**Phase 5: Project Documentation & Submission**

In this part you will document your project and prepare it for submission.

Document the product sales analysis project and prepare it for submission.

**Objective:** The project aims to optimize inventory management and marketing strategies for a retail business. The goal is to enhance inventory accuracy, reduce carrying costs, and improve the effectiveness of marketing efforts to boost sales and customer satisfaction.

**Design Thinking Process:**

1. **Empathize:** Understand current inventory and marketing strategies, identify issues, and gather historical data.
2. **Define:** Define the project's objectives, including reducing overstock and stockouts, improving customer engagement, and increasing marketing effectiveness.
3. **Ideate:** Brainstorm data sources and analysis techniques.
4. **Prototype:** Develop a data analysis plan and visualize insights using Python and relevant libraries.
5. **Test:** Analyze data, validate prototypes, and make necessary adjustments.
6. **Implement:** Implement insights to guide inventory and marketing strategies.

**Development Phases:**

1. **Data Collection and Preprocessing:**
   * Gather historical sales, inventory, and marketing data.
   * Clean, preprocess, and consolidate the data, addressing missing values and outliers.
2. **Data Analysis and Visualization with Python:**
   * Use Python with Pandas, NumPy, and Matplotlib to analyze and visualize data.
   * Analyze sales trends, inventory turnover rates, marketing campaign performance, and customer behavior.
3. **Deriving Actionable Insights with Python:**
   * Identify products with high sales and low turnover, indicating overstock.
   * Detect items with high demand but low inventory, indicating stockouts.
   * Segment customers based on preferences, demographics, and buying patterns.
   * Evaluate past marketing campaigns for their impact on sales.
4. **Optimizing Inventory Management with Python:**
   * Implement reorder points and safety stock levels based on insights to reduce overstock and stockouts.
   * Automate inventory replenishment processes to maintain optimal stock levels.
5. **Enhancing Marketing Strategies with Python:**
   * Tailor marketing campaigns to specific customer segments based on insights.
   * Schedule promotions and advertisements during peak sales periods identified in the analysis.
   * Allocate marketing resources effectively by focusing on channels and strategies with the most impact.

**Python code:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

inventory\_data=pd.read\_csv('inventory.csv')

sales\_data=pd.read\_csv('sales.csv')

marketing\_data = pd.read\_csv('marketing.csv')

sales\_by\_product=sales\_data.groupby('product\_id')['quantity\_sold'].sum()

top\_selling\_products=sales\_by\_product.nlargest(10)

top\_selling\_products.plot(kind='bar', title='Top Selling Products')

plt.xlabel('Product ID')

plt.ylabel('Quantity Sold')

plt.show()

inventory\_data['inventory\_turnover']=inventory\_data['quantity\_sold']/ inventory\_data['opening\_stock']

low\_turnover\_products=inventory\_data[inventory\_data['inventory\_turnover'] < 0.2]

low\_turnover\_products.plot(kind='bar',x='product\_id',y='inventory\_turnover', title='Low Inventory Turnover Products')

plt.xlabel('Product ID')

plt.ylabel('Inventory Turnover')

plt.show()

marketing\_data['conversion\_rate']=marketing\_data['converted\_customers']/ marketing\_data['total\_customers']

successful\_campaigns=marketing\_data[marketing\_data['conversion\_rate'] > 0.1]

successful\_campaigns.plot(kind='bar', x='campaign\_name', y='conversion\_rate', title='Successful Marketing Campaigns')

plt.xlabel('Campaign Name')

plt.ylabel('Conversion Rate')

plt.show()

for product\_id in low\_turnover\_products['product\_id']:

reorder\_point=int(inventory\_data[inventory\_data['product\_id']== product\_id]['opening\_stock'] \* 0.2)

print(f"Product {product\_id}: Set reorder point to {reorder\_point}")

for campaign\_name in successful\_campaigns['campaign\_name']:

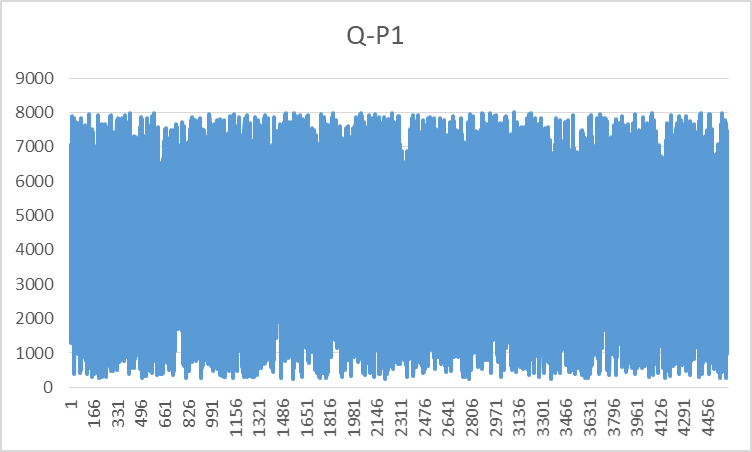
print(f"Launch additional marketing efforts for the {campaign\_name} campaign")

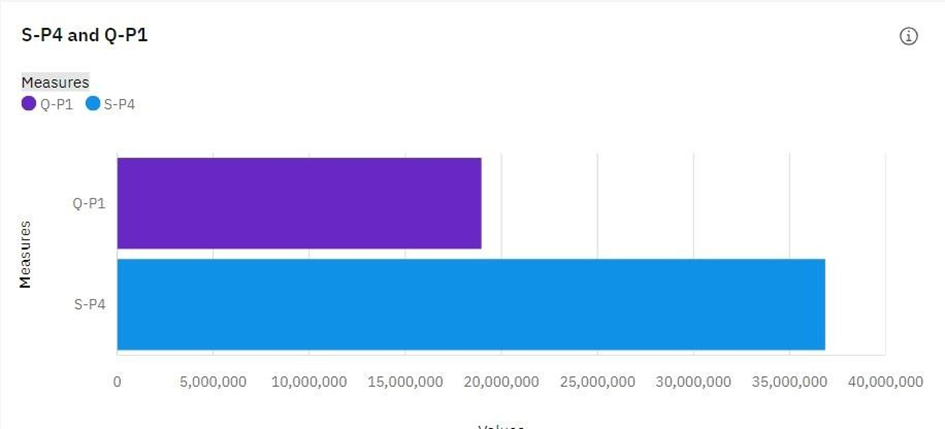
marketing\_channels = marketing\_data['channel'].unique()

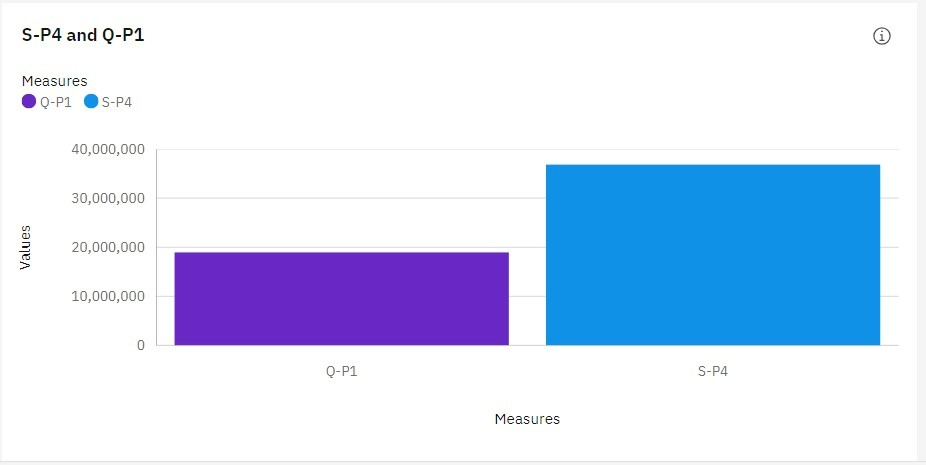
resource\_allocation = {channel: 10000 for channel in marketing\_channels}

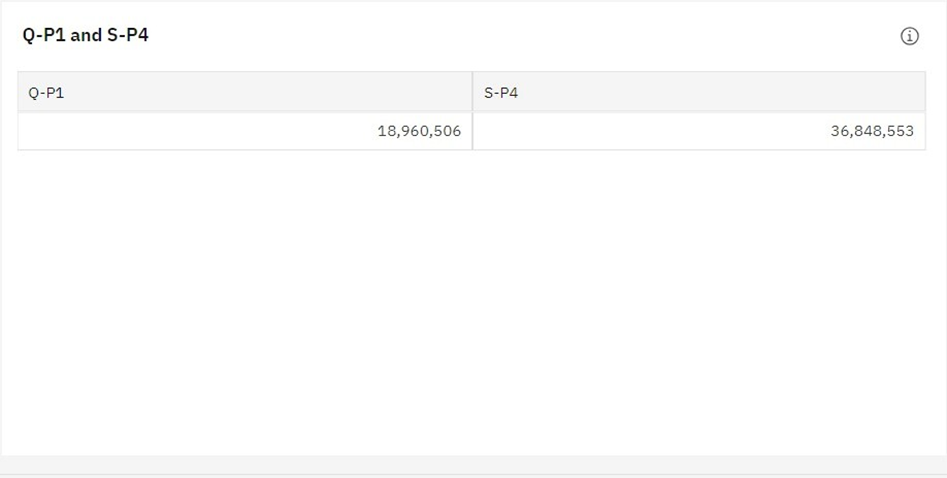
print(f"Allocate marketing resources: {resource\_allocation}")

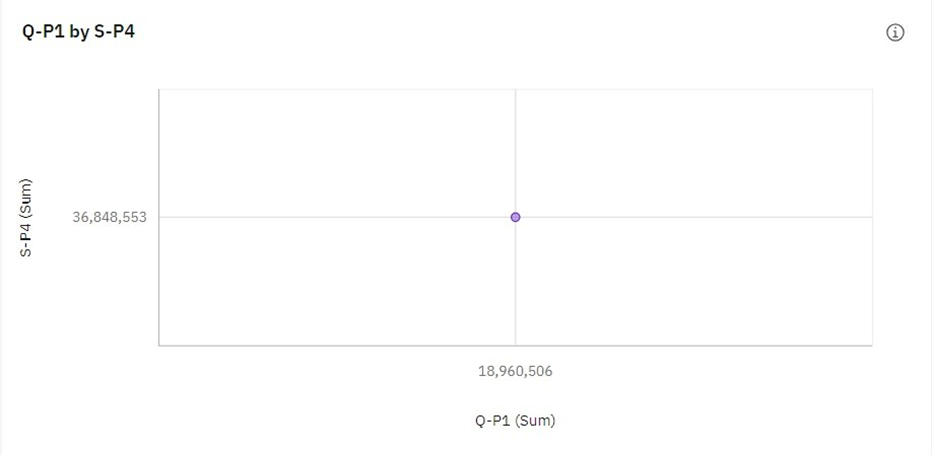
**output:**

****

****

****

****

****