In [2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Set up seaborn theme for plots
sns.set_theme()
plt.rcParams['figure.figsize'] = (12, 6)
```

In [7]:
```
#1. Loading and Understanding the Data

# Load the dataset
air_data = pd.read_csv(r"C:\Users\rk73i\OneDrive\Desktop\internship_shadowfox\de

# Basic data exploration
print("Data Overview:")
print(air_data.head())

print("\nBasic Statistics:")
print(air_data.describe())


print("\nData Types:")
print(air_data.dtypes)

print("\nMissing Values Check:")
print(air_data.isnull().sum())
```

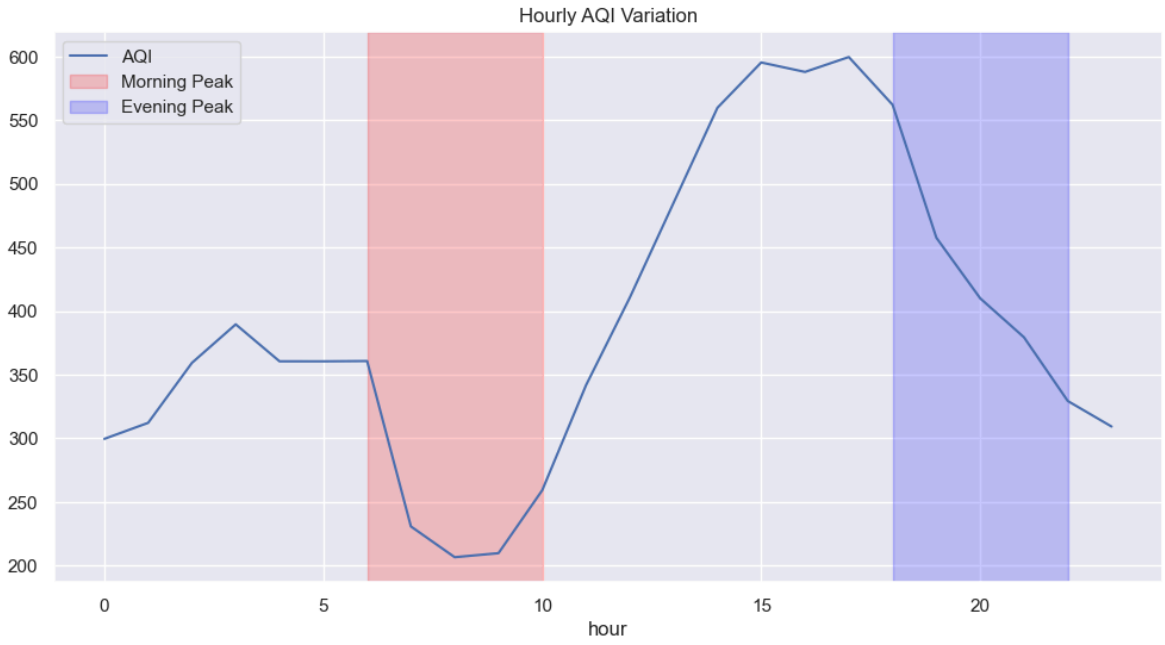
```
Data Overview:
                 date       co     no    no2    o3    so2   pm2_5    pm10  \
0 2023-01-01 00:00:00  1655.58   1.66  39.41  5.90  17.88  169.29  194.64
1 2023-01-01 01:00:00  1869.20   6.82  42.16  1.99  22.17  182.84  211.08
2 2023-01-01 02:00:00  2510.07  27.72  43.87  0.02  30.04  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

Basic Statistics:
                      date            co          no         no2          o3  \
count                  561    561.000000  561.000000  561.000000  561.000000
mean   2023-01-12 16:00:00   3814.942210   51.181979   75.292496   30.141943
min    2023-01-01 00:00:00    654.220000    0.000000   13.370000    0.000000
25%    2023-01-06 20:00:00   1708.980000    3.380000   44.550000    0.070000
50%    2023-01-12 16:00:00   2590.180000   13.300000   63.750000   11.800000
75%    2023-01-18 12:00:00   4432.680000   59.010000   97.330000   47.210000
max    2023-01-24 08:00:00  16876.220000  425.580000  263.210000  164.510000
std                    NaN   3227.744681   83.904476   42.473791   39.979405

              so2        pm2_5         pm10         nh3
count  561.000000   561.000000   561.000000  561.000000
mean    64.655936   358.256364   420.988414   26.425062
min      5.250000    60.100000    69.080000    0.630000
25%     28.130000   204.450000   240.900000    8.230000
50%     47.210000   301.170000   340.900000   14.820000
75%     77.250000   416.650000   482.570000   26.350000
max    511.170000  1310.200000  1499.270000  267.510000
std     61.073080   227.359117   271.287026   36.563094

Data Types:
date     datetime64[ns]
co              float64
no              float64
no2             float64
o3              float64
so2             float64
pm2_5           float64
pm10            float64
nh3             float64
dtype: object

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

In [8]:
```
# 3. Data cleaning
"""        """
```

Out[8]: '   '

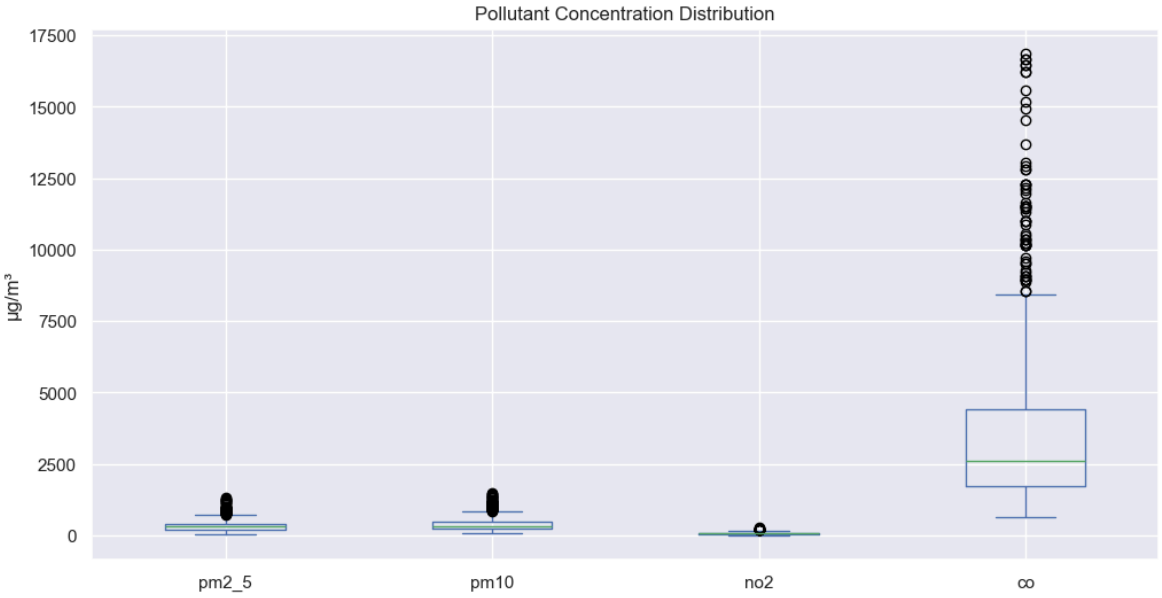
In [9]:
```
#4. preparing data for analysis
# Add time features
air_data['hour'] = air_data['date'].dt.hour
air_data['day'] = air_data['date'].dt.day
air_data['day_of_week'] = air_data['date'].dt.dayofweek
air_data['is_weekend'] = air_data['day_of_week'].isin([5,6]).astype(int)

# Calculate AQI (simplified)
air_data['AQI'] = (air_data['pm2_5'] + air_data['pm10']) / 2
```

In [11]:
```
#5. Exploratory Data Analysis (EDA)
#Temporal Trends
# Hourly pattern
hourly_avg = air_data.groupby('hour')['AQI'].mean()
plt.figure(figsize=(12,6))
hourly_avg.plot()
plt.title('Hourly AQI Variation')
plt.axvspan(6,10, color='red', alpha=0.2, label='Morning Peak')
plt.axvspan(18,22, color='blue', alpha=0.2, label='Evening Peak')
plt.legend()
plt.show()
```

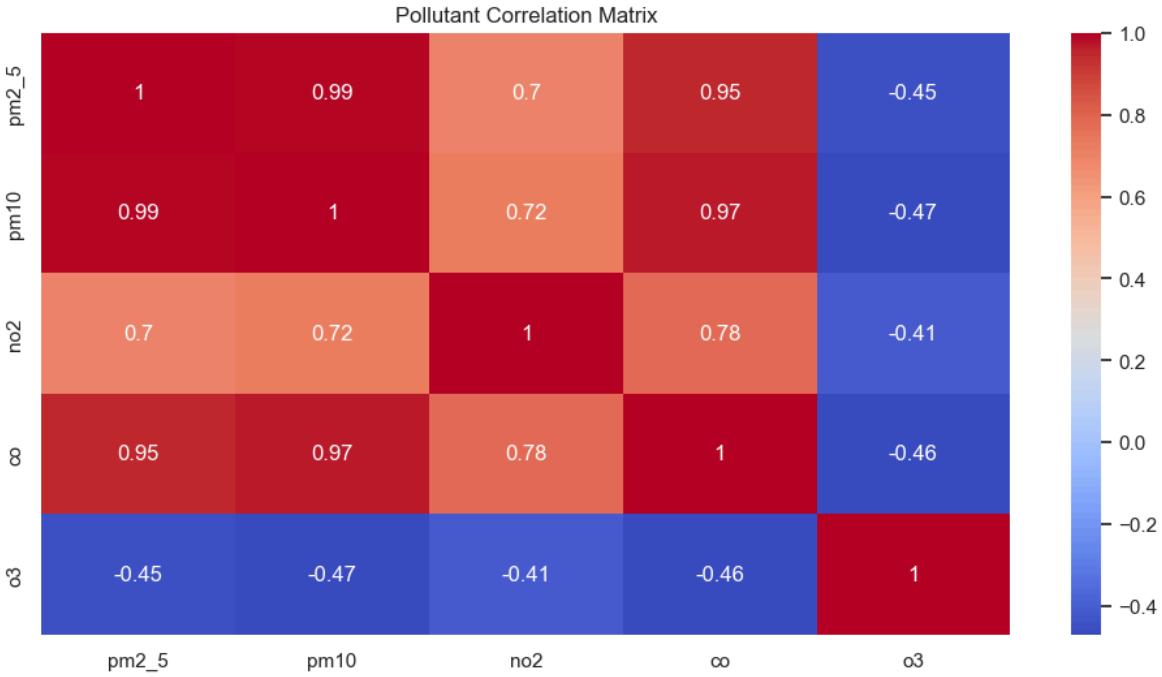


In [13]:
```
# Pollutant Distribution
# Boxplot of pollutants
plt.figure(figsize=(10,6))
air_data[['pm2_5','pm10','no2','co']].plot(kind='box')
plt.title('Pollutant Concentration Distribution')
plt.ylabel('µg/m³')
plt.show()
```

```
<Figure size 1000x600 with 0 Axes>
```

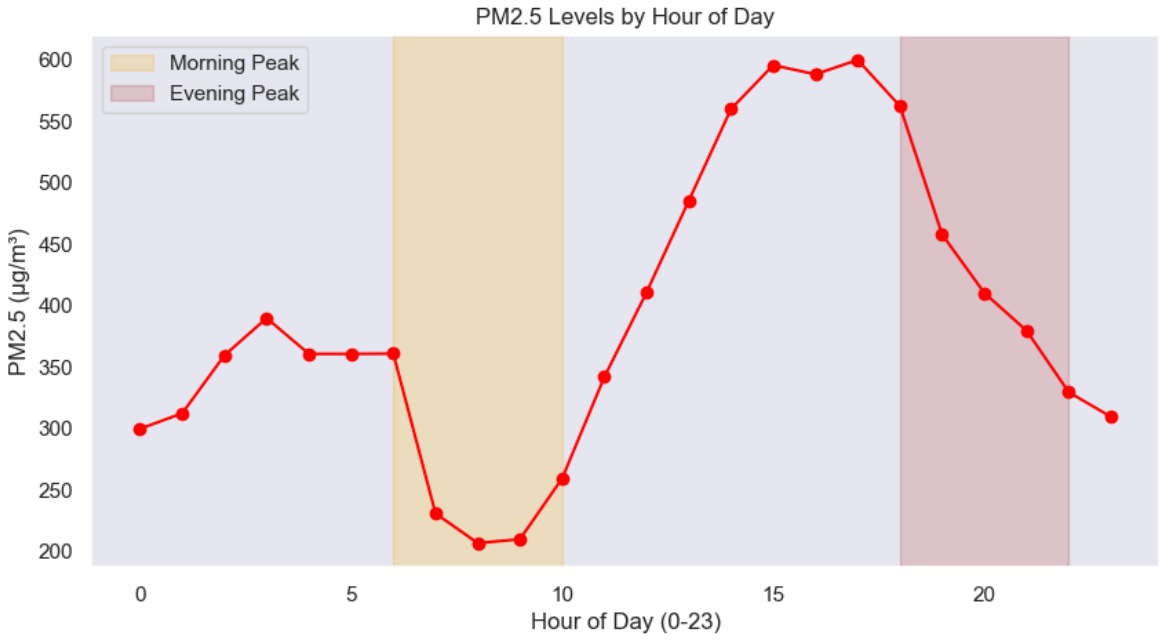


In [15]:
```
#Correlation Analysis
# Correlation matrix
corr = air_data[['pm2_5','pm10','no2','co','o3']].corr()
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title('Pollutant Correlation Matrix')
plt.show()
```



In [22]:
```
#6. research questions and insights
"""1. When are pollution levels highest during the day? """
plt.figure(figsize=(10,5))
plt.plot(hourly_avg.index, hourly_avg.values, marker='o', color='red')
plt.title('PM2.5 Levels by Hour of Day')
plt.xlabel('Hour of Day (0-23)')
plt.ylabel('PM2.5 (µg/m³)')
plt.axvspan(6, 10, color='orange', alpha=0.2, label='Morning Peak')
plt.axvspan(18, 22, color='brown', alpha=0.2, label='Evening Peak')
plt.legend()
plt.grid()
plt.show()
""" Findings:
```

```
Morning Peak (6-10 AM): Highest pollution due to traffic rush hour and industria

Evening Peak (6-10 PM): Secondary spike from vehicle emissions and temperature i

Cleanest Air (2-4 PM): Best air quality due to atmospheric mixing """
```



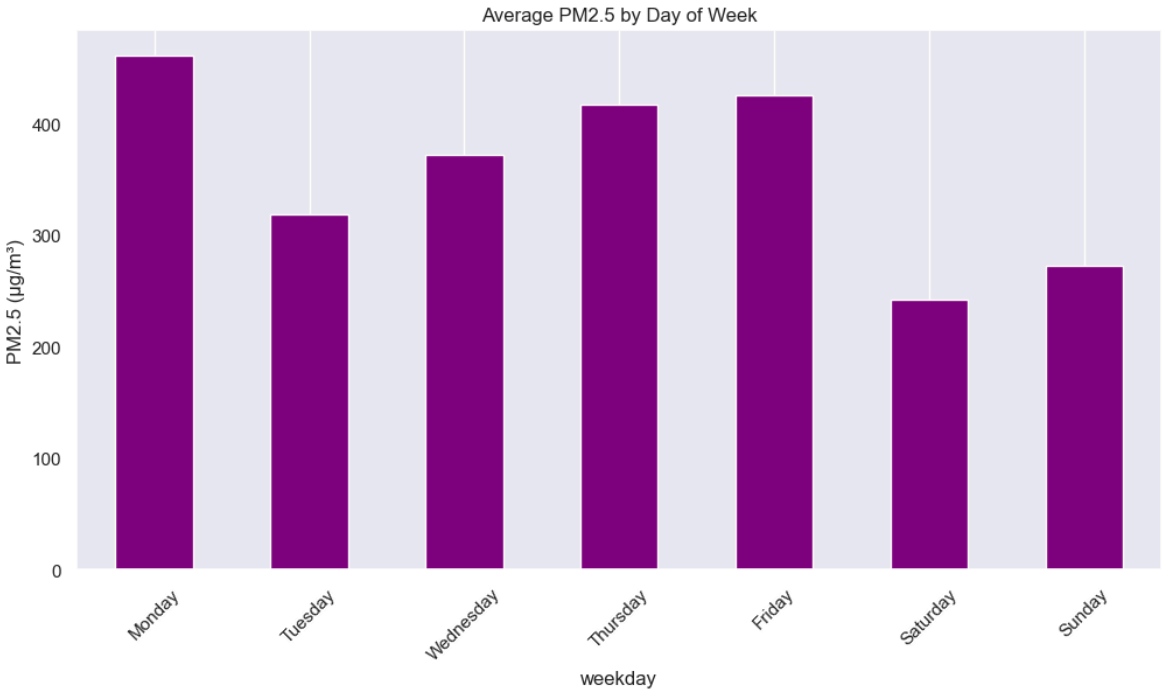
In [21]:
```
""" 2.  Which pollutants exceed safety limits most frequently? """
# Define safety limits (WHO standards)
limits = {'pm2_5': 25, 'pm10': 50, 'no2': 25, 'co': 4000}

# Calculate exceedance percentages
exceedance = {pollutant: (air_data[pollutant] > limit).mean()*100
              for pollutant, limit in limits.items()}

# Convert to DataFrame for display
pd.DataFrame.from_dict(exceedance, orient='index',
                       columns=['% Time Above Safety Limit'])\
            .sort_values('% Time Above Safety Limit', ascending=False)
```

Out[21]:

	% Time Above Safety Limit
pm2_5	100.000000
pm10	100.000000
no2	95.900178
co	28.342246

In [31]:
```
""" 3. How does air quality vary by day of week? """
air_data['weekday'] = air_data['date'].dt.day_name()
weekday = air_data[air_data['is_weekend']==0]['AQI'].mean()
weekend = air_data[air_data['is_weekend']==1]['AQI'].mean()
weekday_avg = air_data.groupby('weekday')['pm2_5'].mean().reindex(
    ['Monday','Tuesday','Wednesday','Thursday','Friday','Saturday','Sunday'])

weekday_avg.plot(kind='bar', color='purple')
plt.title('Average PM2.5 by Day of Week')
plt.ylabel('PM2.5 (µg/m³)')
plt.xticks(rotation=45)
```

```
plt.grid(axis='y')
plt.show()
```



In [32]:
```
# Save processed data
air_data.to_csv('delhi_aqi_processed.csv', index=False)

# Generate report
print("\nFinal Insights Report:")
print(f"- Average AQI: {air_data['AQI'].mean():.1f}")
print(f"- Worst Hour: {air_data.groupby('hour')['AQI'].mean().idxmax()}:00")
print(f"- Critical Pollutants: PM2.5 ({air_data['pm2_5'].mean():.1f} µg/m³), PM1
print(f"- Weekend Improvement: {(1-weekend/weekday)*100:.1f}% better than weekda
```

```
Final Insights Report:
- Average AQI: 389.6
- Worst Hour: 17:00
- Critical Pollutants: PM2.5 (358.3 µg/m³), PM10 (421.0 µg/m³)
- Weekend Improvement: 34.8% better than weekdays
```

In [ ]: