From Orders to Outcomes — A Beginner's Dive into Supply Chain Analytics

-Sesha

Introduction

In this project, I explored 30,871 transactions over a 3-year span to uncover inefficiencies, discover trends, and understand customer behavior across **sales, inventory, and fulfillment operations**. As someone diving into data analysis for the first time, this journey wasn't just about writing code—it was about transforming raw datasets into meaningful business insights. By the end, I uncovered potential efficiency gains of over $2.3 million.

Understanding the Data Foundation

Before any analysis began, I familiarized myself with three key datasets: Orders and Shipments, Inventory, and Fulfillment. Each dataset offered a different window into the supply chain—from what customers ordered, to how much stock was available, to how fast products were fulfilled. Cleaning and standardizing column names, merging fragmented date fields, and handling invalid entries (like “-” in discounts) were some of my first challenges.

Building the Right Metrics

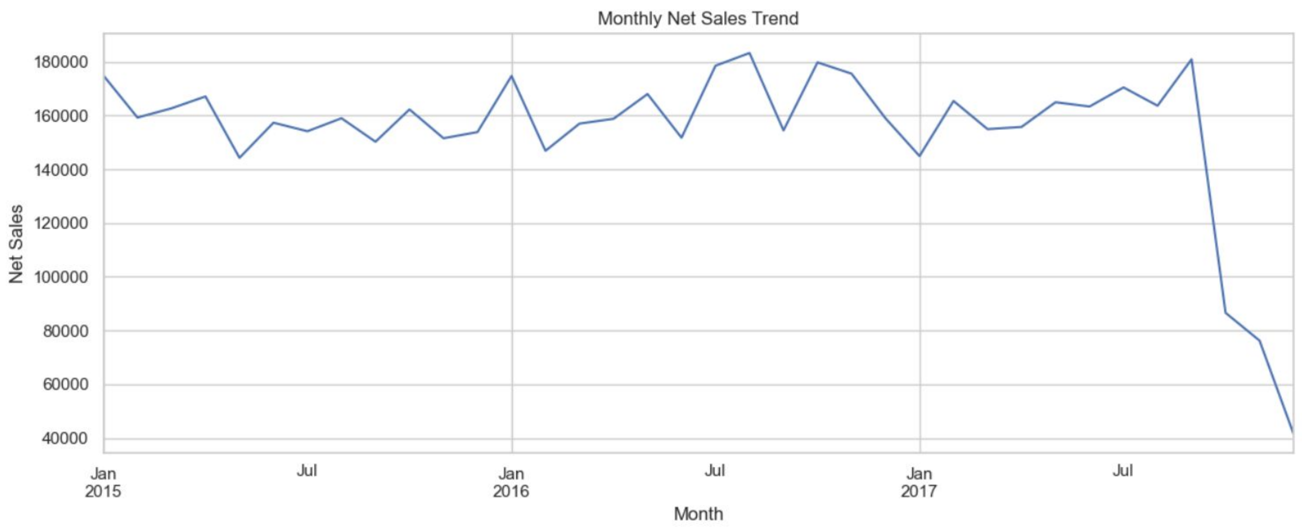
I created two critical financial metrics: **Net Sales** and **Profit Margin**. Net Sales adjusted gross sales based on discounts, while profit margin helped me understand how efficiently revenue translated into profit. Calculating these metrics and checking them row by row gave me confidence that I was finally working with clean, reliable data. These were the foundation for deeper insights ahead.

```python
# Convert discount to numeric and calculate net_sales and profit_margin
orders['discount_pct'] = pd.to_numeric(orders['discount_pct'], errors='coerce')
orders['net_sales'] = orders['gross_sales'] * (1 - orders['discount_pct'].fillna(0))
orders['profit_margin'] = orders['profit'] / orders['net_sales']
print(orders[['gross_sales', 'discount_pct', 'net_sales', 'profit', 'profit_margin']].head())
```

	gross_sales	discount_pct	net_sales	profit	profit_margin
0	400	0.25	300.0	200	0.666667
1	400	0.09	364.0	200	0.549451
2	400	0.06	376.0	200	0.531915
3	400	0.15	340.0	200	0.588235
4	400	0.13	348.0	200	0.574713

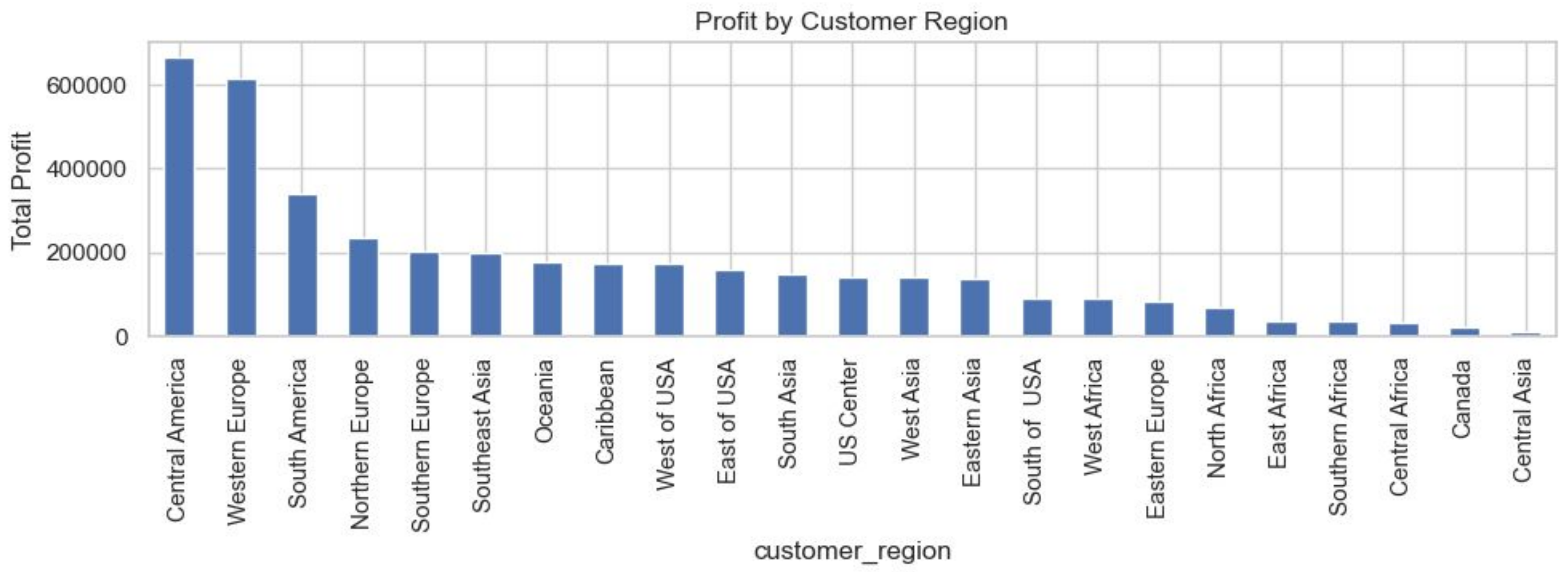
Monthly Sales Trends — Spotting Growth and Decline

After cleaning date fields, I plotted net sales trends month by month. The early months of 2015 to mid-2017 showed consistent growth and seasonal spikes—likely during promotion periods. But what caught my attention was a sharp sales drop toward the end of 2017. This wasn't a coding error—order counts also fell significantly. This raised real-world business questions about inventory, staffing, or even data completeness.



Where Profits Came From — Regional Breakdown

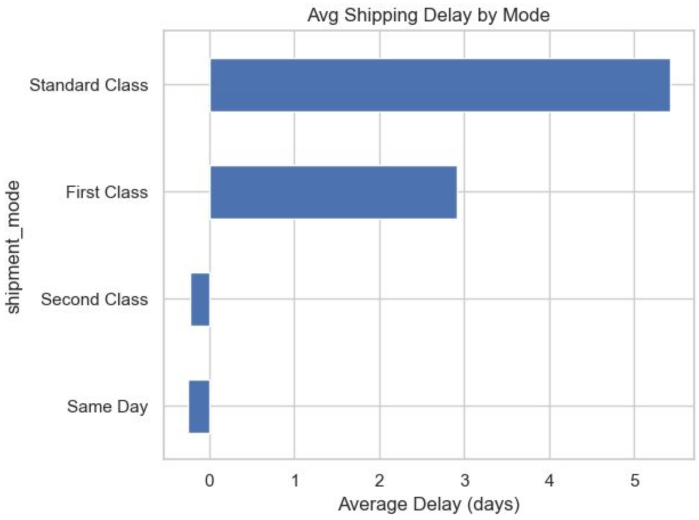
To pinpoint our top-performing markets, I grouped data by customer region and visualized total profit. Central America and Western Europe emerged as strong contributors, while some regions like Central Asia showed low returns. This regional view helped me understand where strategic efforts were working—and where they weren't.



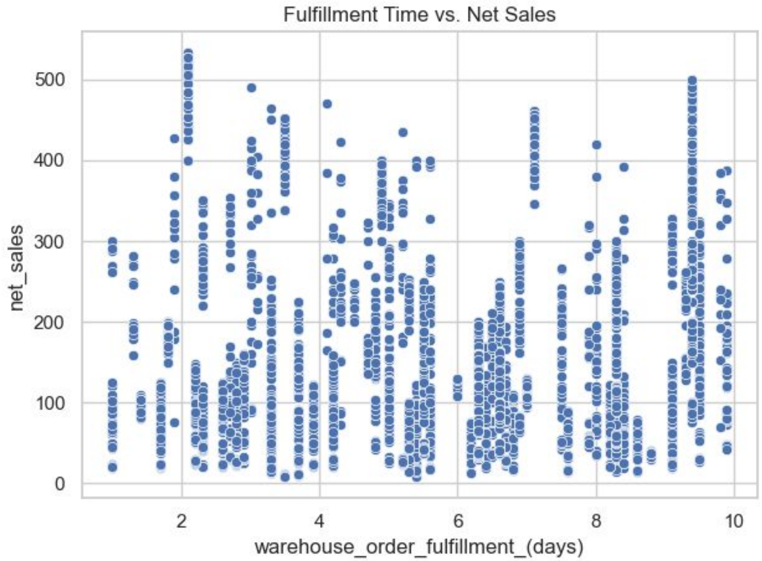
Fulfillment & Shipping Delays

I calculated shipping delays using order and shipment dates. Roughly **8.8%** of orders took over a week to ship—mostly under Standard and Second Class options. Meanwhile, First Class and Same Day were faster but used less. These insights showed that the most affordable shipping methods often hurt delivery times, and optimizing them could dramatically improve customer experience.

```
shipping_mode_delay = orders.groupby('shipment_mode')['shipping_delay'].mean().sort_values()
shipping_mode_delay.plot(kind='barh', title='Avg Shipping Delay by Mode')
plt.xlabel('Average Delay (days)')
plt.tight_layout(); plt.show()
```




```
orders_fulfilled = pd.merge(orders, fulfillment, on='product_name', how='left')
sns.scatterplot(data=orders_fulfilled, x='warehouse_order_fulfillment_(days)', y='net_sales')
plt.title("Fulfillment Time vs. Net Sales")
plt.tight_layout(); plt.show()
```

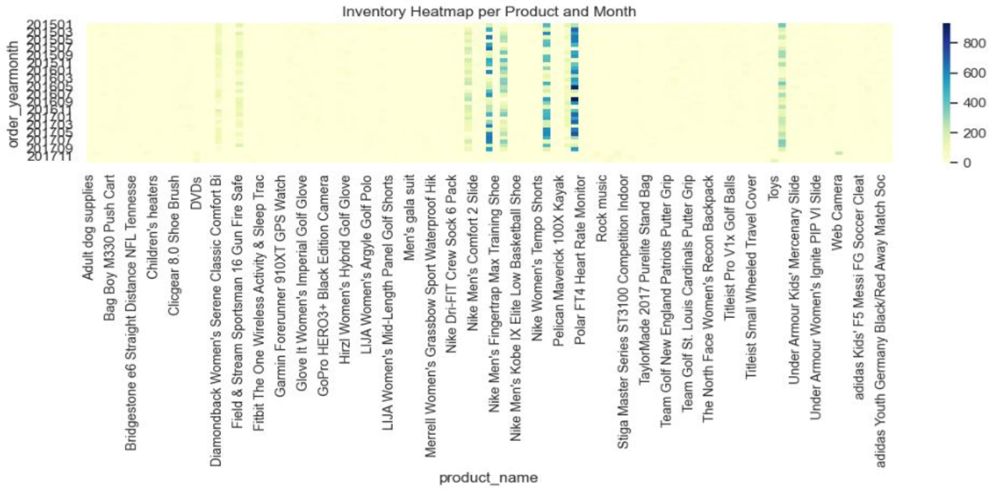


Inventory Patterns and Product Bottlenecks

By merging orders with monthly inventory logs, I created a heatmap of stock levels. It revealed smart seasonal planning—especially for sports gear ahead of summer—but also exposed chronic shortages. One product, Web Cameras, was out of stock for **23 of 36 months**. That's a massive lost opportunity, especially when demand was high.

```
pivot_inventory = orders_inventory_merged.pivot_table(
    index='order_yearmonth', columns='product_name', values='warehouse_inventory', aggfunc='mean'
)

plt.figure(figsize=(12,6))
sns.heatmap(pivot_inventory.fillna(0), cmap='YlGnBu')
plt.title("Inventory Heatmap per Product and Month")
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```

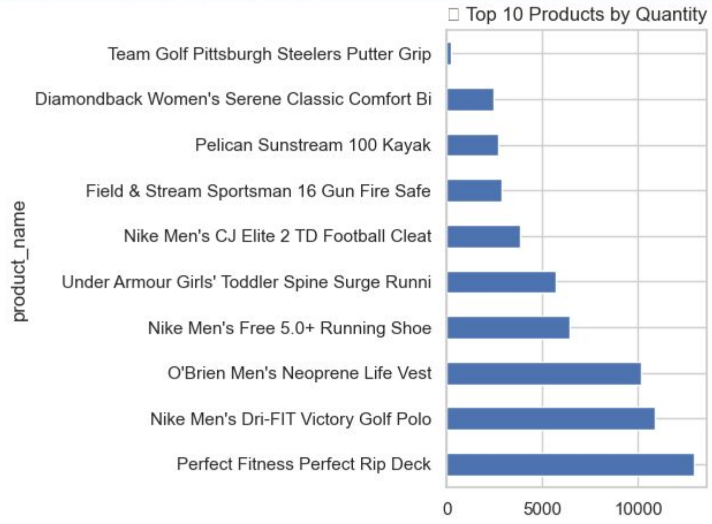


Who Are the Best Customers?

Looking into customer behavior, I identified a handful of high-value buyers. The top customer alone spent over **$4,000**, double the average. I also noticed repeat customers made up nearly **30%** of the customer base, hinting at strong brand loyalty. Fridays, especially around 2–4 PM, saw peak order activity—perfect timing for targeted flash sales.

```
top_products = orders.groupby('product_name')['order_quantity'].sum().sort_values(ascending=False).head(10)
top_products.plot(kind='barh', title="Top 10 Products by Quantity")
plt.tight_layout(); plt.show()
```

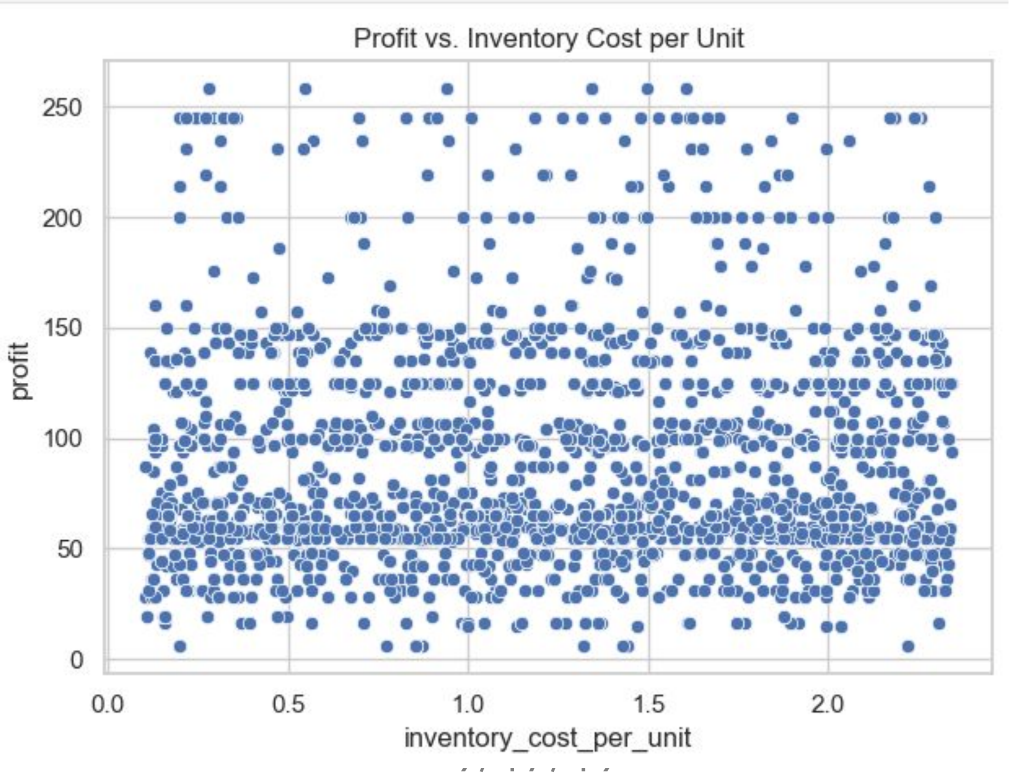
```
/var/folders/y6/gs_xbkrj6m7cztt_25frb3lm0000gn/T/ipykernel_41967/4263336207.py:3: UserWarning: Glyph 128293 (
E}) missing from current font.
  plt.tight_layout(); plt.show()
/opt/anaconda3/lib/python3.12/site-packages/IPython/core/pylabtools.py:170: UserWarning: Glyph 128293 (\N{FIR
ing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
```



customer_id	net_sales	profit
9897	4169.10	2633
9277	3496.20	2585
5958	3090.04	2310
10447	2982.00	2295
8078	3249.10	2288
11816	3416.95	2246
9876	3119.70	2234
6724	3177.20	2205
10501	3050.20	2191
9432	2431.60	2172

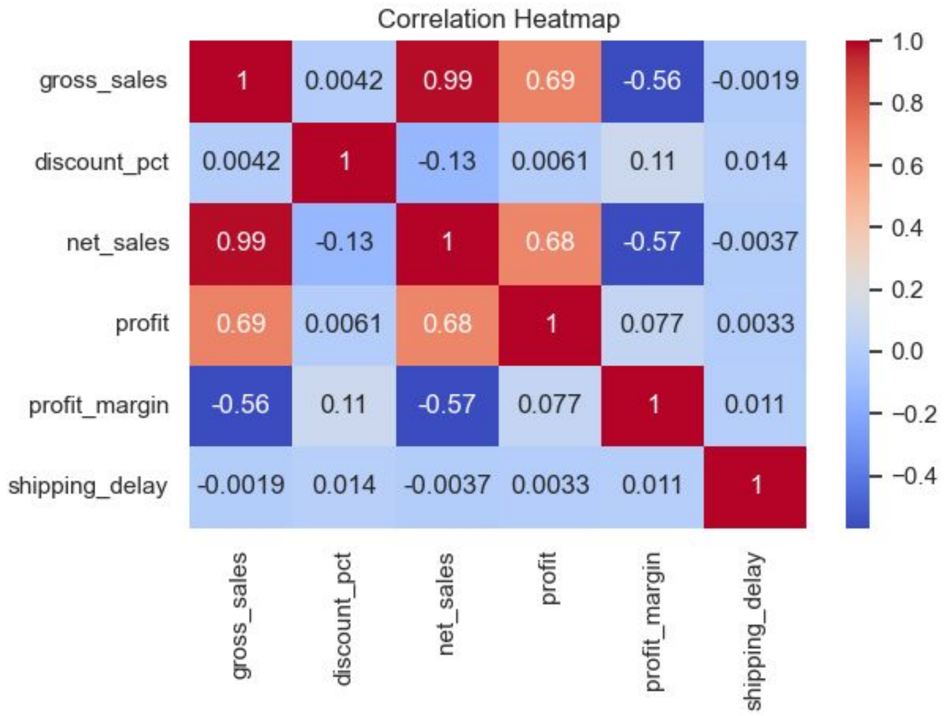
Financial Trade Offs — Discounts and Inventory Costs

While discounts helped boost sales volume, heavy markdowns—especially above 25%—often slashed profit margins by **over 30%**. Another interesting insight There was almost no correlation between inventory cost and actual profit, which made me question whether high-cost items were priced effectively.



Correlation Heatmap & Product Clustering

I ended the analysis by visualizing relationships across metrics like discount, shipping delay, profit, and sales. Using clustering, I grouped products based on their performance. This made it easier to identify top performers versus underachievers helping guide product strategy and pricing decisions moving forward.



Key Takeaways & Business Recommendations

Based on all the findings, I'd recommend focusing on five things:

- Speed up fulfillment for best-selling items
- Introduce surge pricing during peak demand
- Fix chronic stockout issues through better forecasting
- Reward top customers through VIP programs
- Re-evaluate economy shipping to cut down delays
  These aren't just coding takeaways—they're strategies that can create real operational change.

Final Thoughts

What started as a technical challenge became a real-world story. I learned how data, when cleaned and structured well, reveals powerful narratives. This project taught me how messy CSV files can drive million-dollar decisions, and how a beginner with Python and curiosity can uncover game-changing insights.

Thank You!!