**Air Pollution Tracker Report**

**Prepared by:** Rupali Gawate  
**Date:** 22-12-2024

**Table of Contents**

1. Introduction
2. Overview of Data
3. Data Cleaning and Preprocessing
4. Visualizations and Insight
5. Problem Statements
6. Conclusion
7. Recommendations
8. Refferences

**Introduction:**

Air pollution is a pressing public health concern that demands immediate action. Among various pollutants, Ground-level Ozone (O3) plays a pivotal role in this crisis. Elevated O3 levels contribute not only to acute respiratory conditions but also to long-term environmental degradation, such as harm to ecosystems and climate change. Ground-level O3 pollution is particularly concerning in the United States, where it affects the air quality across the country. In recent years, O3 pollution has been exacerbated by various factors, including wildfires that cause simultaneous increases in fine particulate matter and ground-level O3 through the direct emission of ozone precursor compounds in smoke plumes, particularly in the United States. Changes in regional climate and globally enhanced ozone are projected to increase ground-level ozone over most of the United States. Timely monitoring and prediction of O3 levels are critical for targeted policy interventions to mitigate the adverse impacts on public health and the environment

**Importance of the Study:**

The urgency of tackling O3 pollution is highlighted by its rapid changes due to factors like temperature and sunlight. This study is designed to provide actionable insights by focusing on specific geographic locations that have experienced notable increases in O3 levels recently, such as `4545 Navajo St., CO` and `2500 1ST STREET, N.W. WASHINGTON DC`. This targeted strategy allows for more effective and immediate policy action.

**Problem Statement & Methodology:**

Air pollution is an urgent and escalating public health issue, with Ground-level Ozone (O3) being a primary contributor to this crisis. Elevated levels of O3 not only exacerbate respiratory conditions but also contribute to long-term environmental degradation, including harm to ecosystems and climate change. Effective monitoring and prediction of O3 levels are critical for timely interventions by policymakers.

To combat the multi-faceted issue of O3 pollution, this project aims to analyze, model, and predict changes in AQI (Air Quality Index) levels of O3 at specific geographic locations that have experienced notable increases in O3 levels recently. This targeted approach will provide actionable insights for immediate policy reform.

Our goal is to analyze, model, and forecast changes in AQI (Air Quality Index) levels of O3 at the specific addresses mentioned earlier. We intend to employ a train-test split methodology to select the most effective time-series model for forecasting.

**ARIMA (Autoregressive Integrated Moving Average):**ARIMA is a widely-used statistical method for time series forecasting that can capture various patterns in temporal data. It's composed of three parts: autoregressive (AR), which uses the relationship between an observation and a number of lagged observations; integrated (I), which involves differencing the data to make it stationary; and moving average (MA), which uses the dependency between an observation and a residual error from a moving average model applied to lagged observations. ARIMA is best suited for data sets that show a clear trend or pattern over time, making it effective for long-term forecasting without seasonality.

**SARIMAX (Seasonal Autoregressive Integrated Moving-Average with Exogenous Variables):** This model extends ARIMA by adding support for seasonality and external variables. The seasonal part of the model accounts for seasonal variations or patterns that repeat at regular intervals, making it more applicable to data with clear seasonal trends. The exogenous component allows the model to incorporate the influence of external factors or independent variables that might affect the forecast variable. SARIMAX is highly useful in scenarios where both seasonal patterns and external influences play a significant role in the observed time series data.

**Holt-Winters:** Also known as Triple Exponential Smoothing, this technique is well-suited for data with trends and seasonal patterns. Holt-Winters method decomposes the time series into three components: level (the average value in the series), trend (the increasing or decreasing value in the series), and seasonality (the repeating short-term cycle in the series). It then applies exponential smoothing to each of these components. The model is particularly effective for short-term forecasts where seasonality is a significant factor, and it can be adjusted to capture both additive and multiplicative seasonality, making it flexible for different types of seasonal patterns.

Through detailed modelling of O3 levels, we aim to develop a reliable forecasting tool. This tool will help understand and predict future O3 levels at targeted addresses, providing a data-driven basis for policy reforms. By pinpointing peak pollution times, our models aspire to facilitate timely interventions, contributing to the broader goal of mitigating the health and environmental impacts of O3 pollution.

**Public Health Impact:** O3, or ground-level ozone, is harmful to breathe and is especially a concern for certain vulnerable groups, including children, the elderly, and those with certain health conditions. An "F" grade from the American Lung Association signifies that the air quality in a particular area is poor due to high levels of ozone pollution. In recent years, over 100 million Americans have been living in areas with such poor air quality, marked by an "F" grade for ozone smog, which emphasizes the serious public health impact of O3 pollution.

**Comprehensive Exploratory Data Analysis (EDA):**

**Data Distributions**: Conduct a comprehensive Exploratory Data Analysis (EDA) on AQI trends at these locations.

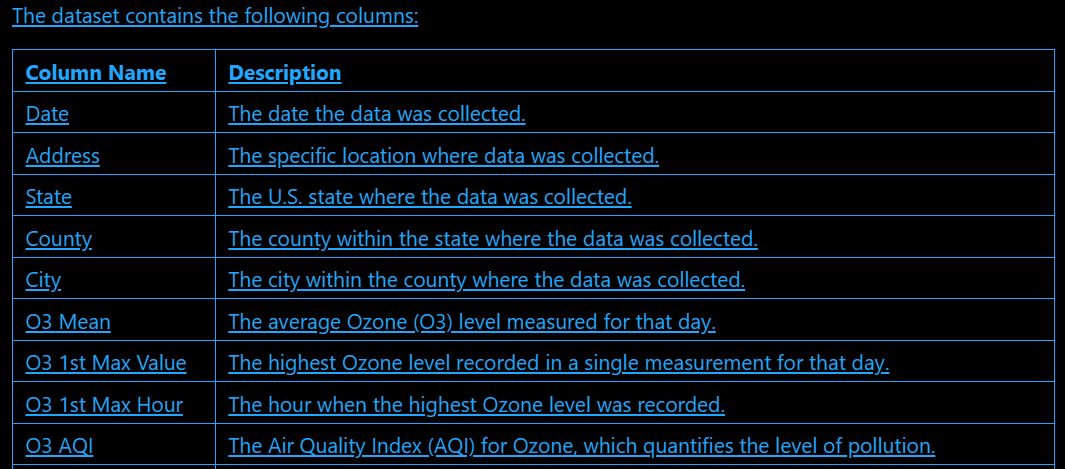
**Modelling** Objectives: Employ a train-test split methodology to identify the most effective model for forecasting future O3 levels.

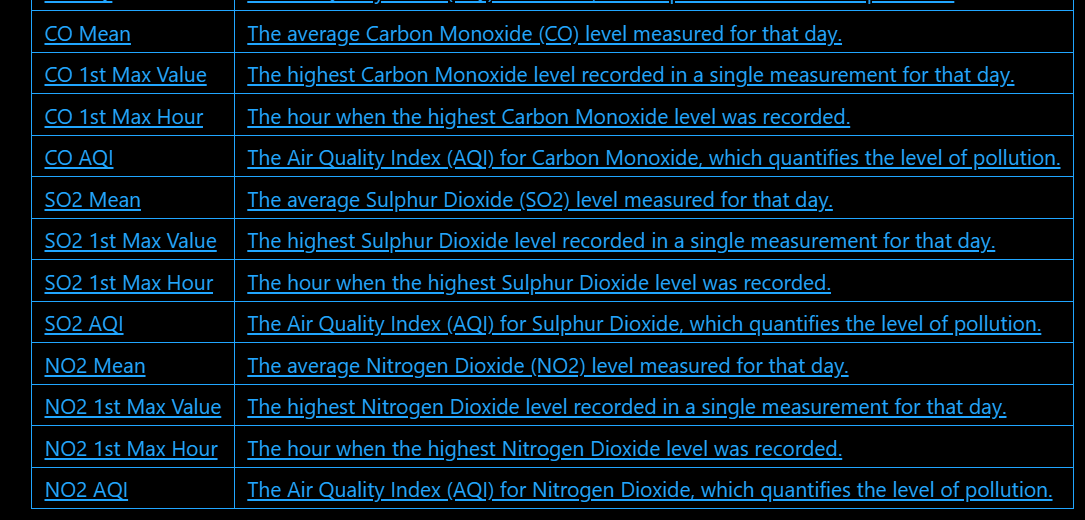
This project has provided a comprehensive analysis of ground-level Ozone (O3) Air Quality Index (AQI) levels at two specific location.

**Overview of Data:**

This dataset initially comprised approximately 665,414 observations across 21 columns. It has since been expanded to incorporate data up to the year 2023, offering a thorough historical overview of air quality in the United States. This dataset continues to be updated. It covers the period from 2000 to 2023, making it an asset for ongoing research and analysis.

**Column descriptions**: The dataset contains the following columns:





**3) Data Cleaning and Preprocessing:**

Finding the nulls: data.isnull()

Handle Duplicates: print(data.duplicated().sum())

data

Handle Outliers: Q1 = data['O3 AQI'].quantile(0.25)

Q3 = data['O3 AQI'].quantile(0.75)

IQR = Q3 - Q1

# Define outlier thresholds

lower\_bound = Q1 - 1.5 \* IQR

upper\_bound = Q3 + 1.5 \* IQR

# Filter out outliers

data = data[(data['O3 AQI'] >= lower\_bound) & (data['O3 AQI'] <= upper\_bound)]

To find last Five Rows: data.tail()

Training and Testing Data : from sklearn.model\_selection import train\_test\_split

X = data.drop('O3 AQI', axis=1) # Features

y = data['O3 AQI'] # Target

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

print("Training Set Shape:", X\_train.shape)

print("Testing Set Shape:", X\_test.shape)

To find Data types of the data : print(X\_train.dtypes)

Training the model: model = LinearRegression()

model.fit(X\_train, y\_train)

print("Model trained successfully!")

Predicting the model: # Predict on the test set

y\_pred = model.predict(X\_test)

# Calculate metrics

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

print("Mean Squared Error (MSE):", mse)

print("R² Score:", r2)

To find Coefficient and Intercept : # Coefficients (importance of each feature)

print("Coefficients:", model.coef\_)

# Intercept

print("Intercept:", model.intercept\_)

4) **Visualizations and Insight:**

import matplotlib.pyplot as plt

residuals = y\_test - y\_pred

plt.scatter(y\_test, residuals)

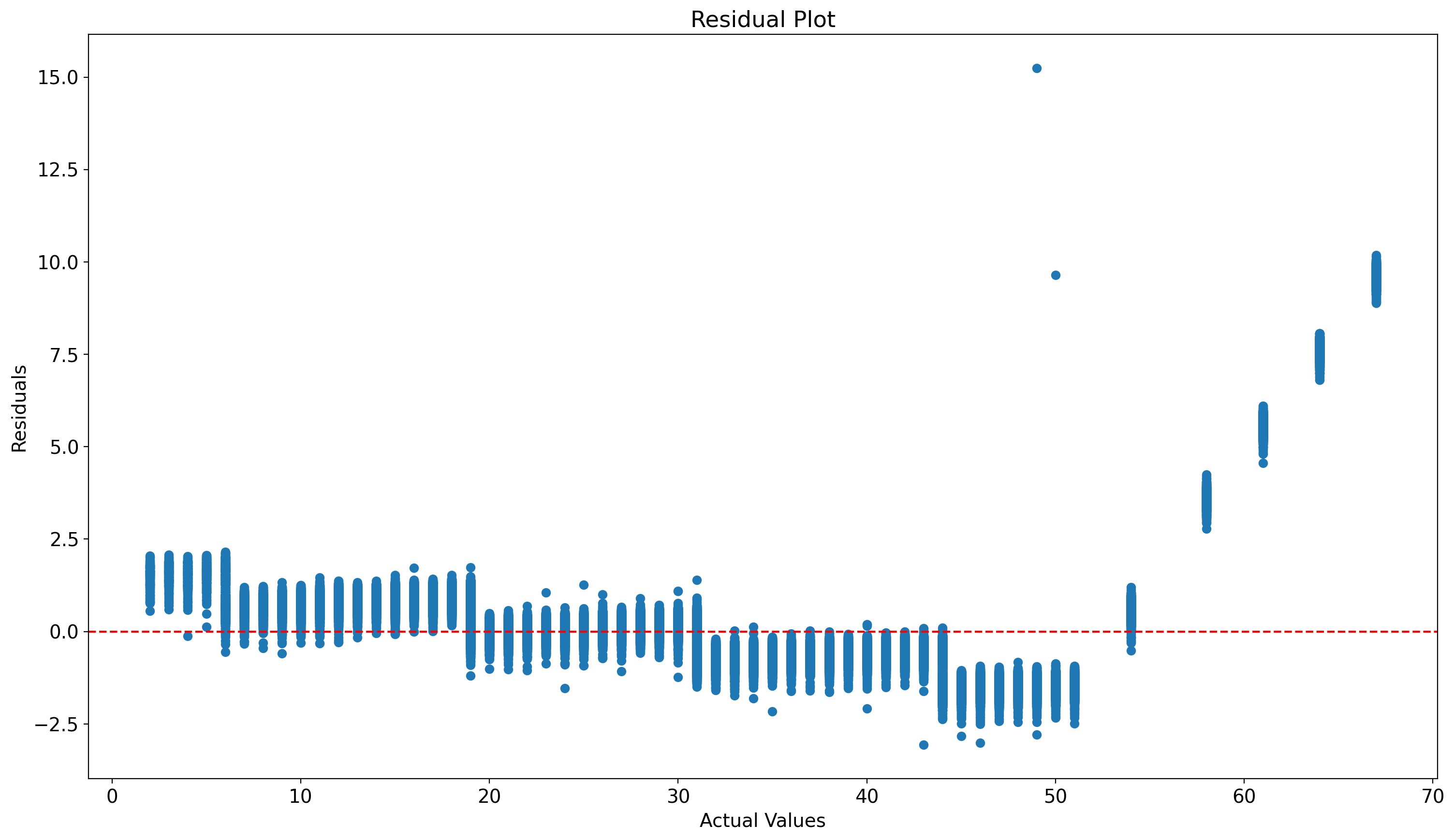
plt.axhline(y=0, color='r', linestyle='--')

plt.xlabel("Actual Values")

plt.ylabel("Residuals")

plt.title("Residual Plot")

plt.show()

****

This chart is a Residual Plot, which is used to evaluate the performance of a regression model. **X-Axis (Actual Values):** Represents the actual observed values of the target variable (y\_actual). **Y-Axis (Residuals):** Represents the residuals, which are the differences between the predicted and actual values of the target variable **Points:** Each point represents one observation in the dataset. **Red Dashed Line:** The horizontal red line represents a residual of 0. Ideally, residuals should scatter randomly around this line. **Clustered Residuals Around Zero:** In many places, the residuals are close to the red dashed line, indicating that the model predictions are fairly accurate in those regions.**Patterns in Residuals:** At the higher end of the actual values (right side of the chart), the residuals are consistently positive, which indicates **systematic underprediction** by the model.

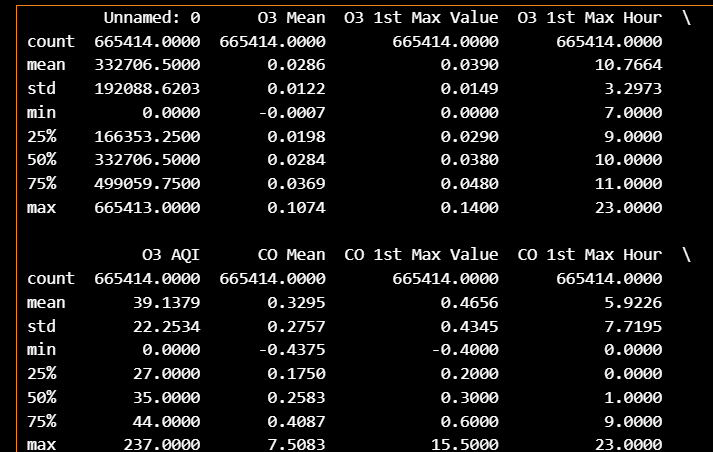
**5)Problem Statements :**

#1) Compute the mean, median, and standard deviation for each numeric column in the dataset.

# Summary statistics

stats = df.describe()

print(stats)

****

Above table is showing the all numeric means statistics columns like mean, mode and median. Standard deviation, count, quantiles minimum value maximum value etc.

**#2) Identify and handle any missing values in the dataset.**

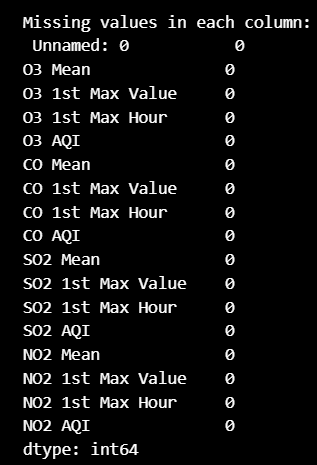
# Check for missing values

missing\_values = df.isnull().sum()

print("Missing values in each column:\n", missing\_values)

# Drop rows with missing values

df\_cleaned = df.dropna()



From the above we are checking for missing values and then finding the is there any null columns available or not.From the above table we can see that there is no any null value is present in the given dataset.

**# 3) Visualize the distribution of the target variable to understand its range and skewness**.

import matplotlib.pyplot as plt

import seaborn as sns

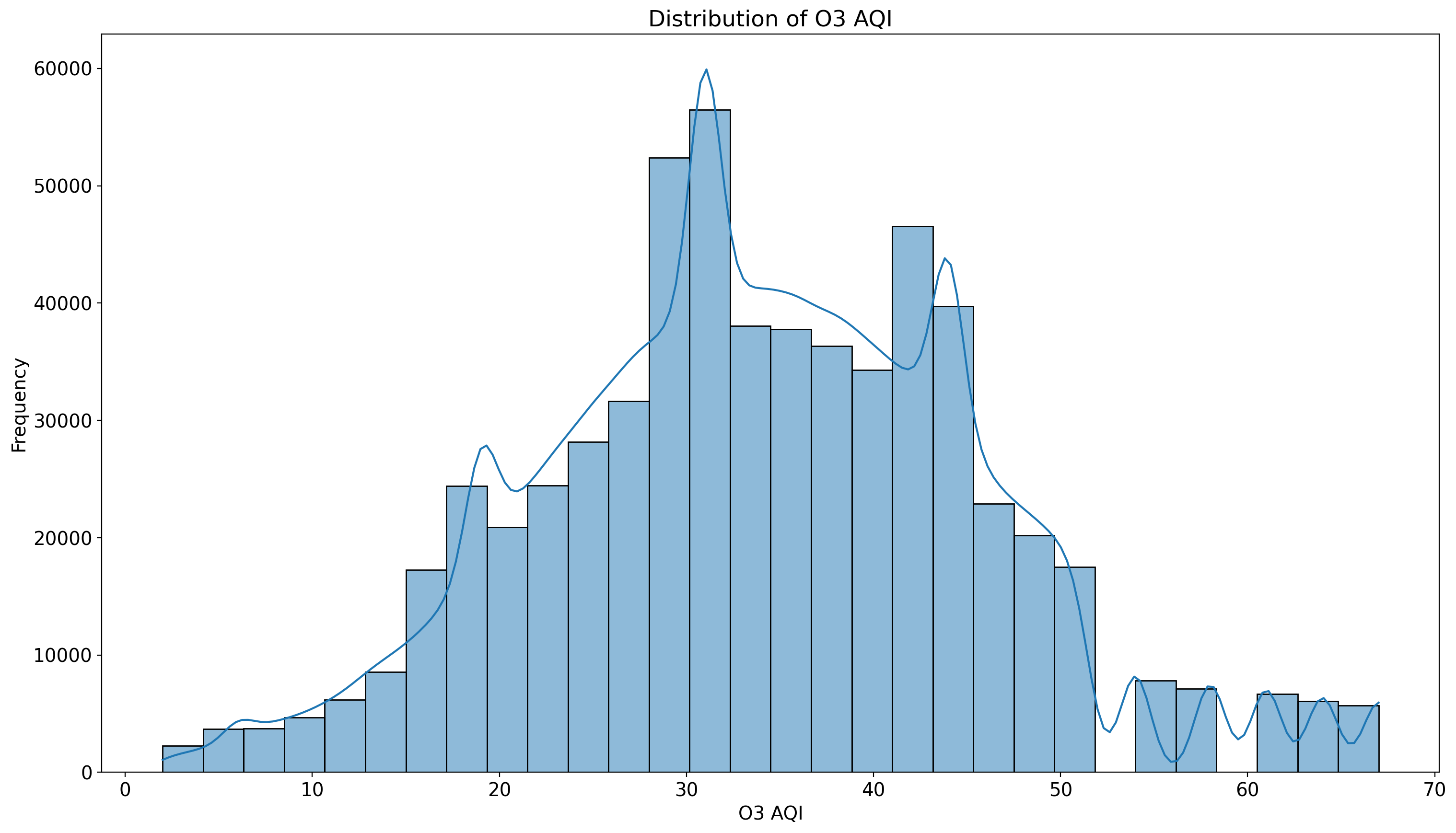
sns.histplot(data['O3 AQI'], kde=True, bins=30)

plt.title('Distribution of O3 AQI')

plt.xlabel('O3 AQI')

plt.ylabel('Frequency')

plt.show()



The above histogram is of distribution of O3 AQI and its frequency. Here O3 AQI is the target variable. Understanding the target's distribution helps choose appropriate transformations or detect outliers.

**# 4) Find which features are most correlated with the target column (O3 AQI).**

# Drop non-numeric columns

numeric\_data = data.select\_dtypes(include=['float64', 'int64'])

# Calculate the correlation matrix

correlation\_matrix = numeric\_data.corr()

# Display correlations with 'O3 AQI'

if 'O3 AQI' in correlation\_matrix.columns:

print(correlation\_matrix['O3 AQI'].sort\_values(ascending=False))

else:

print("Column 'O3 AQI' is not in the dataset after selecting numeric columns.")



The above table is showing that first three rows that are O3 AQI, O3 1st max value and O3 Mean are hightly correlated with each other than other. Highly correlated features can have a significant impact on predictions and should be prioritized.

**# 5) Build a simple linear regression model to predict O3 AQI.**

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

# Drop non-numeric or irrelevant columns

df = data.drop(['Date', 'Address', 'State', 'County', 'City'], axis=1)

# Ensure all columns are numeric

data = data.select\_dtypes(include=['float64', 'int64'])

# Separate features and target

X = data.drop(['O3 AQI'], axis=1)

y = data['O3 AQI']

# Split the dataset into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Train the model

model = LinearRegression()

model.fit(X\_train, y\_train)

# Make predictions

y\_pred = model.predict(X\_test)

# Calculate Mean Squared Error

mse = mean\_squared\_error(y\_test, y\_pred)

print("Mean Squared Error:", mse)

Mean Squared Error: 2.486806499526967

Explanation:- The problem statement is saying to build a simple linear regression model.

So we have to import all necessary libraries with linear regression from sklearn. Then split the dataset into train and testing sets with test size 20% and random state of 42. We have to train the model to fit. Then make some prdictions for testing and then calculate the mean squared error. This problem introduces a predictive model and evaluates its performance using MSE.

**# 6) Create a scatter plot to examine the relationship between O3 AQI and NO2 AQI.**

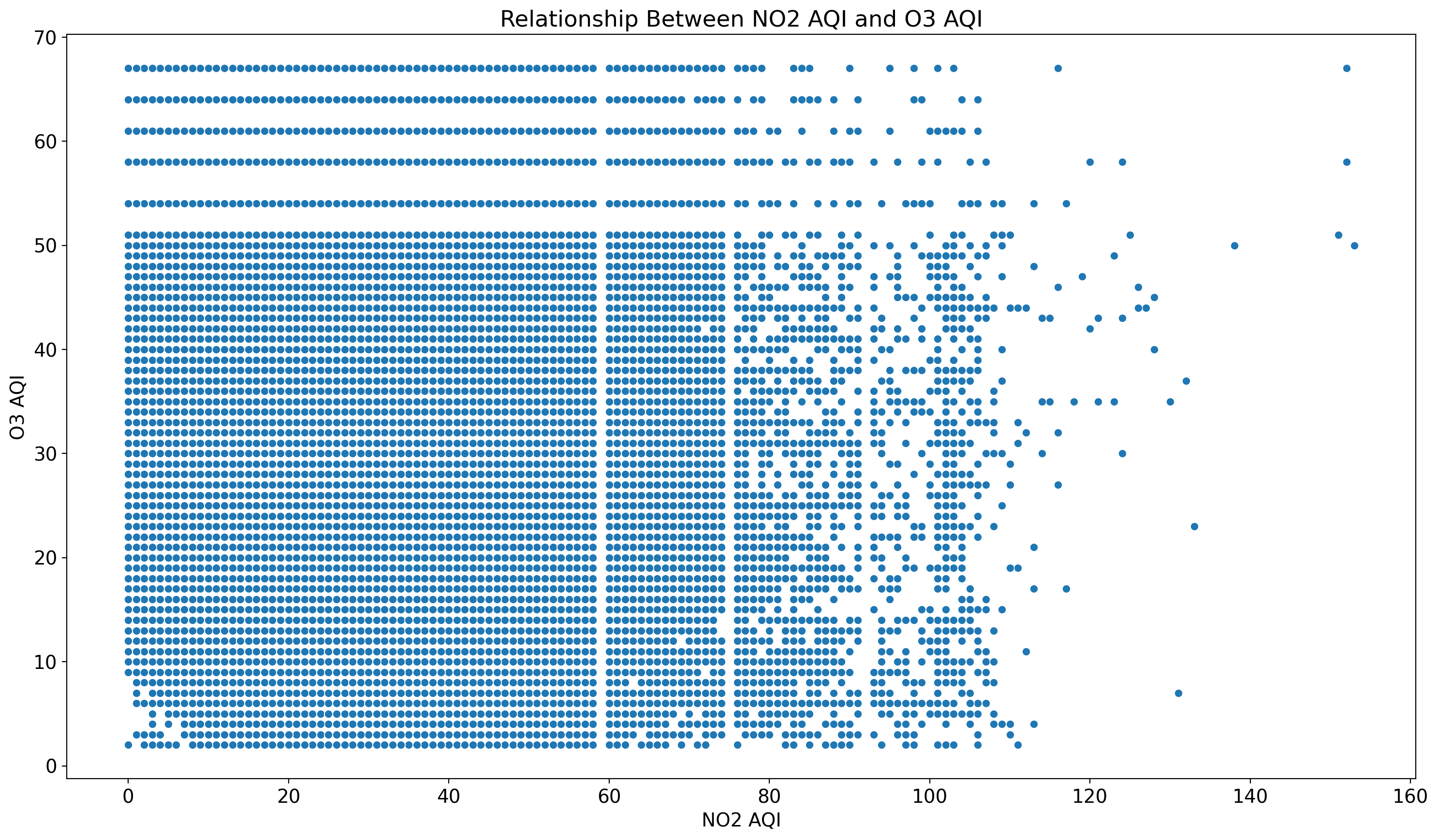
sns.scatterplot(data=data, x='NO2 AQI', y='O3 AQI')

plt.title('Relationship Between NO2 AQI and O3 AQI')

plt.xlabel('NO2 AQI')

plt.ylabel('O3 AQI')

plt.show()



Scatter plots reveal potential relationships or trends between variables. That variables are O3 AQI and NO2 AQI (Nitrogen Dioxide Air Quality Index) on the x-axis and **O3 AQI** (Ozone Air Quality Index) on the y-axis.

**Dense Region (Lower AQI Values)**:

* There is a dense cluster of points where both NO2 AQI and O3 AQI values are relatively low (NO2 AQI below 100, O3 AQI below 50).
* This indicates that most observations in the dataset are concentrated in regions with moderate or low pollution levels for these pollutants.

**Spread for High NO2 Values**:

* As the NO2 AQI increases (above 100), the points spread out more widely on the y-axis, showing varying O3 AQI levels.
* This could imply that higher NO2 concentrations do not directly correlate with higher or lower O3 concentrations.

**Potential Negative Correlation**:

* There appears to be a weak inverse trend where high NO2 AQI values correspond to relatively lower O3 AQI values. This may indicate that nitrogen dioxide might inhibit ozone formation under certain conditions.

**# 7) Analyze how O3 AQI changes over time.**

# Use only the first 10,000 rows for date generation

data\_subset = data.head(10000)

data\_subset['Date'] = pd.date\_range(start='2000-01-01', periods=len(data\_subset))

# Sort data by Date

data\_sorted = data\_subset.sort\_values(by='Date')

# Plot O3 AQI over time

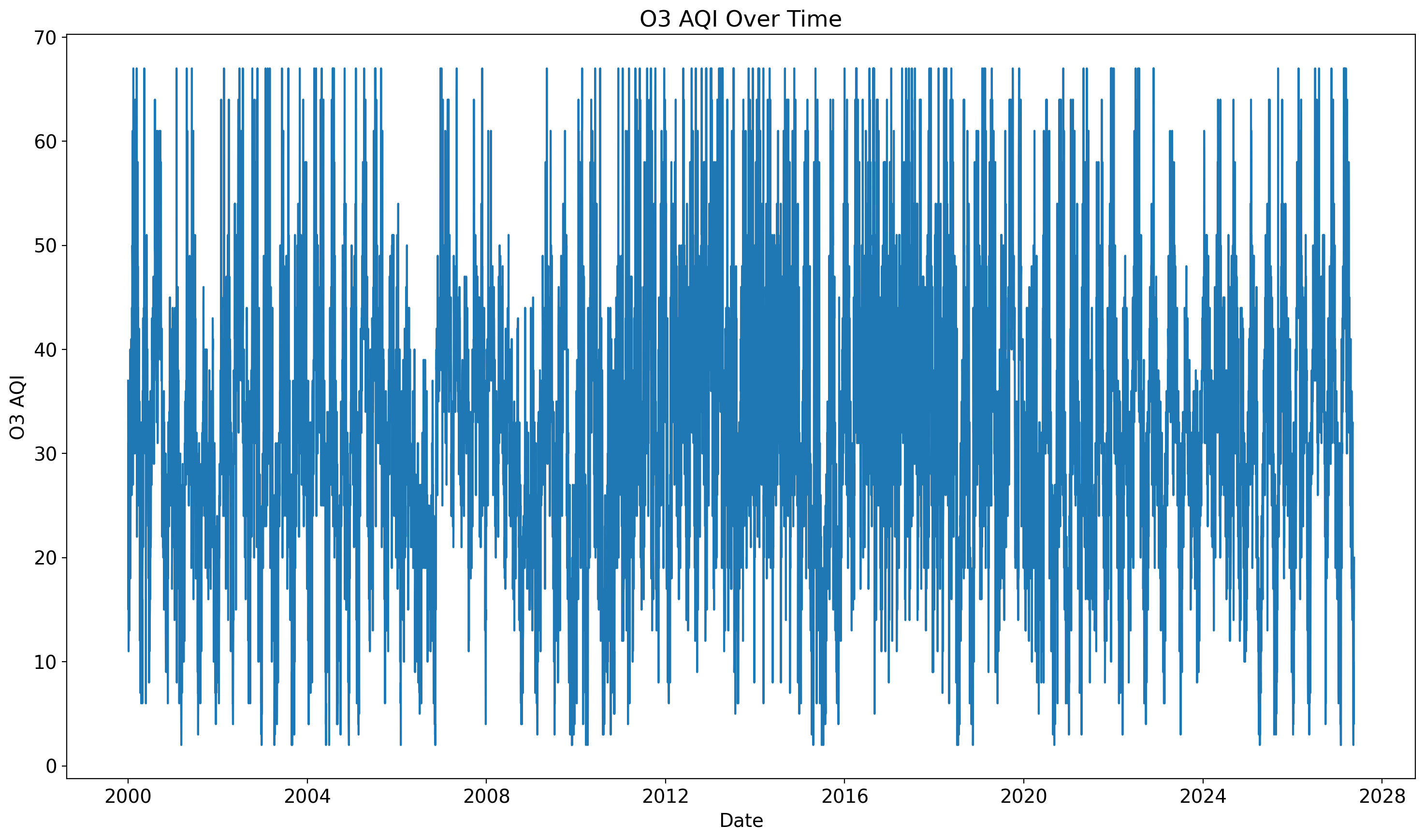
plt.plot(data\_sorted['Date'], data\_sorted['O3 AQI'])

plt.title('O3 AQI Over Time')

plt.xlabel('Date')

plt.ylabel('O3 AQI')

plt.show()



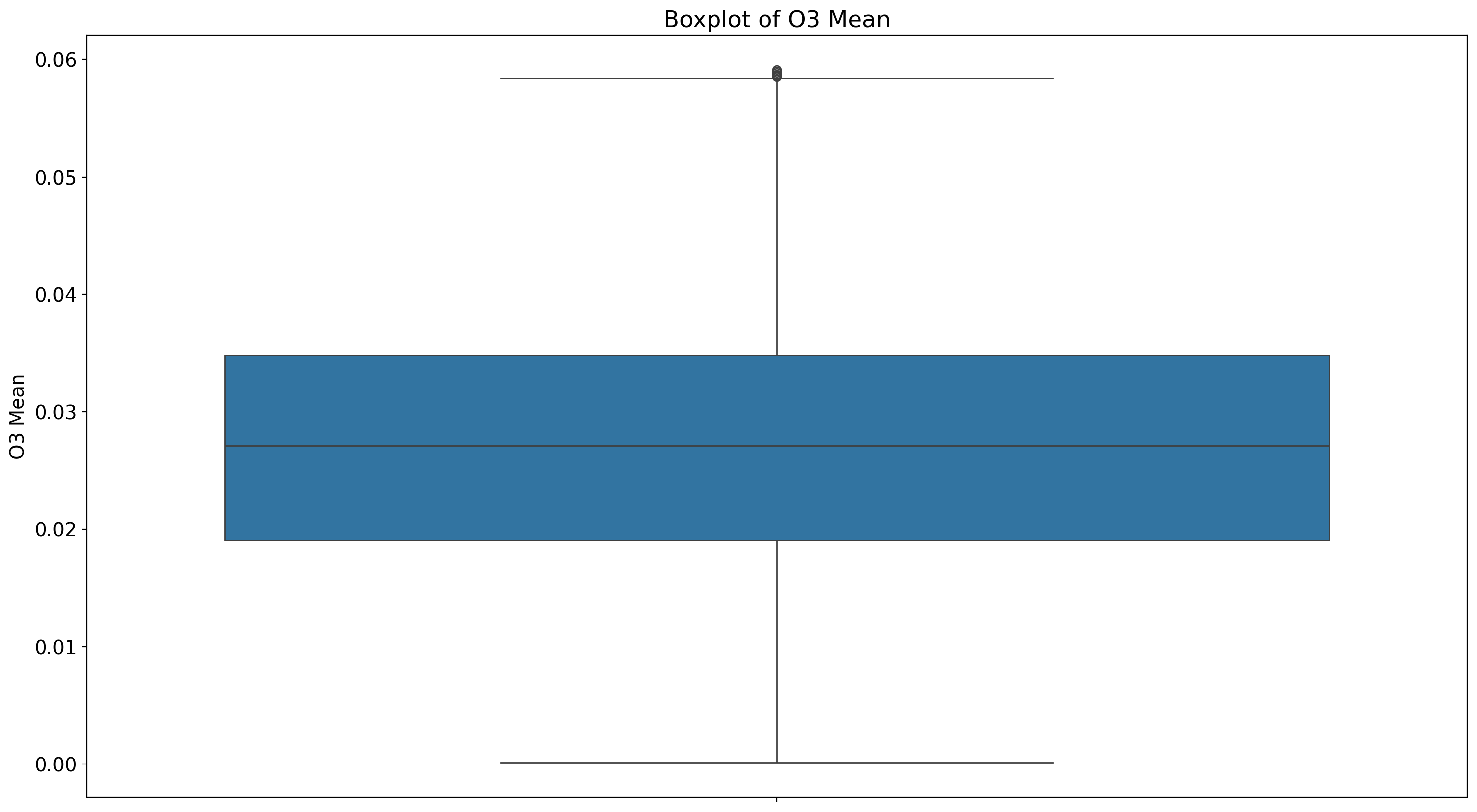
This line chart visualizes the trend of **O3 AQI (Ozone Air Quality Index)** over time. There are frequent and sharp changes in O3 AQI levels, with values oscillating between very low (close to 0) and higher levels (around 70). This indicates substantial temporal variation in ozone concentrations. The chart does not show a consistent increase or decrease in O3 AQI over time. The values remain highly variable throughout the years. There might be some seasonal or periodic patterns, as certain spikes and dips seem to recur over time. This could be due to factors like temperature changes, industrial activity, or traffic patterns.

**# 8) Identify outliers in the O3 Mean column using a boxplot.**

sns.boxplot(data['O3 Mean'])

plt.title('Boxplot of O3 Mean')

plt.show()



This boxplot visualizes the distribution of **O3 Mean** values, providing insight into the data's spread, central tendency, and potential outliers.

**Central Tendency**:

* The **median** of the O3 Mean is approximately **0.03**, indicating that half of the O3 Mean values are below this level.

**Spread**:

* The IQR shows that most O3 Mean values lie between approximately **0.02** and **0.04**.

**Outliers**:

* There is one notable outlier above 0.06, indicating an unusually high O3 Mean value.

**Symmetry**:

* The box is relatively symmetric, suggesting that the data distribution around the median is not highly skewed.

**Range**:

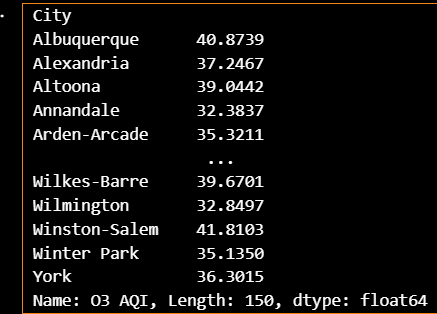
* The whiskers extend from near **0.00** to approximately **0.06**, capturing the majority of the data points.

**# 9) Calculate the average O3 AQI for each city.**

data.columns = data.columns.str.strip()

city\_avg = data.groupby('City')['O3 AQI'].mean()

print(city\_avg)



The above table shows the city having average value og O3 AQI. Here we can conclude that Winston-Salem city has highest O3 AQI value then Albuquerque is at second highest position for O3 AQI values. Annandale has less O3 AQI other than cities.

**# 10) Create new features for the month and year from the Date column.**

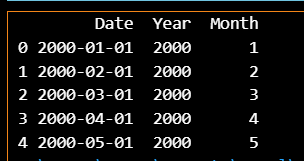
data['Date'] = pd.to\_datetime(data['Date'], errors='coerce')

data = data.dropna(subset=['Date'])

data['Year'] = data['Date'].dt.year

data['Month'] = data['Date'].dt.month

print(data[['Date', 'Year', 'Month']].head())



Here we have created the new column for year and month.

**# 11) Visualize the average CO AQI for different counties.**

county\_avg = data.groupby('County')['CO AQI'].mean()

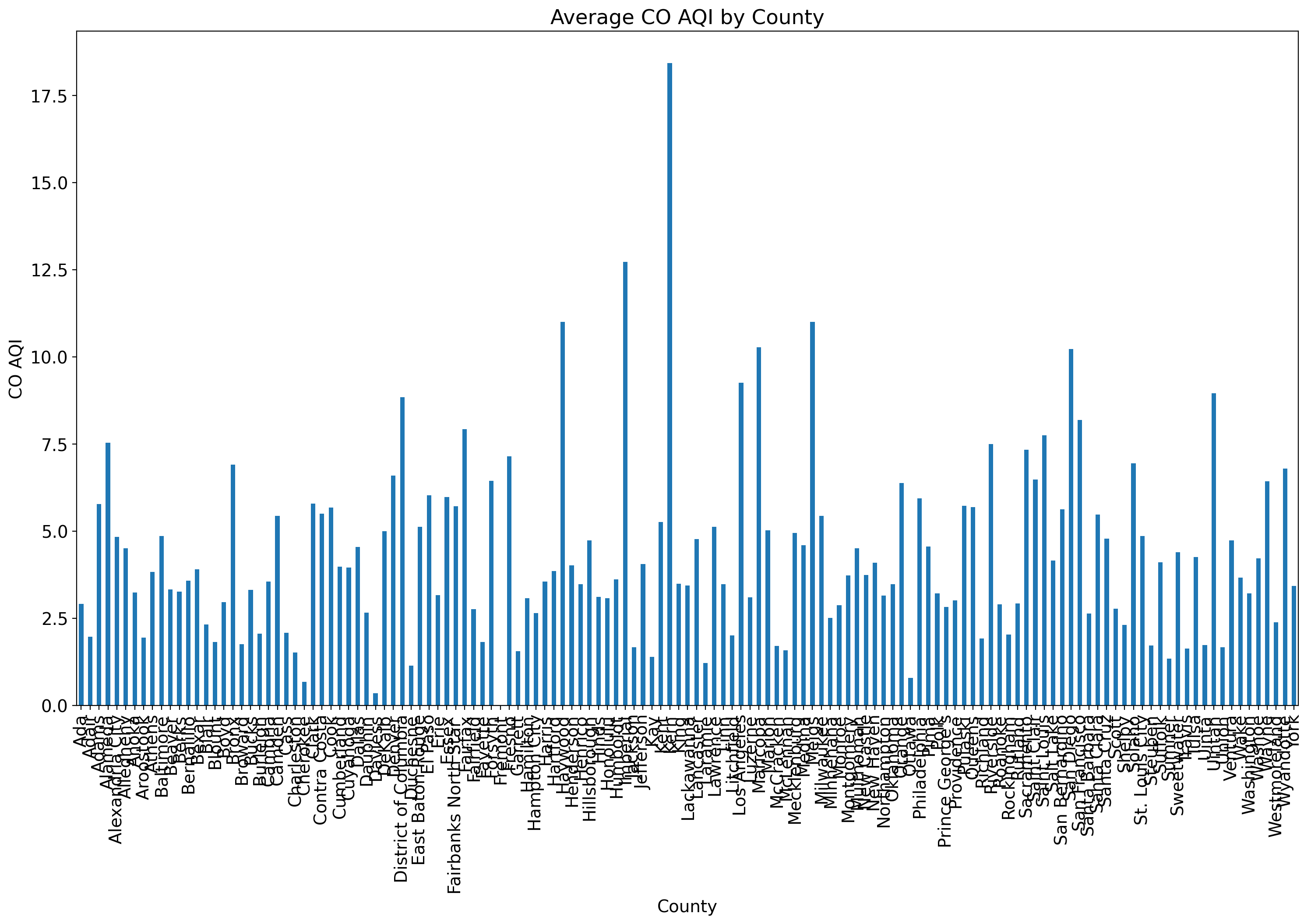
county\_avg.plot(kind='bar')

plt.title('Average CO AQI by County')

plt.xlabel('County')

plt.ylabel('CO AQI')

plt.show()



The above bar chart is showing the co( carbon oxide) AQI level for different countries in US.

The Kerh country.

**# 12) Scale all numeric columns to have a mean of 0 and standard deviation of 1.**

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

data\_scaled = pd.DataFrame(scaler.fit\_transform(data.select\_dtypes(include='number')), columns=data.select\_dtypes(include='number').columns)

print(data\_scaled.head())

Unnamed: 0 O3 Mean O3 1st Max Value O3 1st Max Hour O3 AQI CO Mean \

0 -1.7657 -0.6558 0.3064 -0.2297 0.2535 1.9639

1 -1.7657 -1.0154 -0.3586 -0.2297 -0.3344 2.6363

2 -1.7657 -1.6200 -1.6886 -0.5229 -1.5941 5.1287

3 -1.7657 -1.0154 -0.2755 -0.5229 -0.2504 5.3674

4 -1.7657 -1.8052 -2.0211 -0.5229 -1.9301 8.4866

CO 1st Max Value CO 1st Max Hour CO AQI SO2 Mean SO2 1st Max Value \

0 3.9279 2.1836 3.8830 0.6826 0.7114

1 4.1547 -0.7642 4.0799 0.2477 -0.0976

2 4.6083 0.2611 4.4736 1.6218 0.9810

3 5.7423 2.1836 5.6546 2.3872 1.6551

4 7.3300 -0.5079 7.2294 3.0655 1.5203

SO2 1st Max Hour SO2 AQI NO2 Mean NO2 1st Max Value NO2 1st Max Hour \

0 1.7842 0.8179 0.8409 1.7252 0.9222

1 1.9306 -0.0861 1.2726 0.8678 0.9222

2 1.4915 1.1193 2.9443 1.8571 -0.4748

3 -0.1187 1.8225 3.1797 3.3740 -0.4748

4 -0.2651 1.6216 4.0823 2.5167 1.3032

NO2 AQI Year Month

0 1.7206 -1.8491 -1.5978

1 0.8823 -1.8491 -1.3080

2 1.8604 -1.8491 -1.0183

3 3.5370 -1.8491 -0.7286

4 2.5590 -1.8491 -0.4388

**# 13) Select the top 5 features most correlated with O3 AQI.**

#Identify non-numeric columns and process them appropriately.

print(X.dtypes)

#Drop Non-Numeric Columns: Remove columns that cannot be directly used.

X\_numeric = X.select\_dtypes(include=['float64', 'int64'])

#Convert categorical data into numeric format using techniques like one-hot encoding or label encoding.

X\_encoded = pd.get\_dummies(X, drop\_first=True)

#If the dataset includes date columns, extract numeric features like year, month, or day.

X['Year'] = pd.to\_datetime(X['Date'], errors='coerce').dt.year

X['Month'] = pd.to\_datetime(X['Date'], errors='coerce').dt.month

X = X.drop(columns=['Date']) # Drop the original Date column

#After cleaning the dataset, apply SelectKBest:

from sklearn.feature\_selection import SelectKBest, f\_regression

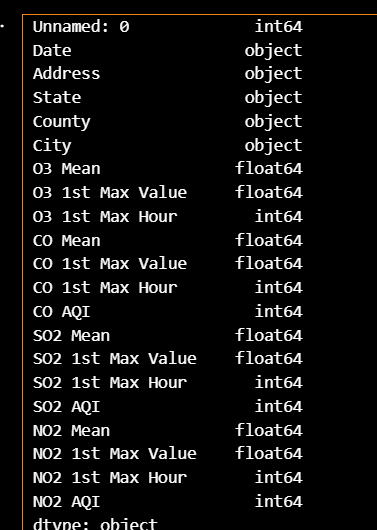
# Ensure only numeric columns are used

X\_numeric = X.select\_dtypes(include=['float64', 'int64'])

selector = SelectKBest(score\_func=f\_regression, k=5)

X\_new = selector.fit\_transform(X\_numeric, y)

print("Selected features:", X\_numeric.columns[selector.get\_support()])

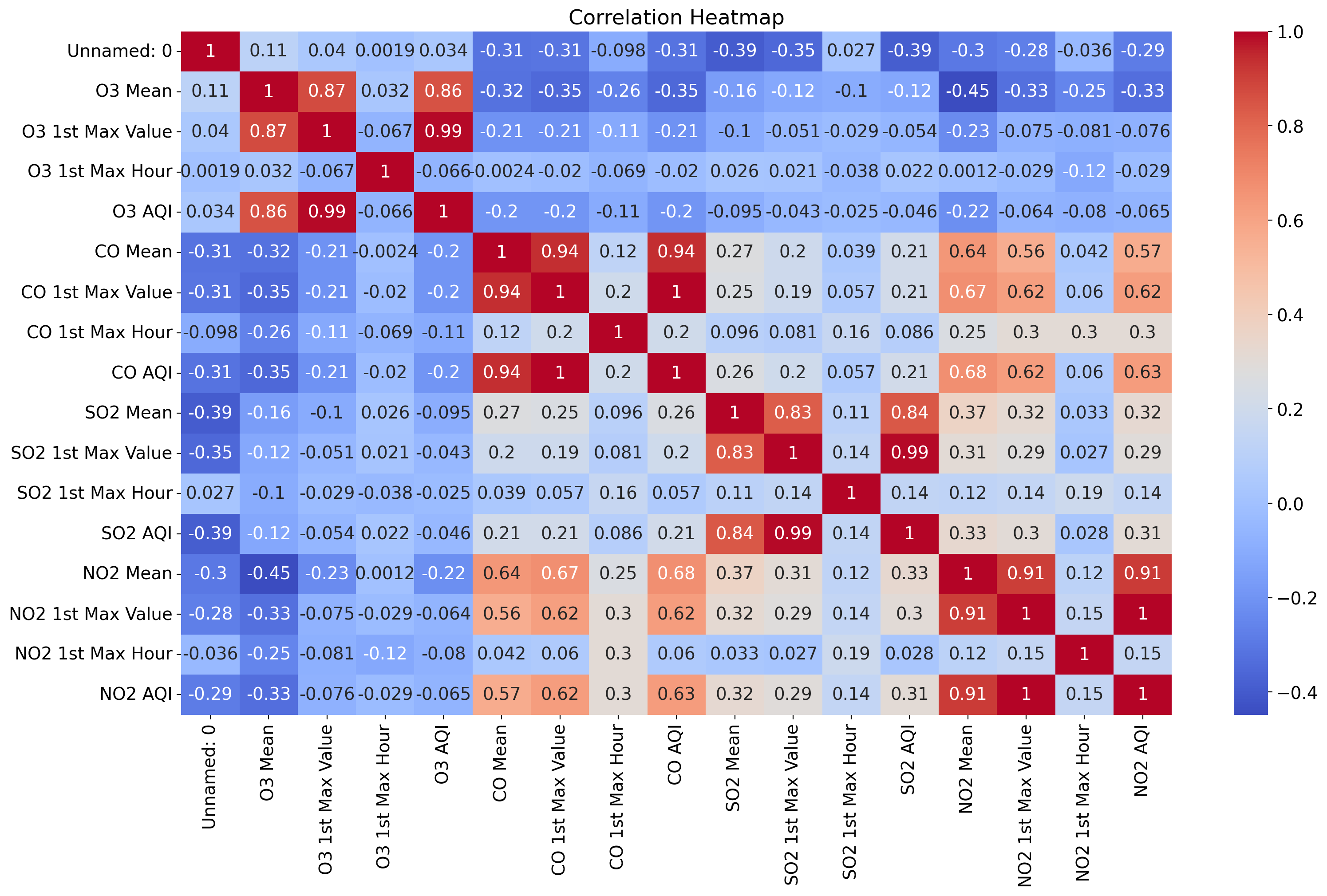


**# 14) Visualize the correlation matrix with a heatmap.**

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm')

plt.title('Correlation Heatmap')

plt.show()



This heatmap visualizes the **correlation matrix** between various air quality metrics and their associated variables (e.g., O3, CO, SO2, NO2) in the dataset. The colours represent the **correlation coefficients**:

* Red (close to 1): Strong positive correlation.
* Blue (close to -1): Strong negative correlation.
* White or light blue (close to 0): Weak or no correlation.

**1**: Perfect positive correlation (as one variable increases, the other also increases).

**-1**: Perfect negative correlation (as one variable increases, the other decreases).

**0**: No correlation.

**CO Mean** and **CO 1st Max Value** (0.94): Indicates a strong relationship between the CO mean concentration and its maximum value.

**SO2 Mean** and **SO2 AQI** (0.99): Strong correlation between the mean SO2 levels and its Air Quality Index.

**NO2 Mean** and **NO2 AQI** (0.91): Indicates that NO2 levels are closely tied to the AQI derived from it.

**O3 1st Max Value** and **O3 AQI** (0.99): Suggests that ozone's maximum value directly influences its AQI.

**# 15) Use cross-validation to evaluate the linear regression model.**

# Target variable

y = data['O3 AQI']

# Features

X = data.drop(columns=['O3 AQI'])

# Drop irrelevant columns

X = X.drop(columns=['Address', 'State', 'County', 'City'], errors='ignore')

# Process date columns

X['Year'] = pd.to\_datetime(data['Date'], errors='coerce').dt.year

X['Month'] = pd.to\_datetime(data['Date'], errors='coerce').dt.month

X = X.drop(columns=['Date'], errors='ignore')

# Ensure numeric features

X = pd.get\_dummies(X, drop\_first=True)

# Handle missing values

X = X.fillna(0)

# Perform cross-validation

model = LinearRegression()

scores = cross\_val\_score(model, X, y, cv=5, scoring='neg\_mean\_squared\_error')

print("Cross-validated MSE:", -scores.mean())

Cross-validated MSE: 2.4918311466600276

**Significance of Monitoring Air Quality:**

Air quality pollution involves the presence of harmful substances in the atmosphere, which adversely affect human health and the environment. This dataset focuses on four primary pollutants: Nitrogen Dioxide (NO2), Sulphur Dioxide (SO2), Carbon Monoxide (CO), and Ozone (O3). Effective monitoring of air quality is paramount for both public health and environmental conservation. Accurate data empowers policy formulation, facilitates the understanding of temporal and spatial trends, and raises public awareness.

**Health and Environmental Impact of Pollutants**

**Nitrogen Dioxide (NO2):** Elevated concentrations of NO2 can irritate the human respiratory system and are especially risky for individuals with asthma or other respiratory ailments. Chronic exposure may result in long-term respiratory diseases.

**Sulphur Dioxide (SO2):** SO2 can chemically react in the atmosphere to produce fine particulate matter, which, when inhaled, can pose significant health risks. It disproportionately affects vulnerable populations such as young children, the elderly, and individuals with pre-existing heart or lung conditions.

**Carbon Monoxide (CO):** High levels of CO can become life-threatening as they disrupt the blood's ability to transport oxygen, thus worsening pre-existing cardiovascular conditions.

**Ozone (O3):** Although Ozone in the upper atmosphere shields us from harmful ultraviolet rays, its presence at ground level can lead to respiratory issues and compound existing chronic respiratory diseases like asthma.

**References:**

[1]: https://www.kaggle.com/datasetssirajahmad/acciojob-ml-capstone-social-media