinatown, Kenmore, and Roxbury) were expected to record PM2.5 values. However, during the selected time period, only Chinatown was actively recording PM2.5 values. We originally thought the issues were related to the SQL queries or data insertion errors. However, we later found out that only Chinatown was actually recording during the selected time, while all three original stations in Boston (Chinatown, Kenmore, and Roxbury) were actually supposed to record PM2.5 values. This taught us to not blindly follow api documentation, as it may be out of date or inaccurate. Also, it was found that although individual parameter data was generally not missing in New York station's reports, sometimes days were missing in New York Data, in particular the data coming from the Queens location did not start reporting until the later half of November.

Other than that, it took some trial and error in Beekeeper studio to find the SQL queries that yielded the results we needed. However, the correct queries were determined, implementing them in Python turned out to be pretty simple. Also, it was surprising to find that setting up the SQLite database and inserting values fetched from the API was quite simple. Resources like **Homework 4 - Voting SQL** and the video "**SQLite Databases With Python - Full Course**" by freeCodeCamp.org on YouTube were very helpful and made the process much easier.

● Descriptions of any insights into the data or domain that you obtained through your work

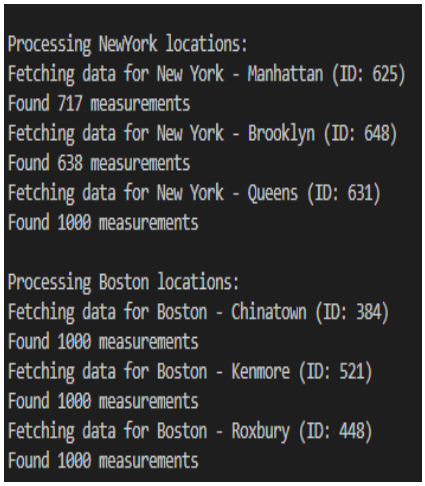




After analyzing the PM2.5 trends in the New York graph, it is pretty clear that Brooklyn shows large spikes in PM2.5 levels, while Manhattan remains more in the middle, and Queens tends to stay on the lower end. From this graph we can infer that on average, from highest to lowest air pollution the three station locations are ranked with Brooklyn by far as the borough with the most air pollution out of the three locations, however Queens and Manhattan are relatively neck and neck, the their averages matching most days, and with the only days of major difference being Manhattan significantly higher on December 10th then Queens, and Queens being significantly higher on December 11th then Manhattan.

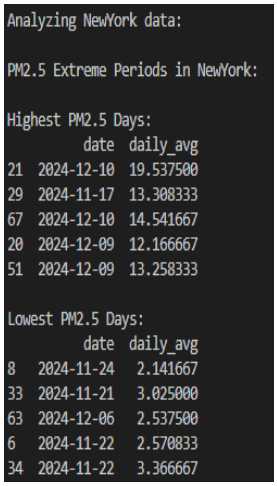
For the Boston graph, it appears that we may have incomplete data. This is because during the first half of the month, only Roxbury recorded data, and later, Chinatown started reporting their data to OpenAQ. This means there may have been issues with the machines or the recording process, as the incomplete data makes it challenging to perform an analysis. Overall, we can infer that Roxbury tends to go through significant ups and downs in their air pollution day to day and that Chinatown, when reporting, seemed to follow this trend.

The Ozone Level Comparison graph shows that the Ozone layers remain relatively the same between New York and Boston. Both cities experience peaks and dips and about the same points, which means that similar factors can be influencing the ozone levels in both cities. We know that the two cities, New York City and Boston, both share a relatively close geographic location, similar vehicle infrastructures and industries which supports the inference from this graph that they experience a very similar ozone, especially due to the lack of any large environmental disaster that would affect the ozone in either area. Overall, all this graph concludes is the assumption that they share a relatively similar ozone layer.



```
Processing NewYork locations:
Fetching data for New York - Manhattan (ID: 625)
Found 717 measurements
Fetching data for New York - Brooklyn (ID: 648)
Found 638 measurements
Fetching data for New York - Queens (ID: 631)
Found 1000 measurements

Processing Boston locations:
Fetching data for Boston - Chinatown (ID: 384)
Found 1000 measurements
Fetching data for Boston - Kenmore (ID: 521)
Found 1000 measurements
Fetching data for Boston - Roxbury (ID: 448)
Found 1000 measurements
```

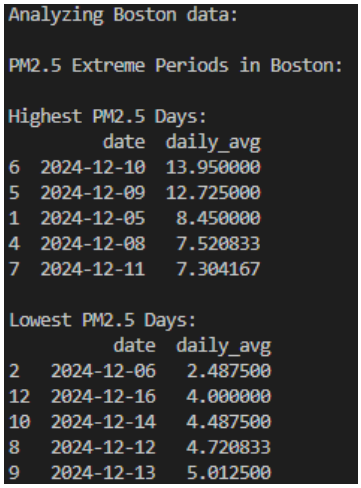


```
Analyzing NewYork data:

PM2.5 Extreme Periods in NewYork:

Highest PM2.5 Days:
         date  daily_avg
21 2024-12-10  19.537500
29 2024-11-17  13.308333
67 2024-12-10  14.541667
20 2024-12-09  12.166667
51 2024-12-09  13.258333

Lowest PM2.5 Days:
         date  daily_avg
8  2024-11-24   2.141667
33 2024-11-21   3.025000
63 2024-12-06   2.537500
6  2024-11-22   2.570833
34 2024-11-22   3.366667
```



```
Analyzing Boston data:

PM2.5 Extreme Periods in Boston:

Highest PM2.5 Days:
         date  daily_avg
6  2024-12-10  13.950000
5  2024-12-09  12.725000
1  2024-12-05   8.450000
4  2024-12-08   7.520833
7  2024-12-11   7.304167

Lowest PM2.5 Days:
          date  daily_avg
2   2024-12-06   2.487500
12  2024-12-16   4.000000
10  2024-12-14   4.487500
8   2024-12-12   4.720833
9   2024-12-13   5.012500
```

Boston and New York shared these highest PM2.5 days, December 10th and December 9th 2024. Boston and New York shared these lowest PM2.5 days, December 6th 2024. For New York, the highest daily average PM2.5 value recorded was 19.5375(µg/m3) and the lowest recorded PM2.5 value was 2.141667(µg/m3), and therefore saw a daily average delta of 17.395833(µg/m3). For Boston, the highest daily average PM2.5 value recorded was 13.95(µg/m3) and the lowest daily average PM2.5 value recorded of 2.4875(µg/m3), with a daily average delta of 11.4625(µg/m3). As you can see, not all data was populated for every day for station Manhattan and station Brooklyn, which can indicate to us that the highest daily average could be higher, the lowest daily average could be lower, and therefore the daily average delta could be larger, but obviously the inverse can not be true. From this analysis, we can infer that on the worst days of air pollution, New York experiences significantly worse air pollution than Boston ever does, at a minimum(remember the missing values in New York) of 40% more air pollution of PM2.5 particles. This level of air pollution difference was directly felt by residents of Boston and New York on December 10th, 2024.

- Ideas for future exploration of the data, including interesting questions raised by your analysis

Seasonal Trends:

We only conducted analysis over a one month span, would analysis over multiple years yield insights into season trends of air pollution?

Impact of Events:

Analyze how specific events (e.g., holidays, storms, or public transportation strikes) influence air pollution levels, in particular we would be interested to see if great gatherings such as the Macy Parade or Boston Sport Teams parades (since we are so lucky they happen so often) affect air quality.

Machine Learning for Prediction:

Use ML models to predict PM2.5 and O_3 levels based on historical data, weather patterns, and human activity indicators.