👋 Hi, I'm A GOKULKRISHNA

📊 Internship in Data Science

🏢 at ShadowFox

AIR QUALITY ANALYSIS (DELHI)

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.simplefilter("ignore")
```

In [2]:
```
df = pd.read_csv(r'C:\Users\A GOKUL
KRISHNA\OneDrive\Desktop\ShadowFox\Shadowfox_DS\I
```

In [3]:
```
df.head()
```

Out[3]:

	date	co	no	no2	o3	so2	pm2_5	pm10	nh3
0	2023-01-01 00:00:00	1655.58	1.66	39.41	5.90	17.88	169.29	194.64	5.83
1	2023-01-01 01:00:00	1869.20	6.82	42.16	1.99	22.17	182.84	211.08	7.66
2	2023-01-01 02:00:00	2510.07	27.72	43.87	0.02	30.04	220.25	260.68	11.40
3	2023-01-01 03:00:00	3150.94	55.43	44.55	0.85	35.76	252.90	304.12	13.55
4	2023-01-01 04:00:00	3471.37	68.84	45.24	5.45	39.10	266.36	322.80	14.19

In [4]:
```
df.shape
```

Out[4]:
```
(561, 9)
```

In [5]:
```
df.dtypes
```

Out[5]:
```
date      object
co       float64
no       float64
no2      float64
o3       float64
so2      float64
pm2_5    float64
pm10     float64
nh3      float64
dtype: object
```

In [6]:
```
df.columns
```

Out[6]:
```
Index(['date', 'co', 'no', 'no2', 'o3', 'so2', 'pm2_5', 'pm10', 'nh3'], dtype='obj
ect')
```

In [7]:
```
df.size
```

Out[7]:
```
5049
```

In [8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 561 entries, 0 to 560
Data columns (total 9 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   date    561 non-null    object 
 1   co      561 non-null    float64
 2   no      561 non-null    float64
 3   no2     561 non-null    float64
 4   o3      561 non-null    float64
 5   so2     561 non-null    float64
 6   pm2_5   561 non-null    float64
 7   pm10    561 non-null    float64
 8   nh3     561 non-null    float64
dtypes: float64(8), object(1)
memory usage: 39.6+ KB
```

In [9]:
```
df.value_counts()
```

Out[9]:
```
date                 co       no     no2     o3      so2     pm2_5   pm10    nh3
2023-01-01 00:00:00  1655.58  1.66   39.41   5.90    17.88   169.29  194.64  5.83
1
2023-01-16 17:00:00  2857.21  8.72   80.20   1.31    41.48   211.19  274.45  31.92
1
2023-01-16 11:00:00  1949.31  11.51  74.03   67.23   58.65   135.85  172.38  22.80
1
2023-01-16 12:00:00  2670.29  15.65  111.04  18.24   59.13   163.88  211.14  29.13
1
2023-01-16 13:00:00  3257.75  28.16  117.90  0.11    60.08   194.19  251.70  36.98
1

..
2023-01-08 13:00:00  4005.43  32.19  124.75  0.44    39.10   370.36  425.01  16.47
1
2023-01-08 12:00:00  2990.72  3.74   112.41  28.97   39.10   327.78  365.97  14.31
1
2023-01-08 11:00:00  2590.18  5.59   76.77   86.55   46.73   325.19  357.19  15.07
1
2023-01-08 10:00:00  2136.23  4.92   50.04   131.61  57.22   308.40  332.44  12.92
1
2023-01-24 08:00:00  1134.87  8.61   56.89   80.11   110.63  123.76  140.26  5.51
1
Name: count, Length: 561, dtype: int64
```

In [10]:
```
display(df.describe().T)
```

	count	mean	std	min	25%	50%	75%	max
co	561.0	3814.942210	3227.744681	654.22	1708.98	2590.18	4432.68	16876.22
no	561.0	51.181979	83.904476	0.00	3.38	13.30	59.01	425.58
no2	561.0	75.292496	42.473791	13.37	44.55	63.75	97.33	263.21
o3	561.0	30.141943	39.979405	0.00	0.07	11.80	47.21	164.51
so2	561.0	64.655936	61.073080	5.25	28.13	47.21	77.25	511.17
pm2_5	561.0	358.256364	227.359117	60.10	204.45	301.17	416.65	1310.20
pm10	561.0	420.988414	271.287026	69.08	240.90	340.90	482.57	1499.27
nh3	561.0	26.425062	36.563094	0.63	8.23	14.82	26.35	267.51

In [11]:
```
df.isna()
```

Out[11]:

	date	co	no	no2	o3	so2	pm2_5	pm10	nh3
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...
556	False	False	False	False	False	False	False	False	False
557	False	False	False	False	False	False	False	False	False
558	False	False	False	False	False	False	False	False	False
559	False	False	False	False	False	False	False	False	False
560	False	False	False	False	False	False	False	False	False

561 rows × 9 columns

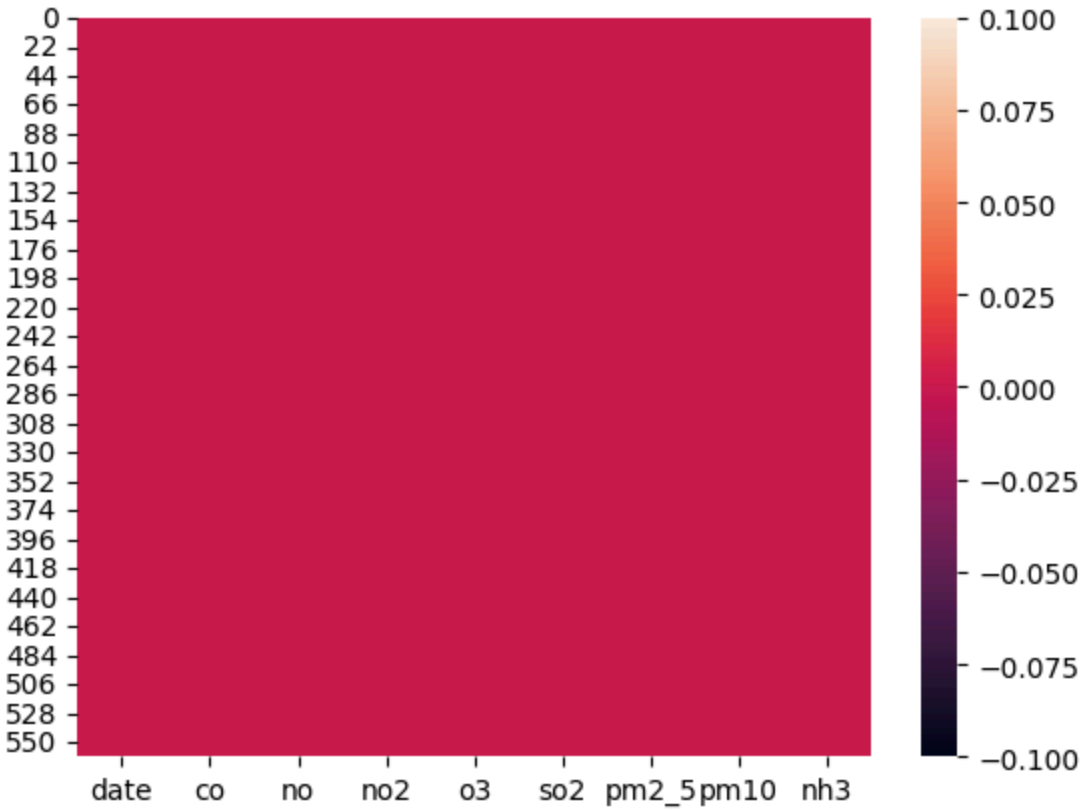
In [12]:
```
df.dropna(inplace=True)
```

In [13]:
```
df.isna().any()
```

Out[13]:
```
date     False
co       False
no       False
no2      False
o3       False
so2      False
pm2_5    False
pm10     False
nh3      False
dtype: bool
```

In [14]:
```
df.fillna(method='ffill',inplace=True)
sns.heatmap(df.isna(),cbar=True)
```

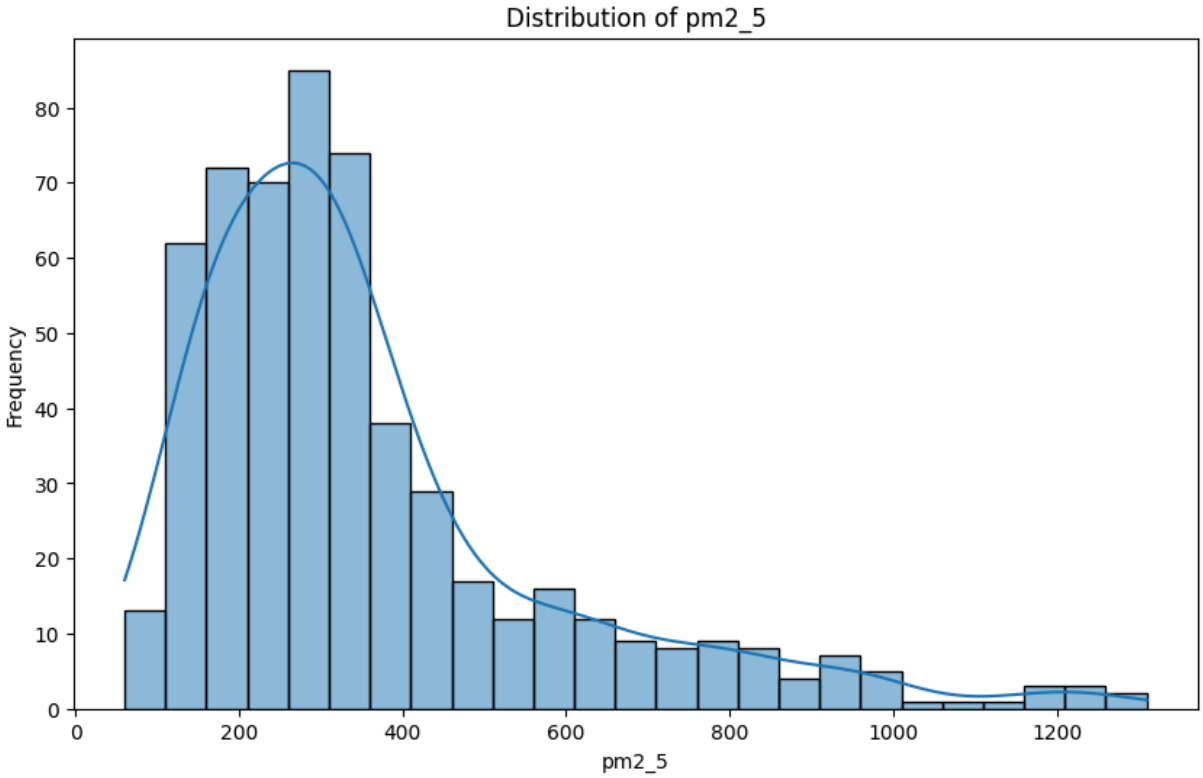
Out[14]: <Axes: >

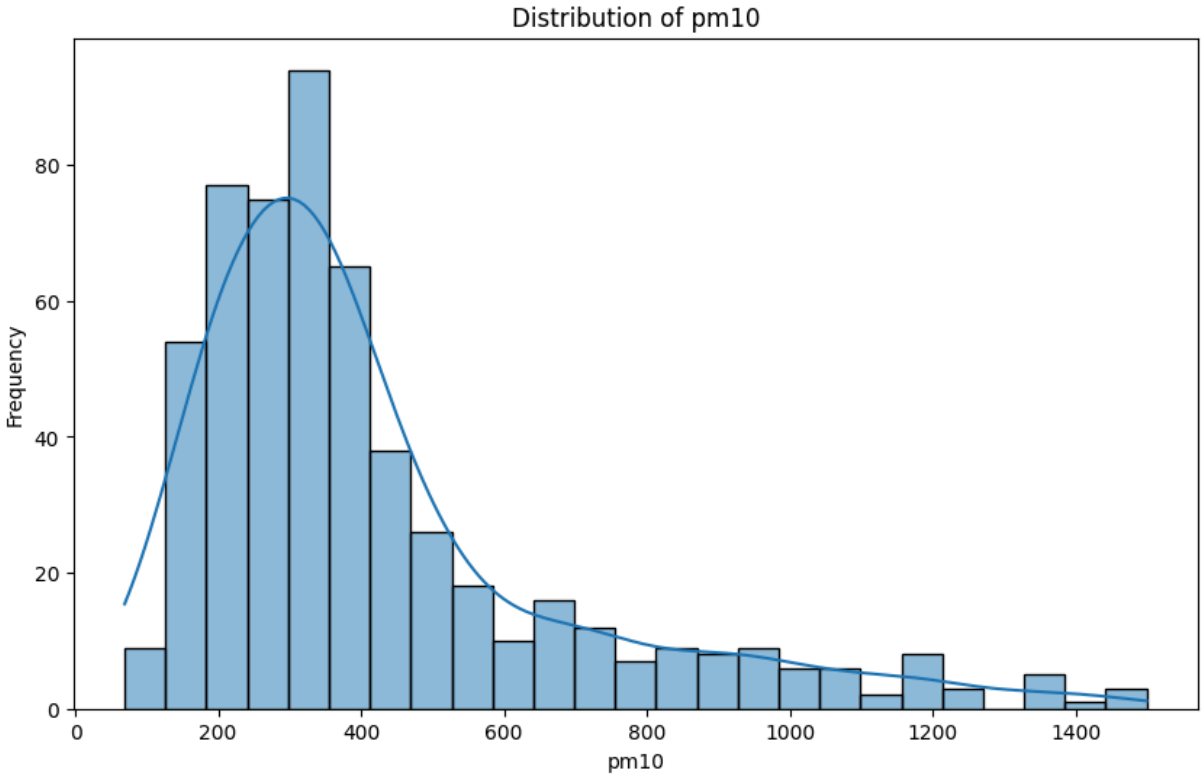


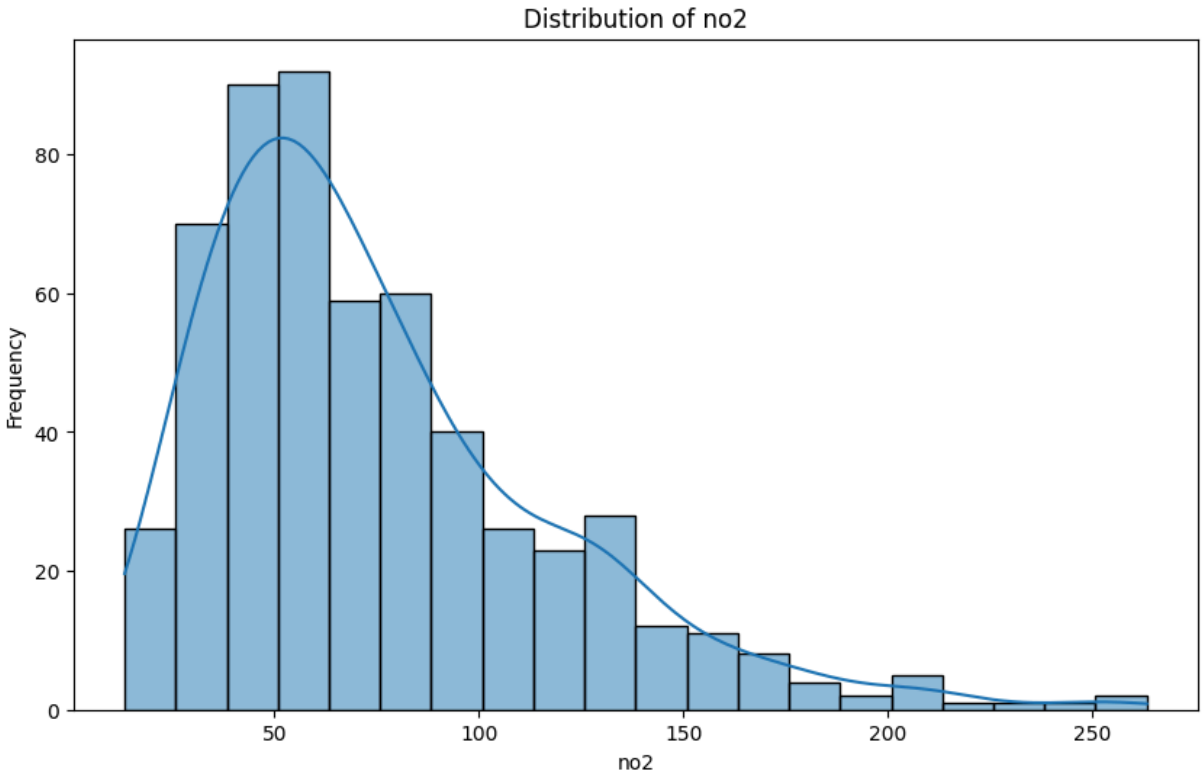
In [15]:
```
df['date'] = pd.to_datetime(df['date'])

def calculate_aqi(row):

    return max(row['pm2_5'], row['pm10'], row['no2'], row['o3'], row['co'], row['so

df['AQI'] = df.apply(calculate_aqi, axis=1)
```

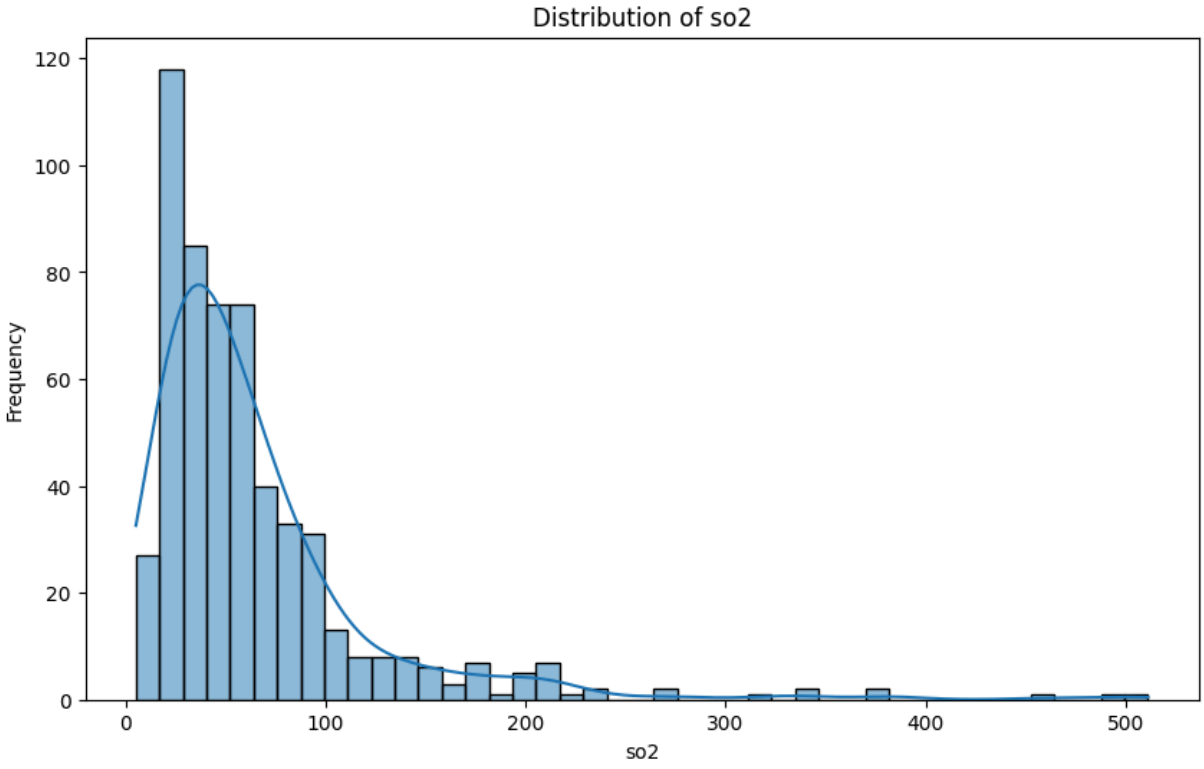
In [16]:
```
pollutants = ['pm2_5', 'pm10', 'no2', 'so2', 'co', 'o3', 'nh3', 'no', 'AQI']

for pollutant in pollutants:
    plt.figure(figsize=(10, 6))
    sns.histplot(df[pollutant].dropna(), kde=True)
    plt.title(f'Distribution of {pollutant}')
```

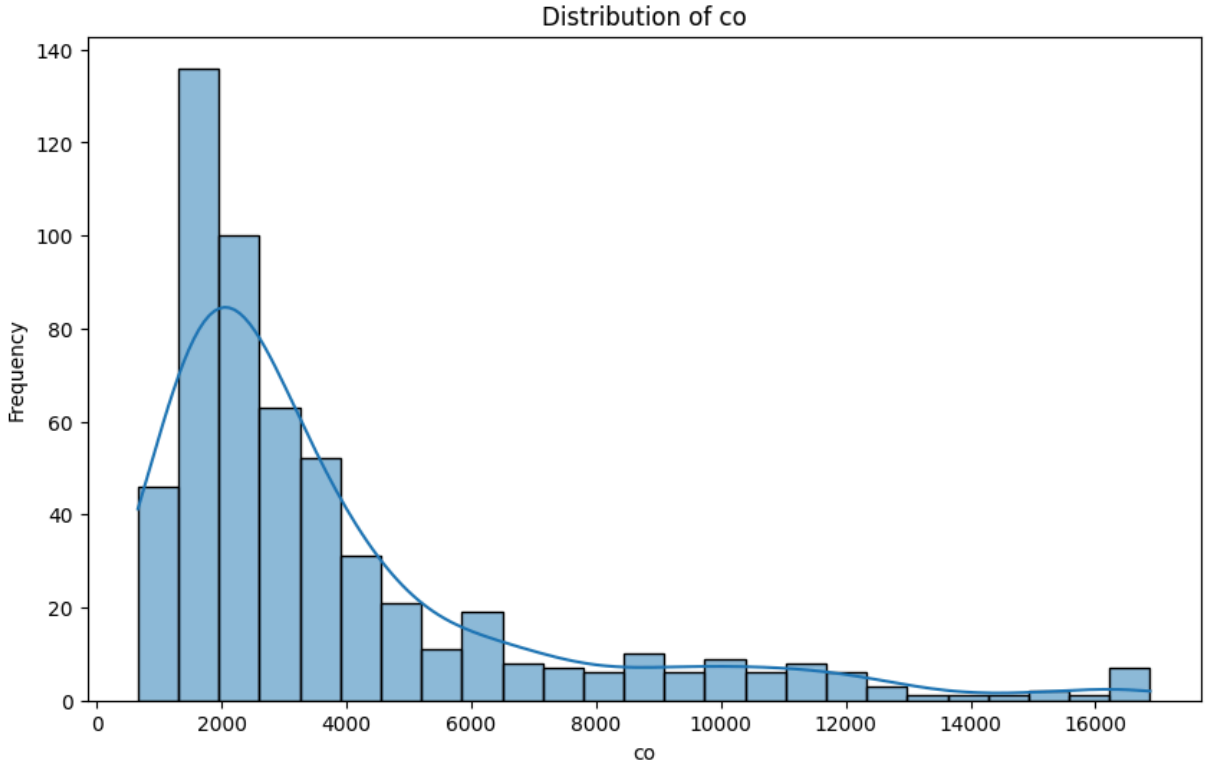
```
    plt.xlabel(pollutant)
    plt.ylabel('Frequency')
    plt.show()
```

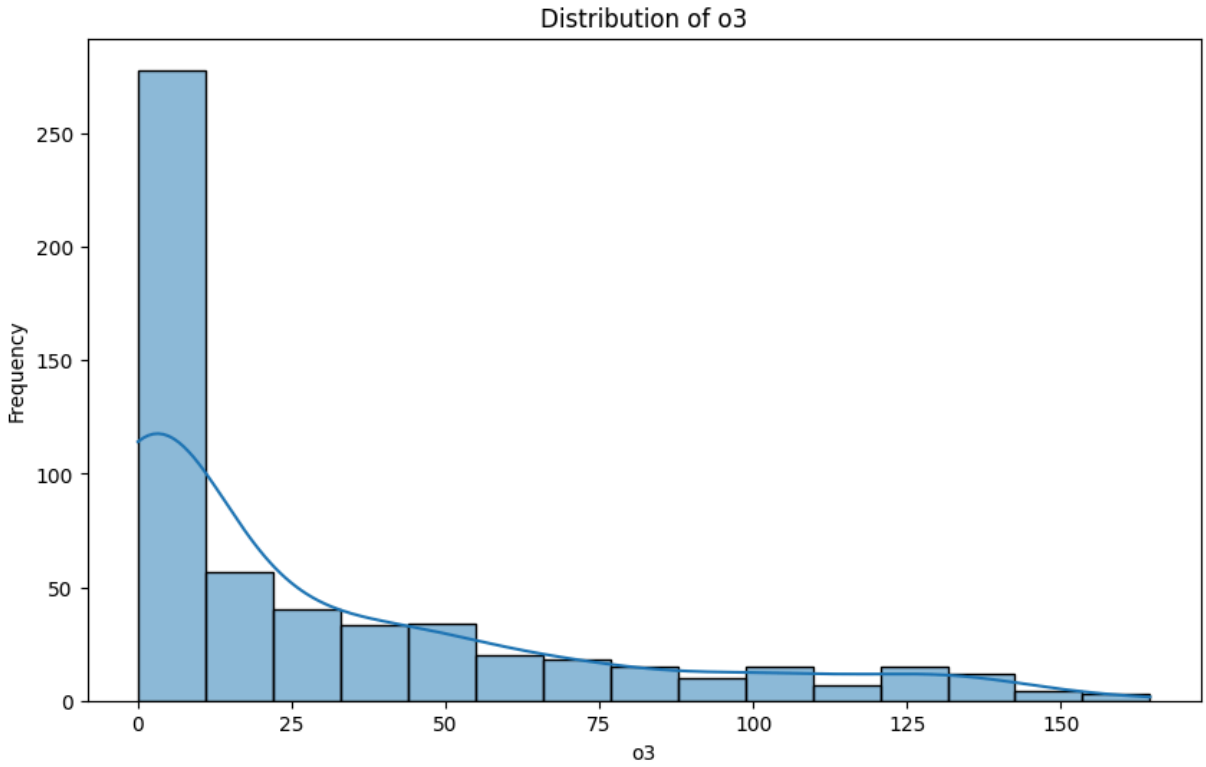


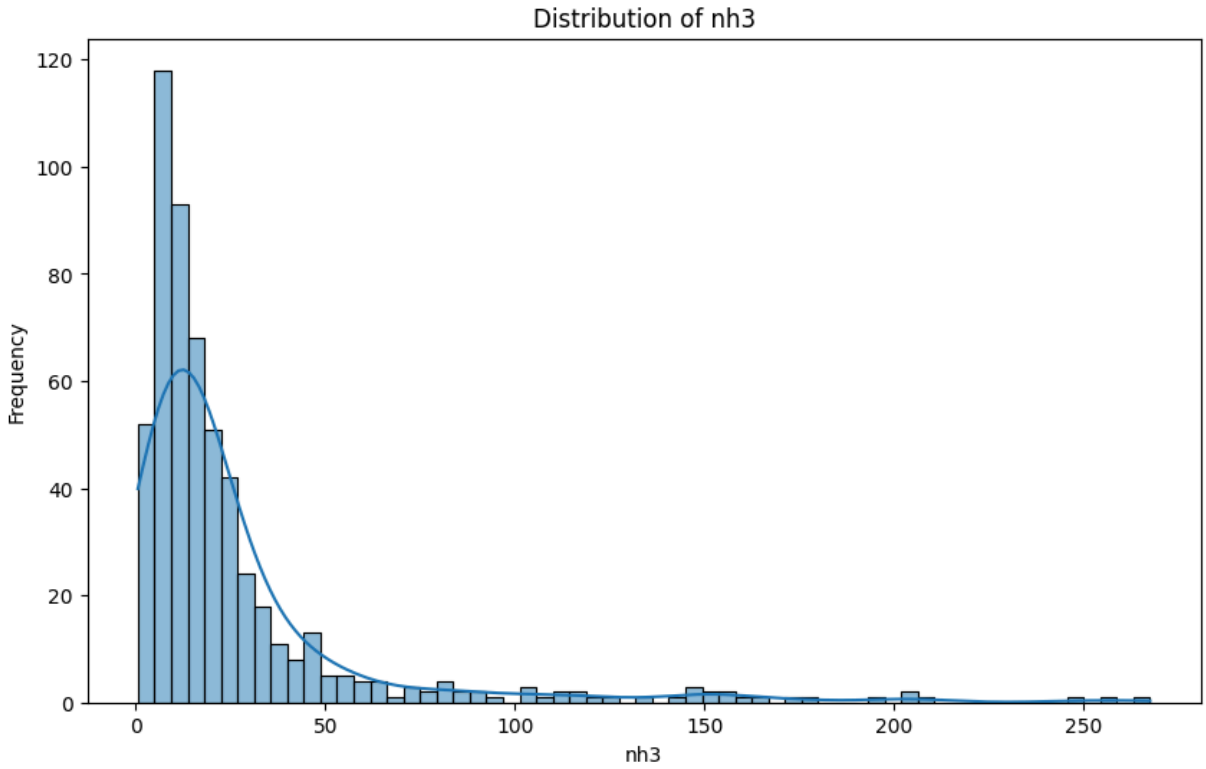


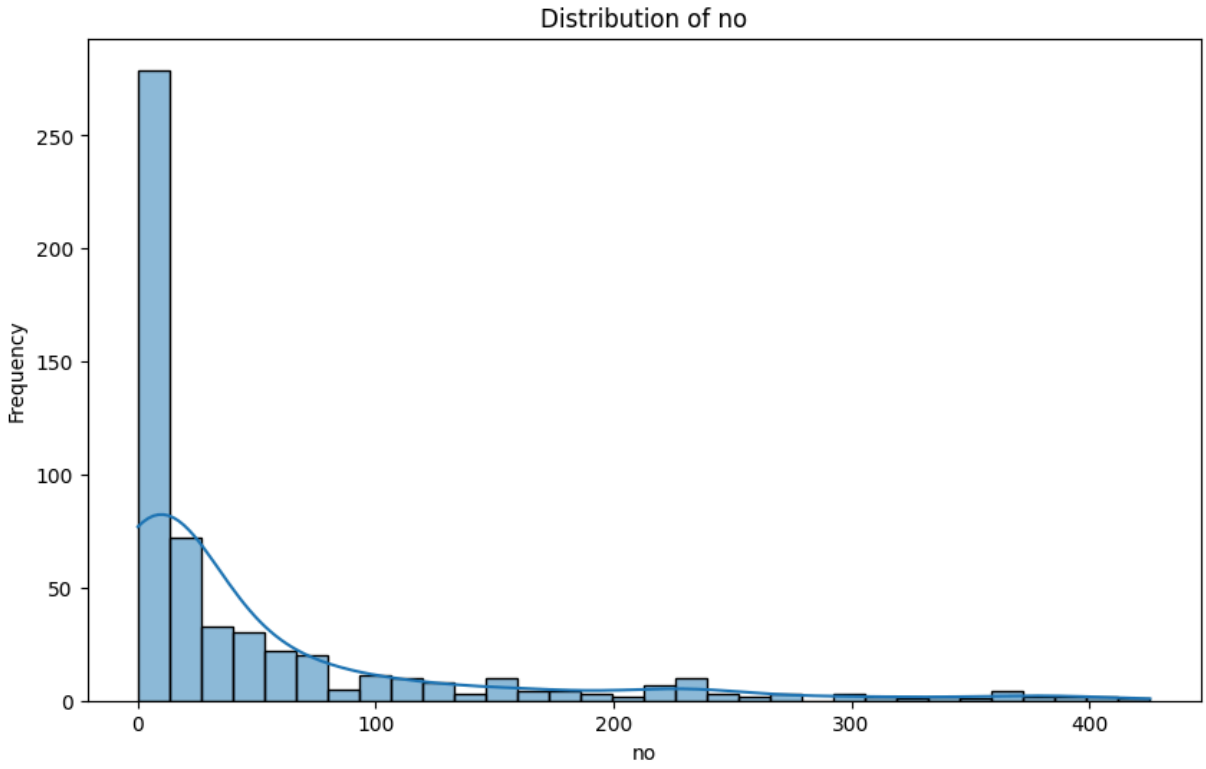

Distribution of no2
Frequency
no2
0
20
40
60
80
50
100
150
200
250

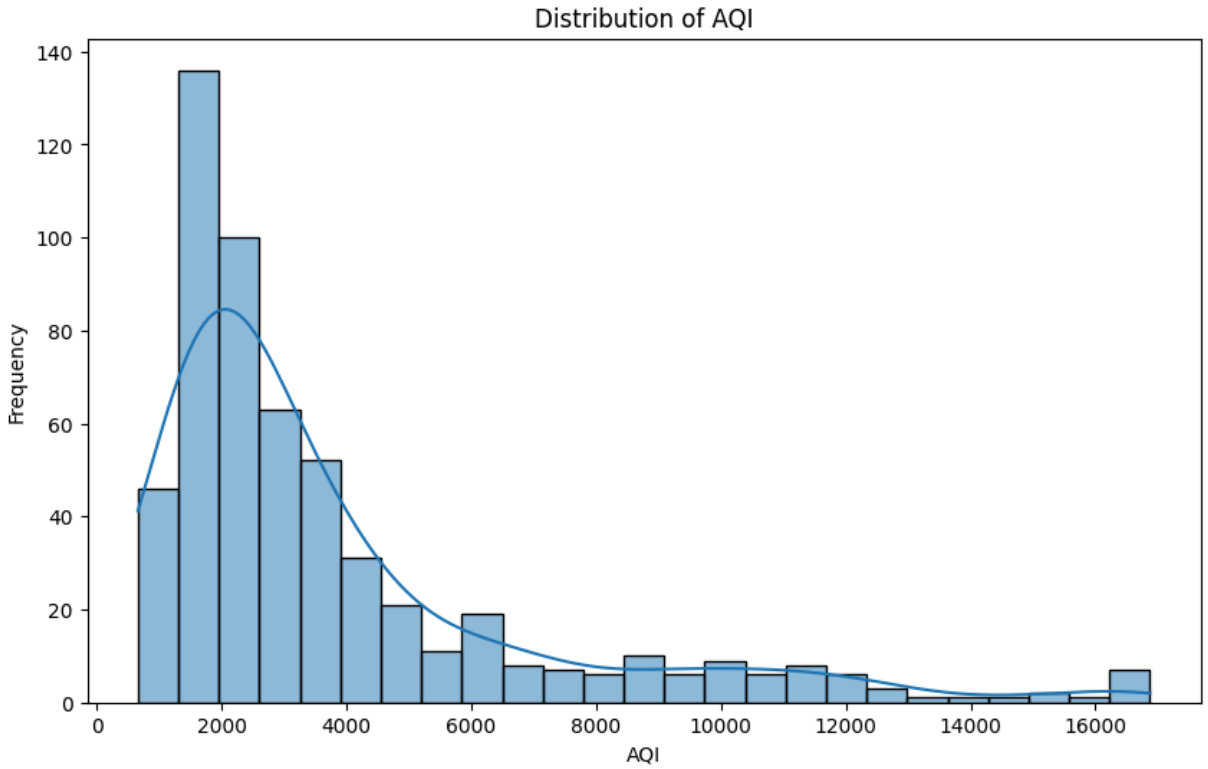

Distribution of so2
Frequency
so2
0
20
40
60
80
100
120
0
100
200
300
400
500


Distribution of co
Frequency
co
0
20
40
60
80
100
120
140
0
2000
4000
6000
8000
10000
12000
14000
16000


Distribution of o3
Frequency
o3
0
50
100
150
200
250
0
25
50
75
100
125
150


Distribution of nh3
Frequency
nh3
0
20
40
60
80
100
120
0
50
100
150
200
250


Distribution of no
Frequency
no
0
50
100
150
200
250
0
100
200
300
400



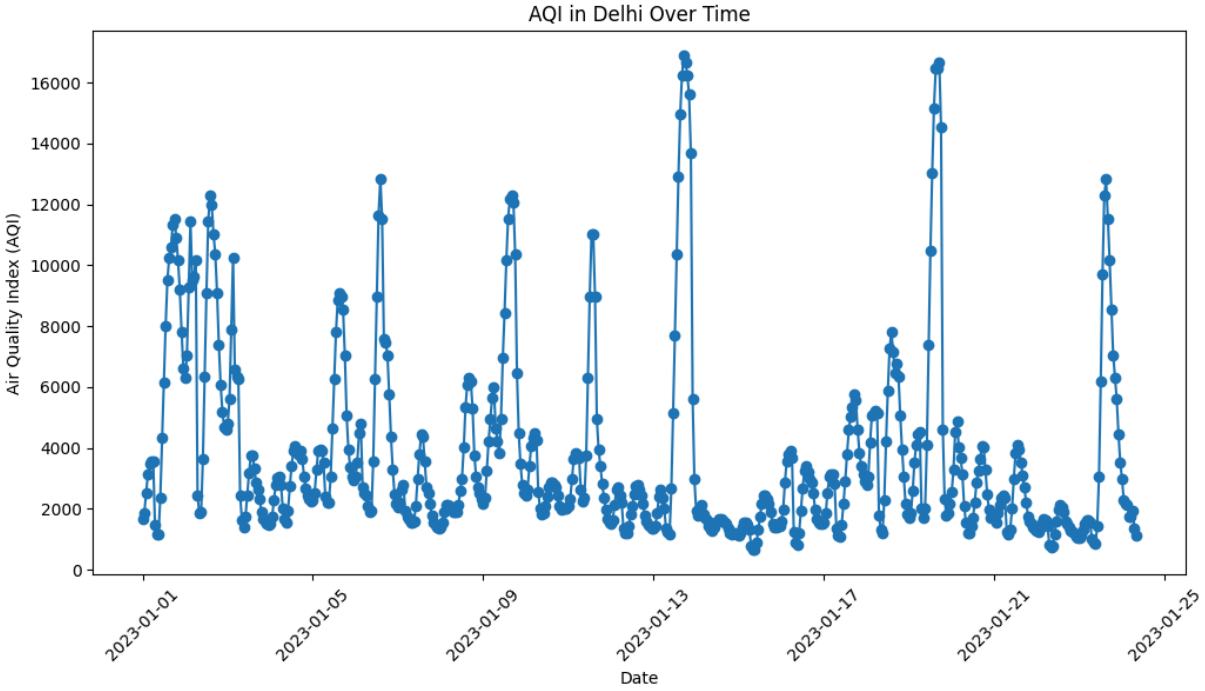
In [17]:
```
corr = df[['pm2_5', 'pm10','no2', 'o3', 'co', 'so2', 'no','nh3','co','AQI']].corr()
print(corr)
```

```
          pm2_5      pm10       no2        o3        co       so2        no  \
pm2_5  1.000000  0.994088  0.698696 -0.450458  0.953083  0.648996  0.888810
pm10   0.994088  1.000000  0.720050 -0.468477  0.966801  0.658325  0.903339
no2    0.698696  0.720050  1.000000 -0.407177  0.776402  0.734961  0.702201
o3    -0.450458 -0.468477 -0.407177  1.000000 -0.463082 -0.049158 -0.377813
co     0.953083  0.966801  0.776402 -0.463082  1.000000  0.716831  0.969740
so2    0.648996  0.658325  0.734961 -0.049158  0.716831  1.000000  0.734503
no     0.888810  0.903339  0.702201 -0.377813  0.969740  0.734503  1.000000
nh3    0.720303  0.754468  0.700254 -0.299663  0.826299  0.843635  0.823638
co     0.953083  0.966801  0.776402 -0.463082  1.000000  0.716831  0.969740
AQI    0.953083  0.966801  0.776402 -0.463082  1.000000  0.716831  0.969740

            nh3        co       AQI
pm2_5  0.720303  0.953083  0.953083
pm10   0.754468  0.966801  0.966801
no2    0.700254  0.776402  0.776402
o3    -0.299663 -0.463082 -0.463082
co     0.826299  1.000000  1.000000
so2    0.843635  0.716831  0.716831
no     0.823638  0.969740  0.969740
nh3    1.000000  0.826299  0.826299
co     0.826299  1.000000  1.000000
AQI    0.826299  1.000000  1.000000
```

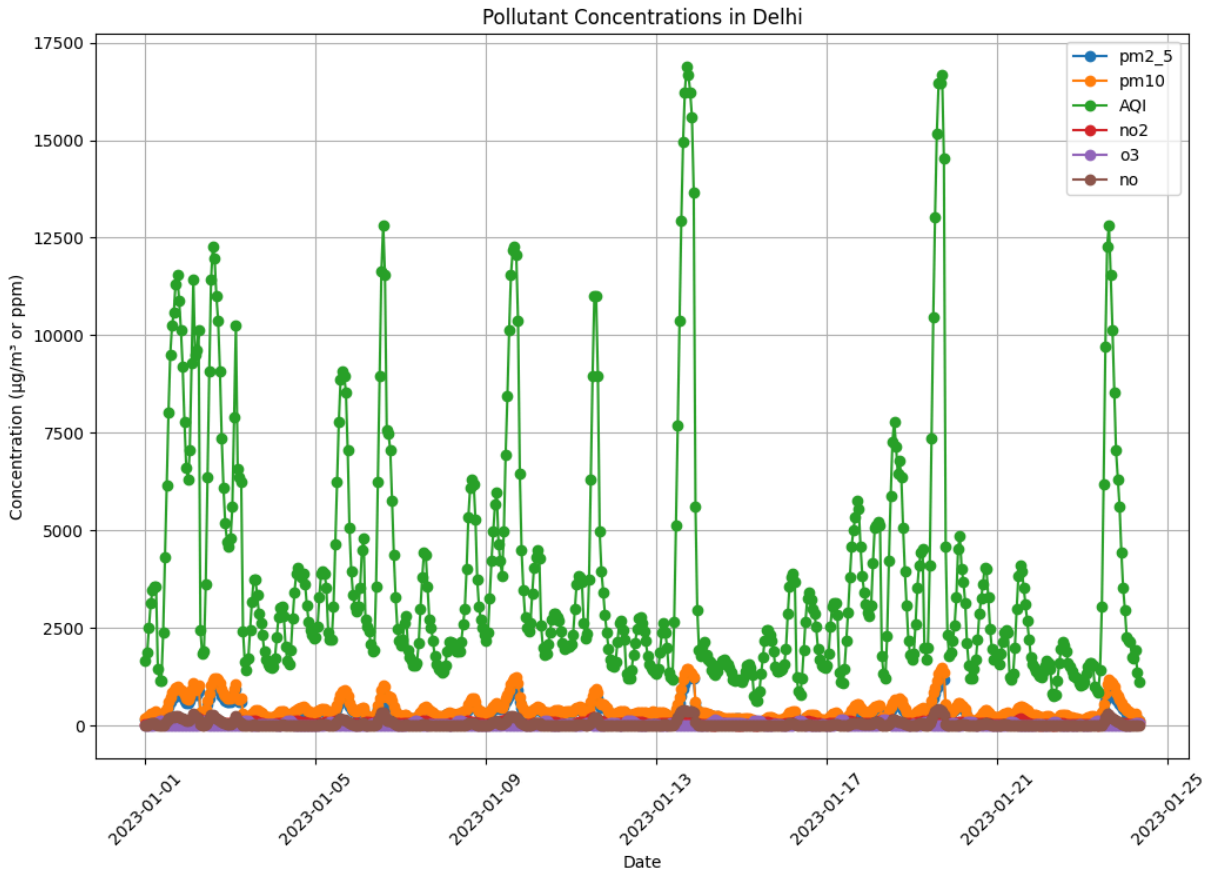
In [18]:
```
seasonal_data = df.groupby('date').agg({'AQI': 'mean', 'pm2_5': 'mean', 'pm10': 'me
print("Seasonal AQI and Pollutant Averages:")
print(seasonal_data.head())
```

```
Seasonal AQI and Pollutant Averages:
                            AQI   pm2_5    pm10
date
2023-01-01 00:00:00  1655.58  169.29  194.64
2023-01-01 01:00:00  1869.20  182.84  211.08
2023-01-01 02:00:00  2510.07  220.25  260.68
2023-01-01 03:00:00  3150.94  252.90  304.12
2023-01-01 04:00:00  3471.37  266.36  322.80
```

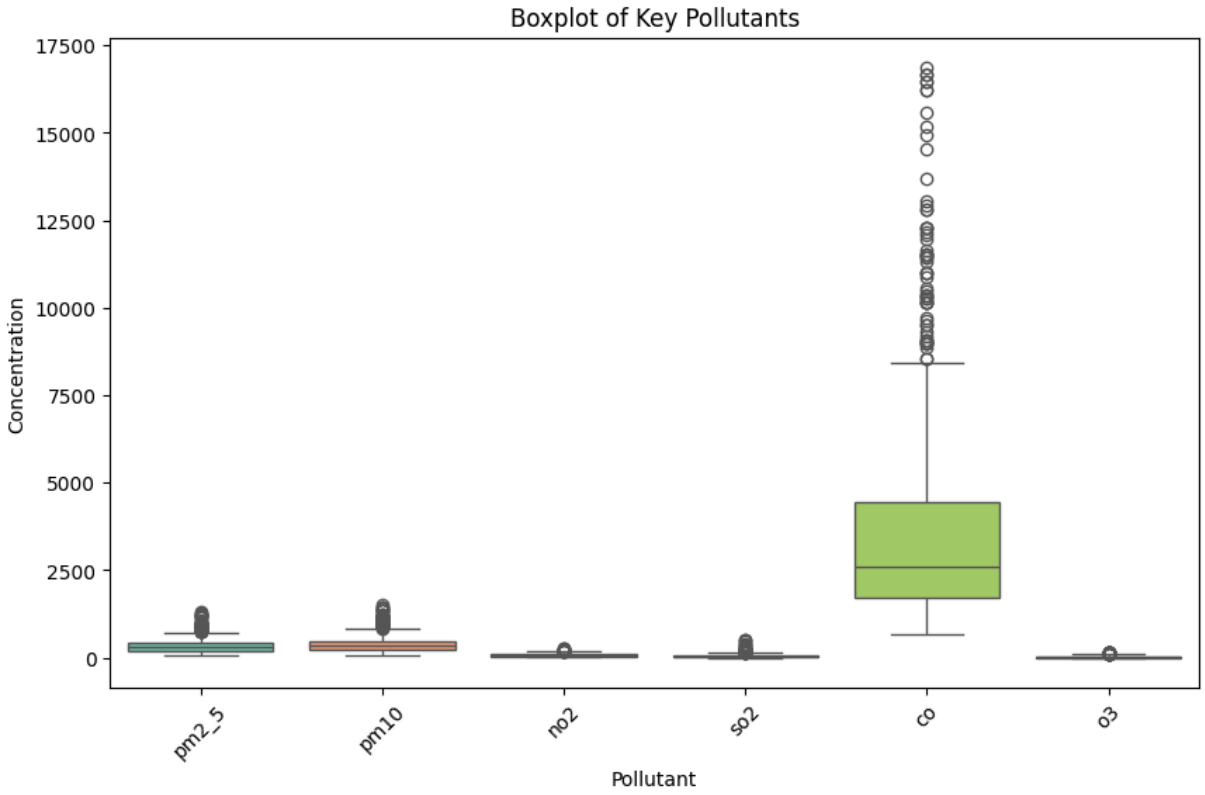
In [19]:
```
plt.figure(figsize=(12, 6))
plt.plot(df['date'], df['AQI'], marker='o', linestyle='-')
plt.xticks(rotation=45)
plt.xlabel('Date')
plt.ylabel('Air Quality Index (AQI)')
plt.title('AQI in Delhi Over Time')
plt.show()
```



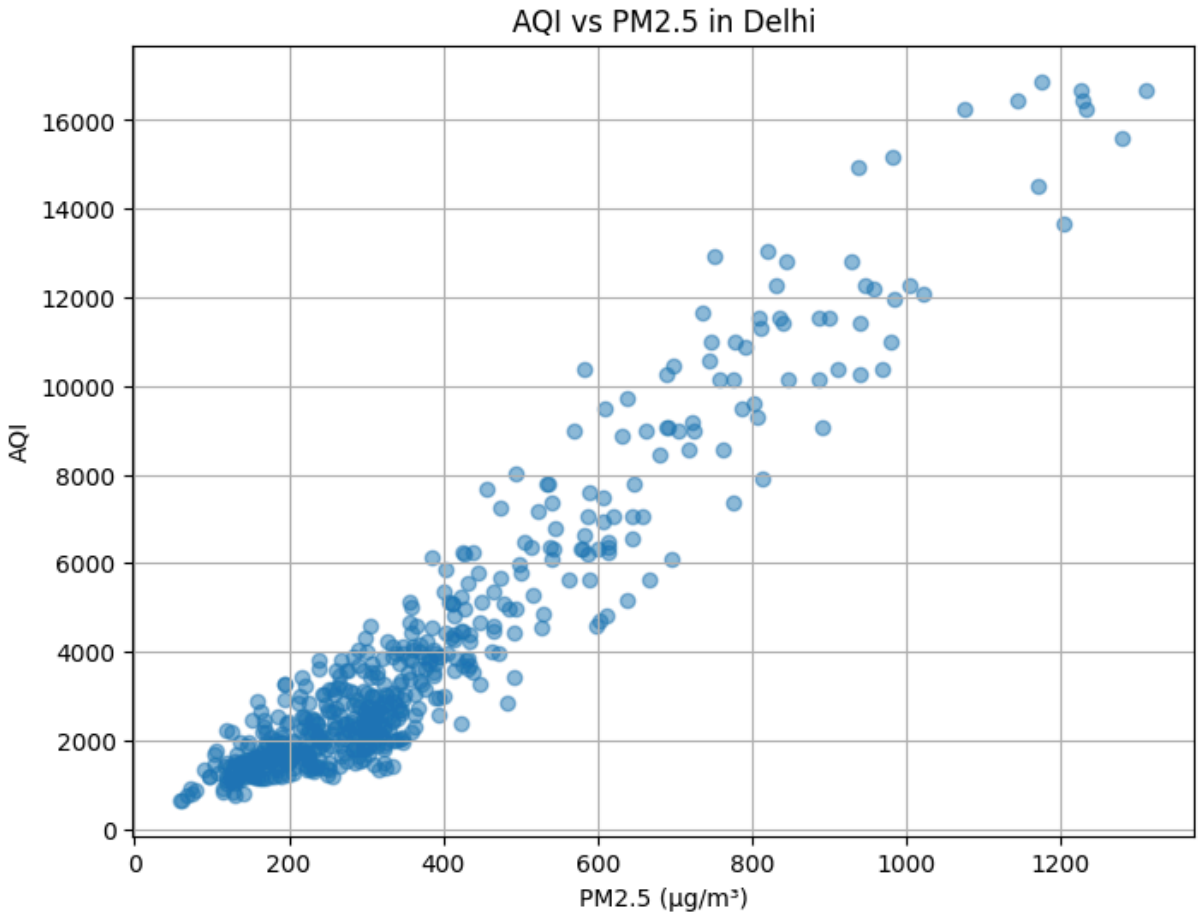
In [20]:
```
plt.figure(figsize=(12, 8))
pollutants = ['pm2_5', 'pm10', 'AQI', 'no2', 'o3', 'no']  # List of pollutants to p
for pollutant in pollutants:
    plt.plot(df['date'], df[pollutant], label=pollutant, marker='o', linestyle='-')
plt.title('Pollutant Concentrations in Delhi')
plt.xlabel('Date')
plt.ylabel('Concentration (µg/m³ or ppm)')
plt.xticks(rotation=45)
plt.legend()
plt.grid(True)
plt.show()
```



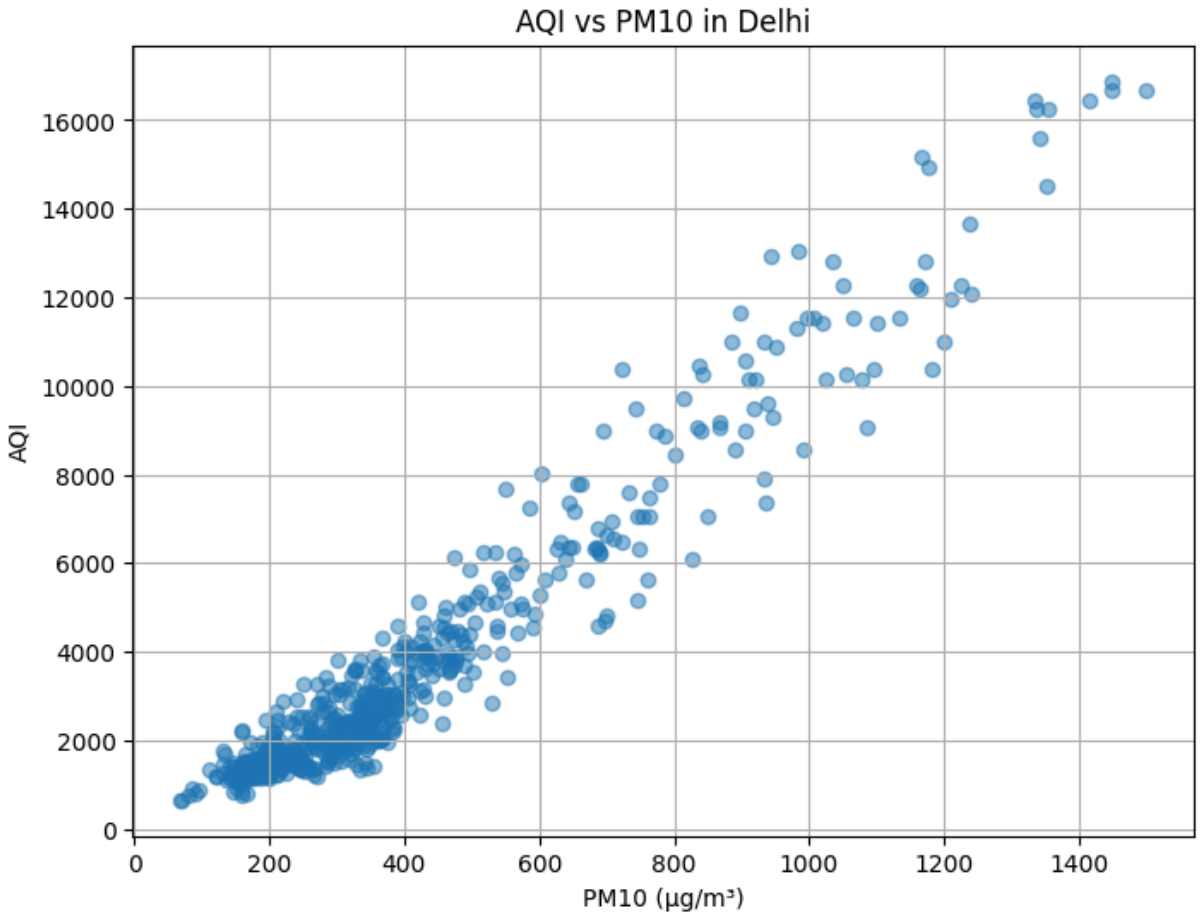
In [21]:
```
plt.figure(figsize=(10, 6))
sns.boxplot(data=df[['pm2_5', 'pm10', 'no2', 'so2', 'co', 'o3']], palette='Set2')
plt.title('Boxplot of Key Pollutants')
plt.xlabel('Pollutant')
plt.ylabel('Concentration')
plt.xticks(rotation=45)
plt.show()
```



In [22]:
```
plt.figure(figsize=(8, 6))
plt.scatter(df['pm2_5'], df['AQI'], alpha=0.5)
plt.title('AQI vs PM2.5 in Delhi')
plt.xlabel('PM2.5 (µg/m³)')
plt.ylabel('AQI')
plt.grid(True)
plt.show()
```

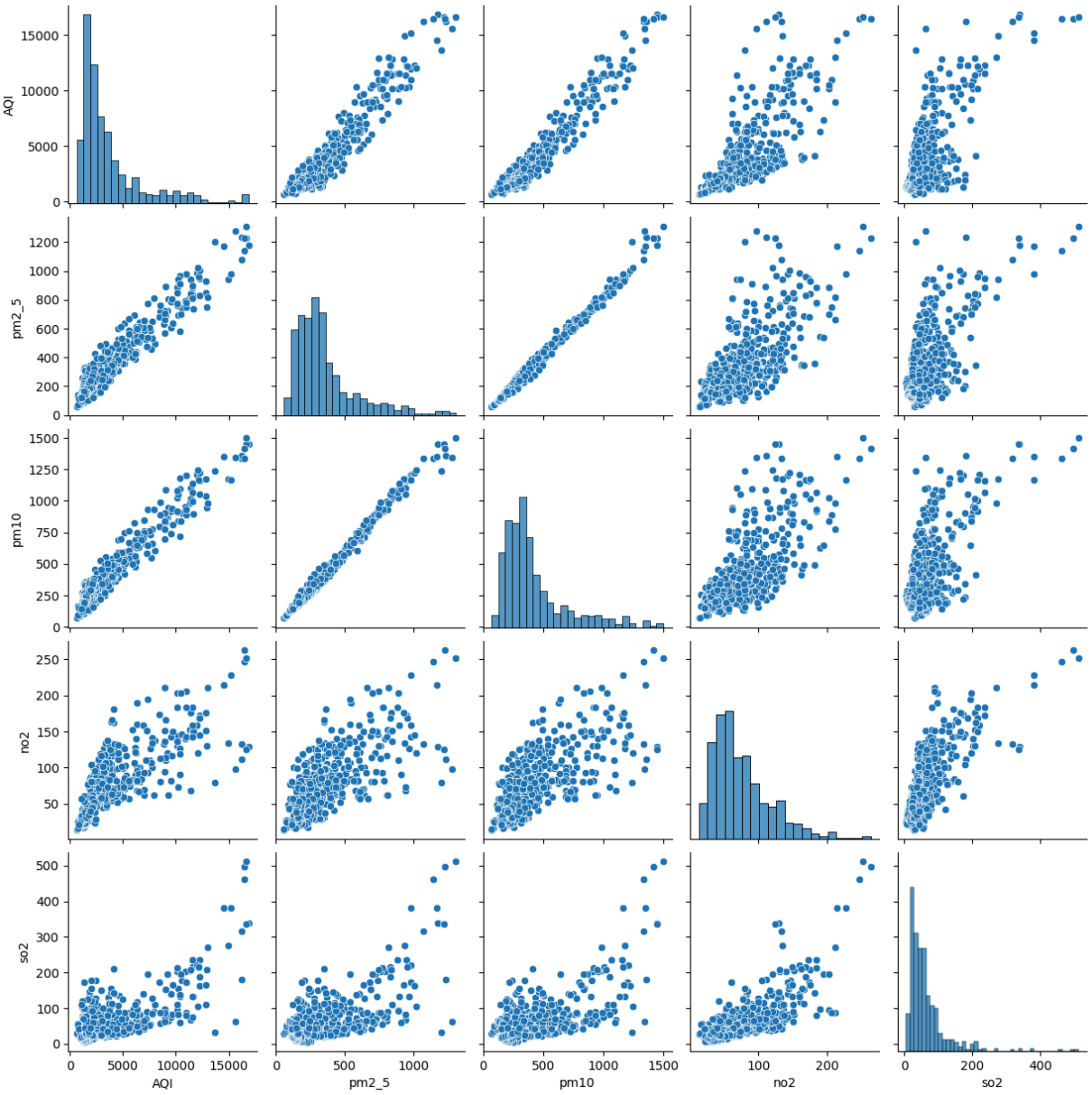


In [23]:
```
plt.figure(figsize=(8, 6))
plt.scatter(df['pm10'], df['AQI'], alpha=0.5)
plt.title('AQI vs PM10 in Delhi')
plt.xlabel('PM10 (µg/m³)')
plt.ylabel('AQI')
plt.grid(True)
plt.show()
```

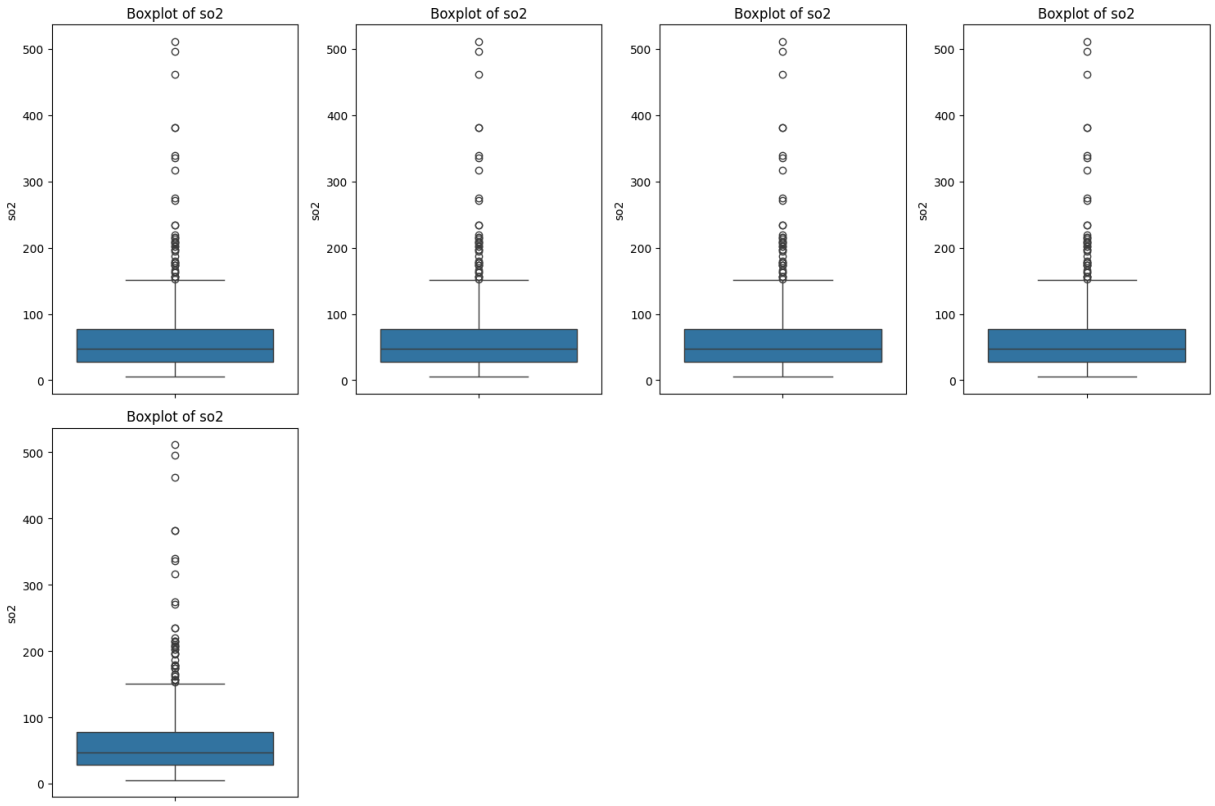


In [ ]:
```
df['AQI'] = pd.to_numeric(df['AQI'], errors='coerce')
pollutants = ['pm2_5', 'pm10', 'no2', 'so2']
for pollutant in pollutants:
    df[pollutant] = pd.to_numeric(df[pollutant], errors='coerce')
```

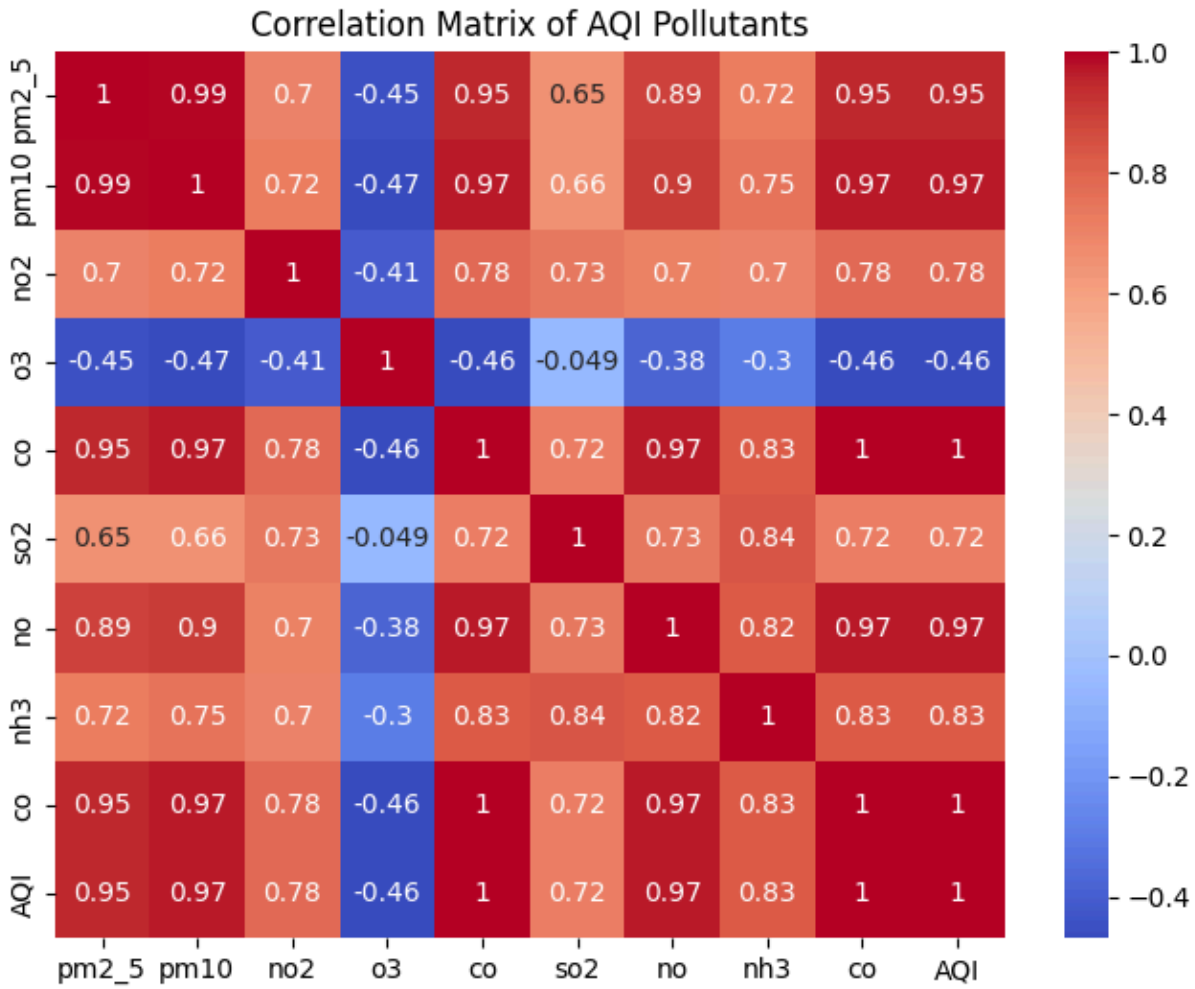
In [25]:
```
sns.pairplot(df[['AQI'] + pollutants])
plt.show()
```



In [26]:
```
plt.figure(figsize=(15, 10))
for i, pollutants in enumerate(['AQI'] + pollutants, 1):
    plt.subplot(2, 4, i)
    sns.boxplot(y=df[pollutant])
    plt.title(f'Boxplot of {pollutant}')
plt.tight_layout()
plt.show()
```



In [27]:
```
plt.figure(figsize=(8, 6))
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix of AQI Pollutants')
plt.show()
```

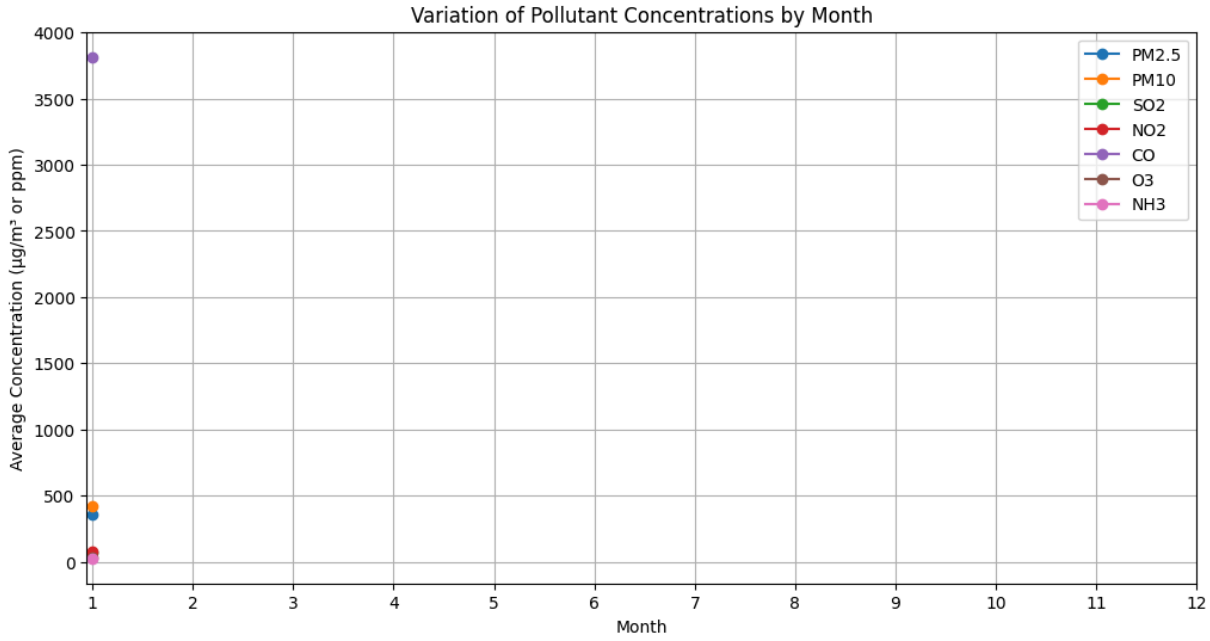


In [28]:
```
avg_concentrations = df[['pm2_5', 'pm10', 'so2', 'no2', 'co', 'o3']].mean()

highest_pollutant = avg_concentrations.idxmax()
highest_concentration = avg_concentrations.max()

print(f"The pollutant with the highest average concentration in Delhi is {highest_p
```

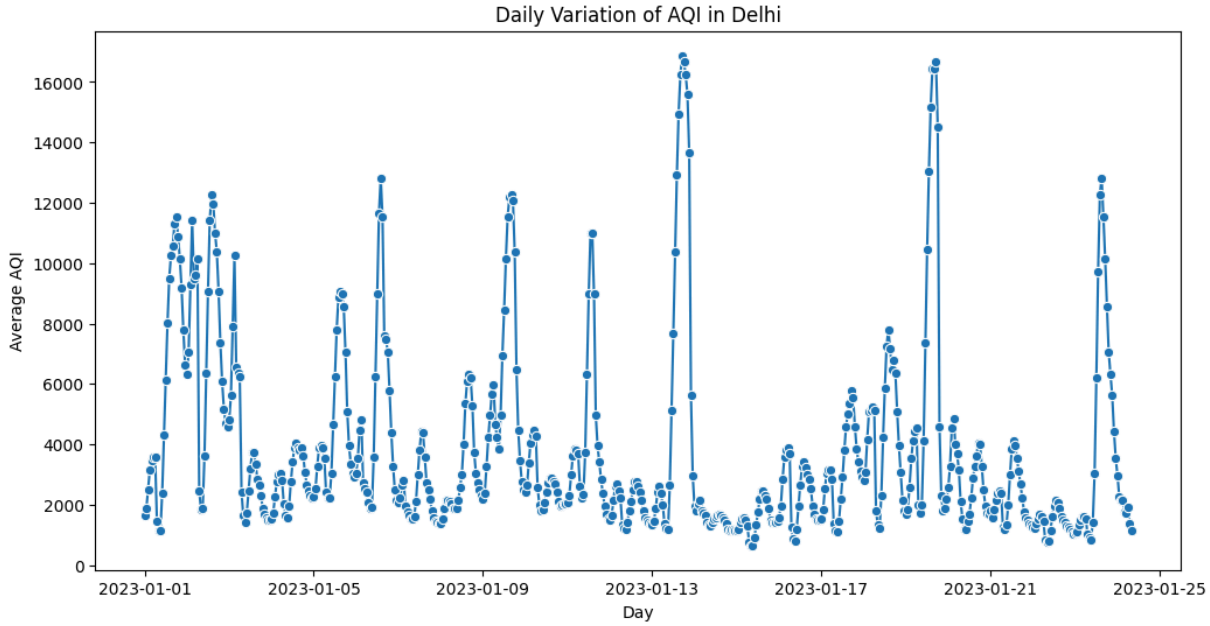
The pollutant with the highest average concentration in Delhi is co with an average concentration of 3814.94 µg/m³ or ppm.

In [29]:
```
df['Month'] = pd.to_datetime(df['date']).dt.month

monthly_avg_concentrations = df.groupby('Month').mean()

plt.figure(figsize=(12, 6))
plt.plot(monthly_avg_concentrations.index, monthly_avg_concentrations['pm2_5'], lab
plt.plot(monthly_avg_concentrations.index, monthly_avg_concentrations['pm10'], labe
plt.plot(monthly_avg_concentrations.index, monthly_avg_concentrations['so2'], label
plt.plot(monthly_avg_concentrations.index, monthly_avg_concentrations['no2'], label
plt.plot(monthly_avg_concentrations.index, monthly_avg_concentrations['co'], label=
plt.plot(monthly_avg_concentrations.index, monthly_avg_concentrations['o3'], label=
plt.plot(monthly_avg_concentrations.index, monthly_avg_concentrations['nh3'], label
plt.title('Variation of Pollutant Concentrations by Month')
plt.xlabel('Month')
plt.ylabel('Average Concentration (µg/m³ or ppm)')
```

```
plt.xticks(range(1, 13))
plt.grid(True)
plt.legend()
plt.show()
```

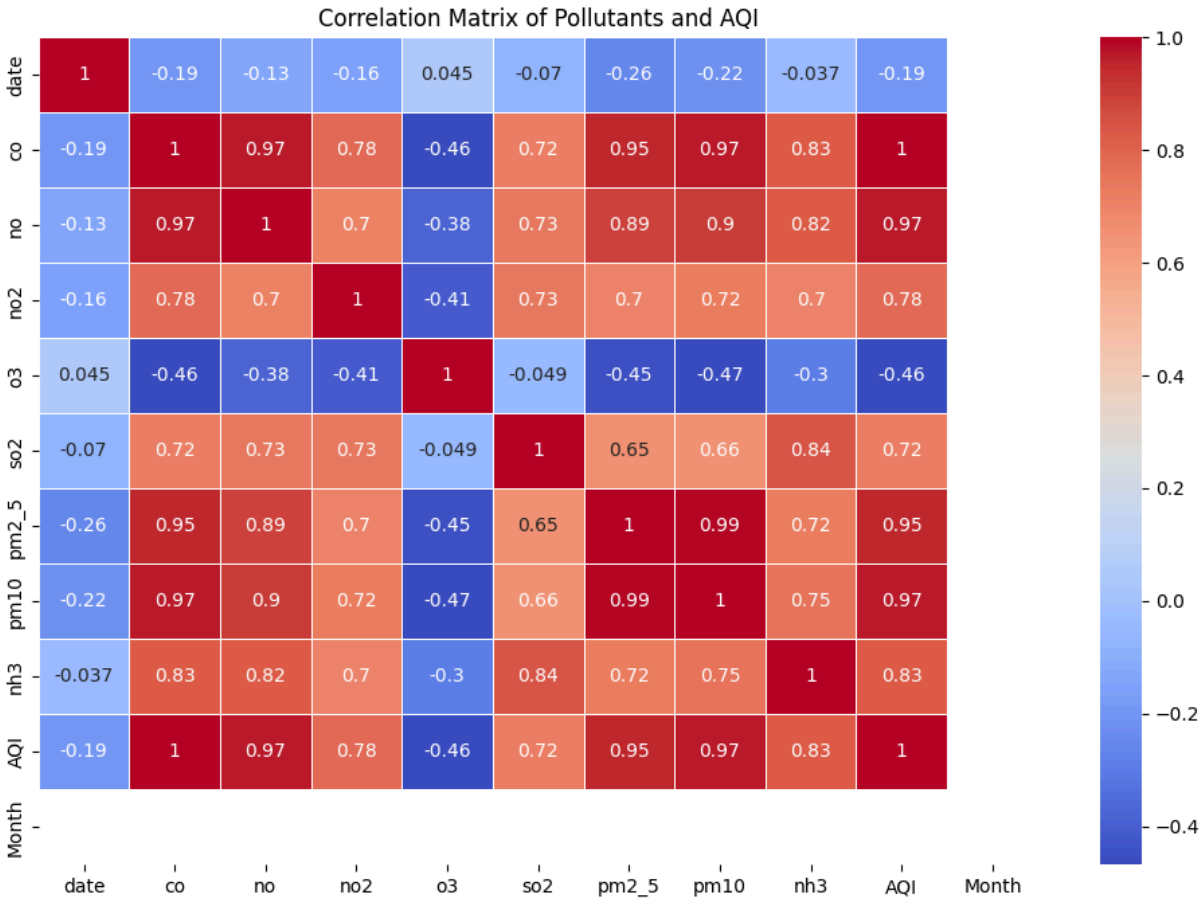


In [30]:
```
Day_aqi = df.groupby('date')['AQI'].mean().reset_index()
plt.figure(figsize=(12, 6))
sns.lineplot(x='date', y='AQI', data=Day_aqi, marker='o')
plt.title('Daily Variation of AQI in Delhi')
plt.xlabel('Day')
plt.ylabel('Average AQI')
plt.show()
```

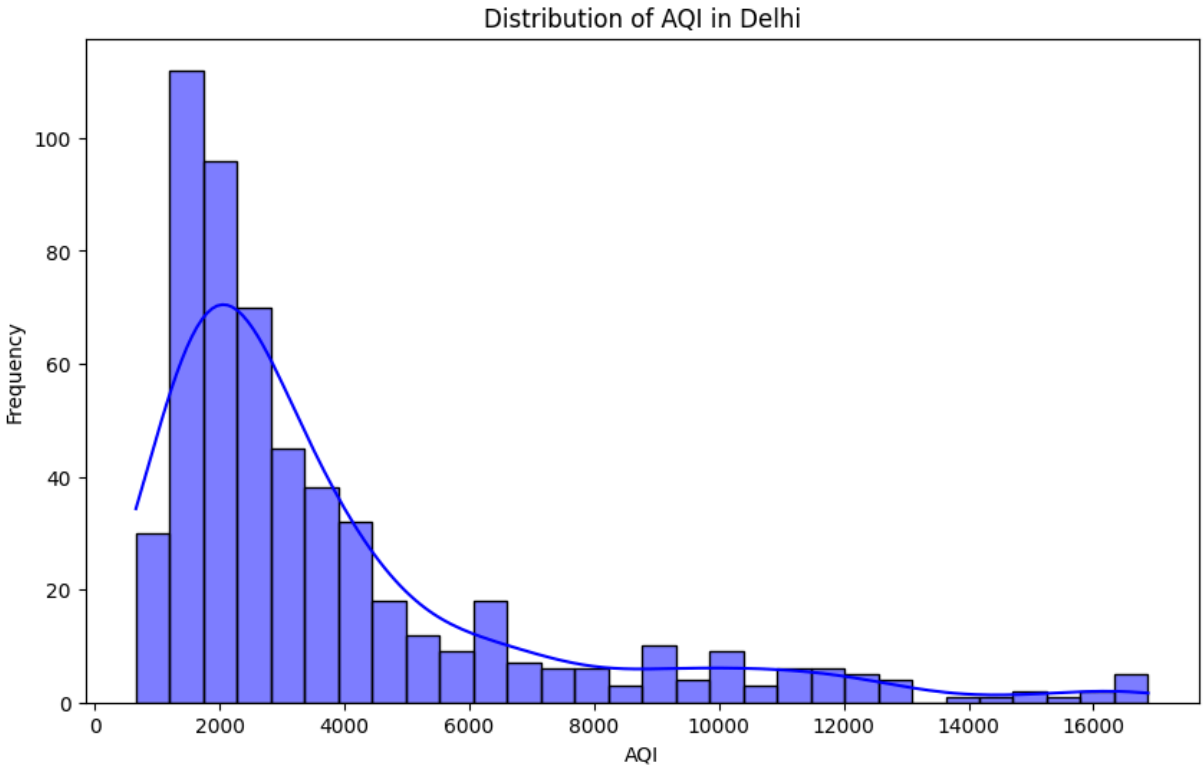


In [31]:
```
correlation_matrix = df.corr()
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', linewidths=0.5)
```

```
plt.title('Correlation Matrix of Pollutants and AQI')
plt.show()
```



In [32]:
```
plt.figure(figsize=(10, 6))
sns.histplot(df['AQI'], bins=30, kde=True, color='blue')
plt.title('Distribution of AQI in Delhi')
plt.xlabel('AQI')
plt.ylabel('Frequency')
plt.show()
```


Distribution of AQI in Delhi
Frequency
AQI
0
20
40
60
80
100
0
2000
4000
6000
8000
10000
12000
14000
16000