Importing Libraries

In [10]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
import sqlite3
from scipy.stats import ttest_ind
import scipy.stats as stats
warnings.filterwarnings('ignore')
```

Loading the dataset

In [11]:
```
# creating database connection
conn = sqlite3.connect('inventory.db')

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

Out[11]:

	VendorNumber	VendorName	Brand	Description	PurchasePrice	ActualPrice	Volume	T
0	2	IRA GOLDMAN AND WILLIAMS, LLP	90085	Ch Lilian 09 Ladouys St Este	23.86	36.99	750.0	
1	2	IRA GOLDMAN AND WILLIAMS, LLP	90609	Flavor Essence Variety 5 Pak	17.00	24.99	162.5	
2	54	AAPER ALCOHOL & CHEMICAL CO	990	Ethyl Alcohol 200 Proof	105.07	134.49	3750.0	
3	60	ADAMBA IMPORTS INTL INC	771	Bak's Krupnik Honey Liqueur	11.44	14.99	750.0	
4	60	ADAMBA IMPORTS INTL INC	3401	Vesica Vodka	11.10	14.99	1750.0	

Exploratory Data Analysis

Previously, we examined the various tables in the database to identify key variables, understand thier relationships, and determine which ones should be included in the final analysis

In this phase of EDA, we will analyze the resultant table to gain insights into the distribution of each column. This will help us understand data patterns, identify anamolies, and ensure data quality before proceeding with further analysis.

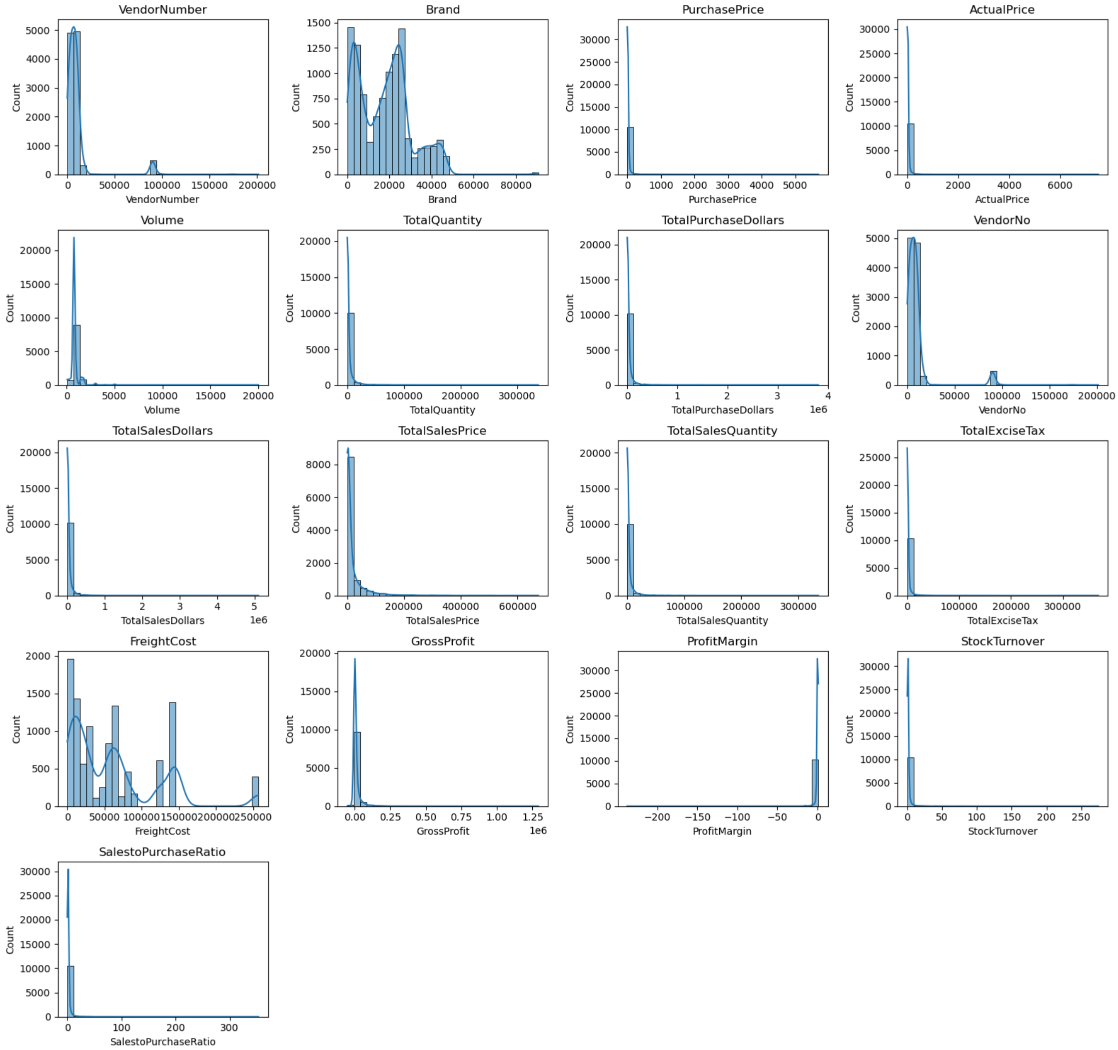
In [12]:
```
# summary statistics
df.describe().T # using T we can transpose also
```

Out[12]:

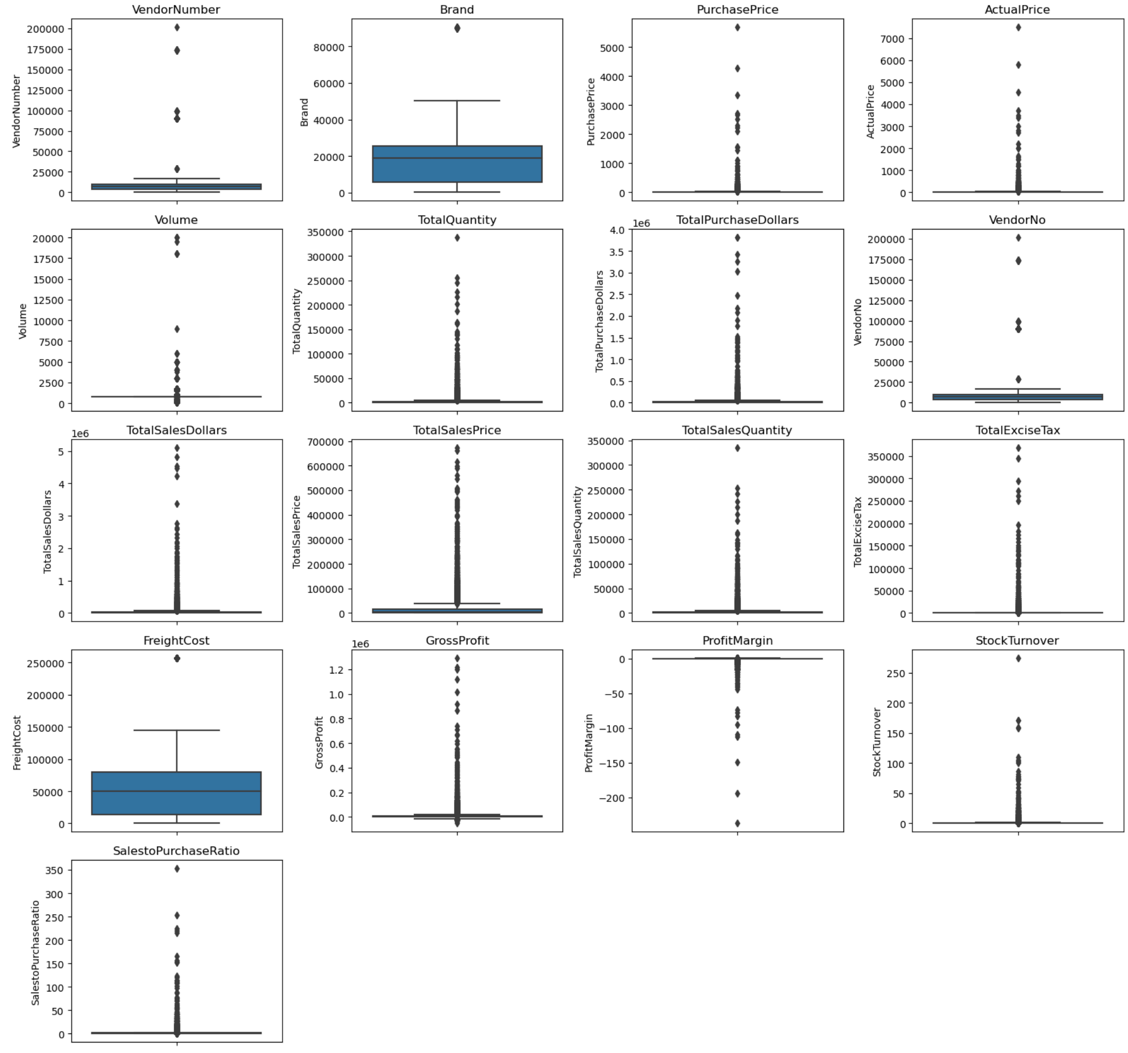
	count	mean	std	min	25%	
VendorNumber	10692.0	1.065065e+04	18753.519148	2.00	3951.000000	715
Brand	10692.0	1.803923e+04	12662.187074	58.00	5793.500000	1876
PurchasePrice	10692.0	2.438530e+01	109.269375	0.36	6.840000	1
ActualPrice	10692.0	3.564367e+01	148.246016	0.49	10.990000	1
Volume	10692.0	8.473605e+02	664.309212	50.00	750.000000	75
TotalQuantity	10692.0	3.140887e+03	11095.086769	1.00	36.000000	26
TotalPurchaseDollars	10692.0	3.010669e+04	123067.799627	0.71	453.457500	365
VendorNo	10692.0	1.042370e+04	18555.092692	0.00	3664.000000	715
TotalSalesDollars	10692.0	4.223907e+04	167655.265984	0.00	729.220000	529
TotalSalesPrice	10692.0	1.879378e+04	44952.773386	0.00	289.710000	285
TotalSalesQuantity	10692.0	3.077482e+03	10952.851391	0.00	33.000000	26
TotalExciseTax	10692.0	1.774226e+03	10975.582240	0.00	4.800000	4
FreightCost	10692.0	6.143376e+04	60938.458032	0.09	14069.870000	5029
GrossProfit	10692.0	1.213238e+04	46224.337964	-52002.78	52.920000	139
ProfitMargin	10692.0	-inf	NaN	-inf	0.133245	
StockTurnover	10692.0	1.706793e+00	6.020460	0.00	0.807229	
SalestoPurchaseRatio	10692.0	2.504390e+00	8.459067	0.00	1.153729	

In [14]:

```
import math
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

numerical_cols = df.select_dtypes(include=np.number).columns
num_cols = len(numerical_cols)
cols_in_grid = 4
rows_in_grid = math.ceil(num_cols / cols_in_grid)

plt.figure(figsize=(cols_in_grid * 4, rows_in_grid * 3))
for i, col in enumerate(numerical_cols):
    plt.subplot(rows_in_grid, cols_in_grid, i + 1)
    sns.histplot(df[col], kde=True, bins=30)
    plt.title(col)
plt.tight_layout()
plt.show()
```



In [16]:
```
import math
import matplotlib.pyplot as plt
import seaborn as sns

numerical_cols = df.select_dtypes(include=np.number).columns
num_cols = len(numerical_cols)
cols_in_grid = 4
rows_in_grid = math.ceil(num_cols / cols_in_grid)

plt.figure(figsize=(cols_in_grid * 4, rows_in_grid * 3))
for i, col in enumerate(numerical_cols):
    plt.subplot(rows_in_grid, cols_in_grid, i + 1)
    sns.boxplot(y=df[col])
    plt.title(col)
plt.tight_layout()
plt.show()
```



Summary Statistics Insights:

Negative and Zero Values:

Gross profit: Minimum Value is -52002.78, indicating losses. Some products or transactions may be selling at a loss due to high costs or selling at discounts lower than the purchase price.

Profit Margin: Has a minimum of -infi, which suggests cases where revenue is zero or even lower than costs.

Total Sales Quantity and Sales Dollars: Minimum values are 0, meaning some products were purchases but never sold. These could be slow-moving or obsolete stock.

Outlier Indicated by High Standard Deviations:

Purchase & Actual Prices: The max values (5681.81 and 7499.99) are significantly higher than the mean(24.39 & 35.64), indicating potential premium products.

Freight Cost: Huge variation, from 0.09 to 257032.07, suggests logistics inefficiencies or bulk shipments.

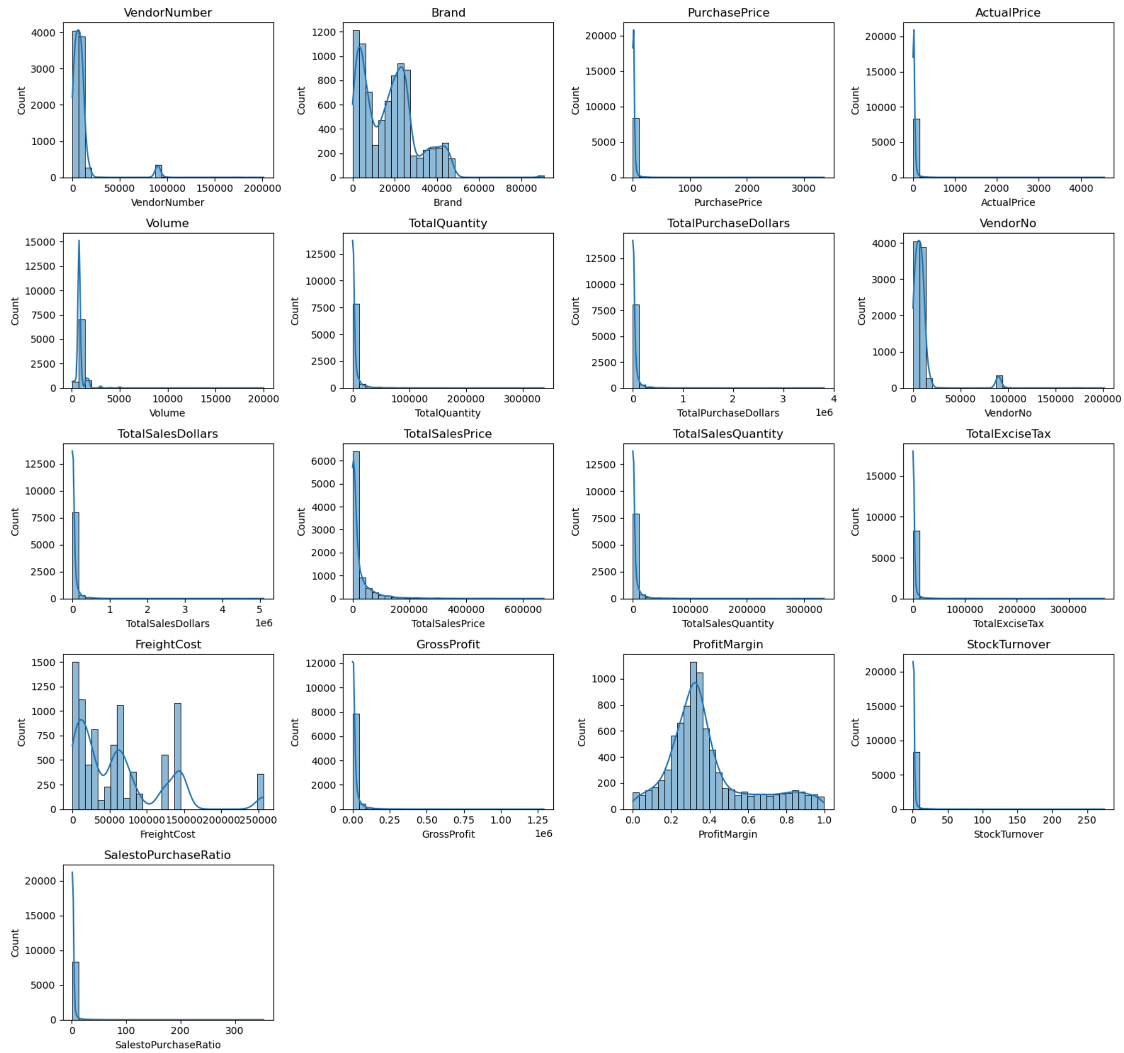
Stock Turnover: Ranges from 0 to 274.5, implying some products sell extremely fast while others remain in stock indefinitely. Value more than 1 indicates that Sold quantity for that

In [17]:

```
# let's filter the data by removing inconsistencies
df = pd.read_sql_query("""SELECT *
FROM vendor_sales_summary
WHERE GrossProfit > 0
AND ProfitMargin > 0
AND TotalSalesQuantity >0""",conn)
```

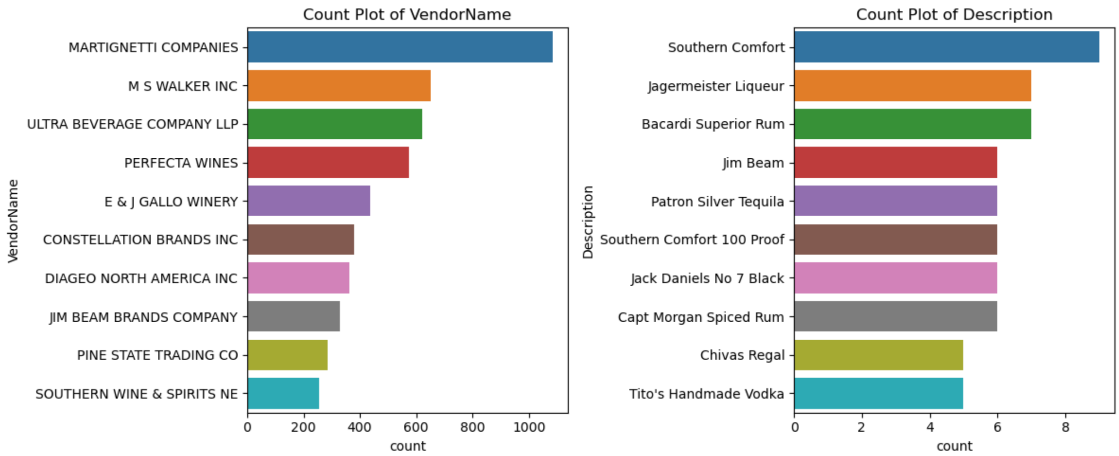
In [18]:

```
import math
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

numerical_cols = df.select_dtypes(include=np.number).columns
num_cols = len(numerical_cols)
cols_in_grid = 4
rows_in_grid = math.ceil(num_cols / cols_in_grid)

plt.figure(figsize=(cols_in_grid * 4, rows_in_grid * 3))
for i, col in enumerate(numerical_cols):
    plt.subplot(rows_in_grid, cols_in_grid, i + 1)
    sns.histplot(df[col], kde=True, bins=30)
    plt.title(col)
plt.tight_layout()
plt.show()
```



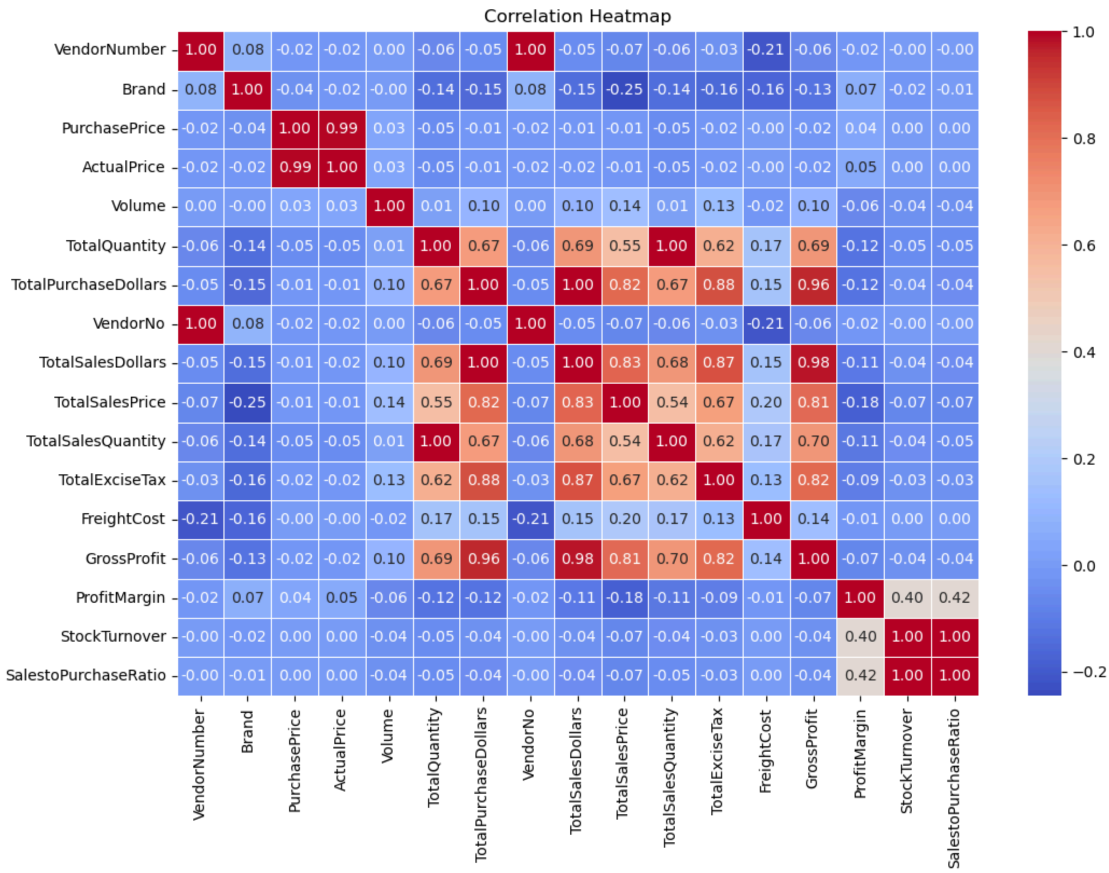
In [19]:

```
# Count Plots for Categorical Columns
categorical_cols = ['VendorName','Description']

plt.figure(figsize=(12,5))
for i, col in enumerate(categorical_cols):
    plt.subplot(1,2,i+1)
    sns.countplot(y=df[col],order = df[col].value_counts().index[:10]) # T
    plt.title(f'Count Plot of {col}')
plt.tight_layout()
plt.show()
```



In [20]:

```
# Correlation Heatmap
plt.figure(figsize=(12,8))
correlation_matrix = df[numerical_cols].corr()
sns.heatmap(correlation_matrix,annot=True,fmt = '.2f',cmap='coolwarm',line
plt.title('Correlation Heatmap')
plt.show()
```



Correlation Insights

PurchasePrice has weak correlation with TotalSalesDollars (-0.012) and GrossProfit (-0.016), suggesting that price variations do not significantly impact sales revenue or profit.

Strong correlation between total purchase quantity and total sales quantity(0.999), confirming efficient inventory turnover.

Negative Correlation between profit margin & total sales price(-0.179) suggests that as sales price increases, margins decrease, possibly due to competitive pricing measures.

StockTurnover has weak negative correlations with both GrossProfit(-0.038) and ProfitMargin (-0.055) indicating faster turnover does not necessarily result in higher profitably.

Data Analysis

Identify Brands that needs Promotional or Pricing Adjustments which exhibits lower sales performance but higher profit margins.

In [23]:
```
brand_performance = df.groupby('Description').agg({'TotalSalesDollars':'su
```

In [24]:
```
low_sales_threshold = brand_performance['TotalSalesDollars'].quantile(0.15
high_margin_threshold = brand_performance['ProfitMargin'].quantile(0.85)
```

In [25]:
```
low_sales_threshold
```

Out[25]:
```
560.299
```

In [26]:
```
high_margin_threshold
```

Out[26]:
```
0.6497017552750112
```

In [27]:

```
# 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'))
```

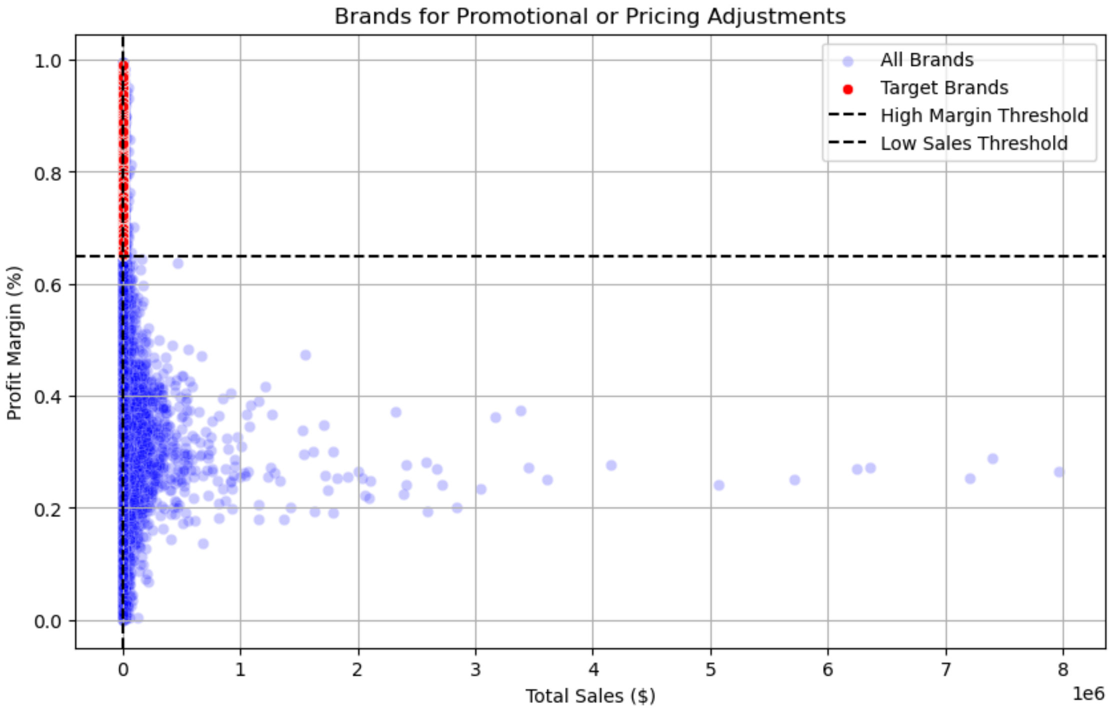
```
Brand with Low Sales but High Profit Margins:
```

	Description	TotalSalesDollars	ProfitMargin
6199	Santa Rita Organic Svgn Bl	9.99	0.664665
2369	Debauchery Pnt Nr	11.58	0.659758
2070	Concannon Glen Ellen Wh Zin	15.95	0.834483
2188	Crown Royal Apple	27.86	0.898062
6237	Sauza Sprklg Wild Berry Marg	27.96	0.821531
...	...	...	...
5074	Nanbu Bijin Southern Beauty	535.68	0.767473
2271	Dad's Hat Rye Whiskey	538.89	0.818516
57	A Bichot Clos Marechaudes	539.94	0.677409
6245	Sbragia Home Ranch Merlot	549.75	0.664447
3326	Goulee Cos d'Estournel 10	558.87	0.694348

198 rows × 3 columns

In [29]:

```
plt.figure(figsize=(10,6))
sns.scatterplot(data = brand_performance,x='TotalSalesDollars',y='ProfitMa
sns.scatterplot(data = target_brands,x='TotalSalesDollars',y = 'ProfitMarg

plt.axhline(high_margin_threshold,linestyle = '--',color = 'black',label =
plt.axvline(low_sales_threshold,linestyle = '--',color = 'black',label = '

plt.xlabel('Total Sales ($)')
plt.ylabel('Profit Margin (%)')
plt.title('Brands for Promotional or Pricing Adjustments')
plt.legend()
plt.grid(True)
plt.show()
```



Which vendors and brands demonstrate the highest sales performance?

In [32]:

```
def format_dollars(value):
    if value>=1000000:
        return f'{value/100000:.2f}M'
    elif value >=1000:
        return f'{value/1000:.2f}K'
    else:
        return str(value)
```

In [30]:

```
# Top Vendors & Brands by Sales performance
top_vendors = df.groupby('VendorName')['TotalSalesDollars'].sum().nlargest
top_brands = df.groupby('Description')['TotalSalesDollars'].sum().nlargest
top_vendors
```

Out[30]:

```
VendorName
DIAGEO NORTH AMERICA INC      6.799010e+07
MARTIGNETTI COMPANIES         3.933036e+07
PERNOD RICARD USA             3.206320e+07
JIM BEAM BRANDS COMPANY       3.142302e+07
BACARDI USA INC               2.485482e+07
CONSTELLATION BRANDS INC      2.421875e+07
E & J GALLO WINERY            1.839990e+07
BROWN-FORMAN CORP             1.824723e+07
ULTRA BEVERAGE COMPANY LLP    1.650254e+07
M S WALKER INC                1.470646e+07
Name: TotalSalesDollars, dtype: float64
```

In [31]:

```
top_brands
```

Out[31]:

```
Description
Jack Daniels No 7 Black    7964746.76
Tito's Handmade Vodka      7399657.58
Grey Goose Vodka           7209608.06
Capt Morgan Spiced Rum     6356320.62
Absolut 80 Proof           6244752.03
Jameson Irish Whiskey      5715759.69
Ketel One Vodka            5070083.56
Baileys Irish Cream        4150122.07
Kahlua                     3604858.66
Tanqueray                  3456697.90
Name: TotalSalesDollars, dtype: float64
```

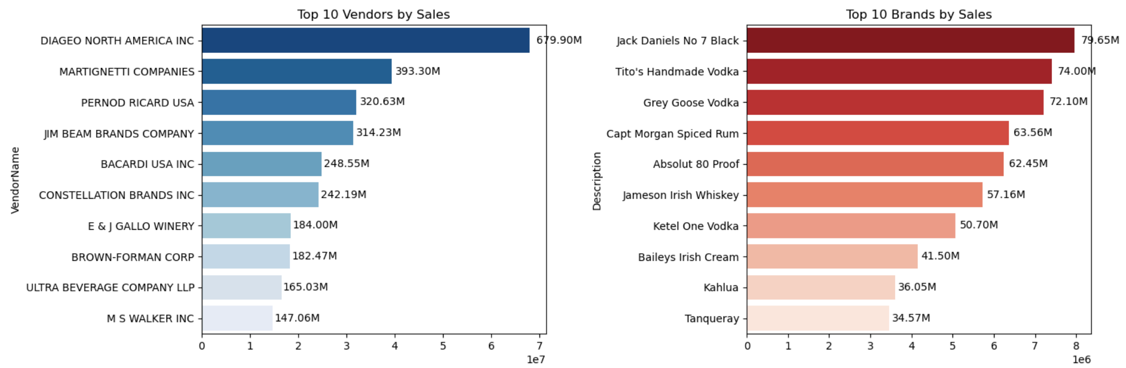
In [33]:

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

Out[33]:

```
Description
Jack Daniels No 7 Black    79.65M
Tito's Handmade Vodka      74.00M
Grey Goose Vodka           72.10M
Capt Morgan Spiced Rum     63.56M
Absolut 80 Proof           62.45M
Jameson Irish Whiskey      57.16M
Ketel One Vodka            50.70M
Baileys Irish Cream        41.50M
Kahlua                     36.05M
Tanqueray                  34.57M
Name: TotalSalesDollars, dtype: object
```

In [39]:
```
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 = 'Blue
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,p
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()
```



Which vendors contribute the most to total purchase dollars?

In [44]:
```
vendor_performance = df.groupby('VendorName').agg({
    'TotalPurchaseDollars':'sum',
    'GrossProfit':'sum',
    'TotalSalesDollars':'sum'
}).reset_index()
```

In [47]:
```
vendor_performance['PurchaseContribution%'] = (vendor_performance['TotalPu
```

In [48]:

```
vendor_performance
```

Out[48]:

	VendorName	TotalPurchaseDollars	GrossProfit	TotalSalesDollars	PurchaseContributio
0	ADAMBA IMPORTS INTL INC	446.16	258.37	704.53	0.000
1	ALISA CARR BEVERAGES	25698.12	78772.82	104470.94	0.008
2	ALTAMAR BRANDS LLC	11706.20	4000.61	15706.81	0.003
3	AMERICAN SPIRITS EXCHANGE	934.08	577.08	1511.16	0.000
4	AMERICAN VINTAGE BEVERAGE	104435.68	35167.85	139603.53	0.033
...	...	...	...	...	
114	WEIN BAUER INC	42694.64	13522.49	56217.13	0.013
115	WESTERN SPIRITS BEVERAGE CO	298416.86	106837.97	405254.83	0.097
116	WILLIAM GRANT & SONS INC	5876538.26	1693337.94	7569876.20	1.912
117	WINE GROUP INC	5203801.17	3100242.11	8304043.28	1.693
118	ZORVINO VINEYARDS	86122.71	38066.88	124189.59	0.028

119 rows × 5 columns

In [50]:

```
vendor_performance.sort_values('PurchaseContribution%',ascending = False,i
```

In [93]:
```
# Display top 10 vendors
top_vendors = vendor_performance.head(10)
top_vendors['TotalSalesDollars'] = top_vendors['TotalSalesDollars'].apply(
top_vendors['TotalPurchaseDollars'] = top_vendors['TotalPurchaseDollars'].
top_vendors['GrossProfit'] = top_vendors['GrossProfit'].apply(format_dolla
top_vendors
```

Out[93]:

	VendorName	TotalPurchaseDollars	GrossProfit	TotalSalesDollars	PurchaseContrib
25	DIAGEO NORTH AMERICA INC	500.97M	178.93M	679.90M	1[illegible]
57	MARTIGNETTI COMPANIES	255.02M	138.28M	393.30M	[illegible]
68	PERNOD RICARD USA	238.51M	82.12M	320.63M	[illegible]
46	JIM BEAM BRANDS COMPANY	234.94M	79.29M	314.23M	[illegible]
6	BACARDI USA INC	174.32M	74.23M	248.55M	[illegible]
20	CONSTELLATION BRANDS INC	152.74M	89.45M	242.19M	[illegible]
11	BROWN-FORMAN CORP	132.39M	50.08M	182.47M	[illegible]
30	E & J GALLO WINERY	120.69M	63.31M	184.00M	[illegible]
106	ULTRA BEVERAGE COMPANY LLP	111.67M	53.35M	165.03M	[illegible]
53	M S WALKER INC	97.64M	49.42M	147.06M	[illegible]

In [52]:
```
top_vendors['PurchaseContribution%'].sum()
```

Out[52]: 65.68957

In [54]:

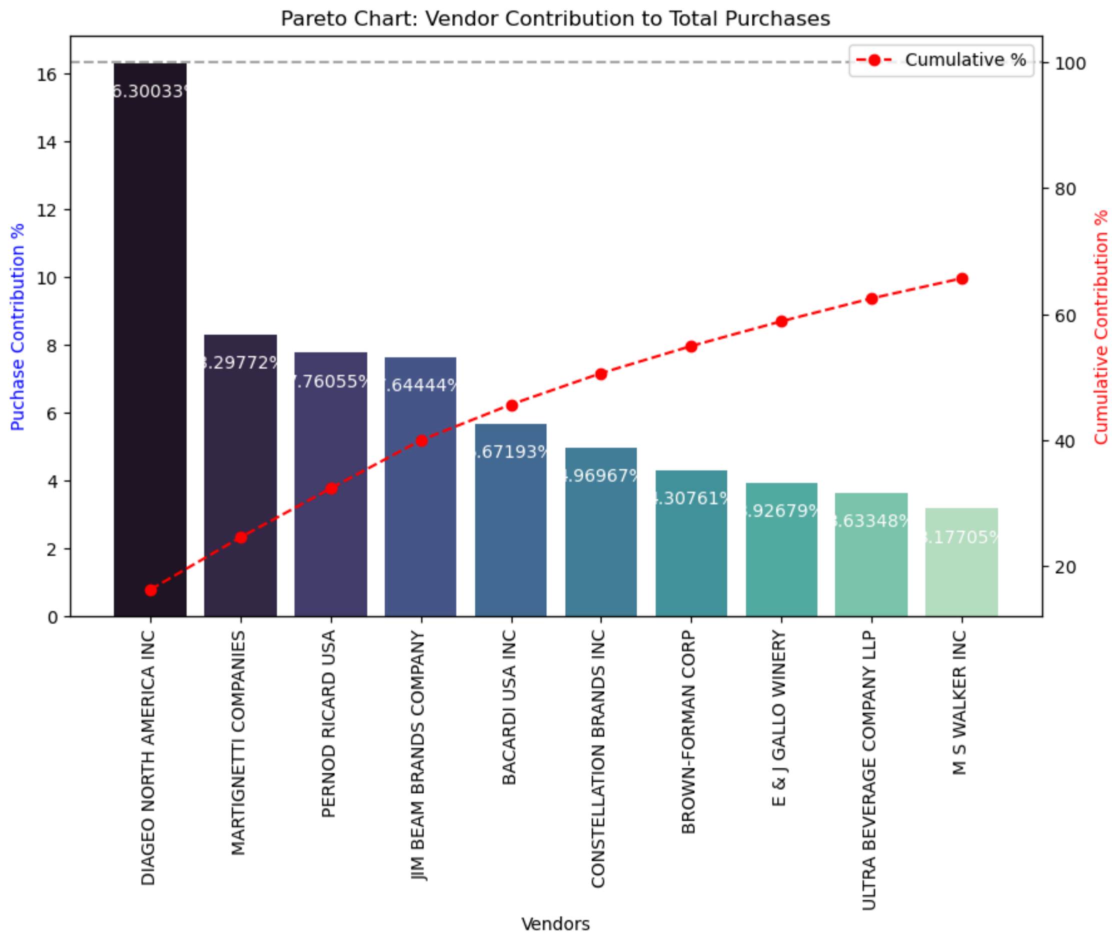
```
top_vendors['Cumulative_Contribution%'] = top_vendors['PurchaseContributio
top_vendors
```

Out[54]:

	VendorName	TotalPurchaseDollars	GrossProfit	TotalSalesDollars	PurchaseContrib
25	DIAGEO NORTH AMERICA INC	500.97M	178.93M	679.90M	[illegible]
57	MARTIGNETTI COMPANIES	255.02M	138.28M	393.30M	[illegible]
68	PERNOD RICARD USA	238.51M	82.12M	320.63M	[illegible]
46	JIM BEAM BRANDS COMPANY	234.94M	79.29M	314.23M	[illegible]
6	BACARDI USA INC	174.32M	74.23M	248.55M	[illegible]
20	CONSTELLATION BRANDS INC	152.74M	89.45M	242.19M	[illegible]
11	BROWN-FORMAN CORP	132.39M	50.08M	182.47M	[illegible]
30	E & J GALLO WINERY	120.69M	63.31M	184.00M	[illegible]
106	ULTRA BEVERAGE COMPANY LLP	111.67M	53.35M	165.03M	[illegible]
53	M S WALKER INC	97.64M	49.42M	147.06M	[illegible]

In [56]:

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

# Bar plot for Purchase Contribution%
sns.barplot(x=top_vendors['VendorName'],y=top_vendors['PurchaseContributio

for i, value in enumerate(top_vendors['PurchaseContribution%']):
    ax1.text(i,value-1,str(value)+'%',ha='center',fontsize=10,color='white

# Line Plot for Cumulative Contribution%
ax2 = ax1.twinx()
ax2.plot(top_vendors['VendorName'],top_vendors['Cumulative_Contribution%']

ax1.set_xticklabels(top_vendors['VendorName'],rotation = 90)
ax1.set_ylabel('Puchase 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()
```



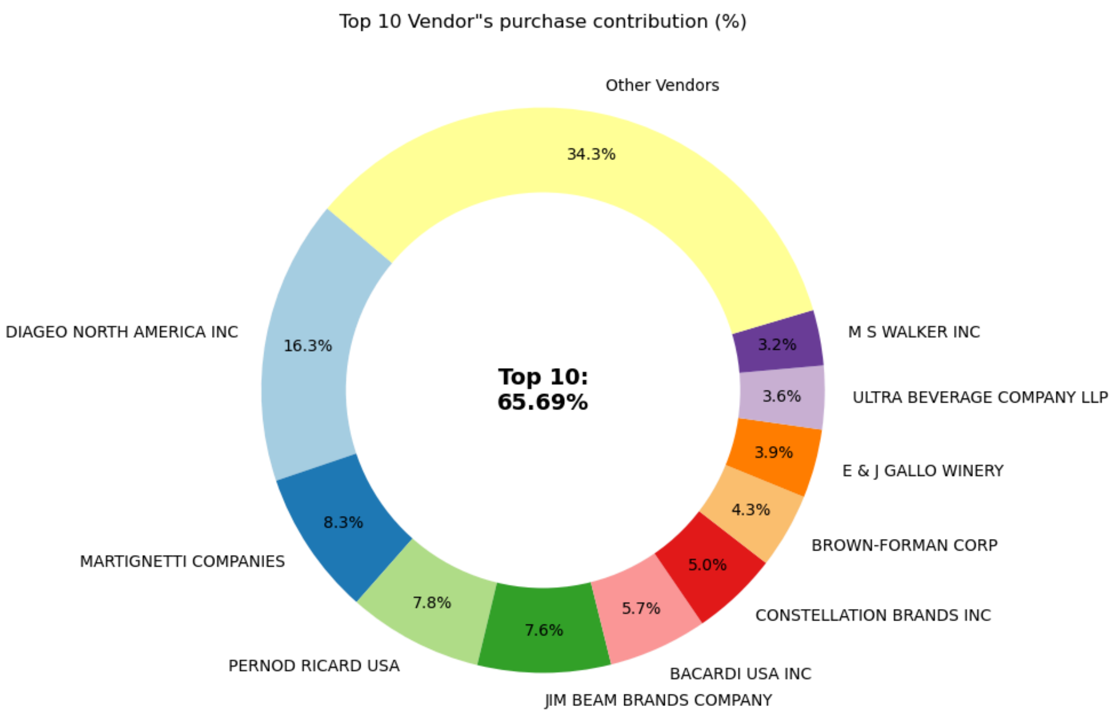
How much of total procurement is dependent on the top vendors?

In [60]:
```
print(f"Total Purchase Contribution of top 10 vendors is {top_vendors['Pur
```

```
Total Purchase Contribution of top 10 vendors is 65.69 %
```

In [62]:
```
vendors = list(top_vendors['VendorName'].values)
purchase_contributions = list(top_vendors['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
                                  startangle = 140,pctdistance = 0.85,color

# 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:\n{total_contribution:.2f}%",fontsize = 14,fontweigh
plt.title('Top 10 Vendor"s purchase contribution (%)')
```

Out[62]:
```
Text(0.5, 1.0, 'Top 10 Vendor"s purchase contribution (%)')
```



Does purchasing in bulk reduce the unit price, and what is the optimal purchase volume for cost savings?

In [64]:
```
df['UnitPurchasePrice'] = df['TotalPurchaseDollars']/df['TotalQuantity']
```

In [66]:
```
df['OrderSize'] = pd.qcut(df['TotalQuantity'],q=3,labels = ['Small','Mediu
```

In [68]:
```
df[['OrderSize','TotalQuantity']]
```

Out[68]:

	OrderSize	TotalQuantity
0	Small	8
1	Small	39
2	Small	12
3	Medium	320
4	Medium	96
...	...	...
8560	Medium	138
8561	Medium	267
8562	Medium	554
8563	Medium	1232
8564	Small	1

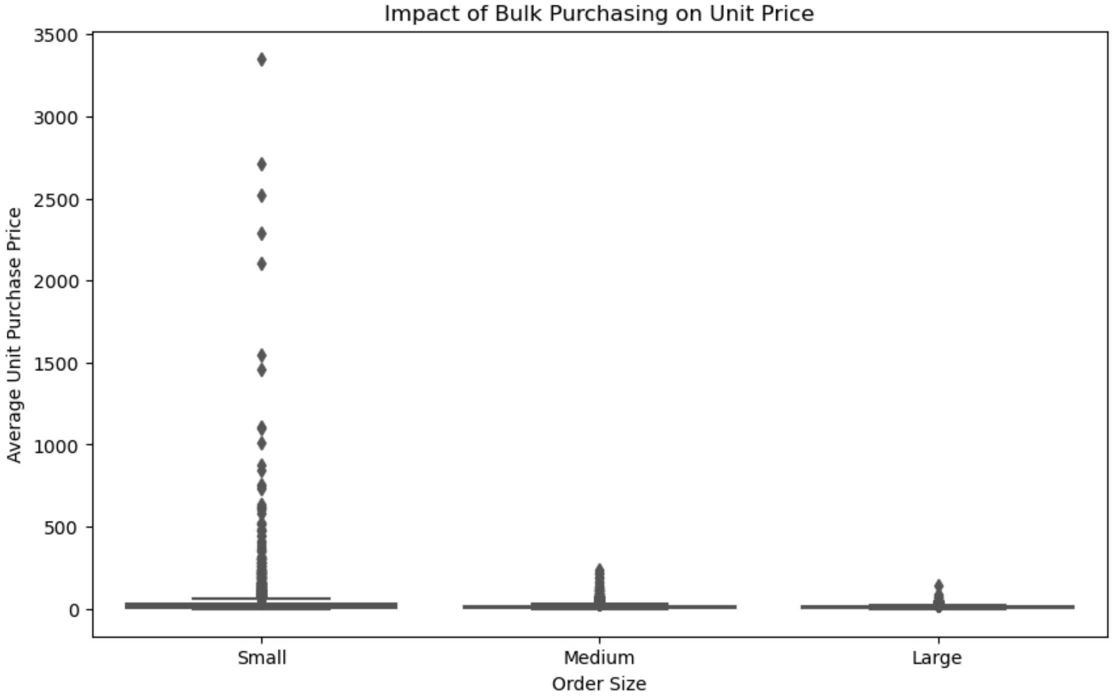
8565 rows × 2 columns

In [69]:
```
df.groupby('OrderSize')[['UnitPurchasePrice']].mean()
```

Out[69]:

	UnitPurchasePrice
OrderSize	
Small	39.057543
Medium	15.486414
Large	10.777625

In [70]:
```
plt.figure(figsize = (10,6))
sns.boxplot(data = df,x = 'OrderSize',y = 'UnitPurchasePrice',palette = 'S
plt.title('Impact of Bulk Purchasing on Unit Price')
plt.xlabel('Order Size')
plt.ylabel('Average Unit Purchase Price')
plt.show()
```



Vendors buying in Bulk (Larger Order Size) get the lowest unit price ($10.78 per unit), meaning higher margins if they can manage inventory efficiently

The price difference between Small and Large orders is substantial (~72% reduction in unit cost)

This suggests that bulk pricing strategies sucessfully encourage vendors to purchase in larger volumes, leading to higher overall sales despite lower per-unit revenue.

Which vendor has low inventory turnver, indicating excess stock and slow-moving products?

In [77]:
```
df[df['StockTurnover']<1].groupby('VendorName')[['StockTurnover']].mean().
```

Out[77]:

VendorName	StockTurnover
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

How much capital is locked in unsold inventory per vendor, and which vendors contribute the most to it?

In [79]:
```
df['UnsoldInventoryValue'] = (df['TotalQuantity'] - df['TotalSalesQuantity
print('Total Unsold Capital:',format_dollars(df['UnsoldInventoryValue'].su
```

```
Total Unsold Capital: 27.08M
```

In [81]:
```
# Aggregate Capital Locked per Vendor
inventory_value_per_vendor = df.groupby('VendorName')['UnsoldInventoryValu

# sort vendors with the highest locked capital
inventory_value_per_vendor = inventory_value_per_vendor.sort_values(by = '
inventory_value_per_vendor['UnsoldInventoryValue'] = inventory_value_per_v
inventory_value_per_vendor.head(10)
```

Out[81]:

	VendorName	UnsoldInventoryValue
25	DIAGEO NORTH AMERICA INC	722.21K
46	JIM BEAM BRANDS COMPANY	554.67K
68	PERNOD RICARD USA	470.63K
116	WILLIAM GRANT & SONS INC	401.96K
30	E & J GALLO WINERY	228.28K
79	SAZERAC CO INC	198.44K
11	BROWN-FORMAN CORP	177.73K
20	CONSTELLATION BRANDS INC	133.62K
61	MOET HENNESSY USA INC	126.48K
77	REMY COINTREAU USA INC	118.60K

What is the 95% confidence intervals for profit margins for profit margins of top-performing and low-performing vendors.

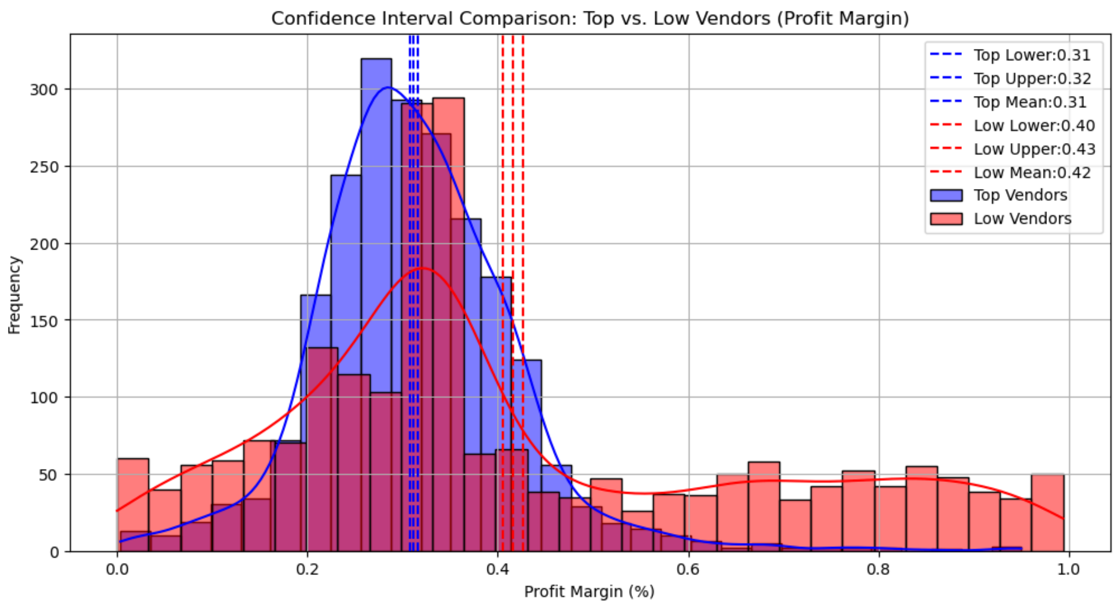
In [82]:
```
top_threshold = df['TotalSalesDollars'].quantile(0.75)
low_threshold = df['TotalSalesDollars'].quantile(0.25)
```

In [87]:
```
top_vendors = df[df['TotalSalesDollars'] >= top_threshold]['ProfitMargin']
low_vendors = df[df['TotalSalesDollars'] <= low_threshold]['ProfitMargin']
```

In [84]:
```
def confidence_interval(data,confidence = 0.95):
    mean_val = np.mean(data)
    std_err = np.std(data,ddof = 1)/np.sqrt(len(data)) # Standard Error
    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 [86]:
```
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_mea
print(f'Low Vendors 95% CI:({low_lower:.2f},{low_upper:.2f}),Mean:{low_mea

plt.figure(figsize = (12,6))

# Top Vendors Plot
sns.histplot(top_vendors,kde = True,color = 'blue',bins = 30,alpha = 0.5,l
plt.axvline(top_lower,color = 'blue',linestyle = '--',label = f'Top Lower:
plt.axvline(top_upper,color = 'blue',linestyle = '--',label = f'Top Upper:
plt.axvline(top_mean,color = 'blue',linestyle = '--',label = f'Top Mean:{t

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

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

```
Top Vendors 95% CI:(0.31,0.32),Mean:0.31
Low Vendors 95% CI:(0.40,0.43),Mean:0.42
```



The confidence interval for low-performing vendors (40.48% to 42.62%) is significantly higher than that of top-performing vendors (30.74% to 31.61%)

This suggests that vendors with lower sales tend to maintain higher profit margins, potentially due to premium pricing or lower operational costs.

For high-performing vendors: If they aim to imporve profitability, they could explore selective price adjustments, cost optimization or bundling strategies.

For low-performing vendors: despite higher margins, thier low sales volume might indicate a need for better marketing, competitive pricing, or improved distribution strategies.

Is there a significant difference in profit margins between top-performing and low-performing vendors?

Hypothesis:

H0(Null Hypothesis) ; There is no significant difference in the mean profit margins of top-performing and low-performing vendors.

H1(Alternative Hypothesis): The mean profit margins of top-performing vendors are significantly different.

In [89]:

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

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

# Perform Two-sample T-test
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 H0 : There is a significant difference in profit margins
else:
    print('Fail to reject H0: No significant difference in profit margins.
```

```
T-Statistic:-17.6440,P-Value:0.0000
Reject H0 : There is a significant difference in profit margins between t
op and low-performing vendors.
```

In [90]:

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

Out[90]:

	VendorNumber	VendorName	Brand	Description	PurchasePrice	ActualPrice	Volume	T
0	2	IRA GOLDMAN AND WILLIAMS, LLP	90085	Ch Lilian 09 Ladouys St Este	23.86	36.99	750.0	
1	2	IRA GOLDMAN AND WILLIAMS, LLP	90609	Flavor Essence Variety 5 Pak	17.00	24.99	162.5	
2	54	AAPER ALCOHOL & CHEMICAL CO	990	Ethyl Alcohol 200 Proof	105.07	134.49	3750.0	
3	60	ADAMBA IMPORTS INTL INC	771	Bak's Krupnik Honey Liqueur	11.44	14.99	750.0	
4	60	ADAMBA IMPORTS INTL INC	3401	Vesica Vodka	11.10	14.99	1750.0	

In [91]:

```
df.to_csv('vendor_sales_summary.csv')
```

In [92]:

Out[92]:

	VendorName	TotalPurchaseDollars	GrossProfit	TotalSalesDollars	PurchaseContri
25	DIAGEO NORTH AMERICA INC	50097226.16	1.789287e+07	6.799010e+07	
57	MARTIGNETTI COMPANIES	25502095.83	1.382826e+07	3.933036e+07	
68	PERNOD RICARD USA	23851164.17	8.212032e+06	3.206320e+07	
46	JIM BEAM BRANDS COMPANY	23494304.32	7.928716e+06	3.142302e+07	
6	BACARDI USA INC	17432020.26	7.422797e+06	2.485482e+07	
...	...	...	...	...	
107	UNCORKED	118.74	5.820000e+01	1.769400e+02	
33	FANTASY FINE WINES CORP	128.64	1.989500e+02	3.275900e+02	
85	SILVER MOUNTAIN CIDERS	77.18	2.653300e+02	3.425100e+02	
16	CAPSTONE INTERNATIONAL	54.64	1.922300e+02	2.468700e+02	
35	FLAVOR ESSENCE INC	17.00	1.457410e+03	1.474410e+03	

119 rows × 5 columns

In [94]:
```
vendor_performance.head(10)
```

Out[94]:

	VendorName	TotalPurchaseDollars	GrossProfit	TotalSalesDollars	PurchaseConti
25	DIAGEO NORTH AMERICA INC	50097226.16	1.789287e+07	6.799010e+07	
57	MARTIGNETTI COMPANIES	25502095.83	1.382826e+07	3.933036e+07	
68	PERNOD RICARD USA	23851164.17	8.212032e+06	3.206320e+07	
46	JIM BEAM BRANDS COMPANY	23494304.32	7.928716e+06	3.142302e+07	
6	BACARDI USA INC	17432020.26	7.422797e+06	2.485482e+07	
20	CONSTELLATION BRANDS INC	15273708.08	8.945038e+06	2.421875e+07	
11	BROWN-FORMAN CORP	13238939.18	5.008291e+06	1.824723e+07	
30	E & J GALLO WINERY	12068539.22	6.331360e+06	1.839990e+07	
106	ULTRA BEVERAGE COMPANY LLP	11167081.61	5.335463e+06	1.650254e+07	
53	M S WALKER INC	9764312.60	4.942146e+06	1.470646e+07	

In [ ]:
```
pd.read_sql_query('SELECT * FROM vendor_invoice',conn)
```

In [ ]:
```

```