Ha Linh Nguyen

LIS 487 - Data Interoperability

DATA INTEROPERABILITY USE CASE

PART 0: ACQUIRING DATA

1.Project Description: For my Data Interoperability project, I have chosen the "Air Quality" dataset from the Data.gov platform. The dataset focuses on New York City's air quality surveillance data, providing information about various pollutants, exposures, and their impacts on different neighborhoods. As air pollution is a critical environmental threat, this dataset offers valuable insights into pollutant emissions, exposure levels, and population vulnerability, contributing to a better understanding of air quality and health in NYC.

2.Data Source: Air Quality Dataset - Data.gov

3.Data Information: The dataset contains 16128 rows with 13 columns in total.

4.Data Description:

- **Unique ID:** A unique identifier for each data entry.
- **Indicator ID:** Identifies the type of air quality indicator, e.g., Nitrogen Dioxide (NO2).
- **Name:** The name of the air quality indicator.
- **Measure:** Type of measurement, e.g., Mean.
- **Measure Info:** Additional information about the measurement unit, e.g., parts per billion (ppb).
- **Geo Type Name:** The type of geographic region.
- **Geo Join ID:** A unique identifier for joining with geographical data.
- **Geo Place Name:** The name of the geographic location or neighborhood.
- **Time Period:** Specifies the time period for the data, e.g., Annual Average 2011.
- **Start_Date:** The start date of the time period.
- **Data Value:** The actual measured value of the air quality indicator.
- **Message:** Additional information or messages related to the data entry.

PART 1: TRANSFORMING DATA

Code:

```
import csv
import json
```

```python
def csv_to_json(csv_file_path, json_file_path):
    #create a dictionary
    data_dict = {}

    #open a csv file handler
    with open(csv_file_path, encoding = 'utf-8') as csv_file_handler:
        csv_reader = csv.DictReader(csv_file_handler)

        #convert each row into a dictionary
        #and add the converted data to the data_variable

        for rows in csv_reader:
            key = rows['Unique ID']
            data_dict[key] = rows

    #open a json file handler and use json.dumps
    #method to dump the data
    with open(json_file_path, 'w', encoding = 'utf-8') as json_file_handler:
        json_file_handler.write(json.dumps(data_dict, indent = 4))

csv_file_path = 'air_quality.csv'
json_file_path = 'air_quality_json_ver.json'

csv_to_json(csv_file_path, json_file_path)

print(f'Transformation complete. JSON file saved at: {json_file_path}')
```

I've chosen to convert the Air Quality data from CSV to JSON for the following reasons:

- Hierarchical Structure: JSON allows for nested structures, which are perfect for intricate data interactions, particularly in hierarchical datasets.
- Web Compatibility: JavaScript and JSON work together effortlessly, which makes JSON ideal for web applications and data sharing via web services. It is also commonly used in web development.
- Flexibility for Integrations: JSON can be easily adapted for future integrations with other systems or tools due to its adaptability and simplicity of parsing across programming languages.
- Readability: JSON is comparatively straightforward to understand, even if it isn't as human-readable as CSV. This makes it better for manual data examination and teamwork.

PART 2: CHARTING DATA

Code:

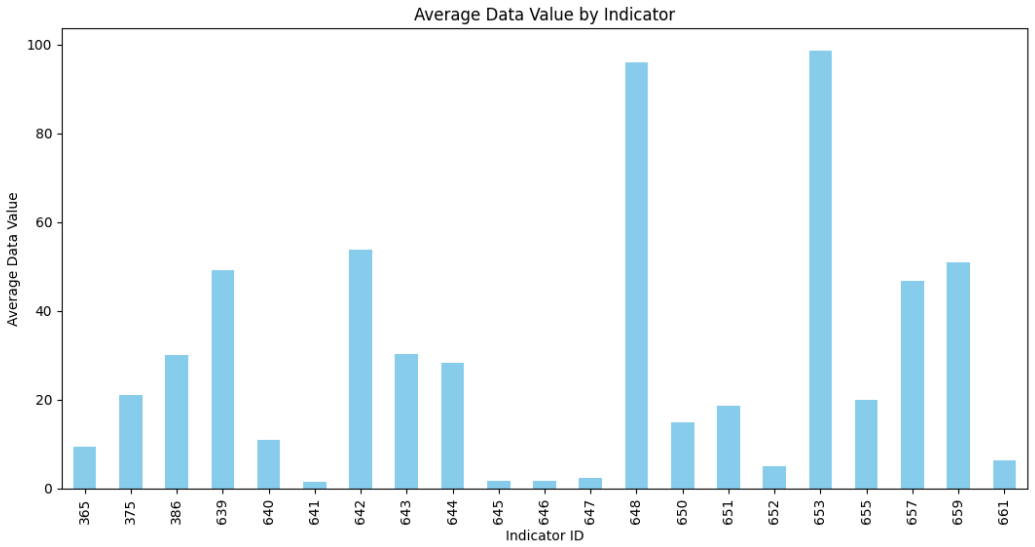
```
import pandas as pd
import matplotlib.pyplot as plt

# Load the data
data = pd.read_csv('air_quality.csv')

# Bar Chart of Average Data Value by Indicator
# Group data by Indicator and calculate the average Data Value
avg_data_by_indicator = data.groupby('Indicator ID')['Data Value'].mean()

# Plotting a bar chart
plt.figure(figsize=(10, 6))
avg_data_by_indicator.plot(kind='bar', color='skyblue')
plt.title('Average Data Value by Indicator')
plt.xlabel('Indicator ID')
plt.ylabel('Average Data Value')
plt.show()
```

Chart implemented from the code:



Interpretation of the Bar Graph:

The bar chart reflects the average data value of each air quality indicator. Notably, Asthma emergency department visits due to PM2.5 and Asthma emergency department visits due to Ozone exhibit high data values, nearly reaching 100. In contrast, Boiler Emissions-Total PM2.5 Emissions, Annual vehicle miles traveled (trucks), Outdoor Air Toxics – Benzene, and Outdoor Air Toxics – Formaldehyde have considerably lower data values, ranging around 2-4. The wide range of average data values across all indicators, from above 0 to nearly 100, highlights the diverse air quality conditions in different neighborhoods. This variation emphasizes the need for

targeted interventions to address specific pollutants and improve overall air quality and health in NYC. In summary, the bar chart provides a visual overview of the average data values for various air quality indicators, highlighting both high-impact and low-impact indicators and emphasizing the need for targeted air quality management strategies in specific geographic areas.

PART 3: ANALYZING DATA

- **What are the unique pollutants (Indicator Names) present in the dataset?**

Code:

```
import pandas as pd
import matplotlib.pyplot as plt

# Assuming your data is in a file named 'Air_Quality.csv'
file_path = 'air_quality.csv'
data = pd.read_csv(file_path)

# Question 1: Distinct the unique pollutants (Indicator Names) present in the
dataset
unique_pollutants = data['Name'].unique()
print("Unique Pollutants in the Dataset:")
for pollutant in unique_pollutants:
    print(pollutant)
print()
```

Result:

```
Unique Pollutants in the Dataset:
Nitrogen dioxide (NO2)
Fine particles (PM 2.5)
Ozone (O3)
Asthma emergency department visits due to PM2.5
Annual vehicle miles traveled
Asthma hospitalizations due to Ozone
Respiratory hospitalizations due to PM2.5 (age 20+)
Boiler Emissions- Total SO2 Emissions
Cardiovascular hospitalizations due to PM2.5 (age 40+)
Boiler Emissions- Total PM2.5 Emissions
Boiler Emissions- Total NOx Emissions
Annual vehicle miles travelled (cars)
Annual vehicle miles travelled (trucks)
Cardiac and respiratory deaths due to Ozone
Asthma emergency departments visits due to Ozone
Outdoor Air Toxics - Formaldehyde
Outdoor Air Toxics - Benzene
Deaths due to PM2.5
```

We can see from the code's output that the dataset includes a wide range of air quality indicators, including pollutants such as nitrogen dioxide (NO2), fine particles (PM 2.5), and ozone (O3). It also includes health-related metrics such as asthma emergency department visits due to PM2.5, cardiovascular hospitalizations due to PM2.5 (age 40+), and information on emissions from sources such as boilers. The list reflects the dataset's richness in capturing various aspects of air quality and its effects on human health, offering a comprehensive inventory of pollutants and associated health outcomes for potential analysis and insights.

- **Compare the air quality in different geographic regions (Geo Place Name) for the average Nitrogen dioxide (NO2).**

Code:

```
# Question 2: Compare the air quality in different geographic regions (Geo Place
Name) for a specific indicator and measure type.
no2_data = data[(data['Indicator ID'] == 375) & (data['Measure'] == 'Mean')]
geo_comparison = no2_data.groupby('Geo Place Name')['Data
Value'].mean().sort_values(ascending=False)
print(geo_comparison)
print()
```

Results:

```
Geo Place Name
Midtown (CD5)                           35.257436
Gramercy Park - Murray Hill             32.964615
Chelsea - Clinton                       30.990256
Stuyvesant Town and Turtle Bay (CD6)    30.703590
Chelsea-Village                         29.882821
                                          ...
Southern SI                             13.340513
Rockaways                               12.974359
Rockaway and Broad Channel (CD14)       12.863077
South Beach - Tottenville               12.385385
Tottenville and Great Kills (CD3)       12.296923
Name: Data Value, Length: 114, dtype: float64
```

The result shows a sorted list of average air quality values of Nitrogen Dioxide (NO2) from various regions, with Midtown (CD5) having the highest mean data value of 35.26, indicative of elevated urban pollution. This indicates that Nitrogen dioxide (NO2) concentration may be higher in the Midtown area and underscores potential health risks for residents and emphasizes the need for targeted interventions and regulatory measures, particularly in identified hotspots like Midtown. Other regions, including Gramercy Park-Murray Hill, Chelsea-Clinton, Stuyvesant

Town, and Turtle Bay (CD6), follow with descending mean data values, providing a comparative overview of Nitrogen dioxide levels in various geographic areas. In contrast, Southern SI, Rockaways, Rockaway and Broad Channel (CD14), South Beach – Tottenville, Tottenville and Great Kills (CD3) have the lowest mean data value of Nitrogen Dioxide (NO2), around 12-13. There are 114 regions in total, allowing for a thorough examination of air quality variations across New York in terms of average Nitrogen Dioxide (NO2). In conclusion, the findings underscore the importance of environmental justice considerations and suggest the need for ongoing monitoring and efforts to mitigate pollution in specific regions.

- **Explore how air quality has changed over time for Ozone (O3) measured by Mean**

Code:

```
# Question 3: Explore how air quality has changed over time for a specific
indicator and measure type.
ozone_data = data[(data['Indicator ID'] == 386) & (data['Measure'] == 'Mean')]
ozone_data.loc[:, 'Start_Date'] = pd.to_datetime(ozone_data['Start_Date'])
time_series_analysis = ozone_data.groupby('Start_Date')['Data Value'].mean()
print(time_series_analysis)
print()
```

Result:

```
Start_Date
2009-06-01    25.888511
2010-06-01    32.439291
2011-06-01    31.818794
2012-06-01    32.921348
2013-06-01    30.037234
2014-06-01    30.451631
2015-06-01    30.910922
2016-06-01    32.975532
2017-06-01    28.791277
2018-06-01    29.924681
2019-06-01    29.646312
2020-06-01    29.728794
2021-06-01    29.734965
Name: Data Value, dtype: float64
```

The code successfully investigates how air quality has changed over time, specifically Ozone (O3) and mean as the measure type. The given result is a time series analysis that shows the average data values for Ozone concentrations for each start date. The data show that Ozone levels fluctuate over the specified time period, with notable peaks in 2010 and 2016. The highest average concentration was found in June 2010 at 32.44, and the lowest was found in 2017 at 28.79. This time series analysis provides valuable insights into Ozone concentrations'

historical trends, assisting in the assessment of air quality dynamics and potential contributing factors over time.

CONCLUSION

In conclusion, my Data Interoperability Use Case project focusing on the "Air Quality" dataset from Data.gov has provided valuable insights into various aspects of air quality in New York City. Through the acquisition, transformation, charting, and analysis of data, several key findings and considerations have emerged.

1. Main Findings:

- Comprehensive inventory of air quality indicators, including pollutants and health-related metrics.
- Significant variations in air quality indicators across neighborhoods, emphasizing the need for targeted interventions.
- Nitrogen Dioxide (NO2) concentrations highlight environmental justice considerations, requiring ongoing monitoring.
- Time series analysis of Ozone (O3) concentrations reveals historical trends and potential contributing factors.

2.Limitations/ Problems in the Data

- Static nature of the dataset may limit real-time insights.
- Absence of contextual factors hinders a comprehensive understanding.
- Geographic granularity may impact precision in localized variations.

3.Suggestions for Future work:

- Incorporate real-time data streams for more dynamic analyses.
- Integrate additional datasets with contextual factors for a nuanced understanding.
- Increase geographic granularity to refine targeted interventions.
- Explore synergies between different pollutants for improved precision in air quality management strategies.