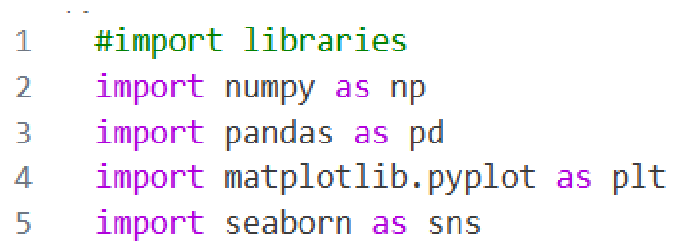
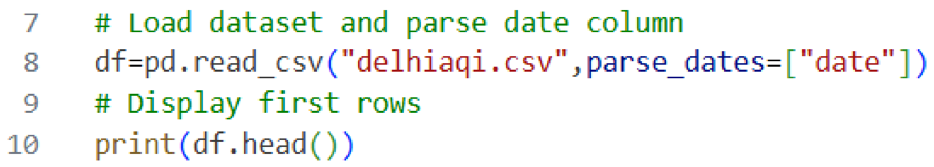
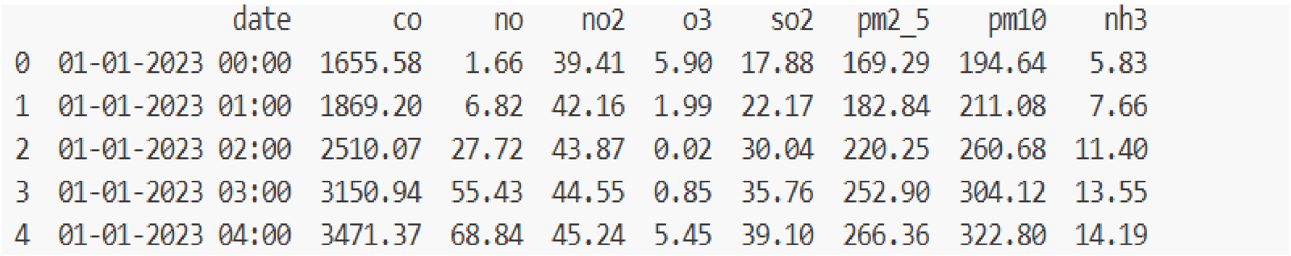


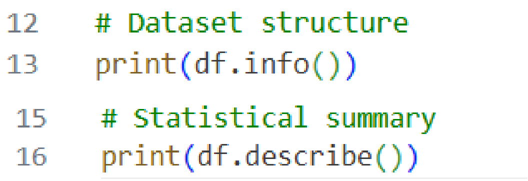
```python
#import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```



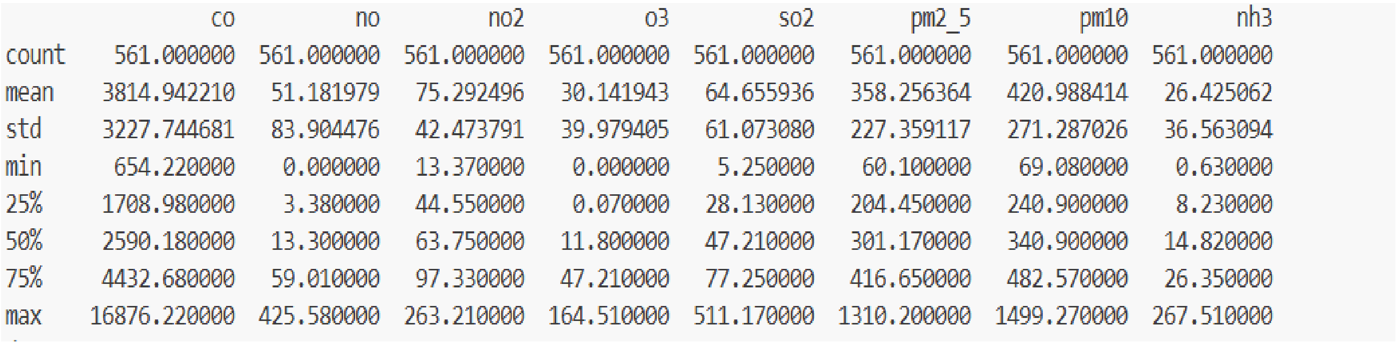
```python
# Load dataset and parse date column
df=pd.read_csv("delhiaqi.csv",parse_dates=["date"])
# Display first rows
print(df.head())
```



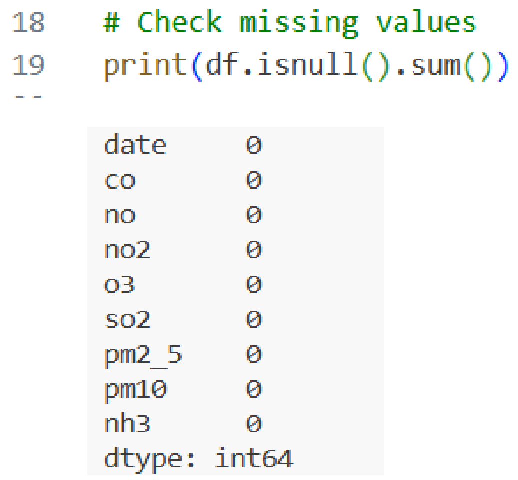
```
               date       co     no    no2    o3    so2   pm2_5    pm10    nh3
0  01-01-2023 00:00  1655.58   1.66  39.41  5.90  17.88  169.29  194.64   5.83
1  01-01-2023 01:00  1869.20   6.82  42.16  1.99  22.17  182.84  211.08   7.66
2  01-01-2023 02:00  2510.07  27.72  43.87  0.02  30.04  220.25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



```python
# Dataset structure
print(df.info())

# Statistical summary
print(df.describe())
```

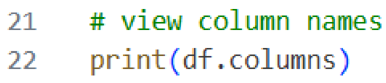


```
                co          no         no2          o3          so2        pm2_5         pm10          nh3
count    561.000000  561.000000  561.000000  561.000000  561.000000   561.000000   561.000000   561.000000
mean    3814.942210   51.181979   75.292496   30.141943   64.655936   358.256364   420.988414    26.425062
std     3227.744681   83.904476   42.473791   39.979405   61.073080   227.359117   271.287026    36.563094
min      654.220000    0.000000   13.370000    0.000000    5.250000    60.100000    69.080000     0.630000
25%     1708.980000    3.380000   44.550000    0.070000   28.130000   204.450000   240.900000     8.230000
50%     2590.180000   13.300000   63.750000   11.800000   47.210000   301.170000   340.900000    14.820000
75%     4432.680000   59.010000   97.330000   47.210000   77.250000   416.650000   482.570000    26.350000
max    16876.220000  425.580000  263.210000  164.510000  511.170000  1310.200000  1499.270000   267.510000
```

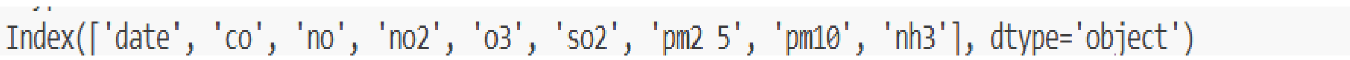


```python
# Check missing values
print(df.isnull().sum())
```

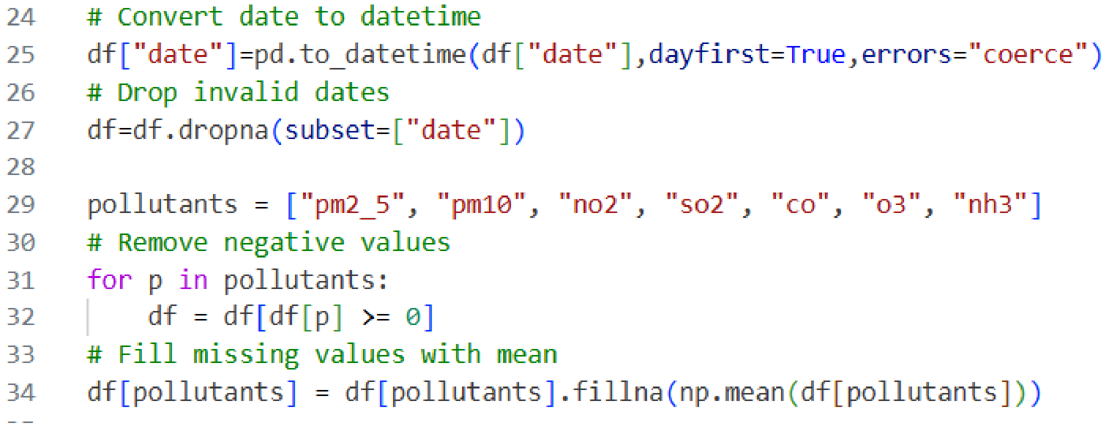
```
date       0
co         0
no         0
no2        0
o3         0
so2        0
pm2_5      0
pm10       0
nh3        0
dtype: int64
```



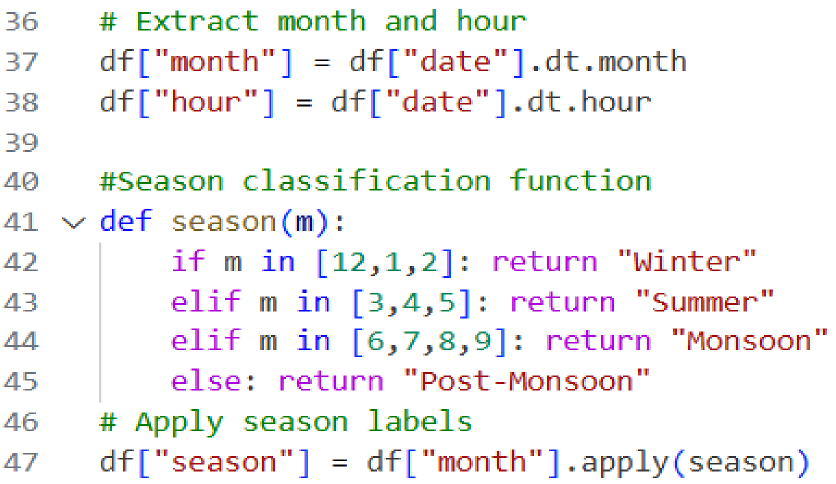
```python
# view column names
print(df.columns)
```



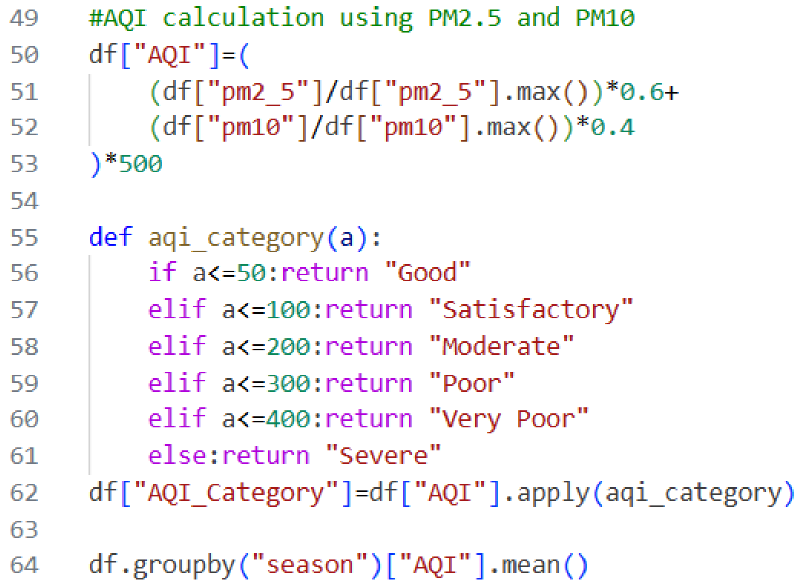
```
Index(['date', 'co', 'no', 'no2', 'o3', 'so2', 'pm2_5', 'pm10', 'nh3'], dtype='object')
```



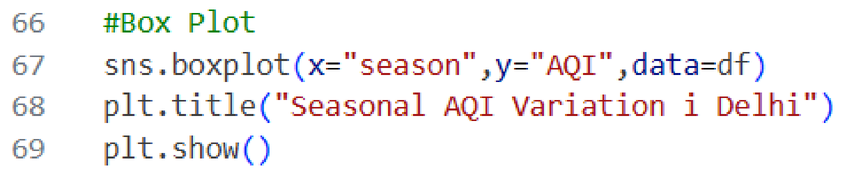
```python
# Convert date to datetime
df["date"]=pd.to_datetime(df["date"],dayfirst=True,errors="coerce")
# Drop invalid dates
df=df.dropna(subset=["date"])

pollutants = ["pm2_5", "pm10", "no2", "so2", "co", "o3", "nh3"]
# Remove negative values
for p in pollutants:
    df = df[df[p] >= 0]
# Fill missing values with mean
df[pollutants] = df[pollutants].fillna(np.mean(df[pollutants]))
```



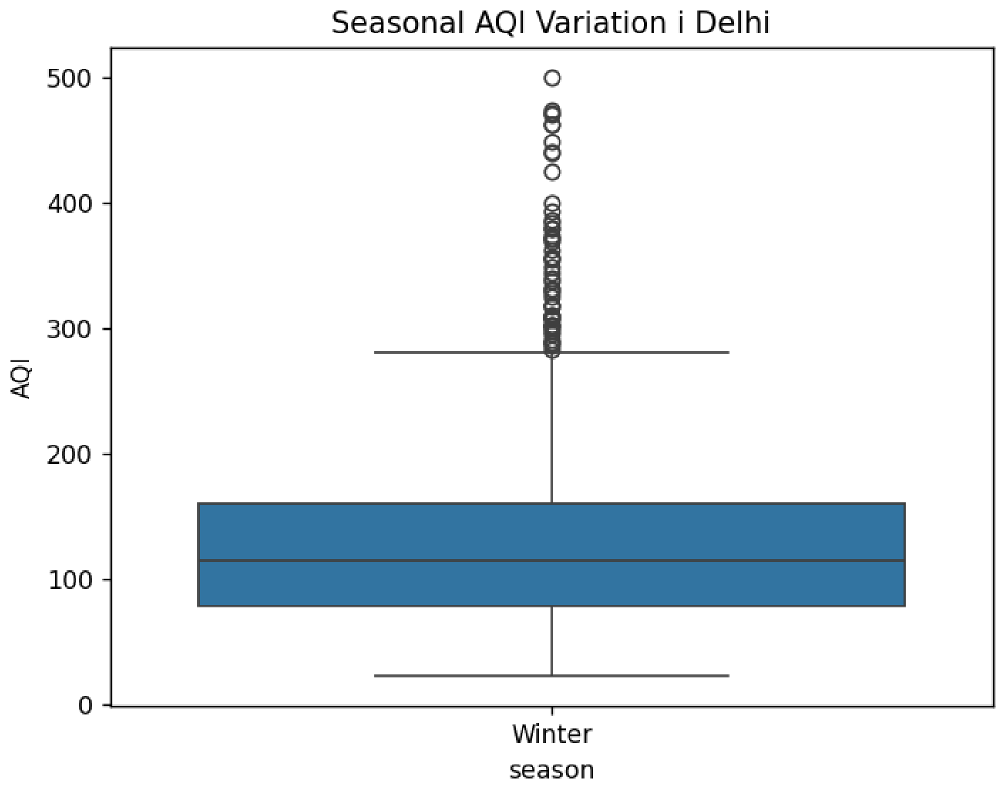
```python
# Extract month and hour
df["month"] = df["date"].dt.month
df["hour"] = df["date"].dt.hour

#Season classification function
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"
# Apply season labels
df["season"] = df["month"].apply(season)
```



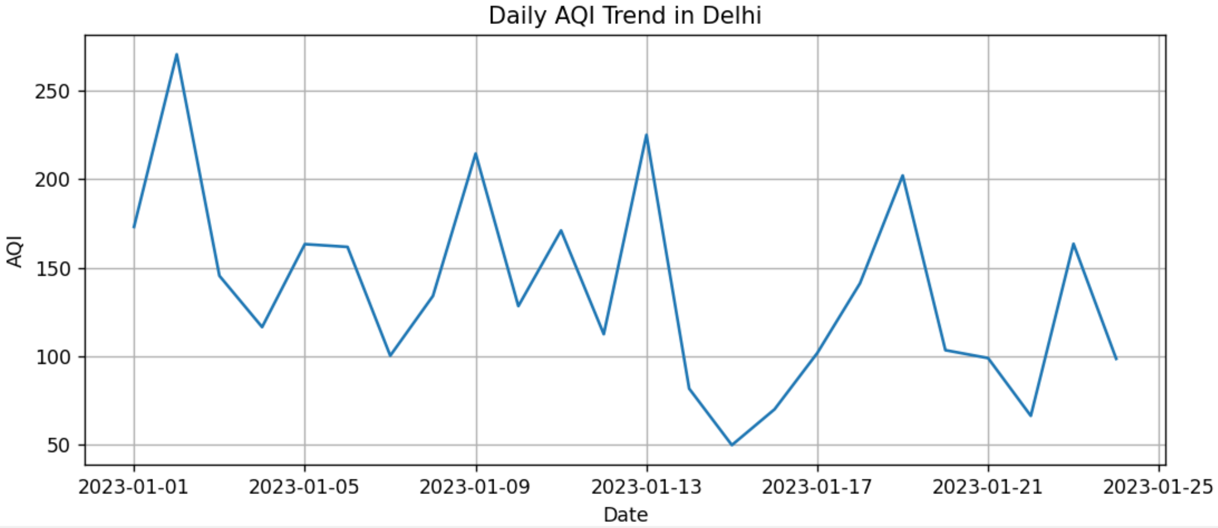
```
#AQI calculation using PM2.5 and PM10
df["AQI"]=(
    (df["pm2_5"]/df["pm2_5"].max())*0.6+
    (df["pm10"]/df["pm10"].max())*0.4
)*500

def aqi_category(a):
    if a<=50:return "Good"
    elif a<=100:return "Satisfactory"
    elif a<=200:return "Moderate"
    elif a<=300:return "Poor"
    elif a<=400:return "Very Poor"
    else:return "Severe"
df["AQI_Category"]=df["AQI"].apply(aqi_category)

df.groupby("season")["AQI"].mean()
```



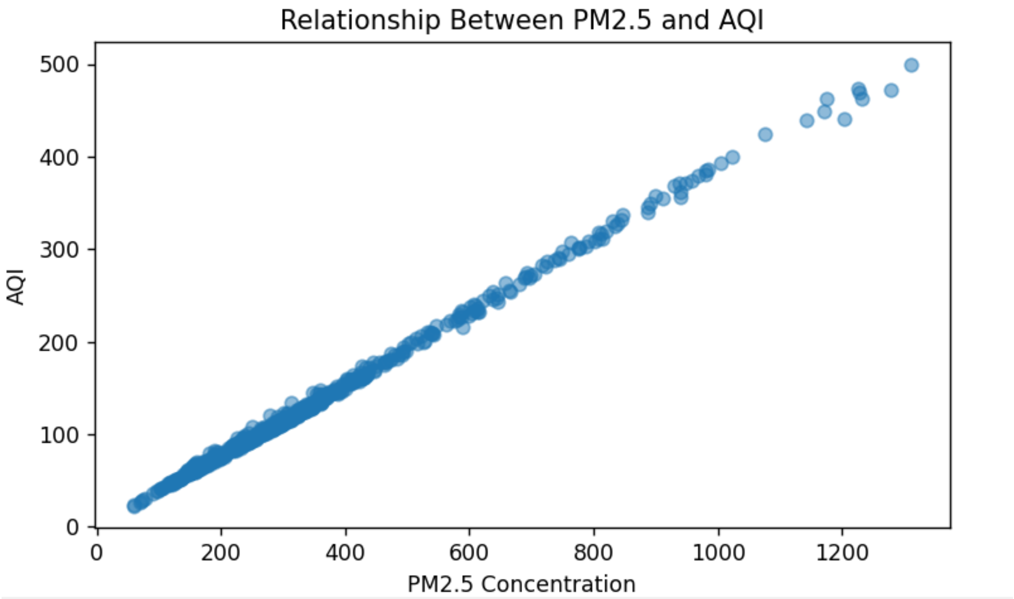
```
#Box Plot
sns.boxplot(x="season",y="AQI",data=df)
plt.title("Seasonal AQI Variation i Delhi")
plt.show()
```



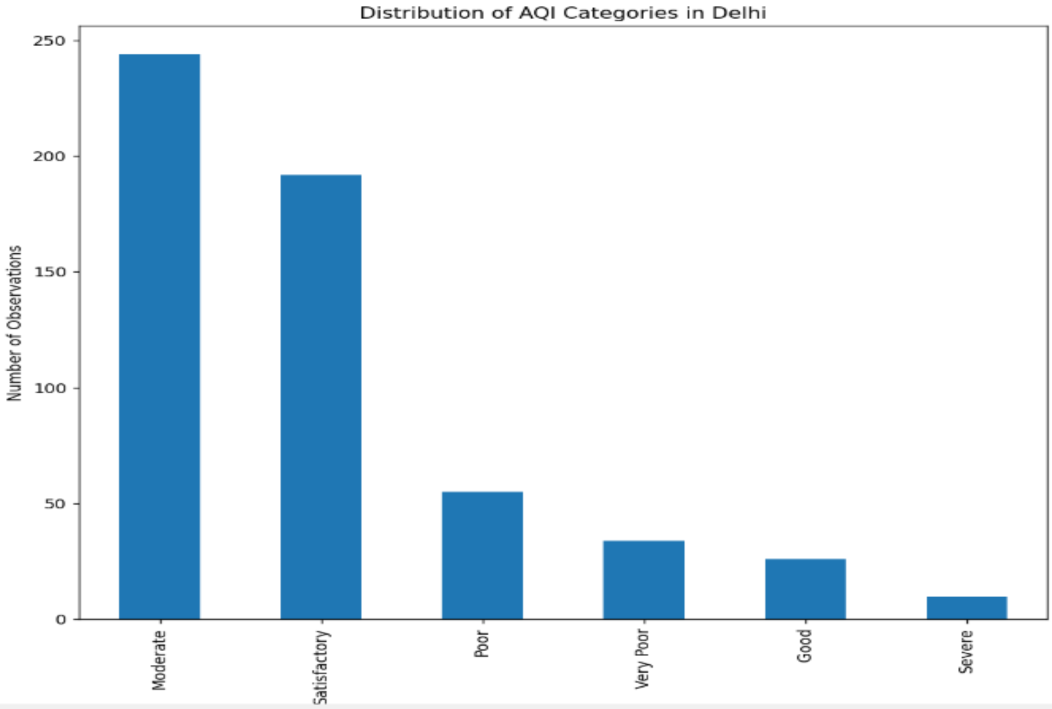
```
#Line Plot
daily_aqi=df.resample("D",on="date")["AQI"].mean()
plt.figure(figsize=(10,4))
plt.plot(daily_aqi)
plt.title("Daily AQI Trend in Delhi")
plt.xlabel("Date")
plt.ylabel("AQI")
plt.grid()
plt.show()
```

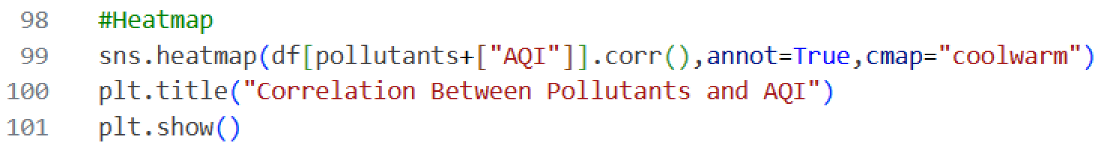


```
#Scatter Plot
plt.figure(figsize=(7,4))
plt.scatter(df["pm2_5"], df["AQI"], alpha=0.5)
plt.xlabel("PM2.5 Concentration")
plt.ylabel("AQI")
plt.title("Relationship Between PM2.5 and AQI")
plt.show()
```

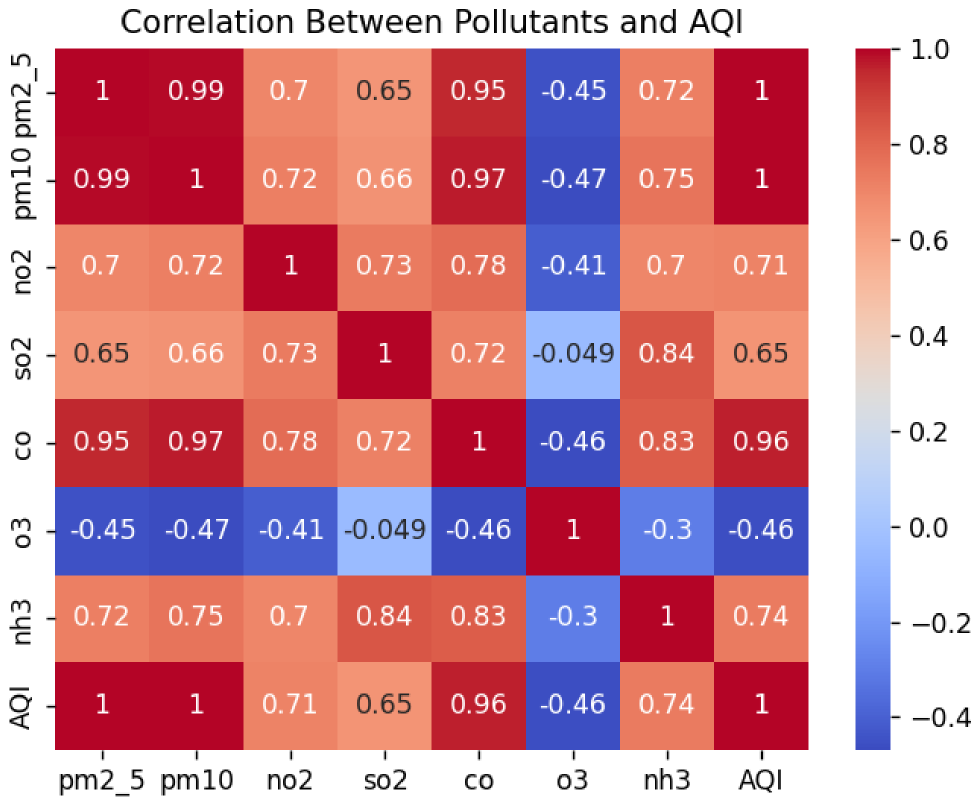


```python
#Bar Plot
aqi_counts = df["AQI_Category"].value_counts()
plt.figure(figsize=(7,4))
aqi_counts.plot(kind="bar")
plt.xlabel("AQI Category")
plt.ylabel("Number of Observations")
plt.title("Distribution of AQI Categories in Delhi")
plt.show()
```





```
#Heatmap
sns.heatmap(df[pollutants+["AQI"]].corr(),annot=True,cmap="coolwarm")
plt.title("Correlation Between Pollutants and AQI")
plt.show()
```


Correlation Between Pollutants and AQI
pm2_5 1 0.99 0.7 0.65 0.95 -0.45 0.72 1
pm10 0.99 1 0.72 0.66 0.97 -0.47 0.75 1
no2 0.7 0.72 1 0.73 0.78 -0.41 0.7 0.71
so2 0.65 0.66 0.73 1 0.72 -0.049 0.84 0.65
co 0.95 0.97 0.78 0.72 1 -0.46 0.83 0.96
o3 -0.45 -0.47 -0.41 -0.049 -0.46 1 -0.3 -0.46
nh3 0.72 0.75 0.7 0.84 0.83 -0.3 1 0.74
AQI 1 1 0.71 0.65 0.96 -0.46 0.74 1
pm2_5 pm10 no2 so2 co o3 nh3 AQI