In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style='whitegrid')
# 1. Import Libraries
```

```
C:\Users\pramo\anaconda3\Lib\site-packages\pandas\core\arrays\masked.py:60: Use
rWarning: Pandas requires version '1.3.6' or newer of 'bottleneck' (version '1.
3.5' currently installed).
  from pandas.core import (
```

In [2]:

```
# 2. Load Data
df = pd.read_csv("supermarket_sales - Sheet1.csv")
df.head()
```

Out[2]:

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7	26.1415	548.9715
1	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	3.8200	80.2200
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7	16.2155	340.5255
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8	23.2880	489.0480
4	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7	30.2085	634.3785

In [3]:

```
# 3. Dataset Overview
df.info()
df.describe()
```

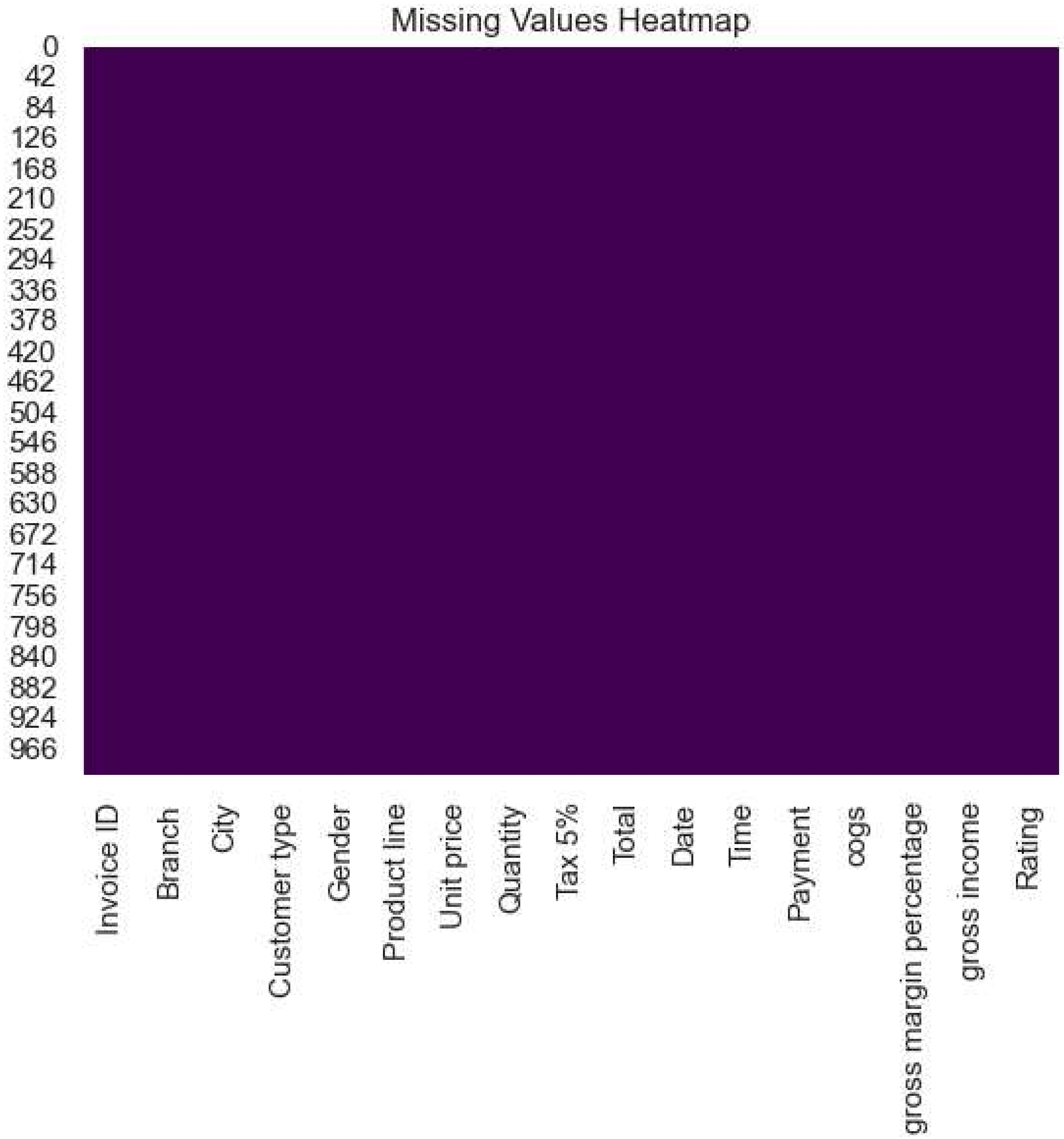
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 17 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Invoice ID               1000 non-null   object 
 1   Branch                   1000 non-null   object 
 2   City                     1000 non-null   object 
 3   Customer type            1000 non-null   object 
 4   Gender                   1000 non-null   object 
 5   Product line             1000 non-null   object 
 6   Unit price               1000 non-null   float64
 7   Quantity                 1000 non-null   int64  
 8   Tax 5%                   1000 non-null   float64
 9   Total                    1000 non-null   float64
 10  Date                     1000 non-null   object 
 11  Time                     1000 non-null   object 
 12  Payment                  1000 non-null   object 
 13  cogs                     1000 non-null   float64
 14  gross margin percentage  1000 non-null   float64
 15  gross income             1000 non-null   float64
 16  Rating                   1000 non-null   float64
dtypes: float64(7), int64(1), object(9)
memory usage: 132.9+ KB
```

Out[3]:

	Unit price	Quantity	Tax 5%	Total	cogs	gross margin percentage	gross income
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.00000	1000.000000	1000.000000
mean	55.672130	5.510000	15.379369	322.966749	307.58738	4.761905	15.379369
std	26.494628	2.923431	11.708825	245.885335	234.17651	0.000000	11.708825
min	10.080000	1.000000	0.508500	10.678500	10.17000	4.761905	0.508500
25%	32.875000	3.000000	5.924875	124.422375	118.49750	4.761905	5.924875
50%	55.230000	5.000000	12.088000	253.848000	241.76000	4.761905	12.088000
75%	77.935000	8.000000	22.445250	471.350250	448.90500	4.761905	22.445250
max	99.960000	10.000000	49.650000	1042.650000	993.00000	4.761905	49.650000

In [4]:

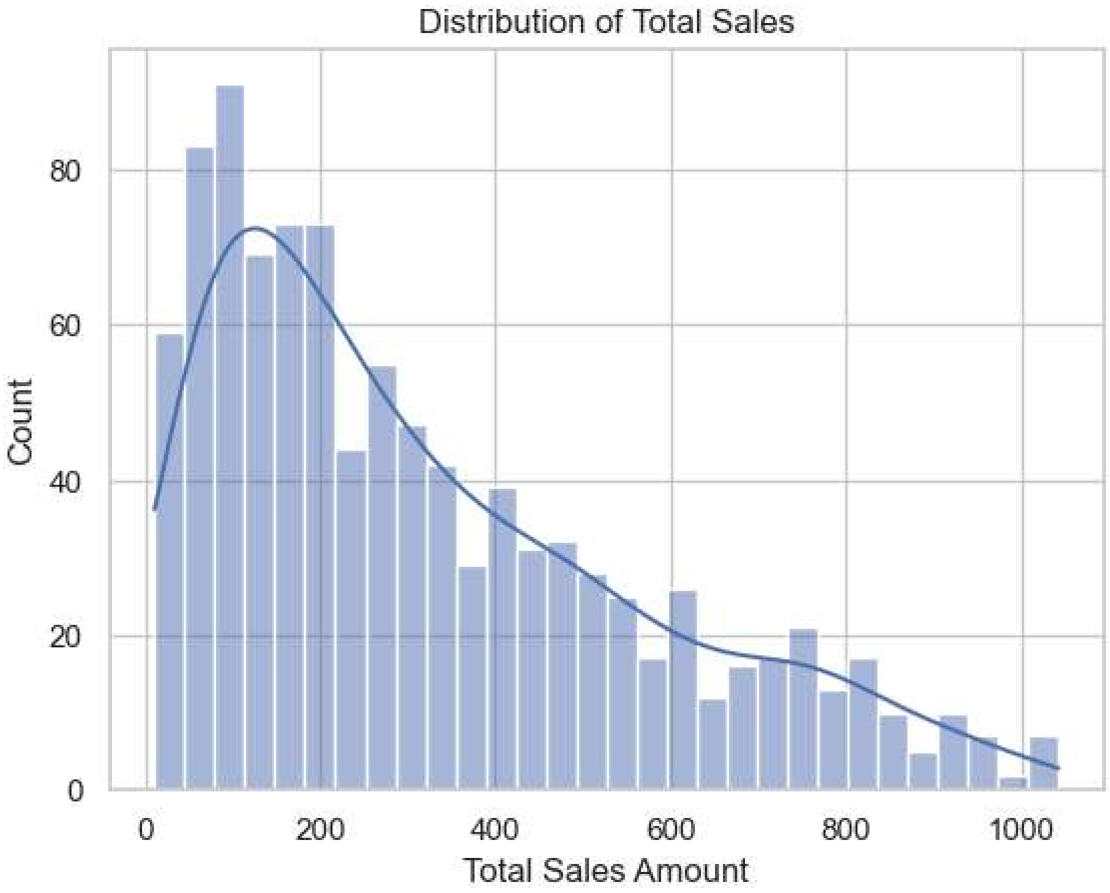
```
# 4. Missing Value Check
sns.heatmap(df.isnull(), cbar=False, cmap='viridis')
plt.title("Missing Values Heatmap")
plt.show()
```



In [5]:

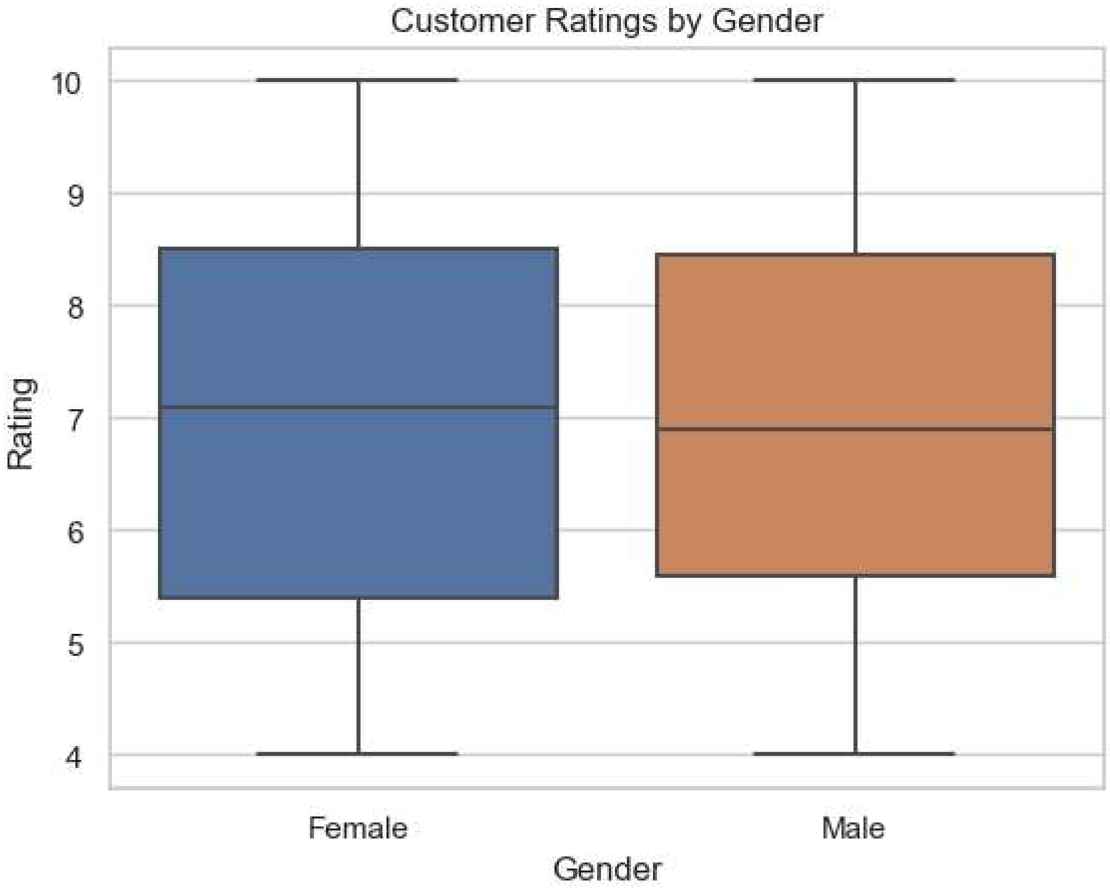
```
# 5. Distribution of Total Sales
sns.histplot(df["Total"], bins=30, kde=True)
plt.title("Distribution of Total Sales")
plt.xlabel("Total Sales Amount")
plt.show()
```

```
C:\Users\pramo\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarn
ing: use_inf_as_na option is deprecated and will be removed in a future versio
n. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

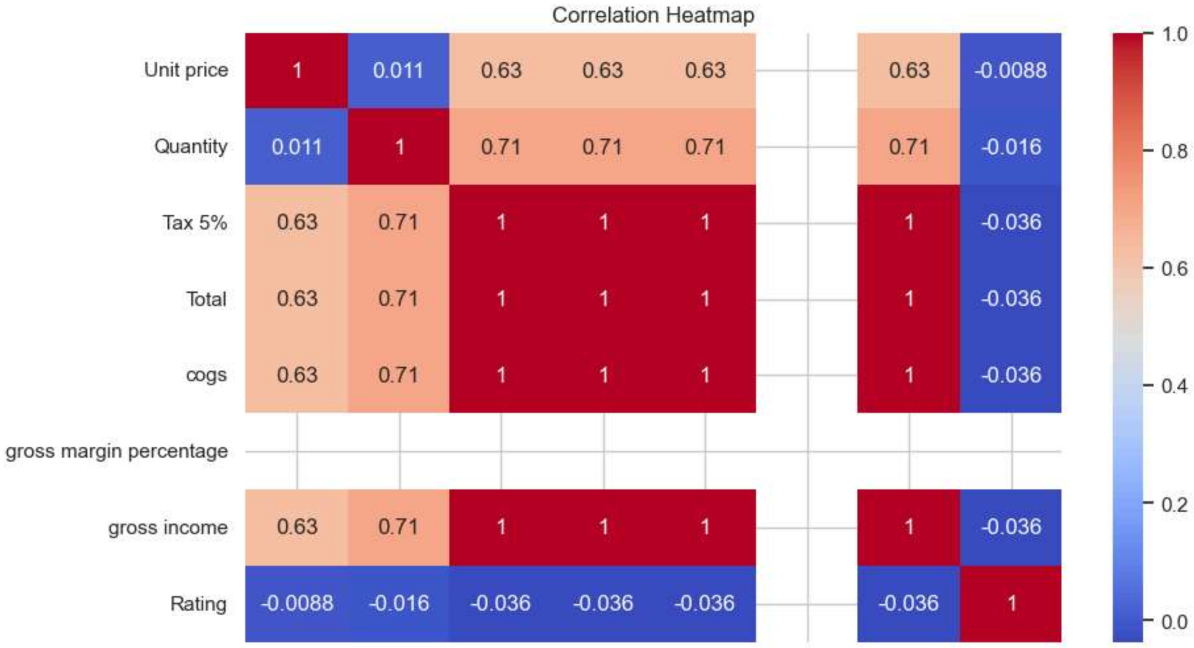


In [6]:

```
# 6. Boxplot: Rating by Gender
sns.boxplot(x='Gender', y='Rating', data=df)
plt.title("Customer Ratings by Gender")
plt.show()
```

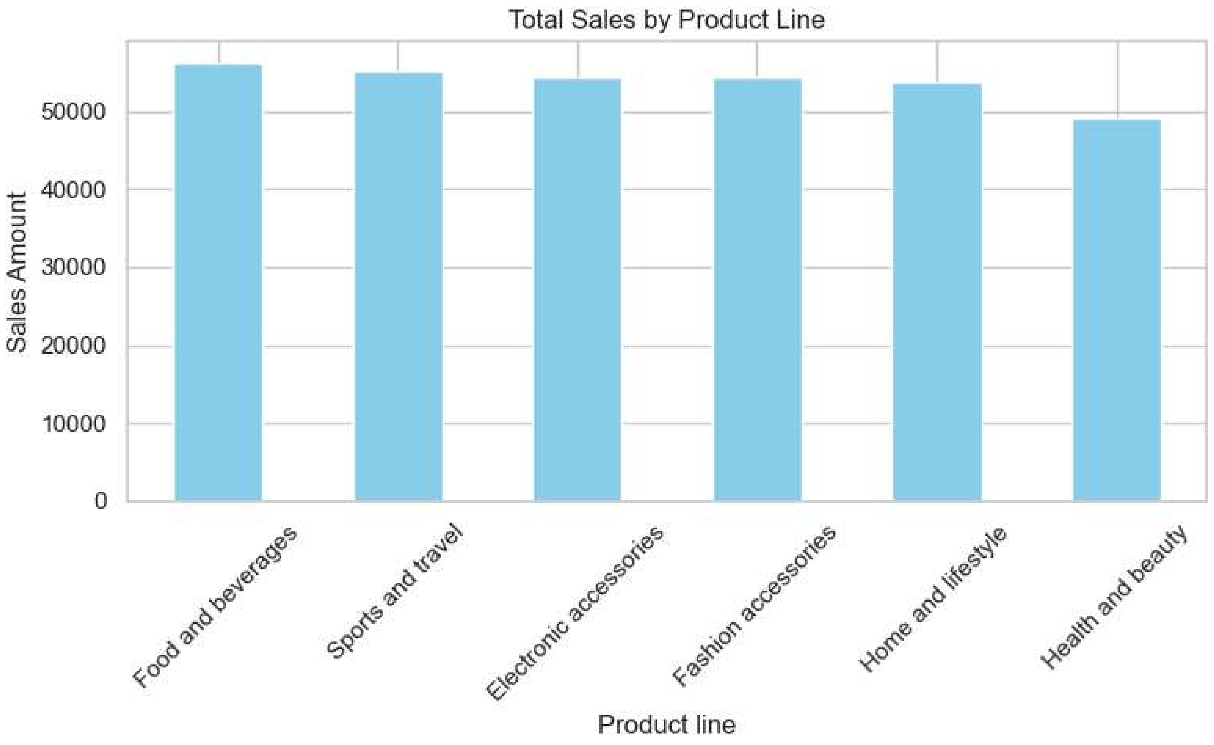


In [7]:
```
# 7. Correlation Heatmap (Fix: Use only numeric columns)
numeric_df = df.select_dtypes(include='number')  # Filters only numeric columns
plt.figure(figsize=(10,6))
sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm')
plt.title("Correlation Heatmap")
plt.show()
```



In [8]:

```
# 8. Total Sales by Product Line
product_sales = df.groupby("Product line")["Total"].sum().sort_values(ascending=
product_sales.plot(kind="bar", color="skyblue", figsize=(8,5))
plt.title("Total Sales by Product Line")
plt.ylabel("Sales Amount")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



In [ ]: