```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
data = pd.read_csv("/content/sales_data_sample.csv", encoding='latin1')
data.head(10)
```

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORDERDATE	STATUS	QTR_ID	MONTH_ID	YEAR_ID	...	ADDRESSLINE
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 D
4	10159	49	100.00	14	5205.27	10/10/2003 0:00	Shipped	4	10	2003	...	7734 Strong St
5	10168	36	96.66	1	3479.76	10/28/2003 0:00	Shipped	4	10	2003	...	9408 Furth Circle
6	10180	29	86.13	9	2497.77	11/11/2003 0:00	Shipped	4	11	2003	...	184, chaussée de Tournai
7	10188	48	100.00	1	5512.32	11/18/2003 0:00	Shipped	4	11	2003	...	Drammen 121, PR 744 Sentrum
8	10201	22	98.57	2	2168.54	12/1/2003 0:00	Shipped	4	12	2003	...	5557 North Pendale Street
9	10211	41	100.00	14	4708.44	1/15/2004 0:00	Shipped	1	1	2004	...	25, rue Lauristo

10 rows × 25 columns

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2823 entries, 0 to 2822
Data columns (total 25 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   ORDERNUMBER       2823 non-null   int64
 1   QUANTITYORDERED   2823 non-null   int64
 2   PRICEEACH         2823 non-null   float64
 3   ORDERLINENUMBER   2823 non-null   int64
 4   SALES             2823 non-null   float64
 5   ORDERDATE         2823 non-null   object
 6   STATUS            2823 non-null   object
 7   QTR_ID            2823 non-null   int64
 8   MONTH_ID          2823 non-null   int64
 9   YEAR_ID           2823 non-null   int64
 10  PRODUCTLINE       2823 non-null   object
 11  MSRP              2823 non-null   int64
 12  PRODUCTCODE       2823 non-null   object
 13  CUSTOMERNAME      2823 non-null   object
 14  PHONE             2823 non-null   object
 15  ADDRESSLINE1      2823 non-null   object
 16  ADDRESSLINE2      302 non-null    object
 17  CITY              2823 non-null   object
 18  STATE             1337 non-null   object
 19  POSTALCODE        2747 non-null   object
 20  COUNTRY           2823 non-null   object
 21  TERRITORY         1749 non-null   object
 22  CONTACTLASTNAME   2823 non-null   object
 23  CONTACTFIRSTNAME  2823 non-null   object
 24  DEALSIZE          2823 non-null   object
dtypes: float64(2), int64(7), object(16)
memory usage: 551.5+ KB
```

DATA PREPROCESSING

```
# Check columns
print(data.columns)

# Convert ORDERDATE to datetime
data['ORDERDATE'] = pd.to_datetime(data['ORDERDATE'])
```

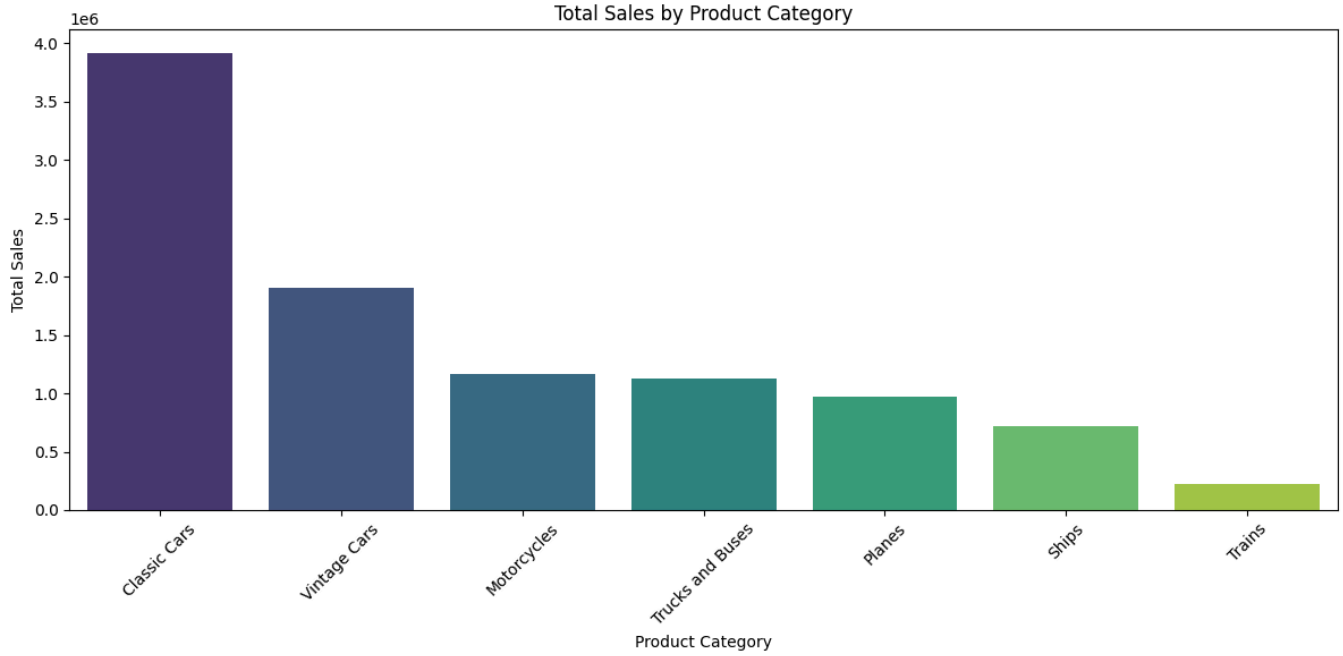
```
# Ensure SALES is numeric
data['SALES'] = pd.to_numeric(data['SALES'], errors='coerce')
```

```
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')
```

Bar Plot – Sales by Product Category

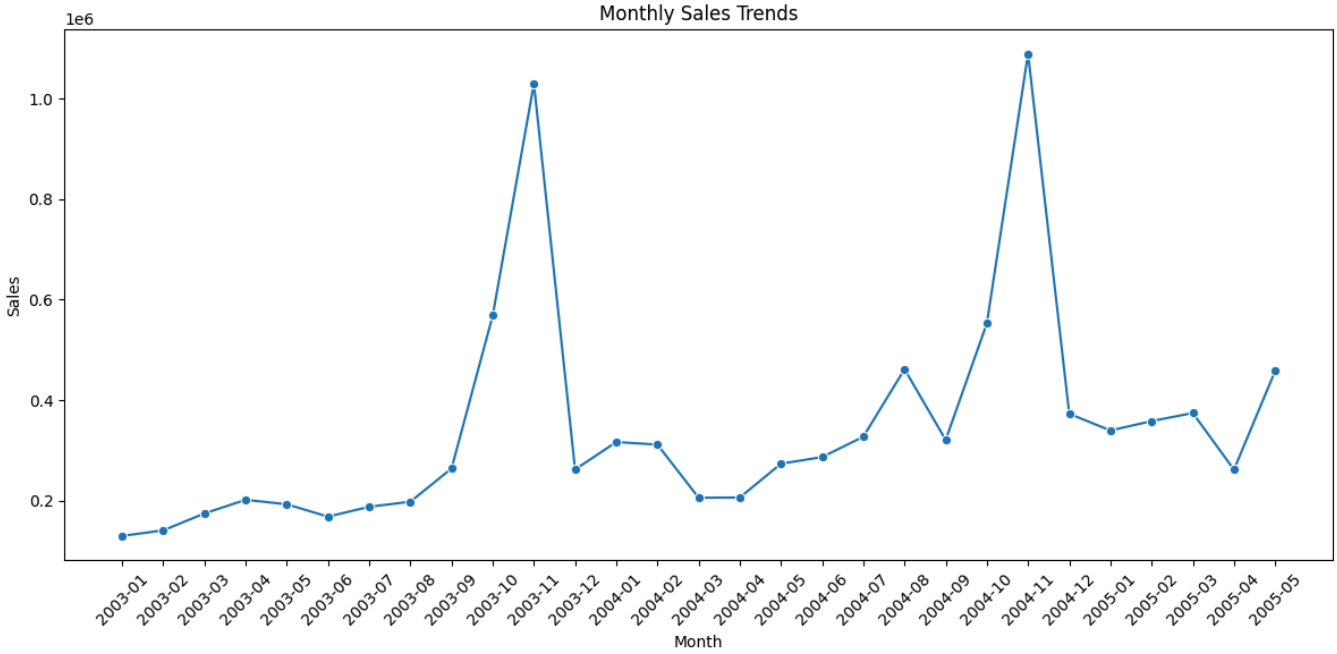
```
plt.figure(figsize=(12,6))
category_sales = data.groupby('PRODUCTLINE')['SALES'].sum().sort_values(ascending=False)
sns.barplot(x=category_sales.index, y=category_sales.values, palette='viridis')
plt.title('Total Sales by Product Category')
plt.ylabel('Total Sales')
plt.xlabel('Product Category')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
/tmp/ipython-input-18-1983515024.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `le

  sns.barplot(x=category_sales.index, y=category_sales.values, palette='viridis')
```



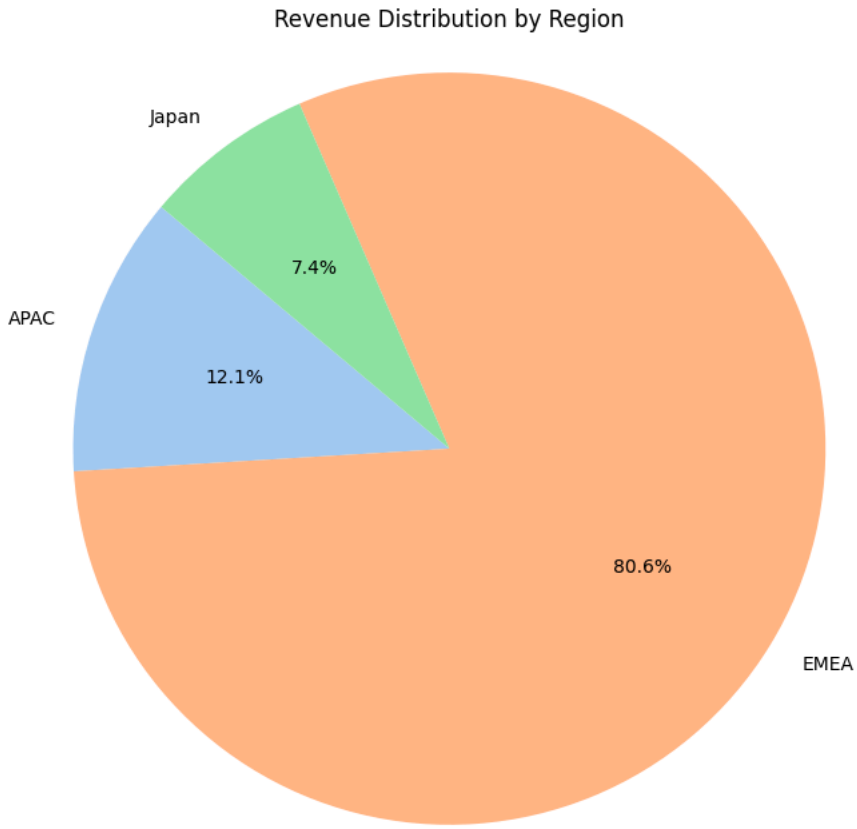
Line Chart – Monthly Sales Trends

```
# Create a 'Month' column
data['Month'] = data['ORDERDATE'].dt.to_period('M').astype(str)

monthly_sales = data.groupby('Month')['SALES'].sum().reset_index()

plt.figure(figsize=(12,6))
sns.lineplot(data=monthly_sales, x='Month', y='SALES', marker='o')
plt.title('Monthly Sales Trends')
plt.xticks(rotation=45)
plt.ylabel('Sales')
plt.xlabel('Month')
plt.tight_layout()
plt.show()
```



Revenue Distribution by Region

```
region_col = 'TERRITORY' if 'TERRITORY' in data.columns else 'COUNTRY'

region_sales = data.groupby(region_col)['SALES'].sum()
plt.figure(figsize=(8,8))
plt.pie(region_sales, labels=region_sales.index, autopct='%1.1f%%', startangle=140, colors=sns.color_palette("pastel"))
plt.title("Revenue Distribution by Region")
plt.axis('equal')  # Equal aspect ratio ensures pie is drawn as a circle.
plt.show()
```



Order Value Distribution

```
plt.figure(figsize=(10,6))
sns.histplot(data['SALES'], bins=30, kde=True, color='coral')
plt.title("Order Value Distribution")
plt.xlabel("Order Value ($)")
plt.ylabel("Frequency")
plt.tight_layout()
plt.show()
```