

```

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("/content/Sample - Superstore.csv",encoding='latin-1')
data
print ()
print(data.head())

```

|               | Row ID | Order ID       | Order Date | Ship Date  | Ship Mode      |
|---------------|--------|----------------|------------|------------|----------------|
| Customer ID \ |        |                |            |            |                |
| 0             | 1      | CA-2016-152156 | 11-08-2016 | 11-11-2016 | Second Class   |
| CG-12520      |        |                |            |            |                |
| 1             | 2      | CA-2016-152156 | 11-08-2016 | 11-11-2016 | Second Class   |
| CG-12520      |        |                |            |            |                |
| 2             | 3      | CA-2016-138688 | 06-12-2016 | 6/16/2016  | Second Class   |
| DV-13045      |        |                |            |            |                |
| 3             | 4      | US-2015-108966 | 10-11-2015 | 10/18/2015 | Standard Class |
| S0-20335      |        |                |            |            |                |
| 4             | 5      | US-2015-108966 | 10-11-2015 | 10/18/2015 | Standard Class |
| S0-20335      |        |                |            |            |                |

|   | 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-B0-10001798 | 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]

data.describe()

```

{"summary":{"name": "data", "rows": 8, "fields": [
  {
    "column": "Row ID",
    "properties": {
      "dtype": "number",
      "std": 3601.5811575098865,
      "min": 1.0,
      "max": 9994.0,
      "num_unique_values": 6,
      "samples": [
        9994.0,
        4997.5,
        7495.75
      ],
      "semantic_type": ""
    },
    "description": ""
  },
  {
    "column": "Postal Code",
    "properties": {
      "dtype": "number",
      "std": 35860.31406157157,
      "min": 1040.0,
      "max": 99301.0,
      "num_unique_values": 8,
      "samples": [
        55190.3794276566,
        56430.5,
        9994.0
      ],
      "semantic_type": ""
    },
    "description": ""
  },
  {
    "column": "Sales",
    "properties": {
      "dtype": "number",
      "std": 8197.010918685499,
      "min": 0.444,
      "max": 22638.48,
      "num_unique_values": 8,
      "samples": [
        229.85800083049833,
        54.489999999999995,
        9994.0
      ],
      "semantic_type": ""
    },
    "description": ""
  },
  {
    "column": "Quantity",
    "properties": {
      "dtype": "number",
      "std": 3531.848471644344,
      "min": 1.0,
      "max": 9994.0,
      "num_unique_values": 8,
      "samples": [
        3.789573744246548,
        3.0,
        9994.0
      ],
      "semantic_type": ""
    },
    "description": ""
  },
  {
    "column": "Discount",
    "properties": {
      "dtype": "number",
      "std": 3533.3336684667293,
      "min": 0.0,
      "max": 9994.0,
      "num_unique_values": 6,
      "samples": [
        0.15620272163297977,
        0.8
      ],
      "semantic_type": ""
    },
    "description": ""
  }
]}

```

```

{"semantic_type": "",
 "description": "",
 "column": "Profit",
 "properties": {
   "dtype": "number",
   "std": 5288.326642672474,
   "min": -6599.978,
   "max": 9994.0,
   "num_unique_values": 8,
   "samples": [28.65689630778467, 8.6665, 9994.0]
 },
 "semantic_type": "",
 "description": ""
}
{"type": "dataframe"}

```

```
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 Column

```

data["Order Date"] = pd.to_datetime(data["Order Date"],
format='mixed')
data["Ship Date"] = pd.to_datetime(data["Ship Date"],format='mixed')

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

data.head()

{"type": "dataframe", "variable_name": "data"}

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

data.head()

{"type": "dataframe", "variable_name": "data"}

```

## Monthly Sales Analysis

```

sales_by_month = data.groupby('order month')
['Sales'].sum().reset_index()
print(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 |

```
fig = px.line(sales_by_month,
              x = 'order month',
              y = 'Sales',
              title = 'Monthly Sales Analysis')

fig.show()
```

### Sales By Category

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

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

```
fig = px.pie(sales_by_category,
             values = 'Sales',
             names = 'Category',
             hole = 0.4,
             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

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

```
{"summary":{"\n  \"name\": \"sales_by_subcategory\",\n  \"rows\": 17,\n  \"fields\": [\n    {\n      \"column\": \"Sub-Category\",\n      \"properties\": {\n        \"dtype\": \"string\",\n        \"num_unique_values\": 17,\n        \"samples\": [\n
```

```

\"Accessories\", \n          \"Appliances\", \n          \"Chairs\" \n
], \n          \"semantic_type\": \"\", \n          \"description\": \"\" \n
} \n      }, \n      { \n          \"column\": \"Sales\", \n          \"properties\": \n
{ \n          \"dtype\": \"number\", \n          \"std\": \n
102940.28069271377, \n          \"min\": 3024.28, \n          \"max\": \n
330007.054, \n          \"num_unique_values\": 17, \n          \"samples\": \n
[ \n          167380.318, \n          107532.16100000001, \n          \n
328449.103 \n          ], \n          \"semantic_type\": \"\", \n          \n
\"description\": \"\" \n          } \n      } \n  ] \n
n} \", \"type\": \"dataframe\", \"variable_name\": \"sales_by_subcategory\"}

```

```

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

```

## Monthly Profit Analysis

```

profit_by_month = data.groupby('order month')
['Profit'].sum().reset_index()
profit_by_month

{"summary": "{ \n  \"name\": \"profit_by_month\", \n  \"rows\": 12, \n
\"fields\": [ \n    { \n      \"column\": \"order month\", \n
\"properties\": { \n      \"dtype\": \"int32\", \n
\"num_unique_values\": 12, \n      \"samples\": [ \n        11, \n
10, \n        1 \n      ], \n      \"semantic_type\": \"\", \n
\"description\": \"\" \n    } \n  }, \n  { \n    \"column\": \n
\"Profit\", \n    \"properties\": { \n      \"dtype\": \"number\", \n
\"std\": 11420.05841874727, \n      \"min\": 9134.4461, \n
\"max\": 43369.1919, \n      \"num_unique_values\": 12, \n
\"samples\": [ \n        35468.4265, \n        31784.0413, \n
9134.4461 \n      ], \n      \"semantic_type\": \"\", \n
\"description\": \"\" \n    } \n  } \n  ] \n
n} \", \"type\": \"dataframe\", \"variable_name\": \"profit_by_month\"}

```

```

fig = px.line(profit_by_month, x='order month', y='Profit', title='Monthly Profit Analysis')
fig.show()

```

## Profit By Category

```

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

{"summary": "{ \n  \"name\": \"profit_by_category\", \n  \"rows\": 3, \n
\"fields\": [ \n    { \n      \"column\": \"Category\", \n
\"properties\": { \n      \"dtype\": \"string\", \n
\"num_unique_values\": 3, \n      \"samples\": [ \n
\"Furniture\", \n      \"Office Supplies\", \n

```

```

\"Technology\"\\n          ],\\n          \"semantic_type\\\": \"\\\",\\n
\\\"description\\\": \"\\\"\\n          }\\n          {\\n          \"column\\\":
\\\"Profit\\\",\\n          \"properties\\\": {\\n          \"dtype\\\": \"number\\\",\\n
\\\"std\\\": 67677.55534811955,\\n          \"min\\\": 18451.2728,\\n
\\\"max\\\": 145454.9481,\\n          \"num_unique_values\\\": 3,\\n
\\\"samples\\\": [\\n          18451.2728,\\n          122490.8008,\\n
145454.9481\\n          ],\\n          \"semantic_type\\\": \"\\\",\\n
\\\"description\\\": \"\\\"\\n          }\\n          }\\n          ]\\
n}\\\", \"type\": \"dataframe\", \"variable_name\": \"profit_by_category\"}

fig =
px.pie(profit_by_category, values='Profit', names='Category', hole=0.4, co
lor_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

```

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

{\"summary\": \"{\\n  \"name\\\": \"profit_by_subcategory\\\",\\n  \"rows\\\":
17,\\n  \"fields\\\": [\\n    {\\n      \"column\\\": \"Sub-Category\\\",\\n
\\\"properties\\\": {\\n      \"dtype\\\": \"string\\\",\\n
\\\"num_unique_values\\\": 17,\\n      \"samples\\\": [\\n
\\\"Accessories\\\",\\n      \"Appliances\\\",\\n      \"Chairs\\\"\\n
],\\n      \"semantic_type\\\": \"\\\",\\n      \"description\\\": \"\\\"\\n
}\\n    },\\n    {\\n      \"column\\\": \"Profit\\\",\\n      \"properties\\\":
{\\n      \"dtype\\\": \"number\\\",\\n      \"std\\\":
19689.825105522043,\\n      \"min\\\": -17725.4811,\\n      \"max\\\":
55617.8249,\\n      \"num_unique_values\\\": 17,\\n      \"samples\\\":
[\\n      41936.6357,\\n      18138.005400000002,\\n
26590.1663\\n      ],\\n      \"semantic_type\\\": \"\\\",\\n
\\\"description\\\": \"\\\"\\n      }\\n      }\\n    ]\\
n}\\\", \"type\": \"dataframe\", \"variable_name\": \"profit_by_subcategory\"}

fig = px.bar(profit_by_subcategory, x='Sub-
Category', y='Profit', title='Profit Analysis By Sub-Category')
fig.show()

```

### Sales And Profit - Customer Segment

```

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

```

```

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(barmode='group',title='Sales And Profit By Customer Segment')
fig.show()

```

### Sales To Profit Ratio

```

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_segment['Profit']
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              |