

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

```
file_path = "/content/delhiaqi.csv"
df = pd.read_csv(file_path)
print(df.head())
```

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

      nh3
0    5.83
1    7.66
2   11.40
3   13.55
4   14.19
```

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

```
df['Year'] = df['date'].dt.year
df['Month'] = df['date'].dt.month
df['Hour'] = df['date'].dt.hour
```

```
def assign_season(month):
    if month in [12, 1, 2]:
        return 'Winter'
    elif month in [3, 4, 5]:
        return 'Summer'
    elif month in [6, 7, 8]:
        return 'Monsoon'
    else:
        return 'Post-Monsoon'
```

```
df['Season'] = df['Month'].apply(assign_season)
```

```
def calculate_aqi(concentration, breakpoints):
    for (low, high, aqi_low, aqi_high) in breakpoints:
        if low <= concentration <= high:
            return ((aqi_high - aqi_low) / (high - low)) * (concentration - low) + aqi_low
    return None
```

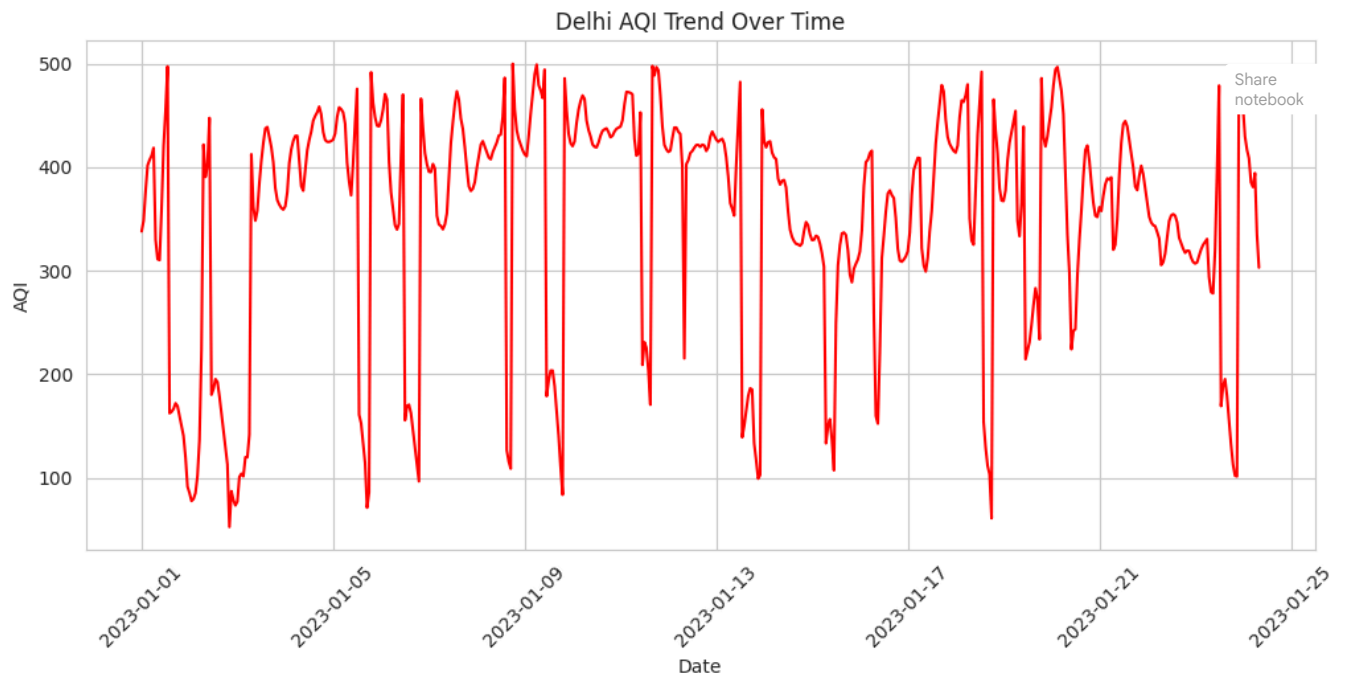
```
aqi_breakpoints = {
    'pm2_5': [(0, 30, 0, 50), (31, 60, 51, 100), (61, 90, 101, 200), (91, 120, 201, 300), (121, 250, 301, 400), (251, 500, 401, 500)],
    'pm10': [(0, 50, 0, 50), (51, 100, 51, 100), (101, 250, 101, 200), (251, 350, 201, 300), (351, 430, 301, 400), (431, 600, 401, 500)],
    '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)],
    'so2': [(0, 40, 0, 50), (41, 80, 51, 100), (81, 380, 101, 200), (381, 800, 201, 300), (801, 1600, 301, 400), (1601, 3000, 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)],
    'o3': [(0, 50, 0, 50), (51, 100, 51, 100), (101, 168, 101, 200), (169, 208, 201, 300), (209, 748, 301, 400), (749, 1000, 401, 500)]
}
```

```
for pollutant, breakpoints in aqi_breakpoints.items():
    df[f"AQI_{pollutant}"] = df[pollutant].apply(lambda x: calculate_aqi(x, breakpoints))
```

```
df["AQI"] = df[["AQI_{p}" for p in aqi_breakpoints.keys()]].max(axis=1)
```

```
sns.set_style("whitegrid")
```

```
plt.figure(figsize=(12, 5))
sns.lineplot(x='date', y='AQI', data=df, color='red')
plt.title('Delhi AQI Trend Over Time')
plt.xlabel('Date')
plt.ylabel('AQI')
plt.xticks(rotation=45)
plt.show()
```



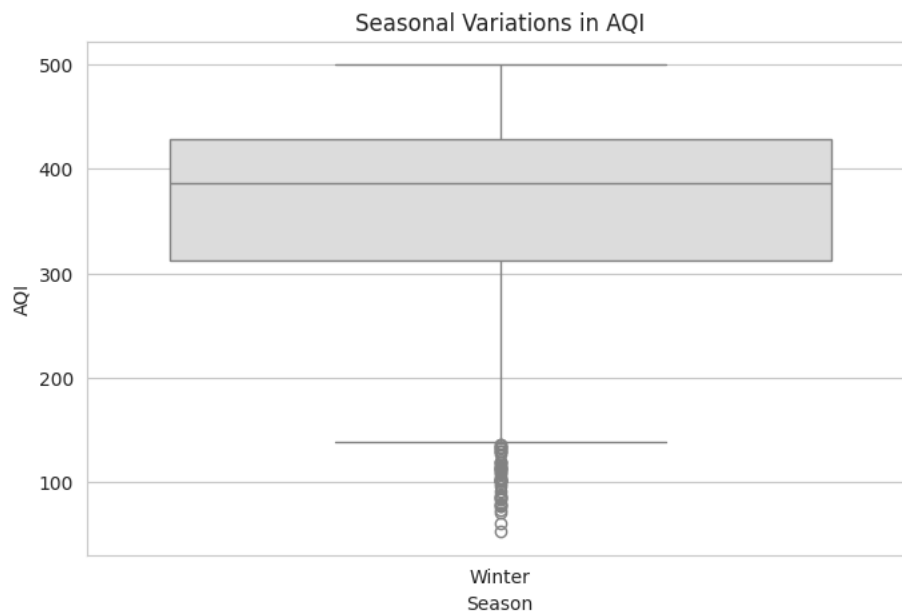
```
plt.figure(figsize=(8, 5))
sns.boxplot(x='Season', y='AQI', data=df, palette='coolwarm')
plt.title('Seasonal Variations in AQI')
plt.show()
```



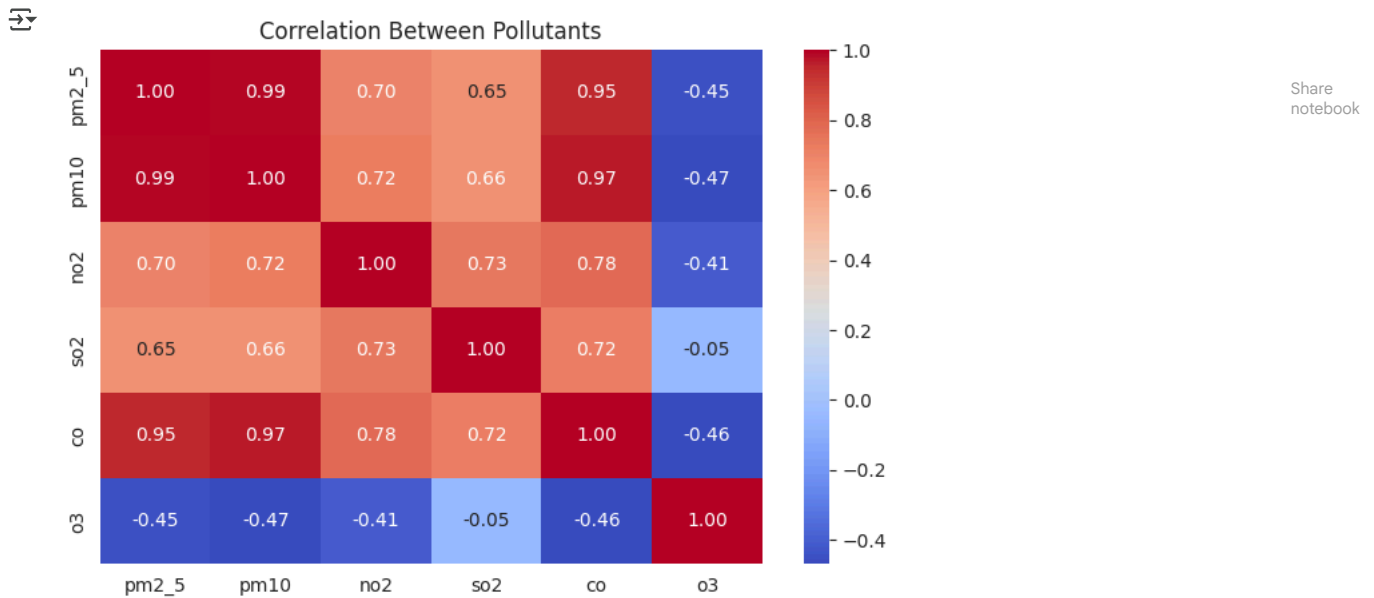
<ipython-input-16-afb8c7466a07>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `le`

```
sns.boxplot(x='Season', y='AQI', data=df, palette='coolwarm')
```

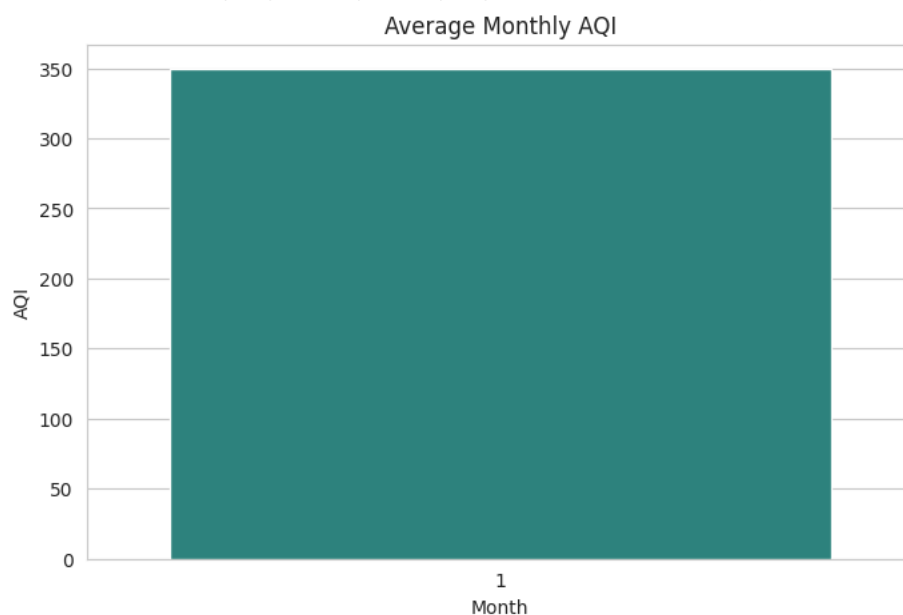


```
pollutants = ['pm2_5', 'pm10', 'no2', 'so2', 'co', 'o3']
plt.figure(figsize=(8, 5))
sns.heatmap(df[pollutants].corr(), annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Between Pollutants')
plt.show()
```



```
monthly_avg = df.groupby('Month')['AQI'].mean()
plt.figure(figsize=(8, 5))
sns.barplot(x=monthly_avg.index, y=monthly_avg.values, palette='viridis')
plt.title('Average Monthly AQI')
plt.xlabel('Month')
plt.ylabel('AQI')
plt.show()
```

`<ipython-input-18-4dfc4dd5b24e>:3: FutureWarning:`
 Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `le`
`sns.barplot(x=monthly_avg.index, y=monthly_avg.values, palette='viridis')`



```
plt.figure(figsize=(8, 5))
df[[f"AQI_{p}" for p in pollutants]].mean().plot(kind='bar', color=['red', 'orange', 'yellow', 'green', 'blue', 'purple'])
plt.title('Average AQI Contribution of Key Pollutants')
plt.xlabel('Pollutants')
plt.ylabel('Mean AQI Contribution')
plt.show()
```



Share
notebook

