

```

1 import pandas as pd
2 import numpy as np
3 import plotly.express as px
4 import plotly.graph_objects as go
5 import plotly.io as pio
6 import plotly.colors as colors
7 pio.templates.default = "plotly_white"

```

```

1 from google.colab import files
2 uploaded = files.upload()

```

  Sample - Superstore.csv

- **Sample - Superstore.csv**(text/csv) - 2287806 bytes, last modified: 04/05/2025 - 100% done

Saving Sample - Superstore.csv to Sample - Superstore.csv

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

```
1 data.head(5)
```



	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Region
0	1	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	S
1	2	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	S
2	3	CA-2016-138688	6/12/2016	6/16/2016	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	...	90036	\
3	4	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	S
4	5	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	S

5 rows × 21 columns

```
1 data.describe()
```



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



```
1 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

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

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

```
1 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(5)
```



	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Region	Pr
0	1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FU 100
1	2	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FU 100
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	Of 100
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	FL 100
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	Of 100

5 rows × 21 columns

```
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
```

```
1 data.head(5)
```



	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Category	Sub-Category
0	1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	Furniture	Bookcases
1	2	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	Furniture	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
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
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

5 rows × 24 columns

## ▼ Monthly sales Analysis

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

```
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:

[Generate code with sales\\_by\\_month](#)
[View recommended plots](#)
[New interactive sheet](#)

```
1 fig = px.line(sales_by_month,
2               x = 'Order Month',
```

```

3     y = 'Sales',
4     title = 'Monthly sales Analysis')
5 fig.show()

```



Monthly sales Analysis



```
1 data.head(5)
```



	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Category	Sub-Category
0	1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	Furniture	Bookcases
1	2	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	Furniture	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
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
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

5 rows × 24 columns

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

1 sales\_by\_category



1 to 3 of 3 entries

Filter



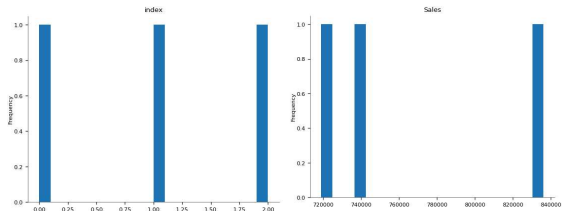
index	Category	Sales
0	Furniture	741999.7953
1	Office Supplies	719047.032
2	Technology	836154.033

Show 25 per page

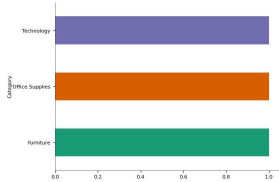


Like what you see? Visit the [data table notebook](#) to learn more about interactive tables.

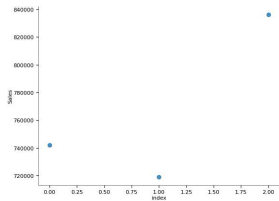
Distributions



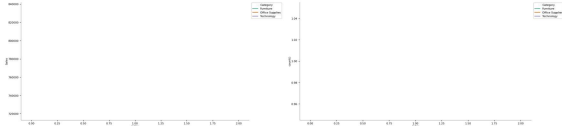
Categorical distributions



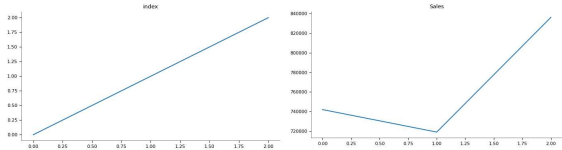
2-d distributions



Time series



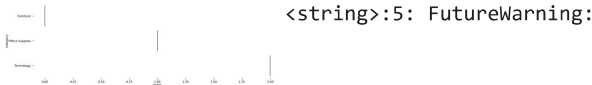
Values



Faceted distributions

<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable



Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable



Next steps:

[Generate code with sales\\_by\\_category](#)[View recommended plots](#)[New interactive sheet](#)

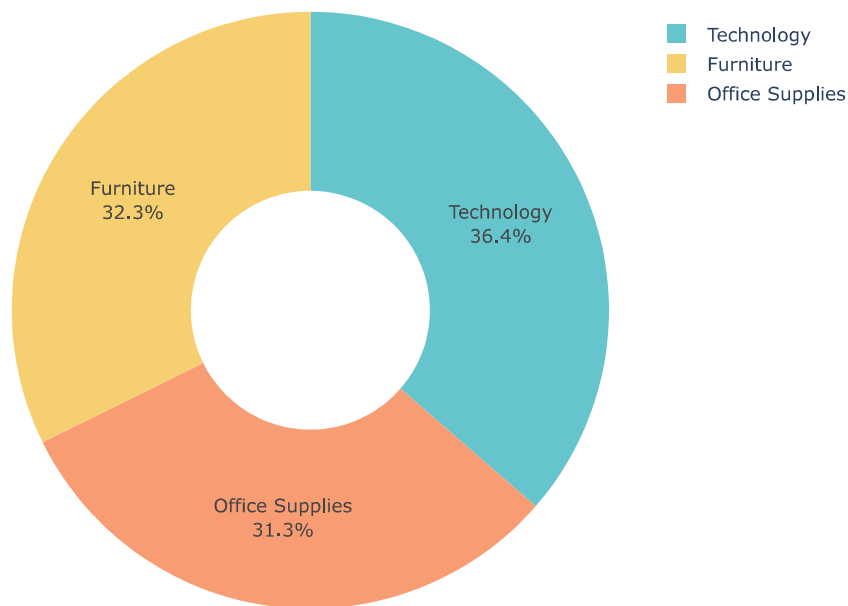
```

1 fig = px.pie(sales_by_category,
2             values = 'Sales', # Changed 'sales' to 'Sales'
3             names = 'Category', # Changed 'category' to 'Category'
4             hole = 0.4,
5             color_discrete_sequence= px.colors.qualitative.Pastel)
6 fig.update_traces(textposition = 'inside', textinfo = 'percent+label')
7 fig.update_layout(title_text = 'sales Analysis by category', title_font= dict(size = 25)
8 fig.show()

```



## 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	

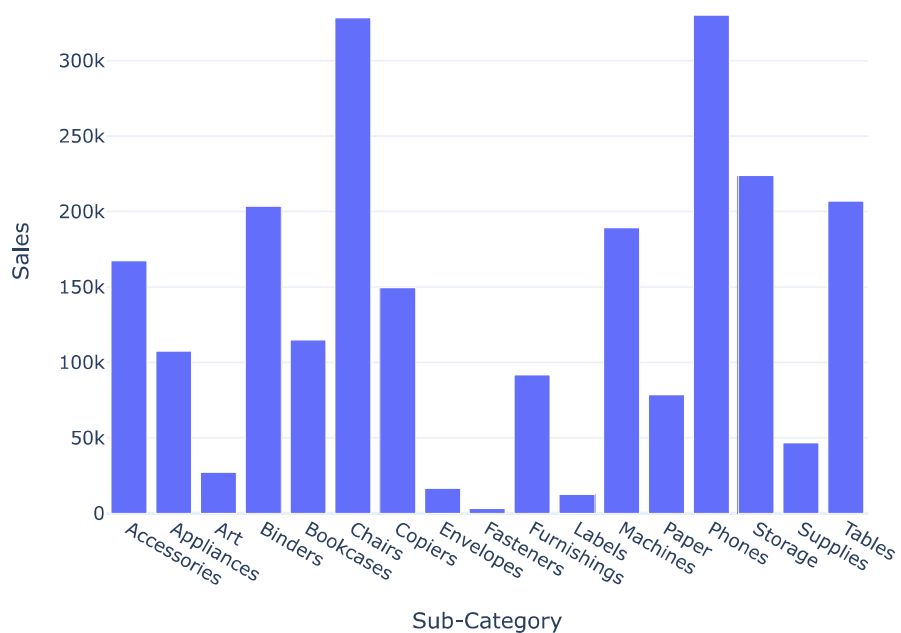
Next steps:

[Generate code with Sales\\_by\\_subCategory](#)[View recommended plots](#)[New interactive sheet](#)

```
1 fig = px.bar(Sales_by_subCategory, # Changed 'sales_by_subcategory' to 'Sales_by_subCat
2           x = 'Sub-Category',
3           y = 'Sales',
4           title = "Sales Analysis by Sub-Category")
5 fig.show()
```



Sales Analysis by Sub-Category





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

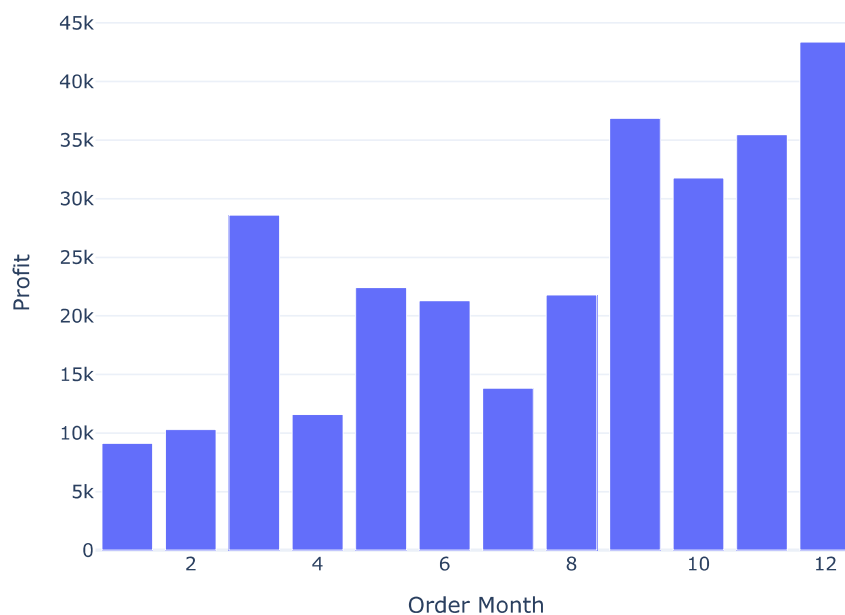
```
1 profit_by_month
```

	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: [Generate code with profit\\_by\\_month](#) [View recommended plots](#) [New interactive sheet](#)

```
1 fig = px.bar(profit_by_month,  
2             x = 'Order Month',  
3             y = 'Profit',  
4             title = 'Monthly Profit Analysis')  
5 fig.show()
```

Monthly Profit Analysis



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

## 1 profit\_by\_category

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

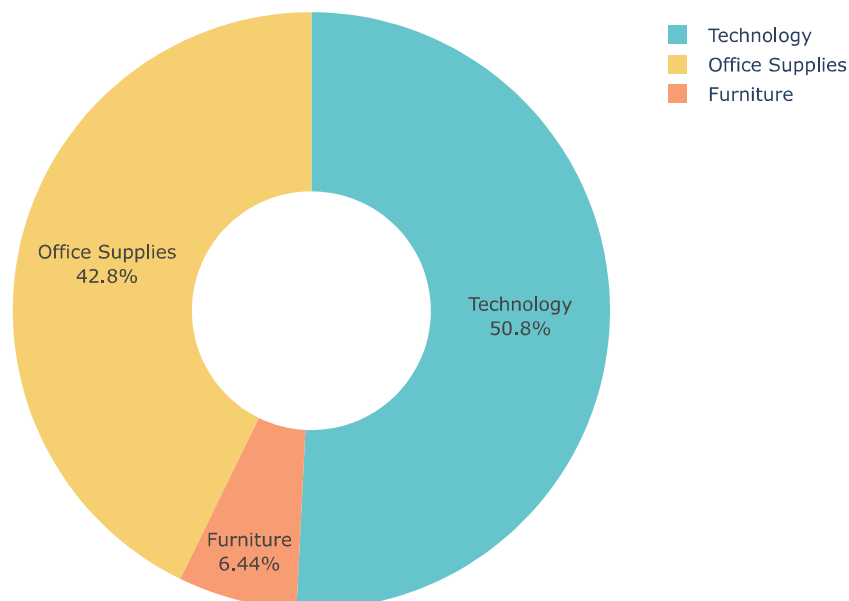
Next steps:

[Generate code with profit\\_by\\_category](#)[View recommended plots](#)[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)
6 fig.update_traces(textposition = 'inside', textinfo = 'percent+label')
7 fig.update_layout(title_text = 'Profit Analysis by category', title_font= dict(size = 2
8 fig.show()
```

## Profit Analysis by category



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

```
1 profit_by_subcategory
```

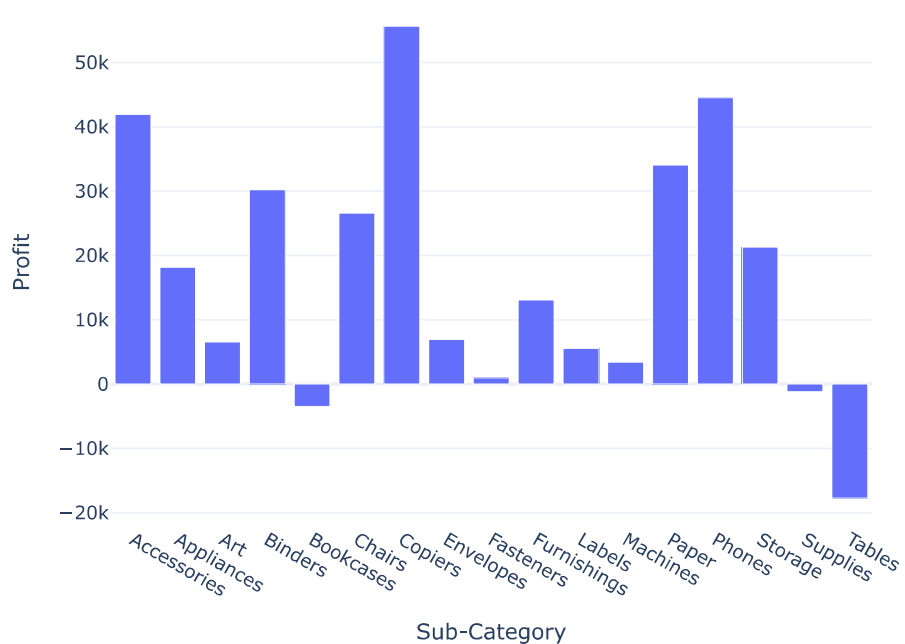
	Sub-Category	Profit	
0	Accessories	41936.6357	
1	Appliances	18138.0054	
2	Art	6527.7870	
3	Binders	30221.7633	
4	Bookcases	-3472.5560	
5	Chairs	26590.1663	
6	Copiers	55617.8249	
7	Envelopes	6964.1767	
8	Fasteners	949.5182	
9	Furnishings	13059.1436	
10	Labels	5546.2540	
11	Machines	3384.7569	
12	Paper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

Next steps: [Generate code with profit\\_by\\_subcategory](#)[View recommended plots](#)[New interactive sheet](#)

```
1 fig = px.bar(profit_by_subcategory,  
2             x = 'Sub-Category',  
3             y = 'Profit',  
4             title = 'profit Analysis by SubCategory')  
5 fig.show()
```



profit Analysis by SubCategory



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

1 color_palatte = colors.qualitative.Pastel
2 fig = go.Figure()

1 fig.add_trace(go.Bar(x= sales_profit_by_segment['Segment'],
2                       y =sales_profit_by_segment['Sales'],
3                       name = 'Sales',
4                       marker_color = color_palatte[0]))
5 fig.add_trace(go.Bar(x= sales_profit_by_segment['Segment'],
6                       y =sales_profit_by_segment['Profit'],
7                       name = 'Profit',
8                       marker_color = color_palatte[1]))
9 fig.update_layout(title = 'Sales and Profit Analysis by Customer Segment',
10                   xaxis_title= 'Customer Segment',yaxis_title='Amount')
11 fig.show()
```