In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from datetime import datetime
import os, time
import logging

from sqlalchemy import create_engine

import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

In [2]:
```
path = '/content/drive/MyDrive/MY_LEARNING/Inventory_Analysis'
data_path = '/content/drive/MyDrive/MY_LEARNING/Inventory_Analysis/Data'
```

In [3]:
```
today = datetime.now().strftime('%Y-%m-%d')

logging.basicConfig(
    filename = f'{path}/logs/{today}.log',
    level = logging.DEBUG,
    format ='%(asctime)s - %(levelname)s - %(message)s',
    filemode = 'a'
)
```

In [4]:
```
for file in os.listdir(f'{data_path}'):
  print(file)
```

```
purchase_prices.csv
begin_inventory.csv
end_inventory.csv
invoice_purchases.csv
purchases.csv
sales.csv
inventory analysis.zip
```

In [5]:
```
for file in os.listdir(f'{data_path}'):
  if '.csv' in file:
    df = pd.read_csv(f'{data_path}/{file}')
    print(f'file name: {file} -> No. of rows: {df.shape[0]} | No. of Columns: {d
```

```
file name: purchase_prices.csv -> No. of rows: 12261 | No. of Columns: 9
file name: begin_inventory.csv -> No. of rows: 206529 | No. of Columns: 9
file name: end_inventory.csv -> No. of rows: 224489 | No. of Columns: 9
file name: invoice_purchases.csv -> No. of rows: 5543 | No. of Columns: 10
file name: purchases.csv -> No. of rows: 2372474 | No. of Columns: 16
file name: sales.csv -> No. of rows: 1048575 | No. of Columns: 14
```

In [6]:
```
engine = create_engine('sqlite:///invetory.db')

def ingest_db(df, table_name, engine):
  '''
  This function will ingest a dataframe into a database.
  '''
  df.to_sql(table_name, engine, if_exists='replace', index=False)
```

```python
def load_raw_data():
  '''
  This function will load the CSVs as dataframes and ingest them into the databa
  '''
  start = time.time()
  for file in os.listdir(f'{data_path}'):
    if '.csv' in file:
      df = pd.read_csv(f'{data_path}/{file}')
      logging.info(f'Ingesting {file} in database..')
      ingest_db(df, file.replace('.csv', ''), engine)
  end = time.time()
  total_time = (end - start)/60
  logging.info(f'-----------Ingestion Complete----------')
  logging.info(f'Total time taken fro ingestion {round(total_time, 2)} minutes')

if __name__ == '__main__':
  load_raw_data()
```

In [7]:
```python
def ingest_db(df, table_name, engine):
  '''
  This function will ingest a dataframe into a database.
  '''
  df.to_sql(table_name, engine, if_exists='replace', index=False)
```

In [8]:
```python
for file in os.listdir(f'{data_path}'):
  if '.csv' in file:
    print(file)
    df = pd.read_csv(f'{data_path}/{file}')
    logging.info(f'Ingesting {file} in database..')
    ingest_db(df, file.replace('.csv', ''), engine)
print(f'-----------Ingestion Complete----------')
```

```
purchase_prices.csv
begin_inventory.csv
end_inventory.csv
invoice_purchases.csv
purchases.csv
sales.csv
-----------Ingestion Complete----------
```

Data preprocessing

In [9]:
```python
import sqlite3
```

In [10]:
```python
# creating database connection
conn = sqlite3.connect('invetory.db')
```

In [11]:
```python
# checking tables present in the database
tables = pd.read_sql_query("SELECT name FROM sqlite_master WHERE type = 'table'"
tables
```

Out[11]:

	name
0	purchase_prices
1	begin_inventory
2	end_inventory
3	invoice_purchases
4	purchases
5	sales

In [12]:

```
pd.read_sql("select count(*) from purchases", conn)
```

Out[12]:

	count(*)
0	2372474

In [13]:

```
for table in tables.name:
  print('-'*50, f'{table}', '-'*50)
  print('Count of records: ', pd.read_sql(f'select count(*) as count from {table
  display(pd.read_sql(f'select * from {table} limit 5', conn))
```

```
-------------------------------------------------- purchase_prices --------------
------------------------------------
Count of records:  12261
```

	Brand	Description	Price	Size	Volume	Classification	PurchasePrice	VendorNumb
0	58	Gekkeikan Black & Gold Sake	12.99	750mL	750	1	9.28	83
1	62	Herradura Silver Tequila	36.99	750mL	750	1	28.67	11
2	63	Herradura Reposado Tequila	38.99	750mL	750	1	30.46	11
3	72	No. 3 London Dry Gin	34.99	750mL	750	1	26.11	91
4	75	Three Olives Tomato Vodka	14.99	750mL	750	1	10.94	72

```
-------------------------------------------------- begin_inventory --------------
------------------------------------
Count of records:  206529
```

	InventoryId	Store	City	Brand	Description	Size	onHand	Price
0	1_HARDERSFIELD_58	1	HARDERSFIELD	58	Gekkeikan Black & Gold Sake	750mL	8	12.99
1	1_HARDERSFIELD_60	1	HARDERSFIELD	60	Canadian Club 1858 VAP	750mL	7	10.99
2	1_HARDERSFIELD_62	1	HARDERSFIELD	62	Herradura Silver Tequila	750mL	6	36.99
3	1_HARDERSFIELD_63	1	HARDERSFIELD	63	Herradura Reposado Tequila	750mL	3	38.99
4	1_HARDERSFIELD_72	1	HARDERSFIELD	72	No. 3 London Dry Gin	750mL	6	34.99

```
-------------------------------------------------- end_inventory ----------------
----------------------------------
Count of records:  224489
```

	InventoryId	Store	City	Brand	Description	Size	onHand	Price
0	1_HARDERSFIELD_58	1	HARDERSFIELD	58	Gekkeikan Black & Gold Sake	750mL	11	12.99
1	1_HARDERSFIELD_62	1	HARDERSFIELD	62	Herradura Silver Tequila	750mL	7	36.99
2	1_HARDERSFIELD_63	1	HARDERSFIELD	63	Herradura Reposado Tequila	750mL	7	38.99
3	1_HARDERSFIELD_72	1	HARDERSFIELD	72	No. 3 London Dry Gin	750mL	4	34.99
4	1_HARDERSFIELD_75	1	HARDERSFIELD	75	Three Olives Tomato Vodka	750mL	7	14.99

```
-------------------------------------------------- invoice_purchases ------------
--------------------------------------
Count of records:  5543
```

	VendorNumber	VendorName	InvoiceDate	PONumber	PODate	PayDate	Quantity
0	105	ALTAMAR BRANDS LLC	2016-01-04	8124	2015-12-21	2016-02-16	6
1	4466	AMERICAN VINTAGE BEVERAGE	2016-01-07	8137	2015-12-22	2016-02-21	15
2	388	ATLANTIC IMPORTING COMPANY	2016-01-09	8169	2015-12-24	2016-02-16	5
3	480	BACARDI USA INC	2016-01-12	8106	2015-12-20	2016-02-05	10100
4	516	BANFI PRODUCTS CORP	2016-01-07	8170	2015-12-24	2016-02-12	1935

```
-------------------------------------------------- purchases -------------------
------------------------------
Count of records:  2372474
```

	InventoryId	Store	Brand	Description	Size	VendorNumber	VendorNam
0	69_MOUNTMEND_8412	69	8412	Tequila Ocho Plata Fresno	750mL	105	ALTAMA BRANDS LL
1	30_CULCHETH_5255	30	5255	TGI Fridays Ultimte Mudslide	1.75L	4466	AMERICA VINTAG BEVERAG
2	34_PITMERDEN_5215	34	5215	TGI Fridays Long Island Iced	1.75L	4466	AMERICA VINTAG BEVERAG
3	1_HARDERSFIELD_5255	1	5255	TGI Fridays Ultimte Mudslide	1.75L	4466	AMERICA VINTAG BEVERAG
4	76_DONCASTER_2034	76	2034	Glendalough Double Barrel	750mL	388	ATLANTI IMPORTIN COMPAN

```
-------------------------------------------------- sales -----------------------
--------------------------
Count of records:  1048575
```

	InventoryId	Store	Brand	Description	Size	SalesQuantity	SalesDollars
0	1_HARDERSFIELD_1004	1	1004	Jim Beam w/2 Rocks Glasses	750mL	1	16.49
1	1_HARDERSFIELD_1004	1	1004	Jim Beam w/2 Rocks Glasses	750mL	2	32.98
2	1_HARDERSFIELD_1004	1	1004	Jim Beam w/2 Rocks Glasses	750mL	1	16.49
3	1_HARDERSFIELD_1004	1	1004	Jim Beam w/2 Rocks Glasses	750mL	1	14.49
4	1_HARDERSFIELD_1005	1	1005	Maker's Mark Combo Pack	375mL 2 Pk	2	69.98

In [14]:
```
print(pd.read_sql('select VendorNumber, count(*) as count from purchases group b
```

```
     VendorNumber   count
0            3960  243326
1           12546  189832
2            1392  185574
3            4425  176781
4            3252  162567
..            ...     ...
121        201359       1
122          9099       1
123          4901       1
124          1439       1
125            54       1

[126 rows x 2 columns]
```

In [15]:
```
purchases_3960 = pd.read_sql('select * from purchases where VendorNumber = 3960'
purchases_3960
```

Out[15]:

	InventoryId	Store	Brand	Description	Size	VendorNumber	Ven
0	5_SUTTON_2443	5	2443	Seagrams VO	1.75L	3960	AME
1	1_HARDERSFIELD_2468	1	2468	Crown Royal Nrth Harvest Rye	750mL	3960	AME
2	76_DONCASTER_8726	76	8726	Baileys Irish Cream	50mL	3960	AME
3	34_PITMERDEN_388	34	388	Smirnoff 80 Proof	Liter	3960	AME
4	67_EANVERNESS_8513	67	8513	Baileys Irish Cream Mini's 4	100mL 4 Pk	3960	AME
...	...	...	...	...	...	...	
243321	69_MOUNTMEND_8178	69	8178	Bulleit Bourbon	750mL	3960	AME
243322	50_MOUNTMEND_8680	50	8680	Crown Royal	1.75L	3960	AME
243323	27_MOUNTMEND_2886	27	2886	Buchanans Deluxe 12Yr Scotch	1.75L	3960	AME
243324	33_HORNSEY_2862	33	2862	Pinch 15-Yr by Haig & Haig	750mL	3960	AME
243325	31_HORNSEY_2894	31	2894	Scoresby Rare	1.75L	3960	AME

243326 rows × 16 columns

In [16]:

```
purchase_price_3960 = pd.read_sql('select * from purchase_prices where VendorNum
purchase_price_3960
```

Out[16]:

	Brand	Description	Price	Size	Volume	Classification	PurchasePrice	Vendoı
0	305	Crown Royal Canadian Whisky	27.99	1000mL	1000	1	22.21	
1	388	Smirnoff 80 Proof	14.49	1000mL	1000	1	11.41	
2	485	Seagrams 7 Crown	12.99	1000mL	1000	1	10.39	
3	497	Capt Morgan Spiced Rum	19.99	1000mL	1000	1	14.70	
4	567	Ketel One Vodka	28.99	1000mL	1000	1	20.71	
...	...	...	...	...	...	...	...	
444	2248	Rhetoric 22 Year Bourbon	109.99	750mL	750	1	80.87	
445	3135	Smirnoff Cinn Sugar Twist	12.99	750mL	750	1	9.48	
446	3442	Smirnoff Grape Vodka	12.99	750mL	750	1	9.55	
447	4763	Ron Zacapa XO Rum	109.99	750mL	750	1	83.96	
448	6199	Zwack Liqueur	18.99	750mL	750	1	14.61	

449 rows × 9 columns

In [17]:

```
vendor_invoice_3960 = pd.read_sql('select * from invoice_purchases where VendorN
vendor_invoice_3960
```

Out[17]:

	VendorNumber	VendorName	InvoiceDate	PONumber	PODate	PayDate	Quantity
0	3960	DIAGEO NORTH AMERICA INC	2016-01-10	8173	2015-12-24	2016-02-19	16602
1	3960	DIAGEO NORTH AMERICA INC	2016-01-14	8271	2015-12-31	2016-02-11	100183
2	3960	DIAGEO NORTH AMERICA INC	2016-01-19	8310	2016-01-03	2016-02-23	75392
3	3960	DIAGEO NORTH AMERICA INC	2016-01-26	8459	2016-01-13	2016-02-27	76438
4	3960	DIAGEO NORTH AMERICA INC	2016-02-03	8575	2016-01-21	2016-03-06	94829
5	3960	DIAGEO NORTH AMERICA INC	2016-02-09	8622	2016-01-25	2016-03-13	89086
6	3960	DIAGEO NORTH AMERICA INC	2016-02-17	8721	2016-02-01	2016-03-20	79259
7	3960	DIAGEO NORTH AMERICA INC	2016-02-29	8899	2016-02-12	2016-03-31	83623
8	3960	DIAGEO NORTH AMERICA INC	2016-03-07	8966	2016-02-17	2016-04-11	92377
9	3960	DIAGEO NORTH AMERICA INC	2016-03-10	9018	2016-02-21	2016-04-09	86177
10	3960	DIAGEO NORTH AMERICA INC	2016-03-21	9199	2016-03-04	2016-04-29	81147
11	3960	DIAGEO NORTH AMERICA INC	2016-03-28	9302	2016-03-11	2016-04-30	81771
12	3960	DIAGEO NORTH AMERICA INC	2016-04-02	9320	2016-03-13	2016-05-13	80495
13	3960	DIAGEO NORTH AMERICA INC	2016-04-06	9451	2016-03-22	2016-05-10	106205
14	3960	DIAGEO NORTH AMERICA INC	2016-04-15	9585	2016-03-31	2016-05-26	82443

	VendorNumber	VendorName	InvoiceDate	PONumber	PODate	PayDate	Quantity
15	3960	DIAGEO NORTH AMERICA INC	2016-04-18	9624	2016-04-03	2016-05-24	75649
16	3960	DIAGEO NORTH AMERICA INC	2016-05-01	9766	2016-04-13	2016-06-02	87795
17	3960	DIAGEO NORTH AMERICA INC	2016-05-07	9904	2016-04-22	2016-06-14	112194
18	3960	DIAGEO NORTH AMERICA INC	2016-05-10	9919	2016-04-24	2016-06-11	98001
19	3960	DIAGEO NORTH AMERICA INC	2016-05-19	10023	2016-05-01	2016-06-21	94602
20	3960	DIAGEO NORTH AMERICA INC	2016-05-31	10126	2016-05-09	2016-07-04	86153
21	3960	DIAGEO NORTH AMERICA INC	2016-06-06	10260	2016-05-17	2016-07-14	117416
22	3960	DIAGEO NORTH AMERICA INC	2016-06-09	10415	2016-05-27	2016-07-13	104451
23	3960	DIAGEO NORTH AMERICA INC	2016-06-15	10468	2016-05-31	2016-07-24	108170
24	3960	DIAGEO NORTH AMERICA INC	2016-06-20	10566	2016-06-07	2016-08-04	109575
25	3960	DIAGEO NORTH AMERICA INC	2016-06-29	10700	2016-06-16	2016-08-09	99841
26	3960	DIAGEO NORTH AMERICA INC	2016-07-10	10754	2016-06-20	2016-08-20	82971
27	3960	DIAGEO NORTH AMERICA INC	2016-07-12	10851	2016-06-25	2016-08-16	18886
28	3960	DIAGEO NORTH AMERICA INC	2016-07-16	10936	2016-06-27	2016-08-13	131712
29	3960	DIAGEO NORTH AMERICA INC	2016-07-26	11028	2016-07-03	2016-08-18	131032

	VendorNumber	VendorName	InvoiceDate	PONumber	PODate	PayDate	Quantity
30	3960	DIAGEO NORTH AMERICA INC	2016-08-01	11191	2016-07-14	2016-09-07	128861
31	3960	DIAGEO NORTH AMERICA INC	2016-08-03	11300	2016-07-21	2016-09-14	114609
32	3960	DIAGEO NORTH AMERICA INC	2016-08-16	11329	2016-07-24	2016-09-14	106690
33	3960	DIAGEO NORTH AMERICA INC	2016-08-17	11493	2016-08-04	2016-10-02	99788
34	3960	DIAGEO NORTH AMERICA INC	2016-08-30	11589	2016-08-10	2016-10-05	119801
35	3960	DIAGEO NORTH AMERICA INC	2016-09-01	11670	2016-08-16	2016-10-13	100897
36	3960	DIAGEO NORTH AMERICA INC	2016-09-09	11794	2016-08-24	2016-10-06	112335
37	3960	DIAGEO NORTH AMERICA INC	2016-09-13	11907	2016-09-01	2016-10-19	108240
38	3960	DIAGEO NORTH AMERICA INC	2016-09-21	11996	2016-09-07	2016-10-25	90424
39	3960	DIAGEO NORTH AMERICA INC	2016-09-29	12044	2016-09-11	2016-11-13	116967
40	3960	DIAGEO NORTH AMERICA INC	2016-10-08	12149	2016-09-18	2016-11-07	118819
41	3960	DIAGEO NORTH AMERICA INC	2016-10-10	12264	2016-09-23	2016-11-07	1742
42	3960	DIAGEO NORTH AMERICA INC	2016-10-13	12397	2016-09-30	2016-11-27	111132
43	3960	DIAGEO NORTH AMERICA INC	2016-10-24	12470	2016-10-05	2016-11-24	110018
44	3960	DIAGEO NORTH AMERICA INC	2016-10-25	12579	2016-10-12	2016-12-05	112498

	VendorNumber	VendorName	InvoiceDate	PONumber	PODate	PayDate	Quantity
45	3960	DIAGEO NORTH AMERICA INC	2016-11-03	12618	2016-10-16	2016-12-14	122740
46	3960	DIAGEO NORTH AMERICA INC	2016-11-12	12771	2016-10-26	2016-12-24	127553
47	3960	DIAGEO NORTH AMERICA INC	2016-11-15	12833	2016-10-30	2016-12-19	141660
48	3960	DIAGEO NORTH AMERICA INC	2016-11-29	12981	2016-11-09	2016-12-30	119586
49	3960	DIAGEO NORTH AMERICA INC	2016-12-05	13032	2016-11-13	2016-12-30	89252
50	3960	DIAGEO NORTH AMERICA INC	2016-12-07	13190	2016-11-23	2017-01-16	116491
51	3960	DIAGEO NORTH AMERICA INC	2016-12-13	13258	2016-11-28	2017-01-29	116065
52	3960	DIAGEO NORTH AMERICA INC	2016-12-21	13354	2016-12-05	2017-01-30	134333
53	3960	DIAGEO NORTH AMERICA INC	2016-12-31	13501	2016-12-14	2017-02-08	124996
54	3960	DIAGEO NORTH AMERICA INC	2017-01-10	13594	2016-12-20	2017-02-05	129816

In [18]:
```
sales_3960 = pd.read_sql('select * from sales where VendorNo = 3960', conn)
sales_3960
```

Out[18]:

	InventoryId	Store	Brand	Description	Size	SalesQuantity	SalesD
0	1_HARDERSFIELD_1380	1	1380	Bulleit 95 Rye	1.75L	1	
1	1_HARDERSFIELD_1380	1	1380	Bulleit 95 Rye	1.75L	1	
2	1_HARDERSFIELD_1380	1	1380	Bulleit 95 Rye	1.75L	1	
3	1_HARDERSFIELD_1380	1	1380	Bulleit 95 Rye	1.75L	1	
4	1_HARDERSFIELD_1384	1	1384	Bulleit 95 Rye	750mL	1	
...	...	...	...	...	...	...	
125864	19_WINTERVALE_3882	19	3882	Smirnoff Strawberry Vodka	750mL	1	
125865	19_WINTERVALE_3882	19	3882	Smirnoff Strawberry Vodka	750mL	1	
125866	19_WINTERVALE_3882	19	3882	Smirnoff Strawberry Vodka	750mL	1	
125867	19_WINTERVALE_3882	19	3882	Smirnoff Strawberry Vodka	750mL	2	
125868	19_WINTERVALE_3882	19	3882	Smirnoff Strawberry Vodka	750mL	1	

125869 rows × 14 columns

In [19]:

```
purchases_3960.groupby(['Brand','PurchasePrice'])[['Quantity','Dollars']].sum()
```

Out[19]:

		Quantity	Dollars
Brand	**PurchasePrice**		
86	**236.21**	7	1653.47
181	**18.65**	6	111.90
187	**11.53**	12	138.36
188	**11.19**	69	772.11
202	**22.30**	168	3746.40
...	**...**	...	...
8801	**0.78**	108	84.24
8893	**12.68**	4739	60090.52
8898	**18.83**	4944	93095.52
8962	**13.84**	3944	54584.96
9145	**0.75**	96151	72113.25

396 rows × 2 columns

In [20]:
```
vendor_invoice_3960['PONumber'].nunique()
```

Out[20]:
```
55
```

In [21]:
```
vendor_invoice_3960.shape
```

Out[21]:
```
(55, 10)
```

In [22]:
```
vendor_invoice_3960.columns
```

Out[22]:
```
Index(['VendorNumber', 'VendorName', 'InvoiceDate', 'PONumber', 'PODate',
       'PayDate', 'Quantity', 'Dollars', 'Freight', 'Approval'],
      dtype='object')
```

In [23]:
```
sales_3960.groupby('Brand')[['SalesDollars','SalesPrice','SalesQuantity']].sum()
```

Out[23]:

Brand	SalesDollars	SalesPrice	SalesQuantity
86	299.99	299.99	1
148	14.49	14.49	1
150	14.49	14.49	1
187	179.88	14.99	12
188	179.88	14.99	12
...	...	...	...
8801	75.24	39.60	76
8893	7866.37	6405.23	463
8898	11253.67	8446.75	433
8962	2270.86	1831.08	114
9145	6570.63	1070.19	6637

332 rows × 3 columns

Observations:

- The purchases table contains actual purchase data, including the date of purchase, products (brands) purchased by vendors, amount paid in dollars, and the quantity purchased.
- The purchase price column is derived from the purchase_prices table, which provides product-wise actual and purchase prices. The combination of vendor and brand is unique in this table.
- The vendor_invoice table aggregates data from the purchases table, summerizing quantity and dollars amonuts along with an additional column for freight. The table maintains uniqueness based on vendor and PO number.
- The sales table captures actual sale transactions, dealing the brands purchased by vendors, the quantity sold, the selling price and the revenue earned.

As the data that we need for analysis is distributed in different tables, we need to create a summary table containing:

- purchase transaction made by vendors
- sales transaction data
- freight cost for each vendor
- actual product prices from vendors

In [24]:

```
freight_summary = pd.read_sql("""SELECT
VendorNumber,
SUM(Freight) as FreightCost
```

```
FROM invoice_purchases
GROUP BY VendorNumber""", conn)
freight_summary
```

Out[24]:

	VendorNumber	FreightCost
0	2	27.08
1	54	0.48
2	60	367.52
3	105	62.39
4	200	6.19
...	...	...
121	98450	856.02
122	99166	130.09
123	172662	178.34
124	173357	202.50
125	201359	0.09

126 rows × 2 columns

In [25]:

```
pd.read_sql_query("""SELECT
p.VendorNumber,
p.VendorName,
p.Brand,
p.PurchasePrice,
pp.Volume,
pp.Price as ActualPrice,
SUM(p.Quantity) as TotalPurchaseQuantity,
SUM(p.Dollars) as TotalPurchaseDollars
FROM purchases p
JOIN purchase_prices pp
on p.Brand = pp.Brand
WHERE p.PurchasePrice > 0
GROUP BY p.VendorNumber, p.VendorName, p.Brand
ORDER BY TotalPurchaseDollars""", conn)
```

Out[25]:

	VendorNumber	VendorName	Brand	PurchasePrice	Volume	ActualPrice	Total
0	7245	PROXIMO SPIRITS INC.	3065	0.71	50	0.99	
1	3960	DIAGEO NORTH AMERICA INC	6127	1.47	200	1.99	
2	3924	HEAVEN HILL DISTILLERIES	9123	0.74	50	0.99	
3	8004	SAZERAC CO INC	5683	0.39	50	0.49	
4	9815	WINE GROUP INC	8527	1.32	750	4.99	
...	...	...	...	...	...	...	
10687	3960	DIAGEO NORTH AMERICA INC	3545	21.89	1750	29.99	
10688	3960	DIAGEO NORTH AMERICA INC	4261	16.17	1750	22.99	
10689	17035	PERNOD RICARD USA	8068	18.24	1750	24.99	
10690	4425	MARTIGNETTI COMPANIES	3405	23.19	1750	28.99	
10691	1128	BROWN-FORMAN CORP	1233	26.27	1750	36.99	

10692 rows × 8 columns

In [26]:

```
pd.read_sql_query("""SELECT
VendorNo,
Brand,
SUM(SalesDollars) as TotalSalesDollars,
SUM(SalesPrice) as TotalSalesPrice,
SUM(SalesQuantity) as TotalSalesQuantity,
SUM(ExciseTax) as TotalExcise
FROM sales
GROUP BY VendorNo, Brand
ORDER BY TotalSalesDollars""", conn)
```

Out[26]:

	VendorNo	Brand	TotalSalesDollars	TotalSalesPrice	TotalSalesQuantity	TotalExci
0	8004	5287	0.98	0.98	2	0.
1	3960	3303	0.99	0.99	1	0.(
2	9206	2773	0.99	0.99	1	0.(
3	9625	8872	0.99	0.99	1	0.(
4	3252	3933	1.98	0.99	2	0.
...	...	...	...	...	...	
7653	4425	3405	275162.97	52289.50	9203	16909.
7654	17035	8068	288135.11	48202.30	11189	20557.!
7655	1128	1233	344712.22	64889.97	9578	17598.
7656	3960	3545	357759.17	52774.51	11883	21833..
7657	3960	4261	444810.74	43304.31	20226	37163.:

7658 rows × 6 columns

In [27]:

```
start = time.time()
final_table = pd.read_sql_query(""" SELECT
pp.VendorNumber,
pp.VendorName,
pp.Brand,
pp.Price AS ActualPrice,
pp.PurchasePrice,
pp.Volume,
SUM(s.SalesDollars) as TotalSalesDollars,
SUM(s.SalesPrice) as TotalSalesPrice,
SUM(s.SalesQuantity) as TotalSalesQuantity,
SUM(s.ExciseTax) as TotalExcise,
SUM(vi.Quantity) as TotalPurchaseQuantity,
SUM(vi.Dollars) as TotalPurchaseDollars,
SUM(vi.Freight) as FreightCost
FROM purchase_prices pp
JOIN sales S
  ON pp.VendorNumber = s.VendorNo
  AND pp.Brand = s.Brand
JOIN invoice_purchases vi
  ON pp.VendorNumber = vi.VendorNumber
GROUP BY pp.VendorNumber, pp.Brand, pp.Price, pp.PurchasePrice
""",conn)
end = time.time()
total_time = (end - start)/60
```

In [28]:

```
final_table
```

Out[28]:

	VendorNumber	VendorName	Brand	ActualPrice	PurchasePrice	Volume	TotalS
0	2	IRA GOLDMAN AND WILLIAMS, LLP	90085	36.99	23.86	750	
1	105	ALTAMAR BRANDS LLC	8412	49.99	35.71	750	
2	105	ALTAMAR BRANDS LLC	8419	6.99	5.30	100	
3	287	APPOLO VINEYARDS LLC	24921	15.49	10.40	750	
4	287	APPOLO VINEYARDS LLC	24922	15.49	10.47	750	
...	...	...	...	...	...	...	
7651	98450	Serralles Usa LLC	6877	18.99	13.56	750	
7652	98450	Serralles Usa LLC	7890	29.99	22.21	750	
7653	98450	Serralles Usa LLC	8543	19.99	14.81	750	
7654	172662	SWEETWATER FARM	4215	25.99	19.40	750	
7655	173357	TAMWORTH DISTILLING	3909	24.99	19.37	750	

7656 rows × 13 columns

In [29]:

```
total_time
```

Out[29]:

```
3.5056567668914793
```

Performance Optimization

- The quesry involves heavy joins and aggregations on large dataset like sales and purchases.
- Storing the pre-agregated results avoids repeated expensive computation.
- Helps in analyzing sales, purchases and pricing for different vendores and brands.
- Future Benefits of storing this data for faster Dashboarding and reporting.
- Instead of running expensive queries each time, dashboards can fetch data quickely from vendor_sales_summary.

In [30]:

```
final_table.dtypes
```

Out[30]:

	0
VendorNumber	int64
VendorName	object
Brand	int64
ActualPrice	float64
PurchasePrice	float64
Volume	object
TotalSalesDollars	float64
TotalSalesPrice	float64
TotalSalesQuantity	int64
TotalExcise	float64
TotalPurchaseQuantity	int64
TotalPurchaseDollars	float64
FreightCost	float64

dtype: object

Observation

- Column Volume has datatype object but storing numering data.

In [31]:

```
final_table.isnull().sum()
```

Out[31]:

	0
VendorNumber	0
VendorName	0
Brand	0
ActualPrice	0
PurchasePrice	0
Volume	0
TotalSalesDollars	0
TotalSalesPrice	0
TotalSalesQuantity	0
TotalExcise	0
TotalPurchaseQuantity	0
TotalPurchaseDollars	0
FreightCost	0

dtype: int64

In [32]:
```
final_table.head()
```

Out[32]:

	VendorNumber	VendorName	Brand	ActualPrice	PurchasePrice	Volume	TotalSales
0	2	IRA GOLDMAN AND WILLIAMS, LLP	90085	36.99	23.86	750	
1	105	ALTAMAR BRANDS LLC	8412	49.99	35.71	750	2
2	105	ALTAMAR BRANDS LLC	8419	6.99	5.30	100	
3	287	APPOLO VINEYARDS LLC	24921	15.49	10.40	750	
4	287	APPOLO VINEYARDS LLC	24922	15.49	10.47	750	

In [33]:
```
final_table['VendorName'].unique()
```

```
Out[33]: array(['IRA GOLDMAN AND WILLIAMS, LLP          ',
       'ALTAMAR BRANDS LLC         ', 'APPOLO VINEYARDS LLC       ',
       'ATLANTIC IMPORTING COMPANY ', 'BACARDI USA INC            ',
       'BANFI PRODUCTS CORP        ', 'STATE WINE & SPIRITS   ',
       'SAZERAC NORTH AMERICA INC. ', 'BRONCO WINE COMPANY        ',
       'BROWN-FORMAN CORP          ', 'BULLY BOY DISTILLERS       ',
       'BLACK ROCK SPIRITS LLC     ', 'CALEDONIA SPIRITS INC      ',
       'CONSTELLATION BRANDS INC   ', 'CAPSTONE INTERNATIONAL     ',
       'CASTLE BRANDS CORP.        ', 'VINEYARD BRANDS INC        ',
       'DIAGEO CHATEAU ESTATE WINES', 'VRANKEN AMERICA            ',
       'Circa Wines                ', 'FABRIZIA SPIRITS LLC       ',
       'ALISA CARR BEVERAGES           ', 'SOUTHERN WINE & SPIRITS NE ',
       'DELICATO VINEYARDS INC     ', 'BLACK PRINCE DISTILLERY INC',
       'DJINN SPIRITS LLC          ', 'DUGGANS DISTILLED PRODUCTS ',
       'DISARONNO INTERNATIONAL LLC', 'EDRINGTON AMERICAS         ',
       'CENTEUR IMPORTS LLC        ', 'SIDNEY FRANK IMPORTING CO  ',
       'E & J GALLO WINERY         ', 'WILLIAM GRANT & SONS INC   ',
       'HEAVEN HILL DISTILLERIES   ', 'HOOD RIVER DISTILLERS, Inc.',
       'DIAGEO NORTH AMERICA INC   ', 'CHARLES JACQUIN ET CIE INC ',
       'MARTIGNETTI COMPANIES', 'MARTIGNETTI COMPANIES ',
       'AMERICAN VINTAGE BEVERAGE  ', 'KLIN SPIRITS LLC           ',
       'KOBRAND CORPORATION        ', 'LAIRD & CO                 ',
       'LATITUDE BEVERAGE COMPANY  ', 'MANGO BOTTLING INC         ',
       'MARSALLE COMPANY           ', 'MILTONS DISTRIBUTING CO    ',
       'MCCORMICK DISTILLING CO    ', 'MHW LTD                    ',
       'MOONLIGHT MEADERY          ', 'NICHE W & S                ',
       'OLE SMOKY DISTILLERY LLC   ', 'PALM BAY INTERNATIONAL INC ',
       'PARK STREET IMPORTS LLC    ', 'PINE STATE TRADING CO      ',
       'PREMIER DISTRIBUTORS       ', 'REMY COINTREAU USA INC     ',
       'PREMIUM PORT WINES, INC.   ', 'PROXIMO SPIRITS INC.       ',
       'PSP WINES ', 'R.P.IMPORTS INC            ',
       'SAZERAC CO INC             ', 'MOET HENNESSY USA INC      ',
       'SEA HAGG DISTILLERY LLC    ', 'SHAW ROSS INT L IMP LTD    ',
       'LUXCO INC                  ', 'STAR INDUSTRIES INC.       ',
       'STOLI GROUP,(USA) LLC      ', 'STE MICHELLE WINE ESTATES  ',
       'SURVILLE ENTERPRISES CORP  ', 'TRINCHERO FAMILY ESTATES   ',
       'TAKARA SAKE USA INC        ', 'ULTRA BEVERAGE COMPANY LLP ',
       'PHILLIPS PRODUCTS CO.      ', 'TY KU LLC                  ',
       'M S WALKER INC             ', 'WEIN BAUER INC             ',
       'WESTERN SPIRITS BEVERAGE CO', 'FREDERICK WILDMAN & SONS   ',
       'VINEDREA WINES LLC         ', 'WINE GROUP INC             ',
       'TREASURY WINE ESTATES      ', 'MAJESTIC FINE WINES        ',
       'Russian Standard Vodka     ', 'PERFECTA WINES             ',
       'CAMPARI AMERICA            ', 'STELLAR IMPORTING CO LLC   ',
       'JIM BEAM BRANDS COMPANY    ', 'FLAG HILL WINERY & VINEYARD',
       'JEWELL TOWNE VINEYARDS     ', 'SEA BREEZE CELLARS LLC     ',
       'PERNOD RICARD USA          ', 'Dunn Wine Brokers          ',
       'SMOKY QUARTZ DISTILLERY LLC', 'TALL SHIP DISTILLERY LLC   ',
       'VINEXTRA INC               ', 'FORTUNE WINE BROKERS LLC   ',
       'THE IMPORTED GRAPE LLC     ', 'VINILANDIA USA             ',
       'SILVER MOUNTAIN CIDERS     ', 'POVERTY LANE ORCHARDS      ',
       'LABELLE VYDS AND WINERY    ', 'FANTASY FINE WINES CORP    ',
       'THE PIERPONT GROUP LLC     ', 'CANDIA VINEYARDS           ',
       'CRUSH WINES                ', 'FULCHINO VINEYARD INC      ',
       'HAUNTING WHISPER VYDS      ', 'INCREDIBREW INC            ',
       'SWEET BABY VINEYARD        ', 'WALPOLE MTN VIEW WINERY    ',
       'ZORVINO VINEYARDS          ', 'Serralles Usa LLC          ',
       'SWEETWATER FARM         ', 'TAMWORTH DISTILLING        '],
      dtype=object)
```

In [34]:
```
final_table['Volume'] = final_table['Volume'].astype('float64')
```

In [35]:
```
final_table['VendorName'] = final_table['VendorName'].str.strip()
```

In [36]:
```
final_table.dtypes
```

Out[36]:

	0
VendorNumber	int64
VendorName	object
Brand	int64
ActualPrice	float64
PurchasePrice	float64
Volume	float64
TotalSalesDollars	float64
TotalSalesPrice	float64
TotalSalesQuantity	int64
TotalExcise	float64
TotalPurchaseQuantity	int64
TotalPurchaseDollars	float64
FreightCost	float64

dtype: object

In [37]:
```
final_table['VendorName'].unique()
```

```
Out[37]: array(['IRA GOLDMAN AND WILLIAMS, LLP', 'ALTAMAR BRANDS LLC',
                'APPOLO VINEYARDS LLC', 'ATLANTIC IMPORTING COMPANY',
                'BACARDI USA INC', 'BANFI PRODUCTS CORP', 'STATE WINE & SPIRITS',
                'SAZERAC NORTH AMERICA INC.', 'BRONCO WINE COMPANY',
                'BROWN-FORMAN CORP', 'BULLY BOY DISTILLERS',
                'BLACK ROCK SPIRITS LLC', 'CALEDONIA SPIRITS INC',
                'CONSTELLATION BRANDS INC', 'CAPSTONE INTERNATIONAL',
                'CASTLE BRANDS CORP.', 'VINEYARD BRANDS INC',
                'DIAGEO CHATEAU ESTATE WINES', 'VRANKEN AMERICA', 'Circa Wines',
                'FABRIZIA SPIRITS LLC', 'ALISA CARR BEVERAGES',
                'SOUTHERN WINE & SPIRITS NE', 'DELICATO VINEYARDS INC',
                'BLACK PRINCE DISTILLERY INC', 'DJINN SPIRITS LLC',
                'DUGGANS DISTILLED PRODUCTS', 'DISARONNO INTERNATIONAL LLC',
                'EDRINGTON AMERICAS', 'CENTEUR IMPORTS LLC',
                'SIDNEY FRANK IMPORTING CO', 'E & J GALLO WINERY',
                'WILLIAM GRANT & SONS INC', 'HEAVEN HILL DISTILLERIES',
                'HOOD RIVER DISTILLERS, Inc.', 'DIAGEO NORTH AMERICA INC',
                'CHARLES JACQUIN ET CIE INC', 'MARTIGNETTI COMPANIES',
                'AMERICAN VINTAGE BEVERAGE', 'KLIN SPIRITS LLC',
                'KOBRAND CORPORATION', 'LAIRD & CO', 'LATITUDE BEVERAGE COMPANY',
                'MANGO BOTTLING INC', 'MARSALLE COMPANY',
                'MILTONS DISTRIBUTING CO', 'MCCORMICK DISTILLING CO', 'MHW LTD',
                'MOONLIGHT MEADERY', 'NICHE W & S', 'OLE SMOKY DISTILLERY LLC',
                'PALM BAY INTERNATIONAL INC', 'PARK STREET IMPORTS LLC',
                'PINE STATE TRADING CO', 'PREMIER DISTRIBUTORS',
                'REMY COINTREAU USA INC', 'PREMIUM PORT WINES, INC.',
                'PROXIMO SPIRITS INC.', 'PSP WINES', 'R.P.IMPORTS INC',
                'SAZERAC CO INC', 'MOET HENNESSY USA INC',
                'SEA HAGG DISTILLERY LLC', 'SHAW ROSS INT L IMP LTD', 'LUXCO INC',
                'STAR INDUSTRIES INC.', 'STOLI GROUP,(USA) LLC',
                'STE MICHELLE WINE ESTATES', 'SURVILLE ENTERPRISES CORP',
                'TRINCHERO FAMILY ESTATES', 'TAKARA SAKE USA INC',
                'ULTRA BEVERAGE COMPANY LLP', 'PHILLIPS PRODUCTS CO.', 'TY KU LLC',
                'M S WALKER INC', 'WEIN BAUER INC', 'WESTERN SPIRITS BEVERAGE CO',
                'FREDERICK WILDMAN & SONS', 'VINEDREA WINES LLC', 'WINE GROUP INC',
                'TREASURY WINE ESTATES', 'MAJESTIC FINE WINES',
                'Russian Standard Vodka', 'PERFECTA WINES', 'CAMPARI AMERICA',
                'STELLAR IMPORTING CO LLC', 'JIM BEAM BRANDS COMPANY',
                'FLAG HILL WINERY & VINEYARD', 'JEWELL TOWNE VINEYARDS',
                'SEA BREEZE CELLARS LLC', 'PERNOD RICARD USA', 'Dunn Wine Brokers',
                'SMOKY QUARTZ DISTILLERY LLC', 'TALL SHIP DISTILLERY LLC',
                'VINEXTRA INC', 'FORTUNE WINE BROKERS LLC',
                'THE IMPORTED GRAPE LLC', 'VINILANDIA USA',
                'SILVER MOUNTAIN CIDERS', 'POVERTY LANE ORCHARDS',
                'LABELLE VYDS AND WINERY', 'FANTASY FINE WINES CORP',
                'THE PIERPONT GROUP LLC', 'CANDIA VINEYARDS', 'CRUSH WINES',
                'FULCHINO VINEYARD INC', 'HAUNTING WHISPER VYDS',
                'INCREDIBREW INC', 'SWEET BABY VINEYARD',
                'WALPOLE MTN VIEW WINERY', 'ZORVINO VINEYARDS',
                'Serralles Usa LLC', 'SWEETWATER FARM', 'TAMWORTH DISTILLING'],
               dtype=object)
```

```
In [38]: final_table['GrossProfit'] = final_table['TotalSalesDollars']-final_table['Total
```

```
In [39]: final_table
```

Out[39]:

	VendorNumber	VendorName	Brand	ActualPrice	PurchasePrice	Volume	TotalS
0	2	IRA GOLDMAN AND WILLIAMS, LLP	90085	36.99	23.86	750.0	
1	105	ALTAMAR BRANDS LLC	8412	49.99	35.71	750.0	
2	105	ALTAMAR BRANDS LLC	8419	6.99	5.30	100.0	
3	287	APPOLO VINEYARDS LLC	24921	15.49	10.40	750.0	
4	287	APPOLO VINEYARDS LLC	24922	15.49	10.47	750.0	
...	...	...	...	...	...	...	
7651	98450	Serralles Usa LLC	6877	18.99	13.56	750.0	
7652	98450	Serralles Usa LLC	7890	29.99	22.21	750.0	
7653	98450	Serralles Usa LLC	8543	19.99	14.81	750.0	
7654	172662	SWEETWATER FARM	4215	25.99	19.40	750.0	
7655	173357	TAMWORTH DISTILLING	3909	24.99	19.37	750.0	

7656 rows × 14 columns

In [40]:
```
final_table['ProfitMargin'] = (final_table['GrossProfit']/final_table['TotalSale
```

In [41]:
```
final_table['StockTurnover'] = (final_table['TotalSalesQuantity']/final_table['T
```

In [42]:
```
final_table['SalestoPurchaseRatio'] = (final_table['TotalSalesDollars']/final_ta
```

In [43]:
```
final_table.columns, final_table.dtypes
```

Out[43]:
```
(Index(['VendorNumber', 'VendorName', 'Brand', 'ActualPrice', 'PurchasePrice',
        'Volume', 'TotalSalesDollars', 'TotalSalesPrice', 'TotalSalesQuantity',
        'TotalExcise', 'TotalPurchaseQuantity', 'TotalPurchaseDollars',
        'FreightCost', 'GrossProfit', 'ProfitMargin', 'StockTurnover',
        'SalestoPurchaseRatio'],
       dtype='object'),
 VendorNumber              int64
 VendorName               object
 Brand                     int64
 ActualPrice             float64
 PurchasePrice           float64
 Volume                  float64
 TotalSalesDollars       float64
 TotalSalesPrice         float64
 TotalSalesQuantity        int64
 TotalExcise             float64
 TotalPurchaseQuantity     int64
 TotalPurchaseDollars    float64
 FreightCost             float64
 GrossProfit             float64
 ProfitMargin            float64
 StockTurnover           float64
 SalestoPurchaseRatio    float64
 dtype: object)
```

In [44]:
```
cursor = conn.cursor()
```

In [45]:
```
cursor.execute("""
CREATE TABLE vendor_sales_summary (
    VendorNumber INTEGER,
    VendorName TEXT,
    Description TEXT,
    Brand TEXT,
    ActualPrice REAL,
    PurchasePrice REAL,
    Volume REAL,
    TotalSalesDollars REAL,
    TotalSalesPrice REAL,
    TotalSalesQuantity REAL,
    TotalExcise REAL,
    TotalPurchaseQuantity REAL,
    TotalPurchaseDollars REAL,
    FreightCost REAL,
    GrossProfit REAL,
    ProfitMargin REAL,
    StockTurnover REAL,
    SalestoPurchaseRatio REAL,
    PRIMARY KEY (VendorNumber, Brand)
);
""")
```

Out[45]: <sqlite3.Cursor at 0x7ba0c6867a40>

In [46]:
```
pd.read_sql_query("SELECT * FROM vendor_sales_summary", conn)
```

Out[46]:

VendorNumber	VendorName	Brand	ActualPrice	PurchasePrice	Volume	TotalSalesD

In [47]:
```
final_table.to_sql('vendor_sales_summary', conn, if_exists='replace', index=Fals
```

Out[47]: 7656

In [48]:
```
pd.read_sql_query("SELECT * FROM vendor_sales_summary", conn)
```

Out[48]:

	VendorNumber	VendorName	Brand	ActualPrice	PurchasePrice	Volume	TotalS
0	2	IRA GOLDMAN AND WILLIAMS, LLP	90085	36.99	23.86	750.0	
1	105	ALTAMAR BRANDS LLC	8412	49.99	35.71	750.0	
2	105	ALTAMAR BRANDS LLC	8419	6.99	5.30	100.0	
3	287	APPOLO VINEYARDS LLC	24921	15.49	10.40	750.0	
4	287	APPOLO VINEYARDS LLC	24922	15.49	10.47	750.0	
...	...	...	...	...	...	...	
7651	98450	Serralles Usa LLC	6877	18.99	13.56	750.0	
7652	98450	Serralles Usa LLC	7890	29.99	22.21	750.0	
7653	98450	Serralles Usa LLC	8543	19.99	14.81	750.0	
7654	172662	SWEETWATER FARM	4215	25.99	19.40	750.0	
7655	173357	TAMWORTH DISTILLING	3909	24.99	19.37	750.0	

7656 rows × 17 columns

Exploratory Data Analysis

- We have examined the varioud tables in the database to identify key variables, understand their relationships, and determine which one should be included in the final analysis.
- In EDA, we will analyse the resultant table to gain insights into the distribution of each column. This will help us to understand data pattern, identify anamolies, and ensure data quality before proceeding with further analysis.

In [138…

```
# Loading database
conn = sqlite3.connect('invetory.db')
```

In [139…

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

In [140…

```
df.head()
```

Out[140…

	VendorNumber	VendorName	Brand	ActualPrice	PurchasePrice	Volume	TotalSales
0	2	IRA GOLDMAN AND WILLIAMS, LLP	90085	36.99	23.86	750.0	
1	105	ALTAMAR BRANDS LLC	8412	49.99	35.71	750.0	2
2	105	ALTAMAR BRANDS LLC	8419	6.99	5.30	100.0	
3	287	APPOLO VINEYARDS LLC	24921	15.49	10.40	750.0	
4	287	APPOLO VINEYARDS LLC	24922	15.49	10.47	750.0	

In [141…

```
#summary statistic
df.describe().T
```

Out[141…

	count	mean	std	min	25
VendorNumber	7656.0	9.729789e+03	1.649077e+04	2.000000e+00	3.924000e+
Brand	7656.0	1.738946e+04	1.316236e+04	5.800000e+01	5.187750e+
ActualPrice	7656.0	2.768236e+01	7.879555e+01	4.900000e-01	1.099000e+
PurchasePrice	7656.0	1.857552e+01	5.858169e+01	3.800000e-01	6.750000e+
Volume	7656.0	8.544943e+02	6.254644e+02	5.000000e+01	7.500000e+
TotalSalesDollars	7656.0	2.495342e+05	8.201858e+05	4.194000e+01	1.099684e+
TotalSalesPrice	7656.0	1.212089e+05	2.539583e+05	2.695000e+01	5.497800e+
TotalSalesQuantity	7656.0	1.828791e+04	5.728049e+04	6.000000e+00	5.927500e+
TotalExcise	7656.0	1.016275e+04	5.617761e+04	6.000000e-01	8.530250e+
TotalPurchaseQuantity	7656.0	2.500968e+08	7.781495e+08	2.000000e+01	2.640411e+
TotalPurchaseDollars	7656.0	2.327185e+09	7.288682e+09	2.315400e+02	2.489156e+
FreightCost	7656.0	1.185050e+07	3.692272e+07	1.080000e+00	1.270786e+
GrossProfit	7656.0	-2.326936e+09	7.288207e+09	-1.058425e+11	-1.309233e+
ProfitMargin	7656.0	-1.019623e+06	2.833533e+06	-9.358998e+07	-1.125731e+
StockTurnover	7656.0	2.931794e-01	2.690776e+00	1.007365e-03	5.150363e-
SalestoPurchaseRatio	7656.0	4.456610e-01	4.030777e+00	1.068489e-04	8.882327e-

In [142…

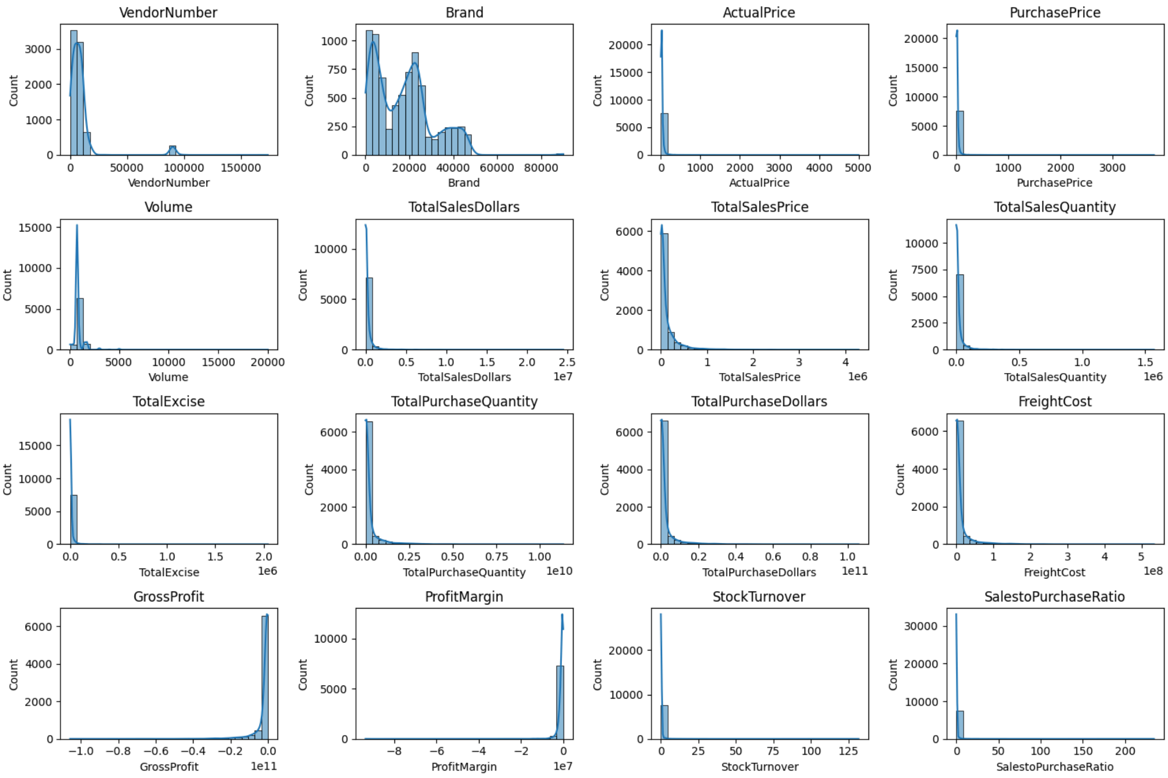
```
# Distribution plot for numerical columns
numerical_cols = df.select_dtypes(include=np.number).columns
numerical_cols
```

Out[142…

```
Index(['VendorNumber', 'Brand', 'ActualPrice', 'PurchasePrice', 'Volume',
       'TotalSalesDollars', 'TotalSalesPrice', 'TotalSalesQuantity',
       'TotalExcise', 'TotalPurchaseQuantity', 'TotalPurchaseDollars',
       'FreightCost', 'GrossProfit', 'ProfitMargin', 'StockTurnover',
       'SalestoPurchaseRatio'],
      dtype='object')
```

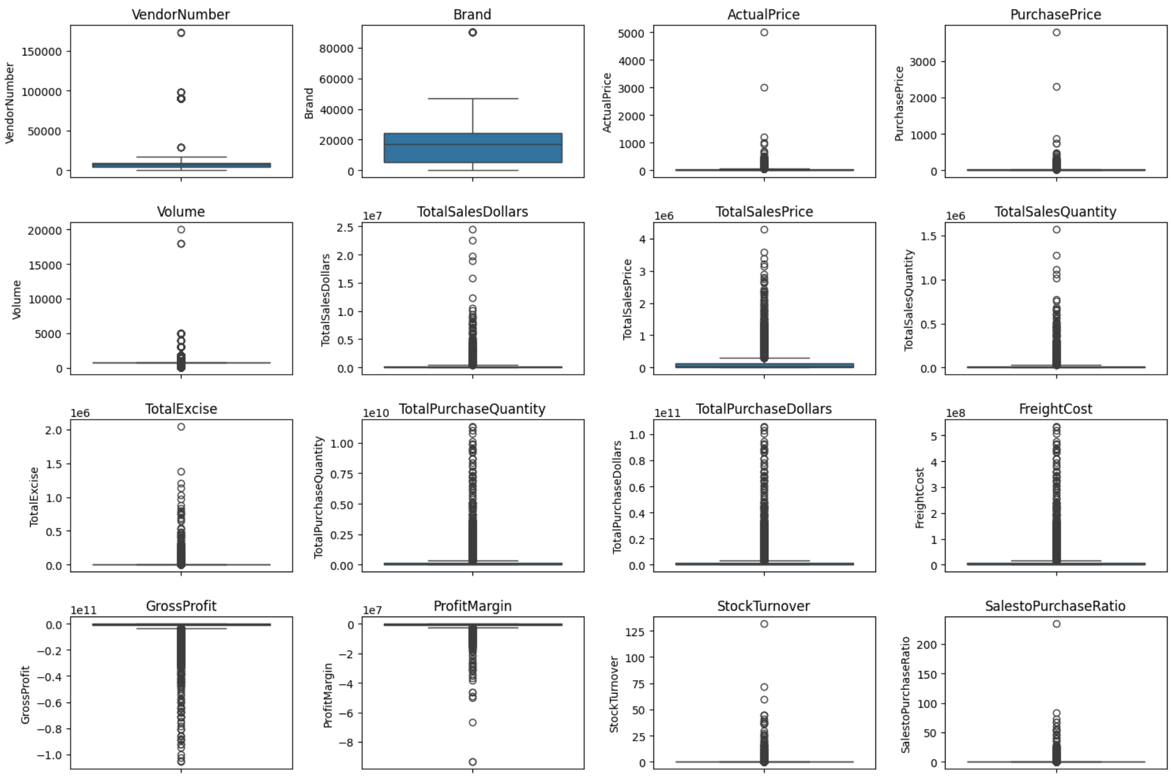
In [143…

```
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 [144…

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



Summary Statistic Insights:

Negative & Zero Values:

- Gross Profit: Minimum value is -105842452617.45, indicating losses. Someproducts or transaction may be seling at a loss due to high costs or selling at discounts lower than the purchase price.
- Profit Margin: Has a minimum of -93589976.85, which suggests cases where revenue is lower than costs.

In [145…

```
# Data after removing inconsistencies
df = pd.read_sql_query("""SELECT *
FROM vendor_sales_summary
WHERE GrossProfit > 0
OR ProfitMargin > 0
OR TotalSalesQuantity > 0
""", conn)
```

In [146…

```
df
```

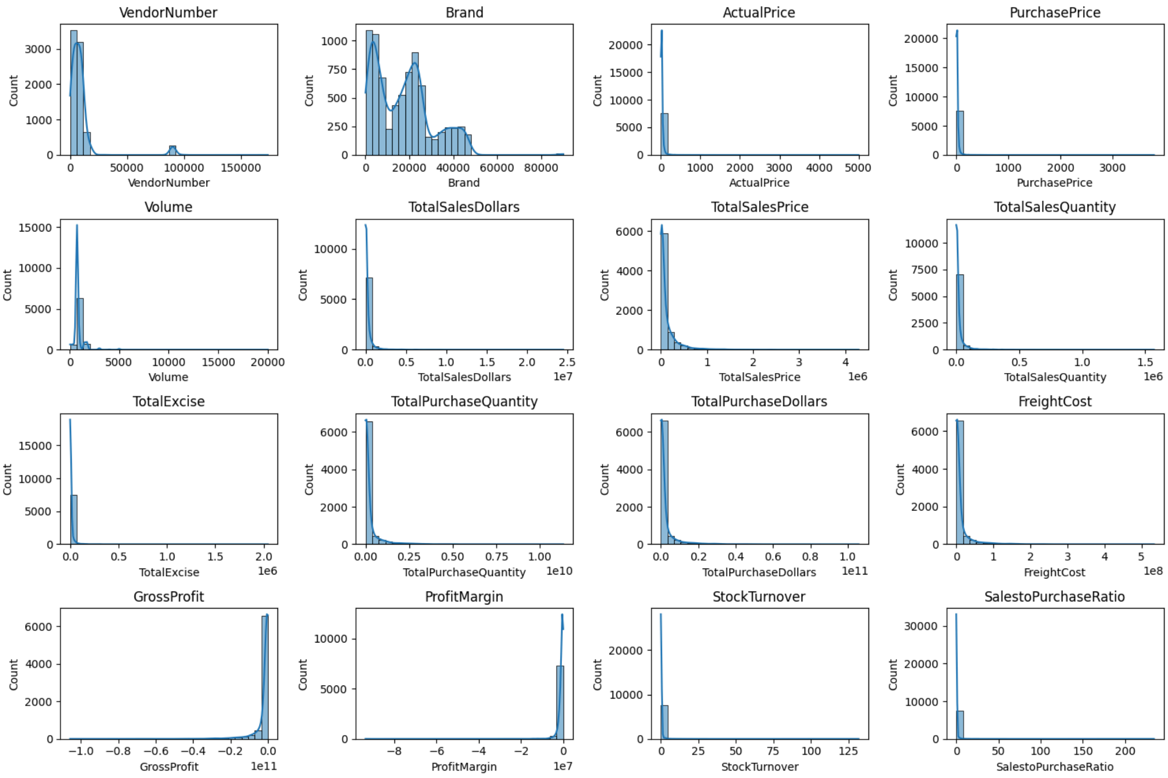
Out[146...

	VendorNumber	VendorName	Brand	ActualPrice	PurchasePrice	Volume	TotalS
0	2	IRA GOLDMAN AND WILLIAMS, LLP	90085	36.99	23.86	750.0	
1	105	ALTAMAR BRANDS LLC	8412	49.99	35.71	750.0	
2	105	ALTAMAR BRANDS LLC	8419	6.99	5.30	100.0	
3	287	APPOLO VINEYARDS LLC	24921	15.49	10.40	750.0	
4	287	APPOLO VINEYARDS LLC	24922	15.49	10.47	750.0	
...	...	...	...	...	...	...	
7651	98450	Serralles Usa LLC	6877	18.99	13.56	750.0	
7652	98450	Serralles Usa LLC	7890	29.99	22.21	750.0	
7653	98450	Serralles Usa LLC	8543	19.99	14.81	750.0	
7654	172662	SWEETWATER FARM	4215	25.99	19.40	750.0	
7655	173357	TAMWORTH DISTILLING	3909	24.99	19.37	750.0	

7656 rows × 17 columns

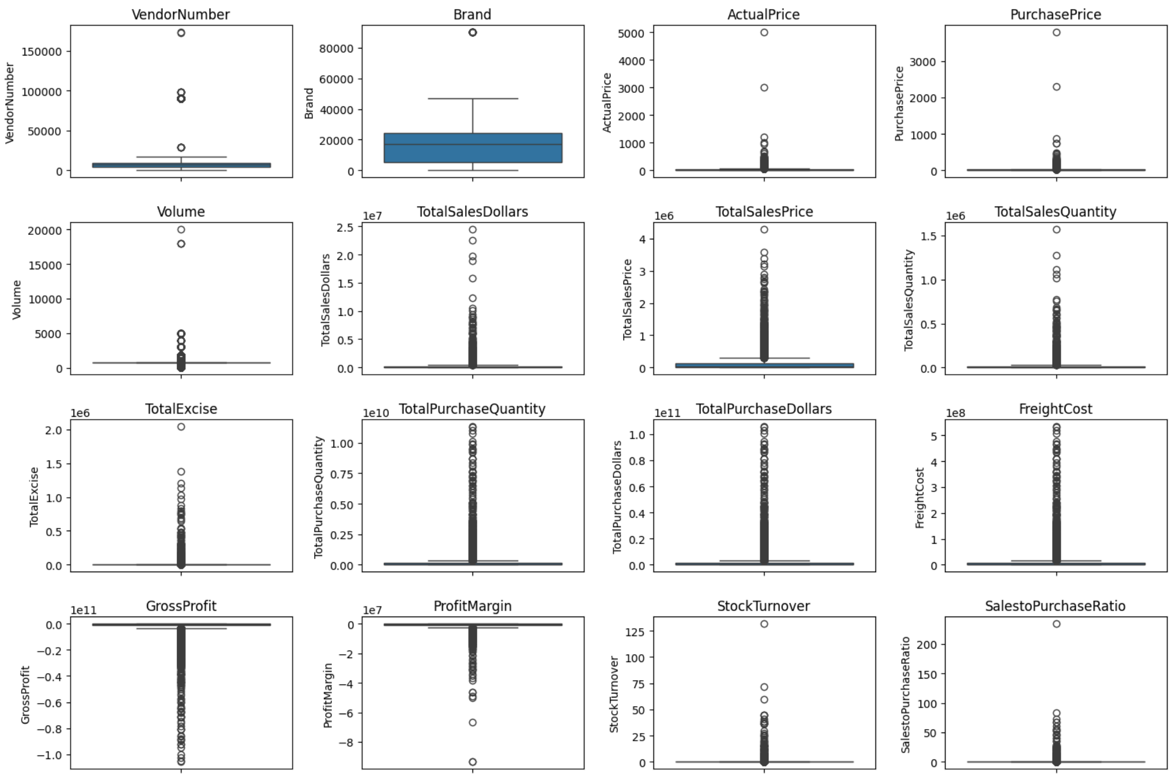
In [147...

```
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 [148…

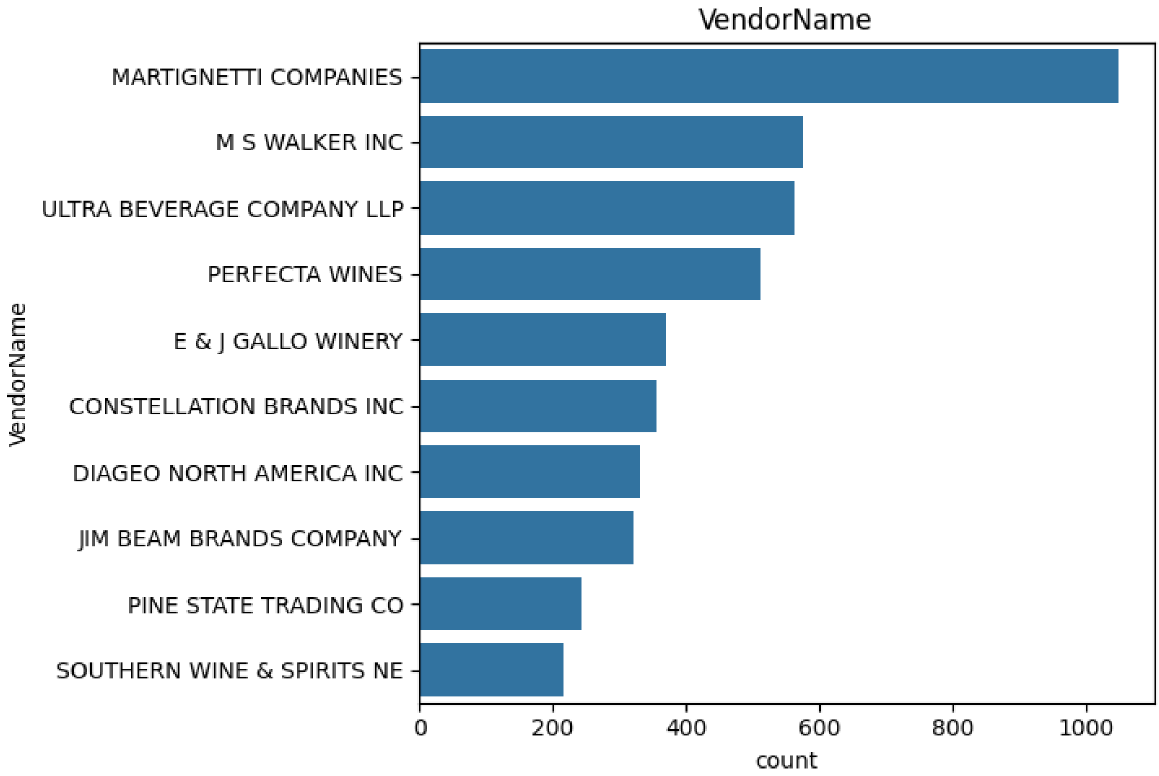
```
# Outlier Detection with Boxplot
plt.figure(figsize=(15,10))
for i, col in enumerate(numerical_cols):
  plt.subplot(4,4,i+1)
  sns.boxplot(df[col])
  plt.title(col)
plt.tight_layout()
plt.show()
```



In [149…

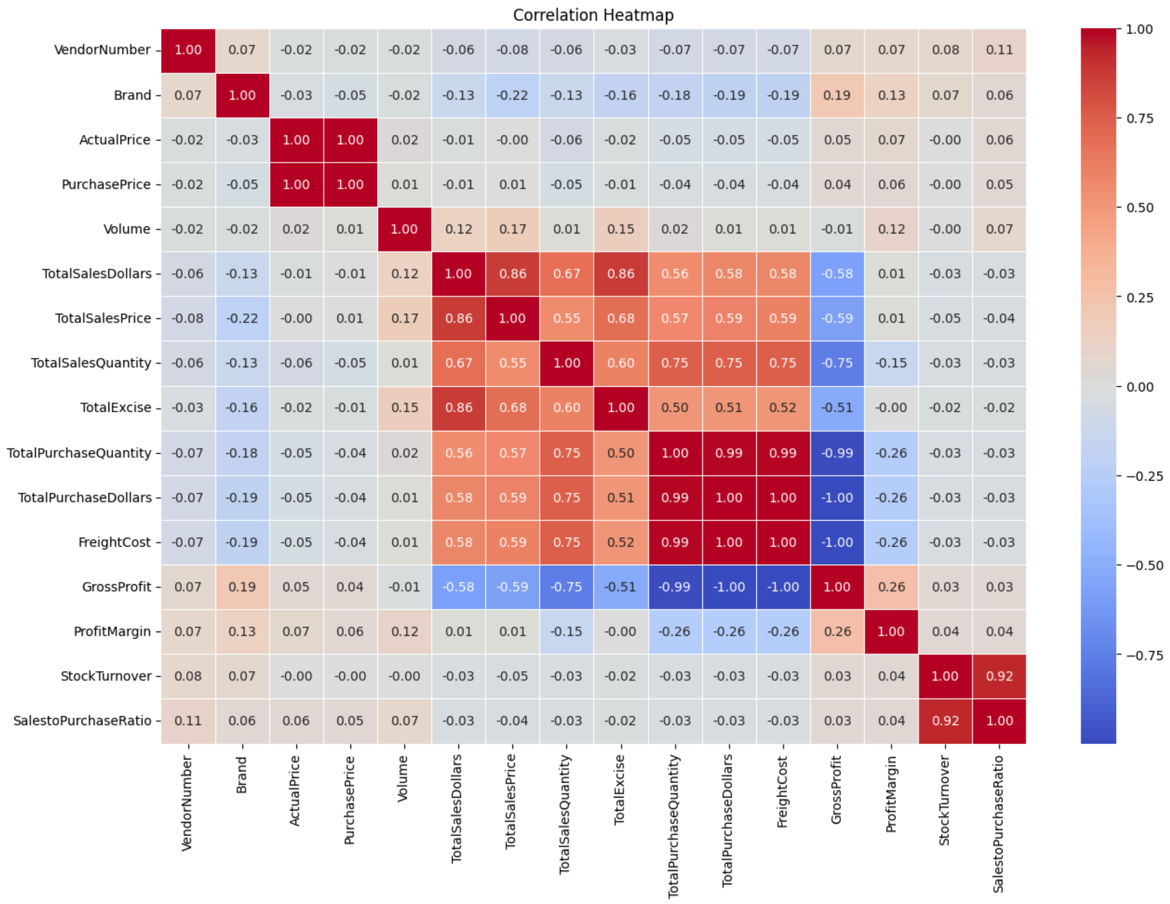
```
# Count plot for Categorical Columns
calregorical_cols = ['VendorName']
```

```
plt.figure(figsize=(12,5))
for i, col in enumerate(calregorical_cols):
  plt.subplot(1,2,i+1)
  sns.countplot(df[col], order=df[col].value_counts().index[:10])
  plt.title(col)
plt.tight_layout()
plt.show()
```



In [150…

```
# correlation Heatmap

plt.figure(figsize=(15,10))
correlation_matrix = df[numerical_cols].corr()
sns.heatmap(correlation_matrix, annot=True, fmt=".2f", cmap='coolwarm', linewidt
plt.title("Correlation Heatmap")
plt.show()
```



Correlation Insights

- PurchasePrice has weak correlations with TotalSalesDollars (-0.01) and GrossProfit (-0.04), suggesting that price variations da not significantiy impuct sales revenue or profit.
- Strong correlation between total purchase quantity and total sales quantity (0.75), 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 competiti pricing pressures.
- StockTurnover has weak negative correlations with both GrossProfit (-0.038) and ProfitMargin (-0.055), indicating that faster turnover does not necessanilty result in higher profitability.

Data Analysis

Identify Brands that needs promotional or pricing adjustments which exhibits lower sales performance but higher profit margines.

In [151…
```
brand_performance = df.groupby('Brand').agg({
    'TotalSalesDollars': 'sum',
    'ProfitMargin': 'mean'}).reset_index()
```

In [152…
```
brand_performance
```

Out[152...

	Brand	TotalSalesDollars	ProfitMargin
0	58	205761.60	-6.429281e+04
1	60	74731.80	-3.206182e+06
2	61	18466.80	-1.073011e+05
3	62	360380.90	-4.954549e+05
4	63	305397.95	-4.739206e+05
...	...	...	...
7651	90084	1215.62	-5.650429e+03
7652	90085	295.92	-1.802839e+03
7653	90086	987.81	-3.438300e+03
7654	90087	8929.81	-2.914045e+02
7655	90089	61554.87	-1.839067e+02

7656 rows × 3 columns

In [153...

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

In [154...

```
low_sales_threshold
```

Out[154...

```
np.float64(5266.800000000009)
```

In [155...

```
high_margin_threshold
```

Out[155...

```
np.float64(-64418.51062874756)
```

In [156...

```
# Filter brands with low sales but high profit margine

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:
```

	Brand	TotalSalesDollars	ProfitMargin
7573	46327	41.94	-452.074392
3943	17576	54.90	-3222.732240
1271	3473	142.87	-1616.448520
3967	17842	143.84	-12953.997497
7634	46950	159.80	-383.003755
...	...	...	...
752	2349	4949.45	-41347.757024
4179	18811	5035.80	-1184.065293
4105	18569	5058.90	-26891.496768
3293	13791	5218.26	-23032.506238
1288	3508	5241.25	-241.823420

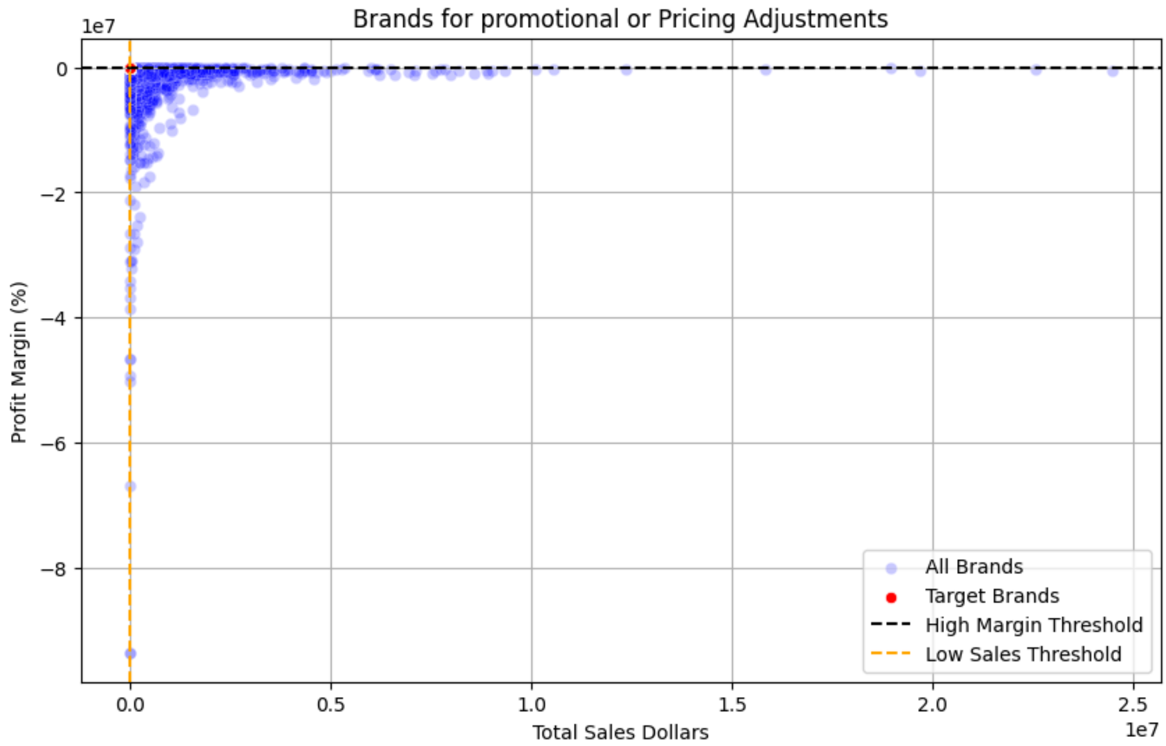
233 rows × 3 columns

In [135…

```
#brand_performance = brand_performance[brand_performance['TotalSalesDollars']<10
```

In [157…

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

plt.axhline(high_margin_threshold, color='black', linestyle='--', label='High Ma
plt.axvline(low_sales_threshold, linestyle='--', color='orange', label='Low Sale

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



Which Vendors and brands demonstrate the highesh sales performance?

In [164…

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

In [161…

```
# top vendors and brands by sales performance
top_vendors = df.groupby('VendorName')['TotalSalesDollars'].sum().nlargest(10)
top_brands = df.groupby('Brand')['TotalSalesDollars'].sum().nlargest(10)
top_brands
```

Out[161...

Brand	TotalSalesDollars
4261	2.446459e+07
3405	2.256336e+07
3545	1.967675e+07
1233	1.895917e+07
8068	1.584743e+07
3858	1.237578e+07
2589	1.054182e+07
4227	1.010130e+07
1376	9.345731e+06
2585	9.043463e+06

dtype: float64

In [162...

```
top_vendors
```

Out[162...

VendorName	TotalSalesDollars
DIAGEO NORTH AMERICA INC	2.657989e+08
MARTIGNETTI COMPANIES	2.556946e+08
JIM BEAM BRANDS COMPANY	1.387612e+08
PERNOD RICARD USA	1.218982e+08
CONSTELLATION BRANDS INC	1.033493e+08
BACARDI USA INC	9.149825e+07
E & J GALLO WINERY	8.400435e+07
ULTRA BEVERAGE COMPANY LLP	7.706078e+07
BROWN-FORMAN CORP	7.424386e+07
M S WALKER INC	7.154752e+07

dtype: float64

In [165...

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

Out[165…

VendorName	TotalSalesDollars
DIAGEO NORTH AMERICA INC	2657.99M
MARTIGNETTI COMPANIES	2556.95M
JIM BEAM BRANDS COMPANY	1387.61M
PERNOD RICARD USA	1218.98M
CONSTELLATION BRANDS INC	1033.49M
BACARDI USA INC	914.98M
E & J GALLO WINERY	840.04M
ULTRA BEVERAGE COMPANY LLP	770.61M
BROWN-FORMAN CORP	742.44M
M S WALKER INC	715.48M

dtype: object

In [166…

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

Out[166…

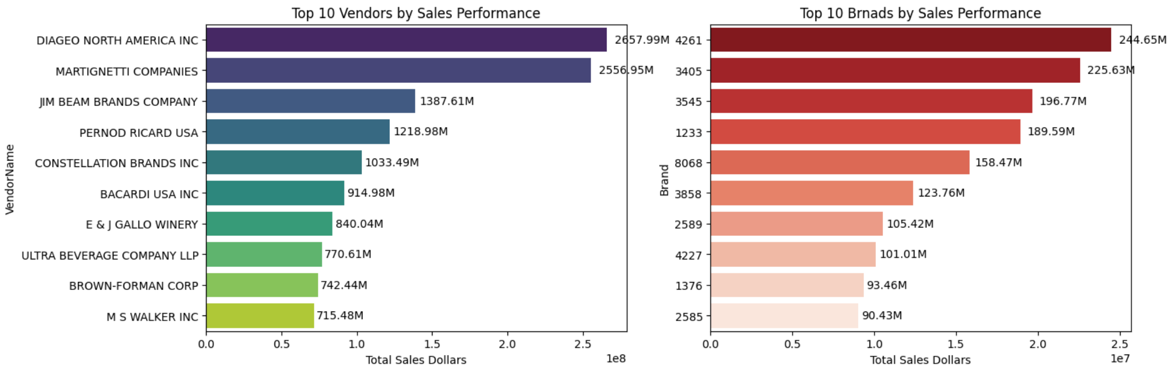
Brand	TotalSalesDollars
4261	244.65M
3405	225.63M
3545	196.77M
1233	189.59M
8068	158.47M
3858	123.76M
2589	105.42M
4227	101.01M
1376	93.46M
2585	90.43M

dtype: object

In [169…

```
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='viridis')
plt.title('Top 10 Vendors by Sales Performance')
plt.xlabel('Total Sales Dollars')
```

```python
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, palette='
plt.title('Top 10 Brnads by Sales Performance')
plt.xlabel('Total Sales Dollars')

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')
```



Which vendors contribute the most to the total purchase dollars?

In [173…

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

In [174…

```python
vendor_performance
```

Out[174...

	VendorName	TotalPurchaseDollars	GrossProfit	TotalSalesDollars
0	ALISA CARR BEVERAGES	1.502922e+06	-1.215825e+06	2.870974e+05
1	ALTAMAR BRANDS LLC	2.341240e+05	-2.024524e+05	3.167160e+04
2	AMERICAN VINTAGE BEVERAGE	6.738999e+07	-6.694060e+07	4.493891e+05
3	APPOLO VINEYARDS LLC	1.919760e+04	-1.758664e+04	1.610960e+03
4	ATLANTIC IMPORTING COMPANY	6.619728e+06	-6.283490e+06	3.362378e+05
...	...	...	...	...
109	WEIN BAUER INC	4.395217e+06	-4.205227e+06	1.899896e+05
110	WESTERN SPIRITS BEVERAGE CO	6.061762e+08	-6.040327e+08	2.143465e+06
111	WILLIAM GRANT & SONS INC	6.851280e+10	-6.848634e+10	2.645071e+07
112	WINE GROUP INC	1.825799e+11	-1.825390e+11	4.091067e+07
113	ZORVINO VINEYARDS	3.582705e+07	-3.544359e+07	3.834540e+05

114 rows × 4 columns

In [175...

```
vendor_performance['purchaceContribution%'] = vendor_performance['TotalPurchaseD
```

In [176...

```
vendor_performance
```

Out[176...

	VendorName	TotalPurchaseDollars	GrossProfit	TotalSalesDollars	purchaceCont
0	ALISA CARR BEVERAGES	1.502922e+06	-1.215825e+06	2.870974e+05	8.4
1	ALTAMAR BRANDS LLC	2.341240e+05	-2.024524e+05	3.167160e+04	1.3
2	AMERICAN VINTAGE BEVERAGE	6.738999e+07	-6.694060e+07	4.493891e+05	3.7
3	APPOLO VINEYARDS LLC	1.919760e+04	-1.758664e+04	1.610960e+03	1.0
4	ATLANTIC IMPORTING COMPANY	6.619728e+06	-6.283490e+06	3.362378e+05	3.7
...	...	...	...	...	
109	WEIN BAUER INC	4.395217e+06	-4.205227e+06	1.899896e+05	2.4
110	WESTERN SPIRITS BEVERAGE CO	6.061762e+08	-6.040327e+08	2.143465e+06	3.4
111	WILLIAM GRANT & SONS INC	6.851280e+10	-6.848634e+10	2.645071e+07	3.8
112	WINE GROUP INC	1.825799e+11	-1.825390e+11	4.091067e+07	1.0
113	ZORVINO VINEYARDS	3.582705e+07	-3.544359e+07	3.834540e+05	2.0

114 rows × 5 columns

In [184...

```
vendor_performance = round(vendor_performance.sort_values('purchaceContribution%
```

In [189...

```
vendor_performance.shape
```

Out[189...

```
(114, 5)
```

In [185...

```
# Display top 10 vendors
top_vendors = vendor_performance.head(10)
top_vendors['TotalSalesDollars'] = top_vendors['TotalSalesDollars'].apply(lambda
top_vendors['TotalPurchaseDollars'] = top_vendors['TotalPurchaseDollars'].apply(
top_vendors['GrossProfit'] = top_vendors['GrossProfit'].apply(lambda x:format_dc
```

In [188...

```
top_vendors
```

Out[188…

	VendorName	TotalPurchaseDollars	GrossProfit	TotalSalesDollars	purcha
24	DIAGEO NORTH AMERICA INC	64142586.70M	-6413992870835.86	2657.99M	
43	JIM BEAM BRANDS COMPANY	22764757.78M	-2276337017104.29	1387.61M	
54	MARTIGNETTI COMPANIES	21349019.98M	-2134646303155.36	2556.95M	
65	PERNOD RICARD USA	11612372.71M	-1161115373115.6	1218.98M	
19	CONSTELLATION BRANDS INC	11025087.96M	-1102405446630.65	1033.49M	
5	BACARDI USA INC	8039536.60M	-803862161441.29	914.98M	
29	E & J GALLO WINERY	8017740.32M	-801690027437.64	840.04M	
102	ULTRA BEVERAGE COMPANY LLP	5410142.72M	-540937211493.68	770.61M	
50	M S WALKER INC	5346411.72M	-534569624455.3	715.48M	
10	BROWN-FORMAN CORP	4217259.59M	-421651714679.64	742.44M	

In [191…

```
top_vendors['purchaceContribution%'].sum()*100
```

Out[191…

```
np.float64(92.0)
```

In [192…

```
top_vendors['Cumulative_Contribution%'] = top_vendors['purchaceContribution%'].c
```

In [202…

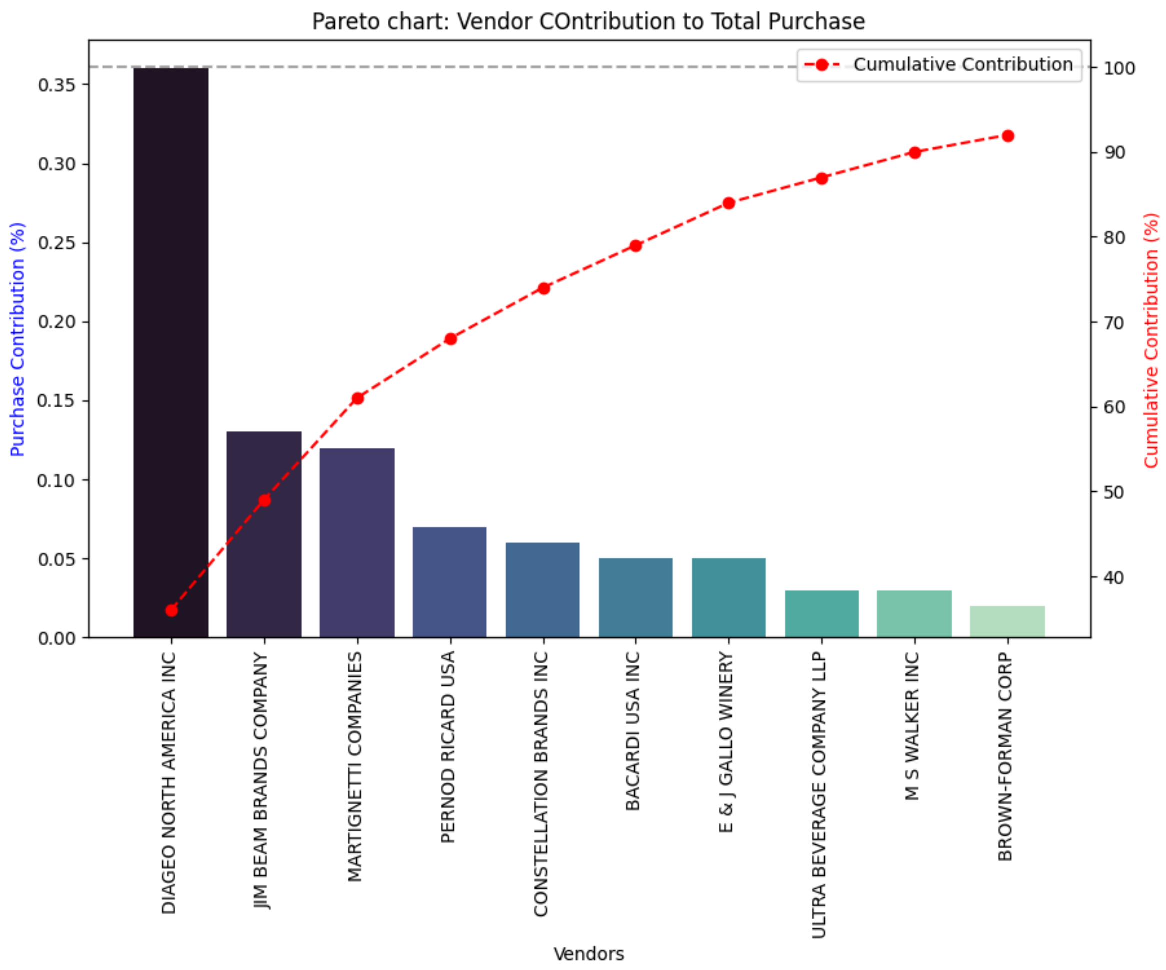
```
top_vendors['Cumulative_Contribution%'] = top_vendors['Cumulative_Contribution%'
top_vendors
```

Out[202...

	VendorName	TotalPurchaseDollars	GrossProfit	TotalSalesDollars	purcha
24	DIAGEO NORTH AMERICA INC	64142586.70M	-6413992870835.86	2657.99M	
43	JIM BEAM BRANDS COMPANY	22764757.78M	-2276337017104.29	1387.61M	
54	MARTIGNETTI COMPANIES	21349019.98M	-2134646303155.36	2556.95M	
65	PERNOD RICARD USA	11612372.71M	-1161115373115.6	1218.98M	
19	CONSTELLATION BRANDS INC	11025087.96M	-1102405446630.65	1033.49M	
5	BACARDI USA INC	8039536.60M	-803862161441.29	914.98M	
29	E & J GALLO WINERY	8017740.32M	-801690027437.64	840.04M	
102	ULTRA BEVERAGE COMPANY LLP	5410142.72M	-540937211493.68	770.61M	
50	M S WALKER INC	5346411.72M	-534569624455.3	715.48M	
10	BROWN-FORMAN CORP	4217259.59M	-421651714679.64	742.44M	

In [203...

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

# Bar plot for purchase contribution
sns.barplot(x = top_vendors['VendorName'], y = top_vendors['purchaceContribution

for i, value in enumerate(top_vendors['purchaceContribution%']):
  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%'], col

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 Purchase")

ax2.axhline(y=100, color = 'gray', linestyle = 'dashed', alpha = 0.7)
ax2.legend(loc='upper right')
plt.tight_layout()

plt.show()
```



Pareto chart: Vendor COntribution to Total Purchase
Cumulative Contribution
Purchase Contribution (%)
Cumulative Contribution (%)
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
40
50
60
70
80
90
100
DIAGEO NORTH AMERICA INC
JIM BEAM BRANDS COMPANY
MARTIGNETTI COMPANIES
PERNOD RICARD USA
CONSTELLATION BRANDS INC
BACARDI USA INC
E & J GALLO WINERY
ULTRA BEVERAGE COMPANY LLP
M S WALKER INC
BROWN-FORMAN CORP
Vendors

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

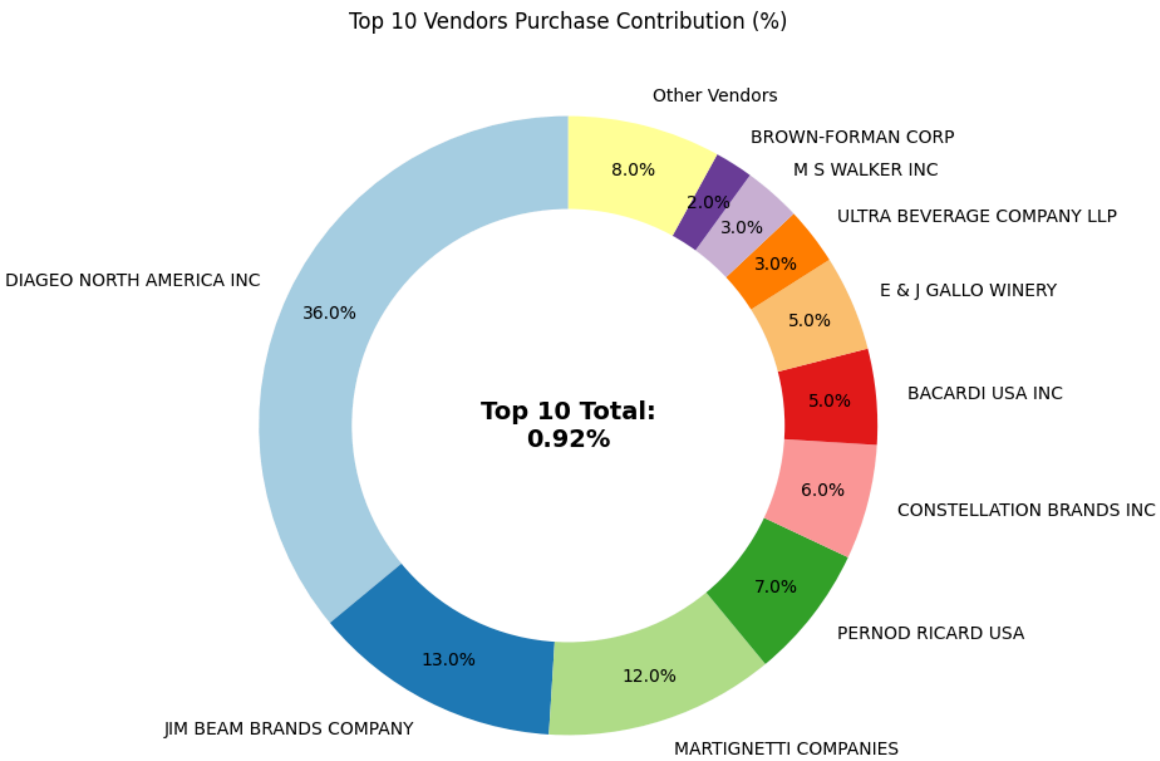
In [205…

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

```
Total purchase contribution of top 10 vendors is 98.0%
```

In [208…

```python
vendors = list(top_vendors['VendorName'].values)
purchase_contribution = list(top_vendors['purchaceContribution%'].values)
total_contribution = sum(purchase_contribution)
remaining_contribution = 1 - total_contribution

# Append 'other Vendors' category
vendors.append('Other Vendors')
purchase_contribution.append(remaining_contribution)

# Donut Chart
fig, ax = plt.subplots(figsize=(8,8))
wedges, texts, autotexts = ax.pie(purchase_contribution, labels=vendors, autopct
                                  startangle=90, pctdistance = 0.85, colors=plt.

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

# add total contribution annotation in the center
plt.text(0,0, f"Top 10 Total:\n{total_contribution:.2f}%", fontsize=14, fontweig

plt.title("Top 10 Vendors Purchase Contribution (%)")
plt.show()
```



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

In [209…

```
df['UnitPurchasePrice'] = df['TotalPurchaseDollars']/df['TotalPurchaseQuantity']
```

In [210…

```
df
```

Out[210…

	VendorNumber	VendorName	Brand	ActualPrice	PurchasePrice	Volume	TotalS
0	2	IRA GOLDMAN AND WILLIAMS, LLP	90085	36.99	23.86	750.0	
1	105	ALTAMAR BRANDS LLC	8412	49.99	35.71	750.0	
2	105	ALTAMAR BRANDS LLC	8419	6.99	5.30	100.0	
3	287	APPOLO VINEYARDS LLC	24921	15.49	10.40	750.0	
4	287	APPOLO VINEYARDS LLC	24922	15.49	10.47	750.0	
...	...	...	...	...	...	...	
7651	98450	Serralles Usa LLC	6877	18.99	13.56	750.0	
7652	98450	Serralles Usa LLC	7890	29.99	22.21	750.0	
7653	98450	Serralles Usa LLC	8543	19.99	14.81	750.0	
7654	172662	SWEETWATER FARM	4215	25.99	19.40	750.0	
7655	173357	TAMWORTH DISTILLING	3909	24.99	19.37	750.0	

7656 rows × 18 columns

In [211…

```
df['OrderSize'] = pd.qcut(df['TotalPurchaseQuantity'], q = 3, labels=['Small','M
```

In [214…

```
df['OrderSize'].value_counts()
```

Out[214...

	count
OrderSize	
Small	2555
Large	2552
Medium	2549

dtype: int64

In [216...

```
df[['TotalPurchaseQuantity','OrderSize']].sort_values('TotalPurchaseQuantity', a
```

Out[216...

	TotalPurchaseQuantity	OrderSize
2143	11339979676	Large
2265	11269002432	Large
2026	11006932608	Large
2198	10750322572	Large
2178	10144286104	Large
...	...	...
4449	151	Small
7543	144	Small
556	82	Small
7525	51	Small
915	20	Small

7656 rows × 2 columns

In [218...

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

Out[218...

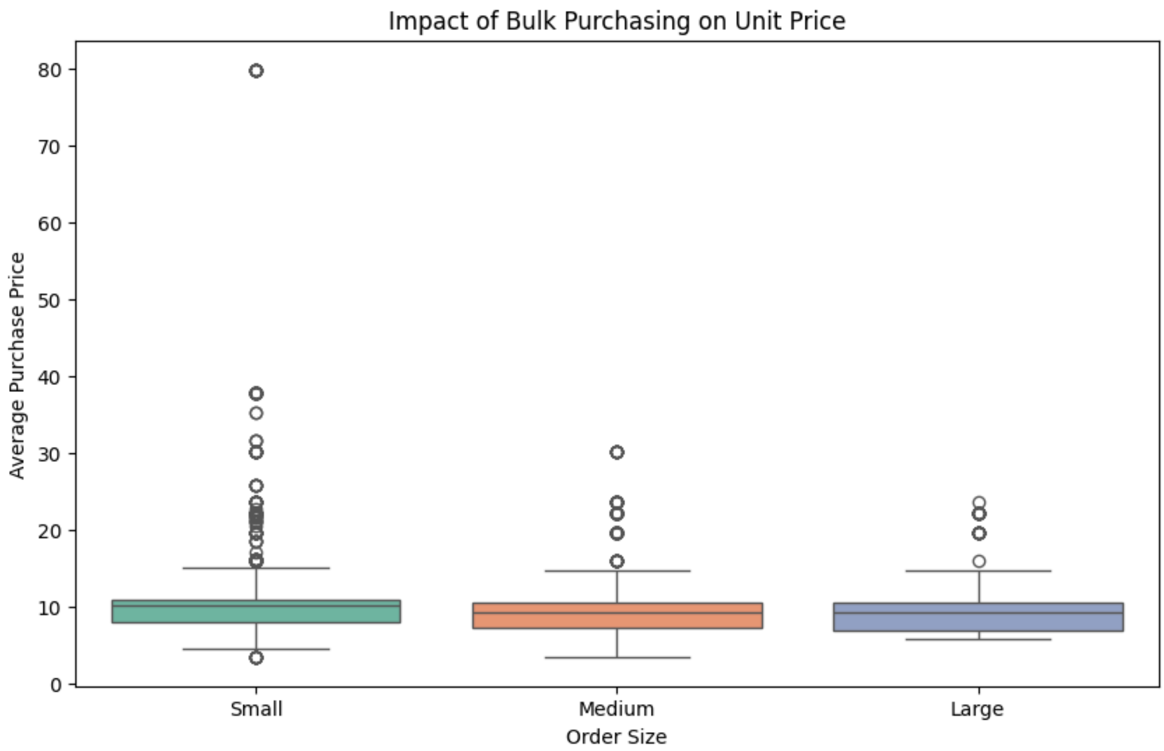
	UnitPurchasePrice
OrderSize	
Small	10.896777
Medium	9.897589
Large	9.441692

dtype: float64

In [219...

```
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 Purchase Price')
plt.show()
```



- Vendors buying in bulk (Large Order Size) get the lowest unit price ($9.44 per unit), meaning higher margins if they can manage inventory efficiently.
- The price difference between Small and Large orders is substantial (-10% reduction in unit cost)
- This suggests that bulk pricing strategies successfully encourage vendors to purchase in larger volumes, leading to higher overall sales despite lower per-unit revenue.

Which Vendor has low inventory turnover, indicating exces stock and slow-moving products?

In [227…

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

Out[227...

VendorName	StockTurnover
DIAGEO NORTH AMERICA INC	0.002687
JIM BEAM BRANDS COMPANY	0.003865
CONSTELLATION BRANDS INC	0.006037
BACARDI USA INC	0.006725
PERNOD RICARD USA	0.007333
MARTIGNETTI COMPANIES	0.007467
M S WALKER INC	0.009995
ULTRA BEVERAGE COMPANY LLP	0.010587
E & J GALLO WINERY	0.010848
BROWN-FORMAN CORP	0.013251

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

In [231...

```
df['UnsoldInventoryValue'] = (df['TotalPurchaseQuantity']- df['TotalSalesQuantit
print(f"Total Unsold Inventory Value: ${format_dollars(df['UnsoldInventoryValue'
```

```
Total Unsold Inventory Value: $178156005.55M
```

In [233...

```
# Aggregate capital Locked per Vendor
inventory_value_per_Vendor = df.groupby('VendorName')['UnsoldInventoryValue'].su

# Sort Vendors with the highest Locked Capital
inventory_value_per_Vendor = inventory_value_per_Vendor.sort_values(by='UnsoldIn
inventory_value_per_Vendor['UnsoldInventoryValue'] = inventory_value_per_Vendor[
inventory_value_per_Vendor.head(10)
```

Out[233...

	VendorName	UnsoldInventoryValue
24	DIAGEO NORTH AMERICA INC	64140410.63M
43	JIM BEAM BRANDS COMPANY	22763748.16M
54	MARTIGNETTI COMPANIES	21347439.45M
65	PERNOD RICARD USA	11611492.34M
19	CONSTELLATION BRANDS INC	11024467.78M
5	BACARDI USA INC	8038846.51M
29	E & J GALLO WINERY	8017219.99M
102	ULTRA BEVERAGE COMPANY LLP	5409565.52M
50	M S WALKER INC	5345928.83M
10	BROWN-FORMAN CORP	4216706.23M

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

In [234...

```
top_threshold = df['TotalSalesDollars'].quantile(0.95)
low_threshold = df['TotalSalesDollars'].quantile(0.05)
```

In [236...

```
top_vendors = df[df['TotalSalesDollars']>top_threshold]['ProfitMargin'].dropna()
low_vendors = df[df['TotalSalesDollars']<low_threshold]['ProfitMargin'].dropna()
```

In [237...

```
top_vendors
```

Out[237...

	ProfitMargin
8	-3.625542e+05
13	-5.801058e+05
40	-3.514382e+05
41	-9.524987e+05
56	-8.034870e+05
...	...
7255	-7.730084e+05
7256	-1.307348e+06
7280	-3.927685e+05
7284	-2.845648e+05
7291	-7.133934e+05

383 rows × 1 columns

dtype: float64

In [238...

```
low_vendors
```

Out[238...

	ProfitMargin
0	-1.802839e+03
3	-1.091687e+03
4	-1.091687e+03
18	-2.466746e+06
108	-1.603720e+06
...	...
7611	-1.999867e+04
7615	-3.722405e+04
7623	-2.892521e+04
7624	-3.156546e+04
7625	-4.344991e+04

383 rows × 1 columns

dtype: float64

In [241...

```
from scipy.stats import ttest_ind
import scipy.stats as stats
```

In [242...

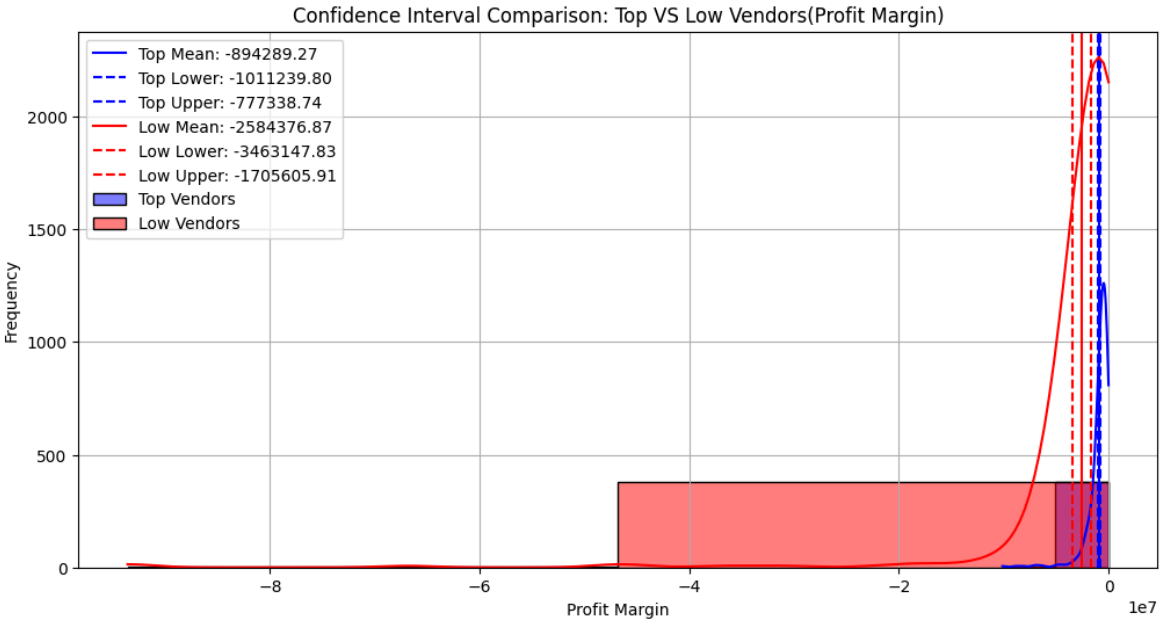
```
def calculate_confidence_interval(data, confidence=0.95):
  mean_val = np.mean(data)
  std_err = np.std(data,ddof=1)/np.sqrt(len(data)) # std error
  t_critical = stats.t.ppf((1+confidence)/2, df=len(data)-1)
  margin_of_error = t_critical*std_err
  lower_bound = mean_val - margin_of_error
  upper_bound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

In [252...

```
top_mean, top_lower, top_upper = calculate_confidence_interval(top_vendors)
low_mean, low_lower, low_upper = calculate_confidence_interval(low_vendors)

print(f"Top Vendor 95% CI:({top_lower:.2f}, {top_upper:.2f}), Mean: {top_mean:.2
print(f"Low Vendor 95% CI:({low_lower:.2f}, {low_upper:.2f}), Mean: {low_mean:.2

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

# Top Vendor Plot
sns.histplot(top_vendors, kde=True, bins=2, color='blue', alpha = 0.5, label='To
plt.axvline(top_mean, color='blue', linestyle='-', label=f"Top Mean: {top_mean:.
plt.axvline(top_lower, color='blue', linestyle='--', label=f"Top Lower: {top_low
plt.axvline(top_upper, color='blue', linestyle='--', label=f"Top Upper: {top_upp

# Low Vendor Plot
sns.histplot(low_vendors, kde=True, bins=2, color='red', alpha = 0.5, label='Low
plt.axvline(low_mean, color='red', linestyle='-', label=f"Low Mean: {low_mean:.2
plt.axvline(low_lower, color='red', linestyle='--', label=f"Low Lower: {low_lowe
plt.axvline(low_upper, color='red', linestyle='--', label=f"Low Upper: {low_uppe
```

```python
# Finalise the 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 Vendor 95% CI:(-1011239.80, -777338.74), Mean: -894289.27
Low Vendor 95% CI:(-3463147.83, -1705605.91), Mean: -2584376.87
```




```
In [ ]:
```