Project Report

Course: *CMSC 6950 -- Fall 2023*

Name: *Mohammadmahdi Mirmojarabian* (Student #: 202292549)

1 Description of Dataset

In this project, I used an air quality dataset from https://archive.ics.uci.edu/dataset/360/air+quality . These data are the output responses of a gas multisensor device deployed on a field in an Italian city to measure the air quality over time. The dataset contains 9357 rows of hourly averaged responses from an array of 5 metal oxide chemical sensors embedded in an air quality chemical multisensor device. Data was recorded for a period of one year (Early March 2004 – Early April 2005). Also, a co-located reference certified analyzer provided Ground Truth hourly averaged concentrations for CO, Non-Methanic Hydrocarbons, Benzene, Total Nitrogen Oxides (NOx), and Nitrogen Dioxide (NO2). Missing values are labeled with "-200".

1.1 Attributes of Dataset

All columns of the data are shown in Table 1:

Table 1: Dataset columns

Date	Date (DD/MM/YYYY)
Time	Time (HH.MM.SS)
CO(GT)	CO(GT): True hourly averaged CO concentration [**mg/m^3**] (reference analyzer)
PT08.S1(CO)	PT08.S1 (tin oxide) hourly averaged sensor response (nominally CO targeted)
NMHC(GT)	Non Metanic HydroCarbons concentration [**µg/ m^3**] (reference analyzer)
C6H6(GT)	True hourly averaged Benzene concentration [**µg/ m^3**] (reference analyzer)
PT08.S2(NMHC)	PT08.S2 (titania) hourly averaged sensor response (nominally NMHC targeted)
NOx (GT)	True hourly averaged NOx concentration [**ppb**] (reference analyzer)
PT08.S3(NOx)	PT08.S3 (tungsten oxide) hourly averaged sensor response
NO2(GT)	True hourly averaged NO2 concentration [**µg/ m^3**] (reference analyzer)
PT08.S4(NO2)	PT08.S4 (tungsten oxide) hourly averaged sensor response
PT08.S5(O3)	PT08.S5 (indium oxide) hourly averaged sensor response (nominally O3 targeted)
T	Temperature [**°C**]
RH	Relative Humidity (**%**)
AH	AH Absolute Humidity

2 Methodology

This project's main code is located in a Jupyter Notebook file named "code_project.ipynb". I also put two of my functions that need unit tests in a .py file called "Functions_For_Test.py" and wrote their test contents in the "Test_Functions_For_Test.py" file. For my project, I have used (pandas, numpy, matplotlib, seaborn, and pytest) libraries in Python.

In the first stage, I constructed a Python panda dataframe from my "AirQuality.csv" file located in the "dataset" folder. I did some initial data cleaning. I dropped extra and all-NaN data (the last 2 columns). I also dropped extra fully empty rows (from index 9357 to 9471). In the dataframe, missing data was tagged with "-200". These values were replaced with NaN for easier handling. A function "NaN_Percentages()" was defined to calculate the percentage of NaN values in each column. This was used to decide which columns should be removed. The column "NMHC(GT)" had %90.23 missing values; so, I dropped this attribute because keeping that will not be that helpful for our analysis.

2.1 Data Visualization

Time Series Illustration: The data was plotted in a series of clearly labeled plots using the matplotlib and pandas libraries in Python. The function "my_plot()" was defined to automatically plot each attribute time series. The 'Date' and 'Time' columns were extracted and converted to datetime objects, and then each attribute was plotted against the date. The x-axis was set to show only monthly ticks and a grid was added for better visualization.

Filling in Missed Values Using Interpolation Techniques: The dataset had many missing values. To handle these, an interpolation technique was used. A copy of the original dataframe was made, and the “interpolate()” function from pandas was used to fill in the missing values. The ‘nearest’ method was used for interpolation, which uses the value of the nearest point to fill the missing values. After filling in the missing values, the function “plot_orig_modif_series()” was used to plot both the original and modified dataframes. This function plotted the original data in red and the modified data in blue, allowing for a clear comparison between the two. The plots were set to show only monthly ticks on the x-axis, and a grid was added for better visualization.

Data Distribution Visualization: To assess the distributions of the data, histograms and probability density plots were created for each attribute in the dataset. A function “plot_histograms_density()” was defined to automatically plot both the histogram and probability density function for each attribute on the same plot. The function takes as input the dataframe and the columns to plot. It creates a figure with subplots, one for each attribute. For each attribute, it creates a normalized histogram and a probability density plot on the same subplot. The plots are labeled, and a legend is added for clarity.

2.2 Extreme Value Analysis

A function “my_boxplot()” was defined to create boxplots for each attribute in the dataset. The boxplots were used to visualize the distribution of the data and identify potential outliers. The whiskers of the boxplot were set to 1.5 times the Interquartile Range (IQR) by default.

A function “show_stats()” was defined to compute various descriptive statistics for each attribute, including count, mean, median, minimum, maximum, standard deviation, variance, and skewness.

The Interquartile Range (IQR) method was used to define what constitutes an “extreme value”. This method was preferred over the Standard Deviation method because it provides specific parameters (IQR, Q1, Q3) for each distribution that can show the behavior of each attribute value uniquely.

A function “remove_outliers_IQR()” was defined to automatically calculate the IQR parameters and the lower and higher limits for each attribute. This function was used to identify and handle outliers in the data. Outliers were defined as values that fall outside of the whiskers in the boxplot. The function also calculated the outlier ratio for each attribute, plotted the outlier ratios in a bar chart, and provided information about the edited dataframe. The outliers were either replaced with the lower or higher limit values or removed from the dataset, depending on the specified mode.

Extreme Value Analysis (scale=1.5): Boxplots were created for each attribute in the dataset using a whisker scale of 1.5. The boxplots were used to visualize the distribution of the data and identify potential outliers. A function “remove_outliers_IQR()” was defined to automatically calculate the IQR parameters and the lower and higher limits for each attribute. This function was used to identify and handle outliers in the data. Outliers were defined as values that fall outside of the whiskers in the boxplot. The function also calculated the outlier ratio for each attribute, plotted the outlier ratios, and provided information about the edited dataframe. The outliers were replaced with lower or higher limit values.

Statistical Analysis of Extreme Values (scale=1.5): Various descriptive statistics were computed for the outliers, including count, mean, median, minimum, maximum, standard deviation, variance, and skewness. The frequency, range, associated times or conditions, variability, and outlier ratios of the extreme values were analyzed. Also, a bar chart was plotted to compare the outlier ratios for all attributes.

Comparison of Original and Modified Data (scale=1.5): The original and modified dataframes were compared to observe the outliers. The original data was plotted in red, and the modified data in blue, allowing for a clear comparison between the two.

Extreme Value Analysis (scale=2): The same analysis was repeated with a whisker scale of 2. This allowed for the exploration of the sensitivity of the results to the definition of “extreme values.”

Statistical Analysis of Extreme Values (scale=2): The same statistical analysis was conducted for the outliers identified with a whisker scale of 2. The results were compared with those obtained with a whisker scale of 1.5.

Comparison of Original and Modified Data (scale=2): The original and modified dataframes were compared again to observe the outliers identified with a whisker scale of 2. The results were compared with those obtained with a whisker scale of 1.5.

2.3 Trend Analysis

Data Cleaning: The dataframe was cleaned by dropping rows with NaN values. The 'Date' and 'Time' columns were formatted to datetime type, and a new column, 'Week Day' was added to the dataframe.

Trend Analysis in CO Values: To identify a trend in CO values, the mean hourly values of CO were calculated for each day of the week. The data was grouped by 'Week Day', and the mean of 'PT08.S1(CO)' was calculated for each group. Bar plots were created to visualize the mean hourly values for each day and for each hour of the day.

Correlation Analysis: Pearson's correlation was used to find the correlation between all the features. A heatmap was created to visualize the correlations.

Pairplot Visualization: Pairplots were created for each pair of features to visualize the relationships between features. The histograms on the diagonal allowed us to see the distribution of each feature.

Seasonal Analysis: A function was defined to assign the seasons based on the astronomical/meteorological definitions. The 'Date' column was used to extract the season information, and a new column 'Season' was added to the dataframe. Pairplots were created for each pair of features, colored by the season, to visualize the relationships between features in different seasons.

Trend Analysis of Each Attribute: To analyze the trend of each attribute, the monthly average points for each attribute were calculated. The data was grouped by 'Month' and 'Month_Name,' and the mean of each attribute was calculated for each group. The grouped data was sorted by 'Month.'

A figure with subplots was created, one for each attribute. For each attribute, a bar plot was created to visualize the monthly average values. A rolling average line was also added to the plot to show the trend over time. The x-axis labels were rotated for better readability. The rolling average of each attribute was calculated with a window size of 1. This provided a smoothed line that represents the trend of the attribute over time.

3 Results

3.1 Data Visualization

I am showing just one of the 12 available time series plots for all 12 attributes. Other plots are shown in the code.

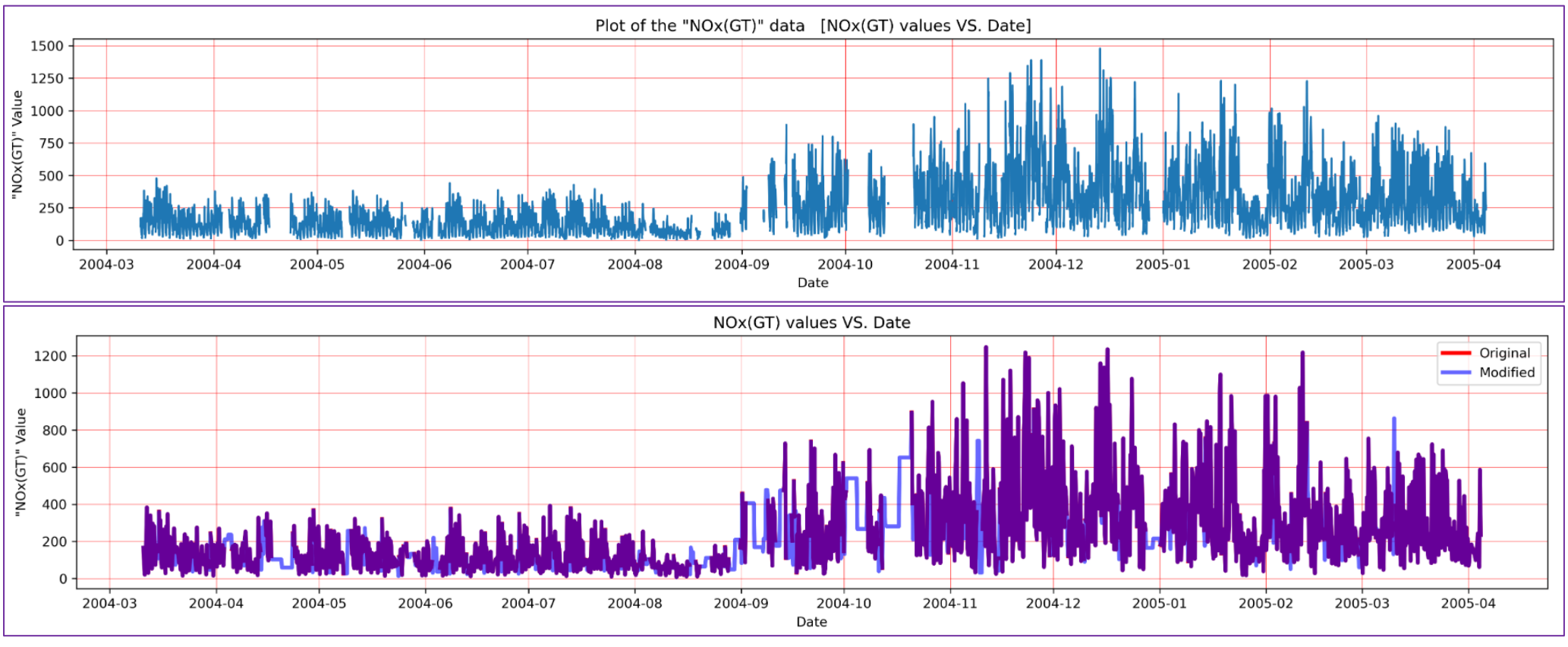


Figure 1: This figure is for task 1. Both plots show the concentration of NOX(GT) [**ppb**] for one year (Early March 2004 – Early April 2005). In the upper plot, we can see the gaps related to missing values. So, we filled in the missing values using the

‘nearest’ method of interpolation and created the bottom plot. The modified part is in blue color. We can compare NOX(GT) values and see the predicted values between these two plots.

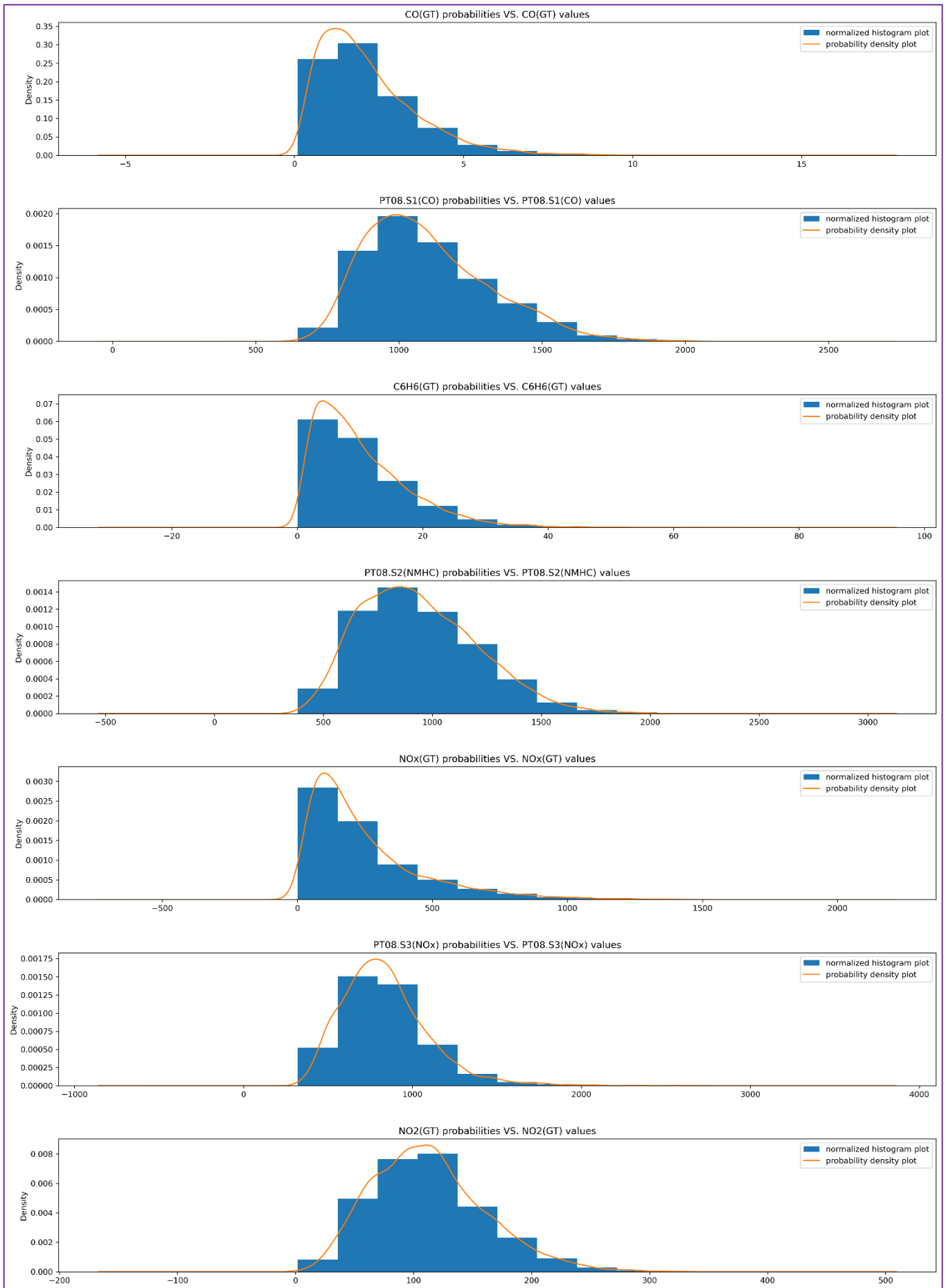


Figure 2: This figure is histograms and probability density plots were created for each attribute in the dataset. We can see and detect the distribution of our attributes. Histograms are a good choice for almost all data types to assess the distributions. When combined with probability density plots, histograms can help us recognize the type of our distribution. Density plot is a smoothed version of the histogram and is used in the same concept. In Table 2, we have recognized the type of distribution for each attribute, and we can see that many of our columns have skewness. So, it would be better to use a median whenever we want to replace some values with a central value.

Table 2: Distribution Results of Our Attributes

Column	Skewness	Outliers
CO(GT)	right skewed	some outliers
PT08.S1(CO)	right skewed	many outliers
C6H6(GT)	right skewed	many outliers
PT08.S2(NMHC)		almost many outliers
NOx(GT)	right skewed	so many outliers
PT08.S3(NOx)		many outliers
NO2(GT)		some outliers
PT08.S4(NO2)		some many outliers
PT08.S5(O3)	right skewed	some many outliers
T	multimodal distribution	no/very few outliers
RH		no outliers
AH	multimodal distribution	no/very few outliers

3.2 Extreme Value Analysis

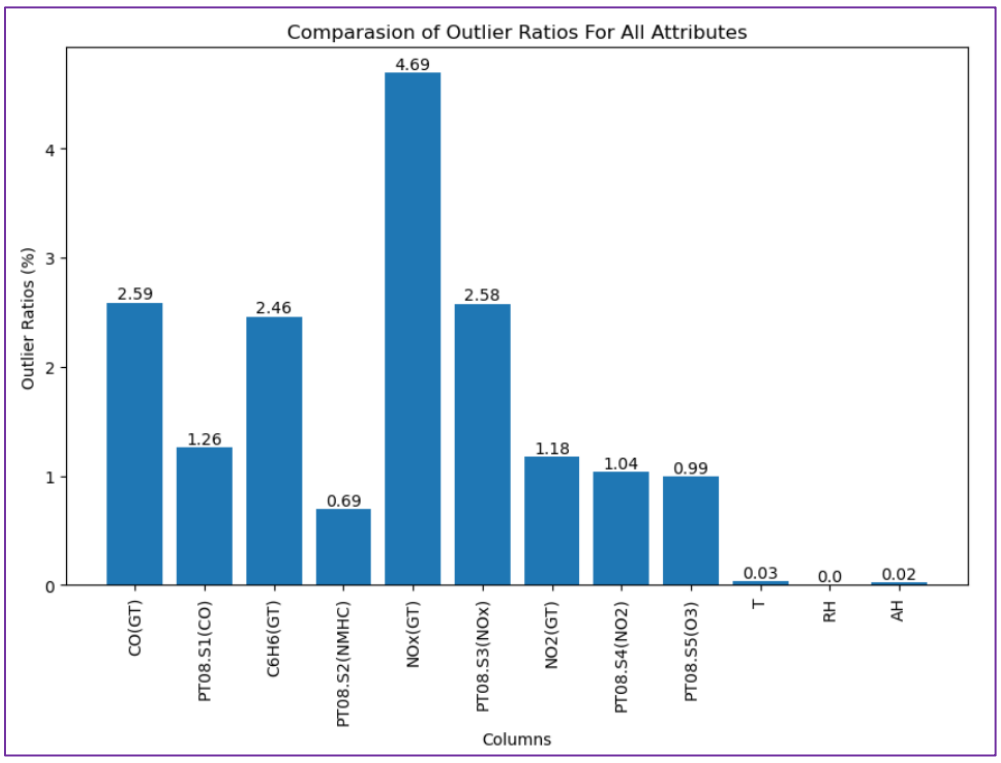
I have computed various descriptive statistics for the outliers in all columns of data, including count, mean, median, minimum, maximum, standard deviation, variance, and skewness that have shown in Table 3 and Table 4. The frequency of extreme values (count values) on scale = 1.5 is more frequency of extreme values on scale = 2. If scale = 2, the extreme values have more ranges (min value, max value, and mean value). The variability of the extreme values (The standard deviation of the extreme values for each feature) in scale = 1.5 is more than scale = 2.

Table 3: Descriptive Statistics for Outliers (whisker scale of 1.5)

Statistic	CO(GT)	PT08.S1(CO)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O3)	T	RH	AH
count	242.0	118.0	230.0	65.0	439.0	241.0	110.0	97.0	93.0	3.0	0.0	2.0
mean	6.7	1774.9	34.4	1799.2	852.3	1651.8	262.2	2467.5	2232.4	44.1		2.2
median	6.4	1754.5	32.6	1770.0	807.0	1581.0	252.5	2464.0	2197.0	44.3		2.2
min	5.6	1673.0	28.4	1689.0	668.0	1437.0	238.0	551.0	2087.0	43.4		2.2
max	11.9	2040.0	63.7	2214.0	1479.0	2683.0	340.0	2775.0	2523.0	44.6		2.2
std	1.1	84.7	5.6	105.3	163.1	216.0	23.1	222.2	124.3	0.6		0.0
var	1.3	7175.9	31.6	11082.1	26604.6	46658.1	535.5	49393.7	15441.7	0.4		0.0
skew	1.6	1.0	1.7	1.3	1.2	1.8	1.3	-6.7	0.9	-1.3		

Table 4: Descriptive Statistics for Outliers (whisker scale of 2)

Statistic	CO(GT)	PT08.S1(CO)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O3)	T	RH	AH
count	113.0	31.0	106.0	13.0	246.0	114.0	34.0	21.0	17.0	0	0	0
mean	7.6	1892.4	38.9	1964.4	957.1	1817.9	291.1	2644.8	2448.4			
median	7.3	1882.0	37.0	1958.0	910.5	1756.5	284.5	2641.0	2452.0			
min	6.5	1819.0	33.2	1889.0	782.0	1593.0	270.0	2568.0	2359.0			
max	11.9	2040.0	63.7	2214.0	1479.0	2683.0	340.0	2775.0	2523.0			
std	1.1	61.7	5.3	84.3	147.4	210.3	18.6	51.9	56.4			
var	1.2	3806.4	28.5	7106.8	21739.6	44232.1	346.4	2696.3	3175.6			
skew	1.5	0.7	1.7	2.4	1.1	1.8	1.1	0.9	-0.1			



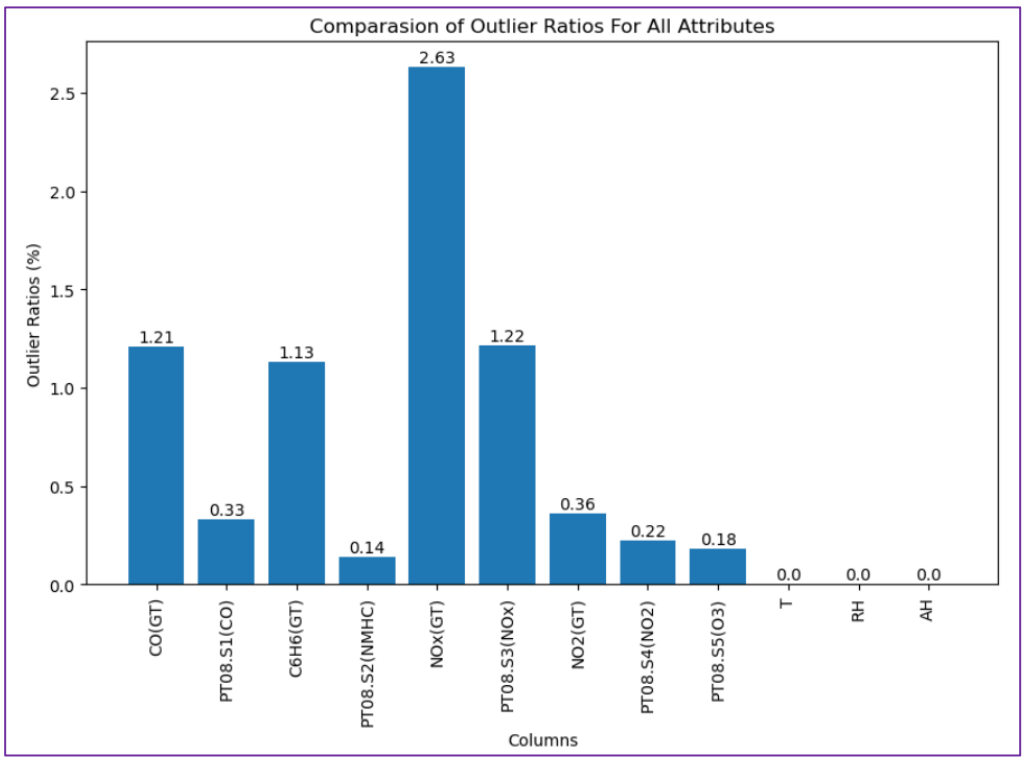


Figure 3: This figure is for task 2 of the project. To define extreme values in our data, we used the IQR method. The left plot has a whisker scale of 1.5, but the right plot uses a whisker scale of 2. We can compare the outlier ratios for all attributes of our data between two different whisker scales. In the left plot, we have more extreme values in all attributes compared to the right plot. The NOX(GT) variable has the most extreme values in both whisker scales. We almost don't have extreme values in the T, RH, and AH variables in both whisker scales.

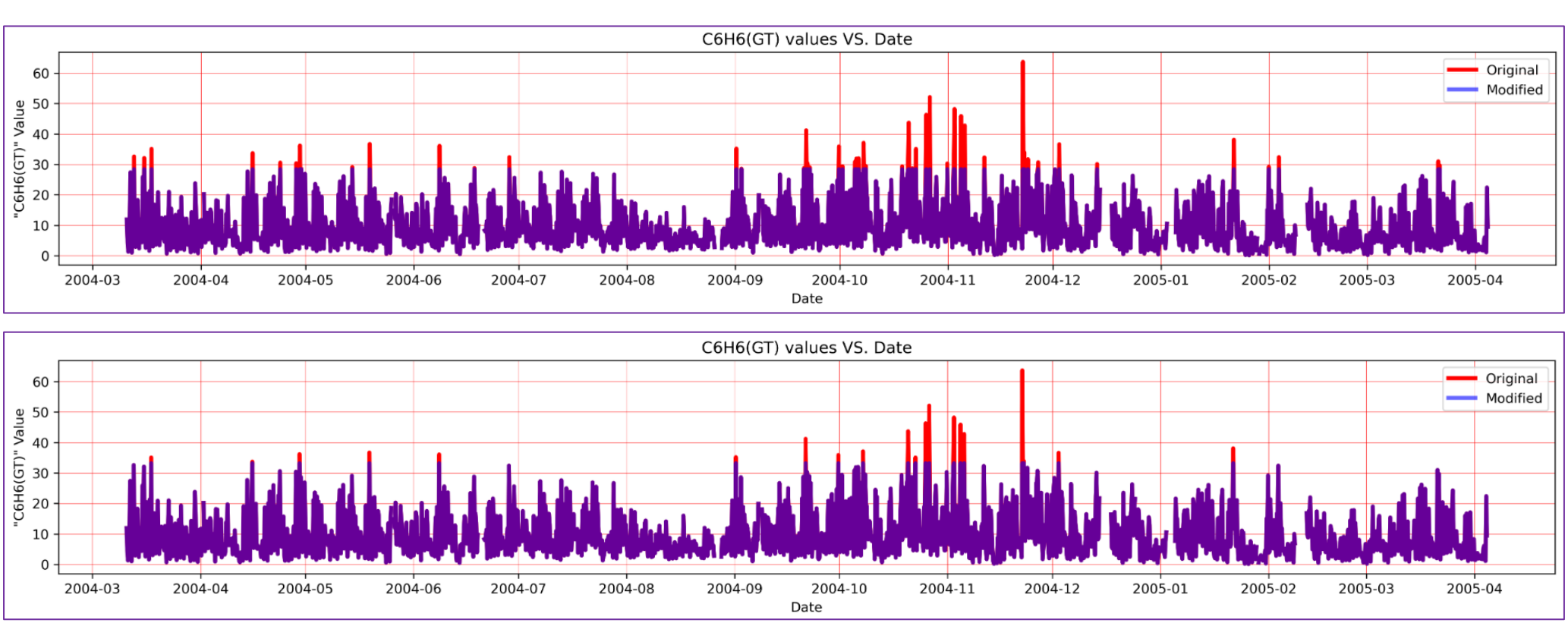


Figure 4: This figure is also for task 2 of the project. Both plots are time series plots that can display outliers clearly. The upper plot has a whisker scale of 1.5, but the bottom plot uses a whisker scale of 2. We can see the extreme values in red color. The upper plot has defined more extreme values compared to the bottom plot. In most variables, when we look at their time series plot, we can find that usually, in Nov and Dec, we have extreme values. It could be related to temperature inversions,

where a layer of warm air traps cooler air near the ground, can prevent pollutants from dispersing, and lead to the buildup of pollutants in the lower atmosphere.

3.3 Trend Analysis

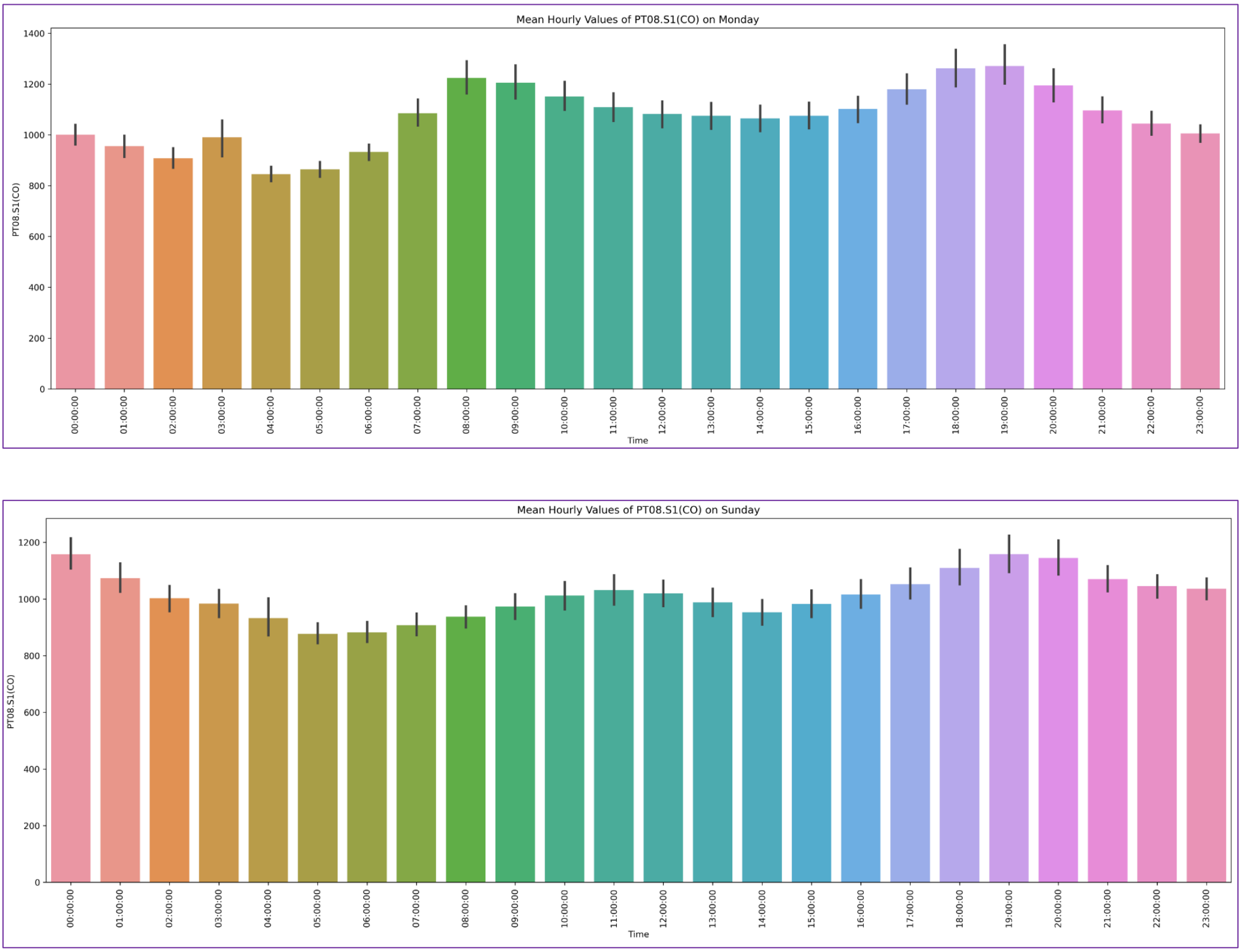


Figure 5: To identify a trend in CO values, I calculated the mean hourly values of PT08.S1(CO) on each day of the week. This plot shows the CO concentration trend over 24 hours on Monday and Sunday (This plot is available for other days of the week in the Jupyter Notebook file). In the upper plot, we can see that the two peaks of CO concentration in the city are 8 AM and 7 PM, the beginning and end of office hours, respectively. But in the bottom plot, we can observe that the peak hours shift to the later hours (11 AM and 7 PM) on weekends, which makes sense.

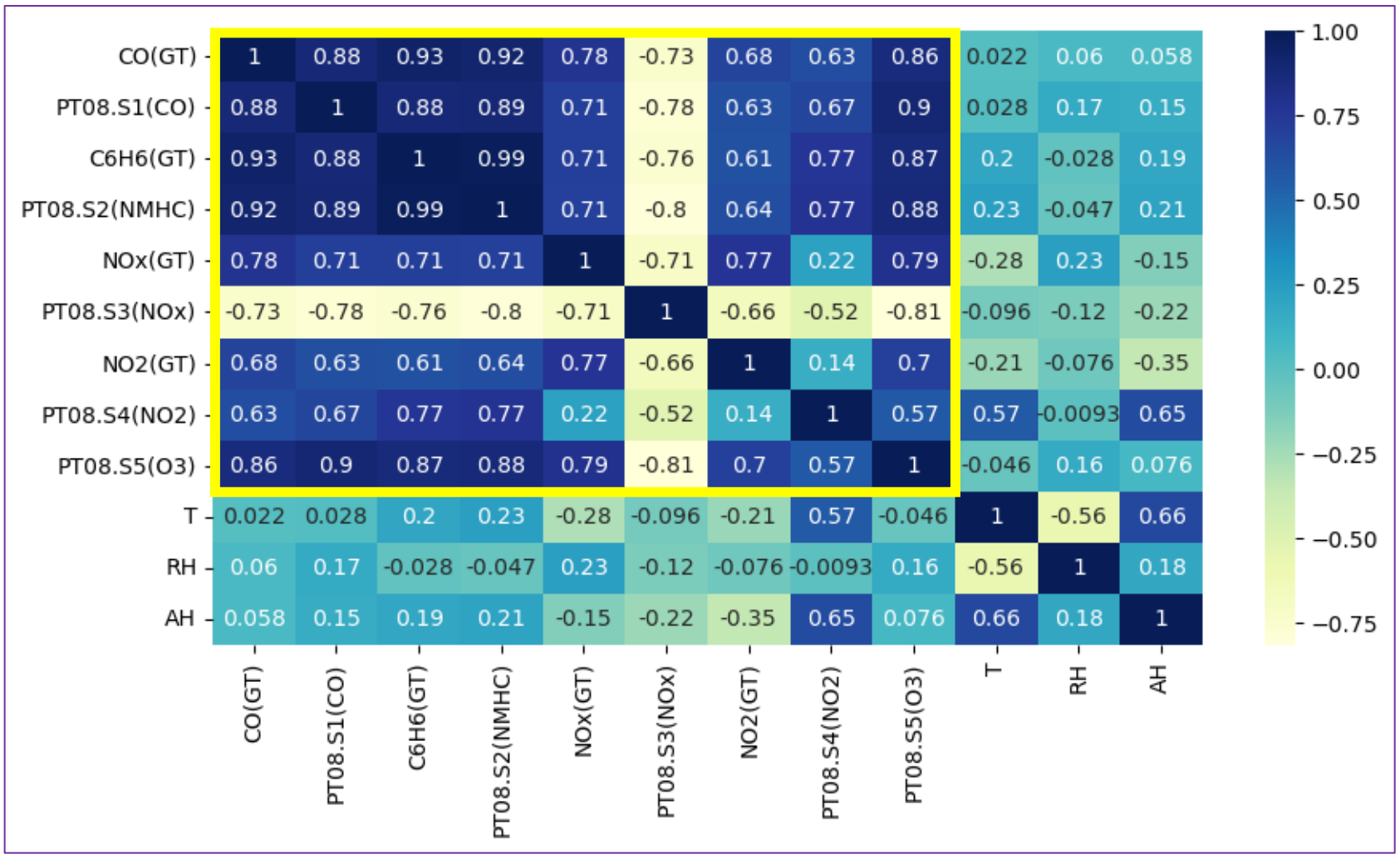


Figure 6: This plot is a heatmap plot that uses a correlation matrix (Pearson's correlation) to visualize the correlations between the features. We can see a correlation between all the pollutants. But we can observe that the columns 'T,' 'RH,' and 'AH' don't have a strong correlation with other features (pollutants). NO2(GT) and NOx(GT) have correlations with other features but are not that strong as compared to CO(GT), C6H6(GT), and Columns with PTs (PT08). CO(GT) and C6H6(GT) can be the columns that are correlated with all other features and can be the target.

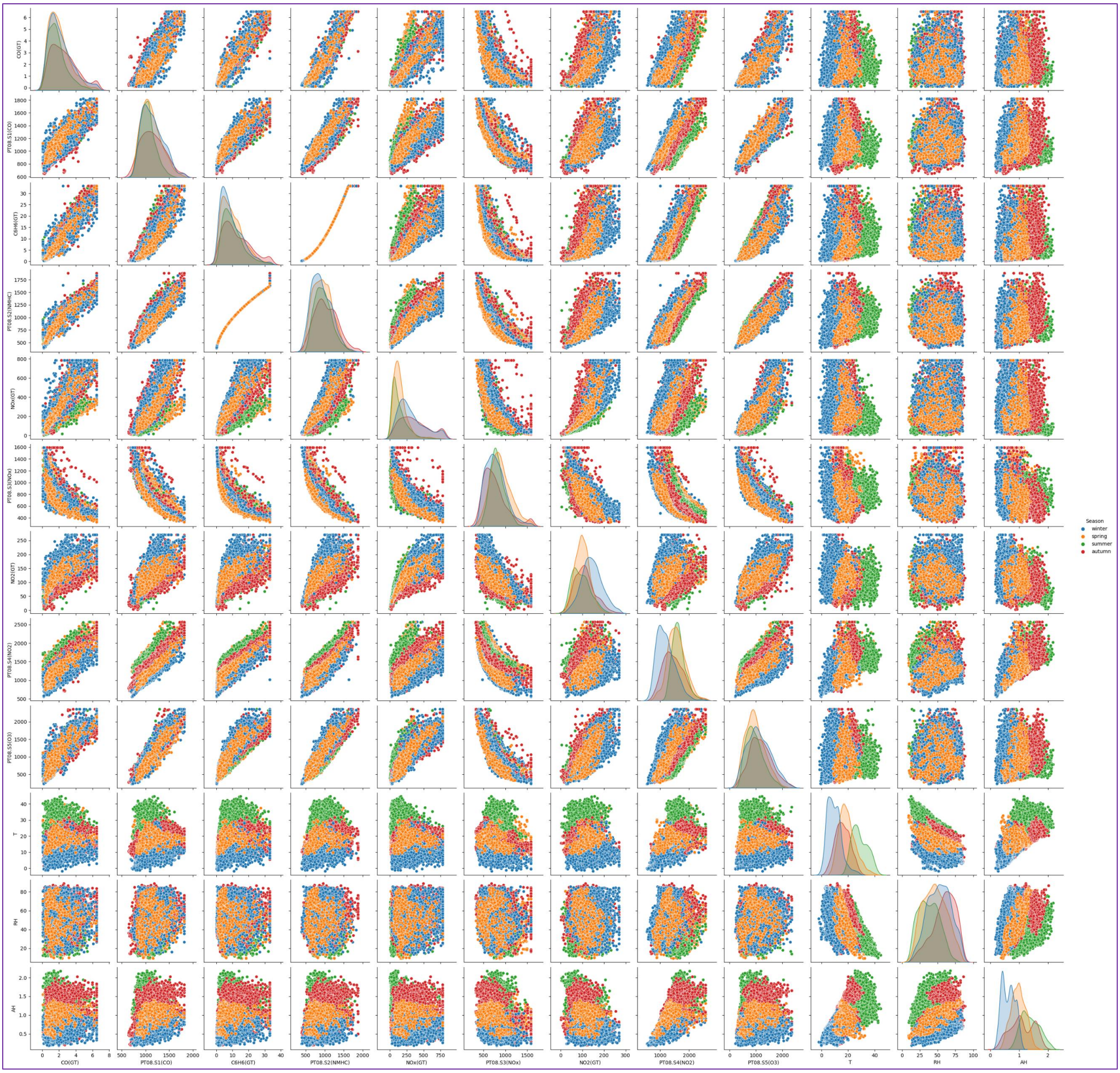
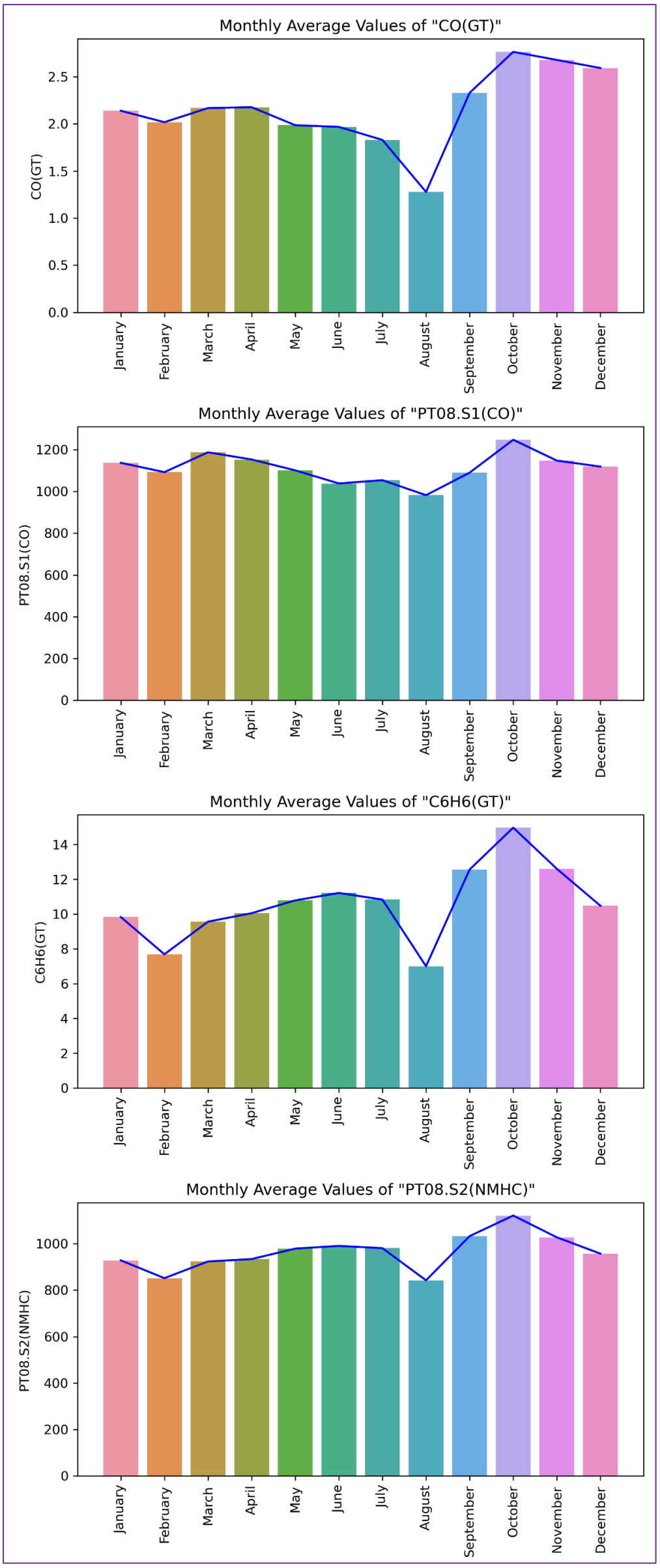
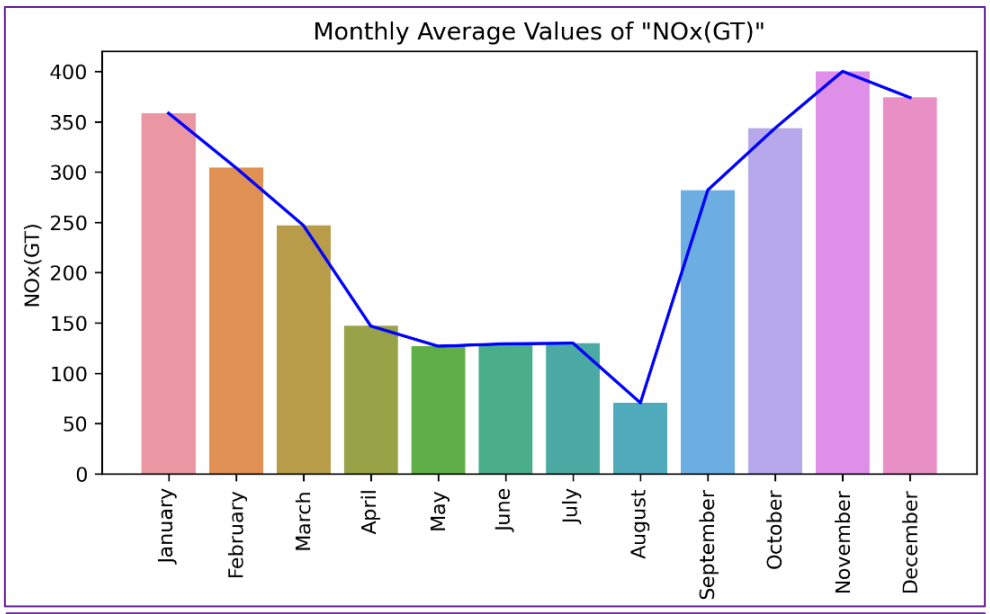
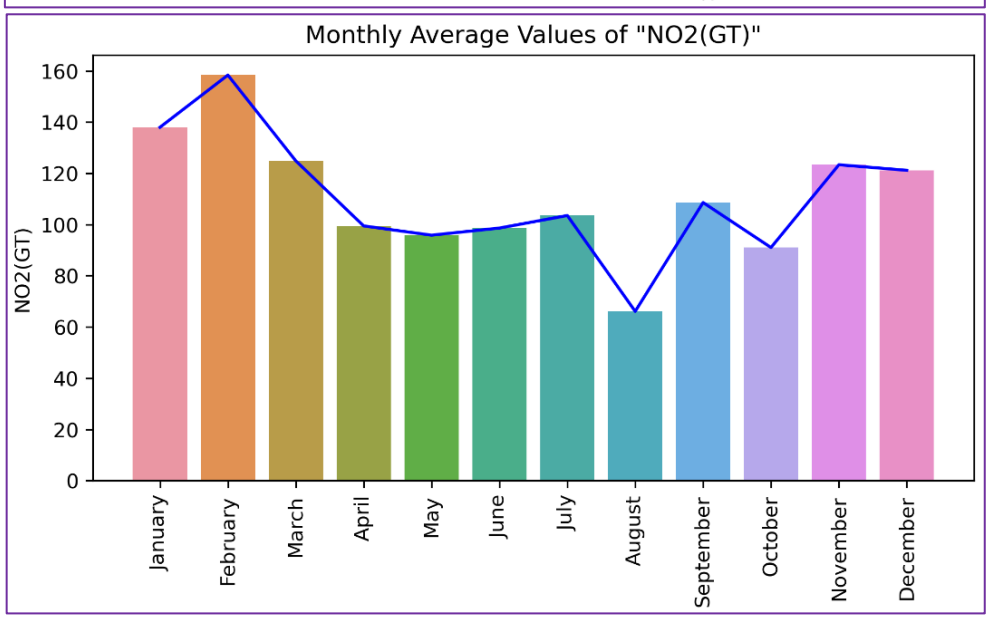
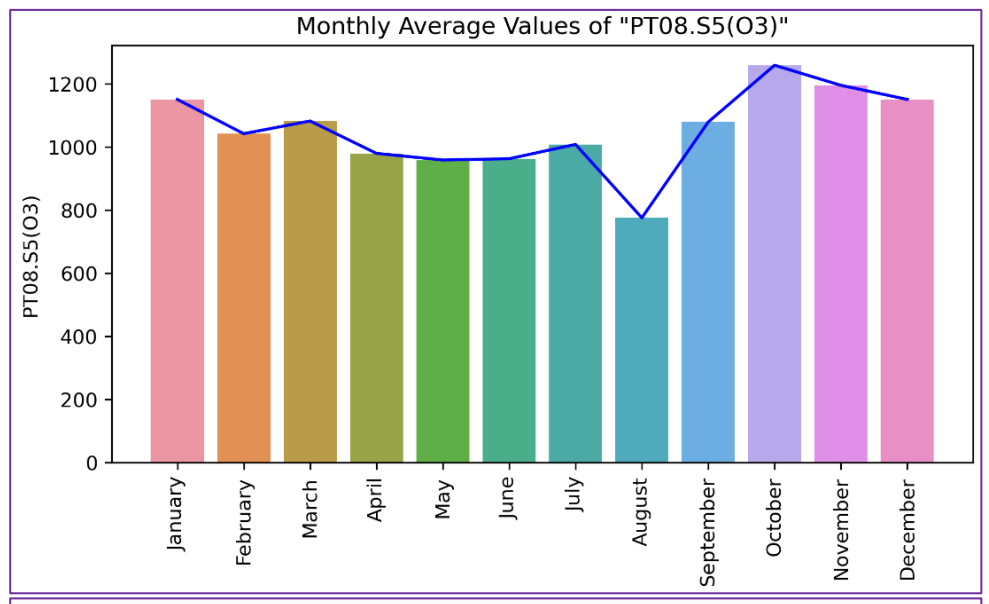


Figure 7: This figure is a pair plot of our attributes that are color-coded based on the calculated season. The blue color is for winter, the orange color is for spring, the green is for summer, and the red is for autumn. In this figure, like the heatmap figure we can see the correlation in all seasons between all the pollutants. The histograms on the diagonal of this figure allow us to see the distribution of each feature. This can help us see if a feature is normally distributed, skewed, or has multiple modes.









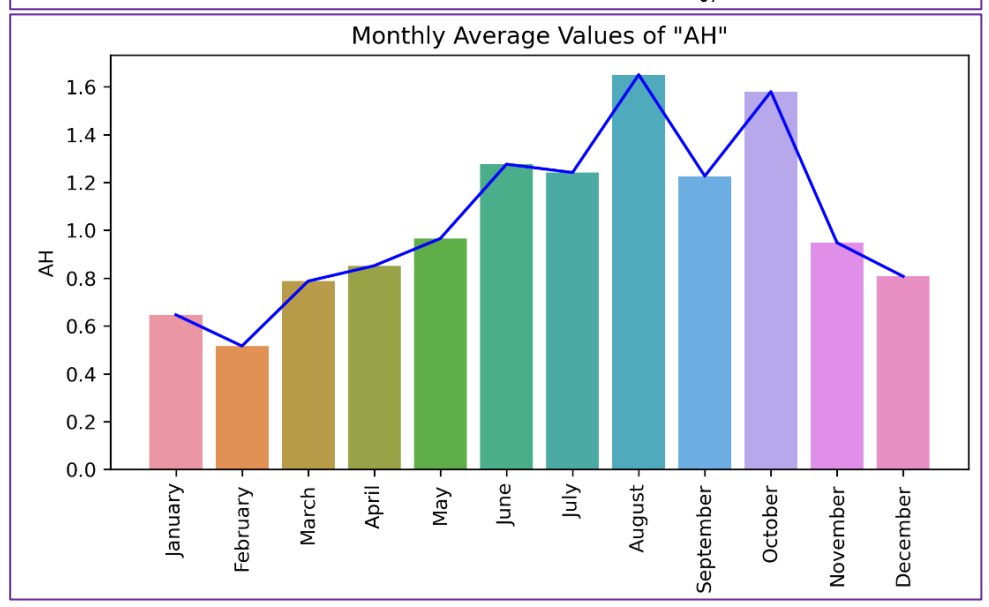


Figure 8: To look at the trend of each attribute, I have extracted Monthly Average Values for each attribute. This figure shows the average values of each attribute of our dataset in 12 months of a year. We can clearly see that in August, we have the minimum values of pollution, maximum temperature, and maximum absolute humidity.