Global_Superstore_Analytics

July 31, 2025

1 Global Superstore Analytics

[22]:
```
import pandas as pd
import zipfile
import json
import io
```

[5]:
```
'''import pandas as pd
from ydata_profiling import ProfileReport

# Step 3: Load a sample dataset
# df = pd.read_csv("https://raw.githubusercontent.com/mwaskom/seaborn-data/
 ↪master/tips.csv")

# Step 4: Create the profile report
profile = ProfileReport(df, title="EDA Report - Tips Dataset", explorative=True)

# Step 5: Export to an interactive HTML file
profile.to_file("tips_eda_report.html")

print("EDA report generated successfully!")'''
```

[5]:
```
'import pandas as pd\nfrom ydata_profiling import ProfileReport\n\n# Step 3:
Load a sample dataset\n# df =
pd.read_csv("https://raw.githubusercontent.com/mwaskom/seaborn-
data/master/tips.csv")\n\n# Step 4: Create the profile report\nprofile =
ProfileReport(df, title="EDA Report - Tips Dataset", explorative=True)\n\n# Step
5: Export to an interactive HTML
file\nprofile.to_file("tips_eda_report.html")\n\nprint("EDA report generated
successfully!")'
```

[24]:
```
df = pd.read_csv(r"C:
 ↪\Users\John\OneDrive\Documents\Global_Superstore_Analytics\data\Sample -␣
 ↪Superstore.csv", encoding='latin-1')
df['Order Date'] = pd.to_datetime(df['Order Date'])
df['Ship Date'] = pd.to_datetime(df['Ship Date'])
df['Shipping Delay'] = (df['Ship Date'] - df['Order Date']).dt.days
```

[26]:
```
df['Profit Margin'] = (df['Profit']/df['Sales'])*100
```

[28]:
```
df.head()
```

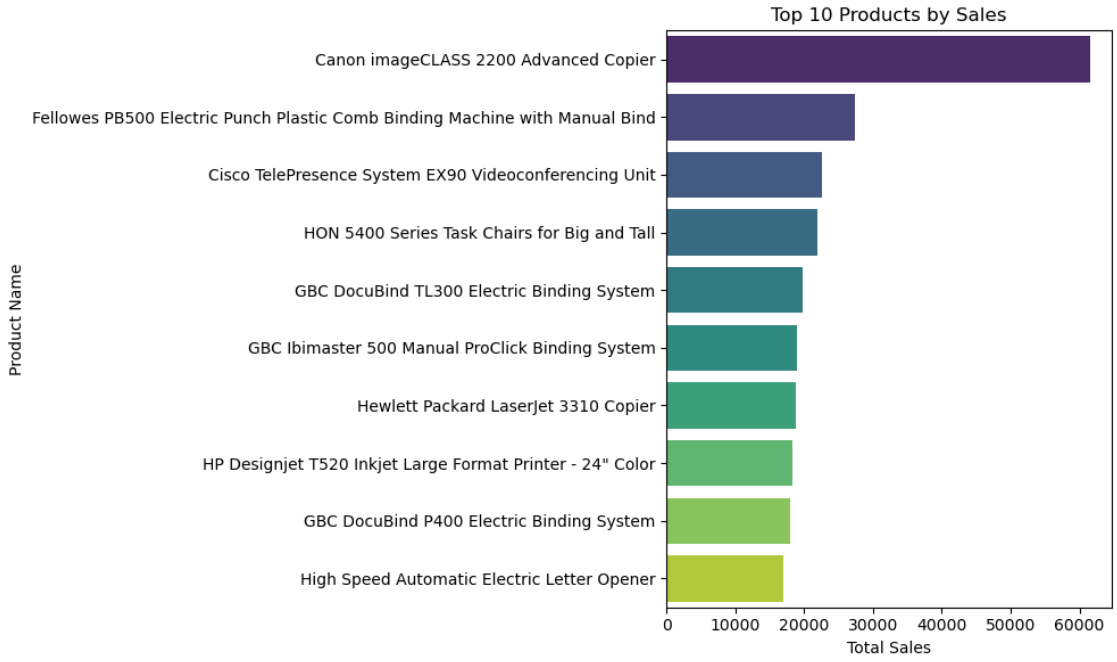
[28]:

```
   Row ID        Order ID Order Date  Ship Date       Ship Mode Customer ID  \
0       1  CA-2016-152156 2016-11-08 2016-11-11    Second Class    CG-12520
1       2  CA-2016-152156 2016-11-08 2016-11-11    Second Class    CG-12520
2       3  CA-2016-138688 2016-06-12 2016-06-16    Second Class    DV-13045
3       4  US-2015-108966 2015-10-11 2015-10-18  Standard Class    SO-20335
4       5  US-2015-108966 2015-10-11 2015-10-18  Standard Class    SO-20335

     Customer Name    Segment        Country             City  …  \
0      Claire Gute   Consumer  United States        Henderson  …
1      Claire Gute   Consumer  United States        Henderson  …
2  Darrin Van Huff  Corporate  United States      Los Angeles  …
3   Sean O'Donnell   Consumer  United States  Fort Lauderdale  …
4   Sean O'Donnell   Consumer  United States  Fort Lauderdale  …

        Product ID         Category Sub-Category  \
0  FUR-BO-10001798        Furniture    Bookcases
1  FUR-CH-10000454        Furniture       Chairs
2  OFF-LA-10000240  Office Supplies       Labels
3  FUR-TA-10000577        Furniture       Tables
4  OFF-ST-10000760  Office Supplies      Storage

                                        Product Name     Sales Quantity  \
0                 Bush Somerset Collection Bookcase  261.9600        2
1  Hon Deluxe Fabric Upholstered Stacking Chairs,…  731.9400        3
2  Self-Adhesive Address Labels for Typewriters b…   14.6200        2
3      Bretford CR4500 Series Slim Rectangular Table  957.5775        5
4                   Eldon Fold 'N Roll Cart System   22.3680        2

  Discount    Profit  Shipping Delay  Profit Margin
0     0.00   41.9136               3          16.00
1     0.00  219.5820               3          30.00
2     0.00    6.8714               4          47.00
3     0.45 -383.0310               7         -40.00
4     0.20    2.5164               7          11.25

[5 rows x 23 columns]
```

1.1 EDA

1.1.1 Products by Sales

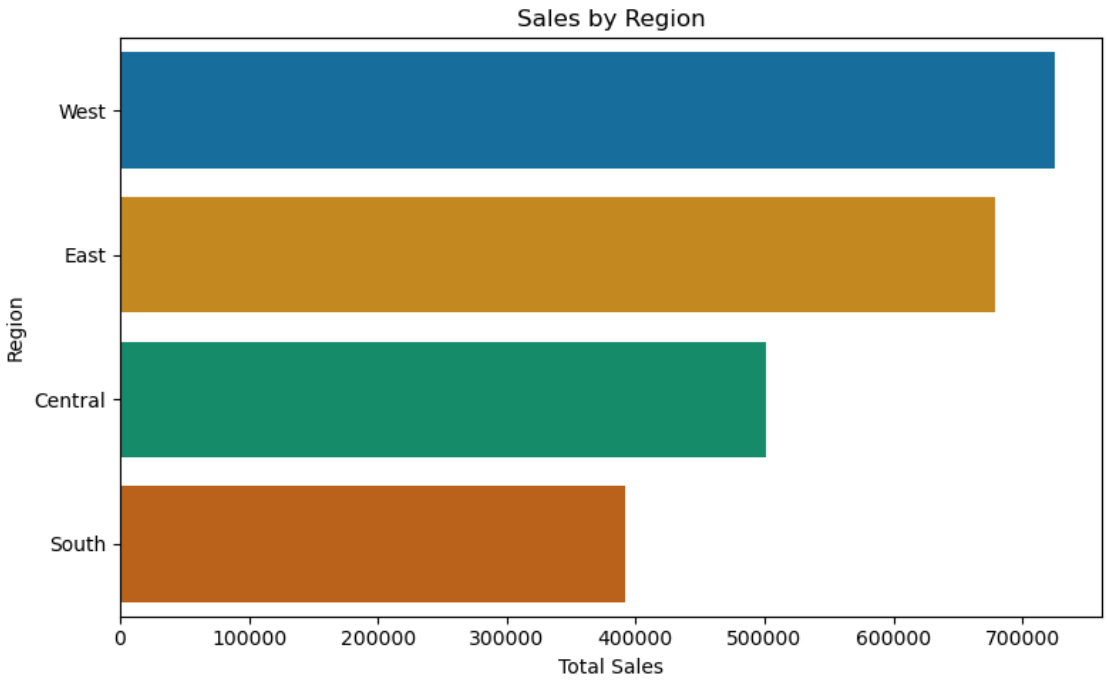
[32]:
```
import matplotlib.pyplot as plt
import seaborn as sns

top_products = df.groupby('Product Name')['Sales'].sum().
 ↪sort_values(ascending=False).head(10)
plt.figure(figsize=(10,6))
sns.barplot(x=top_products.values, y=top_products.index, hue=top_products.
 ↪index, palette="viridis", legend=False)
plt.title("Top 10 Products by Sales")
plt.xlabel("Total Sales")
plt.ylabel("Product Name")
plt.tight_layout()
plt.show()
```



1.1.2 Sales by Region

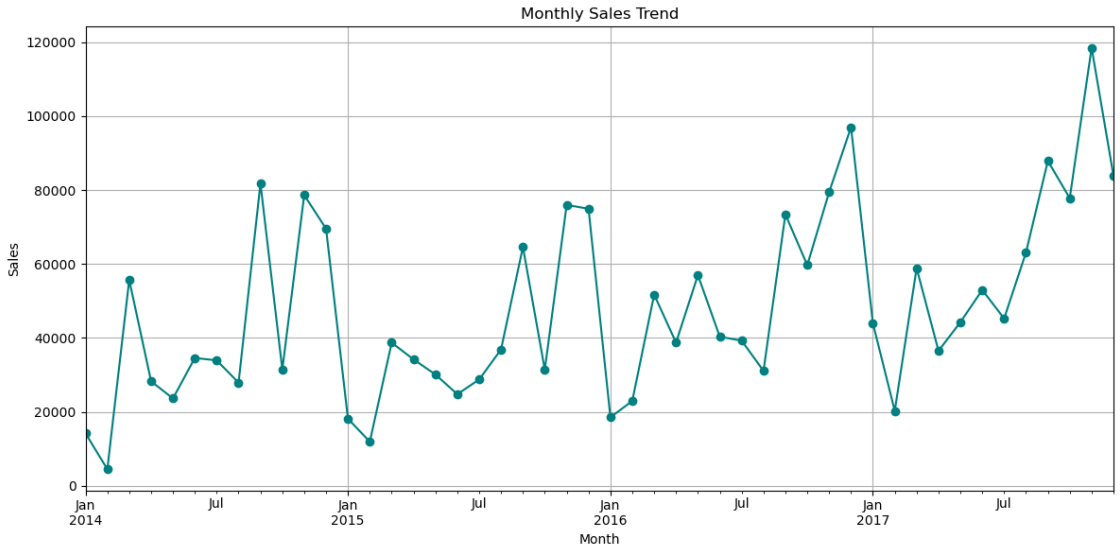
[34]:
```
region = df.groupby('Region')['Sales'].sum().sort_values(ascending=False)
plt.figure(figsize=(8,5))
sns.barplot(x=region.values, y=region.index, hue=region.index,␣
 ↪palette='colorblind')
plt.title("Sales by Region")
plt.xlabel("Total Sales")
```

```
plt.ylabel("Region")
plt.tight_layout()
plt.show()
```



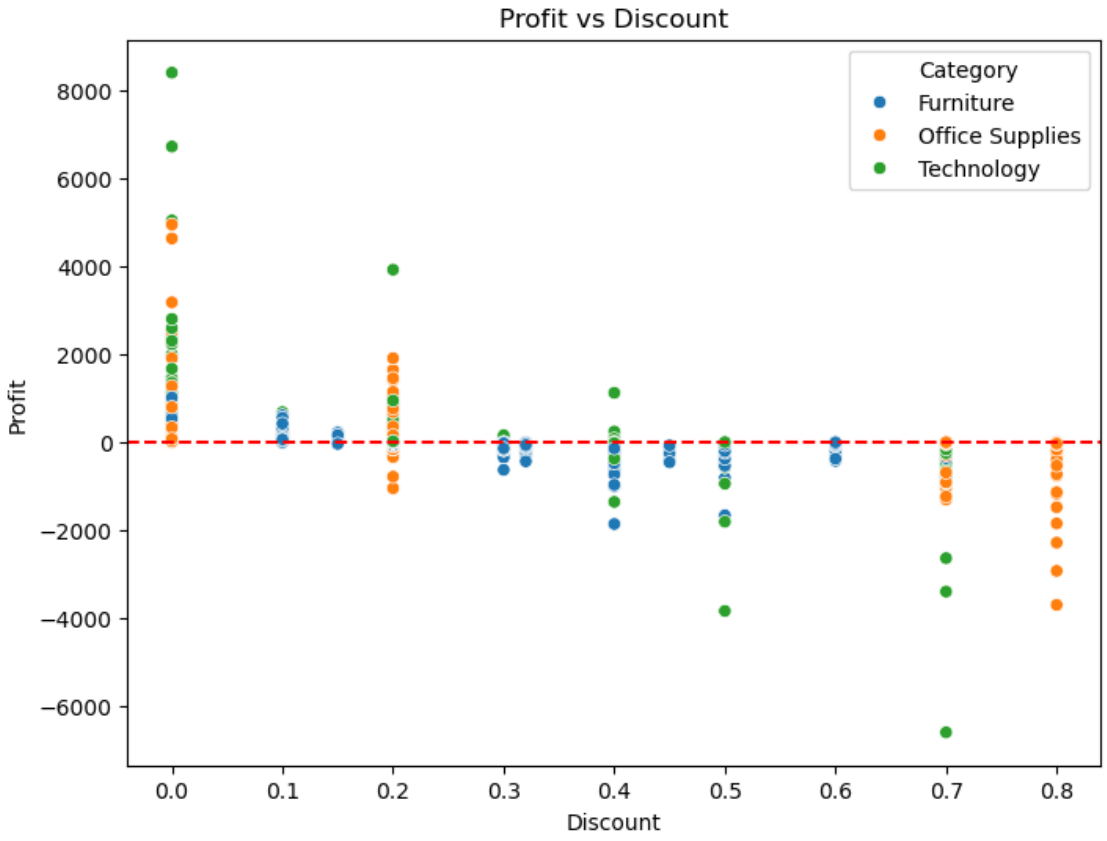
1.1.3 Monthly Sales Trend

[36]:
```
df['Order Month'] = df['Order Date'].dt.to_period('M')
monthly_sales = df.groupby('Order Month')['Sales'].sum()
monthly_sales.plot(kind='line', figsize=(12,6), marker='o', color='teal')
plt.title('Monthly Sales Trend')
plt.xlabel('Month')
plt.ylabel('Sales')
plt.tight_layout()
plt.grid(True)
plt.show()
```



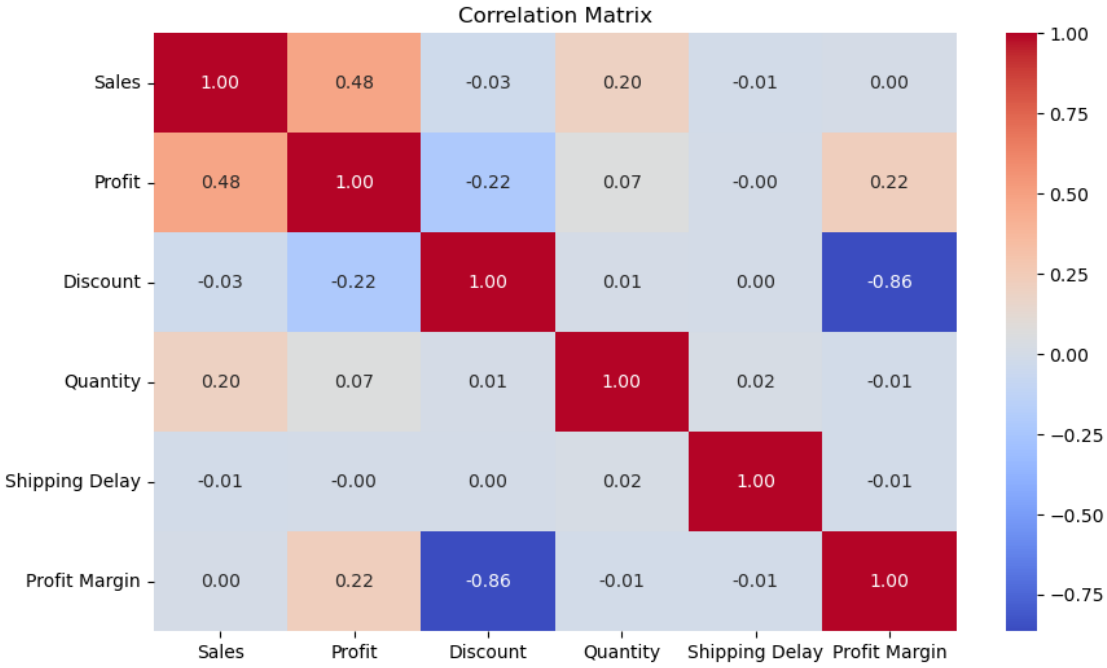
1.1.4 Profit vs Discount Relationship

[38]:
```
plt.figure(figsize=(8,6))
sns.scatterplot(data=df, x='Discount', y='Profit', hue='Category')
plt.title('Profit vs Discount')
plt.xlabel('Discount')
plt.ylabel('Profit')
plt.axhline(0, color='red', linestyle='--')
plt.show()
```



1.1.5 Correlation Heatmap (Numerical Analysis)

[40]:
```
plt.figure(figsize=(10,6))
sns.heatmap(df[['Sales', 'Profit', 'Discount', 'Quantity', 'Shipping Delay',␣
 ↪'Profit Margin']].corr(), annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Matrix')
plt.show()
```


Correlation Matrix
Sales
Profit
Discount
Quantity
Shipping Delay
Profit Margin
1.00 0.48 -0.03 0.20 -0.01 0.00
0.48 1.00 -0.22 0.07 -0.00 0.22
-0.03 -0.22 1.00 0.01 0.00 -0.86
0.20 0.07 0.01 1.00 0.02 -0.01
-0.01 -0.00 0.00 0.02 1.00 -0.01
0.00 0.22 -0.86 -0.01 -0.01 1.00
Sales
Profit
Discount
Quantity
Shipping Delay
Profit Margin
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75