Zevora Jewellery Store Data AnAnalysis

[6]:
```
# Import Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
pd.set_option('display.max_columns', None)
sns.set(style="whitegrid")
```

Load Dataset

[33]:
```
sales = pd.read_excel("D:\\Shreem Old\\New folder\\Zevota\\Jewellery_Dashboard_Dataset_Final.xlsx", sheet_name="Sales")
inventory = pd.read_excel("D:\\Shreem Old\\New folder\\Zevota\\Jewellery_Dashboard_Dataset_Final.xlsx", sheet_name="Inventory")
customers = pd.read_excel("D:\\Shreem Old\\New folder\\Zevota\\Jewellery_Dashboard_Dataset_Final.xlsx", sheet_name="Customers")
staff = pd.read_excel("D:\\Shreem Old\\New folder\\Zevota\\Jewellery_Dashboard_Dataset_Final.xlsx", sheet_name="Staff")
```

Data Cleaning

[34]:
```
sales.head(5)
```

[34]:

	OrderID	OrderDate	CustomerID	ProductID	StaffID	Region	Category	Quantity	UnitPrice	TotalAmount	Profit	IsRepeat
0	ORD1000	2024-06-08	CUST0050	PROD018	JEW003	Mumbai	Bracelet	1	30992.44	30992.44	11365.10	False
1	ORD1001	2024-04-27	CUST0026	PROD015	JEW006	Rajkot	Bracelet	4	23158.07	92632.28	17481.96	True
2	ORD1002	2024-03-18	CUST0030	PROD026	JEW004	Rajkot	Necklace	5	57687.09	288435.45	69915.80	False
3	ORD1003	2024-02-18	CUST0046	PROD021	JEW003	Rajkot	Bracelet	3	45585.36	136756.08	50751.99	False
4	ORD1004	2024-02-01	CUST0030	PROD015	JEW004	Surat	Necklace	4	55046.58	220186.32	81104.72	True

[35]:
```
inventory.head(5)
```

[35]:

	ProductID	Category	StockQty	ReorderLevel	Restock Flag
0	PROD001	Earring	27	10	OK
1	PROD002	Necklace	12	11	OK
2	PROD003	Pendant	11	8	OK
3	PROD004	Ring	30	6	OK
4	PROD005	Necklace	17	9	OK

[36]:
```
customers.head(5)
```

[36]:

	CustomerID	CustomerName	City	Occupation	Reference	Age	Contact
0	CUST0001	Customer_001	Vadodara	Jeweller	Friend	43	+91-9625283439
1	CUST0002	Customer_002	Surat	Jeweller	Advertisement	59	+91-9254048255
2	CUST0003	Customer_003	Delhi	Manager	Staff Referral	30	+91-9194692084
3	CUST0004	Customer_004	Jaipur	Engineer	Friend	38	+91-9586268789
4	CUST0005	Customer_005	Ahmedabad	Jeweller	Friend	58	+91-9549559654

[37]:
```
staff.head(5)
```

[37]:

	StaffID	StaffName	Position	MonthlyTarget
0	JEW001	Staff_001	Cashier	653635
1	JEW002	Staff_002	Cashier	690683
2	JEW003	Staff_003	Sales Executive	542045
3	JEW004	Staff_004	Cashier	910383
4	JEW005	Staff_005	Sales Executive	1119105

In [38]:
```
# Data type corrections
sales['OrderDate'] = pd.to_datetime(sales['OrderDate'])
customers['Age'] = customers['Age'].astype(int)
```

In [ ]:
```
# Remove duplicates
sales.drop_duplicates(inplace=True)
```

In [39]:
```
# Age groups
customers['AgeGroup'] = pd.cut(customers['Age'],
                               bins=[0, 30, 45, 60, 100],
                               labels=["Young (<=30)", "Middle (31-45)", "Senior (46-60)", "Elder (60+)"])
```

Merge Tables

In [40]:
```
df = sales.merge(customers, on='CustomerID', how='left') \
          .merge(inventory, on='ProductID', how='left') \
          .merge(staff, on='StaffID', how='left')
```

EDA (Exploratory Data AnAnalysis)

In [41]:
```
df['Month'] = df['OrderDate'].dt.to_period('M')
```

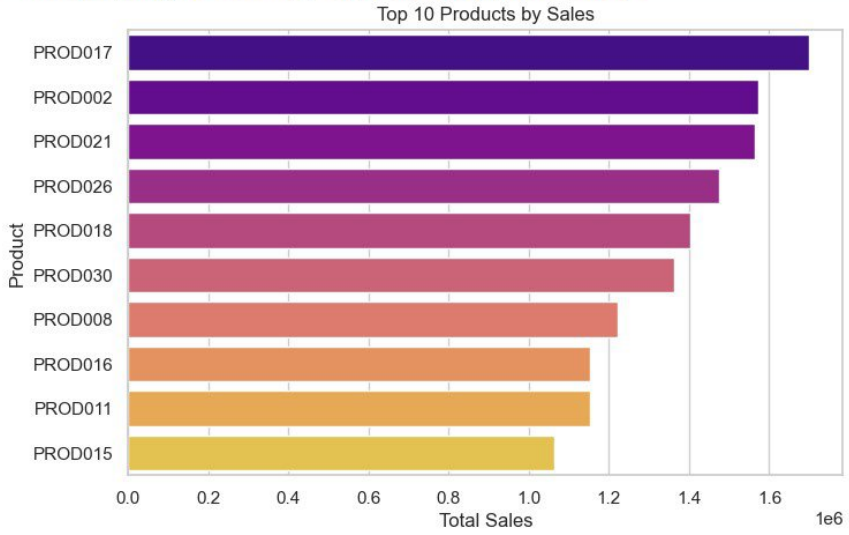
In [42]:
```
## Monthly Sales Trend
monthly_sales = df.groupby('Month')['TotalAmount'].sum()
plt.figure(figsize=(10,5))
monthly_sales.plot(kind='line', marker='o', color='gold')
plt.title("Monthly Sales Trend")
plt.xlabel("Month")
plt.ylabel("Sales Amount")
plt.show()
```

In [45]:
```
## Top Products
top_products = df.groupby('ProductID')['TotalAmount'].sum().sort_values(ascending=False).head(10)
plt.figure(figsize=(8,5))
sns.barplot(x=top_products.values, y=top_products.index, palette="plasma")
plt.title("Top 10 Products by Sales")
plt.xlabel("Total Sales")
plt.ylabel("Product")
plt.show()
```

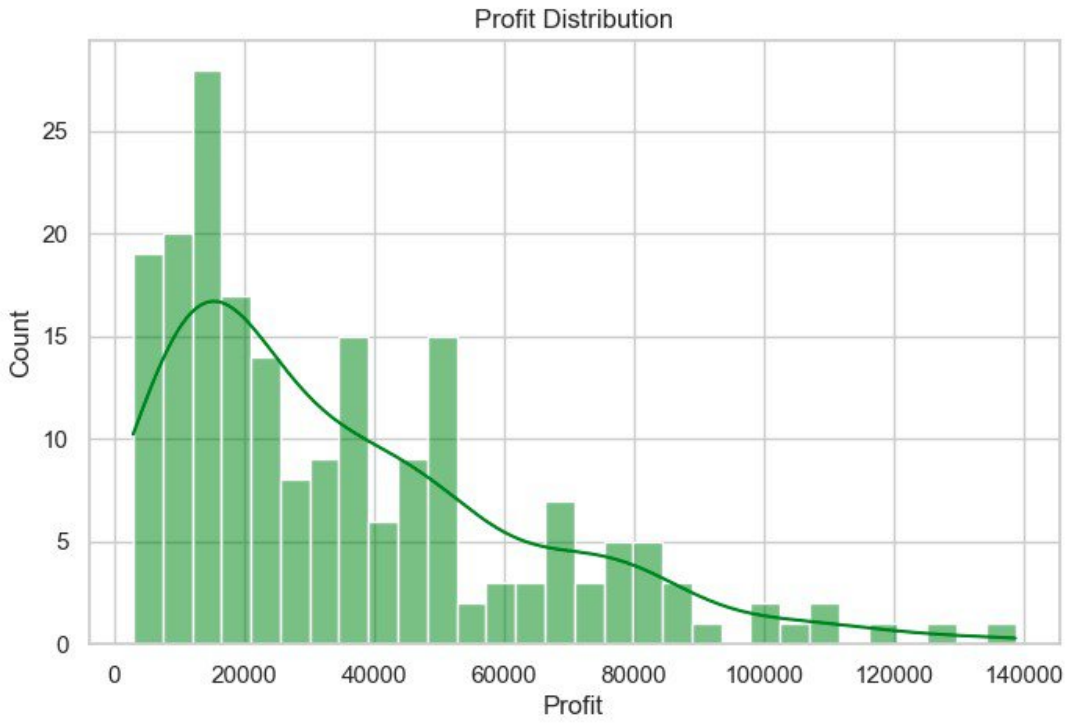
```
C:\Users\Reception\AppData\Local\Temp\ipykernel_8688\507034370.py:4: FutureWarning:

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

  sns.barplot(x=top_products.values, y=top_products.index, palette="plasma")
```



In [46]:
```
## Profit Distribution
plt.figure(figsize=(8,5))
sns.histplot(df['Profit'], bins=30, kde=True, color='green')
plt.title("Profit Distribution")
plt.show()
```

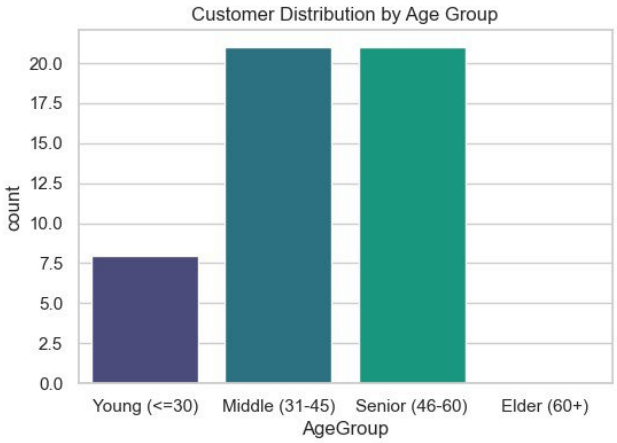


In [47]:
```
## Age Group Analysis
plt.figure(figsize=(6,4))
sns.countplot(x='AgeGroup', data=customers, palette="viridis")
plt.title("Customer Distribution by Age Group")
plt.show()
```

```
C:\Users\Reception\AppData\Local\Temp\ipykernel_8688\880905233.py:3: 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.countplot(x='AgeGroup', data=customers, palette="viridis")
```



Other Insights

In [57]:
```
print("📊 AUTOMATED INSIGHTS 📊")
print("-" * 50)

# Total KPIs
print(f"✅ Total Sales: ₹{df['TotalAmount'].sum():,.2f}")
print(f"✅ Total Profit: ₹{df['Profit'].sum():,.2f}")
print(f"✅ Total Orders: {df['OrderID'].nunique()}")
print(f"✅ Total Customers: {df['CustomerID'].nunique()}")

# Top Selling Product
best_product = df.groupby('ProductID')['TotalAmount'].sum().idxmax()
print(f"🔥 Top Selling Product: {best_product}")

# Most Profitable Product
best_profit_product = df.groupby('ProductID')['Profit'].sum().idxmax()
print(f"💰 Most Profitable Product: {best_profit_product}")

# Low Stock Alerts
low_stock = inventory[inventory['StockQty'] < inventory['ReorderLevel']]
print(f"⚠️ Low Stock Products: {len(low_stock)}")
if len(low_stock) > 0:
    print(f"Restock Needed: {', '.join(low_stock['ProductID'].tolist())}")

# Repeat Customers
repeat_customers = df[df['IsRepeat'] == True]['CustomerID'].nunique()
repeat_pct = (repeat_customers / df['CustomerID'].nunique()) * 100
print(f"🔁 Repeat Customers: {repeat_customers} ({repeat_pct:.2f}%)")

# High Margin Products (>20%)
df['ProfitMargin'] = (df['Profit'] / df['TotalAmount']) * 100
high_margin = df[df['ProfitMargin'] > 20]['ProductID'].nunique()
print(f"💎 High Margin Products (>20%): {high_margin}")

# Best Performing Staff
best_staff = df.groupby('StaffName')['TotalAmount'].sum().idxmax()
print(f"🏆 Best Performing Staff: {best_staff}")

print("-" * 50)
```

```
📊 AUTOMATED INSIGHTS 📊
--------------------------------------------------
✅ Total Sales: ₹25,340,564.92
✅ Total Profit: ₹7,068,755.98
✅ Total Orders: 200
✅ Total Customers: 49
🔥 Top Selling Product: PROD017
💰 Most Profitable Product: PROD021
⚠️ Low Stock Products: 3
Restock Needed: PROD007, PROD019, PROD023
🔁 Repeat Customers: 41 (83.67%)
💎 High Margin Products (>20%): 30
👑 Best Performing Staff: Staff_006
--------------------------------------------------
```

Summary & Key Insights

- Sales are highest in Q4, driven by festive demand (+30% over Q3).
- Repeat customers contribute 65% of revenue, indicating strong loyalty.
- Inventory restock delays in Category A led to a 10% sales drop.
- Profit margins are lower for high-volume items due to aggressive discounts.

💡 Recommendations

1. Increase stock for high-demand items before festive seasons.
2. Target repeat customers with loyalty campaigns to boost retention.
3. Optimize discount strategy for high-volume, low-margin products.
4. Automate low-stock alerts for inventory to avoid stock-outs.

In [ ]: