Uploading dataset.

```
from google.colab import files
uploaded = files.upload()
```

Choose files delhiaqi.csv

delhiaqi.csv(text/csv) - 40158 bytes, last modified: 27/10/2025 - 100% done

```
Saving delhiaqi.csv to delhiaqi.csv
```

Load and view Dataset

```
import pandas as pd
df = pd.read_csv('delhiaqi.csv')
df.head()
```

	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

Next steps: Generate code with df | New interactive sheet

Clean and Prepare the Data.

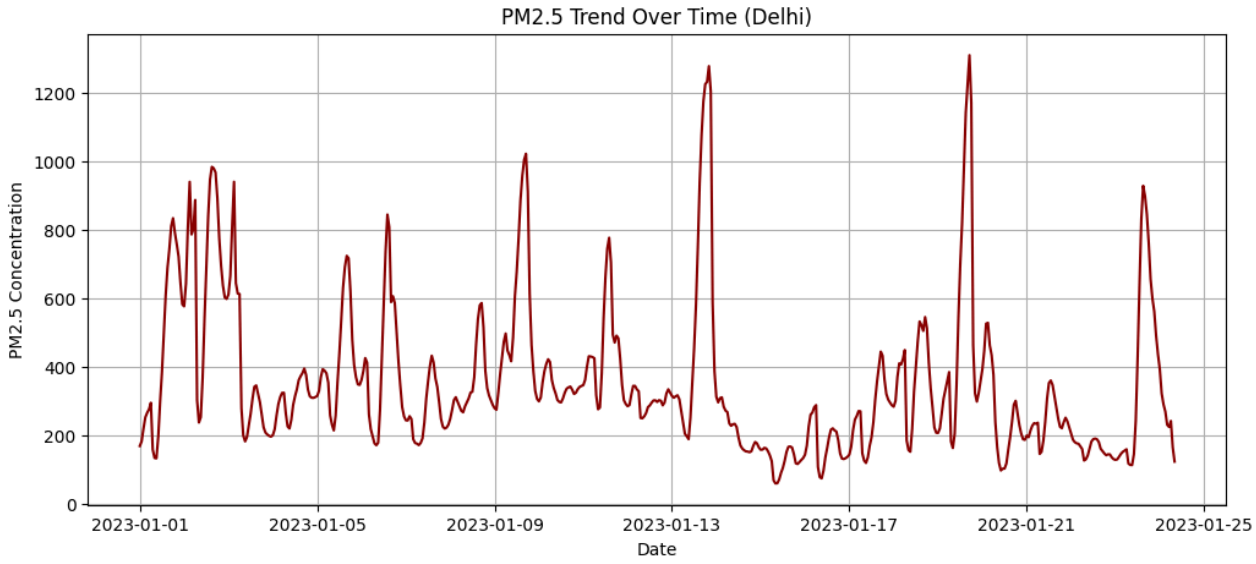
```
# Check data info
df.info()

# Check for missing values
df.isnull().sum()

# Convert date column to datetime type
df['date'] = pd.to_datetime(df['date'])

# Sort data by date
df = df.sort_values('date')
```

```
<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
```

Plot the AQI Trend Over Time

```
import matplotlib.pyplot as plt

plt.figure(figsize=(12,5))
plt.plot(df['date'], df['pm2_5'], color='darkred')
plt.title('PM2.5 Trend Over Time (Delhi)')
plt.xlabel('Date')
plt.ylabel('PM2.5 Concentration')
plt.grid(True)
plt.show()
```



Seasonal Variation

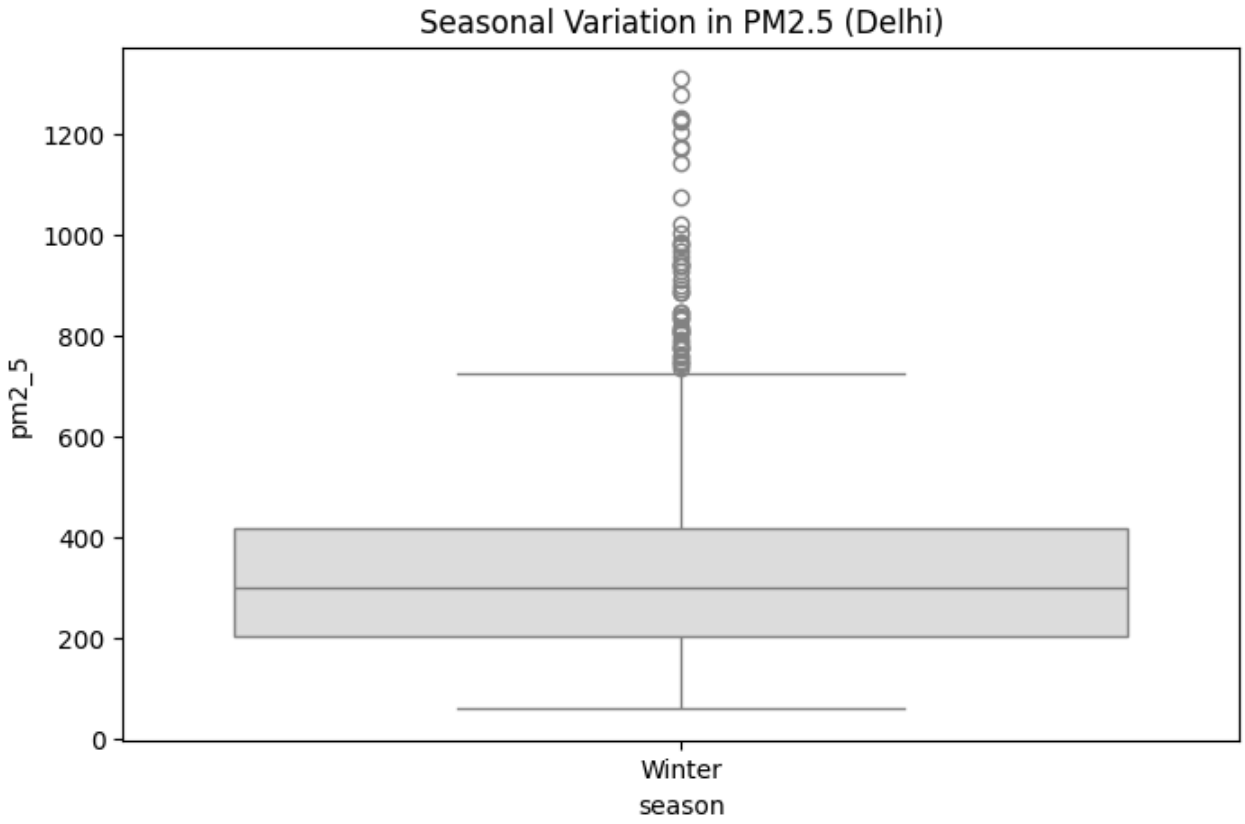
```
df['month'] = df['date'].dt.month

def season(m):
    if m in [12,1,2]:
        return 'Winter'
    elif m in [3,4,5]:
        return 'Summer'
    elif m in [6,7,8,9]:
        return 'Monsoon'
    else:
        return 'Post-Monsoon'
```

```
df['season'] = df['month'].apply(season)
```

For Visualize

```
import seaborn as sns

plt.figure(figsize=(8,5))
sns.boxplot(x='season', y='pm2_5', data=df, palette='coolwarm')
plt.title('Seasonal Variation in PM2.5 (Delhi)')
plt.show()
```

```
/tmp/ipython-input-3330277679.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Ass

  sns.boxplot(x='season', y='pm2_5', data=df, palette='coolwarm')
```

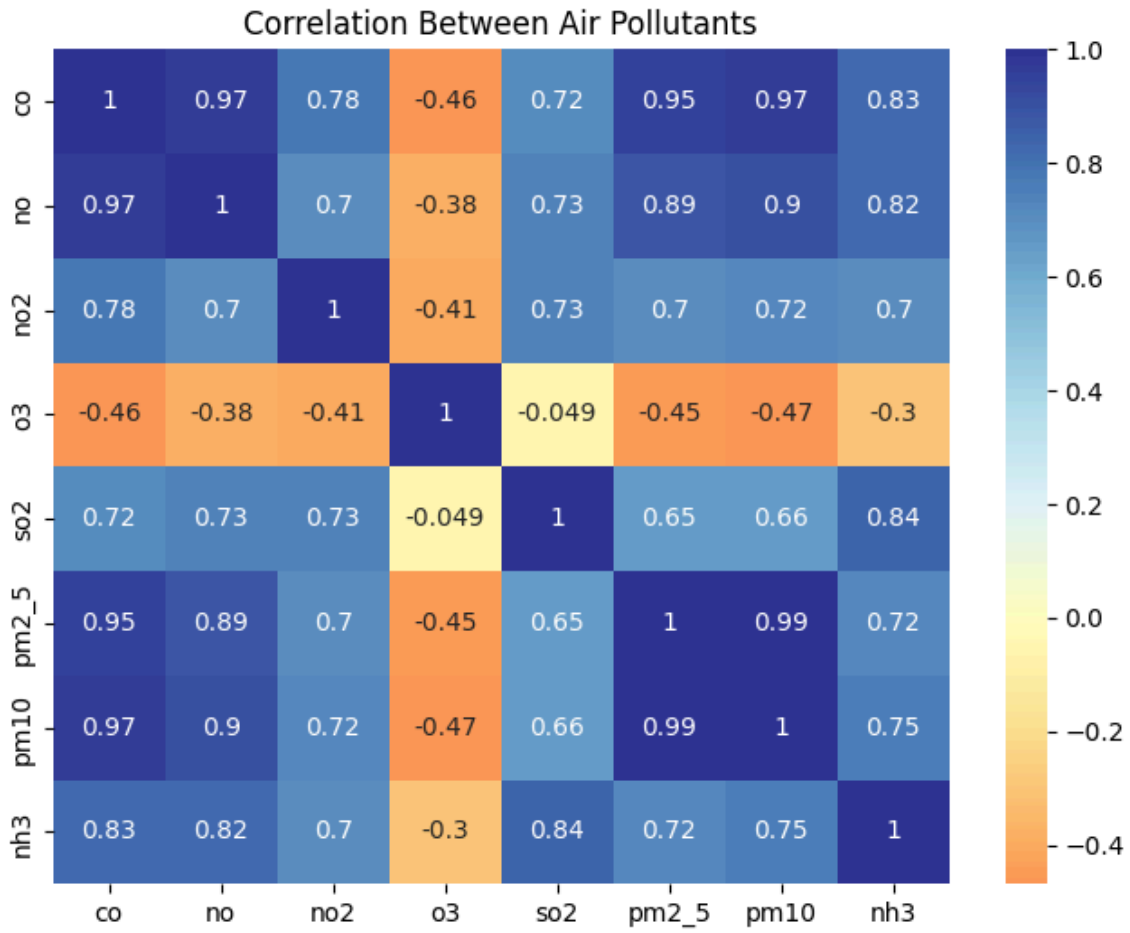


c

Correlation Between Pollutants

```
pollutants = ['co','no','no2','o3','so2','pm2_5','pm10','nh3']
corr = df[pollutants].corr()

plt.figure(figsize=(8,6))
sns.heatmap(corr, annot=True, cmap='RdYlBu', center=0)
plt.title('Correlation Between Air Pollutants')
```

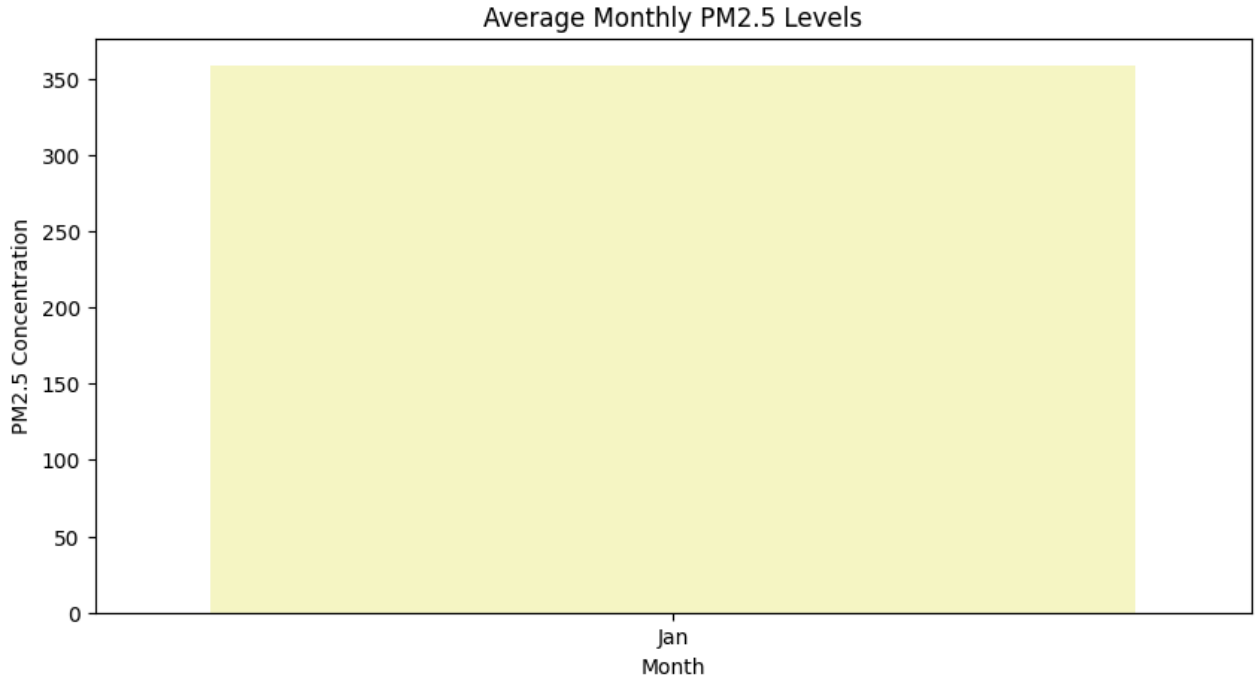
```
plt.show()
```



Monthly Average Air Quality.

```
df['year'] = df['date'].dt.year
df['month_name'] = df['date'].dt.strftime('%b')

monthly_avg = df.groupby('month_name')['pm2_5'].mean()

plt.figure(figsize=(10,5))
sns.barplot(x=monthly_avg.index, y=monthly_avg.values, palette='Spectral')
plt.title('Average Monthly PM2.5 Levels')
plt.xlabel('Month')
plt.ylabel('PM2.5 Concentration')
plt.show()
```

```
/tmp/ipython-input-158190683.py:7: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Ass

  sns.barplot(x=monthly_avg.index, y=monthly_avg.values, palette='Spectral')
```



Insights from AQI Analysis

- PM2.5 and PM10 levels are highest during winter months(Nov-Jan) -due to temperature inversion. lowest in Monsoon.
- CO and PM10 have strong correlation which is from vehicle + dust emission major sources.
- Seasonal spikes indicate stubble buring and temperature inversion effects.