```
In [1]: 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 [2]: data = pd.read_csv("Sample - Superstore.csv",encoding = 'latin-1')
```

```
In [4]: data.head()
```

Out[4]:

	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	South
1	2	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South
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
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
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

5 rows × 21 columns

```
In [6]: data.describe()
```

Out[6]:

	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 [7]: 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

```
In [8]: data['Order Date'] = pd.to_datetime(data['Order Date'])
```

```
In [9]: 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 [10]: data['Ship Date'] = pd.to_datetime(data['Ship Date'])
```

```
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   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 [12]: data.head()
```

Out[12]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Region	Product ID
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
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
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
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
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

5 rows × 21 columns

```
In [13]: 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 [14]: data.head()
```

Out[14]:

	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

```
In [18]: sales_by_month = data.groupby('Order Month')['Sales'].sum().reset_index()
```

```
In [19]: sales_by_month
```

Out[19]:

	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

```
In [21]: fig = px.line(sales_by_month,
                       x='Order Month',
                       y='Sales',
                       title='Monthly Sales Analysis')
         fig.show()
```

Sales by Category

```
In [22]: sales_by_category = data.groupby('Category')['Sales'].sum().reset_index()
```

```
In [23]: sales_by_category
```

Out[23]:

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

```
In [33]: 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(size=24))

         fig.show()
```

Sales Analysis by Sub Category

```
In [34]: sales_by_subcategory = data.groupby('Sub-Category')['Sales'].sum().reset_index()
```

```
In [35]: sales_by_subcategory
```

Out[35]:

	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 [36]: fig = px.bar(sales_by_subcategory,x='Sub-Category', y = 'Sales',title ="Sales analysis by sub category"

         fig.show()
```

Monthly profit analysis

```
In [38]: profit_by_month = data.groupby('Order Month')['Profit'].sum().reset_index()
```

```
In [39]: profit_by_month
```

Out[39]:

	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 [41]: fig = px.line(profit_by_month, x = 'Order Month', y ='Profit', title = 'Monthly profit analysis')

         fig.show()
```

profit by category

```
In [42]: profit_by_category = data.groupby('Category')['Profit'].sum().reset_index()
```

```
In [43]: profit_by_category
```

Out[43]:

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

```
In [45]: 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(size=24))

         fig.show()
```

profit by sub category

```
In [47]: profit_by_subcategory = data.groupby('Sub-Category')['Profit'].sum().reset_index()
         fig = px.bar(profit_by_subcategory, x='Sub-Category',
                      y='Profit',
                      title='Profit Analysis by Sub-Category')
         fig.show()
```

```
In [53]: sales_profit_by_segment = data.groupby('Segment').agg({'Sales': 'sum','Profit':'sum'}).reset_index()

         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()
```

sales to profit ratio

```
In [56]: 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_
         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
```

In [ ]: