Retail Orders Data Analysis Report

1. Project Overview

This project involves analyzing retail orders data to uncover insights about sales performance, profitability, and customer behavior across regions and product categories. The dataset was sourced from Kaggle and analyzed using Python (Pandas, NumPy, SQLAlchemy) and MySQL for database operations and advanced querying.

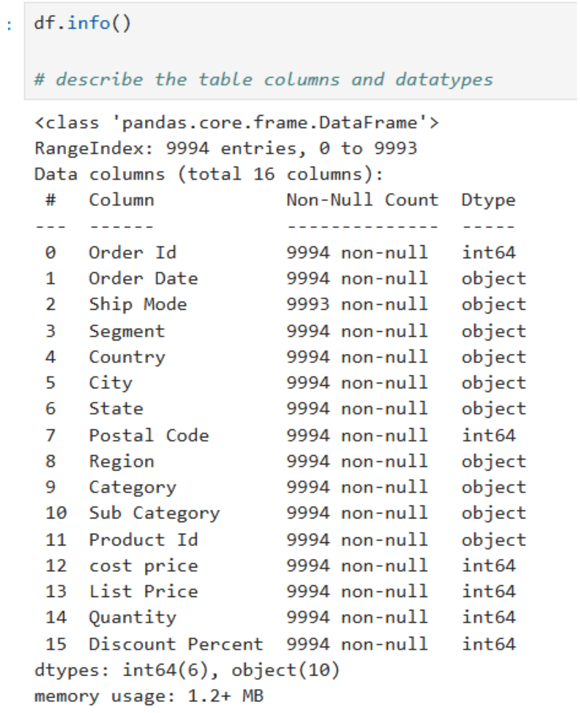
2. Dataset Summary

File Name	orders_data.csv
Rows	9,994
Columns	16
Data Source	Kaggle (Retail Orders Dataset)
Period Covered	2022–2023
Tools Used	Python (Pandas, NumPy, SQLAlchemy), MySQL

Key Columns: Order ID, Order Date, Ship Mode, Segment, Country, City, State, Postal Code, Region, Category, Sub-Category, Product ID, Cost Price, List Price, Quantity, Discount Percent.

3. Data Cleaning and Preparation

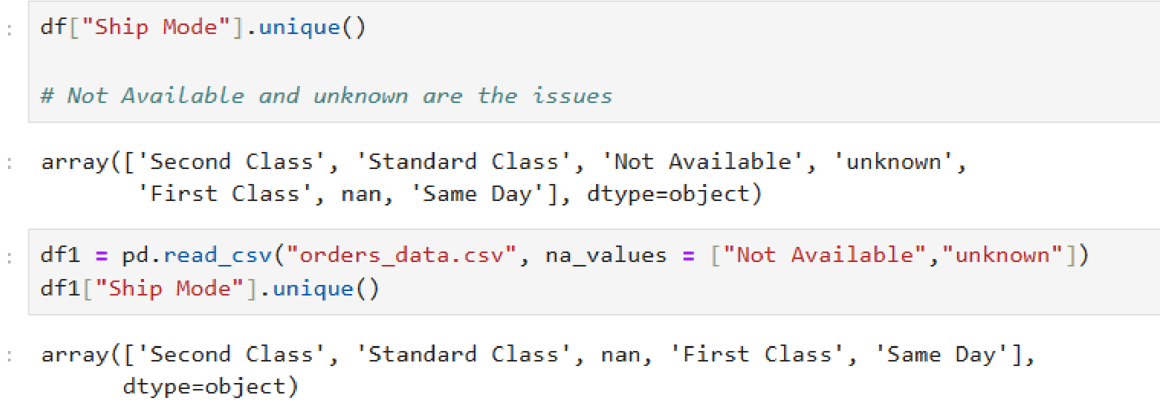
- Imported the dataset using pandas.read_csv() and inspected with df.info().



```
df.info()

# describe the table columns and datatypes
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 16 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Order Id          9994 non-null   int64
 1   Order Date        9994 non-null   object
 2   Ship Mode         9993 non-null   object
 3   Segment           9994 non-null   object
 4   Country           9994 non-null   object
 5   City              9994 non-null   object
 6   State             9994 non-null   object
 7   Postal Code       9994 non-null   int64
 8   Region            9994 non-null   object
 9   Category          9994 non-null   object
 10  Sub Category      9994 non-null   object
 11  Product Id        9994 non-null   object
 12  cost price        9994 non-null   int64
 13  List Price        9994 non-null   int64
 14  Quantity          9994 non-null   int64
 15  Discount Percent  9994 non-null   int64
dtypes: int64(6), object(10)
memory usage: 1.2+ MB
```

- Handled missing values by replacing 'Not Available' and 'unknown' with NaN.



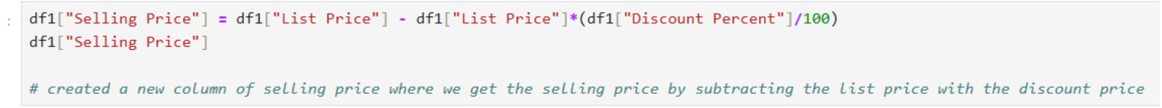
```
df["Ship Mode"].unique()

# Not Available and unknown are the issues
```

```
array(['Second Class', 'Standard Class', 'Not Available', 'unknown',
       'First Class', nan, 'Same Day'], dtype=object)
```

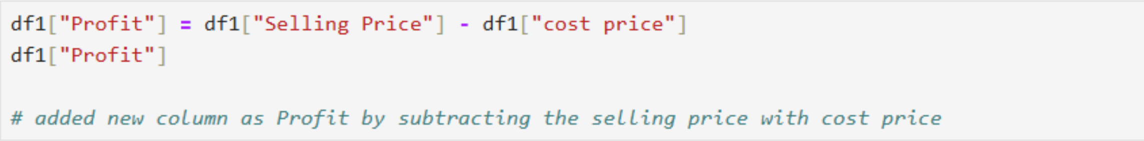
```
df1 = pd.read_csv("orders_data.csv", na_values = ["Not Available","unknown"])
df1["Ship Mode"].unique()
```

```
array(['Second Class', 'Standard Class', nan, 'First Class', 'Same Day'],
      dtype=object)
```

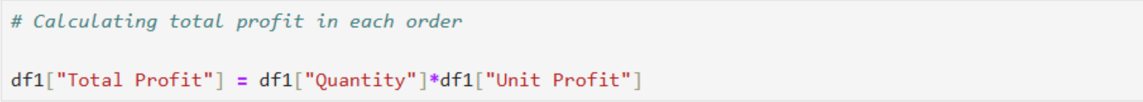
- Created new columns: Unit Selling Price, Unit Profit, Total Sale, and Total Profit.



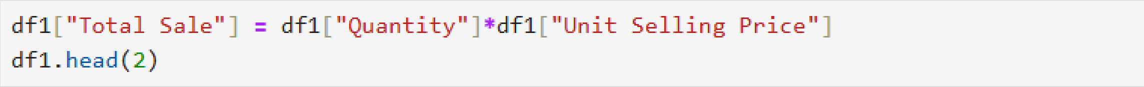
```
df1["Selling Price"] = df1["List Price"] - df1["List Price"]*(df1["Discount Percent"]/100)
df1["Selling Price"]

# created a new column of selling price where we get the selling price by subtracting the list price with the discount price
```



```
df1["Profit"] = df1["Selling Price"] - df1["cost price"]
df1["Profit"]

# added new column as Profit by subtracting the selling price with cost price
```

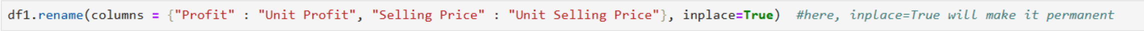


```
# Calculating total profit in each order

df1["Total Profit"] = df1["Quantity"]*df1["Unit Profit"]
```

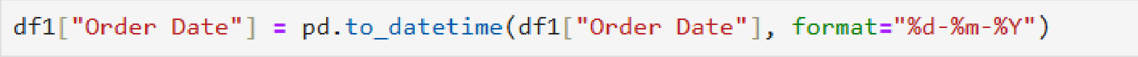


```
df1["Total Sale"] = df1["Quantity"]*df1["Unit Selling Price"]
df1.head(2)
```

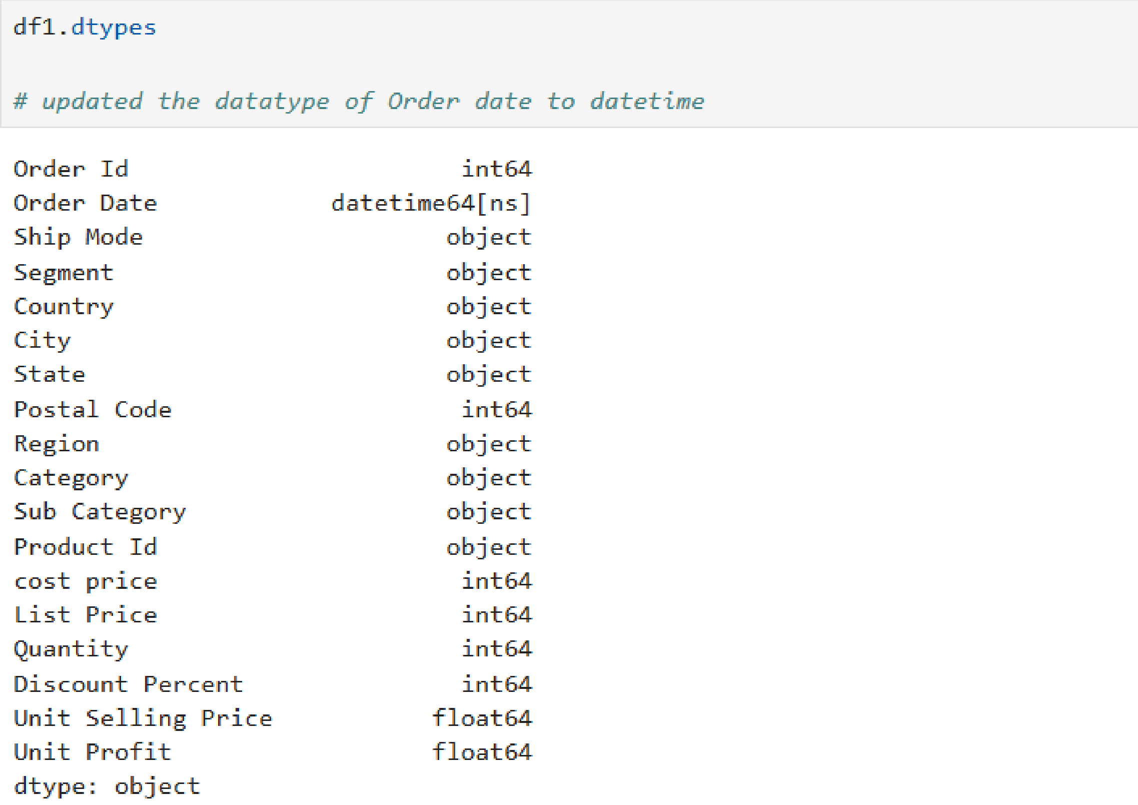
- Renamed columns for clarity and converted Order Date to datetime.



```
df1.rename(columns = {"Profit" : "Unit Profit", "Selling Price" : "Unit Selling Price"}, inplace=True)  #here, inplace=True will make it permanent
```



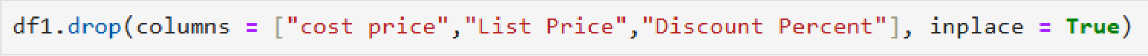
```
df1["Order Date"] = pd.to_datetime(df1["Order Date"], format="%d-%m-%Y")
```



```
df1.dtypes

# updated the datatype of Order date to datetime
```

```
Order Id                       int64
Order Date            datetime64[ns]
Ship Mode                     object
Segment                       object
Country                       object
City                          object
State                         object
Postal Code                    int64
Region                        object
Category                      object
Sub Category                  object
Product Id                    object
cost price                     int64
List Price                     int64
Quantity                       int64
Discount Percent               int64
Unit Selling Price           float64
Unit Profit                  float64
dtype: object
```

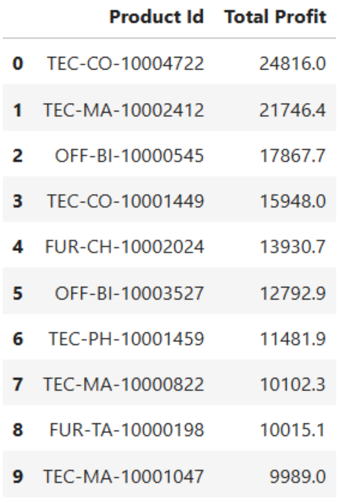
- Dropped unnecessary columns and exported cleaned data to MySQL.



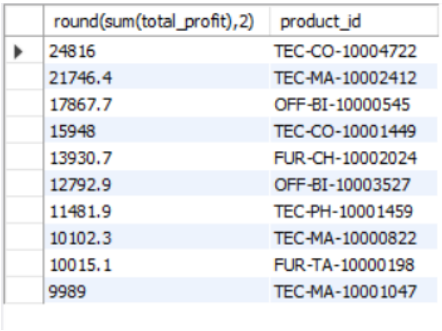
```
df1.drop(columns = ["cost price","List Price","Discount Percent"], inplace = True)
```

4. Exploratory Data Analysis (EDA) using Python and MySQL.

- Top 10 Highest Profit Generating Products: Technology and Office Supplies categories generated the highest profit margins.

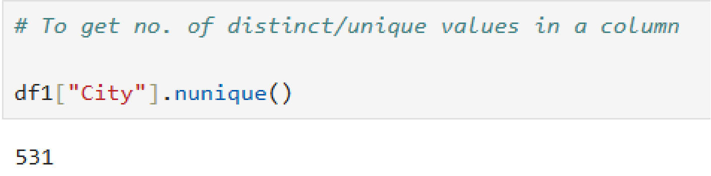


	Product Id	Total Profit
0	TEC-CO-10004722	24816.0
1	TEC-MA-10002412	21746.4
2	OFF-BI-10000545	17867.7
3	TEC-CO-10001449	15948.0
4	FUR-CH-10002024	13930.7
5	OFF-BI-10003527	12792.9
6	TEC-PH-10001459	11481.9
7	TEC-MA-10000822	10102.3
8	FUR-TA-10000198	10015.1
9	TEC-MA-10001047	9989.0



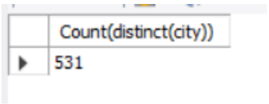
	round(sum(total_profit),2)	product_id
▶	24816	TEC-CO-10004722
	21746.4	TEC-MA-10002412
	17867.7	OFF-BI-10000545
	15948	TEC-CO-10001449
	13930.7	FUR-CH-10002024
	12792.9	OFF-BI-10003527
	11481.9	TEC-PH-10001459
	10102.3	TEC-MA-10000822
	10015.1	FUR-TA-10000198
	9989	TEC-MA-10001047

- Total Unique Cities: 531 distinct cities had orders shipped.



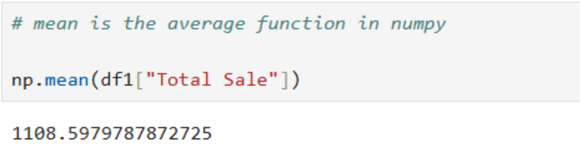
```
# To get no. of distinct/unique values in a column

df1["City"].nunique()
```

```
531
```



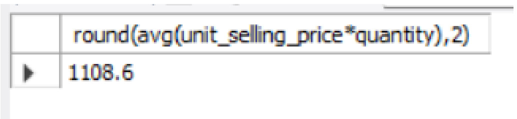
	Count(distinct(city))
▶	531

- Average Order Value (AOV): ₹1,108.60 – average customer spends around ₹1,100 per order.



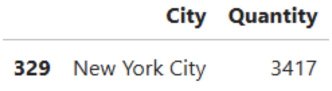
```
# mean is the average function in numpy

np.mean(df1["Total Sale"])
```

```
1108.5979787872725
```

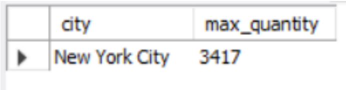


	round(avg(unit_selling_price*quantity),2)
▶	1108.6

- City with Highest Quantity of Orders: New York City – strong sales concentration in metropolitan areas.

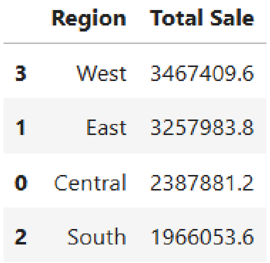


	City	Quantity
329	New York City	3417

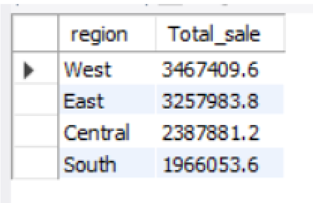


	city	max_quantity
▶	New York City	3417

- Region-Wise Total Sales: West Region recorded highest total sales followed by East.

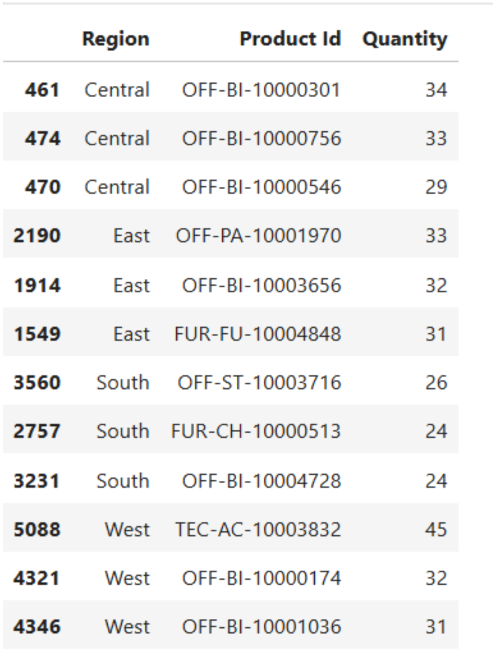


	Region	Total Sale
3	West	3467409.6
1	East	3257983.8
0	Central	2387881.2
2	South	1966053.6

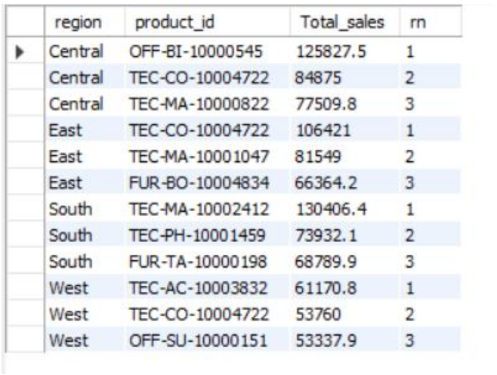


	region	Total_sale
▶	West	3467409.6
	East	3257983.8
	Central	2387881.2
	South	1966053.6

- Top 3 Selling Products by Quantity per Region: Office Supplies and Technology dominate across all regions.

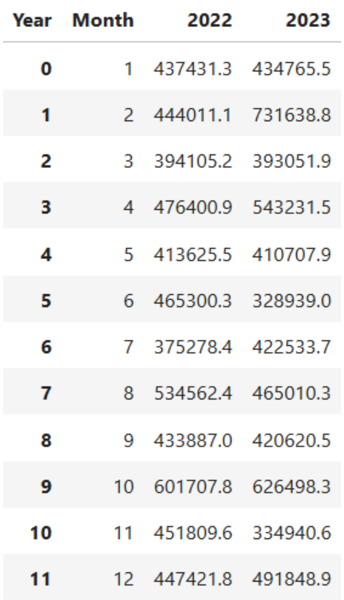


	Region	Product Id	Quantity
461	Central	OFF-BI-10000301	34
474	Central	OFF-BI-10000756	33
470	Central	OFF-BI-10000546	29
2190	East	OFF-PA-10001970	33
1914	East	OFF-BI-10003656	32
1549	East	FUR-FU-10004848	31
3560	South	OFF-ST-10003716	26
2757	South	FUR-CH-10000513	24
3231	South	OFF-BI-10004728	24
5088	West	TEC-AC-10003832	45
4321	West	OFF-BI-10000174	32
4346	West	OFF-BI-10001036	31

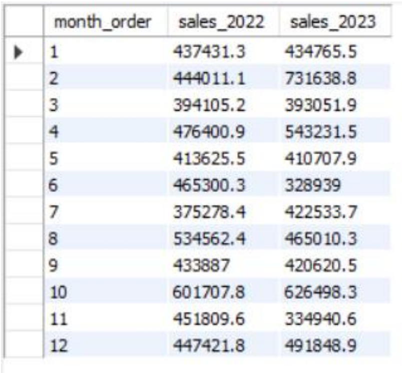


region	product_id	Total_sales	rn
Central	OFF-BI-10000545	125827.5	1
Central	TEC-CO-10004722	84875	2
Central	TEC-MA-10000822	77509.8	3
East	TEC-CO-10004722	106421	1
East	TEC-MA-10001047	81549	2
East	FUR-BO-10004834	66364.2	3
South	TEC-MA-10002412	130406.4	1
South	TEC-PH-10001459	73932.1	2
South	FUR-TA-10000198	68789.9	3
West	TEC-AC-10003832	61170.8	1
West	TEC-CO-10004722	53760	2
West	OFF-SU-10000151	53337.9	3

- Month-over-Month Sales (2022 vs 2023): February 2023 saw largest growth; October–December were peak months.

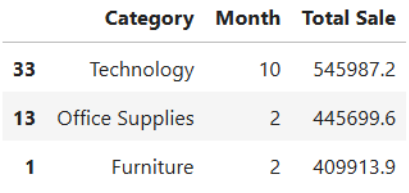


Year	Month	2022	2023
0	1	437431.3	434765.5
1	2	444011.1	731638.8
2	3	394105.2	393051.9
3	4	476400.9	543231.5
4	5	413625.5	410707.9
5	6	465300.3	328939.0
6	7	375278.4	422533.7
7	8	534562.4	465010.3
8	9	433887.0	420620.5
9	10	601707.8	626498.3
10	11	451809.6	334940.6
11	12	447421.8	491848.9

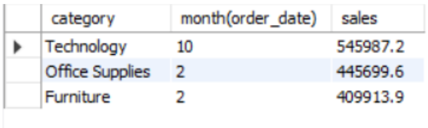


	month_order	sales_2022	sales_2023
▶	1	437431.3	434765.5
	2	444011.1	731638.8
	3	394105.2	393051.9
	4	476400.9	543231.5
	5	413625.5	410707.9
	6	465300.3	328939
	7	375278.4	422533.7
	8	534562.4	465010.3
	9	433887	420620.5
	10	601707.8	626498.3
	11	451809.6	334940.6
	12	447421.8	491848.9

- Category-Wise Month of Highest Sales: Technology peaked in October; Office Supplies and Furniture in February.

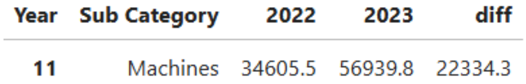


	Category	Month	Total Sale
33	Technology	10	545987.2
13	Office Supplies	2	445699.6
1	Furniture	2	409913.9

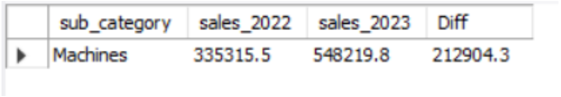


	category	month(order_date)	sales
▶	Technology	10	545987.2
	Office Supplies	2	445699.6
	Furniture	2	409913.9

- Sub-Category with Highest Profit Growth (2023 vs 2022): Machines category showed highest profit growth of ₹22,334.3.



Year	Sub Category	2022	2023	diff
11	Machines	34605.5	56939.8	22334.3



	sub_category	sales_2022	sales_2023	Diff
▶	Machines	335315.5	548219.8	212904.3

5. Key Insights Summary

- Top product category: Technology
- Top region by sales: West
- Most profitable sub-category: Machines
- Highest sales month: October (Festive Season)
- Average Order Value: ₹1,108
- Total unique cities served: 531

6. Business Recommendations

- Increase stock and promotions for Technology and Office Supplies in West and East regions.
- Leverage festive months (Oct–Dec) for high-margin items like Machines and Copiers.
- Investigate underperforming categories like Fasteners and Furnishings to improve profitability.
- Target top-performing cities with loyalty offers to retain high-value customers.
- Optimize shipping modes to improve delivery efficiency.

7. Conclusion

This retail order analysis provided a detailed view of sales trends, profitability, and regional performance. The findings can help decision-makers in strategic pricing, regional sales forecasting, inventory management, and targeted marketing. By combining Python-based EDA and SQL analytics, this project demonstrates a complete data analyst workflow—from data cleaning and transformation to actionable business insights.