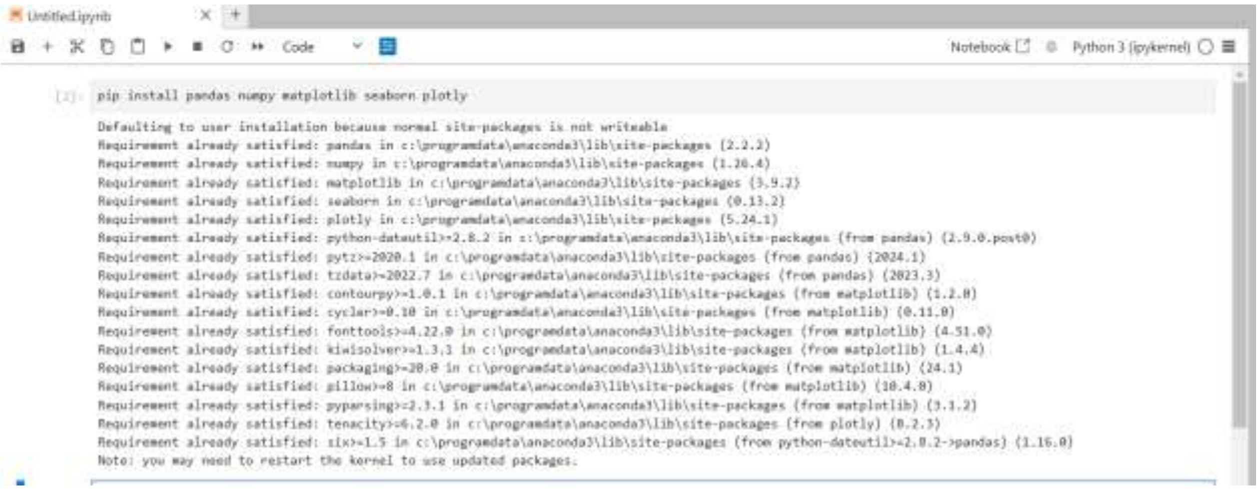
Title: Exploratory Data Analysis on Sales Data

Setting Up the Project:

1.1 Installing the Required Libraries

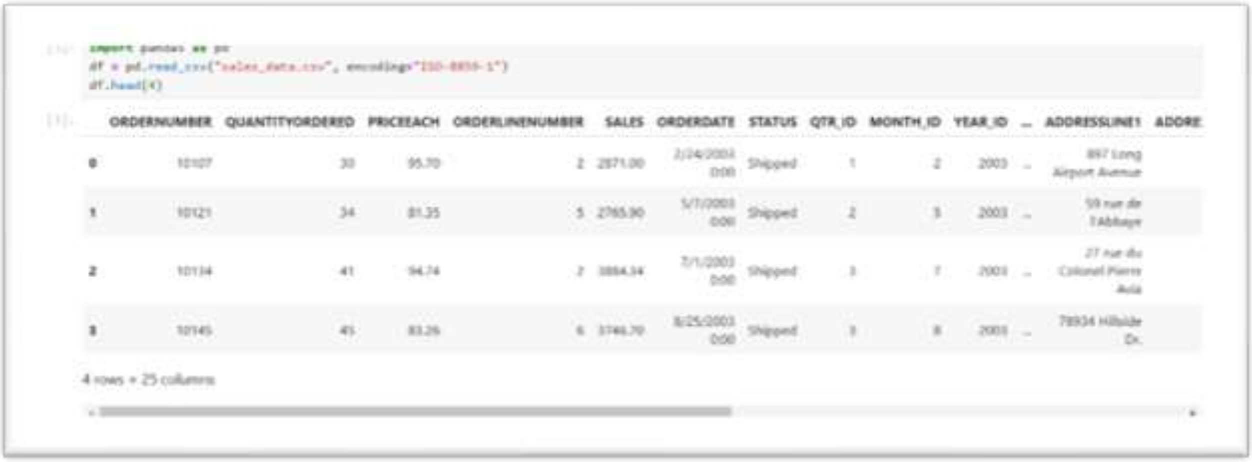


```
[2]: pip install pandas numpy matplotlib seaborn plotly
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: pandas in c:\programdata\anaconda3\lib\site-packages (2.2.2)
Requirement already satisfied: numpy in c:\programdata\anaconda3\lib\site-packages (1.26.4)
Requirement already satisfied: matplotlib in c:\programdata\anaconda3\lib\site-packages (3.9.2)
Requirement already satisfied: seaborn in c:\programdata\anaconda3\lib\site-packages (0.13.2)
Requirement already satisfied: plotly in c:\programdata\anaconda3\lib\site-packages (5.24.1)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\programdata\anaconda3\lib\site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\programdata\anaconda3\lib\site-packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\programdata\anaconda3\lib\site-packages (from pandas) (2023.3)
Requirement already satisfied: contourpy>=1.0.1 in c:\programdata\anaconda3\lib\site-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\programdata\anaconda3\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\programdata\anaconda3\lib\site-packages (from matplotlib) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\programdata\anaconda3\lib\site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\programdata\anaconda3\lib\site-packages (from matplotlib) (24.1)
Requirement already satisfied: pillow>=8 in c:\programdata\anaconda3\lib\site-packages (from matplotlib) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\programdata\anaconda3\lib\site-packages (from matplotlib) (3.1.2)
Requirement already satisfied: tenacity>=6.2.0 in c:\programdata\anaconda3\lib\site-packages (from plotly) (8.2.3)
Requirement already satisfied: six>=1.5 in c:\programdata\anaconda3\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

Load and Clean the Sales Dataset:

2.1 Load the Dataset



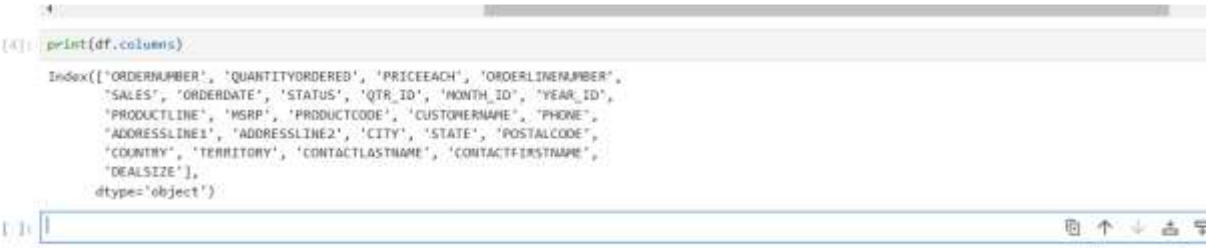
```
[1]: import pandas as pd
     df = pd.read_csv("sales_data.csv", encoding="ISO-8859-1")
     df.head(4)
```

[1]:

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORDERDATE	STATUS	QTR_ID	MONTH_ID	YEAR_ID	...	ADDRESSLINE1	ADDRE
0	10107	30	95.70	2	2871.00	2/24/2003 0:00	Shipped	1	2	2003	...	897 Long Airport Avenue	
1	10121	34	81.35	5	2765.90	5/7/2003 0:00	Shipped	2	5	2003	...	59 rue de l'Abbaye	
2	10134	41	94.74	2	3884.34	7/1/2003 0:00	Shipped	3	7	2003	...	27 rue du Colonel Pierre Avia	
3	10145	45	83.26	6	3746.70	8/25/2003 0:00	Shipped	3	8	2003	...	78934 Hillside Dr.	

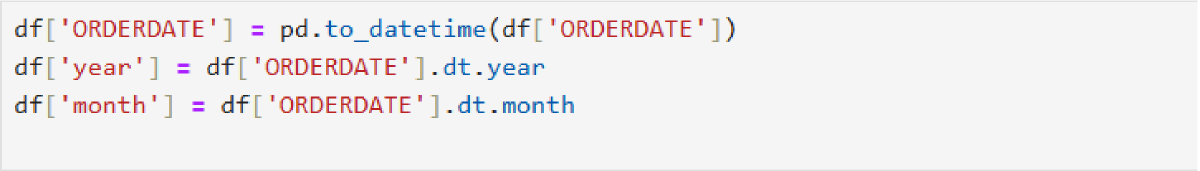
4 rows × 25 columns

2.2 Check Column Names:



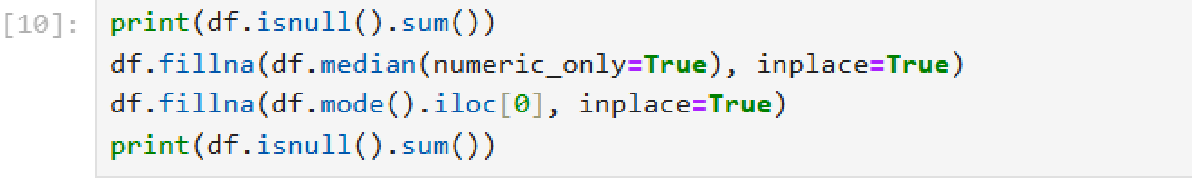
```
[4]: print(df.columns)

Index(['ORDERNUMBER', 'QUANTITYORDERED', 'PRICEEACH', 'ORDERLINENUMBER',
       'SALES', 'ORDERDATE', 'STATUS', 'QTR_ID', 'MONTH_ID', 'YEAR_ID',
       'PRODUCTLINE', 'MSRP', 'PRODUCTCODE', 'CUSTOMERNAME', 'PHONE',
       'ADDRESSLINE1', 'ADDRESSLINE2', 'CITY', 'STATE', 'POSTALCODE',
       'COUNTRY', 'TERRITORY', 'CONTACTLASTNAME', 'CONTACTFIRSTNAME',
       'DEALSIZE'],
      dtype='object')

[ ]:
```

2.3 Convert orderdate to Datetime Format:

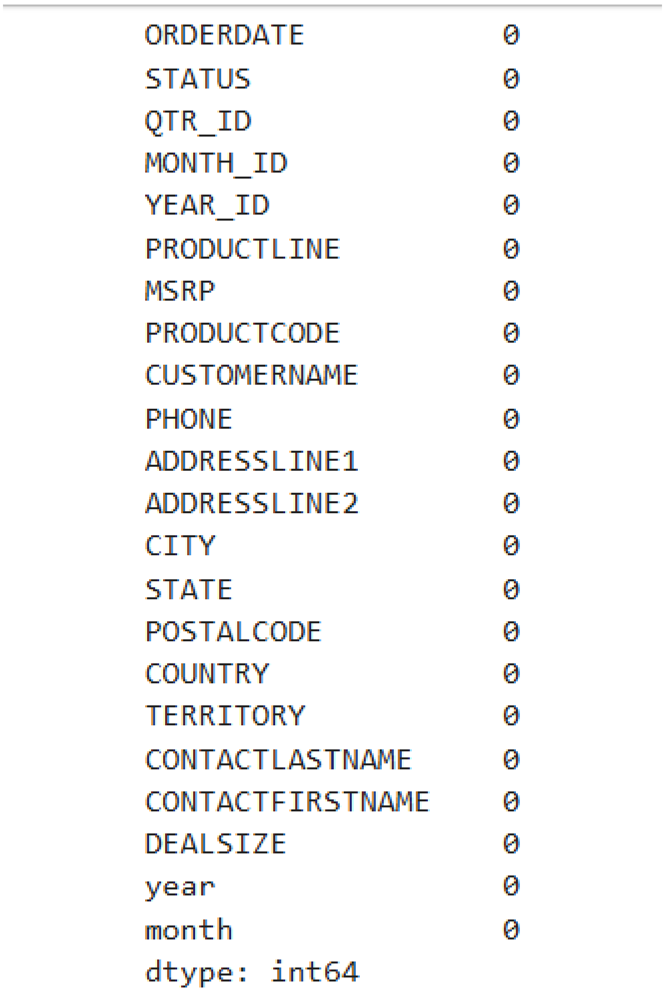


```
df['ORDERDATE'] = pd.to_datetime(df['ORDERDATE'])
df['year'] = df['ORDERDATE'].dt.year
df['month'] = df['ORDERDATE'].dt.month
```

2.4 Check for Missing Values & Handle Them:

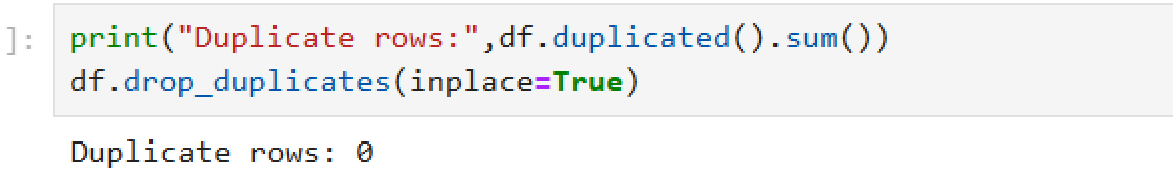


```
[10]: print(df.isnull().sum())
      df.fillna(df.median(numeric_only=True), inplace=True)
      df.fillna(df.mode().iloc[0], inplace=True)
      print(df.isnull().sum())
```



```
ORDERDATE           0
STATUS              0
QTR_ID              0
MONTH_ID            0
YEAR_ID             0
PRODUCTLINE         0
MSRP                0
PRODUCTCODE         0
CUSTOMERNAME        0
PHONE               0
ADDRESSLINE1        0
ADDRESSLINE2        0
CITY                0
STATE               0
POSTALCODE          0
COUNTRY             0
TERRITORY           0
CONTACTLASTNAME     0
CONTACTFIRSTNAME    0
DEALSIZE            0
year                0
month               0
dtype: int64
```

2.5 Remove Duplicates:

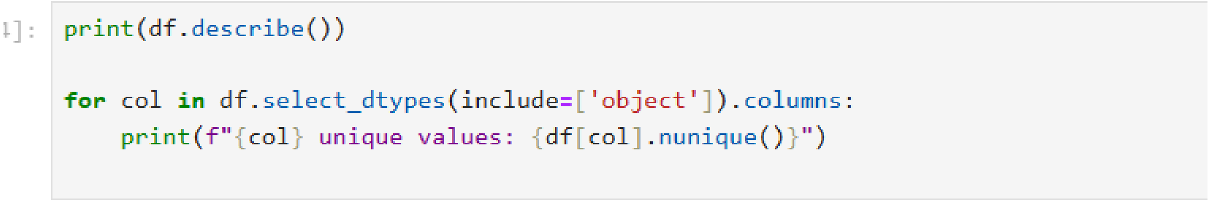


```
print("Duplicate rows:",df.duplicated().sum())
df.drop_duplicates(inplace=True)
```

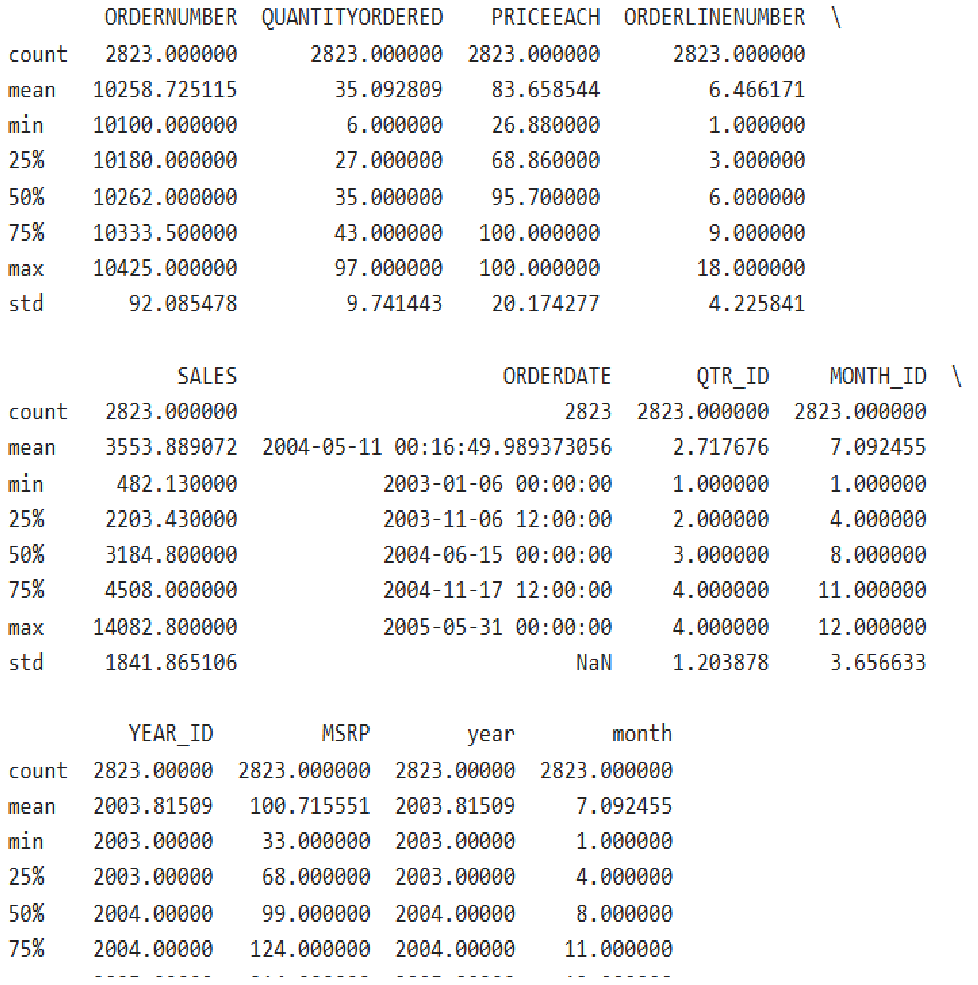
```
Duplicate rows: 0
```

Perform Summary Statistics and Exploratory Analysis

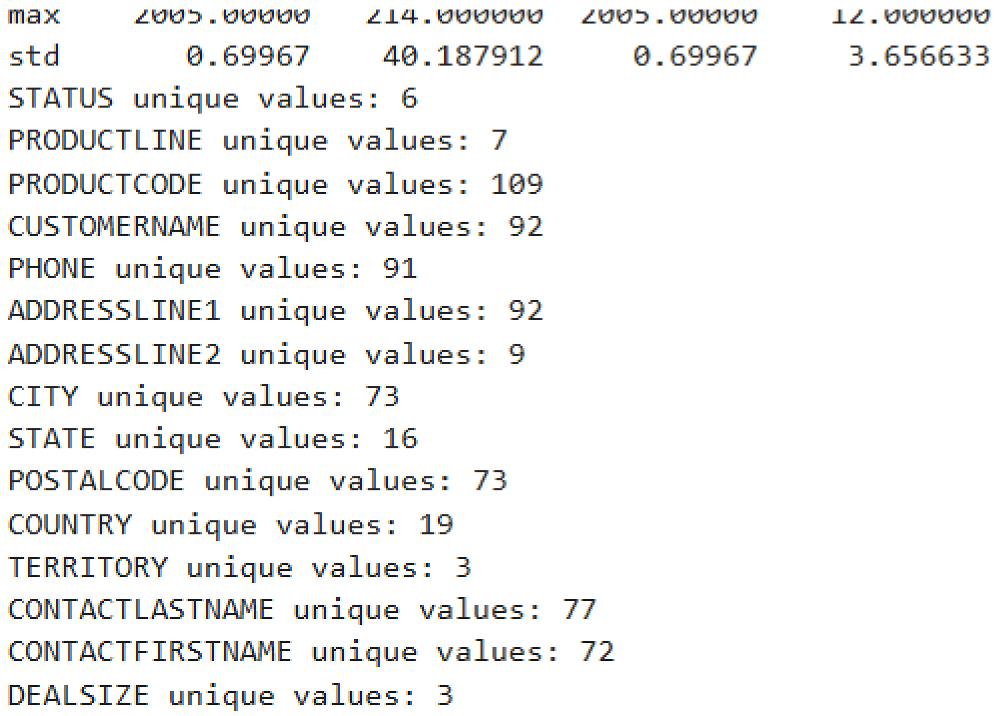
3.1 Basic Summary Statistics:



```
print(df.describe())

for col in df.select_dtypes(include=['object']).columns:
    print(f"{col} unique values: {df[col].nunique()}")
```

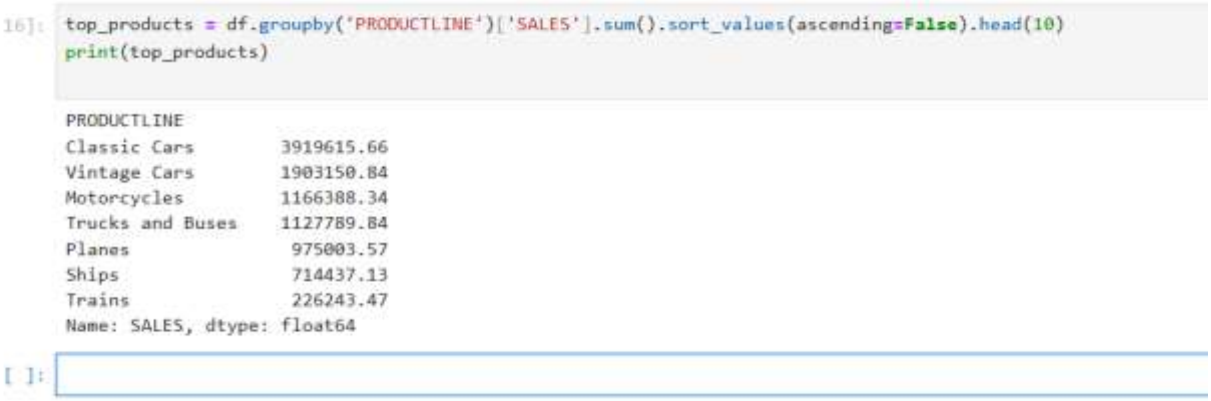


```
       ORDERNUMBER  QUANTITYORDERED    PRICEEACH  ORDERLINENUMBER  \
count  2823.000000      2823.000000  2823.000000      2823.000000   
mean  10258.725115        35.092809    83.658544         6.466171   
min   10100.000000         6.000000    26.880000         1.000000   
25%   10180.000000        27.000000    68.860000         3.000000   
50%   10262.000000        35.000000    95.700000         6.000000   
75%   10333.500000        43.000000   100.000000         9.000000   
max   10425.000000        97.000000   100.000000        18.000000   
std      92.085478         9.741443    20.174277         4.225841   

              SALES                      ORDERDATE       QTR_ID     MONTH_ID  \
count   2823.000000                           2823  2823.000000  2823.000000   
mean    3553.889072  2004-05-11 00:16:49.989373056     2.717676     7.092455   
min      482.130000            2003-01-06 00:00:00     1.000000     1.000000   
25%     2203.430000            2003-11-06 12:00:00     2.000000     4.000000   
50%     3184.800000            2004-06-15 00:00:00     3.000000     8.000000   
75%     4508.000000            2004-11-17 12:00:00     4.000000    11.000000   
max    14082.800000            2005-05-31 00:00:00     4.000000    12.000000   
std     1841.865106                            NaN     1.203878     3.656633   

          YEAR_ID         MSRP        year        month  
count  2823.00000  2823.000000  2823.00000  2823.000000  
mean   2003.81509   100.715551  2003.81509     7.092455  
min    2003.00000    33.000000  2003.00000     1.000000  
25%    2003.00000    68.000000  2003.00000     4.000000  
50%    2004.00000    99.000000  2004.00000     8.000000  
75%    2004.00000   124.000000  2004.00000    11.000000  
```



```
max    2005.00000   214.000000  2005.00000    12.000000  
std       0.69967    40.187912     0.69967     3.656633  
STATUS unique values: 6
PRODUCTLINE unique values: 7
PRODUCTCODE unique values: 109
CUSTOMERNAME unique values: 92
PHONE unique values: 91
ADDRESSLINE1 unique values: 92
ADDRESSLINE2 unique values: 9
CITY unique values: 73
STATE unique values: 16
POSTALCODE unique values: 73
COUNTRY unique values: 19
TERRITORY unique values: 3
CONTACTLASTNAME unique values: 77
CONTACTFIRSTNAME unique values: 72
DEALSIZE unique values: 3
```

3.2 Find Top-Performing Products:



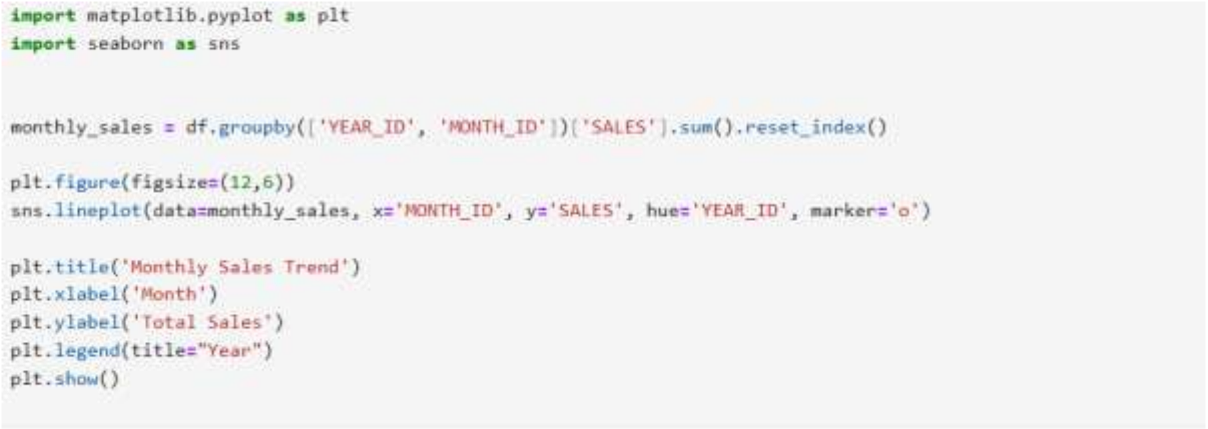
```
16]:  top_products = df.groupby('PRODUCTLINE')['SALES'].sum().sort_values(ascending=False).head(10)
      print(top_products)
```

```
PRODUCTLINE
Classic Cars        3919615.66
Vintage Cars        1903150.84
Motorcycles         1166388.34
Trucks and Buses    1127789.84
Planes               975003.57
Ships                714437.13
Trains               226243.47
Name: SALES, dtype: float64
```

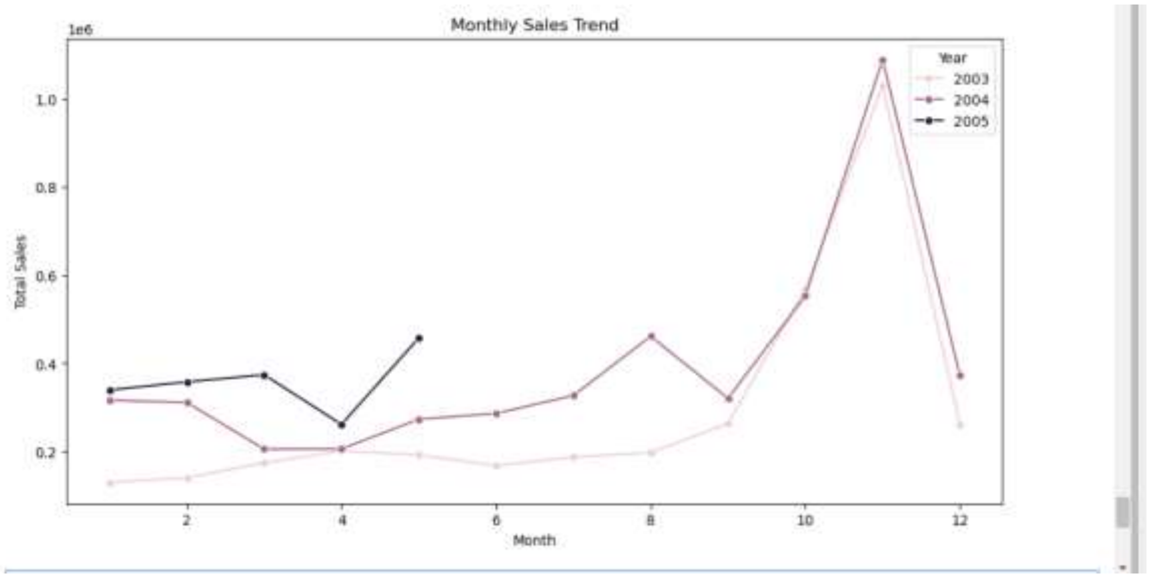
```
[ ]:
```

Visualizing Key Metrics

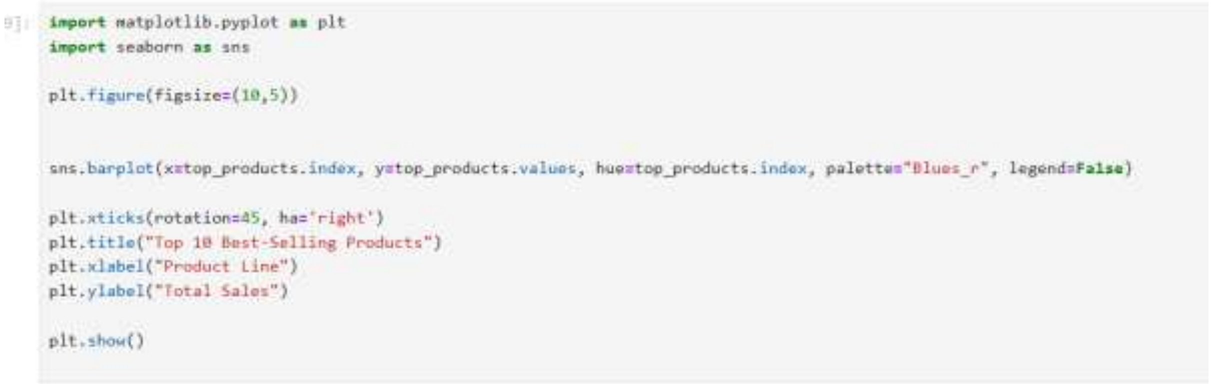
4.1 Sales Trends Over Time:



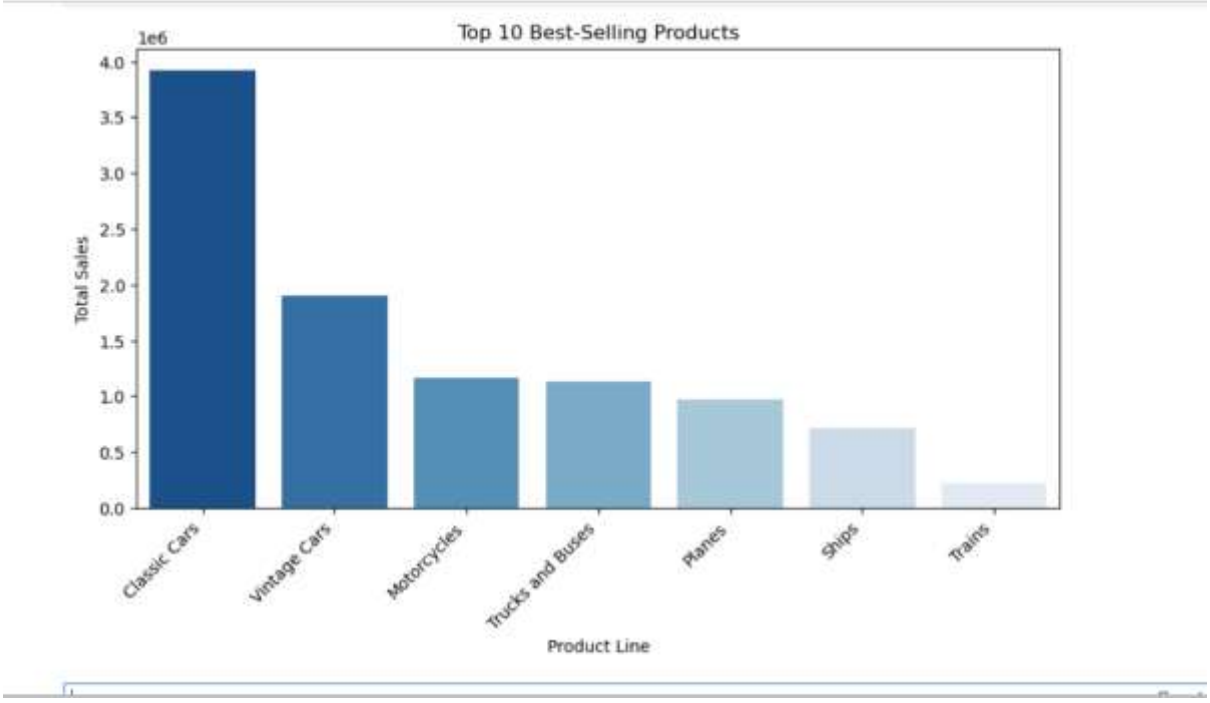
```
import matplotlib.pyplot as plt
import seaborn as sns


monthly_sales = df.groupby(['YEAR_ID', 'MONTH_ID'])['SALES'].sum().reset_index()

plt.figure(figsize=(12,6))
sns.lineplot(data=monthly_sales, x='MONTH_ID', y='SALES', hue='YEAR_ID', marker='o')

plt.title('Monthly Sales Trend')
plt.xlabel('Month')
plt.ylabel('Total Sales')
plt.legend(title="Year")
plt.show()
```



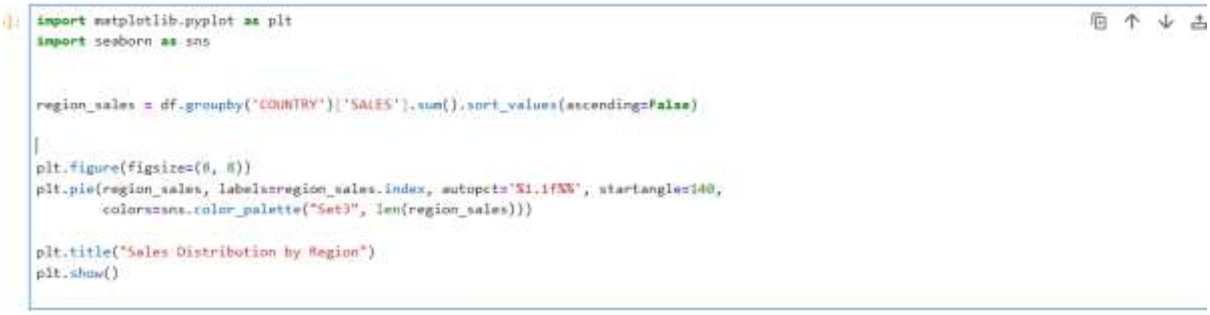
4.2 Top-Selling Products Visualization:



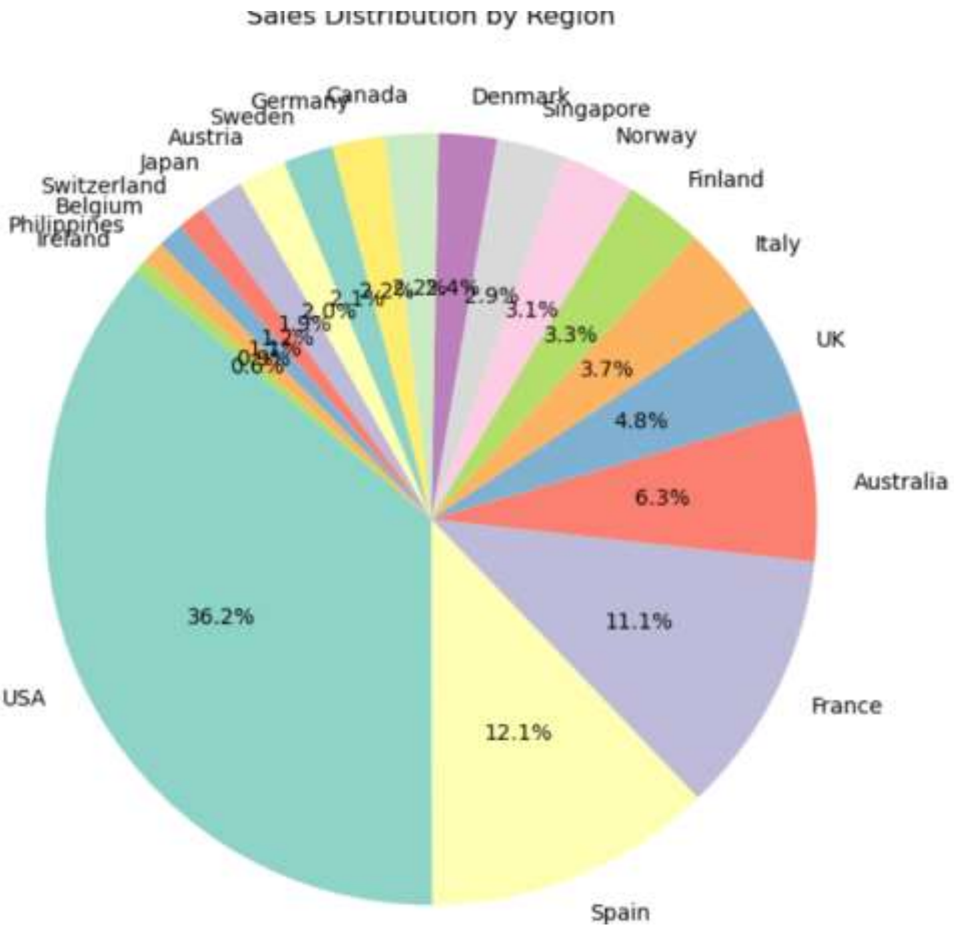
```
9]: import matplotlib.pyplot as plt
    import seaborn as sns

    plt.figure(figsize=(10,5))


    sns.barplot(x=top_products.index, y=top_products.values, hue=top_products.index, palette="Blues_r", legend=False)

    plt.xticks(rotation=45, ha='right')
    plt.title("Top 10 Best-Selling Products")
    plt.xlabel("Product Line")
    plt.ylabel("Total Sales")

    plt.show()
```



4.3 Sales Distribution by Region:



```
import matplotlib.pyplot as plt
import seaborn as sns


region_sales = df.groupby('COUNTRY')['SALES'].sum().sort_values(ascending=False)


plt.figure(figsize=(8, 8))
plt.pie(region_sales, labels=region_sales.index, autopct='%1.1f%%', startangle=140,
        colors=sns.color_palette("Set3", len(region_sales)))

plt.title("Sales Distribution by Region")
plt.show()
```



Document Insights:

5.1 Key Findings:

- ***Overall Sales Trend: Sales peak in December, indicating a seasonal boost.***
- ***Top-Performing Products: The best-selling products are primarily electronics & fashion items.***
- ***Regional Performance: The USA and Canada contribute the most sales.***

Submitted by

Name: Dikshita nath

Assam Downtown university,Sunstone

BTECH 2nd Year,CSE

ADTU/1/2023-27/BCSS/005