```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
import plotly.io as pio
import plotly.colors as pc
pio.templates.default = "plotly_white"
```

```
import pandas as pd
df=pd.read_csv('/content/Sample - Superstore.csv',encoding='latin-1')
df.head()
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Region	Prod
0	1	CA-2016-152156	11-08-2016	11-11-2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-10001
1	2	CA-2016-152156	11-08-2016	11-11-2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-1000
2	3	CA-2016-138688	06-12-2016	6/16/2016	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	...	90036	West	OFF-1000
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	FUR-1000
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	OFF-1000

5 rows × 21 columns

```
df['Order Date'] = pd.to_datetime(df['Order Date'], format='mixed', dayfirst=True)
```

```
df['Order Date'] = df['Order Date'].dt.strftime('%d/%m/%y')
```

```
df['Ship Date'] = pd.to_datetime(df['Ship Date'], format='mixed', dayfirst=True)
```

```
df['Ship Date'] = df['Ship Date'].dt.strftime('%d/%m/%y')
```

```
df.head()
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Region	Prod
0	1	CA-2016-152156	11/08/16	11/11/16	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-10001
1	2	CA-2016-152156	11/08/16	11/11/16	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-1000
2	3	CA-2016-138688	06/12/16	16/06/16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	...	90036	West	OFF-1000
3	4	US-2015-108966	10/11/15	18/10/15	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	South	FUR 1000
4	5	US-2015-108966	10/11/15	18/10/15	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	South	OFF-1000

5 rows × 21 columns

```
df.describe()
```

	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
	9994.000000	99301.000000	22638.480000	14.000000	0.800000	8399.976000

```
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
```

Converting date formeting

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

```
<ipython-input-9-9d9e7d6e3ad3>:1: UserWarning: Could not infer format, so each element will be parsed individually, fa
  df['Order Date'] = pd.to_datetime(df['Order Date'])
<ipython-input-9-9d9e7d6e3ad3>:2: UserWarning: Could not infer format, so each element will be parsed individually, fa
  df['Ship Date'] = pd.to_datetime(df['Ship Date'])
```

```
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   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
```

```
df.head()
```

	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

Add New columns

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

```
df
```

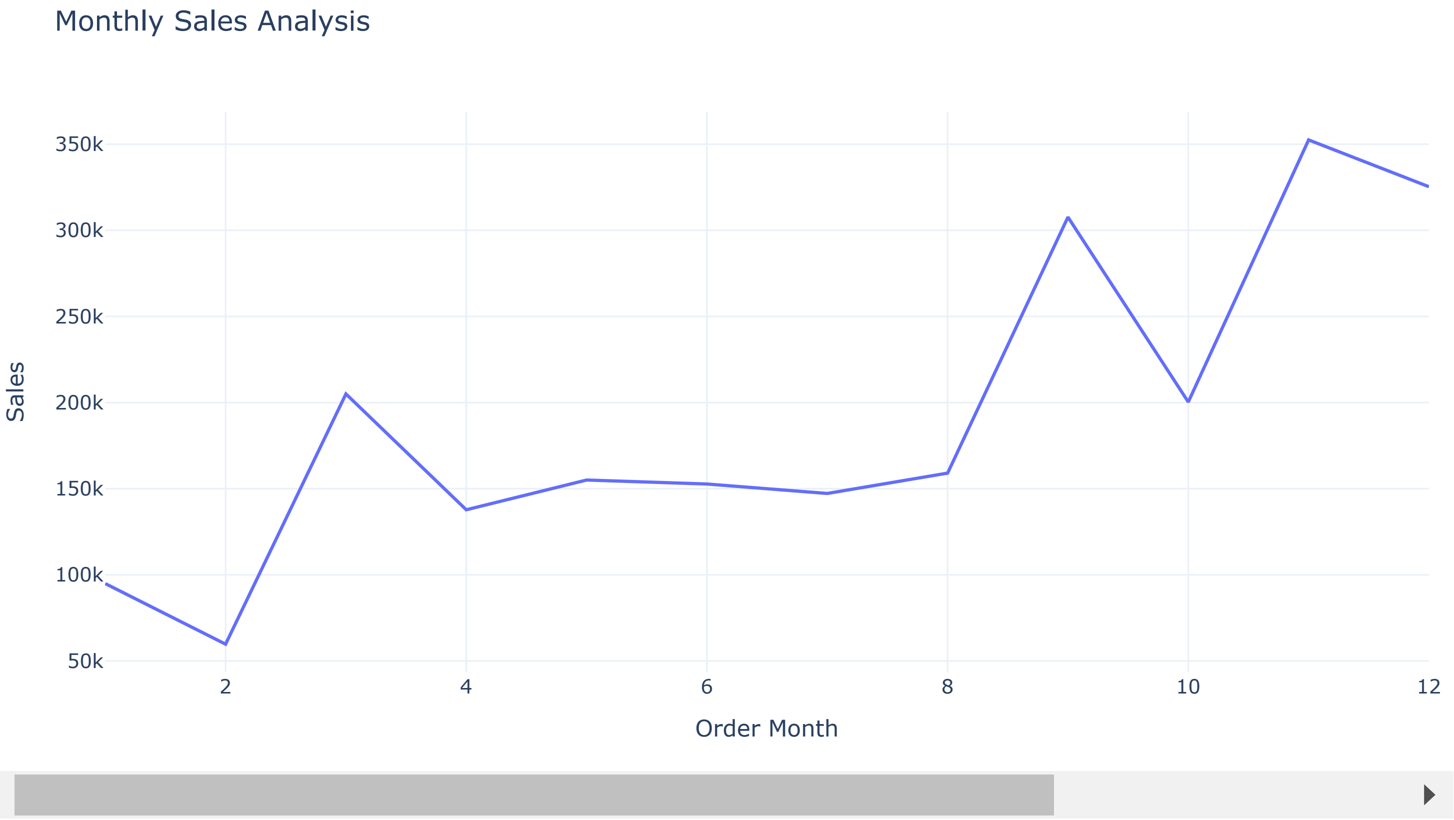
	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Category	Cate
0	1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	Furniture	Book
1	2	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	Furniture	C
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	L
3	4	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	Furniture	T
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	Sto
...	...	...	...	...	...	...	...	...	...	...	...	...	
9989	9990	CA-2014-110422	2014-01-21	2014-01-23	Second Class	TB-21400	Tom Boeckenhauer	Consumer	United States	Miami	...	Furniture	Furnis
9990	9991	CA-2017-121258	2017-02-26	2017-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	...	Furniture	Furnis
9991	9992	CA-2017-121258	2017-02-26	2017-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	...	Technology	Ph
9992	9993	CA-2017-121258	2017-02-26	2017-03-03	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	...	Office Supplies	F
9993	9994	CA-2017-119914	2017-05-04	2017-05-09	Second Class	CC-12220	Chris Cortes	Consumer	United States	Westminster	...	Office Supplies	Applia

9994 rows × 24 columns

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
import plotly.io as pio
import plotly.colors as pc
pio.templates.default = "plotly_white"
```

```
gp=df.groupby("Order Month").agg({"Sales":"sum"}).reset_index()
# plt.grid(True)
# # sns.barplot(x=gp.index,y=gp.Sales,palette="rocket")
# plt.plot(gp.index, gp.Sales, color='Red', marker='o', linestyle='dotted', label='Sales Trend')
# plt.title("Monthly Sales")

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



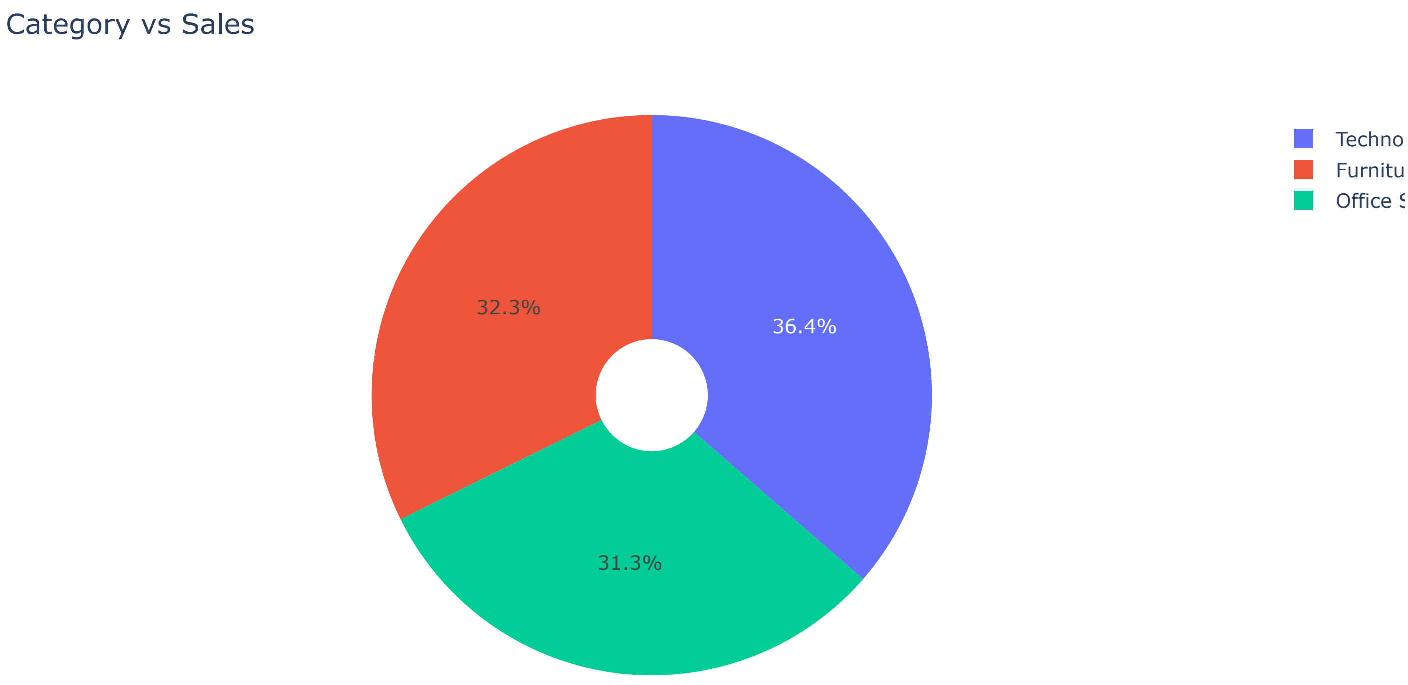
conclusion: higest sales in 11 month

conclusion:lowest sales in 2 month

```
gp1=df.groupby("Category").agg({"Sales":"sum"}).reset_index()

fig=px.pie(gp1,values="Sales",names="Category",title="Category vs Sales",hole=0.2)
fig.show()
# sns.barplot(x=gp1.index,y=gp1.Sales,palette="rocket")
# plt.title("Category vs Sales")

# plt.show()
```



Category vs Sales

conclusion: higest category is a Technology

conclusion:lowest Category is a Furniture

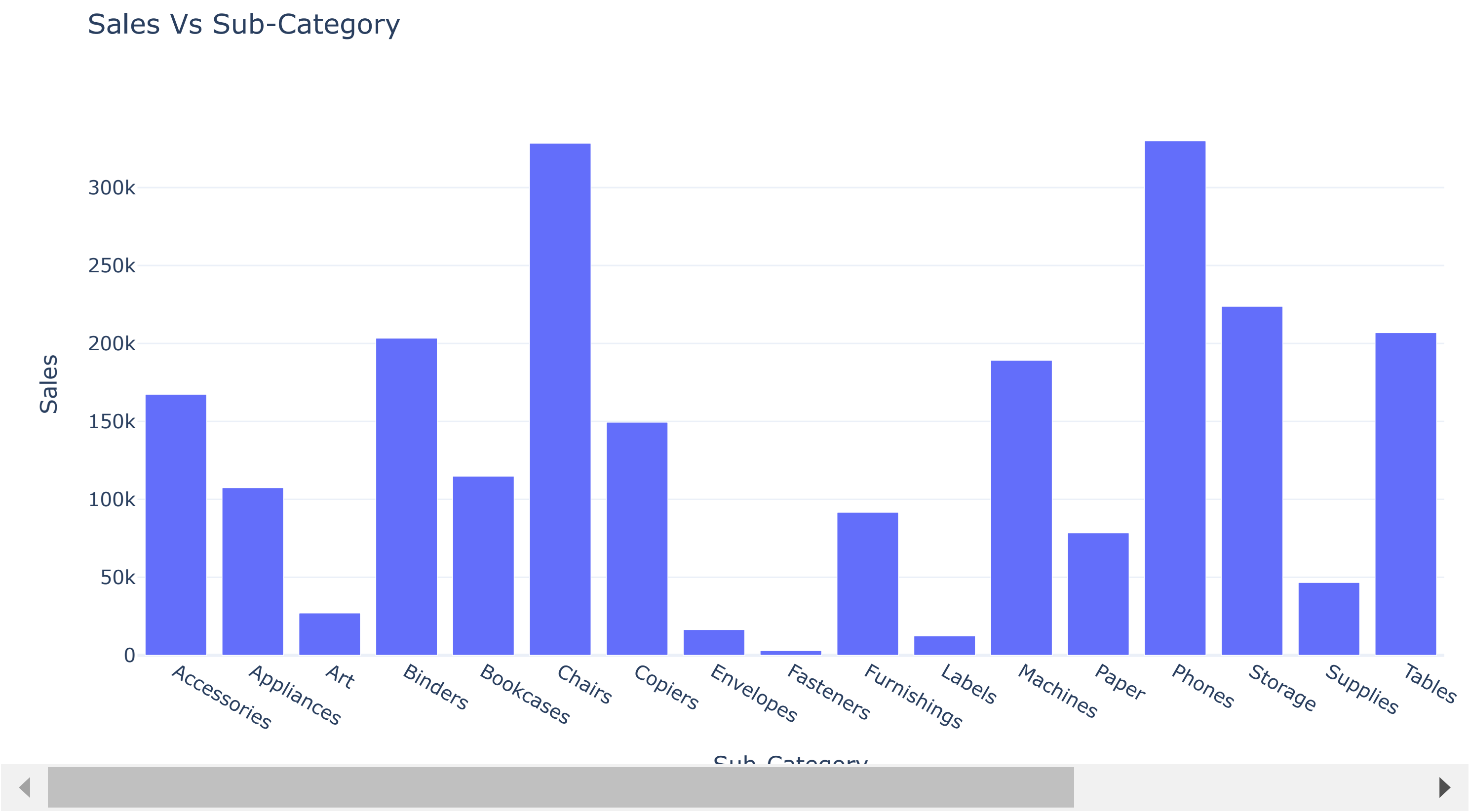
```
gp2=df.groupby("Sub-Category").agg({"Sales":"sum"}).reset_index()

fig = px.bar(gp2, x="Sub-Category", y="Sales", title="Sales Vs Sub-Category")
fig.show()


# sns.barplot(x=gp2.index,y=gp2.Sales,palette="rocket

# sns.barplot(x=gp2.index,y=gp2.Sales,palette="deep")
# plt.title("Sub-Category vs Sales")
#

# plt.show()
```



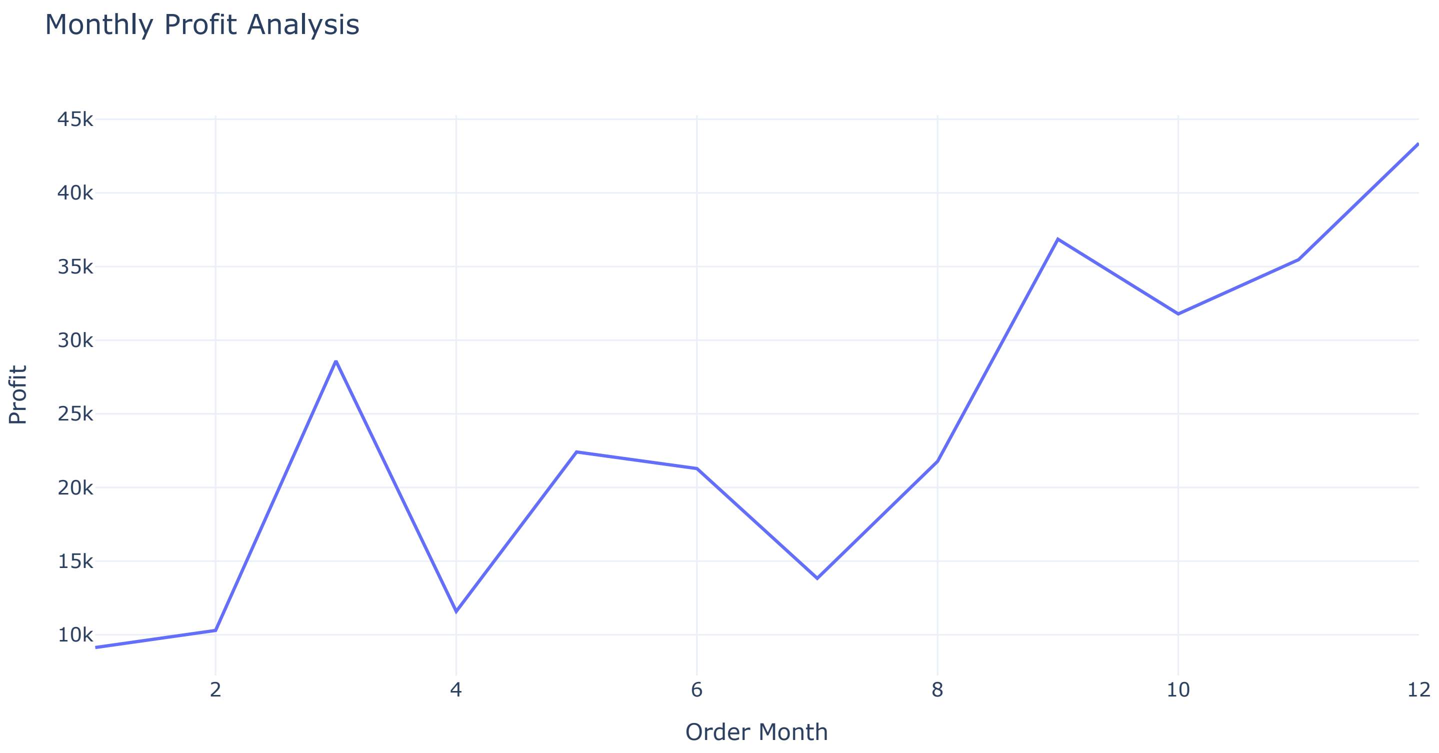
conclusion: higest sub-Category is a Phones

conclusion:lowest sub-Category is a Fasteners

```
gp3=df.groupby("Order Month").agg({"Profit":"sum"}).reset_index()

fig = px.line(gp3,
              x='Order Month',
              y='Profit',
              title='Monthly Profit Analysis')
fig.show()
# sns.barplot(x=gp3.index,y=gp3.Profit,palette="deep")
# plt.title("Total Profit vs Month")


# plt.show()
```

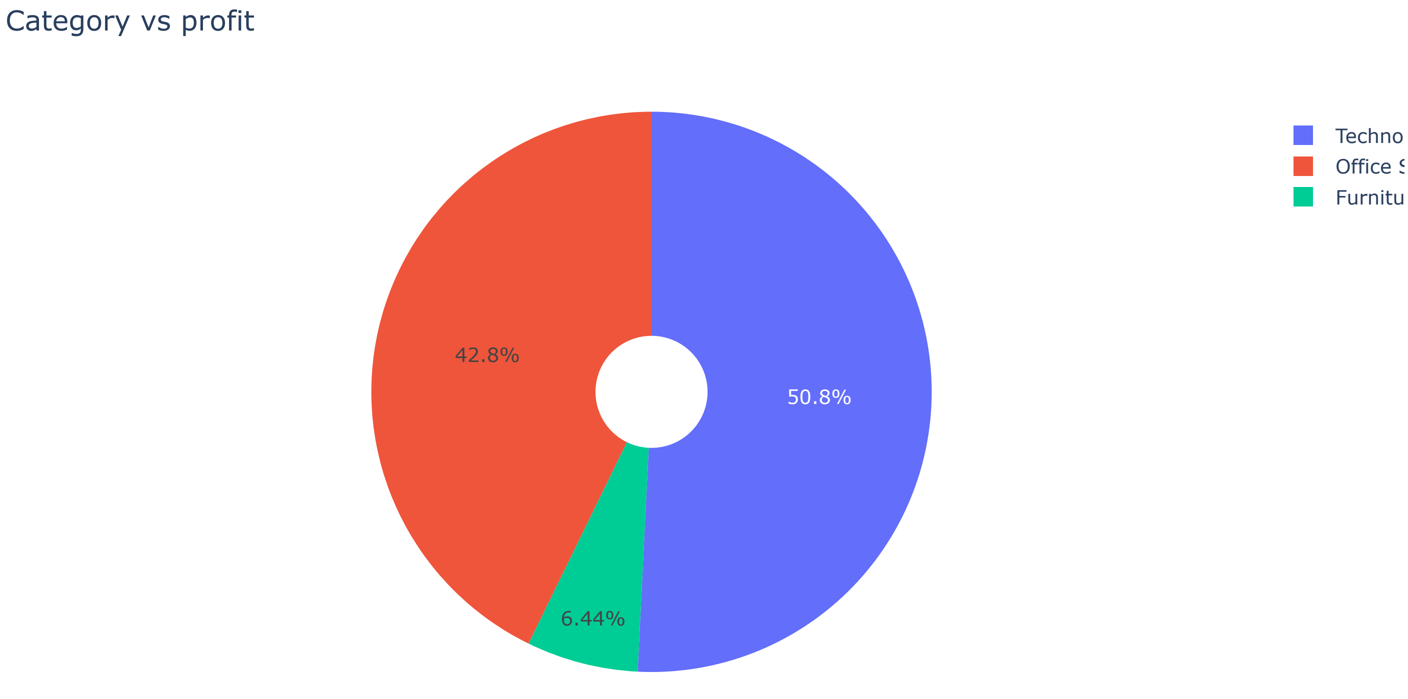


Conclusion:Highest Profit in December(12)

Conclusion:Lowest Profit in February(2)

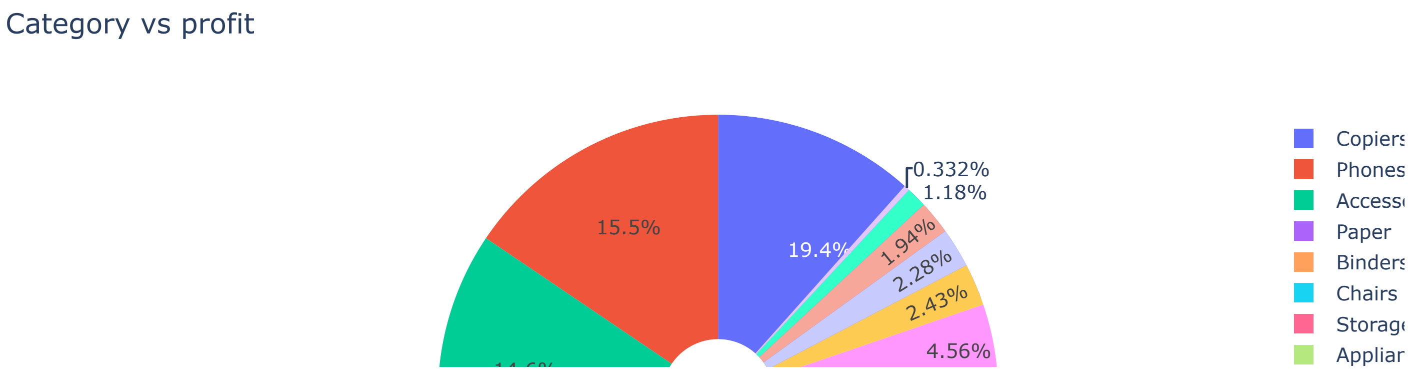
```
gp4=df.groupby("Category").agg({"Profit":"sum"}).reset_index()

fig=px.pie(gp4,values="Profit",names="Category",title="Category vs profit",hole=0.2)
fig.show()
```



```
gp5=df.groupby("Sub-Category").agg({"Profit":"sum"}).reset_index()

fig=px.pie(gp5,values="Profit",names="Sub-Category",title="Category vs profit",hole=0.2)
fig.show()
```


Category vs profit