In [55]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
import sqlite3
from scipy.stats import ttest_ind
import scipy.stats as stats
warnings.filterwarnings('ignore')
```

In [56]:

```
#loading datasets
conn = sqlite3.connect('inventory.db')

#fetching vendor summary data
df = pd.read_sql_query("select * from vendor_sales_summary", conn)
df.head()
```

Out[56]:

	VendorNumber	VendorName	Brand	Description	PurchasePrice	ActualPri
0	1128	BROWN-FORMAN CORP	1233	Jack Daniels No 7 Black	26.27	36.
1	4425	MARTIGNETTI COMPANIES	3405	Tito's Handmade Vodka	23.19	28.
2	17035	PERNOD RICARD USA	8068	Absolut 80 Proof	18.24	24.
3	3960	DIAGEO NORTH AMERICA INC	4261	Capt Morgan Spiced Rum	16.17	22.
4	3960	DIAGEO NORTH AMERICA INC	3545	Ketel One Vodka	21.89	29.

In [57]:

```
### Exploratory Data Analysis

# Previously, we examined the various tables in the database to identify key
# determine which ones should be included in the final analysis.

# In this phase of EDA, we will analyze the resultant table to gain insights
# This will help us understand data patterns, identify anomalies, and ensure
```

In [58]:

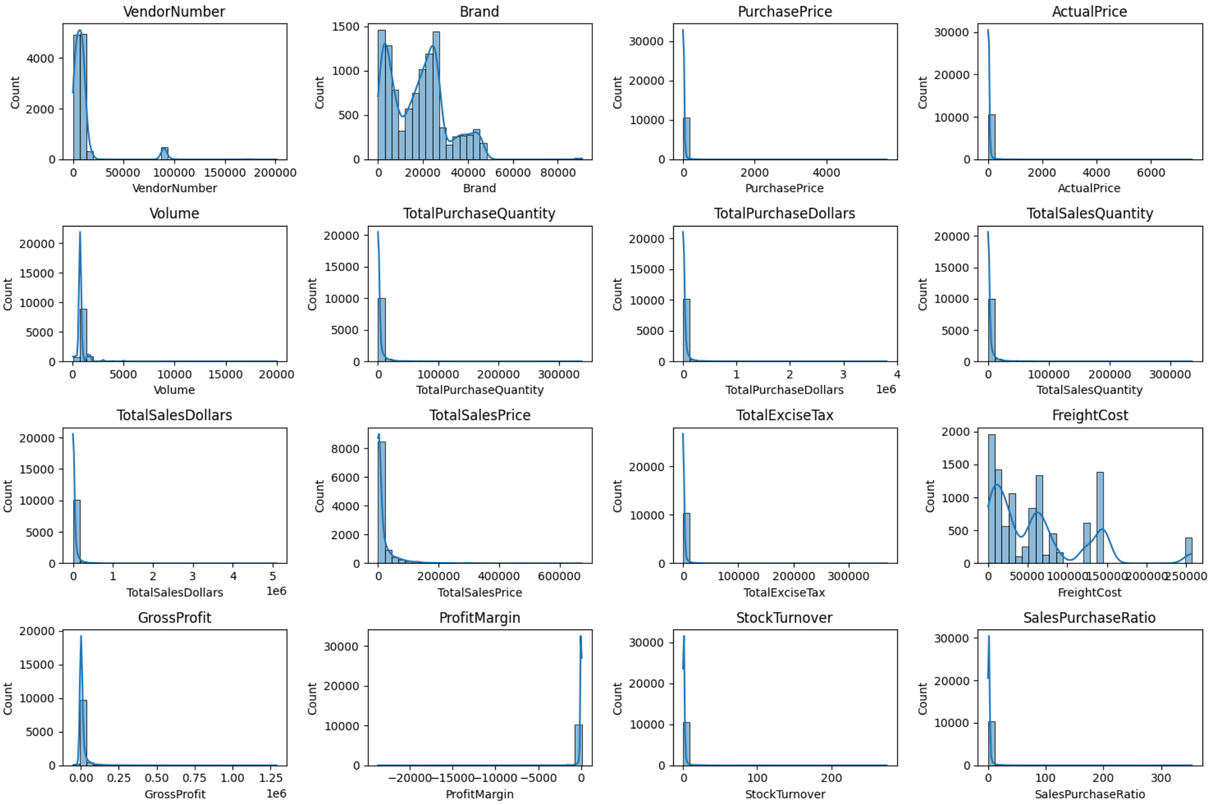
```
#summary statistics

df.describe()
```

Out[58]:

	VendorNumber	Brand	PurchasePrice	ActualPrice	Volun
count	10692.000000	10692.000000	10692.000000	10692.000000	10692.00000
mean	10650.649458	18039.228769	24.385303	35.643671	847.36055
std	18753.519148	12662.187074	109.269375	148.246016	664.30927
min	2.000000	58.000000	0.360000	0.490000	50.00000
25%	3951.000000	5793.500000	6.840000	10.990000	750.00000
50%	7153.000000	18761.500000	10.455000	15.990000	750.00000
75%	9552.000000	25514.250000	19.482500	28.990000	750.00000
max	201359.000000	90631.000000	5681.810000	7499.990000	20000.00000

In [59]:

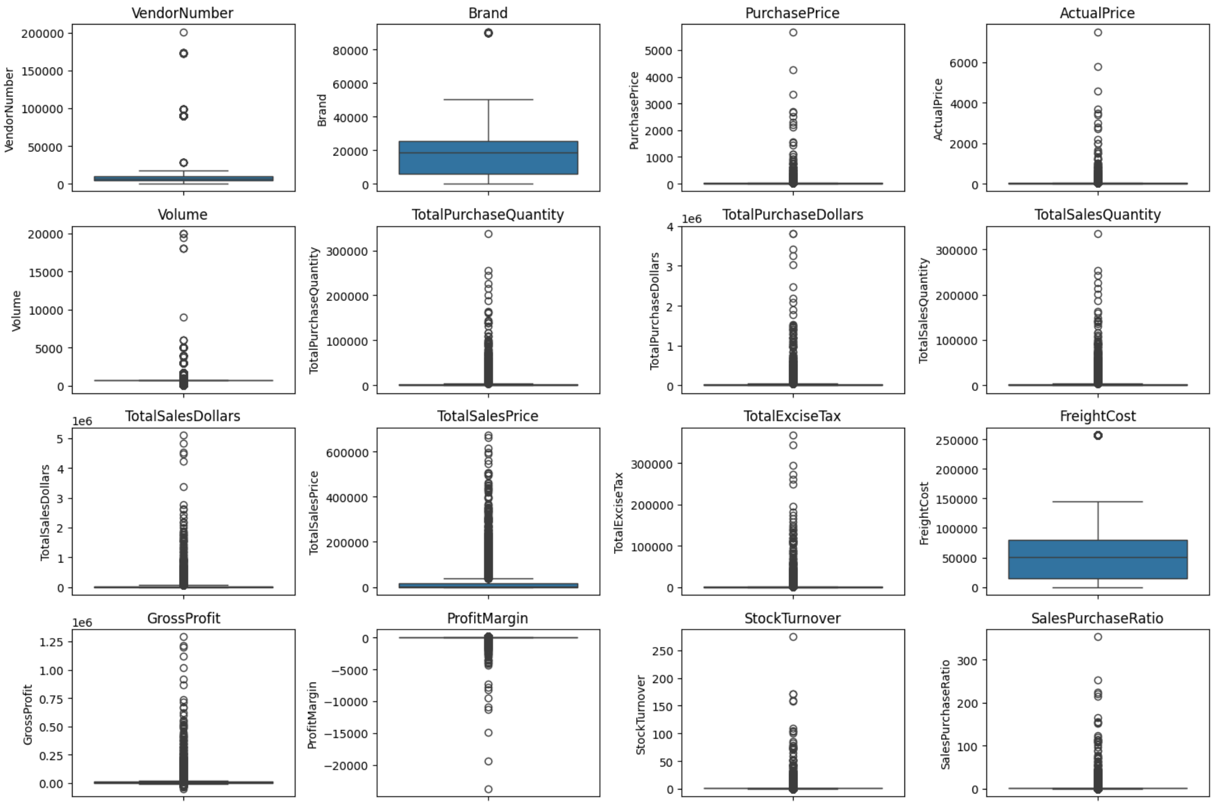
```
#distrubtion plot for numerical columns

numerical_cols = df.select_dtypes(include=np.number).columns

plt.figure(figsize=(15,10))
for i, col in enumerate(numerical_cols):
    plt.subplot(4,4,i+1)
    sns.histplot(df[col], kde = True, bins = 30)
    plt.title(col)
plt.tight_layout()
plt.show()
```



In [6]:

```
#distrubtion plot for numerical columns

numerical_cols = df.select_dtypes(include=np.number).columns
```

```
plt.figure(figsize=(15,10))
for i, col in enumerate(numerical_cols):
    plt.subplot(4,4,i+1)
    sns.boxplot(y=df[col])
    plt.title(col)
plt.tight_layout()
plt.show()
```



In [60]:

```
# Summary Statistics Insights:
# Negative & Zero Values:

# Gross Profit: Minimum value is -52,002.78, indicating losses. Some produc
# Profit Margin: Has a minimum of-00, which suggests cases where revenue is
# Total Sales Quantity & Sales Dollars: Minimum values are 0, meaning some p



# Outliers Indicated by High Standard Deviations:
# Purchase & Actual Prices: The max values (5,681.81 & 7,499.99) are signifi
# Freight Cost: Huge variation, from 0.09 to 257,032.07, suggests logistics
# Stock Turnover: Ranges from 0 to 274.5, implying some products sell extrem
```

In [61]:

```
df = pd.read_sql_query("""SELECT *
FROM vendor_sales_summary
WHERE GrossProfit > 0
AND ProfitMargin > 0
AND TotalSalesQuantity > 0
""", conn)
df
```

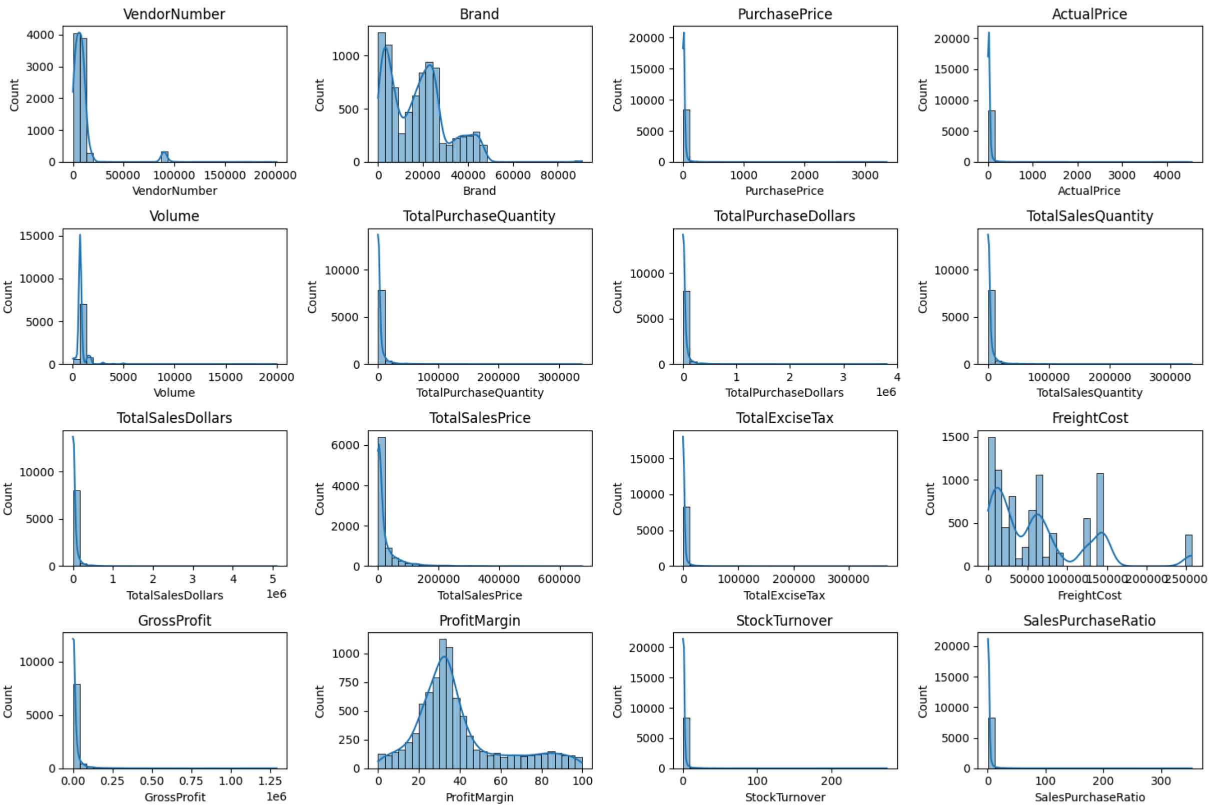
Out[61]:

	VendorNumber	VendorName	Brand	Description	PurchasePrice	Actu
0	1128	BROWN-FORMAN CORP	1233	Jack Daniels No 7 Black	26.27	
1	4425	MARTIGNETTI COMPANIES	3405	Tito's Handmade Vodka	23.19	
2	17035	PERNOD RICARD USA	8068	Absolut 80 Proof	18.24	
3	3960	DIAGEO NORTH AMERICA INC	4261	Capt Morgan Spiced Rum	16.17	
4	3960	DIAGEO NORTH AMERICA INC	3545	Ketel One Vodka	21.89	
...	...	...	...	...	...	
8559	9815	WINE GROUP INC	8527	Concannon Glen Ellen Wh Zin	1.32	
8560	8004	SAZERAC CO INC	5683	Dr McGillicuddy's Apple Pie	0.39	
8561	3924	HEAVEN HILL DISTILLERIES	9123	Deep Eddy Vodka	0.74	
8562	3960	DIAGEO NORTH AMERICA INC	6127	The Club Strawbry Margarita	1.47	
8563	7245	PROXIMO SPIRITS INC.	3065	Three Olives Grape Vodka	0.71	

8564 rows × 18 columns

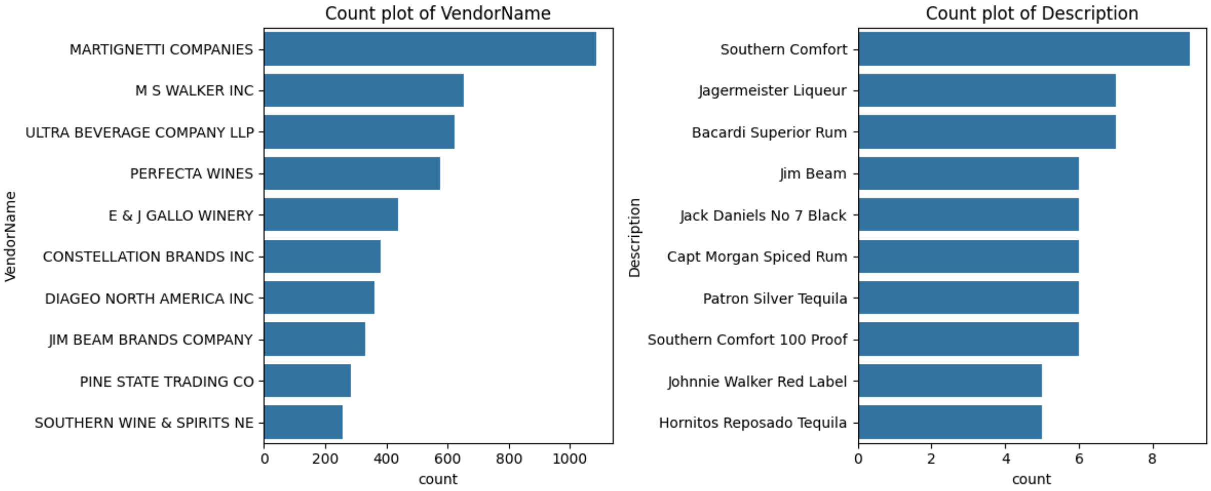
In [62]:

```python
#distrubtion plot for numerical columns

numerical_cols = df.select_dtypes(include=np.number).columns

plt.figure(figsize=(15,10))
for i, col in enumerate(numerical_cols):
    plt.subplot(4,4,i+1)
    sns.histplot(df[col], kde = True, bins = 30)
    plt.title(col)
plt.tight_layout()
plt.show()
```



In [64]:

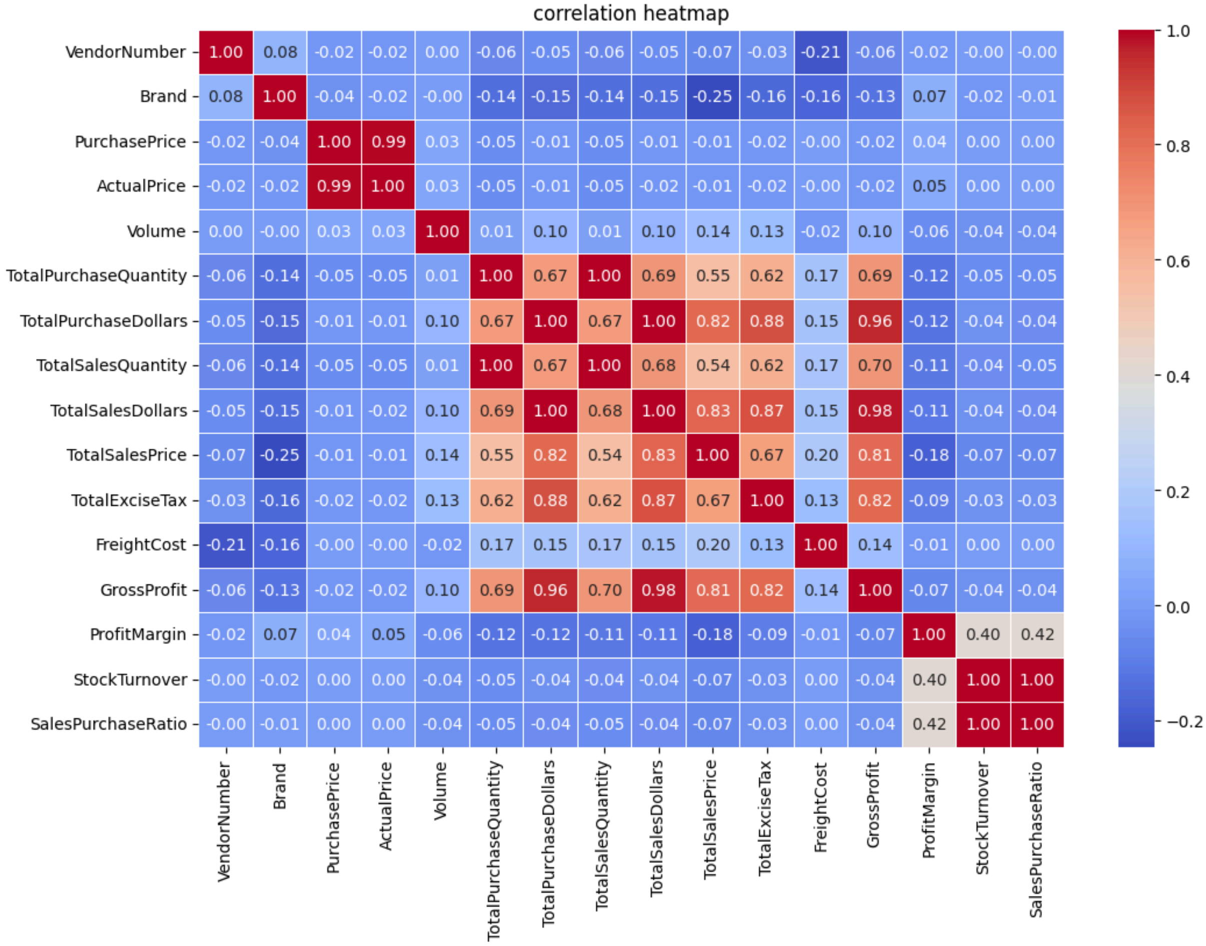
```
#counting plots for categorial columns
categorical_cals = ["VendorName","Description"]

plt.figure(figsize=(12,5))
for i, col in enumerate(categorical_cals):
    plt.subplot(1,2,i+1)
    sns.countplot(y=df[col], order=df[col].value_counts().index[:10]) #top 1
    plt.title(f"Count plot of {col}")
plt.tight_layout()
plt.show()
```



In [65]:

```
#corrrealation heatmap
plt.figure(figsize=(12,8))
correlation_matrix = df[numerical_cols].corr()
```

```
sns.heatmap(correlation_matrix, annot = True, fmt=".2f", cmap="coolwarm", li
plt.title("correlation heatmap")
plt.show()
```



In [66]:
```
# Correlation Insights

# Purchase Price has weak correlations with TotalSales Dollars (-0.012) and
# Strong correlation between total purchase quantity and total sales quantit
# Negative correlation between profit margin & total sales price (-0.179) su
# Stock Turnover has weak negative correlations with both GrossProfit (-0.03
```

In [12]:
```
# Data Analysis

# Identify Brands that needs Promotional or Pricing Adjustments which exhibi
```

In [67]:
```
brand_performance = df.groupby('Description').agg({
    'TotalSalesDollars':'sum',
    'ProfitMargin': 'mean'}).reset_index()
brand_performance
```

Out[67]:

	Description	TotalSalesDollars	ProfitMargin
0	(RI) 1	21519.09	18.060661
1	.nparalleled Svgn Blanc	1094.63	29.978166
2	10 Span Cab Svgn CC	2703.89	20.937612
3	10 Span Chard CC	3325.56	27.806445
4	10 Span Pnt Gris Monterey Cy	2082.22	32.226182
...	...	...	...
7702	Zorvino Vyds Sangiovese	10579.03	29.525675
7703	Zuccardi Q Malbec	1639.18	23.981503
7704	Zum Rsl	10857.34	32.675038
7705	Zwack Liqueur	227.88	16.653502
7706	von Buhl Jazz Rsl	1359.11	90.773374

7707 rows × 3 columns

In [68]:

```
low_sales_threshold = brand_performance['TotalSalesDollars'].quantile(0.15)
high_margin_threshold = brand_performance['ProfitMargin'].quantile(0.85)
```

In [69]:

```
low_sales_threshold
```

Out[69]:

```
np.float64(560.299)
```

In [71]:

```
high_margin_threshold
```

Out[71]:

```
np.float64(64.97017552750113)
```

In [72]:

```
#Filter brands with low sales but high profit margins
target_brands = brand_performance[
    (brand_performance['TotalSalesDollars']<= low_sales_threshold)&
    (brand_performance['ProfitMargin']>= high_margin_threshold)
]

print("Brand with low Sales but high profit margins : ")
display(target_brands.sort_values('TotalSalesDollars'))
```

```
Brand with low Sales but high profit margins :
```

	Description	TotalSalesDollars	ProfitMargin
6199	Santa Rita Organic Svgn Bl	9.99	66.466466
2369	Debauchery Pnt Nr	11.58	65.975820
2070	Concannon Glen Ellen Wh Zin	15.95	83.448276
2188	Crown Royal Apple	27.86	89.806174
6237	Sauza Sprklg Wild Berry Marg	27.96	82.153076
...	...	...	...
5074	Nanbu Bijin Southern Beauty	535.68	76.747312
2271	Dad's Hat Rye Whiskey	538.89	81.851584
57	A Bichot Clos Marechaudes	539.94	67.740860
6245	Sbragia Home Ranch Merlot	549.75	66.444748
3326	Goulee Cos d'Estournel 10	558.87	69.434752

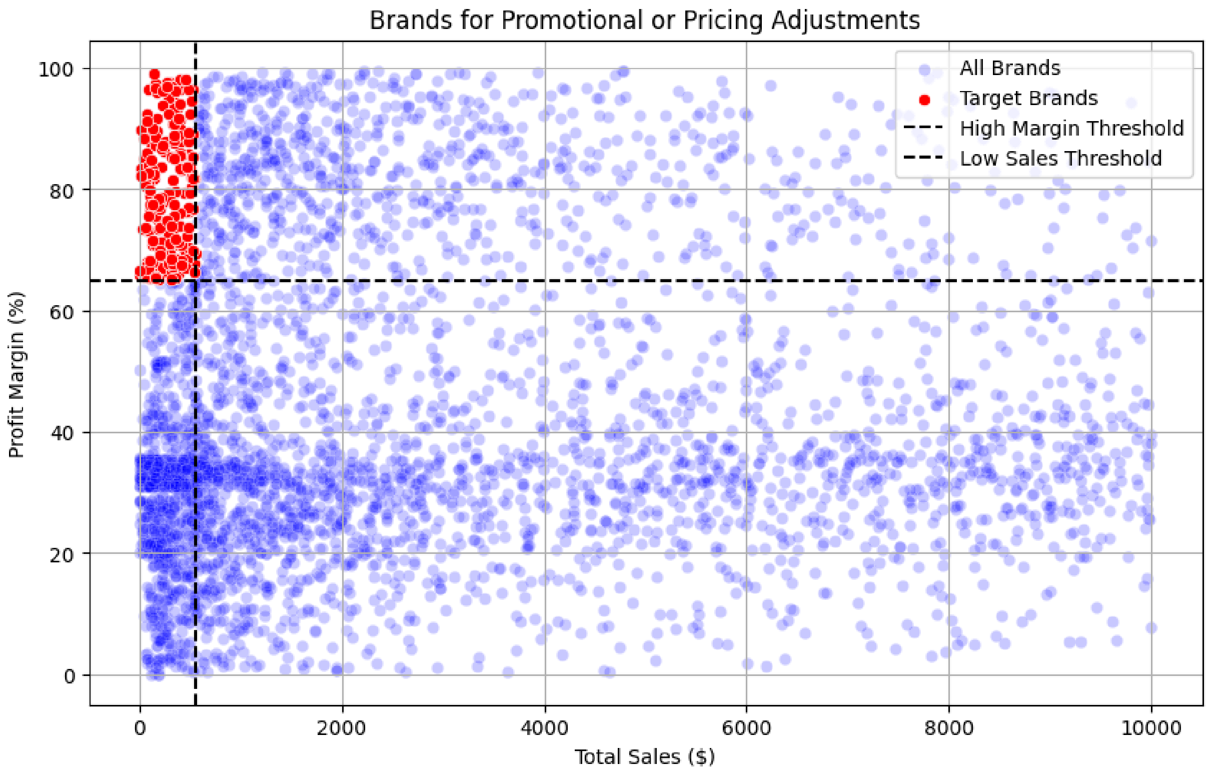
198 rows × 3 columns

In [74]:

```
brand_performance = brand_performance[brand_performance['TotalSalesDollars']
```

In [75]:

```
plt.figure(figsize=(10, 6))
sns.scatterplot(data=brand_performance, x='TotalSalesDollars', y='ProfitMarg
sns.scatterplot(data=target_brands, x='TotalSalesDollars', y='ProfitMargin',
plt.axhline(high_margin_threshold, linestyle='--', color='black', label="Hig
plt.axvline(low_sales_threshold, linestyle='--', color='black', label="Low S
plt.xlabel("Total Sales ($)")
plt.ylabel("Profit Margin (%)")
plt.title("Brands for Promotional or Pricing Adjustments")
plt.legend()
plt.grid(True)
plt.show()
```



In [76]:

```python
def format_dollars(value):
    if value >=1_00_000:
        return f"{value/1_000_000:.2f}M"
    elif value >=1_000:
        return f"{value/1_000:.2f}K"
    else:
        return str(value)
```

In [71]:

```python
# which vendors and brands demonstrate the highest sales performance ?
```

In [77]:

```python
top_vendors = df.groupby("VendorName")["TotalSalesDollars"].sum().nlargest(1
top_brands = df.groupby("Description")["TotalSalesDollars"].sum().nlargest(1
top_vendors.apply(lambda x : format_dollars(x))
```

Out[77]:

```
VendorName
DIAGEO NORTH AMERICA INC      67.99M
MARTIGNETTI COMPANIES         39.33M
PERNOD RICARD USA             32.06M
JIM BEAM BRANDS COMPANY       31.42M
BACARDI USA INC               24.85M
CONSTELLATION BRANDS INC      24.22M
E & J GALLO WINERY            18.40M
BROWN-FORMAN CORP             18.25M
ULTRA BEVERAGE COMPANY LLP    16.50M
M S WALKER INC                14.71M
Name: TotalSalesDollars, dtype: object
```

In [78]:

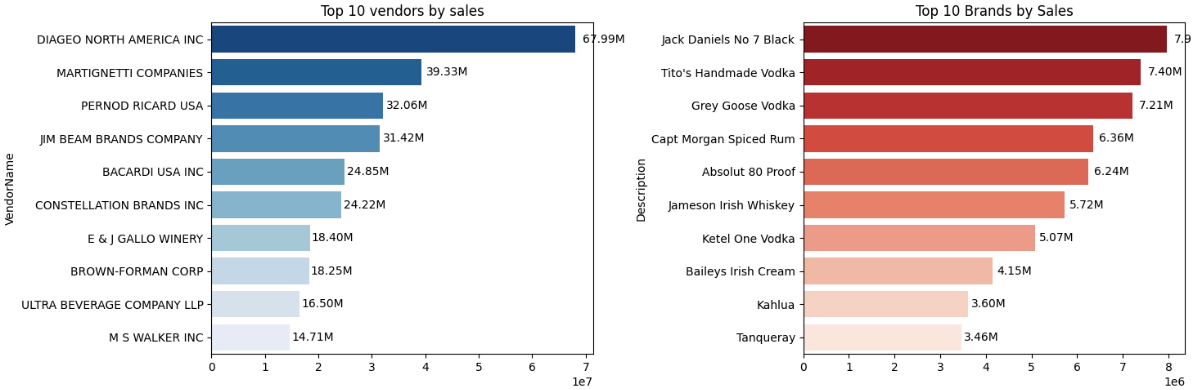
```python
top_brands.apply(lambda x : format_dollars(x))
```

Out[78]:

```
Description
Jack Daniels No 7 Black    7.96M
Tito's Handmade Vodka      7.40M
Grey Goose Vodka           7.21M
Capt Morgan Spiced Rum     6.36M
Absolut 80 Proof           6.24M
Jameson Irish Whiskey      5.72M
Ketel One Vodka            5.07M
Baileys Irish Cream        4.15M
Kahlua                     3.60M
Tanqueray                  3.46M
Name: TotalSalesDollars, dtype: object
```

In [79]:

```
plt.figure(figsize=(15,5))

#plot for top vendors

plt.subplot(1,2,1)
ax1 = sns.barplot(y=top_vendors.index, x=top_vendors.values, palette="Blues_
plt.title("Top 10 vendors by sales")


for bar in ax1.patches:
    ax1.text(bar.get_width()+(bar.get_width()* 0.02),
             bar.get_y() + bar.get_height()/2,
             format_dollars(bar.get_width()),
             ha='left', va='center', fontsize=10, color='black')

#Plot for Top Brands
plt.subplot(1, 2, 2)
ax2 = sns.barplot(y=top_brands.index.astype(str), x=top_brands.values, palet
plt.title("Top 10 Brands by Sales")

for bar in ax2.patches:
    ax2.text(bar.get_width() + (bar.get_width() * 0.02),
        bar.get_y() + bar.get_height() / 2,
        format_dollars(bar.get_width()),
        ha='left', va='center', fontsize=10, color='black')
plt.tight_layout()
plt.show()
```



In [23]:

```
# Which vendors contribute the most to total purchase dollars
```

In [ ]:

```

```

In [80]:

```
vendor_performance = df.groupby('VendorName').agg({
    'TotalPurchaseDollars': 'sum',
    'GrossProfit': 'sum',
    'TotalSalesDollars': 'sum'
}).reset_index()  # <-- this brings VendorName back as a column

# Optional: reorder columns to ensure VendorName is first (though it already
vendor_performance = vendor_performance[['VendorName', 'TotalPurchaseDollars

vendor_performance
```

Out[80]:

	VendorName	TotalPurchaseDollars	GrossProfit	TotalSalesDollars
0	ADAMBA IMPORTS INTL INC	446.16	258.37	704.53
1	ALISA CARR BEVERAGES	25698.12	78772.82	104470.94
2	ALTAMAR BRANDS LLC	11706.20	4000.61	15706.81
3	AMERICAN SPIRITS EXCHANGE	934.08	577.08	1511.16
4	AMERICAN VINTAGE BEVERAGE	104435.68	35167.85	139603.53
...	...	...	...	...
114	WEIN BAUER INC	42694.64	13522.49	56217.13
115	WESTERN SPIRITS BEVERAGE CO	298416.86	106837.97	405254.83
116	WILLIAM GRANT & SONS INC	5876538.26	1693337.94	7569876.20
117	WINE GROUP INC	5203801.17	3100242.11	8304043.28
118	ZORVINO VINEYARDS	86122.71	38066.88	124189.59

119 rows × 4 columns

In [81]:

```
vendor_performance['PurchaseContribution%'] = (vendor_performance['TotalPurc
vendor_performance
```

Out[81]:

	VendorName	TotalPurchaseDollars	GrossProfit	TotalSalesDollars	Purch
0	ADAMBA IMPORTS INTL INC	446.16	258.37	704.53	
1	ALISA CARR BEVERAGES	25698.12	78772.82	104470.94	
2	ALTAMAR BRANDS LLC	11706.20	4000.61	15706.81	
3	AMERICAN SPIRITS EXCHANGE	934.08	577.08	1511.16	
4	AMERICAN VINTAGE BEVERAGE	104435.68	35167.85	139603.53	
...	...	...	...	...	
114	WEIN BAUER INC	42694.64	13522.49	56217.13	
115	WESTERN SPIRITS BEVERAGE CO	298416.86	106837.97	405254.83	
116	WILLIAM GRANT & SONS INC	5876538.26	1693337.94	7569876.20	
117	WINE GROUP INC	5203801.17	3100242.11	8304043.28	
118	ZORVINO VINEYARDS	86122.71	38066.88	124189.59	

119 rows × 5 columns

In [82]:

```
vendor_performance = vendor_performance.sort_values('PurchaseContribution%',
```

In [83]:

```
vendor_performance
```

Out[83]:

	VendorName	TotalPurchaseDollars	GrossProfit	TotalSalesDollars	Pur
25	DIAGEO NORTH AMERICA INC	50097226.16	17892873.26	67990099.42	
57	MARTIGNETTI COMPANIES	25502095.83	13828263.53	39330359.36	
68	PERNOD RICARD USA	23851164.17	8212032.02	32063196.19	
46	JIM BEAM BRANDS COMPANY	23494304.32	7928716.14	31423020.46	
6	BACARDI USA INC	17432020.26	7422796.88	24854817.14	
...	...	...	...	...	
33	FANTASY FINE WINES CORP	128.64	198.95	327.59	
107	UNCORKED	118.74	58.20	176.94	
85	SILVER MOUNTAIN CIDERS	77.18	265.33	342.51	
16	CAPSTONE INTERNATIONAL	54.64	192.23	246.87	
35	FLAVOR ESSENCE INC	17.00	1457.41	1474.41	

119 rows × 5 columns

In [84]:

```
top_vendors = vendor_performance.head(10)
top_vendors['TotalSalesDollars'] = top_vendors['TotalSalesDollars'].apply(fo
top_vendors['TotalPurchaseDollars'] = top_vendors['TotalPurchaseDollars'].ap
top_vendors['GrossProfit'] = top_vendors['GrossProfit'].apply(format_dollars
top_vendors
```

Out[84]:

	VendorName	TotalPurchaseDollars	GrossProfit	TotalSalesDollars	Purc
25	DIAGEO NORTH AMERICA INC	50.10M	17.89M	67.99M	
57	MARTIGNETTI COMPANIES	25.50M	13.83M	39.33M	
68	PERNOD RICARD USA	23.85M	8.21M	32.06M	
46	JIM BEAM BRANDS COMPANY	23.49M	7.93M	31.42M	
6	BACARDI USA INC	17.43M	7.42M	24.85M	
20	CONSTELLATION BRANDS INC	15.27M	8.95M	24.22M	
11	BROWN-FORMAN CORP	13.24M	5.01M	18.25M	
30	E & J GALLO WINERY	12.07M	6.33M	18.40M	
106	ULTRA BEVERAGE COMPANY LLP	11.17M	5.34M	16.50M	
53	M S WALKER INC	9.76M	4.94M	14.71M	

In [93]:

```
top_vendors['Cumulative_Contribution'] = top_vendors['PurchaseContribution%'
top_vendors['PurchaseContribution%'] = top_vendors['PurchaseContribution%']

#diving it by 100 as there was an error in ingestion

top_vendors
```

Out[93]:

	VendorName	TotalPurchaseDollars	GrossProfit	TotalSalesDollars	Purc
25	DIAGEO NORTH AMERICA INC	50.10M	17.89M	67.99M	
57	MARTIGNETTI COMPANIES	25.50M	13.83M	39.33M	
68	PERNOD RICARD USA	23.85M	8.21M	32.06M	
46	JIM BEAM BRANDS COMPANY	23.49M	7.93M	31.42M	
6	BACARDI USA INC	17.43M	7.42M	24.85M	
20	CONSTELLATION BRANDS INC	15.27M	8.95M	24.22M	
11	BROWN-FORMAN CORP	13.24M	5.01M	18.25M	
30	E & J GALLO WINERY	12.07M	6.33M	18.40M	
106	ULTRA BEVERAGE COMPANY LLP	11.17M	5.34M	16.50M	
53	M S WALKER INC	9.76M	4.94M	14.71M	

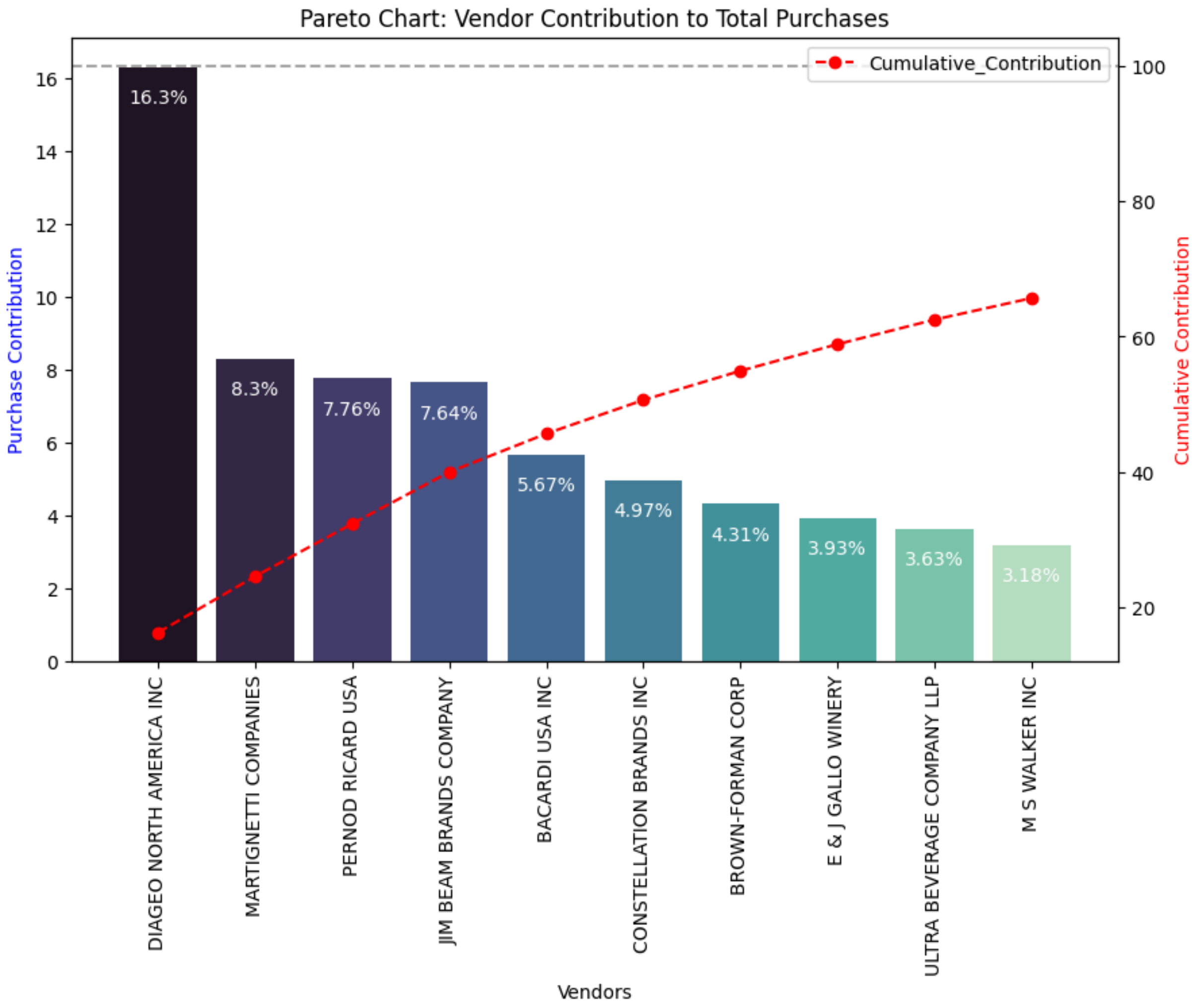
In [94]:

```
print(top_vendors.columns)
```

```
Index(['VendorName', 'TotalPurchaseDollars', 'GrossProfit',
       'TotalSalesDollars', 'PurchaseContribution%',
       'Cumulative_Contribution'],
      dtype='object')
```

In [104…

```
fig, ax1= plt.subplots(figsize=(10, 6))

#Bar plot for Purchase Contribution
sns.barplot(x=top_vendors['VendorName'], y=top_vendors['PurchaseContribution
for i, value in enumerate(top_vendors['PurchaseContribution%']):
    ax1.text(i,value-1, str(value)+'%', ha='center', fontsize=10, color='whi

#Line Plot for Cumulative Contributions
ax2=ax1.twinx()
ax2.plot(top_vendors['VendorName'], top_vendors['Cumulative_Contribution'],
ax1.set_xticklabels (top_vendors['VendorName'], rotation=90)
ax1.set_ylabel('Purchase Contribution', color='blue')
ax2.set_ylabel('Cumulative Contribution', color='red')
ax1.set_xlabel('Vendors')
ax1.set_title('Pareto Chart: Vendor Contribution to Total Purchases')
ax2.axhline(y=100, color='gray', linestyle='dashed', alpha=0.7)
ax2.legend(loc='upper right')

plt.show()
```



In [ ]:

```
# how much of total procurement is dependent on the top vendors?
```

In [98]:

```
print(f"Total purchase contribution of top 10 vendors is {round(top_vendors
```

```
Total purchase contribution of top 10 vendors is 65.69%)
```

In [99]:

```
top_vendors2 = top_vendors
```

In [102…

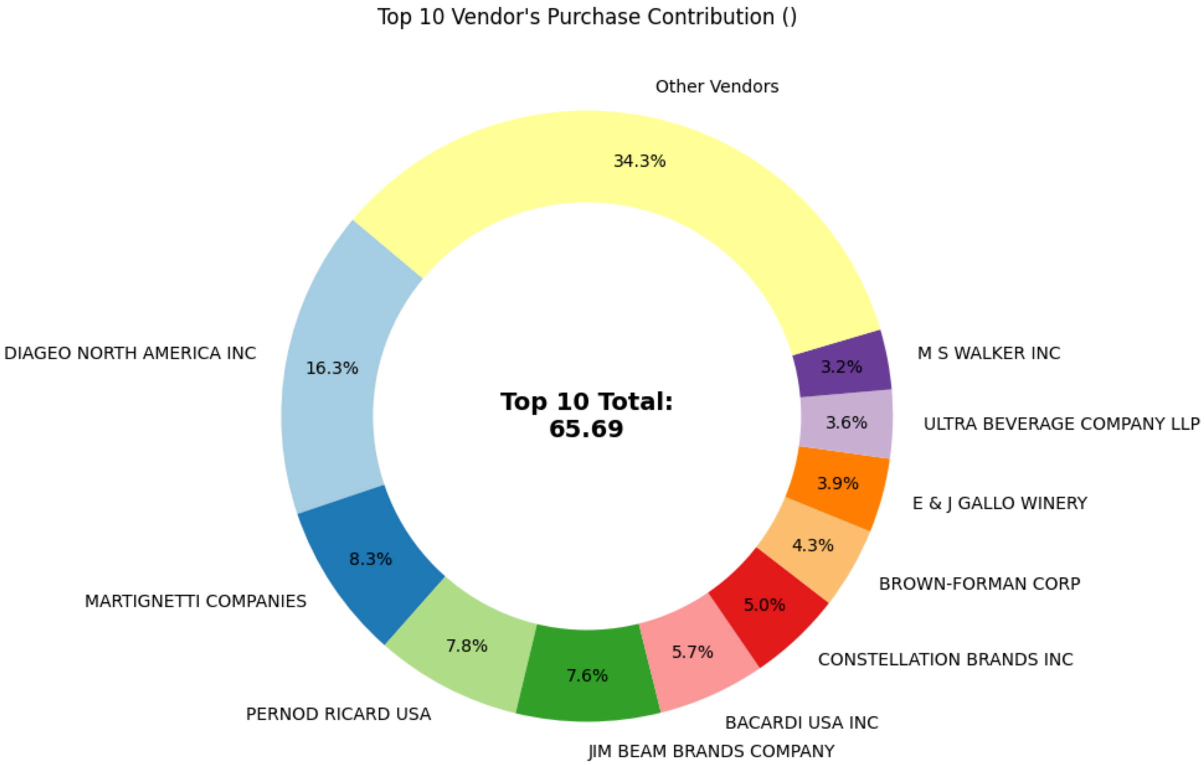
```
top_vendors2['Cumulative_Contribution'] = top_vendors2['PurchaseContributior
top_vendors2['PurchaseContribution%'] = top_vendors2['PurchaseContribution%'

#diving it by 100 as there was an error in ingestion

top_vendors2
```

Out[102…

	VendorName	TotalPurchaseDollars	GrossProfit	TotalSalesDollars	Purc
25	DIAGEO NORTH AMERICA INC	50.10M	17.89M	67.99M	
57	MARTIGNETTI COMPANIES	25.50M	13.83M	39.33M	
68	PERNOD RICARD USA	23.85M	8.21M	32.06M	
46	JIM BEAM BRANDS COMPANY	23.49M	7.93M	31.42M	
6	BACARDI USA INC	17.43M	7.42M	24.85M	
20	CONSTELLATION BRANDS INC	15.27M	8.95M	24.22M	
11	BROWN-FORMAN CORP	13.24M	5.01M	18.25M	
30	E & J GALLO WINERY	12.07M	6.33M	18.40M	
106	ULTRA BEVERAGE COMPANY LLP	11.17M	5.34M	16.50M	
53	M S WALKER INC	9.76M	4.94M	14.71M	

In [105…

```
vendors = list(top_vendors2['VendorName'].values)
purchase_contributions = list(top_vendors2['PurchaseContribution%'].values)
total_contribution =  sum(purchase_contributions)
remaining_contribution = 100 - total_contribution

#Append "Other Vendors" category
vendors.append("Other Vendors")
purchase_contributions.append(remaining_contribution)

#Donut Chart
fig, ax = plt.subplots(figsize=(8, 8))
wedges, texts, autotexts = ax.pie(purchase_contributions, labels=vendors, au


#Draw a white circle in the center to create a "donut" effect
centre_circle = plt.Circle((0, 0), 0.70, fc='white')
fig.gca().add_artist(centre_circle)


#Add Total Contribution annotation in the center
plt.text(0, 0, f"Top 10 Total:\n{total_contribution:.2f}", fontsize=14, font
plt.title("Top 10 Vendor's Purchase Contribution ()")

plt.show()
```



```
In [106…  # Does purchasing in bulk reduce the unit price and what is the optimal purc
```

```
In [ ]:
```

```
In [108…  df['UnitPurchasePrice'] = df['TotalPurchaseDollars'] /df['TotalPurchaseQuant
```

```
In [111…  df["Ordersize"] = pd.qcut(df["TotalPurchaseQuantity"], q=3, labels=["Small",
```

```
In [112…  df
```

Out[112…

	VendorNumber	VendorName	Brand	Description	PurchasePrice	Actu
0	1128	BROWN-FORMAN CORP	1233	Jack Daniels No 7 Black	26.27	
1	4425	MARTIGNETTI COMPANIES	3405	Tito's Handmade Vodka	23.19	
2	17035	PERNOD RICARD USA	8068	Absolut 80 Proof	18.24	
3	3960	DIAGEO NORTH AMERICA INC	4261	Capt Morgan Spiced Rum	16.17	
4	3960	DIAGEO NORTH AMERICA INC	3545	Ketel One Vodka	21.89	
...	...	...	...	...	...	
8559	9815	WINE GROUP INC	8527	Concannon Glen Ellen Wh Zin	1.32	
8560	8004	SAZERAC CO INC	5683	Dr McGillicuddy's Apple Pie	0.39	
8561	3924	HEAVEN HILL DISTILLERIES	9123	Deep Eddy Vodka	0.74	
8562	3960	DIAGEO NORTH AMERICA INC	6127	The Club Strawbry Margarita	1.47	
8563	7245	PROXIMO SPIRITS INC.	3065	Three Olives Grape Vodka	0.71	

8564 rows × 20 columns

In [114…

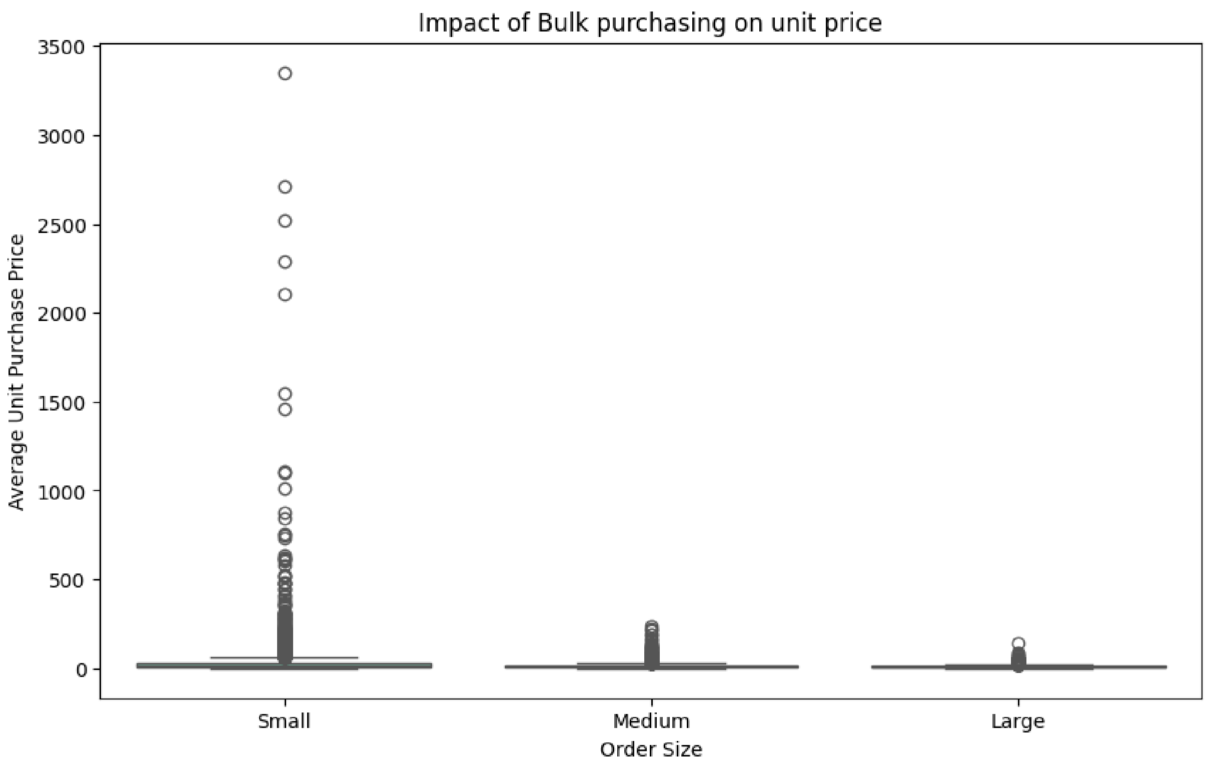
```python
df.groupby('Ordersize')[['UnitPurchasePrice']].mean()
```

Out[114…

	UnitPurchasePrice
Ordersize	
Small	39.068186
Medium	15.486414
Large	10.777625

In [116…

```python
plt.figure(figsize=(10,6))
sns.boxplot(data=df, x="Ordersize", y="UnitPurchasePrice", palette="Set2")
plt.title("Impact of Bulk purchasing on unit price")
plt.xlabel("Order Size")
plt.ylabel("Average Unit Purchase Price")
plt.show()
```



In [118…

```
# Vendors buying in bulk (Large Order Size) get the lowest unit price ($10.7
# The price difference between Small and Large orders is substantial (~72% r
# This suggests that bulk pricing strategies successfully encourage vendors
```

In [120…

```
# Which vendors have low inventory turover, indicating excess stock and slow
```

In [124…

```
df[df['StockTurnover']<1].groupby('VendorName')[['StockTurnover']].mean().sc
```

Out[124…

	StockTurnover
VendorName	
ALISA CARR BEVERAGES	0.615385
HIGHLAND WINE MERCHANTS LLC	0.708333
PARK STREET IMPORTS LLC	0.751306
Circa Wines	0.755676
Dunn Wine Brokers	0.766022
CENTEUR IMPORTS LLC	0.773953
SMOKY QUARTZ DISTILLERY LLC	0.783835
TAMWORTH DISTILLING	0.797078
THE IMPORTED GRAPE LLC	0.807569
WALPOLE MTN VIEW WINERY	0.820548

In [125…

```
# How much captial is locked in unsold inventory per vendor, and which vendo
```

In [126…

```
df['UnsoldInventoryValue'] = (df['TotalPurchaseQuantity'] - df['TotalSalesQu
print("Total Unsold Capital", format_dollars(df["UnsoldInventoryValue"].sum(
```

```
Total Unsold Capital 2.71M
```

In [133…

```
# Aggregate Capital Locked per Vendor

inventory_value_per_vendor = df.groupby("VendorName") ["UnsoldInventoryValue

#Sort Vendors with the Highest Locked Capital
inventory_value_per_vendor = inventory_value_per_vendor.sort_values(by="Unso
inventory_value_per_vendor['UnsoldInventoryValue'] = inventory_value_per_ven
inventory_value_per_vendor.head(10)
```

Out[133…

	VendorName	UnsoldInventoryValue
25	DIAGEO NORTH AMERICA INC	0.72M
46	JIM BEAM BRANDS COMPANY	0.55M
68	PERNOD RICARD USA	0.47M
116	WILLIAM GRANT & SONS INC	0.40M
30	E & J GALLO WINERY	0.23M
79	SAZERAC CO INC	0.20M
11	BROWN-FORMAN CORP	0.18M
20	CONSTELLATION BRANDS INC	0.13M
61	MOET HENNESSY USA INC	0.13M
77	REMY COINTREAU USA INC	0.12M

In [ ]:

```
# What is the 95% confidence intervals for profit margins of top-performing
```

In [142…

```
top_threshold = df["TotalSalesDollars"].quantile(0.75)
low_threshold = df["TotalSalesDollars"].quantile(0.25)
```

In [143…

```
top_vendors = df[df["TotalSalesDollars"] >= top_threshold]['ProfitMargin'].c
low_vendors = df[df["TotalSalesDollars"] <= low_threshold]['ProfitMargin'].c
```

In [145…

```
top_vendors
```

Out[145…

```
0       25.297693
1       21.062810
2       24.675786
3       27.139908
4       28.412764
          ...
3523    79.684817
3681    85.782102
4751    93.085860
4920    95.012530
5050    94.271857
Name: ProfitMargin, Length: 2141, dtype: float64
```

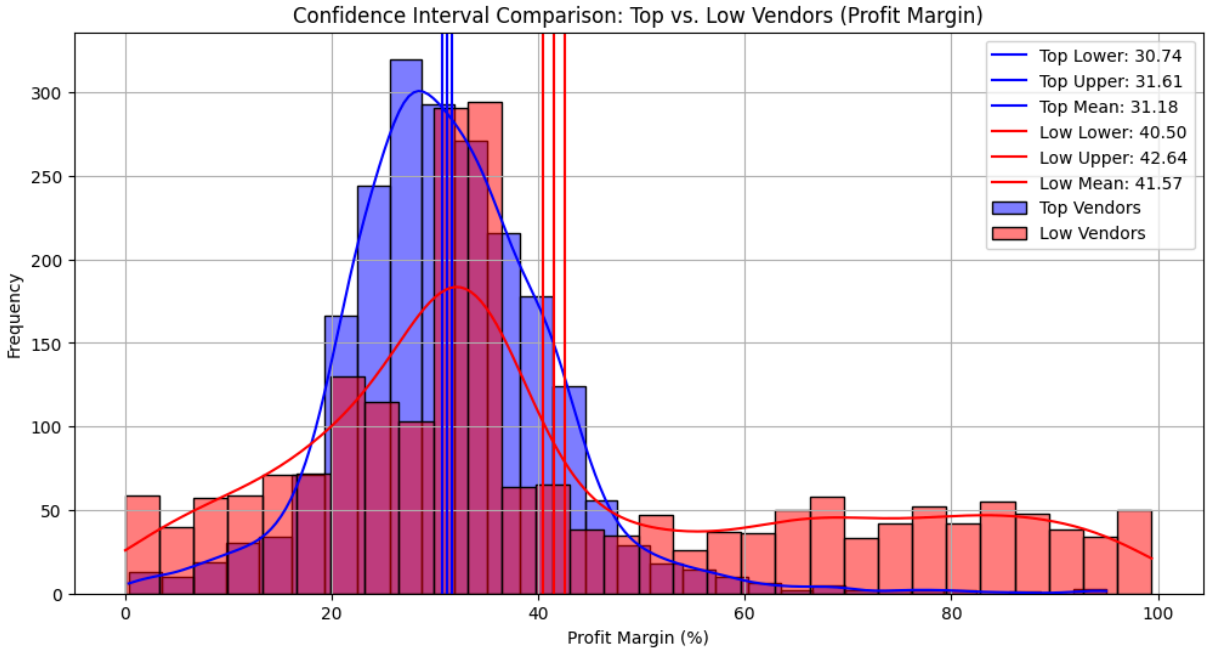
In [147…

```python
def confidence_interval(data, confidence=0.95):
    mean_val = np.mean(data)
    std_err = np.std(data, ddof=1) / np.sqrt(len(data))
    t_critical = stats.t.ppf((1+confidence)/2, df=len(data)-1)
    margin_of_error = t_critical*std_err
    return mean_val, mean_val - margin_of_error, mean_val+margin_of_error
```

In [155…

```python
top_mean, top_lower, top_upper = confidence_interval(top_vendors)
low_mean, low_lower, low_upper = confidence_interval(low_vendors)
print(f"Top Vendors 95% CI: ({top_lower:.2f}, {top_upper:.2f}), Mean: {top_m
print(f"Low Vendors 95% CI: ({low_lower:.2f}, {low_upper:.2f}), Mean: {low_m
plt.figure(figsize=(12, 6))

#Top Vendors Plot
sns.histplot(top_vendors, kde=True, color="blue", bins=30, alpha=0.5, label=
plt.axvline(top_lower, color="blue", linestyle="-", label=f"Top Lower: {top_
plt.axvline(top_upper, color="blue", linestyle="-", label=f"Top Upper: {top_
plt.axvline(top_mean, color="blue", linestyle="-", label=f"Top Mean: {top_me

# Low Vendors Plot
sns.histplot(low_vendors, kde=True, color="red", bins=30, alpha=0.5, label="
plt.axvline(low_lower, color="red", linestyle="-", label=f"Low Lower: {low_l
plt.axvline(low_upper, color="red", linestyle="-", label=f"Low Upper: {low_u
plt.axvline(low_mean, color="red", linestyle="-", label=f"Low Mean: {low_mea


# Finalize Plot
plt.title("Confidence Interval Comparison: Top vs. Low Vendors (Profit Margi
plt.xlabel("Profit Margin (%)")
plt.ylabel("Frequency")
plt.legend()
plt.grid(True)
plt.show()
```

```
Top Vendors 95% CI: (30.74, 31.61), Mean: 31.18
Low Vendors 95% CI: (40.50, 42.64), Mean: 41.57
```



In [ ]:

```
# The confidence interval for low-performing vendors (40.48% to 42.62%) is s
# This suggests that vendors with lower sales tend to maintain higher profit
# For High-Performing Vendors: If they aim to improve profitability, they co
# For Low-Performing Vendors: Despite higher margins, their low sales volume
```

In [ ]:

```
# Is there a significant difference in profit margins between top-performing


# Hypothesis:
# Ho (Null Hypothesis): There is no significant difference in the mean profi
# H. (Alternative Hypothesis): The mean profit margins of top-performing and
```

In [158…

```
top_threshold = df["TotalSalesDollars"].quantile(0.75)
low_threshold = df["TotalSalesDollars"].quantile(0.25)

top_vendors = df[df["TotalSalesDollars"] >= top_threshold]['ProfitMargin'].c
low_vendors = df[df["TotalSalesDollars"] <= low_threshold]['ProfitMargin'].c


t_stat, p_value = ttest_ind(top_vendors, low_vendors, equal_var=False)
#Print results
print(f"T-Statistic: {t_stat:.4f}, P-Value: {p_value:.4f}")
if p_value < 0.05:
    print("Reject Ho: There is a significant difference in profit margins be
else:
    print("Fail to Reject Ho: No significant difference in profit margins.")
```

```
T-Statistic: -17.6695, P-Value: 0.0000
Reject Ho: There is a significant difference in profit margins between top a
nd low-performing vendors.
```

In [ ]: