


```
import pandas as pd

df = pd.read_csv('/content/Global_Superstore.csv')

df.head()
```




	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Postal Code	City	...	Product ID	Category	Cat
0	40098	CA-2014-AB10015140-41954	11/11/2014	11/13/2014	First Class	AB-100151402	Aaron Bergman	Consumer	73120.0	Oklahoma City	...	TEC-PH-5816	Technology	F
1	26341	IN-2014-JR162107-41675	2/5/2014	2/7/2014	Second Class	JR-162107	Justin Ritter	Corporate	NaN	Wollongong	...	FUR-CH-5379	Furniture	
2	25330	IN-2014-CR127307-41929	10/17/2014	10/18/2014	First Class	CR-127307	Craig Reiter	Consumer	NaN	Brisbane	...	TEC-PH-5356	Technology	F
3	13524	ES-2014-KM1637548-41667	1/28/2014	1/30/2014	First Class	KM-1637548	Katherine Murray	Home Office	NaN	Berlin	...	TEC-PH-5267	Technology	F
4	47221	SG-2014-RH9495111-41948	11/5/2014	11/6/2014	Same Day	RH-9495111	Rick Hansen	Consumer	NaN	Dakar	...	TEC-CO-6011	Technology	C

5 rows × 24 columns

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 24 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Row ID                1000 non-null   int64
1   Order ID              1000 non-null   object
2   Order Date            1000 non-null   object
3   Ship Date             1000 non-null   object
4   Ship Mode             1000 non-null   object
5   Customer ID           1000 non-null   object
6   Customer Name         1000 non-null   object
7   Segment               1000 non-null   object
8   Postal Code           194 non-null    float64
9   City                  1000 non-null   object
10  State                 1000 non-null   object
11  Country               1000 non-null   object
12  Region                1000 non-null   object
13  Market                1000 non-null   object
14  Product ID            1000 non-null   object
15  Category              1000 non-null   object
16  Sub-Category          1000 non-null   object
17  Product Name          1000 non-null   object
18  Sales                  1000 non-null   float64
19  Quantity              1000 non-null   int64
20  Discount              1000 non-null   float64
21  Profit                1000 non-null   float64
22  Shipping Cost         1000 non-null   float64
23  Order Priority         1000 non-null   object
dtypes: float64(5), int64(2), object(17)
memory usage: 187.6+ KB

print("Shape of the dataframe:")
print(df.shape)
print("\nDescriptive statistics of the dataframe:")
print(df.describe())
```



Shape of the dataframe:
(1000, 24)
Descriptive statistics of the dataframe:
Row ID Postal Code Sales Quantity Discount \

count	1000.000000	194.000000	1000.000000	1000.000000	1000.000000
mean	25079.328000	53966.170103	1710.971470	5.55800	0.092840
std	12897.726632	33734.306466	1259.239238	2.71846	0.148666
min	58.000000	2920.000000	1.910000	1.00000	0.000000
25%	15118.750000	19134.000000	826.907500	4.00000	0.000000
50%	25084.500000	60564.000000	1585.115000	5.00000	0.000000
75%	34524.000000	88187.500000	2477.812500	7.00000	0.150000
max	51284.000000	98198.000000	9892.740000	14.00000	0.800000

	Profit	Shipping Cost
count	1000.000000	1000.000000
mean	288.920440	272.384897
std	574.504782	176.160716
min	-3059.820000	1.070000
25%	10.037500	209.827500
50%	190.685000	258.897500
75%	518.872500	351.070250
max	4946.370000	923.630000

▼ Data Cleaning & Pre-Processing

Check for missing values Handle any missing or inconsistent data Convert data types if necessary (e.g., dates to datetime objects)

```
print("Missing values before handling:")
print(df.isnull().sum())

# Convert date columns to datetime objects
df['Order Date'] = pd.to_datetime(df['Order Date'])
df['Ship Date'] = pd.to_datetime(df['Ship Date'])

print("\nData types after converting date columns:")
print(df.info())
```

Missing values before handling:

Row ID	0
Order ID	0
Order Date	0
Ship Date	0
Ship Mode	0
Customer ID	0
Customer Name	0
Segment	0
Postal Code	806
City	0
State	0
Country	0
Region	0
Market	0
Product ID	0
Category	0
Sub-Category	0
Product Name	0
Sales	0
Quantity	0
Discount	0
Profit	0
Shipping Cost	0
Order Priority	0

dtype: int64

Data types after converting date columns:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 1000 entries, 0 to 999

Data columns (total 24 columns):

#	Column	Non-Null Count	Dtype
0	Row ID	1000 non-null	int64
1	Order ID	1000 non-null	object
2	Order Date	1000 non-null	datetime64[ns]
3	Ship Date	1000 non-null	datetime64[ns]
4	Ship Mode	1000 non-null	object
5	Customer ID	1000 non-null	object
6	Customer Name	1000 non-null	object
7	Segment	1000 non-null	object
8	Postal Code	194 non-null	float64
9	City	1000 non-null	object
10	State	1000 non-null	object
11	Country	1000 non-null	object
12	Region	1000 non-null	object
13	Market	1000 non-null	object
14	Product ID	1000 non-null	object
15	Category	1000 non-null	object
16	Sub-Category	1000 non-null	object
17	Product Name	1000 non-null	object
18	Sales	1000 non-null	float64

```

19 Quantity      1000 non-null    int64
20 Discount      1000 non-null    float64
21 Profit        1000 non-null    float64
22 Shipping Cost 1000 non-null    float64
23 Order Priority 1000 non-null    object
dtypes: datetime64[ns](2), float64(5), int64(2), object(15)

```

Basic Analysis

Calculate total sales and profit Identify top-selling products Analyze sales by region Examine the relationship between sales and profit

```

# Calculate total sales and profit
total_sales = df['Sales'].sum()
total_profit = df['Profit'].sum()
print(f"Total Sales: {total_sales:.2f}")
print(f"Total Profit: {total_profit:.2f}")

# Identify top-selling products
top_selling_products = df.groupby('Product Name')['Sales'].sum().sort_values(ascending=False).head(10)
print("\nTop 10 Selling Products:")
print(top_selling_products)

# Analyze sales by region
sales_by_region = df.groupby('Region')['Sales'].sum().sort_values(ascending=False)
print("\nSales by Region:")
print(sales_by_region)

# Examine the relationship between sales and profit (using correlation)
sales_profit_correlation = df['Sales'].corr(df['Profit'])
print(f"\nCorrelation between Sales and Profit: {sales_profit_correlation:.2f}")

```

↗ Total Sales: 1710971.47
Total Profit: 288920.44

Top 10 Selling Products:

Product Name	
Motorola Smart Phone, Full Size	48542.52
Apple Smart Phone, Full Size	42303.44
Cisco Smart Phone, Full Size	41325.20
Nokia Smart Phone, Full Size	27661.97
Samsung Smart Phone, Cordless	25438.59
Hoover Stove, Red	23477.81
Motorola Smart Phone, with Caller ID	23239.44
Harbour Creations Executive Leather Armchair, Adjustable	22145.80
Cisco Smart Phone, with Caller ID	22047.89
Cisco Smart Phone, Cordless	21954.87

Name: Sales, dtype: float64

Sales by Region:

Region	
Western Europe	259576.28
Oceania	220809.08
Southern Asia	205466.26
Eastern Asia	193590.60
Southeastern Asia	147337.60
Central America	140966.21
Northern Europe	103888.63
Southern Europe	94441.11
South America	64550.54
Eastern Europe	45662.19
Central Africa	33810.48
Western Asia	33308.21
Caribbean	30746.74
North Africa	26266.44
Southern Africa	22262.28
Eastern Africa	17387.94
Central US	17187.94
Western US	16729.95
Western Africa	15478.79
Eastern US	12256.44
Southern US	8360.75
Canada	887.01

Name: Sales, dtype: float64

Correlation between Sales and Profit: 0.53

Data Visualization

Create a bar chart of sales by product category Plot a line graph of monthly sales over time Generate a scatter plot of sales vs. profit Produce a heatmap of sales by region and product category

```

import matplotlib.pyplot as plt
import seaborn as sns

```

```

# Create a bar chart of sales by product category

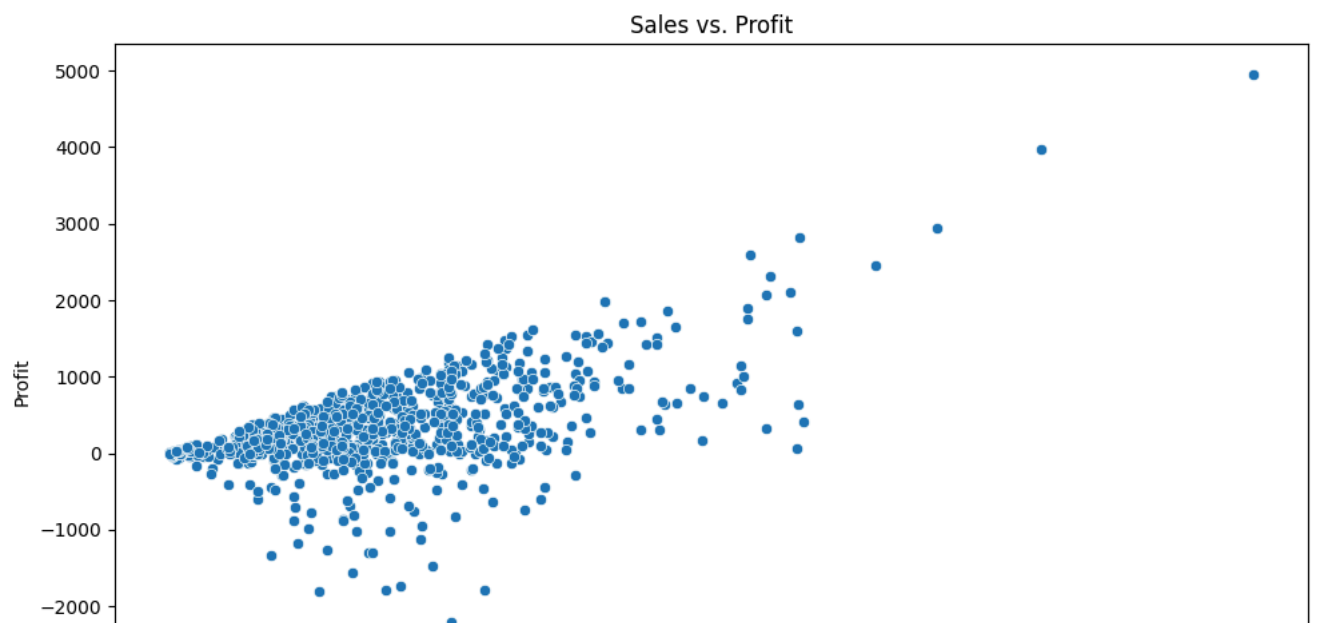
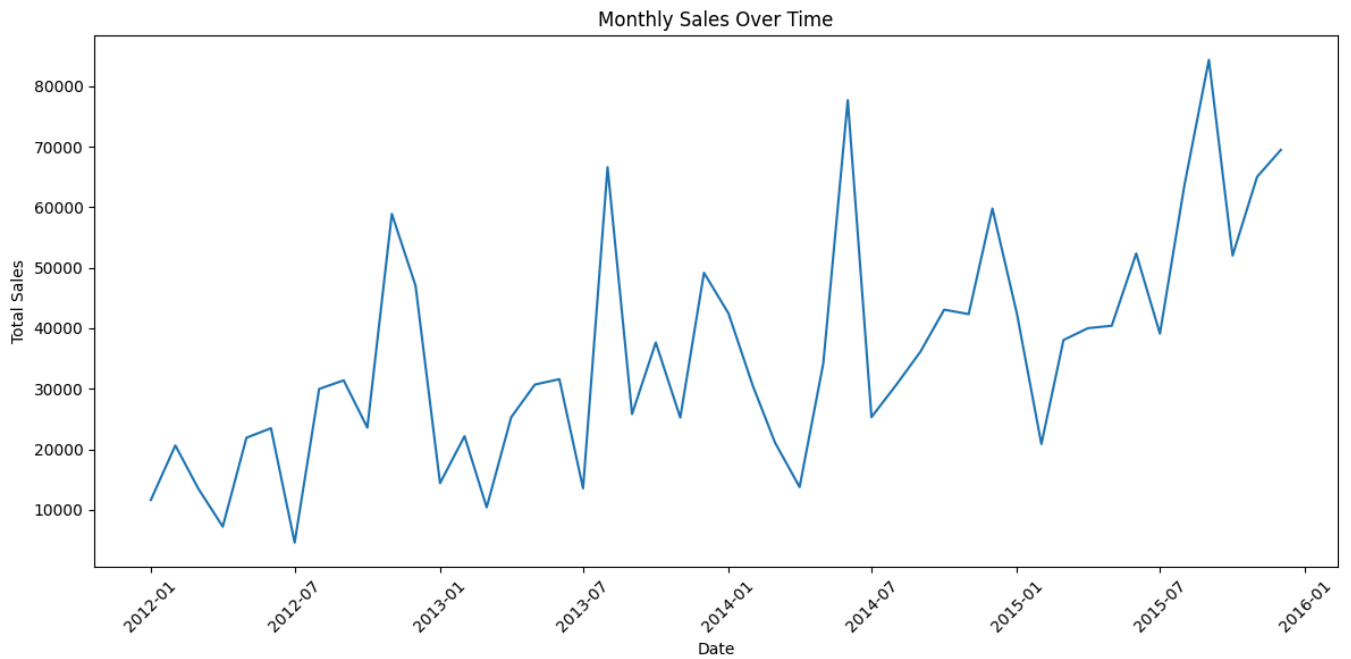
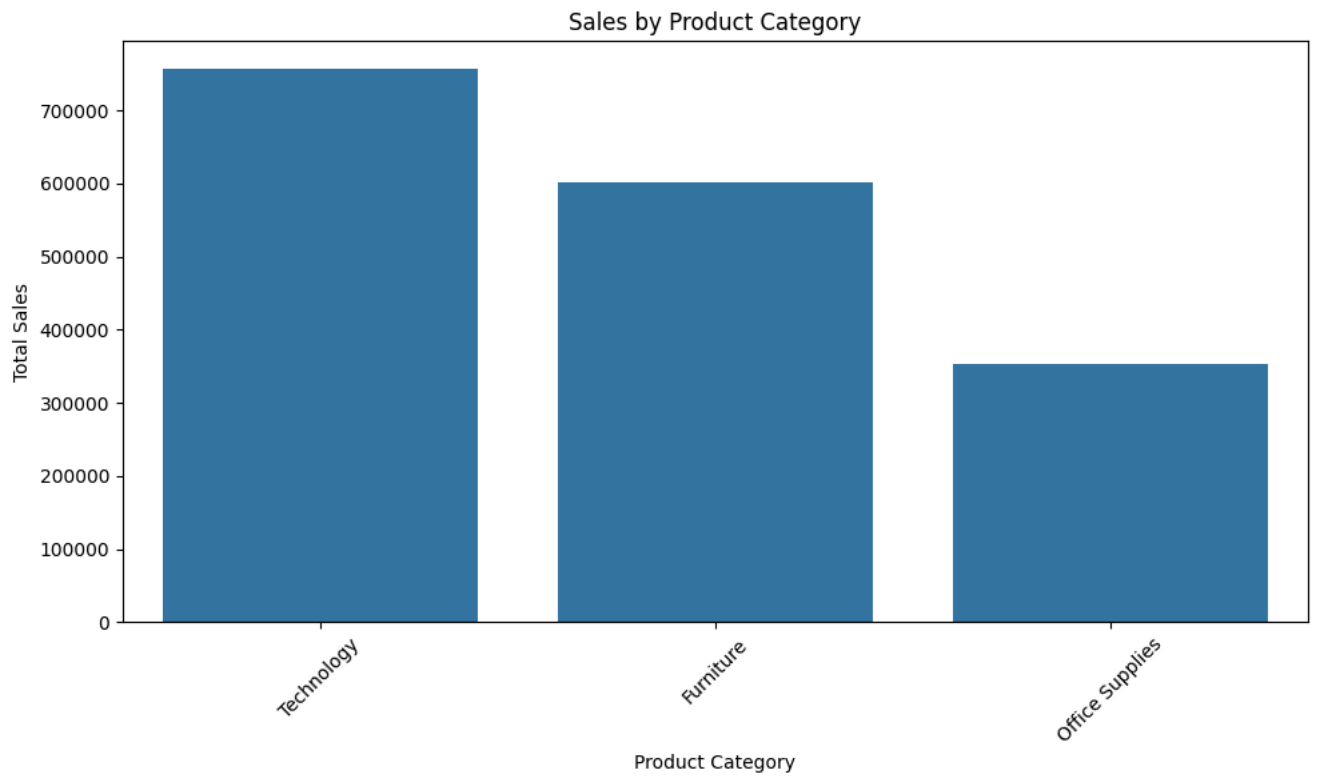
```

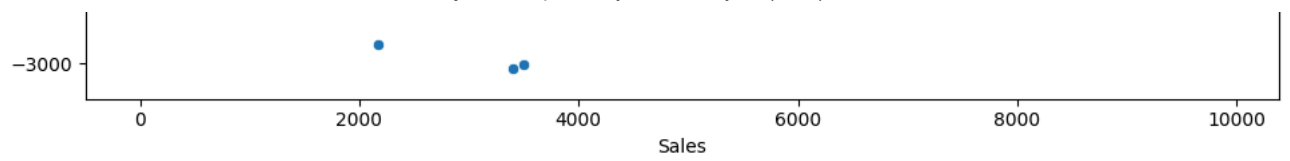
```
sales_by_category = df.groupby('Category')['Sales'].sum().sort_values(ascending=False)
plt.figure(figsize=(10, 6))
sns.barplot(x=sales_by_category.index, y=sales_by_category.values)
plt.title('Sales by Product Category')
plt.xlabel('Product Category')
plt.ylabel('Total Sales')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
# Plot a line graph of monthly sales over time
df['Order Month'] = df['Order Date'].dt.to_period('M')
monthly_sales = df.groupby('Order Month')['Sales'].sum()
monthly_sales.index = monthly_sales.index.to_timestamp()
plt.figure(figsize=(12, 6))
plt.plot(monthly_sales.index, monthly_sales.values)
plt.title('Monthly Sales Over Time')
plt.xlabel('Date')
plt.ylabel('Total Sales')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
# Generate a scatter plot of sales vs. profit
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Sales', y='Profit', data=df)
plt.title('Sales vs. Profit')
plt.xlabel('Sales')
plt.ylabel('Profit')
plt.tight_layout()
plt.show()
```

```
# Produce a heatmap of sales by region and product category
sales_heatmap_data = df.groupby(['Region', 'Category'])['Sales'].sum().unstack()
plt.figure(figsize=(12, 8))
sns.heatmap(sales_heatmap_data, annot=True, fmt=".1f", cmap="YlGnBu")
plt.title('Sales by Region and Product Category')
plt.xlabel('Product Category')
plt.ylabel('Region')
plt.tight_layout()
plt.show()
```





Sales by Region and Product Category

Region	Product Category		
	Furniture	Office Supplies	Technology
Canada		887.0	
Caribbean	12935.2	7971.2	9840.3
Central Africa	13578.5	3129.5	17102.5
Central America	43213.8	25707.0	72045.4
Central US	2085.2	12231.0	2871.8
Eastern Africa	3989.2	4820.0	8578.8
Eastern Asia	76786.8	29599.2	87204.5
Eastern Europe	20590.0	8371.4	16700.8
Eastern US	3258.5	7326.6	1671.4
North Africa	5496.1	5744.9	15025.4
Northern Europe	32728.4	21317.6	49842.6
Oceania	87320.4	36872.9	96615.9
South America	29360.7	14730.7	20459.1
Southeastern Asia	59642.6	37099.7	50595.3
Southern Africa	4129.5	4022.2	14110.6
Southern Asia	82296.2	30840.1	92330.0
Southern Europe	24145.5	40863.6	29432.0
Southern US	1896.4	4275.5	2188.8
Western Africa	1669.2	1239.0	12570.5
Western Asia	9297.9	3425.4	20584.9
Western Europe	80991.0	47146.3	131439.0
Western US	6081.1	4875.4	5773.5

Derive Insights

Based on your analysis and visualizations, list 3-5 key insights about GlobalMart's sales performance

```
# Based on the analysis and visualizations, here are some key insights about GlobalMart's sales performance:
print("Key Insights about GlobalMart's Sales Performance:")
print("- Technology and Furniture are the top-selling product categories, significantly contributing to total sales.")
print("- Sales show a clear upward trend over time, with notable peaks towards the end of each year.")
print("- There is a positive correlation between Sales and Profit (0.53), suggesting that higher sales generally lead to higher profits.")
print("- Western Europe, Oceania, and Southern Asia are the top-performing regions in terms of sales.")
print("- The heatmap reveals that 'Technology' products contribute significantly to sales across most regions, while the performance of
```

```
Key Insights about GlobalMart's Sales Performance:
- Technology and Furniture are the top-selling product categories, significantly contributing to total sales.
- Sales show a clear upward trend over time, with notable peaks towards the end of each year.
- There is a positive correlation between Sales and Profit (0.53), suggesting that higher sales generally lead to higher profits, a
- Western Europe, Oceania, and Southern Asia are the top-performing regions in terms of sales.
- The heatmap reveals that 'Technology' products contribute significantly to sales across most regions, while the performance of 'Fu
```

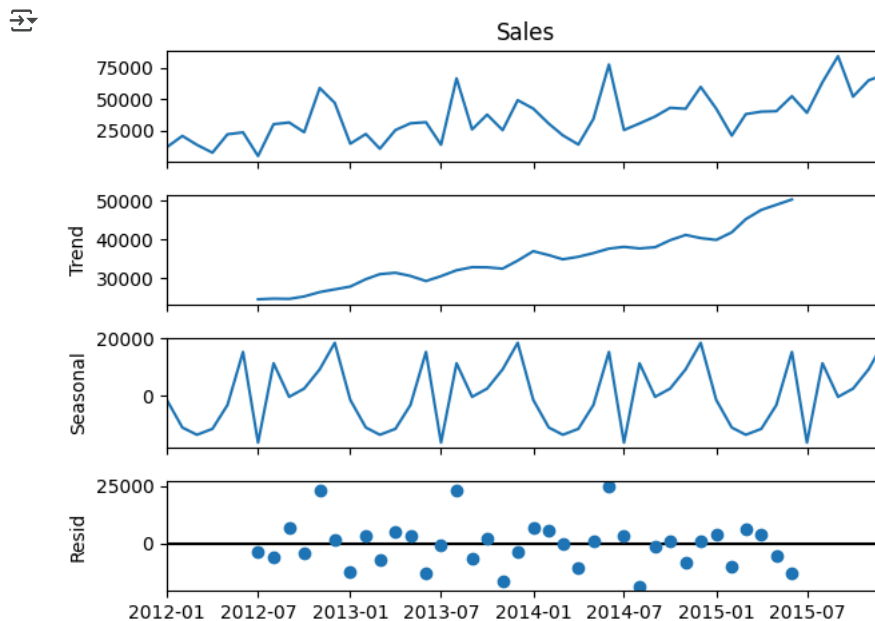
Bonus Challenge

Perform a simple time series analysis to identify any sales trends or seasonality

```
from statsmodels.tsa.seasonal import seasonal_decompose

# Perform seasonal decomposition
decomposition = seasonal_decompose(monthly_sales, model='additive')

# Plot the decomposed components
fig = decomposition.plot()
plt.show()
```



Prepare a Summary

Write a brief report (300-500 words) summarizing your findings and recommendations for GlobalMart

Start coding or [generate](#) with AI.

✓ GlobalMart Sales Performance Summary and Recommendations

Based on the analysis of the GlobalMart sales data, several key insights have been identified, along with recommendations for improving sales performance.

Key Findings:

- 1. Top-Performing Categories:** Technology and Furniture are the leading product categories in terms of total sales, contributing significantly to GlobalMart's revenue. Office Supplies, while still a substantial contributor, has lower sales compared to the other two categories.