

1) Problem: Analyze a retail company's sales data to understand sales trends, top-selling products, and regional performance.

Tasks:

Load and clean the dataset

Find monthly/yearly sales trends

Top 10 products by sales and profit

Region-wise performance

Export summary as Excel report

**** Solution : ****

```
# Import libraries
```

```
import pandas as pd
import matplotlib as plt
import seaborn as sns
import numpy as np
```

Load dataset

```
# Load the Sales Performance dataset
```

```
!pip install xlrd
```

```
data = pd.read_excel(r"C:\Users\Samruddhi Yadav\Documents\Resume
Project\Sales Performance Analysis\Sample - Superstore.xls")
data
```

```
WARNING: Ignoring invalid distribution ~orch (C:\Users\Samruddhi
Yadav\AppData\Roaming\Python\Python312\site-packages)
```

```
WARNING: Ignoring invalid distribution ~orch (C:\Users\Samruddhi
Yadav\AppData\Roaming\Python\Python312\site-packages)
```

```
WARNING: Ignoring invalid distribution ~orch (C:\Users\Samruddhi
Yadav\AppData\Roaming\Python\Python312\site-packages)
```

```
WARNING: Ignoring invalid distribution ~orch (C:\Users\Samruddhi
Yadav\AppData\Roaming\Python\Python312\site-packages)
```

```
WARNING: Ignoring invalid distribution ~orch (C:\Users\Samruddhi
Yadav\AppData\Roaming\Python\Python312\site-packages)
```

```
WARNING: Ignoring invalid distribution ~orch (C:\Users\Samruddhi
Yadav\AppData\Roaming\Python\Python312\site-packages)
```

```
Defaulting to user installation because normal site-packages is not
writeable
```

```
Requirement already satisfied: xlrd in c:\users\samruddhi yadav\
appdata\roaming\python\python312\site-packages (2.0.1)
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	\
0	1	CA-2020-152156	2020-11-08	2020-11-11	Second Class	
1	2	CA-2020-152156	2020-11-08	2020-11-11	Second Class	
2	3	CA-2020-138688	2020-06-12	2020-06-16	Second Class	
3	4	US-2019-108966	2019-10-11	2019-10-18	Standard Class	
4	5	US-2019-108966	2019-10-11	2019-10-18	Standard Class	
...
9989	9990	CA-2018-110422	2018-01-21	2018-01-23	Second Class	
9990	9991	CA-2021-121258	2021-02-26	2021-03-03	Standard Class	
9991	9992	CA-2021-121258	2021-02-26	2021-03-03	Standard Class	
9992	9993	CA-2021-121258	2021-02-26	2021-03-03	Standard Class	
9993	9994	CA-2021-119914	2021-05-04	2021-05-09	Second Class	
	Customer ID	Customer Name	Segment	Country/Region		
City \						
0	CG-12520	Claire Gute	Consumer	United States		
Henderson						
1	CG-12520	Claire Gute	Consumer	United States		
Henderson						
2	DV-13045	Darrin Van Huff	Corporate	United States	Los	
Angeles						
3	S0-20335	Sean O'Donnell	Consumer	United States	Fort	
Lauderdale						
4	S0-20335	Sean O'Donnell	Consumer	United States	Fort	
Lauderdale						
...	
...						
9989	TB-21400	Tom Boeckenhauer	Consumer	United States		
Miami						
9990	DB-13060	Dave Brooks	Consumer	United States		
Costa Mesa						
9991	DB-13060	Dave Brooks	Consumer	United States		
Costa Mesa						
9992	DB-13060	Dave Brooks	Consumer	United States		
Costa Mesa						
9993	CC-12220	Chris Cortes	Consumer	United States		
Westminster						
	Postal Code	Region	Product ID	Category	Sub-	
Category \						
0	...	42420.0	South	FUR-B0-10001798	Furniture	
Bookcases						
1	...	42420.0	South	FUR-CH-10000454	Furniture	
Chairs						
2	...	90036.0	West	OFF-LA-10000240	Office Supplies	
Labels						
3	...	33311.0	South	FUR-TA-10000577	Furniture	
Tables						
4	...	33311.0	South	OFF-ST-10000760	Office Supplies	
Storage						

...
9989	...	33180.0	South	FUR-FU-10001889	Furniture
Furnishings					
9990	...	92627.0	West	FUR-FU-10000747	Furniture
Furnishings					
9991	...	92627.0	West	TEC-PH-10003645	Technology
Phones					
9992	...	92627.0	West	OFF-PA-10004041	Office Supplies
Paper					
9993	...	92683.0	West	OFF-AP-10002684	Office Supplies
Appliances					

Quantity \	Product Name	Sales
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		
...
...		
9989	Ultra Door Pull Handle	25.2480
3		
9990	Tenex B1-RE Series Chair Mats for Low Pile Car...	91.9600
2		
9991	Aastra 57i VoIP phone	258.5760
2		
9992	It's Hot Message Books with Stickers, 2 3/4" x 5"	29.6000
4		
9993	Acco 7-Outlet Masterpiece Power Center, Wihtou...	243.1600
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
...
9989	0.20	4.1028
9990	0.00	15.6332
9991	0.20	19.3932
9992	0.00	13.3200
9993	0.00	72.9480

```
[9994 rows x 21 columns]
```

```
# Dataset info.
```

```
data.info()          # No.of rows,count,dtype
```

```
<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/Region	9994	non-null	object
9	City	9994	non-null	object
10	State	9994	non-null	object
11	Postal Code	9983	non-null	float64
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(4), int64(2), object(13)
```

```
memory usage: 1.6+ MB
```

```
data.describe()      # Describes whole dataset
```

	Row ID	Order Date \
count	9994.000000	9994
mean	4997.500000	2020-04-30 00:07:03.614168576
min	1.000000	2018-01-03 00:00:00
25%	2499.250000	2019-05-23 00:00:00
50%	4997.500000	2020-06-26 00:00:00
75%	7495.750000	2021-05-14 00:00:00
max	9994.000000	2021-12-30 00:00:00
std	2885.163629	NaN

	Ship Date	Postal Code	Sales
Quantity \			

count	9994	9983.000000	9994.000000
9994.000000			
mean	2020-05-03 23:06:58.571142656	55245.233297	229.858001
3.789574			
min	2018-01-07 00:00:00	1040.000000	0.444000
1.000000			
25%	2019-05-27 00:00:00	23223.000000	17.280000
2.000000			
50%	2020-06-29 00:00:00	57103.000000	54.490000
3.000000			
75%	2021-05-18 00:00:00	90008.000000	209.940000
5.000000			
max	2022-01-05 00:00:00	99301.000000	22638.480000
14.000000			
std	NaN	32038.715955	623.245101
2.225110			

	Discount	Profit
count	9994.000000	9994.000000
mean	0.156203	28.656896
min	0.000000	-6599.978000
25%	0.000000	1.728750
50%	0.200000	8.666500
75%	0.200000	29.364000
max	0.800000	8399.976000
std	0.206452	234.260108

Data Cleaning

```
# Check missing data
```

```
data.isnull().sum()
```

Row ID	0
Order ID	0
Order Date	0
Ship Date	0
Ship Mode	0
Customer ID	0
Customer Name	0
Segment	0
Country/Region	0
City	0
State	0
Postal Code	11
Region	0
Product ID	0
Category	0
Sub-Category	0
Product Name	0

```

Sales          0
Quantity       0
Discount       0
Profit         0
dtype: int64

# Handling missing value

data = data.dropna(subset=["Postal Code"])

data.isnull().sum()

Row ID          0
Order ID        0
Order Date      0
Ship Date       0
Ship Mode       0
Customer ID     0
Customer Name   0
Segment        0
Country/Region  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

# Check duplicates data
data.duplicated().sum()

0

```

**** There are no any duplicate data ****

Check Outlier

```

# Visual Methods (Boxplot/Histogram)

sns.boxplot(x = data["Sales"])
plt.show()

```

```

-----
-----

```

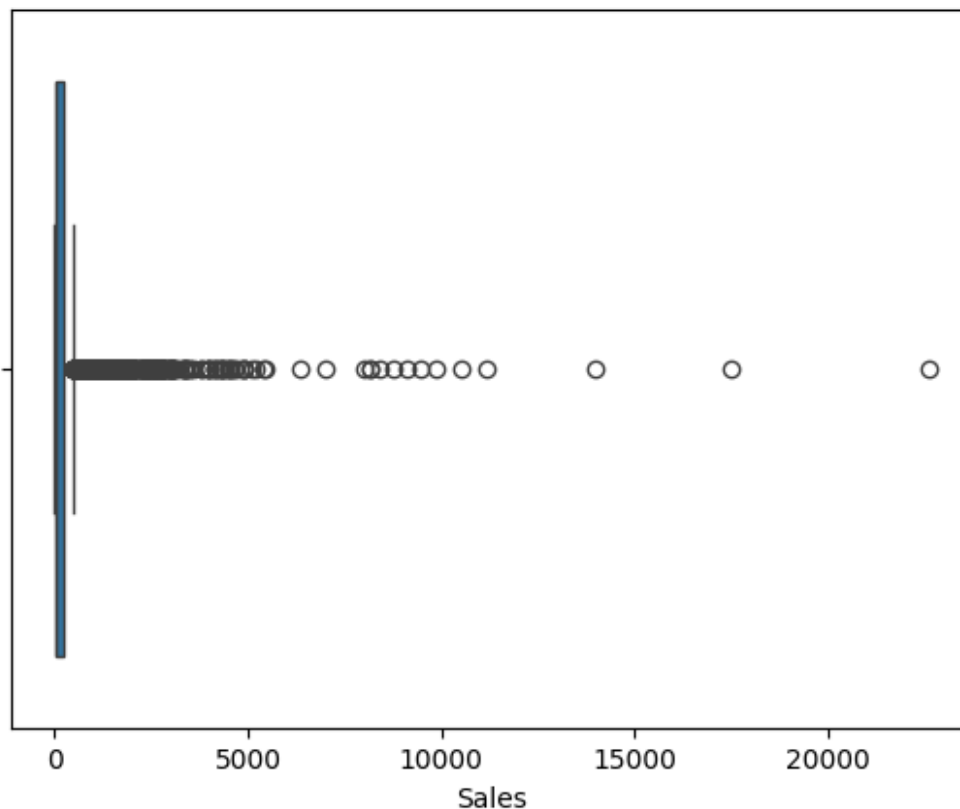
```

AttributeError                                Traceback (most recent call
last)
Cell In[14], line 4
      1 # Visual Methods (Boxplot/Histogram)
      3 sns.boxplot(x = data["Sales"])
----> 4 plt.show()

File ~\AppData\Roaming\Python\Python312\site-packages\matplotlib\_api\
__init__.py:218, in caching_module_getattr.<locals>.__getattr__(name)
    216 if name in props:
    217     return props[name].__get__(instance)
--> 218 raise AttributeError(
    219     f"module {cls.__module__!r} has no attribute {name!r}")

AttributeError: module 'matplotlib' has no attribute 'show'

```



```

# 1. Using the IQR (Interquartile Range) method

for col in ["Sales", "Profit"]:
    Q1 = data[col].quantile(0.25)
    Q3 = data[col].quantile(0.75)
    IQR = Q3 - Q1

```

```
lower_bound = Q1 - 1.5*IQR
upper_bound = Q3 + 1.5*IQR
```

```
Clean_data= data[(data[col] < lower_bound) | (data[col] >
upper_bound)]
print(Clean_data)
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	\
1	2	CA-2020-152156	2020-11-08	2020-11-11	Second Class	
3	4	US-2019-108966	2019-10-11	2019-10-18	Standard Class	
7	8	CA-2018-115812	2018-06-09	2018-06-14	Standard Class	
10	11	CA-2018-115812	2018-06-09	2018-06-14	Standard Class	
13	14	CA-2020-161389	2020-12-05	2020-12-10	Standard Class	
...	
9957	9958	US-2018-143287	2018-11-11	2018-11-17	Standard Class	
9962	9963	CA-2019-168088	2019-03-19	2019-03-22	First Class	
9968	9969	CA-2021-153871	2021-12-11	2021-12-17	Standard Class	
9979	9980	US-2020-103674	2020-12-06	2020-12-10	Standard Class	
9993	9994	CA-2021-119914	2021-05-04	2021-05-09	Second Class	

	Customer ID	Customer Name	Segment	Country/Region	\
1	CG-12520	Claire Gute	Consumer	United States	
3	SO-20335	Sean O'Donnell	Consumer	United States	
7	BH-11710	Brosina Hoffman	Consumer	United States	
10	BH-11710	Brosina Hoffman	Consumer	United States	
13	IM-15070	Irene Maddox	Consumer	United States	
...	
9957	KN-16705	Kristina Nunn	Home Office	United States	
9962	CM-12655	Corinna Mitchell	Home Office	United States	
9968	RB-19435	Richard Bierner	Consumer	United States	
9979	AP-10720	Anne Pryor	Home Office	United States	
9993	CC-12220	Chris Cortes	Consumer	United States	

	City	...	Postal Code	Region	Product ID	\
1	Henderson	...	42420.0	South	FUR-CH-10000454	
3	Fort Lauderdale	...	33311.0	South	FUR-TA-10000577	
7	Los Angeles	...	90032.0	West	TEC-PH-10002275	
10	Los Angeles	...	90032.0	West	FUR-TA-10001539	
13	Seattle	...	98103.0	West	OFF-BI-10003656	
...	
9957	New Rochelle	...	10801.0	East	OFF-PA-10004039	
9962	Houston	...	77041.0	Central	FUR-BO-10004218	
9968	Plainfield	...	7060.0	East	OFF-BI-10004600	
9979	Los Angeles	...	90032.0	West	OFF-BI-10002026	
9993	Westminster	...	92683.0	West	OFF-AP-10002684	

	Category	Sub-Category	\
1	Furniture	Chairs	
3	Furniture	Tables	
7	Technology	Phones	

10	Furniture	Tables
13	Office Supplies	Binders
...
9957	Office Supplies	Paper
9962	Furniture	Bookcases
9968	Office Supplies	Binders
9979	Office Supplies	Binders
9993	Office Supplies	Appliances

Quantity \	Product Name	Sales
1	Hon Deluxe Fabric Upholstered Stacking Chairs,...	731.9400
3		
3	Bretford CR4500 Series Slim Rectangular Table	957.5775
5		
7	Mitel 5320 IP Phone VoIP phone	907.1520
6		
10	Chromcraft Rectangular Conference Tables	1706.1840
9		
13	Fellowes PB200 Plastic Comb Binding Machine	407.9760
3		
...
...		
9957	Xerox 1882	223.9200
4		
9962	Bush Heritage Pine Collection 5-Shelf Bookcase...	383.4656
4		
9968	Ibico Ibimaster 300 Manual Binding System	735.9800
2		
9979	Ibico Recycled Linen-Style Covers	437.4720
14		
9993	Acco 7-Outlet Masterpiece Power Center, Wihtou...	243.1600
2		

	Discount	Profit
1	0.00	219.5820
3	0.45	-383.0310
7	0.20	90.7152
10	0.20	85.3092
13	0.20	132.5922
...
9957	0.00	109.7208
9962	0.32	-67.6704
9968	0.00	331.1910
9979	0.20	153.1152
9993	0.00	72.9480

[1877 rows x 21 columns]

```
print(f"{col} has {Clean_data.shape[0]} Clean_data")
```

Profit has 1877 Clean_data

```
# 2. Using Z-score (from scipy)
```

```
from scipy.stats import zscore
```

```
data['z_score'] = zscore(data['Sales'])
```

```
outliers = data[(data['z_score'] > 3) | (data['z_score'] < -3)]
```

```
print(outliers)
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	\
27	28	US-2019-150630	2019-09-17	2019-09-21	Standard Class	
165	166	CA-2018-139892	2018-09-08	2018-09-12	Standard Class	
251	252	CA-2020-145625	2020-09-11	2020-09-17	Standard Class	
262	263	US-2018-106992	2018-09-19	2018-09-21	Second Class	
263	264	US-2018-106992	2018-09-19	2018-09-21	Second Class	
...	
9660	9661	CA-2020-160717	2020-06-06	2020-06-11	Standard Class	
9774	9775	CA-2018-169019	2018-07-26	2018-07-30	Standard Class	
9857	9858	CA-2019-164301	2019-03-26	2019-03-30	Standard Class	
9929	9930	CA-2020-129630	2020-09-04	2020-09-04	Same Day	
9948	9949	CA-2021-121559	2021-06-01	2021-06-03	Second Class	

	Customer ID	Customer Name	Segment	Country/Region	
City \					
27	TB-21520	Tracy Blumstein	Consumer	United States	
Philadelphia					
165	BM-11140	Becky Martin	Consumer	United States	San
Antonio					
251	KC-16540	Kelly Collister	Consumer	United States	San
Diego					
262	SB-20290	Sean Braxton	Corporate	United States	
Houston					
263	SB-20290	Sean Braxton	Corporate	United States	
Houston					
...	
...					
9660	ME-17320	Maria Etezadi	Home Office	United States	Santa
Barbara					
9774	LF-17185	Luke Foster	Consumer	United States	San
Antonio					
9857	EB-13840	Ellis Ballard	Corporate	United States	
Seattle					
9929	IM-15055	Ionia McGrath	Consumer	United States	San
Francisco					
9948	HW-14935	Helen Wasserman	Corporate	United States	
Indianapolis					

...	Region	Product ID	Category	Sub-Category	\
-----	--------	------------	----------	--------------	---

27	...	East	FUR-B0-10004834	Furniture	Bookcases
165	...	Central	TEC-MA-10000822	Technology	Machines
251	...	West	TEC-AC-10003832	Technology	Accessories
262	...	Central	TEC-MA-10000822	Technology	Machines
263	...	Central	TEC-MA-10003353	Technology	Machines
...
9660	...	West	TEC-PH-10001459	Technology	Phones
9774	...	Central	OFF-BI-10004995	Office Supplies	Binders
9857	...	West	FUR-TA-10001889	Furniture	Tables
9929	...	West	TEC-CO-10003763	Technology	Copiers
9948	...	Central	OFF-AP-10002945	Office Supplies	Appliances

		Product Name	Sales
Quantity \			
27	Riverside Palais Royal Lawyers Bookcase, Royal...		3083.430
7			
165	Lexmark MX611dhe Monochrome Laser Printer		8159.952
8			
251	Logitech P710e Mobile Speakerphone		3347.370
13			
262	Lexmark MX611dhe Monochrome Laser Printer		3059.982
3			
263	Xerox WorkCentre 6505DN Laser Multifunction Pr...		2519.958
7			
...
...			
9660	Samsung Galaxy Mega 6.3		3023.928
9			
9774	GBC DocuBind P400 Electric Binding System		2177.584
8			
9857	Bush Advantage Collection Racetrack Conference...		3393.680
8			
9929	Canon PC1060 Personal Laser Copier		2799.960
5			
9948	Honeywell Enviracaire Portable HEPA Air Clean...		2405.200
8			

	Discount	Profit	z_score
27	0.5	-1665.0522	4.589664
165	0.4	-1359.9920	12.752870
251	0.0	636.0003	5.014088
262	0.4	-509.9970	4.551959
263	0.4	-251.9958	3.683583
...
9660	0.2	226.7946	4.493983
9774	0.8	-3701.8928	3.133035
9857	0.0	610.8624	5.088556
9929	0.2	944.9865	4.133835
9948	0.0	793.7160	3.499049

```
[126 rows x 22 columns]
```

```
C:\Users\Samruddhi Yadav\AppData\Local\Temp\
ipykernel_55264\3714516637.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation:

```
https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#
returning-a-view-versus-a-copy
    data['z_score'] = zscore(data['Sales'])
```

```
-----
-----
```

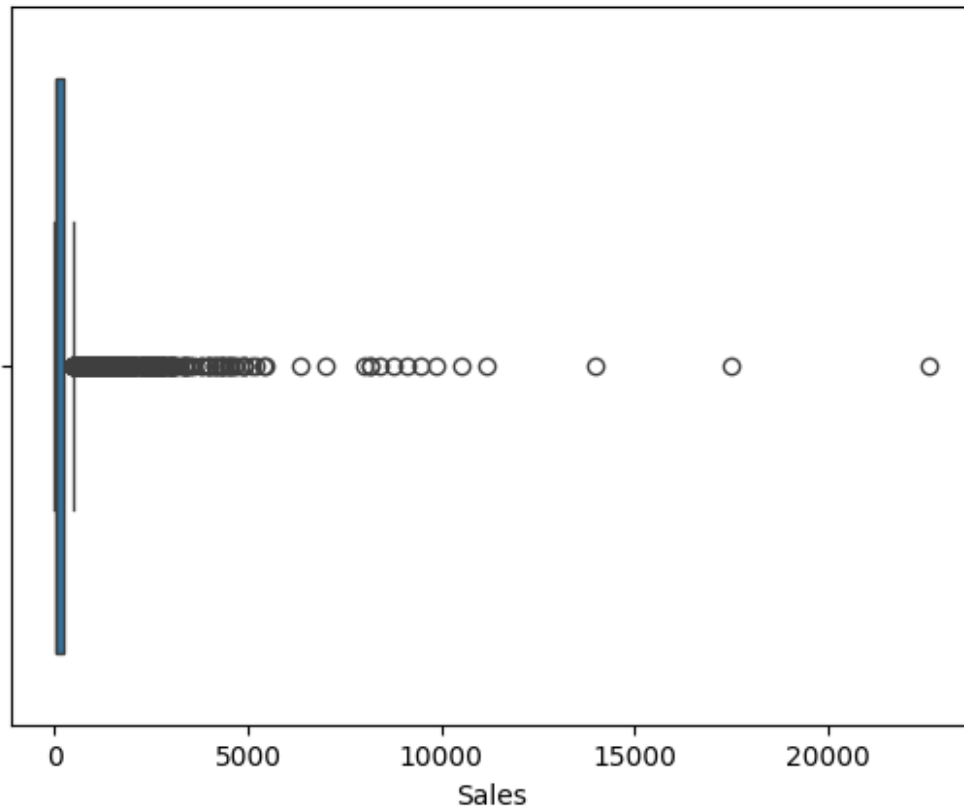
```
AttributeError                                Traceback (most recent call
last)
```

```
Cell In[13], line 4
```

```
    1 # 3. Visual Methods (Boxplot/Histogram)
    3 sns.boxplot(x = data["Sales"])
----> 4 plt.show()
```

```
File ~\AppData\Roaming\Python\Python312\site-packages\matplotlib\_api\
__init__.py:218, in caching_module_getattr.<locals>.__getattr__(name)
    216 if name in props:
    217     return props[name].__get__(instance)
--> 218 raise AttributeError(
    219     f"module {cls.__module__!r} has no attribute {name!r}")
```

```
AttributeError: module 'matplotlib' has no attribute 'show'
```



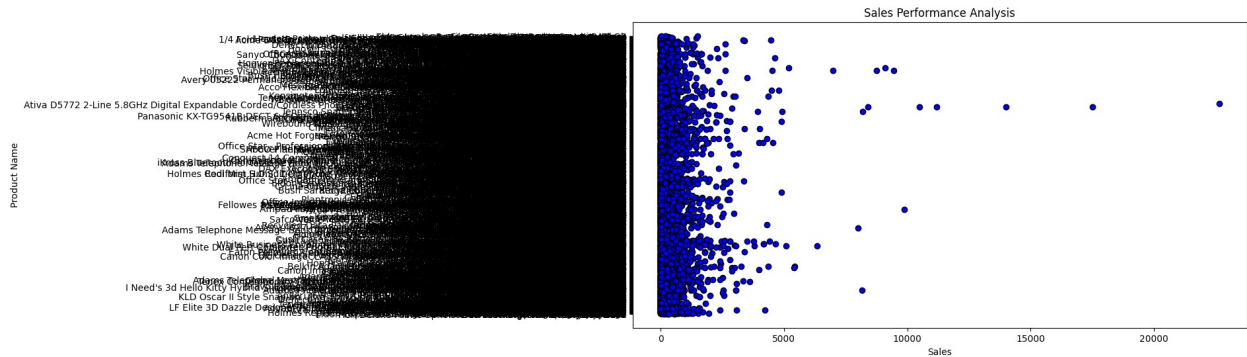
Data Visualization

**** Find monthly/yearly sales trends ****

```
# Scatter plot
import matplotlib.pyplot as plt

plt.figure(figsize=(12, 6))
plt.scatter(x = data["Sales"], y = data["Product Name"], color =
"blue", edgecolor = "black")
plt.title("Sales Performance Analysis")
plt.xlabel("Sales")
plt.ylabel("Product Name")

plt.show()
```



```
# Plot for 10 Products
```

```
import matplotlib.pyplot as plt
```

```
# Group by product and sum sales
```

```
top_products = data.groupby("Product Name")
["Sales"].sum().sort_values(ascending=False).head(10)
```

```
# Plotting
```

```
plt.figure(figsize=(10, 6))
```

```
top_products.plot(kind='barh', color='seagreen', edgecolor='black')
```

```
plt.title("Top 10 Products by Total Sales")
```

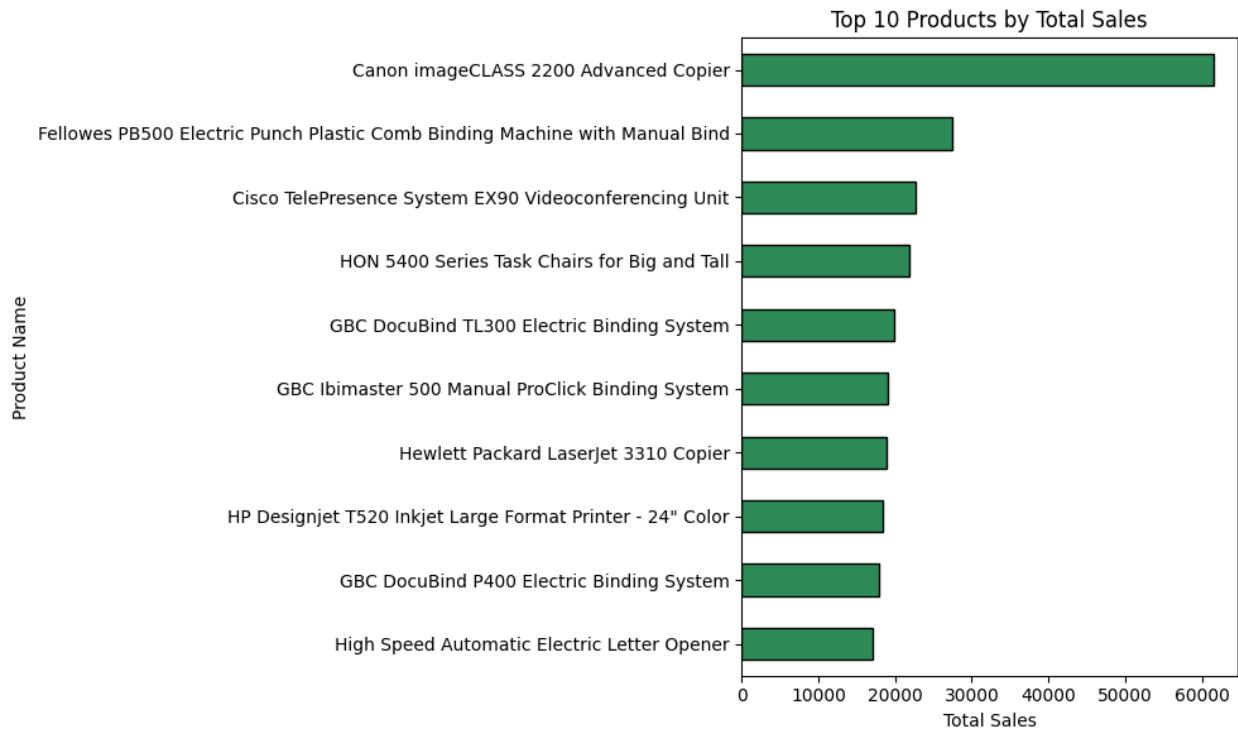
```
plt.xlabel("Total Sales")
```

```
plt.ylabel("Product Name")
```

```
plt.gca().invert_yaxis() # Highest at the top
```

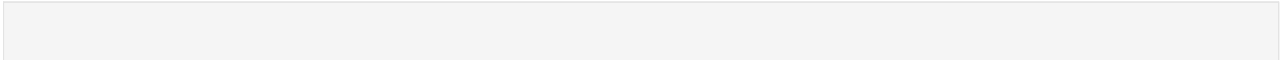
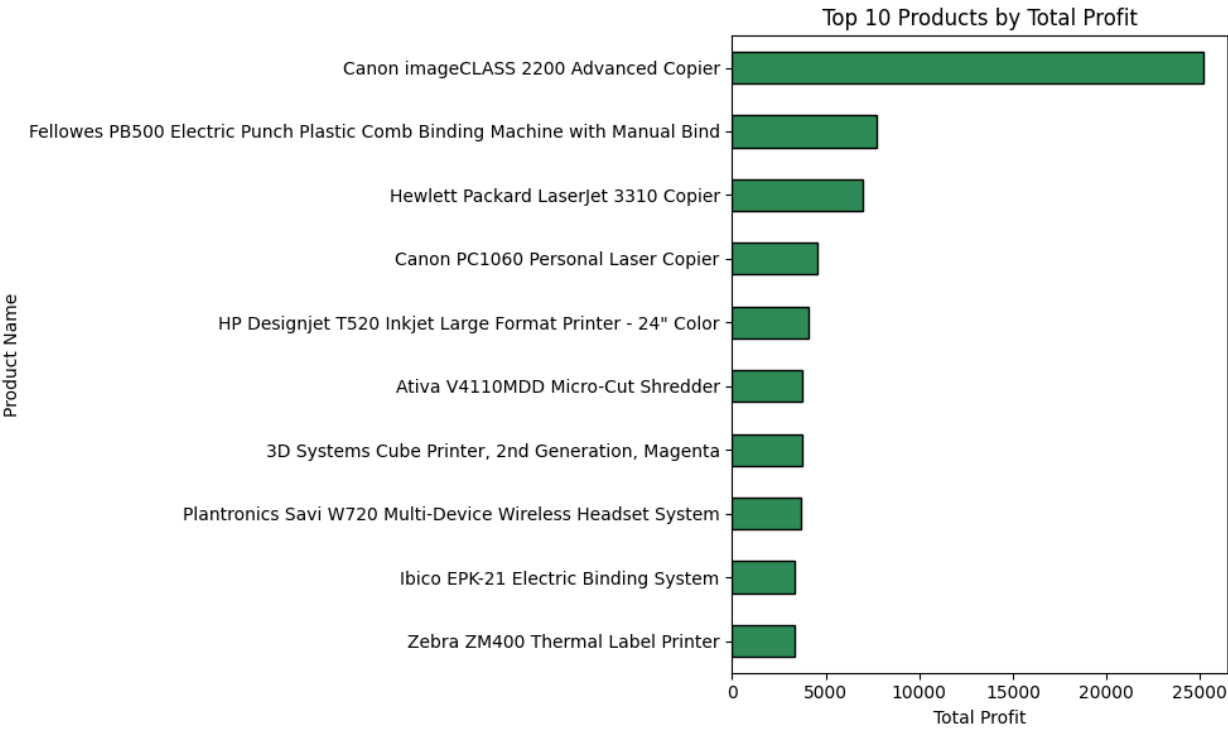
```
plt.tight_layout()
```

```
plt.show()
```



```
top_products1 = data.groupby("Product Name")
["Profit"].sum().sort_values(ascending=False).head(10)

# Plotting
plt.figure(figsize=(10, 6))
top_products1.plot(kind='barh', color='seagreen', edgecolor='black')
plt.title("Top 10 Products by Total Profit")
plt.xlabel("Total Profit")
plt.ylabel("Product Name")
plt.gca().invert_yaxis() # Highest at the top
plt.tight_layout()
plt.show()
```



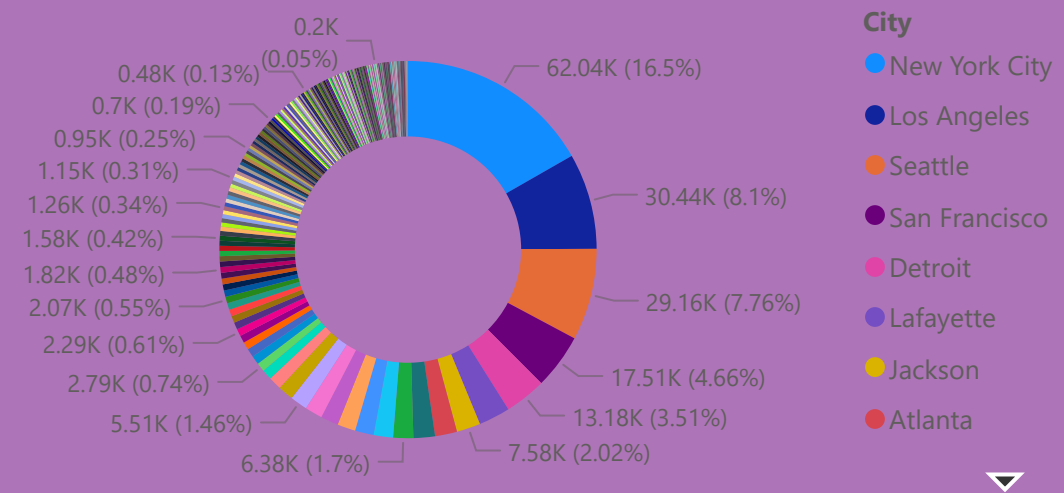
28.66

Average of Profit

9994

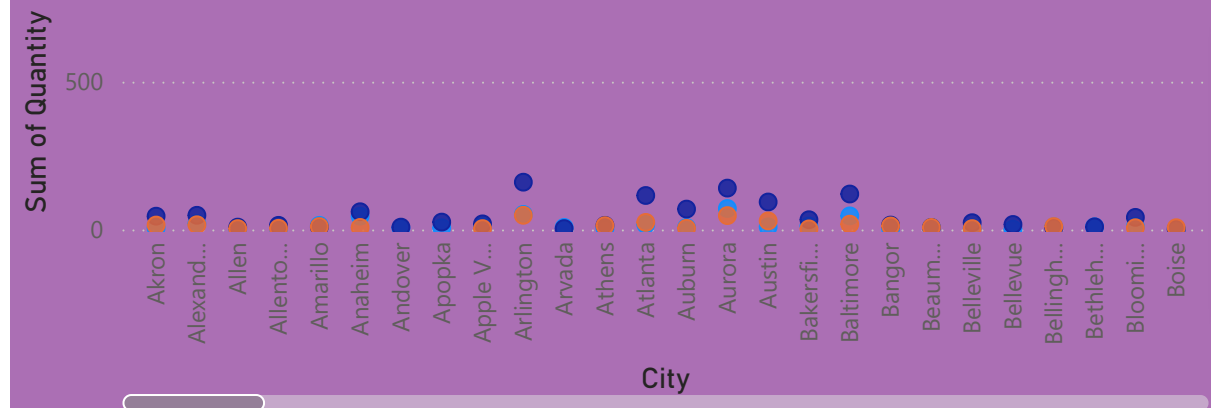
Count of Sales

Sum of Profit by City



Sum of Quantity by Category and City

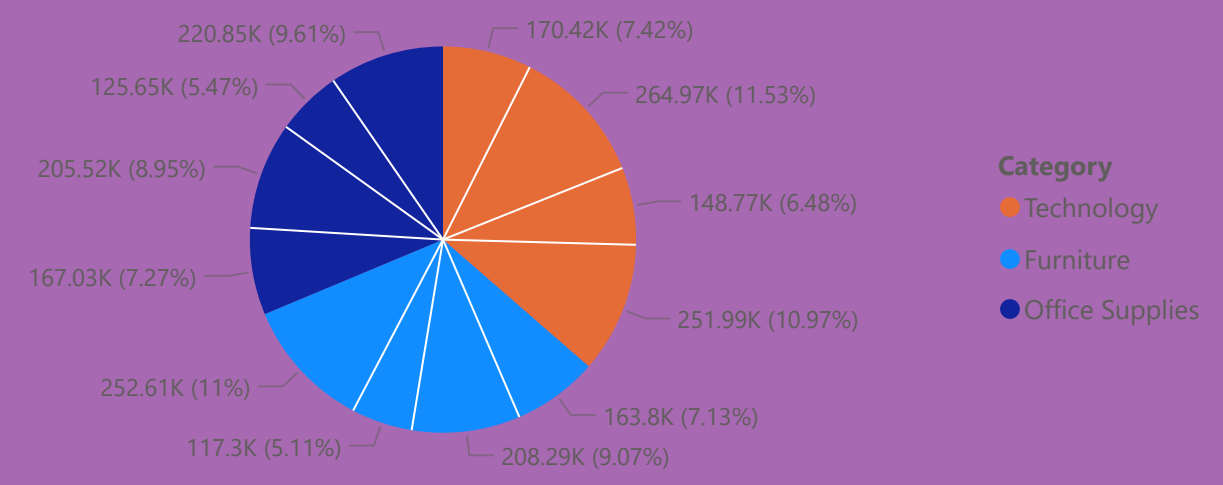
Category Furniture Office Supplies Technology



Sales Performance

Category	Average of Quantity	Sum of Quantity	Min of Quantity	Max of Quantity	Average of Sales	S
Furniture	3.79	8028	1	14	349.83	
Office Supplies	3.80	22906	1	14	119.32	
Technology	3.76	6939	1	14	452.71	
Total	3.79	37873	1	14	229.86	2

Sum of Sales by Category and Region



Sales Performance Analysis

Sum of Sales by City and Category

Category Furniture Office Supplies Technology

