Superstore Dataset - Exploratory and Descriptive Analysis

In this notebook, I carry out an in-depth exploratory and descriptive analysis of the Superstore Dataset, a widely used dataset for understanding sales performance, customer behavior, and profitability based on various transactional and demographic attributes.

This phase of analysis is essential for uncovering sales trends, identifying key performance drivers, and gaining intuition about the dataset's structure before applying any forecasting or optimization procedures. I examine the distribution of key numerical and categorical variables, investigate relationships between product features, customer segments, and geographical regions with sales and profit levels, and use visualizations to summarize insights. Particular focus is placed on sales and profit disparities across *product categories, **customer segments,** geographical regions, and **shipping modes*, helping lay a solid foundation for downstream business intelligence and strategic decision-making.

I begin my analysis by importing the core Python libraries required for *data handling, **numerical computation,** visualization, and **directory management*:

- pandas: Enables efficient manipulation, filtering, and aggregation of structured tabular data, forming the backbone of my analysis pipeline.
- numpy: Provides support for fast numerical operations, array-based computation, and statistical routines.
- os: Facilitates interaction with the file system, allowing me to construct flexible and portable directory paths for data and output management.
- plotly.express: A high-level graphing library that enables the creation of interactive, publication-quality visualizations, which I use extensively to uncover patterns and present insights throughout the notebook.
- plotly.io: Provides functions for input/output operations with Plotly figures, including setting default renderers for display.

I also set `pio.renderers.default = 'notebook'` to ensure that Plotly figures are rendered correctly and interactively within the notebook environment.

```python
# Import libraries
import pandas as pd
import numpy as np
import os
import plotly.express as px
```

```python
import plotly.io as pio
pio.renderers.default = 'notebook'
```

Define and Create Directory Paths

To ensure reproducibility and organized storage, I programmatically create directories if they don't already exist for:

- **raw data**
- **processed data**
- **results**
- **documentation**

These directories will store intermediate and final outputs for reproducibility.

```python
# Get current working directory
current_dir = os.getcwd()

# Go one directory up (assuming script is inside a subfolder like 'notebooks')
project_root_dir = os.path.dirname(current_dir)

# Define key folder paths
data_dir = os.path.join(project_root_dir, 'data')
raw_dir = os.path.join(data_dir, 'raw')
processed_dir = os.path.join(data_dir, 'processed')
results_dir = os.path.join(project_root_dir, 'results')
docs_dir = os.path.join(project_root_dir, 'docs')

# Create directories if they don't exist
os.makedirs(raw_dir, exist_ok=True)
os.makedirs(processed_dir, exist_ok=True)
os.makedirs(results_dir, exist_ok=True)
os.makedirs(docs_dir, exist_ok=True)
```

Loading the Cleaned Dataset

I load the cleaned version of the Superstore Dataset from the processed data directory into a Pandas DataFrame. The `head(10)` function shows the first ten records, giving a glimpse into the data columns such as `Order ID`, `Product Name`, `Sales`, etc.

```
superstore_data_filename = os.path.join(processed_dir, "final_superstore_cleaned.csv")

# Read in the superstore data
superstore_df = pd.read_csv(superstore_data_filename)

# Display the first 10 rows of the Superstore dataset
print(superstore_df.head(10))
```

```
   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
5       6  CA-2014-115812  2014-06-09  2014-06-14  Standard Class    BH-11710
6       7  CA-2014-115812  2014-06-09  2014-06-14  Standard Class    BH-11710
7       8  CA-2014-115812  2014-06-09  2014-06-14  Standard Class    BH-11710
8       9  CA-2014-115812  2014-06-09  2014-06-14  Standard Class    BH-11710
9      10  CA-2014-115812  2014-06-09  2014-06-14  Standard Class    BH-11710

     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  ...
5  Brosina Hoffman   Consumer  United States      Los Angeles  ...
6  Brosina Hoffman   Consumer  United States      Los Angeles  ...
7  Brosina Hoffman   Consumer  United States      Los Angeles  ...
8  Brosina Hoffman   Consumer  United States      Los Angeles  ...
9  Brosina Hoffman   Consumer  United States      Los Angeles  ...

                                        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
5  Eldon Expressions Wood and Plastic Desk Access...   48.8600        7
6                                         Newell 322    7.2800        4
7                     Mitel 5320 IP Phone VoIP phone  907.1520        6
8  DXL Angle-View Binders with Locking Rings by S...   18.5040        3
9                   Belkin F5C206VTEL 6 Outlet Surge  114.9000        5

  Discount    Profit Returned             Person  Shipping Duration  \
0     0.00   41.9136       No  Cassandra Brandow                  3
1     0.00  219.5820       No  Cassandra Brandow                  3
2     0.00    6.8714       No      Anna Andreadi                  4
3     0.45 -383.0310       No  Cassandra Brandow                  7
4     0.20    2.5164       No  Cassandra Brandow                  7
5     0.00   14.1694       No      Anna Andreadi                  5
6     0.00    1.9656       No      Anna Andreadi                  5
7     0.20   90.7152       No      Anna Andreadi                  5
8     0.20    5.7825       No      Anna Andreadi                  5
9     0.00   34.4700       No      Anna Andreadi                  5

   Order Year  Order Month
0        2016           11
1        2016           11
2        2016            6
3        2015           10
4        2015           10
5        2014            6
6        2014            6
7        2014            6
8        2014            6
9        2014            6

[10 rows x 26 columns]
```

Dataset Dimensions and Data Types

Here, I examine the structure of the dataset:

- There are *9,994* entries and *26* variables.
- The dataset includes both **numerical** (e.g., `Sales`, `Profit`, `Quantity`) and **categorical** variables (e.g., `Category`, `Region`, `Segment`).

Understanding data types and null entries is essential before proceeding with analysis.

```
superstore_df.shape
```

```
(9994, 26)
```

```
superstore_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 26 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Row ID             9994 non-null   int64
 1   Order ID           9994 non-null   object
 2   Order Date         9994 non-null   object
 3   Ship Date          9994 non-null   object
 4   Ship Mode          9994 non-null   object
 5   Customer ID        9994 non-null   object
 6   Customer Name      9994 non-null   object
 7   Segment            9994 non-null   object
 8   Country            9994 non-null   object
 9   City               9994 non-null   object
 10  State              9994 non-null   object
 11  Postal Code        9994 non-null   int64
 12  Region             9994 non-null   object
 13  Product ID         9994 non-null   object
 14  Category           9994 non-null   object
 15  Sub-Category       9994 non-null   object
 16  Product Name       9994 non-null   object
 17  Sales              9994 non-null   float64
 18  Quantity           9994 non-null   int64
 19  Discount           9994 non-null   float64
 20  Profit             9994 non-null   float64
 21  Returned           9994 non-null   object
 22  Person             9994 non-null   object
 23  Shipping Duration  9994 non-null   int64
 24  Order Year         9994 non-null   int64
 25  Order Month        9994 non-null   int64
dtypes: float64(3), int64(6), object(17)
memory usage: 2.0+ MB
```

Summary Statistics: Numerical Variables

This summary provides a snapshot of key distribution characteristics. I see that:

- **Sales** values vary widely, from very small amounts (e.g., $0.44) to substantial transactions (up to $22,638). The mean sales value ($229.86) is significantly higher than the median ($54.49), indicating a strong positive skew. This suggests that while most transactions are for smaller amounts, a few high-value sales contribute disproportionately to the total revenue.
- **Quantity** of items per order typically ranges from 1 to 14, with an average of about 3.8 items. The median quantity is 3, suggesting that most customers purchase a small number of items per transaction.
- **Discount** percentages are applied across a broad spectrum, from 0% to 80%. A notable observation is that the median and 75th percentile are both 20%, indicating that a 20% discount is a very common promotional strategy. The presence of 0% discounts suggests many items are sold at full price.
- **Profit** shows a wide distribution, ranging from significant losses (down to -$6,600) to substantial gains (up to $8,400). The mean profit ($28.66) is higher than the median ($8.67), implying that a few highly profitable sales positively skew the overall average, despite the occurrence of numerous loss-making transactions. This highlights the importance of analyzing factors contributing to both high profits and losses.
- **Shipping Duration** typically ranges from 0 to 7 days, with an average and median of approximately 4 days, indicating that most orders are delivered within a week.

```
superstore_df.describe()
```

	Row ID	Postal Code	Sales	Quantity	Discount	Profit	Shipping Du
count	9994.000000	9994.000000	9994.000000	9994.000000	9994.000000	9994.000000	9994.000000
mean	4997.500000	55190.379428	229.858001	3.789574	0.156203	28.656896	3.958175
std	2885.163629	32063.693350	623.245101	2.225110	0.206452	234.260108	1.747567
min	1.000000	1040.000000	0.444000	1.000000	0.000000	-6599.978000	0.000000
25%	2499.250000	23223.000000	17.280000	2.000000	0.000000	1.728750	3.000000
50%	4997.500000	56430.500000	54.490000	3.000000	0.200000	8.666500	4.000000
75%	7495.750000	90008.000000	209.940000	5.000000	0.200000	29.364000	5.000000
max	9994.000000	99301.000000	22638.480000	14.000000	0.800000	8399.976000	7.000000

Summary Statistics: Categorical Variables

This summary provides insights into the distribution and most frequent categories for the object (categorical) variables in the dataset:

- **Ship Mode**: 'Standard Class' is by far the most common shipping method, accounting for nearly 60% of all orders (5968 out of 9994). This suggests a preference for cost-effective shipping or that faster options are less frequently utilized.
- **Segment**: The 'Consumer' segment represents the largest customer base, making up over half of the orders (5191 entries). This indicates that individual consumers are the primary drivers of sales, followed by Corporate and Home Office segments.
- **Country**: The dataset is entirely focused on the 'United States', with all 9994 entries originating from this country. This confirms the geographical scope of the Superstore operations captured in this data.
- **Region**: The 'West' region has the highest number of orders (3203 entries), indicating it is the most active sales region, followed by East, Central, and South.
- **Category**: 'Office Supplies' is the dominant product category, accounting for over 60% of all transactions (6026 entries). This highlights its central role in the Superstore's product offerings, followed by 'Furniture' and 'Technology'.
- **Sub-Category**: Within 'Office Supplies', 'Binders' is the most frequently purchased sub-category (1523 entries), suggesting high demand for these items.
- **Returned**: The vast majority of orders are 'No' (9194 entries), indicating a very low return rate for products. This suggests high customer satisfaction or effective product quality control.

```python
superstore_df.describe(include='object')
```

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Seg
count	9994	9994	9994	9994	9994	9994	999
unique	5009	1237	1334	4	793	793	3
top	CA-2017-100111	2016-09-05	2015-12-16	Standard Class	WB-21850	William Brown	Con
freq	14	38	35	5968	37	37	519

Key Categorical Distributions

Understanding the distribution of key categorical variables provides crucial insights into the Superstore's operational landscape and customer base.

- **Region Distribution**: The dataset shows a clear geographical imbalance. The 'West' region leads in terms of order volume (3,203 entries), closely followed by 'East' (2,848 entries). 'Central' (2,323 entries) and 'South' (1,620 entries) have fewer transactions. This distribution is further reflected in total sales, where the 'West' ($725,457.82) and 'East' ($678,781.24) regions generate the highest revenues, indicating they are the primary sales drivers for the Superstore.
- **Category Distribution**: 'Office Supplies' is the most dominant product category, accounting for approximately 60.3% of all transactions (6,026 entries). This highlights its central role in the Superstore's inventory and customer purchases. 'Furniture' (21.2%) and 'Technology' (18.5%) make up the remaining significant portions, suggesting a diverse product offering but with a strong emphasis on office-related items.
- **Segment Distribution**: The 'Consumer' segment represents the largest customer base, comprising about 51.9% of all orders (5,191 entries). This indicates that individual customers are the primary focus of the Superstore's sales efforts. The 'Corporate' segment accounts for 30.2% (3,020 entries), and 'Home Office' for 17.8% (1,783 entries), showing a substantial presence in business-to-business and small office markets as well.

```python
# Analyze the distribution of 'Region' in the Superstore dataset
print("Distribution of 'Region' in Superstore dataset:")
print(superstore_df['Region'].value_counts())
```

```
Distribution of 'Region' in Superstore dataset:
West       3203
East       2848
Central    2323
South      1620
Name: Region, dtype: int64
```

```python
# Analyze the normalized distribution of 'Category' in the Superstore dataset
print("\nNormalized distribution of 'Category' in Superstore dataset:")
print(superstore_df['Category'].value_counts(normalize=True))
```

```
Normalized distribution of 'Category' in Superstore dataset:
Office Supplies    0.602962
```

```
Furniture          0.212227
Technology         0.184811
Name: Category, dtype: float64
```

```
# Analyze the normalized distribution of 'Segment' in the Superstore dataset
print("\nNormalized distribution of 'Segment' in Superstore dataset:")
print(superstore_df['Segment'].value_counts(normalize=True))
```

```
Normalized distribution of 'Segment' in Superstore dataset:
Consumer       0.519412
Corporate      0.302181
Home Office    0.178407
Name: Segment, dtype: float64
```

```
# Analyze the distribution of 'Region'
print("\nDistribution of 'Region':")
print(superstore_df['Region'].value_counts())
```

```
Distribution of 'Region':
West       3203
East       2848
Central    2323
South      1620
Name: Region, dtype: int64
```

```
# Analyze the distribution of 'Category'
print("\nDistribution of 'Category':")
print(superstore_df['Category'].value_counts())
```

```
Distribution of 'Category':
Office Supplies    6026
Furniture          2121
Technology         1847
Name: Category, dtype: int64
```

```
# Analyze the distribution of 'Segment'
print("\nDistribution of 'Segment':")
print(superstore_df['Segment'].value_counts())
```

```
Distribution of 'Segment':
Consumer       5191
Corporate      3020
Home Office    1783
Name: Segment, dtype: int64
```

```
# Sales Distribution by Region
superstore_sales_region = superstore_df.groupby('Region')['Sales'].sum().reset_index()
print("\nTotal Sales by Region:")
print(superstore_sales_region)
```

```
Total Sales by Region:
    Region        Sales
0  Central  501239.8908
1     East  678781.2400
2    South  391721.9050
3     West  725457.8245
```

Sales Trend Analysis: Monthly Performance

To understand the Superstore's sales seasonality, I analyzed the total sales aggregated by month across all years. This involved converting the 'Order Date' column to a datetime format, extracting the month name, and then grouping the total 'Sales' by these months, ensuring they are ordered chronologically.

Total Sales by Month:

The analysis reveals a distinct seasonal pattern in sales:

- Sales are relatively low at the beginning of the year, with **January** ($94,924) and **February** ($59,751) showing the lowest figures. February stands out as the weakest sales month.
- There's a significant rebound in **March** ($205,005), indicating the start of a stronger sales period.
- Sales remain moderate through the spring and summer months (April to August), generally fluctuating between $137,000 and $159,000.

- A sharp increase is observed in **September** ($307,649), marking the beginning of the peak sales season.
- The highest sales volumes occur towards the end of the year, with **November** ($352,461) being the strongest month, followed closely by **December** ($325,293). This suggests a strong holiday shopping or year-end purchasing trend.

```
# 1. Ensure 'Order Date' is in datetime format
superstore_df['Order Date'] = pd.to_datetime(superstore_df['Order Date'])

# 2. Create 'Order Month' column (as full month name)
superstore_df['Order Month'] = superstore_df['Order Date'].dt.month_name()

# 3. Define month order to keep correct sequence
month_order = ['January', 'February', 'March', 'April', 'May', 'June',
               'July', 'August', 'September', 'October', 'November', 'December']

# 4. Group total sales by month
superstore_sales_month = superstore_df.groupby('Order Month')['Sales'].sum().reindex(month_o

# 5. Print result
print("Total Sales by Month:")
print(superstore_sales_month)

# 6. Plot line chart
fig = px.line(superstore_sales_month,
              x='Order Month',
              y='Sales',
              markers=True,
              title='Total Sales Trend by Month',
              height=600,
              color_discrete_sequence=[px.colors.sequential.Blues[5]])

# 7. Customize layout
fig.update_layout(template="presentation",
                  paper_bgcolor="rgba(0,0,0,0)",
                  plot_bgcolor="rgba(0,0,0,0)",
                  xaxis_title='Month',
                  yaxis_title='Total Sales')

# 8. Show figure
fig.show()

# 9. Save the visual
```

```python
fig.write_image(os.path.join(results_dir, 'total_sales_by_month_all_years_line_chart.jpg'))
fig.write_image(os.path.join(results_dir, 'total_sales_by_month_all_years_line_chart.png'))
fig.write_html(os.path.join(results_dir, 'total_sales_by_month_all_years_line_chart.html'))
```

```
Total Sales by Month:
   Order Month        Sales
0      January   94924.8356
1     February   59751.2514
2        March  205005.4888
3        April  137762.1286
4          May  155028.8117
5         June  152718.6793
6         July  147238.0970
7       August  159044.0630
8    September  307649.9457
9      October  200322.9847
10    November  352461.0710
11    December  325293.5035
```

```
Unable to display output for mime type(s): text/html
```

```
Unable to display output for mime type(s): text/html
```

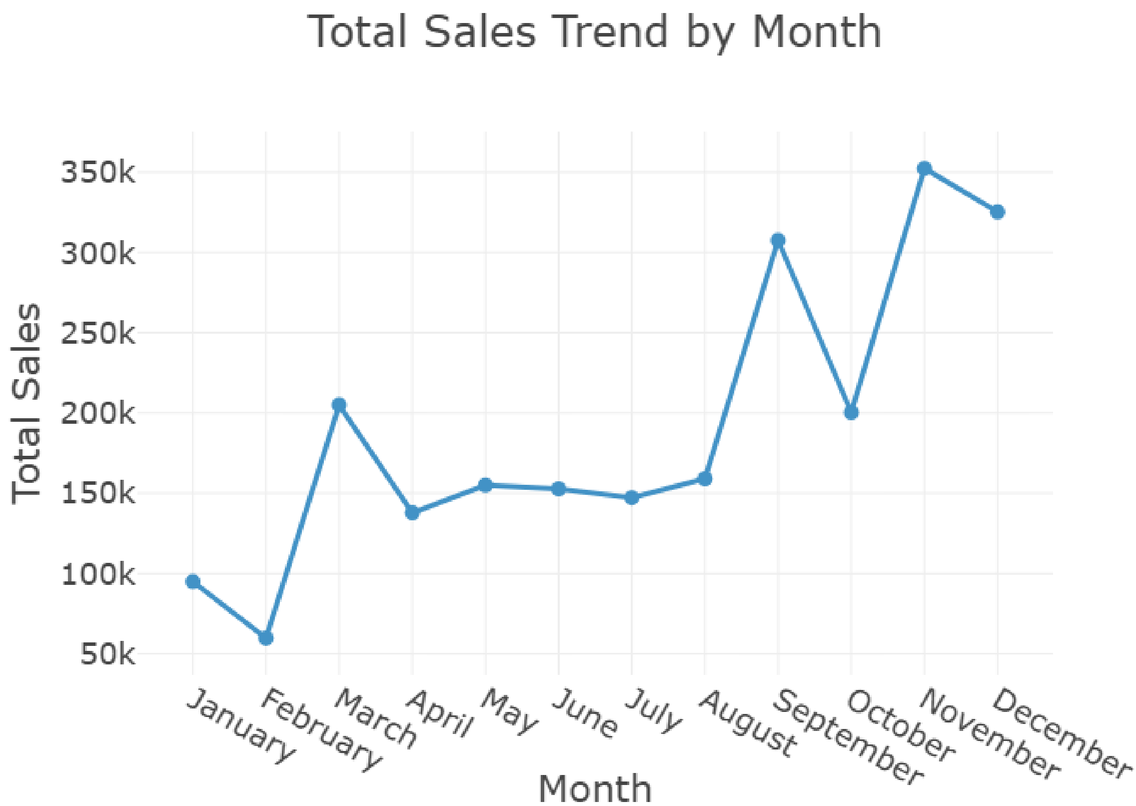


Figure 1: Total Sales Trend by Month

This monthly trend is visually represented using an interactive line chart, which clearly illustrates the fluctuations and highlights the peak sales periods, providing valuable insights for inventory management and marketing strategies.

Quantity Distribution by Segment

To understand which customer segments purchase the most products, we aggregated the total 'Quantity' sold by 'Segment'.

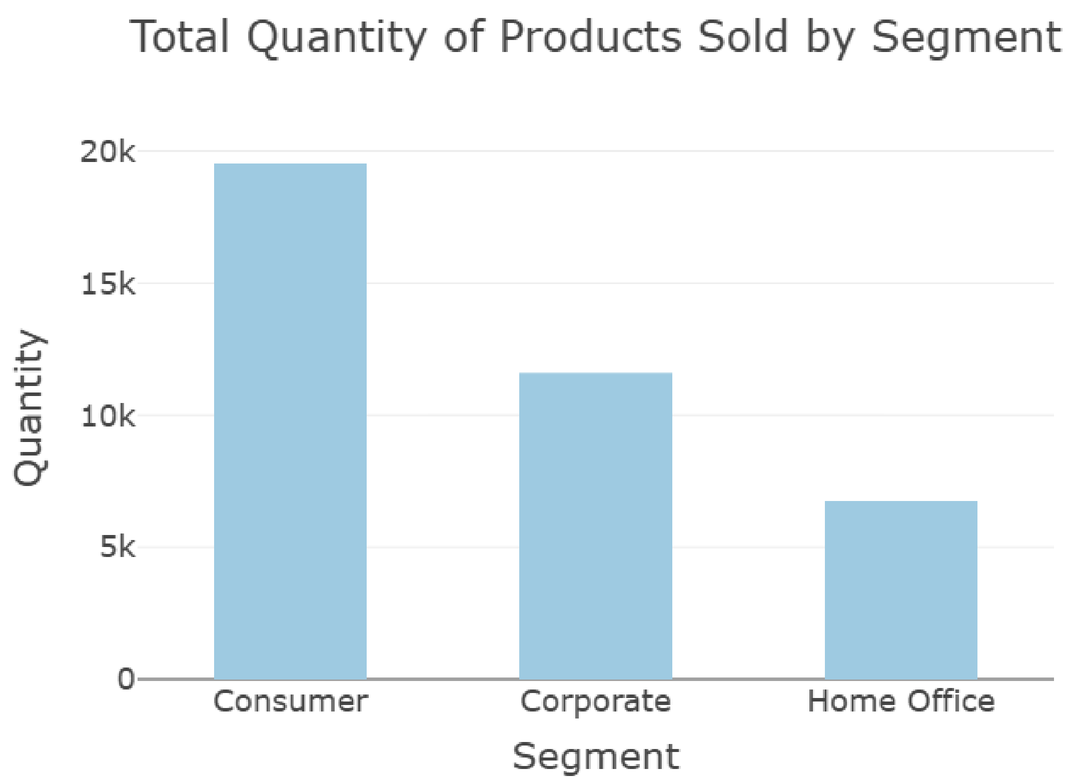
Key Findings:

- The **Consumer** segment accounts for the largest volume of products sold, with a total of 19,521 units.
- The **Corporate** segment follows, purchasing 11,608 units.
- The **Home Office** segment has the lowest quantity sold, with 6,744 units.

```
# Calculate total quantity by Segment
superstore_quantity_segment = superstore_df.groupby('Segment')['Quantity'].sum().reset_index
print("Total Quantity by Segment:")
print(superstore_quantity_segment)

fig = px.bar(superstore_quantity_segment,
             x='Segment',
             y='Quantity',
             title='Total Quantity of Products Sold by Segment',
             height=600,
             color_discrete_sequence=[px.colors.sequential.Blues[3]]
            )


fig.update_layout(template="presentation",
                  paper_bgcolor="rgba(0,0,0,0)",
                  plot_bgcolor="rgba(0,0,0,0)")

# Reduce the width of the bars
fig.update_traces(width=0.5)

# Display the figure
fig.show()

# Save the figure to various formats with a new filename to reflect the color change
fig.write_image(os.path.join(results_dir, 'quantity_by_segment_light_blue_narrow_bar_chart.jp
fig.write_image(os.path.join(results_dir, 'quantity_by_segment_light_blue_narrow_bar_chart.pr
fig.write_html(os.path.join(results_dir, 'quantity_by_segment_light_blue_narrow_bar_chart.htm
```

```
Total Quantity by Segment:
       Segment  Quantity
0     Consumer     19521
1    Corporate     11608
2  Home Office      6744
```

```
Unable to display output for mime type(s): text/html
```



This distribution, visualized through a bar chart, clearly indicates that individual consumers are the primary drivers of product volume for the Superstore, significantly out-purchasing both corporate and home office clients.

Profit by Category

To assess the profitability of different product lines, we calculated the total 'Profit' generated by each 'Category'.

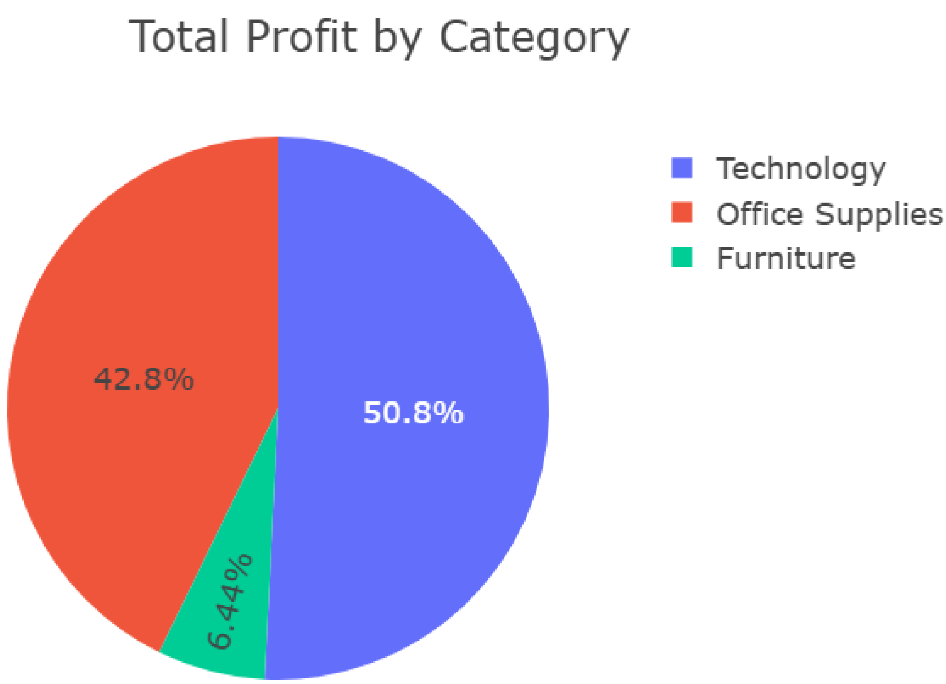
Key Findings:

- **Technology** is the most profitable category, contributing the largest share of total profit at **$145,454.95**, representing approximately **50.8%** of the overall profit.
- **Office Supplies** is the second most profitable, with **$122,490.80** in profit, accounting for about **42.8%**.
- **Furniture** is significantly less profitable, generating only **$18,451.27**, which is a mere **6.44%** of the total profit. This suggests that while furniture might have high sales values, its profit margins are considerably lower, or it incurs higher costs/losses.

```
# Calculate Total Profit by Category
superstore_profit_category = superstore_df.groupby('Category')['Profit'].sum().reset_index()
print("Total Profit by Category:")
print(superstore_profit_category)

# Create the pie chart with different colors
fig = px.pie(superstore_profit_category,
             names='Category',
             values='Profit',
             title='Total Profit by Category',
             color_discrete_sequence=px.colors.qualitative.Plotly)

fig.update_layout(template="presentation",
                  paper_bgcolor="rgba(0,0,0,0)",
                  plot_bgcolor="rgba(0,0,0,0)")

# Display the figure
fig.show()

# Save the figure to various formats
fig.write_image(os.path.join(results_dir, 'profit_by_category_mixed_colors_pie_chart.jpg'))
fig.write_image(os.path.join(results_dir, 'profit_by_category_mixed_colors_pie_chart.png'))
fig.write_html(os.path.join(results_dir, 'profit_by_category_mixed_colors_pie_chart.html'))
```

```
Total Profit by Category:
          Category       Profit
0        Furniture   18451.2728
1  Office Supplies  122490.8008
2       Technology  145454.9481
```

```
Unable to display output for mime type(s): text/html
```



This distribution, clearly illustrated by a pie chart, highlights that Technology and Office Supplies are the primary drivers of the Superstore's profitability, while Furniture contributes minimally.

Sales and Profit by Sub-Category

This section analyzes the sales and profit performance across the top 10 product sub-categories, sorted by their total sales. The bar chart visually compares the sales and profit for each of these sub-categories.

```
# Calculate Sales and Profit by Sub-Category
superstore_sub_category = superstore_df.groupby('Sub-Category')[['Sales', 'Profit']].sum().r
superstore_sub_category = superstore_sub_category.sort_values(by='Sales', ascending=False)

# Filter for only the top 10 sub-categories
superstore_sub_category_top10 = superstore_sub_category.head(10)

print("Top 10 Sales and Profit by Sub-Category (Sorted by Sales):")
```

```
print(superstore_sub_category_top10)

# Create the bar chart for Top 10 Sales and Profit by Sub-Category
fig = px.bar(superstore_sub_category_top10,
             x='Sub-Category',
             y=['Sales', 'Profit'],
             title='Top 10 Sales and Profit by Product Sub-Category',
             barmode='group',
             height=600,
             color_discrete_sequence=[px.colors.sequential.Blues[3], px.colors.sequential.Blu
             text_auto=True
            )

# Update layout for presentation and clarity
fig.update_layout(
    template="presentation",
    xaxis_title='Product Sub-Category',
    yaxis_title='Amount ($)',
    legend_title=dict(text='Metric'),
    paper_bgcolor="rgba(0,0,0,0)",
    plot_bgcolor="rgba(0,0,0,0)",
    xaxis=dict(tickangle=45, title_font=dict(size=14), tickfont=dict(size=12)),
    yaxis=dict(title_font=dict(size=14), tickfont=dict(size=12)),
    title_font_size=18
)


fig.update_traces(width=0.5)

# Display the figure
fig.show()

# Save the figure to various formats with a new filename
fig.write_image(os.path.join(results_dir, 'sales_profit_by_subcategory_top10_bar_chart.jpg'))
fig.write_image(os.path.join(results_dir, 'sales_profit_by_subcategory_top10_bar_chart.png'))
fig.write_html(os.path.join(results_dir, 'sales_profit_by_subcategory_top10_bar_chart.html'))
```

```
Top 10 Sales and Profit by Sub-Category (Sorted by Sales):
   Sub-Category        Sales      Profit
13       Phones  330007.0540  44515.7306
5        Chairs  328449.1030  26590.1663
14      Storage  223843.6080  21278.8264
```

```
16        Tables  206965.5320 -17725.4811
3        Binders  203412.7330  30221.7633
11      Machines  189238.6310   3384.7569
0    Accessories  167380.3180  41936.6357
6        Copiers  149528.0300  55617.8249
4      Bookcases  114879.9963  -3472.5560
1     Appliances  107532.1610  18138.0054
```

```
Unable to display output for mime type(s): text/html
```

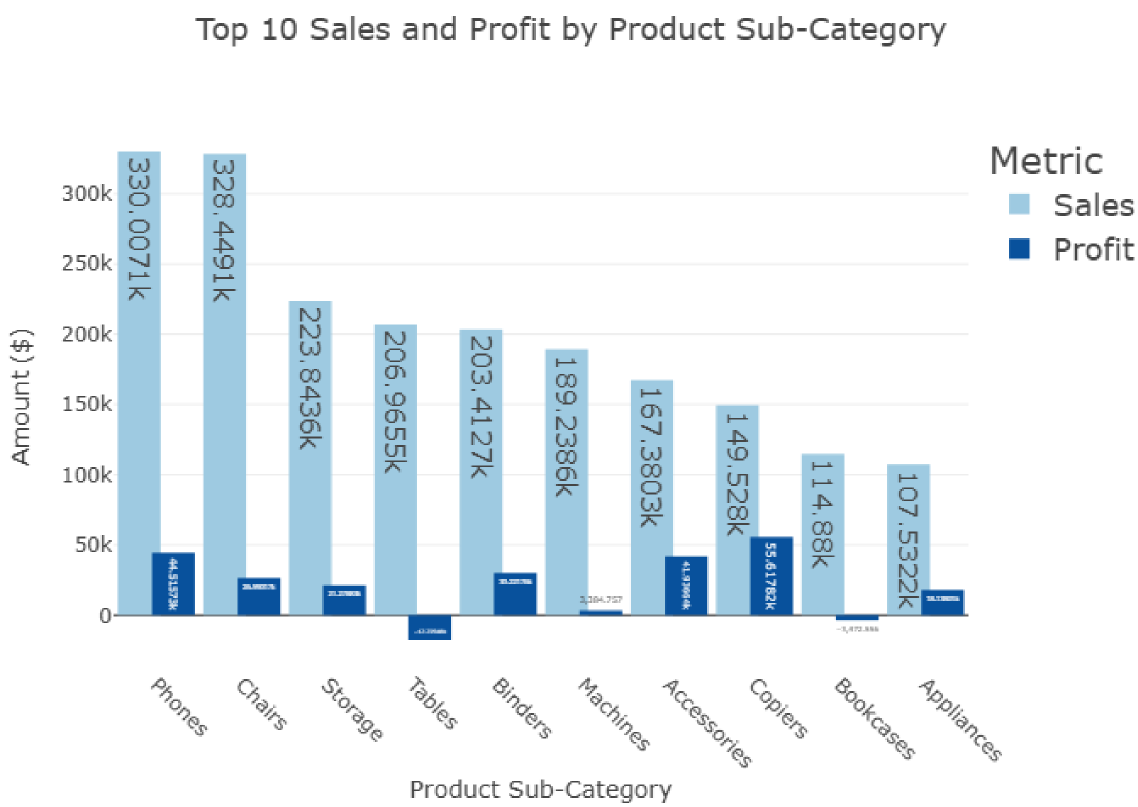


Figure 2: Top 10 Sales and Profit by Product Sub-Category

Key Observations:

Phones and Chairs are the top two sub-categories by sales, generating \$330,007.05 and \$328,449.10 respectively. Both are profitable, with Phones yielding \$44,515.73 in profit and Chairs \$26,590.17.

Copiers stand out as highly profitable, generating $55,617.82 in profit from sales of $149,528.03, indicating a strong profit margin.

Tables and Bookcases are significant concerns, as they are among the top 10 in sales ($206,965.53 and 114,879.99respectively)butincursubstantiallosses(-17,725.48 for Tables and $-3,472.56 for Bookcases). This highlights these sub-categories as major profit drains.

Other notable profitable sub-categories include Accessories ($41,936.64 profit) and Binders ($30,221.76 profit).

Sales by Region

This section examines the total sales generated by each region, sorted from highest to lowest. The bar chart illustrates these regional sales contributions.

Key Observations:

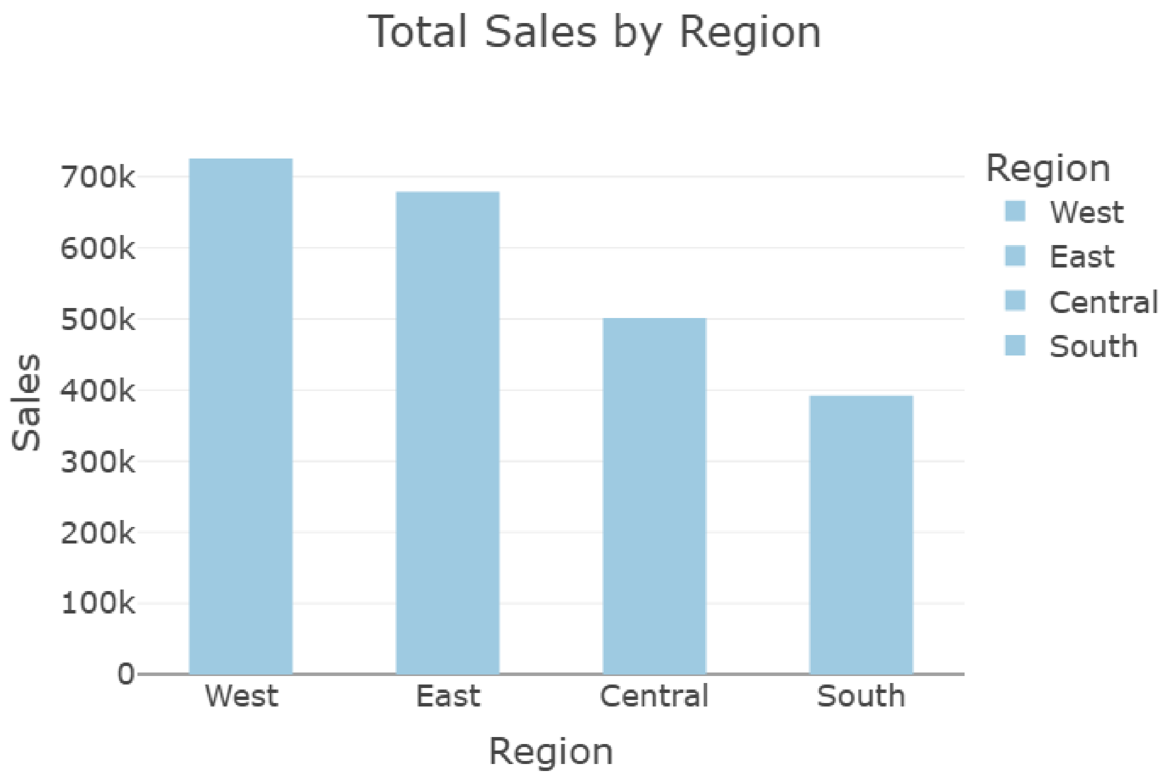
- The West region leads in total sales with $725,457.82.
- The East region follows closely with $678,781.24 in sales.
- The Central region generated $501,239.89 in sales.
- The South region recorded the lowest sales at $391,721.90.

```
superstore_sales_region = superstore_df.groupby('Region')['Sales'].sum().reset_index()

# Sort the data from high to low (descending) by Sales
superstore_sales_region = superstore_sales_region.sort_values(by='Sales', ascending=False)
print("Total Sales by Region (Sorted Descending):")
print(superstore_sales_region)

# Create the bar chart
fig = px.bar(superstore_sales_region,
             x='Region',
             y='Sales',
             title='Total Sales by Region',
             height= 600,
             color='Region',
             color_discrete_sequence=[px.colors.sequential.Blues[3]]
             )


fig.update_layout(template="presentation",
                  paper_bgcolor="rgba(0,0,0,0)",
```

```python
                  plot_bgcolor="rgba(0,0,0,0)")

fig.update_traces(width=0.5)

# Display the figure
fig.show()

# Save the figure to various formats
fig.write_image(os.path.join(results_dir, 'sales_by_region_sorted_bar_chart.jpg'))
fig.write_image(os.path.join(results_dir, 'sales_by_region_sorted_bar_chart.png'))
fig.write_html(os.path.join(results_dir, 'sales_by_region_sorted_bar_chart.html'))
```

```
Total Sales by Region (Sorted Descending):
    Region        Sales
3     West  725457.8245
1     East  678781.2400
0  Central  501239.8908
2    South  391721.9050
```

```
Unable to display output for mime type(s): text/html
```



This distribution highlights the varying sales performance across the Superstore's operational regions, with the West and East being the primary revenue drivers.