Vendor Performance Analysis

June 6, 2025

[13]:
```python
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')
```

[4]:
```python
#Creatings database connection
conn = sqlite3.connect('inventory.db')

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

[4]:
```
   VendorNumber                  VendorName  Brand                Description  \
0          1128            BROWN-FORMAN CORP   1233  Jack Daniels No 7 Black
1          3960  DIAGEO NORTH AMERICA INC   4261    Capt Morgan Spiced Rum
2          4425      MARTIGNETTI COMPANIES   3405    Tito's Handmade Vodka
3         17035           PERNOD RICARD USA   8068          Absolut 80 Proof
4          3960  DIAGEO NORTH AMERICA INC   3545           Ketel One Vodka

   PurchasePrice  ActualPrice  Volume  TotalPurchaseQuantity  \
0          26.27        36.99  1750.0                  60320
1          16.17        22.99  1750.0                  96073
2          23.19        28.99  1750.0                  62385
3          18.24        24.99  1750.0                  75385
4          21.89        29.99  1750.0                  58783

   TotalPurchaseDollars  TotalSalesQuantity  TotalSalesDollars  \
0            1584606.40              9578.0          344712.22
1            1553500.41             20226.0          444810.74
2            1446708.15              9203.0          275162.97
3            1375022.40             11189.0          288135.11
4            1286759.87             11883.0          357759.17
```

```
   TotalSalesPrice  TotalExciseTax  FreightCost  GrossProfit  ProfitMargin  \
0         64889.97        17598.14     68601.68  -1239894.18   -359.689651
1         43304.31        37163.76    257032.07  -1108689.67   -249.249753
2         52289.50        16909.12    144929.24  -1171545.18   -425.764114
3         48202.30        20557.97    123780.22  -1086887.29   -377.214457
4         52774.51        21833.58    257032.07   -929000.70   -259.672086

   StockTurnover  SalesToPurchaseRatio
0       0.158786              0.217538
1       0.210527              0.286328
2       0.147519              0.190199
3       0.148425              0.209549
4       0.202150              0.278031
```

1 Exploratory Data Analysis

- Previously we examined the various tables in the database to identify key variables, understand their 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 of each column. This will help us understand data patterns, identify anomalies, and ensure data quality before proceeding with further analysis.

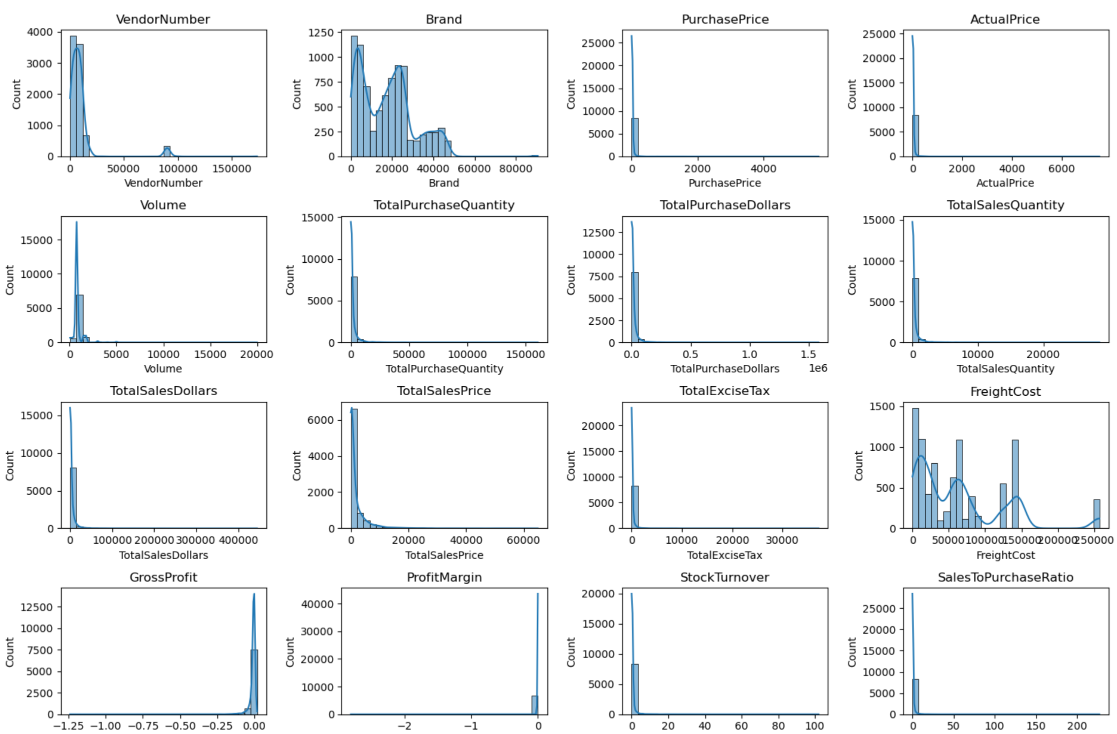
[7]:
```python
# summary statistics
df.describe().T
```

[7]:
```
                       count          mean           std         min  \
VendorNumber          8512.0  1.015346e+04  17718.122212        2.00
Brand                 8512.0  1.760026e+04  13004.702546       58.00
PurchasePrice         8512.0  2.188977e+01    105.829821        0.36
ActualPrice           8512.0  3.218842e+01    144.210224        0.49
Volume                8512.0  8.548756e+02    617.982459       50.00
TotalPurchaseQuantity 8512.0  1.691844e+03   5496.001551        1.00
TotalPurchaseDollars  8512.0  1.578656e+04  56938.194190        0.71
TotalSalesQuantity    8512.0  2.872162e+02    967.741069        0.00
TotalSalesDollars     8512.0  3.873112e+03  13541.331564        0.00
TotalSalesPrice       8512.0  1.890345e+03   4290.868376        0.00
TotalExciseTax        8512.0  1.632617e+02    947.469977        0.00
FreightCost           8512.0  6.391985e+04  62246.272374        0.27
GrossProfit           8512.0 -1.191345e+04  44347.038195 -1239894.18
ProfitMargin          8512.0          -inf           NaN        -inf
StockTurnover         8512.0  5.654312e-01      2.901080        0.00
SalesToPurchaseRatio  8512.0  8.412431e-01      4.576799        0.00

                               25%           50%           75%           max
VendorNumber           3664.000000   7153.000000   9552.000000  1.733570e+05
Brand                  5297.750000  17432.500000  24988.000000  9.063100e+04
```

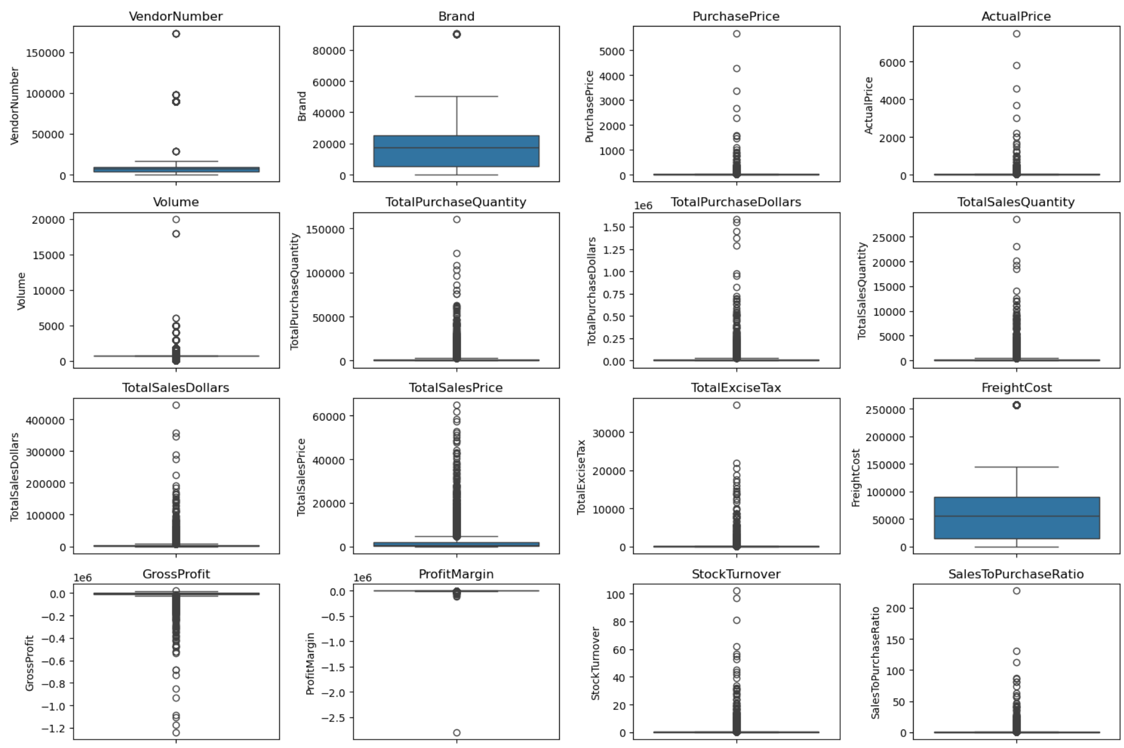
PurchasePrice	6.800000	10.270000	18.240000	5.681810e+03
ActualPrice	10.990000	15.990000	26.990000	7.499990e+03
Volume	750.000000	750.000000	750.000000	2.000000e+04
TotalPurchaseQuantity	24.000000	231.500000	1195.000000	1.607350e+05
TotalPurchaseDollars	344.880000	2840.535000	12601.875000	1.584606e+06
TotalSalesQuantity	3.000000	34.000000	200.000000	2.854400e+04
TotalSalesDollars	59.940000	649.745000	3129.827500	4.448107e+05
TotalSalesPrice	22.692500	363.675000	1831.792500	6.488997e+04
TotalExciseTax	0.340000	5.830000	48.122500	3.716376e+04
FreightCost	14836.570000	55551.820000	89286.270000	2.570321e+05
GrossProfit	-9333.232500	-1923.575000	-170.250000	1.914431e+04
ProfitMargin	-830.408421	-302.341871	-144.561326	9.956005e+01
StockTurnover	0.066946	0.165977	0.268397	1.020000e+02
SalesToPurchaseRatio	0.107480	0.248545	0.408895	2.272985e+02

[14]:

```
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)  # Adjust grid layout if needed
    sns.histplot(df[col], kde=True, bins=30)
    plt.title(col)

plt.tight_layout()
plt.show()
```



```
[15]: # Outliers detection with boxplots
plt.figure(figsize=(15, 10))
for i, col in enumerate(numerical_cols):
    plt.subplot(4, 4, i+1)  # Adjust grid layout if needed
    sns.boxplot(y=df[col])
    plt.title(col)

plt.tight_layout()
plt.show()
```



2 Summary Statistics Insights:

2.0.1 Negative & Zero Values:

• Gross Profit: Minimum value is -52,002.78, indicating losses. Some products or transactions rnay be selling at a loss due to high costs or selling at discounts lower than the purchase price.

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

• Total Sales Quantity & Sales Dollars; Minimum values are 0, meaning some products were purchased but never sold. These could be slow-moving or Obsolete stock

2.0.2 Outliers Indicated by High Standard Deviations:

- Purchase & Actual Prices: The max values (5,681.81 & 7,499.99) are significantly higher than the mean (24.39 & 35.64), indicating potential premium products.

- Freight Cost: Huge variation, frorn 0.09 to 257,032.07. suggests logistics inefficiencies or bulk shipments.

- Stock Turnover: Ranges from 0 to 274.5, implying some products sell extrernely fast while others remain in stock indefinitely. Value more than 1 indicates that Sold quantity for that product is higher than quantity to are being fulfilled older stock.

[118]:
```
# 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 TotalPurchaseQuantity > 0""", conn)
```

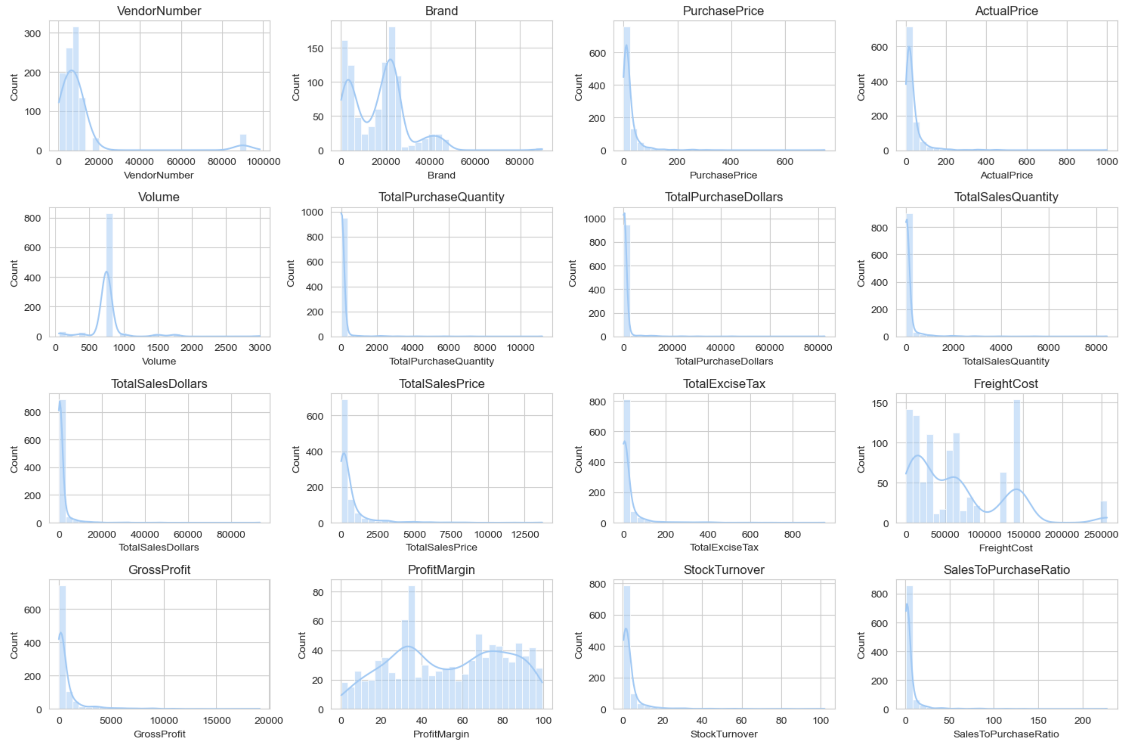
[121]:
```
df
```

[121]:
```
     VendorNumber                VendorName  Brand  \
0            1392  CONSTELLATION BRANDS INC   6650
1            1392  CONSTELLATION BRANDS INC  22143
2             516        BANFI PRODUCTS CORP  18152
3           10754             PERFECTA WINES  25197
4            4425     MARTIGNETTI COMPANIES   8781
..            …                         …      …
977         90024             VINILANDIA USA  46135
978          9815             WINE GROUP INC   8527
979          8004             SAZERAC CO INC   5683
980          9815             WINE GROUP INC  22407
981          7245        PROXIMO SPIRITS INC.   3065

                    Description  PurchasePrice  ActualPrice  Volume  \
0                    Simi Chard           7.38        14.99   750.0
1                 Simi Cab Svgn          10.52        18.99   750.0
2      Banfi Centine Mntcln Tscna          5.26        10.99   750.0
3    Ch La Rousseliere St Estephe         99.33       149.99   750.0
4         Rodney Strong Cab Svgn          10.32        15.99   750.0
..                            …              …            …       …
977      Aresti Pnt Nr Curico Vly           3.28        10.99   750.0
978    Concannon Glen Ellen Wh Zin          1.32         4.99   750.0
979    Dr McGillicuddy's Apple Pie          0.39         0.49    50.0
980            Three Wishes Chard          2.25         3.29   750.0
981      Three Olives Grape Vodka          0.71         0.99    50.0

     TotalPurchaseQuantity  TotalPurchaseDollars  TotalSalesQuantity  \
0                    11199              82648.62              8458.0
```

```
1                    4194         44120.88          2887.0
2                    5723         30102.98          3891.0
3                     249         24733.17           198.0
4                    2241         23127.12          1840.0
..                    …             …                 …
977                     1             3.28            15.0
978                     2             2.64             3.0
979                     6             2.34           128.0
980                     1             2.25             1.0
981                     1             0.71            81.0

     TotalSalesDollars  TotalSalesPrice  TotalExciseTax  FreightCost  \
0             93369.42         13558.87          949.79     79528.99
1             46523.13         13703.56          323.21     79528.99
2             31728.09          7605.68          436.17      8510.41
3             29698.02          1199.92           22.24     28720.52
4             31245.60         11934.97          205.53    144929.24
..                 …                …               …           …
977              74.85            39.92            1.68      2802.64
978               5.97             5.97            0.33     27100.41
979              62.72             0.98            6.72     50293.62
980               3.29             3.29            0.11     27100.41
981              80.19            29.70            4.21     38994.78

     GrossProfit  ProfitMargin  StockTurnover  SalesToPurchaseRatio
0       10720.80     11.482132       0.755246              1.129715
1        2402.25      5.163561       0.688364              1.054447
2        1625.11      5.121991       0.679888              1.053985
3        4964.85     16.717781       0.795181              1.200737
4        8118.48     25.982794       0.821062              1.351037
..           …             …              …                    …
977        71.57     95.617902      15.000000             22.820122
978         3.33     55.778894       1.500000              2.261364
979        60.38     96.269133      21.333333             26.803419
980         1.04     31.610942       1.000000              1.462222
981        79.48     99.114603      81.000000            112.943662

[982 rows x 18 columns]
```

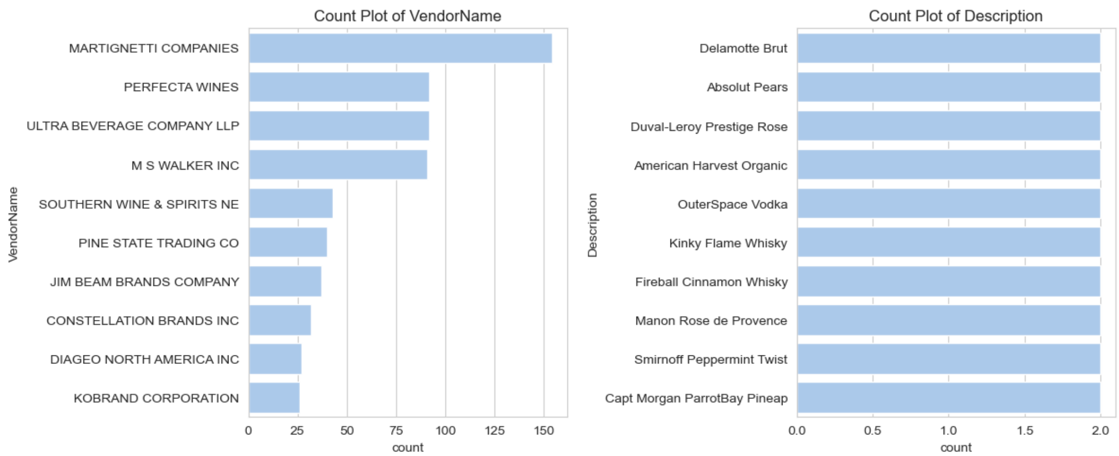
[122]:
```python
numerical_cols = df.select_dtypes(include=np.number).columns

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

```python
plt.tight_layout()
plt.show()
```

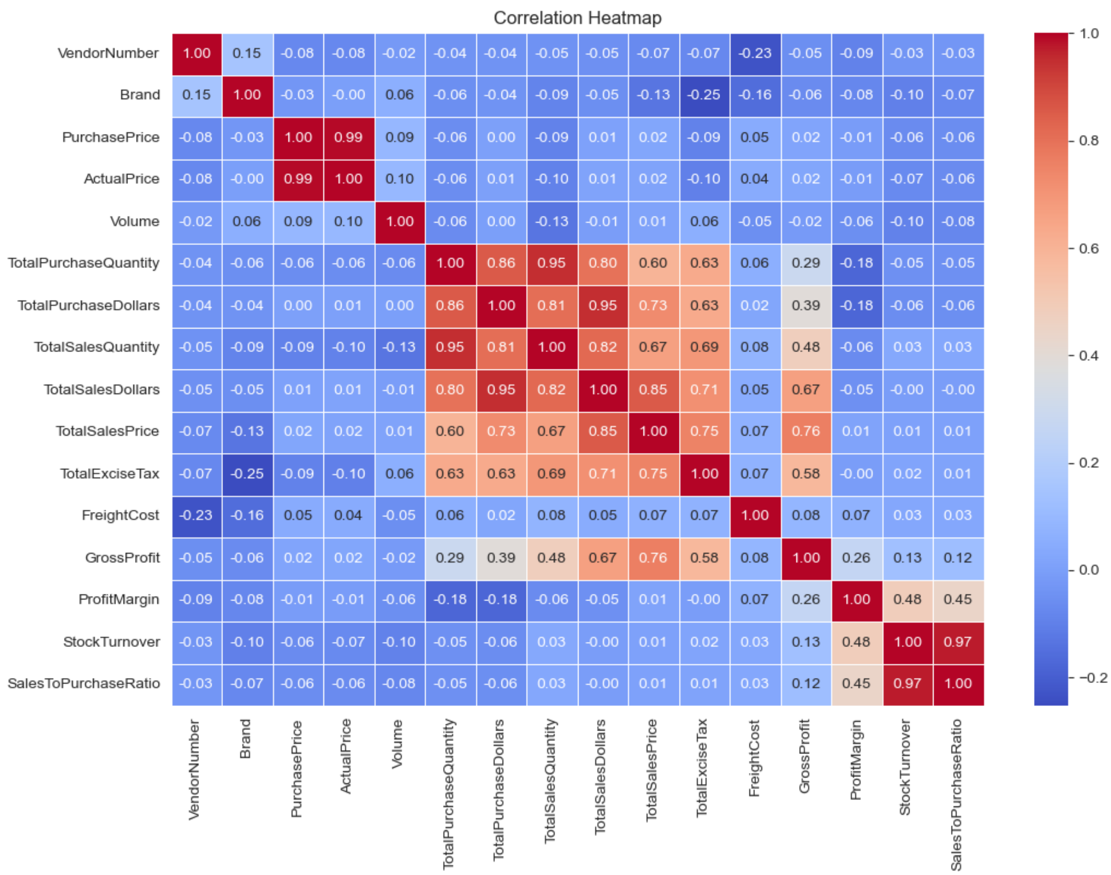


[123]:

```python
# Count Plots for Categorical Columns
categorical_cols = ["VendorName", "Description"]  # fixed variable name and type

plt.figure(figsize=(12, 5))
for i, col in enumerate(categorical_cols):
    plt.subplot(1, 2, i + 1)  # fixed 'l' (should be 1 for rows)
    sns.countplot(y=df[col], order=df[col].value_counts().index[:10]) #Top 10 categories
    plt.title(f"Count Plot of {col}")

plt.tight_layout()
plt.show()
```



[124]:
```
# Correlation Heatmap
plt.figure(figsize=(12, 8))
correlation_matrix = df[numerical_cols].corr()
sns.heatmap(correlation_matrix, annot=True, fmt=".
↪2f",cmap="coolwarm",linewidths=0.5)
plt.title("Correlation Heatmap")
plt.show()
```



3 Correlation Insights

PurchasePrice shows negligible correlation with TotalSalesDollars (-0.01) and GrossProfit (-0.02).

TotalPurchaseQuantity and TotalSalesQuantity are highly correlated (0.999) – strong inventory turnover.

ProfitMargin has a mild positive correlation with TotalSalesPrice (0.26) – higher price slightly improves margin.

StockTurnover has weak/no correlation with GrossProfit (-0.03) and ProfitMargin (0.12) – faster sales don't boost profit much.

4 Data Analysis

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

[125]:
```
brand_performance = df.groupby('Description').agg({
    'TotalSalesDollars':'sum',
```

```
    'ProfitMargin':'mean'}).reset_index()
```

[126]:
```
low_sales_threshold = brand_performance['TotalSalesDollars'].quantile(0.15)
high_margin_threshold = brand_performance['ProfitMargin'].quantile(0.85)
```

[127]:
```
low_sales_threshold
```

[127]: 117.98

[128]:
```
high_margin_threshold
```

[128]: 86.42455041969913

[129]:
```
# 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("Brands with Low Sales but High Profit Margins:" )
display( target_brands.sort_values( 'TotalSalesDollars'))
```

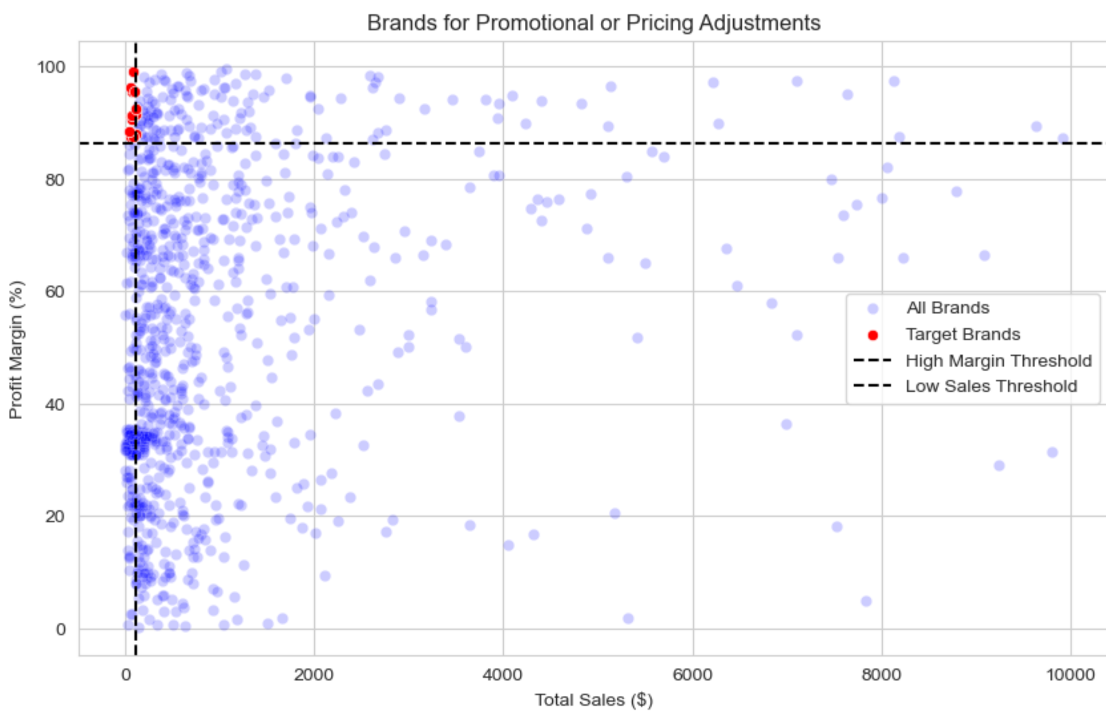
Brands with Low Sales but High Profit Margins:

	Description	TotalSalesDollars	ProfitMargin
915	Tracia Syrah	44.94	88.495772
64	Bacardi Oakheart Spiced Trav	59.94	87.554221
272	Chicken & Turkey Cotes du Rh	59.94	90.990991
349	Dr McGillicuddy's Apple Pie	62.72	96.269133
837	St Elder Elderflower Liqueur	66.33	91.436756
46	Aresti Pnt Nr Curico Vly	74.85	95.617902
324	DeKuyper Buttershots Trav	76.93	90.718835
900	Three Olives Grape Vodka	80.19	99.114603
838	St Germain Liqueur	89.94	87.658439
699	Piehole Apple Pie	98.01	95.592287
153	Capri Natura Limoncello	107.94	87.919214
612	Mojoshot Blue Lagoon RTD	112.86	91.591352
943	Vigne A Porrona Rosso	116.91	92.592593

[130]:
```
brand_performance =␣
↪brand_performance[brand_performance['TotalSalesDollars']<10000]  #for better␣
↪visualization
```

[131]:
```
plt.figure(figsize=(10, 6))
sns.scatterplot(data=brand_performance, x="TotalSalesDollars",␣
↪y="ProfitMargin", color="blue", label="All Brands", alpha=0.2)
sns.scatterplot(data=target_brands, x='TotalSalesDollars', y='ProfitMargin',␣
↪color='red', label='Target Brands')
```

```
plt.axhline(high_margin_threshold, linestyle="--", color="black", label="High␣
↪Margin Threshold")
plt.axvline(low_sales_threshold, linestyle="--", color="black", label="Low␣
↪Sales Threshold")

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.

[132]:
```
def format_dollars(value):
    if value >= 1_000_000:
        return f"{value / 1_000_000:.2f}M"
    elif value >= 1_000:
        return f"{value / 1_000:.2f}K"
    else:
        return str(value)
```

[133]:
```
# Top Vendors & Brands by Sales Performance
top_vendors = df.groupby("VendorName")["TotalSalesDollars"].sum().nlargest(10)
```

```
top_brands = df.groupby("Description")["TotalSalesDollars"].sum().nlargest(10)
top_vendors
```

[133]:
```
VendorName
MARTIGNETTI COMPANIES          243980.10
CONSTELLATION BRANDS INC       180305.21
ULTRA BEVERAGE COMPANY LLP     148726.20
M S WALKER INC                 117693.77
PERFECTA WINES                 114277.24
BACARDI USA INC                 61602.49
BROWN-FORMAN CORP               46884.09
DIAGEO NORTH AMERICA INC        42011.94
JIM BEAM BRANDS COMPANY         41429.49
SOUTHERN WINE & SPIRITS NE      40497.96
Name: TotalSalesDollars, dtype: float64
```

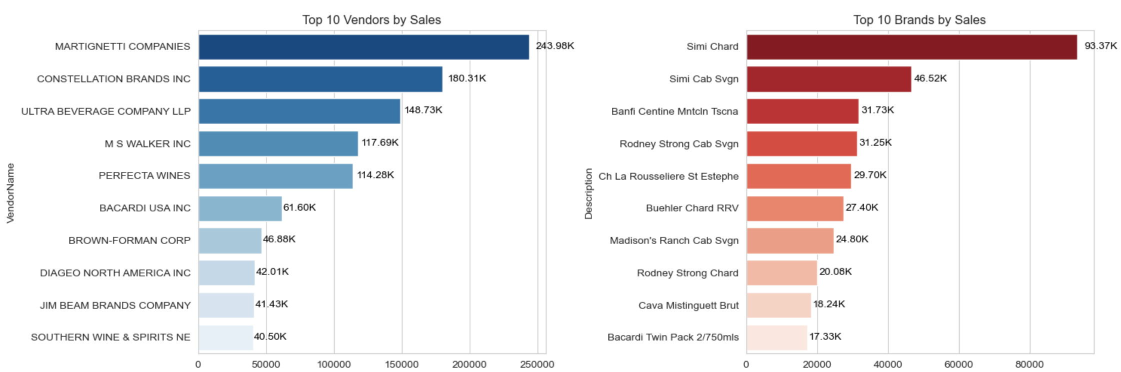
[134]:
```
top_brands.apply(lambda x : format_dollars(x))
```

[134]:
```
Description
Simi Chard                        93.37K
Simi Cab Svgn                     46.52K
Banfi Centine Mntcln Tscna        31.73K
Rodney Strong Cab Svgn            31.25K
Ch La Rousseliere St Estephe      29.70K
Buehler Chard RRV                 27.40K
Madison's Ranch Cab Svgn          24.80K
Rodney Strong Chard               20.08K
Cava Mistinguett Brut             18.24K
Bacardi Twin Pack 2/750mls        17.33K
Name: TotalSalesDollars, dtype: object
```

[135]:
```
import seaborn as sns
```

[136]:
```
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.flatten(),␣
 ↪palette="Blues_r")
plt.title("Top 10 Vendors by Sales")

# Add labels to the bars
for i, bar in enumerate(ax1.patches):
    width = bar.get_width()
    ax1.text(width + (width * 0.02),  # Position slightly to the right of the␣
 ↪bar
             bar.get_y() + bar.get_height() / 2,
```

```
                format_dollars(top_vendors.values.flatten()[i]),
                ha='left', va='center', fontsize=10, color='black')

# Plot for Top Brands
plt.subplot(1, 2, 2)
ax2 = sns.barplot(y=top_brands.index, x=top_brands.values.flatten(),␣
 ↪palette="Reds_r")
plt.title("Top 10 Brands by Sales")

# Add labels to the bars
for i, bar in enumerate(ax2.patches):
    width = bar.get_width()
    ax2.text(width + (width * 0.02),  # Position slightly to the right of the␣
 ↪bar
                bar.get_y() + bar.get_height() / 2,
                format_dollars(top_brands.values.flatten()[i]),
                ha='left', va='center', fontsize=10, color='black')

# Adjust layout and display the plot
plt.tight_layout()
plt.show()
```



Which vendor contribute the most to total purchase dollars?

[137]:
```
vendor_performance = df.groupby('VendorName').agg({
    'TotalPurchaseDollars':'sum',
    'GrossProfit':'sum',
    'TotalSalesDollars':'sum'

}).reset_index()

vendor_performance.shape
```

[137]: (72, 4)

```
[150]: vendor_performance['PurchaseContribution%'] =␣
         ↪vendor_performance['TotalPurchaseDollars'] /␣
         ↪vendor_performance['TotalPurchaseDollars'].sum()*100
```

```
[151]: vendor_performance = round(vendor_performance.
         ↪sort_values('PurchaseContribution%',ascending = False),2)
```

```
[152]: # Display Top 10 Vendors
       top_vendors = vendor_performance.head(10)
       top_vendors["TotalSalesDollars"] = top_vendors["TotalSalesDollars"].
         ↪apply(format_dollars)
       top_vendors["TotalPurchaseDollars"] = top_vendors["TotalPurchaseDollars"].
         ↪apply(format_dollars)
       top_vendors["GrossProfit"] = top_vendors["GrossProfit"].apply(format_dollars)
       top_vendors
```

```
[152]:                  VendorName TotalPurchaseDollars GrossProfit  \
       9     CONSTELLATION BRANDS INC              155.18K      25.12K
       33       MARTIGNETTI COMPANIES              106.12K     137.86K
       42              PERFECTA WINES               65.88K      48.39K
       63   ULTRA BEVERAGE COMPANY LLP              56.97K      91.76K
       31              M S WALKER INC               46.58K      71.11K
       3          BANFI PRODUCTS CORP               33.93K       3.67K
       2              BACARDI USA INC               30.94K      30.66K
       14    DIAGEO NORTH AMERICA INC               20.88K      21.13K
       26     JIM BEAM BRANDS COMPANY               18.69K      22.74K
       55         STATE WINE & SPIRITS              15.26K      22.06K

          TotalSalesDollars  PurchaseContribution%
       9            180.31K                  21.19
       33           243.98K                  14.49
       42           114.28K                   8.99
       63           148.73K                   7.78
       31           117.69K                   6.36
       3             37.59K                   4.63
       2             61.60K                   4.22
       14            42.01K                   2.85
       26            41.43K                   2.55
       55            37.32K                   2.08
```

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

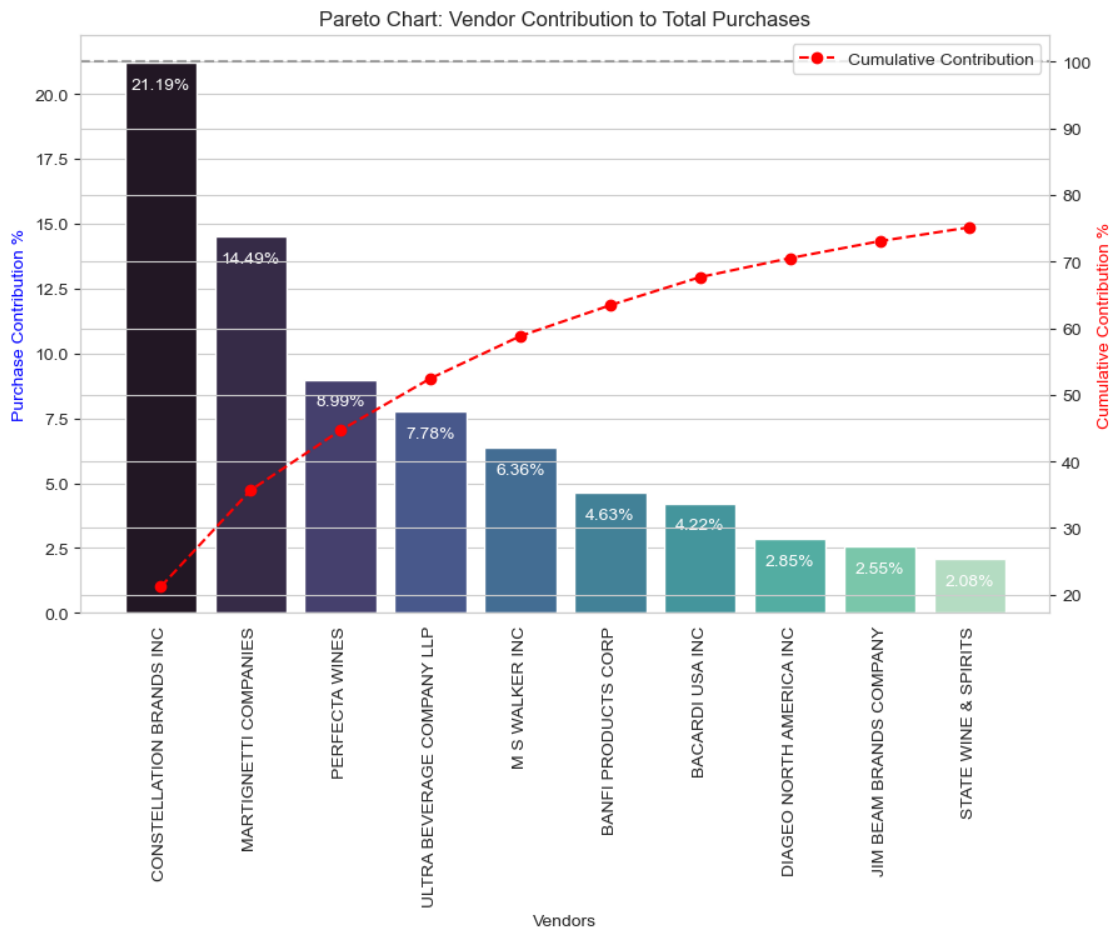
```
[153]: 75.14
```

```
[154]: top_vendors['Cumulative_Contribution%'] = top_vendors['PurchaseContribution%'].
         ↪cumsum()
       top_vendors
```

[154]:
```
                   VendorName TotalPurchaseDollars GrossProfit  \
9     CONSTELLATION BRANDS INC              155.18K      25.12K
33        MARTIGNETTI COMPANIES              106.12K     137.86K
42               PERFECTA WINES               65.88K      48.39K
63  ULTRA BEVERAGE COMPANY LLP               56.97K      91.76K
31               M S WALKER INC               46.58K      71.11K
3           BANFI PRODUCTS CORP               33.93K       3.67K
2               BACARDI USA INC               30.94K      30.66K
14     DIAGEO NORTH AMERICA INC               20.88K      21.13K
26      JIM BEAM BRANDS COMPANY               18.69K      22.74K
55         STATE WINE & SPIRITS               15.26K      22.06K

   TotalSalesDollars  PurchaseContribution%  Cumulative_Contribution%
9            180.31K                  21.19                     21.19
33           243.98K                  14.49                     35.68
42           114.28K                   8.99                     44.67
63           148.73K                   7.78                     52.45
31           117.69K                   6.36                     58.81
3             37.59K                   4.63                     63.44
2             61.60K                   4.22                     67.66
14            42.01K                   2.85                     70.51
26            41.43K                   2.55                     73.06
55            37.32K                   2.08                     75.14
```

[155]:
```python
fig, ax1 = plt.subplots(figsize=(10, 6))

# Bar plot for Purchase Contribution%
sns.barplot(x=top_vendors["VendorName"],
 y=top_vendors["PurchaseContribution%"], palette="mako", ax=ax1)

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%"],
 color='red', marker='o', linestyle='dashed', label='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='grey', linestyle='dashed', alpha=0.7)
```

```
ax2.legend(loc='upper right')

plt.show()
```



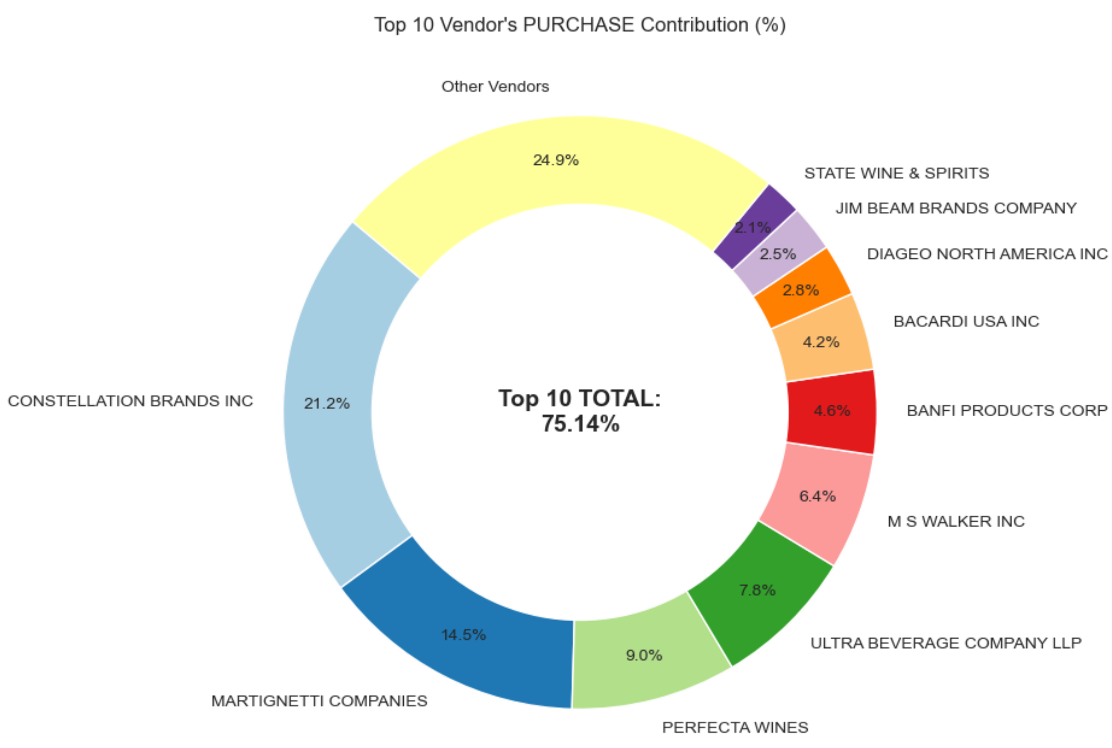
How much total procurement is dependent on the top vendors?

[159]:
```
print(f"TOTAL PURCHASE CONTRIBUTION OF TOP 10 VENDORS IS␣
↪{round(top_vendors['PurchaseContribution%'].sum(), 2)}%")
```

```
TOTAL PURCHASE CONTRIBUTION OF TOP 10 VENDORS IS 75.14%
```

[160]:
```
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,␣
↪autopct='%1.1f%%', startangle=140, pctdistance=0.85, colors=plt.cm.Paired.
↪colors)

# 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,␣
↪fontweight='bold', ha='center', va='center')

plt.title("Top 10 Vendor's PURCHASE Contribution (%)")

plt.show()
```



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

[161]:
```
df['UnitPurchasePrice'] = df['TotalPurchaseDollars] /␣
↪df['TotalPurchaseQuantity']
```

[163]:
```
df["OrderSize"] = pd.qcut(df["TotalPurchaseQuantity"], q=3,␣
↪labels=["Small","Medium","Large"])
```

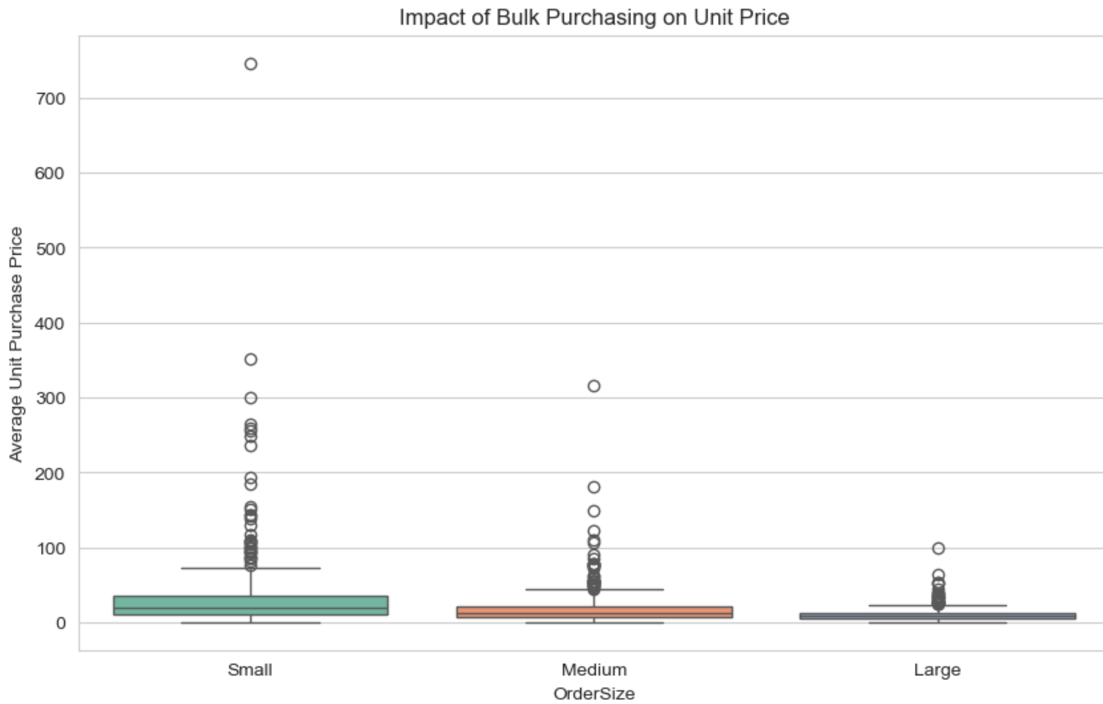
[167]:
```
df[["OrderSize","TotalPurchaseQuantity"]]
```

[167]:
```
    OrderSize  TotalPurchaseQuantity
0       Large                  11199
1       Large                   4194
2       Large                   5723
3       Large                    249
4       Large                   2241
..        …                      …
977     Small                      1
978     Small                      2
979     Small                      6
980     Small                      1
981     Small                      1

[982 rows x 2 columns]
```

[168]:
```
df.groupby('OrderSize')[['UnitPurchasePrice']].mean()
```

[168]:
```
           UnitPurchasePrice
OrderSize
Small              35.900600
Medium             20.687564
Large              11.248938
```

[169]:
```
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("OrderSize")
plt.ylabel("Average Unit Purchase Price")
plt.show()
```



- Vendors buying in bulk (Large Order Size) secure the lowest unit price ($8.45 per unit), resulting in higher margins if they can manage inventory effectively.
- The price difference between Small and Large orders is significant (—65% reduction in unit cost).
- This indicates that bulk pricing strategies effectively incentivize vendors to purchase in larger quantities, driving higher overall sales despite reduced per-unit revenue.

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

[175]:
```
top_threshold = df["TotalSalesDollars"].quantile(0.75)
low_threshold = df ["TotalSalesDollars"].quantile(0.25)
```

[176]:
```
top_vendors = df[df["TotalSalesDollars"] >= top_threshold]["ProfitMargin"].
  ↪dropna()
low_vendors = df[df["TotalSalesDollars"] <= low_threshold]["ProfitMargin"].
  ↪dropna()
```

[177]:
```
top_vendors
```

[177]:
```
0       11.482132
1        5.163561
2        5.121991
3       16.717781
4       25.982794
```

```
         …
812    96.916213
814    97.877620
916    98.763142
923    98.766655
938    98.837168
Name: ProfitMargin, Length: 246, dtype: float64
```

[178]:

```
low_vendors
```

[178]:

```
498     7.320644
505    14.980228
518     0.167224
533    11.739608
553     5.351885
         …
977    95.617902
978    55.778894
979    96.269133
980    31.610942
981    99.114603
Name: ProfitMargin, Length: 246, dtype: float64
```

[179]:

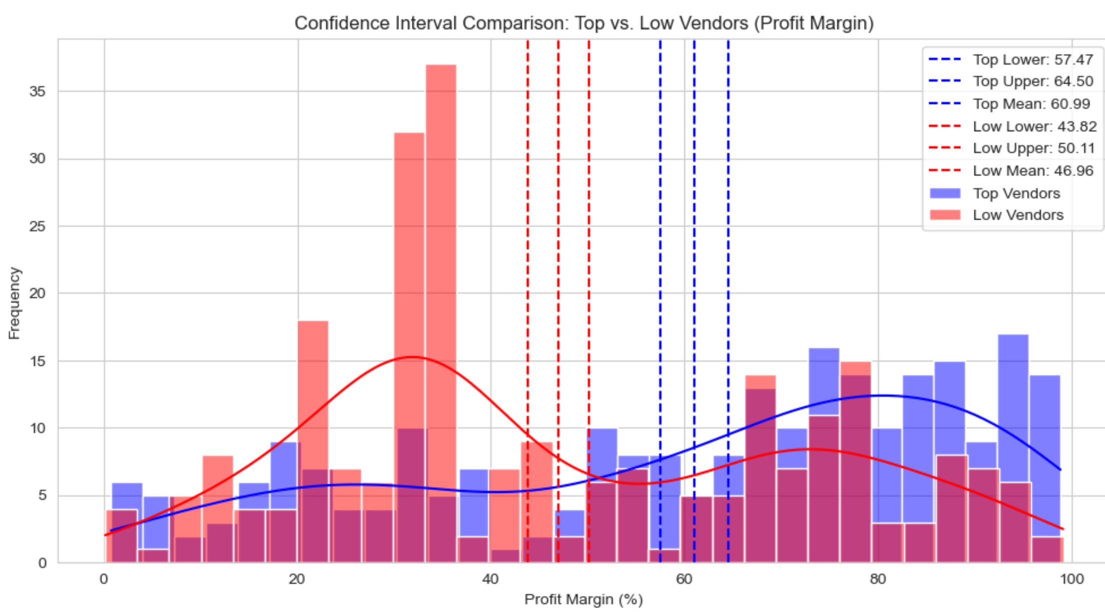
```
def confidence_interval(data, confidence=0.95):
    mean_val = np.mean(data)
    std_err = np.std(data, ddof=1) / np.sqrt(len(data)) # Standard error
    t_critical = stats.t.ppf((1 + confidence) / 2, df=len(data) - 1)
    margin_of_error = t_critical * std_err
    return mean_val, mean_val - margin_of_error,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

[181]:

```
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_mean:
 ↪.2f}")
print(f"Low Vendors 95% CI: ({low_lower:.2f}, {low_upper:.2f}), Mean: {low_mean:
 ↪.2f}")

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

# Top Vendors Plot
sns.histplot(top_vendors, kde=True, color="blue", bins=30, alpha=0.5,␣
 ↪label="Top Vendors")
plt.axvline(top_lower, color="blue", linestyle="--", label=f"Top Lower:␣
 ↪{top_lower:.2f}")
plt.axvline(top_upper, color="blue", linestyle="--", label=f"Top Upper:␣
 ↪{top_upper:.2f}")
```

```
plt.axvline(top_mean, color="blue", linestyle="--", label=f"Top Mean: {top_mean:
 ↪.2f}")

# Low Vendors Plot
sns.histplot(low_vendors, kde=True, color="red", bins=30, alpha=0.5, label="Low␣
 ↪Vendors")
plt.axvline(low_lower, color="red", linestyle="--", label=f"Low Lower:␣
 ↪{low_lower:.2f}")
plt.axvline(low_upper, color="red", linestyle="--", label=f"Low Upper:␣
 ↪{low_upper:.2f}")
plt.axvline(low_mean, color="red", linestyle="--", label=f"Low Mean: {low_mean:.
 ↪2f}")

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

```
Top Vendors 95% CI: (57.47, 64.50), Mean: 60.99
Low Vendors 95% CI: (43.82, 50.11), Mean: 46.96
```



The confidence interval for top-performing vendors (57.47% to 64.50%) is significantly higher than that of low-performing vendors (43.82% to 50.11%). The mean profit margin for top vendors is 60.99%, while for low vendors it is 46.96%.

This suggests that vendors with higher sales tend to maintain higher profit margins, potentially due to economies of scale, stronger brand recognition, or more efficient operations.

For High-Performing Vendors: To maintain or further improve profitability, they could explore optimizing their existing strategies, investing in innovation, or expanding into new markets.

For Low-Performing Vendors: Despite lower margins, their existing market position might indicate a need for strategic adjustments such as competitive pricing, improved product offerings, or enhanced marketing efforts to increase sales volume and profitability.

[ ]:

```

```