

```
# 📦 Importing Essentials
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# 🔍 Read Data
customers = pd.read_csv("customers.csv")
orders = pd.read_csv("orders.csv")
order_items = pd.read_csv("order_items.csv")
products = pd.read_csv("products.csv")
```

```
# ✅ Merge DataFrames
df = order_items.merge(products, on='sku', how='left') \
    .merge(orders, on='order_id', how='left') \
    .merge(customers, on='customer_id', how='left')
```

```
df.rename(columns={'return_flag': 'is_returned'}, inplace=True)
df['is_returned'] = df['is_returned'].astype(bool)
```

```
# 🧠 Data Types + Preview
df['order_date'] = pd.to_datetime(df['order_date'])
df['is_returned'] = df['is_returned'].astype(bool)
print(df.dtypes)
print(df.head())
```

```
➡ order_id      int64
   sku          int64
   quantity     int64
```

```

price                float64
is_returned          bool
product_name          object
category              object
cost_price            float64
customer_id           int64
order_date            datetime64[ns]
total_amount          float64
region                object
signup_date           object
loyalty_flag          int64
profit                float64
dtype: object

```

	order_id	sku	quantity	price	is_returned	product_name	category \
0	3001	2047	4	150.01	False	Product_2047	Electronics
1	3002	2036	4	148.91	False	Product_2036	Beauty
2	3002	2020	3	192.63	False	Product_2020	Home
3	3002	2006	4	96.25	False	Product_2006	Beauty
4	3003	2012	2	125.52	False	Product_2012	Clothing

	cost_price	customer_id	order_date	total_amount	region	signup_date \
0	63.17	1054	2023-03-16	408.41	West	2022-12-26
1	73.18	1060	2023-05-26	256.71	North	2022-09-14
2	22.61	1060	2023-05-26	256.71	North	2022-09-14
3	96.73	1060	2023-05-26	256.71	North	2022-09-14
4	25.25	1097	2023-03-17	428.94	South	2022-01-28

	loyalty_flag	profit
0	1	347.36
1	1	302.92
2	1	510.06
3	1	-1.92
4	0	200.54

#  Feature Engineering

```

df['profit'] = (df['price'] - df['cost_price']) * df['quantity']
df['return_flag'] = df['is_returned'].astype(int)

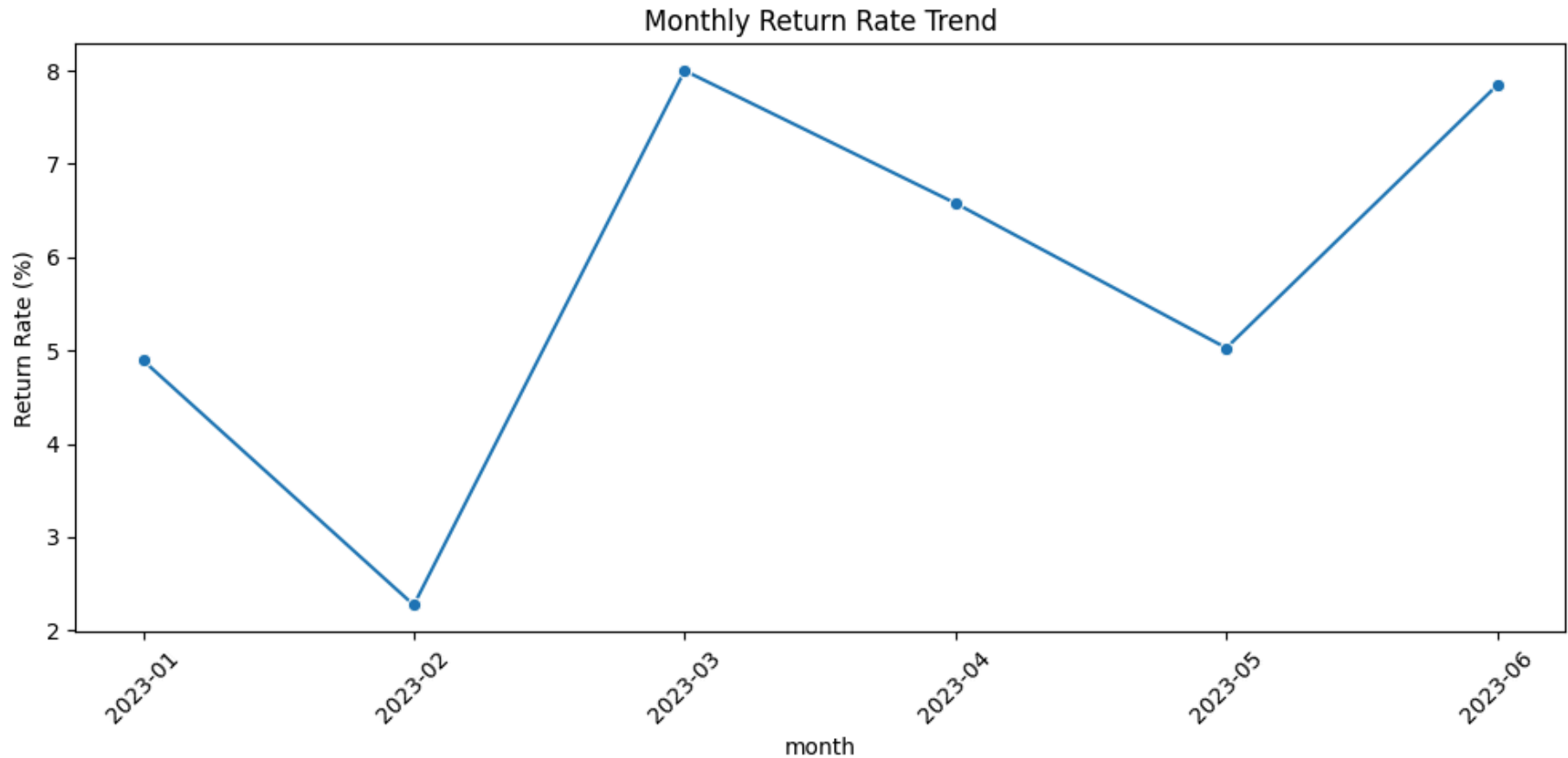
```


```
# 📊 1. Overall Return Rate
total_qty = df['quantity'].sum()
returned_qty = df[df['is_returned'] == True]['quantity'].sum()
overall_return_rate = 100 * returned_qty / total_qty
print(f"Return Rate: {overall_return_rate:.2f}%")
```

➡ Return Rate: 14.44%

```
# 📈 2. Monthly Return Trend
df['month'] = df['order_date'].dt.to_period('M')
monthly = df.groupby('month').agg({
    'quantity': 'sum',
    'return_flag': 'sum'
}).reset_index()
monthly['return_rate'] = 100 * monthly['return_flag'] / monthly['quantity']
```

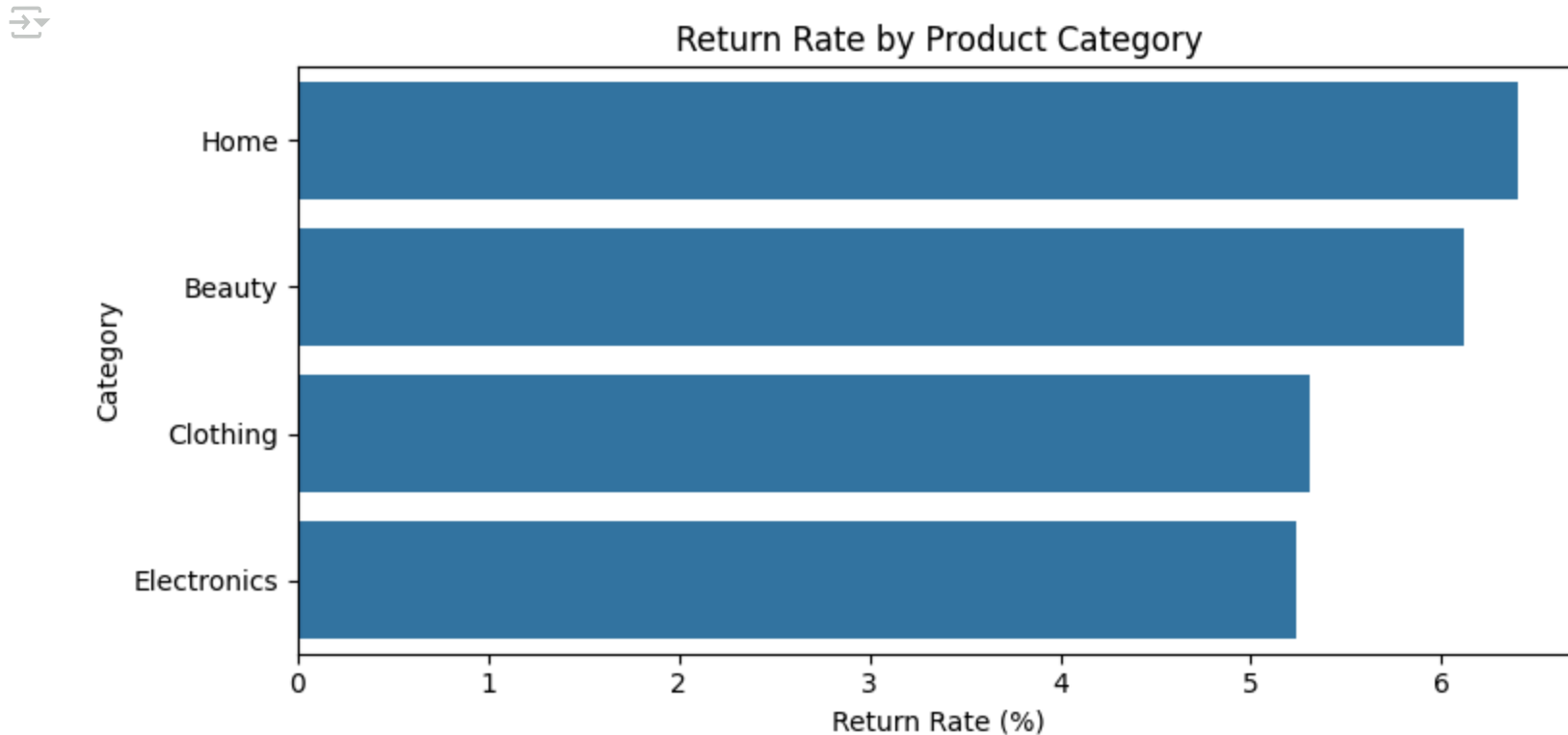
```
# Plot Monthly Return Trend
plt.figure(figsize=(10, 5))
sns.lineplot(x=monthly['month'].astype(str), y='return_rate', data=monthly, marker='o')
plt.title("Monthly Return Rate Trend")
plt.xticks(rotation=45)
plt.ylabel("Return Rate (%)")
plt.tight_layout()
plt.show()
```



```
#  3. Return Rate by Category
category_returns = df.groupby('category').agg({
    'quantity': 'sum',
    'return_flag': 'sum'
}).reset_index()
category_returns['return_rate'] = 100 * category_returns['return_flag'] / category_returns['quar
```

```
# Plot
plt.figure(figsize=(8, 4))
```

```
sns.barplot(data=category_returns.sort_values('return_rate', ascending=False), x='return_rate',
plt.title("Return Rate by Product Category")
plt.xlabel("Return Rate (%)")
plt.ylabel("Category")
plt.tight_layout()
plt.show()
```



```
# 4. Return Rate by Loyalty
loyalty = df.groupby('loyalty_flag').agg({
    'quantity': 'sum',
    'return_flag': 'sum'
}).reset_index()
loyalty['return_rate'] = 100 * loyalty['return_flag'] / loyalty['quantity']
print(loyalty)
```

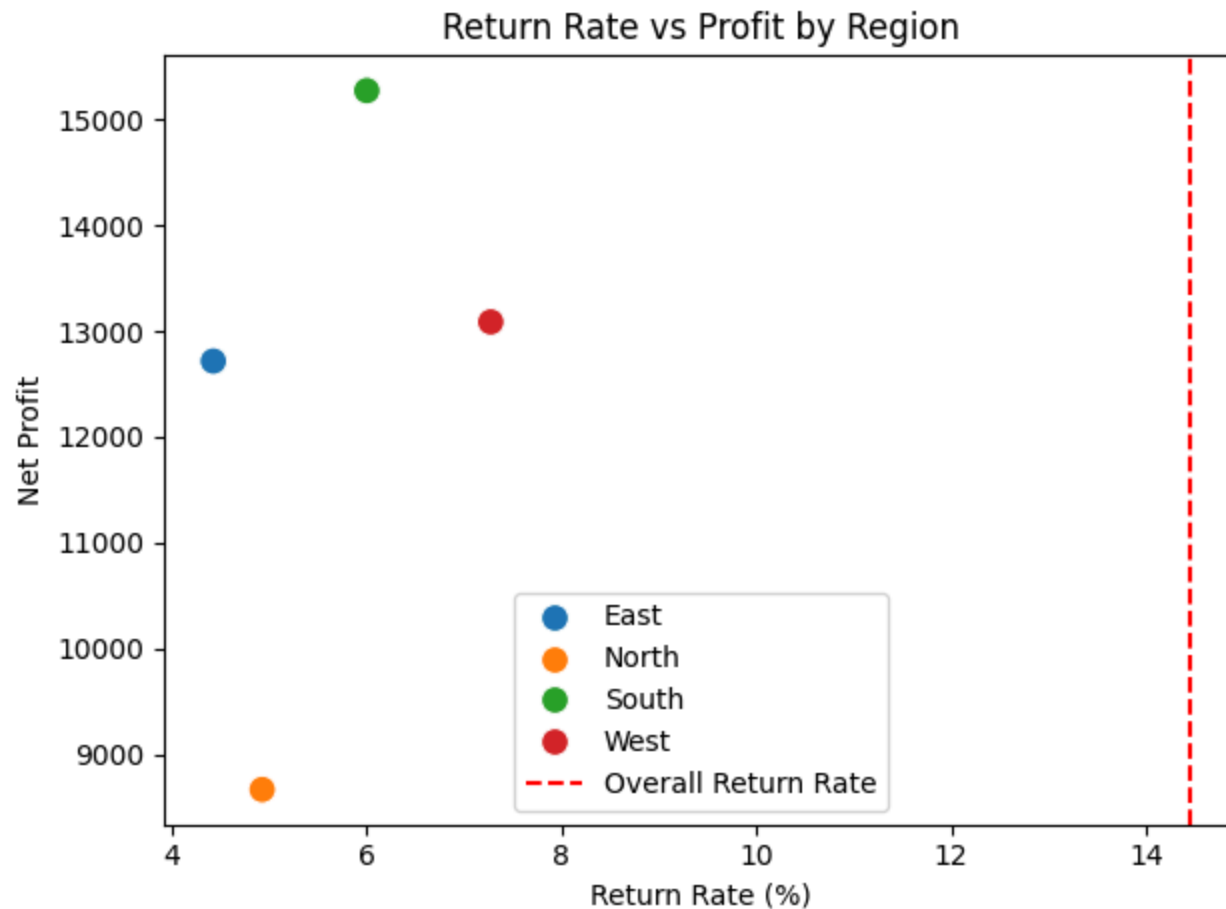
```
⇒
```

	loyalty_flag	quantity	return_flag	return_rate
0	0	636	33	5.188679
1	1	368	25	6.793478

# 🕒 5. Region-Wise Net Profit


```
region_perf = df.groupby('region').agg({
    'profit': 'sum',
    'return_flag': 'sum',
    'quantity': 'sum'
}).reset_index()
region_perf['return_rate'] = 100 * region_perf['return_flag'] / region_perf['quantity']
```

```
sns.scatterplot(data=region_perf, x='return_rate', y='profit', hue='region', s=100)
plt.title("Return Rate vs Profit by Region")
plt.xlabel("Return Rate (%)")
plt.ylabel("Net Profit")
plt.axvline(overall_return_rate, color='red', linestyle='--', label='Overall Return Rate')
plt.legend()
plt.tight_layout()
plt.show()
```



```
# 📌 6. High-Return SKUs
sku_returns = df.groupby('sku').agg({
    'quantity': 'sum',
    'return_flag': 'sum',
    'profit': 'sum'
}).reset_index()
sku_returns['return_rate'] = 100 * sku_returns['return_flag'] / sku_returns['quantity']
sku_top = sku_returns.sort_values('return_rate', ascending=False).head(10)
```

```
print(sku_top[['sku', 'return_rate', 'profit']])
```



	sku	return_rate	profit
12	2013	28.571429	289.85
24	2025	18.750000	242.81
5	2006	13.636364	6.10
34	2035	13.333333	1140.53
39	2040	13.333333	699.63
7	2008	11.764706	2330.30
26	2027	11.111111	329.56
22	2023	9.523810	321.83
29	2030	9.090909	288.08
1	2002	8.695652	2405.80

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.