```
# Import required libraries
import pandas as pd
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset into a DataFrame
df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/retail_sales_dataset.csv')

df.head()
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatic
  and should_run_async(code)
```

	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount
0	1	2023-11-24	CUST001	Male	34	Beauty	3	50	150
1	2	2023-02-27	CUST002	Female	26	Clothing	2	500	1000
2	3	2023-01-13	CUST003	Male	50	Electronics	1	30	30
3	4	2023-05-21	CUST004	Male	37	Clothing	1	500	500
4	5	2023-05-06	CUST005	Male	30	Beauty	2	50	100

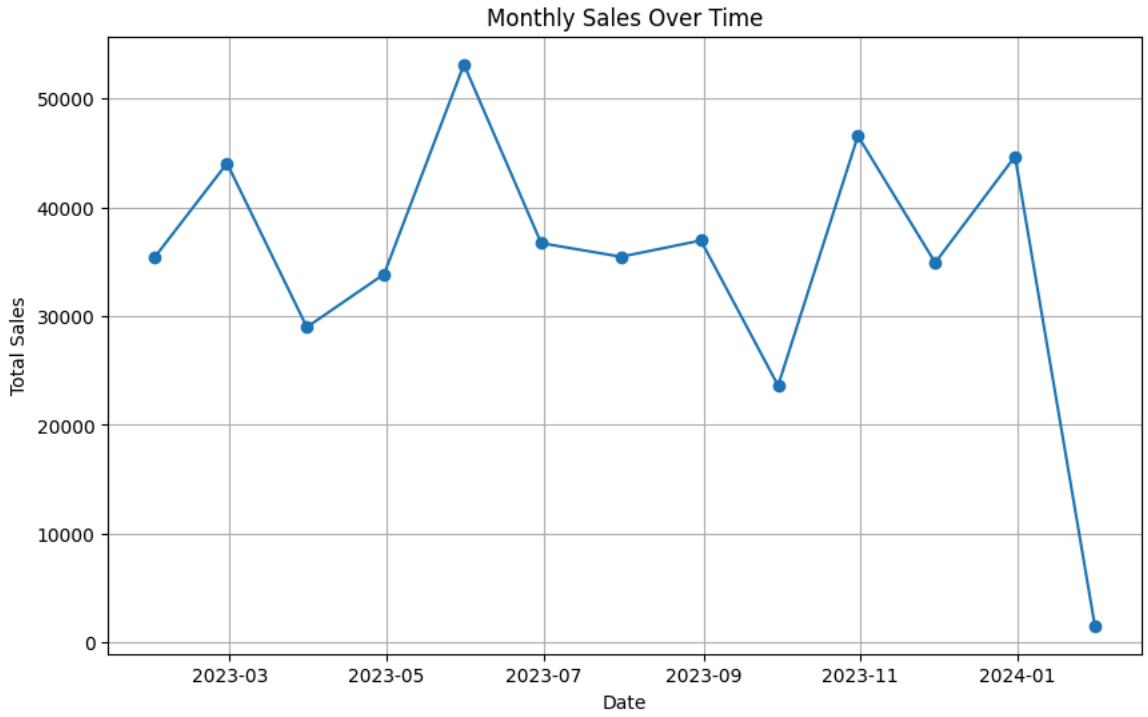
```
df.head(5)
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatic
  and should_run_async(code)
```

	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount
0	1	2023-11-24	CUST001	Male	34	Beauty	3	50	150
1	2	2023-02-27	CUST002	Female	26	Clothing	2	500	1000
2	3	2023-01-13	CUST003	Male	50	Electronics	1	30	30
3	4	2023-05-21	CUST004	Male	37	Clothing	1	500	500
4	5	2023-05-06	CUST005	Male	30	Beauty	2	50	100

```
df['Date'] = pd.to_datetime(df['Date'])
df.set_index('Date', inplace=True)
monthly_sales = df['Total Amount'].resample('M').sum()
plt.figure(figsize=(10, 6))
plt.plot(monthly_sales.index, monthly_sales.values, marker='o')
plt.title('Monthly Sales Over Time')
plt.xlabel('Date')
plt.ylabel('Total Sales')
plt.grid(True)
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatic
  and should_run_async(code)
<ipython-input-89-0ea91bbe2467>:3: FutureWarning: 'M' is deprecated and will be removed in a future version, please use 'ME' instead.
  monthly_sales = df['Total Amount'].resample('M').sum()
```

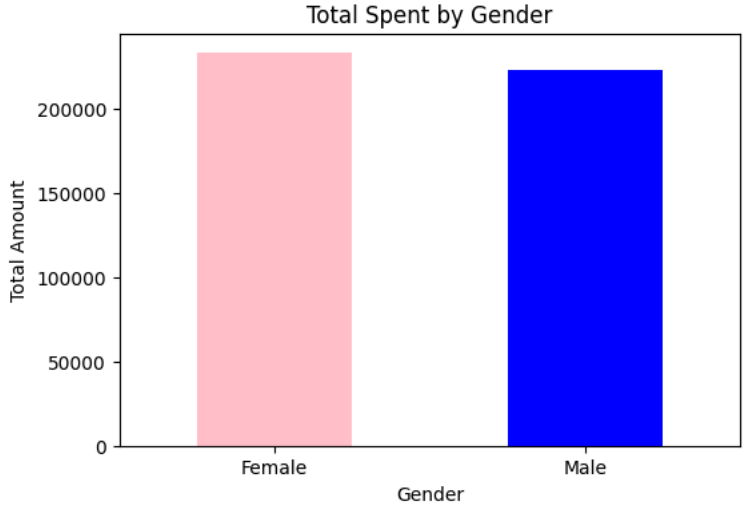



```
gender_consumption = df.groupby('Gender')['Total Amount'].sum()
print(gender_consumption)
```

```
Gender
Female    232840
Male      223160
Name: Total Amount, dtype: int64
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatic
  and should_run_async(code)
```

```
gender_consumption.plot(kind='bar', color=['pink', 'blue'], figsize=(6, 4))
plt.title('Total Spent by Gender')
plt.xlabel('Gender')
plt.ylabel('Total Amount')
plt.xticks(rotation=0)
plt.show()
```

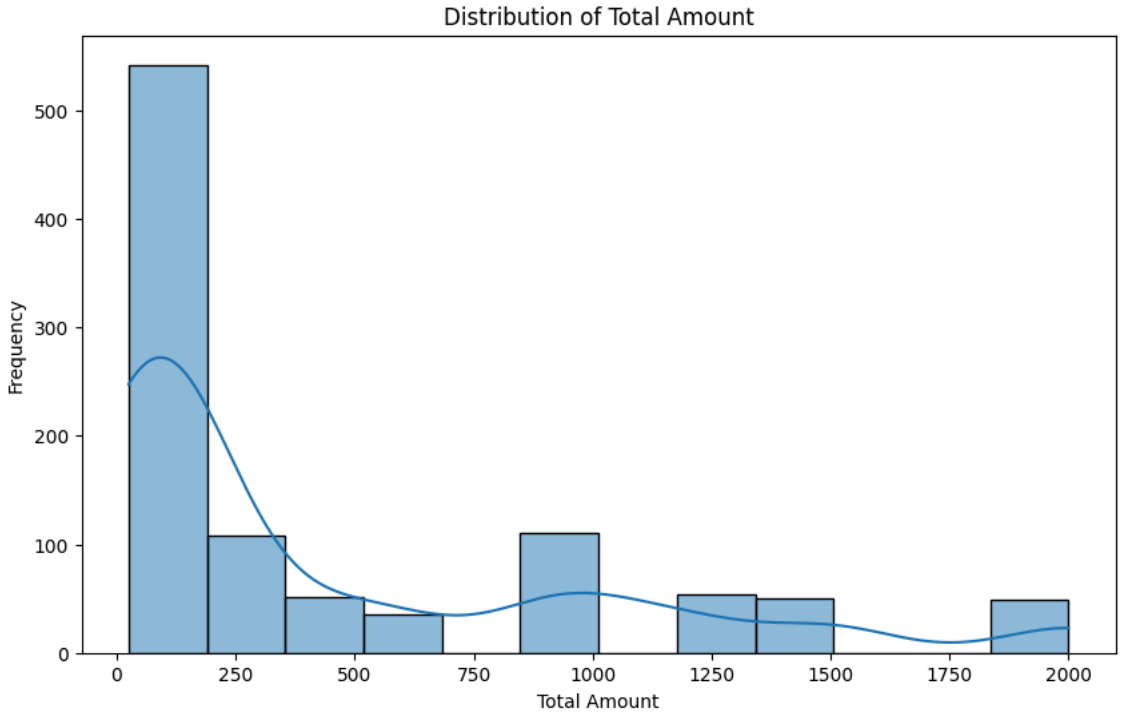
```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatic
  and should_run_async(code)
```




```
plt.figure(figsize=(10, 6))
sns.histplot(df['Total Amount'], kde=True)
plt.title('Distribution of Total Amount')
plt.xlabel('Total Amount')
```

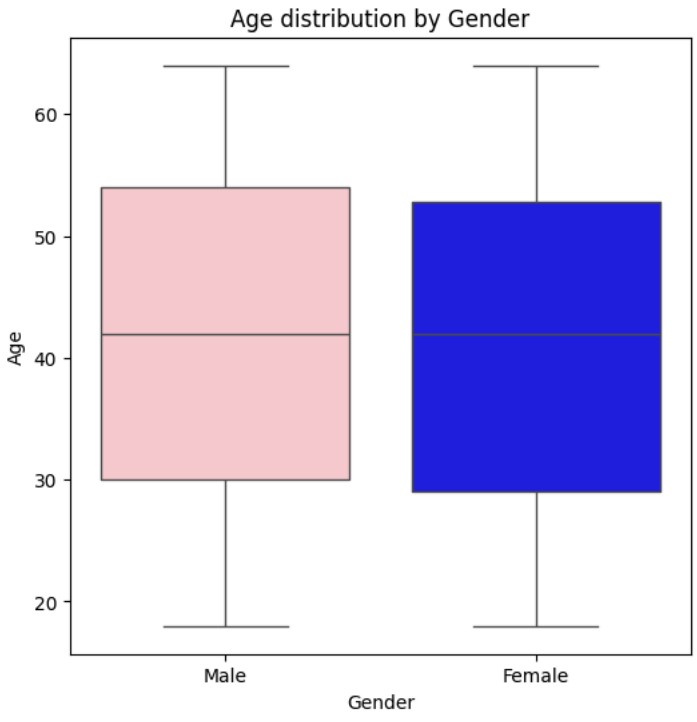
```
plt.ylabel('Frequency')
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatic
  and should_run_async(code)
```




```
plt.figure(figsize=(6, 6))
sns.boxplot( x='Gender', y='Age',data=df, palette=['pink', 'blue'])
plt.title('Age distribution by Gender')
plt.xlabel('Gender')
plt.ylabel('Age')
plt.show()
```

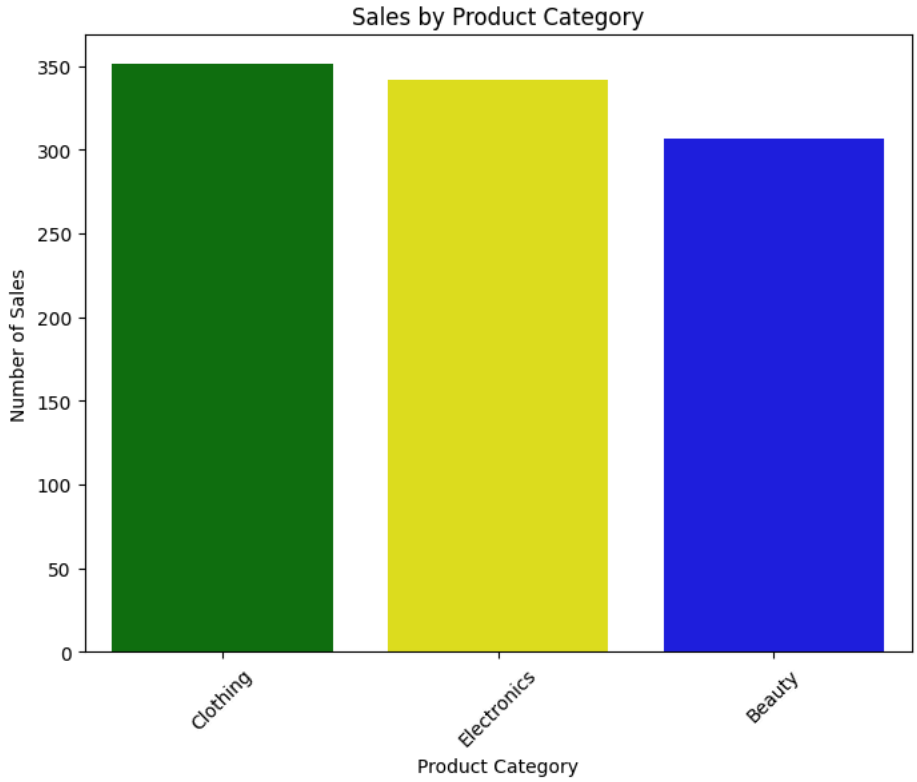
```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatic
  and should_run_async(code)
<ipython-input-105-cccc6f4b5752>:2: FutureWarning:

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` fo

  sns.boxplot( x='Gender', y='Age',data=df, palette=['pink', 'blue'])
```




```
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='Product Category', order=df['Product Category'].value_counts().index , palette=['green', 'yellow', 'blue'])
plt.title('Sales by Product Category')
plt.xlabel('Product Category')
```

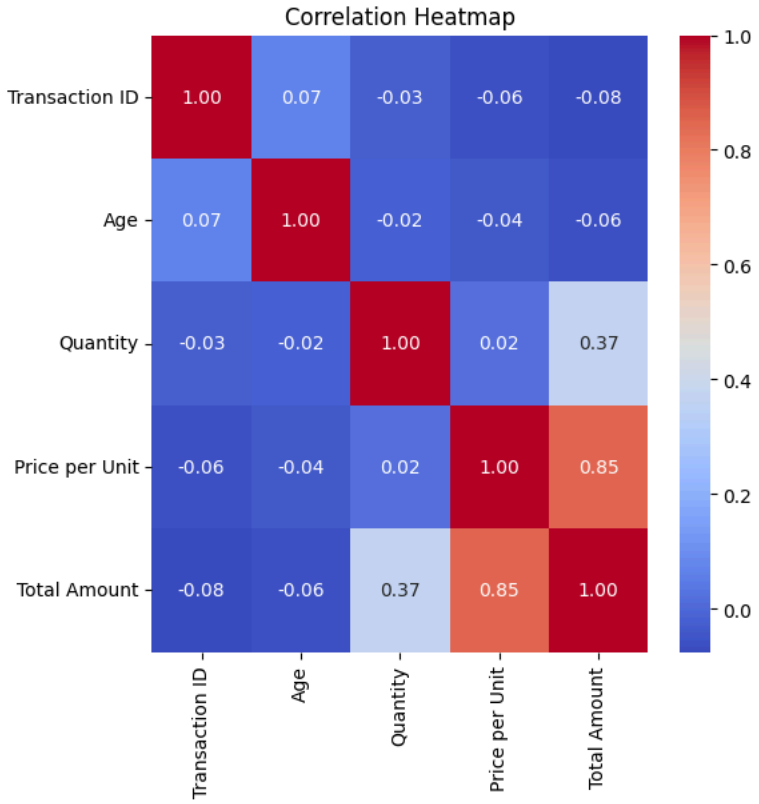
```
plt.ylabel('Number of Sales')
plt.xticks(rotation=45)
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatic
  and should_run_async(code)
<ipython-input-107-c479d984038c>:2: 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` fo

  sns.countplot(data=df, x='Product Category', order=df['Product Category'].value_counts().index , palette=['green', 'yellow', 'blue'])
```




```
numeric_df = df.select_dtypes(include=[np.number])
plt.figure(figsize=(6, 6))
sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Heatmap')
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatic
  and should_run_async(code)
```




```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
X = df[['Age', 'Quantity', 'Price per Unit']]
y = df['Total Amount']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatic
  and should_run_async(code)
```

```
model = LinearRegression()
model.fit(X_train, y_train)
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatic
  and should_run_async(code)
```

LinearRegression

```
LinearRegression()
```

```
y_pred = model.predict(X_test)
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatic
  and should_run_async(code)
```

```
# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f'Mean Squared Error: {mse}')
print(f'R-squared: {r2}')
```

```
Mean Squared Error: 41896.21322134358
R-squared: 0.8568772264250432
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatic
  and should_run_async(code)
```

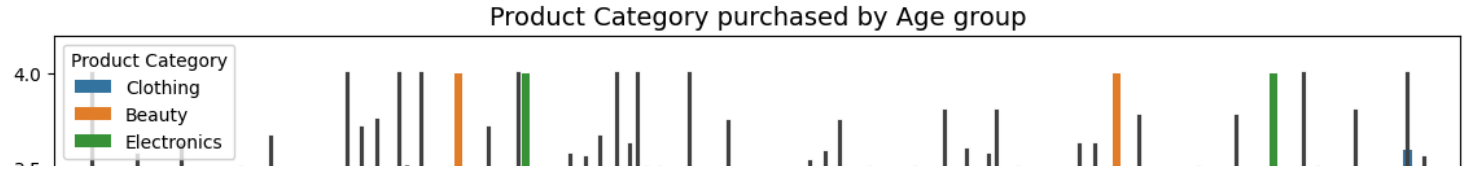
```
plt.figure(figsize=(12,8))
sns.swarmplot(x='Product Category', y='Age', hue='Gender', data=df, palette='Set2')
plt.title('Distribution of Age and Gender across Product Category', fontsize=16)
plt.xlabel('Product Category', fontsize=14)
```

```
plt.ylabel('Age', fontsize=14)
plt.legend(title='Gender', loc ='upper right', fontsize=10)
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatic
  and should_run_async(code)
```


```
# bar plot
plt.figure(figsize=(14,8))
sns.barplot(x='Age', y='Quantity', hue= 'Product Category', data=df)
plt.title('Product Category purchased by Age group', fontsize=14)
plt.xlabel('Age Group', fontsize=10)
plt.ylabel('Quantity Purchased', fontsize=10)
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatic
  and should_run_async(code)
```




```
# Product Preferences by Gender
gender_product_sales = df.groupby(['Gender', 'Product Category'])['Total Amount'].sum().unstack()

plt.figure(figsize=(8, 8))
sns.heatmap(gender_product_sales, annot=True, fmt=".1f", cmap="YlGnBu", linewidths=.5)
plt.title('Product Preferences by Gender')
plt.xlabel('Product Category')
plt.ylabel('Gender')
plt.xticks(rotation=45)
```