In [99]:

```
# FOR DATA CLEANING
import pandas as pd
# FOR DATA VISUALIZATION
import plotly.express as px
# FOR ADVANCE AND CUSTOMERIZED GRAPHS
import plotly.graph_objects as go
# FOR CUSTOMERIZED GRAPHS TEMPLATES
import plotly.io as pio
# FOR CUSTOMERIZED COLOURS
import plotly.colors as colors
# FOR WHITE THEME
pio.templates.default = "plotly_white"
```

In [102…

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

In [101…

```
data
```

Out[101…

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segme
0	1	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consun
1	2	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consun
2	3	CA-2016-138688	6/12/2016	6/16/2016	Second Class	DV-13045	Darrin Van Huff	Corpora
3	4	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consun
4	5	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consun
...	...	...	...	...	...	...	...	
9989	9990	CA-2014-110422	1/21/2014	1/23/2014	Second Class	TB-21400	Tom Boeckenhauer	Consun
9990	9991	CA-2017-121258	2/26/2017	3/3/2017	Standard Class	DB-13060	Dave Brooks	Consun
9991	9992	CA-2017-121258	2/26/2017	3/3/2017	Standard Class	DB-13060	Dave Brooks	Consun
9992	9993	CA-2017-121258	2/26/2017	3/3/2017	Standard Class	DB-13060	Dave Brooks	Consun
9993	9994	CA-2017-119914	5/4/2017	5/9/2017	Second Class	CC-12220	Chris Cortes	Consun

9994 rows × 21 columns

In [17]:
```
data.head()
```

Out[17]:

	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
1	2	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520	Claire Gute	Consumer	U
2	3	CA-2016-138688	6/12/2016	6/16/2016	Second Class	DV-13045	Darrin Van Huff	Corporate	U
3	4	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	U
4	5	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	U

5 rows × 21 columns

In [21]:
```
data.describe()
```

Out[21]:

	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

CONVERTING DTYPE (OBJECT TO DATETIME)

In [25]:
```
data['Order Date'] = pd.to_datetime(data['Order Date'])
```

In [26]:
```
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 [27]:
```
data['Ship Date'] = pd.to_datetime(data['Ship Date'])
```

In [28]:
```
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 [29]:

```
data.head()
```

Out[29]:

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

5 rows × 21 columns

ADDING NEW COLOUMS OF MONTH,YEAR ,DAY OF WEEK

In [33]:

```
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 [34]:

```
data.head()
```

Out[34]:

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

5 rows × 24 columns

MONTHLY SALES ANALYSIS

In [39]:

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

In [43]:

```
sales_by_months
```

Out[43]:

	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 [44]:

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

CATEGORY SALES ANALYSIS

In [45]:
```
sales_by_category = data.groupby('Category')['Sales'].sum().reset_index()
sales_by_category
```

Out[45]:

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

In [62]:
```
fig2 = px.pie(sales_by_category,
              values = 'Sales',
              names = 'Category',
              hole = 0.5,
              color_discrete_sequence = px.colors.qualitative.Pastel)
fig2.update_traces (textposition = 'inside',textinfo = 'percent + label')
fig2.update_layout(title_text = 'Category Sales Analysis',title_font = dict(size
fig2.show()
```

SUB-CATEGORY SALES ANALYSIS

```
In [65]: sales_by_Subcategory = data.groupby('Sub-Category')['Sales'].sum().reset_index()
         sales_by_Subcategory
```

Out[65]:

	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 [67]:

```
fig3 = px.bar(sales_by_Subcategory,
        x = 'Sub-Category',
        y = 'Sales',
        title = 'SUB-CATEGORY SALES ANALYSIS')
fig3.show()
```

In [68]:

```
data.head()
```

Out[68]:

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

5 rows × 24 columns

MONTHLY PROFIT ANALYSIS

In [70]:

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

Out[70]:

	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 [71]:

```
fig4 = px.line(profit_by_months,
               x = 'Order Month',
               y = 'Profit',
            title =  'Monthly Profit Analysis')
fig4.show()
```

In [72]:

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

Out[72]:

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

In [92]:

```
fig5 = px.pie(profit_by_category,
              names = 'Category',
              values = 'Profit',
              hole = 0,
              color_discrete_sequence = px.colors.qualitative.Pastel)
fig5.update_traces(textposition = 'inside',textinfo = 'percent + label')
fig5.update_layout(title_text = 'Profit per Category analysis',title_font = dict
fig5.show()
```

```
In [78]: profit_by_subcategory = data.groupby('Sub-Category')['Profit'].sum().reset_index
         profit_by_subcategory
```

Out[78]:

	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

In [81]:

```
fig6 = px.bar(profit_by_subcategory,
        x = 'Sub-Category',
        y = 'Profit',
        title = 'PROFIT PER SUB-CATEGORY ANALYSIS')
fig6.show()
```

Sales Profit Segment Analysis

In [90]:

```
sales_profit_by_segment = data.groupby('Segment') .agg({'Sales':'sum','Profit':'
color_palette = colors.qualitative.Pastel
fig = go.Figure()
fig.add_traces(go.Bar(x = sales_profit_by_segment['Segment'],
                       y = sales_profit_by_segment['Sales'],
                       name = 'Sales',
                    marker_color = color_palette[0]))
fig.add_traces(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_
fig.show()
```

SALES TO PROFIT RATIO

```
In [96]: sales_profit__by_segment = data.groupby('Segment').agg({'Sales':'sum','Profit':'
         sales_profit_by_segment['Sales_To_Profit_Ratio'] = sales_profit_by_segment['Sale
         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 [ ]:
```