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

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving delhiaqi.csv to delhiaqi (1).csv
```

```
import pandas as pd
df=pd.read_csv("delhiaqi.csv")
df['date'] = pd.to_datetime(df['date'], format='%d-%m-%Y %H:%M')         # Convert to datetime
df['hour'] = df['date'].dt.hour
df['month'] = df['date'].dt.month

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

df['season'] = df['month'].apply(assign_season)
df.to_csv('/content/delhiaqi_updated.csv', index=False)
```

```
import os

file_path = '/content/delhiaqi_updated.csv'

if os.path.exists(file_path):
    print(f"File '{file_path}' exists.")
    # Read the saved file to check its content (optional, for further debugging)
    try:
        saved_df = pd.read_csv(file_path)
        print("First 5 rows of the saved file:")
        display(saved_df)
    except Exception as e:
        print(f"Error reading the saved file: {e}")
else:
    print(f"File '{file_path}' does not exist.")
```

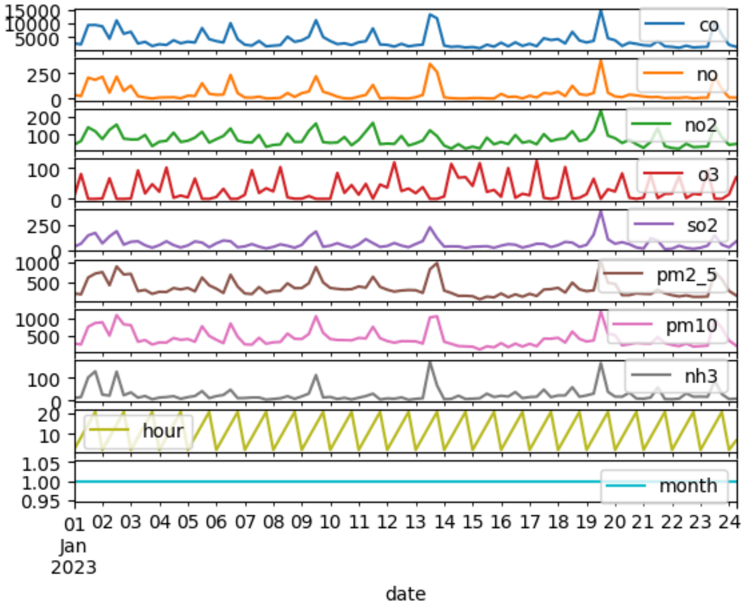
```
File '/content/delhiaqi_updated.csv' exists.
First 5 rows of the saved file:
```

	date	co	no	no2	o3	so2	pm2_5	pm10	nh3	hour	month	season
0	2023-01-01 00:00:00	1655.58	1.66	39.41	5.90	17.88	169.29	194.64	5.83	0	1	Winter
1	2023-01-01 01:00:00	1869.20	6.82	42.16	1.99	22.17	182.84	211.08	7.66	1	1	Winter
2	2023-01-01 02:00:00	2510.07	27.72	43.87	0.02	30.04	220.25	260.68	11.40	2	1	Winter
3	2023-01-01 03:00:00	3150.94	55.43	44.55	0.85	35.76	252.90	304.12	13.55	3	1	Winter
4	2023-01-01 04:00:00	3471.37	68.84	45.24	5.45	39.10	266.36	322.80	14.19	4	1	Winter
...	...	...	...	...	...	...	...	...	...	...	...	...
556	2023-01-24 04:00:00	1762.39	4.64	37.01	33.26	30.52	231.15	289.84	6.27	4	1	Winter
557	2023-01-24 05:00:00	1735.69	6.82	34.96	46.49	34.33	225.08	280.52	9.12	5	1	Winter
558	2023-01-24 06:00:00	1922.61	8.16	40.10	56.51	43.39	242.49	296.07	12.54	6	1	Winter
559	2023-01-24 07:00:00	1361.85	9.05	52.78	71.53	100.14	165.67	191.82	7.47	7	1	Winter
560	2023-01-24 08:00:00	1134.87	8.61	56.89	80.11	110.63	123.76	140.26	5.51	8	1	Winter

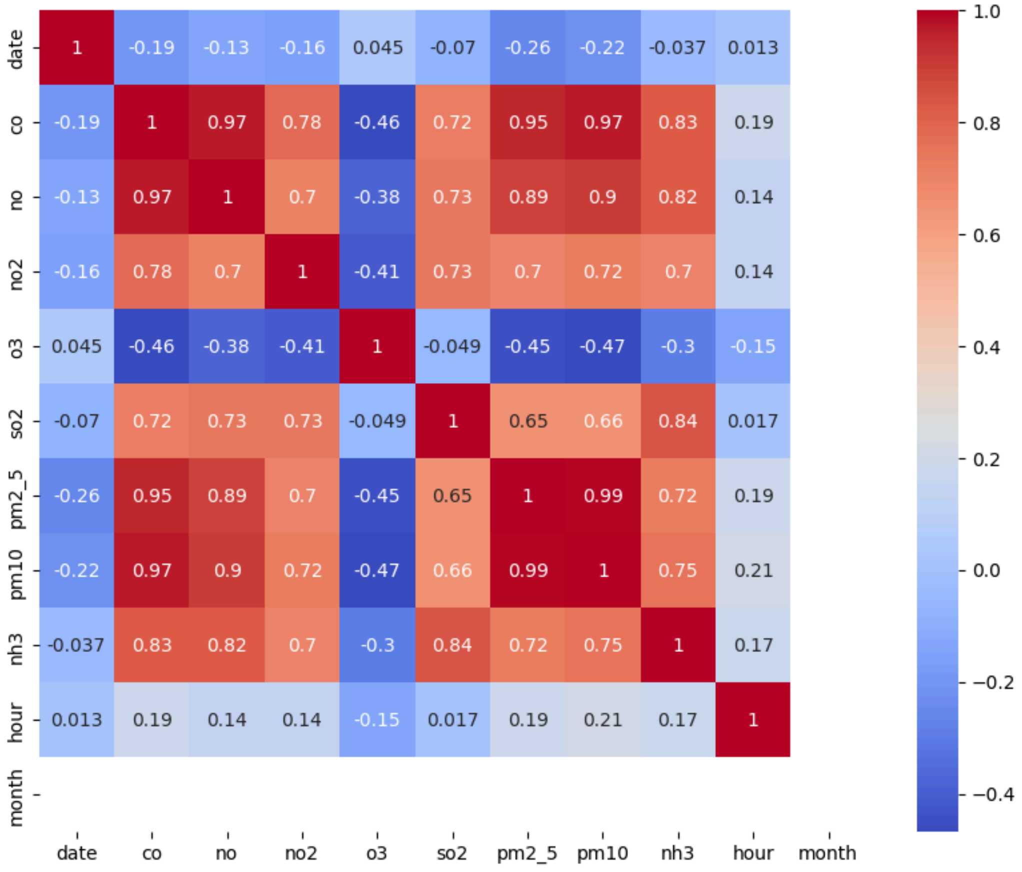
561 rows × 12 columns

```
df.drop('season', axis=1).set_index('date').resample('6h').mean().plot(subplots=True)
```

```
array([<Axes: xlabel='date'>, <Axes: xlabel='date'>,
       <Axes: xlabel='date'>, <Axes: xlabel='date'>,
       <Axes: xlabel='date'>, <Axes: xlabel='date'>,
       <Axes: xlabel='date'>, <Axes: xlabel='date'>,
       <Axes: xlabel='date'>, <Axes: xlabel='date'>], dtype=object)
```



```
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(10, 8))
sns.heatmap(df.drop('season', axis=1).corr(), annot=True, cmap='coolwarm')
plt.show()
```

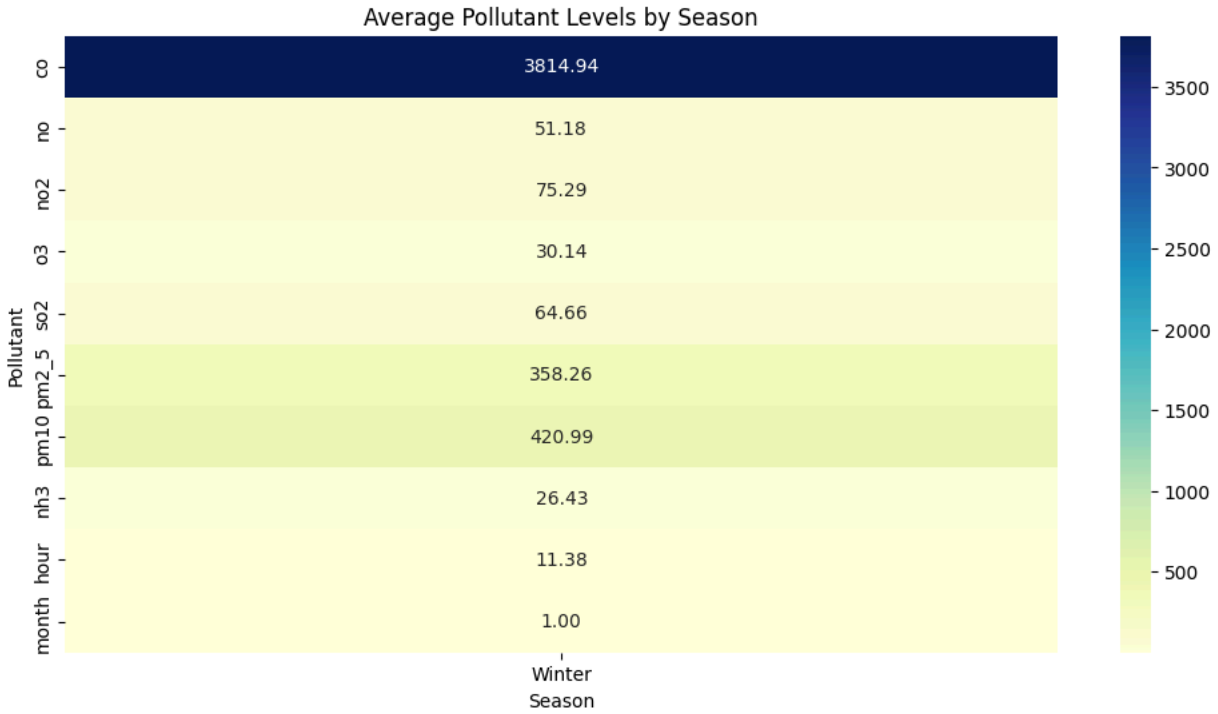


```
seasonal_mean = df.groupby('season').mean()
display(seasonal_mean)
```

	date	co	no	no2	o3	so2	pm2_5	pm10	nh3	hour	month
season											
Winter	2023-01-12 16:00:00	3814.94221	51.181979	75.292496	30.141943	64.655936	358.256364	420.988414	26.425062	11.379679	1.0

```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(12, 6))
sns.heatmap(seasonal_mean.drop('date', axis=1).T, annot=True, cmap='YlGnBu', fmt=".2f")
plt.title('Average Pollutant Levels by Season')
plt.xlabel('Season')
plt.ylabel('Pollutant')
plt.show()
```



Start coding or generate with AI.