visualization

April 29, 2025

[67]:

```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import pandas as pd
```

[68]:

```
df= pd.read_csv('heartdisease.csv')
df.head(5)
```

[68]:

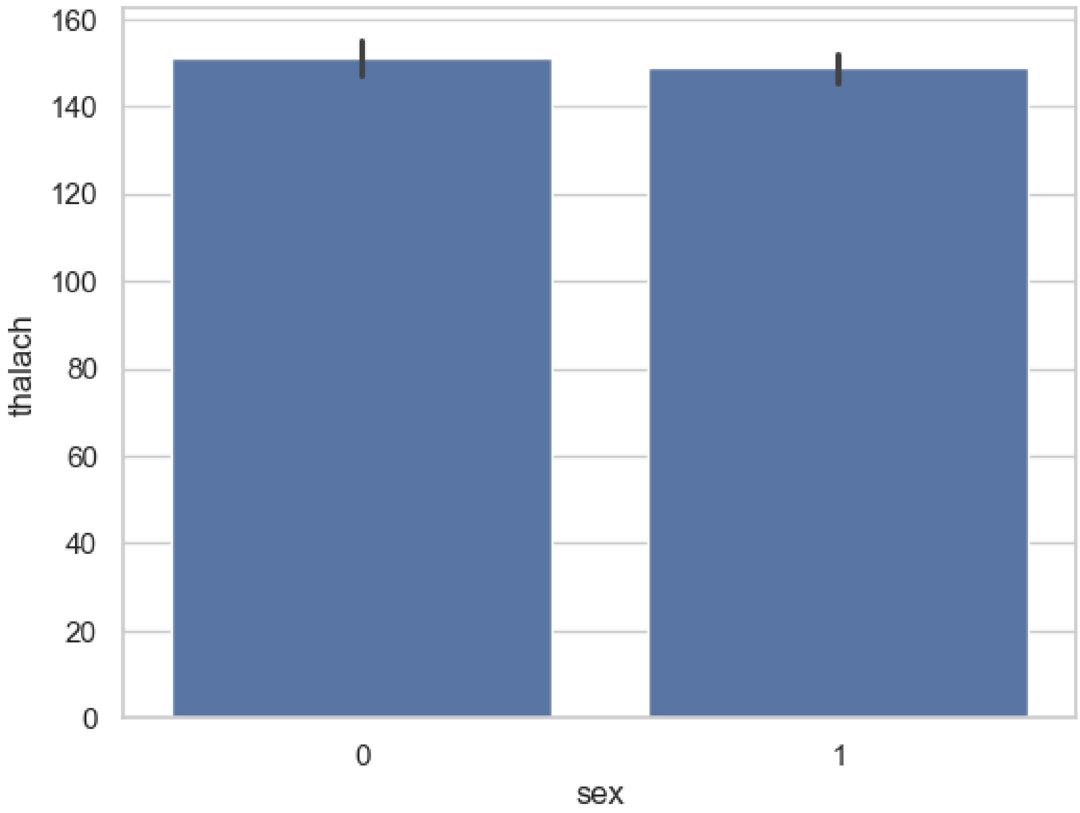
```
   age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  \
0   63    1   1       145   233    1        2      150      0      2.3      3
1   67    1   4       160   286    0        2      108      1      1.5      2
2   67    1   4       120   229    0        2      129      1      2.6      2
3   37    1   3       130   250    0        0      187      0      3.5      3
4   41    0   2       130   204    0        2      172      0      1.4      1

  ca thal  num
0  0    6    0
1  3    3    2
2  2    7    1
3  0    3    0
4  0    3    0
```

[69]:

```
#Barplot using seaborn
sns.barplot(x='sex', y='thalach', data = df)
```

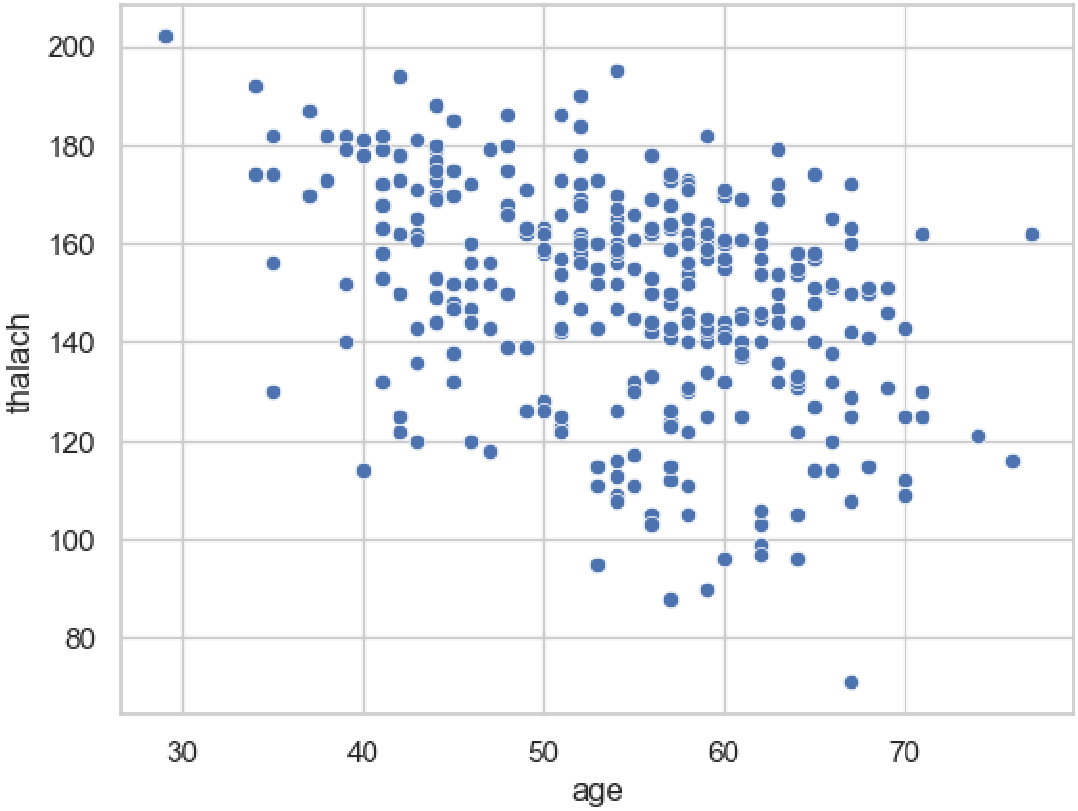
[69]:

```
<Axes: xlabel='sex', ylabel='thalach'>
```



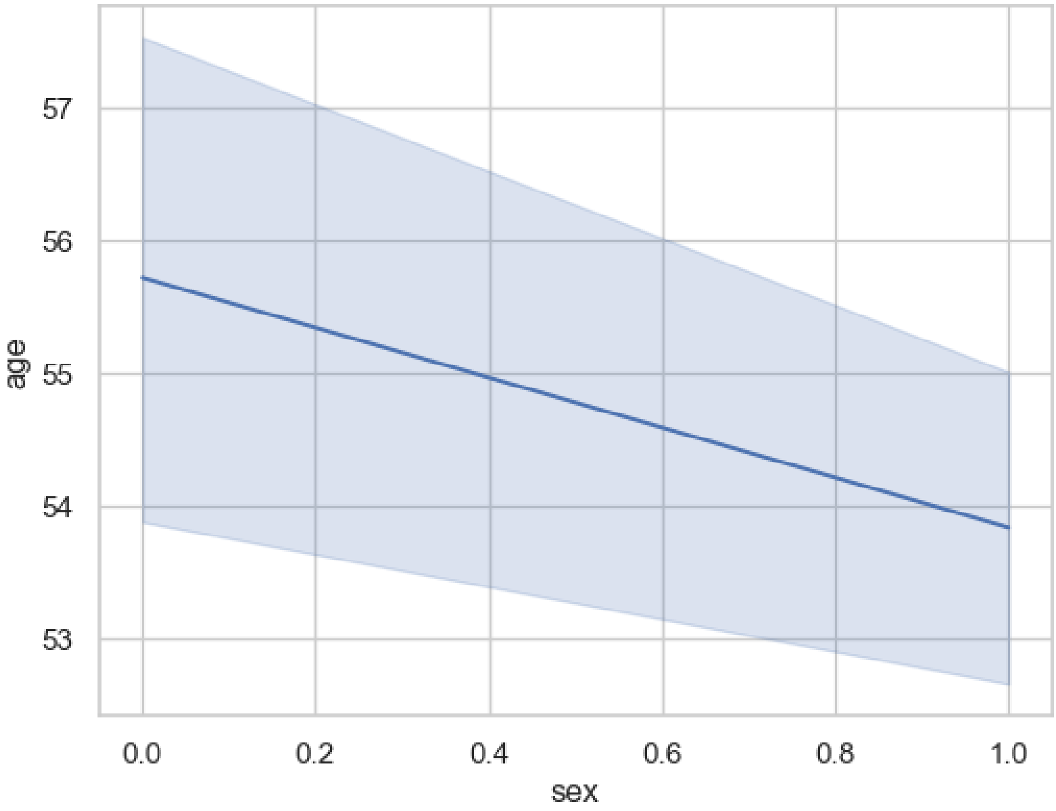
```
[70]: #ScatterPlot using seaborn
      sns.scatterplot(x='age', y='thalach', data = df)
```

```
[70]: <Axes: xlabel='age', ylabel='thalach'>
```



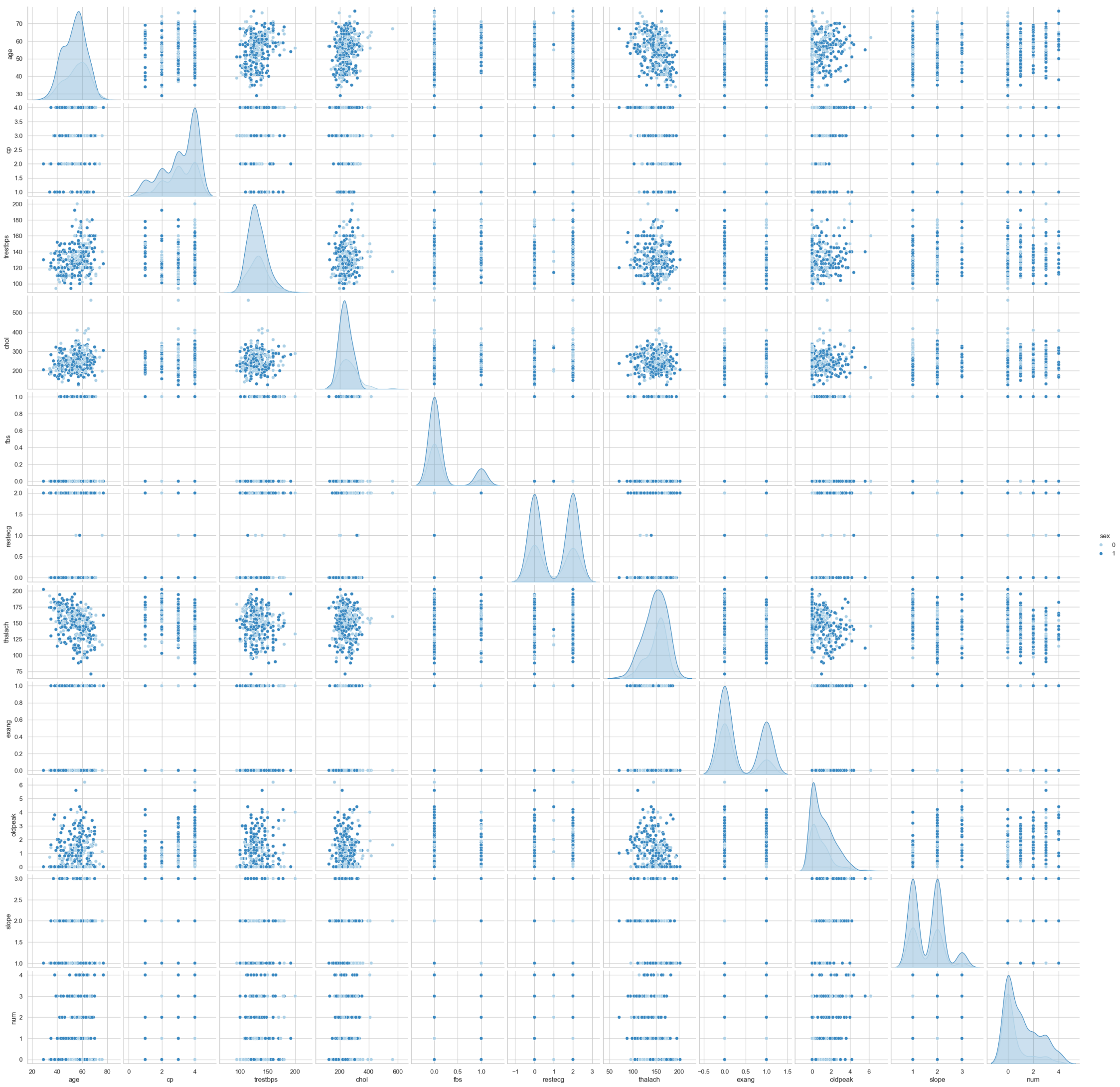
```
[71]: #Lineplot using seaborn
      sns.lineplot(x='sex', y='age', data = df)
```

```
[71]: <Axes: xlabel='sex', ylabel='age'>
```



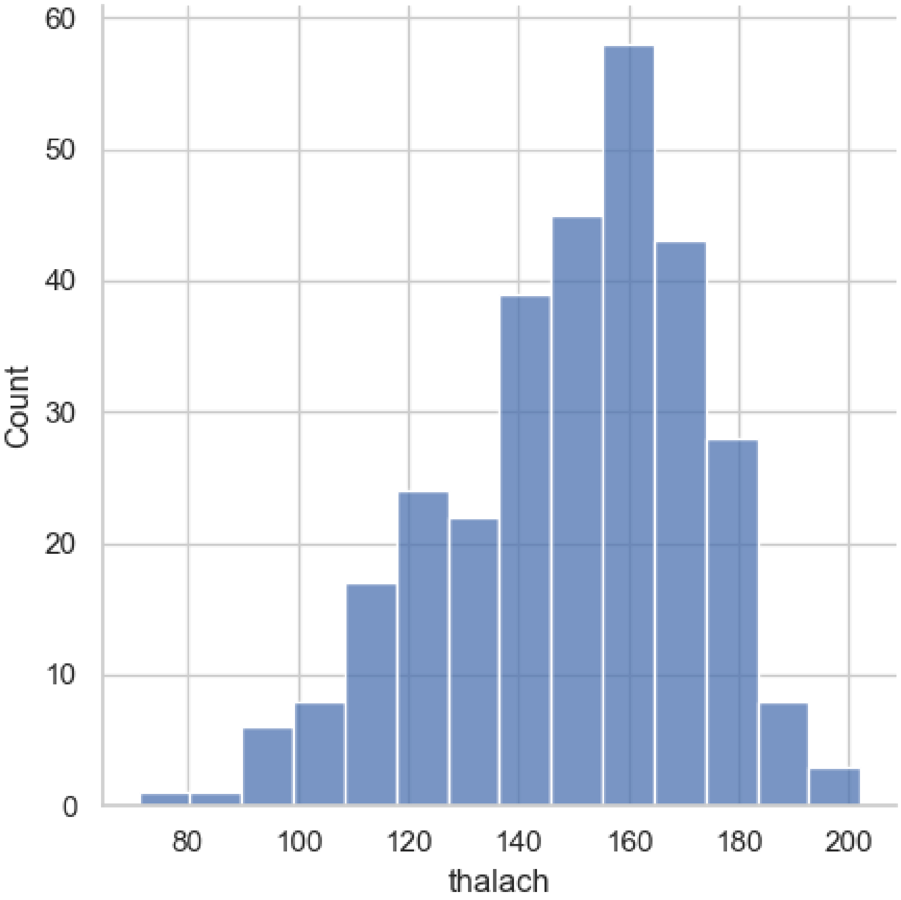
[72]:
```
#Pairplot using seaborn
plt.figure(figsize=(12,12))
sns.pairplot(df, hue= 'sex' , palette = 'Blues')
```

[72]:
```
<seaborn.axisgrid.PairGrid at 0x2ad758de270>

<Figure size 1200x1200 with 0 Axes>
```



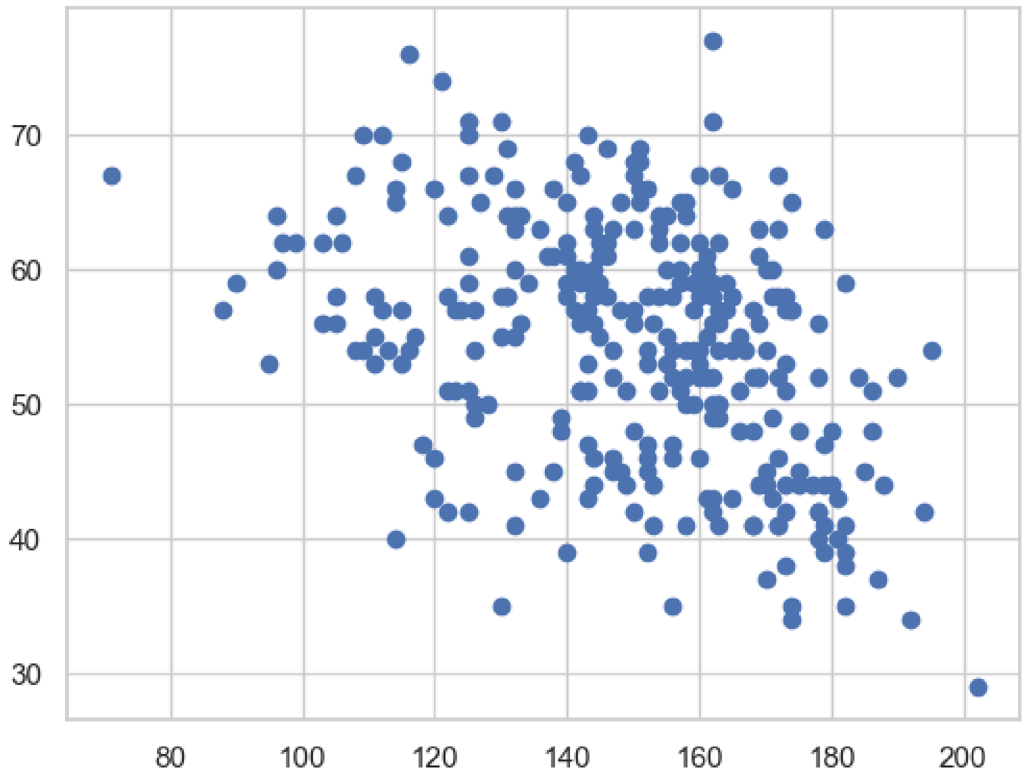
[73]:
```
#Displot using seaborn
sns.displot(df['thalach'])
```

[73]:
```
<seaborn.axisgrid.FacetGrid at 0x2ad7d1babd0>
```



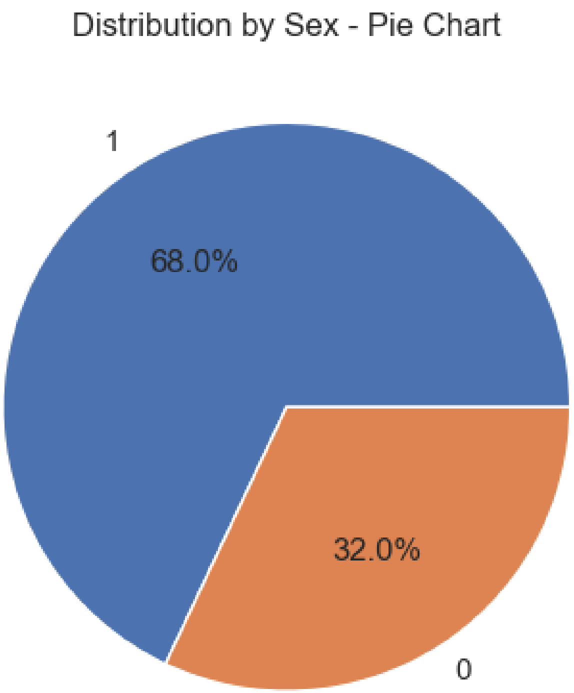
[74]:

```
#ScatterPlot using Matplotlib
plt.scatter(df['thalach'], df['age'])
plt.show()
```



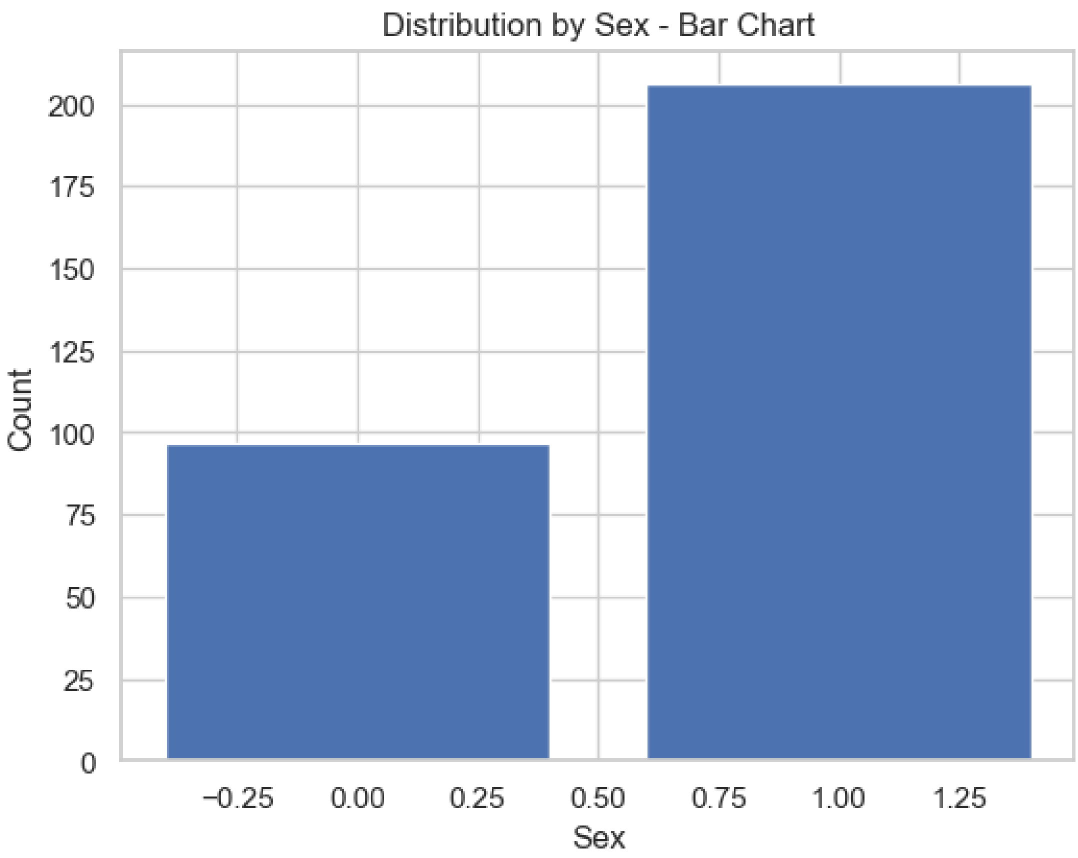
[75]:
```
# First, create the DataFrame correctly
sex_df = pd.DataFrame(df['sex'].value_counts())
sex_df.columns = ['count']  # Rename the column for clarity
```

[76]:
```
# Now plot
plt.pie(sex_df['count'], labels=sex_df.index, autopct='%1.1f%%')
plt.title("Distribution by Sex - Pie Chart")
plt.show()
```



[77]:

```
# Bar chart
plt.bar(sex_df.index, sex_df['count'])
plt.title("Distribution by Sex - Bar Chart")
plt.xlabel("Sex")
plt.ylabel("Count")
plt.show()
```



[78]:

```
# Step 1: Import libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[79]:

```
# Step 2: Load dataset
df = pd.read_csv('AirQuality.csv')  # Ensure the file is in the same directory
df['Datetime'] = pd.to_datetime(df['Date'] + ' ' + df['Time'], dayfirst=True)
```

[80]:

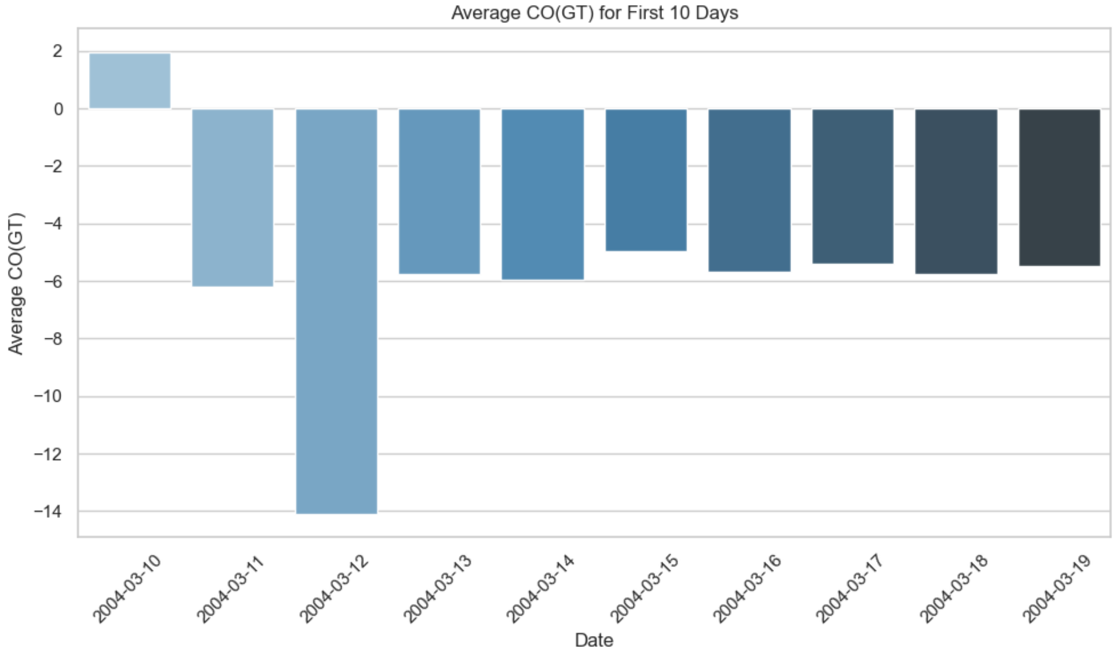
```
# Set seaborn style
sns.set(style='whitegrid')
```

[81]:

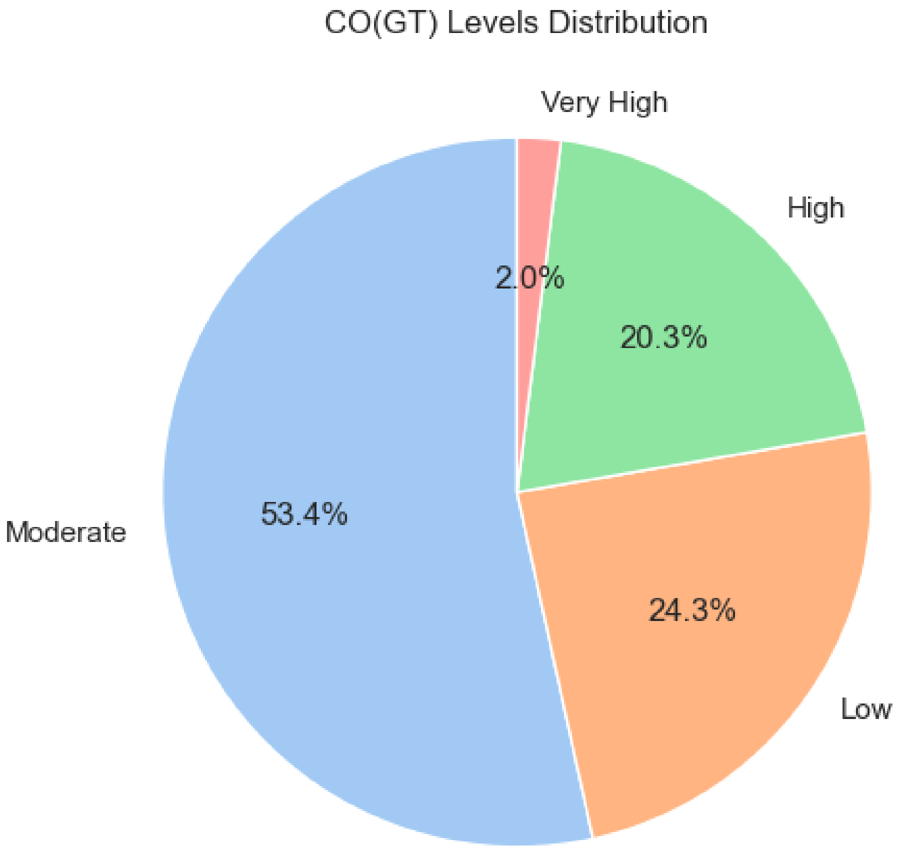
```
# ------------------------------
# BAR PLOT: Average CO per day
# ------------------------------
df['Date'] = pd.to_datetime(df['Date'], dayfirst=True)
daily_avg_co = df.groupby(df['Date'].dt.date)['CO(GT)'].mean().head(10)  #␣
 ↪First 10 days for simplicity
```

```
plt.figure(figsize=(10, 6))
sns.barplot(x=daily_avg_co.index, y=daily_avg_co.values, palette='Blues_d')
plt.xticks(rotation=45)
plt.title('Average CO(GT) for First 10 Days')
plt.xlabel('Date')
plt.ylabel('Average CO(GT)')
plt.tight_layout()
plt.show()
```

```
C:\Users\amans\AppData\Local\Temp\ipykernel_14288\905959813.py:8: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=daily_avg_co.index, y=daily_avg_co.values, palette='Blues_d')
```

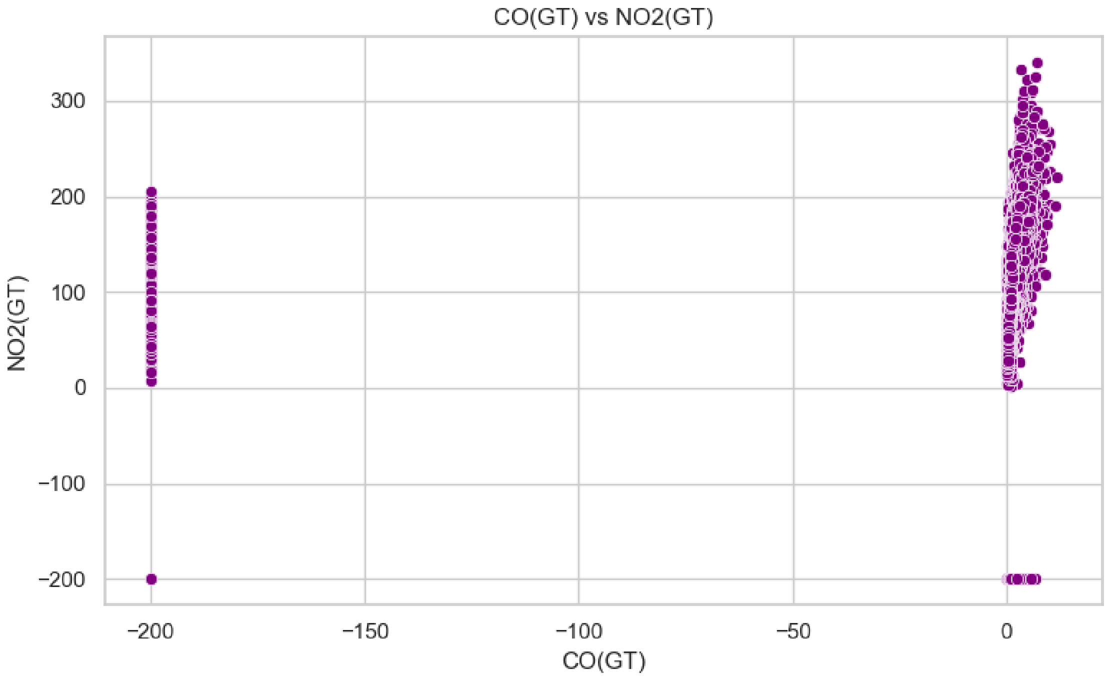


[82]:
```
# ------------------------------
# PIE PLOT: CO(GT) category proportions
# ------------------------------
co_levels = pd.cut(df['CO(GT)'], bins=[-1, 1, 3, 6, 10], labels=['Low',␣
 ↪'Moderate', 'High', 'Very High'])
co_counts = co_levels.value_counts()

plt.figure(figsize=(6, 6))
plt.pie(co_counts, labels=co_counts.index, autopct='%1.1f%%', startangle=90,␣
 ↪colors=sns.color_palette('pastel'))
plt.title('CO(GT) Levels Distribution')
plt.show()
```

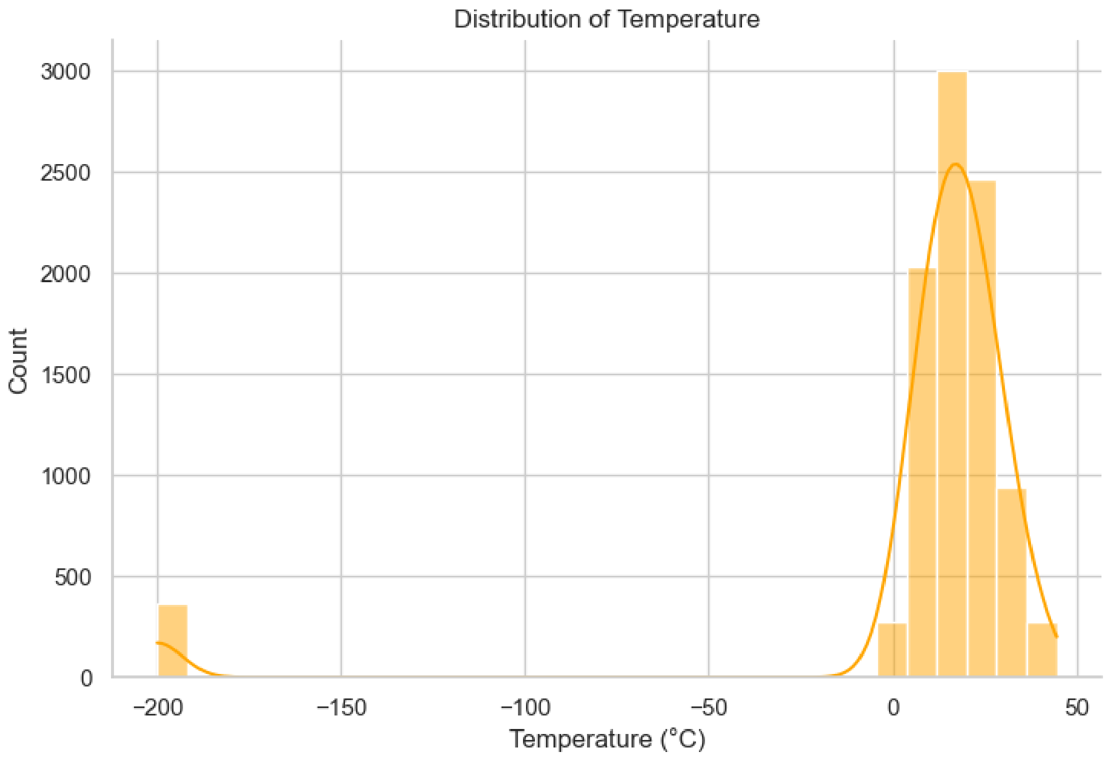


[83]:

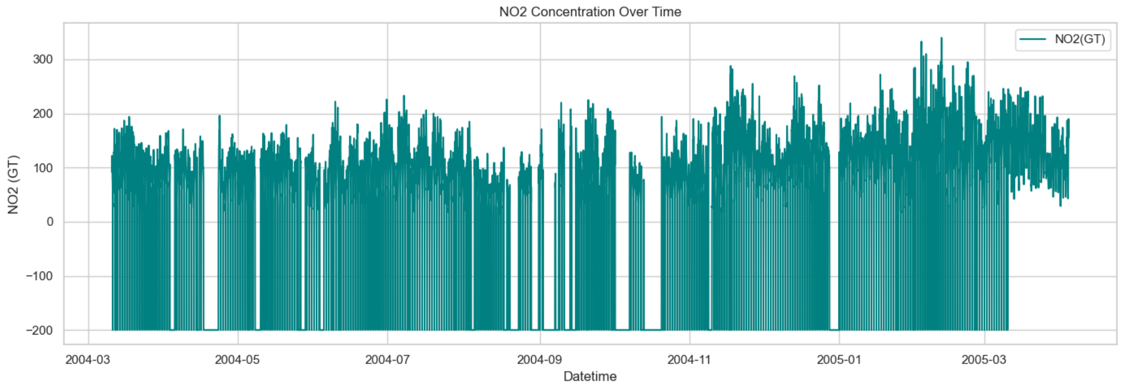
```
# ------------------------------
# SCATTER PLOT: CO(GT) vs NO2(GT)
# ------------------------------
plt.figure(figsize=(8, 5))
sns.scatterplot(data=df, x='CO(GT)', y='NO2(GT)', color='purple')
plt.title('CO(GT) vs NO2(GT)')
plt.xlabel('CO(GT)')
plt.ylabel('NO2(GT)')
plt.tight_layout()
plt.show()
```



[84]:
```
# ----------------------------
# DISPLOT: Distribution of Temperature
# ----------------------------
sns.displot(df['Temp'].dropna(), bins=30, kde=True, color='orange', height=5,␣
  ↪aspect=1.5)
plt.title('Distribution of Temperature')
plt.xlabel('Temperature (°C)')
plt.show()
```

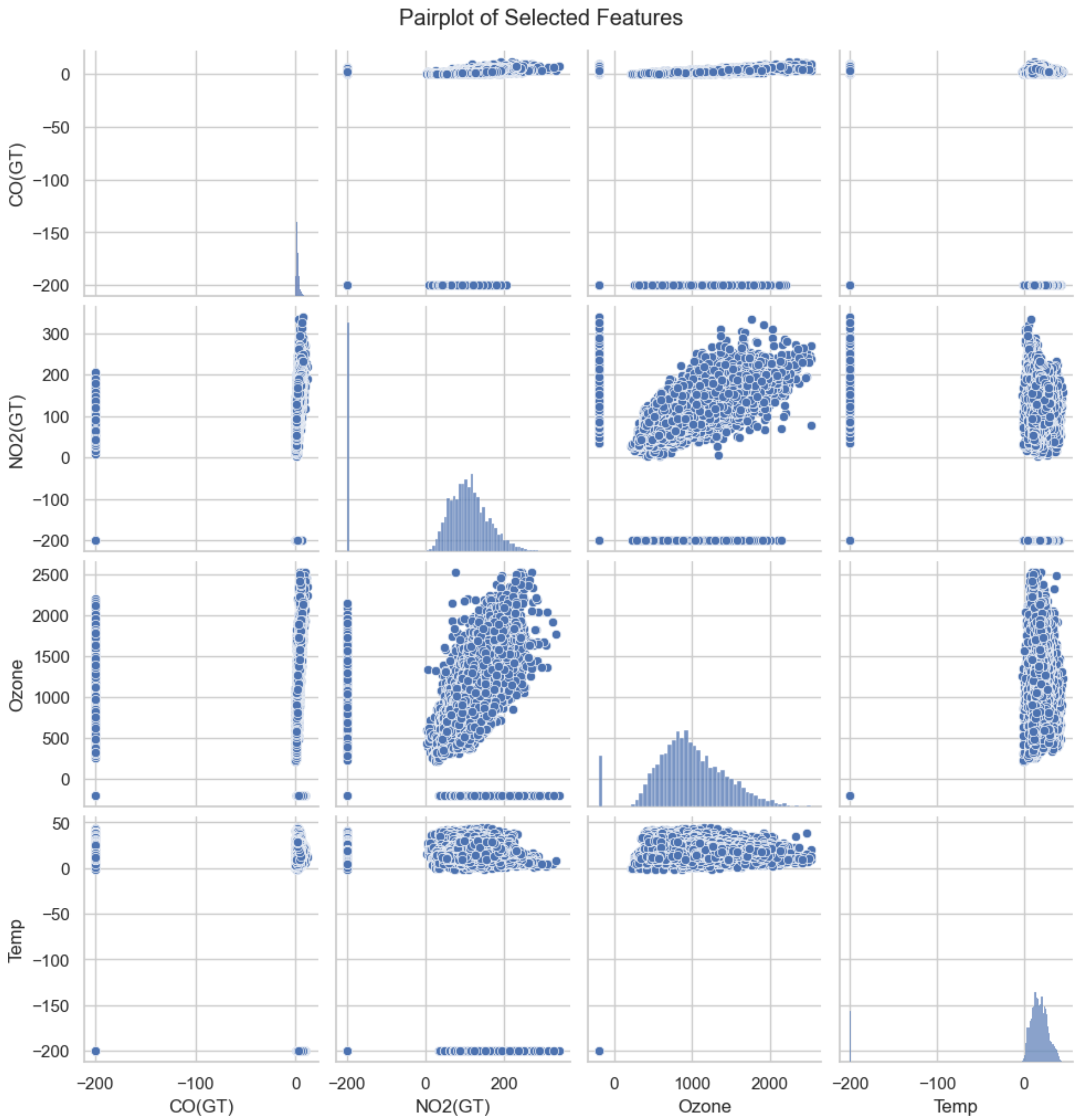


[85]:
```
# -----------------------------
# LINE PLOT: NO2(GT) Over Time
# -----------------------------
plt.figure(figsize=(14, 5))
plt.plot(df['Datetime'], df['NO2(GT)'], label='NO2(GT)', color='teal')
plt.title('NO2 Concentration Over Time')
plt.xlabel('Datetime')
plt.ylabel('NO2 (GT)')
plt.legend()
plt.tight_layout()
plt.show()
```



[86]:

```python
# Step 8: Pair Plot - Selected Features
selected_features = ['CO(GT)', 'NO2(GT)', 'Ozone', 'Temp']
sns.pairplot(df[selected_features].dropna())
plt.suptitle('Pairplot of Selected Features', y=1.02)
plt.show()
```



[ ]:

[ ]: