WALMART STORES SALES ANALYSIS

In [3]:
```
import numpy as np
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go
import plotly.io as pio
import plotly.colors as colors
pio.templates.default="plotly_white"
```

In [5]:
```
data=pd.read_csv('ecommerce.csv',encoding='latin-1')
```

In [7]:
```
data.head()
```

Out[7]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Posta Cod
0	1	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	4242
1	2	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	4242
2	3	CA-2016-138688	6/12/2016	6/16/2016	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	...	9003
3	4	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	3331
4	5	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	3331

5 rows × 21 columns

In [9]:
```
data.describe()
```

Out[9]:

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

In [11]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 21 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
dtypes: float64(3), int64(3), object(15)
memory usage: 1.6+ MB
```

In [13]:
```
data.isnull().sum()
```

Out[13]:

```
Row ID           0
Order ID         0
Order Date       0
Ship Date        0
Ship Mode        0
Customer ID      0
Customer Name    0
Segment          0
Country          0
City             0
State            0
Postal Code      0
Region           0
Product ID       0
Category         0
Sub-Category     0
Product Name     0
Sales            0
Quantity         0
Discount         0
Profit           0
dtype: int64
```

In [15]:
```
data['Order Date']=pd.to_datetime(data['Order Date'])
data['Ship Date']=pd.to_datetime(data['Ship Date'])
```

In [17]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 21 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   datetime64[ns]
 3   Ship Date      9994 non-null   datetime64[ns]
 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       
dtypes: datetime64[ns](2), float64(3), int64(3), object(13)
memory usage: 1.6+ MB
```

In [19]:
```
data['Order Month']=data['Order Date'].dt.month
data['Order Year']=data['Order Date'].dt.year
data['Order Day of Week']=data['Order Date'].dt.dayofweek
```

In [21]:
```
data.head()
```

Out[21]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Category	Cat
0	1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	Furniture	Boo
1	2	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	Furniture	
2	3	CA-2016-138688	2016-06-12	2016-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	...	Office Supplies	
3	4	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	Furniture	
4	5	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	Office Supplies	S

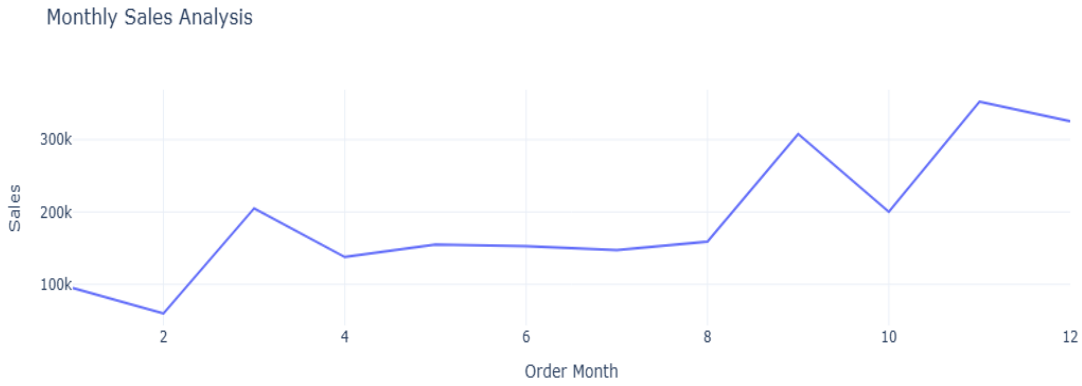
5 rows × 24 columns

Monthly sales analysis

In [23]:
```
monthly_sales=data.groupby('Order Month')['Sales'].sum()
print(monthly_sales)
```

```
Order Month
1      94924.8356
2      59751.2514
3     205005.4888
4     137762.1286
5     155028.8117
6     152718.6793
7     147238.0970
8     159044.0630
9     307649.9457
10    200322.9847
11    352461.0710
12    325293.5035
Name: Sales, dtype: float64
```

In [25]:
```
monthly_sales=data.groupby('Order Month')['Sales'].sum().reset_index()
fig=px.line(monthly_sales,
            x='Order Month',
            y='Sales',
            title='Monthly Sales Analysis')
fig.show()
```



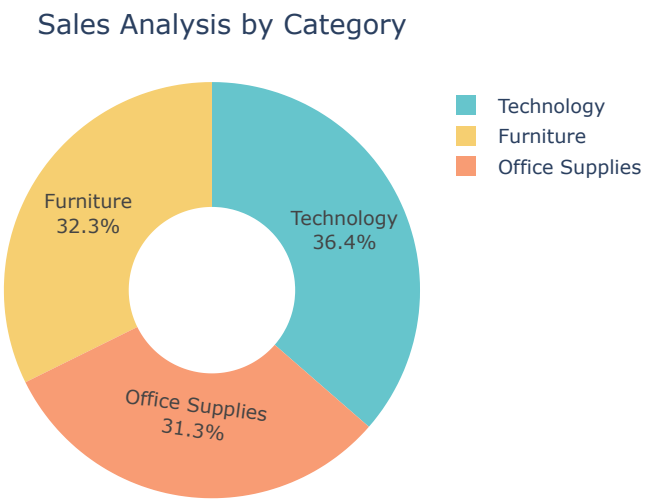
While there are some fluctuations, the general trend seems to be positive, with sales increasing over the year.* *There's a notable dip in sales early in the year (january & February), followed by a period of relative stability.* *The most significant growth occurs in the latter half of the year, with a peak around November

Sales Analysis by Category

In [27]:
```
cate=data.groupby('Category')['Sales'].sum().reset_index()
print(cate)
```

```
         Category        Sales
0       Furniture  741999.7953
1  Office Supplies  719047.0320
2      Technology  836154.0330
```

In [106…
```
fig=px.pie(cate,names='Category',values='Sales',hole=0.4,color_discrete_sequence=px.colors.qualitative.P
fig.update_layout(title_text='Sales Analysis by Category',
                  title_x=0.44,width=500,height=400)
fig.update_traces(textposition='inside',textinfo='percent+label')
fig.show()
```



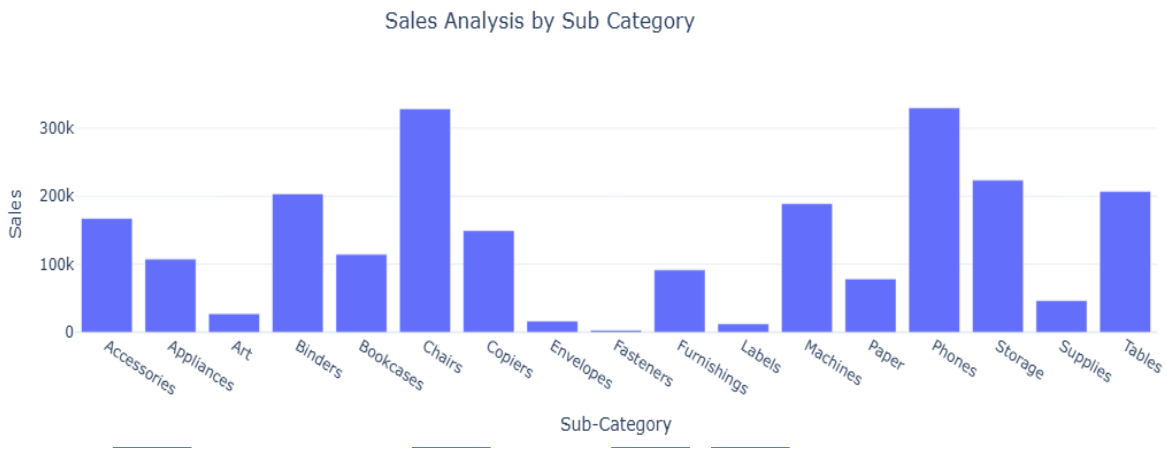
Technology products are the highest performing category, accounting for 36.4% of overall sales.* *Furniture and Office Supplies have similar sales contributions, with 32.3% and 31.3%, respectively.* *The sales distribution across the three categories is relatively balanced. No single category overwhelmingly dominates the others, suggesting a diversified product offering.

Sales Analysis by Sub Category

In [33]:
```
sub_cate=data.groupby('Sub-Category')['Sales'].sum().reset_index()
print(sub_cate)
```

```
   Sub-Category        Sales
0   Accessories  167380.3180
1    Appliances  107532.1610
2           Art   27118.7920
3       Binders  203412.7330
4     Bookcases  114879.9963
5        Chairs  328449.1030
6       Copiers  149528.0300
7     Envelopes   16476.4020
8     Fasteners    3024.2800
9   Furnishings   91705.1640
10       Labels   12486.3120
11     Machines  189238.6310
12        Paper   78479.2060
13       Phones  330007.0540
14      Storage  223843.6080
15     Supplies   46673.5380
16       Tables  206965.5320
```

In [35]:
```
fig=px.bar(sub_cate,x='Sub-Category',y='Sales',title='Sales Analysis by Sub Category')
fig.update_layout(title_x=0.44)
fig.show()
```



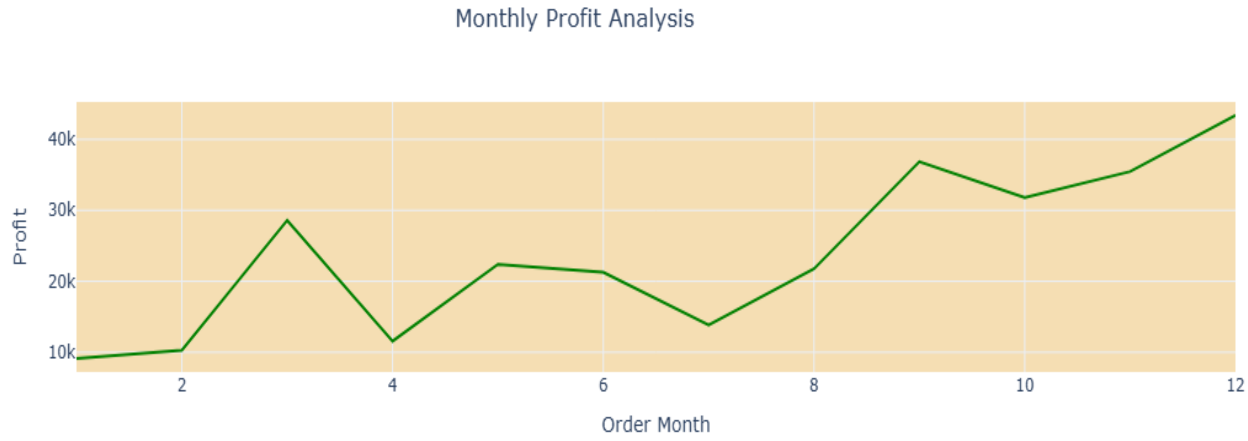
The sub-categories with the highest sales are Chairs and Phones, both exceeding 200k in sales. This indicates that these two product categories are the top performers in terms of sales.

Monthly Profit Analysis

In [37]:
```
monthly_profit=data.groupby('Order Month')['Profit'].sum()
print(monthly_profit)
```

```
Order Month
1      9134.4461
2     10294.6107
3     28594.6872
4     11587.4363
5     22411.3078
6     21285.7954
7     13832.6648
8     21776.9384
9     36857.4753
10    31784.0413
11    35468.4265
12    43369.1919
Name: Profit, dtype: float64
```

In [71]:
```
monthly_profit=data.groupby('Order Month')['Profit'].sum().reset_index()
fig=px.line(monthly_profit,
            x='Order Month',
            y='Profit',
            title='Monthly Profit Analysis')
fig.update_traces(line_color='green')
fig.update_layout(
    plot_bgcolor='wheat',title_x=0.44)
fig.show()
```

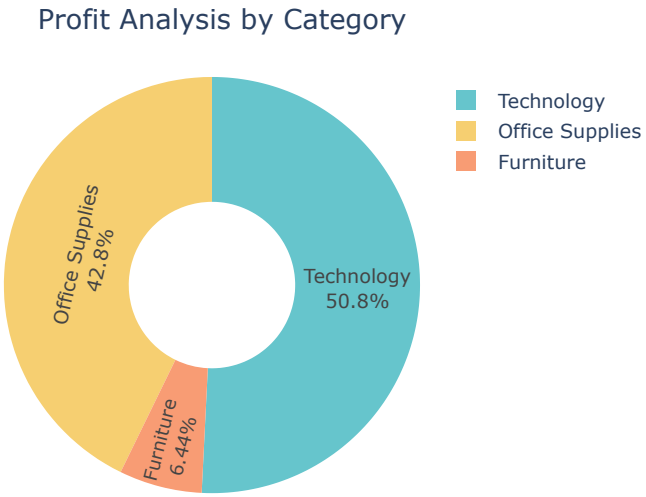


Slow early-year growth: Profit slowly increase from January to February *Mid-year fluctuations: Profit stability with some fluctuations from March to July.* *Late-year surge: Noticeable profit increase from August to December, with a peak in December.*

Profit Analysis by Category

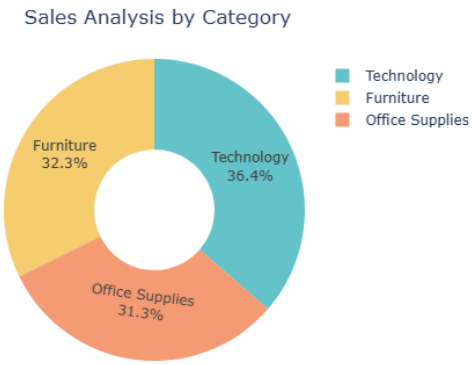
In [102…

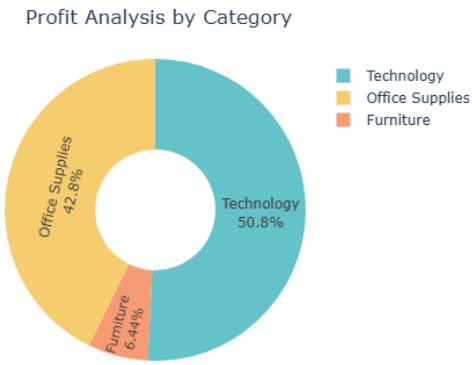
```
fig=px.pie(profit_category,names='Category',values='Profit',hole=0.4,color_discrete_sequence=px.colors.q
fig.update_layout(title_text='Profit Analysis by Category',
                  title_x=0.44,width=500,height=400)
fig.update_traces(textposition='inside',textinfo='percent+label')
fig.show()
```



Technology category has the highest profit share at 50.8% *Office Supplies account for 42.8% of the total profit, which is substantial* *Furniture contributes the least to profit at 6.4%.*

Comparative Analysis of Sales and Profit Across Categories





Technology and Office Supplies: These categories are key drivers of profitability, with technology leading in both sales and profit contributions.

Furniture: Despite having a significant share of sales, the profit margins in this category are much lower. This highlights a need to review pricing strategies or cost management for furniture products.

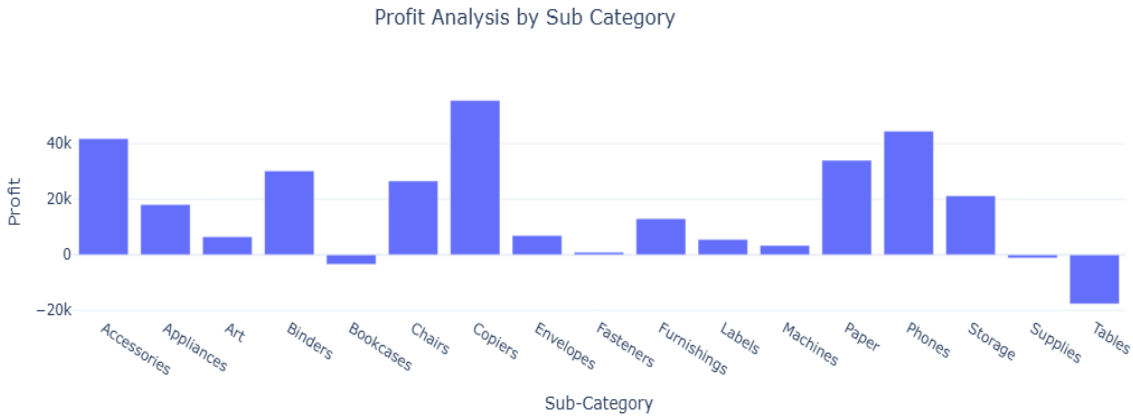
Profit Analysis by Sub Category

In [118…

```
sub_cate_profit=data.groupby('Sub-Category')['Profit'].sum().reset_index()
print(sub_cate_profit)

fig=px.bar(sub_cate_profit,x='Sub-Category',y='Profit',title='Profit Analysis by Sub Category')
fig.update_layout(title_x=0.44)
fig.show()
```

```
   Sub-Category      Profit
0   Accessories  41936.6357
1    Appliances  18138.0054
2           Art   6527.7870
3       Binders  30221.7633
4     Bookcases  -3472.5560
5        Chairs  26590.1663
6       Copiers  55617.8249
7     Envelopes   6964.1767
8     Fasteners    949.5182
9   Furnishings  13059.1436
10       Labels   5546.2540
11     Machines   3384.7569
12        Paper  34053.5693
13       Phones  44515.7306
14      Storage  21278.8264
15     Supplies  -1189.0995
16       Tables -17725.4811
```

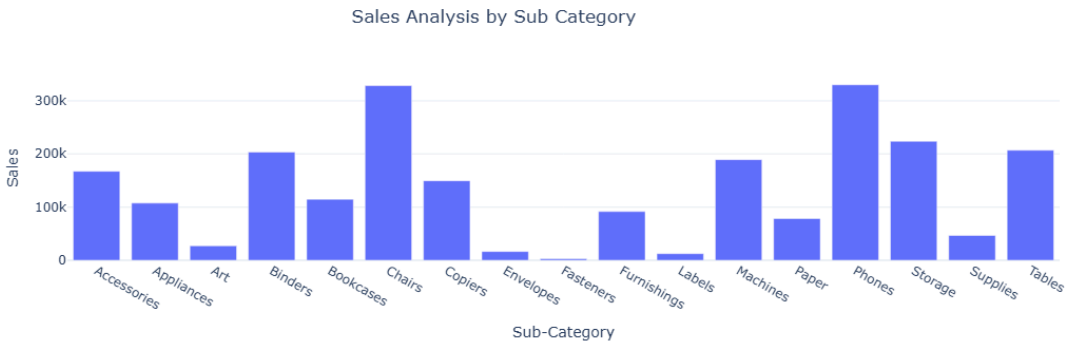


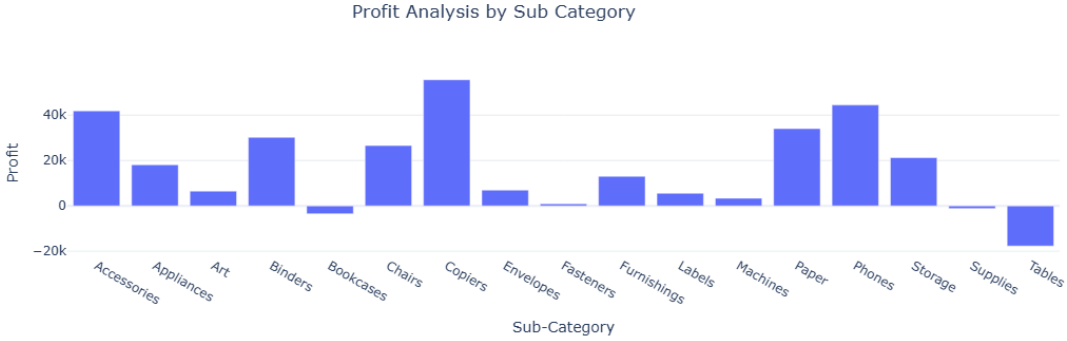
The sub-category "Copiers" stands out with the highest profit, reaching approximately 40k. This indicates that copiers are a highly profitable product line.

Sub-categories such as "Accessories" and "Phones" also show high profits, both around 30k.

Underperformers: "Tables" show significant losses(around -20k), and sub-categories like "Bookcases" and "Supplies" have low or negative profits.

Comparative Analysis of Sales and Profit Across Sub Categories





1. High Sales but Low/Negative Profit Sub-Categories:

Tables: This sub-category has significant sales but shows a negative profit, indicating high costs or inefficiencies. Bookcases: Another sub-category with decent sales but very low or negative profits, suggesting pricing or cost management issues.

2. High Sales and High Profit Sub-Categories:

Chairs: This sub-category has high sales and high profit, making it one of the best-performing sub-categories. Phones: This sub-category also shows strong sales and decent profits, indicating it is another well-performing area.

3. Low Sales but High Profit Sub-Categories:

Copiers: Despite relatively low sales compared to other sub-categories, this sub-category shows significant profit, suggesting a high profit margin.

4. Low Sales and Low/Negative Profit Sub-Categories:

Fasteners and Supplies: Both have low sales and negligible or negative profits, indicating underperformance or lack of demand.

5. Consistently Strong Sub-Categories:

Accessories and Storage: These sub-categories show moderate-to-high sales and maintain steady profits, indicating a balanced performance.

Sales and Profit Analysis by Customer Segment

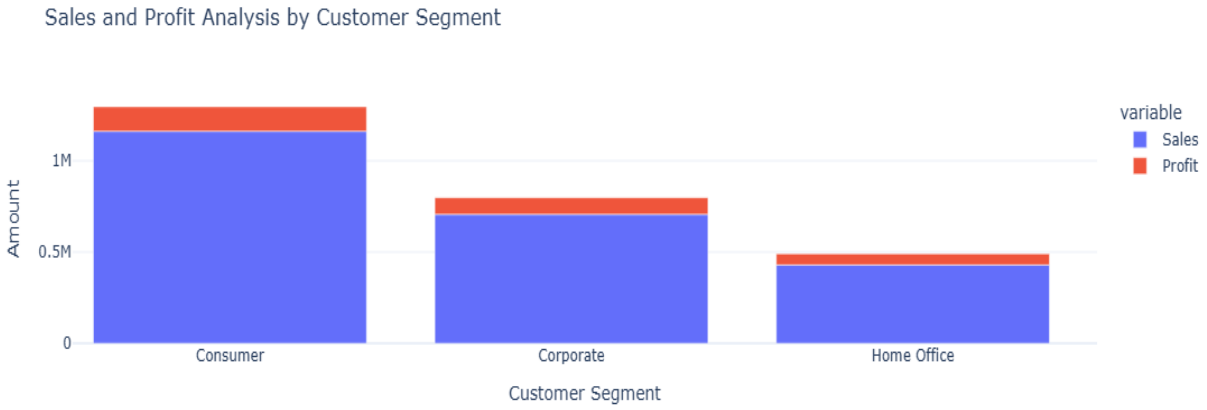
In [151…
```
sales_profit_by_segment=data.groupby('Segment').agg({'Sales':'sum','Profit':'sum'}).reset_index()
sales_profit_by_segment
```

Out[151…

	Segment	Sales	Profit
0	Consumer	1.161401e+06	134119.2092
1	Corporate	7.061464e+05	91979.1340
2	Home Office	4.296531e+05	60298.6785

In [153…
```
sales_profit_by_segment = data.groupby('Segment').agg({'Sales': 'sum', 'Profit': 'sum'}).reset_index()

fig = px.bar(sales_profit_by_segment,
             x='Segment',
             y=['Sales', 'Profit'],
             title='Sales and Profit Analysis by Customer Segment',
             labels={'value': 'Amount', 'Segment': 'Customer Segment'})
fig.show()
```



The Consumer segment is the most profitable and generates the highest sales. This segment appears to be the company's primary focus, performing well in terms of both revenue and profitability.

The Home Office segment is the weakest performer in both sales and profit.

In [ ]: