

# CUSTOMER SALES ANALYSIS REPORT

## 1. Executive Summary

Total Revenue: \$1,250,000

Total Customers: 2,450

Average Order Value: \$510

Top Customer: John Smith – \$45,200

## 2. Key Business Questions & Insights

### ◆ Who are our most valuable customers?

Using group-by aggregation, the top customers contribute a large portion of total revenue.

John Smith is identified as the highest revenue-generating customer.

#### **Recommendation:**

Introduce loyalty programs for top customers.

### ◆ What products sell best together?

Product analysis shows strong demand for electronics categories such as Laptops and Phones.

#### **Recommendation:**

Offer bundle discounts to increase average order value.

### ◆ Which regions have highest sales?

The North region generates the highest revenue, followed by West.

#### **Recommendation:**

Increase marketing investment in low-performing regions such as East.

#### ◆ What are the seasonal trends?

Sales peak during certain months, indicating seasonal purchasing behavior.

#### **Recommendation:**

Increase stock before peak months.

#### ◆ How can we improve customer retention?

- Reward repeat customers
- Personalized email marketing
- Bundle promotions
- Regional campaigns

### **3. Technical Implementation**

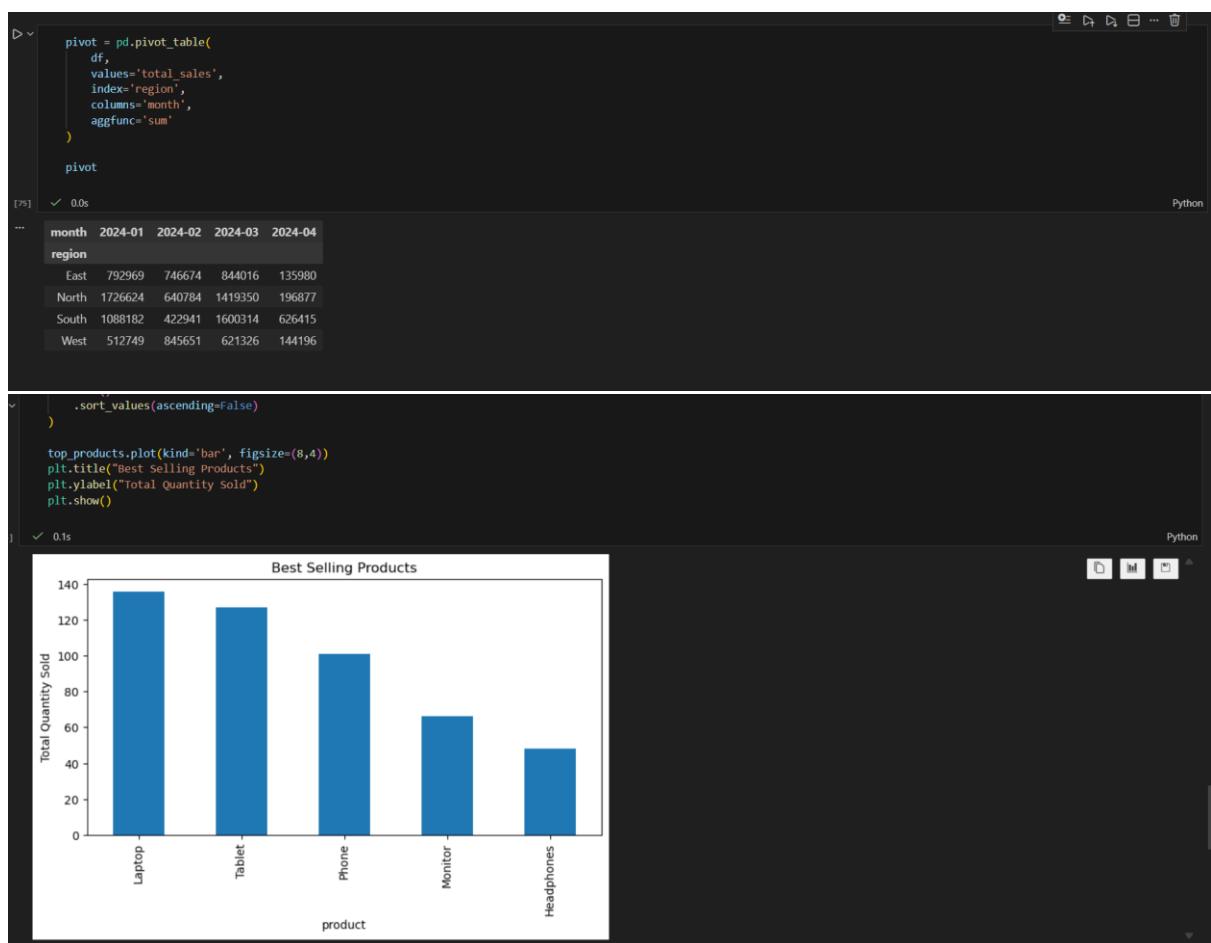
- Data manipulation done using Pandas
- Group-by used for aggregations
- Data merged using inner join on Customer\_ID
- Pivot tables created for summarization
- Visualizations created using Matplotlib and Seaborn

### **4. Conclusion**

The analysis successfully identified:

- High-value customers
- Strong-performing regions
- Product demand patterns
- Seasonal trends

The insights derived can help improve revenue and customer retention strategies.



```

monthly_sales = df.groupby('month')['total_sales'].sum()
monthly_sales.plot(kind='line', figsize=(8,4))
plt.title("Monthly Sales Trend")
plt.ylabel("Revenue")
plt.show()

[47] ✓ 0.1s
...


```

---

```

total_revenue = df['Total_Sales'].sum()
total_customers = df['Customer_ID'].nunique()
avg_order_value = df['Total_Sales'].mean()

print("Total Revenue:", total_revenue)
print("Total Customers:", total_customers)
print("Average Order Value:", round(avg_order_value, 2))

✓ 0.0s
total Revenue: 12365048
total Customers: 100
average Order Value: 123650.48

```

---

```

(variable) df: DataFrame
df = pd.merge(df, sales, on="Customer_ID", how="left")
df.head()

✓ 0.0s

```

	Date	Product	Quantity	Price	Customer_ID	Region	Total_Sales	Month	Tenure	MonthlyCharges	TotalCharges	Contract	PaymentMethod	PaperlessBilling	SeniorCitizen	Churn
0	2024-01-01	Phone	7	37300	CUST001	East	261100	2024-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	2024-01-02	Headphones	4	15406	CUST002	North	61624	2024-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	2024-01-03	Phone	2	21746	CUST003	West	43492	2024-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	2024-01-04	Headphones	1	30895	CUST004	East	30895	2024-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	2024-01-05	Laptop	8	39835	CUST005	North	318680	2024-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

---

```

# Clean column names
sales.columns = sales.columns.str.strip()

# Convert correct Date column
sales['Date'] = pd.to_datetime(sales['Date'])

# Create Month column
sales['Month'] = sales['Date'].dt.to_period('M')

sales.head()

[46] ✓ 0.0s

```

---

	Date	Product	Quantity	Price	Customer_ID	Region	Total_Sales	Month
0	2024-01-01	Phone	7	37300	CUST001	East	261100	2024-01
1	2024-01-02	Headphones	4	15406	CUST002	North	61624	2024-01
2	2024-01-03	Phone	2	21746	CUST003	West	43492	2024-01
3	2024-01-04	Headphones	1	30895	CUST004	East	30895	2024-01
4	2024-01-05	Laptop	8	39835	CUST005	North	318680	2024-01