

```

1 import pandas as pd #Data Cleaning
2
3 import plotly.express as px #Data Visualization
4 import plotly.graph_objects as go #Customizing Graphs
5 import plotly.io as pio #used to customize graphs template
6 import plotly.colors as colors #used for customizing colors
7 pio.templates.default = "plotly_white" #using white theme

```

```

1 data = pd.read_csv("Sample - Superstore.csv",encoding = "latin-1")
2 #Used latin-1 so the python can understand that / in dates and - in orders is to be read as well

```

```
1 data.head() #reading only few rows for cleaner and clutter free experience
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Region	Product ID	Category	Sub-Category	Product Name	Quantity	Unit Price	Total
0	1	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-BO-10001798	Electronics	Peripherals	Monitor	1	\$120.00	\$120.00
1	2	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-CH-10000454	Electronics	Peripherals	Monitor	1	\$120.00	\$120.00
2	3	CA-2016-138688	6/12/2016	6/16/2016	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	...	90036	West	OFF-LA-10000240	Electronics	Peripherals	Monitor	1	\$120.00	\$120.00
3	4	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	South	FUR-TA-10000577	Electronics	Peripherals	Monitor	1	\$120.00	\$120.00
4	5	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	South	OFF-ST-10000760	Electronics	Peripherals	Monitor	1	\$120.00	\$120.00

5 rows × 21 columns

```
1 data.tail()
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Region	Pro
9989	9990	CA-2014-110422	1/21/2014	1/23/2014	Second Class	TB-21400	Tom Boeckenhauer	Consumer	United States	Miami	...	33180	South	FUI 1000
9990	9991	CA-2017-121258	2/26/2017	3/3/2017	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	...	92627	West	FUI 1000
9991	9992	CA-2017-121258	2/26/2017	3/3/2017	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	...	92627	West	TEC 1000
9992	9993	CA-2017-121258	2/26/2017	3/3/2017	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	...	92627	West	OF 1000
9993	9994	CA-2017-119914	5/4/2017	5/9/2017	Second Class	CC-12220	Chris Cortes	Consumer	United States	Westminster	...	92683	West	OF 1000

5 rows × 21 columns

```

1 data.describe() #used for quick descriptive analysis
2 # Here the mean profit acheived was 28.65
3 # loss was min i.e -6599.9 on an order
4 # maximum profit was 8399.97
5 #also can find count of sales and Quantity and discount

```

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

```
1 data.info() #gives info about Datatype of columns and find non null values
```

<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 the Order Date, Ship Date into date datatype

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

```
1 data.info() #successfully converted Order Date into Date Time format
```

```

<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

```

```
1 data['Ship Date'] = pd.to_datetime(data['Ship Date'])
2 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

```

```
1 data.head()
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Region	Product ID	Category
0	1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-BO-10001798	Furniture
1	2	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-CH-10000454	Furniture
2	3	CA-2016-138688	2016-06-12	2016-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	...	90036	West	OFF-LA-10000240	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	...	33311	South	FUR-TA-10000577	Furniture
4	5	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	South	OFF-ST-10000760	Office Supplies

5 rows × 21 columns

## ▼ Finding Monthly Sales and Yearly Sales

```

1 data['Order Month'] = data['Order Date'].dt.month
2 data['Order Year'] = data['Order Date'].dt.year
3 data['Order Day of Week'] = data['Order Date'].dt.dayofweek
4
5 data.head()

```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Category	Sub-Category	Product Name
0	1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	Furniture	Bookcases	Bush Somerset Collection Bookcase
1	2	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	Furniture	Chairs	Hon Deluxe Fabric Upholstered Stacking Chairs,...
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	Labels	Self-Adhesive Address Labels for Typewriters b...
3	4	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	Furniture	Tables	Bretford CR4500 Series Slim Rectangular Table
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	Storage	Eldon Fold 'N Roll Cart System

5 rows × 24 columns

## ▼ Monthly Sales Analysis

```
1 sales_by_month = data.groupby('Order Month')['Sales'].sum().reset_index()
2 # Above we are grouping sales by order months and doing sum of the grouped data
3
```

```
1 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

Next steps: [New interactive sheet](#)

```
1
2 fig = px.line(sales_by_month,
3                 x = 'Order Month',
4                 y = 'Sales',
5                 title = 'Monthly Sales Analysis'
6                 )
7 # Did a line graph above and gave x,y,title names in the px.line()
8 fig.show()
```

Monthly Sales Analysis



The Above Plotted line is dynamic that is we can zoom on certain month

Here we have zoomed on month Feb to May

> Here highest sales is on month of November and lowest Sales is on February

## ▼ Category Wise Sales

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

```
1 sales_by_category
```

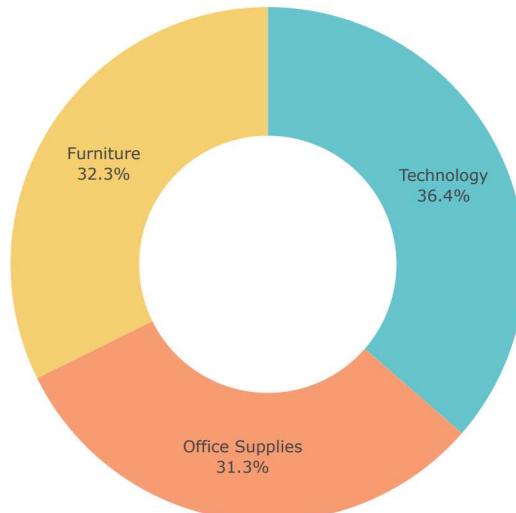
	Category	Sales
0	Furniture	741999.7953
1	Office Supplies	719047.0320
2	Technology	836154.0330

Next steps: [New interactive sheet](#)

```
1 # Creating a Pie Chart for Distribution of Sales by Category
```

```
1 fig = px.pie(sales_by_category,
2                 values = 'Sales',
3                 names = 'Category',
4                 hole = 0.5,
5                 color_discrete_sequence = px.colors.qualitative.Pastel) #colors are given as pastel
6 fig.update_traces(textposition = 'inside', textinfo = 'percent+label') # giving the text of sales and
7 fig.update_layout(title_text = 'Sales Analysis By Category',title_font = dict(size=18)) #title name a
8
9 fig.show()
10
11 # here we used a Pie Chart and Converted it to Donut chart by giving hole size = 0.5 ,also passing va
```

Sales Analysis By Category



## ▼ Sales Analysis By Sub Category

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

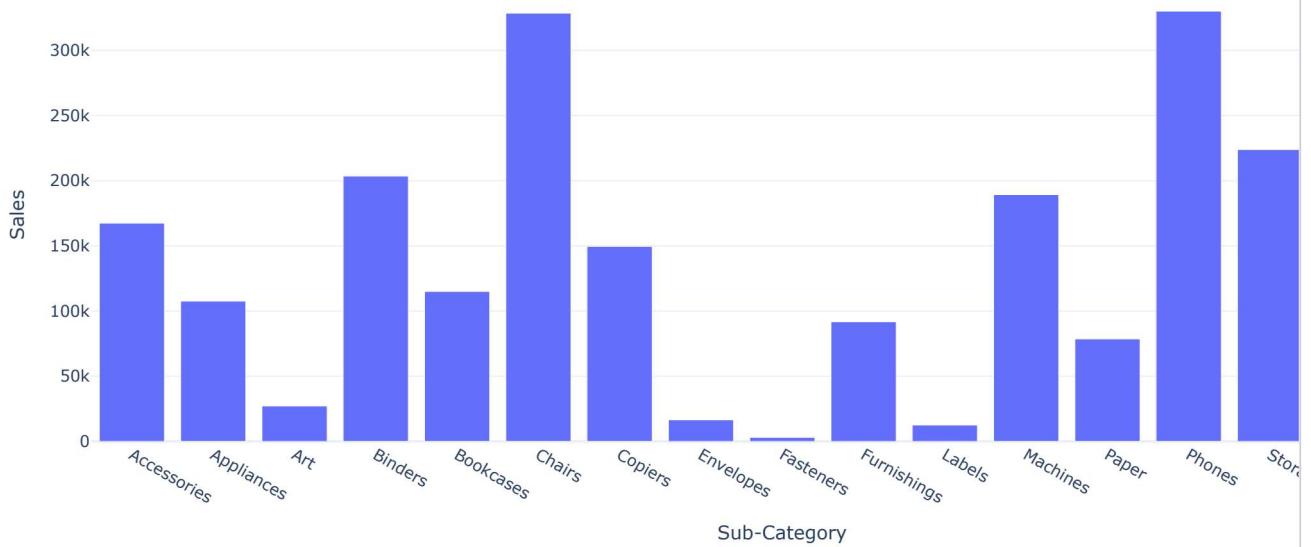
1 sales\_by\_subcategory

	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	

Next steps: [New interactive sheet](#)

```
1 fig = px.bar(sales_by_subcategory,x='Sub-Category',y='Sales',title = 'Sales Analysis By Sub Category'
2
3 fig.show()
```

Sales Analysis By Sub Category



## Profit Analysis By Category and Month

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

```
1 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

Next steps: [New interactive sheet](#)

```
1 fig = px.line(profit_by_month, x= 'Order Month', y = 'Profit', title = 'Profit Based on Months')
2
3 fig.show()
```

Profit Based on Months



```
1 fig = px.bar(profit_by_month, x= 'Order Month', y = 'Profit', title = 'Profit Based on Months',color_
2
3 fig.show()
```

Profit Based on Months



Here we had most profit on month of December and least profit on January

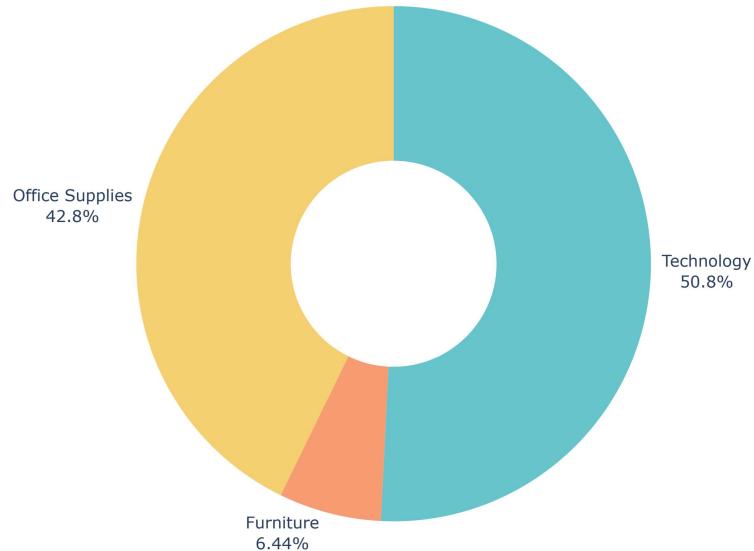
```
1 profit_by_category = data.groupby('Category')['Profit'].sum().reset_index()
2 profit_by_category
```

	Category	Profit	Icon
0	Furniture	18451.2728	grid icon
1	Office Supplies	122490.8008	bar icon
2	Technology	145454.9481	chart icon

Next steps: [New interactive sheet](#)

```
1 fig = px.pie(profit_by_category,
2                 values = 'Profit',
3                 names = 'Category',
4                 hole = 0.4,
5                 color_discrete_sequence = px.colors.qualitative.Pastel) #colors are given as pastel
6 fig.update_traces(textposition = 'outside', textinfo = 'percent+label') # giving the text of sales ar
7 fig.update_layout(title_text = 'Profit Analysis By Category',title_font = dict(size=18)) #title name
8
9 fig.show()
```

### Profit Analysis By Category



Here we have more profit by Technology Category

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

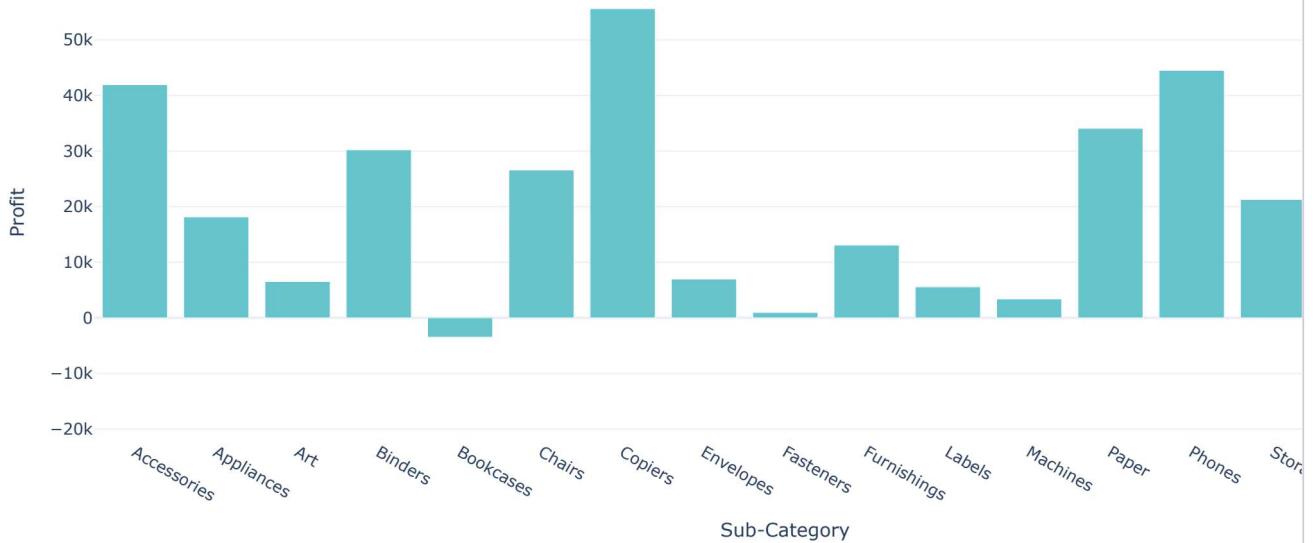
```
1 profit_by_subcategory
```

	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

Next steps: [New interactive sheet](#)

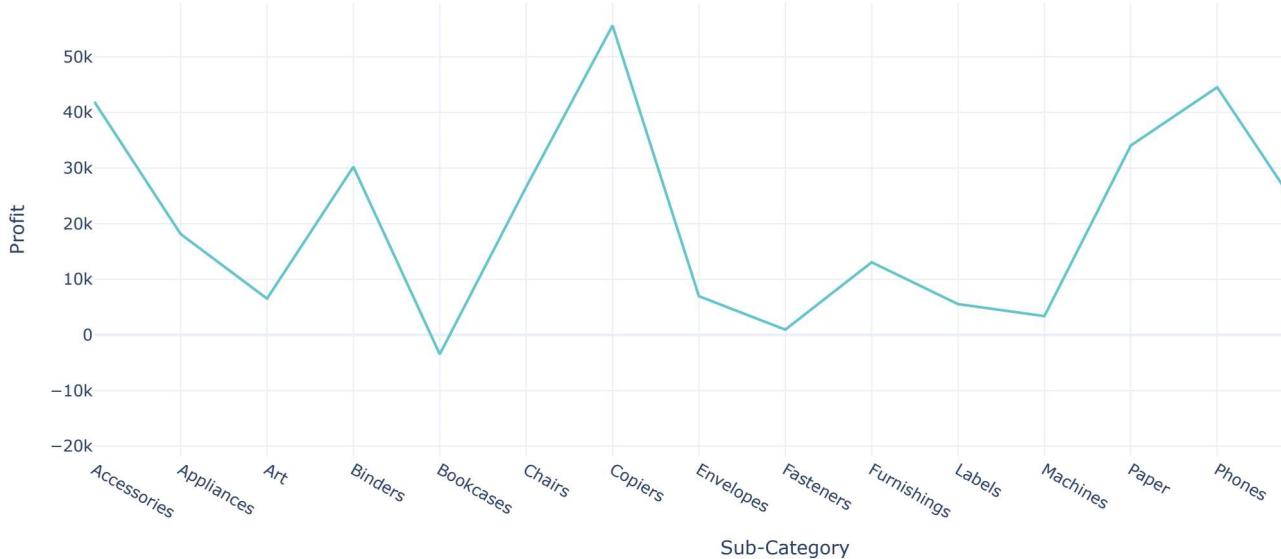
```
1 fig = px.bar(profit_by_subcategory, x= 'Sub-Category', y = 'Profit', title = 'Profit Based on Sub Ca1
2
3 fig.show()
```

Profit Based on Sub Category



```
1 fig = px.line(profit_by_subcategory, x= 'Sub-Category', y = 'Profit', title = 'Profit Based on Sub Ca2
2
3 fig.show()
```

Profit Based on Sub Category



Here we had massive loss on Sub Category Table and massive profit on Copiers

- ▼ Finding Sales and Profit by Customer Segment

1 data.head()

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Category	Sub-Category	Product Name	
0	1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	Furniture	Bookcases	Bush Somerset Collection Bookcase	26
1	2	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	Furniture	Chairs	Hon Deluxe Fabric Upholstered Stacking Chairs...	73
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	Labels	Self-Adhesive Address Labels for Typewriters b...	14
3	4	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	Furniture	Tables	Bretford CR4500 Series Slim Rectangular Table	95
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	Storage	Eldon Fold 'N Roll Cart System	22

5 rows × 24 columns

```

1 sales_profit_by_segment = data.groupby('Segment').agg({'Sales':'sum','Profit':'sum'}).reset_index()
2
3 color_palette = colors.qualitative.Pastel
4 fig= go.Figure()
5 fig.add_trace(go.Bar(x = sales_profit_by_segment['Segment'],
6                     y = sales_profit_by_segment['Sales'],
7                     name = 'Sales',
8                     marker_color = color_palette[0]))
9 fig.add_trace(go.Bar(x = sales_profit_by_segment['Segment'],
10                    y = sales_profit_by_segment['Profit'],
11                    name = 'Profit',
12                    marker_color = color_palette[1]))
13 fig.update_layout(title='Sales and Profit Analysis by Customer Segment',
14                     xaxis_title='Customer Segment',yaxis_title='Amount')
15 fig.show()

```

### Sales and Profit Analysis by Customer Segment

