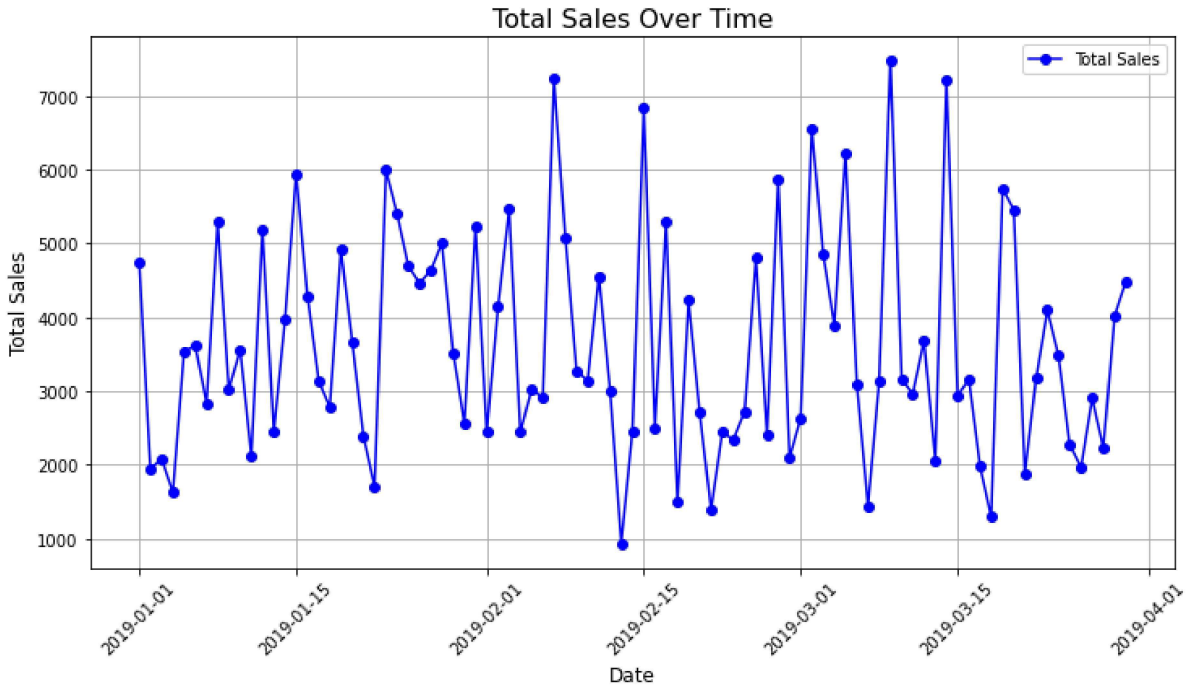
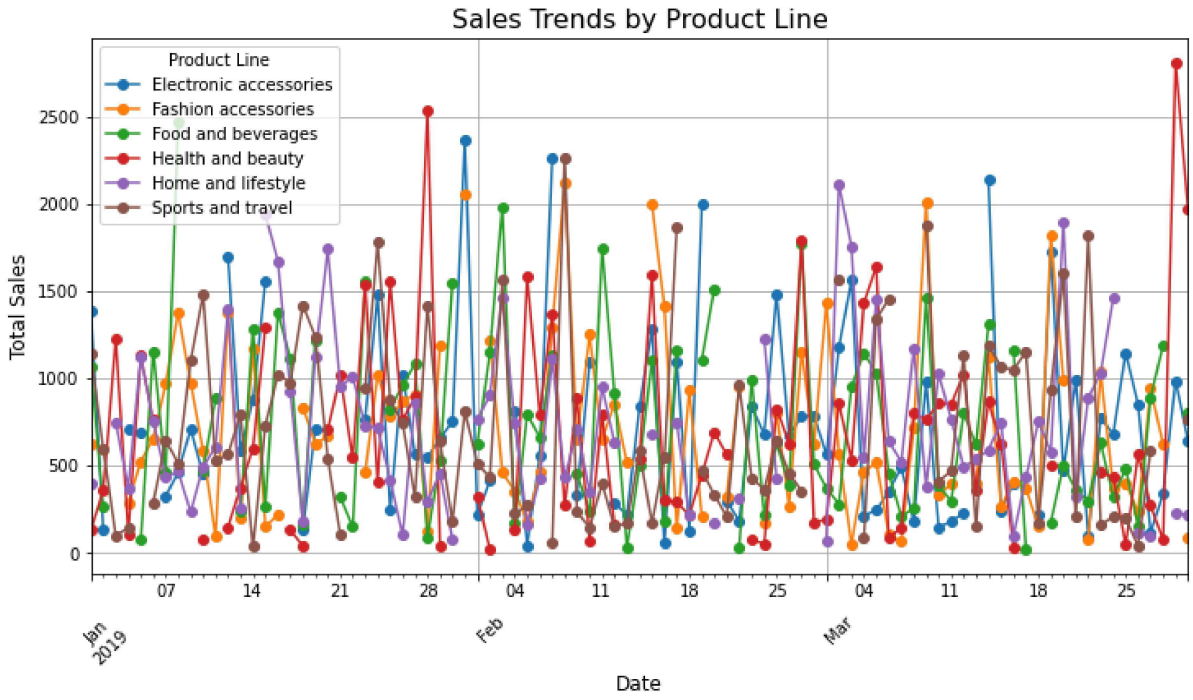
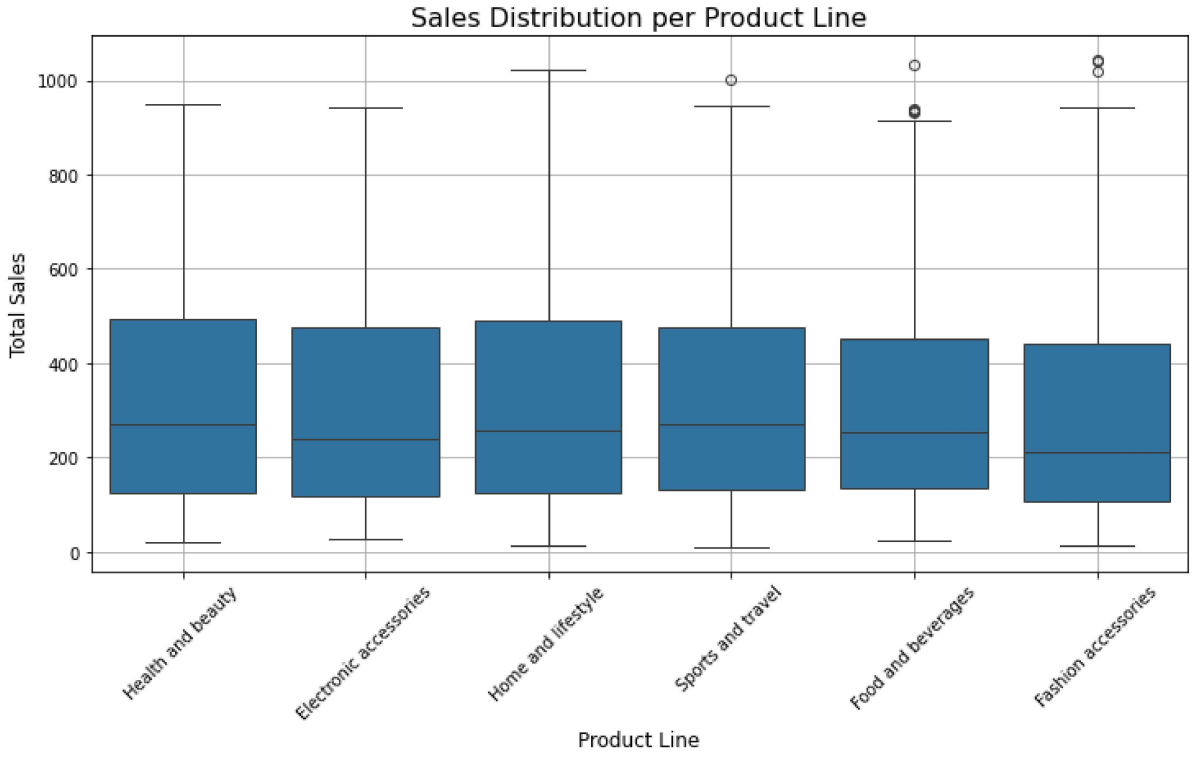
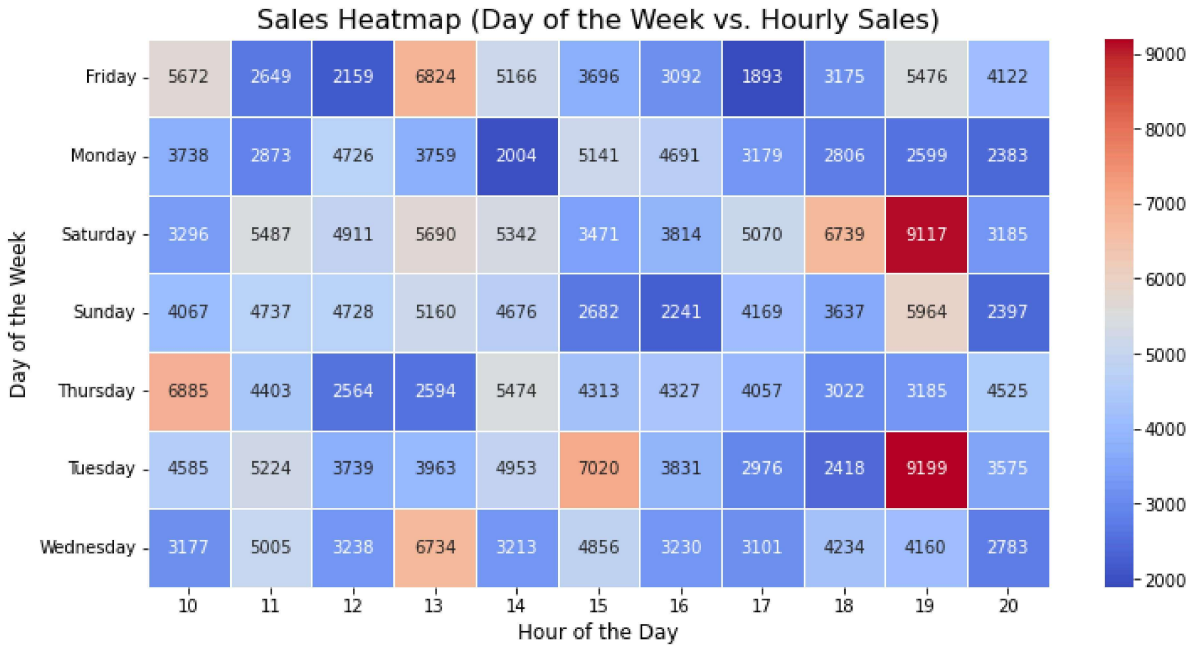
In [10]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
file_path = r"C:\Users\exam\Downloads\archive (2)\supermarket_sales - Sheet1.cs
data = pd.read_csv(file_path)

# Convert 'Date' and 'Time' to a datetime index
data['Datetime'] = pd.to_datetime(data['Date'] + ' ' + data['Time'])
data.set_index('Datetime', inplace=True)
data.drop(['Date', 'Time'], axis=1, inplace=True)

# Resample data to daily frequency
data_daily = data.resample('D').sum()

# 🎯 1️⃣ Total Sales Over Time (Line Plot)
plt.figure(figsize=(12, 6))
plt.plot(data_daily.index, data_daily['Total'], marker='o', linestyle='-', colo
plt.title('Total Sales Over Time', fontsize=16)
plt.xlabel('Date', fontsize=12)
plt.ylabel('Total Sales', fontsize=12)
plt.grid(True)
plt.legend()
plt.xticks(rotation=45)
plt.show()

# 🎯 2️⃣ Sales Trends by Product Line (Line Plot)
data_grouped = data.groupby([data.index.date, 'Product line'])['Total'].sum().u
data_grouped.index = pd.to_datetime(data_grouped.index)

data_grouped.plot(figsize=(12, 6), marker='o')
plt.title('Sales Trends by Product Line', fontsize=16)
plt.xlabel('Date', fontsize=12)
plt.ylabel('Total Sales', fontsize=12)
plt.grid(True)
plt.legend(title='Product Line')
plt.xticks(rotation=45)
plt.show()

# 🎯 3️⃣ Boxplot of Sales Distribution per Product Line
plt.figure(figsize=(12, 6))
sns.boxplot(x='Product line', y='Total', data=data)
plt.xticks(rotation=45)
plt.title('Sales Distribution per Product Line', fontsize=16)
plt.xlabel('Product Line', fontsize=12)
plt.ylabel('Total Sales', fontsize=12)
plt.grid(True)
plt.show()

# 🎯 4️⃣ Sales Heatmap (Day of the Week vs. Hourly Sales)
data['Day'] = data.index.day_name()
data['Hour'] = data.index.hour
heatmap_data = data.pivot_table(index='Day', columns='Hour', values='Total', ag

plt.figure(figsize=(12, 6))
sns.heatmap(heatmap_data, cmap="coolwarm", annot=True, fmt=".0f", linewidths=0
```

```
plt.title('Sales Heatmap (Day of the Week vs. Hourly Sales)', fontsize=16)
plt.xlabel('Hour of the Day', fontsize=12)
plt.ylabel('Day of the Week', fontsize=12)
plt.show()

# 🎯 5️⃣ Customer Traffic Per Hour (Bar Chart)
hourly_traffic = data.groupby(data.index.hour)['Invoice ID'].count()

plt.figure(figsize=(12, 6))
sns.barplot(x=hourly_traffic.index, y=hourly_traffic.values, palette="viridis")
plt.title('Customer Traffic Per Hour', fontsize=16)
plt.xlabel('Hour of the Day', fontsize=12)
plt.ylabel('Number of Transactions', fontsize=12)
plt.grid(True)
plt.show()
```

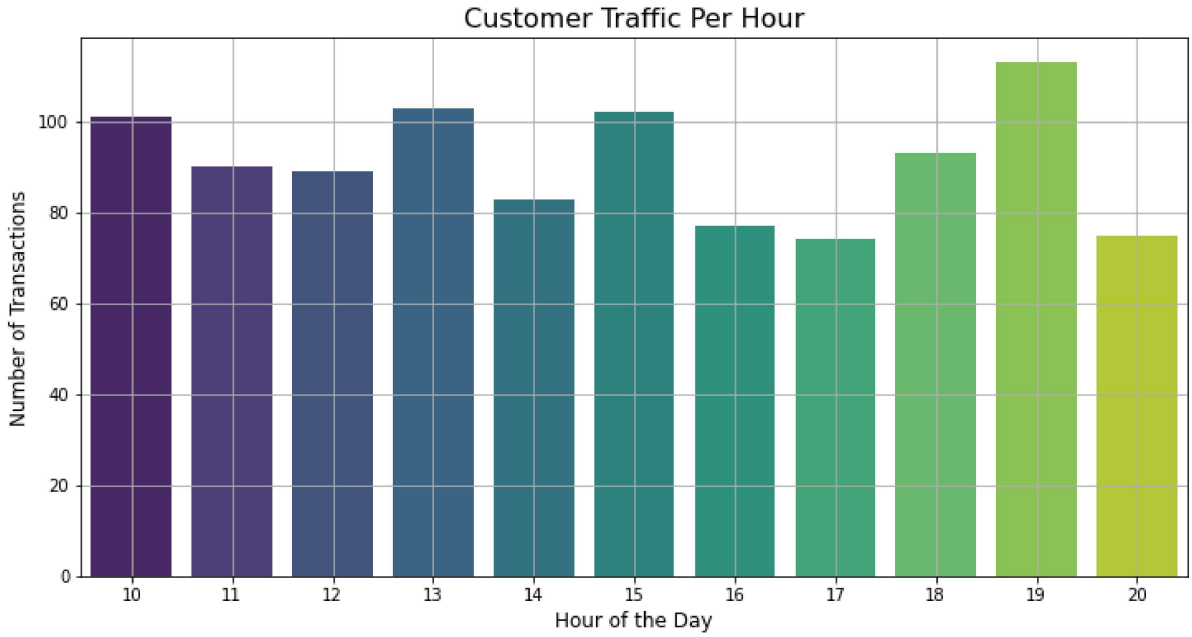



Sales Trends by Product Line
Product Line
Electronic accessories
Fashion accessories
Food and beverages
Health and beauty
Home and lifestyle
Sports and travel
Total Sales
0
500
1000
1500
2000
2500
07
14
21
28
04
11
18
25
04
11
18
25
Jan
2019
Feb
Mar
Date


Sales Distribution per Product Line
Total Sales
0
200
400
600
800
1000
Health and beauty
Electronic accessories
Home and lifestyle
Sports and travel
Food and beverages
Fashion accessories
Product Line




```
<ipython-input-10-5a6266cd064b>:67: FutureWarning:

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

  sns.barplot(x=hourly_traffic.index, y=hourly_traffic.values, palette="virid
is")
```



In [ ]: