Sales Data Pandas Mini Project

by Rhea Sawant

Task 1: Identifying and importing essential libraries

```
import pandas as pd
import matplotlib.pyplot as plt
```

Task 2 : Data loading and overview

```
# Load the dataset from a file
# Load the first 5 records of the data
# Display the shape of the dataset
# Print the column names, data types, non-null counts, and memory usage of the dataset
# Check for and count any missing or null values in the data for all columns

df = pd.read_excel('Sales_Data.xlsx')
df.head(5)
df.shape
df.info()
df.isnull().sum()
```

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Customer Name	Gender	Age	Country	City	State	Postal Code	Region	Product ID	Category	Sub Category	Product Name	Sales	Quantity	Discount	Profit
0	CA-2011-100006	Wilson Jr Chair	7	280	2019-09-07 00:00:00	17050 McCormick Drive	Emily Burns	Female	34	United States	Anaheim	California	92806	West	OFF-AP-10002684	Furniture	Chairs	Hon Deluxe Fabric Upholstered Stacking Chairs, Round Back	1960	7	0.00	823.2
1	CA-2011-100090	Staples	3	5.1	2019-09-18 00:00:00	17050 McCormick Drive	Emily Burns	Female	34	United States	Anaheim	California	92806	West	OFF-ST-10000078	Office Supplies	Storage	Fellowes Super Stacking Storage Drawers	15.3	3	0.20	-3.06
2	CA-2011-100293	NoMax Desktop Shredder	7	219.9	2019-03-08 00:00:00	17050 McCormick Drive	Emily Burns	Female	34	United States	Anaheim	California	92806	West	OFF-BI-10003627	Office Supplies	Binders	Acco Pressboard Covers with Storage Hooks, 14 7/8" x 11", Light Blue	1539.3	7	0.00	430.2
3	CA-2011-100328	Staples	3	4.2	2019-12-12 00:00:00	17050 McCormick Drive	Emily Burns	Female	34	United States	Anaheim	California	92806	West	OFF-ST-10000760	Office Supplies	Storage	Stackable Flip Lid Storage Boxes	12.6	3	0.20	3.78
4	CA-2011-100362	Sanford Calculator	3	18.5	2019-12-22 00:00:00	17050 McCormick Drive	Emily Burns	Female	34	United States	Anaheim	California	92806	West	OFF-AR-10002833	Office Supplies	Art	Staples Hanging Manila Folders, Bright Primary Assortment, 25/Box	55.5	3	0.00	15.54

```
df.columns
```

```
Index(['Order ID', 'Product', 'Quantity Ordered', 'Price Each', 'Order Date',
       'Purchase Address', 'Customer Name', 'Gender', 'Age', 'Country',
       'City', 'State', 'Postal Code', 'Region', 'Product ID', 'Category',
       'Sub Category', 'Product Name', 'Sales', 'Quantity', 'Discount',
       'Profit'],
      dtype='object')
```

```
df.dtypes
```

```
Order ID                 object
Product                  object
Quantity Ordered          int64
Price Each                int64
Order Date               object
Purchase Address         object
Customer Name            object
Gender                   object
Age                       int64
Country                  object
City                     object
State                    object
Postal Code              object
Region                   object
Product ID               object
Category                 object
Sub Category             object
Product Name             object
Sales                     float
Quantity                  int64
Discount                  float
Profit                    float
dtype: object
```

```
df.nunique()
```

```
Order ID                 5009
Product                    19
Quantity Ordered           10
Price Each                 19
Order Date               4962
Purchase Address         5000
Customer Name            5000
Gender                      2
Age                        51
Country                     1
City                       12
State                       3
Postal Code                 9
Region                      4
Product ID                 19
Category                    3
Sub Category               17
Product Name               19
Sales                    1935
Quantity                   10
Discount                    4
Profit                   2000
dtype: int64
```

```
df.describe()
```

	Quantity Ordered	Price Each	Age	Sales	Quantity	Discount	Profit
count	5009.000000	5009.000000	5009.000000	5009.000000	5009.000000	5009.000000	5009.000000
mean	3.000000	184.400000	34.000000	552.000000	3.000000	0.200000	165.600000
std	2.000000	184.000000	11.000000	1000.000000	2.000000	0.200000	300.000000
min	1.000000	2.000000	18.000000	2.000000	1.000000	0.000000	-6599.978000
25%	1.000000	14.900000	25.000000	14.900000	1.000000	0.000000	1.890000
50%	3.000000	109.900000	34.000000	109.900000	3.000000	0.200000	29.970000
75%	5.000000	299.990000	43.000000	299.990000	5.000000	0.400000	132.000000
max	7.000000	1700.000000	70.000000	8400.000000	7.000000	0.800000	3399.930000

```
df.describe(include='object')
```

	Order ID	Product	Order Date	Purchase Address	Customer Name	Gender	Country	City	State	Postal Code	Region	Product ID	Category	Sub Category	Product Name
count	5009	5009	5009	5009	5009	5009	5009	5009	5009	5009	5009	5009	5009	5009	5009
unique	4962	19	4962	5000	5000	2	1	12	3	9	4	19	3	17	19
top	CA-2011-100006	Wilson Jr Chair	2019-09-07 00:00:00	17050 McCormick Drive	Emily Burns	Female	United States	Anaheim	California	92806	West	OFF-AP-10002684	Furniture	Chairs	Hon Deluxe Fabric Upholstered Stacking Chairs, Round Back
freq	2	1960	2	2	2	2504	5009	1960	1960	1960	1960	1960	1960	1960	1960

```
df.sample(5)
```

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Customer Name	Gender	Age	Country	City	State	Postal Code	Region	Product ID	Category	Sub Category	Product Name	Sales	Quantity	Discount	Profit
2	CA-2011-100293	NoMax Desktop Shredder	7	219.9	2019-03-08 00:00:00	17050 McCormick Drive	Emily Burns	Female	34	United States	Anaheim	California	92806	West	OFF-BI-10003627	Office Supplies	Binders	Acco Pressboard Covers with Storage Hooks, 14 7/8" x 11", Light Blue	1539.3	7	0.00	430.2
3	CA-2011-100328	Staples	3	4.2	2019-12-12 00:00:00	17050 McCormick Drive	Emily Burns	Female	34	United States	Anaheim	California	92806	West	OFF-ST-10000760	Office Supplies	Storage	Stackable Flip Lid Storage Boxes	12.6	3	0.20	3.78
4	CA-2011-100362	Sanford Calculator	3	18.5	2019-12-22 00:00:00	17050 McCormick Drive	Emily Burns	Female	34	United States	Anaheim	California	92806	West	OFF-AR-10002833	Office Supplies	Art	Staples Hanging Manila Folders, Bright Primary Assortment, 25/Box	55.5	3	0.00	15.54
5	CA-2011-100375	Apple iPhone 5S	1	700	2019-12-05 00:00:00	17050 McCormick Drive	Emily Burns	Female	34	United States	Anaheim	California	92806	West	TEC-PH-10002275	Technology	Phones	Apple iPhone 5S	700	1	0.00	210
6	CA-2011-100405	Staples	3	4.2	2019-12-12 00:00:00	17050 McCormick Drive	Emily Burns	Female	34	United States	Anaheim	California	92806	West	OFF-ST-10000760	Office Supplies	Storage	Stackable Flip Lid Storage Boxes	12.6	3	0.20	3.78

```
df.tail(5)
```

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Customer Name	Gender	Age	Country	City	State	Postal Code	Region	Product ID	Category	Sub Category	Product Name	Sales	Quantity	Discount	Profit
5004	CA-2011-100001	AT&T EL51325	1	84.1	2019-07-18 00:00:00	193 Spring Street	Sean Miller	Male	41	United States	Los Angeles	California	90036	West	OFF-EN-10001509	Office Supplies	Envelopes	Gyration Business Envelope	15.9	1	0.20	4.77
5005	CA-2011-100006	Wilson Jr Suede Chair	2	125	2019-09-07 00:00:00	17050 McCormick Drive	Emily Burns	Female	34	United States	Anaheim	California	92806	West	OFF-AP-10002684	Furniture	Chairs	Hon Deluxe Fabric Upholstered Stacking Chairs, Round Back	250	2	0.00	105
5006	CA-2011-100090	Eldon Base for stacking storage boxes	2	85.9	2019-09-18 00:00:00	17050 McCormick Drive	Emily Burns	Female	34	United States	Anaheim	California	92806	West	OFF-AR-10003651	Office Supplies	Storage	Eldon Fold 'N Roll Cart System	171.8	2	0.20	19.32
5007	CA-2011-100293	Xerox 1881	3	8.1	2019-03-08 00:00:00	17050 McCormick Drive	Emily Burns	Female	34	United States	Anaheim	California	92806	West	OFF-PA-10002365	Office Supplies	Paper	Xerox 1881	24.3	3	0.00	11.34
5008	CA-2011-100328	Staples	3	4.2	2019-12-12 00:00:00	17050 McCormick Drive	Emily Burns	Female	34	United States	Anaheim	California	92806	West	OFF-ST-10000760	Office Supplies	Storage	Stackable Flip Lid Storage Boxes	12.6	3	0.20	3.78

```
df.info()
```

```
<class'>
RangeIndex: 5009 entries, 0 to 5008
Data columns (total 22):
 #   Order ID         5009 non-null   object
 #   Product          5009 non-null   object
 #   Quantity Ordered 5009 non-null   int64
 #   Price Each       5009 non-null   int64
 #   Order Date       5009 non-null   object
 #   Purchase Address 5009 non-null   object
 #   Customer Name    5009 non-null   object
 #   Gender           5009 non-null   object
 #   Age              5009 non-null   int64
 #   Country          5009 non-null   object
 #   City             5009 non-null   object
 #   State            5009 non-null   object
 #   Postal Code      5009 non-null   object
 #   Region           5009 non-null   object
 #   Product ID       5009 non-null   object
 #   Category         5009 non-null   object
 #   Sub Category     5009 non-null   object
 #   Product Name     5009 non-null   object
 #   Sales            5009 non-null   float
 #   Quantity         5009 non-null   int64
 #   Discount         5009 non-null   float
 #   Profit           5009 non-null   float
dtype: int64
memory usage: 1.0 MB
```

```
df.isnull().sum
```

```
Order ID                 0
Product                  0
Quantity Ordered         0
Price Each               0
Order Date               0
Purchase Address         0
Customer Name            0
Gender                   0
Age                      0
Country                  0
City                     0
State                    0
Postal Code              0
Region                   0
Product ID               0
Category                 0
Sub Category             0
Product Name             0
Sales                    0
Quantity                 0
Discount                 0
Profit                   0
dtype: int64
```

```
df.duplicated()
```

```
0    False
1    False
2    False
3    False
4    False
dtype: bool
```

```
df.duplicated().sum()
```

```
0
```

```
df.drop_duplicates()
```

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Customer Name	Gender	Age	Country	City	State	Postal Code	Region	Product ID	Category	Sub Category	Product Name	Sales	Quantity	Discount	Profit
0	CA-2011-100006	Wilson Jr Chair	7	280	2019-09-07 00:00:00	17050 McCormick Drive	Emily Burns	Female	34	United States	Anaheim	California	92806	West	OFF-AP-10002684	Furniture	Chairs	Hon Deluxe Fabric Upholstered Stacking Chairs, Round Back	1960	7	0.00	823.2
1	CA-2011-100090	Staples	3	5.1	2019-09-18 00:00:00	17050 McCormick Drive	Emily Burns	Female	34	United States	Anaheim	California	92806	West	OFF-ST-10000078	Office Supplies	Storage	Fellowes Super Stacking Storage Drawers	15.3	3	0.20	-3.06
2	CA-2011-100293	NoMax Desktop Shredder	7	219.9	2019-03-08 00:00:00	17050 McCormick Drive	Emily Burns	Female	34	United States	Anaheim	California	92806	West	OFF-BI-10003627	Office Supplies	Binders	Acco Pressboard Covers with Storage Hooks, 14 7/8" x 11", Light Blue	1539.3	7	0.00	430.2
3	CA-2011-100328	Staples	3	4.2	2019-12-12 00:00:00	17050 McCormick Drive	Emily Burns	Female	34	United States	Anaheim	California	92806	West	OFF-ST-10000760	Office Supplies	Storage	Stackable Flip Lid Storage Boxes	12.6	3	0.20	3.78
4	CA-2011-100362	Sanford Calculator	3	18.5	2019-12-22 00:00:00	17050 McCormick Drive	Emily Burns	Female	34	United States	Anaheim	California	92806	West	OFF-AR-10002833	Office Supplies	Art	Staples Hanging Manila Folders, Bright Primary Assortment, 25/Box	55.5	3	0.00	15.54
5004	CA-2011-100001	AT&T EL51325	1	84.1	2019-07-18 00:00:00	193 Spring Street	Sean Miller	Male	41	United States	Los Angeles	California	90036	West	OFF-EN-10001509	Office Supplies	Envelopes	Gyration Business Envelope	15.9	1	0.20	4.77
5005	CA-2011-100006	Wilson Jr Suede Chair	2	125	2019-09-07 00:00:00	17050 McCormick Drive	Emily Burns	Female	34	United States	Anaheim	California	92806	West	OFF-AP-10002684	Furniture	Chairs	Hon Deluxe Fabric Upholstered Stacking Chairs, Round Back	250	2	0.00	105
5006	CA-2011-100090	Eldon Base for stacking storage boxes	2	85.9	2019-09-18 00:00:00	17050 McCormick Drive	Emily Burns	Female	34	United States	Anaheim	California	92806	West	OFF-AR-10003651	Office Supplies	Storage	Eldon Fold 'N Roll Cart System	171.8	2	0.20	19.32
5007	CA-2011-100293	Xerox 1881	3	8.1	2019-03-08 00:00:00	17050 McCormick Drive	Emily Burns	Female	34	United States	Anaheim	California	92806	West	OFF-PA-10002365	Office Supplies	Paper	Xerox 1881	24.3	3	0.00	11.34
5008	CA-2011-100328	Staples	3	4.2	2019-12-12 00:00:00	17050 McCormick Drive	Emily Burns	Female	34	United States	Anaheim	California	92806	West	OFF-ST-10000760	Office Supplies	Storage	Stackable Flip Lid Storage Boxes	12.6	3	0.20	3.78

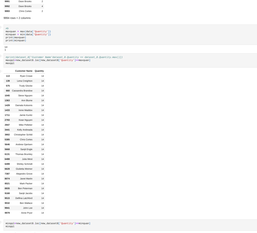
5009 rows × 22 columns

Task 4: Find out the monthly revenue and analyze the findings

- Create a column with Year and month from Order Date
- Create a column with Month from Order Date
- Create a separate dataset from the data which will have two columns, one which is created in step 1 of this task and second the monthly sales / revenue
- Plot the dataset
- Describe your analysis for the monthly revenue observations.

```
df['Year'] = df['Order Date'].dt.year
df['Month'] = df['Order Date'].dt.month
df['Month_Year'] = df['Year'].astype(str) + '-' + df['Month'].astype(str)
monthly_sales = df.groupby('Month_Year')['Sales'].sum()
monthly_sales = monthly_sales[monthly_sales > 0]
plt.figure(figure=(12,6))
plt.plot(monthly_sales.index, monthly_sales.values, marker='o')
plt.xlabel('Month_Year')
plt.ylabel('Sales')
plt.title('Monthly Sales Trend')
plt.xticks(rotation=45)
plt.grid(True)
plt.tight_layout()
plt.show
```

```
<ipyt>:1: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will always be treated as labels (consistent with DataFrame behavior). To access a value by position, use `ser[ser.iloc[pos]]`
```




```
df['Month_Year'].sort_values(ascending=True).unique()
```

```
array(['2019-1', '2019-2', '2019-3', '2019-4', '2019-5', '2019-6',
       '2019-7', '2019-8', '2019-9', '2019-10', '2019-11', '2019-12'],
      dtype=object)
```

```
monthly_sales.sort_values(ascending=True)
```

```
Month_Year
2019-1    1822256.73000000
2019-2    2744967.96000000
2019-3    3519224.16000000
2019-4    3390670.24000000
2019-5    3152606.75000000
2019-6    2577802.26000000
2019-7    2647775.76000000
2019-8    2244467.88000000
2019-9    2097960.29999999
2019-10   3736726.88000000
2019-11   3199603.20000000
2019-12   4613443.34000000
Name: Sales, dtype: float64
```

```
df['Month_Year'].sort_values(ascending=True).unique()
```

```
array(['2019-1', '2019-2', '2019-3', '2019-4', '2019-5', '2019-6',
       '2019-7', '2019-8', '2019-9', '2019-10', '2019-11', '2019-12'],
      dtype=object)
```

Task 5: Find the yearly revenue and analyze the findings

- Create a column with Year from Order Date
- Create a separate dataset from the data which will have two columns, one which is created in step 1 of this task and second the yearly sales / revenue
- Plot the dataset
- Describe your analysis for the yearly revenue observations.

```
df['Year'] = df['Order Date'].dt.year
yearly_sales = df.groupby('Year')['Sales'].sum()
yearly_sales = yearly_sales[yearly_sales > 0]
plt.figure(figure=(12,6))
plt.plot(yearly_sales.index, yearly_sales.values, marker='o')
plt.xlabel('Year')
plt.ylabel('Sales')
plt.title('Yearly Sales Trend')
plt.xticks(rotation=45)
plt.grid(True)
plt.tight_layout()
plt.show
```

```
<ipyt>:1: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will always be treated as labels (consistent with DataFrame behavior). To access a value by position, use `ser[ser.iloc[pos]]`
```

```
df['Year'].sort_values(ascending=True).unique()
```

```
array([2019, 2020, 2021])
```

```
yearly_sales.sort_values(ascending=True)
```

```
Year
2019    10903827.82000000
2020    16964638.28000000
2021    5040000.00000000
Name: Sales, dtype: float64
```

```
df['Year'].sort_values(ascending=True).unique()
```

```
array([2019, 2020, 2021])
```

```
df.describe(include='object')
```

	Order ID	Product	Order Date	Purchase Address	Customer Name	Gender	Country	City	State	Postal Code	Region	Product ID	Category	Sub Category	Product Name
count	5009	5009	5009	5009	5009	5009	5009	5009	5009	5009	5009	5009	5009	5009	5009
unique	4962	19	4962	5000	5000	2	1	12	3	9	4	19	3	17	19
top	CA-2011-100006	Wilson Jr Chair	2019-09-07 00:00:00	17050 McCormick Drive	Emily Burns	Female	United States	Anaheim	California	92806	West	OFF-AP-10002684	Furniture	Chairs	Hon Deluxe Fabric Upholstered Stacking Chairs, Round Back
freq	2	1960	2	2	2	2504	5009	1960	1960	1960	1960	1960	1960	1960	1960

Task 6: Find out the customers who bought most and least from us in terms of quantity

- Create a dataset containing name and quantities bought
- Find out the customer name and quantity who bought maximum in quantity

```
df.groupby('Customer Name')['Quantity'].sum.sort_values(ascending=False)
df.groupby('Customer Name')['Quantity'].sum.sort_values(ascending=True)
```

```
Customer Name
Emily Burns        7
Sean Miller        1
Name: Quantity, dtype: int64
```

```
df.groupby('Customer Name')['Quantity'].sum()
```

```
Customer Name
Emily Burns        7
Sean Miller        1
Name: Quantity, dtype: int64
```

```
df.groupby('Customer Name')['Quantity'].sum().sort_values(ascending=False)
df.groupby('Customer Name')['Quantity'].sum().sort_values(ascending=False).head(1)
```

```
Customer Name
Emily Burns        7
Sean Miller        1
Name: Quantity, dtype: int64
```

```
df.groupby('Customer Name')['Quantity'].sum().sort_values(ascending=False).tail(1)
```

```
Customer Name
Sean Miller        1
Name: Quantity, dtype: int64
```

Task 7: Finding out the most and least sold product in

- Create a new dataset containing the product and the total quantities sold
- Find out the most sold product from the created dataset for that id

```
df.groupby('Product')['Quantity'].sum.sort_values(ascending=False)
df.groupby('Product')['Quantity'].sum.sort_values(ascending=False).head(1)
df.groupby('Product')['Quantity'].sum.sort_values(ascending=False).tail(1)
```

```
Product
Wilson Jr Chair    7
Name: Quantity, dtype: int64
```

```
df.groupby('Product')['Quantity'].sum()
```

```
Product
Apple iPhone 5S                                    1
Eldon Base for stacking storage boxes              2
Eldon Fold 'N Roll Cart System                     2
Hon Deluxe Fabric Upholstered Stacking Chairs,     2
NoMax Desktop Shredder                             7
Sanford Calculator                                 3
Staples                                            3
Wilson Jr Chair                                    7
Wilson Jr Suede Chair                              2
Xerox 1881                                         3
Name: Quantity, dtype: int64
```

```
df.groupby('Product')['Quantity'].sum().sort_values(ascending=False)
df.groupby('Product')['Quantity'].sum().sort_values(ascending=False).head(1)
```

```
Product
Wilson Jr Chair    7
Name: Quantity, dtype: int64
```

```
df.groupby('Product')['Quantity'].sum().sort_values(ascending=False).tail(1)
```

```
Product
Apple iPhone 5S                                    1
Name: Quantity, dtype: int64
```

Task 8: Finding out the customers who bought most and least from us in terms of quantity

- Create a dataset containing name and quantities bought
- Find out the customer name and quantity who bought maximum in quantity

```
df.groupby('Customer Name')['Quantity'].sum.sort_values(ascending=False)
df.groupby('Customer Name')['Quantity'].sum.sort_values(ascending=False).head(1)
df.groupby('Customer Name')['Quantity'].sum.sort_values(ascending=False).tail(1)
```

```
Customer Name
Emily Burns        7
Sean Miller        1
Name: Quantity, dtype: int64
```

```
df.groupby('Customer Name')['Quantity'].sum()
```

```
Customer Name
Emily Burns        7
Sean Miller        1
Name: Quantity, dtype: int64
```

```
df.groupby('Customer Name')['Quantity'].sum().sort_values(ascending=False)
df.groupby('Customer Name')['Quantity'].sum().sort_values(ascending=False).head(1)
df.groupby('Customer Name')['Quantity'].sum().sort_values(ascending=False).tail(1)
```

```
Customer Name
Emily Burns        7
Sean Miller        1
Name: Quantity, dtype: int64
```

```
df.groupby('Customer Name')['Quantity'].sum().sort_values(ascending=False).head(1)
```

```
Customer Name
Emily Burns        7
Name: Quantity, dtype: int64
```

```
df.groupby('Customer Name')['Quantity'].sum().sort_values(ascending=False).tail(1)
```

```
Customer Name
Sean Miller        1
Name: Quantity, dtype: int64
```

Task 10 : Finding out the majority and minority customer cities on basis of

- Number of customers
- Number of sales
- Number of quantity sold

```
df.groupby('Customer Name')['City']
df.groupby('Customer Name')['City'].count
df.groupby('Customer Name')['City'].count().sort_values(ascending=False)
df.groupby('Customer Name')['City'].count().sort_values(ascending=False).head(1)
df.groupby('Customer Name')['City'].count().sort_values(ascending=False).tail(1)
```

```
City, dtype: int64
```

```
df.groupby('Customer Name')['City'].count()
```

```
Customer Name
Emily Burns        2
Sean Miller        2
Name: City, dtype: int64
```

```
df.groupby('Customer Name')['City'].count().sort_values(ascending=False)
df.groupby('Customer Name')['City'].count().sort_values(ascending=False).head(1)
df.groupby('Customer Name')['City'].count().sort_values(ascending=False).tail(1)
```

```
Customer Name
Emily Burns        2
Sean Miller        2
Name: City, dtype: int64
```

```
df.groupby('Customer Name')['City'].count().sort_values(ascending=False).head(1)
```

```
Customer Name
Emily Burns        2
Name: City, dtype: int64
```

```
df.groupby('Customer Name')['City'].count().sort_values(ascending=False).tail(1)
```

```
Customer Name
Sean Miller        2
Name: City, dtype: int64
```

Task 11 : Find out the most and least sold product category from the store

- Value based
- Quantity based

```
df.groupby('Sub Category')
df.groupby('Sub Category')['Sales']
df.groupby('Sub Category')['Sales'].sum
df.groupby('Sub Category')['Sales'].sum().sort_values(ascending=False)
df.groupby('Sub Category')['Sales'].sum().sort_values(ascending=False).head(1)
df.groupby('Sub Category')['Sales'].sum().sort_values(ascending=False).tail(1)
```

```
Sub Category
Binders    1539.3
Name: Sales, dtype: float64
```

```
df.groupby('Sub Category')['Sales'].sum()
```

```
Sub Category
Binders    1539.3
Name: Sales, dtype: float64
```

```
df.groupby('Sub Category')['Sales'].sum().sort_values(ascending=False)
df.groupby('Sub Category')['Sales'].sum().sort_values(ascending=False).head(1)
df.groupby('Sub Category')['Sales'].sum().sort_values(ascending=False).tail(1)
```

```
Sub Category
Binders    1539.3
Name: Sales, dtype: float64
```

```
df.groupby('Sub Category')['Sales'].sum().sort_values(ascending=False).head(1)
```

```
Sub Category
Binders    1539.3
Name: Sales, dtype: float64
```

```
df.groupby('Sub Category')['Sales'].sum().sort_values(ascending=False).tail(1)
```

```
Sub Category
Binders    1539.3
Name: Sales, dtype: float64
```

Task 12 : Find out the most and least sold product sub category from the store

- Value based
- Quantity based

```
df.groupby('Sub Category')
df.groupby('Sub Category')['Quantity']
df.groupby('Sub Category')['Quantity'].sum
df.groupby('Sub Category')['Quantity'].sum().sort_values(ascending=False)
df.groupby('Sub Category')['Quantity'].sum().sort_values(ascending=False).head(1)
df.groupby('Sub Category')['Quantity'].sum().sort_values(ascending=False).tail(1)
```

```
Sub Category
Binders    3
Name: Quantity, dtype: int64
```

```
df.groupby('Sub Category')['Quantity'].sum()
```

```
Sub Category
Binders    3
Name: Quantity, dtype: int64
```

```
df.groupby('Sub Category')['Quantity'].sum().sort_values(ascending=False)
df.groupby('Sub Category')['Quantity'].sum().sort_values(ascending=False).head(1)
df.groupby('Sub Category')['Quantity'].sum().sort_values(ascending=False).tail(1)
```

```
Sub Category
Binders    3
Name: Quantity, dtype: int64
```

```
df.groupby('Sub Category')['Quantity'].sum().sort_values(ascending=False).head(1)
```

```
Sub Category
Binders    3
Name: Quantity, dtype: int64
```

```
df.groupby('Sub Category')['Quantity'].sum().sort_values(ascending=False).tail(1)
```

```
Sub Category
Binders    3
Name: Quantity, dtype: int64
```