

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import geopandas as gpd
import geodatasets
%matplotlib inline
```

```
In [2]: df = pd.read_csv("city_day.csv")
df['Date'] = pd.to_datetime(df['Date'], errors='coerce')
df = df.dropna(subset=['AQI', 'City'])
df.head()
```

Out[2]:

	City	Date	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Toluene	Xylene	AQI	AQI_Bucket
28	Ahmedabad	2015-01-29	83.13	NaN	6.93	28.71	33.72	NaN	6.93	49.52	59.76	0.02	0.00	3.14	209.0	Poor
29	Ahmedabad	2015-01-30	79.84	NaN	13.85	28.68	41.08	NaN	13.85	48.49	97.07	0.04	0.00	4.81	328.0	Very Poor
30	Ahmedabad	2015-01-31	94.52	NaN	24.39	32.66	52.61	NaN	24.39	67.39	111.33	0.24	0.01	7.67	514.0	Severe
31	Ahmedabad	2015-02-01	135.99	NaN	43.48	42.08	84.57	NaN	43.48	75.23	102.70	0.40	0.04	25.87	782.0	Severe
32	Ahmedabad	2015-02-02	178.33	NaN	54.56	35.31	72.80	NaN	54.56	55.04	107.38	0.46	0.06	35.61	914.0	Severe

```
In [3]: print("Dataset Info:")
print(df.info())

print("\nNumber of Cities:", df['City'].nunique())
print("Cities Available:", df['City'].unique()[:10])
```

Dataset Info:

```
<class 'pandas.core.frame.DataFrame'>
Index: 24850 entries, 28 to 29530
Data columns (total 16 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   City        24850 non-null   object  
 1   Date        24850 non-null   datetime64[ns]
 2   PM2.5       24172 non-null   float64 
 3   PM10        17764 non-null   float64 
 4   NO          24463 non-null   float64 
 5   NO2         24459 non-null   float64 
 6   NOx         22993 non-null   float64 
 7   NH3         18314 non-null   float64 
 8   CO          24405 non-null   float64 
 9   SO2         24245 non-null   float64 
 10  O3          24043 non-null   float64 
 11  Benzene     21315 non-null   float64 
 12  Toluene     19024 non-null   float64 
 13  Xylene      9478 non-null   float64 
 14  AQI         24850 non-null   float64 
 15  AQI_Bucket  24850 non-null   object  
dtypes: datetime64[ns](1), float64(13), object(2)
memory usage: 3.2+ MB
None
```

Number of Cities: 26
Cities Available: ['Ahmedabad' 'Aizawl' 'Amaravati' 'Amritsar' 'Bengaluru' 'Bhopal'
'Brajrajnagar' 'Chandigarh' 'Chennai' 'Coimbatore']

```
In [4]: city_aqi = df.groupby('City')['AQI'].mean().sort_values(ascending=False)

print("Top 10 Most Polluted Cities:")
print(city_aqi.head(10))
```

Top 10 Most Polluted Cities:

City	AQI
Ahmedabad	452.122939
Delhi	259.487744
Patna	240.782042
Gurugram	225.123882
Lucknow	217.973059
Talcher	172.886819
Jorapokhar	159.251621
Brajrajnagar	150.280505
Kolkata	140.566313
Guwahati	140.111111

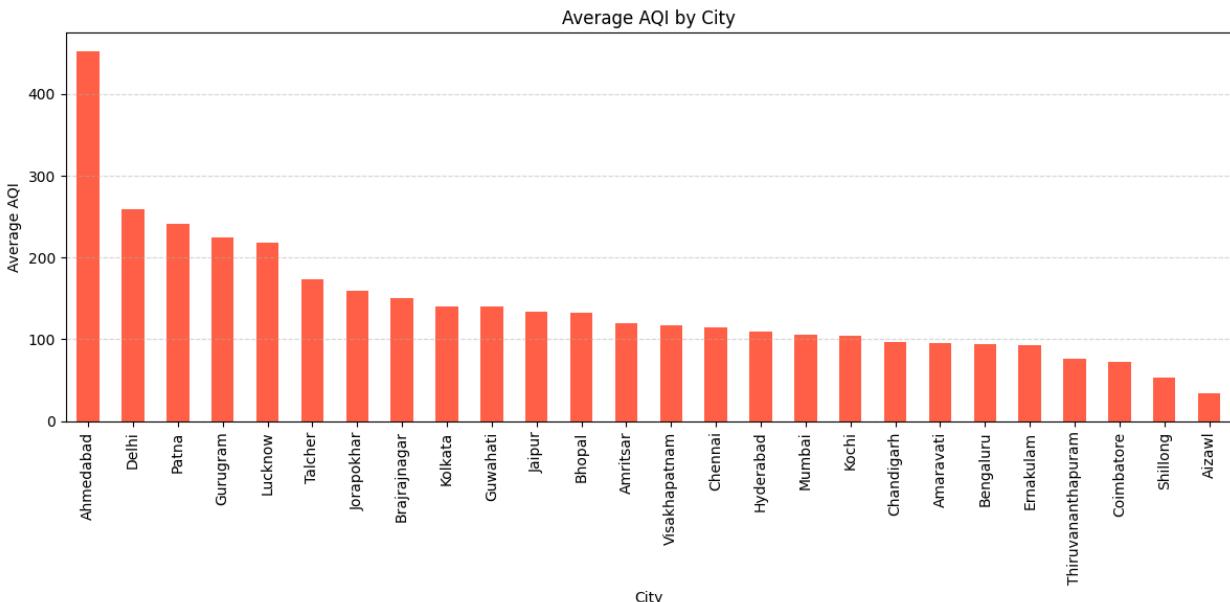
Name: AQI, dtype: float64

```
In [5]: plt.figure(figsize=(12,6))
city_aqi.plot(kind='bar', color='tomato')
plt.title("Average AQI by City")
```

```

plt.ylabel("Average AQI")
plt.xlabel("City")
plt.grid(axis='y', linestyle='--', alpha=0.5)
plt.tight_layout()
plt.show()

```

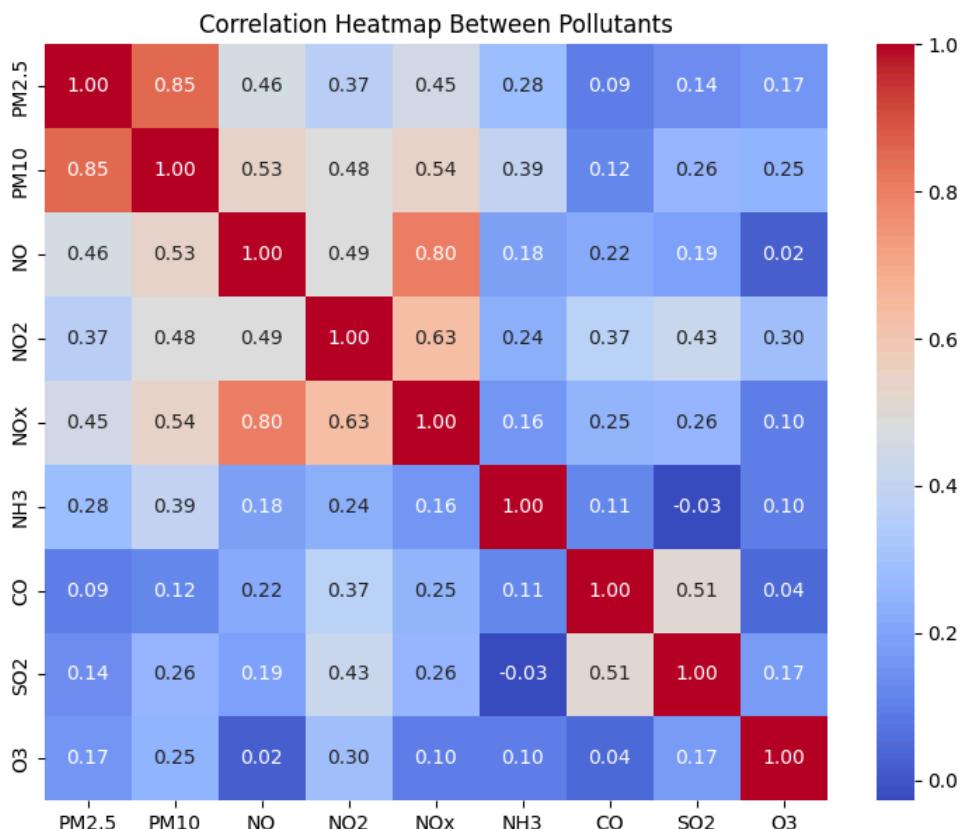


```

In [6]: pollutants = ['PM2.5', 'PM10', 'NO', 'NO2', 'NOx', 'NH3', 'CO', 'SO2', 'O3']
corr = df[pollutants].corr()

plt.figure(figsize=(9,7))
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Correlation Heatmap Between Pollutants")
plt.show()

```



```

In [7]: city_name = "Delhi"
city_data = df[df['City'] == city_name]

plt.figure(figsize=(10,5))
plt.plot(city_data['Date'], city_data['AQI'], color='black', linewidth=2)
plt.title(f"AQI Trend Over Time - {city_name}", fontsize=14, pad=20)
plt.xlabel("Date", fontsize=12)
plt.ylabel("AQI", fontsize=12)
plt.grid(True, linestyle='--', alpha=0.5)

max_aqi = city_data.loc[city_data['AQI'].idxmax()]

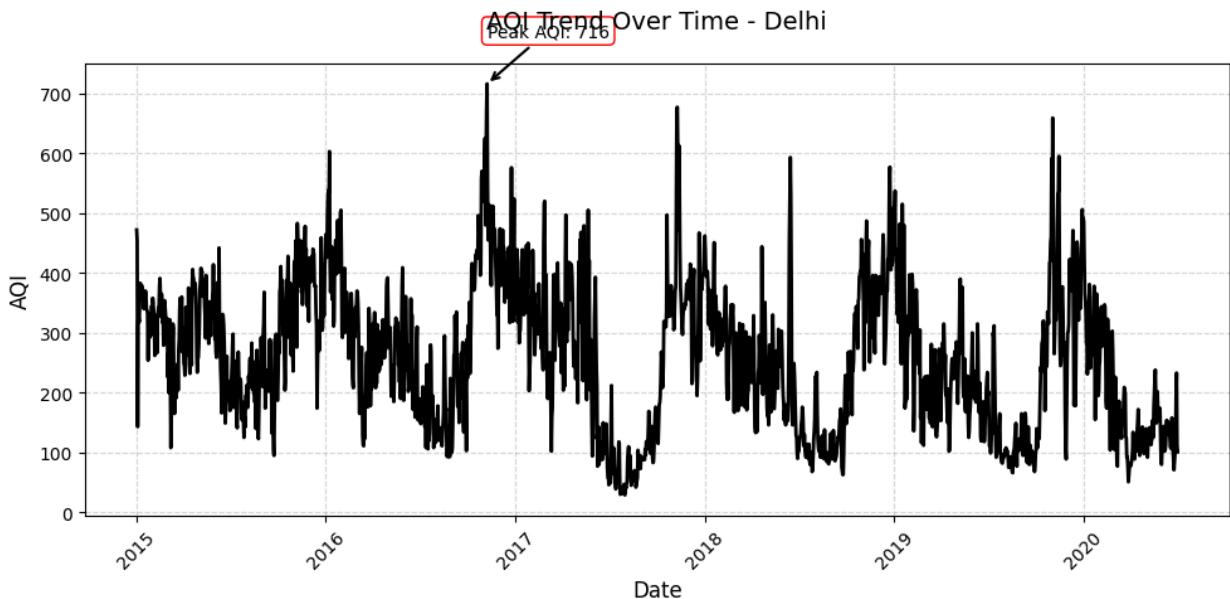
```

```

plt.annotate(f"Peak AQI: {int(max_aqi['AQI'])}",
            xy=(max_aqi['Date'], max_aqi['AQI']),
            xytext=(max_aqi['Date'], max_aqi['AQI'] + 80), # move higher
            textcoords='data',
            arrowprops=dict(facecolor='red', arrowstyle="->", lw=1.5),
            fontsize=10,
            bbox=dict(boxstyle="round,pad=0.3", fc="white", ec="red", lw=1))

plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

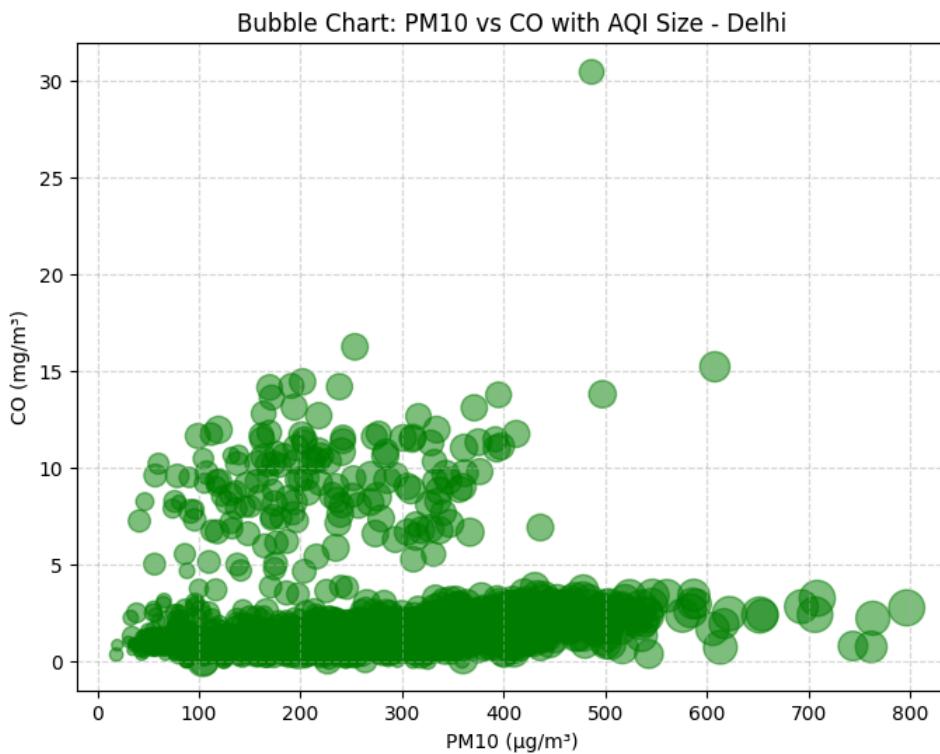
```



```

In [8]: plt.figure(figsize=(8,6))
plt.scatter(city_data['PM10'], city_data['CO'],
           s=city_data['AQI']*0.5, alpha=0.5, c='green')
plt.title(f"Bubble Chart: PM10 vs CO with AQI Size - {city_name}")
plt.xlabel("PM10 ( $\mu\text{g}/\text{m}^3$ )")
plt.ylabel("CO ( $\text{mg}/\text{m}^3$ )")
plt.grid(True, linestyle='--', alpha=0.5)
plt.show()

```



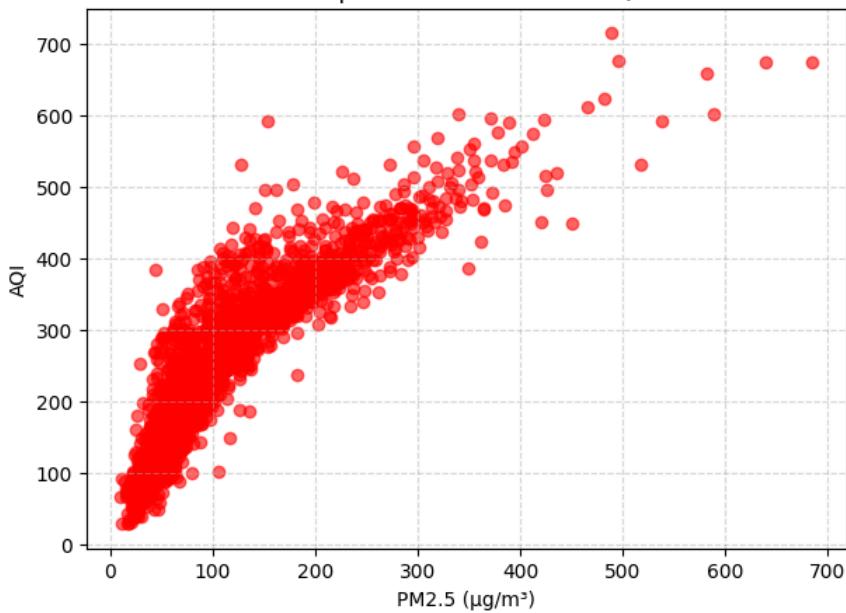
```

In [9]: plt.figure(figsize=(7,5))
plt.scatter(city_data['PM2.5'], city_data['AQI'], c='red', alpha=0.6)
plt.title(f"Relationship Between PM2.5 and AQI - {city_name}")
plt.xlabel("PM2.5 ( $\mu\text{g}/\text{m}^3$ )")
plt.ylabel("AQI")

```

```
plt.grid(True, linestyle='--', alpha=0.5)
plt.show()
```

Relationship Between PM2.5 and AQI - Delhi



```
In [10]: pollutants = ['PM2.5', 'PM10', 'NO2', 'SO2', 'CO']
plt.figure(figsize=(12,6))
for p in pollutants:
    if p in city_data.columns:
        plt.plot(city_data['Date'], city_data[p], label=p, alpha=0.8)
plt.title(f"Pollutant Levels Over Time - {city_name}")
plt.xlabel("Date")
plt.ylabel("Concentration ( $\mu\text{g}/\text{m}^3$ )")
plt.legend()
plt.grid(True, linestyle='--', alpha=0.5)
plt.show()
```

Pollutant Levels Over Time - Delhi

