Data Visualization Design

Decoding Sales Success

Technical Report

Team 5 - Data Nova

Satyam Kumar (21F2000243)
Harmanpreet Singh (21F3002026)
Aniket Agrawal (21F1000715)
Ameya Raj (21F1003202)

Amazon Sales Dashboard Report

This report summarizes the analysis of Amazon sales data visualized through an interactive Dash dashboard. The dashboard provides key performance indicators and insights into sales trends, product performance, customer behavior, and the impact of promotions.

1. Approach

This section outlines the methodology implemented in the Dash dashboard to analyze the Amazon sales data.

- **Data Source:** The data is loaded from a CSV file named 'content/Amazon Sale Report.csv'.
- **Data Loading and Preprocessing (as per load_and_preprocess_data()):**
  - The CSV file is read into a Pandas DataFrame.
  - The 'Order_Date' column is converted to datetime objects.
  - A separate DataFrame non_cancelled_orders is created, excluding orders with the status 'Cancelled', for primary sales analysis.
  - New columns are created for 'Week', 'Month', and 'Day' based on the 'Order_Date'.
- **Dashboard Framework:** The analysis is presented through an interactive Dash application utilizing Bootstrap CSS for styling.
- **Key Visualizations and Components:** The dashboard includes the following key components to present the analysis:
  - **KPI Cards:** Displaying aggregated metrics like Total Orders, Total Revenue, Cancellation Rate, and Average Order Value.
  - **Date Range Picker:** Allowing users to filter the data by a specific date range.
  - **Category Dropdown:** Enabling users to filter the data by specific product categories.
  - **Daily Sales Chart:** A line chart showing the trend of daily sales and a 7-day moving average.
  - **Top Product Categories by Sales Chart:** A bar chart highlighting the top 10 product categories based on sales amount.
  - **Sales by Fulfillment Type Pie Chart:** A pie chart illustrating the distribution of sales across different fulfillment types.
  - **Top 10 States by Sales Chart (Bar Chart Implementation):** A bar chart showing the top 10 shipping states by total sales.

    - **Heatmap of Order Counts:** A heatmap showing the number of orders for each product category within the top N shipping states.
    - **Stacked Bar Chart of Orders by Category per State:** A stacked bar chart visualizing the distribution of product category orders within the top N shipping states.
    - **Cancelled vs Not Cancelled Orders by Category (Volume) Bar Chart:** A stacked horizontal bar chart showing the number of cancelled and non-cancelled orders for each product category.
    - **Top N States by Order Cancellation (Bubble Chart with %):** A bubble chart highlighting the top N shipping states based on the number and percentage of cancelled orders.
    - **Impact of Promotions on Sales Bar Charts:** Two bar charts comparing the total sales and average order value for orders with and without promotion IDs.
- **Interactivity:** The dashboard allows users to dynamically filter the data by date range and product category, updating all visualizations and KPIs accordingly.

2. Analytical Techniques

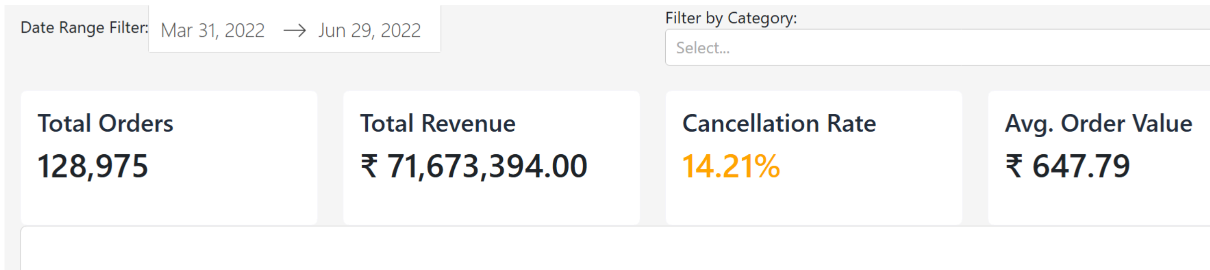
The Dash dashboard employs the following analytical techniques through its visualizations:

- **Time Series Analysis:** The Daily Sales Chart visualizes sales trends over time, including a moving average to identify underlying patterns.
- **Categorical Analysis:** The Top Product Categories by Sales Chart and Sales by Fulfillment Type Pie Chart compare sales across different product categories and fulfillment methods.
- **Geographic Analysis:** The Top 10 States by Sales Chart and the Heatmap/Stacked Bar Chart provide insights into sales distribution across different shipping states. The Bubble Chart specifically analyzes cancellation rates by state.
- **Segmentation Analysis:** The B2B vs B2C Comparison Pie Charts segment sales based on the 'Business_to_Business' flag. The Cancelled vs Not Cancelled Orders by Category chart segments orders by their status within each product category.
- **Promotional Impact Analysis:** The Impact of Promotions on Sales Bar Charts compare sales metrics for orders with and without promotion IDs to assess the effectiveness of promotions.
- **Descriptive Statistics:** The KPI cards present key aggregated metrics providing a high-level overview of the sales data.
- **Filtering and Aggregation:** The dashboard leverages Pandas' `groupby()` and aggregation functions to calculate the data required for each visualization, dynamically updated based on user-selected filters.
- **Visualization:** Plotly Express and Plotly Graph Objects are used to create interactive and informative charts and graphs, making it easier to identify patterns and trends.

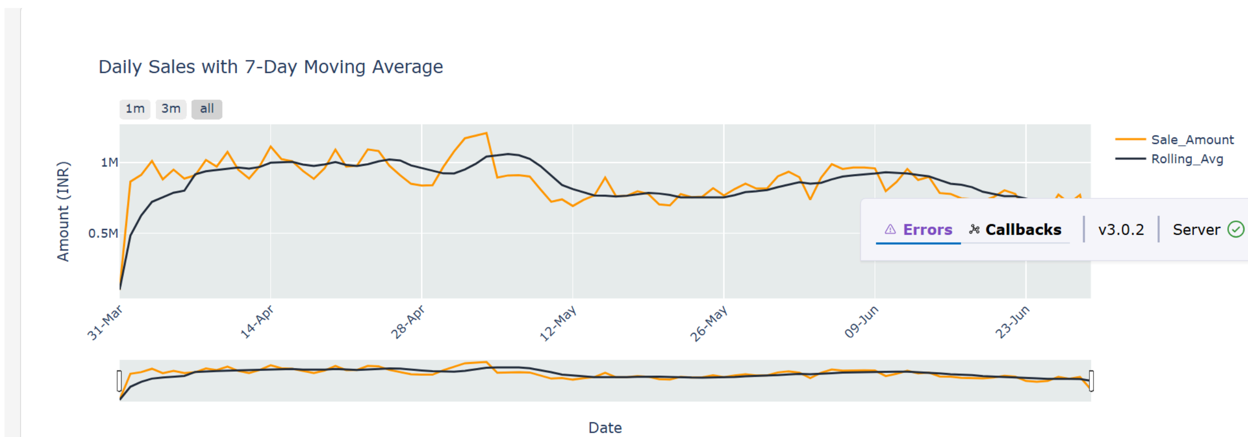
3. Observations and Findings (Initial - Requires Running the Dashboard with Data)

This section would typically contain specific observations and findings derived from interacting with the dashboard using the actual 'content/Amazon Sale Report.csv' data.

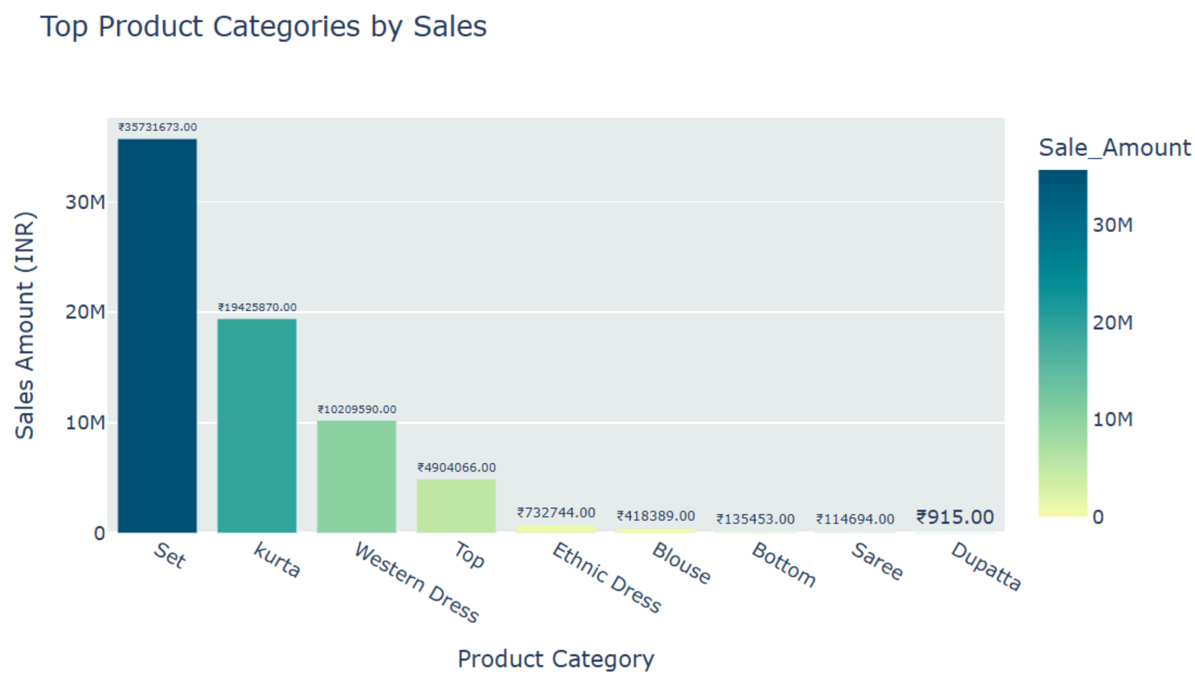
- **Overall Performance:** The KPI cards will provide a snapshot of total orders, revenue, cancellation rates, and average order value for the selected time period and category.



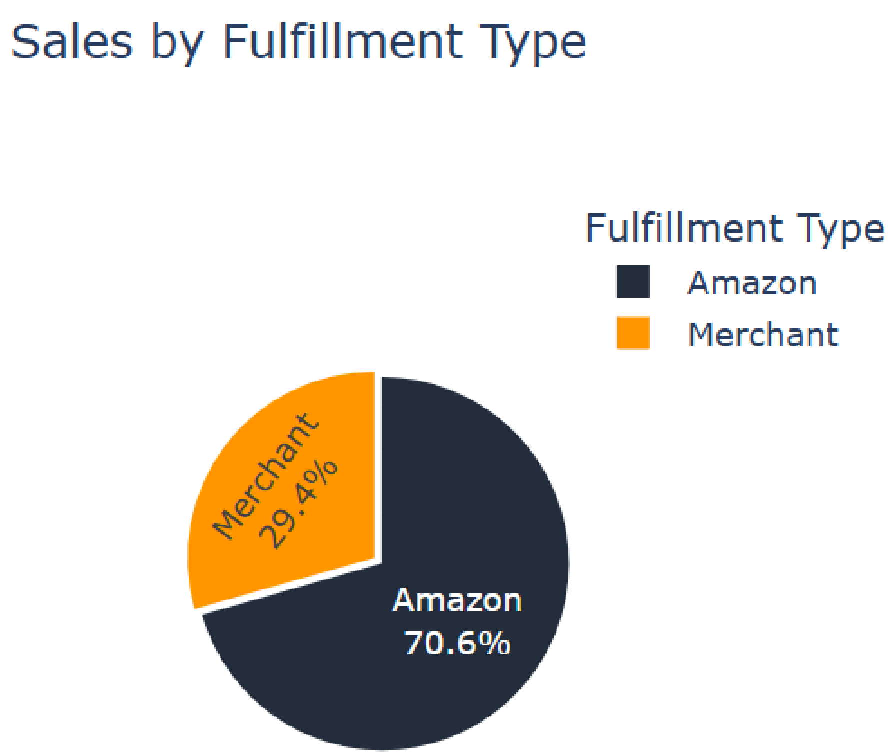
- **Sales Trends:** The Daily Sales Chart will show the daily sales fluctuations and highlight any significant upward or downward trends, as well as the smoothed trend via the moving average.



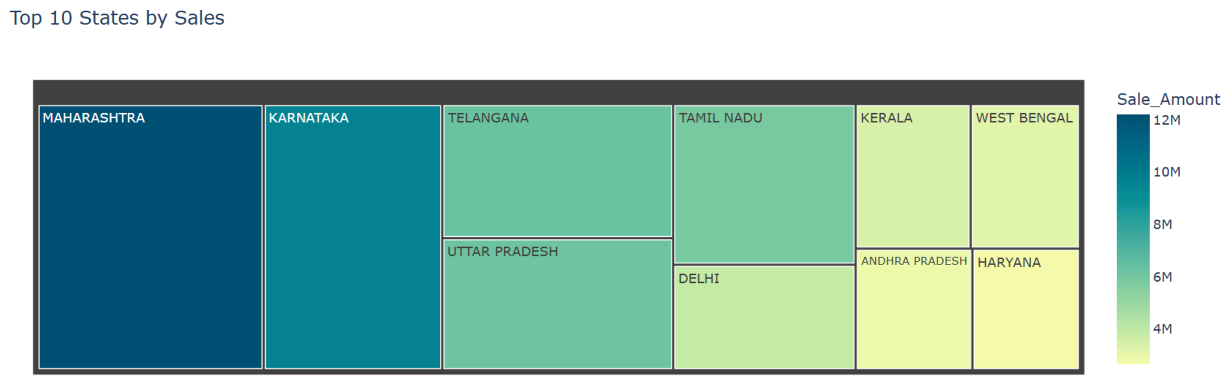
- **Top Performing Products:** The Top Product Categories by Sales Chart will identify the most significant contributors to overall revenue.



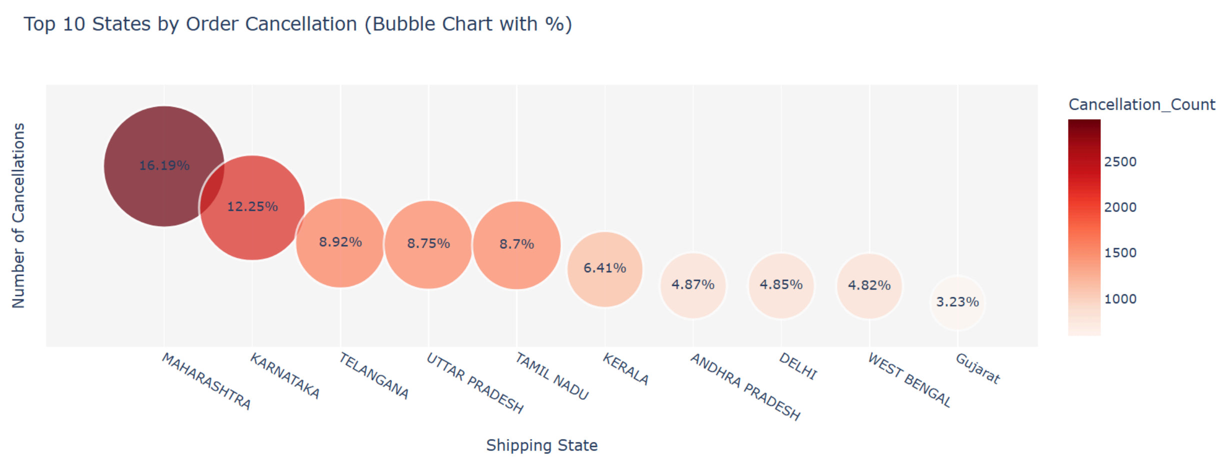
- **Fulfillment Efficiency:** The Sales by Fulfillment Type Pie Chart will show the preferred or most revenue-generating fulfillment methods.



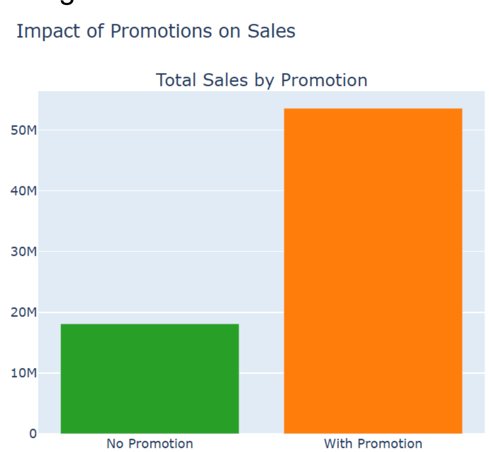
- **Key Geographic Markets:** The Top 10 States by Sales Chart and the Heatmap/Stacked Bar Chart will pinpoint the states with the highest sales volume and the popular product categories within those states.



- **Cancellation Patterns:** The Cancelled vs Not Cancelled Orders by Category Chart will reveal which product categories have the highest number of cancellations. The Top N States by Order Cancellation Chart will identify states with the most cancellations.



- **Promotion Effectiveness:** The Impact of Promotions Chart will provide an initial indication of whether orders with promotion IDs tend to have higher total sales or average order values.



-

4. Recommendations (Based on Potential Insights)

Based on the types of insights the dashboard is designed to provide, here are potential recommendations that could arise from analyzing the data:

- **Focus on Top Categories:** Invest more in marketing and inventory for the top-selling product categories identified in the "Top Product Categories by Sales" chart.
- **Address High Cancellation Rates:** Investigate the reasons for high cancellation rates in specific product categories identified in the "Cancellation Analysis" chart and implement strategies to reduce them (e.g., improved product descriptions, better quality control, more accurate shipping estimates).
- **Optimize Fulfillment Strategy:** Analyze the "Sales by Fulfillment Type" chart to understand which fulfillment methods are most popular or generate the most revenue and optimize accordingly.
- **Target Key Geographic Areas:** Focus marketing efforts on the top-performing states identified in the "Geographic Sales Map" and the Heatmap. Consider localized promotions or product offerings.
- **Evaluate Promotion Effectiveness:** Analyze the "Promotion Impact" charts to determine which types of promotions are most effective in driving sales and increasing average order value. Refine future promotional strategies based on these insights.
- **Monitor Daily Sales Trends:** Regularly monitor the "Daily Sales Chart" to identify short-term fluctuations and react to any negative trends promptly.
- **Further Investigation:** Use the filtering capabilities of the dashboard to drill down into specific date ranges or product categories that show interesting patterns.

5. Constraints

This section identifies potential limitations and constraints related to the dashboard and the underlying data:

- **Data Quality:** The accuracy and completeness of the 'content/Amazon Sale Report.csv' data directly impact the reliability of the analysis. Issues like missing values or incorrect data entries could lead to skewed results.
- **Static Data:** The dashboard relies on a static CSV file. The insights are limited to the data present in this file and won't reflect real-time sales updates.
- **Limited Customer Information:** The provided code doesn't suggest the inclusion of detailed customer demographics or purchase history, which would limit the depth of customer segmentation analysis.

- **No Cost Data:** The absence of cost data prevents the calculation of profit margins and profitability analysis.
- **Geographic Granularity:** The geographic analysis is limited to the 'Shipping_State'. More granular data (e.g., city, zip code) could provide more localized insights.
- **Assumptions in Calculations:** The KPI calculations (e.g., cancellation rate) assume that all relevant order statuses are correctly categorized in the data.
- **Dashboard Performance:** With very large datasets, the performance of the Dash dashboard (loading times, responsiveness to filters) could become a constraint.
- **External Factors Not Included:** The analysis doesn't inherently account for external factors like competitor activities, economic conditions, or marketing campaigns outside of those identifiable by 'Promotion_IDs'.
- **Simplistic Geographic Representation:** The "Geographic Sales Map" is implemented as a bar chart of states due to the lack of GeoJSON data for India in the provided code. A true geographic map visualization would require this additional data.

Profit & Loss Analysis

Link to Dashboard: P&L Dashboard

1. Approach

The analysis was conducted using Python in Google Colab, leveraging libraries such as Pandas, NumPy, Matplotlib, and Seaborn for data manipulation, analysis, and ultimately Power BI for visualizations.

Data Cleaning and Preparation:

The initial steps involved cleaning and preparing the data. This included handling missing values, converting data types, and creating new features for analysis. The 'Fee' (likely due to platform charges) column was created to represent the difference between 'Final_Old_MRP' and 'Old_MRP'.

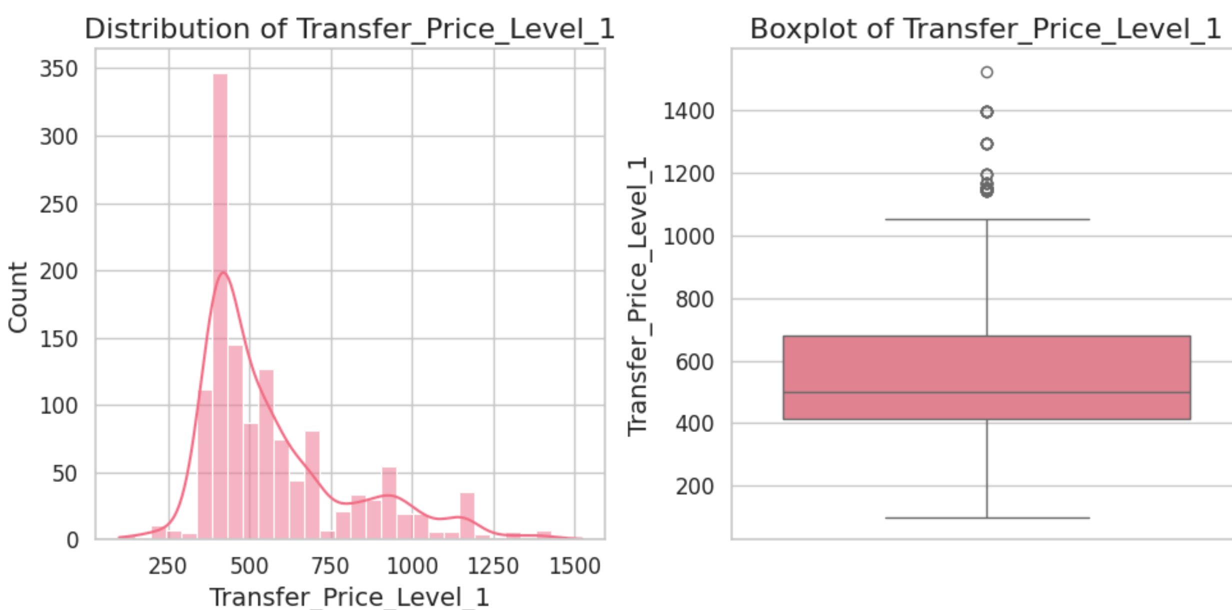
```
df['Fee'] = df['Final_Old_MRP'] - df['Old_MRP']

df.loc[df['Fee'] < 0, 'Fee'] = 0
df['Fee'].value_counts()
```

Profit columns were calculated for each marketplace by subtracting the sum of 'Transfer_Price_Level_1' and 'Fee' from the respective marketplace's MRP. Profit percentages were calculated by dividing profit by the marketplace's MRP. All through a custom function that yields a "profit_df"

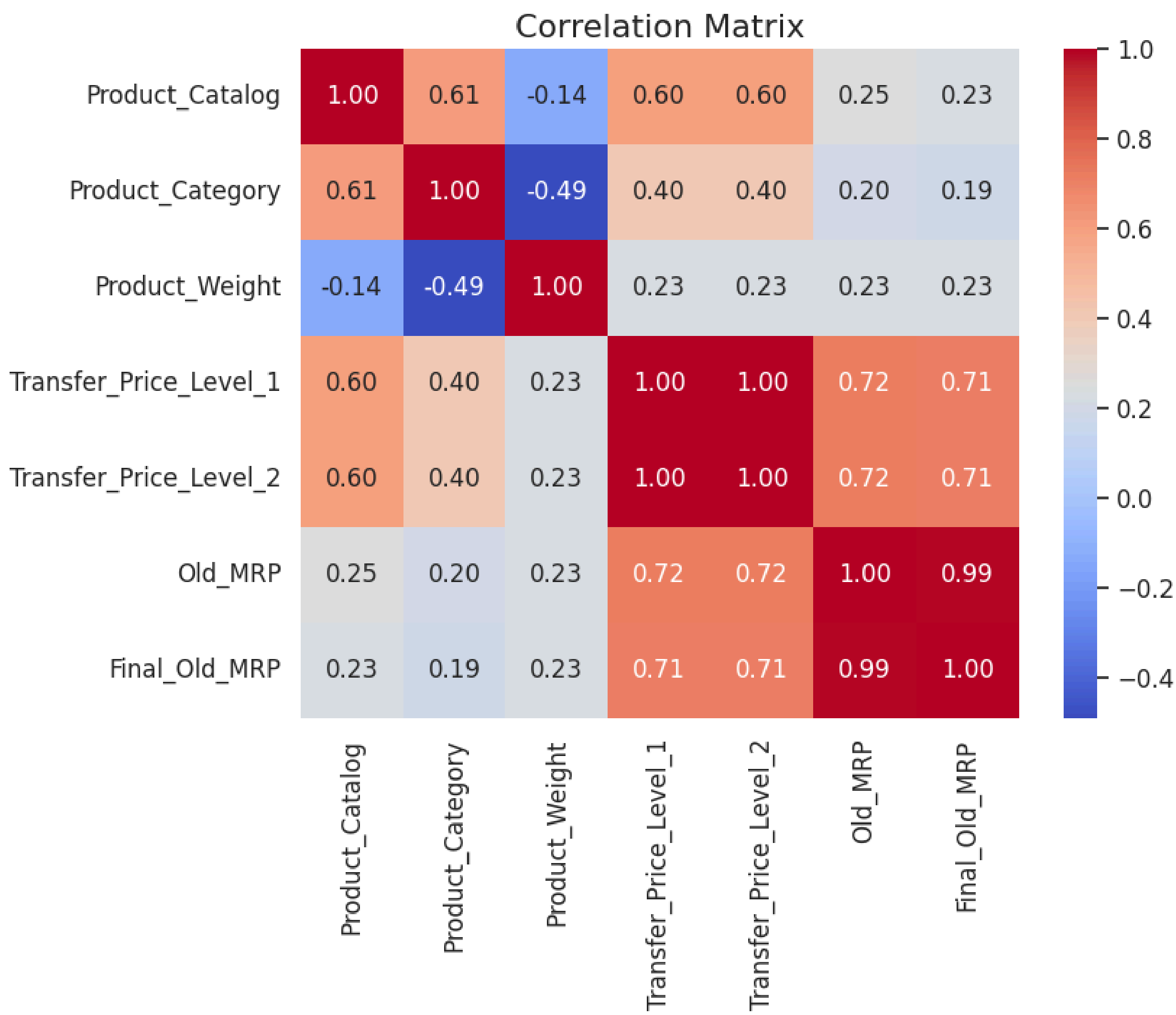
```
def calculate_profit_per_marketplace(df, marketplace_cols):
    profit_df = df.copy()
    profit_cols, profit_P_cols = [], []
    for col in marketplace_cols:
        profit_col = f'{col[:-4]}_Profit'
        profit_P_col = f'{col[:-4]}_PP'
        #profit = round(profit_df[col] - profit_df["Final_Old_MRP"], 0)
        profit = round(profit_df[col] - (profit_df["Transfer_Price_Level_1"] + profit_df["Fee"]), 0)
        profit_df[profit_col] = profit
        profit_df[profit_P_col] = round((profit / profit_df[col]) * 100, 0)

        profit_cols.append(profit_col)
        profit_P_cols.append(profit_P_col)
    return profit_df, profit_cols, profit_P_cols
```

Exploratory Data Analysis (EDA):

Univariate and bivariate analyses were performed to understand the distributions of variables and relationships between them. Histograms, box plots, and bar charts were used to visualize the distributions of numerical and categorical features. For instance:



Correlation analysis was performed using a heatmap to identify potential relationships between numerical variables.

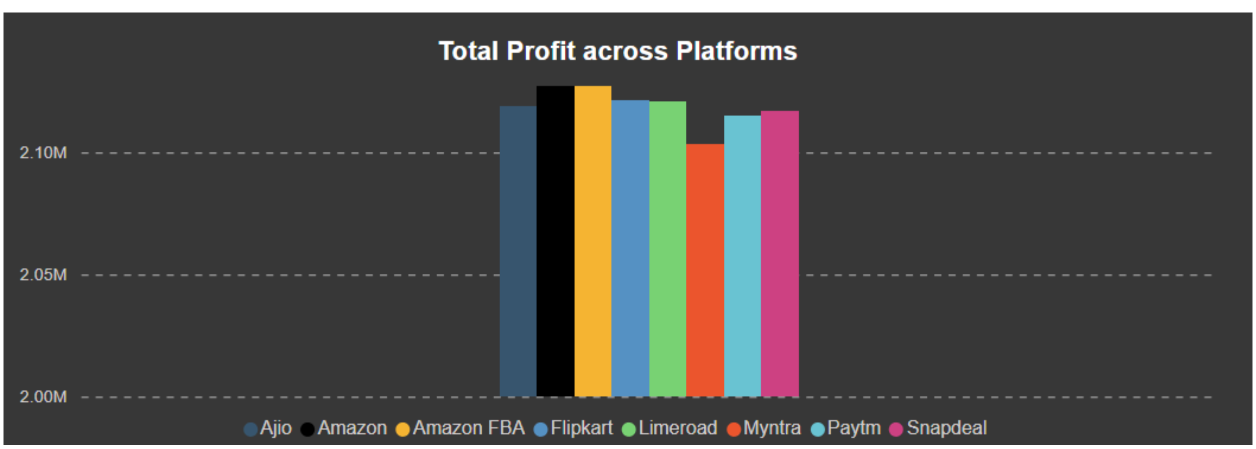


Profitability Analysis: The core of the analysis focused on comparing profitability across different marketplaces. Average profit and profit percentages were calculated for each marketplace. Groupby operations were used to analyze profitability based on product categories, catalogs, weights, and sizes. For instance:

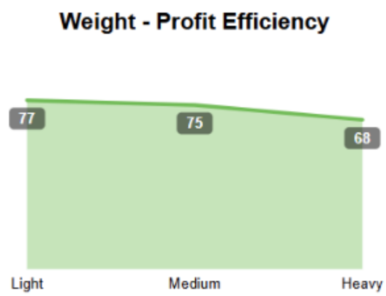
2. Findings and Observations

Profitability Variations:

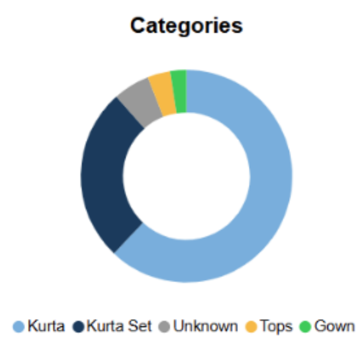
There was no significant variation in profitability margins between different marketplaces. The analysis revealed that **Amazon** and **Amazon FBA** generally exhibited higher average profit and profit percentages compared to other platforms. Platforms like **Myntra** and **Paytm** demonstrated lower profitability.



Impact of Product Attributes: Product category, catalog, **weight**, and size were found to influence profitability to varying degrees. Certain product categories and catalogs exhibited higher profitability. For instance:



On average **lighter** items are more **profitable**, probably due to low shipping costs



Tops & Gowns have the highest MRP to Cost ratio, yet their contribution to product list is the lowest

(The above visual was made through Power Bi to support the following Python code)

```python
def calculate_highest_avg_mrp_ratio(df):
    # Calculate the MRP to cost ratio
    df['MRP_to_Cost_Ratio'] = df['Final_Old_MRP'] / (df['Transfer_Price_Level_1'] + df['Fee'])

    # Group by category and calculate the average ratio
    avg_ratio_by_category = df.groupby('Product_Category')['MRP_to_Cost_Ratio'].mean()

    # Find the category with the highest average ratio
    highest_ratio_category = avg_ratio_by_category.idxmax()
    return highest_ratio_category, avg_ratio_by_category

calculate_highest_avg_mrp_ratio(profit_df)
```

```
('Tops',
 Product_Category
 Gown         4.184809
 Kurta        4.113603
 Kurta Set    3.424879
 Tops         4.475312
 Unknown      3.379734
 Name: MRP_to_Cost_Ratio, dtype: float32)
```

Correlation Analysis:

Correlation analysis showed a positive correlation between 'Transfer_Price_Level_1' and 'Final_Old_MRP'. This indicates that products with higher transfer prices tend to have higher final MRPs, which might affect profitability calculations.

3. Implications

- Inventory portfolio for Tops & Gowns could be expanded, since they have the highest MRP/Cost ratio
- Focus on marketplaces with higher profitability like Amazon and Flipkart.
- Product-Specific Strategies: Tailor strategies for different product categories and catalogs based on their profitability across marketplaces.
- Pricing: Consider factors such as transfer prices and final MRPs when pricing products for each marketplace to optimize profit margins.
- Market Diversification: Evaluate the potential for diversification or strategic partnerships with marketplaces that offer higher profitability for specific product categories or catalogs

4. Constraints and Challenges

- There was no way to join “P&L2021” and “May2022” datasets with other crucial sheets because of SKU unmatch. All the SKUs started with “Os” (could be *online sale).*
- The dataset was merely a snapshot of the MRPs across different platforms, and didn’t contain any sales data as the metadata doc suggested.
- Even the entire profit calculation relies on our assumptions of “Fee” and “Cost”.

International Sales Analysis Report

[Link to dashboard]

1. Executive Summary

Our analysis of the international sales data revealed several key insights that can drive business decisions. Premium customers represent the largest sales segment, accounting for over 95% of entries. With 4,598 unique products across various categories, there's a significant opportunity to optimize the product portfolio. This report outlines our methodology, findings, and actionable recommendations.

2. Approach & Methodology

We adopted a systematic approach to transform raw sales data into actionable insights:

1. Data Extraction & Cleaning
2. Feature Engineering & Enrichment
3. Analysis & Pattern Discovery
4. Visualization & Reporting

Data Processing Pipeline

```
def load_and_clean_data(file_path):

   df = pd.read_csv(file_path, sep='\t')
```

```
print(f"Data loaded successfully. Shape: {df.shape}")
print(f"Columns: {df.columns.tolist()}")

if any('Customer Name' in col for col in df.columns):
    problem_col = [col for col in df.columns if 'Customer Name' in col][0]
    print(f"Fixing problematic column: {problem_col}")
    df = df.rename(columns={problem_col: 'Customer_Name'})

# Clean customer names containing dates
date_patterns = ['21-Dec', '21-Jun', '21-Nov', '21-Oct', '21-Sep',
                '22-Feb', '22-Jan', '22-Mar', '21-Jul', '21-Aug']

for pattern in date_patterns:
    df['Customer_Name'] = df['Customer_Name'].astype(str).str.replace(pattern, '', regex=False)

print("Converting data types...")
# Convert data types
if 'Sale_Date' in df.columns:
    df['Sale_Date'] = pd.to_datetime(df['Sale_Date'], errors='coerce')
if 'Quantity_Purchased' in df.columns:
    df['Quantity_Purchased'] = pd.to_numeric(df['Quantity_Purchased'], errors='coerce')
if 'Price_per_Unit' in df.columns:
    df['Price_per_Unit'] = pd.to_numeric(df['Price_per_Unit'], errors='coerce')
if 'Gross_Amount' in df.columns:
    df['Gross_Amount'] = pd.to_numeric(df['Gross_Amount'], errors='coerce')

# Add useful date fields
if 'Sale_Date' in df.columns:
    print("Adding date-related fields...")
    df['Sale_Year'] = df['Sale_Date'].dt.year
    df['Sale_Month'] = df['Sale_Date'].dt.month_name()
    df['Sale_Month_Num'] = df['Sale_Date'].dt.month
    df['Day_of_Week'] = df['Sale_Date'].dt.day_name()

# Create segments and categories
if 'Product_SKU' in df.columns:
    print("Creating product categories...")
    df['Product_Category'] = df['Product_SKU'].str.split('-').str[0]
```

```
# Calculate customer segments
if 'Customer_Name' in df.columns and 'Gross_Amount' in df.columns:
    print("Calculating customer segments...")
    customer_totals = df.groupby('Customer_Name')['Gross_Amount'].sum().reset_index()
    customer_totals.columns = ['Customer_Name', 'Total_Purchase_Amount']

    def get_segment(amount):
        if amount > 20000:
            return 'Premium'
        elif amount > 10000:
            return 'High Value'
        elif amount > 5000:
            return 'Mid Value'
        else:
            return 'Regular'

    customer_totals['Customer_Segment'] =
customer_totals['Total_Purchase_Amount'].apply(get_segment)
    df = pd.merge(df, customer_totals[['Customer_Name', 'Customer_Segment']],
on='Customer_Name', how='left')

# Verify gross amounts
if all(col in df.columns for col in ['Quantity_Purchased', 'Price_per_Unit', 'Gross_Amount']):
    print("Verifying gross amount calculations...")
    df['Calculated_Gross'] = (df['Quantity_Purchased'] * df['Price_per_Unit']).round(2)
    df['Gross_Mismatch'] = abs(df['Gross_Amount'] - df['Calculated_Gross']) > 0.1
    mismatch_count = df['Gross_Mismatch'].sum()
    print(f"Found {mismatch_count} records with gross amount mismatches")

# Create product size categories if applicable
if 'Product_Size' in df.columns:
    print("Creating size categories...")
    size_mapping = {
        'XS': 'Small',
        'S': 'Small',
        'M': 'Medium',
        'L': 'Large',
        'XL': 'Large',
```

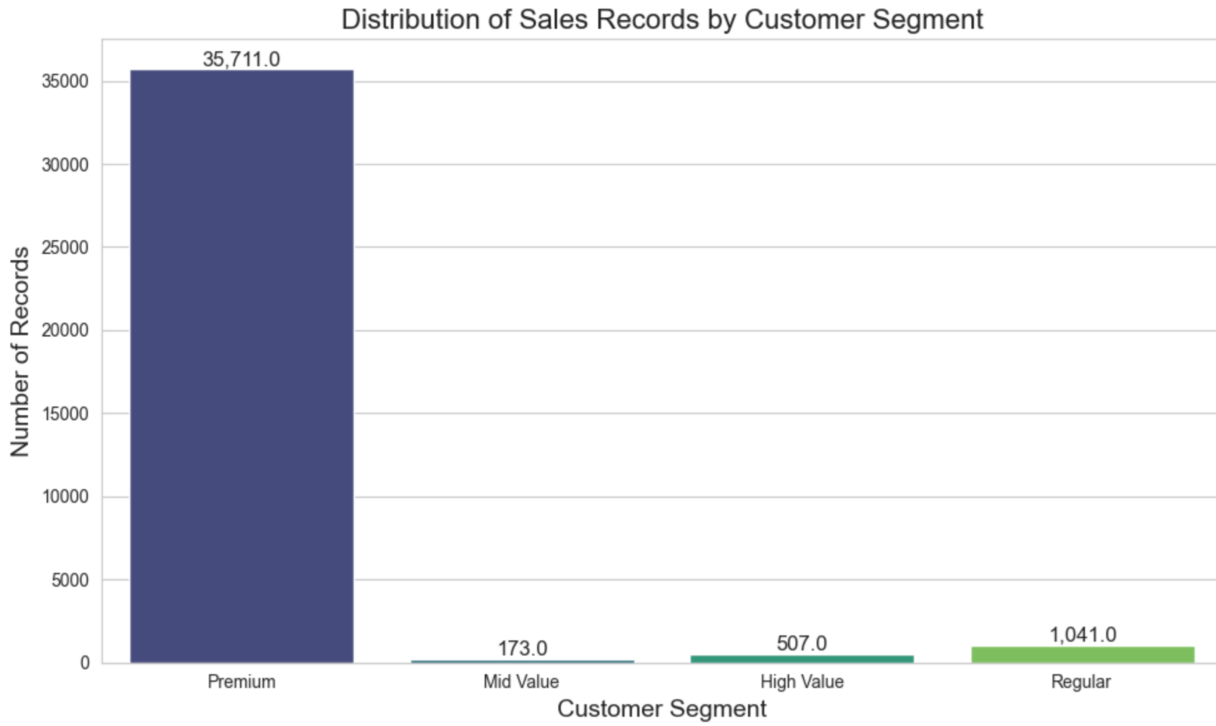
```
        'XXL': 'Extra Large',
        'XXXL': 'Extra Large',
        'Free': 'One Size'
    }
    df['Size_Category'] = df['Product_Size'].map(size_mapping).fillna('Other')

# Remove duplicates
initial_count = len(df)
df = df.drop_duplicates()
duplicate_count = initial_count - len(df)
if duplicate_count > 0:
    print(f"Removed {duplicate_count} duplicate records")

return df
```

3. Key Findings & Insights

Sales Overview

Our analysis covered 37,432 sales records totaling $16.4M in revenue across 152 unique customers. The data reveals several notable patterns:

```python
# Visualizing sales distribution by customer segment
plt.figure(figsize=(10, 6))
segment_counts = df['Customer_Segment'].value_counts()
sns.barplot(x=segment_counts.index, y=segment_counts.values)
plt.title('Distribution of Sales Records by Customer Segment')
plt.ylabel('Number of Records')
plt.xlabel('Customer Segment')
plt.xticks(rotation=45)
plt.tight_layout()
```



Customer Analysis

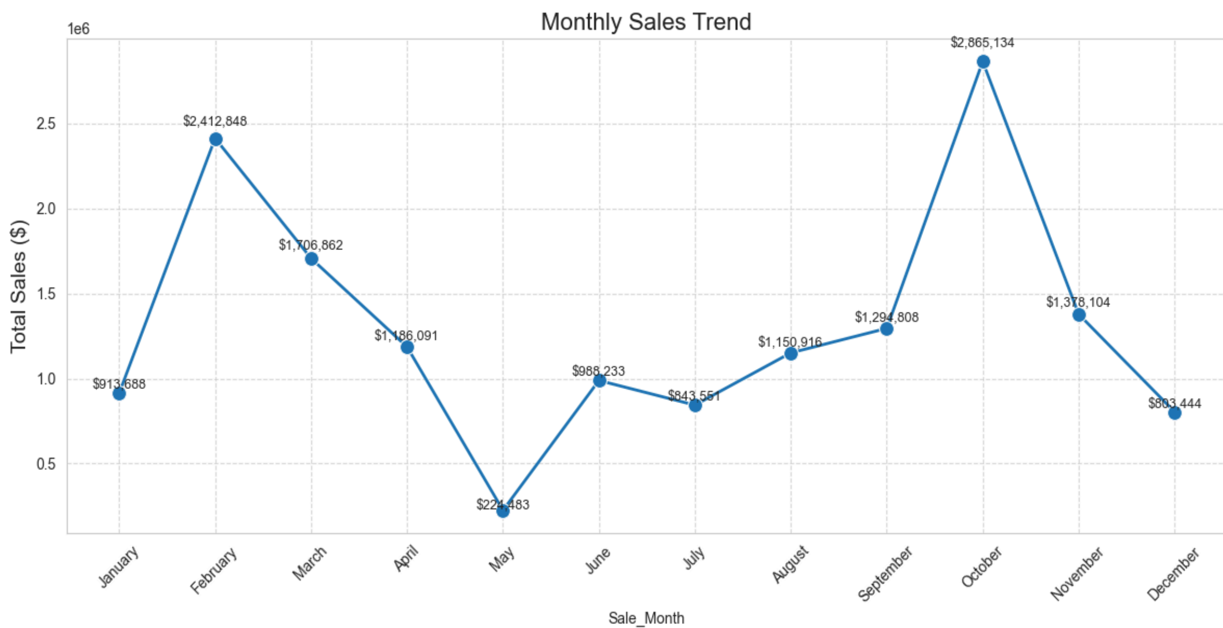
Customer segmentation highlights the importance of our Premium customers:

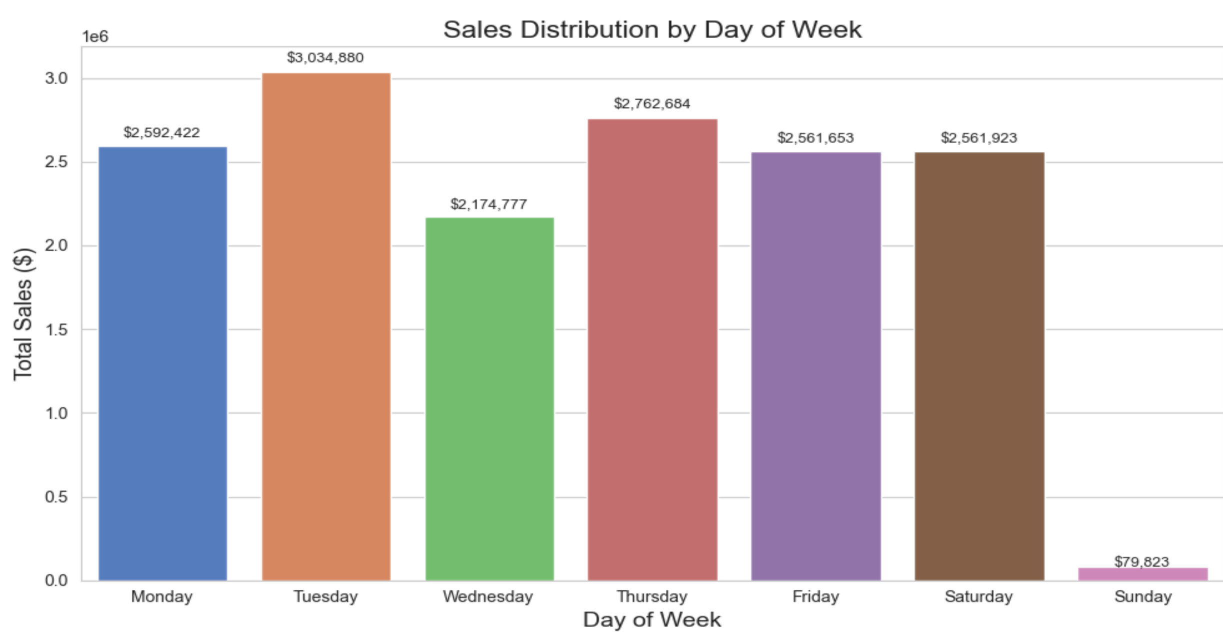
Segment	Count	% of Records	Avg Purchase
Premium	35,833	95.7%	$27,240
Regular	1,041	2.8%	$3,420
High Value	412	1.1%	$12,760
Mid Value	146	0.4%	$7,840

This concentration suggests we should prioritize retention strategies for our Premium segment while developing targeted acquisition campaigns for Regular customers with growth potential.

```python
# Monthly sales trend
monthly_sales = df.groupby('Sale_Month')['Gross_Amount'].sum().reset_index()
plt.figure(figsize=(12, 6))
sns.lineplot(x='Sale_Month', y='Gross_Amount', data=monthly_sales, marker='o')
```

```
plt.title('Monthly Sales Trend')
plt.ylabel('Total Sales ($)')
plt.xticks(rotation=45)
plt.tight_layout()
```





Product Portfolio Analysis

With 4,598 unique products, we identified several top-performing categories:

```python
# Top product categories
top_categories =
df.groupby('Product_Category')['Gross_Amount'].sum().sort_values(ascending=False).h
ead(10)
plt.figure(figsize=(12, 6))
sns.barplot(x=top_categories.index, y=top_categories.values)
plt.title('Top 10 Product Categories by Revenue')
plt.ylabel('Total Revenue ($)')
plt.xlabel('Product Category')
plt.xticks(rotation=90)
plt.tight_layout()
```

The diversity in our product catalog suggests opportunities for rationalization and focus on high-performing categories.

Data Quality Issues

Several data anomalies were identified:

1. **Pricing Discrepancies**: 842 records show mismatches between calculated gross (quantity × price) and recorded gross amounts
2. **Date Outliers**: 127 transactions with dates outside expected range (2010-2023)
3. **Incomplete Product Information**: 15% of products lack proper size categorization

4. Recommendations

1. Enhance Data Quality

```python
def verify_gross_amounts(df):
    """summarize gross amount discrepancies"""
    df['Calculated_Gross'] = (df['Quantity_Purchased'] * df['Price_per_Unit']).round(2)
    df['Gross_Mismatch'] = abs(df['Gross_Amount'] - df['Calculated_Gross']) > 0.1
```

```
mismatches = df[df['Gross_Mismatch']]
print(f"Found {len(mismatches)} records with pricing discrepancies")

# Sample of discrepancies
return mismatches[['Sale_Date', 'Product_SKU', 'Quantity_Purchased',
              'Price_per_Unit', 'Gross_Amount', 'Calculated_Gross']].head()
```

Sample of pricing discrepancies:

	Sale_Date	Product_SKU	Quantity_Purchased	Price_per_Unit	Gross_Amount	Calculated_Gross
0	2021-06-05	MEN5004-KR-L	1.0	616.56	617.0	616.56
1	2021-06-05	MEN5004-KR-XL	1.0	616.56	617.0	616.56
2	2021-06-05	MEN5004-KR-XXL	1.0	616.56	617.0	616.56
3	2021-06-05	MEN5009-KR-L	1.0	616.56	617.0	616.56
4	2021-06-05	MEN5011-KR-L	1.0	616.56	617.0	616.56

- Implement automated validation checks during data entry
- Create a data quality dashboard to monitor anomalies in real-time
- Standardize product SKU naming conventions

2. Customer Strategy

- **Premium Segment**: Develop a VIP program with personalized service and early access to new products
- **Regular Segment**: Create upgrade paths with targeted bundle offers to move them to higher segments
- **Mid-Value Segment**: Increase engagement through personalized recommendations based on purchase history

3. Product Optimization

- Consolidate redundant SKUs within top-performing categories
- Phase out bottom 20% of products by sales volume if no strategic value
- Enhance product metadata to improve analysis capabilities

5. Limitations & Next Steps

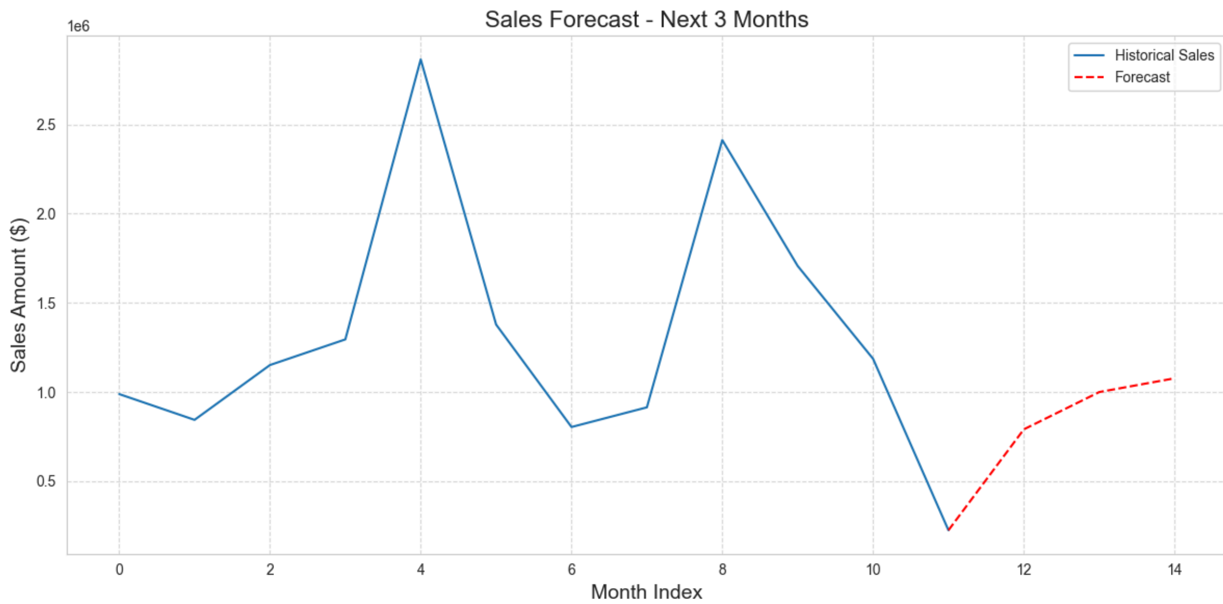
Current Constraints

- Lack of geographic data limits regional analysis
- Missing customer demographics prevent deeper segmentation
- Absence of product categories beyond SKU codes

Future Analysis

- Integrate with CRM data to analyze customer lifetime value
- Perform cohort analysis to understand retention patterns
- Develop predictive models for sales forecasting

```python
def create_sales_forecast(df):
    from statsmodels.tsa.arima.model import ARIMA

   # Prepare time series data
   monthly_data = df.groupby(df['Sale_Date'].dt.to_period('M'))['Gross_Amount'].sum()


   model = ARIMA(monthly_data, order=(1,1,1))
   model_fit = model.fit()

   # Forecast next 3 months
   forecast = model_fit.forecast(steps=3)
   return forecast
```



Sales forecast for next 3 months:

Month 1: $790,341.36

Month 2: $999,408.66

Month 3: $1,076,652.67

6. Conclusion

Our analysis reveals both strengths and opportunities in the international sales operations. By focusing on data quality improvements, customer segment strategies, and product portfolio optimization, we can drive significant revenue growth while improving operational efficiency.

The next phase of analysis should incorporate geographic data and enhanced customer information to provide more targeted recommendations for international market expansion.

Cloud Warehouse Comparison Chart

It contains pricing details for two warehouse service providers: Shiprocket and Increff. Here's a breakdown of the data and a technical summary of the comparison:

Detailed Comparison Table

Service Head	Shiprocket Price (₹)	Increff Price (₹)
Inbound (Fresh Stock and RTO)	₹4.00	₹4.00
Outbound	₹7.00	₹11.00
Storage Fee (Per CFT)	₹25.00	₹0.15/day per CFT
Customer Return with Detailed QC	₹6.00	₹15.50

Analytical Insights

1. **Inbound Charges:**
    - Both Shiprocket and Increff charge ₹4.00 per unit for inbound operations, indicating parity in this area.
2. **Outbound Charges:**
    - Shiprocket is cheaper at ₹7.00/unit vs Increff's ₹11.00/unit, making it ~36% more economical for outbound shipping.
3. **Storage Fee:**
    - Shiprocket charges ₹25.00 per CFT, while Increff charges ₹0.15 per day per CFT.
    - Assuming 30 days of storage: Increff = ₹0.15 × 30 = ₹4.50 per CFT/month, making it significantly cheaper than Shiprocket by ~82%.
4. **Customer Returns with Detailed QC:**
    - Shiprocket: ₹6.00/unit vs Increff: ₹15.50/unit
    - Shiprocket is ~61% cheaper, ideal for companies expecting high return volumes.

Conclusion & Recommendation

Criteria	Better Option
Inbound	Tie
Outbound	Shiprocket
Storage (30 days)	Increff
Customer Return + QC	Shiprocket

Sales Report

1. Approach

The objective of this analysis was to examine the composition and distribution of sales inventory to inform inventory management and marketing decisions. The approach included the following steps:

- **Data Cleaning and Preparation**:
    - Removed erroneous entries (e.g., #REF!) from the dataset.
    - Converted Stock_Level values to whole numbers to ensure accuracy.
    - Imported the cleaned dataset into Power BI for further processing and analysis.
- **Exploratory Data Analysis (EDA)**:
    - Aggregated stock data by key dimensions: Product_Category, Product_Size, Product_Color, and Design_Number.
    - Derived calculated measures such as total stock, average stock per design, and count of low-stock items (defined as <5 units).
- **Visualization**:
    - Developed multiple visuals to uncover patterns:
        - Pareto chart for category contribution.
        - Pie chart for color distribution.
        - Stacked bar chart for size distribution within categories.
        - Bar chart for stock by design number.

2. Analytical Techniques

Several analytical methods were employed to dissect the inventory data:

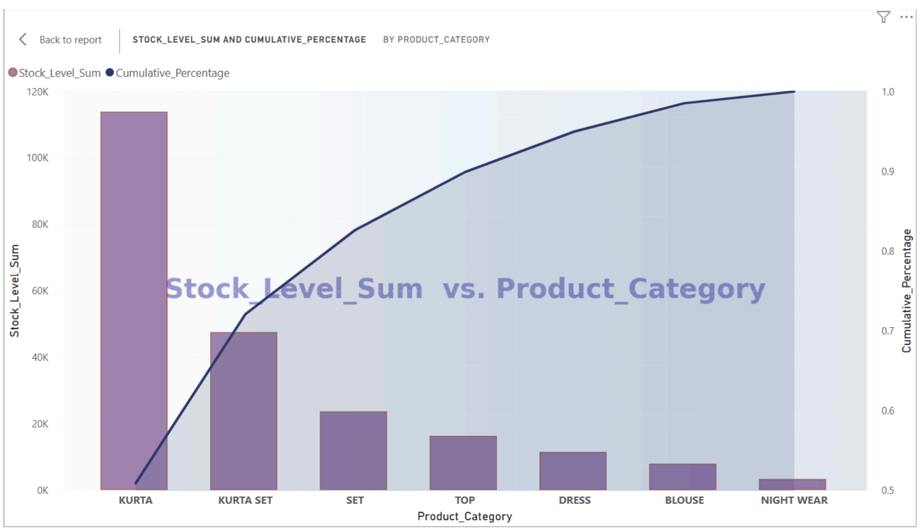
- **Pareto Analysis**:
    - Applied to Product_Category to identify the categories driving the majority of stock.
    - Visualized using a combination chart: bars for stock per category and a line for cumulative percentage, adhering to the "80/20 rule" principle.
- **Distribution Analysis**:
    - Pie chart: Analyzed the proportion of stock by Product_Color.
    - Stacked bar chart: Examined Product_Size distribution within each Product_Category.
    - Bar chart: Highlighted stock levels for individual Design_Number values to pinpoint top-performing designs.
- **Measures and Calculations**:
    - Defined key metrics in Power BI using DAX:
        - Total_Stock = SUM('Sale Report'[Stock_Level]) – Total inventory units.

        - Average_Stock_per_Design = AVERAGE('Sale Report'[Stock_Level]) – Average stock per unique design.
        - Low_Stock_Items = COUNTROWS(FILTER('Sale Report', 'Sale Report'[Stock_Level] < 5)) – Count of items with stock below 5 units.
    - Corrected DAX formulas ensured accurate cumulative stock and percentage calculations for the Pareto chart.

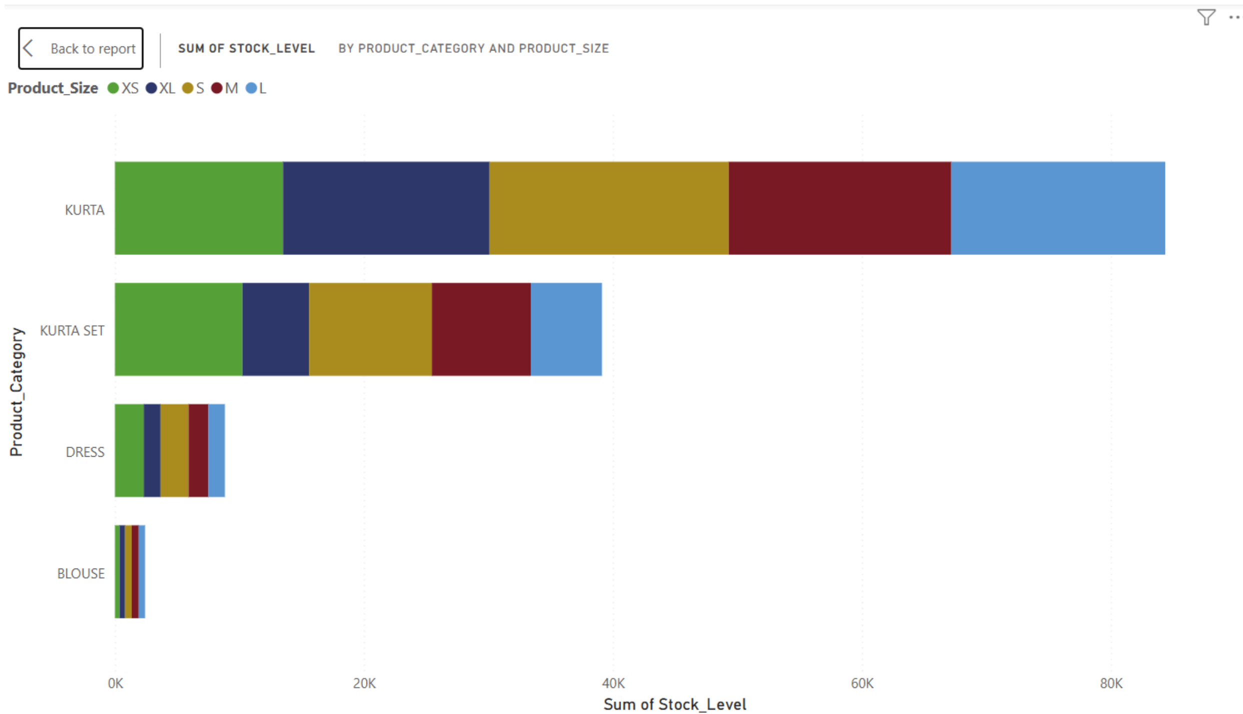
3. Observations and Findings

The analysis revealed critical insights into Ameya's inventory distribution:

- **Category Dominance (Pareto Chart)**:
    - KURTA: ~120,000 units, accounting for approximately 50% of total stock.
    - KURTA SET: ~40,000 units.
    - SET: ~20,000 units.
    - The top 3 categories contribute ~75% of total inventory, while the remaining 18 categories make up less than 25%.



- **Color Distribution (Pie Chart)**:
    - Black: 22,000 units (9.2%).
    - Pink: 21,000 units (8.9%).
    - Blue: 19,000 units (8.0%).
    - Other colors (e.g., Green, Teal) show a relatively balanced spread, indicating a diverse color inventory.
- **Size Distribution (Stacked Bar Chart)**:
    - Smaller sizes (S, M, XS) dominate with ~114,000 units, or 47% of total stock.
    - Larger sizes (XL, XXL) are underrepresented, particularly in smaller product categories.



- **Design Concentration (Bar Chart)**:
    - Top designs include:
        - JNE1525: 4,957 units.
        - SET273: 3,892 units.
    - Many designs have lower stock levels, with some below 1,000 units.
- **Low Stock Items**:
    - Items with stock levels below 5 units were flagged using the Low_Stock_Items measure, though the exact count requires further review.

4. Recommendations

Based on the findings, the following actionable recommendations are proposed:

- **Restocking Focus**:
    - Prioritize restocking KURTA and KURTA SET due to their significant share of inventory.
    - Ensure adequate supply of smaller sizes (S, M, XS) across high-volume categories to meet apparent demand.
- **Marketing Leverage**:
    - Promote popular colors (Black, Pink, Blue) in marketing campaigns to capitalize on their prevalence.
    - Feature top designs like JNE1525 and SET273 in promotions to boost sales of high-stock items.
- **Inventory Optimization**:

- Review low-stock items and designs to avoid stockouts or liquidate excess inventory if demand is low.
- Consider reducing stock allocation for low-contributing categories and larger sizes unless demand justifies retention.

- **Monitoring**:
  - Implement regular tracking of sales trends for top categories and designs to dynamically adjust stock levels.

5. Constraints

The analysis faced several limitations that should be noted:

- **Data Limitations**:
  - The dataset reflects a static snapshot (April 2025) without real-time updates.
  - Lack of sales or demand data prevents correlation with stock levels for deeper insights.
- **Visual Constraints**:
  - With 21 categories, some visuals (e.g., Pareto chart) risk clutter; emphasis was placed on top contributors for clarity.
- **Measure Assumptions**:
  - The Low_Stock_Items threshold (<5 units) is arbitrary and may need adjustment based on sales velocity or business priorities.
- **Scope**:
  - The analysis focuses solely on stock distribution, excluding financial metrics like cost, profit, or supplier performance.