In [2]:
```
!pip3 install plotly
```

```
Requirement already satisfied: plotly in /Library/Frameworks/Python.framewor
k/Versions/3.13/lib/python3.13/site-packages (6.0.0)
Requirement already satisfied: narwhals>=1.15.1 in /Library/Frameworks/Pytho
n.framework/Versions/3.13/lib/python3.13/site-packages (from plotly) (1.24.
1)
Requirement already satisfied: packaging in /Library/Frameworks/Python.frame
work/Versions/3.13/lib/python3.13/site-packages (from plotly) (24.2)
```

In [5]:
```
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 [7]:
```
data = pd.read_csv("salesdata.csv", encoding = "latin-1")
```

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

Out[9]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Co
0	1	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	U S
1	2	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	U S
2	3	CA-2016-138688	6/12/2016	6/16/2016	Second Class	DV-13045	Darrin Van Huff	Corporate	U S
3	4	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	U S
4	5	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	U S

5 rows × 21 columns

In [10]:
```
data.describe()
```

Out[10]:

	Row ID	Postal Code	Sales	Quantity	Discount	
count	9994.000000	9994.000000	9994.000000	9994.000000	9994.000000	9994.
mean	4997.500000	55190.379428	229.858001	3.789574	0.156203	28.
std	2885.163629	32063.693350	623.245101	2.225110	0.206452	234
min	1.000000	1040.000000	0.444000	1.000000	0.000000	-6599
25%	2499.250000	23223.000000	17.280000	2.000000	0.000000	1
50%	4997.500000	56430.500000	54.490000	3.000000	0.200000	8.
75%	7495.750000	90008.000000	209.940000	5.000000	0.200000	29.
max	9994.000000	99301.000000	22638.480000	14.000000	0.800000	8399

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 [12]:

```
#converting date columns
```

In [13]:

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

In [14]:

```
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   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: datetime64[ns](1), float64(3), int64(3), object(14)
memory usage: 1.6+ MB
```

In [15]:

```
data.head()
```

Out[15]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Countr
0	1	CA-2016-152156	2016-11-08	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	Unite State
1	2	CA-2016-152156	2016-11-08	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	Unite State
2	3	CA-2016-138688	2016-06-12	6/16/2016	Second Class	DV-13045	Darrin Van Huff	Corporate	Unite State
3	4	US-2015-108966	2015-10-11	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	Unite State
4	5	US-2015-108966	2015-10-11	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	Unite State

5 rows × 21 columns

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 [20]:

```
data.head()
```

Out[20]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Countr
0	1	CA-2016-152156	2016-11-08	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	Unite State
1	2	CA-2016-152156	2016-11-08	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	Unite State
2	3	CA-2016-138688	2016-06-12	6/16/2016	Second Class	DV-13045	Darrin Van Huff	Corporate	Unite State
3	4	US-2015-108966	2015-10-11	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	Unite State
4	5	US-2015-108966	2015-10-11	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	Unite State

5 rows × 24 columns

In [21]:

```
#Monthly sales analysis
```

In [22]:

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

In [24]:
```
#Sales by Category
```

In [25]:
```
sales_by_category = data.groupby('Category')['Sales'].sum().reset_index()
```

In [26]:
```
sales_by_category
```

Out[26]:

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

In [28]:
```
fig = px.pie(sales_by_category,
             values='Sales',
             names='Category',
             hole=0.5,
             color_discrete_sequence=px.colors.qualitative.Pastel)

fig.update_traces(textposition='inside', textinfo='percent+label')
fig.update_layout(title_text='Sales Analysis by Category', title_font=dict(s
fig.show()
```

In [29]:

```
#Sales analysis by sub-category
```

In [30]:

```
data.head()
```

Out[30]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Countr
0	1	CA-2016-152156	2016-11-08	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	Unite State
1	2	CA-2016-152156	2016-11-08	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	Unite State
2	3	CA-2016-138688	2016-06-12	6/16/2016	Second Class	DV-13045	Darrin Van Huff	Corporate	Unite State
3	4	US-2015-108966	2015-10-11	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	Unite State
4	5	US-2015-108966	2015-10-11	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	Unite State

5 rows × 24 columns

In [31]:

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

In [32]:

```
sales_by_subcategory
```

Out [32]:

	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 [34]:

```
fig = px.bar(sales_by_subcategory, x= 'Sub-Category', y = 'Sales', title= "S

fig.show()
```

```
In [35]: # monthly profit analysis
```

```
In [36]: data.head()
```

Out[36]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Countr
0	1	CA-2016-152156	2016-11-08	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	Unite State
1	2	CA-2016-152156	2016-11-08	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	Unite State
2	3	CA-2016-138688	2016-06-12	6/16/2016	Second Class	DV-13045	Darrin Van Huff	Corporate	Unite State
3	4	US-2015-108966	2015-10-11	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	Unite State
4	5	US-2015-108966	2015-10-11	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	Unite State

5 rows × 24 columns

In [37]:

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

In [38]:

```
profit_by_month
```

Out[38]:

	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

In [39]:

```
fig = px.line(profit_by_month, x = 'Order Month', y= 'Profit', title= 'Month
fig.show()
```

In [40]:

```
#profit by category
```

In [41]:

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

In [42]:

```
profit_by_category
```

Out[42]:

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

In [44]:

```
fig = px.pie(profit_by_category,
             values='Profit',
             names='Category',
             hole=0.5,
             color_discrete_sequence=px.colors.qualitative.Pastel)

fig.update_traces(textposition= 'inside', textinfo = 'percent+label')
fig.update_layout(title_text='Profit Analysis by Category', title_font=dict(
fig.show()
```

In [45]:
```
#profit analysis by sub-category
```

In [46]:
```
profit_by_subcategory = data.groupby('Sub-Category')['Profit'].sum().reset_i
fig = px.bar(profit_by_subcategory, x='Sub-Category',
             y='Profit',
             title='Profit Analysis by Sub-Category')

fig.show()
```

```
In [47]: # sales and profit - customer segment
```

```
In [48]: data.head()
```

Out[48]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Countr
0	1	CA-2016-152156	2016-11-08	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	Unite State
1	2	CA-2016-152156	2016-11-08	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	Unite State
2	3	CA-2016-138688	2016-06-12	6/16/2016	Second Class	DV-13045	Darrin Van Huff	Corporate	Unite State
3	4	US-2015-108966	2015-10-11	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	Unite State
4	5	US-2015-108966	2015-10-11	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	Unite State

5 rows × 24 columns

In [53]:

```
sales_profit_by_segment = data.groupby('Segment').agg({'Sales' : 'sum' ,'Pro
color_palette = colors.qualitative.Pastel

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()
```

```
In [54]: #Sales to profit ratio
```

```
In [58]: sales_profit_by_segment = data.groupby('Segment').agg({'Sales': 'sum', 'Prof
         sales_profit_by_segment['Sales_to_Profit_Ratio'] = sales_profit_by_segment['
         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
```