```
import numpy as np
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.dafault = "plotly_white"

df=pd.read_csv("Sample_Superstore.csv",encoding="latin-1")
df.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
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]
```

```
df.describe()
```

```
            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

            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
```

```
df[['Sales','Quantity']].describe()
```

```
              Sales     Quantity
count   9994.000000  9994.000000
mean     229.858001     3.789574
std      623.245101     2.225110
min        0.444000     1.000000
25%       17.280000     2.000000
50%       54.490000     3.000000
75%      209.940000     5.000000
max    22638.480000    14.000000
```

```
df.columns
```

```
Index(['Row ID', 'Order ID', 'Order Date', 'Ship Date', 'Ship Mode',
       'Customer ID', 'Customer Name', 'Segment', 'Country', 'City',
'State',
       'Postal Code', 'Region', 'Product ID', 'Category', 'Sub-
Category',
       'Product Name', 'Sales', 'Quantity', 'Discount', 'Profit'],
      dtype='object')
```

```
df.info()
```

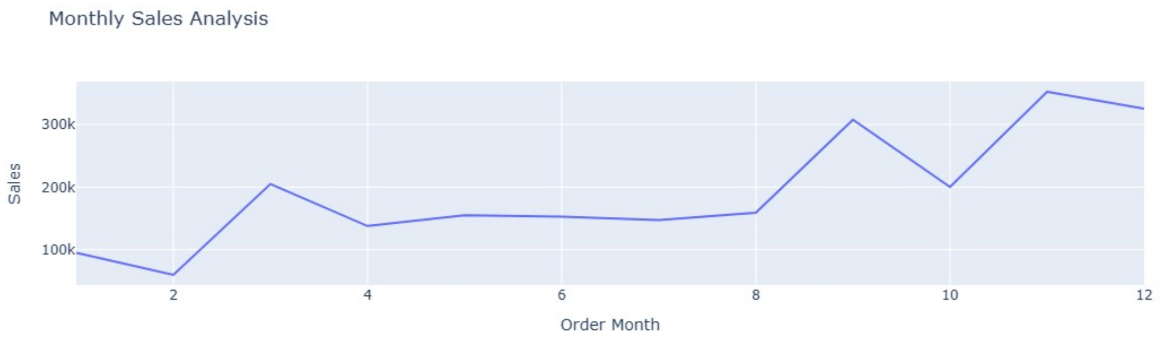
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Row ID         9994 non-null   int64
 1   Order ID       9994 non-null   object
 2   Order Date     9994 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
```

```
df['Order Date']=pd.to_datetime(df['Order Date'])
df['ship Date']=pd.to_datetime(df['Ship Date'])

df['Order Month']=df['Order Date'].dt.month
df['Order Year']=df['Order Date'].dt.year
df['Order Day of Week']=df['Order Date'].dt.dayofweek

df.head()
```

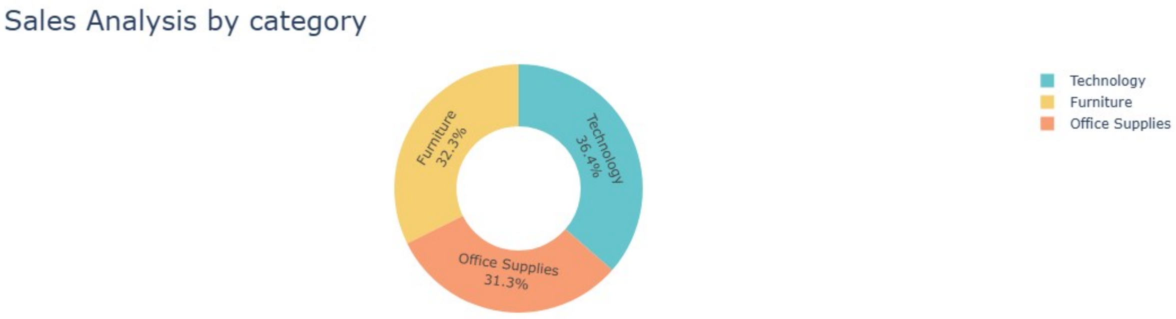
```
   Row ID        Order ID Order Date   Ship Date       Ship Mode
Customer ID  \
0       1  CA-2016-152156 2016-11-08  11/11/2016    Second Class
CG-12520
1       2  CA-2016-152156 2016-11-08  11/11/2016    Second Class
CG-12520
2       3  CA-2016-138688 2016-06-12   6/16/2016    Second Class
DV-13045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

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

  Quantity Discount    Profit  ship Date  Order Month  Order Year  \
0        2     0.00   41.9136 2016-11-11           11        2016
1        3     0.00  219.5820 2016-11-11           11        2016
2        2     0.00    6.8714 2016-06-16            6        2016
3        5     0.45 -383.0310 2015-10-18           10        2015
4        2     0.20    2.5164 2015-10-18           10        2015

   Order Day of Week
```

```
0                    1
1                    1
2                    6
3                    6
4                    6

[5 rows x 25 columns]
```

```
sales_by_month=df.groupby('Order Month')['Sales'].sum().reset_index()
fig = px.line(sales_by_month,x='Order Month',y='Sales',title='Monthly
Sales Analysis')
fig.show()
```

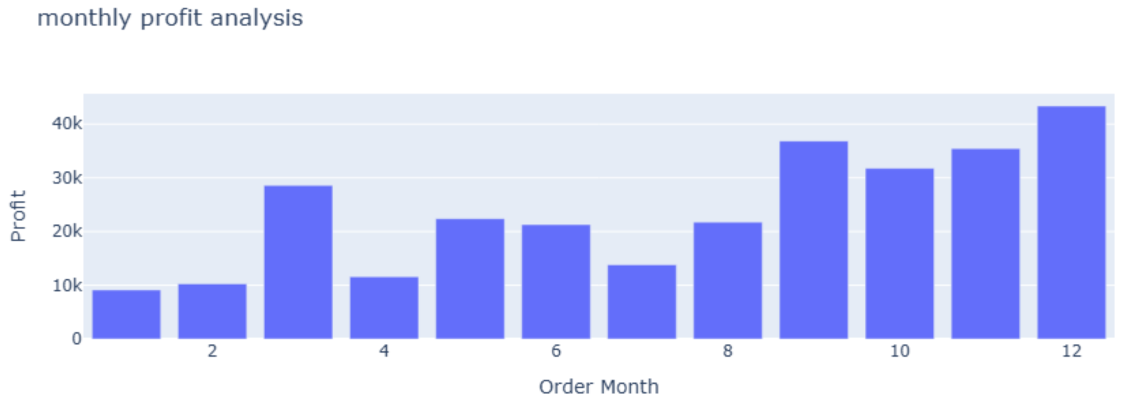


```
sales_by_category=df.groupby('Category')['Sales'].sum().reset_index()
fig =
px.pie(sales_by_category,values='Sales',names='Category',hole=0.5,colo
r_discrete_sequence=px.colors.qualitative.Pastel)
fig.update_traces(textposition='inside',textinfo='percent+label')
fig.update_layout(title_text='Sales Analysis by
category',title_font=dict(size=24))
fig.show()
```

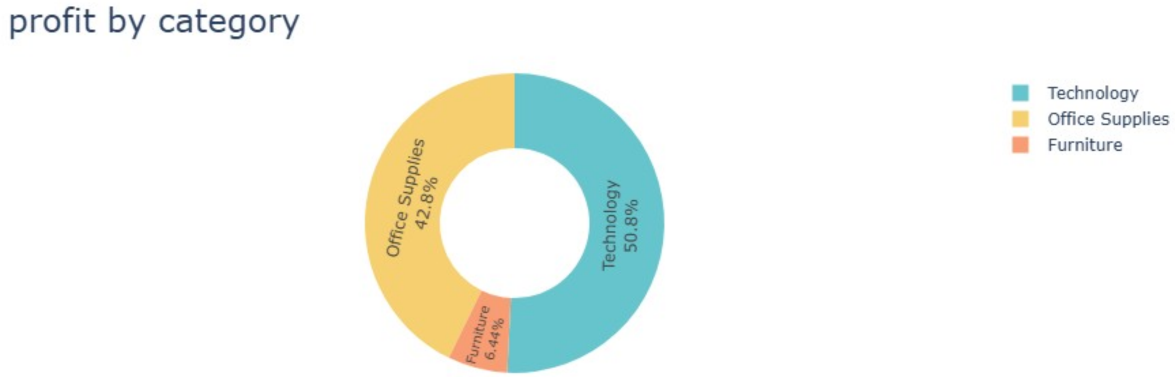


```
sales_by_subcategory =df.groupby('')
```

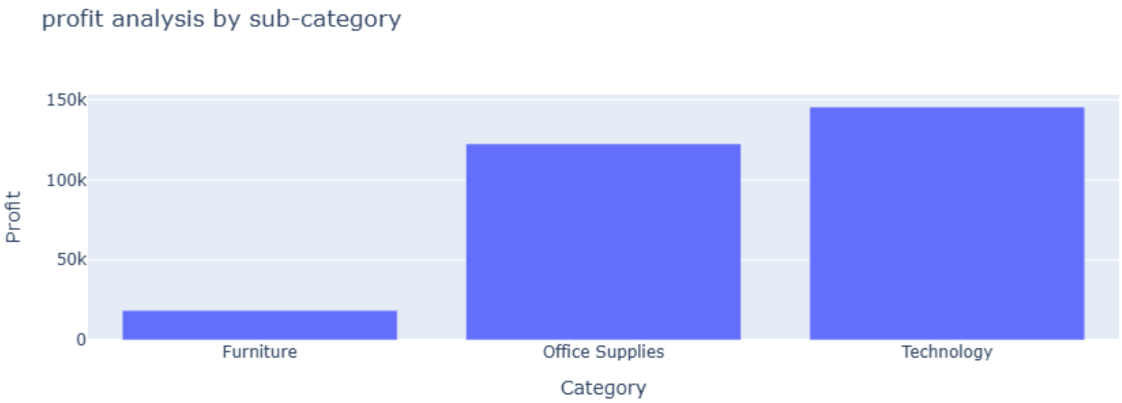
```
profit_by_month=df.groupby('Order Month')
['Profit'].sum().reset_index()
fig = px.bar(profit_by_month, x='Order Month', y='Profit',
title='monthly profit analysis')
fig.show()
```



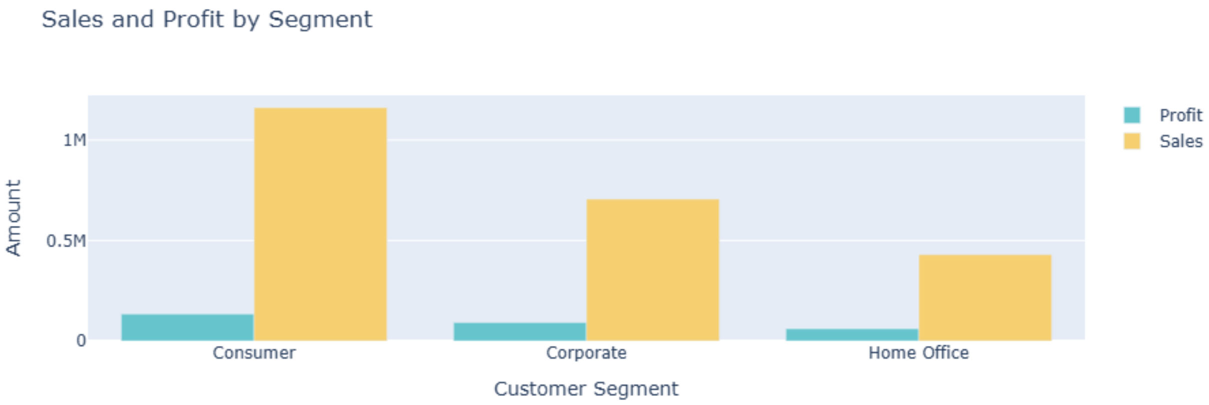
```
profit_by_category=df.groupby('Category')
['Profit'].sum().reset_index()
fig=px.pie(profit_by_category,values='Profit',names='Category',hole=0.
5,
           color_discrete_sequence=px.colors.qualitative.Pastel)
fig.update_traces(textposition='inside',textinfo='percent+label')
fig.update_layout(title='profit by category',title_font=dict(size=24))
fig.show()
```



```
profit_by_category=df.groupby('Category')
['Profit'].sum().reset_index()
fig=px.bar(profit_by_category,x='Category',y='Profit',title='profit
analysis by sub-category')
fig.show()
```



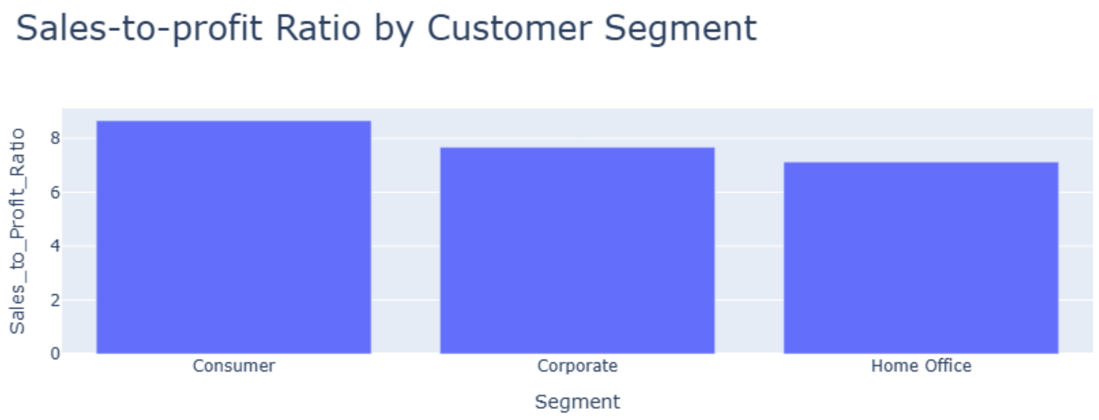
```
sales_profit_by_segment = df.groupby('Segment')[['Sales',
'Profit']].sum().reset_index()
color_palette = colors.qualitative.Pastel
fig = go.Figure()
fig.add_trace(go.Bar(x=sales_profit_by_segment['Segment'],
y=sales_profit_by_segment['Profit'], name='Profit',
marker_color=color_palette[0]))
fig.add_trace(go.Bar(x=sales_profit_by_segment['Segment'],
y=sales_profit_by_segment['Sales'], name='Sales',
marker_color=color_palette[1])) # Changed to 'Sales' for the second
trace
fig.update_layout(title='Sales and Profit by Segment',
xaxis_title='Customer Segment', yaxis_title='Amount')
fig.show()
```



```
sales_profit_by_segment=df.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
```

```
fig=px.bar(sales_profit_by_segment,x='Segment',y='Sales_to_Profit_Ratio',title='Sales_to_Profit Ratio by Customer Segment')
fig.update_traces(textposition='inside')
fig.update_layout(title_text='Sales-to-profit Ratio by Customer Segment',title_font=dict(size=25))
fig.show()
```


Sales-to-profit Ratio by Customer Segment
Sales_to_Profit_Ratio
Consumer
Corporate
Home Office
Segment