CL-I 11 DMV

July 20, 2025

[ ]:
```
''' NAME:Aher Swami Sandip
ROLL NO.01
COURSE: AI&DS
CLASS: BE
SUB:Computer Laboratory-I (DMV) '''
```

[ ]:
```
Data Visualization using matplotlib
Problem Statement: Analyzing Air Quality Index (AQI) Trends in a City
Dataset: "City_Air_Quality.csv"
Description: The dataset contains information about air quality measurements in␣
↪a specific
city over a period of time. It includes attributes such as date, time,␣
↪pollutant levels (e.g., PM2.5,
PM10, CO), and the Air Quality Index (AQI) values. The goal is to use the␣
↪matplotlib library
to create visualizations that effectively represent the AQI trends and patterns␣
↪for different
pollutants in the city.
Tasks to Perform:
1. Import the "City_Air_Quality.csv" dataset.
2. Explore the dataset to understand its structure and content.
3. Identify the relevant variables for visualizing AQI trends, such as date,␣
↪pollutant levels,
and AQI values.
4. Create line plots or time series plots to visualize the overall AQI trend␣
↪over time.
5. Plot individual pollutant levels (e.g., PM2.5, PM10, CO) on separate line␣
↪plots to
visualize their trends over time.
6. Use bar plots or stacked bar plots to compare the AQI values across␣
↪different dates or
time periods.
7. Create box plots or violin plots to analyze the distribution of AQI values␣
↪for different
pollutant categories.
8. Use scatter plots or bubble charts to explore the relationship between AQI␣
↪values and
```

```
pollutant levels.
9. Customize the visualizations by adding labels, titles, legends, and␣
 ↪appropriate color
schemes
```

[5]:
```
#1. Import the "City_Air_Quality.csv" dataset.
```

[33]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

[2]:
```
df=pd.read_csv("city_day.csv")
```

[7]:
```
df
```

[7]:
```
                City        Date  PM2.5   PM10     NO    NO2    NOx    NH3  \
0          Ahmedabad  2015-01-01    NaN    NaN   0.92  18.22  17.15    NaN
1          Ahmedabad  2015-01-02    NaN    NaN   0.97  15.69  16.46    NaN
2          Ahmedabad  2015-01-03    NaN    NaN  17.40  19.30  29.70    NaN
3          Ahmedabad  2015-01-04    NaN    NaN   1.70  18.48  17.97    NaN
4          Ahmedabad  2015-01-05    NaN    NaN  22.10  21.42  37.76    NaN
…                …           …      …      …      …      …      …      …
29526  Visakhapatnam  2020-06-27  15.02  50.94   7.68  25.06  19.54  12.47
29527  Visakhapatnam  2020-06-28  24.38  74.09   3.42  26.06  16.53  11.99
29528  Visakhapatnam  2020-06-29  22.91  65.73   3.45  29.53  18.33  10.71
29529  Visakhapatnam  2020-06-30  16.64  49.97   4.05  29.26  18.80  10.03
29530  Visakhapatnam  2020-07-01  15.00  66.00   0.40  26.85  14.05   5.20

          CO    SO2      O3  Benzene  Toluene  Xylene   AQI    AQI_Bucket
0       0.92  27.64  133.36     0.00     0.02    0.00   NaN           NaN
1       0.97  24.55   34.06     3.68     5.50    3.77   NaN           NaN
2      17.40  29.07   30.70     6.80    16.40    2.25   NaN           NaN
3       1.70  18.59   36.08     4.43    10.14    1.00   NaN           NaN
4      22.10  39.33   39.31     7.01    18.89    2.78   NaN           NaN
…          …      …       …        …        …       …     …             …
29526   0.47   8.55   23.30     2.24    12.07    0.73  41.0          Good
29527   0.52  12.72   30.14     0.74     2.21    0.38  70.0  Satisfactory
29528   0.48   8.42   30.96     0.01     0.01    0.00  68.0  Satisfactory
29529   0.52   9.84   28.30     0.00     0.00    0.00  54.0  Satisfactory
29530   0.59   2.10   17.05      NaN      NaN     NaN  50.0          Good

[29531 rows x 16 columns]
```

[9]:
```
#2. Explore the dataset to understand its structure and content.
```

[9]:
```
df.head(5)
```

[9]:
```
        City        Date  PM2.5  PM10     NO    NO2    NOx  NH3     CO    SO2  \
0  Ahmedabad  2015-01-01    NaN   NaN   0.92  18.22  17.15  NaN   0.92  27.64
1  Ahmedabad  2015-01-02    NaN   NaN   0.97  15.69  16.46  NaN   0.97  24.55
2  Ahmedabad  2015-01-03    NaN   NaN  17.40  19.30  29.70  NaN  17.40  29.07
3  Ahmedabad  2015-01-04    NaN   NaN   1.70  18.48  17.97  NaN   1.70  18.59
4  Ahmedabad  2015-01-05    NaN   NaN  22.10  21.42  37.76  NaN  22.10  39.33

       O3  Benzene  Toluene  Xylene  AQI AQI_Bucket
0  133.36     0.00     0.02    0.00  NaN        NaN
1   34.06     3.68     5.50    3.77  NaN        NaN
2   30.70     6.80    16.40    2.25  NaN        NaN
3   36.08     4.43    10.14    1.00  NaN        NaN
4   39.31     7.01    18.89    2.78  NaN        NaN
```

[11]:
```
df.info
```

[11]:
```
<bound method DataFrame.info of                  City        Date  PM2.5   PM10
NO    NO2    NOx    NH3  \
0          Ahmedabad  2015-01-01    NaN    NaN   0.92  18.22  17.15    NaN
1          Ahmedabad  2015-01-02    NaN    NaN   0.97  15.69  16.46    NaN
2          Ahmedabad  2015-01-03    NaN    NaN  17.40  19.30  29.70    NaN
3          Ahmedabad  2015-01-04    NaN    NaN   1.70  18.48  17.97    NaN
4          Ahmedabad  2015-01-05    NaN    NaN  22.10  21.42  37.76    NaN
…                …           …    …    …     …     …     …     …
29526  Visakhapatnam  2020-06-27  15.02  50.94   7.68  25.06  19.54  12.47
29527  Visakhapatnam  2020-06-28  24.38  74.09   3.42  26.06  16.53  11.99
29528  Visakhapatnam  2020-06-29  22.91  65.73   3.45  29.53  18.33  10.71
29529  Visakhapatnam  2020-06-30  16.64  49.97   4.05  29.26  18.80  10.03
29530  Visakhapatnam  2020-07-01  15.00  66.00   0.40  26.85  14.05   5.20

          CO    SO2      O3  Benzene  Toluene  Xylene   AQI    AQI_Bucket
0       0.92  27.64  133.36     0.00     0.02    0.00   NaN           NaN
1       0.97  24.55   34.06     3.68     5.50    3.77   NaN           NaN
2      17.40  29.07   30.70     6.80    16.40    2.25   NaN           NaN
3       1.70  18.59   36.08     4.43    10.14    1.00   NaN           NaN
4      22.10  39.33   39.31     7.01    18.89    2.78   NaN           NaN
…        …     …      …        …      …       …     …             …
29526   0.47   8.55   23.30     2.24    12.07    0.73  41.0          Good
29527   0.52  12.72   30.14     0.74     2.21    0.38  70.0  Satisfactory
29528   0.48   8.42   30.96     0.01     0.01    0.00  68.0  Satisfactory
29529   0.52   9.84   28.30     0.00     0.00    0.00  54.0  Satisfactory
29530   0.59   2.10   17.05      NaN      NaN     NaN  50.0          Good

[29531 rows x 16 columns]>
```

[13]:
```python
df.describe()
```

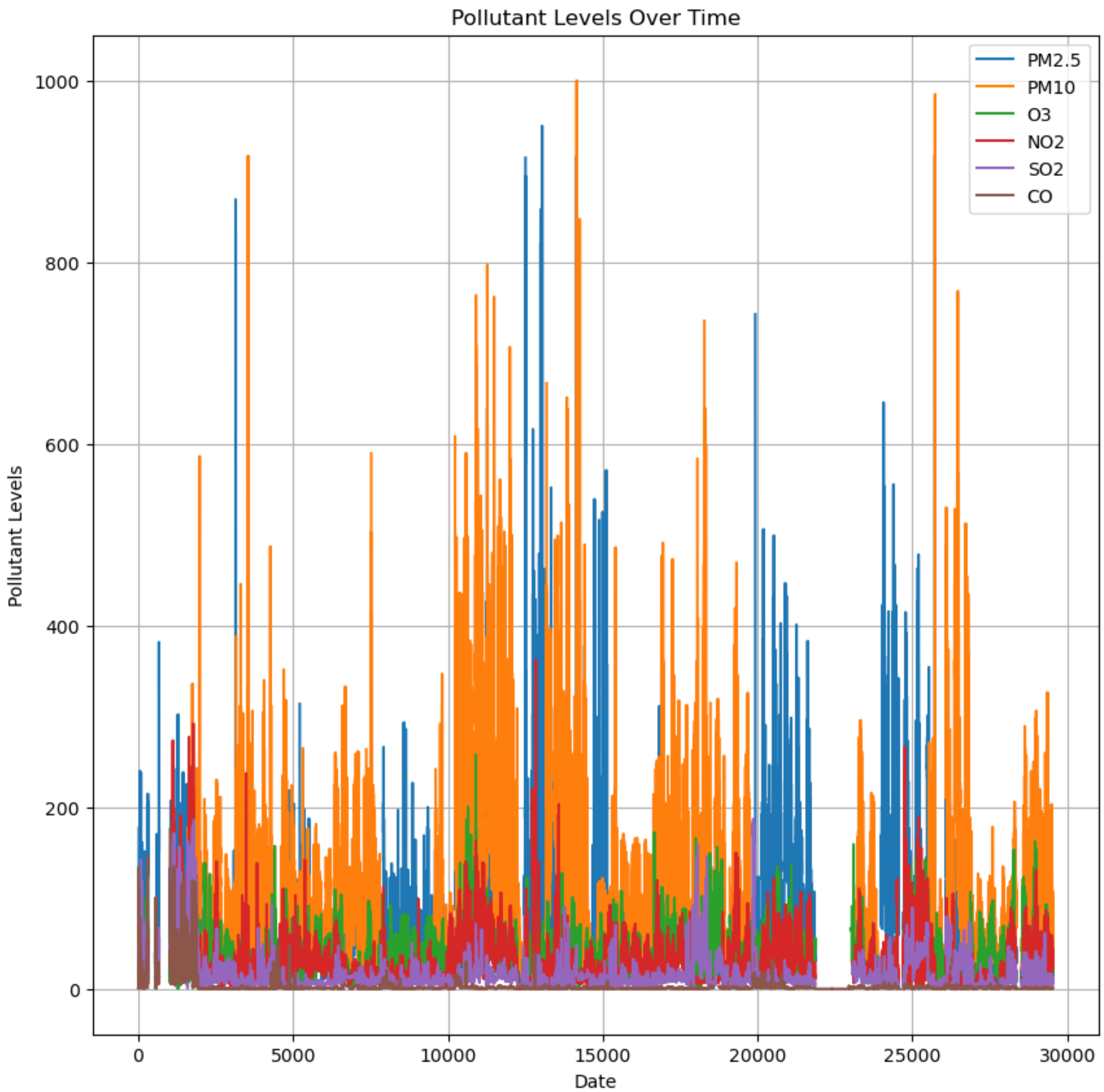
[13]:
```
              PM2.5          PM10            NO           NO2           NOx  \
count  24933.000000  18391.000000  25949.000000  25946.000000  25346.000000
mean      67.450578    118.127103     17.574730     28.560659     32.309123
std       64.661449     90.605110     22.785846     24.474746     31.646011
min        0.040000      0.010000      0.020000      0.010000      0.000000
25%       28.820000     56.255000      5.630000     11.750000     12.820000
50%       48.570000     95.680000      9.890000     21.690000     23.520000
75%       80.590000    149.745000     19.950000     37.620000     40.127500
max      949.990000   1000.000000    390.680000    362.210000    467.630000

                NH3            CO           SO2            O3       Benzene  \
count  19203.000000  27472.000000  25677.000000  25509.000000  23908.000000
mean      23.483476      2.248598     14.531977     34.491430      3.280840
std       25.684275      6.962884     18.133775     21.694928     15.811136
min        0.010000      0.000000      0.010000      0.010000      0.000000
25%        8.580000      0.510000      5.670000     18.860000      0.120000
50%       15.850000      0.890000      9.160000     30.840000      1.070000
75%       30.020000      1.450000     15.220000     45.570000      3.080000
max      352.890000    175.810000    193.860000    257.730000    455.030000

            Toluene        Xylene           AQI
count  21490.000000  11422.000000  24850.000000
mean       8.700972      3.070128    166.463581
std       19.969164      6.323247    140.696585
min        0.000000      0.000000     13.000000
25%        0.600000      0.140000     81.000000
50%        2.970000      0.980000    118.000000
75%        9.150000      3.350000    208.000000
max      454.850000    170.370000   2049.000000
```

[15]:
```python
df['Date'] = pd.to_datetime(df['Date'])
```

[14]:
```python
#3. Identify the relevant variables for visualizing AQI trends, such as date,␣
 ↪pollutant levels, and AQI values.
```

[17]:
```python
# Plotting pollutant levels
pollutants = ['PM2.5', 'PM10', 'O3', 'NO2', 'SO2', 'CO']

plt.figure(figsize=(10, 10))
for pollutant in pollutants:
    if pollutant in df.columns:
        plt.plot(df.index, df[pollutant], label=pollutant)

plt.xlabel('Date')
plt.ylabel('Pollutant Levels')
```

```
plt.title('Pollutant Levels Over Time')
plt.legend()
plt.grid(True)
plt.show()
```



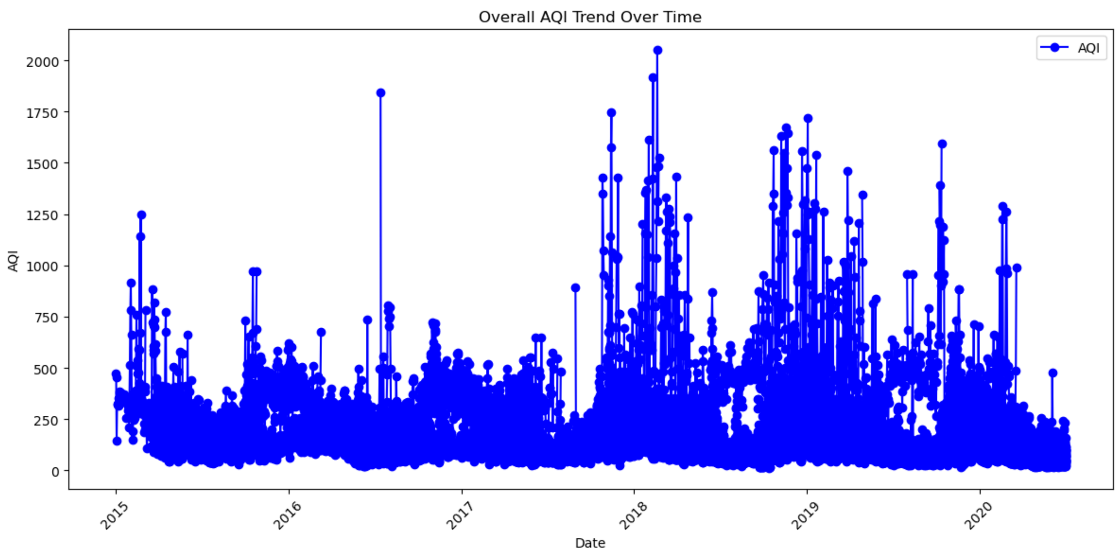
[19]:
```
df.columns
```

[19]:
```
Index(['City', 'Date', 'PM2.5', 'PM10', 'NO', 'NO2', 'NOx', 'NH3', 'CO', 'SO2',
       'O3', 'Benzene', 'Toluene', 'Xylene', 'AQI', 'AQI_Bucket'],
      dtype='object')
```

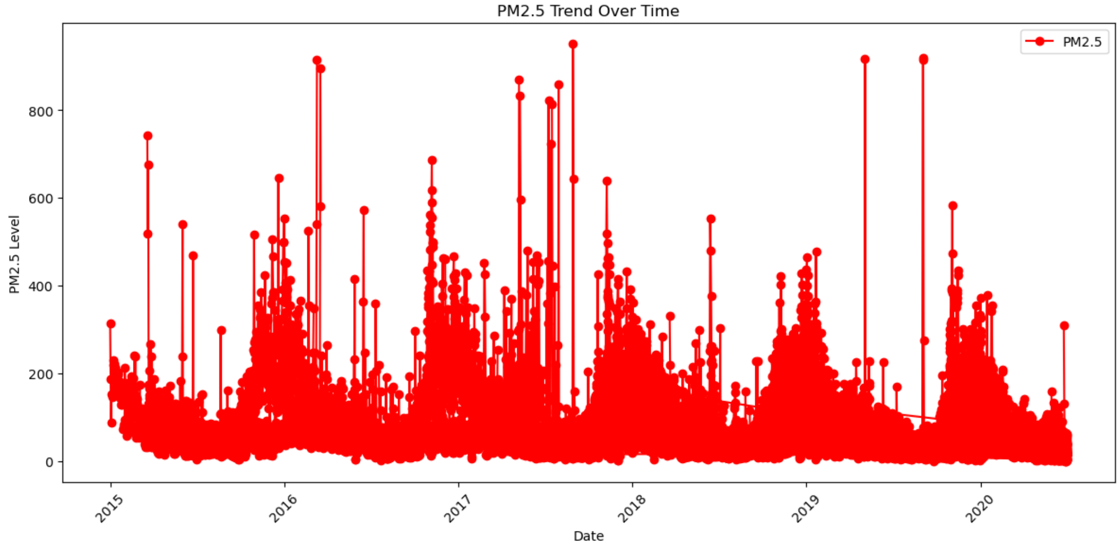
[17]:
```
#4. Create line plots or time series plots to visualize the overall AQI trend␣
↪over time.
```

```
#5. Plot individual pollutant levels (e.g., PM2.5, PM10, CO) on separate line␣
↪plots to visualize their trends over time.
```

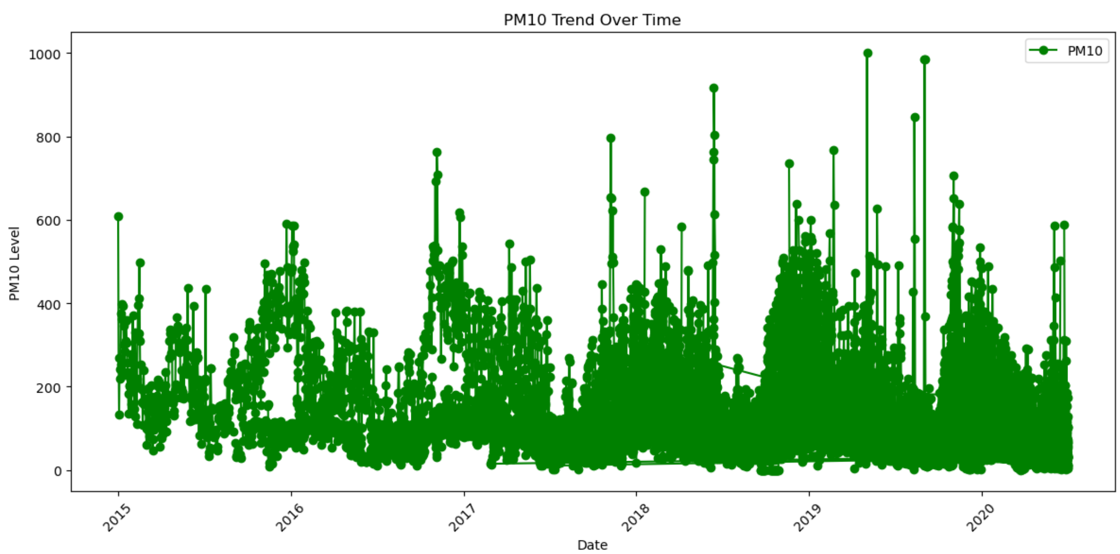
[21]:
```
plt.figure(figsize=(12, 6))
plt.plot(df['Date'], df['AQI'], marker='o', linestyle='-', color='b',␣
↪label='AQI')
plt.xlabel('Date')
plt.ylabel('AQI')
plt.title('Overall AQI Trend Over Time')
plt.xticks(rotation=45)
plt.legend()
plt.tight_layout()
plt.show()
```



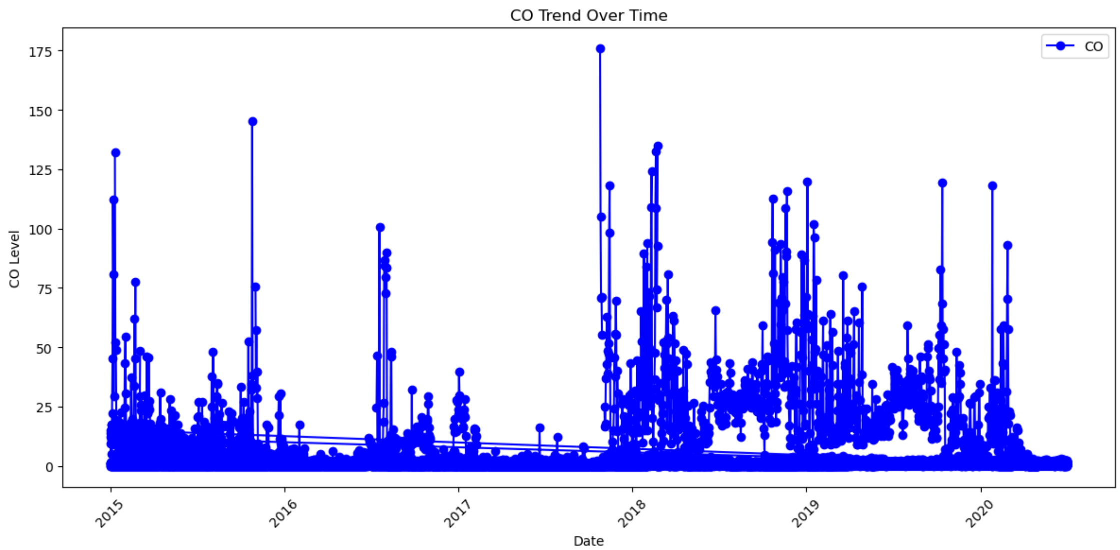
[19]:
```
plt.figure(figsize=(12, 6))
plt.plot(df['Date'], df['PM2.5'], marker='o', linestyle='-', color='r',␣
↪label='PM2.5')
plt.xlabel('Date')
plt.ylabel('PM2.5 Level')
plt.title('PM2.5 Trend Over Time')
plt.xticks(rotation=45)
plt.legend()
plt.tight_layout()
plt.show()
```



[23]:
```
plt.figure(figsize=(12, 6))
plt.plot(df['Date'], df['PM10'], marker='o', linestyle='-', color='g',␣
  ↪label='PM10')
plt.xlabel('Date')
plt.ylabel('PM10 Level')
plt.title('PM10 Trend Over Time')
plt.xticks(rotation=45)
plt.legend()
plt.tight_layout()
plt.show()
```

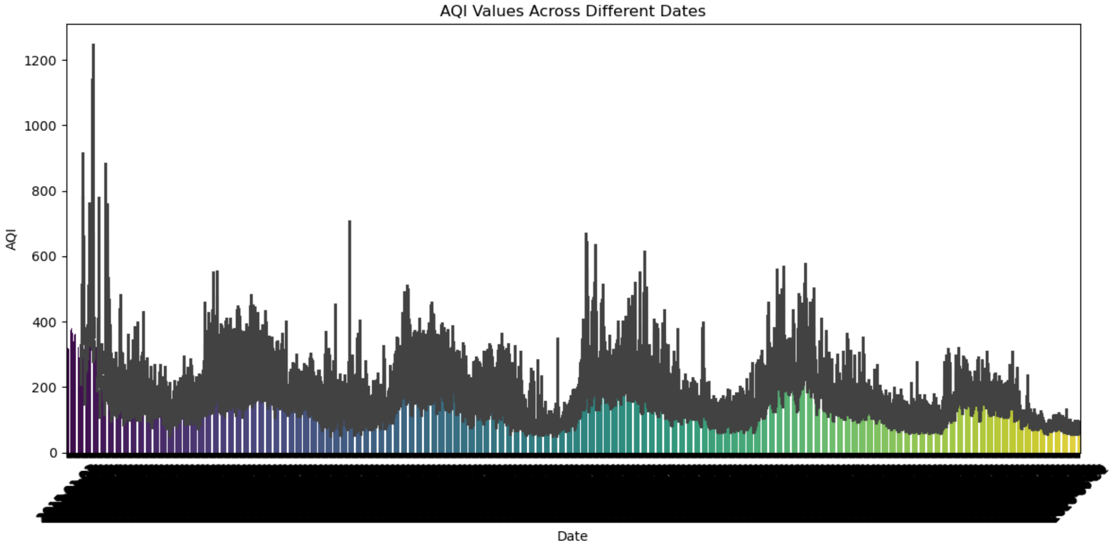


[25]:
```
plt.figure(figsize=(12, 6))
plt.plot(df['Date'], df['CO'], marker='o', linestyle='-', color='b', label='CO')
plt.xlabel('Date')
plt.ylabel('CO Level')
plt.title('CO Trend Over Time')
plt.xticks(rotation=45)
plt.legend()
plt.tight_layout()
plt.show()
```

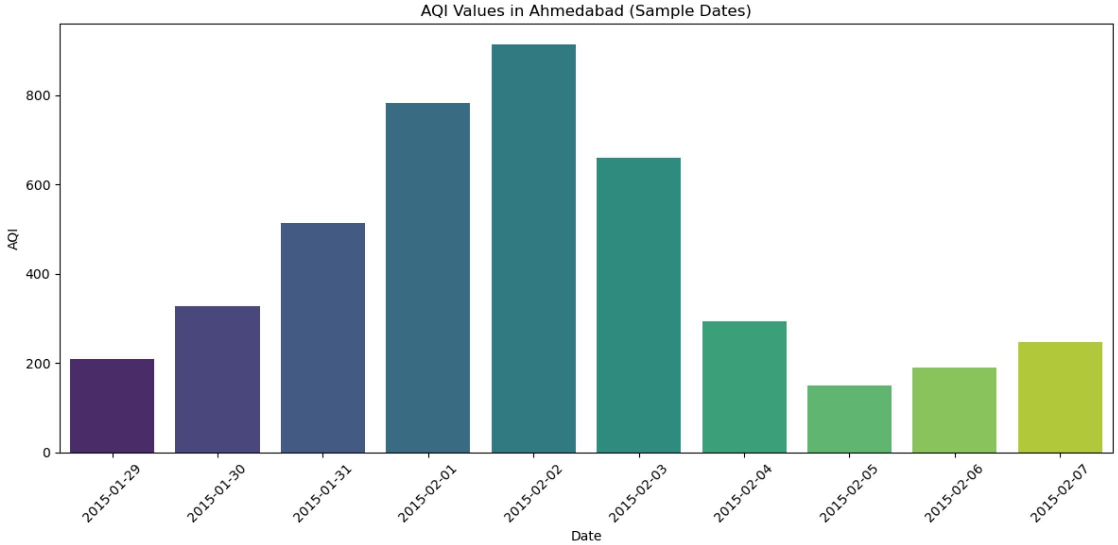


[22]:
```
#6. Use bar plots or stacked bar plots to compare the AQI values across␣
↪different dates or time periods.
```

[37]:
```
plt.figure(figsize=(12, 6))
# Assign 'Date' to hue to avoid deprecation warning, and suppress legend
sns.barplot(x='Date', y='AQI', data=df, hue='Date', palette='viridis',␣
↪dodge=False, legend=False)
plt.xlabel('Date')
plt.ylabel('AQI')
plt.title('AQI Values Across Different Dates')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

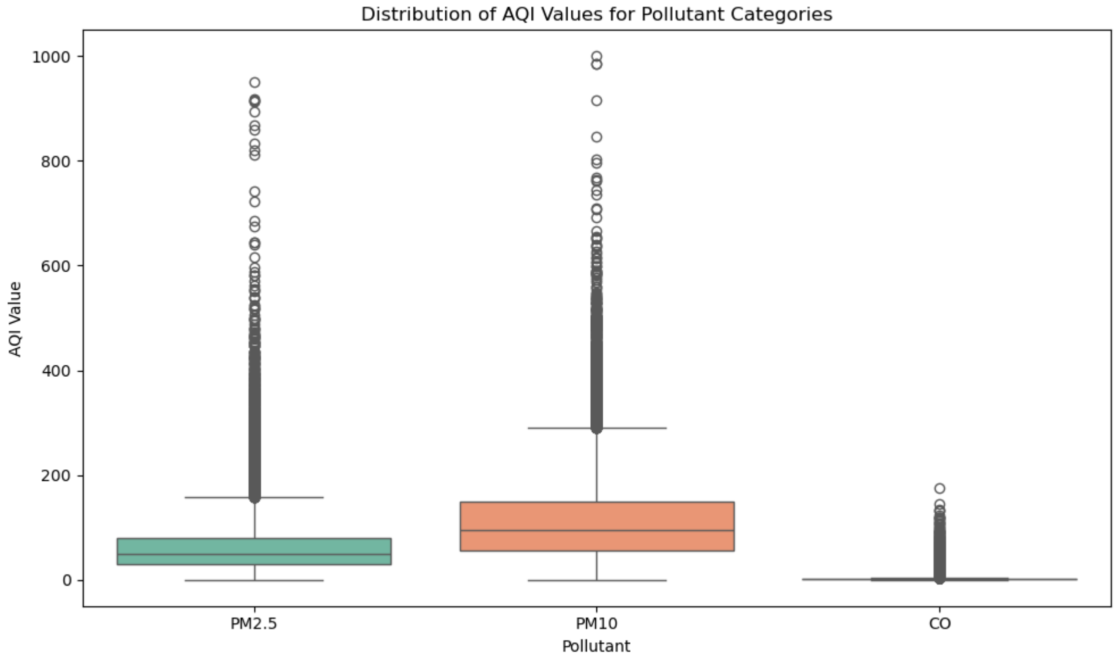


[45]:
```
df['Date'] = pd.to_datetime(df['Date'])
df = df.dropna(subset=['AQI'])

# Sample: Take data for a single city and 10 dates
df_sample = df[df['City'] == 'Ahmedabad'].sort_values('Date').head(10)

plt.figure(figsize=(12, 6))
sns.barplot(x='Date', y='AQI', data=df_sample, hue='Date', palette='viridis',␣
 ↪dodge=False, legend=False)
plt.xlabel('Date')
plt.ylabel('AQI')
plt.title('AQI Values in Ahmedabad (Sample Dates)')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



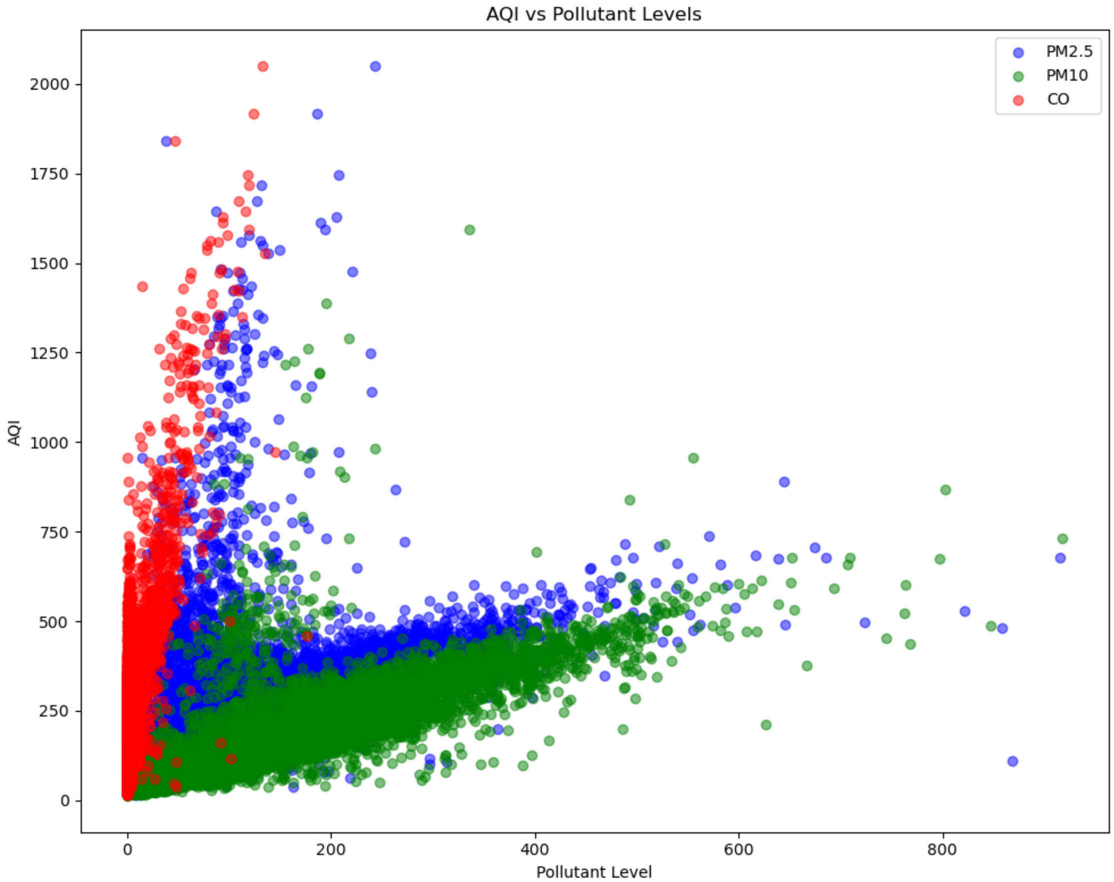
[29]:
```
#7. Create box plots or violin plots to analyze the distribution of AQI values␣
↪for different pollutant categories.
```

[30]:
```
plt.figure(figsize=(10, 6))
sns.boxplot(data=df[['PM2.5', 'PM10', 'CO']], palette='Set2')
plt.xlabel('Pollutant')
plt.ylabel('AQI Value')
plt.title('Distribution of AQI Values for Pollutant Categories')
plt.tight_layout()
plt.show()
```




```
[ ]: #8. Use scatter plots or bubble charts to explore the relationship between AQI␣
     ↪values and pollutant levels.
```

```
[31]: plt.figure(figsize=(10, 8))
      plt.scatter(df['PM2.5'], df['AQI'], color='b', alpha=0.5, label='PM2.5')
      plt.scatter(df['PM10'], df['AQI'], color='g', alpha=0.5, label='PM10')
      plt.scatter(df['CO'], df['AQI'], color='r', alpha=0.5, label='CO')
      plt.xlabel('Pollutant Level')
      plt.ylabel('AQI')
      plt.title('AQI vs Pollutant Levels')
      plt.legend()
      plt.tight_layout()
      plt.show()
```


AQI vs Pollutant Levels
PM2.5
PM10
CO
2000
1750
1500
1250
1000
750
500
250
0
AQI
0
200
400
600
800
Pollutant Level

```
[ ]:
```