

```
In [2]: pip install pandas matplotlib seaborn

Requirement already satisfied: pandas in c:\users\geraldine nyika\anaconda3\lib\site-packages (2.2.2)Note: you may need to restart the kernel to use updated packages.

Requirement already satisfied: matplotlib in c:\users\geraldine nyika\anaconda3\lib\site-packages (3.9.2)
Requirement already satisfied: seaborn in c:\users\geraldine nyika\anaconda3\lib\site-packages (0.13.2)
Requirement already satisfied: numpy<1.16.0 in c:\users\geraldine nyika\anaconda3\lib\site-packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil<=2.8.2 in c:\users\geraldine nyika\anaconda3\lib\site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz<2023.1 in c:\users\geraldine nyika\anaconda3\lib\site-packages (from pandas) (2023.3)
Requirement already satisfied: tzdata>=2022.7 in c:\users\geraldine nyika\anaconda3\lib\site-packages (from pandas) (2023.3)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\geraldine nyika\anaconda3\lib\site-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\geraldine nyika\anaconda3\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\geraldine nyika\anaconda3\lib\site-packages (from matplotlib) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\geraldine nyika\anaconda3\lib\site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\geraldine nyika\anaconda3\lib\site-packages (from matplotlib) (24.1)
Requirement already satisfied: pillow>=8 in c:\users\geraldine nyika\anaconda3\lib\site-packages (from matplotlib) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\geraldine nyika\anaconda3\lib\site-packages (from matplotlib) (3.1.2)
Requirement already satisfied: six>=1.5 in c:\users\geraldine nyika\anaconda3\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)

In [4]: import pandas as pd

# Load the dataset
file_path = "C:\Users\Geraldine Nyika\Downloads\superstore.csv"
data = pd.read_csv(file_path)

# Display the first few rows of the dataset
print(data.head())

UncaughtException: Traceback (most recent call last)
Cell In[4], line 5
      3 # Load the dataset
      4 # file_path = "C:\Users\Geraldine Nyika\Downloads\superstore.csv"
----> 5 data = pd.read_csv(file_path)
      6 # Display the first few rows of the dataset
      7 print(data.head())

File ~\anaconda3\lib\site-packages\pandas\io\parsers\readers.py:1026, in TextFileReader.__init__(self, filepath_or_buffer, sep, delimiter, header, names, index_col, usecols, dtype, engine, converters, true_value, false_value, skipinitialspace, skiprows, skipfooter, nrows, na_values, keep_default_na, na_filter, verbose, skip_blank_lines, parse_dates, infer_datetime_format, keep_date_col, data_parser, f, date_format, dayfirst, cache_dates, iterator, chunksize, compression, thousands, decimal, lineterminator, quoting, doublequote, escapechar, comment, encoding, encoding_errors, dialect, on_bad_lines, delta_is_datetime, low_memory, memory_map, float_precision, storage_options, dtype_backend)
    1013 kwds_defaults = _infer_defaults_kwds()
    1014 dialect,
    1015 delimiter,
    (...)
    1022 dtype_backend=dtype_backend,
    1023 )
-> 1026 return _read(filepath_or_buffer, kwds)

File ~\anaconda3\lib\site-packages\pandas\io\parsers\readers.py:620, in TextFileReader._read(filepath_or_buffer, kwds)
    617 validate_names(kwargs.get("names", None))
    619 # Create the parser.
--> 620 parser = TextFileReader(filepath_or_buffer, **kwds)
    621 # Chunksize on iterator
    623     return parser

File ~\anaconda3\lib\site-packages\pandas\io\parsers\readers.py:1620, in TextFileReader._init__(self, f, engine, **kwds)
    1617     self.options["has_index_names"] = kwds["has_index_names"]
    1619 self.header_iterator = None
-> 1620 self._engine = self._parse_engine(f, self.engine)

File ~\anaconda3\lib\site-packages\pandas\io\parsers\readers.py:1898, in TextFileReader._parse_engine(self, f, engine)
    1895     raise ValueError(msg)
-> 1898 try:
    1899     return mapping(engine)(f, **self.optional)
    1899 except Exception:
    1900     if self._handles is not None:

File ~\anaconda3\lib\site-packages\pandas\io\parsers\c_parser_wrapper.py:93, in CParserWrapper._init__(self, src, **kwds)
    90 # Fail here loudly instead of in cython after reading
    92     _import_optional_dependency("pyarrow")
--> 93 self._reader = parsers.TextReader(src, **kwds)
    94 self._unnamed_cols = self._reader._unnamed_cols
    97 # error: Cannot determine type of 'names'

File parsers.pyx:574, in pandas._libs.parsers.TextReader._cinit__()
File parsers.pyx:663, in pandas._libs.parsers.TextReader._get_header()
File parsers.pyx:874, in pandas._libs.parsers.TextReader._tokenize_rows()
File parsers.pyx:891, in pandas._libs.parsers.TextReader._check_tokenize_status()
File parsers.pyx:2053, in pandas._libs.parsers.raise_parser_error()

File <frozen code>:322, in _decode(self, input, final)

UncaughtExceptionError: 'utf-8' codec can't decode byte 0xa0 in position 2944: invalid start byte

In [6]: # Question 1
import pandas as pd

# Load the dataset with a specified encoding
file_path = "C:\Users\Geraldine Nyika\Downloads\superstore.csv"
data = pd.read_csv(file_path, encoding="ISO-8859-1")

# Display the first few rows
print(data.head())

Row ID      Order ID  Order Date   Ship Date      Ship Mode Customer ID \
0      1  CA-2016-152156   11/8/2016  11/11/2016      Second Class    CG-10520
1      2  CA-2016-152156   11/8/2016  11/11/2016      Second Class    CG-10520
2      3  CA-2016-138648    6/12/2016   6/16/2016      Second Class    DW-13045
3      4  US-2015-108966   10/11/2015  10/18/2015      Standard Class  SO-20335
4      5  US-2015-108966   10/11/2015  10/18/2015      Standard Class  SO-20335

Customer Name  Segment      Country      City ... \
0  Claire Gote      Consumer      United States      Henderson ...
1  Claire Gote      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 ...

Postal Code  Region      Product ID      Category Sub-Category \
0      42420      South      FUR-BQ-10007798      Furniture      Bookcases
1      42420      South      FUR-CH-10000454      Furniture      Chairs
2      90036      West      OFF-CA-10000240      Office Supplies      Labels
3      33311      South      FUR-TA-10000577      Furniture      Tables
4      33311      South      OFF-ST-10000760      Office Supplies      Storage

Product Name      Sales Quantity \
0  Bush Somerset Collection Bookcase      261.9600      2
1  Non-Deluxe Fabric Upholstered Reading Chair...      731.9400      3
2  Self-Adhesive Address Labels for Typewriters B...      14.6200      2
3  Brevard CR4500 Series Sile Rectangular Table      957.5775      5
4  Eidon Fold 'N Roll Cart System      22.3680      2

Discount      Profit
0      0.00      41.9136
1      0.00      219.3820
2      0.00      6.8714
3      0.45      -383.0310
4      0.20      2.5164

[5 rows x 21 columns]

In [8]: # Group by Product Name to get total sales and profit
product_performance = data.groupby("Product Name")[["Sales", "Profit"]].sum()

In [10]: # Find top 10 products by sales and profit
top_sales_products = product_performance.sort_values(by="Sales", ascending=False).head(10)
top_profit_products = product_performance.sort_values(by="Profit", ascending=False).head(10)

In [12]: # Find bottom 10 products by sales and profit
lowest_sales_products = product_performance.sort_values(by="Sales", ascending=True).head(10)
lowest_profit_products = product_performance.sort_values(by="Profit", ascending=True).head(10)

In [14]: print("Top 10 Products by Sales:\n", top_sales_products)
print("\nBottom 10 Products by Sales:\n", lowest_sales_products)
print("\nTop 10 Products by Profit:\n", top_profit_products)
print("\nBottom 10 Products by Profit:\n", lowest_profit_products)

Top 10 Products by Sales:
Product Name      Sales      Profit
Canon ImageCLASS 2200 Advanced Copier      61599.824      2.519999e+04
Fellowes PB500 Electric Punch Plastic Comb Bind...      27453.384      7.753099e+03
Cisco TelePresence System EX90 Videoconferencin...      22638.480      -1.811078e+03
Hewlett Packard LaserJet 3110 Copier      21870.275      5.681942e-14
GBC DocuBind T1300 Electric Binding System      19823.479      2.235004e+03
GBC Disinater 300 Manual ProClock Binding System      19024.500      7.609800e+02
Hewlett Packard LaserJet 3110 Copier      18939.486      6.383884e+03
HP DesignJet T520 Inkjet Large Format Printer ...      18374.895      4.094977e+03
GBC DocuBind 1400 Electric Binding System      17965.000      -1.878164e+03
High Speed Automatic Electric Letter Opener      17030.312      -2.620048e+02

Bottom 10 Products by Sales:
Product Name      Sales      Profit
Rusaka Disposable Bags for Sanitary Groc...      1.624      -4.4660
Avery 5      5.760      2.8224
Xerox 20      6.480      3.1104
Grip Seal Envelopes      7.072      2.3868
Avery Hi-Liter Pen Style Six-Color Fluorescent Set      7.700      3.1570
Avery Hi-Liter Comfort Grip Fluorescent Highlig...      7.800      3.0400
Xerox 1989      7.968      2.6892
4009 Highlighters      8.040      2.7336
Stonewell Gold Paper Clips      8.096      3.6240
Hewlett 308      8.400      2.1000

Top 10 Products by Profit:
Product Name      Sales      Profit
Canon ImageCLASS 2200 Advanced Copier      61599.824      25199.9280
Fellowes PB500 Electric Punch Plastic Comb Bind...      27453.384      7753.0390
Hewlett Packard LaserJet 3110 Copier      18939.486      6383.8836
Canon PCL1060 Personal Laser Copier      18119.834      4570.9347
HP DesignJet T520 Inkjet Large Format Printer ...      18374.895      4094.9768
Active ValueLine Micro-Out Shredder      7699.490      3733.9461
GBC DocuBind T1300 Electric Binding System      17965.068      -1878.1662
Cisco TelePresence System EX90 Videoconferencin...      22638.480      -1811.0784
Hewlett Packard LaserJet 3110 Copier      18939.486      -1939.1836
Balt Solid Wood Round Tables      6518.754      -1201.0581
BoonOffice By Design Rectangular and Half-Moon R...      1706.250      -1148.4375

In [16]: # Group by Region to get total sales and profit
regional_performance = data.groupby("Region")[["Sales", "Profit"]].sum()

In [18]: # Plotting sales by region
import matplotlib.pyplot as plt
import seaborn as sns

In [20]: sns.barplot(x=regional_performance.index, y=regional_performance["Sales"])
plt.title("Sales by Region")
plt.xlabel("Region")
plt.ylabel("Total Sales")
plt.show()

Sales by Region
Total Sales
700000
600000
500000
400000
300000
200000
100000
0
Central East South West
Region

In [22]: # Plotting profit by region
sns.barplot(x=regional_performance.index, y=regional_performance["Profit"])
plt.title("Profit by Region")
plt.xlabel("Region")
plt.ylabel("Total Profit")
plt.show()

Profit by Region
Total Profit
100000
80000
60000
40000
20000
0
Central East South West
Region

In [24]: # Group by Category to get total sales and profit
category_performance = data.groupby("Category")[["Sales", "Profit"]].sum()

# Plotting sales by category
sns.barplot(x=category_performance.index, y=category_performance.values)
plt.title("Sales by Category")
plt.xlabel("Category")
plt.ylabel("Total Sales")
plt.show()

Sales by Category
Total Sales
800000
700000
600000
500000
400000
300000
200000
100000
0
Furniture Office Supplies Technology
Category

In [26]: # Convert 'Order Date' to datetime format
data["Order Date"] = pd.to_datetime(data["Order Date"])

# Extract the month from the Order Date
data["Order Month"] = data["Order Date"].dt.month

monthly_sales = data.groupby("Order Month")["Sales"].sum()

# Plotting monthly sales trend
plt.plot(monthly_sales.index, monthly_sales.values, marker="o")
plt.title("Monthly Sales Trend")
plt.xlabel("Month")
plt.ylabel("Total Sales")
plt.xticks(range(1, 13)) # Show each month
plt.show()

Monthly Sales Trend
Total Sales
350000
300000
250000
200000
150000
100000
50000
0
1 2 3 4 5 6 7 8 9 10 11 12
Month

In [30]: # Group by Ship Mode to get average sales and profit
shipping_performance = data.groupby("Ship Mode")[["Sales", "Profit"]].mean()

# Plotting average sales by shipping mode
sns.barplot(x=shipping_performance.index, y=shipping_performance["Sales"])
plt.title("Average Sales by Shipping Mode")
plt.xlabel("Shipping Mode")
plt.ylabel("Average Sales")
plt.show()

Average Sales by Shipping Mode
Average Sales
200
150
100
50
0
First Class Same Day Second Class Standard Class
Shipping Mode

In [32]: # Analyzing the Impact of discount on profit
sns.scatterplot(data=data, x="Discount", y="Profit")
plt.title("Impact of Discount on Profit")
plt.xlabel("Discount")
plt.ylabel("Profit")
plt.show()

Impact of Discount on Profit
Profit
8000
6000
4000
2000
0
-2000
-4000
-6000
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8
Discount

In [30]: # Question 2

In [70]: # Set up the figure with a grid layout for the dashboard
plt.figure(figsize=(15, 10))

# Subplot 1: Sales by Region
plt.subplot(2, 3, 1) # (2 rows, 3 columns, chart 1)
sns.barplot(x=regional_sales.index, y=regional_sales.values, palette="Blues_d")
plt.title("Sales by Region")
plt.xlabel("Region")
plt.ylabel("Total Sales")

# Subplot 2: Profit by Region
plt.subplot(2, 3, 2) # (2 rows, 3 columns, chart 2)
sns.barplot(x=regional_profit.index, y=regional_profit.values, palette="Greens_d")
plt.title("Profit by Region")
plt.xlabel("Region")
plt.ylabel("Total Profit")

# Subplot 3: Sales by Category
plt.subplot(2, 3, 3) # (2 rows, 3 columns, chart 3)
sns.barplot(x=category_sales.index, y=category_sales.values, palette="Purples_d")
plt.title("Sales by Category")
plt.xlabel("Category")
plt.ylabel("Total Sales")
plt.xticks(rotation=45, ha="right") # Rotate labels for better readability

# Subplot 4: Monthly Sales Trend
plt.subplot(2, 1, 2) # (2 rows, 1 column, chart 4 occupies entire second row)
plt.plot(monthly_sales.index, monthly_sales.values, marker="o", linestyle="--", color="skyblue")
plt.title("Monthly Sales Trend")
plt.xlabel("Month")
plt.ylabel("Total Sales")
plt.xticks(range(1, 13)) # Show each month on the x-axis

# Adjust layout for better spacing
plt.tight_layout()
plt.show()

C:\Users\Geraldine Nyika\AppData\Local\Temp\ipykernel_3376\4072038368.py:6: 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.barplot(x=regional_sales.index, y=regional_sales.values, palette="Blues_d")
C:\Users\Geraldine Nyika\AppData\Local\Temp\ipykernel_3376\4072038368.py:13: 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.barplot(x=regional_profit.index, y=regional_profit.values, palette="Greens_d")
C:\Users\Geraldine Nyika\AppData\Local\Temp\ipykernel_3376\4072038368.py:20: 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.barplot(x=category_sales.index, y=category_sales.values, palette="Purples_d")

Sales by Region Profit by Region Sales by Category
Total Sales
700000
600000
500000
400000
300000
200000
100000
0
Central East South West
Region
Total Profit
100000
80000
60000
40000
20000
0
Central East South West
Region
Total Sales
800000
700000
600000
500000
400000
300000
200000
100000
0
Furniture Office Supplies Technology
Category

Monthly Sales Trend
Total Sales
350000
300000
250000
200000
150000
100000
50000
0
1 2 3 4 5 6 7 8 9 10 11 12
Month

In [76]: # Question 3
# 3-minute story
# 3-minute story and Big Idea
print("!!! 3-Minute Story")
Good day to you all. The findings of our data analysis on GlobalMart's performance across different regions, categories, and customer segments are something I'm thrilled to share with you.

The analysis uncovered some insightful key points:
- Regional Insights: Although profits differ greatly, the East and West regions have the largest sales volumes. Despite having significant sales, the South region has lower profit margins than the East and West, which are the most lucrative regions. This implies a chance to increase profitability in the South by streamlining our inventory or maximizing expenses.
- Category Performance: Of our product categories, Technology brings in the most money, followed by Office Supplies and Furniture. While investing in high-performing categories like Technology may yield higher returns, we may need to reconsider our strategy in other categories, especially those where sales are stagnating.
- Seasonal and Discount Trends: We can infer seasonal demand from our monthly sales trend, which shows peak sales periods near the end of the year. Discounts do increase sales, but if they are not properly managed, they can drastically lower profit margins according to the analysis. A more calculated approach to discounting might aid in striking a balance between profitability and customer attraction.

To sum it up, concentrating on high-performing areas and product categories, cutting expenses in low-margin areas, and honing our discount strategy will put us in a position to boost profitability and expand our market reach. By adopting a data-driven approach, we can ensure that every strategic choice we make is based on knowledge that will spur growth. I am eager to talk with you more about these suggestions and explore how we can successfully put them into practice.

2. The Big Idea
GlobalMart can increase profitability and bolster its position as a market leader by utilizing data-driven insights to strategically manage discounts, invest in high-performing categories, and optimize regional strategies.

In [77]:
In [78]:
In [79]:
In [80]:
```

