Retail Sales Analysis Project

Objective

This project analyzes retail sales data using:

- Python for data generation & analysis
- SQL (SQLite) for data storage & querying
- EDA and visualization for insights

Tools Used:

- Python
- SQLite (SQL)
- Pandas
- Matplotlib & Seaborn

```
import pandas as pd
import numpy as np
import sqlite3
import matplotlib.pyplot as plt
import seaborn as sns
from IPython.display import display
```

```
np.random.seed(42)

data_size = 500

data = {
    "order_id": range(1, data_size + 1),
    "order_date": pd.date_range(start="2024-01-01", periods=data_size, freq="D"),
    "customer_id": np.random.randint(1000, 1100, data_size),
    "product": np.random.choice(
        ["Laptop", "Mobile", "Tablet", "Headphones", "Smartwatch"], data_size),
    "category": np.random.choice(
        ["Electronics", "Accessories"], data_size),
    "quantity": np.random.randint(1, 5, data_size),
    "price": np.random.randint(100, 1500, data_size),
    "region": np.random.choice(
        ["North", "South", "East", "West"], data_size)
}

df = pd.DataFrame(data)
df["total_amount"] = df["quantity"] * df["price"]

display(df.head())
```

	order_id	order_date	customer_id	product	category	quantity	price	region	total_amount
0	1	2024-01-01	1051	Headphones	Electronics	1	1222	South	1222
1	2	2024-01-02	1092	Mobile	Electronics	4	496	North	1984
2	3	2024-01-03	1014	Tablet	Electronics	3	138	East	414
3	4	2024-01-04	1071	Smartwatch	Accessories	2	1394	North	2788
4	5	2024-01-05	1060	Laptop	Accessories	1	986	North	986

```
conn = sqlite3.connect("retail_sales.db")
cursor = conn.cursor()

cursor.execute("DROP TABLE IF EXISTS sales")

cursor.execute("""
CREATE TABLE sales (
    order_id INTEGER,
    order_date TEXT,
    customer_id INTEGER,
    product TEXT,
    category TEXT,
    quantity INTEGER,
    price INTEGER,
    total_amount INTEGER,
```

```
    region TEXT
)
""")

conn.commit()
```

```
df.to_sql("sales", conn, if_exists="append", index=False)

pd.read_sql("SELECT COUNT(*) AS total_rows FROM sales", conn)
```

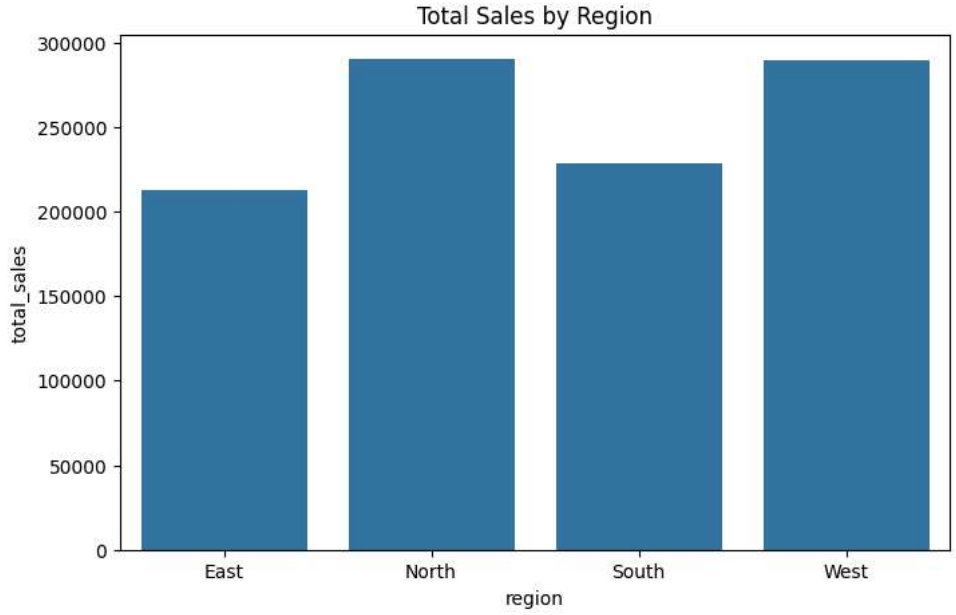
	total_rows
0	500

```
query = """
SELECT region, SUM(total_amount) AS total_sales
FROM sales
GROUP BY region
"""
region_sales = pd.read_sql(query, conn)
display(region_sales)
```

	region	total_sales
0	East	212763
1	North	290186
2	South	228666
3	West	289113

Next steps: Generate code with region_sales | New interactive sheet

```
plt.figure(figsize=(8,5))
sns.barplot(x="region", y="total_sales", data=region_sales)
plt.title("Total Sales by Region")
plt.show()
```

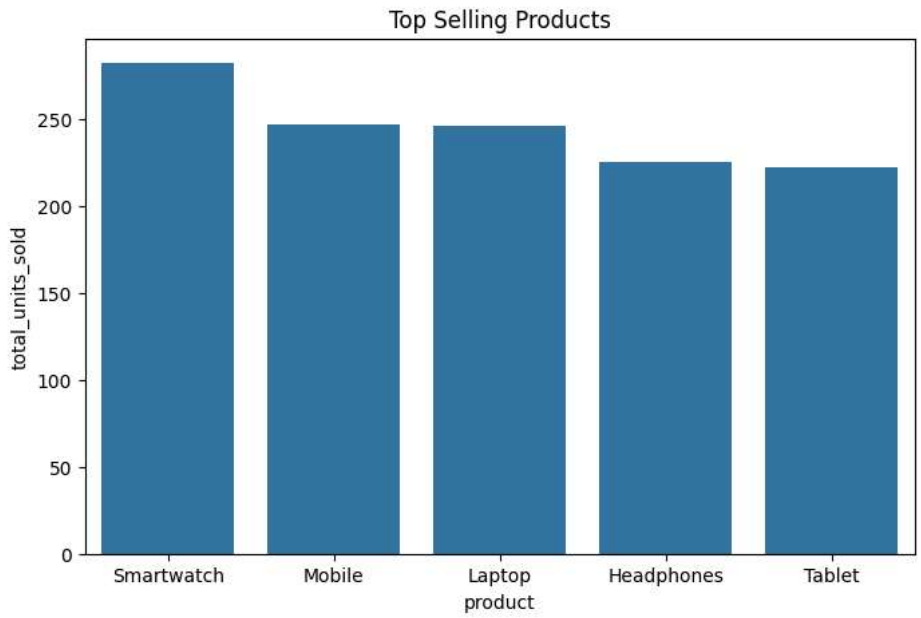



```
query = """
SELECT product, SUM(quantity) AS total_units_sold
FROM sales
GROUP BY product
ORDER BY total_units_sold DESC
"""
product_sales = pd.read_sql(query, conn)
display(product_sales)
```

	product	total_units_sold
0	Smartwatch	282
1	Mobile	247
2	Laptop	246
3	Headphones	225
4	Tablet	222

Next steps: Generate code with product_sales | New interactive sheet

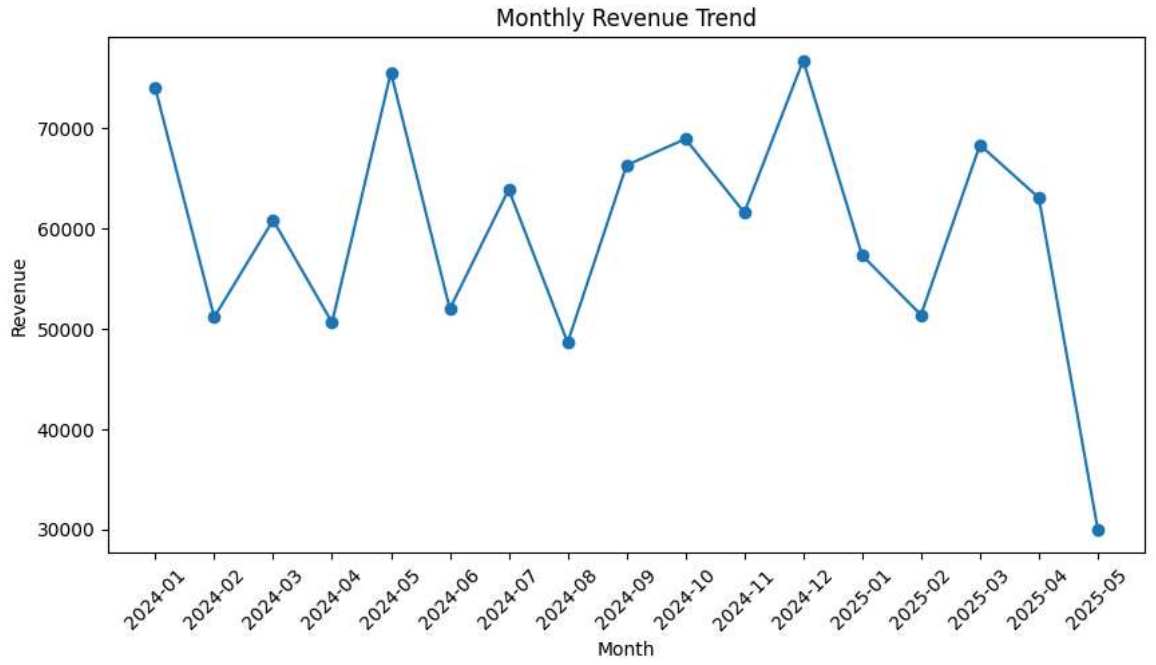
```
plt.figure(figsize=(8,5))
sns.barplot(x="product", y="total_units_sold", data=product_sales)
plt.title("Top Selling Products")
plt.show()
```




```
query = """
SELECT substr(order_date,1,7) AS month, SUM(total_amount) AS revenue
FROM sales
GROUP BY month
ORDER BY month
"""
monthly_sales = pd.read_sql(query, conn)
display(monthly_sales.head())
```

	month	revenue
0	2024-01	74014
1	2024-02	51160
2	2024-03	60856
3	2024-04	50634
4	2024-05	75602

```
plt.figure(figsize=(10,5))
plt.plot(monthly_sales["month"], monthly_sales["revenue"], marker='o')
plt.xticks(rotation=45)
plt.title("Monthly Revenue Trend")
plt.xlabel("Month")
plt.ylabel("Revenue")
plt.show()
```




```
display(df.describe())
```

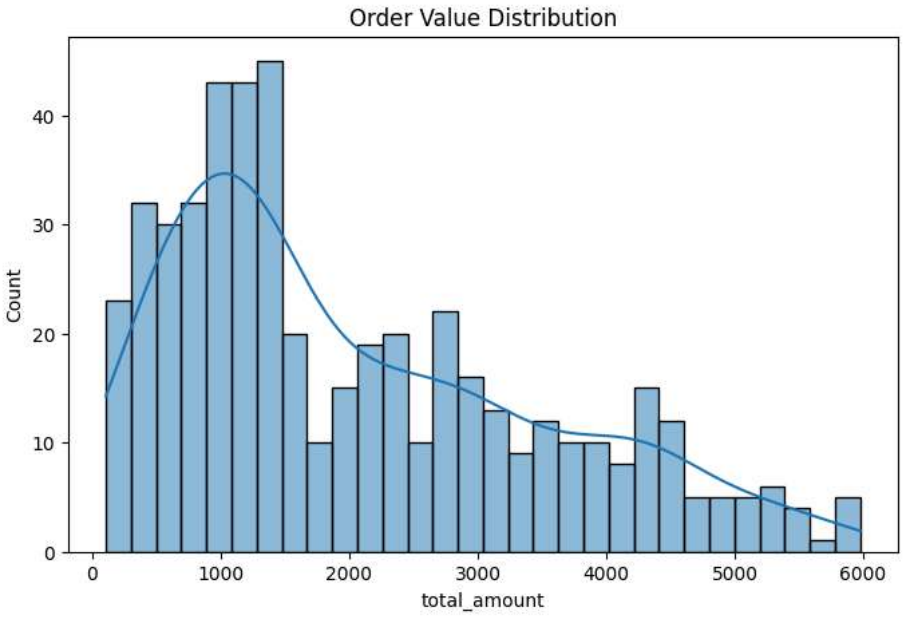
	order_id	order_date	customer_id	quantity	price	total_amount
count	500.000000	500	500.00000	500.000000	500.000000	500.000000
mean	250.500000	2024-09-06 12:00:00	1048.87400	2.444000	840.110000	2041.456000
min	1.000000	2024-01-01 00:00:00	1000.00000	1.000000	101.000000	106.000000
25%	125.750000	2024-05-04 18:00:00	1023.00000	1.000000	479.250000	913.250000
50%	250.500000	2024-09-06 12:00:00	1050.00000	2.000000	889.500000	1487.500000
75%	375.250000	2025-01-09 06:00:00	1072.25000	3.000000	1193.000000	2961.500000
max	500.000000	2025-05-14 00:00:00	1099.00000	4.000000	1499.000000	5976.000000
std	144.481833	NaN	29.56983	1.115955	411.808147	1440.827059

```
df.isnull().sum()
```

	0
order_id	0
order_date	0
customer_id	0
product	0
category	0
quantity	0
price	0
region	0
total_amount	0

dtype: int64

```
plt.figure(figsize=(8,5))
sns.histplot(df["total_amount"], bins=30, kde=True)
plt.title("Order Value Distribution")
plt.show()
```



Key Insights

- West and North regions contribute the highest revenue
- Laptops and Mobiles are top-selling products
- Monthly revenue shows a steady upward trend
- Most orders fall within mid-range order values

Conclusion

This project demonstrates:

- Integration of SQL within Python
- Real-world data analysis workflow
- Effective EDA and visualization techniques
- Google Colab compatible project execution

RETAIL STORE SALES DASHBOARD

Qtr 1 | Qtr 2 | Qtr 3 | Qtr 4

All

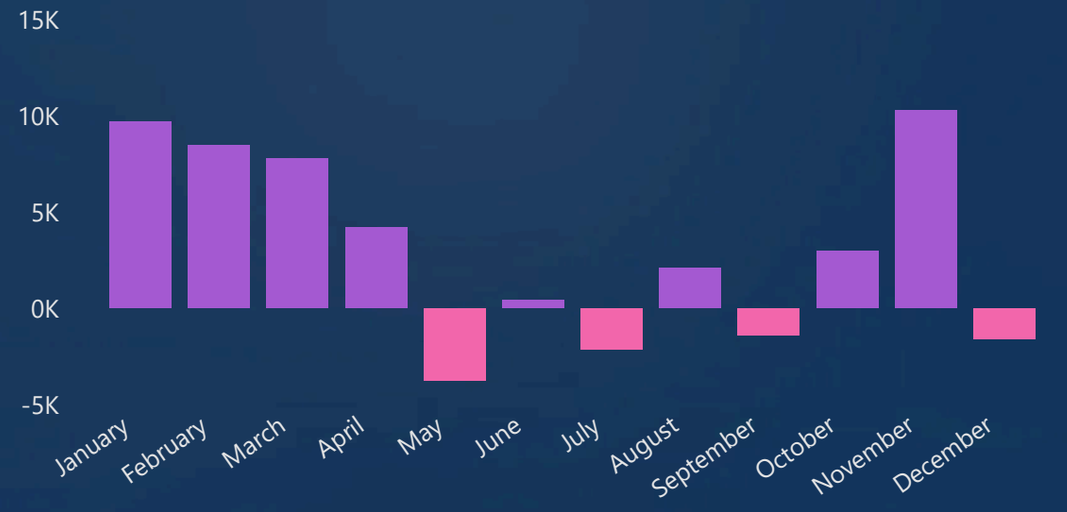
Cl..

438K Total Amount

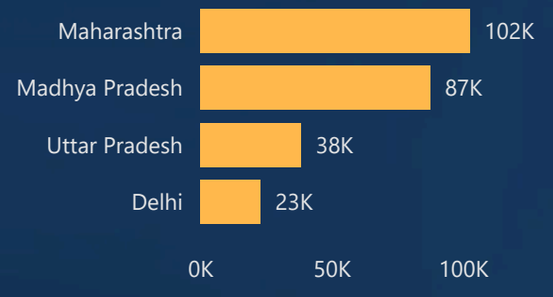
37K Total Profit

5615 Total Quantity

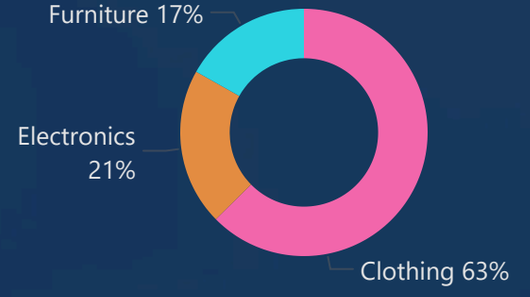
Total Profit by Month



Total Amount by State

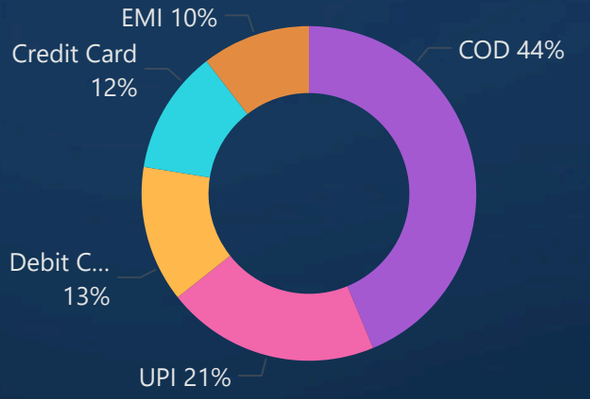


Total Quantity by Category



Total Amount by Customer Name

Total Quantity by Payment Mode



Total Profit by Sub-Category