Practical - 5

Data Visualization using matplotlib :

- ### Import Required Libraries :

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

- ### Load the dataset :

```
try:
    df = pd.read_csv('/content/AirQuality.csv', sep=';', decimal=',')
except FileNotFoundError:
    print("Make sure the 'AirQuality.csv' file is in the same
directory as your script.")
    exit()
```

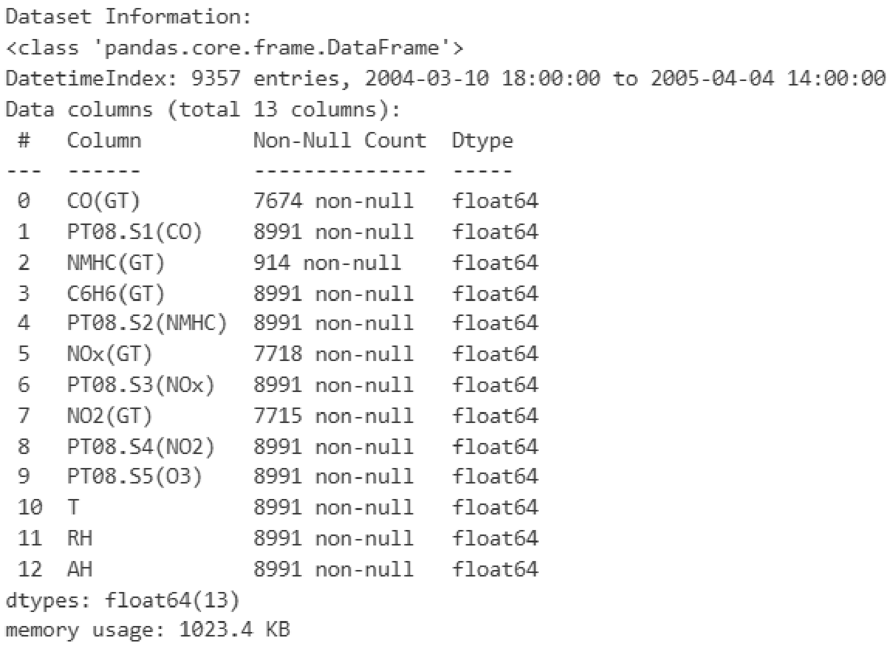
- ### Preprocessing the data :

```
# Drop empty columns that might be at the end of the file
df = df.dropna(axis=1, how='all')

# Clean up rows that are completely empty before processing
df.dropna(how='all', inplace=True)

# Ensure the column is treated as a string before replacing
df['Time'] = df['Time'].astype(str).str.replace('.', ':',
regex=False)

# Combine 'Date' and 'Time' into a single datetime column with the
correct format
df['DateTime'] = pd.to_datetime(df['Date'] + ' ' + df['Time'],
format='%d/%m/%Y %H:%M:%S')

# Set the new 'DateTime' column as the index
df.set_index('DateTime', inplace=True)

# Drop the original 'Date' and 'Time' columns
df.drop(['Date', 'Time'], axis=1, inplace=True)

# Replace the placeholder -200 with NaN (Not a Number) to
represent missing values
df.replace(to_replace=-200, value=np.nan, inplace=True)
```

- ### Exploring the dataset :

```
# Display basic information about the dataset
print("Dataset Information:")
df.info()
```

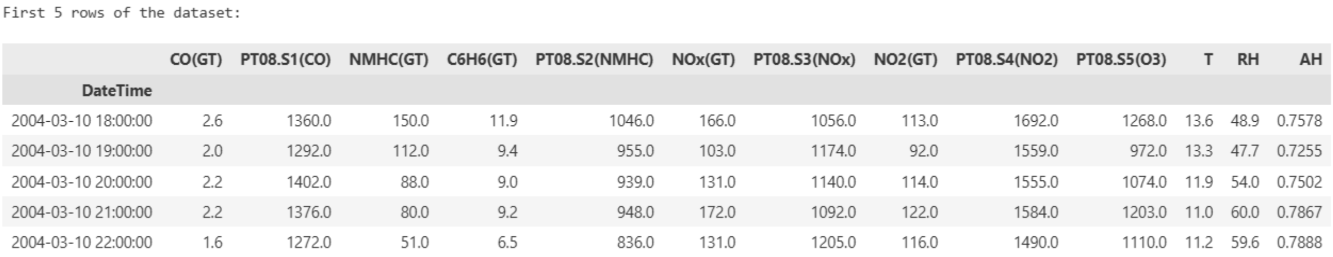
➢ Output -



```
Dataset Information:
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 9357 entries, 2004-03-10 18:00:00 to 2005-04-04 14:00:00
Data columns (total 13 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   CO(GT)         7674 non-null   float64
 1   PT08.S1(CO)    8991 non-null   float64
 2   NMHC(GT)       914 non-null    float64
 3   C6H6(GT)       8991 non-null   float64
 4   PT08.S2(NMHC)  8991 non-null   float64
 5   NOx(GT)        7718 non-null   float64
 6   PT08.S3(NOx)   8991 non-null   float64
 7   NO2(GT)        7715 non-null   float64
 8   PT08.S4(NO2)   8991 non-null   float64
 9   PT08.S5(O3)    8991 non-null   float64
 10  T              8991 non-null   float64
 11  RH             8991 non-null   float64
 12  AH             8991 non-null   float64
dtypes: float64(13)
memory usage: 1023.4 KB
```

```
# Display the first 5 rows of the dataset
print("\nFirst 5 rows of the dataset:")
df.head()
```

➢ Output -

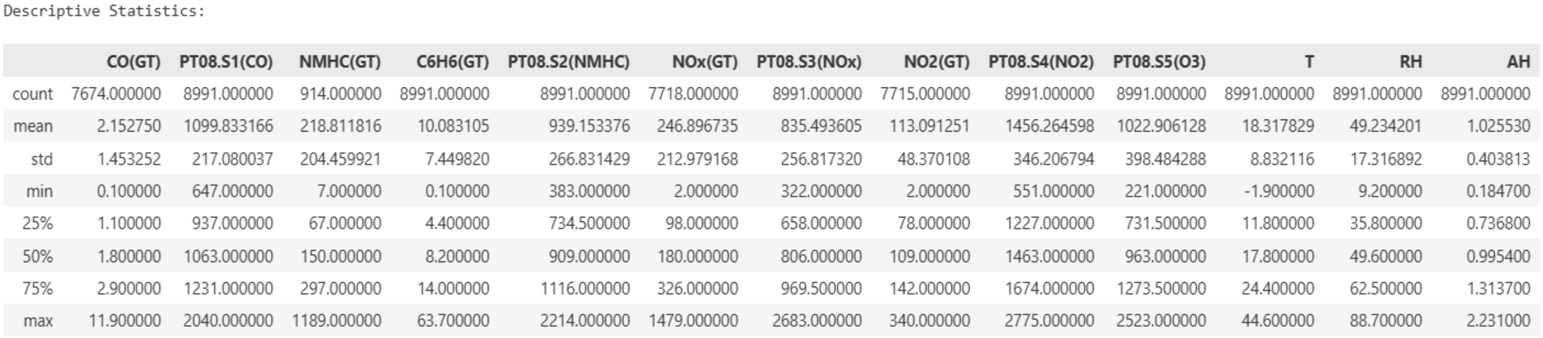


First 5 rows of the dataset:

	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O3)	T	RH	AH
DateTime													
2004-03-10 18:00:00	2.6	1360.0	150.0	11.9	1046.0	166.0	1056.0	113.0	1692.0	1268.0	13.6	48.9	0.7578
2004-03-10 19:00:00	2.0	1292.0	112.0	9.4	955.0	103.0	1174.0	92.0	1559.0	972.0	13.3	47.7	0.7255
2004-03-10 20:00:00	2.2	1402.0	88.0	9.0	939.0	131.0	1140.0	114.0	1555.0	1074.0	11.9	54.0	0.7502
2004-03-10 21:00:00	2.2	1376.0	80.0	9.2	948.0	172.0	1092.0	122.0	1584.0	1203.0	11.0	60.0	0.7867
2004-03-10 22:00:00	1.6	1272.0	51.0	6.5	836.0	131.0	1205.0	116.0	1490.0	1110.0	11.2	59.6	0.7888

```
# Display descriptive statistics
print("\nDescriptive Statistics:")
df.describe()
```

➢ Output :



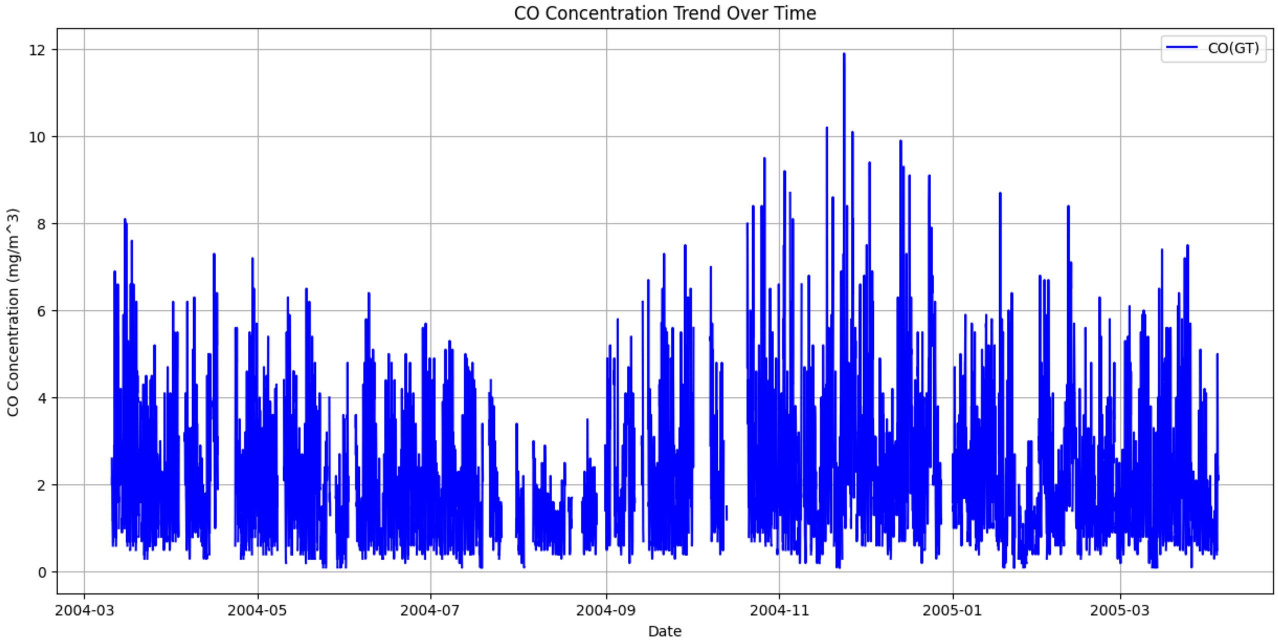
Descriptive Statistics:

	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O3)	T	RH	AH
count	7674.000000	8991.000000	914.000000	8991.000000	8991.000000	7718.000000	8991.000000	7715.000000	8991.000000	8991.000000	8991.000000	8991.000000	8991.000000
mean	2.152750	1099.833166	218.811816	10.083105	939.153376	246.896735	835.493605	113.091251	1456.264598	1022.906128	18.317829	49.234201	1.025530
std	1.453252	217.080037	204.459921	7.449820	266.831429	212.979168	256.817320	48.370108	346.206794	398.484288	8.832116	17.316892	0.403813
min	0.100000	647.000000	7.000000	0.100000	383.000000	2.000000	322.000000	2.000000	551.000000	221.000000	-1.900000	9.200000	0.184700
25%	1.100000	937.000000	67.000000	4.400000	734.500000	98.000000	658.000000	78.000000	1227.000000	731.500000	11.800000	35.800000	0.736800
50%	1.800000	1063.000000	150.000000	8.200000	909.000000	180.000000	806.000000	109.000000	1463.000000	963.000000	17.800000	49.600000	0.995400
75%	2.900000	1231.000000	297.000000	14.000000	1116.000000	326.000000	969.500000	142.000000	1674.000000	1273.500000	24.400000	62.500000	1.313700
max	11.900000	2040.000000	1189.000000	63.700000	2214.000000	1479.000000	2683.000000	340.000000	2775.000000	2523.000000	44.600000	88.700000	2.231000

- **Create a line plot for CO(GT) concentration over time :**

```
plt.figure(figsize=(15, 7))
plt.plot(df.index, df['CO(GT)'], label='CO(GT)', color='blue')

# Add titles and labels for clarity
plt.title('CO Concentration Trend Over Time')
plt.xlabel('Date')
plt.ylabel('CO Concentration (mg/m^3)')
plt.legend()
plt.grid(True)

# Show the plot
plt.show()
```

➢ **Output -**



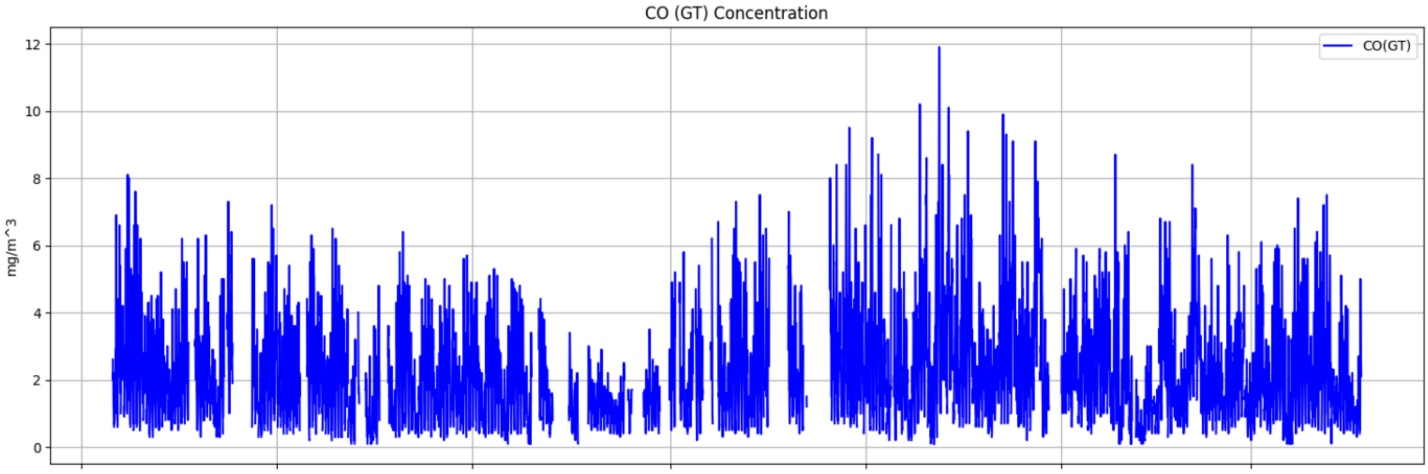
- **Create subplots for individual pollutants :**

```
fig, axes = plt.subplots(4, 1, figsize=(15, 20), sharex=True)
```

- **Plot for CO(GT) :**

```
axes[0].plot(df.index, df['CO(GT)'], label='CO(GT)', color='blue')
axes[0].set_title('CO (GT) Concentration')
axes[0].set_ylabel('mg/m^3')
axes[0].legend()
axes[0].grid(True)
plt.tight_layout()
plt.show()
```

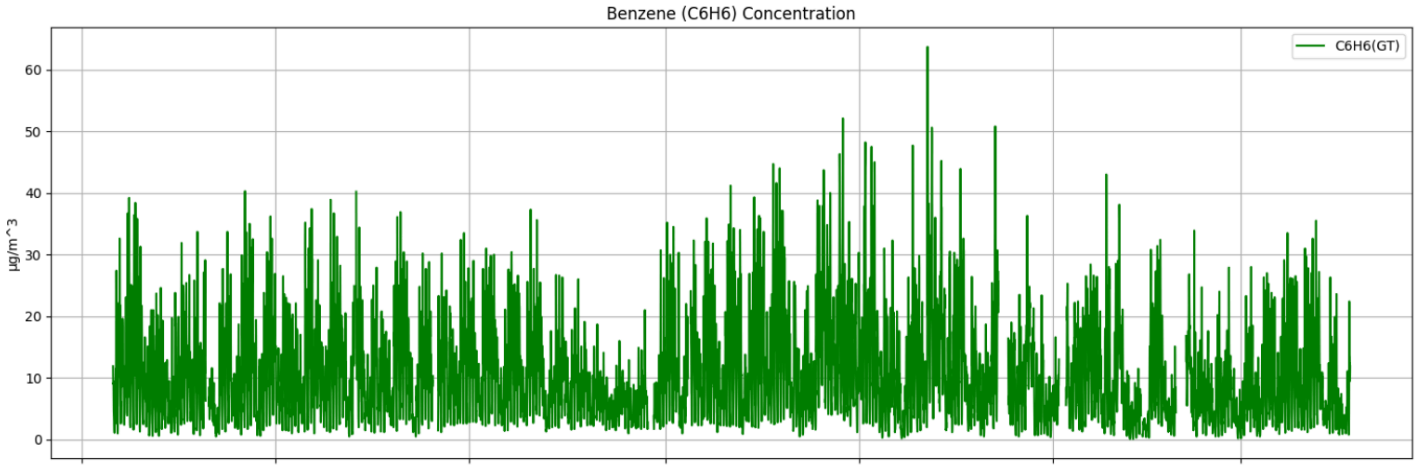
➢ **Output -**



- **Plot for C6H6(GT) :**

```
axes[1].plot(df.index, df['C6H6(GT)'], label='C6H6(GT)',
color='green')
axes[1].set_title('Benzene (C6H6) Concentration')
axes[1].set_ylabel('µg/m^3')
axes[1].legend()
axes[1].grid(True)
plt.tight_layout()
plt.show()
```

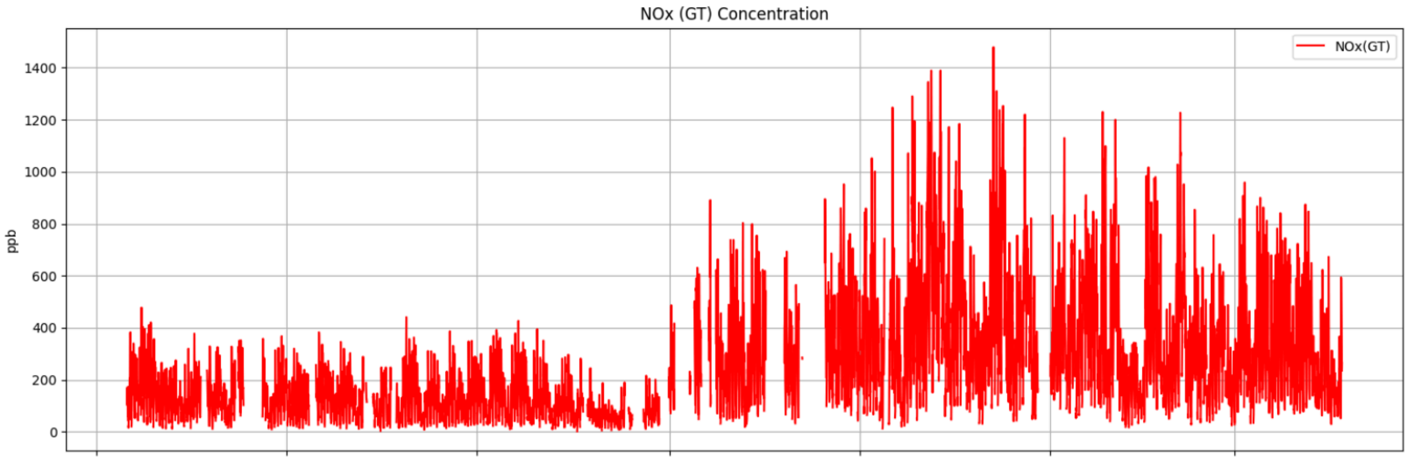
➢ **Output -**



- **Plot for NOx(GT) :**

```
axes[2].plot(df.index, df['NOx(GT)'], label='NOx(GT)',
color='red')
axes[2].set_title('NOx (GT) Concentration')
axes[2].set_ylabel('ppb')
axes[2].legend()
axes[2].grid(True)
plt.tight_layout()
plt.show()
```

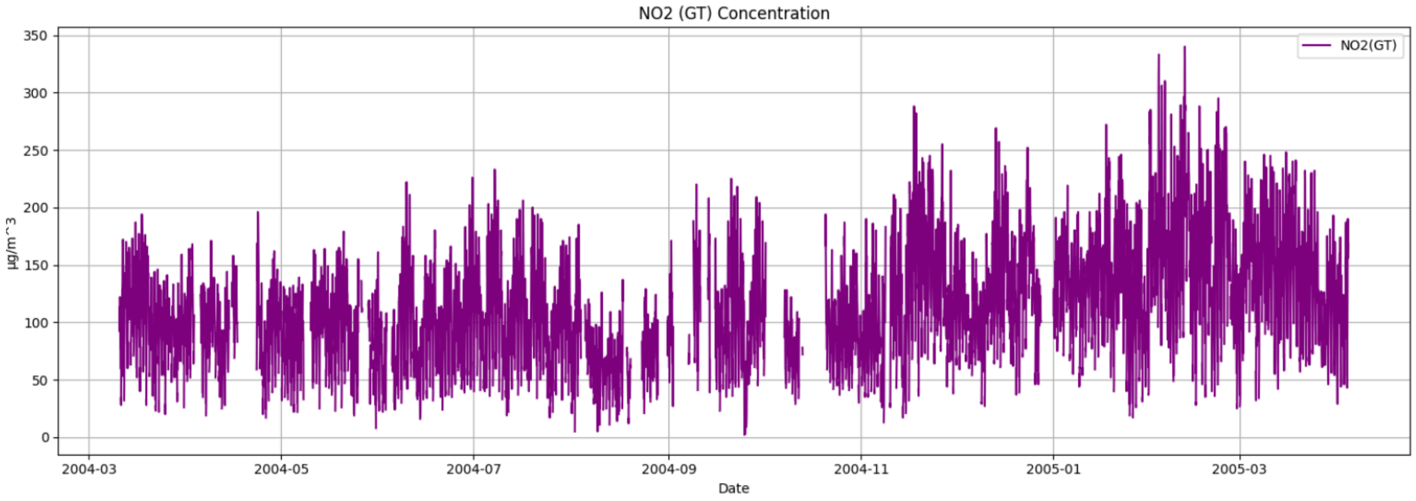
➢ **Output -**



- **Plot for NO2(GT) :**

```
axes[3].plot(df.index, df['NO2(GT)'], label='NO2(GT)',
color='purple')
axes[3].set_title('NO2 (GT) Concentration')
axes[3].set_xlabel('Date')
axes[3].set_ylabel('µg/m^3')
axes[3].legend()
axes[3].grid(True)
plt.tight_layout()
plt.show()
```

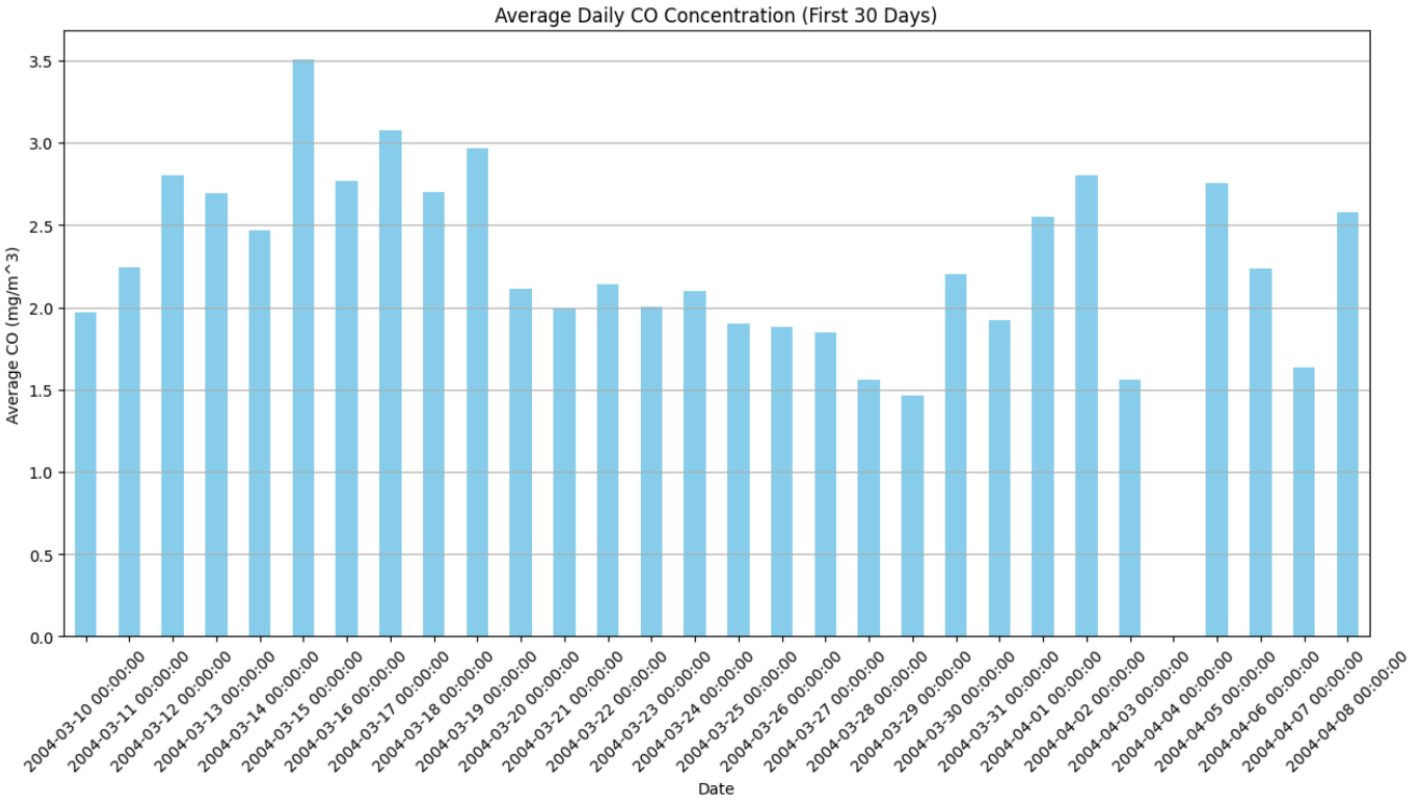
➢ **Output -**



- **Create a bar plot :**

```python
# Resample data to get the daily mean for CO(GT)
daily_co = df['CO(GT)'].resample('D').mean()

# Create a bar plot for the first 30 days
plt.figure(figsize=(15, 7))
daily_co.head(30).plot(kind='bar', color='skyblue')

# Add titles and labels
plt.title('Average Daily CO Concentration (First 30 Days)')
plt.xlabel('Date')
plt.ylabel('Average CO (mg/m^3)')
plt.xticks(rotation=45)
plt.grid(axis='y')

# Show the plot
plt.show()
```

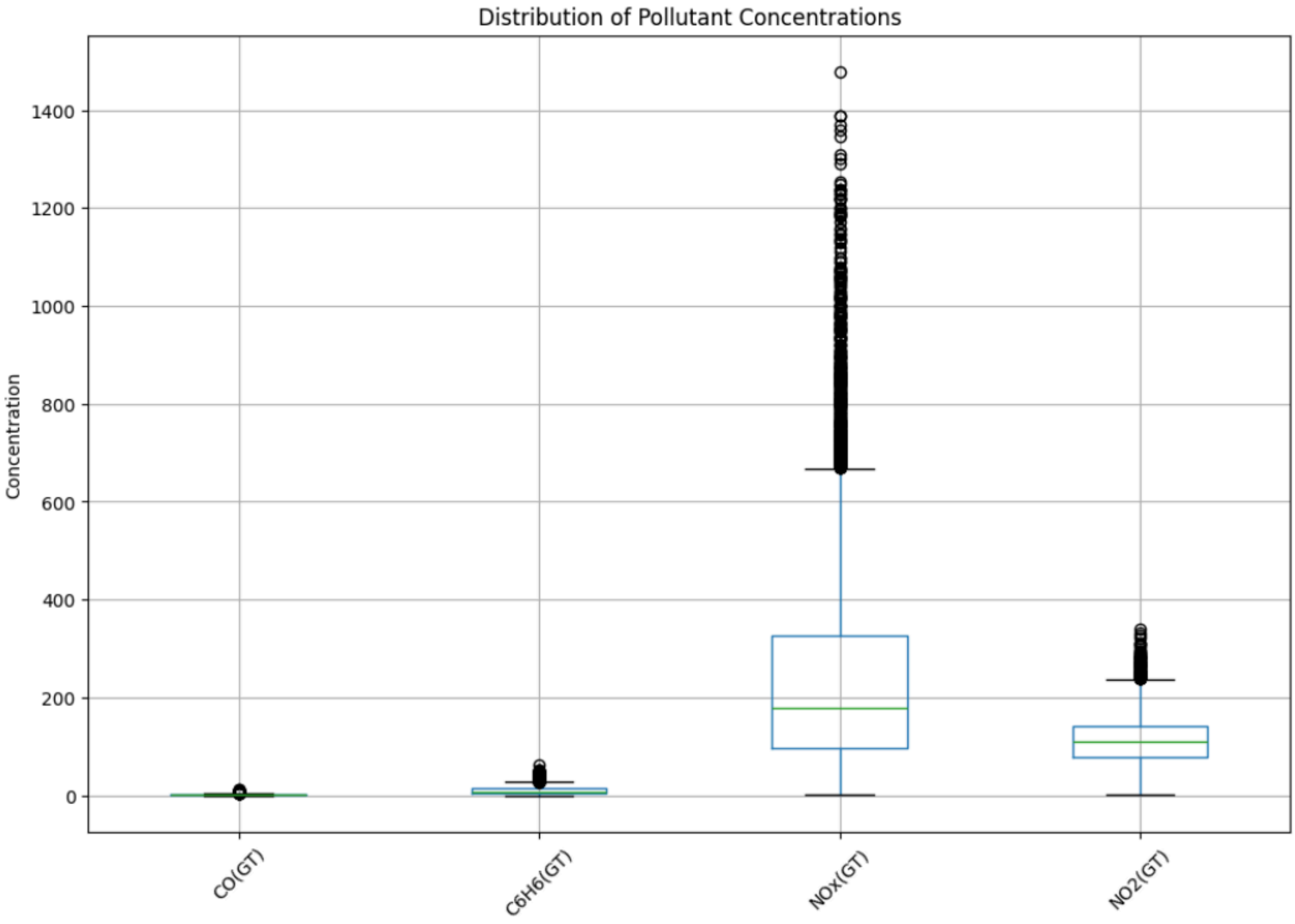
➢ **Output -**



- **Create the box plot :**

```
# Select a few pollutant columns for the box plot
pollutants = df[['CO(GT)', 'C6H6(GT)', 'NOx(GT)', 'NO2(GT)']]

# Create the box plot
plt.figure(figsize=(12, 8))
pollutants.boxplot()

# Add titles and labels
plt.title('Distribution of Pollutant Concentrations')
plt.ylabel('Concentration')
plt.xticks(rotation=45)

# Show the plot
plt.show()
```

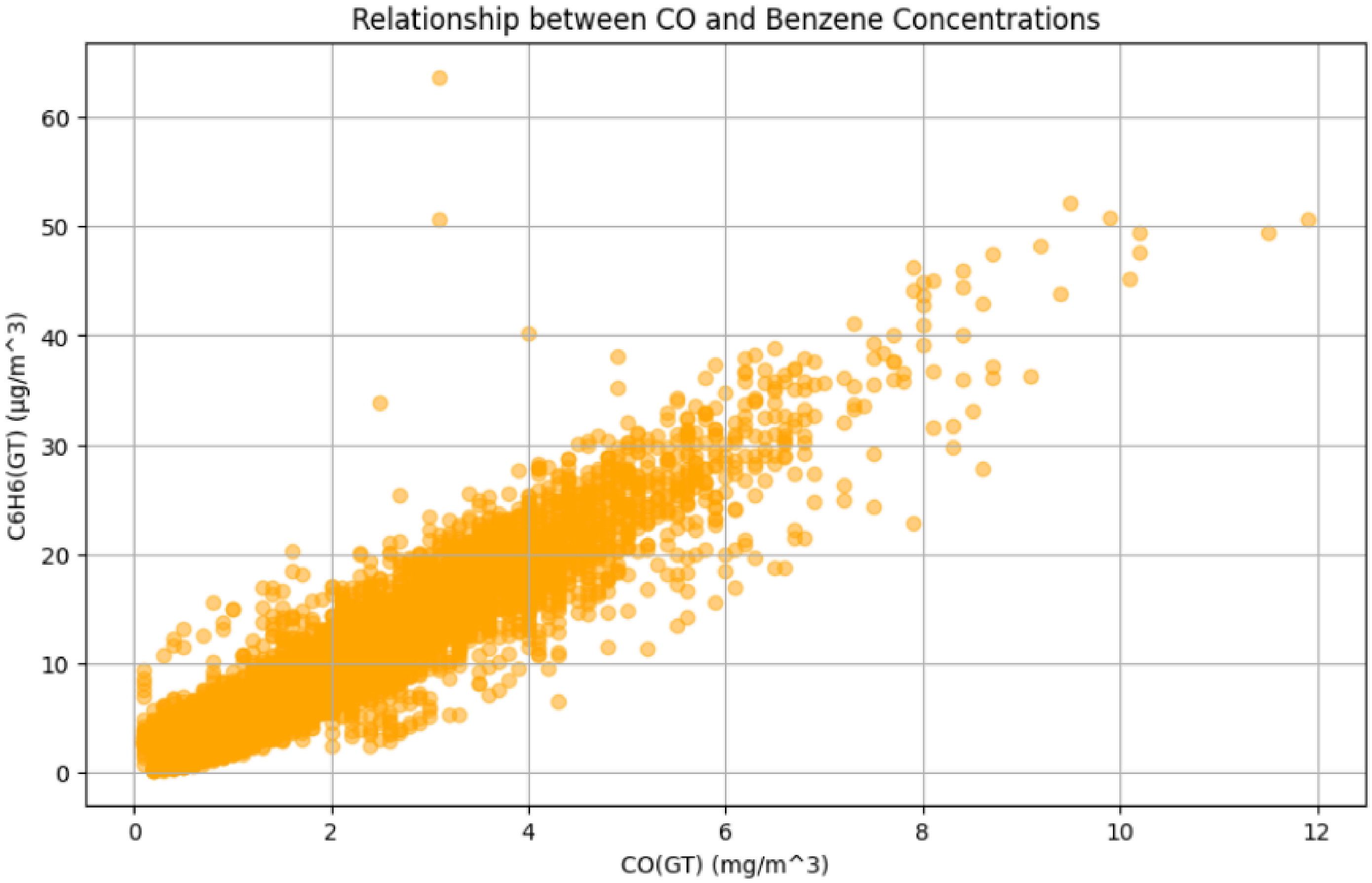
➢ **Output -**



- **Create a scatter plot :**

```
# Select a few pollutant columns for the box plot
pollutants = df[['CO(GT)', 'C6H6(GT)', 'NOx(GT)', 'NO2(GT)']]

# Create the box plot
plt.figure(figsize=(12, 8))
pollutants.boxplot()

# Add titles and labels
plt.title('Distribution of Pollutant Concentrations')
plt.ylabel('Concentration')
plt.xticks(rotation=45)

# Show the plot
plt.show()
```

➢ **Output -**


Relationship between CO and Benzene Concentrations
C6H6(GT) (µg/m^3)
CO(GT) (mg/m^3)