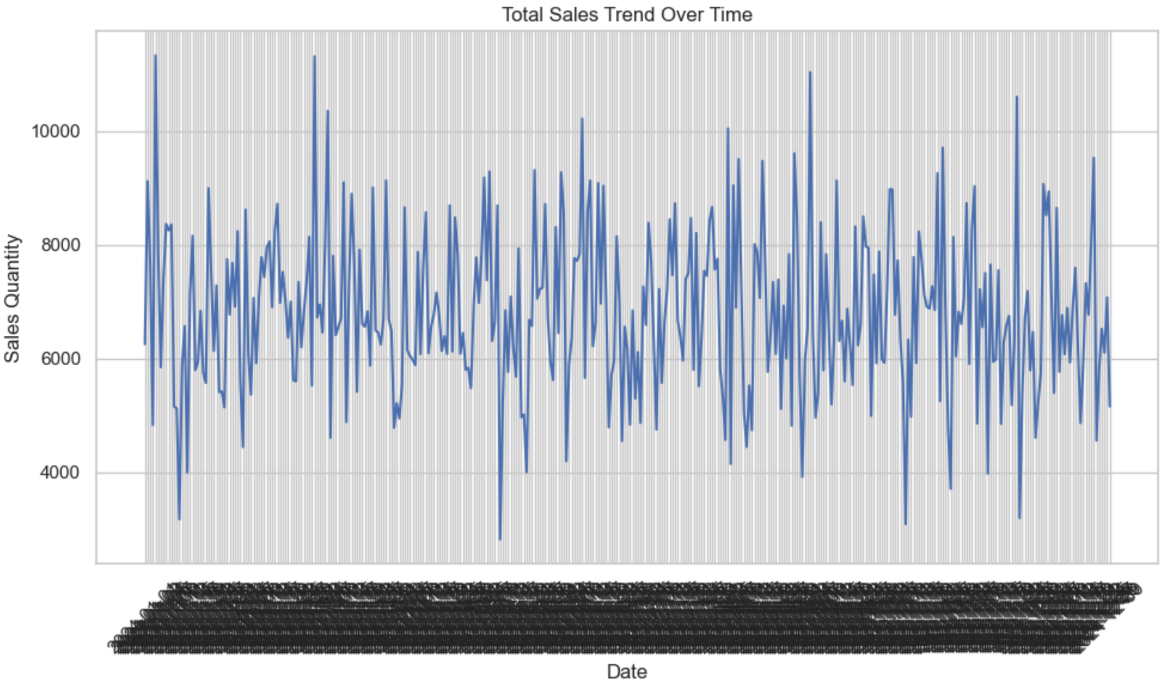
In [5]:

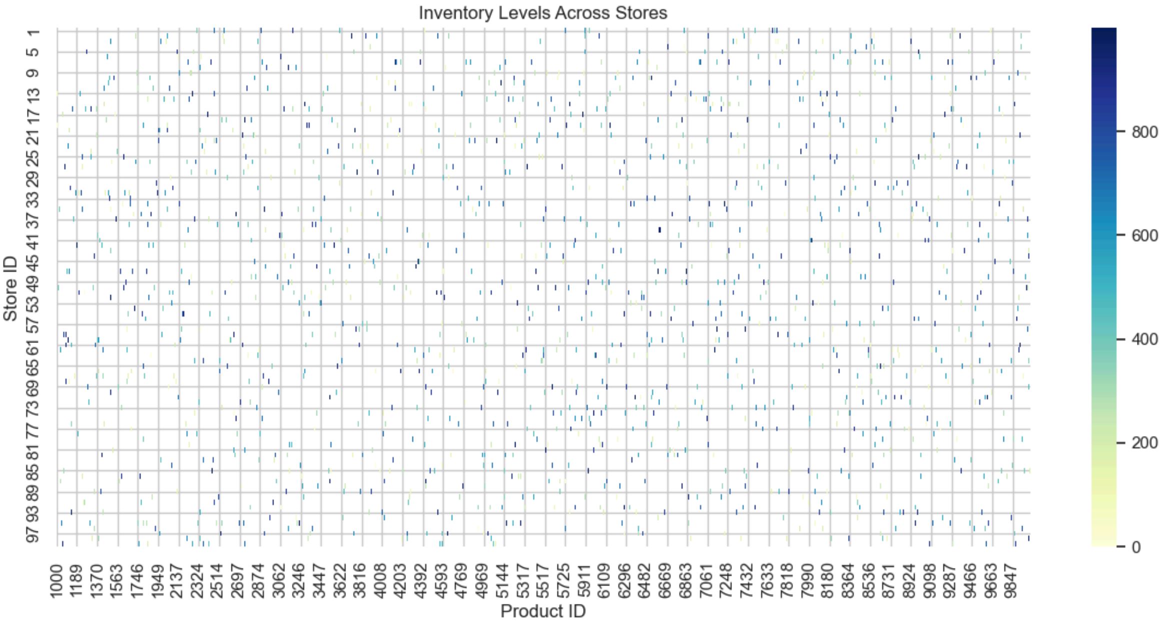
```
# ========================================
# Milestone 2: Retail Optimization Workflow
# Cleaning + EDA + Illustrations
# ========================================

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# -------------------------------
# 1. Setup
# -------------------------------
sns.set(style='whitegrid')  # Plot style

# -------------------------------
# 2. Load Datasets
# -------------------------------
df_demand = pd.read_csv('../data/demand_forecasting.csv', parse_dates=['Date'],
df_inventory = pd.read_csv('../data/inventory_monitoring.csv')
df_pricing = pd.read_csv('../data/pricing_optimization.csv')

# Quick overview
print("Demand Dataset:")
print(df_demand.head())
print(df_demand.info())

print("Inventory Dataset:")
print(df_inventory.head())
print(df_inventory.info())

print("Pricing Dataset:")
print(df_pricing.head())
print(df_pricing.info())

# -------------------------------
# 3. Light Cleaning
# -------------------------------
# Check for missing values
print(df_demand.isnull().sum())
print(df_inventory.isnull().sum())
print(df_pricing.isnull().sum())

# Remove duplicate rows if any
df_demand.drop_duplicates(inplace=True)
df_inventory.drop_duplicates(inplace=True)
df_pricing.drop_duplicates(inplace=True)

# Ensure numeric columns are correct type
numeric_cols_demand = ['Sales Quantity', 'Price']
df_demand[numeric_cols_demand] = df_demand[numeric_cols_demand].apply(pd.to_nume

numeric_cols_inventory = ['Stock Levels','Supplier Lead Time (days)','Stockout F
df_inventory[numeric_cols_inventory] = df_inventory[numeric_cols_inventory].appl

numeric_cols_pricing = ['Price','Competitor Prices','Discounts','Sales Volume','
df_pricing[numeric_cols_pricing] = df_pricing[numeric_cols_pricing].apply(pd.to_

# -------------------------------
# 4. Exploratory Data Analysis (EDA) & Figures
# -------------------------------

# --- 4a. Sales Trend Over Time ---

# Aggregating total sales by date
sales_trend = df_demand.groupby('Date')['Sales Quantity'].sum().reset_index()

# Plotting the sales trend
plt.figure(figsize=(10,6))
sns.lineplot(data=sales_trend, x='Date', y='Sales Quantity')
plt.title('Total Sales Trend Over Time')
plt.xlabel('Date')
plt.ylabel('Sales Quantity')
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig('../figures/sales_trend.png', dpi=300)
plt.show()

# --- 4b. Inventory Levels Across Stores (pivot_table fix) ---

# Pivot table of inventory by Store and Product
inventory_pivot = df_inventory.pivot_table(
    index='Store ID',
    columns='Product ID',
    values='Stock Levels',
    aggfunc='mean'  # Handles duplicates
)

# Plotting the heatmap
plt.figure(figsize=(12,6))
sns.heatmap(inventory_pivot, cmap='YlGnBu')
plt.title('Inventory Levels Across Stores')
plt.xlabel('Product ID')
plt.ylabel('Store ID')
plt.tight_layout()
plt.savefig('../figures/inventory_heatmap.png', dpi=300)
plt.show()

# --- 4c. Price vs Sales Relationship (duplicate column fix) ---

# Merge demand and pricing datasets
df_merged = pd.merge(
    df_demand[['Product ID','Store ID','Sales Quantity','Price']],
    df_pricing[['Product ID','Store ID','Price']],
    on=['Product ID','Store ID'],
    how='left',
    suffixes=('_demand','_pricing')
)

# Scatterplot: Price vs Sales Quantity
plt.figure(figsize=(10,6))
sns.scatterplot(data=df_merged, x='Price_demand', y='Sales Quantity')
plt.title('Price vs Sales Quantity')
plt.xlabel('Price')
plt.ylabel('Sales Quantity')
plt.tight_layout()
plt.savefig('../figures/price_vs_sales.png', dpi=300)
plt.show()
```

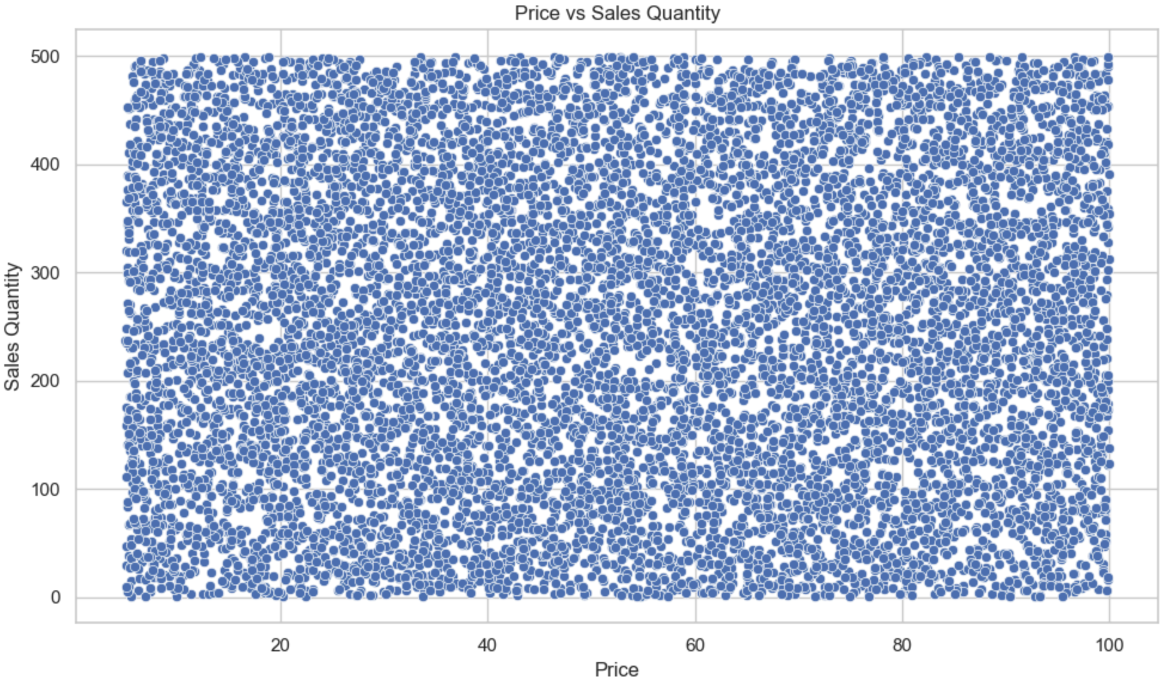
```
# --- 4d. Correlation Heatmap ---

# Subset inventory and pricing datasets
df_inventory_small = df_inventory[['Product ID','Store ID','Stock Levels']]
df_pricing_small = df_pricing[['Product ID','Store ID','Price','Discounts','Elas

# Merge all datasets
df_all = pd.merge(
    df_demand[['Product ID','Store ID','Sales Quantity','Price']],
    df_inventory_small, on=['Product ID','Store ID'], how='left'
)
df_all = pd.merge(df_all, df_pricing_small, on=['Product ID','Store ID'], how='l

# Correlation heatmap
plt.figure(figsize=(10,8))
sns.heatmap(df_all.corr(), annot=True, fmt=".2f", cmap='coolwarm')
plt.title('Correlation Heatmap of Sales, Inventory, and Pricing Features')
plt.tight_layout()
plt.savefig('../figures/correlation_heatmap.png', dpi=300)
plt.show()

# --- 4e. Promotions vs Sales Bar Chart---

# Average sales by promotion status
promo_sales = df_demand.groupby('Promotions')['Sales Quantity'].mean().reset_ind

# Bar chart: Promotions vs Average Sales
plt.figure(figsize=(6,4))
sns.barplot(data=promo_sales, x='Promotions', y='Sales Quantity', palette='paste
plt.title('Average Sales with and without Promotions')
plt.ylabel('Average Sales Quantity')
plt.tight_layout()
plt.savefig('../figures/promotion_sales.png', dpi=300)
plt.show()
```

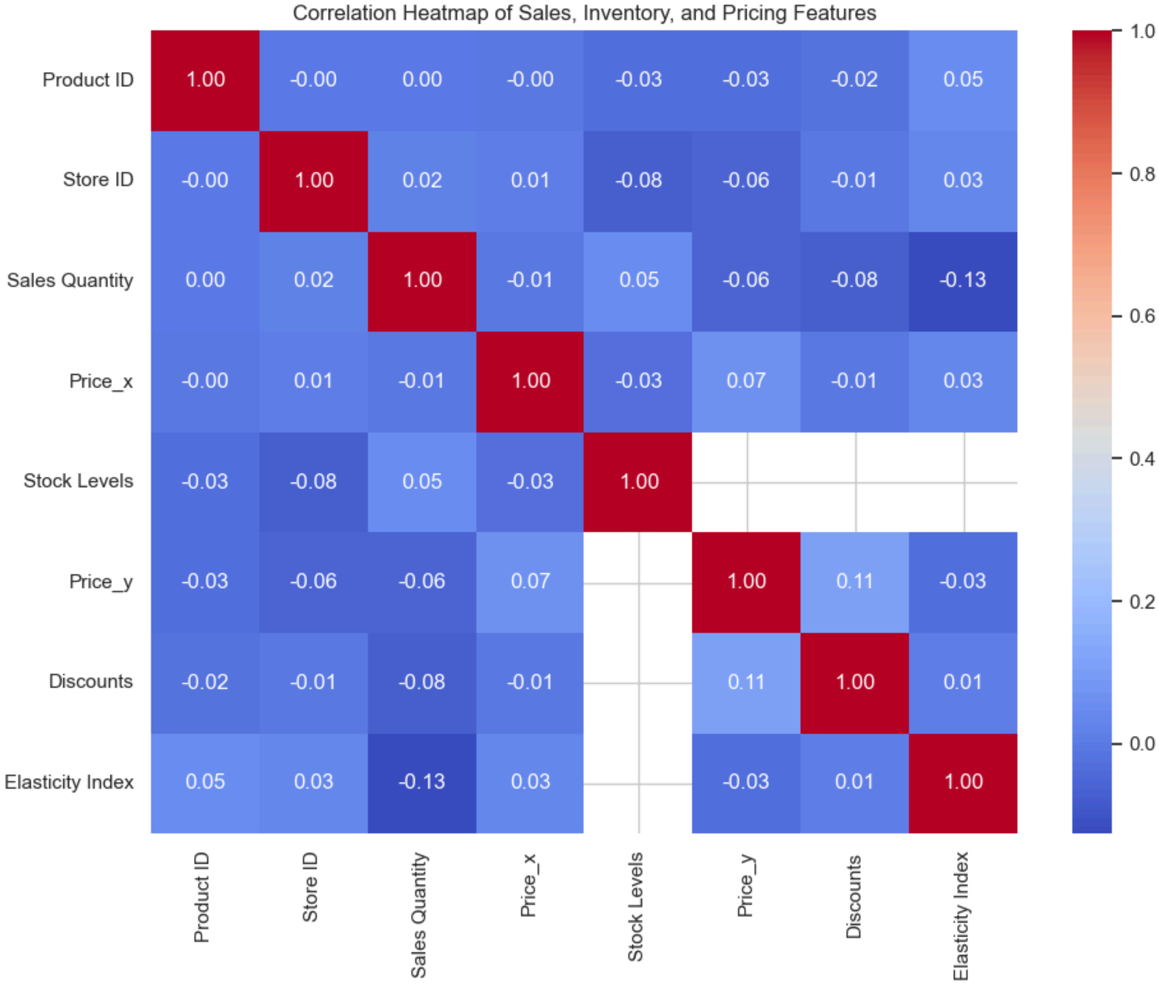
```
Demand Dataset:
   Product ID        Date  Store ID  Sales Quantity  Price Promotions  \
0        4277  2024-01-03        48             330  24.38         No   
1        5540  2024-04-29        10             334  74.98        Yes   
2        5406  2024-01-11        67             429  24.83        Yes   
3        5617  2024-04-04        17             298  13.41         No   
4        3480  2024-12-14        33             344  94.96        Yes   

  Seasonality Factors    External Factors Demand Trend Customer Segments  
0            Festival  Competitor Pricing   Increasing           Regular  
1             Holiday             Weather       Stable           Premium  
2             Holiday  Economic Indicator   Decreasing           Premium  
3                 NaN  Economic Indicator       Stable           Regular  
4            Festival             Weather   Increasing           Regular  
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 10 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Product ID           10000 non-null  int64  
 1   Date                 10000 non-null  object 
 2   Store ID             10000 non-null  int64  
 3   Sales Quantity       10000 non-null  int64  
 4   Price                10000 non-null  float64
 5   Promotions           10000 non-null  object 
 6   Seasonality Factors  6685 non-null   object 
 7   External Factors     7574 non-null   object 
 8   Demand Trend         10000 non-null  object 
 9   Customer Segments    10000 non-null  object 
dtypes: float64(1), int64(3), object(6)
memory usage: 781.4+ KB
None
Inventory Dataset:
   Product ID  Store ID  Stock Levels  Supplier Lead Time (days)  \
0        9286        16           700                         10   
1        2605        60            82                         11   
2        2859        55           145                         25   
3        2374        24           151                         17   
4        7678         5           714                         12   

   Stockout Frequency  Reorder Point Expiry Date  Warehouse Capacity  \
0                  14            132  2024-01-15                1052   
1                   1            127  2024-12-16                1262   
2                  14            192  2024-04-30                1457   
3                   6             19  2024-12-16                2944   
4                   2             21  2024-08-05                3739   

   Order Fulfillment Time (days)  
0                              6  
1                              9  
2                             12  
3                              3  
4                              7  
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 9 columns):
 #   Column                         Non-Null Count  Dtype  
---  ------                         --------------  -----  
 0   Product ID                     10000 non-null  int64  
 1   Store ID                       10000 non-null  int64  
 2   Stock Levels                   10000 non-null  int64 
 3   Supplier Lead Time (days)      10000 non-null  int64 
 4   Stockout Frequency             10000 non-null  int64 
 5   Reorder Point                  10000 non-null  int64 
 6   Expiry Date                    10000 non-null  object
 7   Warehouse Capacity             10000 non-null  int64 
 8   Order Fulfillment Time (days)  10000 non-null  int64 
dtypes: int64(8), object(1)
memory usage: 703.3+ KB
None
Pricing Dataset:
   Product ID  Store ID  Price  Competitor Prices  Discounts  Sales Volume  \
0        9502        13  31.61              56.14      19.68           255   
1        2068        77  35.51              63.04      16.88             5   
2        7103        59   6.54              30.61      10.86           184   
3        5288        19  13.61              15.94      45.28           337   
4        7212        66  62.68              30.64      33.48            80   

   Customer Reviews  Return Rate (%)  Storage Cost  Elasticity Index  
0                 3            13.33          6.72              1.78  
1                 3             1.50          8.38              1.67  
2                 3             9.44          3.86              2.46  
3                 1            15.11          8.80              0.88  
4                 3            19.62          9.74              1.00  
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 10 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Product ID         10000 non-null  int64  
 1   Store ID           10000 non-null  int64  
 2   Price              10000 non-null  float64
 3   Competitor Prices  10000 non-null  float64
 4   Discounts          10000 non-null  float64
 5   Sales Volume       10000 non-null  int64  
 6   Customer Reviews   10000 non-null  int64  
 7   Return Rate (%)    10000 non-null  float64
 8   Storage Cost       10000 non-null  float64
 9   Elasticity Index   10000 non-null  float64
dtypes: float64(6), int64(4)
memory usage: 781.4 KB
None
Product ID                0
Date                      0
Store ID                  0
Sales Quantity            0
Price                     0
Promotions                0
Seasonality Factors    3315
External Factors       2426
Demand Trend              0
Customer Segments         0
dtype: int64
Product ID                       0
Store ID                         0
Stock Levels                     0
Supplier Lead Time (days)        0
Stockout Frequency               0
Reorder Point                    0
Expiry Date                      0
```

```
Warehouse Capacity            0
Order Fulfillment Time (days)    0
dtype: int64
Product ID           0
Store ID             0
Price                0
Competitor Prices    0
Discounts            0
Sales Volume         0
Customer Reviews     0
Return Rate (%)      0
Storage Cost         0
Elasticity Index     0
dtype: int64
```

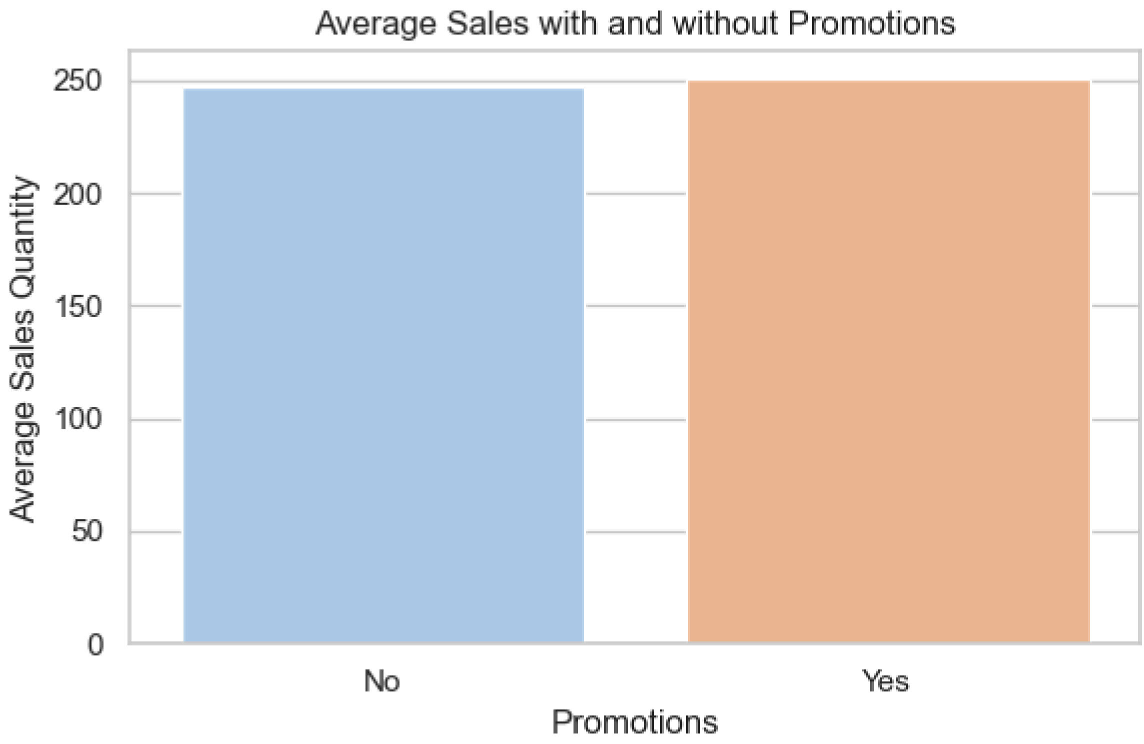








```
C:\Users\karth\AppData\Local\Temp\ipykernel_24968\3497958404.py:126: FutureWarnin
g:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.barplot(data=promo_sales, x='Promotions', y='Sales Quantity', palette='past
el')
```


Average Sales with and without Promotions
Average Sales Quantity
250
200
150
100
50
0
No
Yes
Promotions

```
In [ ]:
```