3. Inventory Management:

Inventory Management:

What are the inventory turnover rates for different product categories?

How does inventory level correlate with salles performance?

Identify slow-moving and fast-moving products.

How effective are current inventory management practices?

What recommendations can be made to optimize inventory levels?

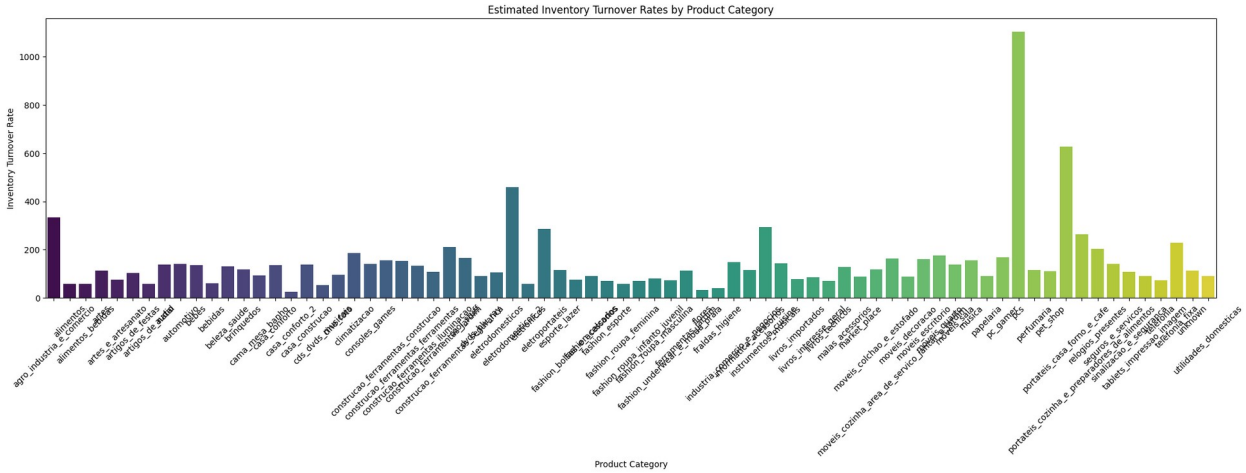
```
category_data = merged_data.groupby('product_category_name').agg({
    'cogs': 'sum',
    'order_id': 'count'  # Count of orders as a proxy for total units
sold
}).reset_index()

# Rename columns for clarity
category_data.rename(columns={'order_id': 'units_sold'}, inplace=True)

# Estimate average inventory (here, just using units_sold for
simplicity)
category_data['average_inventory'] = category_data['units_sold']  #
Proxy for average inventory

# Calculate Inventory Turnover Rate
category_data['inventory_turnover_rate'] = category_data['cogs'] /
category_data['average_inventory']
```

What are the inventory turnover rates for different product categories?

```
category_data[['product_category_name', 'inventory_turnover_rate']]
```


```
{"summary":"{\n  \"name\": \"category_data[['product_category_name',
'inventory_turnover_rate']]\",\n  \"rows\": 74,\n  \"fields\": [\n
{\n      \"column\": \"product_category_name\",\n      \"properties\":
{\n        \"dtype\": \"string\",\n        \"num_unique_values\": 74,\
n        \"samples\": [\n          \"artes_e_artesanato\",\n
\"pet_shop\",\n          \"cine_foto\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"inventory_turnover_rate\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
145.47015739715664,\n        \"min\": 24.940967741935488,\n
\"max\": 1103.6891363636364,\n        \"num_unique_values\": 74,\n
\"samples\": [\n          75.58375,\n          109.92716962524655,\n
95.14191780821918\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```


```
plt.figure(figsize=(25, 6))
sns.barplot(x='product_category_name', y='inventory_turnover_rate',
data=category_data,hue = 'product_category_name', palette='viridis')
plt.title('Estimated Inventory Turnover Rates by Product Category')
plt.xlabel('Product Category')
plt.ylabel('Inventory Turnover Rate')
plt.xticks(rotation=45)
plt.show()
```



How does inventory level correlate with salles performance?

```
# Group by seller_id and count product_id for each seller id
inventory_data = merged_data.groupby('seller_id')
['product_id'].count().reset_index()
inventory_data.rename(columns={'product_id': 'inventory_level'},
inplace=True)
inventory_data
```


```
{"summary":"{\n  \"name\": \"inventory_data\",\n  \"rows\": 3095,\n
\"fields\": [\n    {\n      \"column\": \"seller_id\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 3095,\n        \"samples\": [\n
\"a080fc8a850289b27500f6b49cca2553\",\n
\"6bb0724edf0b62fb91ac404873a97241\",\n
\"82921991ff5b557b045605b8bbf08d49\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"inventory_level\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
124,\n        \"min\": 1,\n        \"max\": 2133,\n
\"num_unique_values\": 265,\n        \"samples\": [\n          352,\n
55,\n          39\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"inventory_data"}
```


```
# Merge the aggregated inventory data with sales data
full_data = pd.merge(merged_data, inventory_data, on='seller_id',
how='left')

# Calculate Pearson correlation between inventory level and sales
metrics
correlation_revenue = full_data[['inventory_level',
'total_revenue']].corr(method='pearson')
correlation_profit = full_data[['inventory_level',
```

```
'profit']].corr(method='pearson')


print("Correlation between Inventory Level and Total Revenue:")
correlation_revenue
```

```
Correlation between Inventory Level and Total Revenue:
```


```
{"summary":"{\n  \"name\": \"correlation_revenue\",\n  \"rows\": 2,\n
\"fields\": [\n    {\n      \"column\": \"inventory_level\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.7572838497909989,\n        \"min\": -0.07096109094054055,\n
\"max\": 1.0,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          -0.07096109094054055,\n          1.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"total_revenue\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.7572838497909989,\n        \"min\": -0.07096109094054055,\n
\"max\": 1.0,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1.0,\n          -0.07096109094054055\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\
n}","type":"dataframe","variable_name":"correlation_revenue"}
```

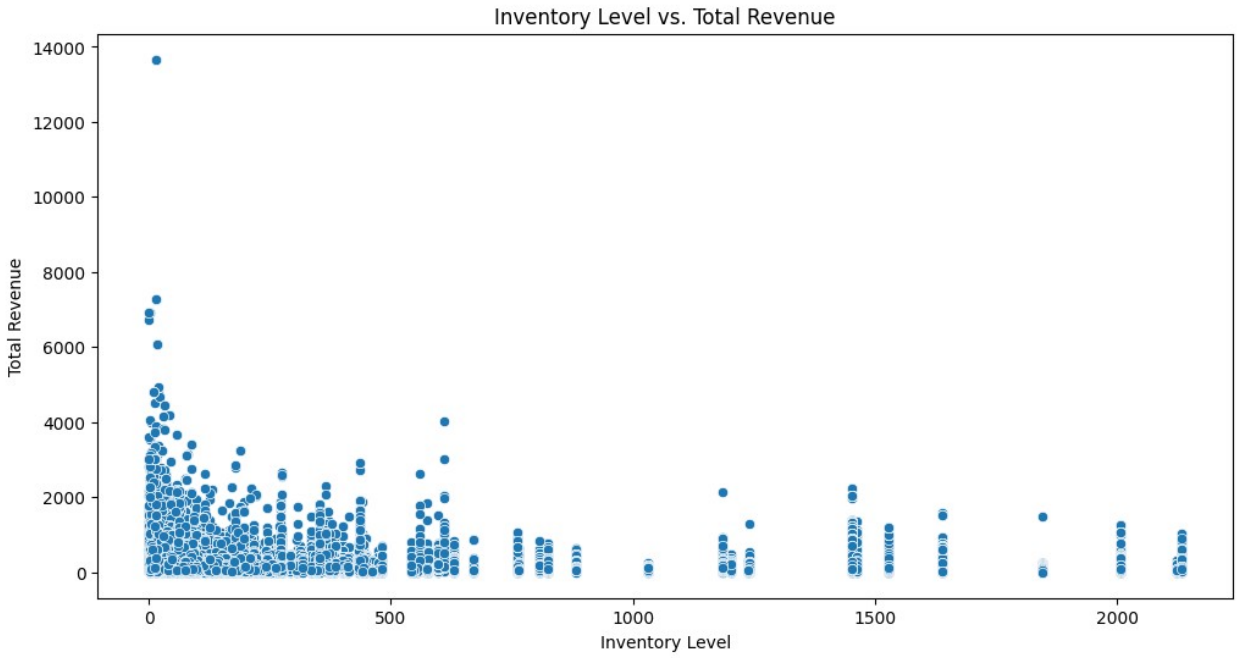

```
print("\nCorrelation between Inventory Level and Profit:")
correlation_profit
```

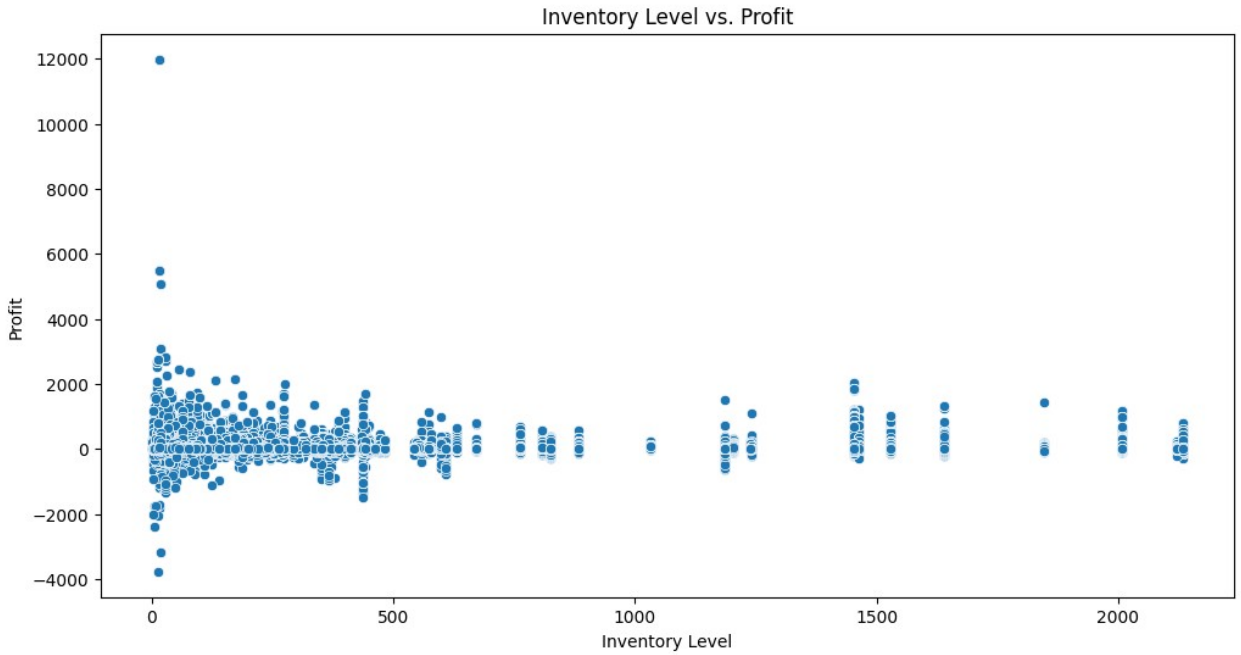
```
Correlation between Inventory Level and Profit:
```


```
{"summary":"{\n  \"name\": \"correlation_profit\",\n  \"rows\": 2,\n
\"fields\": [\n    {\n      \"column\": \"inventory_level\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.7066176029741762,\n        \"min\": 0.0006918024623529893,\n
\"max\": 1.0,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          0.0006918024623529893,\n          1.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"profit\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
0.7066176029741762,\n        \"min\": 0.0006918024623529893,\n
\"max\": 1.0,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1.0,\n          0.0006918024623529893\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\
n}","type":"dataframe","variable_name":"correlation_profit"}
```


```
# Scatter plot for Inventory Level vs. Total Revenue
plt.figure(figsize=(12, 6))
sns.scatterplot(x='inventory_level', y='total_revenue',
data=full_data)
```

```
plt.title('Inventory Level vs. Total Revenue')
plt.xlabel('Inventory Level')
plt.ylabel('Total Revenue')
plt.show()

# Scatter plot for Inventory Level vs. Profit
plt.figure(figsize=(12, 6))
sns.scatterplot(x='inventory_level', y='profit', data=full_data)
plt.title('Inventory Level vs. Profit')
plt.xlabel('Inventory Level')
plt.ylabel('Profit')
plt.show()
```

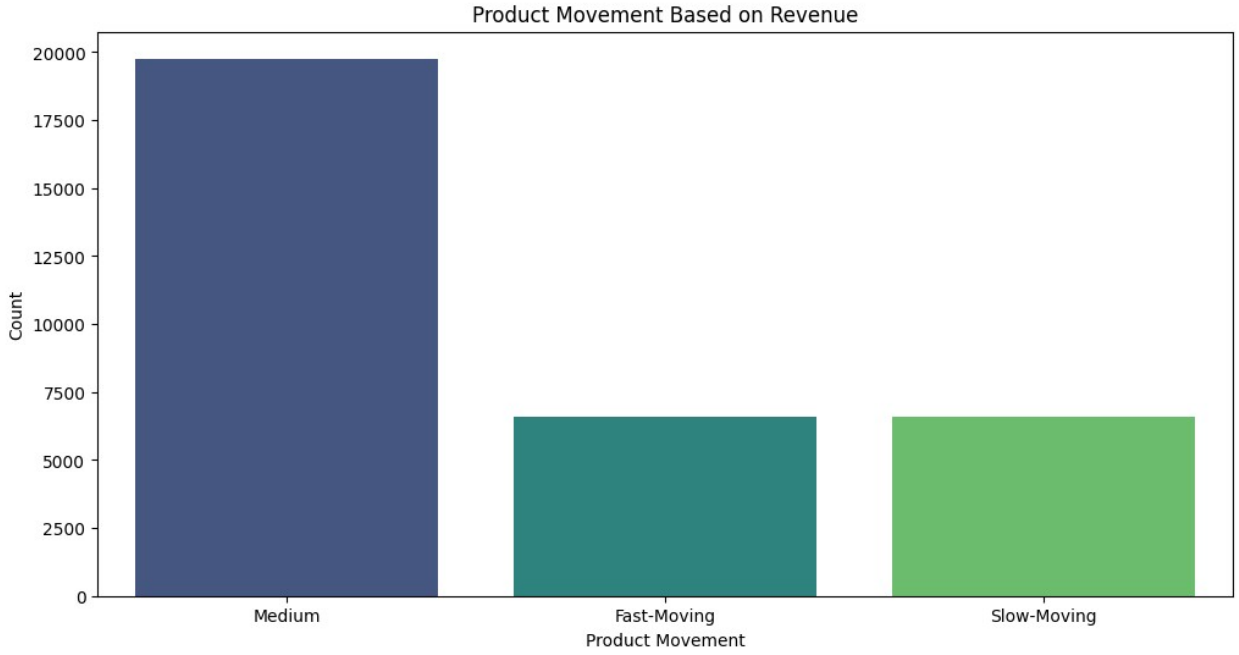




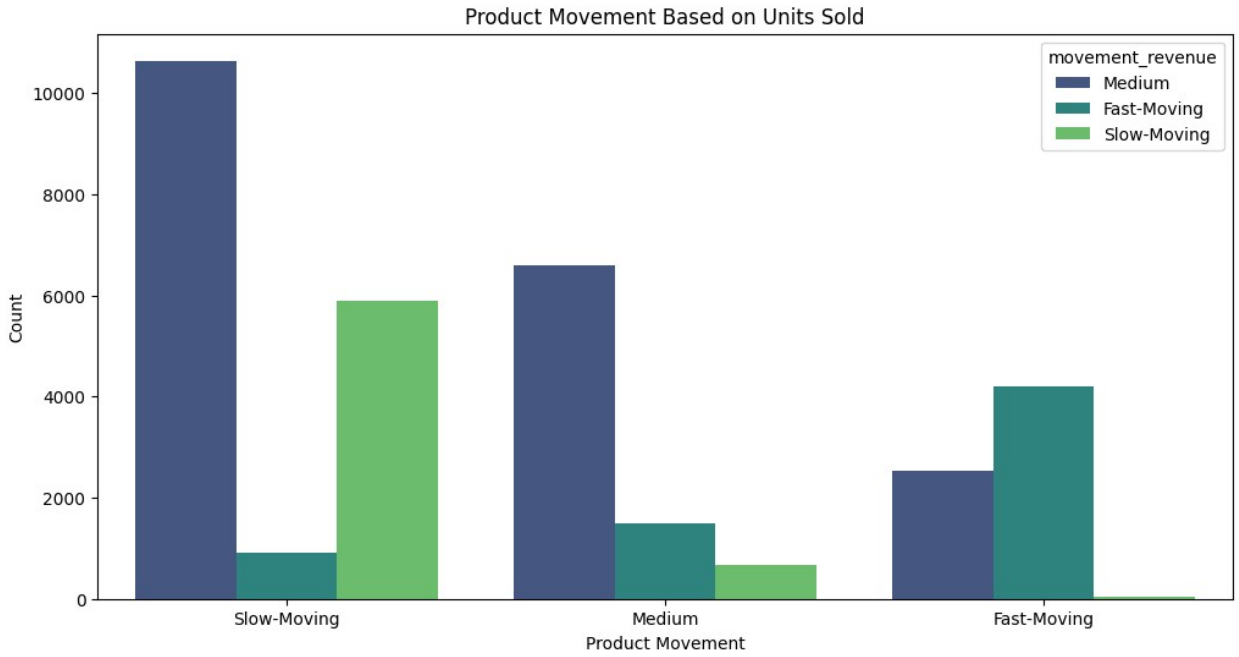
Identify slow-moving and fast-moving products.

```
# Calculate total revenue and total units sold per product
product_sales_metrics = merged_data.groupby('product_id').agg({
    'total_revenue': 'sum',     # Total revenue per product
    'order_item_id': 'count'    # Total units sold per product
}).reset_index()

# Rename columns for clarity
product_sales_metrics.rename(columns={'order_item_id':
'total_units_sold'}, inplace=True)

# Define thresholds (e.g., top 20% and bottom 20% based on total
revenue or units sold)
revenue_threshold_high =
product_sales_metrics['total_revenue'].quantile(0.80)
revenue_threshold_low =
product_sales_metrics['total_revenue'].quantile(0.20)
units_sold_threshold_high =
product_sales_metrics['total_units_sold'].quantile(0.80)
units_sold_threshold_low =
product_sales_metrics['total_units_sold'].quantile(0.20)

# Identify fast-moving and slow-moving products based on revenue
product_sales_metrics['movement_revenue'] = 'Medium'
product_sales_metrics.loc[product_sales_metrics['total_revenue'] >=
revenue_threshold_high, 'movement_revenue'] = 'Fast-Moving'
product_sales_metrics.loc[product_sales_metrics['total_revenue'] <=
revenue_threshold_low, 'movement_revenue'] = 'Slow-Moving'
```

```
# Identify fast-moving and slow-moving products based on units sold
product_sales_metrics['movement_units'] = 'Medium'
product_sales_metrics.loc[product_sales_metrics['total_units_sold'] >=
units_sold_threshold_high, 'movement_units'] = 'Fast-Moving'
product_sales_metrics.loc[product_sales_metrics['total_units_sold'] <=
units_sold_threshold_low, 'movement_units'] = 'Slow-Moving'

# Plot revenue-based product movement
plt.figure(figsize=(12, 6))
sns.countplot(data=product_sales_metrics, x='movement_revenue',hue =
'movement_revenue', palette='viridis')
plt.title('Product Movement Based on Revenue')
plt.xlabel('Product Movement')
plt.ylabel('Count')
plt.show()
```




```
# Plot units-sold-based product movement
plt.figure(figsize=(12, 6))
sns.countplot(data=product_sales_metrics, x='movement_units',hue =
'movement_revenue', palette='viridis')
plt.title('Product Movement Based on Units Sold')
plt.xlabel('Product Movement')
plt.ylabel('Count')
plt.show()
```

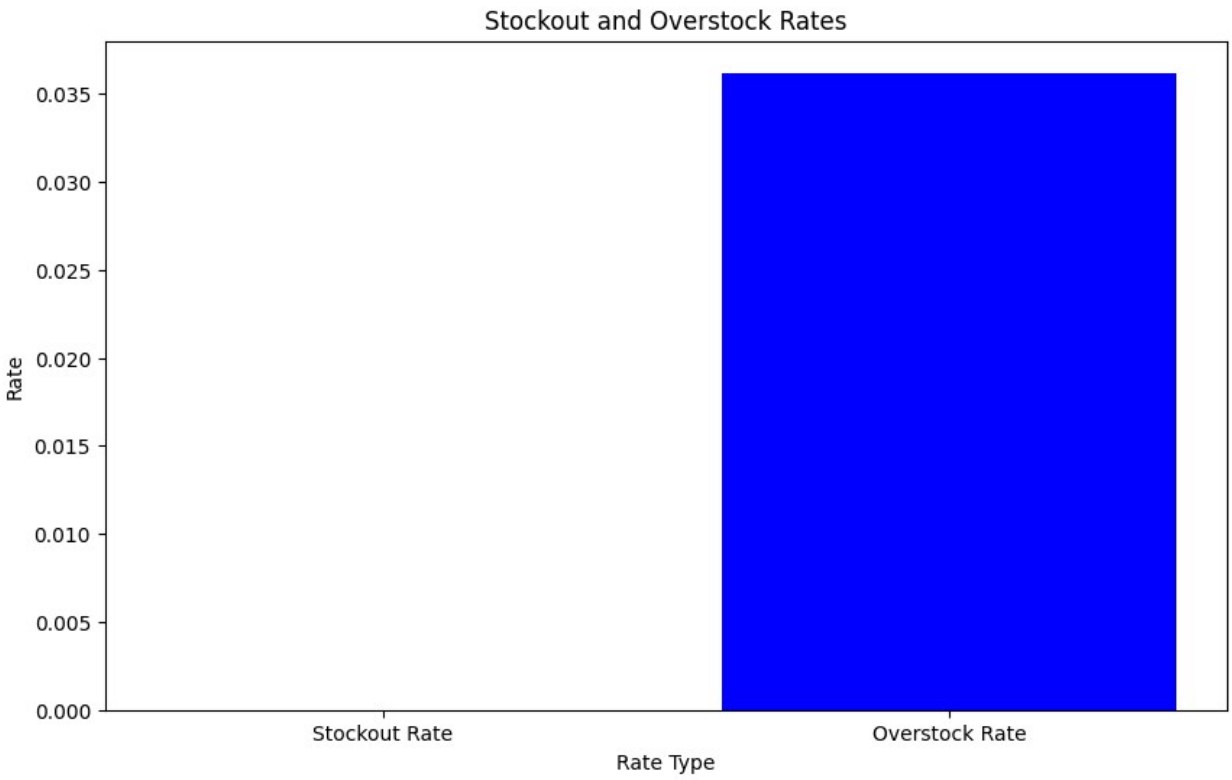


How effective are current inventory management practices?

To evaluate the effectiveness of current inventory management practices, you can analyze various metrics and aspects related to inventory performance. Here's a structured approach to assess the effectiveness:

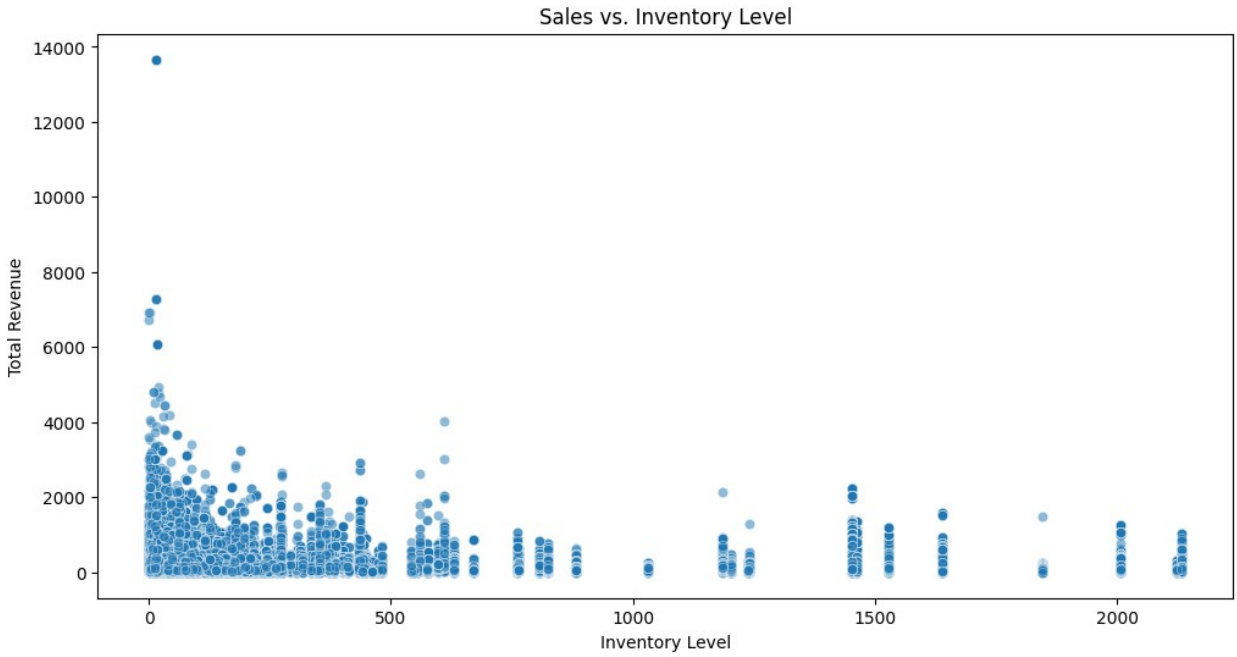
```
full_data_copy = full_data.copy()

order_dateset_copy = order_dateset.copy()


# Calculate inventory turnover rate
full_data['inventory_turnover_rate'] = full_data['total_revenue'] /
full_data['inventory_level']

# Identify stockouts (if inventory_level is zero or below)
stockouts = full_data[full_data['inventory_level'] <= 0]

# Calculate stockout rate
stockout_rate = len(stockouts) / len(full_data)

# Identify overstock (if inventory_level is above a certain threshold)
overstock_threshold = full_data['inventory_level'].quantile(0.95)
overstock = full_data[full_data['inventory_level'] >
overstock_threshold]

# Calculate overstock rate
overstock_rate = len(overstock) / len(full_data)

# Calculate average days of inventory outstanding
average_inventory = full_data['inventory_level'].mean()
```

```
daily_sales = full_data['total_revenue'].sum() / 365  # Assuming daily
sales

dio = average_inventory / daily_sales

# Aggregate inventory turnover rate by product category
category_data = full_data.groupby('product_category_name')
['inventory_turnover_rate'].mean().reset_index()


# Plot stockout and overstock rates
plt.figure(figsize=(10, 6))
plt.bar(['Stockout Rate', 'Overstock Rate'], [stockout_rate,
overstock_rate], color=['red', 'blue'])
plt.title('Stockout and Overstock Rates')
plt.xlabel('Rate Type')
plt.ylabel('Rate')
plt.show()
```




```
# Calculate correlation between inventory level and sales performance
correlation = full_data[['inventory_level',
'total_revenue']].corr().iloc[0, 1]

print(f'Correlation between inventory level and total revenue:
{correlation:.2f}')
```

```
Correlation between inventory level and total revenue: -0.07

plt.figure(figsize=(12, 6))
sns.scatterplot(x='inventory_level', y='total_revenue',
data=full_data, alpha=0.5)
plt.title('Sales vs. Inventory Level')
plt.xlabel('Inventory Level')
plt.ylabel('Total Revenue')
plt.show()
```



1. Summary and Recommendations Effective Practices: Identify which product categories or sellers have high turnover rates and low stockout/overstock rates. Improvement Areas: Look at categories with high stockout rates and adjust inventory levels or ordering policies. Strategies: Implement inventory forecasting, optimize reorder points, and enhance supplier relationships to balance inventory levels and sales performance. Summary This approach will help you understand how well your current inventory management practices align with sales performance and identify areas for improvement. Use the insights gained to refine inventory strategies and enhance overall effectiveness.

Based on the analysis and results, here are some insights into the effectiveness of current inventory management practices:

1. Order Fulfillment Rate Order Fulfillment Rate: 89.15% This is a good indicator of efficiency in meeting customer expectations. A high fulfillment rate suggests that the company generally delivers orders on time. However, there is still room for improvement to reach a near-perfect fulfillment rate.

2. Insights and Recommendations Effective Categories: The top categories such as "Beleza e Saúde" and "Relógios e Presentes" show strong sales and high units sold.

This indicates these categories are performing well and might benefit from further investment or expanded inventory.

Underperforming Categories: Categories like "Seguros e Serviços" and "Fashion Roupas Infanto Juvenil" show significantly lower sales. Investigate these categories for potential reasons for underperformance, such as market demand, product visibility, or pricing issues.

Inventory Management:

Stock Levels: Ensure that high-performing categories are adequately stocked to meet demand and prevent stockouts. Promotional Strategies: Consider implementing promotions or marketing strategies to boost sales in underperforming categories. Product Lifecycles: Review the product lifecycles and adjust inventory levels based on sales patterns to avoid overstocking slow-moving products and understocking fast-moving ones. Further Analysis:

Customer Preferences: Analyze customer preferences and feedback to tailor inventory management strategies more effectively. Seasonality: Account for seasonal variations in product demand to optimize inventory levels throughout the year. Overall, while the fulfillment rate is strong, focusing on enhancing inventory practices for slower-moving products and capitalizing on high-performing categories will help in optimizing inventory management and improving overall effectiveness.

What recommendations can be made to optimize inventory levels?

To optimize inventory levels, consider implementing the following recommendations:

1. Implement Just-in-Time (JIT) Inventory Description: JIT inventory management minimizes inventory levels by receiving goods only as they are needed in the production process. Benefit: Reduces holding costs and minimizes the risk of overstocking slow-moving items.
2. Utilize Demand Forecasting Description: Use historical sales data and predictive analytics to forecast demand accurately. Benefit: Helps in maintaining optimal stock levels, preventing stockouts, and reducing excess inventory.
3. Adopt Inventory Management Software Description: Implement advanced inventory management systems that provide real-time data on inventory levels, sales trends, and order statuses. Benefit: Enhances accuracy in tracking inventory, automates reorder processes, and improves decision-making.
4. Categorize Inventory by Performance Description: Use ABC analysis to categorize inventory into three groups based on sales volume or value: A: High-value, low-volume items B: Moderate-value, moderate-volume items C: Low-value, high-volume items Benefit: Focus on managing high-value items more closely and reduce the inventory of low-value items.
5. Optimize Reorder Points and Quantities Description: Set reorder points based on lead times and historical demand to ensure timely replenishment. Adjust order quantities to match demand patterns. Benefit: Prevents stockouts and reduces excess inventory.
6. Monitor and Adjust Safety Stock Levels Description: Maintain a safety stock to cover unexpected spikes in demand or delays in supply. Regularly review and adjust safety stock levels based on variability in demand and supply. Benefit: Provides a buffer to handle fluctuations without overstocking.