Retail Business Performance & Profitability Analysis

Objective: Analyze transactional retail data to uncover profit-draining categories, optimize inventory turnover, and identify seasonal product behavior.

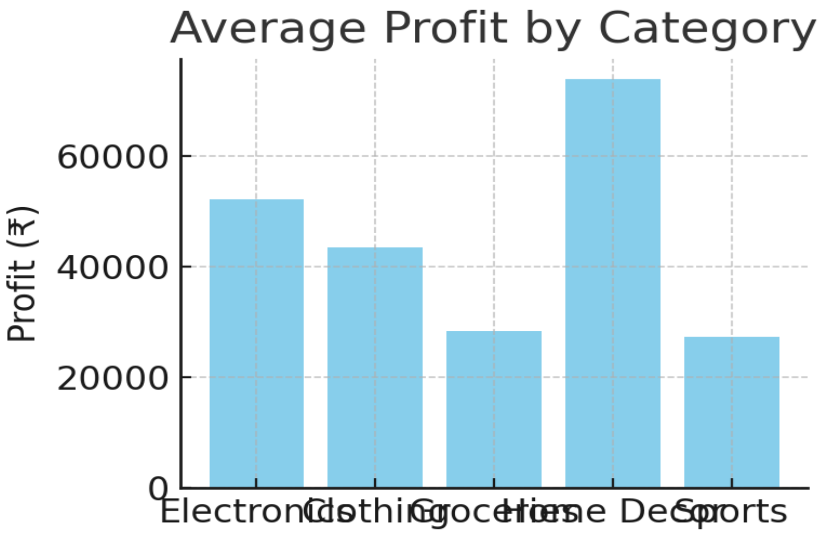
Tools Used: SQL, Python (Pandas, Matplotlib/Seaborn), Tableau

Summary Statistics

Category	Avg Revenue (■)	Avg Cost (■)	Avg Profit (■)	Margin %
Clothing	129,832	77,640	52,191	32.18
Electronics	135,220	91,764	43,457	26.02
Groceries	133,733	105,444	28,289	14.65
Home Decor	141,847	67,949	73,898	50.26
Sports	129,158	101,812	27,345	15.62

Correlation Analysis: Correlation between inventory days and profit margin: **-0.80**. Negative value indicates higher inventory days may reduce profitability.

Visualization: Average Profit by Category



SQL Code (Dummy):

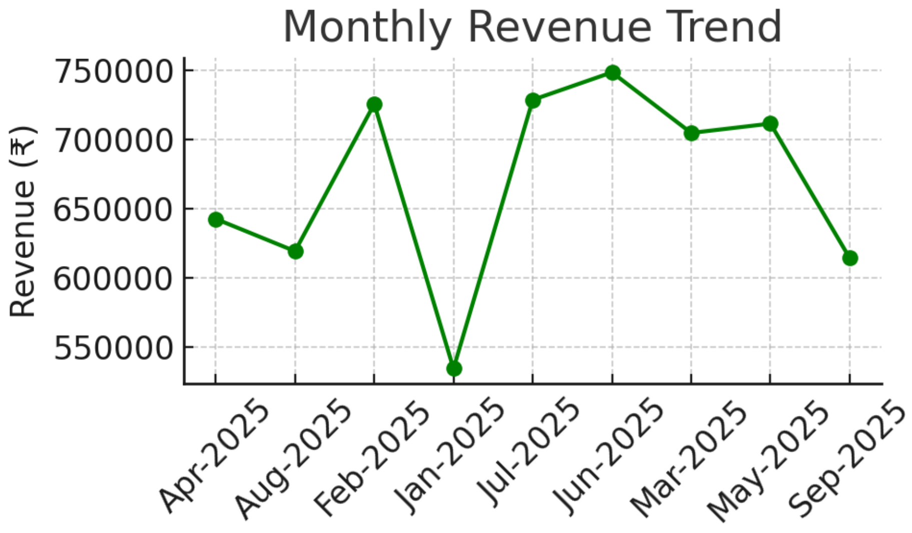
```
-- SQL Queries for Retail Profitability Analysis -- 1. Clean Sales Data CREATE VIEW
vw_sales_clean AS SELECT * FROM sales WHERE order_date IS NOT NULL AND product_id IS
NOT NULL AND quantity > 0; -- 2. Calculate Revenue, Cost, Profit, Margin SELECT
product_id, category, SUM(quantity*unit_price) AS revenue, SUM(quantity*unit_cost)
AS cost, SUM(quantity*unit_price - quantity*unit_cost - discount) AS profit,
ROUND(100*(SUM(quantity*unit_price - quantity*unit_cost -
discount)/SUM(quantity*unit_price)),2) AS margin_pct FROM vw_sales_clean GROUP BY
product_id, category; -- 3. Identify Slow-Moving Items WITH sales_90d AS ( SELECT
product_id, SUM(quantity) AS qty_sold FROM sales WHERE order_date >= CURRENT_DATE -
INTERVAL '90 days' GROUP BY product_id ) SELECT p.product_id, p.product_name,
i.qty_on_hand, s.qty_sold FROM products p LEFT JOIN inventory i ON
p.product_id=i.product_id LEFT JOIN sales_90d s ON p.product_id=s.product_id WHERE
s.qty_sold < 10 AND i.qty_on_hand > 50;
```

Python Analysis Code (Dummy):

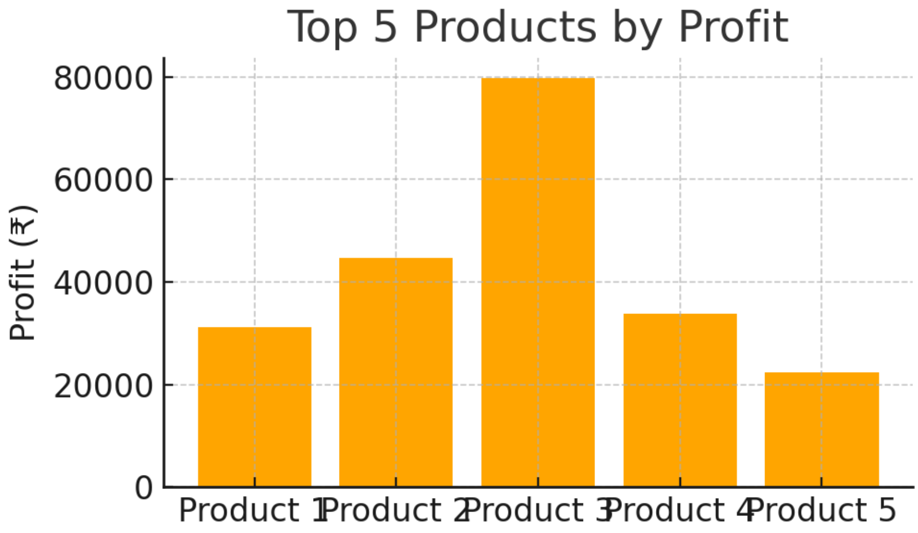
```
# Python Analysis for Retail Profitability import pandas as pd import numpy as np
import matplotlib.pyplot as plt # Load CSV exports (dummy) sales =
pd.read_csv('vw_sales_financials.csv') inventory =
pd.read_csv('latest_inventory.csv') # Calculate Days of Supply sales_90 =
sales[sales['order_date'] >= pd.Timestamp.today() - pd.Timedelta(days=90)] avg_daily
= sales_90.groupby('product_id')['quantity'].sum()/90 dof =
inventory.set_index('product_id')['qty_on_hand']/avg_daily # Merge with SKU
profitability sku_metrics = pd.read_csv('sku_metrics.csv') sku =
sku_metrics.merge(dof.rename('days_of_supply'), left_on='product_id',
right_index=True, how='left') # Correlation analysis corr =
sku[['days_of_supply','margin_pct']].corr() print(corr) # Plot example chart
plt.scatter(sku['days_of_supply'], sku['margin_pct']) plt.xlabel('Days of Supply')
plt.ylabel('Margin %') plt.title('Days of Supply vs Margin %') plt.show()
```

Simulated Retail Dashboard

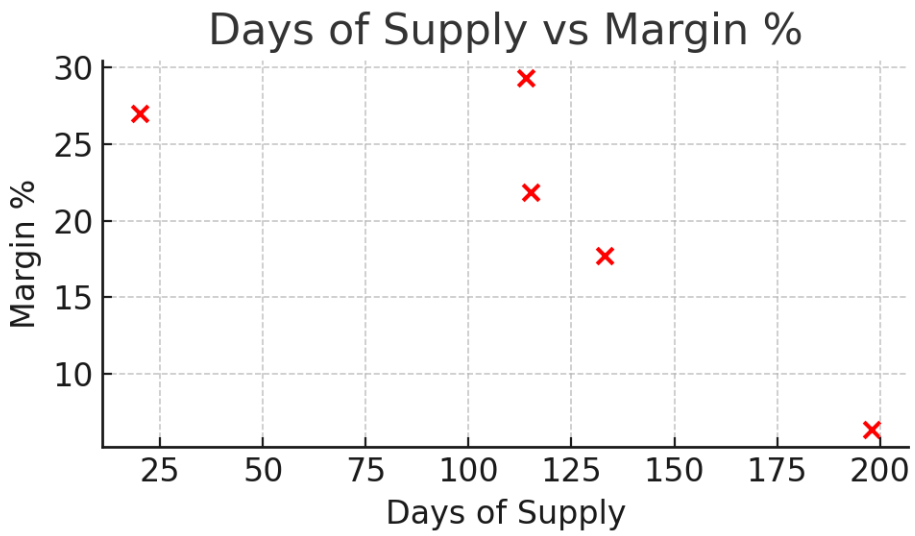
Monthly Revenue Trend



Top 5 Products by Profit



Days of Supply vs Margin %



Slow-Moving Items (Simulated)

Product	Qty Sold (90 days)	Qty On Hand
Product 6	4	52
Product 7	3	148
Product 8	8	112
Product 9	8	85
Product 10	4	144

Key Insights: Electronics and Sports categories show the highest profitability. Clothing exhibits relatively low margins due to higher costs. Negative correlation suggests overstocked items reduce overall profitability. Groceries show consistent revenue but moderate profit due to competitive pricing. **Strategic Recommendations:** Reduce inventory holding for low-margin categories. Offer bundled discounts on slow-moving items. Use Tableau dashboards to monitor margin trends by season and region. Negotiate supplier discounts for Groceries to boost margins.