```
# Import Libraries
import pandas as pd

# Loading the dataset
df = pd.read_csv("walmart_data_cleansed.csv")

# Preview the data
df.head()
```

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax_5	
0	750-67-8428	Alex	Yangon	Member	Female	Health and beauty	74.69	7	26.1415	548
1	226-31-3081	Giza	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	3.8200	80
2	631-41-3108	Alex	Yangon	Normal	Female	Home and lifestyle	46.33	7	16.2155	340
3	123-19-1176	Alex	Yangon	Member	Female	Health and beauty	58.22	8	23.2880	489
4	373-73-7910	Alex	Yangon	Member	Female	Sports and travel	86.31	7	30.2085	634

Next steps: Generate code with df View recommended plots New interactive sheet

```
# Standardizing column names
df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_')

# Converting 'date' column to datetime
df['date'] = pd.to_datetime(df['date'])

# Confirming the column types
df.dtypes
```

	0
invoice_id	object
branch	object
city	object
customer_type	object
gender	object
product_line	object
unit_price	float64
quantity	int64
tax_5	float64
sales	float64
date	datetime64[ns]
time	object
payment	object
cogs	float64
gross_margin_percentage	float64
gross_income	float64
rating	float64

dtype: object

```
# Basic summary statistics
df.describe()
```

	unit_price	quantity	tax_5	sales	date	cogs	gro
count	1000.000000	1000.000000	1000.000000	1000.000000	1000	1000.00000	
mean	55.672130	5.510000	15.379369	322.966749	2019-02-14 00:05:45.600000	307.58738	
min	10.080000	1.000000	0.508500	10.678500	2019-01-01 00:00:00	10.17000	
25%	32.875000	3.000000	5.924875	124.422375	2019-01-24 00:00:00	118.49750	
50%	55.230000	5.000000	12.088000	253.848000	2019-02-13 00:00:00	241.76000	
75%	77.935000	8.000000	22.445250	471.350250	2019-03-08 00:00:00	448.90500	
max	99.960000	10.000000	49.650000	1042.650000	2019-03-30 00:00:00	993.00000	
std	26.494628	2.923431	11.708825	245.885335	NaN	234.17651	

```
# Quick facts about the dataset
print("🧾 Total Rows:", df.shape[0])
print("🧾 Total Columns:", df.shape[1])
print("\n📍 Unique Branches:", df['branch'].nunique())
print("📍 Cities:", df['city'].unique())
print("📍 Product Lines:", df['product_line'].unique())
print("📍 Payment Methods:", df['payment'].unique())
```

```
🧾 Total Rows: 1000
🧾 Total Columns: 17

📍 Unique Branches: 3
📍 Cities: ['Yangon' 'Naypyitaw' 'Mandalay']
📍 Product Lines: ['Health and beauty' 'Electronic accessories' 'Home and lifestyle'
 'Sports and travel' 'Food and beverages' 'Fashion accessories']
📍 Payment Methods: ['Ewallet' 'Cash' 'Credit card']
```

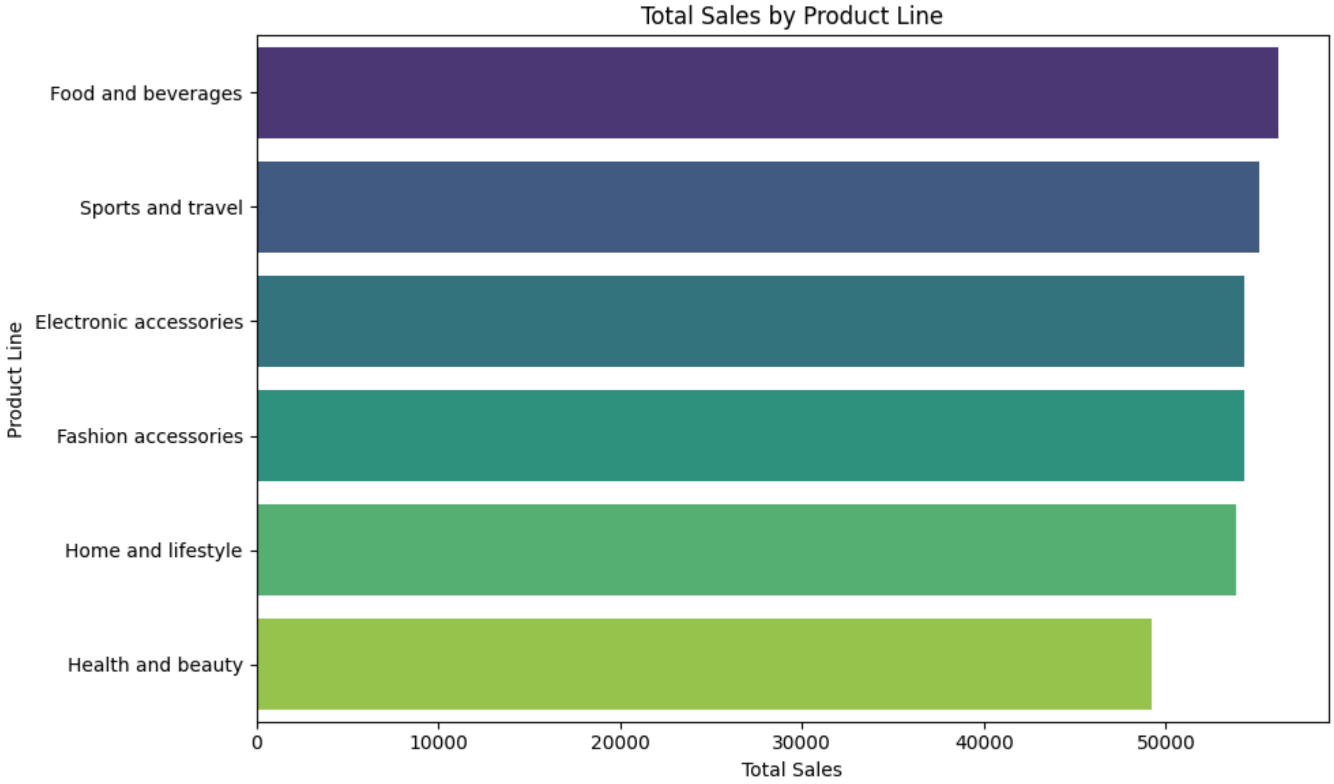
```
# Total sales by product line
sales_by_category = df.groupby('product_line')['sales'].sum().sort_values(ascending=False)

# Display result
sales_by_category
```

	sales
product_line	
Food and beverages	56144.8440
Sports and travel	55122.8265
Electronic accessories	54337.5315
Fashion accessories	54305.8950
Home and lifestyle	53861.9130
Health and beauty	49193.7390

dtype: float64

```
import seaborn as sns
import matplotlib.pyplot as plt

# Plot bar chart
plt.figure(figsize=(10,6))
sns.barplot(x=sales_by_category.values, y=sales_by_category.index, palette="viridis")
plt.title('Total Sales by Product Line')
plt.xlabel('Total Sales')
plt.ylabel('Product Line')
plt.tight_layout()
plt.show()
```

```
/tmp/ipython-input-8-3688651746.py:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.

  sns.barplot(x=sales_by_category.values, y=sales_by_category.index, palette="viridis")
```

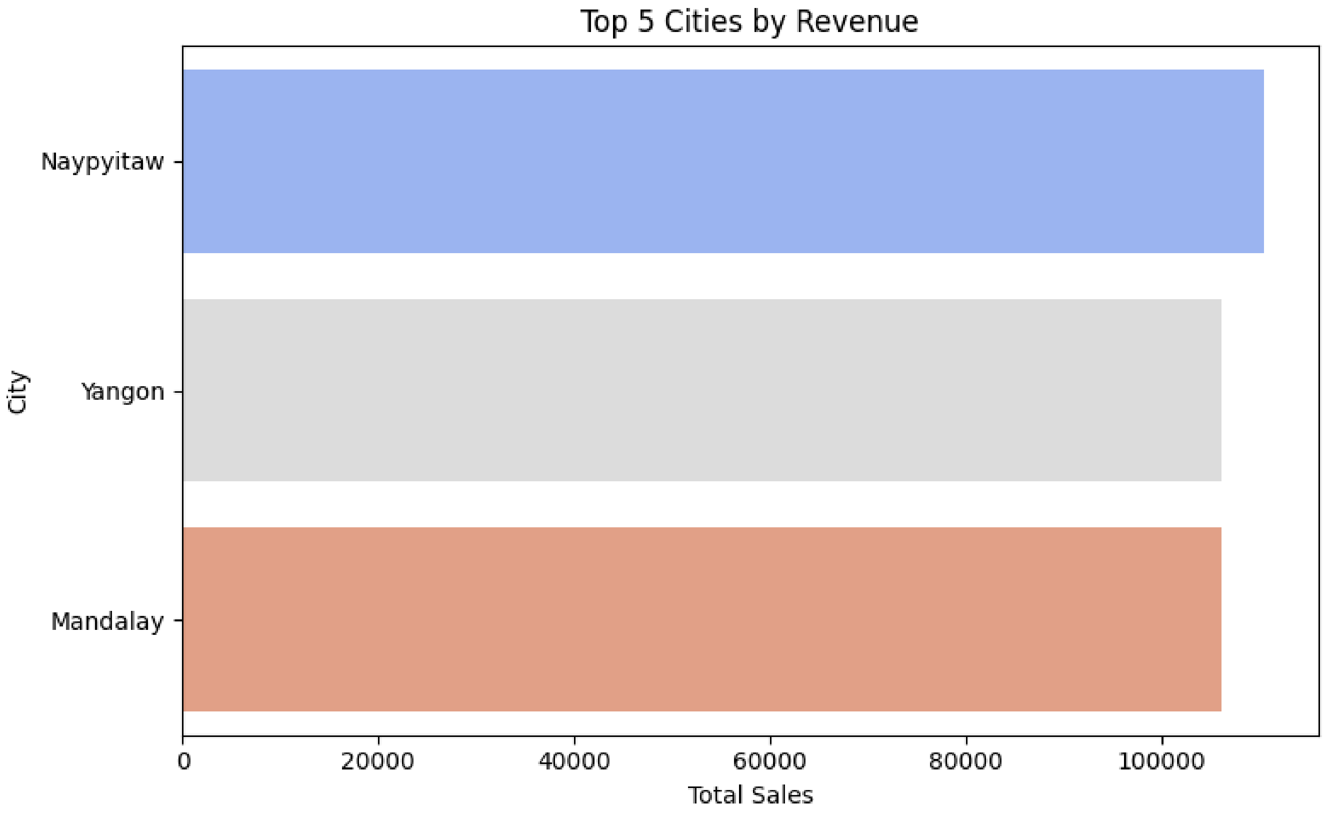


```
# Top 5 cities by revenue
city_sales = df.groupby('city')['sales'].sum().sort_values(ascending=False).head(5)

# Display result
city_sales
```

```
                    sales
         city
Naypyitaw     110568.7065
  Yangon      106200.3705
 Mandalay     106197.6720

dtype: float64
```

```
# Plotting bar chart for top 5 cities
plt.figure(figsize=(8,5))
sns.barplot(x=city_sales.values, y=city_sales.index, palette="coolwarm")
plt.title('Top 5 Cities by Revenue')
plt.xlabel('Total Sales')
plt.ylabel('City')
plt.tight_layout()
plt.show()
```

```
/tmp/ipython-input-10-2425129531.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.

  sns.barplot(x=city_sales.values, y=city_sales.index, palette="coolwarm")
```



```
# Count of payment methods
payment_counts = df['payment'].value_counts()

# Show result
payment_counts
```

```
              count
    payment
    Ewallet     345
       Cash     344
Credit card     311
```

```
# Plotting pie chart
```