Data Preprocessing

```
import pandas as pd
import os
from google.colab import drive

# Step 1: Mount Google Drive (Required if files are stored there)
drive.mount('/content/drive')

# Step 2: Set the main folder path (update this based on your Drive location)
main_folder = "/content/drive/My Drive/AQI"
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
# Step 3: Define the columns you need
required_columns = ['Timestamp', 'PM2.5 (µg/m³)', 'PM10 (µg/m³)', 'NO2 (µg/m³)', 'CO (mg/m³)', 'Ozone (µg/m³)']

# Step 4: Loop through all city folders and process CSV files
all_data = []

for city_folder in os.listdir(main_folder):
  city_path = os.path.join(main_folder, city_folder)

  if os.path.isdir(city_path):
      for filename in os.listdir(city_path):
          if filename.endswith(".csv"):
              file_path = os.path.join(city_path, filename)
              try:
                  # Read the CSV file with specified columns only
                  df = pd.read_csv(file_path, usecols=required_columns)

                  # Add a 'City' column to identify the source city
                  df['City'] = city_folder

                  # Add the city's data to the list
                  all_data.append(df)

              except pd.errors.ParserError as e:
                  print(f"Error parsing {file_path}: {e}")
              except FileNotFoundError:
                  print(f"File not found: {file_path}")
              except Exception as e:
                  print(f"An unexpected error occurred while processing {file_path}: {e}")


# Step 5: Concatenate all DataFrames into a single DataFrame
if all_data:
    combined_df = pd.concat(all_data, ignore_index=True)
    print(combined_df.head())  # Display the first few rows of the combined DataFrame
else:
    print("No data found in the specified folders.")
```

```
    Timestamp  PM2.5 (µg/m³)  PM10 (µg/m³)  NO2 (µg/m³)  CO (mg/m³)  \
0  2024-01-01         120.16        168.83        61.56        1.45
1  2024-01-02         104.79        146.30        44.64        1.35
2  2024-01-03          90.14        120.57        37.63        1.41
3  2024-01-04         114.69        161.47        39.49        1.54
4  2024-01-05         110.96        160.04        40.04        1.50

   Ozone (µg/m³)    City
0          63.08  Indore
1          53.47  Indore
2          49.11  Indore
3          38.37  Indore
4          44.21  Indore
```

```
# prompt: download this processed file

# Assuming 'combined_df' from the previous code is available

# Save the combined DataFrame to a CSV file in your Google Drive
combined_df.to_csv('/content/drive/My Drive/combined_aqi_data.csv', index=False)

# Download the file from Google Drive to your local machine
from google.colab import files
files.download('/content/drive/My Drive/combined_aqi_data.csv')
```

```
df = pd.read_csv('/content/drive/MyDrive/combined_aqi_data.csv')
df.head()
```

	Timestamp	PM2.5 (µg/m³)	PM10 (µg/m³)	NO2 (µg/m³)	CO (mg/m³)	Ozone (µg/m³)	City
0	2024-01-01	120.16	168.83	61.56	1.45	63.08	Indore
1	2024-01-02	104.79	146.30	44.64	1.35	53.47	Indore
2	2024-01-03	90.14	120.57	37.63	1.41	49.11	Indore
3	2024-01-04	114.69	161.47	39.49	1.54	38.37	Indore
4	2024-01-05	110.96	160.04	40.04	1.50	44.21	Indore

Next steps: Generate code with df View recommended plots New interactive sheet

```
# Rename columns
df = df.rename(columns={
    'Timestamp': 'Date',
    'PM2.5 (µg/m³)': 'PM2.5',
    'PM10 (µg/m³)': 'PM10',
    'NO2 (µg/m³)': 'NO2',
    'CO (mg/m³)': 'CO',
    'Ozone (µg/m³)': 'Ozone'
})

df.head()
```

	Date	PM2.5	PM10	NO2	CO	Ozone	City
0	2024-01-01	120.16	168.83	61.56	1.45	63.08	Indore
1	2024-01-02	104.79	146.30	44.64	1.35	53.47	Indore
2	2024-01-03	90.14	120.57	37.63	1.41	49.11	Indore
3	2024-01-04	114.69	161.47	39.49	1.54	38.37	Indore
4	2024-01-05	110.96	160.04	40.04	1.50	44.21	Indore

Next steps: Generate code with df View recommended plots New interactive sheet

```
# Remove rows with any null values
df = df.dropna()

# Reset the index after removing rows
df = df.reset_index(drop=True)

# Now you can further analyze the cleaned data
df.head()
```

	Date	PM2.5	PM10	NO2	CO	Ozone	City
0	2024-01-01	120.16	168.83	61.56	1.45	63.08	Indore
1	2024-01-02	104.79	146.30	44.64	1.35	53.47	Indore
2	2024-01-03	90.14	120.57	37.63	1.41	49.11	Indore
3	2024-01-04	114.69	161.47	39.49	1.54	38.37	Indore
4	2024-01-05	110.96	160.04	40.04	1.50	44.21	Indore

Next steps: Generate code with df View recommended plots New interactive sheet

```
# AQI Breakpoints for India
import numpy as np # import numpy library

aqi_breakpoints = {
    "PM2.5": [(0, 30, 0, 50), (31, 60, 51, 100), (61, 90, 101, 200), (91, 120, 201, 300), (121, 250, 301, 400), (251, 500, 401, 500)],
    "PM10": [(0, 50, 0, 50), (51, 100, 51, 100), (101, 250, 101, 200), (251, 350, 201, 300), (351, 430, 301, 400), (431, 600, 401, 500)
    "NO2": [(0, 40, 0, 50), (41, 80, 51, 100), (81, 180, 101, 200), (181, 280, 201, 300), (281, 400, 301, 400), (401, 1000, 401, 500)],
    "CO": [(0, 1, 0, 50), (1.1, 2, 51, 100), (2.1, 10, 101, 200), (10.1, 17, 201, 300), (17.1, 34, 301, 400), (34.1, 50, 401, 500)],
    "Ozone": [(0, 50, 0, 50), (51, 100, 51, 100), (101, 168, 101, 200), (169, 208, 201, 300), (209, 748, 301, 400), (749, 1000, 401, 50
}

# Function to calculate AQI sub-index
def compute_aqi_subindex(concentration, pollutant):
    for (bp_low, bp_high, i_low, i_high) in aqi_breakpoints[pollutant]:
        if bp_low <= concentration <= bp_high:
            return ((i_high - i_low) / (bp_high - bp_low)) * (concentration - bp_low) + i_low
    return np.nan  # If out of range # np refers to numpy here

# Calculate AQI for each pollutant
df["AQI_PM2.5"] = df["PM2.5"].apply(lambda x: compute_aqi_subindex(x, "PM2.5"))
df["AQI_PM10"] = df["PM10"].apply(lambda x: compute_aqi_subindex(x, "PM10"))
df["AQI_NO2"] = df["NO2"].apply(lambda x: compute_aqi_subindex(x, "NO2"))
df["AQI_CO"] = df["CO"].apply(lambda x: compute_aqi_subindex(x, "CO"))
df["AQI_Ozone"] = df["Ozone"].apply(lambda x: compute_aqi_subindex(x, "Ozone"))

# Final AQI (max of all sub-indices)
df["AQI"] = df[["AQI_PM2.5", "AQI_PM10", "AQI_NO2", "AQI_CO", "AQI_Ozone"]].max(axis=1)
```

```
df.head()
```

```
         Date   PM2.5    PM10    NO2    CO  Ozone   City   AQI_PM2.5    AQI_PM10    AQI_NO2     AQI_CO  AQI_Ozone         AQI
0  2024-01-01  120.16  168.83  61.56  1.45  63.08  Indore         NaN  146.068255  76.831795  70.055556      63.08  146.068255
1  2024-01-02  104.79  146.30  44.64  1.35  53.47  Indore  248.076207  131.098658  55.573333  64.611111      53.47  248.076207
2  2024-01-03   90.14  120.57  37.63  1.41  49.11  Indore         NaN  114.002886  47.037500  67.877778      49.11  114.002886
3  2024-01-04  114.69  161.47  39.49  1.54  38.37  Indore  281.872759  141.178054  49.362500  74.955556      38.37  281.872759
4  2024-01-05  110.96  160.04  40.04  1.50  44.21  Indore  269.139310  140.227919        NaN  72.777778      44.21  269.139310
```

Next steps: Generate code with df View recommended plots New interactive sheet

```
df.to_csv("AQICities22-25.csv", index=False)
```

```
df = df.dropna()

# Reset the index after removing rows
df = df.reset_index(drop=True)

# Now you can further analyze the cleaned data
df.head()
```

```
         Date   PM2.5    PM10    NO2    CO  Ozone   City   AQI_PM2.5    AQI_PM10    AQI_NO2     AQI_CO  AQI_Ozone         AQI
0  2024-01-02  104.79  146.30  44.64  1.35  53.47  Indore  248.076207  131.098658  55.573333  64.611111      53.47  248.076207
1  2024-01-04  114.69  161.47  39.49  1.54  38.37  Indore  281.872759  141.178054  49.362500  74.955556      38.37  281.872759
2  2024-01-06   88.81  122.08  48.94  1.53  43.09  Indore  195.937586  115.006174  60.975897  74.411111      43.09  195.937586
3  2024-01-07   66.61   93.93  46.25  1.25  30.89  Indore  120.151379   93.930000  57.596154  59.166667      30.89  120.151379
4  2024-01-08   94.56  141.99  51.93  1.64  49.13  Indore  213.153103  128.234966  64.732564  80.400000      49.13  213.153103
```

Next steps: Generate code with df View recommended plots New interactive sheet

```
df.to_csv("AQICities22-25(1).csv", index=False)
```