

Case Study

Store Sales and Profit Analysis using Python



```
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"

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

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID
\	0	1	CA-2016-152156	11/8/2016	11/11/2016	Second Class CG-12520
	1	2	CA-2016-152156	11/8/2016	11/11/2016	Second Class CG-12520
	2	3	CA-2016-138688	6/12/2016	6/16/2016	Second Class DV-13045
	3	4	US-2015-108966	10/11/2015	10/18/2015	Standard Class SO-20335
	4	5	US-2015-108966	10/11/2015	10/18/2015	Standard Class SO-20335

	Customer Name	Segment	Country	City	...	\
0	Claire Gute	Consumer	United States	Henderson	...	
1	Claire Gute	Consumer	United States	Henderson	...	
2	Darrin Van Huff	Corporate	United States	Los Angeles	...	
3	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	
4	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	

	Postal Code	Region	Product ID	Category	Sub-Category	\
0	42420	South	FUR-BO-10001798	Furniture	Bookcases	
1	42420	South	FUR-CH-10000454	Furniture	Chairs	
2	90036	West	OFF-LA-10000240	Office Supplies	Labels	
3	33311	South	FUR-TA-10000577	Furniture	Tables	
4	33311	South	OFF-ST-10000760	Office Supplies	Storage	

	Product Name	Sales	Quantity	\
0	Bush Somerset Collection Bookcase	261.9600	2	
1	Hon Deluxe Fabric Upholstered Stacking Chairs,...	731.9400	3	
2	Self-Adhesive Address Labels for Typewriters b...	14.6200	2	
3	Bretford CR4500 Series Slim Rectangular Table	957.5775	5	
4	Eldon Fold 'N Roll Cart System	22.3680	2	

	Discount	Profit
0	0.00	41.9136
1	0.00	219.5820
2	0.00	6.8714
3	0.45	-383.0310
4	0.20	2.5164

[5 rows x 21 columns]

In [2]: `print(data.describe())`

```

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

count      Profit
count  9994.000000
mean    28.656896
std     234.260108
min    -6599.978000
25%      1.728750
50%      8.666500
75%     29.364000
max     8399.976000

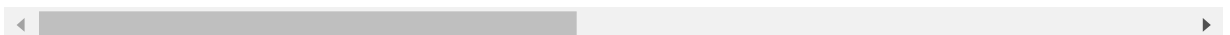
```

In [3]: `data.head()`

Out[3]:

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

5 rows × 21 columns



The dataset has an order date column. We can use this column to create new columns like order month, order year, and order day, which will be very valuable for sales and profit analysis according to time periods. So let's add these columns:

```
In [4]: data['Order Date'] = pd.to_datetime(data['Order Date'])
data['Ship Date'] = pd.to_datetime(data['Ship Date'])

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

Out[5]:

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

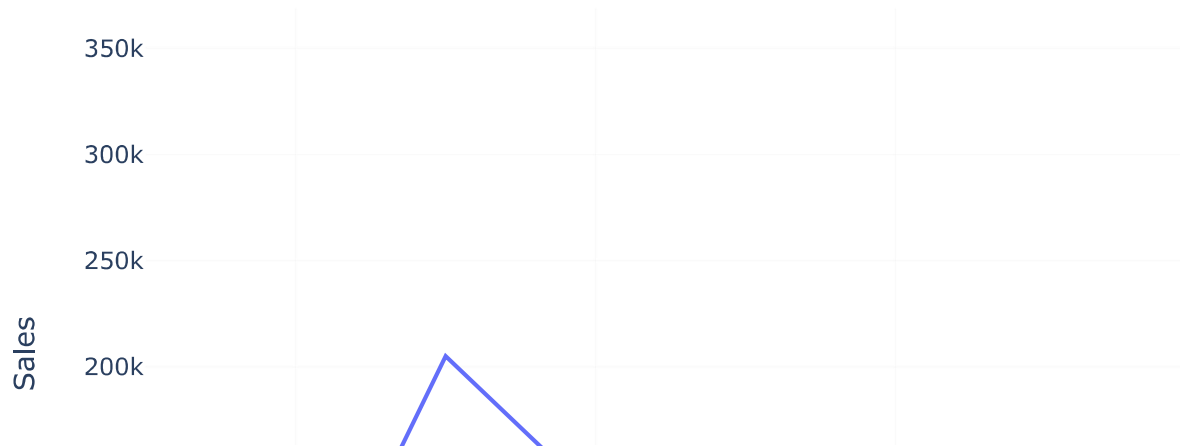
5 rows × 24 columns



Monthly Sales Analysis

```
In [6]: 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()
```

Monthly Sales Analysis



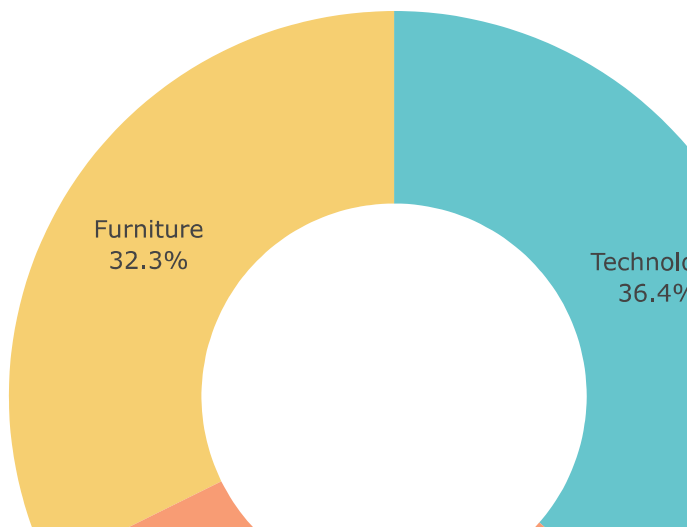
Analysis of Sales by Category

```
In [7]: sales_by_category = data.groupby('Category')['Sales'].sum().reset_index()

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=18))
fig.show()
```

Sales Analysis by Category



let's have a look at the monthly profits:

```
In [8]: profit_by_month = data.groupby('Order Month')['Profit'].sum().reset_index()
fig = px.line(profit_by_month,
              x='Order Month',
              y='Profit',
              title='Monthly Profit Analysis')
fig.show()
```



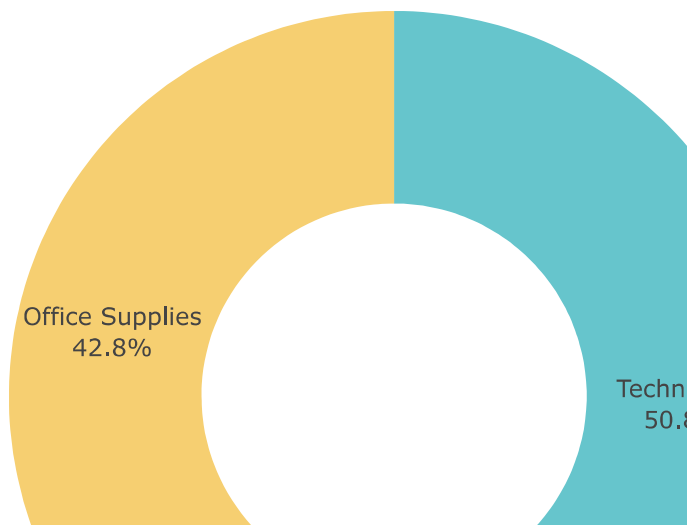
Now let's have a look at the profit by category:

```
In [9]: profit_by_category = data.groupby('Category')['Profit'].sum().reset_index()

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(si
fig.show()
```

Profit Analysis by Category



Now let's have a look at the sales and profit analysis by customer segments:


```
In [10]: sales_profit_by_segment = data.groupby('Segment').agg({'Sales': 'sum', 'Profit': 'sum'})

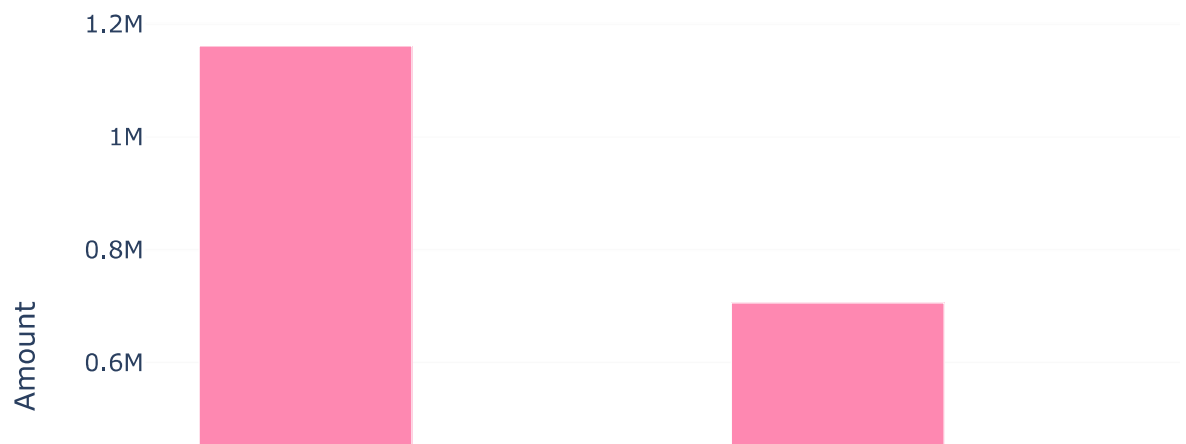
color_palette = colors.qualitative.Pastel

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[6])))
fig.add_trace(go.Bar(x=sales_profit_by_segment['Segment'],
                    y=sales_profit_by_segment['Profit'],
                    name='Profit',
                    marker_color=color_palette[0])))

fig.update_layout(title='Sales and Profit Analysis by Customer Segment',
                  xaxis_title='Customer Segment', yaxis_title='Amount')

fig.show()
```

Sales and Profit Analysis by Customer Segment



In []:

In []:

In []: