DSBDA Assignment 7 - Visualization on Air Quality Dataset

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
df = pd.read_csv('AirQuality_visualization.csv',delimiter=';')
```

In [3]:
```
df.head()
```

Out[3]:

	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(N
0	10/03/2004	18.00.00	2,6	1360.0	150.0	11,9	1046.0	166.0	1056.0	113.0	16
1	10/03/2004	19.00.00	2	1292.0	112.0	9,4	955.0	103.0	1174.0	92.0	15
2	10/03/2004	20.00.00	2,2	1402.0	88.0	9,0	939.0	131.0	1140.0	114.0	15
3	10/03/2004	21.00.00	2,2	1376.0	80.0	9,2	948.0	172.0	1092.0	122.0	15
4	10/03/2004	22.00.00	1,6	1272.0	51.0	6,5	836.0	131.0	1205.0	116.0	14

In [4]:
```
df = df.rename(columns={'T': 'Temperature'})
df = df.rename(columns={'RH': 'Relative Humidity'})
df = df.rename(columns={'AH': 'Absolute Humidity'})
df
```

Out[4]:

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

9471 rows × 17 columns

In [5]:
```
df = df.drop(['Unnamed: 15', 'Unnamed: 16'],axis=1)
df
```

Out[5]:

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

9471 rows × 15 columns

In [6]:
```
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
```

Out[6]:

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

9471 rows × 15 columns

In [7]:
```
df=df.drop_duplicates()
```

In [8]:
```
df.isna().sum()
```

Out[8]:
```
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
```

In [9]:
```
df = df.fillna(df.mean(numeric_only=True))  # Fill NaNs in numeric columns
df = df.dropna()  # Drop remaining NaNs (if any in non-numeric columns)
```

In [10]:
```
df.isna().sum()
```

Out[10]:
```
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
```

In [11]:
```
df['Absolute Humidity'] = df['Absolute Humidity'].multiply(100)
```

In [12]:
```
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]
```

In [13]:
```
df.columns
```

Out[13]:
```
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')
```

In [14]:
```
# 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])
```

In [15]:
```
df["Date"] = pd.to_datetime(df["Date"], format="%d/%m/%Y", errors='coerce')
df["Year"] = df["Date"].dt.year
df["Month"] = df["Date"].dt.month
```

In [16]:
```
df
```

Out[16]:

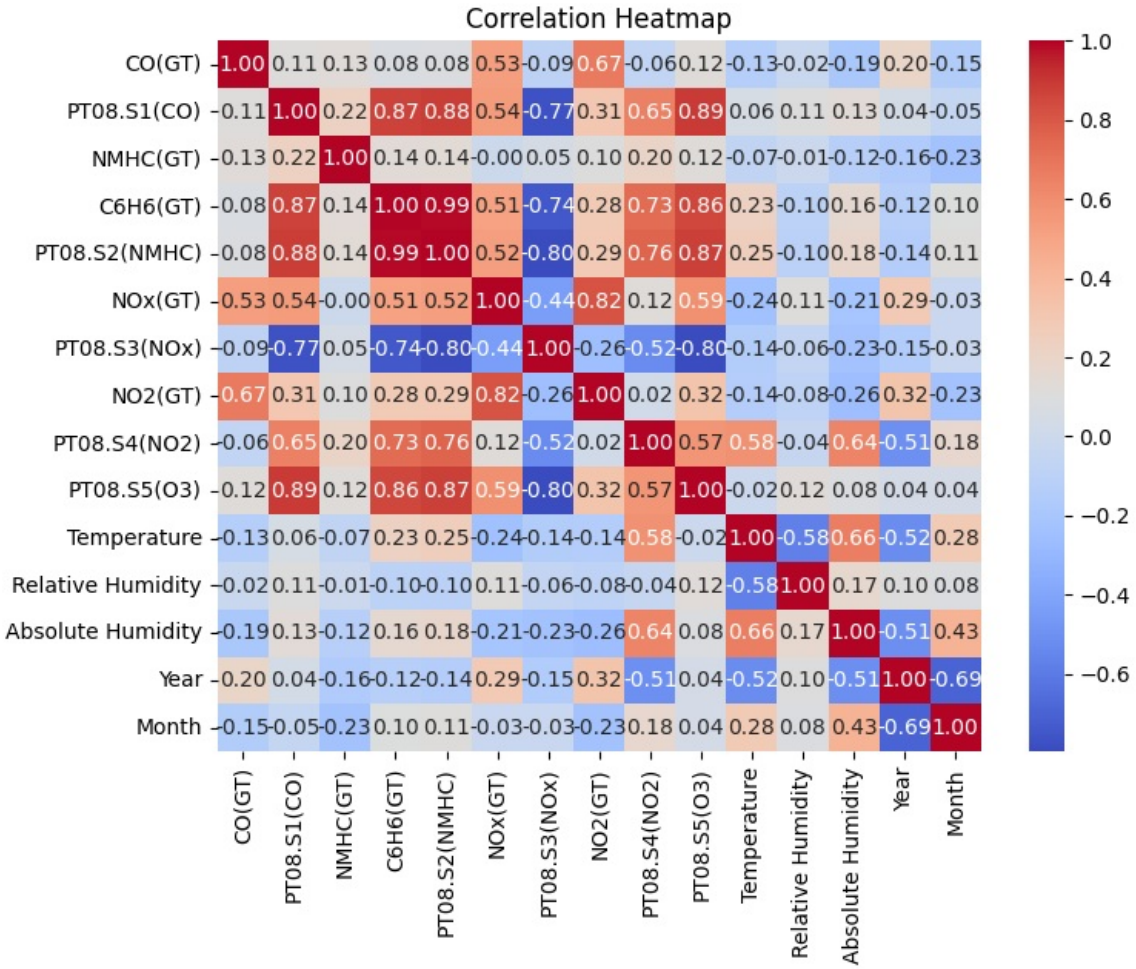
	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO
0	2004-03-10	18.00.00	2.6	1360.0	150.0	11.9	1046.0	166.0	1056.0	113.0	1692
1	2004-03-10	19.00.00	2.0	1292.0	112.0	9.4	955.0	103.0	1174.0	92.0	1559
2	2004-03-10	20.00.00	2.2	1402.0	88.0	9.0	939.0	131.0	1140.0	114.0	1555
3	2004-03-10	21.00.00	2.2	1376.0	80.0	9.2	948.0	172.0	1092.0	122.0	1584
4	2004-03-10	22.00.00	1.6	1272.0	51.0	6.5	836.0	131.0	1205.0	116.0	1490
...	...	...	...	...	...	...	...	...	...	...	
9352	2005-04-04	10.00.00	3.1	1314.0	-200.0	13.5	1101.0	472.0	539.0	190.0	1374
9353	2005-04-04	11.00.00	2.4	1163.0	-200.0	11.4	1027.0	353.0	604.0	179.0	1264
9354	2005-04-04	12.00.00	2.4	1142.0	-200.0	12.4	1063.0	293.0	603.0	175.0	1241
9355	2005-04-04	13.00.00	2.1	1003.0	-200.0	9.5	961.0	235.0	702.0	156.0	1041
9356	2005-04-04	14.00.00	2.2	1071.0	-200.0	11.9	1047.0	265.0	654.0	168.0	1129

9357 rows × 17 columns

In [17]:
```
df['yearr']= df.Year.astype(str)
df['month']= df.Month.astype(str)
```

1)Heatmap

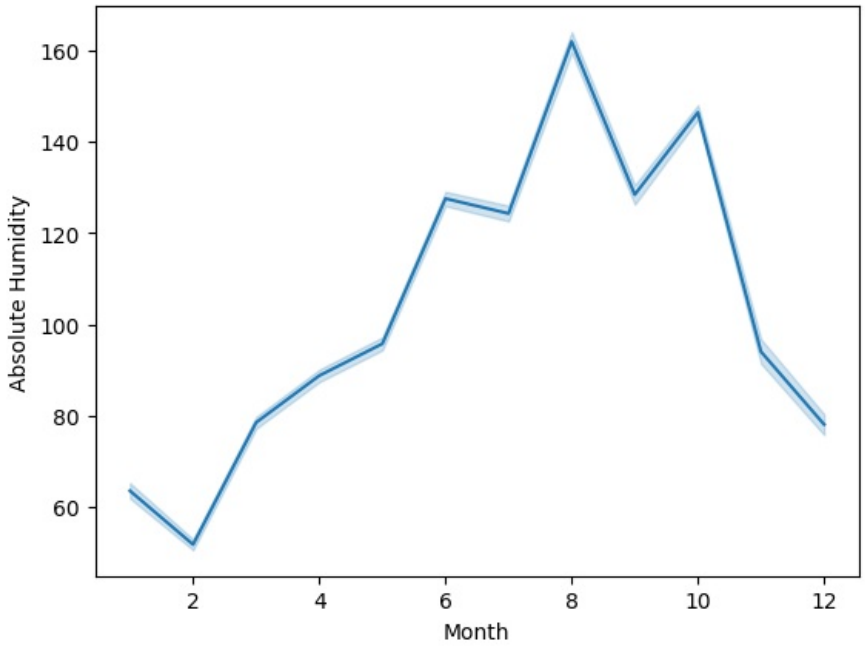
In [18]:
```
plt.figure(figsize=(8, 6))
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap')
plt.show()
```



2)Line plot

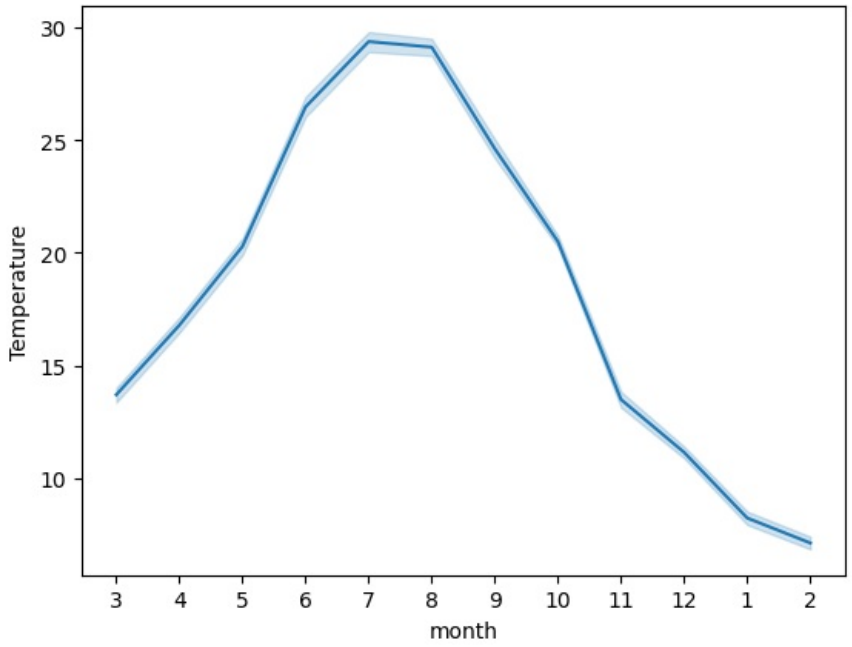
In [20]:
```
sns.lineplot(df,x="Month",y='Absolute Humidity')
```

Out[20]:
```
<Axes: xlabel='Month', ylabel='Absolute Humidity'>
```



In [25]:
```
sns.lineplot(df,x="month",y='Temperature',)
```

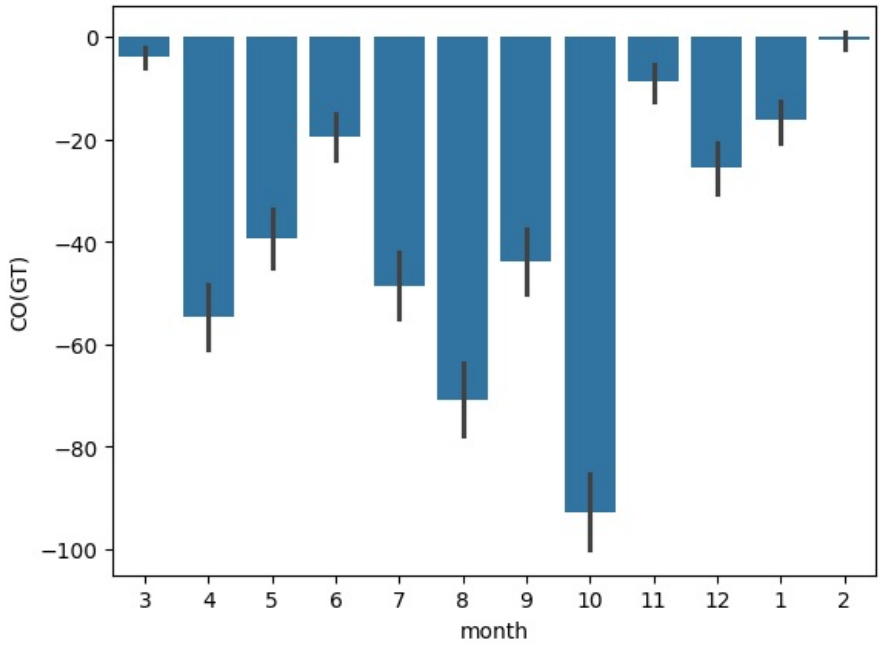
Out[25]: <Axes: xlabel='month', ylabel='Temperature'>



3)Bar plot

In [28]:
```
sns.barplot(df,x=df.month,y=df['CO(GT)'])
```

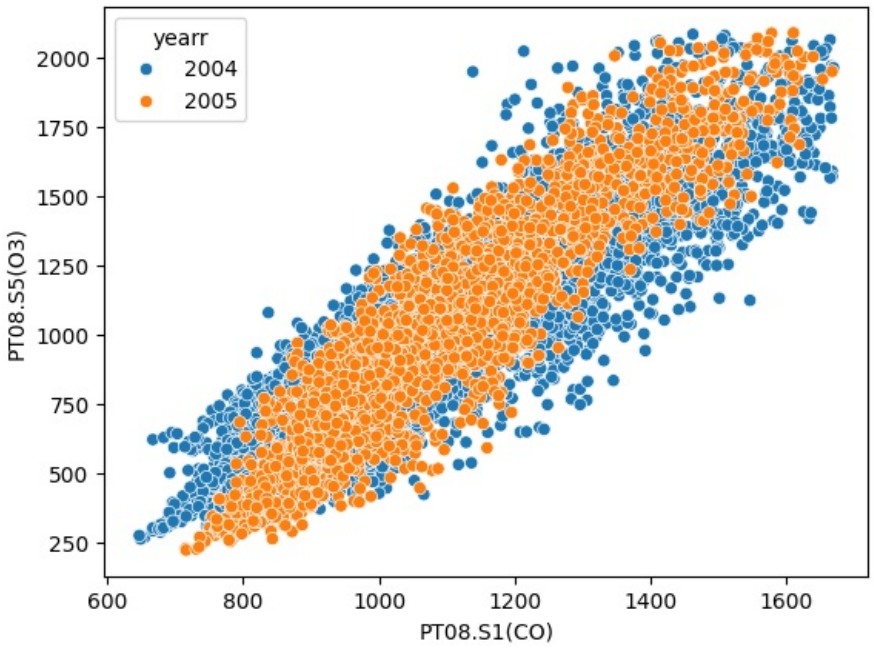
Out[28]: <Axes: xlabel='month', ylabel='CO(GT)'>



4)Scatter Plot

In [29]:
```
sns.scatterplot(df,x='PT08.S1(CO)',y='PT08.S5(O3)', hue='yearr')
```

Out[29]: <Axes: xlabel='PT08.S1(CO)', ylabel='PT08.S5(O3)'>

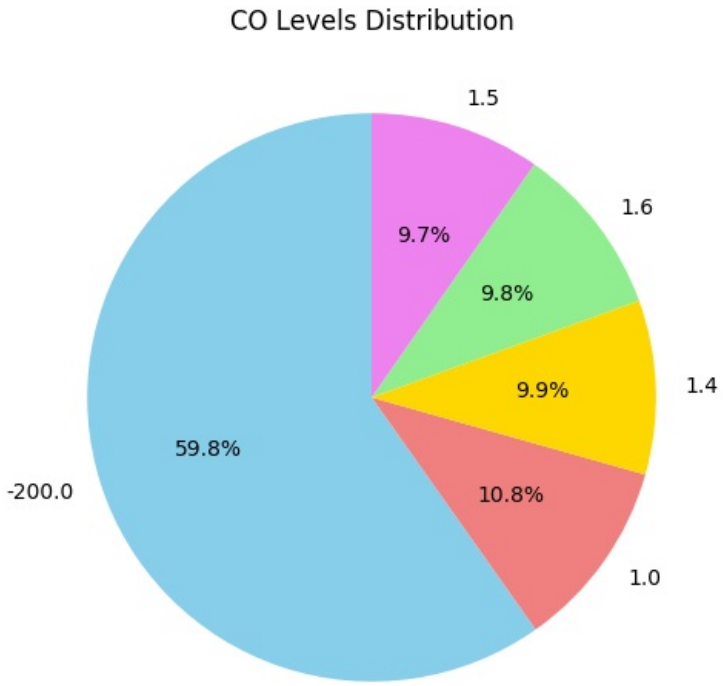


5)Pie Chart

In [36]:

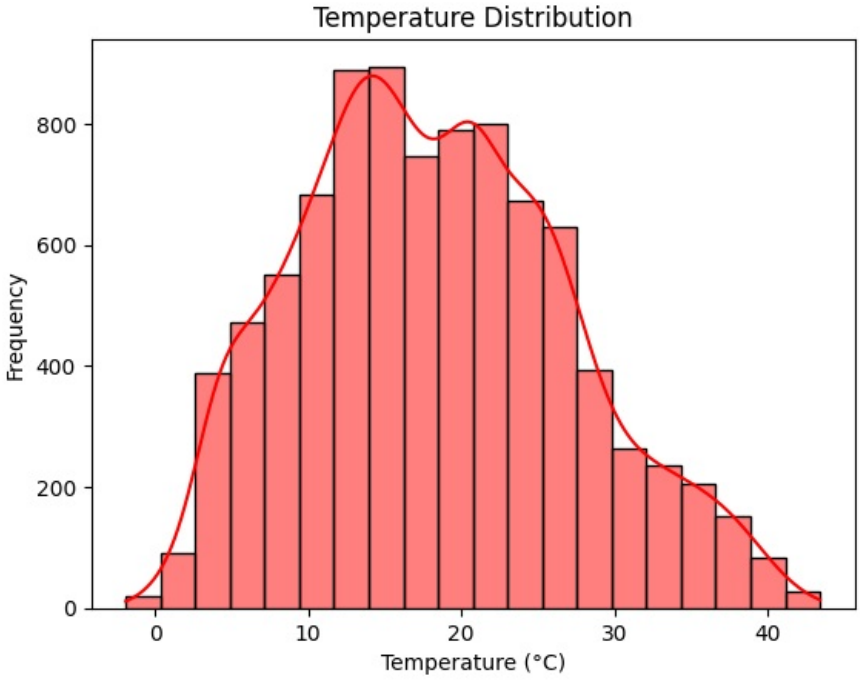
```
# Get the top 5 most frequent CO values
co_counts = df["CO(GT)"].value_counts().nlargest(5)

# Simple Pie Chart
plt.figure(figsize=(6, 6))
co_counts.plot.pie(
    autopct="%1.1f%%",
    colors=["skyblue", "lightcoral", "gold", "lightgreen", "violet"],
    startangle=90
)
plt.title("CO Levels Distribution")
plt.ylabel("")  # Hide default y-label
plt.show()
```



6)Histogram

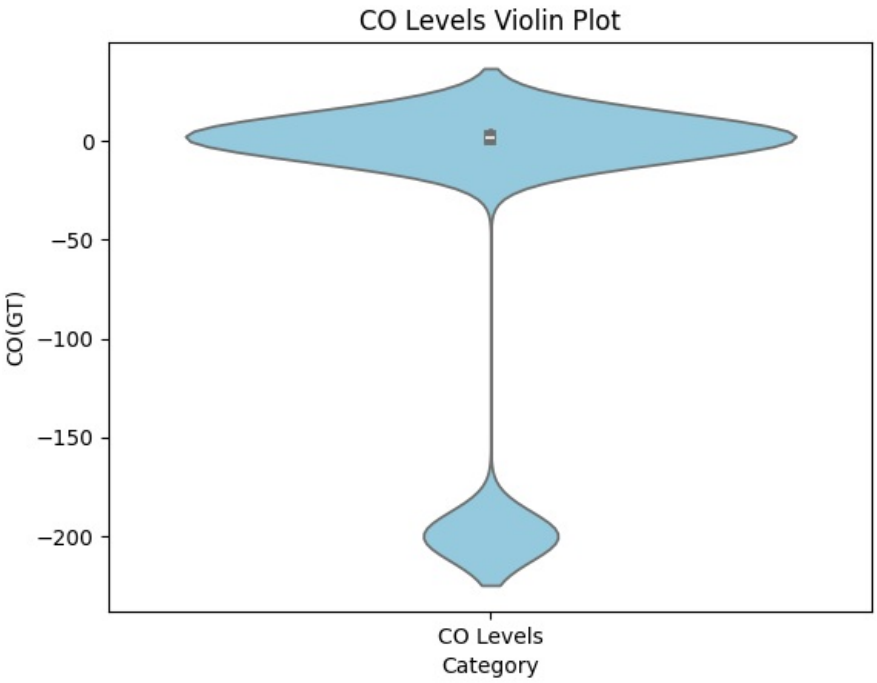
In [40]:
```
sns.histplot(df["Temperature"], bins=20, kde=True, color="red")
plt.title("Temperature Distribution")
plt.xlabel("Temperature (°C)")
plt.ylabel("Frequency")
plt.show()
```



7)Violin Plot

In [39]:
```
# Create a dummy category to place CO(GT) on x-axis
df["Category"] = "CO Levels"

sns.violinplot(x=df["Category"], y=df["CO(GT)"], color="skyblue")
plt.title("CO Levels Violin Plot")
plt.xlabel("Category")
plt.ylabel("CO(GT)")
plt.show()
```



8)Box plot

In [48]:
```
sns.boxplot(data=df, x='Month', y='Temperature')
plt.title('Box Plot of Temperature by Month')
plt.xlabel('Month')
plt.ylabel('Temperature')
plt.show()
```