E-COMMERCE ANALYSIS

```
import pandas as pd #For Data Cleaning

import plotly.express as px #For Visualization
import plotly.graph_objects as go #For Advanced and Customized Graphs
import plotly.io as pio #For Customizing the templates of graphs
import plotly.colors as colors
pio.templates.default = "plotly_white"

data = pd.read_csv("Sample - Superstore.csv", encoding = 'latin-1')

data.head()
```

```
   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  ...

          Category  Sub-Category  \
0        Furniture     Bookcases
1        Furniture        Chairs
2  Office Supplies        Labels
3        Furniture        Tables
4  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  Order Month  Order Year  Order Day of Week
0     0.00   41.9136           11        2016                  1
1     0.00  219.5820           11        2016                  1
2     0.00    6.8714            6        2016                  6
3     0.45 -383.0310           10        2015                  6
4     0.20    2.5164           10        2015                  6

[5 rows x 24 columns]
```

```
data.describe()
```

```
            Row ID   Postal Code         Sales     Quantity
Discount  \
count  9994.000000   9994.000000   9994.000000  9994.000000
9994.000000
mean   4997.500000  55190.379428    229.858001     3.789574
0.156203
std    2885.163629  32063.693350    623.245101     2.225110
0.206452
min       1.000000   1040.000000      0.444000     1.000000
0.000000
25%    2499.250000  23223.000000     17.280000     2.000000
0.000000
50%    4997.500000  56430.500000     54.490000     3.000000
0.200000
75%    7495.750000  90008.000000    209.940000     5.000000
0.200000
max    9994.000000  99301.000000  22638.480000    14.000000
0.800000

            Profit
count  9994.000000
mean     28.656896
std     234.260108
min   -6599.978000
25%       1.728750
50%       8.666500
75%      29.364000
max    8399.976000
```

```
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
```

Converting Date Columns

```
data['Order Date']= pd.to_datetime(data['Order Date'])

data['Ship Date']= pd.to_datetime(data['Ship Date'])

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
```

To calculate Monthly, Yearly Analysis Make New Columns And To Calcuate Order Date and Delivery Date Days Difference

```
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

data.head(5)

   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  ...

          Category  Sub-Category  \
0        Furniture     Bookcases
1        Furniture        Chairs
2  Office Supplies        Labels
3        Furniture        Tables
```

```
4  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  Order Month  Order Year  Order Day of Week
0     0.00    41.9136           11        2016                  1
1     0.00   219.5820           11        2016                  1
2     0.00     6.8714            6        2016                  6
3     0.45  -383.0310           10        2015                  6
4     0.20     2.5164           10        2015                  6

[5 rows x 24 columns]
```

Monthly Sales Analysis

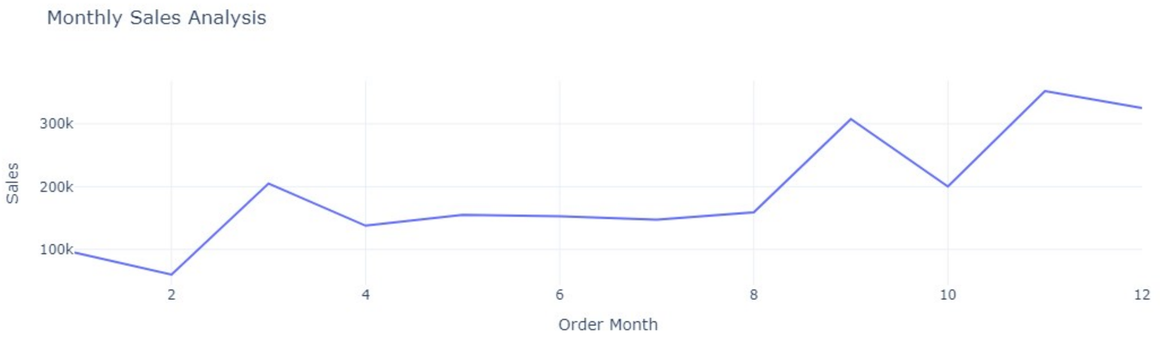
Q.1 You need to calculate the monthly sales of the store and identify which month had the highest sales and which month had the lowest sales.

```
sales_by_month = data.groupby('Order Month')
['Sales'].sum().reset_index()

sales_by_month

    Order Month        Sales
0             1   94924.8356
1             2   59751.2514
2             3  205005.4888
3             4  137762.1286
4             5  155028.8117
5             6  152718.6793
6             7  147238.0970
7             8  159044.0630
8             9  307649.9457
9            10  200322.9847
10           11  352461.0710
11           12  325293.5035
```

```
fig = px.line(sales_by_month,
              x='Order Month',
              y='Sales',
              title='Monthly Sales Analysis')
fig.show()
```



SALES BY CATEGORY

Q.2 You need to analyze sales based on product categories and determine which category has the lowest sales and which category has the highest sales.

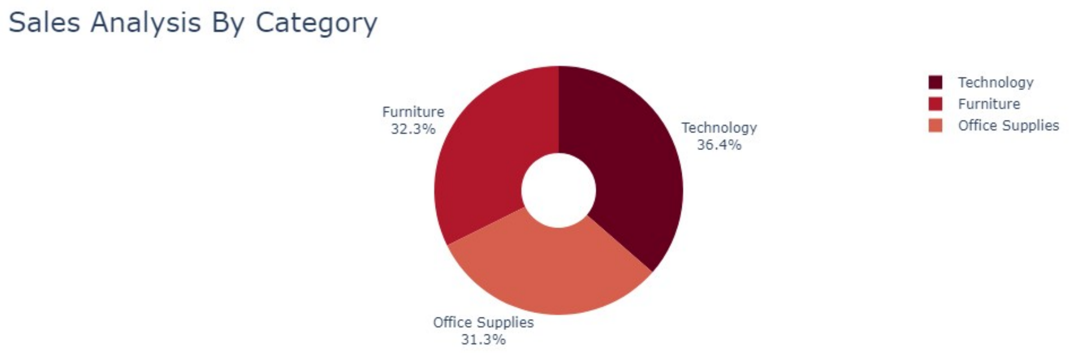
```
sales_by_category = data.groupby('Category')
['Sales'].sum().reset_index()

sales_by_category

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

fig = px.pie(sales_by_category,
             values='Sales',
             names='Category',
             hole=0.3,
             color_discrete_sequence=px.colors.sequential.RdBu)

fig.update_traces(textposition = 'outside', textinfo='percent+label')
fig.update_layout(title_text='Sales Analysis By Category',
title_font=dict(size=24))
fig.show()
```



Sales Analysis By Category

Sales Analysis By Sub Category

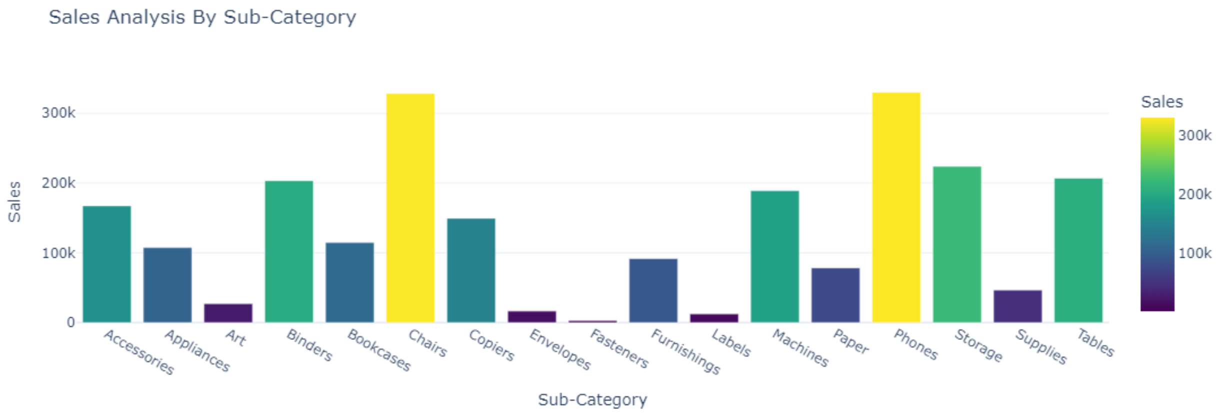
Q.3 The sales analysis needs to be done based on sub-categories

```
sales_by_subcategory=data.groupby('Sub-Category')
['Sales'].sum().reset_index()

sales_by_subcategory

   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

fig = px.bar(sales_by_subcategory,
             x='Sub-Category',
             y='Sales',
             title='Sales Analysis By Sub-Category',
             color="Sales",
            color_continuous_scale = 'viridis',
             height=400)
fig.show()
```



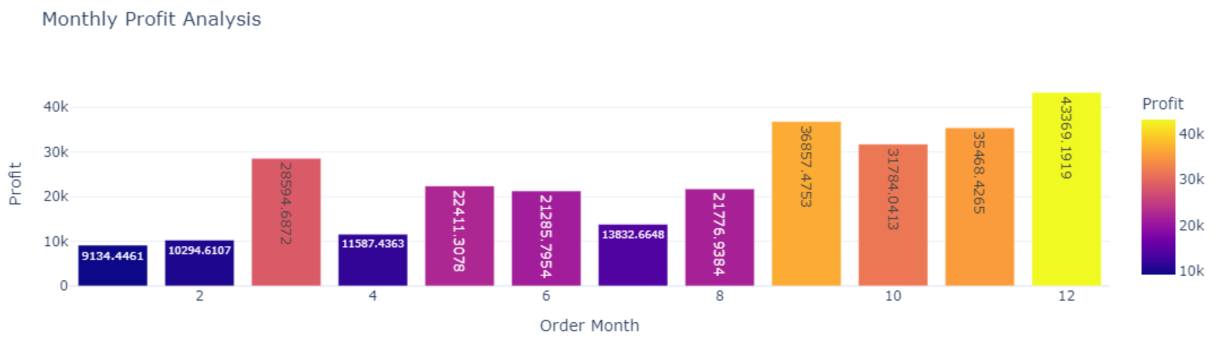
MONTHLY PROFIT

Q.4 You need to analyze the monthly profit from sales and determine which month had the highest profit.

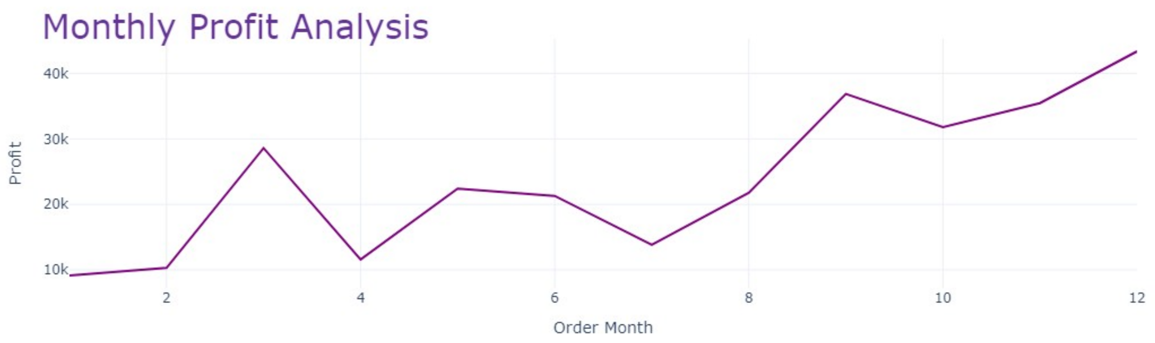
```
profit_by_month = data.groupby("Order Month")
["Profit"].sum().reset_index()
profit_by_month

    Order Month       Profit
0             1    9134.4461
1             2   10294.6107
2             3   28594.6872
3             4   11587.4363
4             5   22411.3078
5             6   21285.7954
6             7   13832.6648
7             8   21776.9384
8             9   36857.4753
9            10   31784.0413
10           11   35468.4265
11           12   43369.1919

fig = px.bar(profit_by_month,
             x='Order Month',
             y='Profit',
             text='Profit',
             title='Monthly Profit Analysis',
             color='Profit')
fig.show()
```



```
fig = px.line(profit_by_month,
             x='Order Month',
             y='Profit',
             )
fig.update_layout(title=dict(text="Monthly Profit Analysis",
                             font=dict(size=30,color="RebeccaPurple"),
automargin=True, yref='paper'))
fig.update_traces(line_color='purple')
fig.show()
```



ANALYZING PROFIT BY CATEGORY & SUB-CATEGORY

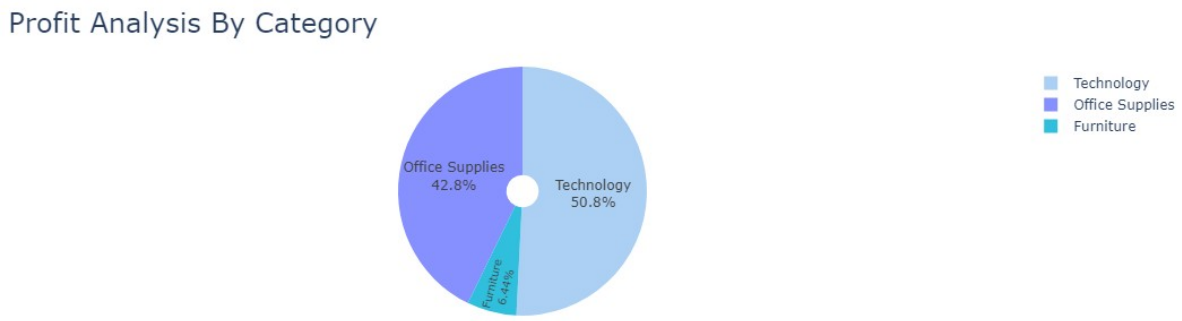
Q.5 Analyze the profit by category and sub-category

```
profit_by_category = data.groupby('Category')
['Profit'].sum().reset_index()
profit_by_category
```

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

fig = px.pie(profit_by_category,
             values='Profit',
             names='Category',
             hole=0.13,
             color = ['G1', 'G2', 'G3'],
             color_discrete_map = {'G1': '#30BFDD',
                                   'G2': '#8690FF',
                                   'G3': '#ACD0F4'})

fig.update_traces(textposition = 'inside', textinfo='percent+label')
fig.update_layout(title_text='Profit Analysis By Category',
title_font=dict(size=24,color="Sky Blue"))
fig.show()
```

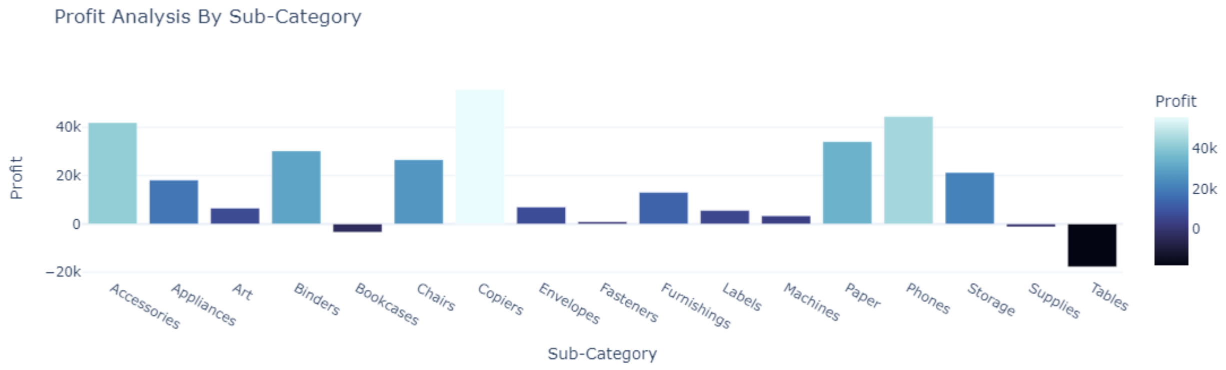


```
profit_by_subcategory = data.groupby('Sub-Category')
['Profit'].sum().reset_index()
profit_by_subcategory

   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

fig = px.bar(profit_by_subcategory,
             x='Sub-Category',
             y='Profit',
             color='Profit', color_continuous_scale = 'ice',
             title='Profit Analysis By Sub-Category')
fig.show()
```



SALES & PROFIT BY CUSTOMER SEGMENT

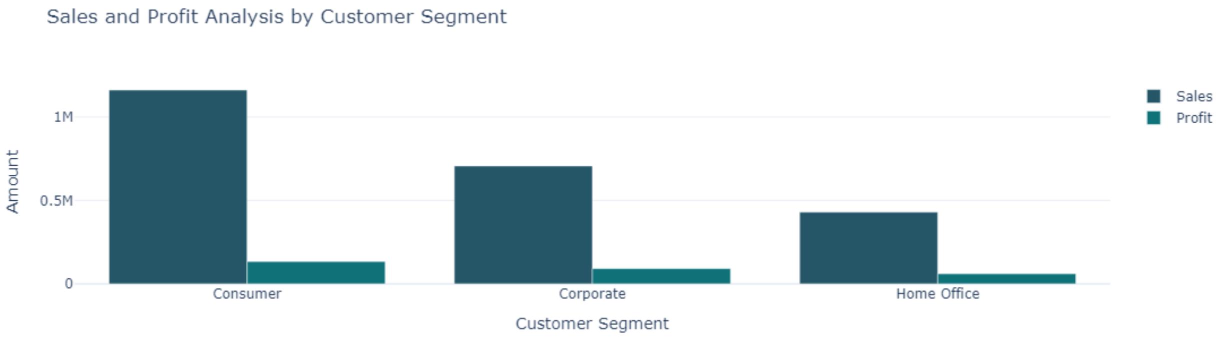
Q.6 Analyze the sales and profit by customer segment

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

color_palette = px.colors.sequential.Aggrnyl
fig = go.Figure()
fig.add_trace(go.Bar(x = sales_profit_by_segment['Segment'],
                    y = sales_profit_by_segment['Sales'],
                    name='Sales',
                    marker_color = color_palette[0]))

fig.add_trace(go.Bar(x=sales_profit_by_segment['Segment'],
                    y=sales_profit_by_segment['Profit'],
                    name='Profit',
                    marker_color = color_palette[1]))

fig.update_layout(title='Sales and Profit Analysis by Customer
Segment',
                 xaxis_title='Customer Segment',
                  yaxis_title='Amount')
fig.show()
```



SALES TO PROFIT RATIO

Q.7 Analyze the sales to profit ratio

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

sales_profit_by_segment['Sales_to_Profit_Ratio'] =
sales_profit_by_segment['Sales'] / sales_profit_by_segment['Profit']

print(sales_profit_by_segment[['Segment', 'Sales_to_Profit_Ratio']])
```

```
       Segment  Sales_to_Profit_Ratio
0     Consumer               8.659471
1    Corporate               7.677245
2  Home Office               7.125416
```