2nd Assignement

Air Quality Data NYC

Objective

Analyze and study data representing the air quality index in New York (US). Develop a **Python program** that uses Pandas to read and parse the given dataset and plot some useful analyses. You will use **pandas** to implement this assignment.

Descritpion

You are given two datasets representing the quality of data in New York:

- *new-york-air-quality.csv* is downloaded from EPA website (https://www.epa.gov/aqs) and contains data daily measurement from 2014 to 2025 for the major pollutants reported in a given city: PM2.5, Ozone (O3), Nitrogene Dioxine (NO2) and Carbon Monoxide (CO). **It does not contain the air quality index (AQI)**
- *daily_aqi_nyc.csv* is also downloaded from EPA website (https://aqs.epa.gov/aqsweb/airdata/download_files.html) but it contains pre-aggregated data in terms of AQI (Air Quality Index) and the defining parameter (PM, O3, NO2 or CO) from 2012 to 2024.

Eventually, you are **only interested in the AQI**. Your task is to:

1. Compute the AQI from the second dataset. To compute the AQI you need first to compute the AQI index for each pollutant and then take the one with highest value. These are the steps:
    a. **Breakpoints for each pollutant**: Using the table below "Table 1 – Pollutant conversion", compute the breakpoints for each pollutant (PM2.5, O3, NO2 and CO). For example, a PM2.5 of 46 corresponds to the third category "Unhealty for Sensitive Groups" and the two breakpoints are 35.5 and 55.4 and respective AQI breakpoints are 101 and 150.
    b. **Index for each pollutant:** Using the formula "Equation 1 – Index calculation", compute the AQI for each pollutant (PM2.5, O3, NO2 and CO). For example, using the two breakpoints above you have: ((150-101)/(55.4-35.5)) * (46-35.5) + 101 = **127 as AQI for PM2.5 pollutant.**
    c. **Final AQI:** Find the dominant pollutant, which is the one giving you the highest AQI. This is the final AQI you are interested in.

    For more information please see https://document.airnow.gov/technical-assistance-document-for-the-reporting-of-daily-air-quailty.pdf.
    Please check a complete example at the end of the document.

2. Check the differences between the AQI calculated from the first dataset with the one in the second dataset.
    a. Is there any difference?
    b. For which dates you note a difference?
    c. Are there any missing data?
3. Bonus: plots AQI data against time (at daily level and monthly / yearly aggregated).
    a. Which trend can you observe?

These Breakpoints…							**…equal this AQI**	**…and this category**
O_3 (ppm) 8-hour	O_3 (ppm) 1-hour[1]	$PM_{2.5}$ (µg/m³) 24-hour	PM_{10} (µg/m³) 24-hour	CO (ppm) 8-hour	SO_2 (ppb) 1-hour	NO_2 (ppb) 1-hour	AQI	
0.000 - 0.054	-	0.0 – 9.0	0 - 54	0.0 - 4.4	0 - 35	0 - 53	0 - 50	Good
0.055 - 0.070	-	9.1 – 35.4	55 - 154	4.5 - 9.4	36 - 75	54 - 100	51 - 100	Moderate
0.071 - 0.085	0.125 - 0.164	35.5 – 55.4	155 - 254	9.5 - 12.4	76 - 185	101 - 360	101 - 150	Unhealthy for Sensitive Groups
0.086 - 0.105	0.165 - 0.204	(55.5 - 125.4)[3]	255 - 354	12.5 - 15.4	[3] 186 - 304	361 - 649	151 - 200	Unhealthy
0.106 - 0.200	0.205 - 0.404	(125.5 - (225.4)[3]	355 - 424	15.5 - 30.4	[3] 305 - 604)	650 - 1249	201 - 300	Very unhealthy
0.201-([2])	0.405+	225.5+	425+	30.5+	[3] 605+	1250+	301+	Hazardous [4]

Table 1- Pollutant conversion

Equation 1:

$$I_p = \frac{I_{Hi} - I_{Lo}}{BP_{HI} - BP_{Lo}}(C_p - BP_{Lo}) + I_{Lo}.$$

Where I_p = the index for pollutant p

C_p = the truncated concentration of pollutant p

BP_{Hi} = the concentration breakpoint that is greater than or equal to C_p

BP_{Lo} = the concentration breakpoint that is less than or equal to C_p

I_{Hi} = the AQI value corresponding to BP_{Hi}

I_{Lo} = the AQI value corresponding to BP_{Lo}

Equation 1- Index calculation

However, you can compute the AQI from the first file using the following formula:

Guidelines

- Use **pandas** library;
- Do all the tests you want, but when you deliver your notebooks please clean all the useless cells and deliver a clean notebook.
- You can submit either a **.py** file or **.ipynb** file (Jupyter notebook)
- **Tips**:
  - To convert strings data into numeric use the *pd.to_numeric* function, e.g.,
    - *df_air_quality['pm25'] = pd.to_numeric(df_air_quality['pm25'])*
  - You can read dates directly using *pd.read_csv()* with *parse_dates* option, e.g.,
    - *pd.read_csv('your_file.csv', header=0, parse_dates=['date_column'])*

Grading criteria

Criteria	**Points**
Correct input parsing & storage	20
Accurate calculations	20
Functions & code modularity	20
Explanations, documentation and discussions	20
Plots	20
Total	**100**