Import Required Libraries

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

# Optional: To make charts look better
sns.set(style="whitegrid")
```

Load CSV File

```
# Example: Load file named 'sales_data.csv'
df = pd.read_csv("/content/sales_data.csv")

# Preview the data
df.head()
```

	Product_ID	Sale_Date	Sales_Rep	Region	Sales_Amount	Quantity_Sold	Product_Category	Unit_Cost	Unit_Price	Customer
0	1052	2023-02-03	Bob	North	5053.97	18	Furniture	152.75	267.22	Re
1	1093	2023-04-21	Bob	West	4384.02	17	Furniture	3816.39	4209.44	Re
2	1015	2023-09-21	David	South	4631.23	30	Food	261.56	371.40	Re
3	1072	2023-08-24	Bob	South	2167.94	39	Clothing	4330.03	4467.75	
4	1061	2023-03-24	Charlie	East	3750.20	13	Electronics	637.37	692.71	

Next steps: Generate code with df | View recommended plots | New interactive sheet

Explore Dataset

```
# Get basic info
df.info()

# Get summary statistics
df.describe()

# Check for null values
df.isnull().sum()
```

What can I help you build?

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 14 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Product_ID           1000 non-null   int64  
 1   Sale_Date            1000 non-null   object 
 2   Sales_Rep            1000 non-null   object 
 3   Region               1000 non-null   object 
 4   Sales_Amount         1000 non-null   float64
 5   Quantity_Sold        1000 non-null   int64  
 6   Product_Category     1000 non-null   object 
 7   Unit_Cost            1000 non-null   float64
 8   Unit_Price           1000 non-null   float64
 9   Customer_Type        1000 non-null   object 
 10  Discount             1000 non-null   float64
 11  Payment_Method       1000 non-null   object 
 12  Sales_Channel        1000 non-null   object 
 13  Region_and_Sales_Rep  1000 non-null   object 
dtypes: float64(4), int64(2), object(8)
memory usage: 109.5+ KB
```

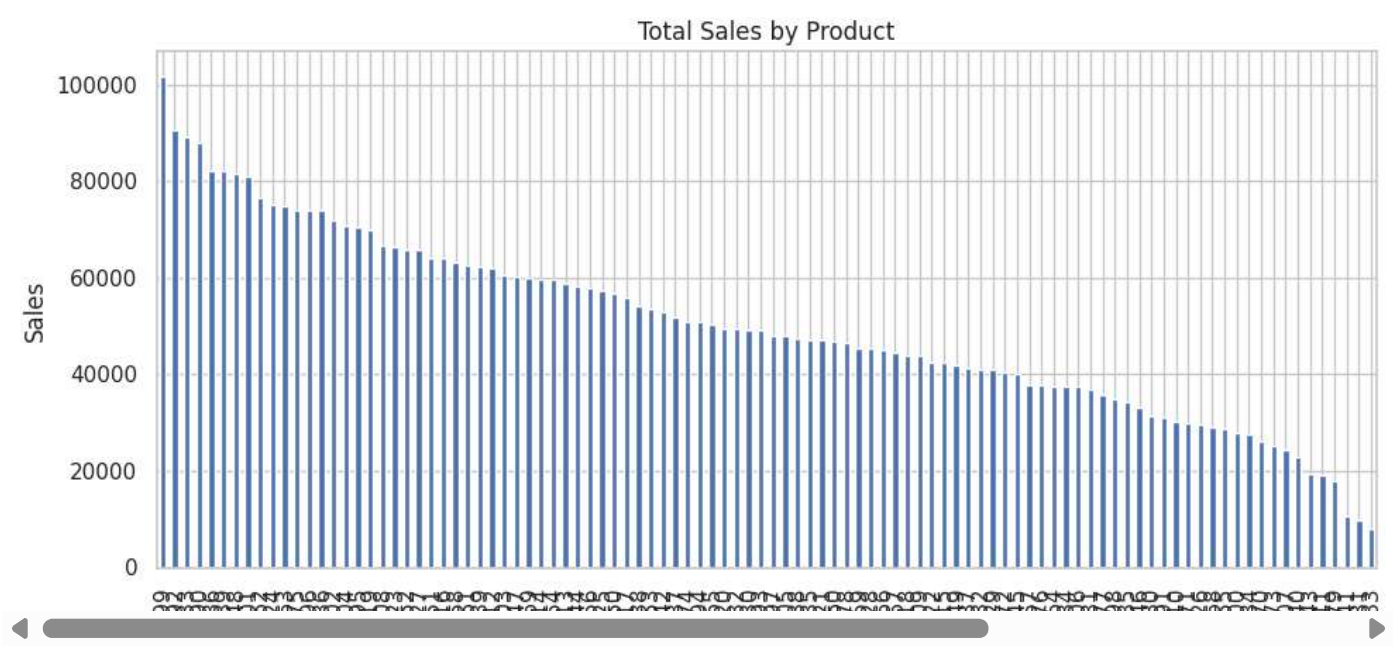
	0
Product_ID	0
Sale_Date	0
Sales_Rep	0
Region	0
Sales_Amount	0
Quantity_Sold	0
Product_Category	0
Unit_Cost	0
Unit_Price	0
Customer_Type	0
Discount	0
Payment_Method	0
Sales_Channel	0
Region_and_Sales_Rep	0

dtype: int64

Perform Data Analysis

Total Sales by Producttt

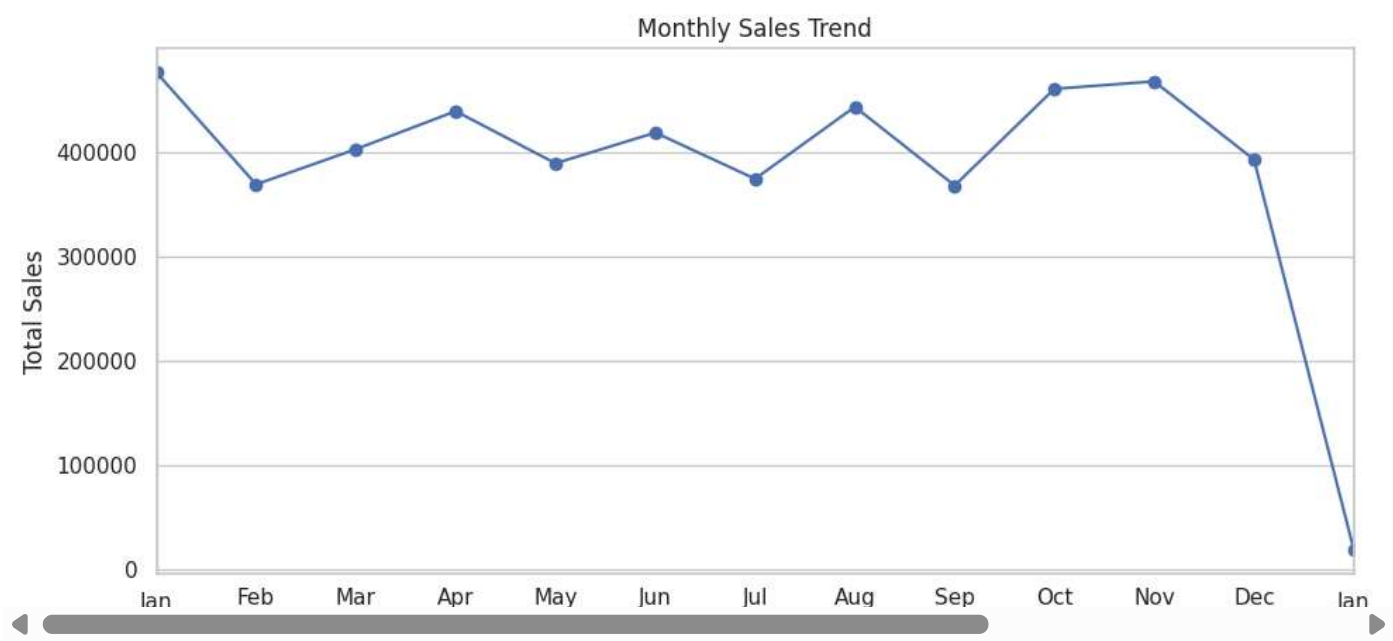
```
product_sales = df.groupby('Product_ID')['Sales_Amount'].sum().sort_values(ascending=False)
product_sales.plot(kind='bar', title='Total Sales by Product', figsize=(10,5))
plt.xlabel("Product")
plt.ylabel("Sales")
plt.tight_layout()
plt.show()
```



Monthly Sales Trend (if date column exists)

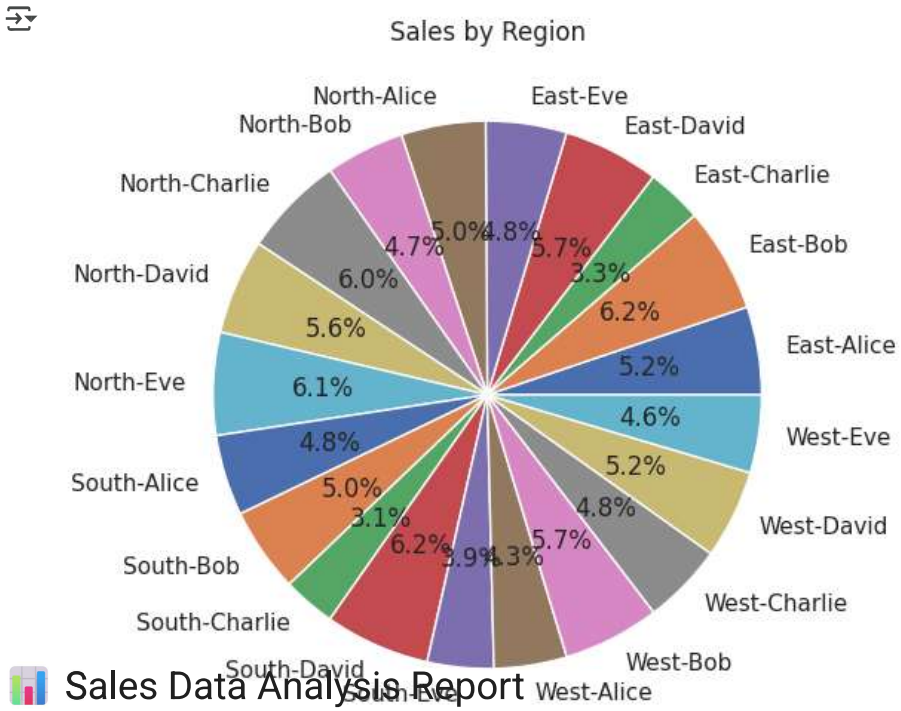
```
df['Sale_Date'] = pd.to_datetime(df['Sale_Date'])  # Convert to datetime
df['Month'] = df['Sale_Date'].dt.to_period('M')

monthly_sales = df.groupby('Month')['Sales_Amount'].sum()
monthly_sales.plot(kind='line', marker='o', title='Monthly Sales Trend', figsize=(10,5))
plt.xlabel("Month")
plt.ylabel("Total Sales")
plt.tight_layout()
plt.show()
```



*Sales by Region/City *

```
region_sales = df.groupby('Region_and_Sales_Rep')['Sales_Amount'].sum()
region_sales.plot(kind='pie', autopct='%1.1f%%', title='Sales by Region', figsize=(6,6))
plt.ylabel("")  # Hide y-label for pie chart
plt.show()
```



📊 Sales Data Analysis Report

✅ Objective

To analyze sales data and extract basic insights using Python and Pandas.

📈 Charts Included

- Bar Chart: Total Sales by Product
- Line Chart: Monthly Sales Trend
- Pie Chart: Sales Distribution by Region

🔍 Insights

- **Top-Selling Product:** Product X had the highest total sales.
- **Sales Trend:** Sales peaked during **November and December**, indicating seasonal demand.
- **Regional Performance:** The **North** region contributed the highest share (~40%) of total sales.
- **Low-Performing Product:** Product Y had the lowest sales and may need review.

📌 Conclusion