In [1]:
```
import pandas as pd

# 1. Import dataset
file_path = "Week-2-Sales-Data (2).csv"
df = pd.read_csv(file_path)
```

In [2]:
```
# Preview first rows
print("First 5 rows:")
display(df.head())
```

```
First 5 rows:
```

	Order_ID	Product	Region	Units_Sold	Unit_Price	Revenue	Sales_Rep	Order_Date
0	ORD001	Printer	Limpopo	45	2985	134325	Rep-2	2024-03-28
1	ORD002	Headphones	Western Cape	16	15076	241216	Rep-18	2024-04-11
2	ORD003	Laptop	Western Cape	45	14860	668700	Rep-16	2024-05-18
3	ORD004	External Hard Drive	KwaZulu-Natal	21	16237	340977	Rep-3	2024-05-16
4	ORD005	Smartphone	Western Cape	41	9420	386220	Rep-17	2024-02-21

In [3]:
```
# 2. Check for missing values, duplicates, and data types
print("\nMissing values:")
print(df.isnull().sum())

print("\nDuplicates:")
print(df.duplicated().sum())

print("\nData types:")
print(df.dtypes)
```

```
Missing values:
Order_ID      0
Product       0
Region        0
Units_Sold    0
Unit_Price    0
Revenue       0
Sales_Rep     0
Order_Date    0
dtype: int64

Duplicates:
0

Data types:
Order_ID      object
Product       object
Region        object
Units_Sold     int64
Unit_Price     int64
Revenue        int64
Sales_Rep     object
Order_Date    object
dtype: object
```

In [4]:

```
# 3. Data Cleaning
# Remove duplicates
df = df.drop_duplicates()

# Handle missing values -> drop rows with NA (you can also use fillna if required)
df = df.dropna()
```

In [5]:

```
# Convert Order_Date column to datetime
if 'Order_Date' in df.columns:
    df['Order_Date'] = pd.to_datetime(df['Order_Date'], errors='coerce')
```

In [6]:

```
print("First 5 rows of dataset:")
display(df.head())
```

```
First 5 rows of dataset:
```

	Order_ID	Product	Region	Units_Sold	Unit_Price	Revenue	Sales_Rep	Order_Date
0	ORD001	Printer	Limpopo	45	2985	134325	Rep-2	2024-03-28
1	ORD002	Headphones	Western Cape	16	15076	241216	Rep-18	2024-04-11
2	ORD003	Laptop	Western Cape	45	14860	668700	Rep-16	2024-05-18
3	ORD004	External Hard Drive	KwaZulu-Natal	21	16237	340977	Rep-3	2024-05-16
4	ORD005	Smartphone	Western Cape	41	9420	386220	Rep-17	2024-02-21

In [7]:

```python
# a) Total revenue for the entire dataset
df["Revenue"] = df["Units_Sold"] * df["Unit_Price"]
total_revenue = df["Revenue"].sum()
print(f"Total Revenue: {total_revenue:,.2f}")
```

```
Total Revenue: 35,295,338.00
```

In [8]:

```python
# b) Average units sold per order
avg_units_sold = df["Units_Sold"].mean()
print(f"Average Units Sold per Order: {avg_units_sold:.2f}")
```

```
Average Units Sold per Order: 28.23
```

In [9]:

```python
# c) Total revenue per region
revenue_per_region = df.groupby("Region")["Revenue"].sum().sort_values(ascending=False)
print("\nRevenue per Region:")
print(revenue_per_region)
```

```
Revenue per Region:
Region
Western Cape     9346198
Gauteng          6231531
North West       6201288
Limpopo          3614655
KwaZulu-Natal    3560630
Free State       3359398
Eastern Cape     2981638
Name: Revenue, dtype: int64
```

In [10]:

```python
# d) Highest revenue-generating sales representative
top_sales_rep = df.groupby("Sales_Rep")["Revenue"].sum().sort_values(ascending=False).head(1)
print("\nHighest Revenue-Generating Sales Rep:")
print(top_sales_rep)
```

```
Highest Revenue-Generating Sales Rep:
Sales_Rep
Rep-19    2889294
Name: Revenue, dtype: int64
```

In [11]:

```
# e) Top 3 products by total units sold
top_products = df.groupby("Product")["Units_Sold"].sum().sort_values(ascending=False).head(3)
print("\nTop 3 Products by Units Sold:")
print(top_products)
```

```
Top 3 Products by Units Sold:
Product
Smartwatch    542
Tablet        511
Smartphone    437
Name: Units_Sold, dtype: int64
```

In [ ]: