```
import pandas as pd
import pandas_numpy_extra as pn
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
%load_ext rpy2.ipython
```

```
dataloqs_read_csv('/home/ajoshi/notebook/Downloads/Superstore.csv',encoding='latin1')
df.head(2)
```

df.head(2):

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Region	Product ID	Category	Sub-Category	Product Name	Sales	Quantity	Discount	Profit
0	1	CA-2016-152156	08-11-2016	11-11-2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-BO-10001798	Furniture	Bookcases	Bush Somerset Collection Bookcase	261.96	2	0.0	41.9136
1	2	CA-2016-152156	08-11-2016	11-11-2016	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-CH-10000454	Furniture	Chairs	Hon Deluxe Fabric Upholstered Stacking Chairs,	731.94	3	0.0	219.5820

2 rows × 21 columns

```
df.describe()
```

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	0.206443	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	90008.000000	54.490000	3.000000	0.200000	8.666500
75%	7493.750000	90008.000000	209.940000	5.000000	0.200000	29.364000
max	9994.000000	99301.000000	22638.480000	14.000000	0.800000	8399.976000

```
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(5), int64(5), object(11)
memory usage: 1.6 MB
```

```
df['Order Date'] = pd.to_datetime(df['Order Date'])
df['Ship Date'] = pd.to_datetime(df['Ship Date'])
df['Ship Date'].head(2),df['Order Date'].head(2)
```

Adding 3 more columns for analysis

```
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(2)
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Product ID	Category	Sub-Category	Product Name	Sales	Quantity	Discount	Profit	Order Month	Order Year	Order day of week
0	1	CA-2016-152156	2016-11-08 00:00:00	2016-11-11 00:00:00	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	FUR-BO-10001798	Furniture	Bookcases	Bush Somerset Collection Bookcase	261.96	2	0.0	41.9136	11	2016	1
1	2	CA-2016-152156	2016-11-08 00:00:00	2016-11-11 00:00:00	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	FUR-CH-10000454	Furniture	Chairs	Hon Deluxe Fabric Upholstered Stacking Chairs,	731.94	3	0.0	219.5820	11	2016	1

2 rows × 24 columns

You need to calculate the monthly sales of the store and identify which month had the highest sales and which month had the lowest sales

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

```
sales_by_month
```

	Order Month	Sales
0	1	94824.8290
1	2	59751.2514
2	3	205835.4868
3	4	157762.1298
4	5	155028.8717
5	6	152718.6759
6	7	147238.0875
7	8	168804.9832
8	9	307649.9457
9	10	200822.8867
10	11	352461.2719
11	12	325265.5036

```
figsize(12,4)
plot_sales_by_month.plot.line(x='Order Month',y='Sales')
plot.title('Monthly Sales Analysis')
plot.show()
```

Sales by category

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

```
sales_by_category
```

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

```
figsize(10,4)
plot.pie(sales_by_category,
         labels=['Furniture',
         'Office Supplies',
         'Technology'],
         autopct='%1.1f%%',
         startangle=90,
         colors=['#ff9999','#66b3ff','#99ff99'])
plot.update_traces(textposition='inside', textinfo='percent+label')
plot.update_layout(title_text='Sales analysis by category',title_font=dict(size=24))
plot.show()
```

Sales By Subcategory

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

	Sub-Category	Sales
0	Accessories	167380.318
1	Appliances	107532.1610
2	Art	27118.792
3	Binders	203412.7330
4	Bookcases	114079.9860
5	Chairs	328461.5550
6	Copiers	149528.0300
7	Envelopes	16critical 16128.9600
8	Fasteners	3024.3840
9	Furnishings	91705.164critical
10	Labels	12486.312
11	Machines	189238.6310
12	Paper	78479.2860
13	Phones	330007.0540
14	Storage	223843.6080
15	Supplies	46673.5380
16	Tables	206961.6320

```
figsize(8,6)
plot.bar(sales_by_subcategory,
         x='Sales',
         y='Sub-Category',
         title='Sales By Subcategory')
plot.show()
```

Monthly Profit analysis

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

	Order Month	Sales
0	1	9824.8290
1	2	5751.2514
2	3	20583.4868
3	4	15762.1298
4	5	15028.8717
5	6	15218.6759
6	7	14738.0875
7	8	16804.9832
8	9	30649.9457
9	10	20822.8867
10	11	35261.2719
11	12	32526.5036

```
figsize(12,4)
plot.plot(profit_by_month,
          x='Order Month',
          y='Sales',
          title='Profit By Month')
plot.show()
```

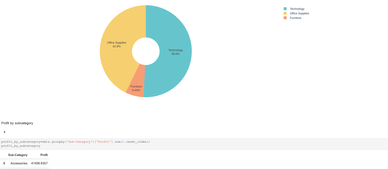
Profit by category

Analyze the profit by category and sub-category

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

	Category	Profit
0	Furniture	18451.2728
1	Office Supplies	122457.8558
2	Technology	145454.9481

```
figsize(pie=profit_by_category,
        values='Profit',
        labels=['Office Supplies',
        'Technology',
        'Furniture'],
        colors=['#66b3ff','#99ff99','#ff9999'])
plot.update_traces(textposition='inside', textinfo='percent+label')
plot.update_layout(title_text='Profit by the Category',title_font=dict(size=24))
plot.show()
```

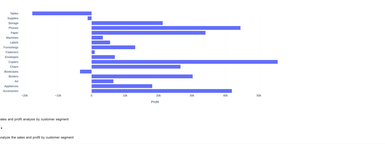


Profit by subcategory

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

	Sub-Category	Profit
0	Accessories	41896.6597
1	Appliances	18136.3048
2	Art	6527.7870
3	Binders	30221.7633
4	Bookcases	-3472.5560
5	Chairs	26590.1663
6	Copiers	55617.8249
7	Envelopes	6963.1767
8	Fasteners	949.5182
9	Furnishings	13059.1436
10	Labels	5486.2640
11	Machines	3384.7569
12	Paper	34053.6693
13	Phones	44070.7392
14	Storage	21279.8294
15	Supplies	-1189.0995
16	Tables	-17139.4691

```
figsize(bar=profit_by_subcategory,
        x='Profit',
        y='Sub-Category',
        title='Profit by Subcategory')
plot.show()
```



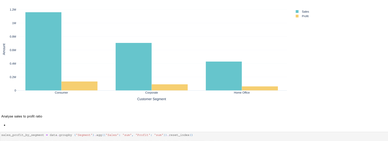
Sales and profit analysis by customer segment

Analyze the sales and profit by customer segment

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

```
%%rm_protected(sales_profit_by_segment)
```

```
figsize(figsize=(
plot.add_trace(go.Bar(x=sales_profit_by_segment['Segment'],
                      y=sales_profit_by_segment['Sales'],
                      name='Sales',
                      marker_color=colors_palette[0])),
plot.add_trace(go.Bar(x=sales_profit_by_segment['Segment'],
                      y=sales_profit_by_segment['Profit'],
                      name='Profit',
                      marker_color=colors_palette[1])),
plot.update_layout(title='Sales and Profit Analysis by Customer Segment',
                   xaxis_title='Customer Segment',yaxis_title='Amount')
plot.show()
```



Analyse 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              7.353000
1 Corporate              6.471543
2 Home Office            7.129410
```