0 Date (DD/MM/YYYY)
1 Time (HH.MM.SS)
2 True hourly averaged concentration CO in mg/m^3 (reference analyzer)
3 PT08.S1 (tin oxide) hourly averaged sensor response (nominally CO targeted)
4 True hourly averaged overall Non Metanic HydroCarbons concentration in microg/m^3 (reference analyzer)
5 True hourly averaged Benzene concentration in microg/m^3 (reference analyzer)
6 PT08.S2 (titania) hourly averaged sensor response (nominally NMHC targeted)
7 True hourly averaged NOx concentration in ppb (reference analyzer)
8 PT08.S3 (tungsten oxide) hourly averaged sensor response (nominally NOx targeted)
9 True hourly averaged NO2 concentration in microg/m^3 (reference analyzer)
10 PT08.S4 (tungsten oxide) hourly averaged sensor response (nominally NO2 targeted)
11 PT08.S5 (indium oxide) hourly averaged sensor response (nominally O3 targeted)
12 Temperature in Â°C
13 Relative Humidity (%)
14 AH Absolute Humidity

```
import pandas as pd
import numpy as np

df = pd.read_csv('AirQuality_visualization.csv',delimiter=';')

df
```

	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O
0	10/03/2004	18.00.00	2,6	1360.0	150.0	11,9	1046.0	166.0	1056.0	113.0	1692.0	126
1	10/03/2004	19.00.00	2	1292.0	112.0	9,4	955.0	103.0	1174.0	92.0	1559.0	97
2	10/03/2004	20.00.00	2,2	1402.0	88.0	9,0	939.0	131.0	1140.0	114.0	1555.0	107
3	10/03/2004	21.00.00	2,2	1376.0	80.0	9,2	948.0	172.0	1092.0	122.0	1584.0	120
4	10/03/2004	22.00.00	1,6	1272.0	51.0	6,5	836.0	131.0	1205.0	116.0	1490.0	111
...	...	...	...	...	...	...	...	...	...	...	...	
9466	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
9467	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
9468	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
9469	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
9470	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N

9471 rows × 17 columns

```
df = df.rename(columns={'T': 'Temperature'})
df = df.rename(columns={'RH': 'Relative Humidity'})
df = df.rename(columns={'AH': 'Absolute Humidity'})
df
```

	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O
0	10/03/2004	18.00.00	2,6	1360.0	150.0	11,9	1046.0	166.0	1056.0	113.0	1692.0	126
1	10/03/2004	19.00.00	2	1292.0	112.0	9,4	955.0	103.0	1174.0	92.0	1559.0	97
2	10/03/2004	20.00.00	2,2	1402.0	88.0	9,0	939.0	131.0	1140.0	114.0	1555.0	107
3	10/03/2004	21.00.00	2,2	1376.0	80.0	9,2	948.0	172.0	1092.0	122.0	1584.0	120
4	10/03/2004	22.00.00	1,6	1272.0	51.0	6,5	836.0	131.0	1205.0	116.0	1490.0	111
...	...	...	...	...	...	...	...	...	...	...	...	
9466	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
9467	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
9468	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
9469	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
9470	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N

9471 rows × 17 columns

```
df = df.drop(['Unnamed: 15', 'Unnamed: 16'],axis=1)

df
```

	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O
0	10/03/2004	18.00.00	2,6	1360.0	150.0	11,9	1046.0	166.0	1056.0	113.0	1692.0	126
1	10/03/2004	19.00.00	2	1292.0	112.0	9,4	955.0	103.0	1174.0	92.0	1559.0	97
2	10/03/2004	20.00.00	2,2	1402.0	88.0	9,0	939.0	131.0	1140.0	114.0	1555.0	107
3	10/03/2004	21.00.00	2,2	1376.0	80.0	9,2	948.0	172.0	1092.0	122.0	1584.0	120
4	10/03/2004	22.00.00	1,6	1272.0	51.0	6,5	836.0	131.0	1205.0	116.0	1490.0	111
...	...	...	...	...	...	...	...	...	...	...	...	
9466	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
9467	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
9468	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
9469	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
9470	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N

9471 rows × 15 columns

```
df['CO(GT)'] = df['CO(GT)'].str.replace(',', '.').astype(float)
df['C6H6(GT)'] = df['C6H6(GT)'].str.replace(',', '.').astype(float)
df['Temperature'] = df['Temperature'].str.replace(',', '.').astype(float)
df['Relative Humidity'] = df['Relative Humidity'].str.replace(',', '.').astype(float)
df['Absolute Humidity'] = df['Absolute Humidity'].str.replace(',', '.').astype(float)
df
```

	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O
0	10/03/2004	18.00.00	2.6	1360.0	150.0	11.9	1046.0	166.0	1056.0	113.0	1692.0	126
1	10/03/2004	19.00.00	2.0	1292.0	112.0	9.4	955.0	103.0	1174.0	92.0	1559.0	97
2	10/03/2004	20.00.00	2.2	1402.0	88.0	9.0	939.0	131.0	1140.0	114.0	1555.0	107
3	10/03/2004	21.00.00	2.2	1376.0	80.0	9.2	948.0	172.0	1092.0	122.0	1584.0	120
4	10/03/2004	22.00.00	1.6	1272.0	51.0	6.5	836.0	131.0	1205.0	116.0	1490.0	111
...	...	...	...	...	...	...	...	...	...	...	...	
9466	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
9467	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
9468	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
9469	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
9470	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N

9471 rows × 15 columns

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9471 entries, 0 to 9470
Data columns (total 15 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Date               9357 non-null   object 
 1   Time               9357 non-null   object 
 2   CO(GT)             9357 non-null   float64
 3   PT08.S1(CO)        9357 non-null   float64
 4   NMHC(GT)           9357 non-null   float64
 5   C6H6(GT)           9357 non-null   float64
 6   PT08.S2(NMHC)      9357 non-null   float64
 7   NOx(GT)            9357 non-null   float64
 8   PT08.S3(NOx)       9357 non-null   float64
 9   NO2(GT)            9357 non-null   float64
 10  PT08.S4(NO2)       9357 non-null   float64
 11  PT08.S5(O3)        9357 non-null   float64
 12  Temperature        9357 non-null   float64
 13  Relative Humidity  9357 non-null   float64
 14  Absolute Humidity  9357 non-null   float64
dtypes: float64(13), object(2)
memory usage: 1.1+ MB
```

```
df.head()
```

	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O3)
0	10/03/2004	18.00.00	2.6	1360.0	150.0	11.9	1046.0	166.0	1056.0	113.0	1692.0	1268.0
1	10/03/2004	19.00.00	2.0	1292.0	112.0	9.4	955.0	103.0	1174.0	92.0	1559.0	972.0
2	10/03/2004	20.00.00	2.2	1402.0	88.0	9.0	939.0	131.0	1140.0	114.0	1555.0	1074.0
3	10/03/2004	21.00.00	2.2	1376.0	80.0	9.2	948.0	172.0	1092.0	122.0	1584.0	1203.0
4	10/03/2004	22.00.00	1.6	1272.0	51.0	6.5	836.0	131.0	1205.0	116.0	1490.0	1110.0

```
df=df.drop_duplicates()
```

```
df.isna().sum()
```

```
Date                1
Time                1
CO(GT)              1
PT08.S1(CO)         1
NMHC(GT)            1
C6H6(GT)            1
PT08.S2(NMHC)       1
NOx(GT)             1
```

```
PT08.S3(NOx)         1
NO2(GT)              1
PT08.S4(NO2)         1
PT08.S5(O3)          1
Temperature          1
Relative Humidity    1
Absolute Humidity    1
dtype: int64
```

```
df = df.fillna(df.mean())
df = df.dropna()
```

```
C:\Users\Kumbh\AppData\Local\Temp\ipykernel_11904\1870714312.py:1: FutureWarning: The default value of numeric_only in DataFrame.mean is
  df = df.fillna(df.mean())
```

```
df.isna().sum()
```

```
Date                 0
Time                 0
CO(GT)               0
PT08.S1(CO)          0
NMHC(GT)             0
C6H6(GT)             0
PT08.S2(NMHC)        0
NOx(GT)              0
PT08.S3(NOx)         0
NO2(GT)              0
PT08.S4(NO2)         0
PT08.S5(O3)          0
Temperature          0
Relative Humidity    0
Absolute Humidity    0
dtype: int64
```

```
df
```

	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O
0	10/03/2004	18.00.00	2.6	1360.0	150.0	11.9	1046.0	166.0	1056.0	113.0	1692.0	126
1	10/03/2004	19.00.00	2.0	1292.0	112.0	9.4	955.0	103.0	1174.0	92.0	1559.0	97
2	10/03/2004	20.00.00	2.2	1402.0	88.0	9.0	939.0	131.0	1140.0	114.0	1555.0	107
3	10/03/2004	21.00.00	2.2	1376.0	80.0	9.2	948.0	172.0	1092.0	122.0	1584.0	120
4	10/03/2004	22.00.00	1.6	1272.0	51.0	6.5	836.0	131.0	1205.0	116.0	1490.0	111
...	...	...	...	...	...	...	...	...	...	...	...	
9352	04/04/2005	10.00.00	3.1	1314.0	-200.0	13.5	1101.0	472.0	539.0	190.0	1374.0	172
9353	04/04/2005	11.00.00	2.4	1163.0	-200.0	11.4	1027.0	353.0	604.0	179.0	1264.0	126
9354	04/04/2005	12.00.00	2.4	1142.0	-200.0	12.4	1063.0	293.0	603.0	175.0	1241.0	109
9355	04/04/2005	13.00.00	2.1	1003.0	-200.0	9.5	961.0	235.0	702.0	156.0	1041.0	77
9356	04/04/2005	14.00.00	2.2	1071.0	-200.0	11.9	1047.0	265.0	654.0	168.0	1129.0	81

9357 rows × 15 columns

```
df['Absolute Humidity'] = df['Absolute Humidity'].multiply(100)
df
```

	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O
0	10/03/2004	18.00.00	2.6	1360.0	150.0	11.9	1046.0	166.0	1056.0	113.0	1692.0	126
1	10/03/2004	19.00.00	2.0	1292.0	112.0	9.4	955.0	103.0	1174.0	92.0	1559.0	97
2	10/03/2004	20.00.00	2.2	1402.0	88.0	9.0	939.0	131.0	1140.0	114.0	1555.0	107
3	10/03/2004	21.00.00	2.2	1376.0	80.0	9.2	948.0	172.0	1092.0	122.0	1584.0	120
4	10/03/2004	22.00.00	1.6	1272.0	51.0	6.5	836.0	131.0	1205.0	116.0	1490.0	111
...	...	...	...	...	...	...	...	...	...	...	...	
9352	04/04/2005	10.00.00	3.1	1314.0	-200.0	13.5	1101.0	472.0	539.0	190.0	1374.0	172
9353	04/04/2005	11.00.00	2.4	1163.0	-200.0	11.4	1027.0	353.0	604.0	179.0	1264.0	126
9354	04/04/2005	12.00.00	2.4	1142.0	-200.0	12.4	1063.0	293.0	603.0	175.0	1241.0	109
9355	04/04/2005	13.00.00	2.1	1003.0	-200.0	9.5	961.0	235.0	702.0	156.0	1041.0	77
9356	04/04/2005	14.00.00	2.2	1071.0	-200.0	11.9	1047.0	265.0	654.0	168.0	1129.0	81

9357 rows × 15 columns

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


def remove_outliers(column):
    Q1 = column.quantile(0.25)
    Q3 = column.quantile(0.75)
    IQR = Q3 - Q1
    threshold = 1.5 * IQR
    outlier_mask = (column < Q1 - threshold) | (column > Q3 + threshold)
    return column[~outlier_mask]


df.columns
```

```
Index(['Date', 'Time', 'CO(GT)', 'PT08.S1(CO)', 'NMHC(GT)', 'C6H6(GT)',
       'PT08.S2(NMHC)', 'NOx(GT)', 'PT08.S3(NOx)', 'NO2(GT)', 'PT08.S4(NO2)',
       'PT08.S5(O3)', 'Temperature', 'Relative Humidity', 'Absolute Humidity'],
      dtype='object')
```

```
# Remove outliers for each column using a loop
col_name = ['Temperature','Relative Humidity','Absolute Humidity','PT08.S4(NO2)','PT08.S5(O3)','C6H6(GT)',
       'PT08.S2(NMHC)','PT08.S1(CO)']
for col in col_name:
    df[col] = remove_outliers(df[col])


df['Year'] = pd.to_datetime(df['Date']).dt.year
df['Month'] = pd.to_datetime(df['Date']).dt.month

df
```

```
C:\Users\Kumbh\AppData\Local\Temp\ipykernel_11904\2904881021.py:1: UserWarning: Parsing dates in DD/MM/YYYY format when dayfirst=False (
  df['Year'] = pd.to_datetime(df['Date']).dt.year
C:\Users\Kumbh\AppData\Local\Temp\ipykernel_11904\2904881021.py:2: UserWarning: Parsing dates in DD/MM/YYYY format when dayfirst=False (
  df['Month'] = pd.to_datetime(df['Date']).dt.month
```

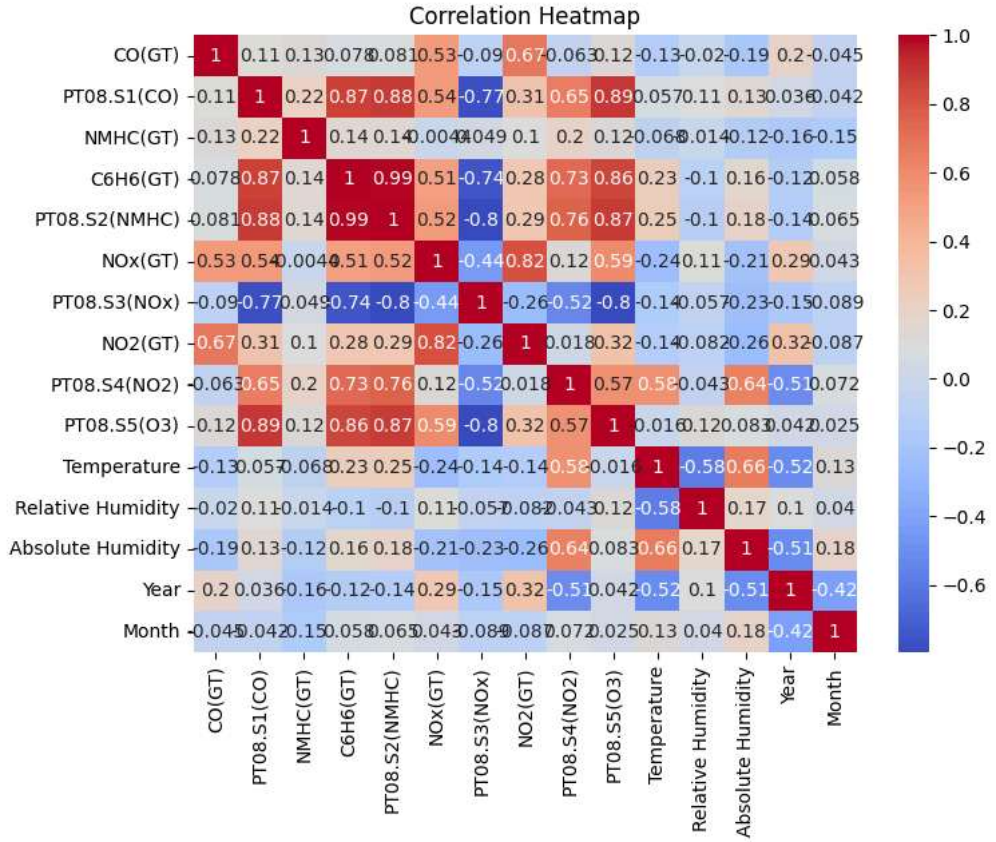
	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O
0	10/03/2004	18.00.00	2.6	1360.0	150.0	11.9	1046.0	166.0	1056.0	113.0	1692.0	126
1	10/03/2004	19.00.00	2.0	1292.0	112.0	9.4	955.0	103.0	1174.0	92.0	1559.0	97
2	10/03/2004	20.00.00	2.2	1402.0	88.0	9.0	939.0	131.0	1140.0	114.0	1555.0	107
3	10/03/2004	21.00.00	2.2	1376.0	80.0	9.2	948.0	172.0	1092.0	122.0	1584.0	120
4	10/03/2004	22.00.00	1.6	1272.0	51.0	6.5	836.0	131.0	1205.0	116.0	1490.0	111
...	...	...	...	...	...	...	...	...	...	...	...	
9352	04/04/2005	10.00.00	3.1	1314.0	-200.0	13.5	1101.0	472.0	539.0	190.0	1374.0	172
9353	04/04/2005	11.00.00	2.4	1163.0	-200.0	11.4	1027.0	353.0	604.0	179.0	1264.0	126
9354	04/04/2005	12.00.00	2.4	1142.0	-200.0	12.4	1063.0	293.0	603.0	175.0	1241.0	109
9355	04/04/2005	13.00.00	2.1	1003.0	-200.0	9.5	961.0	235.0	702.0	156.0	1041.0	77
9356	04/04/2005	14.00.00	2.2	1071.0	-200.0	11.9	1047.0	265.0	654.0	168.0	1129.0	81

9357 rows × 17 columns

```
df['yearr']= df.Year.astype(str)
df['month']= df.Month.astype(str)


# Plot correlation heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```

```
C:\Users\Kumbh\AppData\Local\Temp\ipykernel_11904\2547940491.py:3: FutureWarning: The default value of numeric_only in DataFrame.corr is
  sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
```



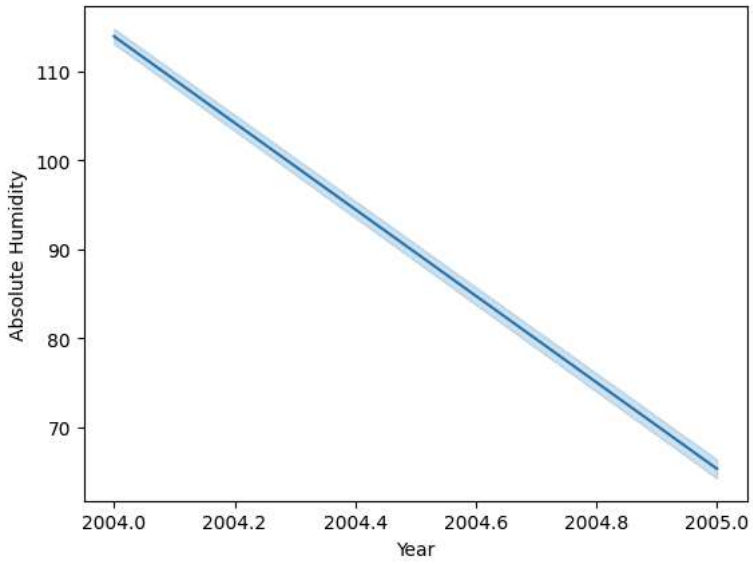
```
df.columns
```

```
Index(['Date', 'Time', 'CO(GT)', 'PT08.S1(CO)', 'NMHC(GT)', 'C6H6(GT)',
       'PT08.S2(NMHC)', 'NOx(GT)', 'PT08.S3(NOx)', 'NO2(GT)', 'PT08.S4(NO2)',
       'PT08.S5(O3)', 'Temperature', 'Relative Humidity', 'Absolute Humidity',
       'Year', 'Month', 'yearr', 'month'],
      dtype='object')
```

LinePlot

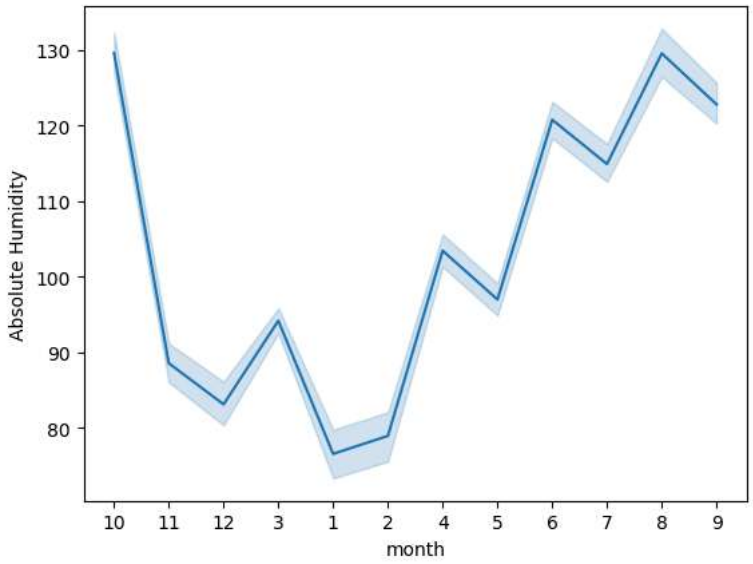
```
sns.lineplot(df,x="Year",y='Absolute Humidity')
```

```
<Axes: xlabel='Year', ylabel='Absolute Humidity'>
```



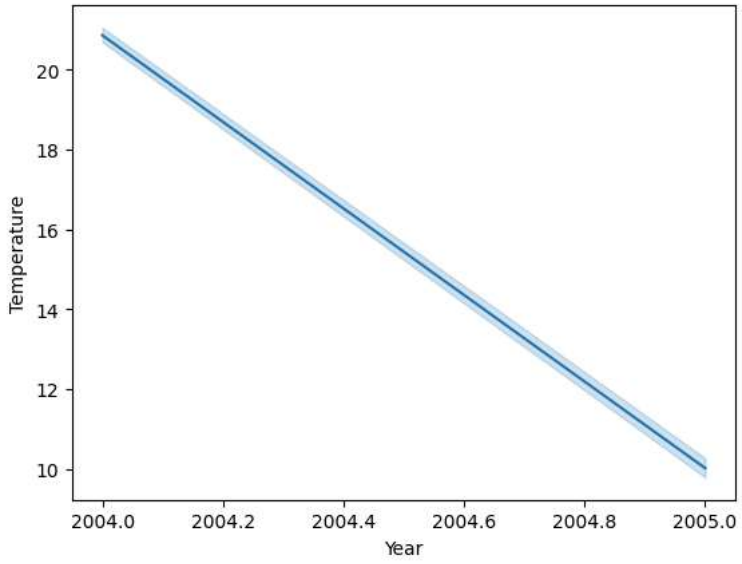

```
sns.lineplot(df,x="month",y='Absolute Humidity')
```

```
<Axes: xlabel='month', ylabel='Absolute Humidity'>
```



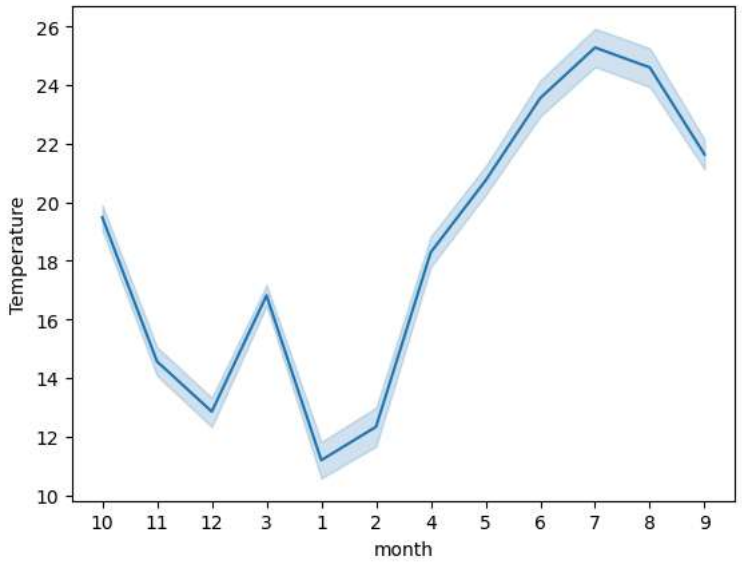

```
sns.lineplot(df,x="Year",y='Temperature')
```

```
<Axes: xlabel='Year', ylabel='Temperature'>
```



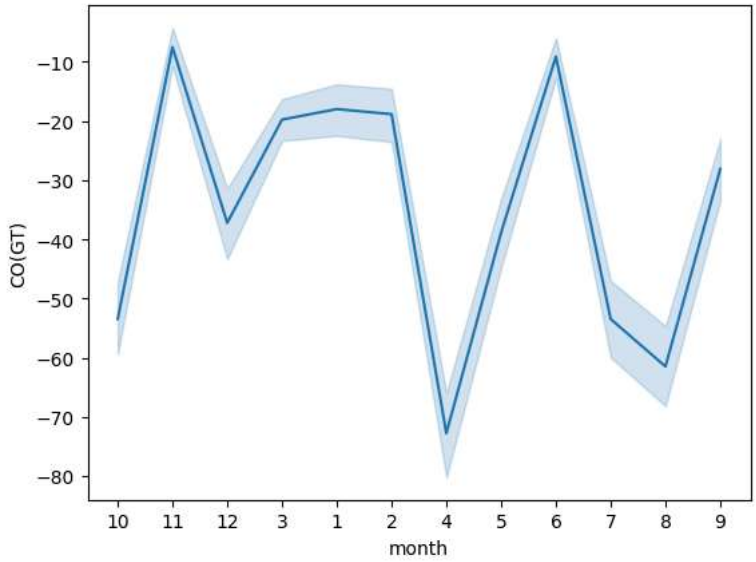

```
sns.lineplot(df,x="month",y='Temperature',)
```

```
<Axes: xlabel='month', ylabel='Temperature'>
```



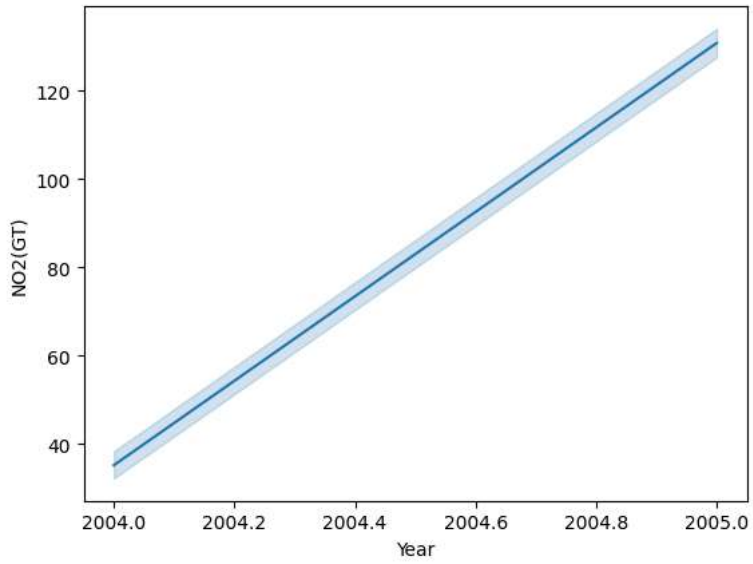

```
sns.lineplot(df,x="month",y='CO(GT)',)
```

```
<Axes: xlabel='month', ylabel='CO(GT)'>
```



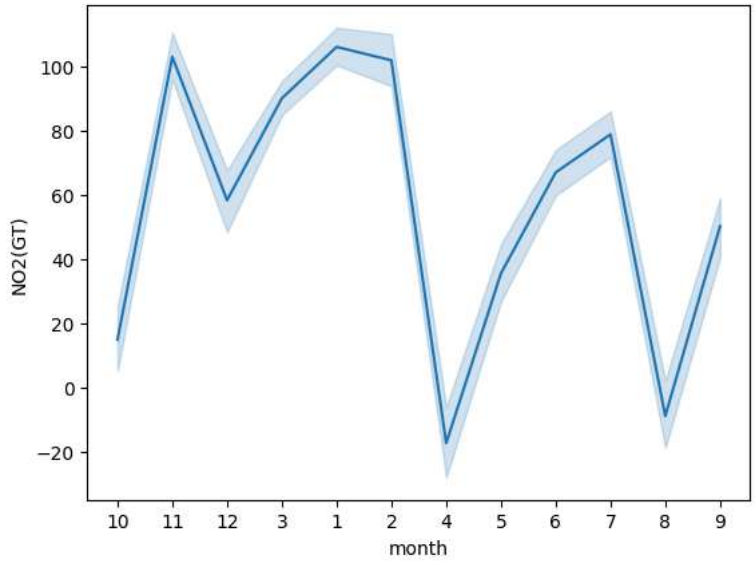

```
sns.lineplot(df,x="Year",y='NO2(GT)')
```

```
<Axes: xlabel='Year', ylabel='NO2(GT)'>
```



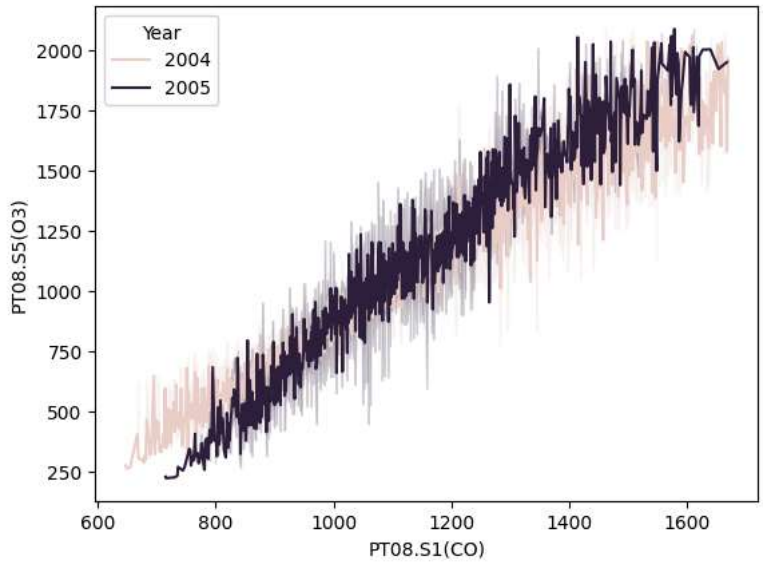

```
sns.lineplot(df,x="month",y='NO2(GT)')
```

```
<Axes: xlabel='month', ylabel='NO2(GT)'>
```



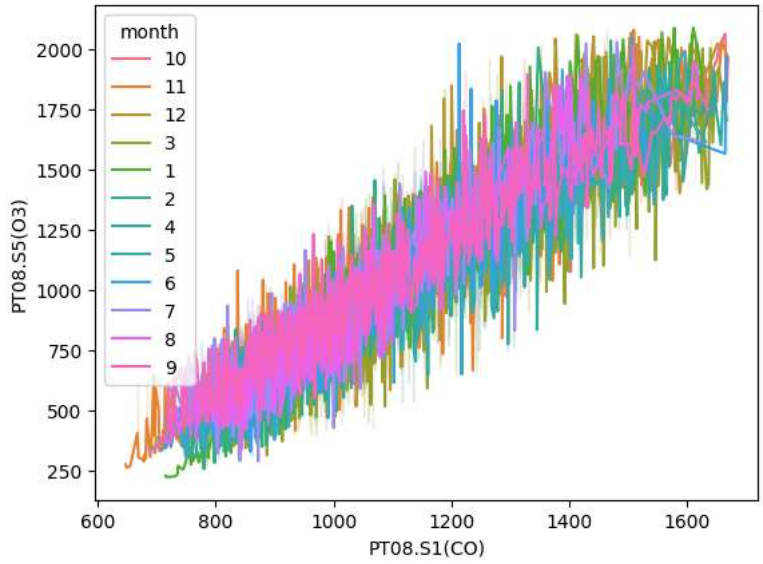

```
sns.lineplot(df,x='PT08.S1(CO)',y='PT08.S5(O3)',hue='Year')
```

```
<Axes: xlabel='PT08.S1(CO)', ylabel='PT08.S5(O3)'>
```




```
# sns.lineplot(df,x='PT08.S1(CO)',y='PT08.S5(O3)',hue='month')
```

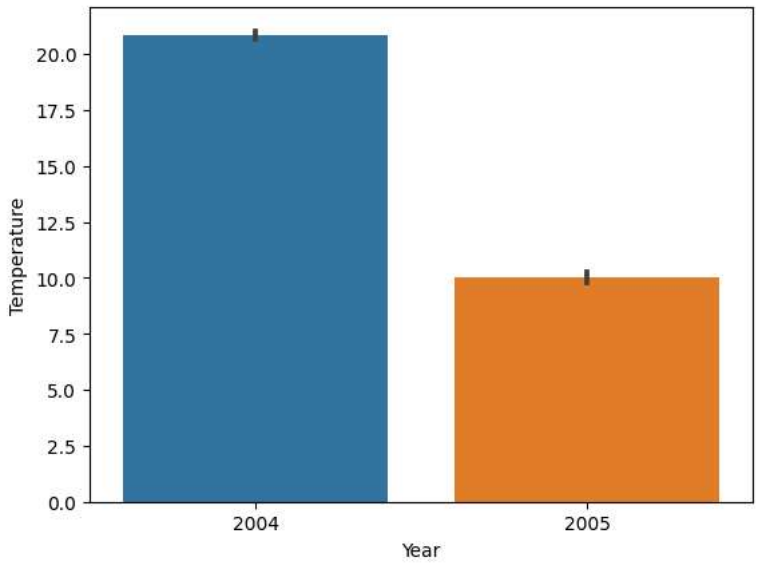
```
<Axes: xlabel='PT08.S1(CO)', ylabel='PT08.S5(O3)'>
```



Barplot

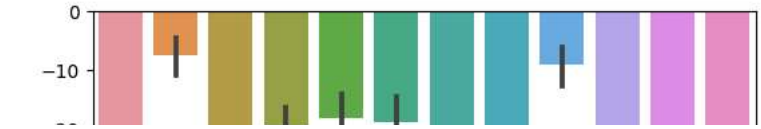
```
sns.barplot(df,x=df.Year,y=df.Temperature)
```

```
<Axes: xlabel='Year', ylabel='Temperature'>
```



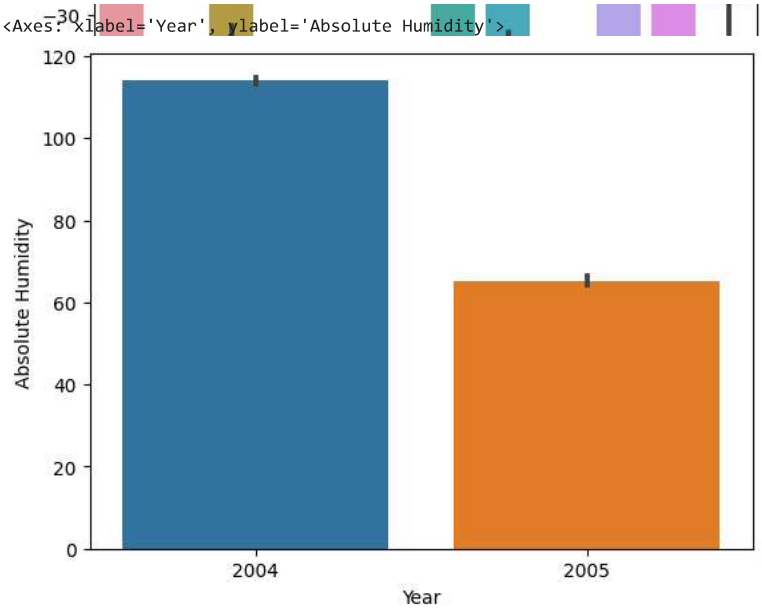

```
sns.barplot(df,x=df.month,y=df['CO(GT)'])
```

```
<Axes: xlabel='month', ylabel='CO(GT)'>
```



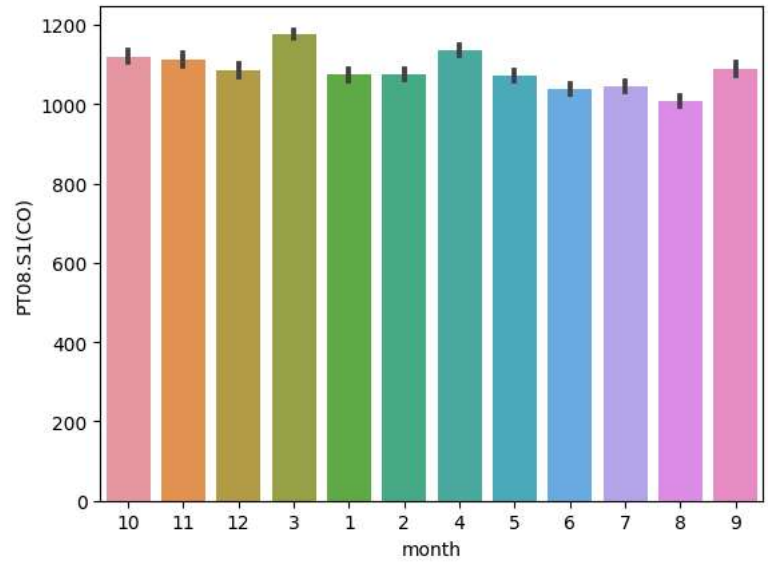
```
sns.barplot(df,x=df.Year,y='Absolute Humidity')
```

```
<Axes: xlabel='Year', ylabel='Absolute Humidity'>
```




```
sns.barplot(df,x=df.month,y='PT08.S1(CO)')
```

```
<Axes: xlabel='month', ylabel='PT08.S1(CO)'>
```



Scatter Plot