

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

Intel MKL WARNING: Support of Intel(R) Streaming SIMD Extensions 4.2 (Intel(R) SSE4.2) enabled only processors has been deprecated. Intel oneAPI Math Kernel Library 2025.0 will require Intel(R) Advanced Vector Extensions (Intel(R) AVX) instructions.

Intel MKL WARNING: Support of Intel(R) Streaming SIMD Extensions 4.2 (Intel(R) SSE4.2) enabled only processors has been deprecated. Intel oneAPI Math Kernel Library 2025.0 will require Intel(R) Advanced Vector Extensions (Intel(R) AVX) instructions.

```
In [2]: df = pd.read_excel('/Users/mohamad/Desktop/Projects/wayaak purchasing.xlsx')
df = df.drop(df.columns[0], axis=1)
df = df[df['Product name'] != 'debit note']
df = df[df['Product name'] != 'credit note']
# df = df[df['Quantity'] > 0.0001]
# df2 = pd.read_excel('/Users/mohamad/Desktop/Projects/wayaak purchasing.xlsx')

pd.set_option('display.float_format', '{:,.1f}'.format)
```

```
In [3]: df['Total'].sum()
```

```
Out[3]: 266512442.2873409
```

```
In [4]: df[df['Quantity'] < 0.0001].sort_values(by='new discount', ascending=False)
```

Out[4] :

	Date	Bill No	Store ID	Type	Store Name	Product id	Produc nar
32456	04-03-2024	2024102278413	1	Purchase	وياك المعادي	23100	LEPTICU 400 MG CAF
94968	22-07-2024	8771117595	1	Purchase	وياك المعادي	19760	SEROP HAIR SERU 100 M
94957	22-07-2024	107822037	1	Purchase	وياك المعادي	22468	LIPOSOM I 10.000 I. 30 CAF
94958	22-07-2024	2024107982506	1	Purchase	وياك المعادي	21853	SULF PLI MASSA GEL 60 G
144461	20-10-2024	8777163860	1	Return	وياك المعادي	21780	GENUP ORIGINAL F.C. TA

110418	21-08-2024	Nan	1	Return	وياك المعادي	23170	GLEPTOM 50/1000N 30FIL COATI TABLE
99537	30-07-2024	808993	1	Return	وياك المعادي	8642	ISOPT MAXITR EYE DROPS
41688	27-03-2024	70940	1	Return	وياك المعادي	7242	GAST-R 100MG TA
110419	21-08-2024	Nan	1	Return	وياك المعادي	23318	METFORMI EVA XR 10 MG 30 TAE
110420	21-08-2024	Nan	1	Return	وياك المعادي	21788	MUCOBRA 600 MG SACHE

6993 rows x 28 columns

In [5]: df[df['Quantity'].between(0, 0.05)].head(1)

Out[5]:

	Date	Bill No	Store ID	Type	Store Name	Product id	Product name
32456	04-03-2024	2024102278413		1 Purchase	وياك المعادي	23100	LEPTICURE 400 MG 30 CAPS.

1 rows × 28 columns

```
In [6]: df['Product name'] = df['Product name'].str.strip()
df['Product name'] = df['Product name'].str.replace(r'\s+', ' ', re
```

```
In [7]: # df = df.merge(df2, how='left', on='Product name')
# df
# List of suppliers
preferred_suppliers = ['شركة سوفيكو فارم', 'شركة فارما اوفر سيز']

# Apply conditions using np.where
df['new discount adds'] = np.where(
    (df['Supplier'].isin(preferred_suppliers)) & (df['Category'] == df['new discount']) + (((df['total after discount tax'] * 0.08)
    np.where(
        (df['Supplier'].isin(preferred_suppliers)) & (df['Category'] == df['new discount']) + (((df['total after discount tax'] * 0.
        df['new discount']) # fallback to original discount if no c
    )
)

df['total after discount tax 2'] = df['total public'] - (df['total p
df['total discount 2'] = df['total public'] * df['new discount adds']
```

```
In [8]: df.sort_values(by='new discount adds', ascending=False).head(10)
```

Out[8] :

	Date	Bill No	Store ID	Type	Store Name	Product id	Product r
116355	02-09-2024	0000002293708	5	Purchase	المخزن المركزي	21463	c zinc 30
103659	07-08-2024	2024107982504	4	Purchase	وياك صيدلية الهرم	23222	SANS 10.00 28T
103690	07-08-2024	8771970070	4	Purchase	وياك صيدلية الهرم	1382	DELTAVID 1M SUBLIN
103716	07-08-2024	8772409565	4	Purchase	وياك صيدلية الهرم	19599	SULFAX (FORM MASS CREAM 60
102408	05-08-2024	2024108202883	4	Purchase	وياك صيدلية الهرم	19599	SULFAX (FORM MASS CREAM 60
103726	07-08-2024	8772371958	4	Purchase	وياك صيدلية الهرم	21726	RETARD+ 20 C
55172	29-04-2024	801667	1	Purchase	وياك المعادى	3675	CITAPRONEX 10MG F.C
36322	11-03-2024	2024102585216	1	Purchase	وياك المعادى	9675	SILDEN 50 4 F.C. T
36533	11-03-2024	2024102585217	1	Purchase	وياك المعادى	22378	PARACETAMOL CID 20
2993	04-01-2024	2023109040580	4	Return	وياك صيدلية الهرم	21955	JUSTEC 20MG 28 T

10 rows × 31 columns

In [9]: df[df['Product name'] == 'CITAPRONEX 10MG 14 F.C.TAB.'].sort_values

Out[9]:

	Date	Bill No	Store ID	Type	Store Name	Product id	Product name
55172	29-04-2024	801667	1	Purchase	وياك المعادي	3675	CITAPRONEX 10MG 14 F.C.TAB.
7048	11-01-2024	8762292785	4	Return	وياك صيدلية الهرم	3675	CITAPRONEX 10MG 14 F.C.TAB.
58878	06-05-2024	29597	1	Purchase	وياك المعادي	3675	CITAPRONEX 10MG 14 F.C.TAB.
126775	22-09-2024	8776369075	1	Purchase	وياك المعادي	3675	CITAPRONEX 10MG 14 F.C.TAB.
64334	19-05-2024	8768654404	1	Purchase	وياك المعادي	3675	CITAPRONEX 10MG 14 F.C.TAB.
66617	24-05-2024	8768947571	1	Purchase	وياك المعادي	3675	CITAPRONEX 10MG 14 F.C.TAB.
70502	31-05-2024	8769340575	1	Purchase	وياك المعادي	3675	CITAPRONEX 10MG 14 F.C.TAB.
86074	03-07-2024	8771164409	4	Purchase	وياك صيدلية الهرم	3675	CITAPRONEX 10MG 14 F.C.TAB.
87331	05-07-2024	8771297956	4	Purchase	وياك صيدلية الهرم	3675	CITAPRONEX 10MG 14 F.C.TAB.
89149	07-07-2024	8771374221	1	Purchase	وياك المعادي	3675	CITAPRONEX 10MG 14 F.C.TAB.

10 rows × 31 columns

```
In [10]: name_replacements = {
    'شركة ميديكال فارما لتجارة توزيع الادوية': 'شركة جلوبال فارما'
}

# Apply the replacement
df['Supplier'] = df['Supplier'].replace(name_replacements)
```

```
In [11]: distribution_suppliers = 'شركة سوفيكيو فارم', 'شركة فارما اوفر سيز',  
Bulk_suppliers = 'WeCare', 'شركة الاهرام شبين الكوم', 'الفاروق',  
'فارم', 'هيثم محمد', 'AKHNATON COMPANY', 'الشمس فارم',  
'فارم السنبلاوين', 'سينا فارما تريد لتجارة والتوزيع'
```

```
[جاسمين فارما', 'مخزن فارما بيور نيو',
'i', 'ميديكال فارما _ اونر نقمي', 'متنوعون الهرم',
ي فارما للتجارة والتوزيع', 'مخزن الأخوة المتخدون',
ة مالتي ستورز فارما', 'رامكو فارم لتجارة الادوية',
ركة شفاء', 'الاصدقاء فارم', 'شركة ستوك فارما جروب',
', 'مخزن كيان فارم لتجارة وتوزيع الادوية',
'EBDA PHARMA', 'KIAN FARMS FOR TRADE & DISTRIBUTION LTD.',

df['Vendor Type'] = np.where(
    df['Supplier'].isin(distribution_suppliers),
    'Distribution',
    np.where(
        df['Supplier'].isin(Bulk_suppliers),
        'Bulk',
        np.where(
            df['Supplier'].isin(Retail_suppliers),
            'Retail',
            'NA' # fallback to original discount if no condition matches
        )
    )
)
# fallback to original discount if no condition matches
```

In [12]: df.head(10)

Out[12]:

	Date	Bill No	Store ID	Type	Store Name	Product id	Product name	Su
0	01-01-2024	2023108920907	4	Purchase	وياك صيدلية الهرم	3083	EXFORGE 5MG/160MG 14 F.C. TAB.	١
1	01-01-2024	2023108920907	4	Purchase	وياك صيدلية الهرم	10393	GLAPTIVIA PLUS 50/1000MG 30 F.C.TAB.	١
2	01-01-2024	2023108920907	4	Purchase	وياك صيدلية الهرم	3080	EXFORGE 10MG/160MG 14 F.C. TAB.	١
3	01-01-2024	10253	4	Purchase	وياك صيدلية الهرم	11444	ERALONER 25 MG 20 F.C.TAB.	٦
4	01-01-2024	10253	4	Purchase	وياك صيدلية الهرم	8683	PANTOLOC 20MG 14 E.C. TAB.	٦
5	01-01-2024	10253	4	Purchase	وياك صيدلية الهرم	5177	CARDURA 4 MG 14 TAB.	٦
6	01-01-2024	10253	4	Purchase	وياك صيدلية الهرم	5756	AMARYL 4 MG 30 TAB.	٦
7	01-01-2024	10253	4	Purchase	وياك صيدلية الهرم	18453	DIAMICRON 60 MG 30 M.R. SCORED TAB.	٦
8	01-01-2024	10253	4	Purchase	وياك صيدلية الهرم	5312	ATOR 10MG 7 TAB.	٦
9	01-01-2024	10253	4	Purchase	وياك صيدلية الهرم	3044	ATOR 40MG 10 F.C. TAB.	٦

10 rows × 32 columns

In [13]:

```
df['Date'] = pd.to_datetime(df['Date'], format='%d-%m-%Y')
df['month'] = df['Date'].dt.month
```

In [14]:

```
df_purchased = df.copy()
```

In [15]:

```
# max_discount = df_purchased[df_purchased['new discount'] < 100].g
```

```
# discount_map = max_discount.set_index('Product name')['new discount']
# # discount_map

# df_purchased['new discount pot'] = df_purchased['Product name'].map(discount_map)

# df_purchased['Potential Diff %'] = (df_purchased['new discount pot'] - df_purchased['original price']) / df_purchased['original price']

# df_purchased['potential total discount'] = df_purchased['new discount pot'] * df_purchased['original price']

# # # Value of missed opportunity per transaction
# df_purchased['Potential Value'] = df_purchased['Potential Diff %'] * df_purchased['original price']
```

In [16]:

```
# Step 1: Remove outliers in discount per product
def remove_discount_outliers(group):
    q1 = group['new discount adds'].quantile(0.25)
    q3 = group['new discount adds'].quantile(0.75)
    iqr = q3 - q1
    lower = q1 - 1.5 * iqr
    upper = q3 + 1.5 * iqr
    return group[(group['new discount adds'] >= lower) & (group['new discount adds'] <= upper)]

# Step 2: Apply to each product
df_cleaned = df_purchased.groupby('Product name', group_keys=False).apply(remove_discount_outliers)

# Step 3: Compute max discount per product (without outliers)
max_discount = df_cleaned.groupby('Product name')['new discount adds'].max()

# Step 4: Create a mapping and apply
discount_map = max_discount.set_index('Product name')['new discount adds']
df_purchased['new discount pot'] = df_purchased['Product name'].map(discount_map)

# Step 5: Calculate opportunity columns
df_purchased['Potential Diff %'] = df_purchased['new discount pot'] - df_purchased['original price']
df_purchased['potential total discount'] = df_purchased['new discount pot'] * df_purchased['original price']
df_purchased['potential value'] = df_purchased['Potential Diff %'] * df_purchased['original price']
```

```
/var/folders/vf/sz32t6rx7635wcdmj4ydx580000gn/T/ipykernel_50536/2097340451.py:11: DeprecationWarning: DataFrameGroupBy.apply operated on the grouping columns. This behavior is deprecated, and in a future version of pandas the grouping columns will be excluded from the operation. Either pass `include_groups=False` to exclude the groupings or explicitly select the grouping columns after groupby to silence this warning.
    df_cleaned = df_purchased.groupby('Product name', group_keys=False).apply(remove_discount_outliers)
```

In [17]:

```
# potential2 = supplier_product_df.merge(max_discount, on='Product name')
# potential2['Potential Diff'] = (potential2['new discount adds'] - potential2['original price'])
# potential2['potential value'] = potential2['new discount adds'] * potential2['original price']
print(f"Total: {df_purchased['Total'].sum():,.2f}")
print(f"Total Actual Savings: {df_purchased['Total'].sum() - df_purchased['original price'].sum():,.2f}")
print(f"Total Potential Savings value: {df_purchased['potential total discount'].sum():,.2f}")
print(f"Potential Difference: {df_purchased['potential value'].sum():,.2f}
```

Total: 266,512,442.29
 Total Actual Savings: 64,087,776.23
 Total Potential Savings value: 81,163,940.87
 Potential Difference: 17,076,269.63

In [18]: df_cleaned.head(10)

Out[18]:

	Date	Bill No	Store ID	Type	Store Name	Product id	Product name	€
22362	2024-02-11	1535978	1	Purchase	وياك المعادي	23083	0	
22793	2024-02-12	1540553	1	Purchase	وياك المعادي	23083	0	
25906	2024-02-20	2024101783259	1	Purchase	وياك المعادي	23083	0	
27938	2024-02-26	2024102009718	1	Purchase	وياك المعادي	23083	0	
30448	2024-03-01	1605149	1	Purchase	وياك المعادي	23083	0	
30560	2024-03-01	70736	1	Purchase	وياك المعادي	23083	0	
30933	2024-03-02	70743	1	Purchase	وياك المعادي	23083	0	
37320	2024-03-13	70865	1	Purchase	وياك المعادي	23083	0	
8219	2024-01-13	70442	1	Purchase	وياك المعادي	441	A-VITON 50.000 I.U. 20 CAPS.	
9694	2024-01-15	70460	1	Purchase	وياك المعادي	441	A-VITON 50.000 I.U. 20 CAPS.	

10 rows × 33 columns

In [19]: max_discount.sort_values(by = 'new discount adds', ascending = False)

Out[19]:

	Product name	new discount adds
2582	LEPTICURE 400 MG 30 CAPS.	108.0
4277	TADARAXIA 20MG 2 FLIM COATED TABLETS	108.0
3679	Pregnancy Check-Up Package	100.0
2313	INFINITY ANTI-AGING SPF 30 DAY SERUM 40 ML	100.0
4278	TADATRONA 20 MG 2 TABS.	100.0
4140	STERILE WATER FOR INJ. 5 ML	98.0
4673	VIGOTON 30 TABS	76.0
4580	V-GONE 100MG 8TAB	75.0
1641	EREC 100 MG 12 F.C. TABS.	73.4
3080	NAPIFIT 100MG 8 TAB .	72.0
4148	STIMU-MAX 100MG 5 F.C.TAB.	72.0
3989	SILDAVA 100MG 12 F.C. TAB.	72.0
4597	VARDENA 20 MG 4 F.C. TABS	71.0
4690	VIRECTA 100MG 12F.C. TAB.	71.0
1763	FAROVIGA 100MG 12 F.C.TAB.	70.0
2891	MESTAL MASSAGE GEL 60 GM	70.0
4257	SYRINGE ADULT 3CM	68.7
4258	SYRINGE ADULT 5 CM	68.5
4667	VIAVAG 100MG 10 TAB	68.0
3882	SANSO C PLUS 14 CAPS	68.0

In [20]: df[df['Product name'] == 'c zinc 30 cap'].sort_values(by = 'new dis

Out[20]:

	Date	Bill No	Store ID	Type	Store Name	Product id	Product name
116355	2024-09-02	0000002293708	5	Purchase	المخزن المركزي	21463	c zinc 30 cap

1 rows × 33 columns

In [21]: df_purchased['day'] = df_purchased['Date'].dt.day
day_df = df_purchased.groupby('day')['total after discount tax 2'].

In [22]: df.info()

```

<class 'pandas.core.frame.DataFrame'>
Index: 195663 entries, 0 to 198268
Data columns (total 33 columns):
 #   Column           Non-Null Count   Dtype  
--- 
 0   Date             195663 non-null    datetime64[ns]
 1   Bill No          195459 non-null    object  
 2   Store ID         195663 non-null    int64   
 3   Type             195663 non-null    object  
 4   Store Name       195663 non-null    object  
 5   Product id       195663 non-null    int64   
 6   Product name     195662 non-null    object  
 7   Supplier          195663 non-null    object  
 8   Category          195663 non-null    object  
 9   Unit Price        195663 non-null    float64 
 10  Discount %       195663 non-null    float64 
 11  Net Unit Price   195663 non-null    float64 
 12  Quantity          195663 non-null    float64 
 13  Total             195663 non-null    float64 
 14  Bonus QTY         195663 non-null    float64 
 15  VAT               195663 non-null    float64 
 16  Discount          195663 non-null    float64 
 17  Net Cos            195663 non-null    float64 
 18  price after discount include tax 195663 non-null    float64 
 19  price after discount exclude tax 195663 non-null    float64 
 20  tax                195663 non-null    float64 
 21  total public       195663 non-null    float64 
 22  total public without tax 195663 non-null    float64 
 23  total tax          195663 non-null    float64 
 24  total after discount tax 195663 non-null    float64 
 25  total discount      195663 non-null    float64 
 26  actual unit price   195663 non-null    float64 
 27  new discount        195663 non-null    float64 
 28  new discount adds    195661 non-null    float64 
 29  total after discount tax 2 195661 non-null    float64 
 30  total discount 2     195661 non-null    float64 
 31  Vendor Type        195663 non-null    object  
 32  month              195663 non-null    int32  
dtypes: datetime64[ns](1), float64(22), int32(1), int64(2), object(7)
memory usage: 50.0+ MB

```

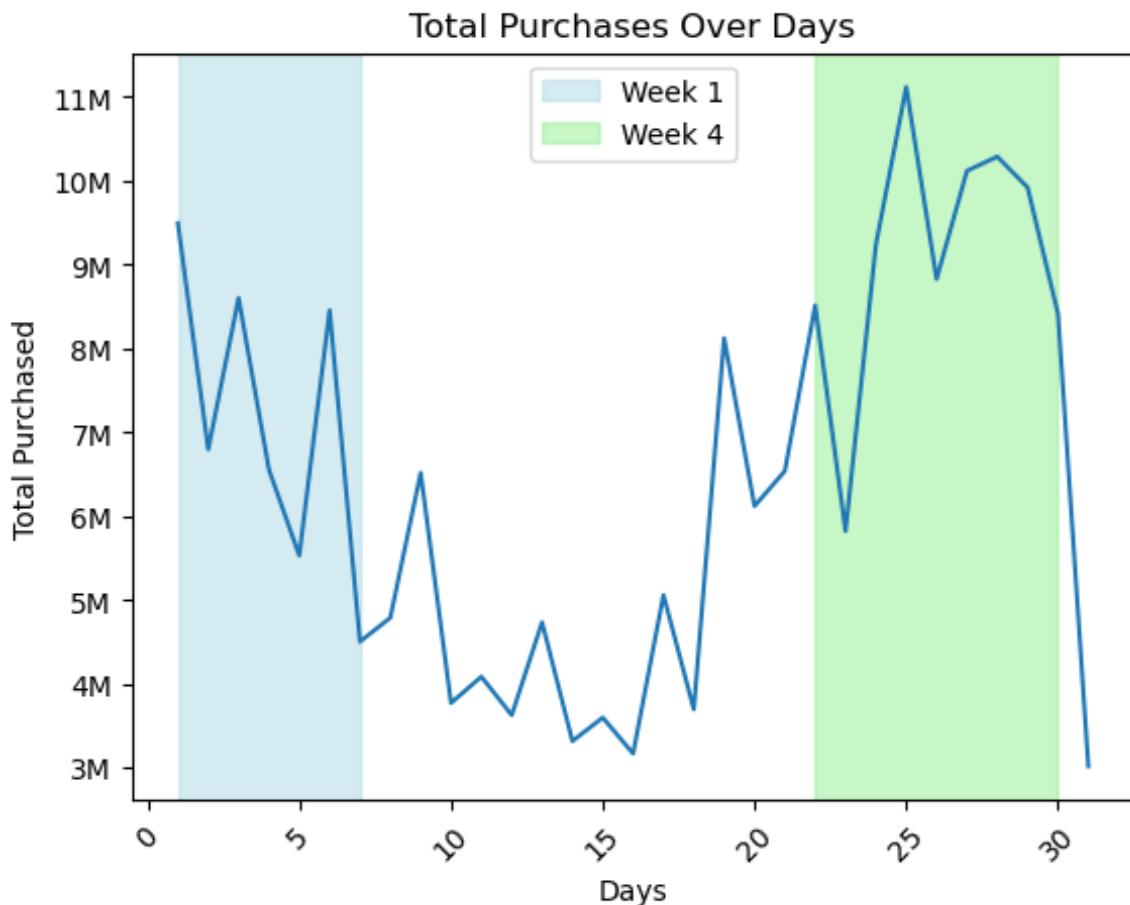
Time

```
In [23]: sns.lineplot(data=day_df, x = 'day', y = 'total purchased')
plt.axvspan(1, 7, color='lightblue', alpha=0.5, label='Week 1')
plt.axvspan(22, 30, color='lightgreen', alpha=0.5, label='Week 4')
# plt.text(4, 50, 'Week 1', horizontalalignment='center', fontsize=14)
# plt.text(25.5, 50, 'Week 4', horizontalalignment='center', fontsize=14)

plt.gca().yaxis.set_major_formatter(mticker.FuncFormatter(lambda x,
```

```
plt.title('Total Purchases Over Days')
plt.xlabel('Days')
plt.ylabel('Total Purchased')
plt.xticks(rotation=45)
plt.legend()
```

Out[23]: <matplotlib.legend.Legend at 0x13d7eb290>



Week 1 & week 4 are the active ones

In [24]: `fig, ax = plt.subplots(3,1, figsize = (10,10))`

```
df_bill_month = df_purchased[df_purchased['month'].isin([1,2,3,4,5,
ax[1].plot(df_bill_month['month'], df_bill_month['Bill Count'], color='red')
ax[1].set_title('Bills Count Per Month')
ax[1].set_xlabel('')
ax[1].set_xlim(300,2000)
# ax[1].grid()
sns.despine(ax = ax[1])
ax[1].axhline(y = df_bill_month['Bill Count'].mean(), color='green')
ax[1].text(x=len(df_bill_month['month']) + 1, y=df_bill_month['Bill Count'].mean(), color='red', verticalalignment='bottom', horizontalalignment='right')

max_value = df_bill_month['Bill Count'].max()
min_value = df_bill_month['Bill Count'].min()
max_month = df_bill_month['month'][df_bill_month['Bill Count'].idxmax()]
min_month = df_bill_month['month'][df_bill_month['Bill Count'].idxmin()]
```

```
# Mark maximum point
ax[1].plot(max_month, max_value, 'bo', markersize=5) # Red point for max
ax[1].text(max_month, max_value + 100, f'Max: {max_value}', color='red')
# Mark minimum point
# ax[1].plot(min_month, min_value, 'ro', markersize=5) # Blue point for min
# ax[1].text(min_month, min_value - 200, f'Min: {min_value}', color='blue')
ax[1].axvspan(9, 12, color='lightblue', alpha=0.5, label='Q4')

for i, row in df_bill_month.iterrows():
    if row['Bill Count'] < df_bill_month['Bill Count'].mean():
        ax[1].plot(row['month'], row['Bill Count'], 'o', color='red')

df_supplier_month = df_purchased[df_purchased['month'].isin([1,2,3,4,5,6,7,8,9,10,11,12])]

ax[2].plot(df_supplier_month['month'], df_supplier_month['Supplier Count'])
sns.despine(ax = ax[2])
ax[2].set_title('Supplier Count Per Month')
ax[2].set_xlabel('')
ax[2].set_xlim(20,45)
ax[2].set_ylim(20,45)
sns.despine(ax = ax[2])
ax[2].axhline(y = df_supplier_month['Supplier Count'].mean(), color='black')
ax[2].text(x=len(df_bill_month['month']) + 1, y=df_supplier_month['Supplier Count'].mean() + 5, s=f'AVG: {int(df_supplier_month['Supplier Count'].mean())}', verticalalignment='bottom', horizontalalignment='right')
max_value_2 = df_supplier_month['Supplier Count'].max()
min_value_2 = df_supplier_month['Supplier Count'].min()
max_month_2 = df_supplier_month['month'][df_supplier_month['Supplier Count'] == max_value_2]
min_month_2 = df_supplier_month['month'][df_supplier_month['Supplier Count'] == min_value_2]

# Mark maximum point
ax[2].plot(max_month_2, max_value_2, 'bo', markersize=5) # Red point for max
ax[2].text(max_month_2 + 0.2, max_value_2 + 2, f'Max: {max_value_2}', color='red')
# Mark minimum point
ax[2].plot(min_month_2, min_value_2, 'ro', markersize=5) # Blue point for min
ax[2].text(min_month_2, min_value_2 - 5, f'Min: {min_value_2}', color='blue')

for i, row in df_supplier_month.iterrows():
    if row['Supplier Count'] < df_supplier_month['Supplier Count'].mean():
        ax[2].plot(row['month'], row['Supplier Count'], 'o', color='red')

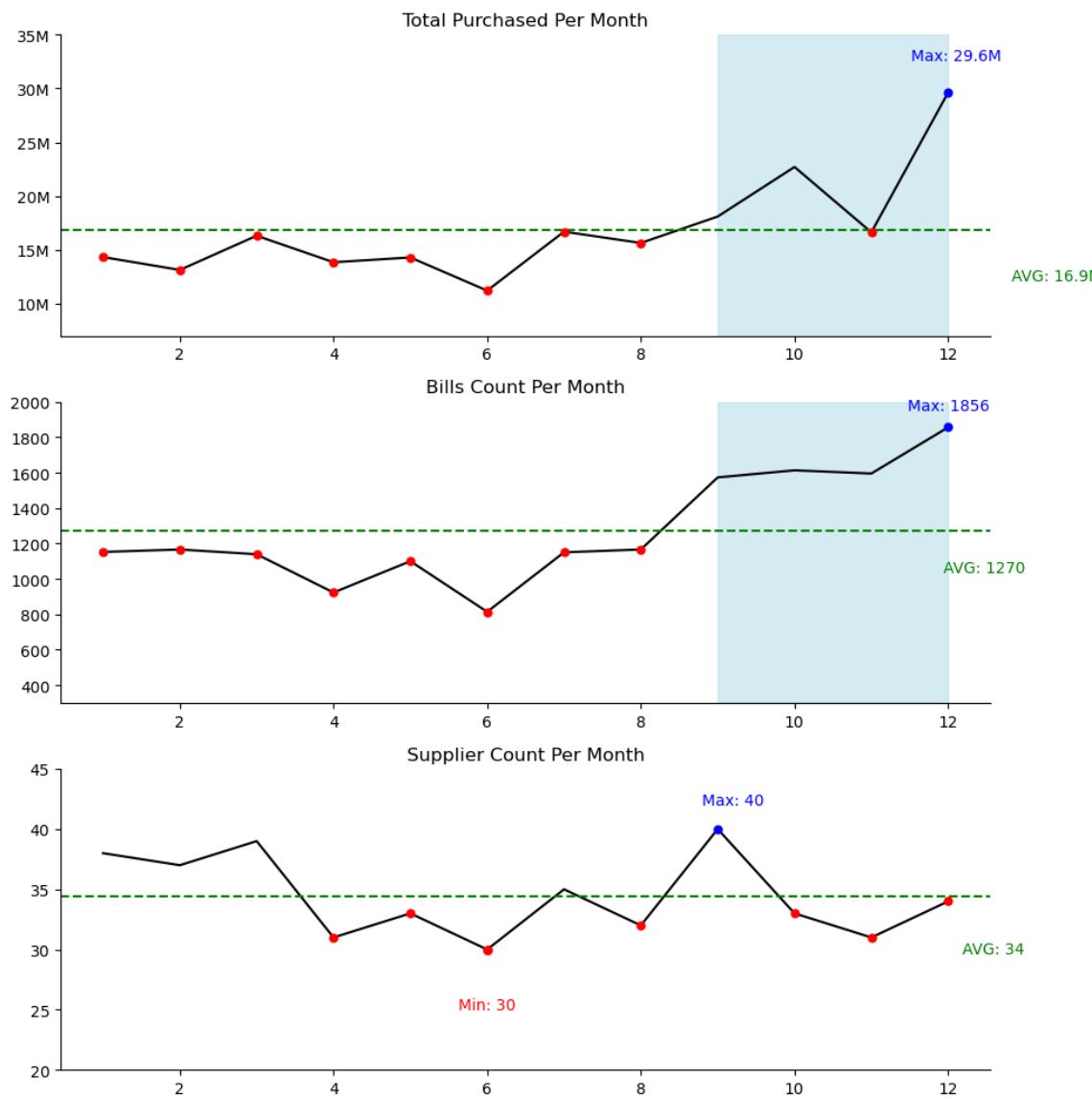
df_purchasing_month = df_purchased[df_purchased['month'].isin([1,2,3,4,5,6,7,8,9,10,11,12])]

ax[0].plot(df_purchasing_month['month'], df_purchasing_month['Total Purchased'])
sns.despine(ax = ax[0])
ax[0].set_title('Total Purchased Per Month')
ax[0].set_xlabel('')
sns.despine(ax = ax[0])
ax[0].yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:.0f}'))
ax[0].axhline(y = df_purchasing_month['Total Purchased'].mean(), color='black')
ax[0].text(x=len(df_bill_month['month']) + 2, y=df_purchasing_month['Total Purchased'].mean() + 5, s=f'AVG: {int(df_purchasing_month["Total Purchased"].mean())}', verticalalignment='bottom', horizontalalignment='right')
max_value_0 = df_purchasing_month['Total Purchased'].max()
min_value_0 = df_purchasing_month['Total Purchased'].min()
max_month_0 = df_purchasing_month['month'][df_purchasing_month['Total Purchased'] == max_value_0]
min_month_0 = df_purchasing_month['month'][df_purchasing_month['Total Purchased'] == min_value_0]

# Mark maximum point
ax[0].plot(max_month_0, max_value_0, 'bo', markersize=5) # Red point for max
ax[0].text(max_month_0 + 0.2, max_value_0 + 2, f'Max: {max_value_0}', color='red')
# Mark minimum point
ax[0].plot(min_month_0, min_value_0, 'ro', markersize=5) # Blue point for min
ax[0].text(min_month_0, min_value_0 - 5, f'Min: {min_value_0}', color='blue')

for i, row in df_purchasing_month.iterrows():
    if row['Total Purchased'] < df_purchasing_month['Total Purchased'].mean():
        ax[0].plot(row['month'], row['Total Purchased'], 'o', color='red')
```

```
s=f"AVG: {int(df_purchasing_month['Total Purchased'].mean())}M  
verticalalignment='bottom', horizontalalignment='right')  
ax[0].set_ylim(7000000,35000000)  
ax[0].axvspan(9, 12, color='lightblue', alpha=0.5, label='Q4')  
  
def millions(x, pos):  
    return f'{x * 1e-6:.0f}M'  
ax[0].yaxis.set_major_formatter(FuncFormatter(millions))  
  
max_value_0 = df_purchasing_month['Total Purchased'].max()  
min_value_0 = df_purchasing_month['Total Purchased'].min()  
max_month_0 = df_purchasing_month['month'][df_purchasing_month['Total Purchased'].idxmax()]  
min_month_0 = df_purchasing_month['month'][df_purchasing_month['Total Purchased'].idxmin()]  
  
# Mark maximum point  
ax[0].plot(max_month_0, max_value_0, 'bo', markersize=5) # Red point  
ax[0].text(max_month_0 + 0.1, max_value_0 + 3000000, f'Max: {int(max_value_0)}M')  
# Mark minimum point  
# ax[0].plot(min_month_0, min_value_0, 'ro', markersize=5) # Blue point  
# ax[0].text(min_month_0, min_value_0 - 1800000, f'Min: {int(min_value_0)}M')  
  
for i, row in df_purchasing_month.iterrows():  
    if row['Total Purchased'] < df_purchasing_month['Total Purchased'].mean():  
        ax[0].plot(row['month'], row['Total Purchased'], 'o', color='red')  
    else:  
        ax[0].plot(row['month'], row['Total Purchased'], 'o', color='green')  
  
plt.tight_layout()
```



```
In [25]: df_purchased['Supplier'].nunique()
```

```
Out[25]: 72
```

```
In [26]: # bills_month_df = df_purchased[df_purchased['month'].isin([1,2,3,4
#                                         'Bill No' : 'بills'
#                                         'average per ticket' ] = bills_month_df['total afte
#                                         'bills_df_haram = df_purchased[(df_purchased['Store Name'] == 'بهم
#                                         'Bill No' : 'بهم'
#                                         'average per ticket' ] = bills_df_haram['total afte
#
#                                         'bills_df_maadi = df_purchased[(df_purchased['Store Name'] == 'عادي
#                                         'Bill No' : 'عادي'
#                                         'average per ticket' ] = bills_df_maadi['total afte
#
#                                         'bills_df_central = df_purchased[(df_purchased['Store Name'] == 'المركز
#                                         'Bill No' : 'المركز'
#                                         'average per ticket' ] = bills_df_central['total afte
#
# fig, ax = plt.subplots(4,1, figsize = (6,6))
```

```
# ax[0].plot(bills_month_df['month'], bills_month_df['average per ticket'])
# ax[0].set_title('Monthly AVG Per Ticket')
# ax[0].set_xlabel('')
# ax[0].set_xlim(9000,20000)
# # ax[1].grid()
# sns.despine(ax = ax[0])
# ax[0].axhline(y = bills_month_df['average per ticket'].mean(), color='red')
# ax[0].text(x=len(bills_month_df['month']) +0.3 , y=bills_month_df['average per ticket'].mean() + 100, s=f'Average: {int(bills_month_df['average per ticket'].mean())}', verticalalignment='bottom', horizontalalignment='right')

# # max_value = bills_month_df['average per ticket'].max()
# # min_value = bills_month_df['average per ticket'].min()
# # max_month = bills_month_df['month'][bills_month_df['average per ticket'].idxmax()]
# # min_month = bills_month_df['month'][bills_month_df['average per ticket'].idxmin()]

# # # Mark maximum point
# # ax[0].plot(max_month, max_value, 'bo', markersize=5) # Red point
# # ax[0].text(max_month, max_value + 400, f'Max: {int(max_value)}')
# # # Mark minimum point
# # ax[0].plot(min_month, min_value, 'ro', markersize=5) # Blue point
# # ax[0].text(min_month, min_value - 1100, f'Min: {int(min_value)}')

# for i, row in bills_month_df.iterrows():
#     if row['average per ticket'] < bills_month_df['average per ticket'].mean() - 3000:
#         ax[0].plot(row['month'], row['average per ticket'], 'o', color='red')
#     else:
#         ax[0].plot(row['month'], row['average per ticket'], 'o', color='blue')

# ax[1].plot(bills_df_haram['month'], bills_df_haram['average per ticket'])
# ax[1].set_title('Haram AVG Per Ticket')
# ax[1].set_xlabel('')
# ax[1].set_xlim(10000,38000)
# # ax[1].grid()
# sns.despine(ax = ax[1])
# ax[1].axhline(y = bills_df_haram['average per ticket'].mean(), color='red')
# ax[1].text(x=len(bills_df_haram['month']) +0.3 , y=bills_df_haram['average per ticket'].mean() + 100, s=f'Average: {int(bills_df_haram['average per ticket'].mean())}', verticalalignment='bottom', horizontalalignment='right')

# # max_value_1 = bills_df_haram['average per ticket'].max()
# # min_value_1 = bills_df_haram['average per ticket'].min()
# # max_month_1 = bills_df_haram['month'][bills_df_haram['average per ticket'].idxmax()]
# # min_month_1 = bills_df_haram['month'][bills_df_haram['average per ticket'].idxmin()]

# # # Mark maximum point
# # ax[1].plot(max_month_1, max_value_1, 'bo', markersize=5) # Red point
# # ax[1].text(max_month_1, max_value_1 + 1500, f'Max: {int(max_value_1)}')
# # # Mark minimum point
# # ax[1].plot(min_month_1, min_value_1, 'ro', markersize=5) # Blue point
# # ax[1].text(min_month_1, min_value_1 - 3000, f'Min: {int(min_value_1)}')

# for i, row in bills_df_haram.iterrows():
#     if row['average per ticket'] < bills_df_haram['average per ticket'].mean() - 3000:
#         ax[1].plot(row['month'], row['average per ticket'], 'o', color='red')
#     else:
#         ax[1].plot(row['month'], row['average per ticket'], 'o', color='blue')
```

```
# ax[2].plot(bills_df_maadi['month'], bills_df_maadi['average per ticket'])
# ax[2].set_title('Maadi AVG Per Ticket')
# ax[2].set_xlabel('')
# ax[2].set_xlim(3000,10000)
# # ax[1].grid()
# sns.despine(ax = ax[2])
# ax[2].axhline(y = bills_df_maadi['average per ticket'].mean(), color='red')
# ax[2].text(x=len(bills_df_maadi['month']) + 0.3, y=bills_df_maadi['average per ticket'].mean() + 100, s=f'Average: {int(bills_df_maadi["average per ticket"].mean())}', verticalalignment='bottom', horizontalalignment='right')

# # max_value_2 = bills_df_maadi['average per ticket'].max()
# # min_value_2 = bills_df_maadi['average per ticket'].min()
# # max_month_2 = bills_df_maadi['month'][bills_df_maadi['average per ticket'].idxmax()]
# # min_month_2 = bills_df_maadi['month'][bills_df_maadi['average per ticket'].idxmin()]

# # # Mark maximum point
# # ax[2].plot(max_month_2, max_value_2, 'bo', markersize=5) # Red
# # ax[2].text(max_month_2, max_value_2 + 400, f'Max: {int(max_value_2)}')

# # # Mark minimum point
# # ax[2].plot(min_month_2, min_value_2, 'ro', markersize=5) # Blue
# # ax[2].text(min_month_2, min_value_2 - 1100, f'Min: {int(min_value_2)}')

# for i, row in bills_df_maadi.iterrows():
#     if row['average per ticket'] < bills_df_maadi['average per ticket'].mean() - 1000:
#         ax[2].plot(row['month'], row['average per ticket'], 'o', color='red')
#     else:
#         ax[2].plot(row['month'], row['average per ticket'], 'o', color='blue')

# ax[3].plot(bills_df_central['month'], bills_df_central['average per ticket'])
# ax[3].set_title('Central AVG Per Ticket')
# ax[3].set_xlabel('')
# ax[3].set_xlim(0,13)
# ax[3].set_ylim(3000,90000)
# # ax[1].grid()
# sns.despine(ax = ax[3])
# ax[3].axhline(y = bills_df_central['average per ticket'].mean(), color='red')
# ax[3].text(x=len(bills_df_central['month']) + 0.3, y=bills_df_central['average per ticket'].mean() + 100, s=f'Average: {int(bills_df_central["average per ticket"].mean())}', verticalalignment='bottom', horizontalalignment='right')

# # max_value_3 = bills_df_central['average per ticket'].max()
# # min_value_3 = bills_df_central['average per ticket'].min()
# # max_month_3 = bills_df_central['month'][bills_df_central['average per ticket'].idxmax()]
# # min_month_3 = bills_df_central['month'][bills_df_central['average per ticket'].idxmin()]

# # # Mark maximum point
# # ax[3].plot(max_month_3, max_value_3, 'bo', markersize=5) # Red
# # ax[3].text(max_month_3, max_value_3 + 400, f'Max: {int(max_value_3)}')

# # # Mark minimum point
# # ax[3].plot(min_month_3, min_value_3, 'ro', markersize=5) # Blue
# # ax[3].text(min_month_3, min_value_3 - 1100, f'Min: {int(min_value_3)}')

# for i, row in bills_df_central.iterrows():
#     if row['average per ticket'] < bills_df_central['average per ticket'].mean() - 1000:
#         ax[3].plot(row['month'], row['average per ticket'], 'o', color='red')
#     else:
#         ax[3].plot(row['month'], row['average per ticket'], 'o', color='blue')
```

```
#     if row['average per ticket'] < bills_df_central['average per
#             ax[3].plot(row['month'], row['average per ticket'], 'o',
# # plt.tight_layout()
```

```
In [27]: bills_month_df = df_purchased[df_purchased['month'].isin([1,2,3,4,5
                                         'Bill No' : 'nu
bills_month_df['average per ticket'] = bills_month_df['total after

bills_df_haram = df_purchased[(df_purchased['Store Name'] == 'الهرم
                                         'Bill No' : 'nu
bills_df_haram['average per ticket'] = bills_df_haram['total after

bills_df_maadi = df_purchased[(df_purchased['Store Name'] == 'معادي
                                         'Bill No' : 'nu
bills_df_maadi['average per ticket'] = bills_df_maadi['total after

bills_df_central = df_purchased[(df_purchased['Store Name'] == 'كزى
                                         'Bill No' : 'nu
bills_df_central['average per ticket'] = bills_df_central['total af

fig, ax = plt.subplots(figsize=(10, 6)) # Single plot instead of 2

# Main plot
ax.plot(bills_month_df['month'], bills_month_df['average per ticket'])

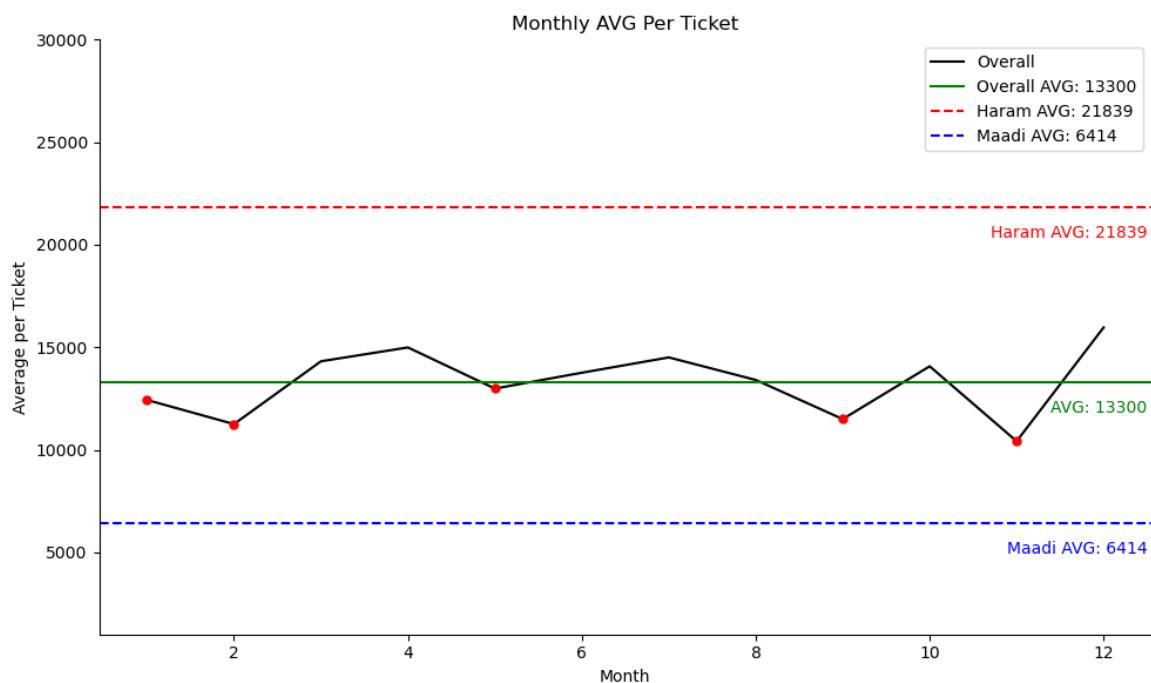
# Horizontal lines for averages
overall_avg = bills_month_df['average per ticket'].mean()
haram_avg = bills_df_haram['average per ticket'].mean()
maadi_avg = bills_df_maadi['average per ticket'].mean()

ax.axhline(y=overall_avg, color='green', linestyle='--', label=f'Overall AVG: {int(overall_avg)}')
ax.axhline(y=haram_avg, color='red', linestyle='--', label=f'Haram AVG: {int(haram_avg)}')
ax.axhline(y=maadi_avg, color='blue', linestyle='--', label=f'Maadi AVG: {int(maadi_avg)}')

# Text annotations
ax.text(12.5, overall_avg - 1500, f"AVG: {int(overall_avg)}", color='green')
ax.text(12.5, haram_avg - 1500, f"Haram AVG: {int(haram_avg)}", color='red')
ax.text(12.5, maadi_avg - 1500, f"Maadi AVG: {int(maadi_avg)}", color='blue')

# Highlight points below average
for i, row in bills_month_df.iterrows():
    if row['average per ticket'] < overall_avg:
        ax.plot(row['month'], row['average per ticket'], 'o', color='red')

# Aesthetics
ax.set_title('Monthly AVG Per Ticket')
ax.set_ylim(1000, 30000)
ax.set_xlabel('Month')
ax.set_ylabel('Average per Ticket')
sns.despine()
ax.legend()
plt.tight_layout()
plt.show()
```



In [28]: bills_df_central

Out [28]:

	month	total after discount tax 2	Bill No	average per ticket
0	8	8,240,539.5	201	40,997.7
1	9	9,713,188.2	230	42,231.3
2	10	9,447,068.5	224	42,174.4
3	11	5,861,724.3	193	30,371.6
4	12	11,979,555.6	222	53,962.0

In [29]: bills_df_maadi

Out [29]:

	month	total after discount tax 2	Bill No	average per ticket
0	1	4,944,715.4	752	6,575.4
1	2	4,979,251.2	813	6,124.5
2	3	5,465,472.4	791	6,909.6
3	4	4,357,332.5	626	6,960.6
4	5	5,746,323.5	813	7,068.0
5	6	4,304,604.2	581	7,409.0
6	7	5,884,271.9	736	7,994.9
7	8	3,315,882.7	663	5,001.3
8	9	4,637,956.2	972	4,771.6
9	10	4,184,971.3	898	4,660.3
10	11	5,134,374.4	876	5,861.2
11	12	7,483,346.0	980	7,636.1

In [30]: bills_df_haram

Out [30]:

	month	total after discount tax 2	Bill No	average per ticket
0	1	9,376,625.6	401	23,383.1
1	2	8,143,209.9	353	23,068.6
2	3	10,841,155.9	349	31,063.5
3	4	9,480,124.8	298	31,812.5
4	5	8,537,641.1	291	29,339.0
5	6	6,895,184.5	234	29,466.6
6	7	10,797,094.4	415	26,017.1
7	8	4,075,643.1	305	13,362.8
8	9	3,736,424.7	371	10,071.2
9	10	9,067,674.0	494	18,355.6
10	11	5,623,703.7	529	10,630.8
11	12	10,169,605.2	656	15,502.4

In [31]: fig, ax = plt.subplots(figsize=(8, 6)) # Single plot instead of 2x

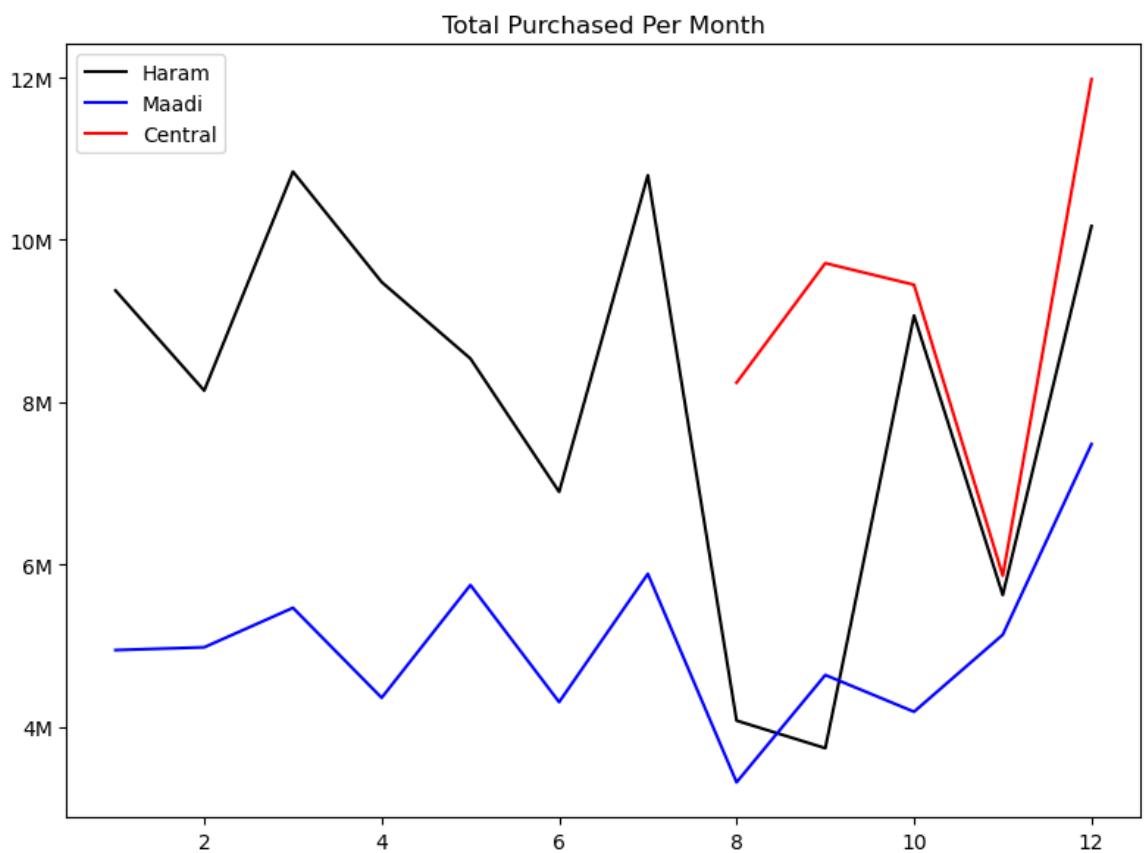
```
ax.plot(bills_df_haram['month'], bills_df_haram['total after discou
ax.plot(bills_df_maadi['month'], bills_df_maadi['total after discou
ax.plot(bills_df_central['month'], bills_df_central['total after di
```

```
def millions(x, pos):
    return f'{x * 1e-6:.0f}M'
```

```
ax.yaxis.set_major_formatter(FuncFormatter(millions))

ax.set_title('Total Purchased Per Month')

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



```
In [32]: bills_month_store = df_purchased[df_purchased['month'].isin([1,2,3,
                                                               'Bill No' : 'nu
bills_month_store
```

Out [32] :

	month	Store Name	total after discount tax 2	Bill No
0	1	وياك المعادي	4,944,715.4	752
1	1	وياك صيدلية الهرم	9,376,625.6	401
2	2	وياك المعادي	4,979,251.2	813
3	2	وياك صيدلية الهرم	8,143,209.9	353
4	3	وياك المعادي	5,465,472.4	791
5	3	وياك صيدلية الهرم	10,841,155.9	349
6	4	وياك المعادي	4,357,332.5	626
7	4	وياك صيدلية الهرم	9,480,124.8	298
8	5	وياك المعادي	5,746,323.5	813
9	5	وياك صيدلية الهرم	8,537,641.1	291
10	6	وياك المعادي	4,304,604.2	581
11	6	وياك صيدلية الهرم	6,895,184.5	234
12	7	وياك المعادي	5,884,271.9	736
13	7	وياك صيدلية الهرم	10,797,094.4	415
14	8	المخزن المركزي	8,240,539.5	201
15	8	وياك المعادي	3,315,882.7	663
16	8	وياك صيدلية الهرم	4,075,643.1	305
17	9	المخزن المركزي	9,713,188.2	230
18	9	وياك المعادي	4,637,956.2	972
19	9	وياك صيدلية الهرم	3,736,424.7	371
20	10	المخزن المركزي	9,447,068.5	224
21	10	وياك المعادي	4,184,971.3	898
22	10	وياك صيدلية الهرم	9,067,674.0	494
23	11	المخزن المركزي	5,861,724.3	193
24	11	وياك المعادي	5,134,374.4	876
25	11	وياك صيدلية الهرم	5,623,703.7	529
26	12	over stock	1.3	3
27	12	المخزن المركزي	11,979,555.6	222
28	12	وياك المعادي	7,483,346.0	980
29	12	وياك صيدلية الهرم	10,169,605.2	656

In [33] : bills_bill_store = df_purchased.groupby(['Bill No', 'Store Name', 'D

bills_bill_store

Out[33]:

		Bill No	Store Name	Date	total after discount tax 2
0		0000002284421	وياك صيدلية الهرم	2024-08-29	50,889.2
1		132758	وياك صيدلية الهرم	2024-10-02	277.9
2		2024102632938	وياك المعادى	2024-03-29	302.4
3		39229	وياك صيدلية الهرم	2024-08-26	21,648.7
4		47900	وياك صيدلية الهرم	2024-11-29	6,754.6
...					
15796		99900	وياك صيدلية الهرم	2024-03-21	428.5
15797		99949	وياك صيدلية الهرم	2024-11-28	9,569.2
15798	SINUPRET 50 SUGAR COATED TAB		وياك المعادى	2024-10-10	445.5
15799	TIRATAM 100MG/ML ORAL SOLUTION 120 ML		وياك المعادى	2024-01-16	-66.7
15800	limitless prenatal Max 30 soft gel cap		وياك المعادى	2024-09-26	-686.9

15801 rows × 4 columns

In [34]:

```
# sns.scatterplot(data= bills_bill_store, y= 'total after discount'
# plt.ylim(0,200000)
```

Haram AVG per ticket 4 times Maadi AVG per ticket

Decline after incline in Haraam

Maadi makes baby but stable steps in AVG per ticket

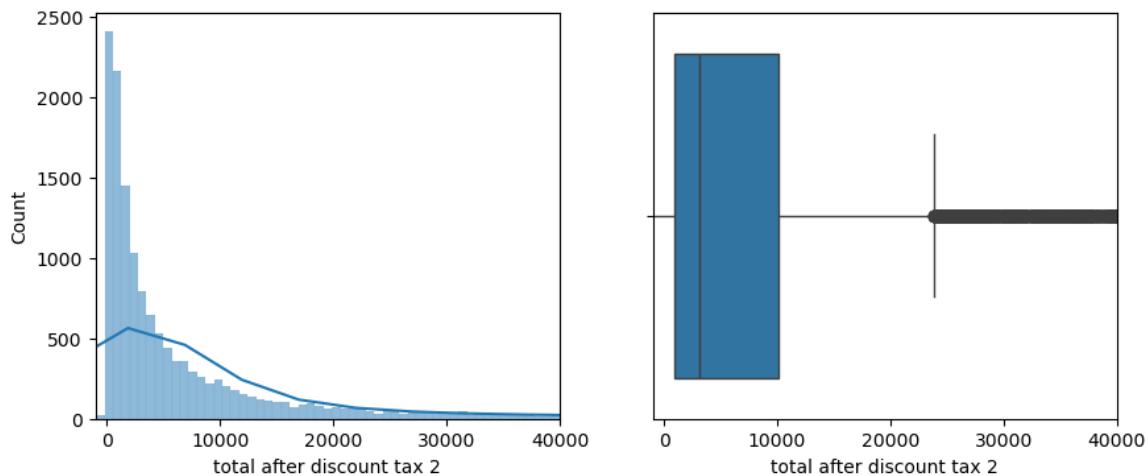
decrease number of bills in order to increase avg per ticket

Distribution of Bills

```
In [35]: fig, ax = plt.subplots(1,2, figsize = (10,4))
sns.histplot(df_purchased.groupby('Bill No')['total after discount tax 2'])
ax[0].set_xlim(-1000, 40000) # Set x limits

sns.boxplot(df_purchased.groupby('Bill No')['total after discount tax 2'])
ax[1].set_xlim(-1000, 40000) # Set x limits
```

Out[35]: (-1000.0, 40000.0)



The curve is right-skewed

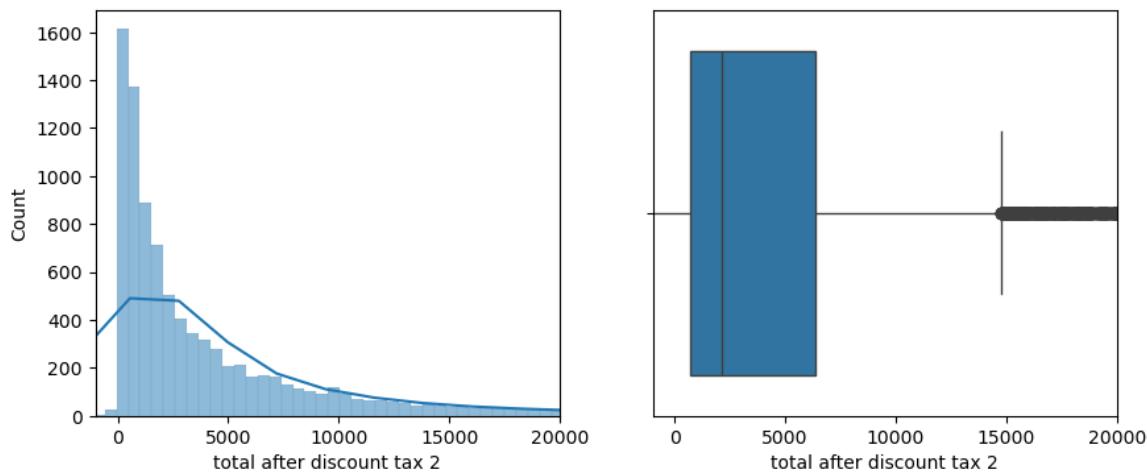
The majority of bills between 0 & 25000 EGP

The problem rise from retail bills

```
In [36]: fig, ax = plt.subplots(1,2, figsize = (10,4))
sns.histplot(df_purchased[df_purchased['Store Name'] == 'باق المعادي']
ax[0].set_xlim(-1000, 20000) # Set x limits

sns.boxplot(df_purchased[df_purchased['Store Name'] == 'باق المعادي']
ax[1].set_xlim(-1000, 20000) # Set x limits
```

Out[36]: (-1000.0, 20000.0)



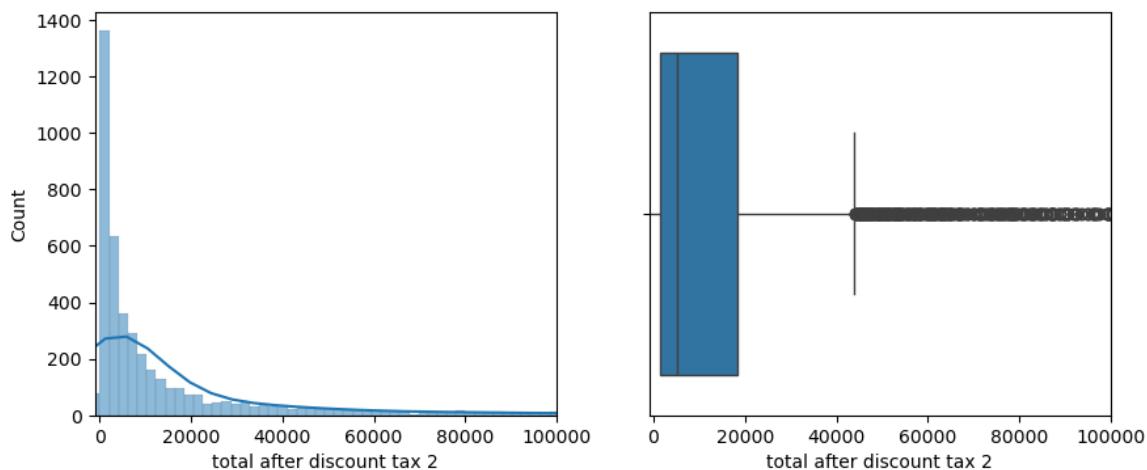
The curve is right-skewed

The majority of bills between 0 & 13000 EGP

```
In [37]: fig, ax = plt.subplots(1,2, figsize = (10,4))
sns.histplot(df_purchased[df_purchased['Store Name'] == 'يدلية الهرم'].set_xlim(-1000, 100000) # Set x limits

sns.boxplot(df_purchased[df_purchased['Store Name'] == 'يدلية الهرم'].set_xlim(-1000, 100000) # Set x limits
```

Out[37]: (-1000.0, 100000.0)



The curve is right-skewed

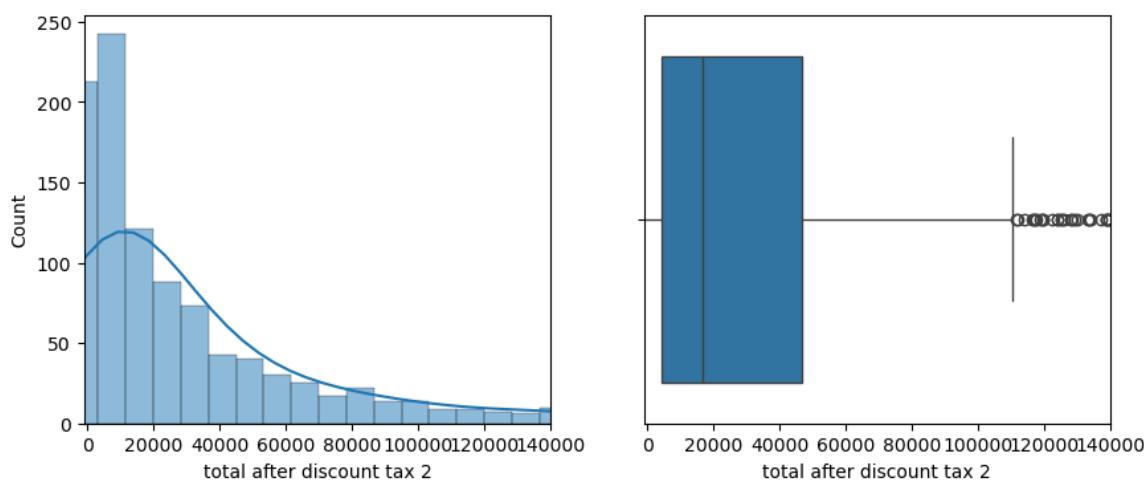
The majority of bills between 0 &

60000 EGP

```
In [38]: fig, ax = plt.subplots(1,2, figsize = (10,4))
sns.histplot(df_purchased[df_purchased['Store Name'] == 'خزن المركزي'] ,ax[0].set_xlim(-1000, 140000) # Set x limits

sns.boxplot(df_purchased[df_purchased['Store Name'] == 'خزن المركزي'] ,ax[1].set_xlim(-1000, 140000) # Set x limits
```

Out[38]: (-1000.0, 140000.0)



```
In [39]: bills_df = df_purchased.groupby(['Bill No','Supplier','Vendor Type'
                                         'Total'
                                         'total'
                                         'potential'
                                         'Discount %'] = bills_df['total discount 2'] / bills_df['Total']

average_per_ticket = bills_df['total after discount tax 2'].sum()/
print(f'average_per_ticket = {int(average_per_ticket)}')

bills_df_haram = df_purchased[df_purchased['Store Name'] == 'الهرم'
average_per_ticket_haram = bills_df_haram['total after discount tax 2'].sum()/
print(f'average_per_ticket_haram = {int(average_per_ticket_haram)}')

bills_df_maadi = df_purchased[df_purchased['Store Name'] == 'المعادي'
average_per_ticket_maadi = bills_df_maadi['total after discount tax 2'].sum()/
print(f'average_per_ticket_maadi = {int(average_per_ticket_maadi)}')

average_per_ticket = 13360
average_per_ticket_haram = 21184
average_per_ticket_maadi = 6439
```

Retail Bills (< 25000 EGP & < 20% Discount & > 25 bills per supplier(3 bills/month))

```
In [40]: retail_df = bills_df[bills_df['total after discount tax 2'] < 25000
retail_count = retail_df['Supplier'].value_counts().reset_index()
retail_supplier_sum = retail_df.groupby(['Supplier','Vendor Type'])
                                         'Total' :
```

```
'total dis
'potential
retail_supplier_sum['Discount %'] = retail_supplier_sum['total disc
retail_count_sum = retail_count[retail_count['count'] >= 25].merge(
print(round(retail_count_sum[retail_count_sum['Discount %'] < 20]['
print(round(retail_count_sum[retail_count_sum['Discount %'] < 20]['

# retail_count_sum
retail_count_sum[retail_count_sum['Discount %'] < 20]
```

16179058

6273

Out[40]:

	Supplier count	Vendor Type	total after discount tax 2	Total	total discount 2	potential total discount
0	متنوعون المعادي	2782	Retail	4,428,014.9	4,883,764.8	455,749.9
3	متنوعون الهرم	1067	Retail	2,856,426.6	2,996,209.3	139,782.7
5	الفاتح	684	Retail	2,260,614.1	2,809,484.1	548,870.0
7	مخزن الأخوة المتحدون	555	Retail	1,426,686.4	1,763,619.9	336,933.5
8	شركة مالقي ستورز فارما	370	Retail	1,207,925.2	1,453,382.3	245,457.1
10	شركة المصري	255	Retail	709,829.8	885,215.2	175,385.4
12	الشافعي فارما للتجارة والتوزيع	202	NA	905,322.2	1,100,116.9	194,794.7
15	مخزن جلوبال مصطفى	177	Retail	776,333.5	957,191.7	180,858.2
20	Imported (Maadi)	55	Retail	417,729.0	493,128.0	75,399.0
21	مخزن السعيد	40	NA	583,050.0	710,137.0	127,087.0
24	شركة ماкро فارما	32	NA	65,873.3	81,735.4	15,862.0
25	مخزن جلوبال / فارم د/ ابراهيم الروبي	29	Retail	354,733.0	434,856.6	80,123.6
26	شركة شفاء	25	Retail	186,519.6	230,418.8	43,899.2
						73,081.5

In [41]: # top_suppliers

In [42]: print(df_purchased['total after discount tax 2'].sum())
print(bills_df['Bill No'].count())

202424666.05273923

15164

In [43]: round(retail_count_sum[retail_count_sum['Discount %'] < 20]['total

Out[43]: 8.0

In [44]: round(retail_count_sum[retail_count_sum['Discount %'] < 20]['count'])

Out[44]: 41.4

In [45]: retail_supplier_sum

Out[45]:

	Supplier	Vendor Type	total after discount tax 2	Total	total discount 2	potential total discount	Disc%
0	el farouk	NA	5,761.0	7,486.0	1,725.1	2,650.4	!
1	المتحده للصيادله	NA	1,143,052.6	1,448,830.6	305,778.0	447,098.3	!
2	هيثم محمد	NA	12,527.5	25,055.0	12,527.5	12,843.8	!
3	محزن 2020	Retail	8,505.8	10,322.8	1,816.9	3,826.2	!
4	AKHNATON COMPANY	Bulk	93,444.3	118,405.6	24,961.3	33,220.4	!
...
61	مصر ميديكال	NA	5,050.1	5,962.0	911.9	1,923.5	!
62	ميديكال فارما _ اونر نقدي	NA	960,980.4	1,225,476.7	264,496.3	367,793.9	!
63	ميديكال فارما - اونر	Retail	7,438.4	9,419.0	1,980.6	3,231.3	!
64	نالي فارم انترناشيونال	NA	16,192.7	19,313.2	3,120.6	5,603.4	!
65	هاي ستورز فارما	NA	50,127.0	61,128.9	11,001.9	17,175.6	!

66 rows × 7 columns

In [46]: bills_df

Out[46]:

	Bill No	Supplier	Vendor Type	total after discount tax 2	Total	total discount 2
0	0000002284421	شركة سوفييكو فارم	Distribution	50,889.2	62,060.0	11,170.8
1	132758	الفاتح	Retail	277.9	348.0	70.1
2	2024102632938	ابن سينا فارما	Distribution	302.4	360.0	57.6
3	39229	شركة جلوبال فارما	Retail	21,648.7	28,668.5	7,019.8
4	47900	شركة جلوبال فارما	Retail	6,754.6	8,944.0	2,189.4
...
15159	99900	مخزن الأخوة المتخدون	Retail	428.5	629.0	200.5
15160	99949	مخزن جلوبال فارم / ابراهيم الروبي	Retail	9,569.2	11,681.0	2,111.8
15161	SINUPRET 50 SUGAR COATED TAB	متتنوعون المعادي	Retail	445.5	479.0	33.5
15162	TIRATAM 100MG/ML ORAL SOLUTION 120 ML	مخزن جلوبال مصطفى	Retail	-66.7	-78.5	-11.8
15163	limitless prenatal Max 30 soft gel cap	متتنوعون الهرم	Retail	-686.9	-723.0	-36.1

15164 rows × 8 columns

In [47]:

```
haram_monthly_discount = df_purchased[df_purchased['Store Name'] == 'Haram Discount %']

haram_monthly_discount['Haram Discount %'] = haram_monthly_discount['Haram Discount %'].sum()
haram_monthly_discount['Haram Discount %'] = haram_monthly_discount['Haram Discount %'].mean()

maadi_monthly_discount = df_purchased[df_purchased['Store Name'] == 'Maadi Discount %']

maadi_monthly_discount['Maadi Discount %'] = maadi_monthly_discount['Maadi Discount %'].sum()
maadi_monthly_discount['Maadi Discount %'] = maadi_monthly_discount['Maadi Discount %'].mean()
```

```

maadi_monthly_discount['Maadi Discount %'] = maadi_monthly_discount
maadi_monthly_discount

central_monthly_discount = df_purchased[df_purchased['Store Name'] == 'Central']

central_monthly_discount['Central Discount %'] = central_monthly_discount

```

In [48]: haram_monthly_discount

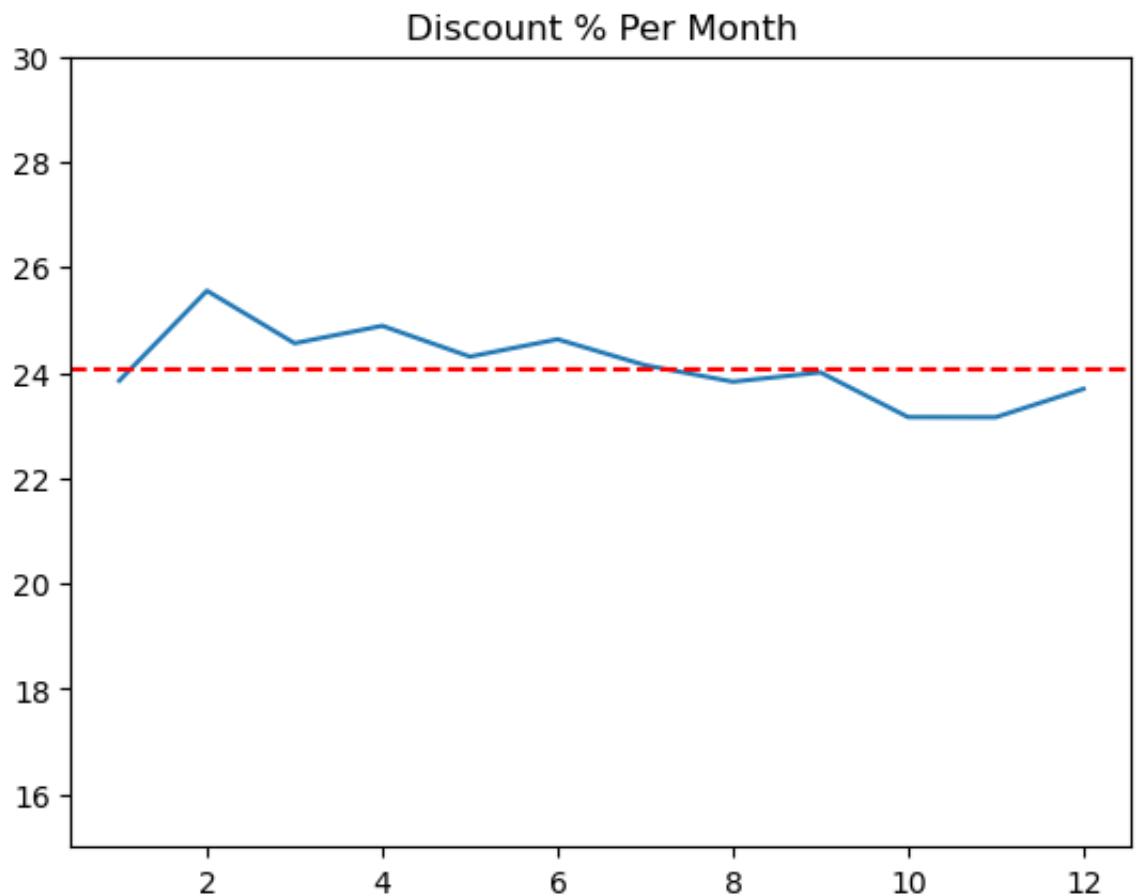
Out[48]:

	month	total after discount tax 2	Total	total discount 2	Haram Discount %
0	1	9,376,625.6	12,285,051.4	2,908,425.8	23.7
1	2	8,143,209.9	11,030,031.0	2,886,821.1	26.2
2	3	10,841,155.9	14,359,792.1	3,518,636.2	24.5
3	4	9,480,124.8	12,662,430.9	3,182,306.1	25.1
4	5	8,537,641.1	11,298,544.3	2,760,903.2	24.4
5	6	6,895,184.5	9,240,810.4	2,345,625.9	25.4
6	7	10,797,094.4	14,332,569.4	3,535,475.0	24.7
7	8	4,075,643.1	5,228,148.0	1,152,504.9	22.0
8	9	3,736,424.7	4,691,782.0	955,357.3	20.4
9	10	9,067,674.0	11,480,541.9	2,412,867.9	21.0
10	11	5,623,703.7	7,067,319.5	1,443,615.8	20.4
11	12	10,169,605.2	13,011,635.9	2,842,030.7	21.8

In [49]: df_discount_month = round(df_purchased.groupby('month').agg({'total': 'sum', 'Total': 'sum', 'total': 'sum'}), 2)

df_discount_month['Discount %'] = df_discount_month['total discount
Discount_month_plot = df_discount_month.plot(kind = 'line', x = 'month',
Discount_month_plot.set_title('Discount % Per Month')
Discount_month_plot.set_xlabel('')
Discount_month_plot.get_legend().remove()
Discount_month_plot.axhline(y = (df_discount_month['total discount']
plt.ylim(15, 30)

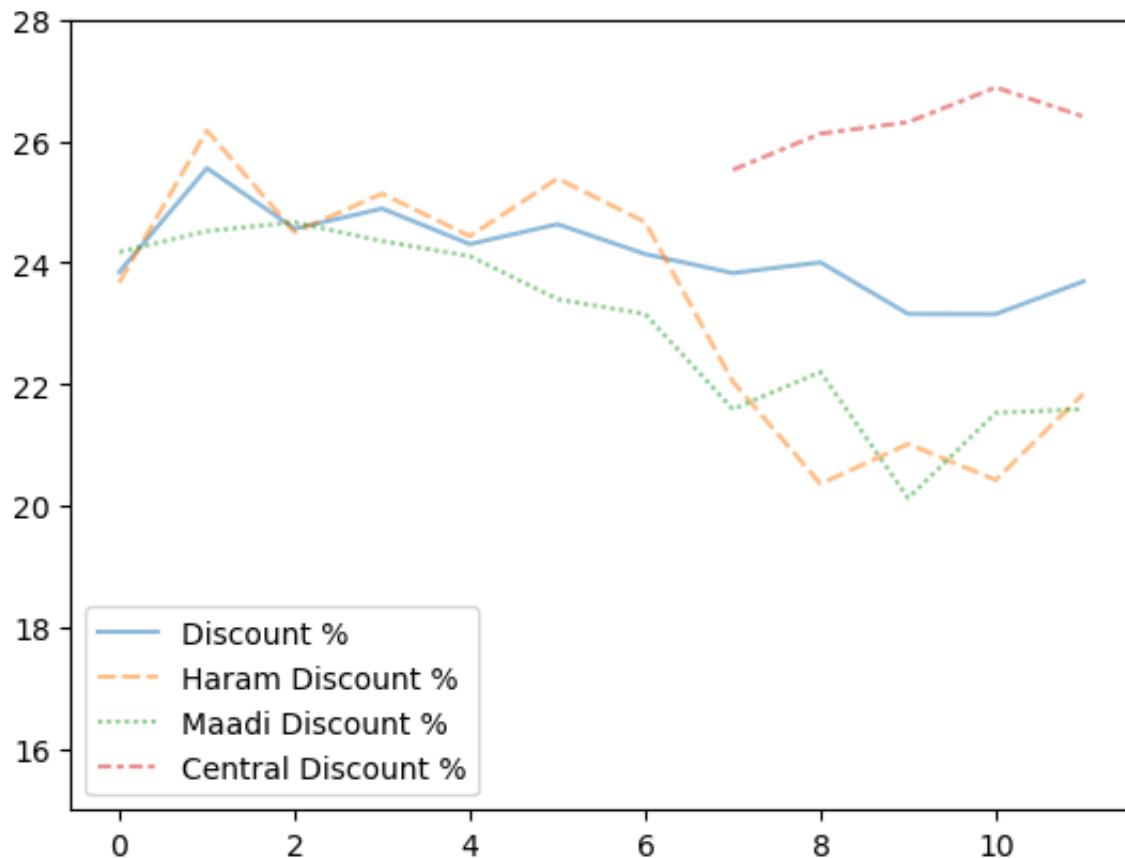
Out[49]: (15.0, 30.0)



In []:

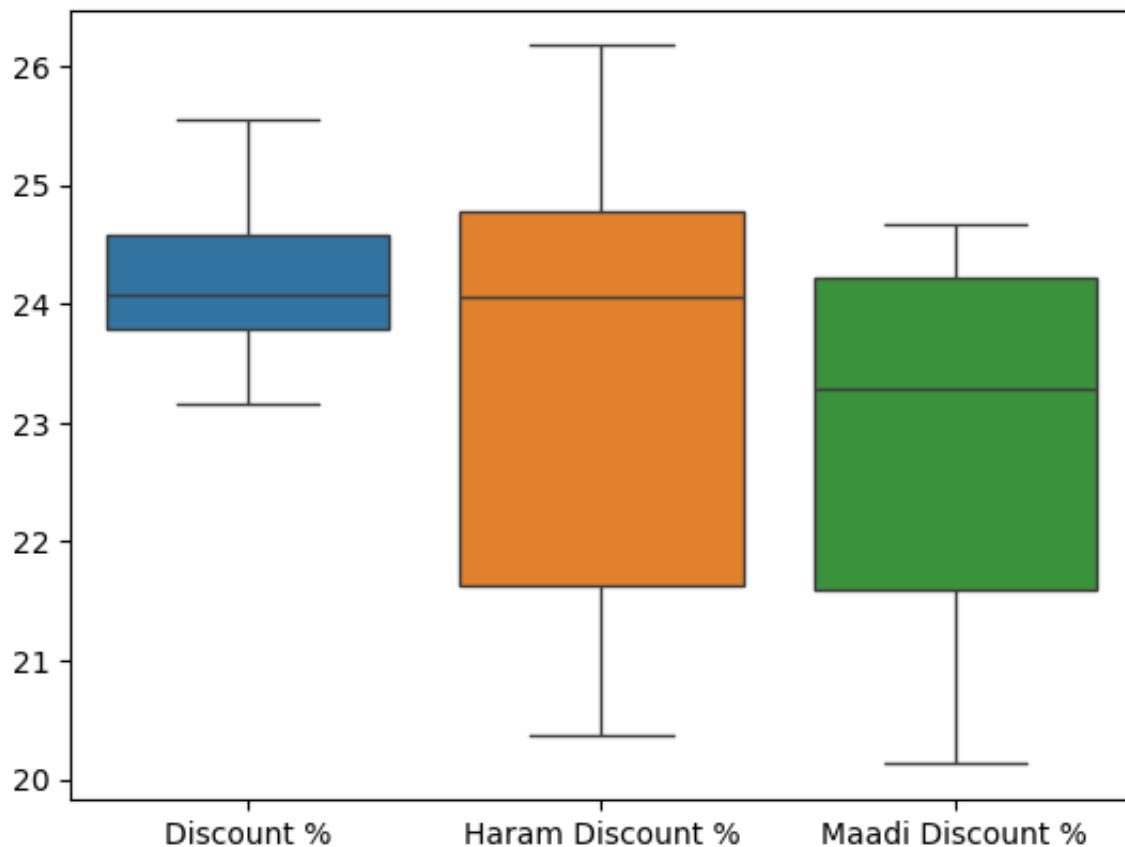
```
In [50]: monthly_discount = df_discount_month.merge(haram_monthly_discount[[  
monthly_discount = monthly_discount.merge(maadi_monthly_discount[[  
monthly_discount = monthly_discount.merge(central_monthly_discount[  
  
sns.lineplot(data = monthly_discount[['Discount %', 'Haram Discount  
# sns.lineplot(data = haram_monthly_discount[['month', 'Haram Disco  
  
plt.ylim(15,28)
```

Out[50]: (15.0, 28.0)



```
In [51]: sns.boxplot(data = monthly_discount[['Discount %', 'Haram Discount %', 'Maadi Discount %', 'Central Discount %']])
```

```
Out[51]: <Axes: >
```

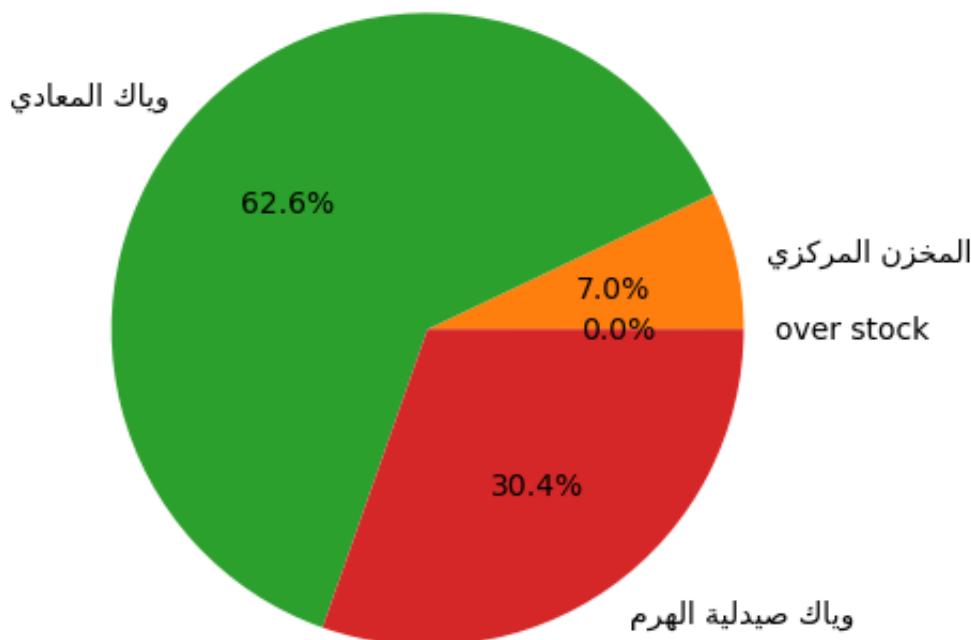


```
In [52]: import arabic_reshaper  
from bidi.algorithm import get_display
```

```
def reshape_arabic(text):
    reshaped_text = arabic_reshaper.reshape(text) # Reshape the Arabic text
    return get_display(reshaped_text) # Return text suitable for R

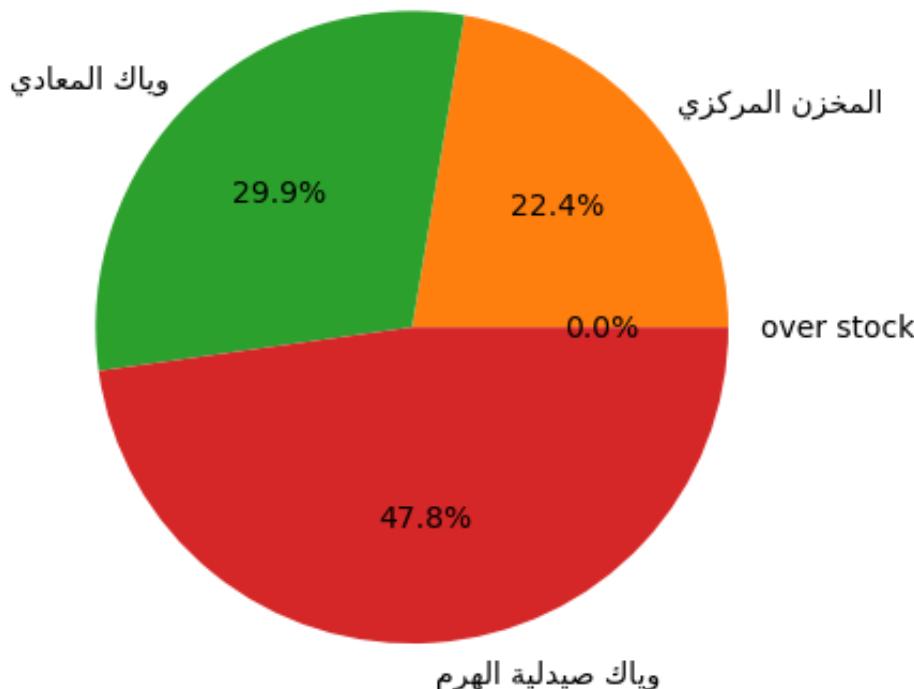
df_bill_store = df_purchased.groupby('Store Name')['Bill No'].nunique()
df_bill_store['Store Name'] = df_bill_store['Store Name'].apply(reshape_arabic)
bills_store_plot = df_bill_store.plot(kind = 'pie',y= 'Bills Count')
bills_store_plot.set_title('Bills No Per Store')
bills_store_plot.set_xlabel('')
bills_store_plot.set_ylabel('')
bills_store_plot.get_legend().remove()
```

Bills No Per Store



```
In [53]: df_price_store = df_purchased.groupby('Store Name')['total after discount'].sum()
df_price_store['Store Name'] = df_price_store['Store Name'].apply(reshape_arabic)
Price_store_plot = df_price_store.plot(kind = 'pie',y= 'Total Price')
Price_store_plot.set_title('Total Purchased')
Price_store_plot.set_xlabel('')
Price_store_plot.set_ylabel('')
Price_store_plot.get_legend().remove()
```

Total Purchased



```
In [54]: df_return_filtered = df[df['Type'] == 'Return']
df_return_filtered[['Store Name', 'total after discount tax 2']]
return_items = df_return_filtered.groupby('Store Name')['total after d
return_items['Total Return'] = return_items['Total Return'].abs()

# df_purchased_filtered = df[df['Type'] == 'Purchase']
# df_purchased_filtered[['Store Name', 'total public without tax']]
purchased_items = df_purchased.groupby('Store Name')['total after d

df_category = return_items.merge(purchased_items)

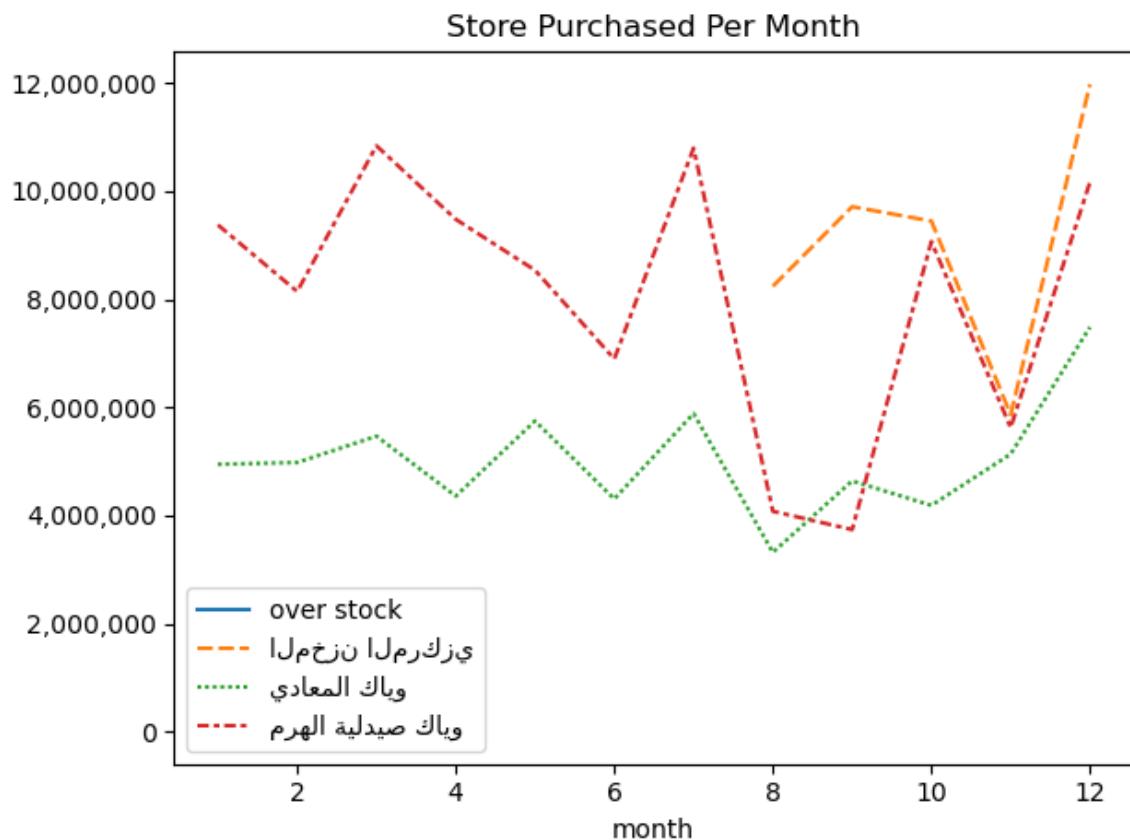
df_category['Return %'] = round(df_category['Total Return'] / (df_c
# df_category['Total Return'] = df_category['Total Return'].apply(l
# df_category['Total Purchased'] = df_category['Total Purchased'].a

df_category
```

	Store Name	Total Return	Total Purchased	Return %
0	over stock	9,962.3	1.3	100.0
1	المخزن المركزي	1,652,649.4	45,242,076.1	3.5
2	وياك المعادي	1,543,014.1	60,438,501.7	2.5
3	وياك صيدلية الهرم	4,014,969.7	96,744,086.9	4.0

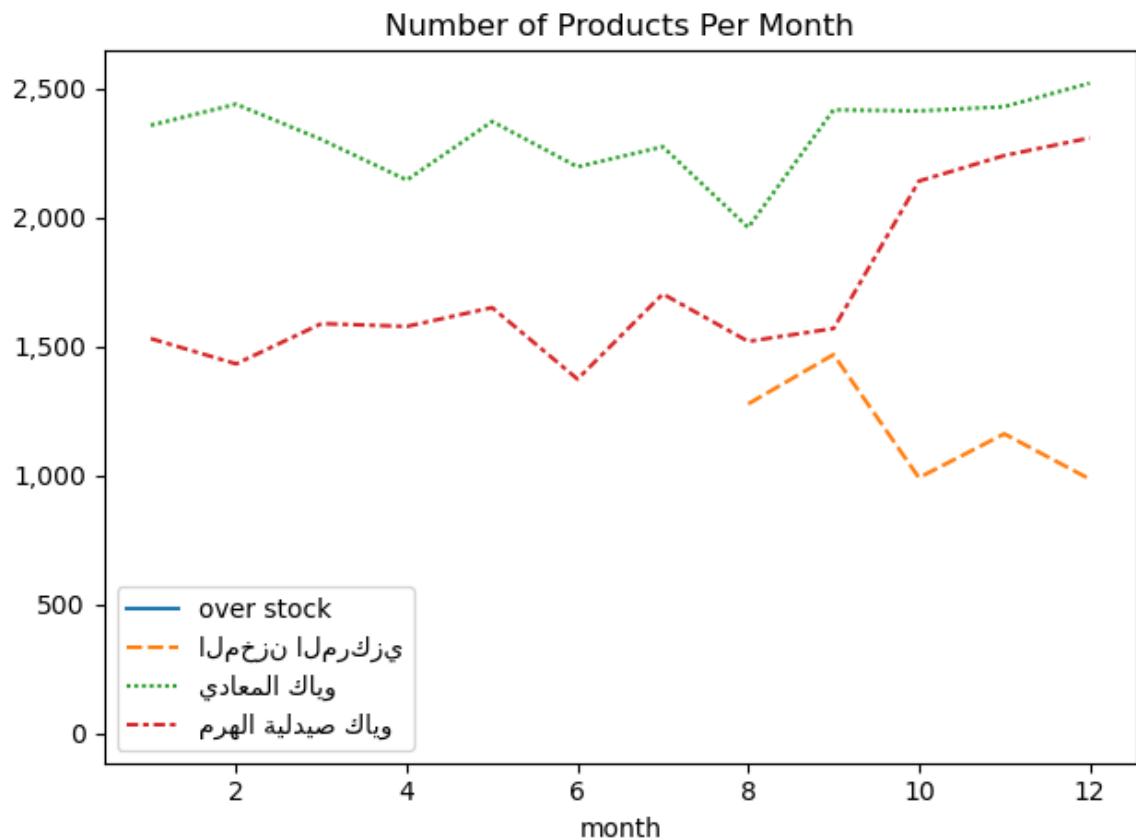
```
In [55]: import matplotlib.ticker as ticker
```

```
df_stores_month = df_purchased.pivot_table(index = 'month', columns='store', values='quantity').round(0)
df_stores_month = df_stores_month.rename(columns={0: 'وياك المعادي', 1: 'وياك صيدلية الهرم' })
sns.lineplot(data = df_stores_month)
plt.gca().yaxis.set_major_formatter(ticker.FuncFormatter(lambda x, pos: '{:,}'.format(x)))
plt.legend(loc='best')
plt.title('Store Purchased Per Month')
plt.tight_layout()
plt.show()
```

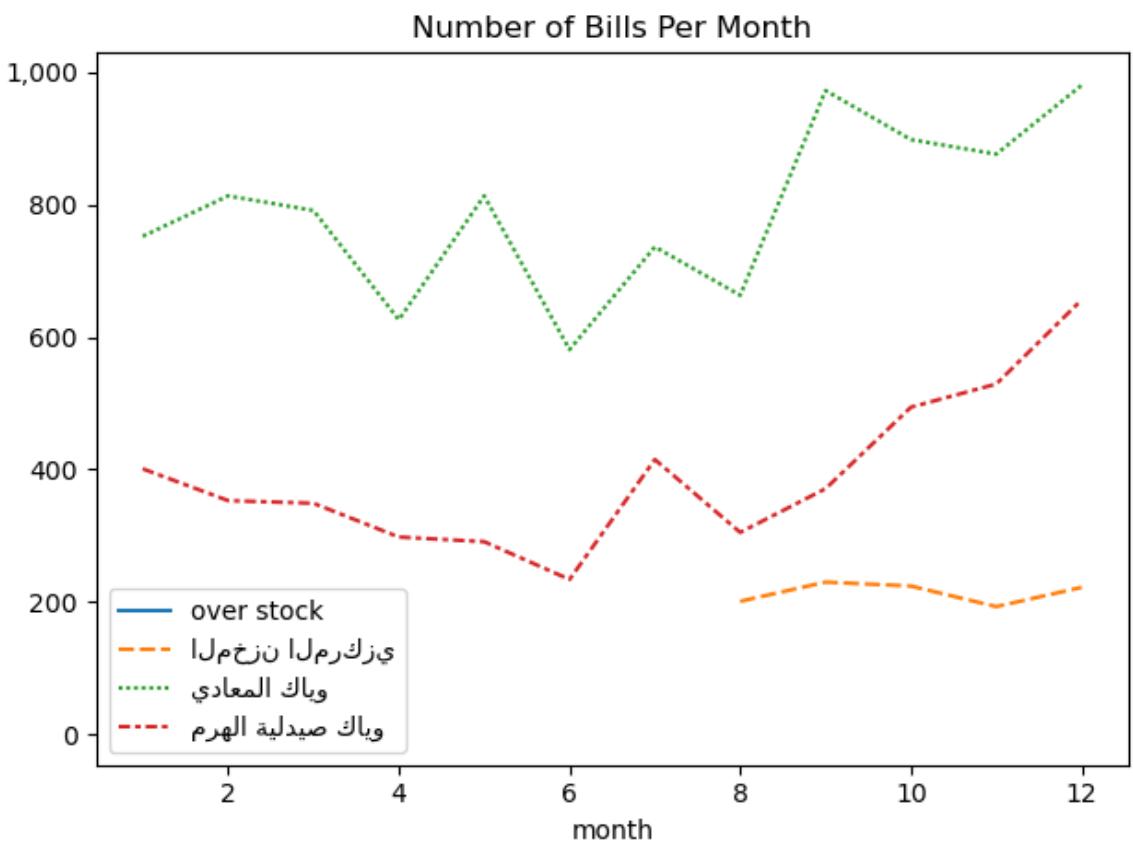


```
In [56]:
```

```
df_stores_products_month = df_purchased.pivot_table(index = 'month', columns='store', values='products').round(0)
df_stores_products_month = df_stores_products_month.rename(columns={0: 'وياك صيدلية الهرم', 1: 'وياك المعادي', 2: 'يزكرملا نزحملأا', 3: 'over stock'})
sns.lineplot(data = df_stores_products_month)
plt.gca().yaxis.set_major_formatter(ticker.FuncFormatter(lambda x, pos: '{:,}'.format(x)))
plt.legend(loc='best')
plt.title('Number of Products Per Month')
plt.tight_layout()
plt.show()
```



```
In [57]: df_stores_bills_month = df_purchased.pivot_table(index = 'month', c  
df_stores_bills_month = df_stores_bills_month.round(0)  
df_stores_bills_month = df_stores_bills_month.rename(columns={  
    'عادي': 'ي ز ك ر م ل ا ن ZX م ل ا',  
    'ياك صيدلية الهرم': 'و ياك المعادي'  
})  
sns.lineplot(data = df_stores_bills_month)  
plt.gca().yaxis.set_major_formatter(ticker.FuncFormatter(lambda x,  
    plt.legend(loc='best')  
    plt.title('Number of Bills Per Month')  
    plt.tight_layout()  
    plt.show()
```



Top Suppliers by Sales

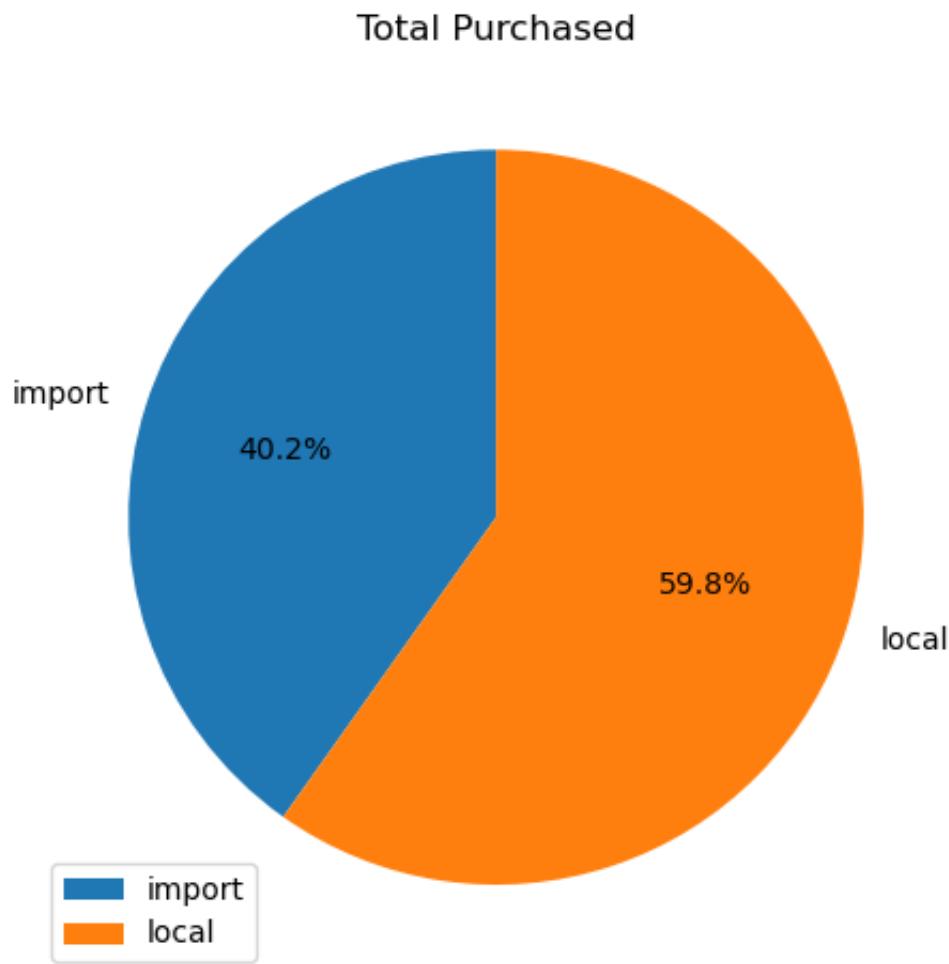
```
In [58]: top_suppliers = df_purchased.groupby('Supplier').agg({'total after tax': 'sum', 'total discount': 'sum'})  
top_suppliers['Discount %'] = top_suppliers['total discount 2'] / top_suppliers['total after tax']  
# top_suppliers['total public without tax'] = top_suppliers['total after tax'] - top_suppliers['total discount']  
top_suppliers.round(1).reset_index().head(10)
```

Out [58] :

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
0	ابن سينا فارما	53,135,119.5	70,413,196.8	17,278,077.4	24.5
1	شركة فارما اوفر سبيز	43,213,845.1	57,262,833.3	14,048,988.2	24.5
2	شركة جلوبال فارما	19,423,966.0	25,645,432.6	6,221,466.5	24.3
3	شركة الاهرام شبين الكوم	17,587,243.4	23,432,663.0	5,845,419.5	24.9
4	مخزن السعيد	6,821,898.3	8,542,555.0	1,720,656.7	20.1
5	الفاروق	5,804,709.8	7,653,719.0	1,849,009.3	24.2
6	AUG PHARMA	4,947,294.1	8,448,029.0	3,500,734.9	41.4
7	Wecare	4,670,226.4	6,317,893.2	1,647,666.8	26.1
8	متنوعون المعادى	4,506,066.3	4,964,999.4	458,933.1	9.2
9	مخزن داوى	4,339,712.2	5,320,831.5	981,119.3	18.4

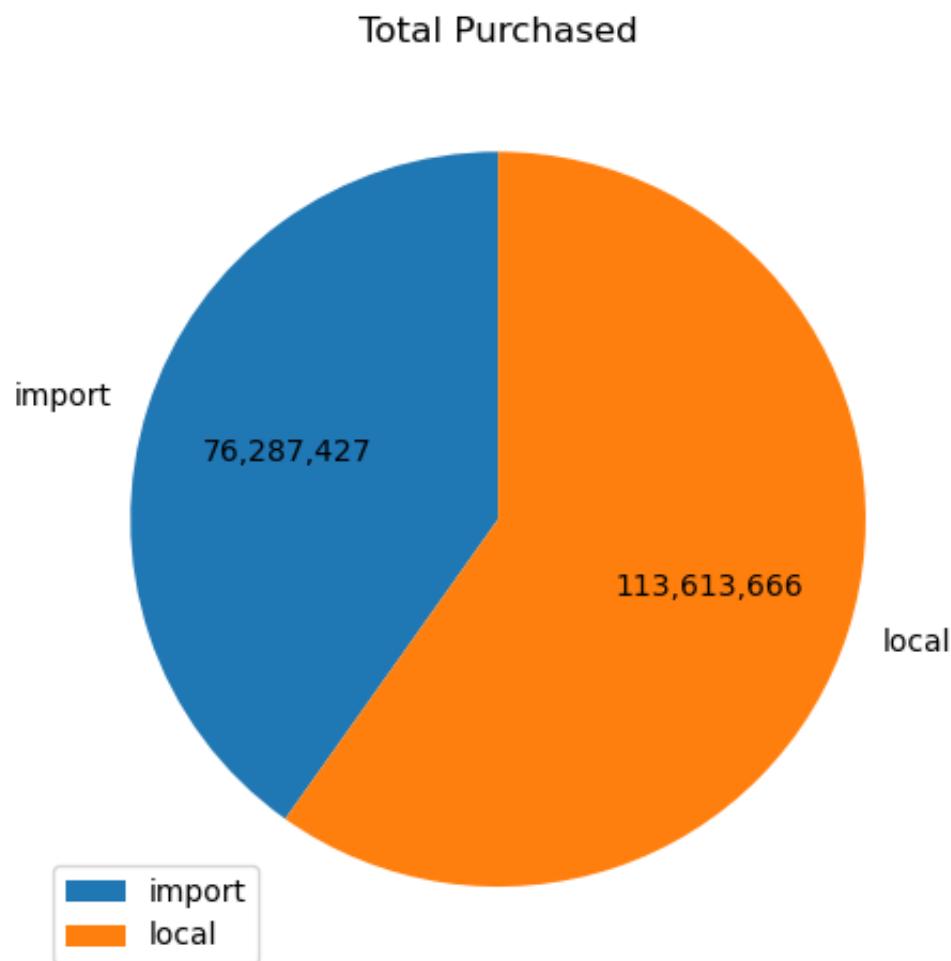
In [59] :

```
df_category_purchased = df_purchased[df_purchased['Category'].isin(
category_purchased_plot = df_category_purchased.plot(kind = 'pie',y
category_purchased_plot.set_title('Total Purchased')
category_purchased_plot.set_xlabel('')
category_purchased_plot.set_ylabel('')
plt.legend(loc='lower left')
plt.tight_layout()
```



```
In [60]: def label_func(pct, all_vals):
    total = sum(all_vals)
    val = int(round(total * pct / 100.0))
    return f'{val:,}'

df_category_purchased = df_purchased[df_purchased['Category'].isin(
category_purchased_plot = df_category_purchased.plot(kind = 'pie',y
category_purchased_plot.set_title('Total Purchased')
category_purchased_plot.set_xlabel('')
category_purchased_plot.set_ylabel('')
plt.legend(loc='lower left')
plt.tight_layout()
```



```
In [61]: df_category_purchased
```

```
Out[61]:    Category  Total Purchased
```

	Category	Total Purchased
0	import	76,287,426.9
1	local	113,613,665.4

```
In [62]: supplier_category = df_purchased.pivot_table(index = 'Supplier', columns = 'Category', values = 'Purchased', aggfunc = sum)  
supplier_category  
# supplier_category['local'].apply(lambda x: '{:, .2f}'.format(x))
```

Out[62]:

Category	Supplier	local	import
10	ابن سينا فارما	31,146,352.4	20,826,212.9
40	شركة فارما اوفر سيز	25,401,736.6	16,926,580.8
34	شركة جلوبال فارما	11,682,477.7	7,648,712.0
30	شركة الاهرام شبين الكوم	8,659,048.0	8,905,384.1
5	AUG PHARMA	4,947,294.1	NaN
...
25	ديفارت	-45,000.0	NaN
21	القصـوـاء	-82,380.9	NaN
51	مجموعة اتش اس فارما	NaN	6,500.0
55	مخزن السعيد	NaN	1,925,098.3
71	هنا فارم	NaN	39,468.0

72 rows × 3 columns

In [63]:

```

df_supplier = supplier_category.merge(top_suppliers, on= 'Supplier'
# df_supplier['local'] = df_supplier['local'].apply(lambda x : '{}',
# df_supplier['import'] = df_supplier['import'].apply(lambda x : '{}'

df_supplier['local %'] = round(df_supplier['local'] / df_supplier['total'] * 100)
df_supplier['import %'] = round(df_supplier['import'] / df_supplier['total'] * 100)

df_supplier['Discount %'] = round(df_supplier['Discount %'],1)
df_supplier = df_supplier.sort_values(by= 'total after discount tax', ascending=False)

# df_supplier['local'] = df_supplier['local'].apply(lambda x : '{}',
# df_supplier['import'] = df_supplier['import'].apply(lambda x : '{}'
# df_supplier['total public without tax r'] = df_supplier['total public without tax r'].apply(lambda x : '{}'

df_supplier = df_supplier.reset_index(drop= True)

df_supplier['Supplier Rank by Sales'] = df_supplier['total after discount tax'].rank(ascending=False)

df_supplier.head(10)

```

Out[63]:

	Supplier	local	import	total after discount tax 2	Total	to discoun
0	ابن سينا فارما	31,146,352.4	20,826,212.9	53,135,119.5	70,413,196.8	17,278,07
1	شركة فارما أوفر سيز	25,401,736.6	16,926,580.8	43,213,845.1	57,262,833.3	14,048,98
2	شركة جلوبال فارما	11,682,477.7	7,648,712.0	19,423,966.0	25,645,432.6	6,221,46
3	شركة الاهرام شبين الكوم	8,659,048.0	8,905,384.1	17,587,243.4	23,432,662.9	5,845,41
4	مخزن السعيد	Nan	1,925,098.3	6,821,898.3	8,542,555.0	1,720,65
5	الفاروق	3,569,318.8	2,234,570.5	5,804,709.8	7,653,719.0	1,849,00
6	AUG PHARMA	4,947,294.1	Nan	4,947,294.1	8,448,029.0	3,500,73
7	Wecare	3,178,931.0	1,456,617.0	4,670,226.4	6,317,893.2	1,647,66
8	متنوعون المعادى	2,403,361.0	1,935,660.5	4,506,066.3	4,964,999.4	458,93
9	مخزن داوى	320,621.3	631,990.9	4,339,712.2	5,320,831.5	981,11

In [64]: df_purchased.pivot_table(index= 'month', columns= 'Store Name', val

Out[64]:

Store Name	over stock	المخزن المركزي	وياك المعادى	وياك صيدلية الهرم	وياك صيدلية الهرم	over stock	المخزن المركزي	المخزن المركزي			
month											
1	NaN		NaN	4,944,715.4	9,376,625.6	NaN		NaN	1,5		
2	NaN		NaN	4,979,251.2	8,143,209.9	NaN		NaN	1,6		
3	NaN		NaN	5,465,472.4	10,841,155.9	NaN		NaN	1,7		
4	NaN		NaN	4,357,332.5	9,480,124.8	NaN		NaN	1,4		
5	NaN		NaN	5,746,323.5	8,537,641.1	NaN		NaN	1,8		
6	NaN		NaN	4,304,604.2	6,895,184.5	NaN		NaN	1,3		
7	NaN		NaN	5,884,271.9	10,797,094.4	NaN		NaN	1,7		
8	NaN	8,240,539.5	3,315,882.7	4,075,643.1	NaN	2,824,399.4	9				
9	NaN	9,713,188.2	4,637,956.2	3,736,424.7	NaN	3,434,193.5	1,3				
10	NaN	9,447,068.5	4,184,971.3	9,067,674.0	NaN	3,372,916.9	1,0				
11	NaN	5,861,724.3	5,134,374.4	5,623,703.7	NaN	2,155,148.3	1,4				
12	1.3	11,979,555.6	7,483,346.0	10,169,605.2	-1.3	4,297,236.3	2,0				

In [65]:

```
top_suppliers_sales = df_supplier.loc[:10,'Supplier']
# top_suppliers_sales = top_suppliers_sales.to_list()

new_df = df_purchased[df_purchased['Supplier'].isin(top_suppliers_s
# new_df['Supplier 2'] = new_df['Supplier'].apply(reshape_arabic)

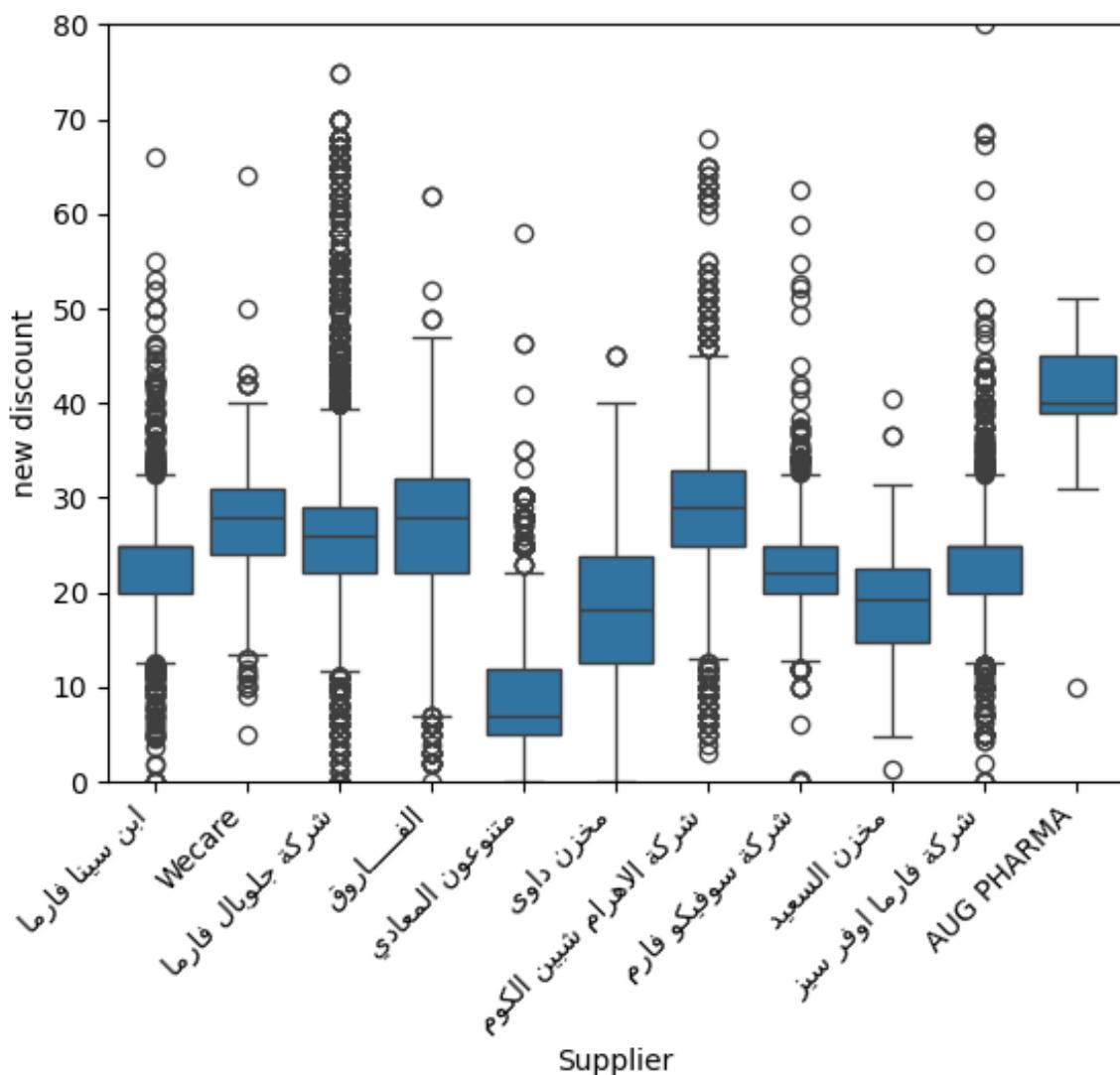
sns.boxplot(data = new_df, x = new_df['Supplier'], y = 'new discoun

ax = plt.gca() # Get the current axis
tick_labels = ax.get_xticklabels() # Retrieve current tick labels
formatted_labels = [reshape_arabic(label.get_text()) for label in t

plt.xticks(ticks=ax.get_xticks(), labels=formatted_labels)

plt.xticks(rotation=45, ha='right')

plt.ylim(0,80)
plt.show()
```



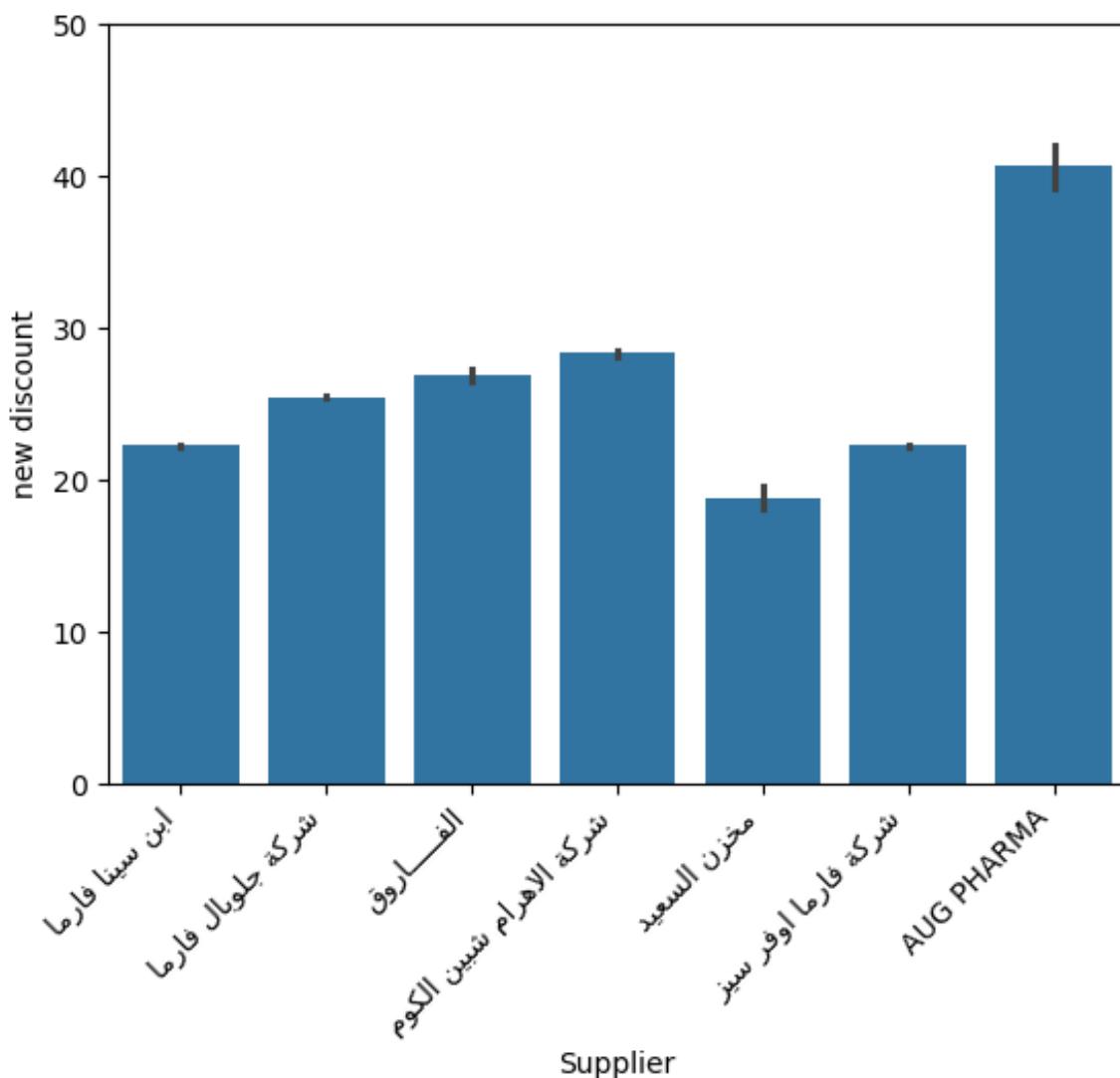
```
In [66]: top_suppliers_sales = df_supplier.loc[:6,'Supplier']
# top_suppliers_sales = top_suppliers_sales.to_list()

new_df = df_purchased[df_purchased['Supplier'].isin(top_suppliers_s
# new_df['Supplier 2'] = new_df['Supplier'].apply(reshape_arabic)

sns.barplot(data = new_df, x = new_df['Supplier'], y = 'new discoun
ax = plt.gca() # Get the current axis
tick_labels = ax.get_xticklabels() # Retrieve current tick labels
formatted_labels = [reshape_arabic(label.get_text()) for label in t
plt.xticks(ticks=ax.get_xticks(), labels=formatted_labels)

plt.xticks(rotation=45, ha='right')

plt.ylim(0,50)
plt.show()
```



```
In [67]: df_return_filtered = df[df['Type'] == 'Return']
df_return_filtered[['Supplier', 'total after discount tax 2']]
return_items_supplier = df_return_filtered.groupby('Supplier')['tot
return_items_supplier['Total Return'] = return_items_supplier['Tot

# df_purchased_filtered_supplier = df[df['Type'] == 'Purchase']
# df_purchased_filtered_supplier[['Supplier', 'total public without
purchased_items_supplier = df_purchased_supplier.groupby('Supplier')['total

df_category_supplier = return_items_supplier.merge(purchased_items_
df_category_supplier['Return %'] = round(df_category_supplier['Tota
# df_category_supplier['Total Return'] = df_category_supplier['Total
# df_category_supplier['Total Purchased'] = df_category_supplier['T

df_category_supplier.head(20)
```

Out[67]:

	Supplier	Total Return	Total Purchased	Return %
0	el farouk	6,219.9	0.0	100.0
1	المتحده للصيادله	46,592.8	2,158,475.4	2.1
2	محزن 2020	78,686.6	-0.1	100.0
3	AKHNATON COMPANY	265.5	520,705.4	0.1
4	AUG PHARMA	259,240.0	4,947,294.1	5.0
5	Imported (Maadi)	3,357.4	1,206,899.3	0.3
6	Wecare	42,552.2	4,670,226.4	0.9
7	ابن سينا فارما	1,507,782.1	53,135,119.5	2.8
8	اكتوبر فارما	22.4	-0.0	100.1
9	الاصدقاء فارم	1,510.5	201,699.6	0.7
10	البدري تو ام للتجارة والتوزيع	189.0	25,342.2	0.7
11	الشافعي فارما للتجارة والتوزيع	64,176.8	1,479,774.5	4.2
12	الشركة العربية لتجارة الادوية والمستلزمات الطبية	7,878.1	229,485.1	3.3
13	الشركة الوطنية للصيادلة	157.5	61,075.9	0.3
14	الشمس فارم	23,490.6	1,472,381.7	1.6
15	الفتح لتجارة وتوزيع الادوية	37,328.9	1,600,282.7	2.3
16	الفاتح	58,341.1	2,394,842.8	2.4
17	الفاروق	277,120.3	5,804,709.8	4.6
18	القصوء	82,985.9	-82,380.9	13,716.7
19	المحمدية	9,212.4	303,849.2	2.9

In [68]: df_category_supplier['Total Return'].sum()

Out[68]: 7220595.408871965

Suppliers Data

In [69]: df_supplier_data = df_supplier.merge(df_category_supplier[['Supplier', 'Category', 'SubCategory', 'Total Return', 'Total Purchased', 'Return %']], left_on='Supplier', right_on='Supplier')

Out[69]:

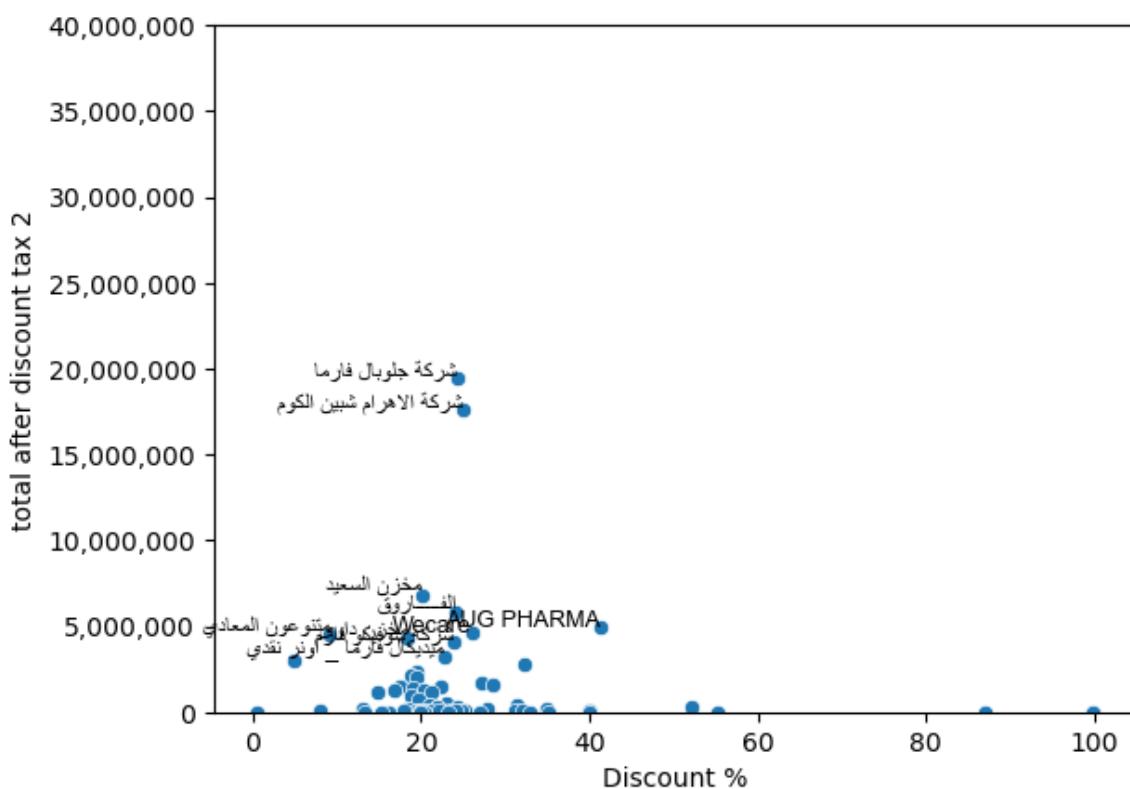
	Supplier	local	import	total after discount tax 2	Total	total discount
0	ابن سينا فارما	31,146,352.4	20,826,212.9	53,135,119.5	70,413,196.8	17,278,011.0

1	شركة فارما اوفر سيز	25,401,736.6	16,926,580.8	43,213,845.1	57,262,833.3	14,048,98
2	شركة جلوبال فارما	11,682,477.7	7,648,712.0	19,423,966.0	25,645,432.6	6,221,46
3	شركة الاهرام شبين الكوم	8,659,048.0	8,905,384.1	17,587,243.4	23,432,662.9	5,845,4
4	مخزن السعيد	NaN	1,925,098.3	6,821,898.3	8,542,555.0	1,720,65
5	الفاروق	3,569,318.8	2,234,570.5	5,804,709.8	7,653,719.0	1,849,00
6	AUG PHARMA	4,947,294.1	NaN	4,947,294.1	8,448,029.0	3,500,73
7	Wecare	3,178,931.0	1,456,617.0	4,670,226.4	6,317,893.2	1,647,66
8	متنوعون المعادى	2,403,361.0	1,935,660.5	4,506,066.3	4,964,999.4	458,9
9	مخزن داوى	320,621.3	631,990.9	4,339,712.2	5,320,831.5	981,1
10	شركة سوفييكو فارم	2,166,414.9	1,872,924.1	4,078,066.4	5,360,520.5	1,282,4!
11	ميديكال فارما _ اونر نقمي	1,490,353.6	1,741,568.5	3,235,502.7	4,186,592.2	951,08
12	متنوعون الهرم	1,672,338.5	1,245,852.9	2,980,311.5	3,131,901.2	151,58
13	شركة ريو فارما	2,747,267.0	NaN	2,747,267.0	4,055,330.0	1,308,06
14	الفاتح	1,619,077.3	733,594.4	2,394,842.8	2,969,461.9	574,6
15	المتحده للصيادله	1,112,036.1	1,045,723.9	2,158,475.4	2,660,521.6	502,04
16	مخزن جلوبال فارم د/ ابراهيم الروبي	933,332.6	983,267.7	2,016,814.3	2,506,675.8	489,86
17	مخزن السلام السنبلاويين	1,341,134.3	357,744.9	1,698,879.2	2,332,770.1	633,89
18	الفتح لتجارة وتوزيع الادوية	1,147,728.9	440,728.8	1,600,282.7	2,236,649.5	636,36

19	الشافعي فارما للتجارة والتوزيع	876,252.9	603,229.6	1,479,774.5	1,790,721.4	310,94
----	---	-----------	-----------	-------------	-------------	--------

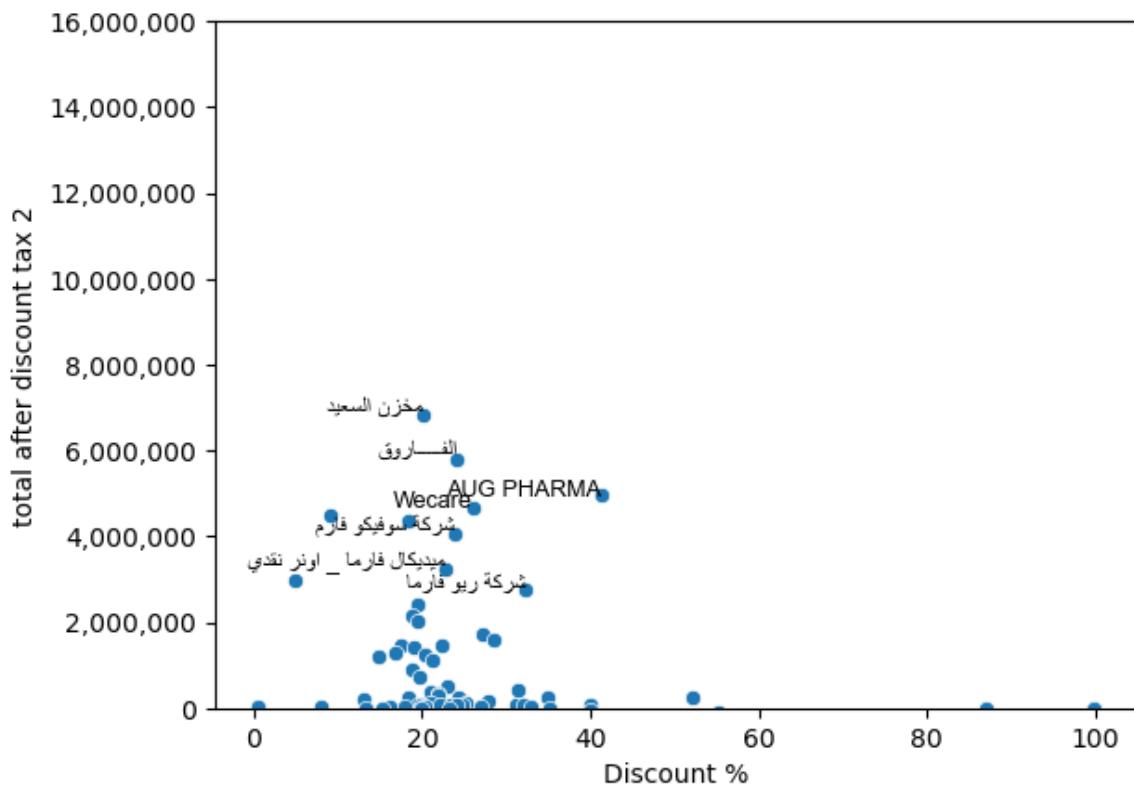
```
In [70]: sns.scatterplot(data= df_supplier_data, y= 'total after discount tax 2'
plt.gca().yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f}'))
plt.ylim(0,40000000)

# Adding text labels for each point
for i, row in df_supplier_data[df_supplier_data['total after discount tax 2'] > 0].iterrows():
    arabic_supplier = reshape_arabic(row['Supplier'])
    plt.text(row['Discount %'], row['total after discount tax 2'],
             fontsize=9, ha='right', fontname='Arial')
```



```
In [71]: sns.scatterplot(data= df_supplier_data, y= 'total after discount tax 2'
plt.gca().yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f}'))
plt.ylim(0,16000000)

# Adding text labels for each point
for i, row in df_supplier_data[(df_supplier_data['total after discount tax 2'] > 0)].iterrows():
    arabic_supplier = reshape_arabic(row['Supplier'])
    plt.text(row['Discount %'], row['total after discount tax 2'],
             fontsize=9, ha='right', fontname='Arial')
```



```
In [72]: product_df_purchased = df_purchased.groupby('Product name').agg({'Q  
        'total after discount tax 2' : 'sum  
        'Total' : 'sum',  
        'total discount 2' : 'sum'}).reset_  
product_df_purchased['Discount %'] = product_df_purchased['total di  
product_df_purchased['Product Rank by Sales'] = product_df_purchase  
product_df_purchased.round().head(15)
```

Out[72]:

		Product name	Quantity	total after discount tax 2	Total	total discount 2	Di
3548		PLAVIX 75MG 28 FILM COATED TAB.	22,436.0	4,776,691.0	5,962,951.0	1,186,260.0	
4456		TRESIBA 100 I.U./ML FLEXTOUCH PRE-FILLED PEN	10,214.0	4,043,689.0	4,794,089.0	750,400.0	
1132		CRESTOR 20MG 28 F.C.TAB.	9,652.0	3,662,441.0	4,970,145.0	1,307,704.0	
2545		LANTUS SOLOSTAR 100 I.U./ML 5 PENS	4,822.0	3,616,924.0	4,220,389.0	603,465.0	
3240		NOVORAPID 100 I.U./ML 5 FLEXPEN	3,431.0	3,132,696.0	3,658,428.0	525,732.0	
1903		FORXIGA 10 MG 28 TABS.	6,306.0	2,539,995.0	3,077,587.0	537,592.0	
1589		ENTRESTO 50 MG (24/26 MG) 28 F.C. TABS.	1,938.0	2,487,919.0	2,954,129.0	466,210.0	
4882		ZURCAL 40 MG 14 GASTRO RESISTANT TAB.	42,982.0	2,255,546.0	4,001,042.0	1,745,496.0	
1064		CONTROLOC 40MG 14 GASTRORESISTANT TAB.	15,630.0	2,036,508.0	2,496,986.0	460,478.0	
3617		PROCORALAN 5MG 28 F.C. TABS.	10,223.0	1,992,002.0	2,718,532.0	726,530.0	
4614		VASTAREL MR 35MG 30 F.C.TAB.	21,335.0	1,978,816.0	2,881,823.0	903,007.0	
3868		RYZODEG 100 I.U./ML FLEXTOUCH PRE-FILLED PEN	4,362.0	1,900,731.0	2,268,196.0	367,465.0	
1129		CRESTOR 10MG28F.C. TAB.	8,226.0	1,863,353.0	2,524,056.0	660,703.0	
1343		DIBAVALLY PLUS 50/1000 MG 28 TABS.	16,899.0	1,667,992.0	2,743,993.0	1,076,001.0	
328		ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	36,370.0	1,644,969.0	2,109,683.0	464,714.0	

```
In [73]: product_df_quantity = df_purchased.groupby('Product name').agg({'Qu
        'total after discount tax 2' : 'sum',
        'Total' : 'sum',
        'total discount 2' : 'sum'}).reset_
product_df_quantity['Discount %'] = product_df_quantity['total disc
product_df_quantity.round().head(15)
product_df_quantity['Product Rank by Sales'] = product_df_quantity[product_df_quantity
```

Out[73]:

	Product name	Quantity	total after discount tax 2	Total	total discount 2	Discount %
4882	ZURCAL 40 MG 14 GASTRO RESISTANT TAB.	42,982.0	2,255,545.9	4,001,042.0	1,745,496.1	43.6
328	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	36,370.3	1,644,969.1	2,109,683.0	464,713.8	22.0
3548	PLAVIX 75MG 28 FILM COATED TAB.	22,436.0	4,776,690.6	5,962,951.0	1,186,260.4	19.9
4614	VASTAREL MR 35MG 30 F.C.TAB.	21,335.0	1,978,816.4	2,881,823.0	903,006.6	31.3
1330	DIAMICRON 60 MG 30 M.R. SCORED TAB.	19,823.0	1,228,034.5	1,648,404.0	420,369.5	25.5
3066	Medicine organizer	-3.0	-280.8	-312.0	-31.2	10.0
2400	JASMOLINE MASSAGE SPRAY 120 ML	-54.0	-3,078.0	-5,130.0	-2,052.0	40.0
4700	VITACID C PLUS 12 TABS	-72.0	-1,569.6	-2,518.5	-948.9	37.7
2392	JANUMET 50/850MG 14 F.C. TAB.	-194.0	-17,675.6	-21,534.0	-3,858.4	17.9
2431	KALOGENA 100 MG 10 SACHETS	-500.0	-45,000.0	-75,000.0	-30,000.0	40.0

5001 rows × 7 columns

In [74]:

```
product_supplier_df = df_purchased.groupby(['Product name', 'Supplier Rank by Sales'])

product_supplier_df['Supplier Rank by Sales'] = product_supplier_df
```

```
product_supplier_df = product_supplier_df.merge(product_df_purchase)

product_supplier_df

# product_supplier_df[product_supplier_df['Supplier Rank']== 1].sort_values(by='total after discount tax 2', ascending=False)
# product_supplier_df[(product_supplier_df['Supplier Rank']== 1) & (product_supplier_df['Product name'] == 'TAMSU')]
```

Out[74] :

	Product name	Supplier	total after discount tax 2	Supplier Rank by Sales	Product Rank by Sales
0	0	ابن سينا فارما	750.0	1.0	2447
1	0	شركة سوفيكوفارم	562.5	3.0	2447
2	0	متنوعون المعادى	712.5	2.0	2447
3	A-VITON 50.000 I.U. 20 CAPS.	متنوعون المعادى	180.1	1.0	4168
4	ABC YEAST 30 TABS	شركة مالتي ستورز فارما	147.6	1.0	4279
...
37929	ماء اكسجين 10%	متنوعون المعادى	30.8	2.0	4520
37930	ماء حقن 5 مل	شركة فارما اوفر سيز	0.0	1.0	4988
37931	مذيب 50 مل	شركة فارما اوفر سيز	0.0	1.0	4988
37932	مصل ثعبان	مخزن داوى	66,500.0	1.0	508
37933	مصل عقرب	مخزن داوى	30,875.0	1.0	802

37934 rows × 5 columns

In [75] :

```
# product_df = round(product_df)
# product_df_quantity = product_df.copy().head(50)

# product_df_purchased = product_df.copy().sort_values(by= 'total after discount tax 2', ascending=False)
# product_df_purchased

product_purchased_quantity_intersection = product_df_quantity.head(50)
product_purchased_quantity_intersection
```

Out[75] :

Product name	Quantity	total after discount tax 2	Total	total discount 2	Disc%

0	ZURCAL 40 MG 14 GASTRO RESISTANT TAB.	42,982.0	2,255,545.9	4,001,042.0	1,745,496.1	4
1	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	36,370.3	1,644,969.1	2,109,683.0	464,713.8	2
2	PLAVIX 75MG 28 FILM COATED TAB.	22,436.0	4,776,690.6	5,962,951.0	1,186,260.4	
3	VASTAREL MR 35MG 30 F.C.TAB.	21,335.0	1,978,816.4	2,881,823.0	903,006.6	
4	DIAMICRON 60 MG 30 M.R. SCORED TAB.	19,823.0	1,228,034.5	1,648,404.0	420,369.5	2
5	MOVINERVE SUB LINGUAL 30 TABLETS.	19,037.0	1,152,235.3	1,770,463.0	618,227.7	3
6	DIBAVALLY PLUS 50/1000 MG 28 TABS.	16,899.0	1,667,992.0	2,743,993.0	1,076,001.0	2
7	CONTROLOC 40MG 14 GASTRORESISTANT TAB.	15,630.0	2,036,507.8	2,496,986.0	460,478.2	
8	MILGA ADVANCE 30 F.C. TAB	15,256.6	1,422,920.1	1,907,545.1	484,625.0	2
9	OMEGA-3 PLUS 30 CAPS	14,513.0	1,391,806.3	1,700,415.0	308,608.7	
10	PEPON PLUS 20 CAPS.	14,475.0	952,438.8	1,329,610.0	377,171.2	2
11	ESMORAP 40MG 14 CAPS.	14,221.0	888,359.3	1,551,736.0	663,376.7	4
12	ATOR 40MG 10 F.C. TAB.	13,950.0	831,280.8	1,152,088.0	320,807.2	
13	TAMSULIN 0.4MG 28 CAPS.	13,519.5	1,085,827.8	1,450,129.5	364,301.7	
14	Laprin 120 gm 10 sachets	12,044.0	1,486,553.4	2,130,030.0	643,476.6	2
15	JANAGLIP PLUS 50/1000MG 28 F.C. TAB.	10,344.0	859,002.5	1,303,344.0	444,341.5	
16	PROCORALAN 5MG 28 F.C. TABS.	10,223.2	1,992,002.4	2,718,532.0	726,529.6	2
	TRESIBA 100 I.U./ML					

17	FLEXTOUCH PRE-FILLED PEN	10,214.0	4,043,688.9	4,794,089.0	750,400.1	:
-----------	--------------------------	----------	-------------	-------------	-----------	---

18	CRESTOR 20MG 28 F.C.TAB.	9,652.0	3,662,440.6	4,970,145.0	1,307,704.4	:
-----------	--------------------------	---------	-------------	-------------	-------------	---

19	EXFORGE HCT 10/160/25MG 14 F.C. TAB.	9,458.0	1,473,003.9	1,756,313.0	283,309.1	:
-----------	--------------------------------------	---------	-------------	-------------	-----------	---

20	FORADIL 12 MCG 30 CAPS.+INHALER	9,106.0	1,448,938.1	1,937,816.0	488,877.9	:
-----------	---------------------------------	---------	-------------	-------------	-----------	---

21	EXFORGE HCT 5/160/12.5MG 14 F.C. TAB.	8,246.5	1,315,481.4	1,584,389.1	268,907.7	:
-----------	---------------------------------------	---------	-------------	-------------	-----------	---

22	CRESTOR 10MG28F.C. TAB.	8,225.5	1,863,352.8	2,524,056.0	660,703.2	:
-----------	-------------------------	---------	-------------	-------------	-----------	---

23	GLIPTUS PLUS 50/1000MG 30 TABLETS	8,094.0	916,231.9	1,313,013.0	396,781.1	:
-----------	-----------------------------------	---------	-----------	-------------	-----------	---

24	EXFORGE 5MG/160MG 14 F.C. TAB.	7,487.0	927,894.5	1,276,036.5	348,142.0	:
-----------	--------------------------------	---------	-----------	-------------	-----------	---

25	THIOTACID COMPOUND 600 MG 30 F.C.CAPLETS	7,119.0	1,055,944.8	1,497,268.5	441,323.7	:
-----------	--	---------	-------------	-------------	-----------	---

```
In [76]: monthly_category = df_purchased.pivot_table(index= 'month', columns=monthly_category
# monthly_category['d'] = monthly_category['total discount 2']/ mon
# # discount = monthly_category[('total discount 2', slice(None))]
# # total_after_discount = monthly_category[('total after discount
# # # Calculate Discount %
# # monthly_category[('Discount %', '')] = (discount / total_after_
```

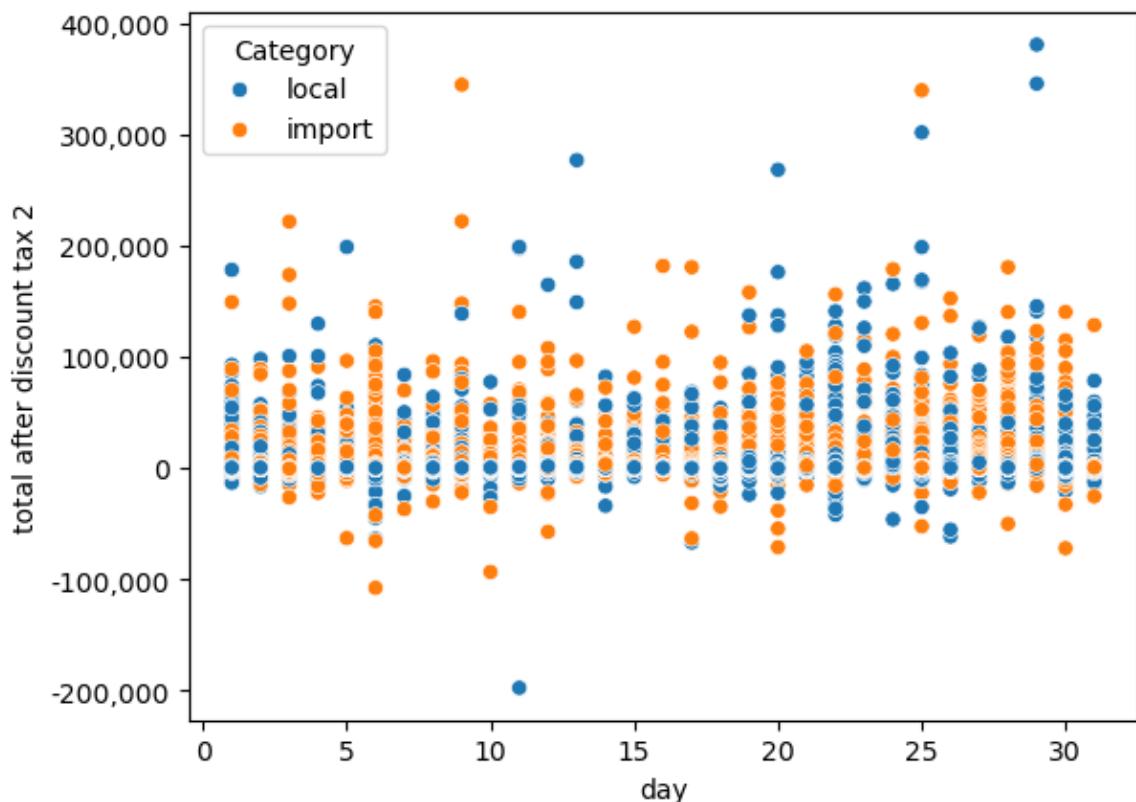
Out[76]:

Category	Chemotherapy	Medical device	TABLETS	Toll	import	import_
month						
1	881,770.0	NaN	NaN	7,000.0	5,906,252.0	
2	697,750.0	NaN	NaN	NaN	4,632,054.0	
3	573,500.0	22,000.0	NaN	NaN	6,646,992.0	
4	262,040.0	164.0	NaN	NaN	5,081,850.0	
5	869,604.0	NaN	NaN	700.0	5,351,139.0	
6	221,760.0	NaN	NaN	31,243.0	3,590,691.0	
7	587,500.0	NaN	NaN	27,683.0	6,112,988.0	
8	381,500.0	NaN	NaN	41,214.0	6,085,229.0	
9	517,500.0	179.0	206.0	30,001.0	6,863,876.0	
10	603,040.0	NaN	1,742.0	17,281.0	9,119,657.0	
11	606,670.0	NaN	NaN	27,292.0	5,932,486.0	
12	1,039,530.0	NaN	NaN	88,361.0	10,964,212.0	

In []:

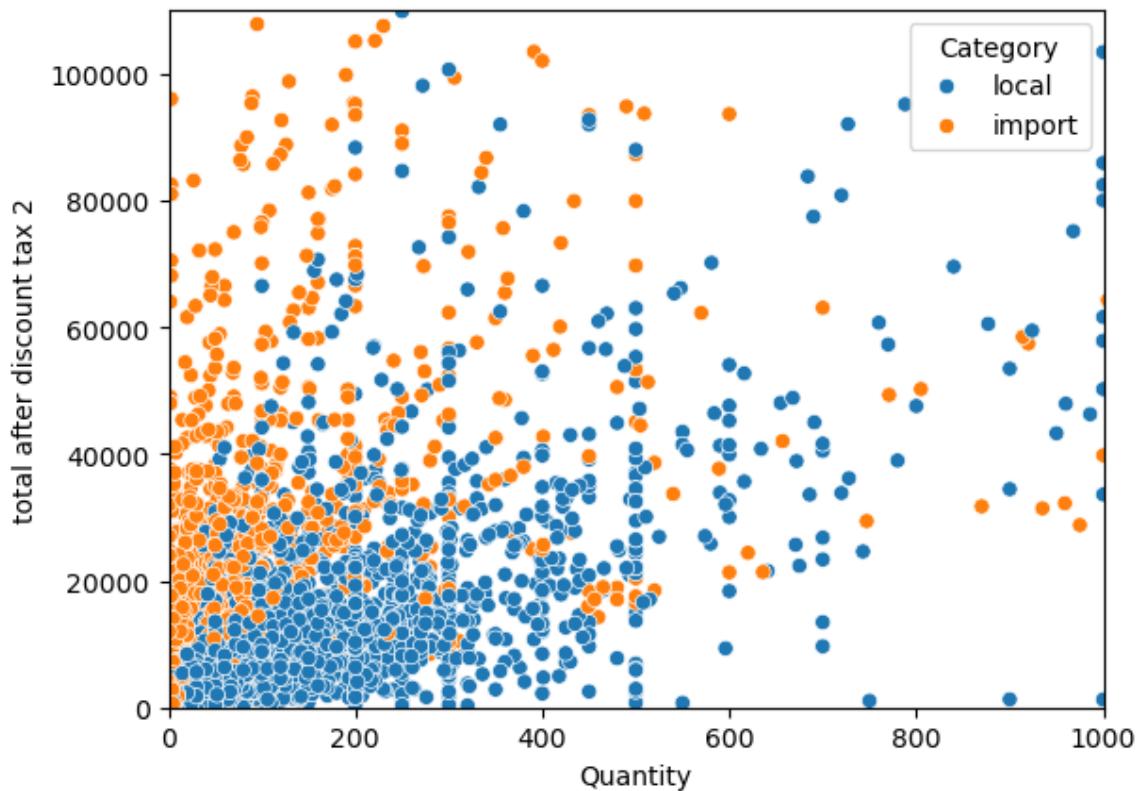
In [77]:

```
sns.scatterplot(data= df_purchased[df_purchased['Category'].isin(['
```



```
In [78]: sns.scatterplot(data= df_purchased[df_purchased['Category'].isin(['local','import'])],  
plt.xlim(0,1000)  
plt.ylim(0,110000)
```

Out[78]: (0.0, 110000.0)

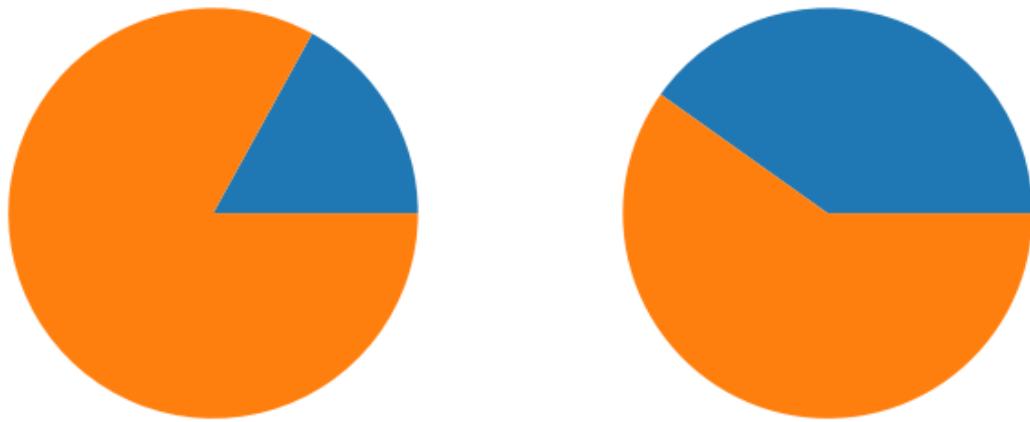


```
In [79]: local_import_df = df_purchased[df_purchased['Category'].isin(['local','import'])]  
local_import_df['AVG Price per Unit'] = local_import_df.apply(lambda x: x['Price per Unit'].mean(), axis=1)  
local_import_df
```

```
fig, ax = plt.subplots(1,2, figsize=(8, 8))

ax[0].pie(data = local_import_df, x = 'Quantity')
ax[1].pie(data = local_import_df, x = 'total after discount tax 2')
```

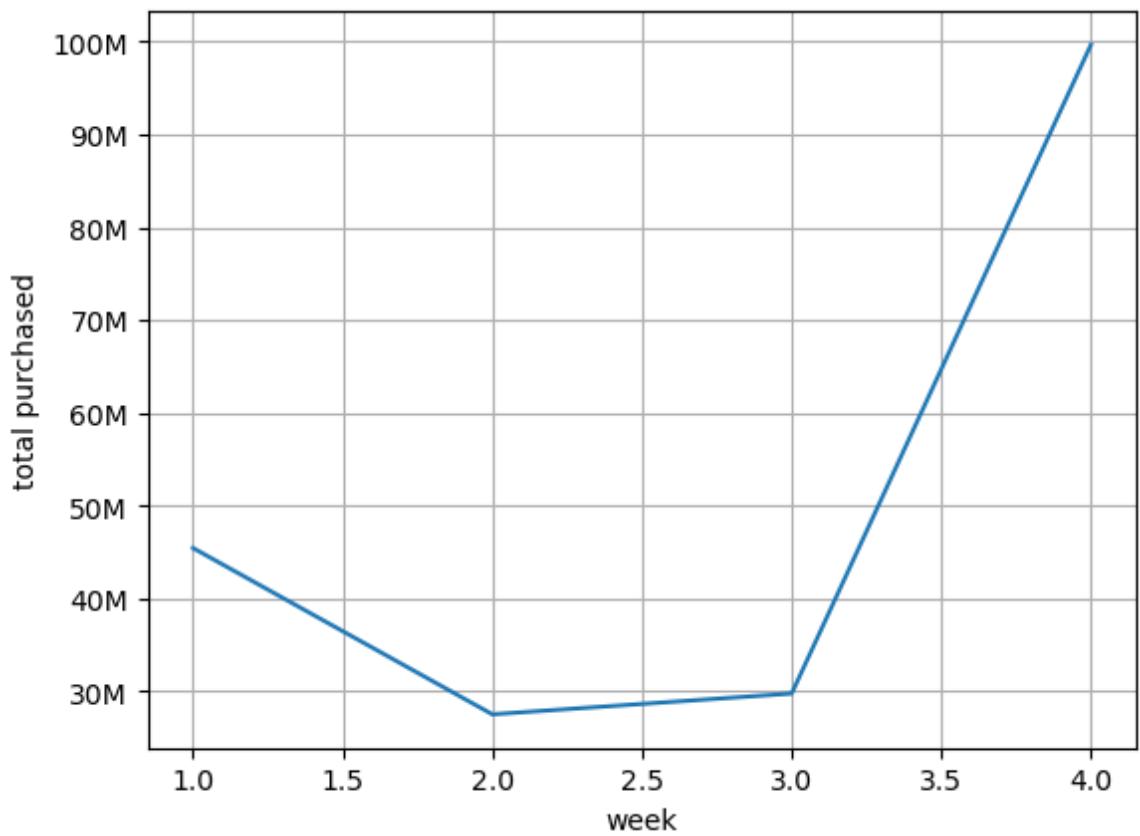
Out[79]: ([<matplotlib.patches.Wedge at 0x13ec34490>,<matplotlib.patches.Wedge at 0x13ebf2410>],[Text(0.3342545422314756, 1.0479856396905574, ''),Text(-0.3342545422314757, -1.0479856396905574, '')])



In [80]: # sns.distplot(data= df_purchased[df_purchased['Category'].isin(['l

In [81]: # insert number of week
df_purchased['week'] = df_purchased.apply(lambda x : 1 if x['day']

In [82]: week_df = df_purchased.groupby('week')['total after discount tax 2']
sns.lineplot(data= week_df, x = 'week', y = 'total purchased')
plt.gca().yaxis.set_major_formatter(mticker.FuncFormatter(lambda x,



```
In [83]: week_df
```

```
Out[83]: week  total purchased
```

0	1	45,447,516.7
1	2	27,536,353.5
2	3	29,776,843.1
3	4	99,663,952.7

```
In [84]: df_purchased[df_purchased['Bonus QTY'] > 0][['Discount %', 'new dis
```

Out[84]:

	Discount %	new discount
340	20.0	27.3
853	25.0	29.1
857	20.0	32.8
864	25.0	31.8
873	20.0	27.0
...
197645	25.0	55.0
198127	25.0	31.8
198136	20.0	21.5
198164	25.0	29.7
198227	25.0	40.0

1388 rows × 2 columns

Top 5 Suppliers with low return % & sales above 1 M

In [85]:

```
low_return_suppliers = df_supplier_data[df_supplier_data['total aft
low_return_suppliers['Supplier Rank by Sales'] = low_return_supplie
low_return_suppliers
```

Out[85]:

	Supplier	local	import	total after discount tax 2	Total	total discount 2
3	شركة الاهرام شبين الكوم	8,659,048.0	8,905,384.1	17,587,243.4	23,432,662.9	5,845,419.5
23	رامكو فارم لتجارة الادوية	712,518.9	496,290.8	1,225,582.0	1,537,277.6	311,695.7
24	Imported (Maadi)	11,811.7	602,969.1	1,206,899.3	1,415,942.0	209,042.7
13	شركة ريو فارما	2,747,267.0	Nan	2,747,267.0	4,055,330.0	1,308,063.0
25	Samir Supplier	251,425.8	39,250.0	1,131,363.4	1,437,027.0	305,663.6

Worst 13 Suppliers with low return % & sales above 1 M

In [86]: df_supplier_data[df_supplier_data['total after discount tax 2'] >=

Out [86]:

	Supplier	local	import	total after discount tax 2	Total	t discou
12	متنوعون الهرم	1,672,338.5	1,245,852.9	2,980,311.5	3,131,901.2	151,5%
1	شركة فارما اوفر سيز	25,401,736.6	16,926,580.8	43,213,845.1	57,262,833.3	14,048,9%
6	AUG PHARMA	4,947,294.1	NaN	4,947,294.1	8,448,029.0	3,500,7%
8	متنوعون المعادى	2,403,361.0	1,935,660.5	4,506,066.3	4,964,999.4	458,9%
2	شركة جلوبال فارما	11,682,477.7	7,648,712.0	19,423,966.0	25,645,432.6	6,221,4%
5	الفاروق	3,569,318.8	2,234,570.5	5,804,709.8	7,653,719.0	1,849,0%
19	الشافعى فارما للتجارة والتوزيع	876,252.9	603,229.6	1,479,774.5	1,790,721.4	310,9%
11	ميديكال فارما _ اونر نقدي	1,490,353.6	1,741,568.5	3,235,502.7	4,186,592.2	951,0%
22	شركة مالقى ستورز فارما	644,939.1	603,085.1	1,294,501.4	1,554,977.3	260,4%
10	شركة سوفيكو فارم	2,166,414.9	1,872,924.1	4,078,066.4	5,360,520.5	1,282,4%
17	مخزن السلام السنبلاويين	1,341,134.3	357,744.9	1,698,879.2	2,332,770.1	633,8%
16	مخزن جلوبال فارم د/ ابراهيم الروبي	933,332.6	983,267.7	2,016,814.3	2,506,675.8	489,8%
0	ابن سينا فارما	31,146,352.4	20,826,212.9	53,135,119.5	70,413,196.8	17,278,0%

```
In [87]: supplier_product_df = df_purchased.groupby(['Supplier','Product nam  
        'total after discount tax 2' : 'sum'  
        'Total' : 'sum',  
        'total discount 2': 'sum',  
        'potential total discount': 'sum',  
        'potential value': 'sum'  
    }).reset_index().sort_values(by = '  
supplier_product_df['Discount %'] = supplier_product_df['total disc  
  
supplier_product_df = supplier_product_df.merge(df_supplier_data[['  
# supplier_product_df.drop(['total public without tax_y'], axis=1,  
supplier_product_df = supplier_product_df.merge(product_df_purchase  
supplier_product_df
```

Out[87]:

	Supplier	Product name	Quantity	total after discount tax 2	Total	total discount 2
0	AUG PHARMA	ZURCAL 40 MG 14 GASTRO RESISTANT TAB.	41,322.0	2,186,333.5	3,879,972.0	1,693,638.5
1	ابن سينا فارما	TRESIBA 100 I.U./ML FLEXTOUCH PRE-FILLED PEN	5,856.0	2,098,395.4	2,525,276.0	426,880.6
2	ابن سينا فارما	CRESTOR 20MG 28 F.C.TAB.	5,080.0	1,889,180.1	2,575,315.0	686,134.9
3	ابن سينا فارما	NOVORAPID 100 I.U./ML 5 FLEXPEN	1,970.0	1,657,809.9	1,951,587.0	293,777.1
4	شركة الاهرام شبيبن الكوم	PLAVIX 75MG 28 FILM COATED TAB.	7,539.0	1,480,685.9	1,864,329.0	383,643.1
...
37929	متنوعون الهرم	FORXIGA 10 MG 28 TABS.	15.0	-13,398.4	-12,806.0	592.4
37930	مخزن داوى	AFINITOR 10MG 30 TABS	-1.0	-15,000.0	-19,100.0	-4,100.0
37931	شركة سوفيكو فارم	JANUMET 50/850MG 14 F.C. TAB.	-194.0	-17,675.6	-21,534.0	-3,858.4
37932	ديفارت	KALOGENA 100 MG 10 SACHETS	-500.0	-45,000.0	-75,000.0	-30,000.0
37933	القصواع	DIBAVALLY PLUS 50/1000 MG 28 TABS.	-1,008.0	-63,504.0	-141,120.0	-77,616.0

37934 rows × 11 columns

In [88]:

```
df_return_filtered = df[df['Type'] == 'Return']
return_products = df_return_filtered.groupby(['Supplier', 'Product n
return_products['total after discount tax 2'] = return_products['to
return_products.rename(columns={'total after discount tax 2' : 'Tot
```

```
return_products = return_products.merge(df_supplier_data[['Supplier']]
```

Out[88]:

	Supplier	Product name	Total Return	Supplier Rank by Sales
0	el farouk	COVERAM 5/10MG 15 TABS.	80.0	63
1	el farouk	COVERAM 5/5MG 15 TABS.	80.0	63
2	el farouk	DIGLIFLOZ 5 MG 30 F.C. TABS.	993.6	63
3	el farouk	DOWNOPRAZOL 20/1680MG PD. FOR ORAL SUSP. 7 SAC...	89.6	63
4	el farouk	ERASTAPEX CO 5/20MG 30 F.C. TABS.	165.6	63
...	
4818	نالی فارم انترناشیونال	VIVIDOL HAIR OIL 250 ML	166.5	56
4819	های ستورز فارما	ARTHROFAST 150 MG 14 M.R. TABLETS	874.8	51
4820	های ستورز فارما	DRAMENEX 50MG 20 TABS	23.0	51
4821	های ستورز فارما	KANSARTAN 300MG 30 TAB.	77.7	51
4822	های ستورز فارما	VESICARE 5 MG 30 F.C. TAB	265.0	51

4823 rows × 4 columns

Analysis for top 10 ranked Suppliers

In [89]: `product_df_quantity.rename(columns={'total after discount tax 2': '`

In [90]: `product_df_quantity.head(5)`

Out [90]:

		Product name	Quantity	total after discount	Total	total discount 2	Discount %
		ZURCAL 40 MG 14					
4882		GASTRO RESISTANT TAB.	42,982.0	2,255,545.9	4,001,042.0	1,745,496.1	43.6
328		ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	36,370.3	1,644,969.1	2,109,683.0	464,713.8	22.0
3548		PLAVIX 75MG 28 FILM COATED TAB.	22,436.0	4,776,690.6	5,962,951.0	1,186,260.4	19.9
4614		VASTAREL MR 35MG 30 F.C.TAB.	21,335.0	1,978,816.4	2,881,823.0	903,006.6	31.3
1330		DIAMICRON 60 MG 30 M.R. SCORED TAB.	19,823.0	1,228,034.5	1,648,404.0	420,369.5	25.5

In [91]:

supplier_product_df

Out[91]:

	Supplier	Product name	Quantity	total after discount tax 2	Total	total discount %
0	AUG PHARMA	ZURCAL 40 MG 14 GASTRO RESISTANT TAB.	41,322.0	2,186,333.5	3,879,972.0	1,693,638.5
1	ابن سينا فارما	TRESIBA 100 I.U./ML FLEXTOUCH PRE-FILLED PEN	5,856.0	2,098,395.4	2,525,276.0	426,880.6
2	ابن سينا فارما	CRESTOR 20MG 28 F.C.TAB.	5,080.0	1,889,180.1	2,575,315.0	686,134.9
3	ابن سينا فارما	NOVORAPID 100 I.U./ML 5 FLEXPEN	1,970.0	1,657,809.9	1,951,587.0	293,777.1
4	شركة الاهرام شبيبن الكوم	PLAVIX 75MG 28 FILM COATED TAB.	7,539.0	1,480,685.9	1,864,329.0	383,643.1
...
37929	متنوعون الهرم	FORXIGA 10 MG 28 TABS.	15.0	-13,398.4	-12,806.0	592.4
37930	مخزن داوى	AFINITOR 10MG 30 TABS	-1.0	-15,000.0	-19,100.0	-4,100.0
37931	شركة سوفيكيو فارم	JANUMET 50/850MG 14 F.C. TAB.	-194.0	-17,675.6	-21,534.0	-3,858.4
37932	ديفارت	KALOGENA 100 MG 10 SACHETS	-500.0	-45,000.0	-75,000.0	-30,000.0
37933	القصواع	DIBAVALLY PLUS 50/1000 MG 28 TABS.	-1,008.0	-63,504.0	-141,120.0	-77,616.0

37934 rows × 11 columns

In [92]:

```
supplier_product_df = supplier_product_df.merge(return_products[['Supplier ID', 'Return %']])
supplier_product_df = supplier_product_df.merge(product_df_quantity)

supplier_product_df['Return %'] = supplier_product_df['Total Return'] * 100 / supplier_product_df['Total Quantity']
```

In [93]: supplier_product_df

Out[93]:

	Supplier	Product name	Quantity	total after discount tax 2	Total	total discount %
0	AUG PHARMA	ZURCAL 40 MG 14 GASTRO RESISTANT TAB.	41,322.0	2,186,333.5	3,879,972.0	1,693,638.5
1	ابن سينا فارما	TRESIBA 100 I.U./ML FLEXTOUCH PRE-FILLED PEN	5,856.0	2,098,395.4	2,525,276.0	426,880.6
2	ابن سينا فارما	CRESTOR 20MG 28 F.C.TAB.	5,080.0	1,889,180.1	2,575,315.0	686,134.9
3	ابن سينا فارما	NOVORAPID 100 I.U./ML 5 FLEXPEN	1,970.0	1,657,809.9	1,951,587.0	293,777.1
4	شركة الاهرام شبين الكوم	PLAVIX 75MG 28 FILM COATED TAB.	7,539.0	1,480,685.9	1,864,329.0	383,643.1
...						
37929	متنوعون الهرم	FORXIGA 10 MG 28 TABS.	15.0	-13,398.4	-12,806.0	592.4
37930	مخزن داوى	AFINITOR 10MG 30 TABS	-1.0	-15,000.0	-19,100.0	-4,100.0
37931	شركة سوفيكو فارم	JANUMET 50/850MG 14 F.C. TAB.	-194.0	-17,675.6	-21,534.0	-3,858.4
37932	ديفارت	KALOGENA 100 MG 10 SACHETS	-500.0	-45,000.0	-75,000.0	-30,000.0
37933	القصـوة	DIBAVALLY PLUS 50/1000 MG 28 TABS.	-1,008.0	-63,504.0	-141,120.0	-77,616.0

37934 rows × 14 columns

In [94]: rank_supplier = {}

```

for i in range(1,11):
    rank_supplier[i] = supplier_product_df[supplier_product_df['SupplierID'] == i]
    rank_supplier[i]['% from total product Sales'] = rank_supplier[i].sum()['Sales'] / total_sales * 100
# Example: Display the DataFrame for rank 1
display(rank_supplier[6])

```

		Supplier	Product name	Quantity	total after discount tax 2	Total	total discount 2	p d
18	الفاروق		PLAVIX 75MG 28 FILM COATED TAB.	4,301.0	831,407.7	1,067,911.0	236,503.3	30
71	الفاروق		OMEGA-3 PLUS 30 CAPS	3,605.0	332,769.9	387,495.0	54,725.1	11
90	الفاروق		EXFORGE HCT 10/160/25MG 14 F.C. TAB.	1,864.0	282,700.2	314,520.0	31,819.8	7
113	الفاروق		MILGA 40 F.C. TAB	4,779.0	248,712.2	341,012.0	92,299.8	1
124	الفاروق		MILGA ADVANCE 30 F.C. TAB	2,766.0	228,355.0	308,920.5	80,565.5	6
...
37815	الفاروق		EXAMIDE 10MG 30 TAB	0.0	0.0	0.0	0.0	0.0
37820	الفاروق		ZURCAL 40 MG 28 GASTRO RESISTANT TAB.	0.0	0.0	0.0	0.0	0.0
37821	الفاروق		VOLTAREN 1% EMULGEL 25 GM	0.0	0.0	0.0	0.0	0.0
37824	الفاروق		VIBRAMYCIN 100MG 10 CAPS.	0.0	0.0	0.0	0.0	0.0
37825	الفاروق		VENTAL COMPOSITUM 200 DOSES SPRAY	0.0	0.0	0.0	0.0	0.0

454 rows × 15 columns

Rank 1

In [95]: `display(rank_supplier[1])`

		Supplier	Product name	Quantity	total after discount tax 2	Total	total discount 2
1	ابن سينا فارما	TRESIBA 100 I.U./ML FLEXTOUCH PRE-FILLED PEN	5,856.0	2,098,395.4	2,525,276.0	426,880.6	
2	ابن سينا فارما	CRESTOR 20MG 28 F.C.TAB.	5,080.0	1,889,180.1	2,575,315.0	686,134.9	
3	ابن سينا فارما	NOVORAPID 100 I.U./ML 5 FLEXPEN	1,970.0	1,657,809.9	1,951,587.0	293,777.1	
8	ابن سينا فارما	CRESTOR 10MG28F.C. TAB.	5,529.0	1,196,774.9	1,635,024.0	438,249.1	
9	ابن سينا فارما	LANTUS SOLOSTAR 100 I.U./ML 5 PENS	1,555.0	1,158,352.8	1,352,850.0	194,497.2	
...
37836	ابن سينا فارما	ATCOEPRILAX 100 MG 20 F.C.TABS	0.0	-2.6	-6.0	-3.4	
37841	ابن سينا فارما	SORAL 20 MG 20 CAPS.	-4.0	-18.8	-32.5	-13.7	
37853	ابن سينا فارما	PARACETAMOL CID 20 TABS	9.0	-53.0	72.0	125.0	
37906	ابن سينا فارما	GLIPTAMET 50/1000MG 56 F.C.TAB.	-5.0	-527.6	-787.5	-259.9	
37908	ابن سينا فارما	ENERA SHOTS 10 DRINKABLE AMPOULES * 25 ML.	-2.0	-677.9	-720.0	-42.1	

3204 rows × 15 columns

In [96]: `df_purchased[df_purchased['Product name'] == 'TRESIBA 100 I.U./ML F']`

Out [96]: 4043688.9187894217

In [97]: df_purchased = df_purchased.merge(df_supplier_data[['Supplier', 'Supplier Name', 'Address', 'Phone', 'Email', 'Website', 'City', 'State', 'Country', 'Postal Code', 'Latitude', 'Longitude']], left_on='Supplier ID', right_index=True)

Out[97]:

	Date	Bill No	Store ID	Type	Store Name	Product id	Prod
0	2024-01-01	2023108920907	4	Purchase	وياك صيدلية الهرم	3083	EXFC 5MG/16 14 F.C.
1	2024-01-01	2023108920907	4	Purchase	وياك صيدلية الهرم	10393	GLAP1 F 50/100 30 F.C.
2	2024-01-01	2023108920907	4	Purchase	وياك صيدلية الهرم	3080	EXFC 10MG/16 14 F.C.
3	2024-01-01	10253	4	Purchase	وياك صيدلية الهرم	11444	ERALONE M F.C.
4	2024-01-01	10253	4	Purchase	وياك صيدلية الهرم	8683	PANTC 20MG 14
...							
195658	2024-12-31	30575	1	Purchase	وياك المعادى	1082	NIZA 20MG SHAMPOO
195659	2024-12-31	30575	1	Purchase	وياك المعادى	8784	OXALEP 300 M F.C. T
195660	2024-12-31	30575	1	Purchase	وياك المعادى	2592	MICONAZ ORAL GE
195661	2024-12-31	30575	1	Purchase	وياك المعادى	23805	IMUTRE 5 M T
195662	2024-12-31	30575	1	Purchase	وياك المعادى	9926	GLUCOPH XR 100 30 T

195663 rows × 40 columns

In [98]: import matplotlib.ticker as mticker

```
supplier_monthly_data = df_purchased[df_purchased['Supplier Rank by Product name' : 'nunique', 'Total' : 'sum', 'total discount 2' : 'sum']].reset_index().groupby('Supplier Rank by Product name').sum().sort_values('Total', ascending=False).head(10)
```

```

supplier_monthly_data['Discount %'] = supplier_monthly_data['total']

print(supplier_monthly_data)
print("-----")
print("-----")

supplier_category_data = df_purchased[df_purchased['Supplier Rank b
-----


print(supplier_category_data)

fig, ax = plt.subplots(3,1, figsize=(8, 8))

sns.lineplot(data = supplier_monthly_data, x = 'month', y = 'total'
sns.lineplot(data = supplier_monthly_data, x = 'month', y = 'Produc
# sns.lineplot(data = supplier_category_data[['local', 'import']],x
supplier_category_data[['local', 'import']].plot(kind = 'line', ax

ax[0].set_title('Total Purchased per Month')
ax[0].set_xlabel('')
ax[0].set_ylabel('')

ax[1].set_title('Products Number per Month')
ax[1].set_xlabel('')
ax[1].set_ylabel('')

ax[2].set_title('Local & import Purchased per Month')
ax[2].set_xlabel('')
ax[2].set_ylabel('')

# avoid scientific notation
ax[0].yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f
ax[1].yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f
ax[2].yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f

# plt.gca().yaxis.set_major_formatter(mticker.StrMethodFormatter('{

plt.tight_layout()
plt.show()

```

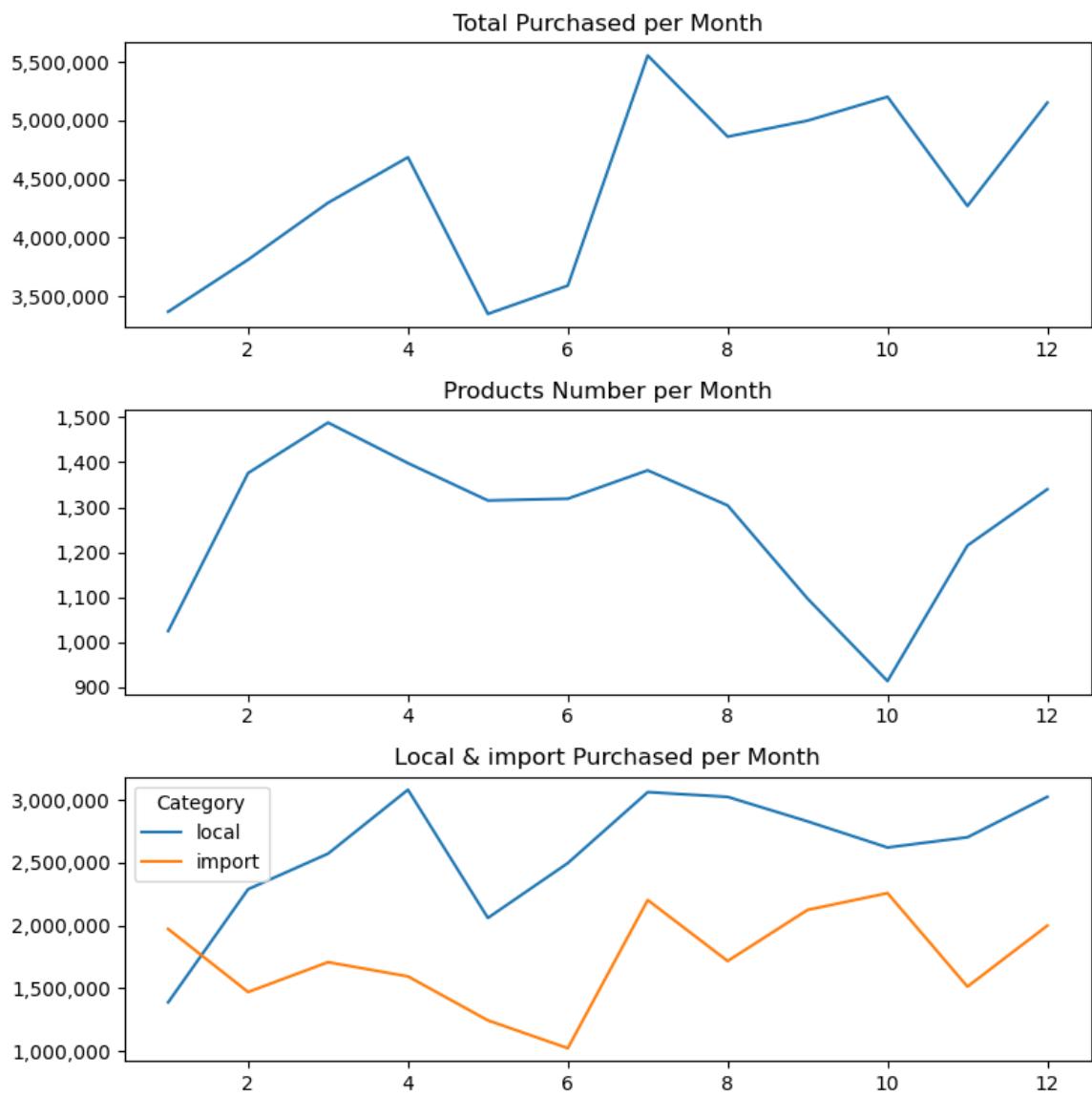
	month	total	after discount	tax	2	Product	name	Total	\
0	1		3,366,052.7			1025	4,439,224.4		
1	2		3,811,029.1			1376	5,056,464.5		
2	3		4,296,776.0			1488	5,700,133.0		
3	4		4,685,685.7			1398	6,279,933.2		
4	5		3,347,409.9			1315	4,504,164.7		
5	6		3,588,883.7			1319	4,805,041.7		
6	7		5,555,377.9			1382	7,319,930.9		
7	8		4,861,731.4			1304	6,442,565.7		
8	9		4,998,655.4			1097	6,605,495.4		
9	10		5,202,320.4			914	6,747,123.5		

10	11	4,269,114.6	1215	5,706,321.2
11	12	5,152,082.5	1340	6,806,798.7

	total discount	2	Discount %
0	1,073,171.7		24.2
1	1,245,435.4		24.6
2	1,403,356.9		24.6
3	1,594,247.6		25.4
4	1,156,754.8		25.7
5	1,216,158.0		25.3
6	1,764,553.0		24.1
7	1,580,834.3		24.5
8	1,606,839.9		24.3
9	1,544,803.1		22.9
10	1,437,206.6		25.2
11	1,654,716.1		24.3

Category	Chemotherapy	Toll	import	import_re_registration
\month				
1	NaN	NaN	1,971,956.7	5,940.5
2	NaN	NaN	1,470,496.6	52,033.2
3	NaN	NaN	1,707,740.6	17,343.7
4	NaN	NaN	1,594,736.2	10,426.2
5	NaN	NaN	1,244,569.7	42,764.4
6	NaN	11,933.4	1,023,351.2	56,978.0
7	NaN	2,406.7	2,202,510.3	289,006.8
8	NaN	1,022.8	1,717,057.0	120,028.2
9	NaN	2,077.8	2,124,072.8	43,973.8
10	NaN	1,095.8	2,257,492.3	322,835.6
11	NaN	2,382.0	1,513,277.4	51,434.5
12	18,781.3	3,823.5	1,998,952.1	106,266.1

Category	local
\month	
1	1,388,155.5
2	2,288,499.2
3	2,571,691.8
4	3,080,523.3
5	2,060,075.8
6	2,496,621.0
7	3,061,454.2
8	3,023,623.5
9	2,828,531.1
10	2,620,896.8
11	2,702,020.7
12	3,024,259.5



```
In [99]: import matplotlib.ticker as mticker

supplier_monthly_data = df_purchased[df_purchased['Supplier Rank by
                                         'Product name' : 'nunique',
                                         'Total' : 'sum',
                                         'total discount 2': 'sum']].r
supplier_monthly_data['Discount %'] = supplier_monthly_data['total

print(supplier_monthly_data)
print("-----")
print("-----")

supplier_category_data = df_purchased[df_purchased['Supplier Rank b

print(supplier_category_data)

fig, ax = plt.subplots(3,1, figsize=(8, 8))
```

```

sns.lineplot(data = supplier_monthly_data, x = 'month', y = 'total')
sns.lineplot(data = supplier_monthly_data, x = 'month', y = 'Product')
# sns.lineplot(data = supplier_category_data[['local', 'import']], x = 'month', y = 'Total')
supplier_category_data[['local', 'import']].plot(kind = 'line', ax = [ax[1], ax[2]])

ax[0].set_title('Total Purchased per Month')
ax[0].set_xlabel('')
ax[0].set_ylabel('')

ax[1].set_title('Products Number per Month')
ax[1].set_xlabel('')
ax[1].set_ylabel('')

ax[2].set_title('Local & import Purchased per Month')
ax[2].set_xlabel('')
ax[2].set_ylabel('')

# avoid scientific notation
ax[0].yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f}'))
ax[1].yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f}'))
ax[2].yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f}'))

# plt.gca().yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f}'))

plt.tight_layout()
plt.show()

```

	month	total	after discount	tax	2	Product name	Total	\
0	1		1,349,890.7			816	1,850,527.2	
1	2		1,528,132.2			1272	2,037,819.9	
2	3		1,224,934.7			1066	1,612,620.9	
3	4		1,383,562.7			721	1,866,391.9	
4	5		2,783,159.6			1431	3,735,231.6	
5	6		1,894,914.9			945	2,516,082.8	
6	7		3,699,753.8			1270	4,933,200.3	
7	8		1,957,012.8			689	2,544,819.0	
8	9		5,552,087.5			1432	7,429,594.1	
9	10		7,040,928.8			1416	9,228,197.8	
10	11		4,687,726.6			1626	6,136,363.5	
11	12		10,111,740.8			1560	13,371,984.3	

	total	discount	2	Discount %
0	500,636.5		27.1	
1	509,687.7		25.0	
2	387,686.1		24.0	
3	482,829.3		25.9	
4	952,072.0		25.5	
5	621,167.8		24.7	
6	1,233,446.5		25.0	
7	587,806.2		23.1	
8	1,877,506.6		25.3	
9	2,187,269.0		23.7	
10	1,448,636.9		23.6	
11	3,260,243.6		24.4	

Category local \ month	TABLETS	Toll	import	import_re_registration		
1 7,685.7	NaN	NaN	388,086.7		54,118.3	90
2 3,046.9	NaN	NaN	586,685.3		48,400.0	89
3 9,613.2	NaN	NaN	499,368.0		15,953.6	70
4 9,995.2	NaN	NaN	421,876.9		1,690.5	95
5 3,537.2	NaN	NaN	895,692.4		13,930.0	1,87
6 4,469.2	NaN	NaN	684,644.8		5,801.0	1,20
7 9,758.0	NaN	4,175.5	1,332,103.4		103,716.9	2,25
8 0,811.0	NaN	4,390.1	951,811.7		NaN	1,00
9 2,046.7	96.8	4,268.2	2,055,137.1		110,538.7	3,38
10 7,580.4	1,741.5	3,324.7	2,925,427.9		122,854.3	3,98
11 4,881.3	NaN	3,962.6	2,183,523.3		85,359.4	2,41
12 8,311.9	NaN	65,957.4	4,002,223.4		228,673.0	5,80

Category month	medical supply
1	NaN
2	NaN
3	NaN
4	NaN
5	NaN
6	NaN
7	NaN
8	NaN
9	NaN
10	NaN
11	NaN
12	6,575.1



```
In [100]: supplier_category_data
```

Out [100...]

Category	TABLETS	Toll	import	import_re_registration	local
month					
1	NaN	NaN	388,086.7	54,118.3	907,685.1
2	NaN	NaN	586,685.3	48,400.0	893,046.9
3	NaN	NaN	499,368.0	15,953.6	709,613.1
4	NaN	NaN	421,876.9	1,690.5	959,995.1
5	NaN	NaN	895,692.4	13,930.0	1,873,537.1
6	NaN	NaN	684,644.8	5,801.0	1,204,469.1
7	NaN	4,175.5	1,332,103.4	103,716.9	2,259,758.1
8	NaN	4,390.1	951,811.7	NaN	1,000,811.1
9	96.8	4,268.2	2,055,137.1	110,538.7	3,382,046.1
10	1,741.5	3,324.7	2,925,427.9	122,854.3	3,987,580.1
11	NaN	3,962.6	2,183,523.3	85,359.4	2,414,881.1
12	NaN	65,957.4	4,002,223.4	228,673.0	5,808,311.1

In [101...]

```
top_suppliers_sales = df_supplier.loc[:10,'Supplier']
# top_suppliers_sales = top_suppliers_sales.to_list()

new_df = df_purchased[df_purchased['Supplier'].isin(top_suppliers_sales)]
# new_df['Supplier 2'] = new_df['Supplier'].apply(reshape_arabic)

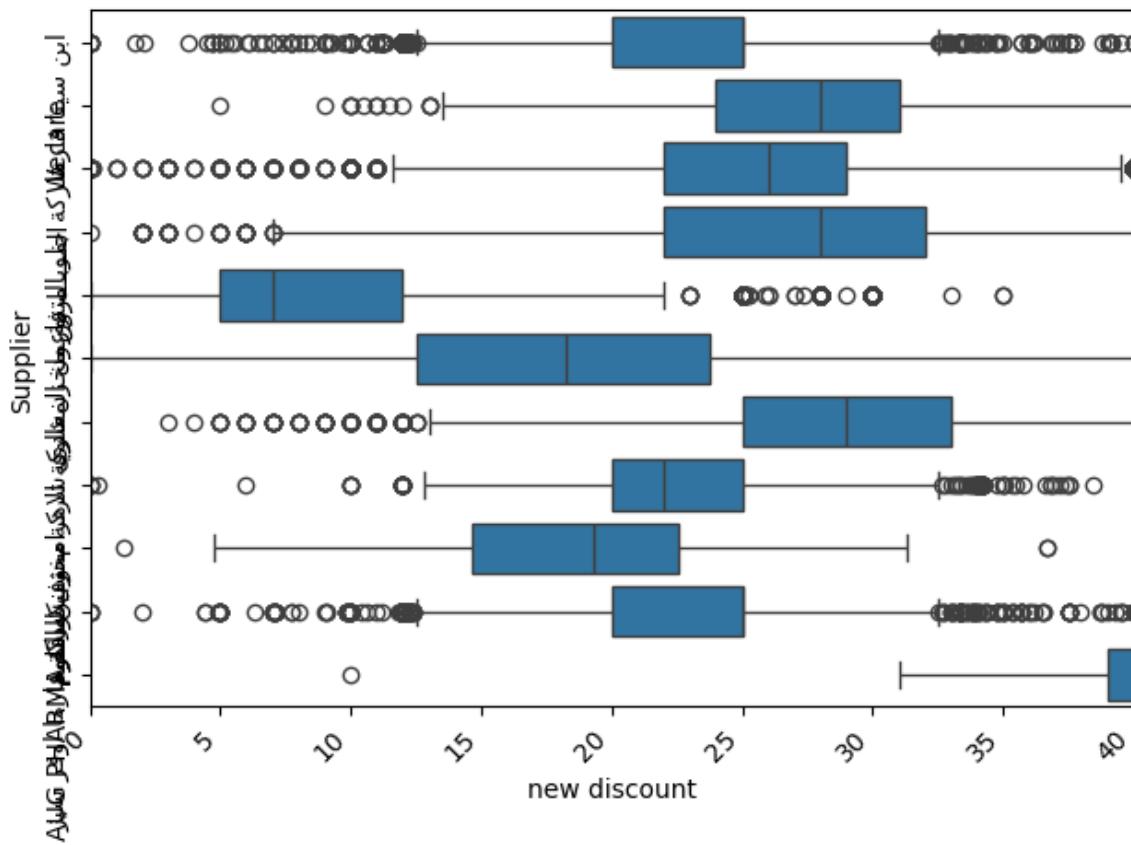
sns.boxplot(data = new_df, y = new_df['Supplier'], x = 'new discount')

ax = plt.gca() # Get the current axis
tick_labels = ax.get_yticklabels() # Retrieve current tick labels
formatted_labels = [reshape_arabic(label.get_text()) for label in tick_labels]
plt.yticks(ticks=ax.get_yticks(), labels=formatted_labels)

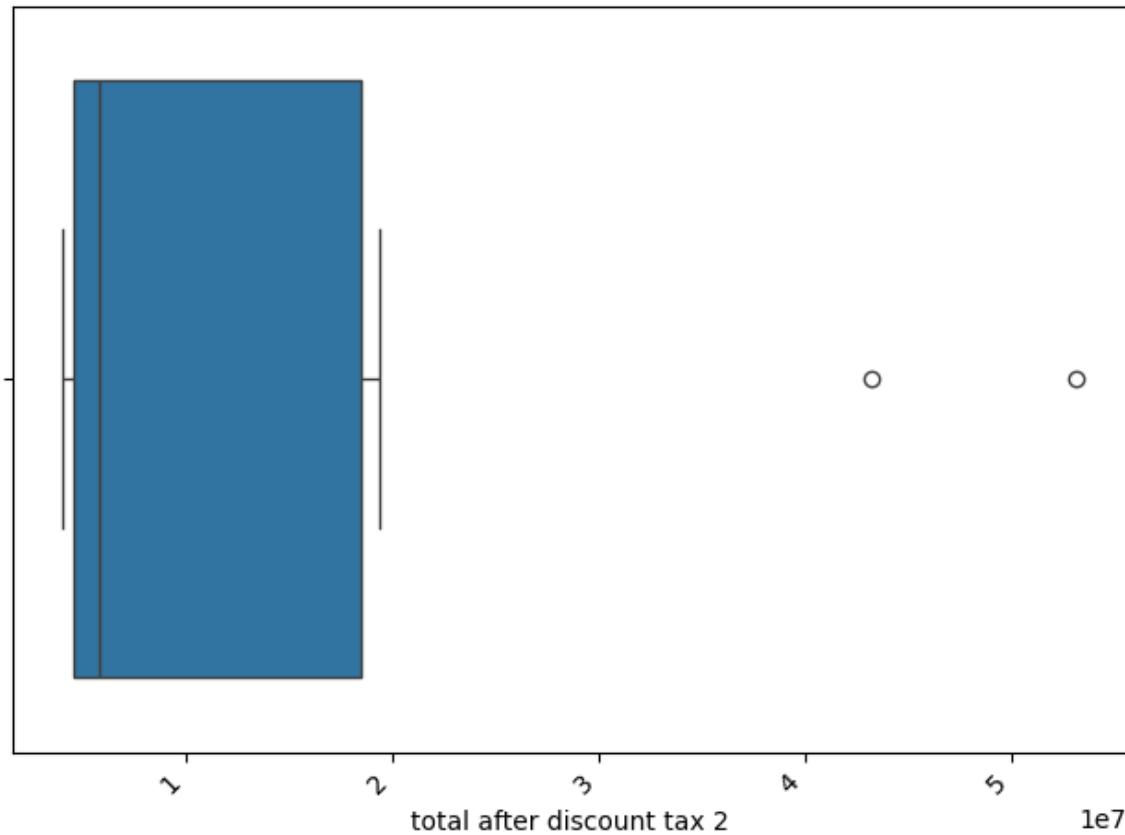
plt.xticks(rotation=45, ha='right')
plt.yticks(rotation=90, ha='right')

plt.xlim(0,40)

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



```
In [102]:  
top_suppliers_sales = df_supplier.loc[:10,'Supplier']  
# top_suppliers_sales = top_suppliers_sales.to_list()  
  
new_df = df_purchased[df_purchased['Supplier'].isin(top_suppliers_sales)]  
# new_df['Supplier 2'] = new_df['Supplier'].apply(reshape_arabic)  
  
sns.boxplot(data = new_df.groupby('Supplier')['total after discount'])  
  
ax = plt.gca() # Get the current axis  
tick_labels = ax.get_yticklabels() # Retrieve current tick labels  
formatted_labels = [reshape_arabic(label.get_text()) for label in tick_labels]  
  
plt.yticks(ticks=ax.get_yticks(), labels=formatted_labels)  
  
plt.xticks(rotation=45, ha='right')  
plt.yticks(rotation=90, ha='right')  
  
# plt.xlim(38275000,38300000)  
plt.tight_layout()  
plt.show()
```



```
In [103...]: # rank_1_supplier[rank_1_supplier['Product Rank by Sales'] == 4446]
# rank_1_supplier_return[rank_1_supplier_return['Product name'] ==
# rank_1_supplier = supplier_product_df[supplier_product_df['Suppli
# rank_1_supplier[rank_1_supplier['Product name'] == 'DELTARHINO NA
# rank_1_supplier_return = return_products[return_products['Supplie
# rank_1_supplier_return[rank_1_supplier_return['Product name'] ==
# df[(df['Product name'] == 'DELTARHINO NASAL SPRAY 15 ML') & (df['
```

Top 15 Products by Sales

```
In [104...]: rank_supplier[1].head(15)
```

```
Out[104...]:
```

	Supplier	Product name	Quantity	total after discount tax 2	Total	total discount 2	per unit
1	ابن سينا فارما	TRESIBA 100 I.U./ML FLEXTOUCH PRE-FILLED PEN	5,856.0	2,098,395.4	2,525,276.0	426,880.6	5
2	ابن سينا فارما	CRESTOR 20MG 28	5,080.0	1,889,180.1	2,575,315.0	686,134.9	7

F.C.TAB.								
3	ابن سينا فارما	NOVORAPID 100 I.U./ML 5 FLEXPEN	1,970.0	1,657,809.9	1,951,587.0	293,777.1	3	
8	ابن سينا فارما	CRESTOR 10MG28F.C. TAB.	5,529.0	1,196,774.9	1,635,024.0	438,249.1	5	
9	ابن سينا فارما	LANTUS SOLOSTAR 100 I.U./ML 5 PENS	1,555.0	1,158,352.8	1,352,850.0	194,497.2	2	
13	ابن سينا فارما	RYZODEG 100 I.U./ML FLEXTOUCH PRE-FILLED PEN	2,492.0	1,027,198.4	1,234,362.0	207,163.6	2	
15	ابن سينا فارما	FORXIGA 10 MG 28 TABS.	2,325.0	950,189.7	1,149,660.0	199,470.3	2	
19	ابن سينا فارما	JANAGLIP PLUS 50/1000MG 28 F.C. TAB.	8,889.0	735,358.5	1,120,014.0	384,655.5	3	
20	ابن سينا فارما	BRILIQUE 90MG 56 F.C. TABS.	1,704.0	721,704.3	821,328.0	99,623.7	1	
23	ابن سينا فارما	ELIQUIS 5 MG 20 F.C. TABS.	2,080.0	685,831.6	836,380.0	150,548.4	1	
24	ابن سينا فارما	ASPIRIN PROTECT 100 MG 30 GASTRO- RESISTANT TAB.	12,086.0	652,842.6	797,238.0	144,395.4	2	
29	ابن سينا فارما	VASTAREL MR 35MG 30 F.C.TAB.	6,059.0	586,737.4	850,200.0	263,462.6	2	
35	ابن سينا فارما	ENTRESTO 50 MG (24/26 MG) 28 F.C. TABS.	408.0	516,494.9	607,292.0	90,797.1	1	
36	ابن سينا فارما	PROCORALAN 5MG 28 F.C. TABS.	2,428.0	501,526.1	681,532.0	180,005.9	2	
39	ابن سينا فارما	NEXIUM 40 MG 28 F.C. TAB.	1,876.0	477,346.9	648,728.0	171,381.1	1	

In [105...]: rank_supplier[1].iloc[[0]]['Product name']

```
Out[105...]: 1      TRESIBA 100 I.U./ML FLEXTOUCH PRE-FILLED PEN
Name: Product name, dtype: object
```

No of Products

```
In [106...]: len(rank_supplier[1])
```

```
Out[106...]: 3204
```

Products to be revised (sales > 0.5% & Return > 5%)

```
In [107...]: rank_supplier[1][(rank_supplier[1]['% from total product Sales'] >=
```

```
Out[107...]:
```

	Supplier	Product name	Quantity	total after discount tax 2	Total	total discount 2	pc di
53	ابن سينا فارما	PEPON PLUS 20 CAPS.	6,507.0	407,190.0	578,260.0	171,070.0	19
62	ابن سينا فارما	CONCOR 5 MG 30 F.C. TABS	7,621.0	371,230.8	490,815.0	119,584.2	14
69	ابن سينا فارما	PLAVIX 75MG 28 FILM COATED TAB.	1,444.0	334,862.9	396,509.0	61,646.1	11
74	ابن سينا فارما	JUSTECHOL 20MG 28 F.C. TABS.	5,380.0	325,456.8	618,700.0	293,243.2	40
89	ابن سينا فارما	EXFORGE 5MG/160MG 14 F.C. TAB.	2,272.0	283,268.1	409,184.0	125,915.9	13
107	ابن سينا فارما	TAREG 80MG 14 F.C.TAB.	2,705.0	259,209.3	352,272.8	93,063.5	10
108	ابن سينا فارما	EXFORGE HCT 5/160/12.5MG 14 F.C. TAB.	1,600.0	257,873.6	314,480.0	56,606.4	7
111	ابن سينا فارما	EXFORGE HCT 10/160/25MG 14 F.C. TAB.	1,679.0	255,408.7	311,474.0	56,065.3	7
115	ابن سينا فارما	CONCOR COR 2.5 MG 30 F.C. TABS	6,179.0	245,974.6	333,456.0	87,481.4	10

138	ابن سينا فارما	CHOLEROSE PLUS 10/20 MG 28 F.C.TAB.	1,269.0	206,654.8	298,653.0	91,998.2	10
168	ابن سينا فارما	EXFORGE 10MG/160MG 14 F.C. TAB.	1,389.0	175,909.6	254,874.0	78,964.4	9
188	ابن سينا فارما	ATACAND PLUS 16/12.5 MG 14 TABS.	1,885.0	161,105.2	218,660.0	57,554.8	7
191	ابن سينا فارما	TAREG 160MG 28 F.C. TAB.	772.0	159,748.5	216,108.0	56,359.5	6
218	ابن سينا فارما	TAREG 40MG 15 F.C.TAB.	1,512.0	146,734.7	199,113.0	52,378.3	5
253	ابن سينا فارما	ENTRESTO 200 MG (97/103 MG) 56 F.C. TABS.	51.0	131,253.6	154,416.0	23,162.4	3
278	ابن سينا فارما	MOSAPRIDE 5 MG 30 F.C.TAB.	1,793.0	119,711.4	175,299.0	55,587.6	5
334	ابن سينا فارما	LIMITLESS OSOFORTIN D3 5000IU 30 TAB	1,329.0	101,675.7	137,599.0	35,923.3	4
366	ابن سينا فارما	ATROZEMB 20/10MG 30 F.C. TAB.	793.0	95,138.1	137,982.0	42,843.9	4
407	ابن سينا فارما	LIPOCOMB 10/ 20 30 CAP	217.0	83,840.9	101,298.0	17,457.1	2
434	ابن سينا فارما	ZYROVAZET 10/10MG 30 F.C. TABS	603.0	80,163.4	111,996.0	31,832.6	3
447	ابن سينا فارما	NEVILOB 5MG 21 TAB.	1,296.0	77,724.1	112,570.5	34,846.4	4
453	ابن سينا فارما	SOLOFRESH 0.2% EYE DROPS 10 ML	2,060.0	76,745.7	110,850.0	34,104.3	3
458	ابن سينا فارما	GAPTIN 300MG 30 CAPS.	1,148.0	76,091.5	115,219.5	39,128.0	4
460	ابن سينا فارما	JUSPRIN 81MG 60 ENTERIC COATED TAB.	1,806.0	75,823.7	107,862.0	32,038.3	3
	ابن سينا	ALPHINTERN					

488	فارما	30 F.C. TABS.	1,306.0	71,330.9	96,330.0	24,999.1	2
561	ابن سينا فارما	LIPITOR 10 MG 28 TAB.	439.0	63,258.8	85,836.0	22,577.2	2
562	ابن سينا فارما	TEBONINA FORTE 40MG 20 F.C. TAB.	1,016.0	63,198.0	91,440.0	28,242.0	3
603	ابن سينا فارما	CEREBROMAP 200MG 30 CAPS.	657.0	59,902.4	86,431.5	26,529.1	3
642	ابن سينا فارما	DAFLON 500MG 30 F.C. TABS.	452.0	56,114.6	81,106.0	24,991.4	2
666	ابن سينا فارما	ATROZEMB 10/10MG 30 F.C. TAB.	450.0	54,183.2	78,300.0	24,116.8	2
678	ابن سينا فارما	VAXATO 15 MG 30 TABS	194.0	53,219.4	75,660.0	22,440.6	2
683	ابن سينا فارما	APIDRA 100I.U./ML 5 CARTIRIDGES	140.0	52,928.8	60,232.0	7,303.2	1
690	ابن سينا فارما	ROWATINEX 45 SOFT GELATIN CAPS.	1,147.0	52,616.1	76,298.0	23,681.9	3
768	ابن سينا فارما	APIDRA 100I.U./ML 5 PREFILLED PEN	109.0	46,266.9	52,723.5	6,456.6	1
772	ابن سينا فارما	LIPITOR 40 MG 14 TAB	363.0	45,854.4	62,238.0	16,383.6	1
782	ابن سينا فارما	PLETAAL 50 MG 20 TAB.	709.0	45,106.7	65,228.0	20,121.3	2
863	ابن سينا فارما	ULTIBRO BREEZHALER 110/50 MCG 30 INH. CAPS. + ...	87.0	40,316.5	47,699.2	7,382.7	1
871	ابن سينا فارما	JANAGLIP 100MG 28 F.C. TAB.	253.0	40,079.7	58,190.0	18,110.3	1
872	ابن سينا فارما	HYFRESH 0.2% EYE DROPS 10 ML	822.0	40,063.9	49,639.5	9,575.6	1
899	ابن سينا فارما	ATACAND 16MG 14 F.C. TAB.	497.0	38,746.8	55,451.0	16,704.2	1

911	ابن سينا فارما	ARTHFREE 20MG 30 F.C.TAB.	263.0	38,293.9	51,945.0	13,651.1	1
920	ابن سينا فارما	PROSTRIKE 5MG 30 CAPS.	383.0	37,570.3	54,366.0	16,795.7	1
956	ابن سينا فارما	CO-TAREG 320/25 MG 14 F.C. TAB.	324.0	36,076.5	52,147.0	16,070.5	1
982	ابن سينا فارما	CRESTOLIP 10MG 30 F.C. TABS.	483.0	35,357.1	51,228.0	15,870.9	1
983	ابن سينا فارما	COAPROVEL 300/12.5MG 14 TAB.	381.0	35,331.7	40,383.0	5,051.3	1
1013	ابن سينا فارما	OMEGA-3 PLUS 30 CAPS	396.0	34,344.4	45,072.0	10,727.6	1
1078	ابن سينا فارما	CONCOR 10 MG 30 F.C. TABLETS	512.0	32,014.5	44,019.0	12,004.5	1
1093	ابن سينا فارما	diacurimap 10 mg 30 tab	227.0	31,269.0	41,826.0	10,557.0	1
1116	ابن سينا فارما	OMEZ 40MG 20 CAPS.	408.0	30,493.0	44,094.0	13,601.0	1
1140	ابن سينا فارما	COVERSYL 5MG 30 F.C. TAB.	458.0	29,655.5	40,228.0	10,572.5	1

In [108...]: supplier_product_df

Out [108...]

	Supplier	Product name	Quantity	total after discount tax 2	Total	total discount %
0	AUG PHARMA	ZURCAL 40 MG 14 GASTRO RESISTANT TAB.	41,322.0	2,186,333.5	3,879,972.0	1,693,638.5
1	ابن سينا فارما	TRESIBA 100 I.U./ML FLEXTOUCH PRE-FILLED PEN	5,856.0	2,098,395.4	2,525,276.0	426,880.6
2	ابن سينا فارما	CRESTOR 20MG 28 F.C.TAB.	5,080.0	1,889,180.1	2,575,315.0	686,134.9
3	ابن سينا فارما	NOVORAPID 100 I.U./ML 5 FLEXPEN	1,970.0	1,657,809.9	1,951,587.0	293,777.1
4	شركة الاهرام شبيبن الكوم	PLAVIX 75MG 28 FILM COATED TAB.	7,539.0	1,480,685.9	1,864,329.0	383,643.1
...
37929	متنوعون الهرم	FORXIGA 10 MG 28 TABS.	15.0	-13,398.4	-12,806.0	592.4
37930	مخزن داوى	AFINITOR 10MG 30 TABS	-1.0	-15,000.0	-19,100.0	-4,100.0
37931	شركة سوفيكيو فارم	JANUMET 50/850MG 14 F.C. TAB.	-194.0	-17,675.6	-21,534.0	-3,858.4
37932	ديفارت	KALOGENA 100 MG 10 SACHETS	-500.0	-45,000.0	-75,000.0	-30,000.0
37933	القصواع	DIBAVALLY PLUS 50/1000 MG 28 TABS.	-1,008.0	-63,504.0	-141,120.0	-77,616.0

37934 rows × 14 columns

Rank 2

In [109... rank_supplier[2]

Out [109...]

		Supplier	Product name	Quantity	total after discount tax 2	Total	total discount
7	شركة فارما اوفر سيز	PLAVIX 75MG 28 FILM COATED TAB.		5,107.0	1,297,726.7	1,581,002.0	283,275.3
11	شركة فارما اوفر سيز	TRESIBA 100 I.U./ML FLEXTOUCH PRE-FILLED PEN		2,385.0	1,105,138.7	1,301,090.0	195,951.3
17	شركة فارما اوفر سيز	NOVORAPID 100 I.U./ML 5 FLEXPEN		826.0	866,508.2	1,019,596.0	153,087.8
21	شركة فارما اوفر سيز	LANTUS SOLOSTAR 100 I.U./ML 5 PENS		945.0	718,710.5	825,890.0	107,179.5
25	شركة فارما اوفر سيز	ENTRESTO 50 MG (24/26 MG) 28 F.C. TABS.		469.0	643,943.0	757,580.0	113,637.0
...
37787	شركة فارما اوفر سيز	SIMILAC TOTAL COMFORT MILK 357 GM		0.0	0.0	0.0	0.0
37807	شركة فارما اوفر سيز	SYRINGE ADULT 5 CM		0.0	0.0	0.0	0.0
37844	شركة فارما اوفر سيز	COVERSYL 5MG 15 F.C. TAB.		0.0	-22.5	-42.0	-19.5
37877	شركة فارما اوفر سيز	CONVARSIOTAM 100 MG 20 F.C.TABS.		-1.0	-134.0	-200.0	-66.0
37901	شركة فارما اوفر سيز	PROTOFIX 40MG VIAL I.V.		-10.0	-374.4	-585.0	-210.4

3049 rows x 15 columns

Top 15 Products by Sales

In [110... rank_supplier[2].head(15)

Out [110...]

		Supplier	Product name	Quantity	total after discount tax 2	Total	total discount 2
7	شركة فارما اوفر سيز		PLAVIX 75MG 28 FILM COATED TAB.	5,107.0	1,297,726.7	1,581,002.0	283,275.3
11	شركة فارما اوفر سيز		TRESIBA 100 I.U./ML FLEXTOUCH PRE-FILLED PEN	2,385.0	1,105,138.7	1,301,090.0	195,951.3
17	شركة فارما اوفر سيز		NOVORAPID 100 I.U./ML 5 FLEXPEN	826.0	866,508.2	1,019,596.0	153,087.8
21	شركة فارما اوفر سيز		LANTUS SOLOSTAR 100 I.U./ML 5 PENS	945.0	718,710.5	825,890.0	107,179.5
25	شركة فارما اوفر سيز		ENTRESTO 50 MG (24/26 MG) 28 F.C. TABS.	469.0	643,943.0	757,580.0	113,637.0
26	شركة فارما اوفر سيز		FORXIGA 10 MG 28 TABS.	1,533.0	639,860.6	771,464.0	131,603.4
27	شركة فارما اوفر سيز		CRESTOR 20MG 28 F.C.TAB.	1,572.0	639,380.1	866,760.0	227,379.9
30	شركة فارما اوفر سيز		RYZODEG 100 I.U./ML FLEXTOUCH PRE-FILLED PEN	1,196.0	579,974.1	686,872.0	106,897.9
37	شركة فارما اوفر سيز		PROCORALAN 5MG 28 F.C. TABS.	2,434.0	496,693.3	674,978.0	178,284.7
42	شركة فارما اوفر سيز		NEXIUM 40 MG 28 F.C. TAB.	1,422.0	462,225.4	624,472.0	162,246.6
46	شركة فارما اوفر سيز		CONTROLOC 40MG 14 GASTRORESISTANT TAB.	2,940.0	441,853.0	541,079.0	99,226.0
47	شركة فارما اوفر سيز		DIAMICRON 60 MG 30 M.R. SCORED TAB.	6,084.0	430,448.9	584,901.0	154,452.1
48	شركة فارما اوفر سيز		VASTAREL MR 35MG 30 F.C.TAB.	3,922.0	429,833.3	622,975.0	193,141.7
59	شركة فارما اوفر سيز		CRESTOR 10MG28F.C. TAB.	1,540.0	385,158.2	522,984.0	137,825.8
66	شركة فارما اوفر سيز		SINGULAIR 10MG 14 F.C. TAB.	2,587.0	342,754.7	496,704.0	153,949.3

No of Products

```
In [111... len(rank_supplier[2])
```

```
Out[111... 3049
```

Products to be revised (sales > 1% & Return > 5%)

```
In [112... rank_supplier[2][(rank_supplier[2]['% from total product Sales'] >=
```

```
Out[112...
```

	Supplier	Product name	Quantity	total after discount tax 2	Total	to discou
7	شركة فارما اوفر سيز	PLAVIX 75MG 28 FILM COATED TAB.	5,107.0	1,297,726.7	1,581,002.0	283,275
11	شركة فارما اوفر سيز	TRESIBA 100 I.U./ML FLEXTOUCH PRE-FILLED PEN	2,385.0	1,105,138.7	1,301,090.0	195,955
21	شركة فارما اوفر سيز	LANTUS SOLOSTAR 100 I.U./ML 5 PENS	945.0	718,710.5	825,890.0	107,175
25	شركة فارما اوفر سيز	ENTRESTO 50 MG (24/26 MG) 28 F.C. TABS.	469.0	643,943.0	757,580.0	113,630
27	شركة فارما اوفر سيز	CRESTOR 20MG 28 F.C.TAB.	1,572.0	639,380.1	866,760.0	227,375
30	شركة فارما اوفر سيز	RYZODEG 100 I.U./ML FLEXTOUCH PRE-FILLED PEN	1,196.0	579,974.1	686,872.0	106,895
42	شركة فارما اوفر سيز	NEXIUM 40 MG 28 F.C. TAB.	1,422.0	462,225.4	624,472.0	162,245
59	شركة فارما اوفر سيز	CRESTOR 10MG28F.C. TAB.	1,540.0	385,158.2	522,984.0	137,825
76	شركة فارما اوفر سيز	RECORMON 5000 I.U. 6 PREFILLED SYRINGE	165.0	324,489.1	381,786.0	57,295
81	شركة فارما اوفر سيز	XGEVA 120 MG VIAL S.C. INJ.	47.0	308,732.6	358,636.0	49,905
86	شركة فارما اوفر سيز	GENUPHIL ORIGINAL 50 F.C. TAB.	1,525.8	293,926.2	342,235.0	48,305
136	شركة فارما	EXFORGE HCT 5/160/12.5MG 14 F.C.	1,264.0	207,699.7	253,296.0	45,595

		اوفر سيز	TAB.				
151	شركة فارما اوفر سيز	PULMICORT 0.5MG/ML 20 NEBULIZER VIAL SUSP.	346.0	190,644.0	224,396.0	33,751	
152	شركة فارما اوفر سيز	TRULICITY 1.5MG/0.5ML 4 PRE-FILLED PENS	70.0	190,421.8	222,250.0	31,828	
173	شركة فارما اوفر سيز	PEPON PLUS 20 CAPS.	2,521.0	173,108.1	250,420.0	77,311	
175	شركة فارما اوفر سيز	SAXENDA 18MG/3ML 1 PRE-FILLED PEN S.C.	103.0	172,473.5	202,910.0	30,436	
179	شركة فارما اوفر سيز	BRILIQUE 90MG 56 F.C. TABS.	391.0	166,354.1	186,898.0	20,543	
184	شركة فارما اوفر سيز	LIPITOR 20 MG 28 TAB	851.0	163,114.2	221,274.0	58,159	
206	شركة فارما اوفر سيز	DULOXEPHRIN 30 MG 28 CAPS.	1,275.0	153,758.5	208,556.0	54,791	
220	شركة فارما اوفر سيز	PIASCLEDINE 300MG 15 CAPS.	443.0	144,430.0	164,125.0	19,695	
221	شركة فارما اوفر سيز	MILGA 40 F.C. TAB	2,243.0	144,352.5	197,459.0	53,106	
222	شركة فارما اوفر سيز	ATOR 40MG 10 F.C. TAB.	2,447.0	144,284.2	195,760.0	51,475	
230	شركة فارما اوفر سيز	NORVASC 5MG 30 TAB.	1,502.0	140,982.5	192,426.0	51,445	
231	شركة فارما اوفر سيز	MELLITOFIX MET 12.5/1000 MG 30 F.C. TABS.	947.0	140,578.6	203,131.5	62,552	
249	شركة فارما اوفر سيز	FORADIL 12 MCG 30 CAPS.+INHALER	756.0	132,532.3	161,784.0	29,251	
259	شركة فارما اوفر سيز	BETASERC 24 MG 40 TAB.	776.0	128,725.8	157,068.0	28,342	
269	شركة فارما اوفر سيز	DEPOVIT B12-1000MCG/ML 5 I.M. AMP.	2,307.0	125,259.4	183,252.5	57,991	
287	شركة فارما اوفر سيز	ATOREZA 40/10MG 28 F.C. TAB.	820.0	115,414.3	167,000.0	51,581	
289	شركة فارما اوفر سيز	CO-TAREG 160/12.5MG 14 F.C. TAB	1,340.0	115,077.7	166,686.0	51,608	
291	شركة فارما اوفر سيز	CH ALPHA PLUS 10 SACHETS	382.0	113,754.1	129,266.0	15,511	

298	شركة فارما اوفر سيز	DOXIUM 500MG 30 CAPS.	734.0	109,834.7	159,745.0	49,910
309	شركة فارما اوفر سيز	GAPTIN 300MG 30 CAPS.	1,283.0	106,418.6	161,140.5	54,720
311	شركة فارما اوفر سيز	OZEMPIC 1MG 1 PREFILLED PEN	32.0	106,250.0	125,000.0	18,750
314	شركة فارما اوفر سيز	ULTIBRO BREEZHALER 110/50 MCG 30 INH. CAPS. + ...	226.0	105,680.0	124,329.4	18,649
317	شركة فارما اوفر سيز	CONCOR COR 2.5 MG 30 F.C. TABS	3,115.0	104,925.3	142,303.5	37,378
325	شركة فارما اوفر سيز	CIPRAPRO 10MG 30 F.C. TAB.	739.0	103,396.8	164,598.0	61,200
342	شركة فارما اوفر سيز	FORTEO (20MCG/80MCL) PRE-FILLED PEN 2.4 ML	17.0	99,911.6	107,559.0	7,640
343	شركة فارما اوفر سيز	FEMARA 2.5MG 30 F.C. TABS.	152.0	99,831.3	115,368.0	15,536
346	شركة فارما اوفر سيز	NORVASC 10 MG 15 TAB.	1,619.0	99,570.1	135,124.5	35,554
378	شركة فارما اوفر سيز	ATOR 20MG 10 F.C. TAB.	1,903.0	90,887.5	122,877.0	31,989
417	شركة فارما اوفر سيز	LIPITOR 10 MG 28 TAB.	561.0	82,524.9	112,116.0	29,590
435	شركة فارما اوفر سيز	FORBUDES 400/12MCG 60 INHALATION CAPS.+INHALER	366.0	80,075.2	115,764.0	35,688
436	شركة فارما اوفر سيز	JUSTECHOL 10 MG 28 F.C. TABS.	1,705.0	79,640.1	148,930.0	69,289
451	شركة فارما اوفر سيز	ICOSALIP 1 GM 20 CAPS.	442.0	77,001.3	111,366.0	34,364
463	شركة فارما اوفر سيز	CLEXANE 40MG/0.4ML 2 PREFILLED SYRINGE	342.0	75,237.6	88,890.0	13,652
515	شركة فارما اوفر سيز	SPIRIVA 18MCG 30 INH. CAPS.+HANDIHALER	198.0	67,541.8	82,368.0	14,826
599	شركة فارما اوفر سيز	BLADOGRA XR 50 MG 30 F.C. TABS.	291.0	59,952.3	86,661.0	26,708
607	شركة فارما اوفر سيز	TIRATAM 1000MG 30 F.C. TAB.	312.0	59,566.7	85,752.0	26,185

611	شركة فارما اوفر سيز	ZYROVAZET 10/40MG 30 TABS	196.0	59,050.9	79,440.0	20,38
615	شركة فارما اوفر سيز	PROCORALAN 7.5MG 28 F.C. TABS.	319.0	58,457.5	78,474.0	20,016

In [113]: df_purchased[df_purchased['Product name'] == 'JANAGLIP PLUS 50/1000 mg F.C. CAPLETS']
supplier_monthly_data['Discount %'] = supplier_monthly_data['total discount 2'] / supplier_monthly_data['Total'] * 100

Out[113]:

month		total after discount tax 2	Product name	Total	total discount 2	Discount %
0	1	1,349,890.7	816	1,850,527.2	500,636.5	27.1
1	2	1,528,132.2	1272	2,037,819.9	509,687.7	25.0
2	3	1,224,934.7	1066	1,612,620.9	387,686.1	24.0
3	4	1,383,562.7	721	1,866,391.9	482,829.3	25.9
4	5	2,783,159.6	1431	3,735,231.6	952,072.0	25.5
5	6	1,894,914.9	945	2,516,082.8	621,167.8	24.7
6	7	3,699,753.8	1270	4,933,200.3	1,233,446.5	25.0
7	8	1,957,012.8	689	2,544,819.0	587,806.2	23.1
8	9	5,552,087.5	1432	7,429,594.1	1,877,506.6	25.3
9	10	7,040,928.8	1416	9,228,197.8	2,187,269.0	23.7
10	11	4,687,726.6	1626	6,136,363.5	1,448,636.9	23.6
11	12	10,111,740.8	1560	13,371,984.3	3,260,243.6	24.4

In [114]: df[df['Product id'] == 23054][['Date', 'new discount', 'Product name', 'Supplier', 'Unit Price']]

Out[114]:

	Date	new discount	Product name	Supplier	Unit Price
197981	2024-12-31	28.0	THIOTACID COMPOUND 600 MG 30 F.C.CAPLETS	الشمس فارم	249.0
197057	2024-12-30	15.0	THIOTACID COMPOUND 600 MG 30 F.C.CAPLETS	متنوعون المعادي	249.0
196792	2024-12-30	25.0	THIOTACID COMPOUND 600 MG 30 F.C.CAPLETS	شركة فارما اوفر سيز	249.0
195974	2024-12-29	30.0	THIOTACID COMPOUND 600 MG 30 F.C.CAPLETS	ميديكال فارما اوفر نقصي	249.0
195527	2024-12-28	25.0	THIOTACID COMPOUND 600 MG 30 F.C.CAPLETS	شركة فارما اوفر سيز	249.0
191921	2024-	25.0	THIOTACID COMPOUND	ابن سينا فارما	249.0

	12-24		600 MG 30 F.C.CAPLETS		
191688	2024-12-23	25.0	THIOTACID COMPOUND 600 MG 30 F.C.CAPLETS	شركة فارما اوفر سيز	-249.0
191134	2024-12-23	28.0	THIOTACID COMPOUND 600 MG 30 F.C.CAPLETS	شركة جلوبال فارما	249.0
191084	2024-12-23	25.0	THIOTACID COMPOUND 600 MG 30 F.C.CAPLETS	شركة فارما اوفر سيز	249.0
190190	2024-12-21	25.0	THIOTACID COMPOUND 600 MG 30 F.C.CAPLETS	شركة فارما اوفر سيز	249.0
189964	2024-12-20	28.0	THIOTACID COMPOUND 600 MG 30 F.C.CAPLETS	شركة جلوبال فارما	249.0
189805	2024-12-20	25.0	THIOTACID COMPOUND 600 MG 30 F.C.CAPLETS	شركة فارما اوفر سيز	249.0
188941	2024-12-18	25.0	THIOTACID COMPOUND 600 MG 30 F.C.CAPLETS	ميديكال فارما _ اوفر نكدي	180.0
188708	2024-12-18	25.0	THIOTACID COMPOUND 600 MG 30 F.C.CAPLETS	ميديكال فارما _ اوفر نكدي	180.0
188183	2024-12-17	12.0	THIOTACID COMPOUND 600 MG 30 F.C.CAPLETS	متنوعون المعادي	249.0
186841	2024-12-15	12.0	THIOTACID COMPOUND 600 MG 30 F.C.CAPLETS	متنوعون المعادي	249.0
186232	2024-12-14	27.0	THIOTACID COMPOUND 600 MG 30 F.C.CAPLETS	شركة جلوبال فارما	249.0
185069	2024-12-12	28.0	THIOTACID COMPOUND 600 MG 30 F.C.CAPLETS	الشمس فارم	249.0
185035	2024-12-12	28.0	THIOTACID COMPOUND 600 MG 30 F.C.CAPLETS	الشمس فارم	249.0
184161	2024-12-11	29.0	THIOTACID COMPOUND 600 MG 30 F.C.CAPLETS	شركة مالتي ستورز فارما	249.0
181726	2024-12-08	31.0	THIOTACID COMPOUND 600 MG 30 F.C.CAPLETS	شركة الاهرام شبين الكوم	249.0
179462	2024-12-06	27.0	THIOTACID COMPOUND 600 MG 30 F.C.CAPLETS	ميديكال فارما _ اوفر نكدي	-249.0
179459	2024-12-06	12.0	THIOTACID COMPOUND 600 MG 30 F.C.CAPLETS	ميديكال فارما _ اوفر نكدي	163.5
179429	2024-12-06	27.0	THIOTACID COMPOUND 600 MG 30 F.C.CAPLETS	ميديكال فارما _ اوفر نكدي	249.0
179355	2024-12-06	27.0	THIOTACID COMPOUND 600 MG 30 F.C.CAPLETS	شركة المصري	249.0
178804	2024-12-06	31.0	THIOTACID COMPOUND 600 MG 30 F.C.CAPLETS	شركة الاهرام شبين الكوم	249.0
177723	2024-12-04	26.0	THIOTACID COMPOUND 600 MG 30 F.C.CAPLETS	مخزن الأخوة المتحدون	249.0

177575	2024-12-04	27.0	THIOTACID COMPOUND 600 MG 30 F.C.CAPLETS	الشمس فارم	249.0
176704	2024-12-03	28.0	THIOTACID COMPOUND 600 MG 30 F.C.CAPLETS	ميديكال فارما اونر نكدي	249.0
175911	2024-12-02	29.0	THIOTACID COMPOUND 600 MG 30 F.C.CAPLETS	شركة مالي ستورز فارما	249.0

In [115...]

```
product_name = rank_supplier[2].iloc[1]['Product name']

product_1 = (df_purchased[df_purchased['Product name'] == product_name]
             .groupby('Supplier')
             .agg({
                 'total after discount tax 2': 'sum',
                 'Total': 'sum',
                 'total discount 2': 'sum'
             })
             .reset_index()
             .sort_values(by='total after discount tax 2', ascending=False)

# Calculate Discount %
product_1['Discount %'] = (product_1['total discount 2'] / product_1['Total']) * 100

# Display the resulting DataFrame
product_1[product_1['total after discount tax 2'] > 100000]
```

Out[115...]

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
0	ابن سينا فارما	2,098,395.4	2,525,276.0	426,880.6	16.9
8	شركة فارما اوفر سيز	1,105,138.7	1,301,090.0	195,951.3	15.1
5	شركة الاهرام شبين الكوم	213,117.8	248,794.0	35,676.2	14.3
7	شركة جلوبال فارما	177,478.5	214,616.0	37,137.5	17.3

In [116...]

```
# supplier_1 = product_1['Discount %'][product_1['Supplier'] == 'فارما اونر نكدي']
# if supplier_1.size > 0:
#     supplier_1 = supplier_1[0]

for i, row in enumerate(rank_supplier[1]['Product name'].head(45)):
    product_1 = (df_purchased[df_purchased['Product name'] == row]
                 .groupby('Supplier')
                 .agg({
                     'total after discount tax 2': 'sum',
                     'Total': 'sum',
                     'total discount 2': 'sum'
                 })
                 .reset_index())
```

```

        .sort_values(by='total after discount tax 2', ascending=False)

product_1['Discount %'] = (product_1['total discount 2'] / product_1['Total']) * 100
supplier_1 = product_1['Discount %'][product_1['Supplier'] == 'ابن سينا فارما']
if supplier_1.size > 0:
    supplier_1 = supplier_1[0]
# Print the index and filtered DataFrame
if supplier_1 is not None:
    filtered_product_1 = product_1[
        (product_1['total after discount tax 2'] > 100000) &
        (product_1['Discount %'] >= supplier_1) &
        (product_1['Supplier'].iloc[0] == 'ابن سينا فارما')

    if len(filtered_product_1) > 1:
        print(f"Product Index: {i}, Product Name: {row}")
        display(filtered_product_1) # Display the DataFrame

# print(f"Product Index: {i}, Product Name: {row}")
# display(filtered_product_1)
# print() # Add an empty line for better readability between products

```

0, Product Name: TRESIBA 100 I.U./ML FLEXTOUCH PRE-FILLED PEN

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
0	ابن سينا فارما	2,098,395.4	2,525,276.0	426,880.6	16.9
7	شركة جلوبال فارما	177,478.5	214,616.0	37,137.5	17.3

1, Product Name: CRESTOR 20MG 28 F.C.TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
3	ابن سينا فارما	1,889,180.1	2,575,315.0	686,134.9	26.6
9	شركة الاهرام شبين الكوم	395,213.4	538,940.0	143,726.6	26.7
10	شركة جلوبال فارما	308,268.2	427,110.0	118,841.8	27.8
21	ميديكال فارما _ اونر نيدي	128,154.0	176,400.0	48,246.0	27.4

2, Product Name: NOVORAPID 100 I.U./ML 5 FEXPEN

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
2	ابن سينا فارما	1,657,809.9	1,951,587.0	293,777.1	15.1
9	شركة جلوبال فارما	149,976.5	179,983.5	30,007.0	16.7

4, Product Name: LANTUS SOLOSTAR 100 I.U./ML 5 PENS

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
3	ابن سينا فارما	1,158,352.8	1,352,850.0	194,497.2	14.4
8	شركة الاهرام شبين الكوم	468,321.0	561,150.0	92,829.0	16.5
10	شركة جلوبال فارما	400,705.5	486,416.0	85,710.5	17.6

6, Product Name: FORXIGA 10 MG 28 TABS.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
1	ابن سينا فارما	950,189.7	1,149,660.0	199,470.3	17.4
10	شركة جلوبال فارما	241,170.3	293,165.0	51,994.7	17.7
8	شركة الاهرام شبين الكوم	211,995.4	262,314.0	50,318.6	19.2
7	رامكو فارم لتجارة الادوية	130,888.4	159,620.0	28,731.6	18.0

8, Product Name: BRILIQUE 90MG 56 F.C. TABS.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
0	ابن سينا فارما	721,704.3	821,328.0	99,623.7	12.1
9	شركة الاهرام شبين الكوم	130,508.3	152,960.0	22,451.7	14.7

9, Product Name: ELIQUIS 5 MG 20 F.C. TABS.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
3	ابن سينا فارما	685,831.6	836,380.0	150,548.4	18.0
12	شركة فارما اوفر سيز	300,293.8	366,212.0	65,918.2	18.0
9	شركة الاهرام شبين الكوم	117,987.7	150,626.0	32,638.3	21.7

10, Product Name: ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
1	ابن سينا فارما	652,842.6	797,238.0	144,395.4	18.1
12	شركة فارما اوفر سيز	325,260.5	405,378.0	80,117.5	19.8
9	شركة الاهرام شبين الكوم	281,944.3	391,590.0	109,645.7	28.0
10	شركة جلوبال فارما	139,081.7	194,154.0	55,072.3	28.4

11, Product Name: VASTAREL MR 35MG 30 F.C.TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
2	ابن سينا فارما	586,737.4	850,200.0	263,462.6	31.0
14	شركة فارما اوفر سيز	429,833.3	622,975.0	193,141.7	31.0
9	شركة الاهرام شبين الكوم	269,332.6	404,312.5	134,979.9	33.4
1	Wecare	163,309.5	237,712.5	74,403.0	31.3
7	الفاروق	153,936.2	232,250.0	78,313.8	33.7

13, Product Name: PROCORALAN 5MG 28 F.C. TABS.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
2	ابن سينا فارما	501,526.1	681,532.0	180,005.9	26.4
12	شركة فارما اوفر سيز	496,693.3	674,978.0	178,284.7	26.4
10	شركة جلوبال فارما	331,727.7	459,320.0	127,592.3	27.8
11	شركة سوفيكو فارم	210,776.1	293,842.0	83,065.9	28.3
1	Wecare	175,051.1	241,818.0	66,766.9	27.6

15, Product Name: ZYROVAZET 10/20MG 30 F.C. TABS

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
2	ابن سينا فارما	456,688.6	635,442.0	178,753.4	28.1
12	شركة فارما اوفر سيز	152,305.1	214,068.0	61,762.9	28.9

18, Product Name: MILGA ADVANCE 30 F.C. TAB

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
1	ابن سينا فارما	425,925.2	580,552.5	154,627.3	26.6
7	شركة الاهرام شبين الكوم	153,858.8	210,642.0	56,783.2	27.0

19, Product Name: DAPAGLIF 10 MG 14 F.C.TABS.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
1	ابن سينا فارما	413,825.1	598,910.0	185,084.9	30.9
13	شركة فارما اوفر سيز	182,251.6	264,089.5	81,837.9	31.0

20, Product Name: PEPON PLUS 20 CAPS.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
2	ابن سينا فارما	407,190.0	578,260.0	171,070.0	29.6
12	شركة فارما اوفر سيز	173,108.1	250,420.0	77,311.9	30.9

21, Product Name: FORADIL 12 MCG 60 CAPS.+INHALER

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
1	ابن سينا فارما	389,368.0	458,080.0	68,712.0	15.0
7	شركة فارما اوفر سيز	226,576.0	266,560.0	39,984.0	15.0
3	شركة الاهرام شبين الكوم	213,315.2	252,000.0	38,684.8	15.4

30, Product Name: SYSTANE ULTRA EYE DROPS 10 ML

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
1	ابن سينا فارما	296,020.2	337,760.0	41,739.8	12.4
9	شركة جلوبال فارما	261,665.2	325,820.0	64,154.8	19.7
7	شركة الاهرام شبين الكوم	120,107.2	140,200.0	20,092.8	14.3

31, Product Name: EXFORGE 5MG/160MG 14 F.C. TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
1	ابن سينا فارما	283,268.1	409,184.0	125,915.9	30.8
10	شركة فارما اوفر سيز	237,183.4	343,744.0	106,560.6	31.0

32, Product Name: ATOR 40MG 10 F.C. TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
2	ابن سينا فارما	279,762.4	381,278.0	101,515.6	26.6
8	الفاروق	155,582.4	222,160.0	66,577.6	30.0

37, Product Name: TAREG 80MG 14 F.C.TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
0	ابن سينا فارما	259,209.3	352,272.8	93,063.5	26.4
9	شركة فارما اوفر سيز	151,115.3	205,777.0	54,661.7	26.6

38, Product Name: EXFORGE HCT 5/160/12.5MG 14 F.C. TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
2	ابن سينا فارما	257,873.6	314,480.0	56,606.4	18.0
15	شركة فارما اوفر سيز	207,699.7	253,296.0	45,596.3	18.0
13	شركة جلوبال فارما	141,109.5	172,787.5	31,678.0	18.3
26	ميديكال فارما _ اوذر نقدي	124,266.7	165,908.0	41,641.3	25.1

40, Product Name: CONCOR COR 2.5 MG 30 F.C. TABS

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
1	ابن سينا فارما	245,974.6	333,456.0	87,481.4	26.2
13	شركة فارما اوفر سيز	104,925.3	142,303.5	37,378.2	26.3

In [117...]: rank_supplier[1].head(20)

Out[117...]:

	Supplier	Product name	Quantity	total after discount tax 2	Total	tot discou
1	ابن سينا فارما	TRESIBA 100 I.U./ML FLEXTOUCH PRE- FILLED PEN	5,856.0	2,098,395.4	2,525,276.0	426,880
2	ابن سينا فارما	CRESTOR 20MG 28 F.C.TAB.	5,080.0	1,889,180.1	2,575,315.0	686,134
3	ابن سينا فارما	NOVORAPID 100 I.U./ML 5 FLEXPEN	1,970.0	1,657,809.9	1,951,587.0	293,778
8	ابن سينا فارما	CRESTOR 10MG28F.C. TAB.	5,529.0	1,196,774.9	1,635,024.0	438,249
9	ابن سينا فارما	LANTUS SOLOSTAR 100 I.U./ML 5 PENS	1,555.0	1,158,352.8	1,352,850.0	194,497
13	ابن سينا فارما	RYZODEG 100 I.U./ML FLEXTOUCH PRE- FILLED PEN	2,492.0	1,027,198.4	1,234,362.0	207,163
15	ابن سينا فارما	FORXIGA 10 MG 28 TABS.	2,325.0	950,189.7	1,149,660.0	199,470
19	ابن سينا فارما	JANAGLIP PLUS 50/1000MG 28 F.C. TAB.	8,889.0	735,358.5	1,120,014.0	384,655
20	ابن سينا فارما	BRILIQUE 90MG 56 F.C. TABS.	1,704.0	721,704.3	821,328.0	99,623

23	ابن سينا فارما	ELIQUIS 5 MG 20 F.C. TABS.	2,080.0	685,831.6	836,380.0	150,548
24	ابن سينا فارما	ASPIRIN PROTECT 100 MG 30 GASTRO- RESISTANT TAB.	12,086.0	652,842.6	797,238.0	144,395
29	ابن سينا فارما	VASTAREL MR 35MG 30 F.C.TAB.	6,059.0	586,737.4	850,200.0	263,462
35	ابن سينا فارما	ENTRESTO 50 MG (24/26 MG) 28 F.C. TABS.	408.0	516,494.9	607,292.0	90,797
36	ابن سينا فارما	PROCORALAN 5MG 28 F.C. TABS.	2,428.0	501,526.1	681,532.0	180,005
39	ابن سينا فارما	NEXIUM 40 MG 28 F.C. TAB.	1,876.0	477,346.9	648,728.0	171,381
43	ابن سينا فارما	ZYROVAZET 10/20MG 30 F.C. TABS	2,431.0	456,688.6	635,442.0	178,753
44	ابن سينا فارما	COSENTYX 150 MG/ML S.C. PREFILLED PEN	80.0	452,626.8	504,176.0	51,549
45	ابن سينا فارما	CONTROLOC 40MG 14 GASTRORESISTANT TAB.	3,169.0	444,845.3	542,674.0	97,828
49	ابن سينا فارما	MILGA ADVANCE 30 F.C. TAB	4,961.0	425,925.2	580,552.5	154,627
50	ابن سينا فارما	DAPAGLIF 10 MG 14 F.C.TABS.	4,420.0	413,825.1	598,910.0	185,084

In [118]: df_supplier_data[['Supplier', 'Supplier Rank by Sales']]

Out [118...]

	Supplier	Supplier Rank by Sales
0	ابن سينا فارما	1
1	شركة فارما اوفر سيز	2
2	شركة جلوبال فارما	3
3	شركة الاهرام شبين الكوم	4
4	مخزن السعيد	5
...
67	محزن 2020	68
68	شركة فارما ايمدج	69
69	شركة صيادلة المستقبل	70
70	ديفارت	71
71	القصواع	72

72 rows × 2 columns

In [119...]

```

supplier_name = df_supplier_data['Supplier'][df_supplier_data['Supplier'].size > 0]
if supplier_name.size > 0:
    supplier_name = supplier_name[0]
if supplier_name is not None:
    for i, row in enumerate(rank_supplier[7]['Product name']):
        rank_product = (df_purchased[df_purchased['Product name'] == row].groupby('Supplier')
                        .agg({
                            'total after discount tax 2': 'sum',
                            'Total': 'sum',
                            'total discount 2': 'sum'
                        })
                        .reset_index()
                        .sort_values(by='total after discount tax 2', ascending=False))
        product['Discount %'] = (product['total discount 2'] / product['Total']) * 100
        supplier_discount = product['Discount %'][product['Supplier'] == supplier_name]
        if supplier_discount.size > 0:
            supplier_discount = supplier_discount[0]
        # Print the index and filtered DataFrame
        if supplier_discount is not None:
            filtered_product = product[
                (product['total after discount tax 2'] > 100000) &
                (product['Discount %'] >= supplier_discount) &
                (product['Supplier'].iloc[0] == supplier_name)]
            if len(filtered_product) > 1:
                print(f'{i}, Product Name: {row}')
                display(filtered_product) # Display the DataFrame

```

1, Product Name: DIBAVALLY PLUS 50/1000 MG 28 TABS.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
1	AUG PHARMA	1,442,756.2	2,383,780.0	941,023.8	39.5
2	Samir Supplier	127,008.0	211,680.0	84,672.0	40.0

2, Product Name: ESMORAP 40MG 14 CAPS.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
1	AUG PHARMA	548,421.4	921,908.0	373,486.6	40.5
3	ابن سينا فارما	158,674.6	272,250.0	113,575.4	41.7

```
In [120]: for supplierrank in range(1,10):
    supplier_name = df_supplier_data['Supplier'][df_supplier_data['Supplier'].size > 0]
    print(f"Supplier Name: {supplier_name}, Supplier Rank: {supplierrank}")
    if supplier_name.size > 0:
        supplier_name = supplier_name[0]
    if supplier_name is not None:
        for i , row in enumerate(rank_supplier[supplierrank]['Product']):
            product = (df_purchased[df_purchased['Product name'] == row['Product']].groupby('Supplier').agg({
                'total after discount tax 2': 'sum',
                'Total': 'sum',
                'total discount 2': 'sum'
            }))
            .reset_index()
            .sort_values(by='total after discount tax 2', ascending=False)
            product['Discount %'] = (product['total discount 2'] / product['Total']) * 100
            supplier_discount = product['Discount %'][product['Supplier'] == supplier_name]
            if supplier_discount.size > 0:
                supplier_discount = supplier_discount[0]
            # Print the index and filtered DataFrame
            if supplier_discount is not None:
                filtered_product = product[
                    (product['total after discount tax 2'] > 1000) &
                    (product['Discount %'] >= supplier_discount) &
                    (product['Supplier'].iloc[0] == supplier_name)]
                if len(filtered_product) > 1:
                    print(f"Product Rank:: {i +1}, Product Name: {row['Product']}")
                    display(filtered_product) # Display the DataFrame
    print(" ")
    print(" ")
    print(" ")
    print(" ")
```

Supplier Name: [ابن سينا فارما'], Supplier Rank: 1

Product Rank:: 1, Product Name: TRESIBA 100 I.U./ML FLEXTOUCH PRE-FILLED PEN

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
0	ابن سينا فارما	2,098,395.4	2,525,276.0	426,880.6	16.9
7	شركة جلوبال فارما	177,478.5	214,616.0	37,137.5	17.3

Product Rank:: 2, Product Name: CRESTOR 20MG 28 F.C.TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
3	ابن سينا فارما	1,889,180.1	2,575,315.0	686,134.9	26.6
9	شركة الاهرام شبين الكوم	395,213.4	538,940.0	143,726.6	26.7
10	شركة جلوبال فارما	308,268.2	427,110.0	118,841.8	27.8
21	مبديكال فارما _ اونر نيدي	128,154.0	176,400.0	48,246.0	27.4

Product Rank:: 3, Product Name: NOVORAPID 100 I.U./ML 5 FLEXPEN

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
2	ابن سينا فارما	1,657,809.9	1,951,587.0	293,777.1	15.1
9	شركة جلوبال فارما	149,976.5	179,983.5	30,007.0	16.7

Product Rank:: 5, Product Name: LANTUS SOLOSTAR 100 I.U./ML 5 PENS

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
3	ابن سينا فارما	1,158,352.8	1,352,850.0	194,497.2	14.4
8	شركة الاهرام شبين الكوم	468,321.0	561,150.0	92,829.0	16.5
10	شركة جلوبال فارما	400,705.5	486,416.0	85,710.5	17.6

Product Rank:: 7, Product Name: FORXIGA 10 MG 28 TABS.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
1	ابن سينا فارما	950,189.7	1,149,660.0	199,470.3	17.4
10	شركة جلوبال فارما	241,170.3	293,165.0	51,994.7	17.7
8	شركة الاهرام شبين الكوم	211,995.4	262,314.0	50,318.6	19.2
7	رامكو فارم لتجارة الادوية	130,888.4	159,620.0	28,731.6	18.0

Product Rank:: 9, Product Name: BRILIQUE 90MG 56 F.C. TABS.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
0	ابن سينا فارما	721,704.3	821,328.0	99,623.7	12.1
9	شركة الاهرام شبين الكوم	130,508.3	152,960.0	22,451.7	14.7

Product Rank:: 10, Product Name: ELIQUIS 5 MG 20 F.C. TABS.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
3	ابن سينا فارما	685,831.6	836,380.0	150,548.4	18.0
12	شركة فارما اوفر سيز	300,293.8	366,212.0	65,918.2	18.0
9	شركة الاهرام شبين الكوم	117,987.7	150,626.0	32,638.3	21.7

Product Rank:: 11, Product Name: ASPIRIN PROTECT 100 MG 30 GASTRO-RE
SISTANT TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
1	ابن سينا فارما	652,842.6	797,238.0	144,395.4	18.1
12	شركة فارما اوفر سيز	325,260.5	405,378.0	80,117.5	19.8
9	شركة الاهرام شبين الكوم	281,944.3	391,590.0	109,645.7	28.0
10	شركة جلوبال فارما	139,081.7	194,154.0	55,072.3	28.4

Product Rank:: 12, Product Name: VASTAREL MR 35MG 30 F.C.TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
2	ابن سينا فارما	586,737.4	850,200.0	263,462.6	31.0
14	شركة فارما اوفر سيز	429,833.3	622,975.0	193,141.7	31.0
9	شركة الاهرام شبين الكوم	269,332.6	404,312.5	134,979.9	33.4
1	Wecare	163,309.5	237,712.5	74,403.0	31.3
7	الفاروق	153,936.2	232,250.0	78,313.8	33.7

Product Rank:: 14, Product Name: PROCORALAN 5MG 28 F.C. TABS.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
2	ابن سينا فارما	501,526.1	681,532.0	180,005.9	26.4
12	شركة فارما اوفر سيز	496,693.3	674,978.0	178,284.7	26.4
10	شركة جلوبال فارما	331,727.7	459,320.0	127,592.3	27.8
11	شركة سوفيكو فارم	210,776.1	293,842.0	83,065.9	28.3
1	Wecare	175,051.1	241,818.0	66,766.9	27.6

Product Rank:: 16, Product Name: ZYROVAZET 10/20MG 30 F.C. TABS

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
2	ابن سينا فارما	456,688.6	635,442.0	178,753.4	28.1
12	شركة فارما اوفر سيز	152,305.1	214,068.0	61,762.9	28.9

Product Rank:: 19, Product Name: MILGA ADVANCE 30 F.C. TAB

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
1	ابن سينا فارما	425,925.2	580,552.5	154,627.3	26.6
7	شركة الاهرام شبين الكوم	153,858.8	210,642.0	56,783.2	27.0

Product Rank:: 20, Product Name: DAPAGLIF 10 MG 14 F.C.TABS.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
1	ابن سينا فارما	413,825.1	598,910.0	185,084.9	30.9
13	شركة فارما اوفر سيز	182,251.6	264,089.5	81,837.9	31.0

Product Rank:: 21, Product Name: PEPON PLUS 20 CAPS.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
2	ابن سينا فارما	407,190.0	578,260.0	171,070.0	29.6
12	شركة فارما اوفر سيز	173,108.1	250,420.0	77,311.9	30.9

Product Rank:: 22, Product Name: FORADIL 12 MCG 60 CAPS.+INHALER

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
1	ابن سينا فارما	389,368.0	458,080.0	68,712.0	15.0
7	شركة فارما اوفر سيز	226,576.0	266,560.0	39,984.0	15.0
3	شركة الاهرام شبين الكوم	213,315.2	252,000.0	38,684.8	15.4

Product Rank:: 31, Product Name: SYSTANE ULTRA EYE DROPS 10 ML

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
1	ابن سينا فارما	296,020.2	337,760.0	41,739.8	12.4
9	شركة جلوبال فارما	261,665.2	325,820.0	64,154.8	19.7
7	شركة الاهرام شبين الكوم	120,107.2	140,200.0	20,092.8	14.3

Product Rank:: 32, Product Name: EXFORGE 5MG/160MG 14 F.C. TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
1	ابن سينا فارما	283,268.1	409,184.0	125,915.9	30.8
10	شركة فارما اوفر سيز	237,183.4	343,744.0	106,560.6	31.0

Product Rank:: 33, Product Name: ATOR 40MG 10 F.C. TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
2	ابن سينا فارما	279,762.4	381,278.0	101,515.6	26.6
8	الفاروق	155,582.4	222,160.0	66,577.6	30.0

Product Rank:: 38, Product Name: TAREG 80MG 14 F.C.TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
0	ابن سينا فارما	259,209.3	352,272.8	93,063.5	26.4
9	شركة فارما اوفر سيز	151,115.3	205,777.0	54,661.7	26.6

Product Rank:: 39, Product Name: EXFORGE HCT 5/160/12.5MG 14 F.C. TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
2	ابن سينا فارما	257,873.6	314,480.0	56,606.4	18.0
15	شركة فارما اوفر سيز	207,699.7	253,296.0	45,596.3	18.0
13	شركة جلوبال فارما	141,109.5	172,787.5	31,678.0	18.3
26	ميديكال فارما _ اونر نيدي	124,266.7	165,908.0	41,641.3	25.1

Product Rank:: 41, Product Name: CONCOR COR 2.5 MG 30 F.C. TABS

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
1	ابن سينا فارما	245,974.6	333,456.0	87,481.4	26.2
13	شركة فارما اوفر سيز	104,925.3	142,303.5	37,378.2	26.3

Product Rank:: 47, Product Name: THIOTACID 600MG ORIGINAL 30 F.C.TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
0	ابن سينا فارما	209,198.1	284,102.0	74,903.9	26.4
9	شركة فارما اوفر سيز	154,957.2	212,108.0	57,150.8	26.9

Product Rank:: 58, Product Name: EXFORGE 10MG/160MG 14 F.C. TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
2	ابن سينا فارما	175,909.6	254,874.0	78,964.4	31.0
12	شركة فارما اوفر سيز	147,676.8	214,018.0	66,341.2	31.0

Product Rank:: 62, Product Name: NORVASC 5MG 30 TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
1	ابن سينا فارما	166,794.9	226,815.0	60,020.1	26.5
11	شركة فارما اوفر سيز	140,982.5	192,426.0	51,443.5	26.7

Product Rank:: 63, Product Name: ATACAND PLUS 16/12.5 MG 14 TABS.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
1	ابن سينا فارما	161,105.2	218,660.0	57,554.8	26.3
8	شركة فارما اوفر سيز	139,433.9	190,008.0	50,574.1	26.6

Product Rank:: 80, Product Name: BETOLVEX 1MG/ML 2 AMP. I.M.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
2	ابن سينا فارما	127,442.0	175,307.0	47,865.0	27.3
12	شركة فارما اوفر سيز	107,737.0	149,157.0	41,420.0	27.8

Product Rank:: 84, Product Name: CO-TAREG 80/12.5MG 14 F.C.TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
1	ابن سينا فارما	122,070.8	176,222.0	54,151.2	30.7
10	شركة فارما اوفر سيز	108,730.2	157,478.0	48,747.8	31.0

Supplier Name: ['شركة فارما اوفر سيز'], Supplier Rank: 2

Product Rank:: 5, Product Name: ENTRESTO 50 MG (24/26 MG) 28 F.C. TA BS.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
11	شركة فارما اوفر سيز	643,943.0	757,580.0	113,637.0	15.0
9	شركة جلوبال فارما	577,693.9	697,005.0	119,311.1	17.1
7	شركة الاهرام شبين الكوم	345,412.4	415,607.0	70,194.6	16.9

Product Rank:: 15, Product Name: SINGULAIR 10MG 14 F.C. TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
10	شركة فارما اوفر سيز	342,754.7	496,704.0	153,949.3	31.0
9	شركة سوفيكو فارم	109,527.5	159,744.0	50,216.5	31.4

Product Rank:: 16, Product Name: ELIQUIS 2.5 MG 20 F.C. TABS.

Supplier		total after discount tax 2	Total	total discount 2	Discount %
13	شركة فارما اوفر سيز	339,917.8	412,098.0	72,180.2	17.5
1	ابن سينا فارما	333,074.2	406,188.0	73,113.8	18.0
9	شركة الاهرام شبين الكوم	100,533.7	128,702.0	28,168.3	21.9

Product Rank:: 20, Product Name: THIOTACID COMPOUND 600 MG 30 F.C.CA PLETS

Supplier		total after discount tax 2	Total	total discount 2	Discount %
14	شركة فارما اوفر سيز	316,839.1	458,910.0	142,070.9	31.0
2	ابن سينا فارما	203,296.6	294,927.0	91,630.4	31.1
10	شركة الاهرام شبين الكوم	174,055.8	253,098.0	79,042.2	31.2

Product Rank:: 21, Product Name: GLIPTUS PLUS 50/1000MG 30 TABLETS

Supplier		total after discount tax 2	Total	total discount 2	Discount %
16	شركة فارما اوفر سيز	315,911.5	458,691.0	142,779.5	31.1
11	شركة الاهرام شبين الكوم	185,692.7	269,841.0	84,148.3	31.2
3	ابن سينا فارما	133,347.6	195,009.0	61,661.4	31.6

Product Rank:: 22, Product Name: XGEVA 120 MG VIAL S.C. INJ.

Supplier		total after discount tax 2	Total	total discount 2	Discount %
2	شركة فارما اوفر سيز	308,732.6	358,636.0	49,903.4	13.9
1	ابن سينا فارما	132,008.4	155,304.0	23,295.6	15.0

Product Rank:: 24, Product Name: GENUPHIL ORIGINAL 50 F.C. TAB.

Supplier		total after discount tax 2	Total	total discount 2	Discount %
14	شركة فارما اوفر سيز	293,926.2	342,235.0	48,308.9	14.1
10	شركة الاهرام شبين الكوم	192,470.7	275,460.0	82,989.3	30.1
2	ابن سينا فارما	159,957.3	197,519.6	37,562.3	19.0

Product Rank:: 28, Product Name: TAMSULIN 0.4MG 28 CAPS.

Supplier	total after discount tax 2	Total	total discount 2	Discount %
16 شركة فارما اوفر سيز	255,768.1	347,293.0	91,524.9	26.4
11 شركة الاهرام شبين الكوم	177,342.0	244,218.0	66,876.0	27.4

Product Rank:: 31, Product Name: APIDRA 100I.U./ML 5 PREFILLED PEN

Supplier	total after discount tax 2	Total	total discount 2	Discount %
9 شركة فارما اوفر سيز	216,537.3	259,402.5	42,865.2	16.5
7 شركة جلوبال فارما	154,130.4	185,738.8	31,608.4	17.0
5 شركة الاهرام شبين الكوم	102,553.3	125,023.5	22,470.2	18.0

Product Rank:: 38, Product Name: TRULICITY 1.5MG/0.5ML 4 PRE-FILLED PENS

Supplier	total after discount tax 2	Total	total discount 2	Discount %
4 شركة فارما اوفر سيز	190,421.8	222,250.0	31,828.2	14.3
0 AKHNATON COMPANY	135,402.0	158,500.0	23,098.0	14.6

Product Rank:: 39, Product Name: METHYLTECHNO 1000MG 30TAB

Supplier	total after discount tax 2	Total	total discount 2	Discount %
12 شركة فارما اوفر سيز	189,947.0	258,075.0	68,128.0	26.4
9 شركة الاهرام شبين الكوم	142,956.5	202,965.0	60,008.6	29.6

Product Rank:: 41, Product Name: RECORMON 4000 I.U. 6 PREFILLED SYRINE

Supplier	total after discount tax 2	Total	total discount 2	Discount %
3 شركة فارما اوفر سيز	181,129.9	213,094.0	31,964.1	15.0
0 ابن سينا فارما	172,503.5	203,376.0	30,872.5	15.2

Product Rank:: 69, Product Name: ZOLADEX DEPOT 10.8MG PREF. SYRINGE

Supplier	total after discount tax 2	Total	total discount 2	Discount %
9 شركة فارما اوفر سيز	139,154.4	158,130.0	18,975.6	12.0
1 ابن سينا فارما	125,207.9	145,071.0	19,863.2	13.7

Product Rank:: 75, Product Name: CRESTOR 5MG 28 F.C. TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
9	شركة فارما اوفر سيز	118,643.4	161,568.0	42,924.6	26.6
1	ابن سينا فارما	110,068.3	150,328.0	40,259.7	26.8

Supplier Name: ['شركة جلوبال فارما'], Supplier Rank: 3

Product Rank:: 2, Product Name: ENTRESTO 100 MG (49/51 MG) 28 F.C. T ABS.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
8	شركة جلوبال فارما	467,445.7	564,439.0	96,993.3	17.2
6	شركة الاهرام شبين الكوم	135,695.3	164,280.0	28,584.7	17.4

Product Rank:: 7, Product Name: OPTAMINNESS 150 TABS.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
12	شركة جلوبال فارما	262,017.8	375,885.0	113,867.2	30.3
9	شركة الاهرام شبين الكوم	189,991.4	284,512.5	94,521.1	33.2

Supplier Name: ['شركة الاهرام شبين الكوم'], Supplier Rank: 4

Product Rank:: 1, Product Name: PLAVIX 75MG 28 FILM COATED TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
10	شركة الاهرام شبين الكوم	1,480,685.9	1,864,329.0	383,643.1	20.6
7	الفاروق	831,407.7	1,067,911.0	236,503.3	22.1
12	شركة جلوبال فارما	220,519.7	297,758.0	77,238.3	25.9

Product Rank:: 2, Product Name: CONTROLLOC 40MG 14 GASTRORESISTANT TA B.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
10	شركة الاهرام شبين الكوم	563,492.0	698,978.0	135,486.0	19.4
7	الفاروق	165,571.5	210,878.0	45,306.5	21.5

Product Rank:: 5, Product Name: EXFORGE HCT 10/160/25MG 14 F.C. TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
11	شركة الاهرام شبين الكوم	395,174.5	467,470.0	72,295.5	15.5
3	ابن سينا فارما	255,408.7	311,474.0	56,065.3	18.0
15	شركة فارما اوفر سيز	197,624.9	241,006.0	43,381.1	18.0

Product Rank:: 8, Product Name: TRAJENTA 5 MG 30 TABS.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
9	شركة الاهرام شبين الكوم	280,190.0	339,141.0	58,951.0	17.4
2	ابن سينا فارما	139,466.6	169,623.0	30,156.4	17.8

Product Rank:: 29, Product Name: CIPRAPRO 10MG 30 F.C. TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
4	شركة الاهرام شبين الكوم	130,837.7	207,516.0	76,678.3	37.0
8	شركة فارما اوفر سيز	103,396.8	164,598.0	61,201.2	37.2

Supplier Name: ['مخزن السعيد'], Supplier Rank: 5

Product Rank:: 7, Product Name: LENVIMA 4 MG 30 H.G. CAPS.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
0	مخزن السعيد	282,000.0	377,730.0	95,730.0	25.3
1	مخزن داوى	156,000.0	209,850.0	53,850.0	25.7

Product Rank:: 8, Product Name: STELARA 90 MG/ML PREF. SYRINGE FOR S.C.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
1	مخزن السعيد	274,998.3	325,512.0	50,513.7	15.5
0	Imported (Maadi)	162,168.1	196,231.0	34,062.9	17.4

Supplier Name: ['الفاروق'], Supplier Rank: 6

Product Rank:: 2, Product Name: OMEGA-3 PLUS 30 CAPS

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
12	الفاروق	332,769.9	387,495.0	54,725.1	14.1
15	شركة الاهرام شبين الكوم	246,061.4	312,471.0	66,409.6	21.3
17	شركة جلوبال فارما	138,915.0	175,320.0	36,405.0	20.8

Supplier Name: ['AUG PHARMA'], Supplier Rank: 7

Product Rank:: 2, Product Name: DIBAVALLY PLUS 50/1000 MG 28 TABS.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
1	AUG PHARMA	1,442,756.2	2,383,780.0	941,023.8	39.5
2	Samir Supplier	127,008.0	211,680.0	84,672.0	40.0

Product Rank:: 3, Product Name: ESMORAP 40MG 14 CAPS.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %
1	AUG PHARMA	548,421.4	921,908.0	373,486.6	40.5
3	ابن سينا فارما	158,674.6	272,250.0	113,575.4	41.7

Supplier Name: ['Wecare'], Supplier Rank: 8

Supplier Name: ['متنوعون المعادى'], Supplier Rank: 9

```
In [121... supplier_name = df_supplier_data['Supplier'][df_supplier_data['Supp
supplier_name
```

```
Out[121... array(['متنوعون المعادى'], dtype=object)
```

```
In [122... supplier_name = df_supplier_data['Supplier'][df_supplier_data['Supp
if supplier_name.size > 0:
    supplier_name = supplier_name[0]
if supplier_name is not None:
    for i , row in enumerate(rank_supplier[9]['Product name'][rank_
        product = (df_purchased[df_purchased['Product name'] == row
            .groupby('Supplier')
            .agg({
                'total after discount tax 2': 'sum',
                'Total': 'sum',
                'total discount 2': 'sum'
            })
            .reset_index()
            .sort_values(by='total after discount tax 2', ascen
product['Discount %'] = (product['total discount 2'] / prod
supplier_discount = product['Discount %'][product['Supplier
if supplier_discount.size > 0:
    supplier_discount = supplier_discount[0]
# Print the index and filtered DataFrame
if supplier_discount is not None:
    filtered_product = product[
        (product['total after discount tax 2'] > 100000
        (product['Discount %'] >= supplier_discount)&
        (product['Supplier'].iloc[0] == supplier_name)]
if len(filtered_product) > 1:
    print(f"{i}, Product Name: {row}")
    display(filtered_product) # Display the DataFrame
```

```
In [123... supplier_product_df
```

Out [123...]

	Supplier	Product name	Quantity	total after discount tax 2	Total	total discount %
0	AUG PHARMA	ZURCAL 40 MG 14 GASTRO RESISTANT TAB.	41,322.0	2,186,333.5	3,879,972.0	1,693,638.5
1	ابن سينا فارما	TRESIBA 100 I.U./ML FLEXTOUCH PRE-FILLED PEN	5,856.0	2,098,395.4	2,525,276.0	426,880.6
2	ابن سينا فارما	CRESTOR 20MG 28 F.C.TAB.	5,080.0	1,889,180.1	2,575,315.0	686,134.9
3	ابن سينا فارما	NOVORAPID 100 I.U./ML 5 FLEXPEN	1,970.0	1,657,809.9	1,951,587.0	293,777.0
4	شركة الاهرام شبيبن الكوم	PLAVIX 75MG 28 FILM COATED TAB.	7,539.0	1,480,685.9	1,864,329.0	383,643.1
...						
37929	متنوعون الهرم	FORXIGA 10 MG 28 TABS.	15.0	-13,398.4	-12,806.0	592.4
37930	مخزن داوى	AFINITOR 10MG 30 TABS	-1.0	-15,000.0	-19,100.0	-4,100.0
37931	شركة سوفيكيو فارم	JANUMET 50/850MG 14 F.C. TAB.	-194.0	-17,675.6	-21,534.0	-3,858.4
37932	ديفارت	KALOGENA 100 MG 10 SACHETS	-500.0	-45,000.0	-75,000.0	-30,000.0
37933	القصواع	DIBAVALLY PLUS 50/1000 MG 28 TABS.	-1,008.0	-63,504.0	-141,120.0	-77,616.0

37934 rows × 14 columns

In [124...]

```
for supplierrank in range(1,10):
    supplier_name = df_supplier_data['Supplier'][df_supplier_data['
print(f"Supplier Name: {supplier_name}, Supplier Rank: {supplie
print(" ")
```

```

if supplier_name.size > 0:
    supplier_name = supplier_name[0]
if supplier_name is not None:
    for i, row in enumerate(rank_supplier[supplierrank]['Product']):
        product = (df_purchased[df_purchased['Product name'] == row]
                    .groupby('Supplier')
                    .agg({
                        'total after discount tax 2': 'sum',
                        'Total': 'sum',
                        'total discount 2': 'sum'
                    })
                    .reset_index()
                    .sort_values(by='total after discount tax 2', ascending=False))
        product['Discount %'] = (product['total discount 2'] / product['Total']) * 100
        product = product.merge(supplier_product_df[supplier_product_df['Supplier'] == supplier_name])
        supplier_return = product['Return %'][product['Supplier'] == supplier_name]
        if supplier_return.size > 0:
            supplier_return = supplier_return[0]
        # Print the index and filtered DataFrame
        if supplier_return is not None:
            filtered_product = product[
                (product['total after discount tax 2'] > 10)
                & (product['Return %'] <= supplier_return) &
                # (product['Return %'] >= 1) &
                (product['Supplier'].iloc[0] == supplier_name)]
            if len(filtered_product) > 1:
                print(f"Product Rank:: {i + 1}, Product Name: {row}")
                display(filtered_product) # Display the DataFrame
            print(" ")
            print(" ")
            print(" ")
            print(" ")

```

Supplier Name: [ابن سينا فارما], Supplier Rank: 1

Product Rank:: 2, Product Name: CRESTOR 20MG 28 F.C.TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %	Return %
0	ابن سينا فارما	1,889,180.1	2,575,315.0	686,134.9	26.6	1.0
2	شركة الاهرام شبين الكوم	395,213.4	538,940.0	143,726.6	26.7	0.3

Product Rank:: 11, Product Name: ASPIRIN PROTECT 100 MG 30 GASTRO-RE
SISTANT TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %	Return %
0	ابن سينا فارما	652,842.6	797,238.0	144,395.4	18.1	0.7
1	شركة فارما اوفر سيز	325,260.5	405,378.0	80,117.5	19.8	0.1
2	شركة الاهرام شبيبن الکوم	281,944.3	391,590.0	109,645.7	28.0	0.1

Product Rank:: 12, Product Name: VASTAREL MR 35MG 30 F.C.TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %	Return %
0	ابن سينا فارما	586,737.4	850,200.0	263,462.6	31.0	0.8
4	الفاروق	153,936.2	232,250.0	78,313.8	33.7	0.1

Product Rank:: 19, Product Name: MILGA ADVANCE 30 F.C. TAB

	Supplier	total after discount tax 2	Total	total discount 2	Discount %	Return %
0	ابن سينا فارما	425,925.2	580,552.5	154,627.3	26.6	2.2
2	الفاروق	228,355.0	308,920.5	80,565.5	26.1	1.4

Product Rank:: 20, Product Name: DAPAGLIF 10 MG 14 F.C.TABS.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %	Return %
0	ابن سينا فارما	413,825.1	598,910.0	185,084.9	30.9	4.5
1	شركة فارما اوفر سيز	182,251.6	264,089.5	81,837.9	31.0	0.5

Product Rank:: 21, Product Name: PEPON PLUS 20 CAPS.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %	Return %
0	ابن سينا فارما	407,190.0	578,260.0	171,070.0	29.6	14.3
1	شركة فارما اوفر سيز	173,108.1	250,420.0	77,311.9	30.9	6.1

Product Rank:: 24, Product Name: CONCOR 5 MG 30 F.C. TABS

	Supplier	total after discount tax 2	Total	total discount 2	Discount %	Return %
0	ابن سينا فارما	371,230.8	490,815.0	119,584.2	24.4	5.2
1	شركة فارما اوفر سيز	170,929.1	222,756.0	51,826.9	23.3	2.5

Product Rank:: 38, Product Name: TAREG 80MG 14 F.C.TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %	Return %
0	ابن سينا فارما	259,209.3	352,272.8	93,063.5	26.4	5.4
1	شركة فارما اوفر سيز	151,115.3	205,777.0	54,661.7	26.6	0.4

Product Rank:: 39, Product Name: EXFORGE HCT 5/160/12.5MG 14 F.C. TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %	Return %
0	ابن سينا فارما	257,873.6	314,480.0	56,606.4	18.0	11.8
1	شركة الاهرام شبين الكوم	257,392.6	303,480.0	46,087.4	15.2	0.1
4	شركة جلوبال فارما	141,109.5	172,787.5	31,678.0	18.3	8.4

Product Rank:: 41, Product Name: CONCOR COR 2.5 MG 30 F.C. TABS

	Supplier	total after discount tax 2	Total	total discount 2	Discount %	Return %
0	ابن سينا فارما	245,974.6	333,456.0	87,481.4	26.2	9.2
1	شركة فارما اوفر سيز	104,925.3	142,303.5	37,378.2	26.3	7.7

Product Rank:: 58, Product Name: EXFORGE 10MG/160MG 14 F.C. TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %	Return %
0	ابن سينا فارما	175,909.6	254,874.0	78,964.4	31.0	5.0
1	شركة فارما اوفر سيز	147,676.8	214,018.0	66,341.2	31.0	0.1

Product Rank:: 63, Product Name: ATACAND PLUS 16/12.5 MG 14 TABS.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %	Return %
0	ابن سينا فارما	161,105.2	218,660.0	57,554.8	26.3	13.0
1	شركة فارما اوفر سيز	139,433.9	190,008.0	50,574.1	26.6	0.5

Supplier Name: ['شركة فارما اوفر سيز'], Supplier Rank: 2

Product Rank:: 5, Product Name: ENTRESTO 50 MG (24/26 MG) 28 F.C. TABS.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %	Return %
0	شركة فارما اوفر سيز	643,943.0	757,580.0	113,637.0	15.0	5.1
1	شركة جلوبال فارما	577,693.9	697,005.0	119,311.1	17.1	4.0

Product Rank:: 12, Product Name: DIAMICRON 60 MG 30 M.R. SCORED TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %	Return %
0	شركة فارما اوفر سيز	430,448.9	584,901.0	154,452.1	26.4	0.6
2	شركة الاهرام شبين الكوم	155,016.5	207,495.0	52,478.6	25.3	0.1
3	الفاروق	137,336.9	184,612.5	47,275.7	25.6	0.0
4	شركة جلوبال فارما	119,710.8	160,191.0	40,480.2	25.3	0.4

Product Rank:: 20, Product Name: THIOTACID COMPOUND 600 MG 30 F.C.CA PLETS

	Supplier	total after discount tax 2	Total	total discount 2	Discount %	Return %
0	شركة فارما اوفر سيز	316,839.1	458,910.0	142,070.9	31.0	3.2
1	ابن سينا فارما	203,296.6	294,927.0	91,630.4	31.1	2.6
2	شركة الاهرام شبين الكوم	174,055.8	253,098.0	79,042.2	31.2	2.7
3	شركة جلوبال فارما	116,175.9	160,084.5	43,908.6	27.4	1.8

Product Rank:: 24, Product Name: GENUPHIL ORIGINAL 50 F.C. TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %	Return %
0	شركة فارما اوفر سيز	293,926.2	342,235.0	48,308.9	14.1	8.0
1	شركة الاهرام شبين الكوم	192,470.7	275,460.0	82,989.3	30.1	2.8
2	ابن سينا فارما	159,957.3	197,519.6	37,562.3	19.0	1.2

Product Rank:: 28, Product Name: TAMSULIN 0.4MG 28 CAPS.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %	Return %
0	شركة فارما اوفر سيز	255,768.1	347,293.0	91,524.9	26.4	4.1
2	شركة الاهرام شبين الكوم	177,342.0	244,218.0	66,876.0	27.4	0.3

Product Rank:: 50, Product Name: CELLCEPT 500 MG 50 TAB.

	Supplier	total after discount tax 2	Total	total discount 2	Discount %	Return %
0	شركة فارما اوفر سيز	157,501.7	185,310.0	27,808.3	15.0	4.5
1	ابن سينا فارما	150,934.5	177,570.0	26,635.5	15.0	2.0

Supplier Name: ['شركة جلوبال فارما'], Supplier Rank: 3

Supplier Name: ['شركة الاهرام شبين الكوم'], Supplier Rank: 4

Supplier Name: ['مخزن السعيد'], Supplier Rank: 5

Supplier Name: ['الفاروق'], Supplier Rank: 6

Product Rank:: 2, Product Name: OMEGA-3 PLUS 30 CAPS

	Supplier	total after discount tax 2	Total	total discount 2	Discount %	Return %
0	الفاروق	332,769.9	387,495.0	54,725.1	14.1	15.9
2	شركة جلوبال فارما	138,915.0	175,320.0	36,405.0	20.8	1.0

Supplier Name: ['AUG PHARMA'], Supplier Rank: 7

Supplier Name: ['Wecare'], Supplier Rank: 8

Supplier Name: ['متعاونون المعادي'], Supplier Rank: 9

In [125... df[(df['Supplier'] == 'Wecare') & (df['Type'] == 'Return')]]

Out[125...]

	Date	Bill No	Store ID	Type	Store Name	Product id	Product name	Supplier
29672	2024-02-29	10273	4	Return	وياك صيدلية الهرم	21613	EMPACOZA 25 MG 30 TAB	Wecare
106463	2024-08-13	10294	5	Return	المخزن المركزي	23406	FREESTYLE LIBRE II	Wecare
126761	2024-09-22	10297	5	Return	المخزن المركزي	3083	EXFORGE 5MG/160MG 14 F.C. TAB.	Wecare
167011	2024-11-24	10302	5	Return	المخزن المركزي	10175	MAVILOR 5MG 30 TAB.	Wecare
167012	2024-11-24	10302	5	Return	المخزن المركزي	23406	FREESTYLE LIBRE II	Wecare

5 rows x 33 columns

In [126... df[df['Bill No']== '7814']]

Out[126...]

Date	Bill No	Store ID	Type	Store Name	Product id	Product name	Supplier	Category	Un Pric
0 rows × 33 columns									

In [127...]

```
# pd.set_option('display.max_rows', None)
df_purchased[df_purchased['Supplier'] == 'Wecare'].groupby(['month'])
```

Out[127...]

	month	Bill No	total after discount tax 2
0	1	10246	550.0
1	1	10253	107,026.6
2	1	10254	76,707.7
3	1	10255	81,721.0
4	1	10256	96,032.5
5	1	10257	36,540.0
6	1	10258	30,198.8
7	1	10259	37,680.1
8	1	10261	52,655.3
9	1	10262	49,197.3
10	1	10263	20,813.5
11	1	10264	100,337.0
12	1	10265	92,376.7
13	1	10266	5,394.9
14	2	10267	150,146.2
15	2	10268	73,596.2
16	2	10269	85,122.4
17	2	10270	62,645.8
18	2	10271	172,760.9
19	2	10272	109,208.2
20	2	10273	192,601.8
21	3	10273	4,495.5
22	3	10274	51,450.0
23	3	10275	22,000.0
24	3	10276	147,237.7

25	3	10277	64,186.7
26	3	10278	145,564.2
27	3	10279	7,500.0
28	4	10280	78,219.6
29	4	10281	189,089.3
30	4	10282	153,860.4
31	5	10284	46,333.2
32	5	10285	7,012.7
33	5	10286	9,207.6
34	5	10287	30,675.8
35	6	10288	197,628.6
36	6	10289	179,038.1
37	6	10290	77,946.5
38	7	10291	96,117.3
39	7	10292	4,582.7
40	7	10293	46,219.2
41	8	10294	203,931.9
42	8	10295	125,099.9
43	8	10296	95,379.4
44	9	10297	116,961.1
45	9	10298	48,853.8
46	9	10299	74,607.9
47	10	10300	225,762.5
48	10	10301	243,527.0
49	11	10302	59,160.9
50	12	10303	103,458.8
51	12	10304	181,805.1

In [128]: # pd.set_option('display.max_rows', None)

```
# df_purchased[df['Supplier'] == 'Wecare']
```

In [248]: df[df['Product id'] == 22434][['Date', 'new discount', 'Product name', 'Supplier', 'Unit Price']]

Out[248]:

	Date	new discount	Product name	Supplier	Unit Price

197630	2024-12-30	20.0	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	شركة فارما اوفر سيز	78.0
196868	2024-12-30	20.0	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	شركة فارما اوفر سيز	78.0
196817	2024-12-30	20.0	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	شركة فارما اوفر سيز	78.0
196313	2024-12-29	30.0	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	شركة جلوبال فارما	78.0
195354	2024-12-28	20.0	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	شركة فارما اوفر سيز	78.0
193348	2024-12-25	20.0	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	شركة فارما اوفر سيز	78.0
191926	2024-12-24	18.0	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	ابن سينا فارما	78.0
191799	2024-12-24	27.0	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	شركة جلوبال فارما	78.0
191119	2024-12-23	28.0	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	شركة جلوبال فارما	78.0
189033	2024-12-19	18.0	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	ابن سينا فارما	78.0
188837	2024-12-18	28.0	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	شركة جلوبال فارما	78.0
188338	2024-12-18	20.0	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	شركة فارما اوفر سيز	78.0
188287	2024-12-18	20.0	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	شركة فارما اوفر سيز	78.0
187601	2024-12-17	27.0	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	الشافعي فارما للتجارة والتوزيع	78.0
187703	2024-12-17	27.0	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	الفاتح	78.0
182615	2024-12-09	18.0	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	ابن سينا فارما	78.0

TAB.						
181611	2024-12-08	18.0	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	ابن سينا فارما	78.0	
181385	2024-12-08	20.0	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	متنوعون الهرم	78.0	
179562	2024-12-06	27.0	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	الشافعي فارما للتجارة والتوزيع	78.0	
179549	2024-12-06	27.0	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	الشافعي فارما للتجارة والتوزيع	78.0	
178769	2024-12-06	36.0	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	شركة الاهرام شبين الكوم	78.0	
178671	2024-12-06	20.0	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	شركة فارما اوفر سيز	78.0	
177868	2024-12-04	27.0	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	الشافعي فارما للتجارة والتوزيع	78.0	
177798	2024-12-04	31.0	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	شركة جلوبال فارما	78.0	
176034	2024-12-02	27.0	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	الشافعي فارما للتجارة والتوزيع	78.0	
175654	2024-12-02	28.0	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	الفاتح	78.0	
175336	2024-12-02	30.0	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	شركة جلوبال فارما	78.0	
173630	2024-12-01	30.0	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	شركة جلوبال فارما	78.0	
173217	2024-12-01	33.0	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	ميديكال فارما _ اونر نقدي	78.0	
174517	2024-12-01	20.0	ASPIRIN PROTECT 100 MG 30 GASTRO-RESISTANT TAB.	شركة فارما اوفر سيز	78.0	

In [130...]: product_df_purchased

Out [130...]

	Product name	Quantity	total after discount tax 2	Total	total discount 2	Discount %
3548	PLAVIX 75MG 28 FILM COATED TAB.	22,436.0	4,776,690.6	5,962,951.0	1,186,260.4	19.9
4456	TRESIBA 100 I.U./ML FLEXTOUCH PRE-FILLED PEN	10,214.0	4,043,688.9	4,794,089.0	750,400.1	15.7
1132	CRESTOR 20MG 28 F.C.TAB.	9,652.0	3,662,440.6	4,970,145.0	1,307,704.4	26.3
2545	LANTUS SOLOSTAR 100 I.U./ML 5 PENS	4,822.1	3,616,924.0	4,220,389.0	603,465.0	14.3
3240	NOVORAPID 100 I.U./ML 5 FLEXPEN	3,430.8	3,132,696.2	3,658,428.1	525,731.9	14.4
3066	Medicine organizer	-3.0	-280.8	-312.0	-31.2	10.0
4700	VITACID C PLUS 12 TABS	-72.0	-1,569.6	-2,518.5	-948.9	37.7
2400	JASMOLINE MASSAGE SPRAY 120 ML	-54.0	-3,078.0	-5,130.0	-2,052.0	40.0
2392	JANUMET 50/850MG 14 F.C. TAB.	-194.0	-17,675.6	-21,534.0	-3,858.4	17.9
2431	KALOGENA 100 MG 10 SACHETS	-500.0	-45,000.0	-75,000.0	-30,000.0	40.0

5001 rows x 7 columns

In [131...]

```
bills_products_df = bills_df.merge(df_purchased[['Product name','Bi
bills_products_df = bills_products_df[['Bill No','Product name','Su

bills_products_discount = bills_products_df.merge(product_df_purcha
bills_products_discount
```

Out [131...]

		Bill No	Product name	Supplier	Discount %
0	0000002284421		FORADIL 12 MCG 30 CAPS.+INHALER	شركة سوفيكوم فارم	25.2
1	132758		OSTEONATE 70MG 4 TAB.	الفاتح	28.5
2	132758		PROCTAN S.T. 30 CAPS	الفاتح	27.1
3	132758		KEROVIT 15 SOFT GELATIN CAPS.	الفاتح	22.0
4	132758		LISTERINE MOUTH WASH 250 ML	الفاتح	14.3
...
203110	SINUPRET 50 SUGAR COATED TAB		SOLVADOL ADULTS THROAT SPRAY 30 ML	متنوعون المعادي	6.6
203111	SINUPRET 50 SUGAR COATED TAB		SINUPRET 50 SUGAR COATED TAB	متنوعون المعادي	16.7
203112	TIRATAM 100MG/ML ORAL SOLUTION 120 ML		TIRATAM 100MG/ML ORAL SOLUTION 120 ML	مخزن جلوبال مصطفى	27.1
203113	limitless prenatal Max 30 soft gel cap		limitless prenatal Max 30 soft gel cap	متنوعون الهرم	18.9
203114	limitless prenatal Max 30 soft gel cap		LELIPEL 4MG 14 ORAL GRANULES IN SACHETS	متنوعون الهرم	20.5

203115 rows × 4 columns

In [132...]

```
bills_products = df_purchased.groupby(['Bill No', 'Supplier', 'Product name'])['Total'].sum()
bills_products['Discount % / item'] = bills_products['total discount'] / bills_products['Total']
bills_products = bills_products.merge(bills_products_discount[['Bill No', 'Supplier', 'Discount % / item']])
bills_products['opportunity cost'] = bills_products['Total'] * (bills_products['Discount % / item'] - 1)
bills_products
```

Out [132...]

		Bill No	Supplier	Product name	total after discount tax 2	Total	disc
0	0000002284421		شركة سوفيكت فارم	FORADIL 12 MCG 30 CAPS.+INHALER	50,889.2	62,060.0	11,170.80
1	132758	الفاتح		KEROVIT 15 SOFT GELATIN CAPS.	123.0	150.0	
2	132758	الفاتح		LISTERINE MOUTH WASH 250 ML	81.3	98.0	
3	132758	الفاتح		OSTEONATE 70MG 4 TAB.	30.4	40.0	
4	132758	الفاتح		PROCTAN S.T. 30 CAPS	43.2	60.0	
...
206552	SINUPRET 50 SUGAR COATED TAB		متتنوعون المعادي	SINUPRET 50 SUGAR COATED TAB	143.2	154.0	
206553	SINUPRET 50 SUGAR COATED TAB		متتنوعون المعادي	SOLVADOL ADULTS THROAT SPRAY 30 ML	302.3	325.0	
206554	TIRATAM 100MG/ML ORAL SOLUTION 120 ML		مخزن جلوبال مصطفى	TIRATAM 100MG/ML ORAL SOLUTION 120 ML	-66.7	-78.5	
206555	limitless prenatal Max 30 soft gel cap		متتنوعون الهرم	LELIPEL 4MG 14 ORAL GRANULES IN SACHETS	-88.3	-93.0	
206556	limitless prenatal Max 30 soft gel cap		متتنوعون الهرم	limitless prenatal Max 30 soft gel cap	-598.5	-630.0	

206557 rows × 9 columns

Vendors Metrics

Top 10 vendors

In [133...]: top_suppliers = df_purchased.groupby('Supplier').agg({'total after discount tax 2': 'sum'})

```
'Total' : 'su
          'total discou
top_suppliers['Discount %'] = top_suppliers['total discount 2'] / t
# top_suppliers['total public without tax'] = top_suppliers['total
top_suppliers_4M = top_suppliers[top_suppliers['total after discoun
top_suppliers_4M

top_suppliers_4M = top_suppliers_4M.rename(columns={
    'total after discount tax 2': 'Total Purchased',
    'total discount 2': 'Total Discount'
})

def format_compact_number(x):
    if pd.isna(x):
        return ''
    if x >= 1_000_000_000:
        return f'{x / 1_000_000_000:.1f} B'
    elif x >= 1_000_000:
        return f'{x / 1_000_000:.1f} M'
    elif x >= 1_000:
        return f'{x / 1_000:.1f} K'
    else:
        return f'{x:.0f}'

import matplotlib.pyplot as plt
from matplotlib.table import Table
from matplotlib.ticker import FuncFormatter
import pandas as pd
import arabic_reshaper
from bidi.algorithm import get_display

# Number formatting
def format_compact_number(x):
    if pd.isna(x):
        return ''
    if x >= 1_000_000_000:
        return f'{x / 1_000_000_000:.1f} B'
    elif x >= 1_000_000:
        return f'{x / 1_000_000:.1f} M'
    elif x >= 1_000:
        return f'{x / 1_000:.1f} K'
    else:
        return f'{x:.0f}'

def save_table_as_image(df, filename='table.png', dpi=200, font_fam
import matplotlib
matplotlib.rcParams['font.family'] = font_family # Set your fo

df_formatted = df.copy()

# Reshape Arabic for supplier column if it exists
if 'Supplier' in df_formatted.columns:
    df_formatted['Supplier'] = df_formatted['Supplier'].apply(r
```

```

# Format numeric columns
for col in df_formatted.select_dtypes(include=['number']).columns:
    if col == 'Discount %':
        df_formatted[col] = df_formatted[col].apply(lambda x: f'{x:.2%}')
    else:
        df_formatted[col] = df_formatted[col].apply(format_compact)

fig, ax = plt.subplots(figsize=(14, len(df_formatted) * 0.6))
ax.axis('off')

table = ax.table(cellText=df_formatted.values,
                  colLabels=df_formatted.columns,
                  cellLoc='center',
                  loc='center')

table.auto_set_font_size(False)
table.set_fontsize(10)
table.scale(1, 1.5)

for col_idx in range(len(df_formatted.columns)):
    cell = table[0, col_idx]
    cell.set_text_props(weight='bold', color='white')
    cell.set_facecolor('#4B72B0')

for row in range(1, len(df_formatted) + 1):
    for col in range(len(df_formatted.columns)):
        cell = table[row, col]
        cell.set_facecolor('#f2f2f2' if row % 2 == 0 else 'white')

plt.tight_layout()
plt.savefig(filename, dpi=dpi, bbox_inches='tight')
plt.close()

def reshape_arabic_text(text):
    if pd.isna(text):
        return ''
    try:
        reshaped_text = arabic_reshaper.reshape(text)
        bidi_text = get_display(reshaped_text)
        return bidi_text
    except Exception:
        return text # fallback if reshaping fails

save_table_as_image(top_suppliers_4M, 'top_suppliers.png')

```

Top Vendors by Purchasing Value

In [134...]

```

import matplotlib

# Function to format numbers with K, M, B suffixes
def format_compact(value, _): # <-- fixed: added second arg for FutureWarning
    if value >= 1_000_000_000:
        return f'{value/1_000_000_000:.1f} B'
    elif value >= 1_000_000:

```

```
        return f'{value/1_000_000:.1f} M'
    elif value >= 1_000:
        return f'{value/1_000:.1f} K'
    else:
        return f'{value:.0f}'

# Create the barplot
ax = sns.barplot(data=top_suppliers_4M, y='Supplier', x='Total Purchased')

# Set compact number formatting for x-axis
ax.xaxis.set_major_formatter(FuncFormatter(format_compact))

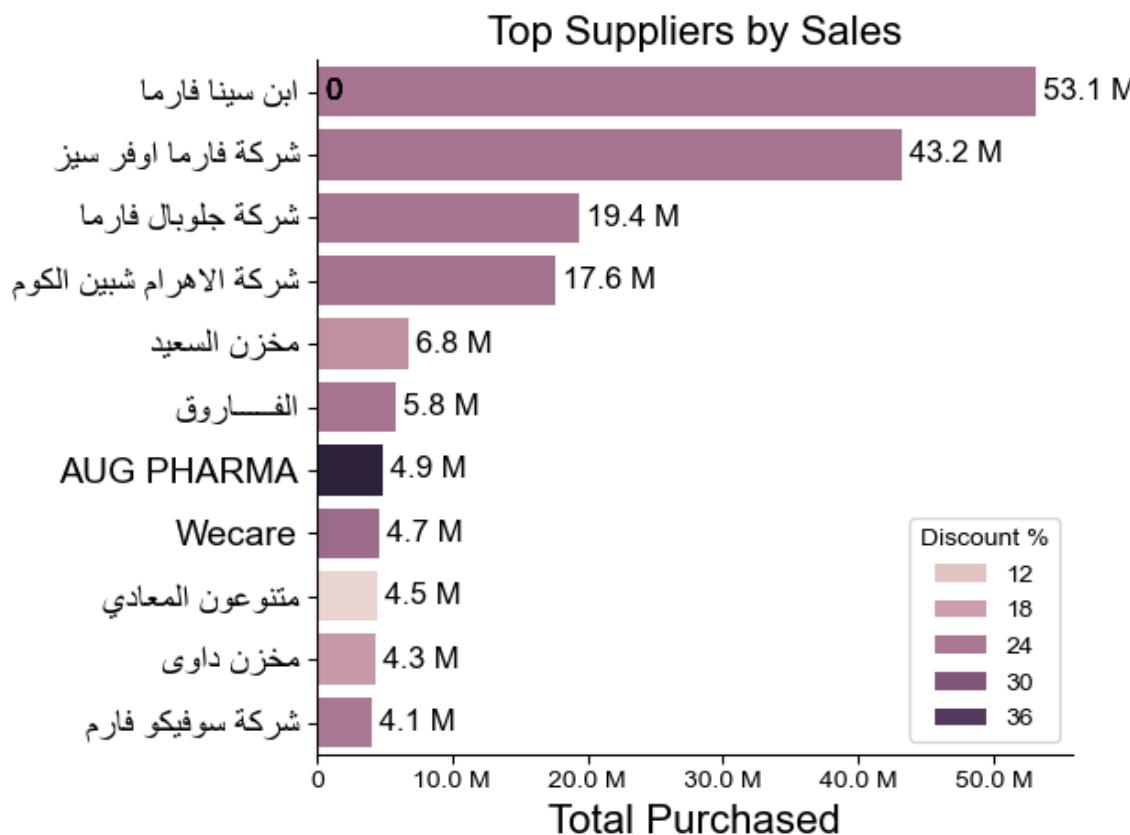
matplotlib.rcParams['font.family'] = 'Arial' # Change to 'Amiri' or 'Times New Roman'

# Reshape Arabic supplier names if needed
tick_labels = ax.get_yticklabels()
formatted_labels = [reshape_arabic(label.get_text()) for label in tick_labels]
plt.yticks(ticks=ax.get_yticks(), labels=formatted_labels, fontsize=14)

# Add formatted labels outside bars
for bar in ax.patches:
    width = bar.get_width()
    y = bar.get_y() + bar.get_height() / 2
    label = format_compact(width, None)
    ax.text(width + (max(ax.get_xlim()) * 0.01), y, label, va='center')

sns.despine()
plt.title('Top Suppliers by Sales', fontsize=16)
plt.xlabel('Total Purchased', fontsize=16)
plt.ylabel('')
plt.tight_layout()
plt.savefig('Vendor Sales.jpg', bbox_inches = 'tight', dpi = 200)

plt.show()
```



```
In [135...]: top_suppliers_4M['Total Purchased'].sum()
```

```
Out[135...]: 168528147.5
```

```
In [136...]: df['total after discount tax 2'].sum()
```

```
Out[136...]: 202424666.05273923
```

Top Vendors by Total Discount

```
# Function to format numbers with K, M, B suffixes
def format_compact(value, _=None):
    if value >= 1_000_000_000:
        return f'{value/1_000_000_000:.1f} B'
    elif value >= 1_000_000:
        return f'{value/1_000_000:.0f} M'
    elif value >= 1_000:
        return f'{value/1_000:.1f} K'
    else:
        return f'{value:.1f}'

# Create the barplot
ax = sns.barplot(data=top_suppliers_4M.sort_values(by = 'Discount %', ascending=False))

# Set compact number formatting for x-axis
ax.xaxis.set_major_formatter(FuncFormatter(format_compact))

# Reshape Arabic supplier names if needed
top_suppliers_4M['Supplier'] = top_suppliers_4M['Supplier'].str.replace(' ', '_')
```

```

    tick_labels = ax.get_yticklabels()
    formatted_labels = [reshape_arabic(label.get_text()) for label in t
    plt.yticks(ticks=ax.get_yticks(), labels=formatted_labels, fontsize=12)

    for bar in ax.patches:
        width = bar.get_width()
        y = bar.get_y() + bar.get_height() / 2

        # Skip ghost bars (like 0-width or 0-height)
        if width <= 0 or bar.get_height() <= 0:
            continue

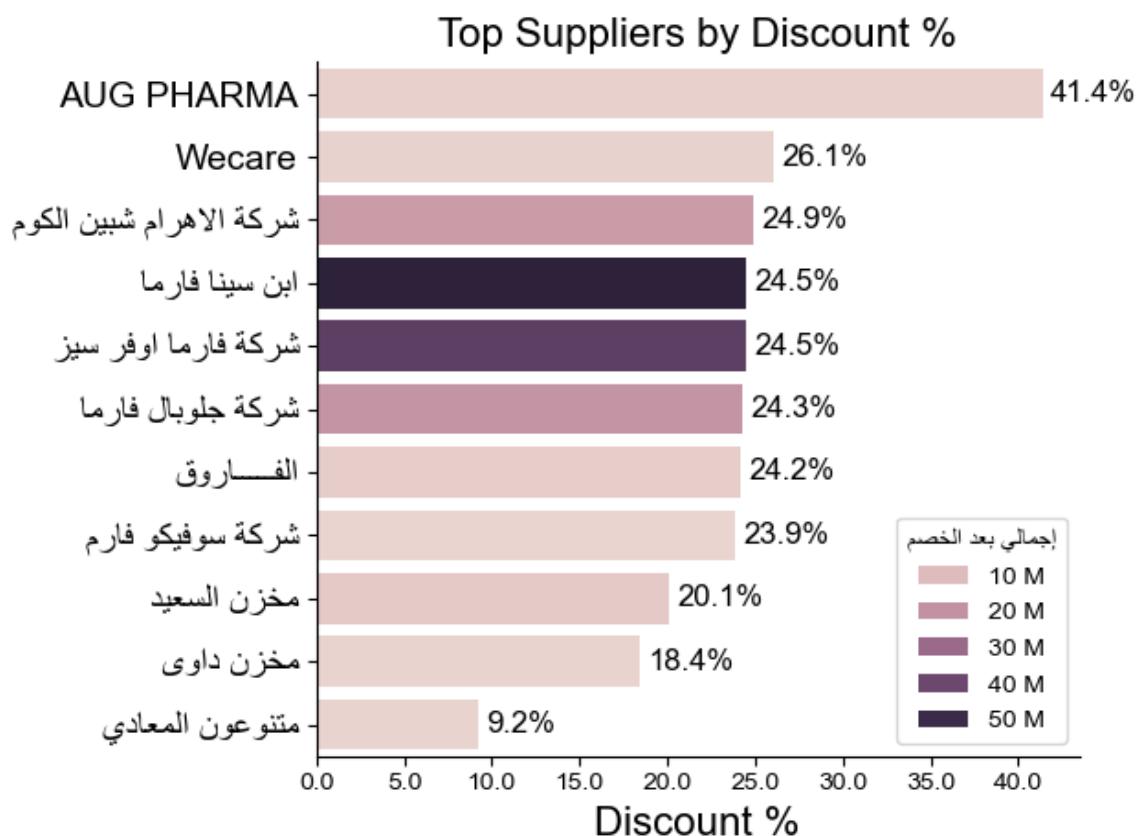
        label = f"{format_compact(width, None)}%" # Append percent sign
        ax.text(width + (max(ax.get_xlim()) * 0.01), y, label, va='center')

    # Reformat hue legend labels
    handles, labels = ax.get_legend_handles_labels()
    formatted_labels = [format_compact(float(label)) for label in labels]
    ax.legend(handles, formatted_labels, title=reshape_arabic("عدد الخصم"))

    sns.despine()
    plt.title('Top Suppliers by Discount %', fontsize=16)
    plt.xlabel('Discount %', fontsize=16)
    plt.ylabel('')
    plt.tight_layout()
    plt.savefig('Vendor Discount.jpg', bbox_inches = 'tight', dpi = 200)

    plt.show()

```



Discount vs. Purchased Value Matrix

In [138...]

```
# Function for compact number formatting
def format_compact(value, _=None):
    if value >= 1_000_000_000:
        return f'{value/1_000_000_000:.1f} B'
    elif value >= 1_000_000:
        return f'{value/1_000_000:.1f} M'
    elif value >= 1_000:
        return f'{value/1_000:.1f} K'
    else:
        return f'{value:.0f}'

# Filter top suppliers
filtered = top_suppliers[(top_suppliers['total after discount tax 2'] >= 1000000) & (top_suppliers['Supplier'] != 'Total')]

# Create scatterplot with larger points
plt.figure(figsize=(12, 7))
ax = sns.scatterplot(
    data=filtered,
    y='total after discount tax 2',
    x='Discount %',
    s=300, # size of points
    color='#2c7bb6' # optional: change point color
)
import matplotlib.patches as patches

# Get current axis limits
xlim = ax.get_xlim()
ylim = ax.get_ylim()

# Define rectangle dimensions (e.g., top 25% of both axes)
width = (xlim[1] - xlim[0]) * 0.5
height = (ylim[1] - ylim[0]) * 0.5

x_start = xlim[1] - width
y_start = ylim[1] - height

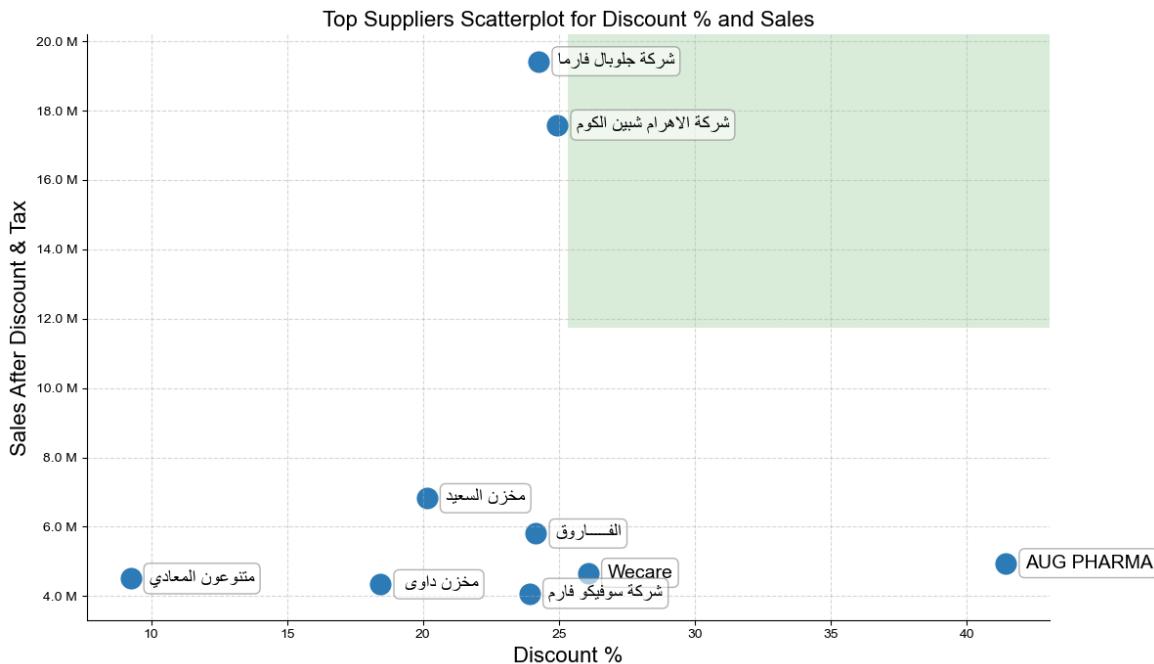
# Add translucent rectangle
rect = patches.Rectangle((x_start, y_start), width, height,
                        linewidth=0, edgecolor=None, facecolor='green', alpha=0.2)
ax.add_patch(rect)
# Format y-axis
ax.yaxis.set_major_formatter(FuncFormatter(format_compact))

# Label each point with clearer text
for _, row in filtered.iterrows():
    supplier_name = reshape_arabic(row['Supplier']) # if Arabic replace with English name
    plt.text(
        row['Discount %'] + 0.7, # horizontal offset
        row['total after discount tax 2'],
        supplier_name,
        fontsize=14,
        va='center',
        ha='left',
        bbox=dict(boxstyle="round,pad=0.3", fc="white", alpha=0.6),
    )
sns.despine()
```

```

plt.title('Top Suppliers Scatterplot for Discount % and Sales', fontweight='bold')
plt.xlabel('Discount %', fontsize=16)
plt.ylabel('Sales After Discount & Tax', fontsize=16)
plt.tight_layout()
plt.grid(linestyle='--', alpha=0.5)
plt.savefig('Vendor scatterplot.jpg', bbox_inches = 'tight', dpi =
plt.show()

```



In [139...]

```

# Function for compact number formatting
def format_compact(value, _=None):
    if value >= 1_000_000_000:
        return f'{value/1_000_000_000:.1f} B'
    elif value >= 1_000_000:
        return f'{value/1_000_000:.1f} M'
    elif value >= 1_000:
        return f'{value/1_000:.1f} K'
    else:
        return f'{value:.0f}'

# Filter top suppliers
filtered = top_suppliers[top_suppliers['total after discount tax 2' > 0]]
# Create scatterplot with larger points
plt.figure(figsize=(12, 7))
ax = sns.scatterplot(
    data=filtered,
    y='total after discount tax 2',
    x='Discount %',
    s=300, # size of points
    color='#2c7bb6' # optional: change point color
)

# Format y-axis
ax.yaxis.set_major_formatter(FuncFormatter(format_compact))

from adjustText import adjust_text

```

```
texts = []

for _, row in filtered.iterrows():
    supplier_name = reshape_arabic(row['Supplier'])
    txt = plt.text(
        row['Discount %'] + 0.7,
        row['total after discount tax 2'],
        supplier_name,
        fontsize=14,
        va='center',
        ha='left',
        bbox=dict(boxstyle="round, pad=0.3", fc="white", alpha=0.6,
    )
    texts.append(txt)

adjust_text(texts, ax=ax, arrowprops=dict(arrowstyle='-', color='gr

import matplotlib.patches as patches

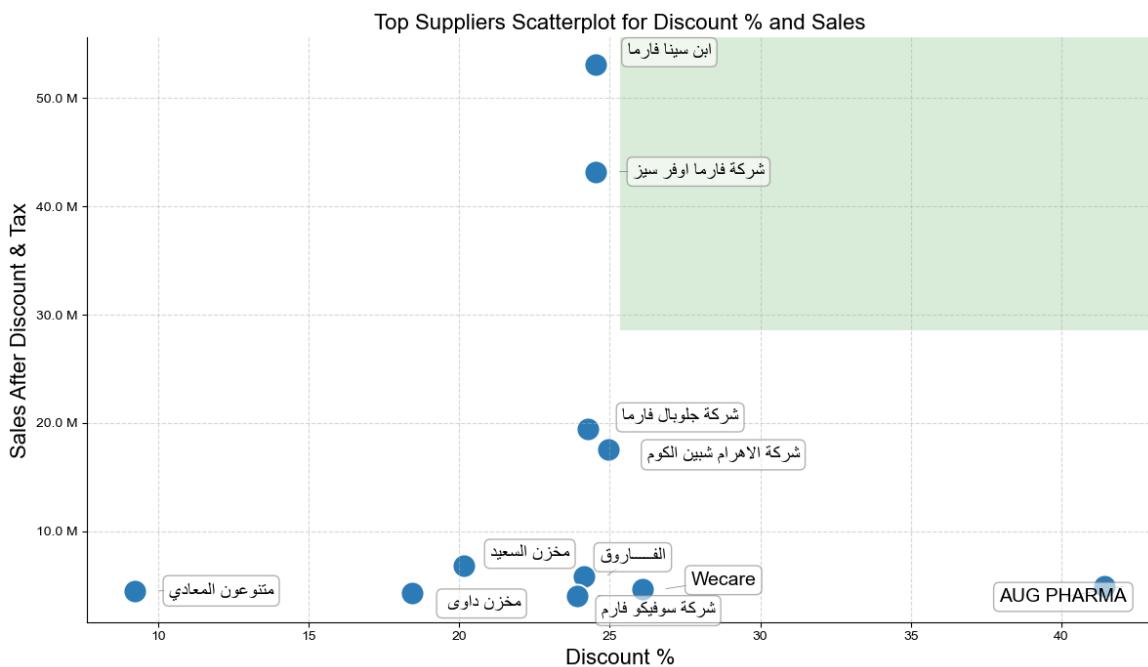
# Get current axis limits
xlim = ax.get_xlim()
ylim = ax.get_ylim()

# Define rectangle dimensions (e.g., top 25% of both axes)
width = (xlim[1] - xlim[0]) * 0.5
height = (ylim[1] - ylim[0]) * 0.5

x_start = xlim[1] - width
y_start = ylim[1] - height

# Add translucent rectangle
rect = patches.Rectangle((x_start, y_start), width, height,
                        linewidth=0, edgecolor=None, facecolor='gr
ax.add_patch(rect)

sns.despine()
plt.title('Top Suppliers Scatterplot for Discount % and Sales', font
plt.xlabel('Discount %', fontsize=16)
plt.ylabel('Sales After Discount & Tax', fontsize=16)
plt.tight_layout()
plt.grid(linestyle='--', alpha=0.5)
plt.savefig('Vendor scatterplot2.jpg', bbox_inches = 'tight', dpi =
plt.show()
```



In [140...]

```
# Set suppliers to highlight
highlighted_suppliers = {'ابن سينا فارما', 'الفاروق', 'شركة جلوبال فارما', 'شركة الاهرام شبين الكوم', 'الفاروق', 'WeCare', 'شركة سوفيكو فارم', 'مخزن داوى', 'مخزن السعيد', 'متبرعون المعادى'}
```

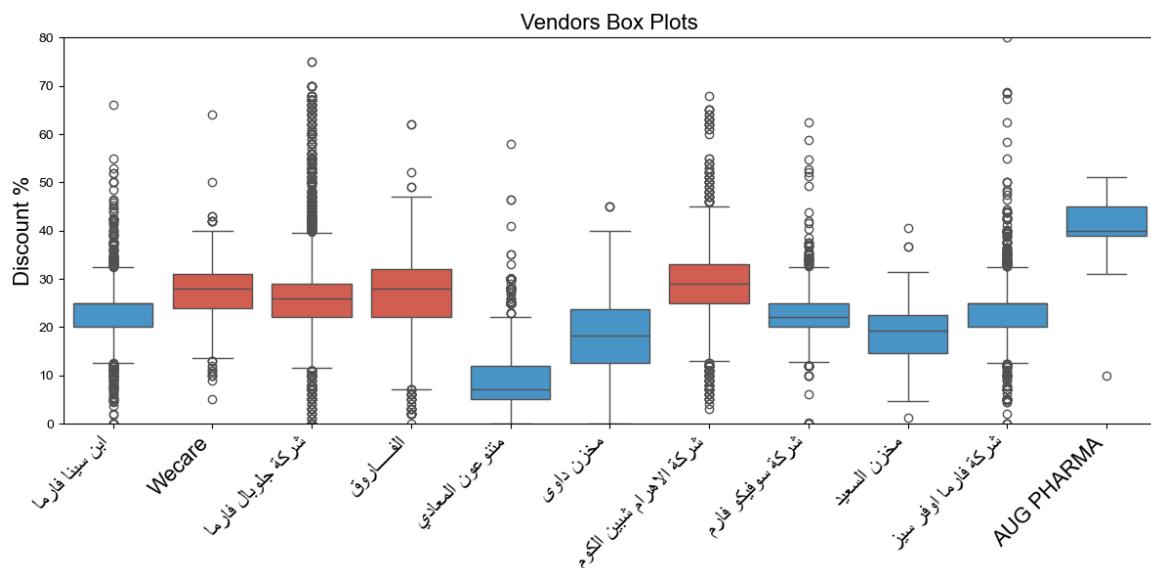
```
# Build a palette: use a different color for highlighted suppliers
all_suppliers = new_df['Supplier'].unique()
palette = {
    supplier: '#e74c3c' if supplier in highlighted_suppliers else '#f9a86a'
    for supplier in all_suppliers
}
```

```
# Plot
fig, ax = plt.subplots(figsize=(12, 6))
sns.boxplot(
    data=new_df,
    x='Supplier',
    y='new_discount',           # required to map colors via palette
    hue='Supplier',             # set hue same as x
    palette=palette,
    dodge=False,                # prevent duplicate boxes
    legend=False                # suppress duplicate legend
)

# Arabic reshaping
tick_labels = ax.get_xticklabels()
formatted_labels = [reshape_arabic(label.get_text()) for label in tick_labels]
plt.xticks(ticks=ax.get_xticks(), labels=formatted_labels, rotation=90)

plt.ylim(0, 80)
plt.title('Vendors Box Plots', fontsize=16)
plt.xlabel('')
plt.ylabel('Discount %', fontsize=16)
plt.tight_layout()
plt.savefig('Vendor boxplots.jpg', bbox_inches = 'tight', dpi = 200)

plt.show()
```



In [141]:

```
# Set suppliers to highlight
highlighted_suppliers = ['الكوم', 'شركة جلوبال فارما', 'الفقاروق']

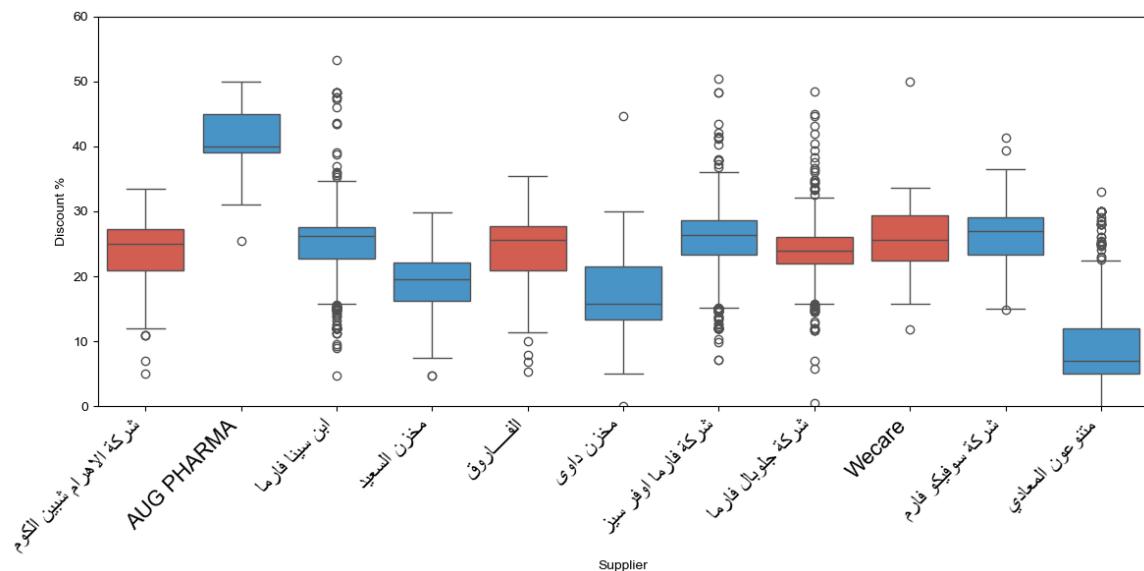
# Build a palette: use a different color for highlighted suppliers
all_suppliers = new_df['Supplier'].unique()
palette = {
    supplier: '#e74c3c' if supplier in highlighted_suppliers else '#e74c3c'
    for supplier in all_suppliers
}

box_plot_vendor = new_df.groupby(['Bill No', 'Supplier']).agg({'total': 'sum',
                                                               'Total' : 'sum',
                                                               'total discount' : 'sum'})
box_plot_vendor['Discount %'] = box_plot_vendor['total discount'] / box_plot_vendor['Total']

# Plot
fig, ax = plt.subplots(figsize=(12, 6))
sns.boxplot(
    data=box_plot_vendor,
    x='Supplier',
    y='Discount %',
    hue='Supplier',                      # required to map colors via palette
    palette=palette,
    dodge=False,                         # prevent duplicated boxes due to hue
    legend=False                          # suppress duplicate legend
)

# Arabic reshaping
tick_labels = ax.get_xticklabels()
formatted_labels = [reshape_arabic(label.get_text()) for label in tick_labels]
plt.xticks(ticks=ax.get_xticks(), labels=formatted_labels, rotation=45)

plt.ylim(0, 60)
plt.tight_layout()
plt.show()
```



Discount Metrics

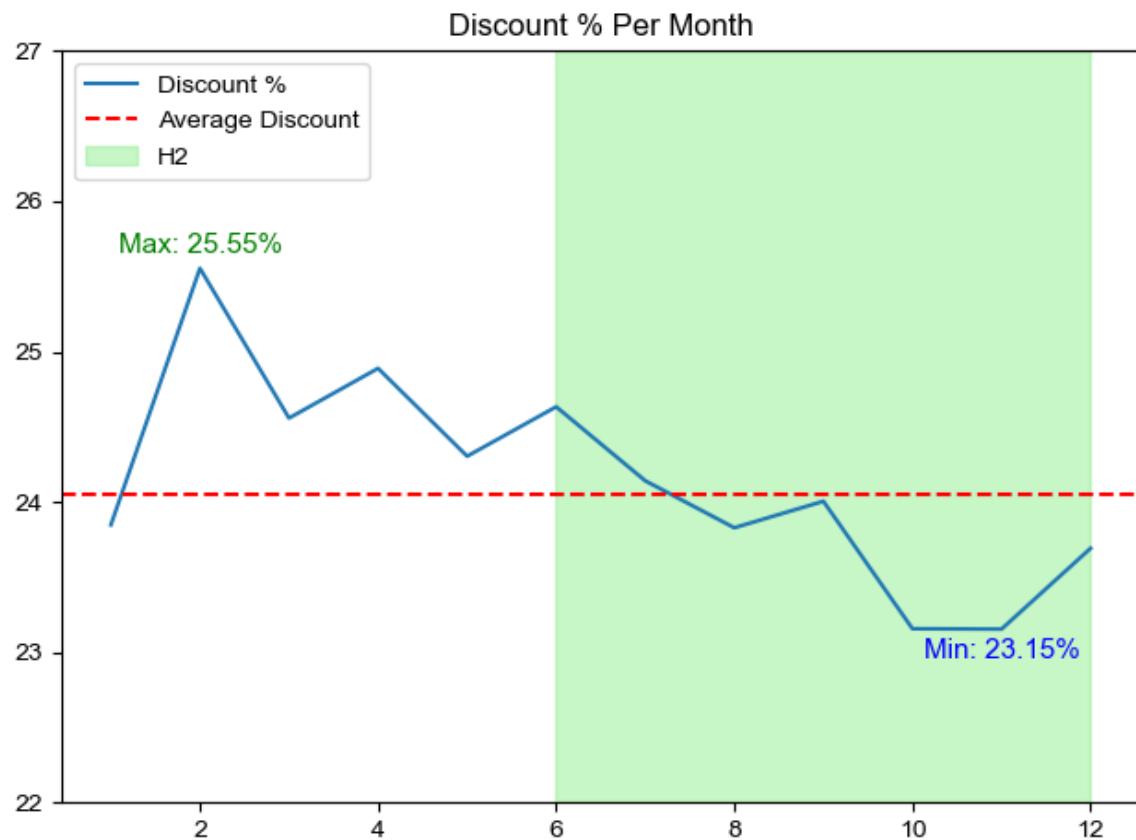
Discount Trend over Months

```
In [142]: df_discount_month = round(df_purchased.groupby('month').agg({'total_discount': 'sum'}), 2)
df_discount_month['Discount %'] = df_discount_month['total_discount'] / df_discount_month['total']
df_discount_month.plot(kind = 'line', x = 'month')
df_discount_month.set_title('Discount % Per Month')
df_discount_month.set_xlabel('')
df_discount_month.get_legend().remove()
df_discount_month.axhline(y = (df_discount_month['total_discount'].max() + df_discount_month['total_discount'].min()) / 2)
plt.ylim(22, 27)
plt.axvspan(6, 12, color='lightgreen', alpha=0.5, label='H2')

# Add min/max labels
max_idx = df_discount_month['Discount %'].idxmax()
min_idx = df_discount_month['Discount %'].idxmin()

max_month = df_discount_month.loc[max_idx, 'month']
max_value = df_discount_month.loc[max_idx, 'Discount %']
min_month = df_discount_month.loc[min_idx, 'month']
min_value = df_discount_month.loc[min_idx, 'Discount %']

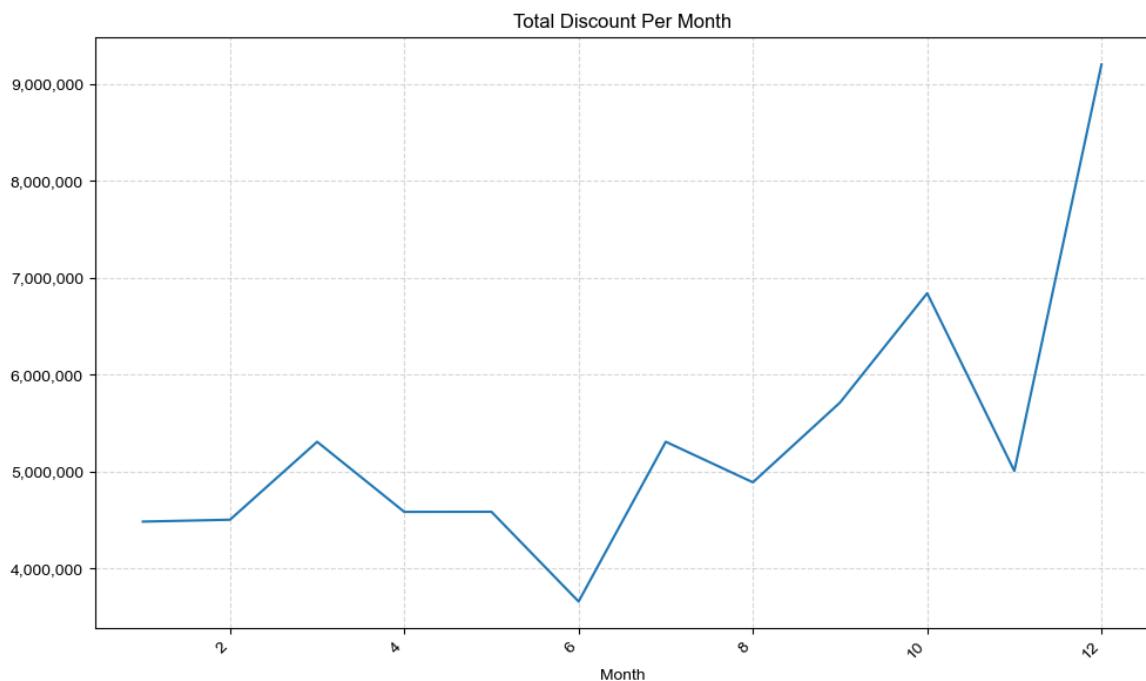
plt.text(max_month, max_value + 0.1, f"Max: {max_value:.2f}%", ha='center')
plt.text(min_month, min_value - 0.2, f"Min: {min_value:.2f}%", ha='center')
plt.legend()
plt.tight_layout()
plt.show()
```



Total Discount Value per Month

```
In [143]: plt.figure(figsize=(10, 6)) # Set the figure size here

sns.lineplot(data=df_discount_month, x = 'month', y = 'total discount')
plt.gca().yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.2f}'))
plt.xticks(rotation=45, ha='right')
plt.title('Total Discount Per Month')
plt.xlabel('Month')
plt.ylabel('')
plt.tight_layout()
plt.grid(linestyle='--', alpha=0.5)
plt.savefig('Total Discount Per Month.jpg', bbox_inches = 'tight',
plt.show()
```



In [144...]: df_discount_month

Out[144...]:

month	total after discount tax 2	Total	total discount 2	Discount %
0	14,321,341.0	18,806,032.0	4,484,691.0	23.8
1	13,122,461.0	17,626,735.0	4,504,274.0	25.6
2	16,306,628.0	21,614,573.0	5,307,945.0	24.6
3	13,837,457.0	18,422,510.0	4,585,053.0	24.9
4	14,283,965.0	18,870,100.0	4,586,136.0	24.3
5	11,199,789.0	14,860,276.0	3,660,488.0	24.6
6	16,681,366.0	21,989,756.0	5,308,390.0	24.1
7	15,632,065.0	20,521,993.0	4,889,928.0	23.8
8	18,087,569.0	23,800,723.0	5,713,154.0	24.0
9	22,699,714.0	29,540,042.0	6,840,328.0	23.2
10	16,619,802.0	21,627,328.0	5,007,525.0	23.2
11	29,632,508.0	38,832,374.0	9,199,866.0	23.7

In [145...]: max_discount = df_purchased.groupby('Product name')['new discount adds'].max()

Potential vs. Actual Savings

In [146...]:

```

potential = product_df_purchased.merge(max_discount, on='Product name')
potential['Potential Diff'] = (potential['new discount adds'] - potential['actual discount adds'])
potential['potential value'] = potential['new discount adds'] * potential['actual discount adds']
print(f"Total: {potential['Total'].sum():,.2f}")

```

```
print(f"Total Actual Savings: {(potential['Total'].sum() - potential['Actual Savings']):.2f}")
print(f"Total Potential Savings value: {potential['potential value'].sum():,.2f}")
print(f"Potential Difference: {potential['Potential Diff'].sum():,.2f}")

Total: 266,511,942.29
Total Actual Savings: 64,087,671.23
Total Potential Savings value: 661,278,686,800.89
Potential Difference: 661,214,599,142.42
```

In [147...]: potential['new discount adds'].sort_values(ascending = False).head()

```
Out[147...]: 1051    131,568,212.6
  151     97,401,460.7
  330     96,739,055.7
  853     82,412,762.3
  1349    15,989,531.6
  ...
  2026      60.0
  3238      60.0
  4001      59.0
  494       59.0
  1998      59.0
Name: new discount adds, Length: 65, dtype: float64
```

In [148...]: max_discount.head(1)

	Product name	new discount adds
0	0	25.0

In [149...]: max_discount.sort_values(by = 'new discount adds', ascending = False).head(1)

Out [149...]

	Product name	new discount adds
4910	c zinc 30 cap	131,568,212.6
3886	SANSO D3 10.000 I.U. 28TABS.	97,401,460.7
1249	DELTAVIT B12 1MG 30 SUBLINGUAL TAB.	96,739,055.7
4195	SULFAX (CMO FORMULA) MASSAGE CREAM 60 GM	82,412,762.3
676	C-RETARD+ZINC 20 CAPS.	15,989,531.6
957	CITAPRONEX 10MG 14 F.C.TAB.	510.0
3991	SILDEN 50 MG 4 F.C. TABS.	215.2
3453	PARACETAMOL CID 20 TABS	192.0
2582	LEPTICURE 400 MG 30 CAPS.	108.0
2012	GENUPHIL ORIGINAL 50 F.C. TAB.	108.0
373	ATOREZA 80/10MG 28 F.C. TAB.	108.0
2422	JUSTECHOL 20MG 28 F.C. TABS.	108.0
4277	TADARAXIA 20MG 2 FLIM COATED TABLETS	108.0
4958	vixaport 30 tap	108.0
3305	OMEZ 20MG 14 CAPS.	107.9
4314	TEBOFORTIN FORTE 80MG 30 F.C.TAB.	101.4
4278	TADATRONA 20 MG 2 TABS.	100.0
3284	OMEGA RX JELLY CANDY 60 PCS.	100.0
2696	LIPOSOM D3 10.000 I.U. 30 CAPS.	100.0
2313	INFINITY ANTI-AGING SPF 30 DAY SERUM 40 ML	100.0

In [150...]

product_df_purchased.head(2)

Out [150...]

	Product name	Quantity	total after discount tax 2	Total	total discount 2	Discount %
3548	PLAVIX 75MG 28 FILM COATED TAB.	22,436.0	4,776,690.6	5,962,951.0	1,186,260.4	19.9
4456	TRESIBA 100 I.U./ML FLEXTOUCH PRE-FILLED PEN	10,214.0	4,043,688.9	4,794,089.0	750,400.1	15.7

```
In [151... top_suppliers_monthly = df_purchased.groupby(['month', 'Supplier']).  
      'Total' : 'su  
      'total discou  
      'Bill No': 'n  
top_suppliers_monthly['Discount %'] = top_suppliers_monthly['total  
top_suppliers_monthly
```

Out[151...]

	month	Supplier	total after discount tax 2	Total	total discount 2	Bill No	Discount %
0	12	شركة فارما اوفر سيز	10,111,740.8	13,371,984.3	3,260,243.6	531	24.4
1	10	شركة فارما اوفر سيز	7,040,928.8	9,228,197.8	2,187,269.0	436	23.7
2	7	ابن سينا فارما	5,555,377.9	7,319,930.9	1,764,553.0	188	24.1
3	9	شركة فارما اوفر سيز	5,552,087.5	7,429,594.1	1,877,506.6	407	25.3
4	10	ابن سينا فارما	5,202,320.4	6,747,123.5	1,544,803.1	99	22.9
...
408	10	2020 محزن	-170.1	-189.0	-18.9	1	10.0
409	4	شركة الفنج شبين الكوم	-16,403.5	-23,878.0	-7,474.5	1	31.3
410	12	ميديكال فارما - اونر	-22,818.7	-31,239.4	-8,420.7	1	27.0
411	1	ديفارت	-45,000.0	-75,000.0	-30,000.0	1	40.0
412	1	القصـوـاء	-82,380.9	-183,793.0	-101,412.1	3	55.2

413 rows × 7 columns

```
In [152... top_suppliers_sales = df_supplier.loc[:10, 'Supplier']  
# top_suppliers_sales = df_supplier[df_supplier['Supplier'] != 'AUG'  
  
# top_suppliers_sales = top_suppliers_sales.to_list()  
  
# Pivot the data to make it suitable for a heatmap  
heatmap_data = top_suppliers_monthly[top_suppliers_monthly['Supplier'].  
                                         'Supplier' != 'AUG']  
heatmap_data.index = heatmap_data.index.map(reshape_arabic)  
  
# Set up the figure
```

```

plt.figure(figsize=(12, 8))

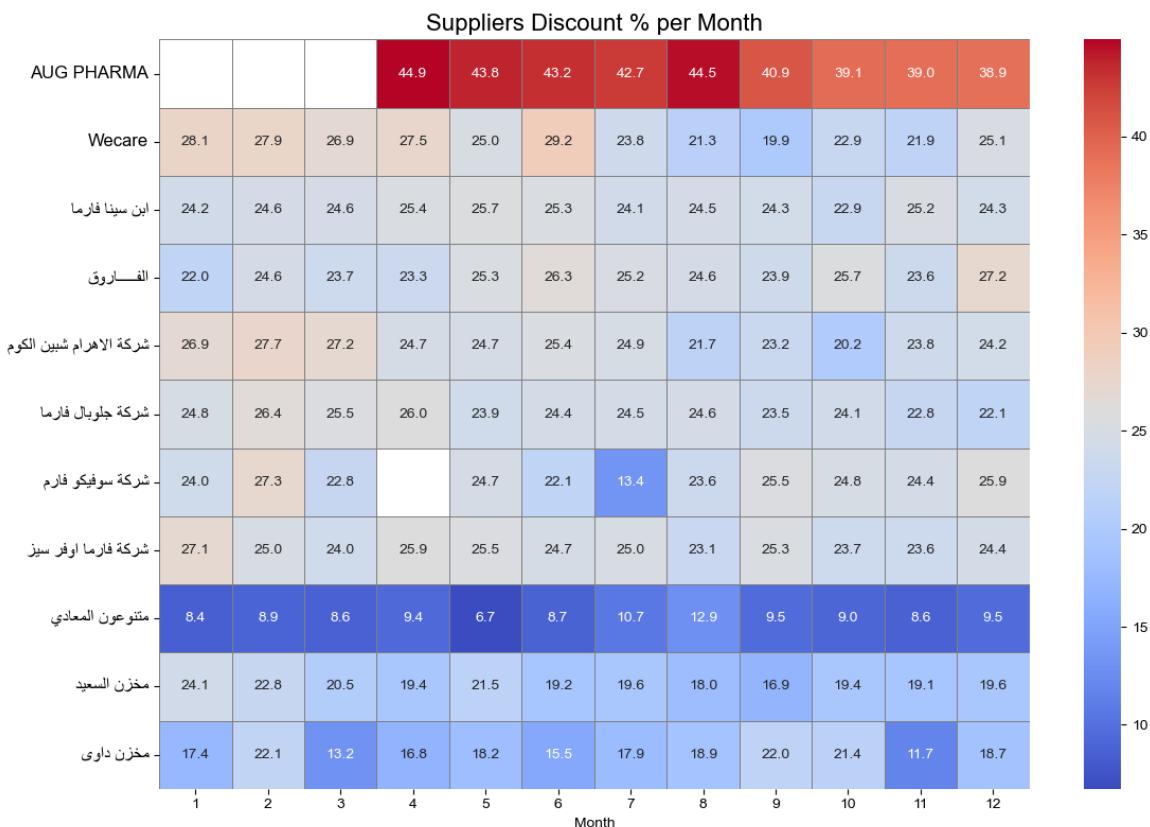
# Create the heatmap
sns.heatmap(heatmap_data, annot=True, fmt=".1f", cmap='coolwarm', l

# Add titles and labels
plt.title("Suppliers Discount % per Month", fontsize=16)
plt.xlabel("Month")
plt.ylabel("")
plt.yticks(fontsize=12)

plt.tight_layout()

plt.savefig('Vendor Discount heatmap.jpg', bbox_inches = 'tight', d
plt.show()

```



```

In [153]: # top_suppliers_sales = df_supplier.loc[:10,'Supplier']
top_suppliers_sales = df_supplier[df_supplier['Supplier'] != 'AUG P

# top_suppliers_sales = top_suppliers_sales.to_list()

# Pivot the data to make it suitable for a heatmap
heatmap_data = top_suppliers_monthly[top_suppliers_monthly['Supplier'].isin(
    top_suppliers_sales)]
heatmap_data.index = heatmap_data.index.map(reshape_arabic)

```

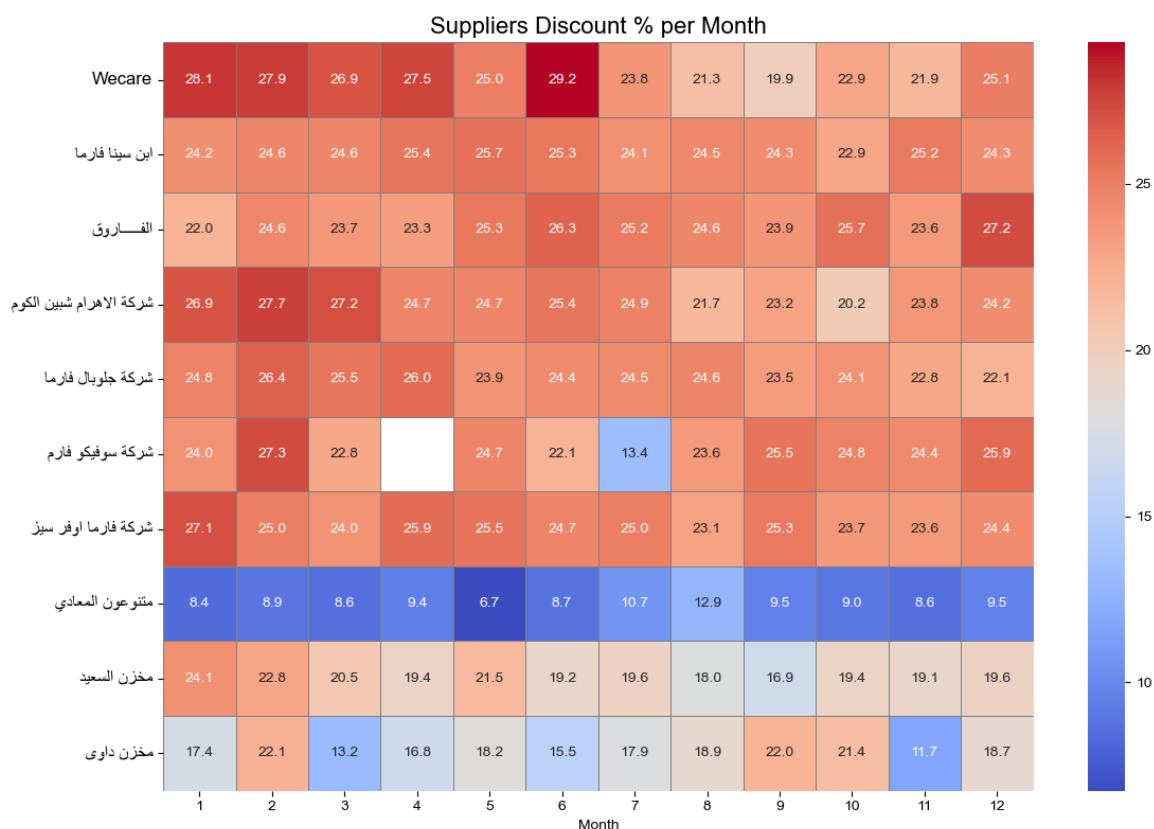
```
# Set up the figure
plt.figure(figsize=(12, 8))

# Create the heatmap
sns.heatmap(heatmap_data, annot=True, fmt=".1f", cmap='coolwarm', l

# Add titles and labels
plt.title("Suppliers Discount % per Month", fontsize=16)
plt.xlabel("Month")
plt.ylabel("")
plt.yticks(fontsize=12)

plt.tight_layout()

plt.savefig('Vendor Discount heatmap without AUG.jpg', bbox_inches='tight')
plt.show()
```



In [154...]

```
# Function to format numbers as K, M, B
def format_compact(val):
    if np.isnan(val):
        return ''
    elif val >= 1_000_000_000:
        return f'{val / 1_000_000_000:.1f} B'
    elif val >= 1_000_000:
        return f'{val / 1_000_000:.0f} M'
    elif val >= 1_000:
        return f'{val / 1_000:.0f} K'
    else:
```

```
    return f'{val:.0f}'\n\n# Filter top suppliers and pivot\ntop_suppliers_sales = df_supplier.loc[:10, 'Supplier']\nheatmap_data = top_suppliers_monthly[top_suppliers_monthly['Supplier']\n    .index='Supplier', columns='month', values='total after discount']\nheatmap_data.index = heatmap_data.index.map(reshape_arabic)\n\n# Apply compact formatting to annotations\nannotations = heatmap_data.applymap(format_compact)\n\n# Set up figure\nplt.figure(figsize=(12, 8))\n\n# Draw heatmap with custom annotations\nsns.heatmap(\n    heatmap_data,\n    annot=annotations,\n    fmt=' ',\n    cmap='coolwarm',\n    linewidths=0.5,\n    linecolor='gray'\n)\n\nplt.title("Suppliers Total Purchased per Month", fontsize=16)\nplt.xlabel("Month")\nplt.ylabel("Supplier")\nplt.yticks(fontsize=12)\n\nplt.tight_layout()\nplt.savefig('Vendor total purchased heatmap.jpg', bbox_inches = 'tight')\nplt.show()
```

```
/var/folders/vf/sz32t6rx7635wcdmj4ydx580000gn/T/ipykernel_50536/2441422606.py:23: FutureWarning: DataFrame.applymap has been deprecated. Use DataFrame.map instead.\n    annotations = heatmap_data.applymap(format_compact)
```



```
# Function to format numbers as K, M, B
def format_compact(val):
    if np.isnan(val):
        return ''
    elif val >= 1_000_000_000:
        return f'{val / 1_000_000_000:.1f} B'
    elif val >= 1_000_000:
        return f'{val / 1_000_000:.0f} M'
    elif val >= 1_000:
        return f'{val / 1_000:.0f} K'
    else:
        return f'{val:.0f}'

# Filter top suppliers and pivot
top_suppliers_sales = df_supplier[df_supplier['Supplier'] != 'ر سيز']
heatmap_data = top_suppliers_monthly[top_suppliers_monthly['Supplier'].isin(['Suppliers'])]
index='Supplier', columns='month', values='total after discount')
)
heatmap_data.index = heatmap_data.index.map(reshape_arabic)

# Apply compact formatting to annotations
annotations = heatmap_data.applymap(format_compact)

# Set up figure
plt.figure(figsize=(12, 8))

# Draw heatmap with custom annotations
sns.heatmap(
    heatmap_data,
    annot=annotations,
    fmt=' ',
```

```

        cmap='coolwarm',
        linewidths=0.5,
        linecolor='gray'
    )

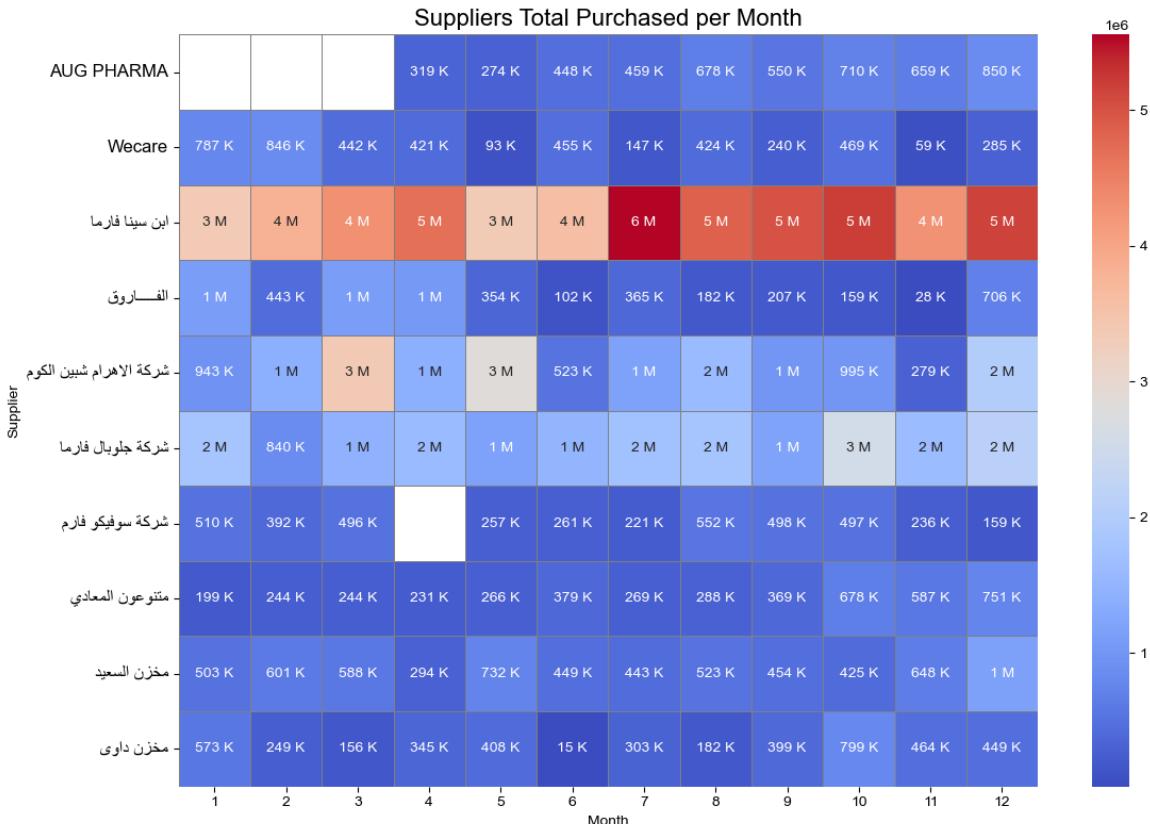
plt.title("Suppliers Total Purchased per Month", fontsize=16)
plt.xlabel("Month")
plt.ylabel("Supplier")
plt.yticks(fontsize=12)

plt.tight_layout()
plt.savefig('Vendor total purchased heatmap without overseas.jpg',
            dpi=300)
plt.show()

```

/var/folders/vf/sz32t6rx7635wcdmj4ydx580000gn/T/ipykernel_50536/1152778858.py:23: FutureWarning: DataFrame.applymap has been deprecated. Use DataFrame.map instead.

```
annotations = heatmap_data.applymap(format_compact)
```



In [156]:

```
top_suppliers_sales = df_supplier.loc[:10, 'Supplier']
# top_suppliers_sales = top_suppliers_sales.to_list()
```

```
# Pivot the data to make it suitable for a heatmap
heatmap_data = top_suppliers_monthly[top_suppliers_monthly['Supplier'].isin(top_suppliers_sales)]
heatmap_data.index = heatmap_data.index.map(reshape_arabic)

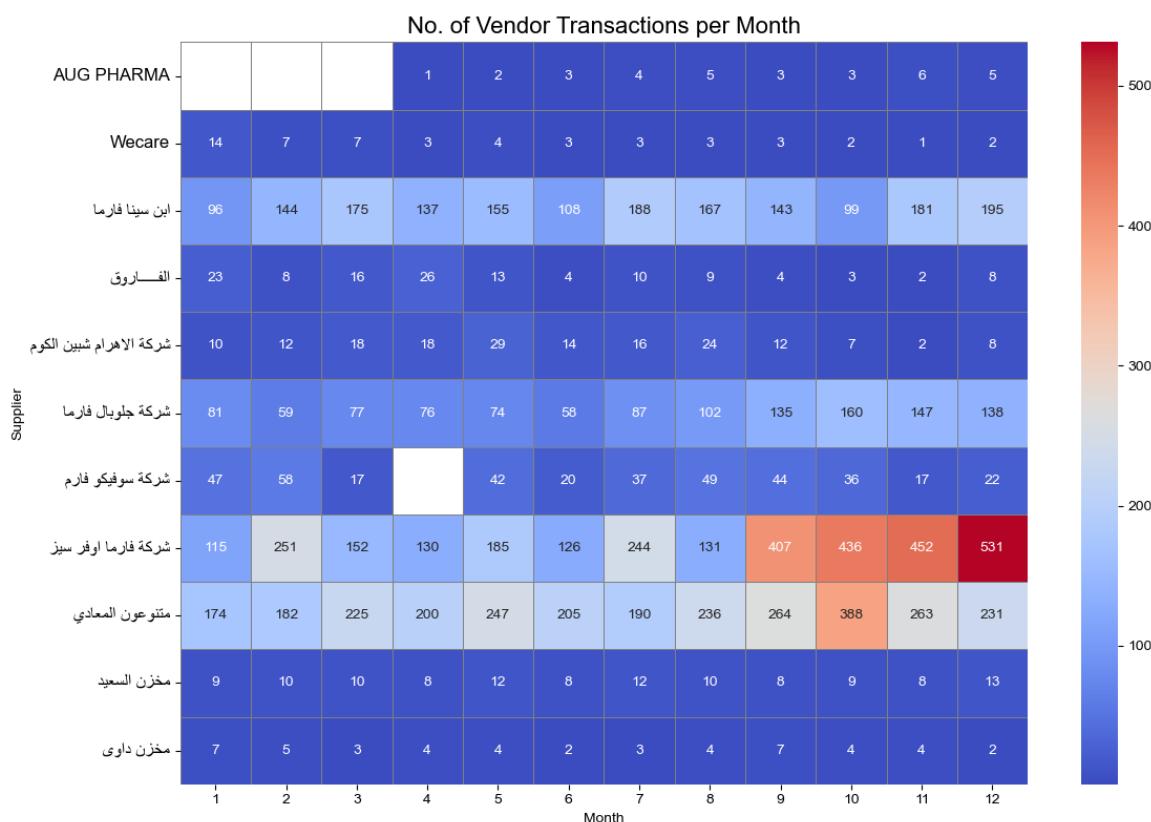
# Set up the figure
plt.figure(figsize=(12, 8))
```

```
# Create the heatmap
sns.heatmap(heatmap_data, annot=True, fmt=".0f", cmap='coolwarm', linewidths=1)

# Add titles and labels
plt.title("No. of Vendor Transactions per Month", fontsize=16)
plt.xlabel("Month")
plt.ylabel("Supplier")
plt.yticks(fontsize=12)

plt.tight_layout()
plt.savefig('No. of Vendor Transactions per Month.jpg', bbox_inches='tight')

plt.show()
```



In [157...]: top_suppliers_4M

Out[157...]

	Supplier	Total Purchased	Total	Total Discount	Discount %
0	ابن سينا فارما	53,135,119.5	70,413,196.8	17,278,077.4	24.5
1	شركة فارما اوفر سيز	43,213,845.1	57,262,833.3	14,048,988.2	24.5
2	شركة جلوبال فارما	19,423,966.0	25,645,432.6	6,221,466.5	24.3
3	شركة الاهرام شبين الكوم	17,587,243.4	23,432,663.0	5,845,419.5	24.9
4	مخزن السعيد	6,821,898.3	8,542,555.0	1,720,656.7	20.1
5	الفاروق	5,804,709.8	7,653,719.0	1,849,009.3	24.2
6	AUG PHARMA	4,947,294.1	8,448,029.0	3,500,734.9	41.4
7	Wecare	4,670,226.4	6,317,893.2	1,647,666.8	26.1
8	متنوعون المعادى	4,506,066.3	4,964,999.4	458,933.1	9.2
9	مخزن داوى	4,339,712.2	5,320,831.5	981,119.3	18.4
10	شركة سوفيكو فارم	4,078,066.4	5,360,520.5	1,282,454.1	23.9

In [158...]

top_suppliers_4M['Total Purchased'].sum()

Out[158...]

168528147.5

In [159...]

df_purchased['total after discount tax 2'].sum()

Out[159...]

202424666.05273923

In [160...]

top_suppliers_4M['Total Purchased'].sum()/df_purchased['total after

Out[160...]

83.25474893266815

In [161...]

```
# potential2 = supplier_product_df.merge(max_discount, on= 'Product ID')
# potential2['Potential Diff'] = (potential2['new discount adds'] - potential2['actual adds'])
# potential2['potential value'] = potential2['new discount adds'] * potential2['Potential Diff']
# print(f"Total: {potential2['Total'].sum():,.2f}")
# print(f"Total Actual Savings: {(potential2['Total'].sum() - potential2['actual adds']):,.2f}")
# print(f"Total Potential Savings value: {potential2['potential value'].sum():,.2f}")
# print(f"Potential Difference: {potential2['Potential Diff'].sum():,.2f}")
```

In [162...]

supplier_product_df.head(2)

Out [162...]

	Supplier	Product name	Quantity	total after discount tax 2	Total	total discount 2	P	C
0	AUG PHARMA	ZURCAL 40 MG 14 GASTRO RESISTANT TAB.	41,322.0	2,186,333.5	3,879,972.0	1,693,638.5	2,1%	
1	ابن سينا فارما	TRESIBA 100 I.U./ML FLEXTOUCH PRE-FILLED PEN	5,856.0	2,098,395.4	2,525,276.0	426,880.6	5%	

In [163...]

```

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import arabic_reshaper
from bidi.algorithm import get_display

# Reshape Arabic function
def reshape_arabic(text):
    try:
        return get_display(arabic_reshaper.reshape(text))
    except:
        return text

# Prepare data
potential_supplier_saving = df_purchased.groupby('Supplier').agg({
    'potential total discount': 'sum',
    'potential value': 'sum',
    'total after discount tax 2': 'sum',
    'total discount 2': 'sum',
    'Total': 'sum'
}).reset_index()

# Calculate percentage
potential_supplier_saving['Potential Diff %'] = (
    potential_supplier_saving['potential value'] / potential_supplier_saving['Total']
) * 100

# Top 11 suppliers
top_suppliers = potential_supplier_saving.sort_values(by='total aft

# Reshape Arabic supplier names
top_suppliers['Supplier'] = top_suppliers['Supplier'].apply(reshape_arabic)

# Sort by Potential Diff %
sorted_df = top_suppliers.sort_values(by='Potential Diff %', ascending=False)

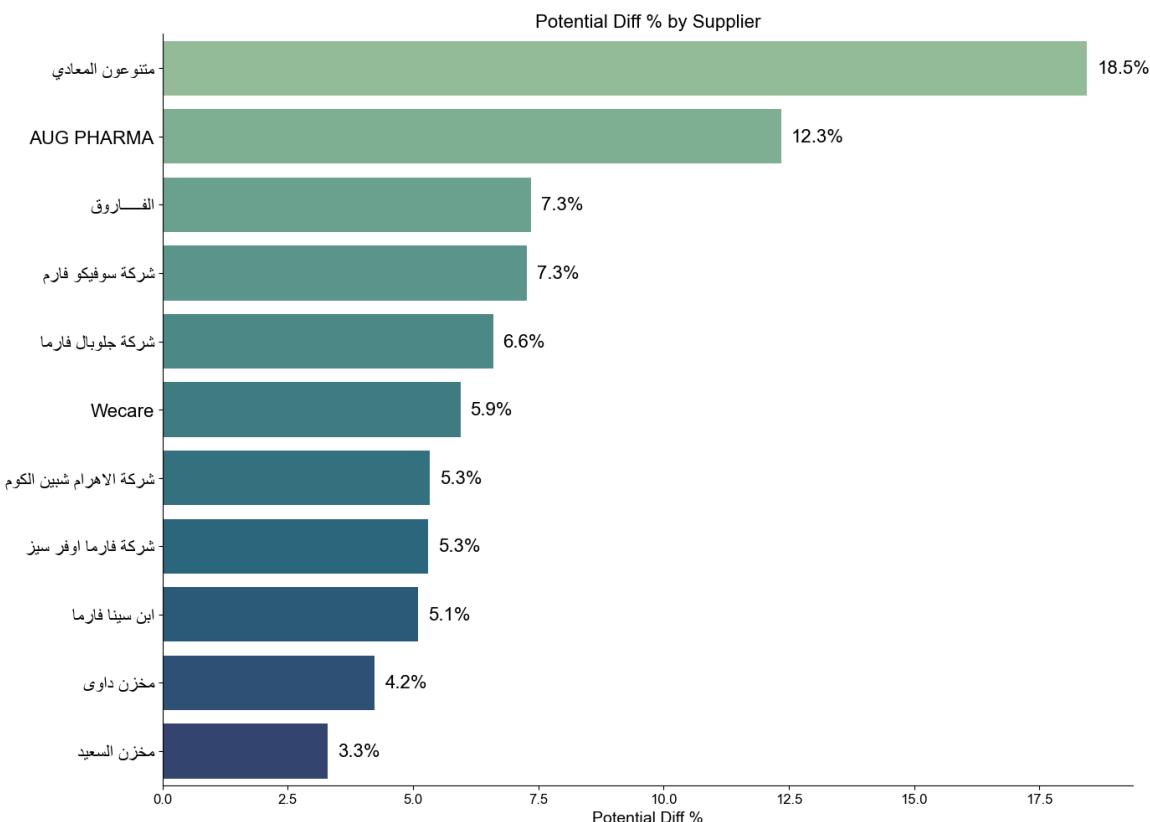
# Plot
plt.figure(figsize=(14, 10))
ax = sns.barplot(data=sorted_df, y='Supplier', x='Potential Diff %')

```

```
# Add value labels
for i, (value, supplier) in enumerate(zip(sorted_df['Potential Diff %'], sorted_df['Supplier'])):
    ax.text(value + 0.2, i, f'{value:.1f}%', va='center', fontsize=12)

# Titles and labels
plt.title('Potential Diff % by Supplier', fontsize=16)
plt.xlabel('Potential Diff %', fontsize=14)
plt.ylabel('')
plt.xticks(fontsize=12)
plt.yticks(fontsize=16)
plt.tight_layout()
sns.despine()
plt.savefig('Potential Diff % by Supplier.jpg', dpi = 200)

plt.show()
```



In [164...]

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import arabic_reshaper
from bidi.algorithm import get_display

# Reshape Arabic function
def reshape_arabic(text):
    try:
        return get_display(arabic_reshaper.reshape(text))
    except:
        return text

# Prepare data
potential_supplier_saving = df_purchased.groupby('Supplier').agg({
    'potential total discount': 'sum',
    'potential value': 'sum',
```

```
'total after discount tax 2': 'sum',
'total discount 2': 'sum',
'Total': 'sum'
}).reset_index()

# Calculate percentage
potential_supplier_saving['Potential Diff %'] = (
    potential_supplier_saving['potential value'] / potential_supplier_saving['Total'])
* 100

# Top 11 suppliers
top_suppliers = potential_supplier_saving.sort_values(by='total after discount tax 2', ascending=False).head(11)

# Reshape Arabic supplier names
top_suppliers['Supplier'] = top_suppliers['Supplier'].apply(reshape_arabic)

# Sort by Potential Diff %
sorted_df = top_suppliers.sort_values(by='potential value', ascending=False)

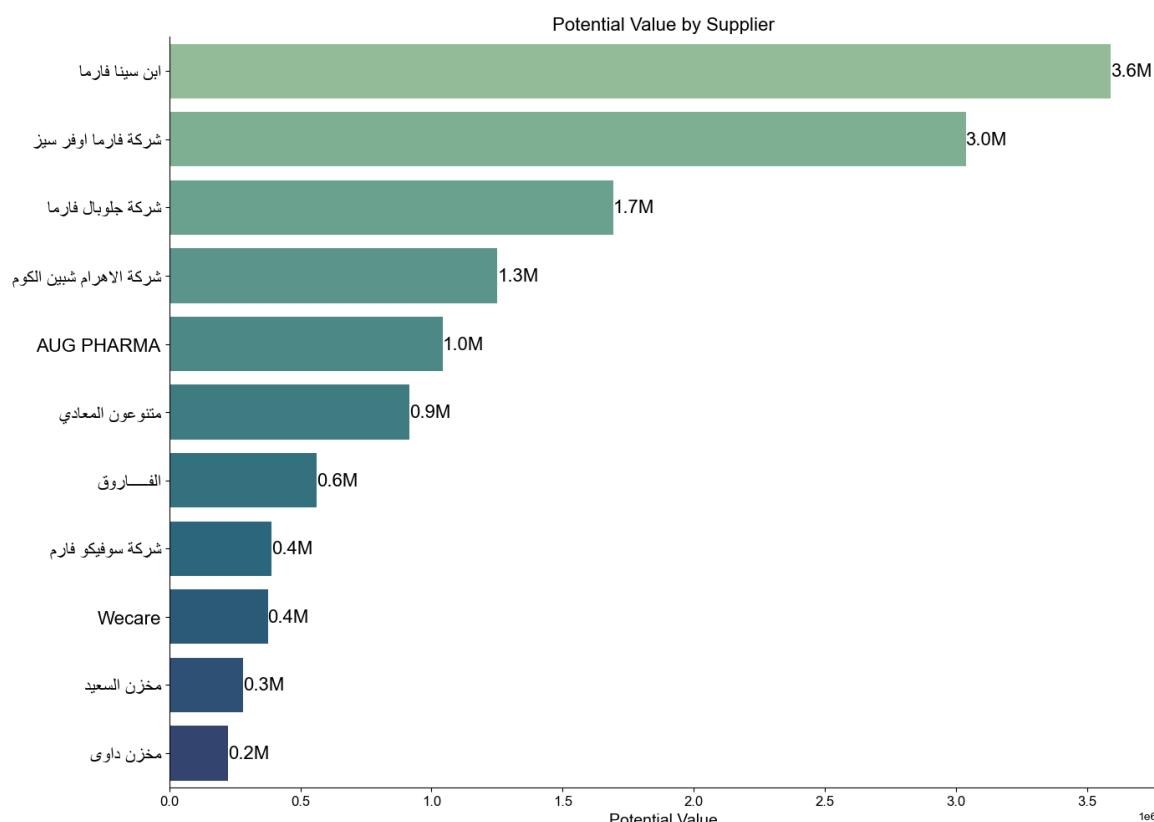
# Plot
plt.figure(figsize=(14, 10))
ax = sns.barplot(data=sorted_df, y='Supplier', x='potential value', color='blue')

# Add value labels
for i, (value, supplier) in enumerate(zip(sorted_df['potential value'], sorted_df['Supplier'])):
    ax.text(value + 0.9, i, f'{value/1_000_000:.1f}M', va='center', color='red')

# Titles and labels
plt.title('Potential Value by Supplier', fontsize=16)
plt.xlabel('Potential Value', fontsize=14)
plt.ylabel('')
plt.xticks(fontsize=12)
plt.yticks(fontsize=16)
plt.tight_layout()
sns.despine()
plt.savefig('Potential Value by Supplier.jpg', dpi = 200)

plt.show()

print(f"Total Potential Savings value: {top_suppliers['potential value'].sum()}")
print(f"Total Potential Savings percentage: {top_suppliers['Potential Diff %'].sum()}%")
```



Total Potential Savings value: 13,361,445.55

In [165... `supplier_product_df.head(2)`

	Supplier	Product name	Quantity	total after discount tax 2	Total	total discount 2	%
0	AUG PHARMA	ZURCAL 40 MG 14 GASTRO RESISTANT TAB.	41,322.0	2,186,333.5	3,879,972.0	1,693,638.5	2,1%
1	ابن سينا فارما	TRESIBA 100 I.U./ML FLEXTOUCH PRE-FILLED PEN	5,856.0	2,098,395.4	2,525,276.0	426,880.6	5%

In [166... `top_suppliers`

Out [166...]

	Supplier	potential total discount	potential value	total after discount tax 2	total discount 2	Total
10	امراطفانليس نبا	20,865,666.1	3,587,588.8	53,135,119.5	17,278,077.4	70,413,19
40	زيس رفوا امرافة كرش	17,085,099.5	3,036,111.3	43,213,845.1	14,048,988.2	57,262,83
34	امراطف لابولاج ةكرش	7,913,504.2	1,692,142.7	19,423,966.0	6,221,466.5	25,645,43
30	موكلا نيپيش مارهلا ةكرش	7,096,322.9	1,250,903.3	17,587,243.4	5,845,419.5	23,432,66
55	ديعسلا نزخم	2,002,278.4	281,621.6	6,821,898.3	1,720,656.7	8,542,55
20	قورا فلا	2,411,487.0	562,477.7	5,804,709.8	1,849,009.3	7,653,71
5	AUG PHARMA	4,544,017.5	1,043,282.7	4,947,294.1	3,500,734.9	8,448,02
9	Wecare	2,023,342.1	375,675.3	4,670,226.4	1,647,666.8	6,317,89
49	يداعملا نوعونتم	1,375,015.4	916,082.3	4,506,066.3	458,933.1	4,964,99
62	ىواد نزخم	1,206,632.6	225,513.3	4,339,712.2	981,119.3	5,320,83
37	امراطف وكيفوس ةكرش	1,672,500.6	390,046.6	4,078,066.4	1,282,454.1	5,360,52

In [167...]

```

list1 = ['شركة الاهرام شبين الكوم', 'شركة جلوبال فارما', 'الفاروق']
all_ven = df_purchased[df_purchased['Supplier'].isin(list1)].groupby('Supplier').sum()
all_ven['Discount %'] = all_ven['total discount 2'] / all_ven['total value']

def reshape_arabic(text):
    return get_display(arabic_reshaper.reshape(text))

# Reshape supplier names
all_ven['Supplier'] = all_ven['Supplier'].apply(reshape_arabic)
sns.lineplot(data=all_ven, x='month', y='Discount %', hue='Supplier')

# Set up the figure

# Create the heatmap

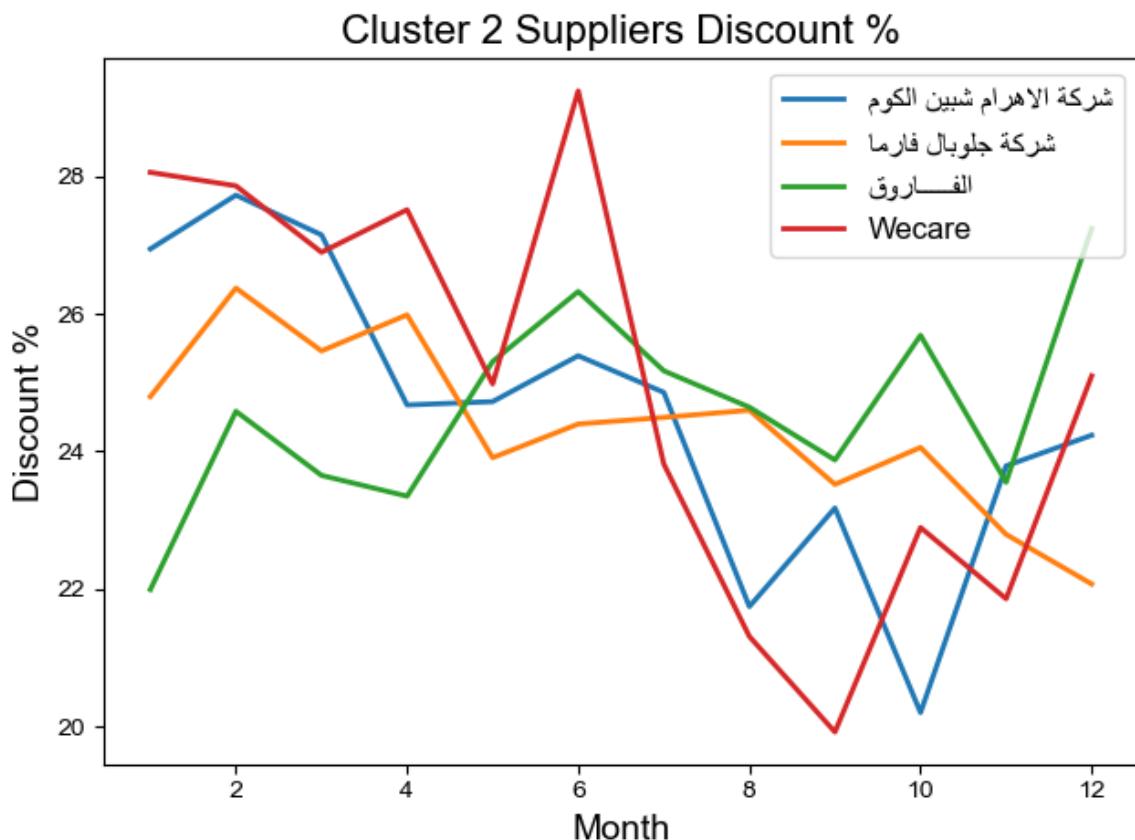
# Add titles and labels
plt.title("Cluster 2 Suppliers Discount %", fontsize=16)
plt.tight_layout()
plt.xlabel("Month", fontsize=14)

```

```

plt.ylabel("Discount %", fontsize=14)
plt.legend(fontsize=12)
plt.savefig('Cluster 2 Suppliers Discount %.jpg', bbox_inches = 'tight')
plt.show()

```



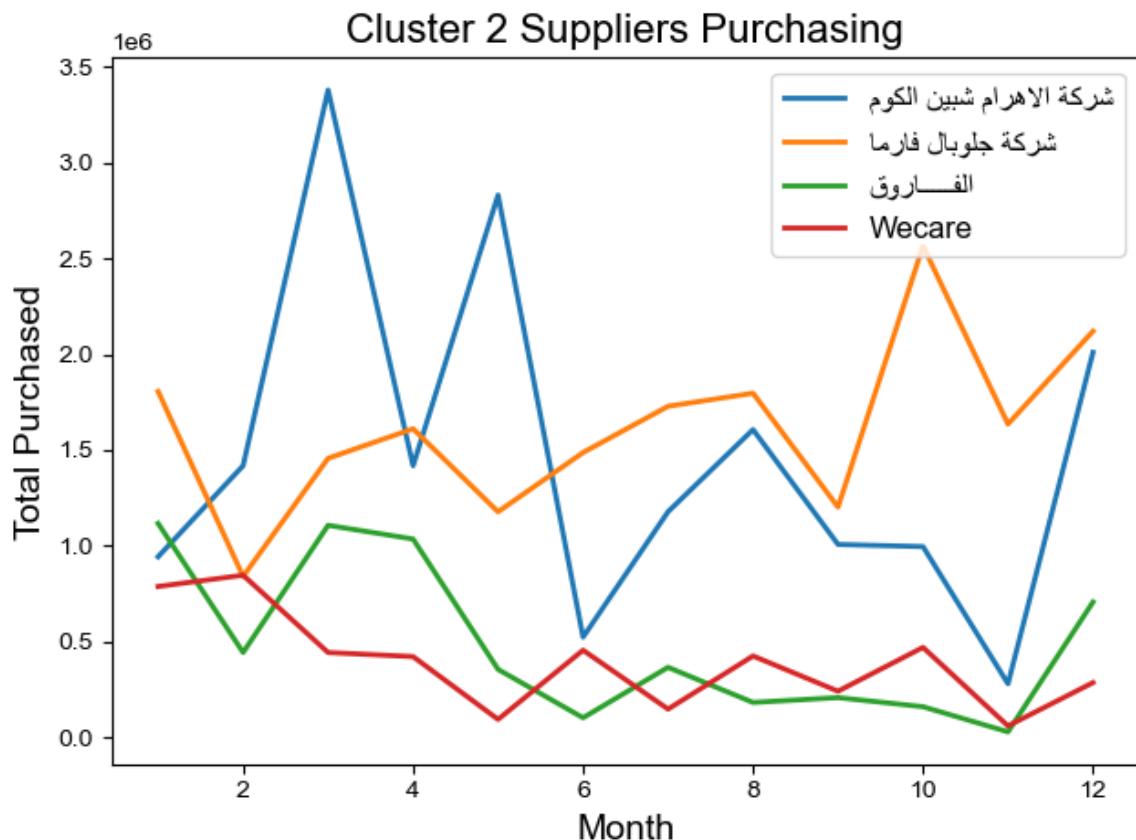
```

In [168]: list1 = ['شركة الاهرام شبين الكوم', 'شركة جلوبال فارما', 'الفاروق']
all_ven = df_purchased[df_purchased['Supplier'].isin(list1)].groupby(['Supplier'])
    .sum()
    .reset_index()
all_ven['Discount %'] = all_ven['total discount 2'] / all_ven['Total Purchased']
def reshape_arabic(text):
    return get_display(arabic_reshaper.reshape(text))

# Reshape supplier names
all_ven['Supplier'] = all_ven['Supplier'].apply(reshape_arabic)
sns.lineplot(data=all_ven, x='month', y='total after discount tax 2')

# Add titles and labels
plt.title("Cluster 2 Suppliers Purchasing", fontsize=16)
plt.tight_layout()
plt.xlabel("Month", fontsize=14)
plt.ylabel("Total Purchased", fontsize=14)
plt.legend(fontsize=12)
plt.savefig('Cluster 2 Suppliers purchasing.jpg', bbox_inches = 'tight')
plt.show()

```



In [169]:

```
df_purchased.groupby(['month', 'Category']).agg({'total after discount': 'sum',  
                                              'Total' : 'sum',  
                                              'total discount 2' : 'sum'}).re
```

Out [169...]

	month	Category	total after discount tax 2	Total	total discount 2
0	1	Chemotherapy	881,770.0	1,128,991.0	247,221.0
1	1	Toll	7,000.0	8,404.0	1,404.0
2	1	import	5,906,252.5	7,148,861.1	1,242,608.6
3	1	import_re_registration	238,891.6	270,333.0	31,441.4
4	1	local	7,287,427.0	10,249,443.3	2,962,016.2
5	2	Chemotherapy	697,750.0	902,437.0	204,687.0
6	2	import	4,632,054.0	5,714,737.3	1,082,683.3
7	2	import_re_registration	259,445.1	285,327.0	25,881.9
8	2	local	7,533,212.0	10,724,233.5	3,191,021.5
9	3	Chemotherapy	573,500.0	711,924.0	138,424.0
10	3	Medical device	22,000.0	32,800.0	10,800.0
11	3	import	6,646,992.3	8,104,182.1	1,457,189.8
12	3	import_re_registration	214,269.7	242,364.0	28,094.3
13	3	local	8,849,866.3	12,523,302.9	3,673,436.6
18	4	local	8,284,842.8	11,712,957.6	3,428,114.8
16	4	import	5,081,849.9	6,156,321.9	1,074,472.1
17	4	import_re_registration	208,560.8	235,108.5	26,547.7
14	4	Chemotherapy	262,040.0	317,958.0	55,918.0
15	4	Medical device	163.8	164.0	0.2
19	5	Chemotherapy	869,604.0	1,091,260.0	221,656.0

In [170...]

```
# top_generic = df_purchased.groupby('Generic Name')['total after discount tax 2'].sum()
# print(top_generic['total after discount tax 2'].sum())
# print(df_purchased['total after discount tax 2'].sum())
```

In [171...]

```
df_discount_month
```

Out[171...]

	month	total after discount tax 2	Total	total discount 2	Discount %
0	1	14,321,341.0	18,806,032.0	4,484,691.0	23.8
1	2	13,122,461.0	17,626,735.0	4,504,274.0	25.6
2	3	16,306,628.0	21,614,573.0	5,307,945.0	24.6
3	4	13,837,457.0	18,422,510.0	4,585,053.0	24.9
4	5	14,283,965.0	18,870,100.0	4,586,136.0	24.3
5	6	11,199,789.0	14,860,276.0	3,660,488.0	24.6
6	7	16,681,366.0	21,989,756.0	5,308,390.0	24.1
7	8	15,632,065.0	20,521,993.0	4,889,928.0	23.8
8	9	18,087,569.0	23,800,723.0	5,713,154.0	24.0
9	10	22,699,714.0	29,540,042.0	6,840,328.0	23.2
10	11	16,619,802.0	21,627,328.0	5,007,525.0	23.2
11	12	29,632,508.0	38,832,374.0	9,199,866.0	23.7

In [172...]

```

from sklearn.linear_model import LinearRegression

# Simulate your data
data = {
    'Discount %': [23.8, 25.6, 24.6, 24.9, 24.3, 24.6, 24.1, 23.8,
    ]
df_try = pd.DataFrame(data)
df_try['month_num'] = np.arange(len(df_try)) # 0 to 11

# Fit Linear Regression
X = df_try[['month_num']]
y = df_try['Discount %']
model = LinearRegression().fit(X, y)

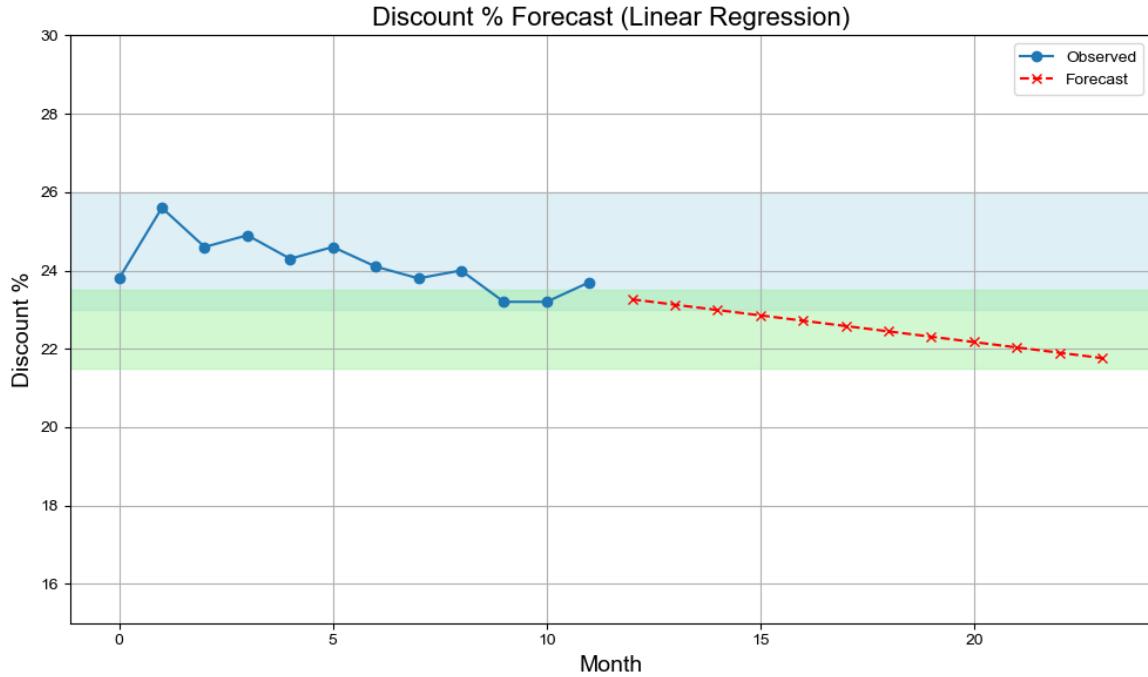
# Forecast next 12 months
future_months = np.arange(12, 24).reshape(-1, 1)
future_preds = model.predict(future_months)

# Plot
plt.figure(figsize=(10, 6))
plt.plot(df_try['month_num'], y, marker='o', label='Observed')
plt.plot(future_months, future_preds, linestyle='--', color='red',
plt.title('Discount % Forecast (Linear Regression)', fontsize=16)
plt.xlabel('Month', fontsize=14)
plt.ylabel('Discount %', fontsize=14)
plt.ylim(15, 30)
plt.legend()
plt.grid(True)
plt.axhspan(23, 26, color='lightblue', alpha=0.4, label='Target Ran
plt.axhspan(21.5, 23.5, color='lightgreen', alpha=0.4, label='Targe
plt.tight_layout()

```

```
plt.savefig('discount forecast.jpg', bbox_inches = 'tight', dpi = 200)
plt.show()
```

```
/Users/mohamad/opt/anaconda3/envs/test_env/lib/python3.11/site-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```



In [173]:

```
from sklearn.linear_model import LinearRegression

# # Simulate your data
# data = {
#     'Discount %': [24.6, 25.9, 24.9, 25.5, 24.9, 25.4, 25.0, 24.4,
#     # ]
# df_try = df_discount_month['total after discount tax 2']
# df_try['month_num'] = np.arange(len(df_try)) # 0 to 11

# Fit Linear Regression
X = df_discount_month[['month']]
y = df_discount_month['total after discount tax 2']
model = LinearRegression().fit(X, y)

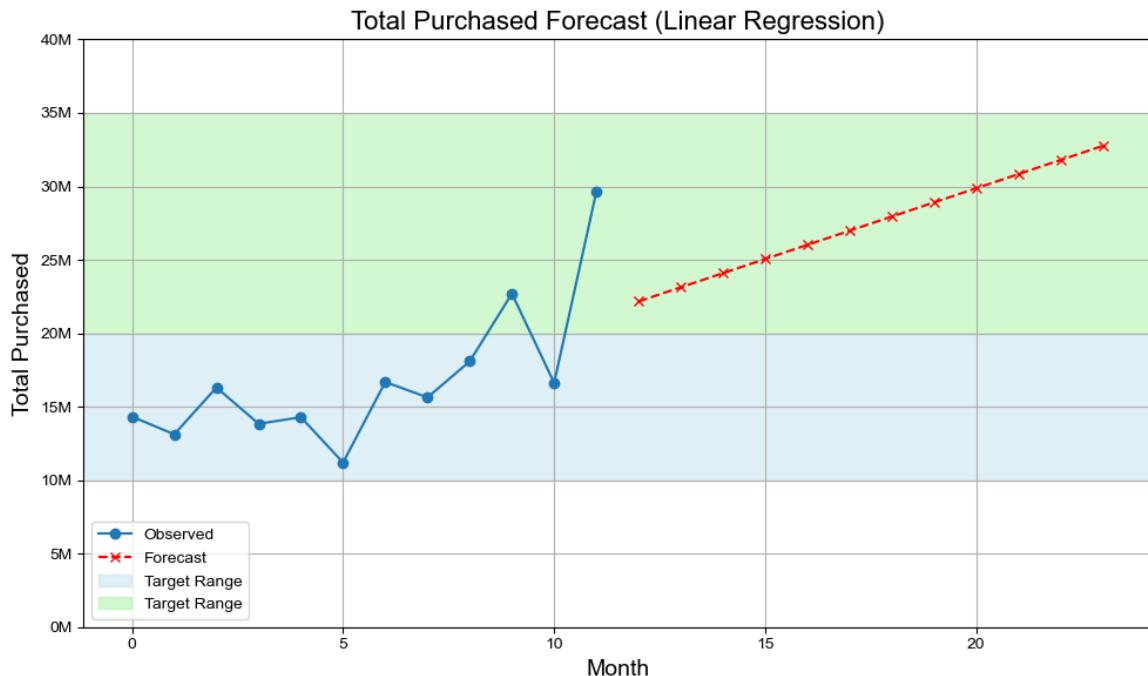
# Forecast next 12 months
future_months = np.arange(12, 24).reshape(-1, 1)
future_preds = model.predict(future_months)

# Plot
plt.figure(figsize=(10, 6))
plt.plot(df_try['month_num'], y, marker='o', label='Observed')
plt.plot(future_months, future_preds, linestyle='--', color='red',
plt.gca().yaxis.set_major_formatter(mticker.FuncFormatter(lambda x,
plt.ylim(0, 40_000_000)
plt.axhspan(10000000, 20000000, color='lightblue', alpha=0.4, label=
plt.axhspan(20000000, 35000000, color='lightgreen', alpha=0.4, labe
```

```
plt.title('Total Purchased Forecast (Linear Regression)', fontsize=14)
plt.xlabel('Month', fontsize=14)
plt.ylabel('Total Purchased', fontsize=14)
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.savefig('total purchased forecast.jpg', bbox_inches = 'tight',
            plt.show()
```

/Users/mohamad/opt/anaconda3/envs/test_env/lib/python3.11/site-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names

```
    warnings.warn(
```



```
In [174...]: future_df = pd.DataFrame(future_preds, columns=['Prediction']) # a
future_df.iloc[:4].sum()
```

```
Out[174...]: Prediction    94,439,764.1
dtype: float64
```

```
# from statsmodels.tsa.holtwinters import SimpleExpSmoothing

# # SES Model
# ses_model = SimpleExpSmoothing(df_try['Discount %']).fit()
# ses_forecast = ses_model.forecast(12)

# # Plot
# plt.figure(figsize=(10, 6))
# plt.plot(df_try['Discount %'], marker='o', label='Observed')
# plt.plot(np.arange(12, 24), ses_forecast, linestyle='--', color='red')
# plt.title('Discount % Forecast (Simple Exponential Smoothing)')
# plt.xlabel('Month')
# plt.ylabel('Discount %')
# plt.legend()
```

```
# plt.grid(True)
# plt.tight_layout()
# plt.show()
```

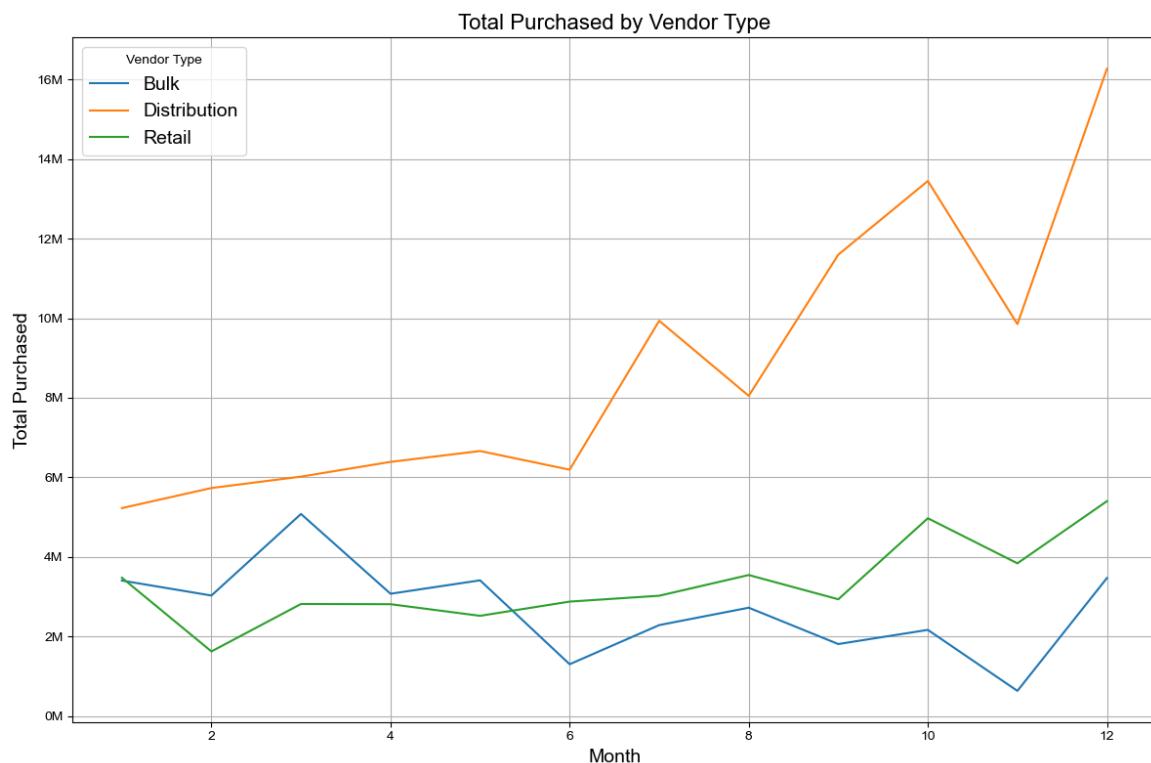
In [176... df_try

Out[176...

	Discount %	month_num
0	23.8	0
1	25.6	1
2	24.6	2
3	24.9	3
4	24.3	4
5	24.6	5
6	24.1	6
7	23.8	7
8	24.0	8
9	23.2	9
10	23.2	10
11	23.7	11

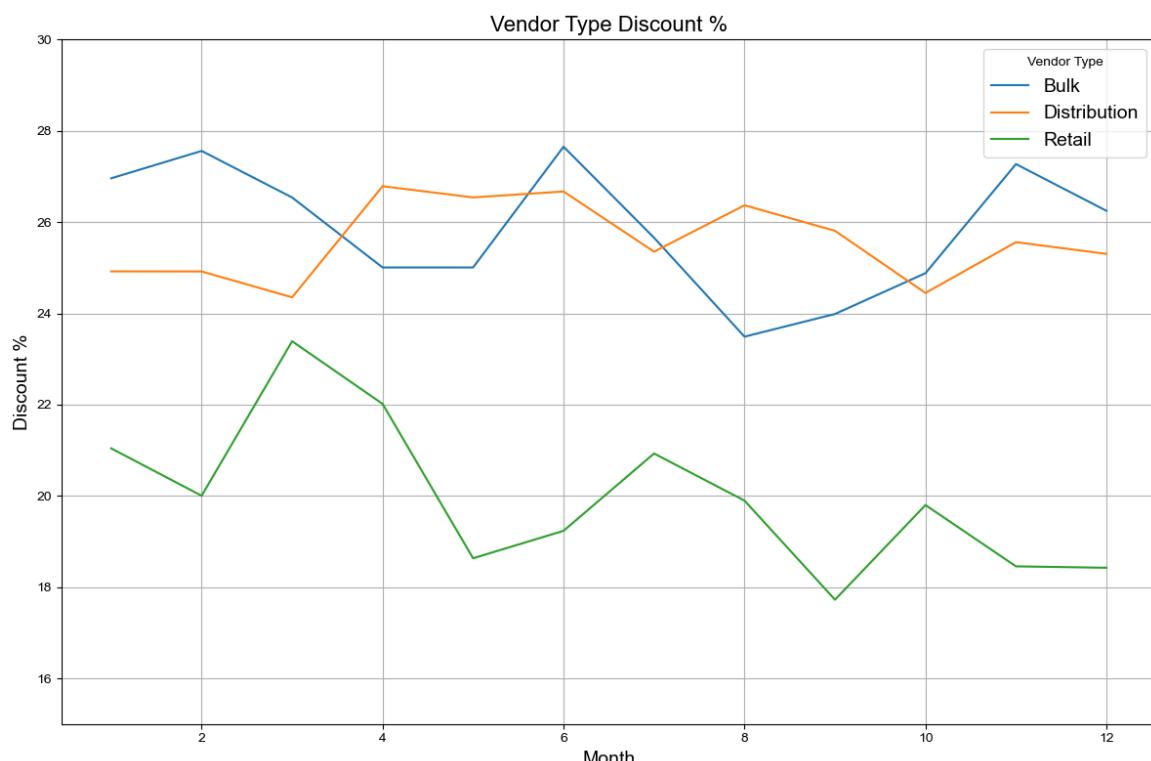
	Discount %	month_num
0	23.8	0
1	25.6	1
2	24.6	2
3	24.9	3
4	24.3	4
5	24.6	5
6	24.1	6
7	23.8	7
8	24.0	8
9	23.2	9
10	23.2	10
11	23.7	11

```
In [177... vendor_type = df_purchased.groupby(['month', 'Vendor Type']).agg({'Total' : 'sum',
    'total discount 2' : 'sum'}).re
vendor_type['Discount %'] = vendor_type['total discount 2'] / vend
vendor_type[vendor_type['Vendor Type'] != 'NA'].pivot(index='month'
plt.gca().yaxis.set_major_formatter(mticker.FuncFormatter(lambda x,
plt.title('Total Purchased by Vendor Type', fontsize=16)
plt.xlabel('Month', fontsize=14)
plt.ylabel('Total Purchased', fontsize=14)
plt.legend(title='Vendor Type', fontsize=14)
plt.grid(True)
plt.tight_layout()
plt.savefig('Total Purchased by Vendor Type.jpg', bbox_inches = 'ti
plt.show()
```



In [178]:

```
vendor_type[vendor_type['Vendor Type'] != 'NA'].pivot(index='month'  
plt.title('Vendor Type Discount %', fontsize=16)  
plt.xlabel('Month', fontsize=14)  
plt.ylabel('Discount %', fontsize=14)  
plt.legend(title='Vendor Type', fontsize=14)  
plt.grid(True)  
plt.ylim(15, 30)  
plt.tight_layout()  
plt.savefig('Vendor Type Discount %.jpg', bbox_inches = 'tight', dp  
plt.show()
```



```
In [179... vendor_type.groupby('Vendor Type')['total after discount tax 2'].su
```

```
Out[179... Vendor Type
Bulk           32,373,878.9
Distribution   105,374,325.0
NA             24,849,122.5
Retail          39,827,339.6
Name: total after discount tax 2, dtype: float64
```

```
In [180... vendor_type.groupby('Vendor Type')['total after discount tax 2'].su
```

```
Out[180... Vendor Type
Bulk           32,373,878.9
Distribution   105,374,325.0
NA             24,849,122.5
Retail          39,827,339.6
Name: total after discount tax 2, dtype: float64
```

```
In [181... vendor_type['total after discount tax 2'].sum()
177,575,544
```

```
Out[181... (177, 575, 544)
```

```
In [182... import matplotlib.ticker as mticker
```

```
supplier_monthly_data = df_purchased[df_purchased['Supplier Rank by
                                         'Product name' : 'nunique',
                                         'Total' : 'sum',
                                         'total discount 2': 'sum']].r
supplier_monthly_data['Discount %'] = supplier_monthly_data['total
                                         'Discount %'] / supplier_monthly_data['Total
                                         'Discount %'] * 100
```

```
overseas = df_purchased[df_purchased['Supplier'] == 'فاما اوفر سيز'
                           'Total' : 'sum',
                           'total discount 2' : 'sum']).re
overseas['Discount %'] = overseas['total discount 2'] / overseas['T
```

```
fig, ax = plt.subplots(4,1, figsize=(10, 8))

sns.lineplot(data = supplier_monthly_data, x = 'month', y = 'total
sns.lineplot(data = supplier_monthly_data, x = 'month', y = 'Produc
# sns.lineplot(data = supplier_category_data[['local', 'import']],x
supplier_category_data[['local', 'import']].plot(kind = 'line', ax
sns.lineplot(data = overseas,x = 'month',y = 'Discount %', ax= ax[3]
ax[3].set_title('Overseas Discount %')
ax[3].set_xlabel('Month')
ax[3].set_xlim(22, 30)
ax[3].grid(True)
```

```
ax[0].set_title('Total Purchased per Month')
ax[0].set_xlabel('')
```

```
ax[0].set_ylabel('')
ax[0].grid(True)

ax[1].set_title('Products Number per Month')
ax[1].set_xlabel('')
ax[1].set_ylabel('')
ax[1].grid(True)

ax[2].set_title('Local & import Purchased per Month')
ax[2].set_xlabel('')
ax[2].set_ylabel('')
ax[2].grid(True)

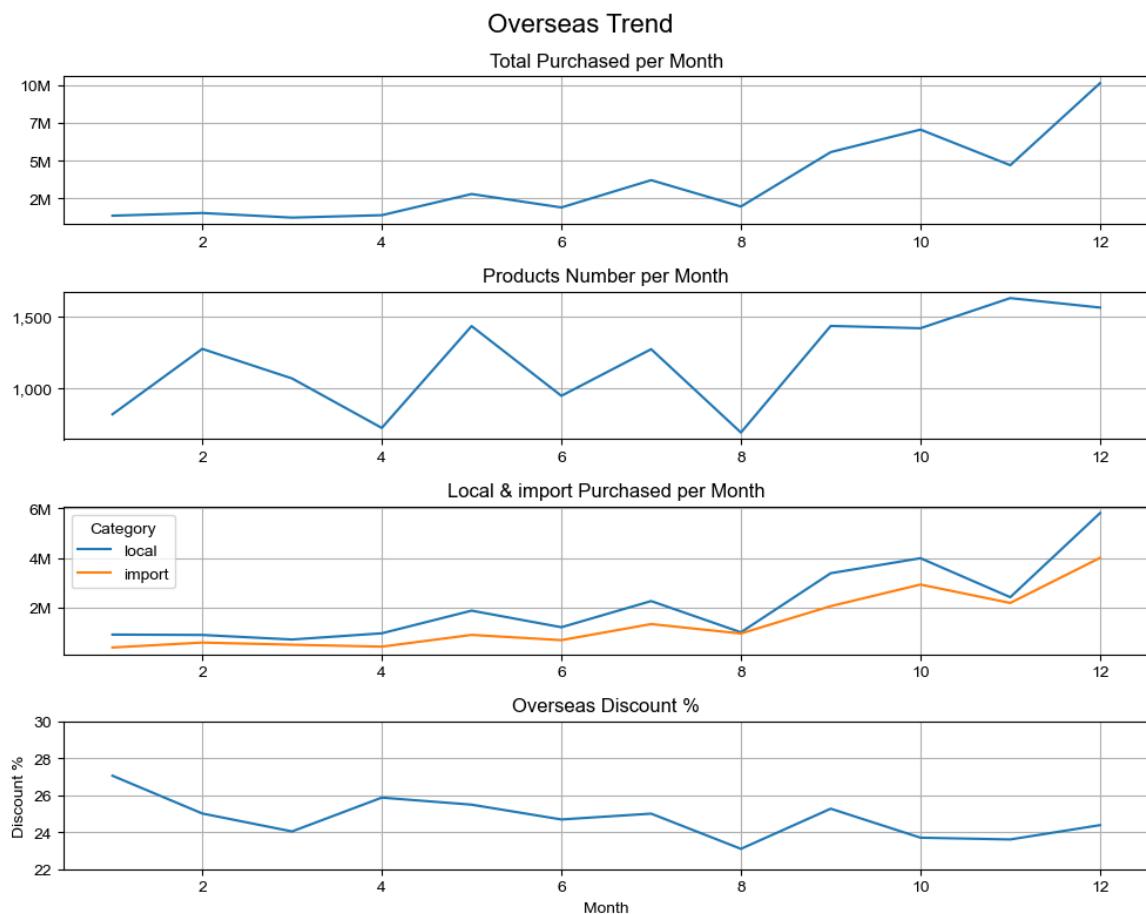
# avoid scientific notation
ax[0].yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f'))
ax[1].yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f'))
ax[2].yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f')

ax[0].yaxis.set_major_formatter(mticker.FuncFormatter(lambda x, _:
ax[2].yaxis.set_major_formatter(mticker.FuncFormatter(lambda x, _:

plt.suptitle('Overseas Trend', fontsize=16)

plt.tight_layout()
plt.savefig('overseas trend.jpg', bbox_inches = 'tight', dpi = 200)

plt.show()
```



```
In [183]: import matplotlib.ticker as mticker

supplier_monthly_data = df_purchased[df_purchased['Supplier Rank by
                                         'Product name' : 'nunique',
                                         'Total' : 'sum',
                                         'total discount 2': 'sum']].r
supplier_monthly_data['Discount %'] = supplier_monthly_data['total

supplier_category_data = df_purchased[df_purchased['Supplier Rank b

Global = df_purchased[df_purchased['Supplier'] == 'Global
                                         'Total' : 'sum',
                                         'total discount 2' : 'sum']).re
Global['Discount %'] = Global['total discount 2'] / Global['Total']

fig, ax = plt.subplots(4,1, figsize=(10, 8))

sns.lineplot(data = supplier_monthly_data, x = 'month', y = 'total
sns.lineplot(data = supplier_monthly_data, x = 'month', y = 'Produc
# sns.lineplot(data = supplier_category_data[['local', 'import']],x
supplier_category_data[['local', 'import']].plot(kind = 'line', ax
sns.lineplot(data = Global,x = 'month',y = 'Discount %', ax= ax[3])
ax[3].set_title('Global Discount %')
ax[3].set_xlabel('Month')
ax[3].set_xlim(20, 30)
ax[3].grid(True)
```

```
ax[0].set_title('Total Purchased per Month')
ax[0].set_xlabel('')
ax[0].set_ylabel('')
ax[0].grid(True)

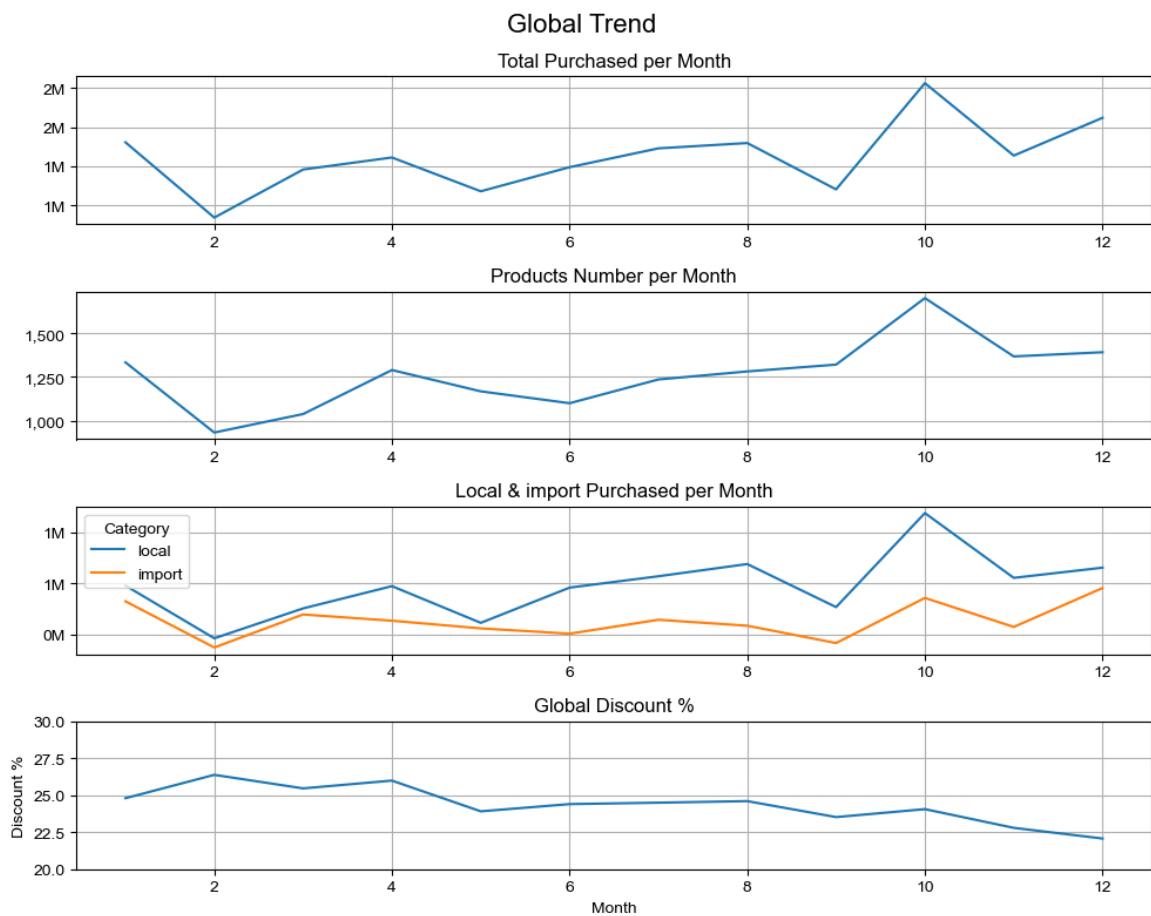
ax[1].set_title('Products Number per Month')
ax[1].set_xlabel('')
ax[1].set_ylabel('')
ax[1].grid(True)

ax[2].set_title('Local & import Purchased per Month')
ax[2].set_xlabel('')
ax[2].set_ylabel('')
ax[2].grid(True)

# avoid scientific notation
ax[0].yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f'))
ax[1].yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f')
ax[2].yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f')

ax[0].yaxis.set_major_formatter(mticker.FuncFormatter(lambda x, _:
ax[2].yaxis.set_major_formatter(mticker.FuncFormatter(lambda x, _:

plt.suptitle('Global Trend', fontsize=16)
plt.tight_layout()
plt.savefig('Global Trend.jpg', bbox_inches = 'tight', dpi = 200)
plt.show()
```



```
In [184]: supplier_monthly_data = df_purchased[df_purchased['Supplier Rank by
                           'Product name' : 'nunique',
                           'Total' : 'sum',
                           'total discount 2': 'sum']].r
supplier_monthly_data['Discount %'] = supplier_monthly_data['total
supplier_category_data = df_purchased[df_purchased['Supplier Rank b

Ahram = df_purchased[df_purchased['Supplier'] == 'الاهرام شبين الكوم'
                           'Total' : 'sum',
                           'total discount 2' : 'sum']).re
Ahram['Discount %'] = Ahram['total discount 2'] / Ahram['Total'] *

fig, ax = plt.subplots(4,1, figsize=(10, 8))

sns.lineplot(data = supplier_monthly_data, x = 'month', y = 'total
sns.lineplot(data = supplier_monthly_data, x = 'month', y = 'Produc
# sns.lineplot(data = supplier_category_data[['local', 'import']],x
supplier_category_data[['local', 'import']].plot(kind = 'line', ax =
sns.lineplot(data = Ahram,x = 'month',y = 'Discount %', ax= ax[3])
ax[3].set_title('Ahram Discount %')
ax[3].set_xlabel('Month')
ax[3].set_xlim(20, 30)
ax[3].grid(True)
```

```
ax[0].set_title('Total Purchased per Month')
ax[0].set_xlabel('')
ax[0].set_ylabel('')
ax[0].grid(True)

ax[1].set_title('Products Number per Month')
ax[1].set_xlabel('')
ax[1].set_ylabel('')
ax[1].grid(True)

ax[2].set_title('Local & import Purchased per Month')
ax[2].set_xlabel('')
ax[2].set_ylabel('')
ax[2].grid(True)

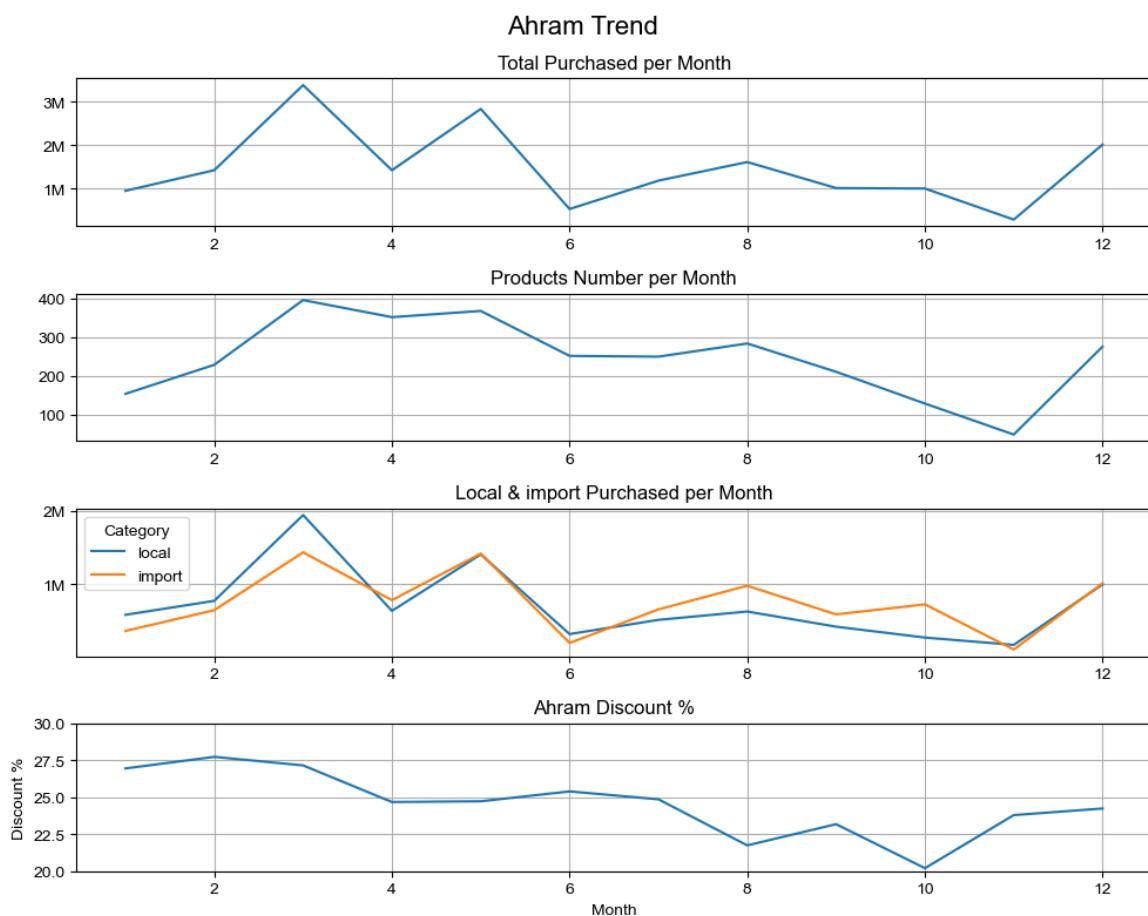
# avoid scientific notation
ax[0].yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f'))
ax[1].yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f'))
ax[2].yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f'))

ax[0].yaxis.set_major_formatter(mticker.FuncFormatter(lambda x, _:
ax[2].yaxis.set_major_formatter(mticker.FuncFormatter(lambda x, _:

plt.suptitle('Ahram Trend', fontsize=16)

plt.tight_layout()
plt.savefig('Ahram Trend.jpg', bbox_inches = 'tight', dpi = 200)

plt.show()
```



In [185...]: df_purchased[df_purchased['Supplier'] == 'شركة الاهرام شبين الكوم'].
 'Total' : 'sum',
 'total discount 2' : 'sum']).re

Out[185...]:

	month	total after discount tax 2	Total	total discount 2
0	1	942,693.4	1,290,414.8	347,721.3
1	2	1,418,051.1	1,962,121.1	544,070.0
2	3	3,379,715.6	4,639,385.2	1,259,669.6
3	4	1,417,751.3	1,882,201.8	464,450.4
4	5	2,831,203.0	3,761,140.5	929,937.5
5	6	522,677.4	700,566.5	177,889.1
6	7	1,177,622.7	1,567,226.4	389,603.7
7	8	1,606,538.5	2,052,897.8	446,359.2
8	9	1,006,820.3	1,310,576.5	303,756.2
9	10	995,432.9	1,247,419.0	251,986.1
10	11	279,156.8	366,298.5	87,141.7
11	12	2,009,580.3	2,652,415.0	642,834.7

In [186...]: import matplotlib.ticker as mticker

```
supplier_monthly_data = df_purchased[df_purchased['Supplier Rank by
                                         'Product name' : 'nunique',
                                         'Total' : 'sum',
                                         'total discount 2': 'sum'}].r
supplier_monthly_data['Discount %'] = supplier_monthly_data['total

supplier_category_data = df_purchased[df_purchased['Supplier Rank b

Wecare = df_purchased[df_purchased['Supplier'] == 'Wecare'].groupby
                                         'Total' : 'sum',
                                         'total discount 2' : 'sum'}).re
Wecare['Discount %'] = Wecare['total discount 2'] / Wecare['Total']

fig, ax = plt.subplots(4,1, figsize=(10, 8))

sns.lineplot(data = supplier_monthly_data, x = 'month', y = 'total
sns.lineplot(data = supplier_monthly_data, x = 'month', y = 'Produc
supplier_category_data[['local', 'import']].plot(kind = 'line', ax =
sns.lineplot(data = Wecare,x = 'month',y = 'Discount %', ax= ax[3])
ax[3].set_title('Wecare Discount %')
ax[3].set_xlabel('Month')
ax[3].set_ylim(15, 30)
ax[3].grid(True)

ax[0].set_title('Total Purchased per Month')
ax[0].set_xlabel('')
ax[0].set_ylabel('')
ax[0].grid(True)
# ax[0].set_ylim(1000000, 7000000)

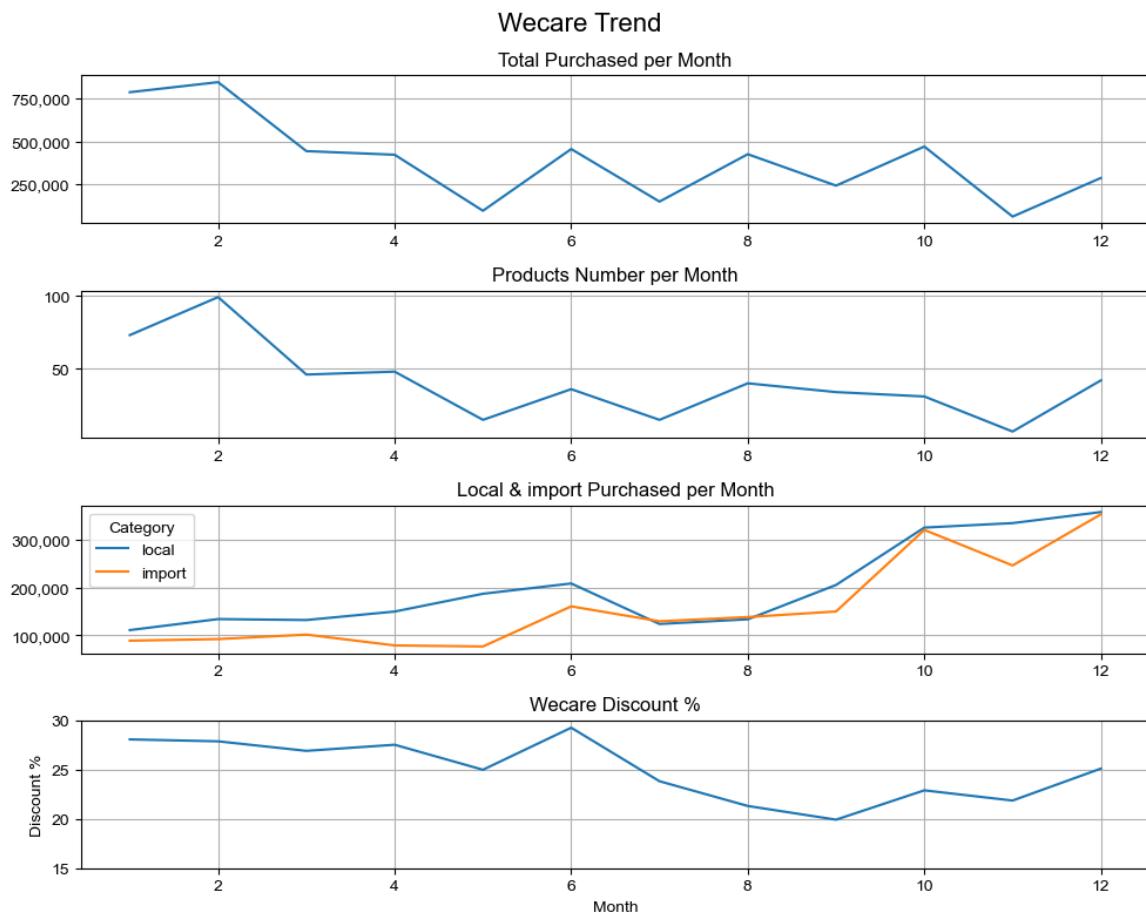
ax[1].set_title('Products Number per Month')
ax[1].set_xlabel('')
ax[1].set_ylabel('')
ax[1].grid(True)

ax[2].set_title('Local & import Purchased per Month')
ax[2].set_xlabel('')
ax[2].set_ylabel('')
ax[2].grid(True)

# avoid scientific notation
ax[0].yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f
ax[1].yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f
ax[2].yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f

# ax[0].yaxis.set_major_formatter(mticker.FuncFormatter(lambda x, _
```

```
# ax[2].yaxis.set_major_formatter(mticker.FuncFormatter(lambda x, _  
  
plt.suptitle('Wecare Trend', fontsize=16)  
plt.tight_layout()  
plt.savefig('Wecare Trend.jpg', bbox_inches = 'tight', dpi = 200)  
  
plt.show()
```



In [187]:

```
import matplotlib.ticker as mticker  
  
supplier_monthly_data = df_purchased[df_purchased['Supplier Rank by  
    'Product name' : 'nunique',  
    'Total' : 'sum',  
    'total discount 2': 'sum']].re  
supplier_monthly_data['Discount %'] = supplier_monthly_data['total  
  
supplier_category_data = df_purchased[df_purchased['Supplier Rank b  
  
Farouk = df_purchased[df_purchased['Supplier'] == 'الفرع'].g  
    'Total' : 'sum',  
    'total discount 2' : 'sum']).re  
Farouk['Discount %'] = Farouk['total discount 2'] / Farouk['Total']  
  
fig, ax = plt.subplots(4,1, figsize=(10, 8))  
  
sns.lineplot(data = supplier_monthly_data, x = 'month', y = 'total
```

```
sns.lineplot(data = supplier_monthly_data, x = 'month', y = 'Product')
supplier_category_data[['local', 'import']].plot(kind = 'line', ax = ax[1])
sns.lineplot(data = Farouk,x = 'month',y = 'Discount %', ax= ax[3])
ax[3].set_title('El Farouk Discount %')
ax[3].set_xlabel('Month')
ax[3].set_ylim(15, 30)
ax[3].grid(True)

ax[0].set_title('Total Purchased per Month')
ax[0].set_xlabel('')
ax[0].set_ylabel('')
ax[0].grid(True)
# ax[0].set_ylim(1000000, 7000000)

ax[1].set_title('Products Number per Month')
ax[1].set_xlabel('')
ax[1].set_ylabel('')
ax[1].grid(True)

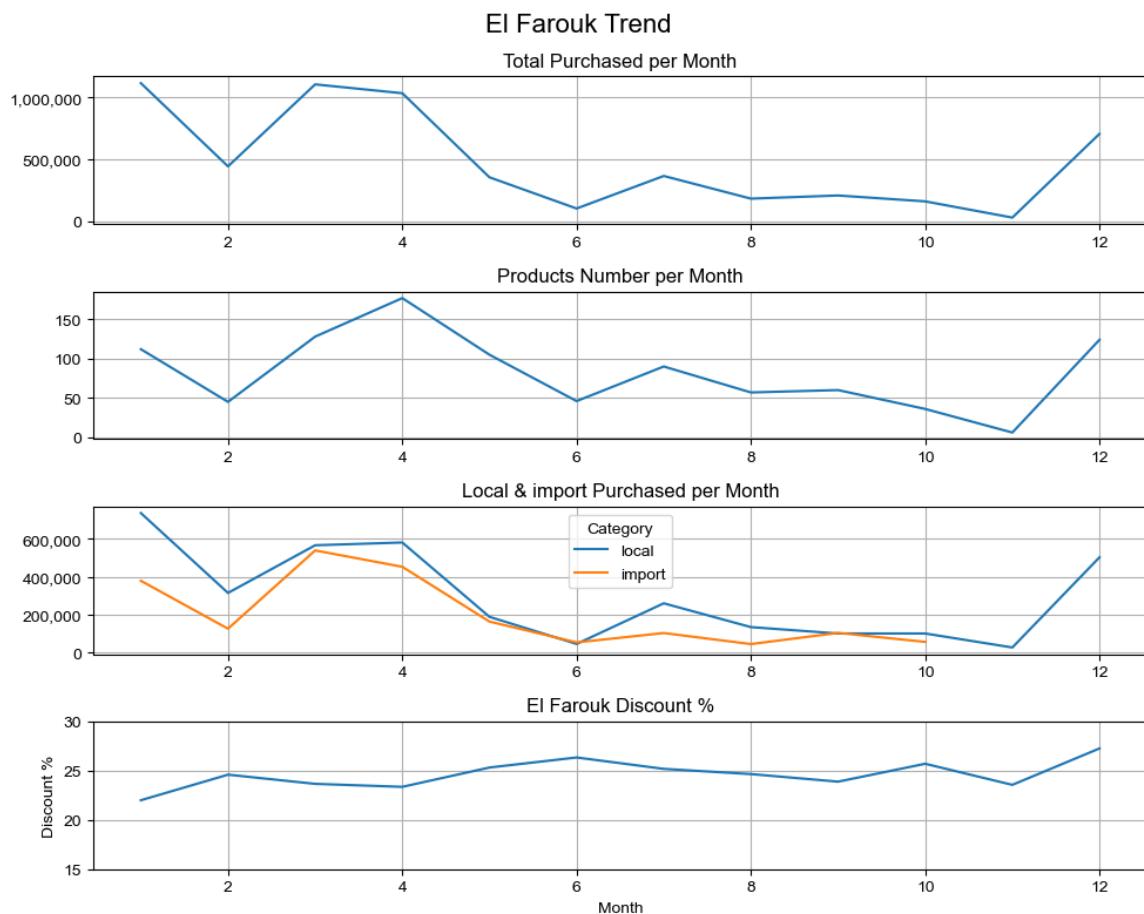
ax[2].set_title('Local & import Purchased per Month')
ax[2].set_xlabel('')
ax[2].set_ylabel('')
ax[2].grid(True)

# avoid scientific notation
ax[0].yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f'))
ax[1].yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f'))
ax[2].yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f'))

# ax[0].yaxis.set_major_formatter(mticker.FuncFormatter(lambda x, _
# ax[2].yaxis.set_major_formatter(mticker.FuncFormatter(lambda x, _

plt.suptitle('El Farouk Trend', fontsize=16)
plt.tight_layout()
plt.savefig('Farouk trend.jpg', bbox_inches = 'tight', dpi = 200)

plt.show()
```



In [188]:

```
import matplotlib.ticker as mticker

supplier_monthly_data = df_purchased[df_purchased['Supplier Rank by
                                         'Product name' : 'nunique',
                                         'Total' : 'sum',
                                         'total discount 2': 'sum']].r
supplier_monthly_data['Discount %'] = supplier_monthly_data['total

supplier_category_data = df_purchased[df_purchased['Supplier Rank b

ibn_sina = df_purchased[df_purchased['Supplier'] == 'ابن سينا فارما
                                         'Total' : 'sum',
                                         'total discount 2' : 'sum']).re
ibn_sina['Discount %'] = ibn_sina['total discount 2'] / ibn_sina['T

fig, ax = plt.subplots(4,1, figsize=(10, 8))

sns.lineplot(data = supplier_monthly_data, x = 'month', y = 'total
sns.lineplot(data = supplier_monthly_data, x = 'month', y = 'Produc
# sns.lineplot(data = supplier_category_data[['local', 'import']],x
supplier_category_data[['local', 'import']].plot(kind = 'line', ax
sns.lineplot(data = ibn_sina,x = 'month',y = 'Discount %', ax= ax[3]
ax[3].set_title('Ibn Sina Discount %')
ax[3].set_xlabel('Month')
ax[3].set_xlim(22, 30)
```

```
ax[3].grid(True)

ax[0].set_title('Total Purchased per Month')
ax[0].set_xlabel('')
ax[0].set_ylabel('')
ax[0].grid(True)
ax[0].set_ylim(1000000, 7000000)

ax[1].set_title('Products Number per Month')
ax[1].set_xlabel('')
ax[1].set_ylabel('')
ax[1].grid(True)

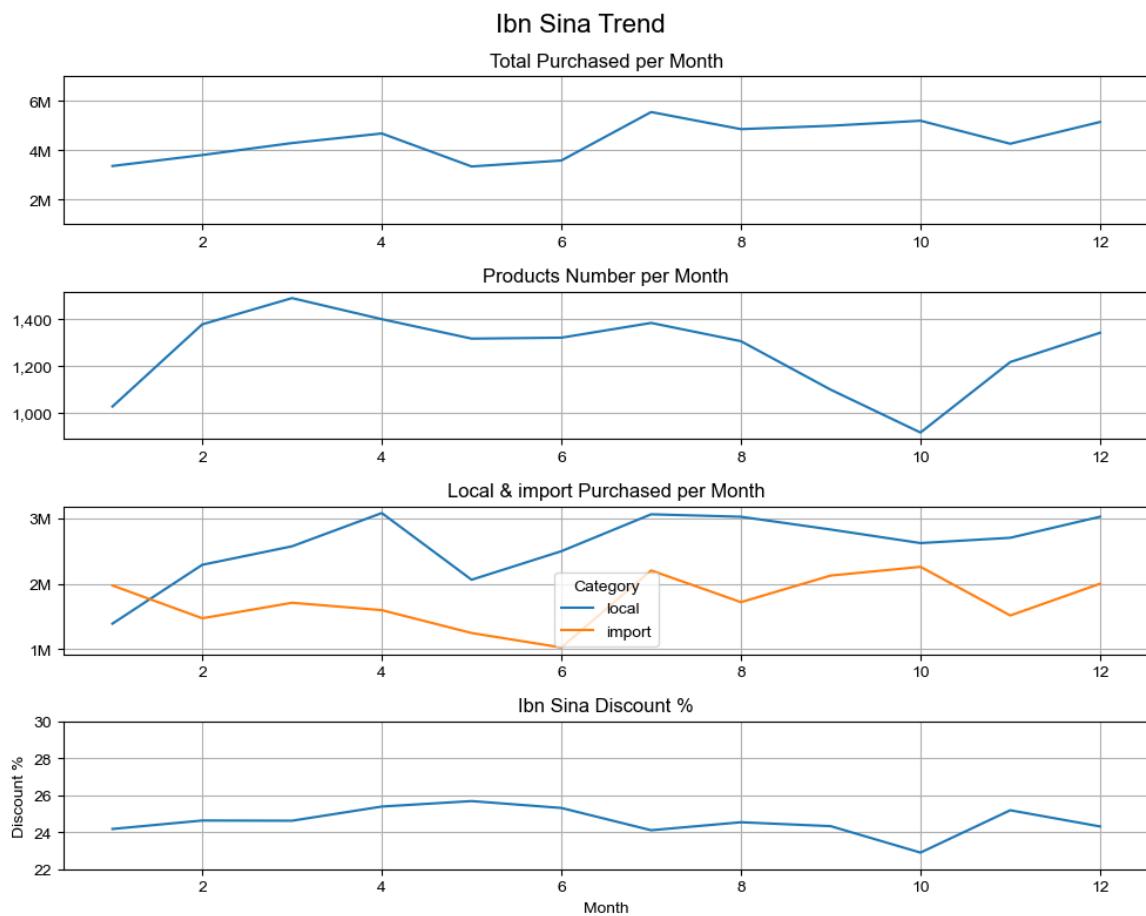
ax[2].set_title('Local & import Purchased per Month')
ax[2].set_xlabel('')
ax[2].set_ylabel('')
ax[2].grid(True)

# avoid scientific notation
ax[0].yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f'))
ax[1].yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f')
ax[2].yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f')

ax[0].yaxis.set_major_formatter(mticker.FuncFormatter(lambda x, _:
ax[2].yaxis.set_major_formatter(mticker.FuncFormatter(lambda x, _:

plt.suptitle('Ibn Sina Trend', fontsize=16)
plt.tight_layout()
plt.savefig('Ibn_sina_trend.jpg', bbox_inches = 'tight', dpi = 200)

plt.show()
```



```
In [189]: from scipy.stats import pearsonr
corr, _ = pearsonr(df_purchased['Discount %'], df_purchased['Quantity'])
print(f"Correlation between Discount and Quantity: {corr}")
```

Correlation between Discount and Quantity: 0.0728259230395191

```
In [190]: import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
import seaborn as sns
import arabic_reshaper
from bidi.algorithm import get_display
from adjustText import adjust_text

# Aggregate supplier metrics
supplier_data = df_purchased.groupby('Supplier').agg({
    'total after discount tax 2': 'sum',
    'Quantity': 'sum',
    'Total': 'sum',
    'Unit Price': 'mean',
    'total discount 2': 'sum'
}).reset_index()
supplier_data['Discount %'] = supplier_data['total discount 2'] / s

# Select top 10 suppliers by Total
top_suppliers = supplier_data.nlargest(11, 'Total')

# Prepare features
```

```
features = top_suppliers[['Discount %', 'Quantity', 'Total', 'Unit
scaler = StandardScaler()
scaled_features = scaler.fit_transform(features)

# KMeans clustering
kmeans = KMeans(n_clusters=6, random_state=42)
top_suppliers['Cluster'] = kmeans.fit_predict(scaled_features)

# PCA for 2D plot
pca = PCA(n_components=2)
pca_result = pca.fit_transform(scaled_features)
top_suppliers['PC1'] = pca_result[:, 0]
top_suppliers['PC2'] = pca_result[:, 1]

# Reshape Arabic names
top_suppliers['Supplier_ar'] = top_suppliers['Supplier'].apply(lambda x: x.replace(' ', '_'))

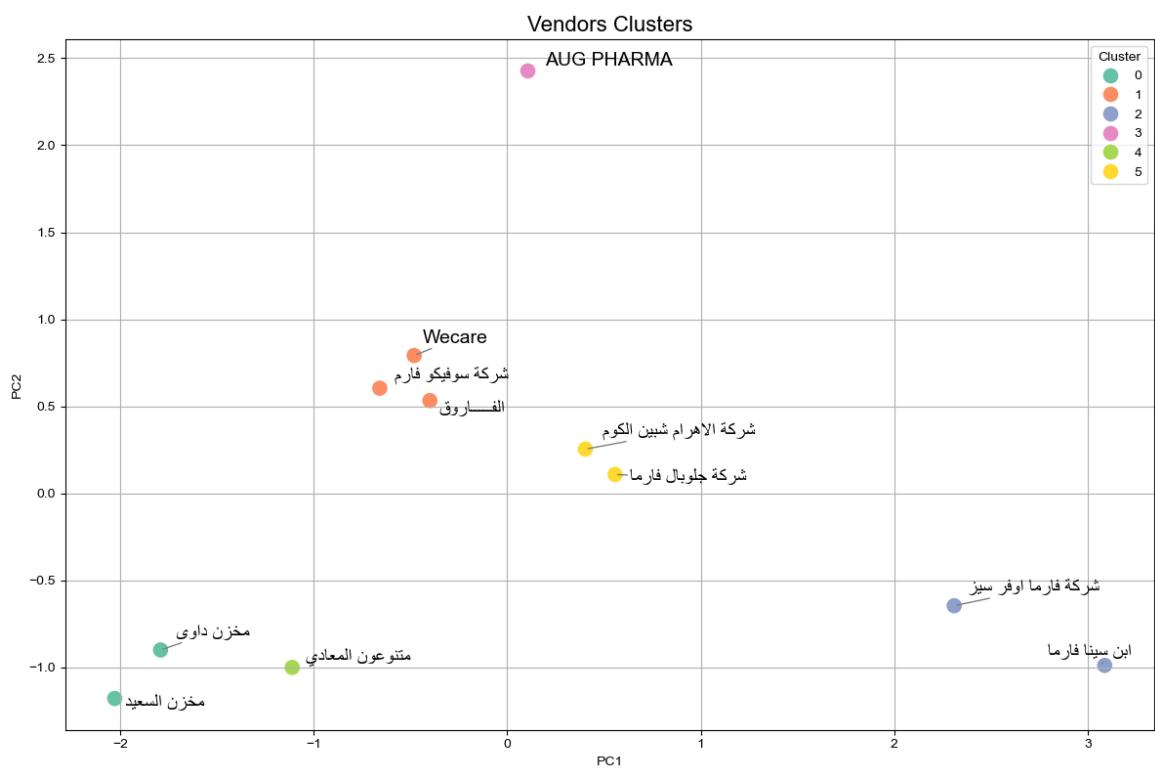
# Plot with non-overlapping labels
plt.figure(figsize=(12, 8))
sns.scatterplot(data=top_suppliers, x='PC1', y='PC2', hue='Cluster')

# Add adjusted labels
texts = []
for _, row in top_suppliers.iterrows():
    texts.append(plt.text(row['PC1'], row['PC2'], row['Supplier_ar'],
                        horizontalalignment='center',
                        verticalalignment='bottom',
                        rotation=45))

adjust_text(texts, arrowprops=dict(arrowstyle='-', color='gray', lw=1))

plt.title('Vendors Clusters', fontsize=16)
# plt.xlabel('الملون الرئيسي 1')
# plt.ylabel('الملون الرئيسي 2')
plt.grid(True)
plt.tight_layout()
plt.savefig('Vendors clusters.jpg', bbox_inches = 'tight', dpi = 200)

plt.show()
```



```
In [191...]: top_suppliers
```

Out[191...]

	Supplier	total after discount tax 2	Quantity	Total	Unit Price	total discount 2
10	ابن سينا فارما	53,135,119.5	493,936.7	70,413,196.8	138.3	17,278,077.4
40	شركة فارما اوفر سيز	43,213,845.1	396,075.6	57,262,833.3	149.8	14,048,988.2
34	شركة جلوبال فارما	19,423,966.0	190,439.4	25,645,432.6	134.4	6,221,466.5
30	شركة الاهرام شبين الكوم	17,587,243.4	163,987.6	23,432,662.9	171.6	5,845,419.5
55	مخزن السعيد	6,821,898.3	391.5	8,542,555.0	23,176.4	1,720,656.7
5	AUG PHARMA	4,947,294.1	74,780.0	8,448,029.0	109.0	3,500,734.9
20	الفاروق	5,804,709.8	82,144.0	7,653,719.0	80.5	1,849,009.3
9	Wecare	4,670,226.4	59,150.0	6,317,893.2	185.6	1,647,666.8
37	شركة سوفيكو فارم	4,078,066.4	37,736.6	5,360,520.5	141.9	1,282,454.1
62	مخزن داوى	4,339,712.2	3,584.0	5,320,831.5	16,109.6	981,119.3
49	متنوعون المعادى	4,506,066.3	37,026.1	4,964,999.4	143.7	458,933.1

In [192...]

```
# Ensure pandas prints all rows
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)

# Calculate mean and standard deviation
mean_discount = df_purchased['new discount adds'].mean()
std_discount = df_purchased['new discount adds'].std()

# Z-score calculation
df_purchased['z_discount'] = (df_purchased['new discount adds'] - m
                               ) / std

# Detect anomalies (more than 2 std deviations from the mean)
anomalies = df_purchased[np.abs(df_purchased['new discount adds']) >= 2 * std]

# Show full table of anomalies
# anomalies[['Supplier', 'Discount %', 'z_discount']].to_excel("dis

grouped_anomalies = anomalies.groupby('Supplier').agg({
    'new discount adds': ['count', 'mean', 'min', 'max'],
    'Discount %': ['mean', 'min', 'max'],
    'z_discount': ['mean', 'min', 'max']
})
```

```

        'z_discount': ['min', 'max']
    }).reset_index()

# Rename columns for clarity
grouped_anomalies.columns = ['Supplier', 'Anomaly Count', 'Avg Disc
                             'Min Z-Score', 'Max Z-Score']

# Sort from highest to lowest by Anomaly Count
grouped_anomalies = grouped_anomalies.sort_values(by='Anomaly Count')
grouped_anomalies # Show top 10 suppliers

```

Out[192...]

	Supplier	Anomaly Count	Avg Discount %	Min Discount %	Max Discount %	Min Z-Score	Max Z-Score
34	شركة جلوبال فارما	38797	25.5	-76.0	75.0	-0.0	-0.0
40	شركة فارما اوفر سيز	34779	5,638.4	4.4	96,739,055.7	-0.0	191.4
10	ابن سينا فارما	33609	5,377.7	2.1	97,401,460.7	-0.0	192.7
49	متنوعون المعادى	21933	9.0	2.0	510.0	-0.0	-0.0
19	الفاتح	6420	21.3	3.0	69.0	-0.0	-0.0
67	ميديكال فارما _ اونر نقمي	5276	22.9	3.0	70.0	-0.0	-0.0
50	متنوعون الهرم	5201	6.3	2.0	90.0	-0.0	-0.0
53	مخزن الأخوة المتحدون	4947	21.1	2.0	70.0	-0.0	-0.0
30	شركة الاهرام شبين الكوم	4930	28.3	3.0	68.0	-0.0	-0.0
37	شركة سوفيكو فارم	4261	30,905.2	6.0	131,568,212.6	-0.0	260.3
1	المتحده للصيادله	2995	22.2	10.0	100.0	-0.0	-0.0
61	مخزن جلوبال مصطفى	2903	21.1	3.0	70.0	-0.0	-0.0
33	شركة المصري	2798	22.9	2.0	70.0	-0.0	-0.0
44	شركة مالي ستورز فارما	2728	20.4	2.0	66.0	-0.0	-0.0
17	الشمس فارم	2634	25.1	7.0	72.0	-0.0	-0.0
60	مخزن جلوبال فارم د/ابراهيم الروبي	2316	20.4	2.0	56.0	-0.0	-0.0
56	مخزن السلام السنبلاويين	2208	28.9	3.0	71.0	-0.0	-0.0

14	الشافعي فارما للتجارة والتوزيع	2170	19.8	3.0	75.0	-0.0	-0.0
23	تريند فارما	1639	23.6	2.0	71.0	-0.0	-0.0
20	الفاروق	1416	27.1	2.0	62.0	-0.0	-0.0
26	رامكو فارم لتجارة الادوية	1389	22.1	7.0	37.5	-0.0	-0.0
45	فارما توداي	1098	23.0	4.0	76.0	-0.0	-0.0
22	المحمدية	799	25.1	5.0	72.0	-0.0	-0.0
18	الفتح لتجارة وتوزيع الادوية	611	28.7	3.0	57.0	-0.0	-0.0
9	Wecare	587	27.2	5.0	64.0	-0.0	-0.0
38	شركة شفاء	347	20.2	3.0	57.0	-0.0	-0.0
68	ميديكال فارما - اونر	324	24.0	3.0	97.0	-0.0	-0.0
32	شركة الماسة فارما	305	24.5	3.0	73.0	-0.0	-0.0
43	شركة ماкро فارما	284	21.8	9.0	65.0	-0.0	-0.0
55	مخزن السعيد	264	18.9	4.7	40.6	-0.0	-0.0
57	مخزن العربية فاقوس	264	29.1	4.0	61.0	-0.0	-0.0
15	الشركة العربية لتجارة الادوية والمستلزمات الطبية	154	34.3	14.0	68.5	-0.0	-0.0
7	Imported (Maadi)	153	16.3	3.8	32.0	-0.0	-0.0
36	شركة ستوك فارما جروب	146	28.5	10.0	43.0	-0.0	-0.0
62	مخزن داوى	141	19.0	3.3	45.0	-0.0	-0.0
52	محجوب فارم السنبلاويين	138	29.7	5.0	69.0	-0.0	-0.0
70	هاي ستورز فارما	101	22.0	3.0	58.0	-0.0	-0.0
3	مخزن 2020	89	18.5	4.0	30.0	-0.0	-0.0
59	مخزن تبارك فارم	78	24.4	2.0	62.0	-0.0	-0.0
28	سينا فارما ترید لتجارة والتوزيع	78	28.6	12.0	52.0	-0.0	-0.0

5	AUG PHARMA	63	40.7	10.0	51.0	-0.0	-0.0
69	نالي فارم انترناشيونال	56	23.0	10.0	65.0	-0.0	-0.0
6	EBDA PHARM	54	23.7	18.0	30.0	-0.0	-0.0
64	مخزن فارما بيور نيو	54	31.5	15.0	45.0	-0.0	-0.0
35	شركة ريو فارما	53	31.8	30.0	35.0	-0.0	-0.0
8	Samir Supplier	39	16.3	7.3	55.0	-0.0	-0.0
4	AKHNATON COMPANY	34	23.0	13.8	28.7	-0.0	-0.0
12	الاصدقاء فارم	28	11.9	3.0	23.0	-0.0	-0.0
41	شركة فارما ايمدج	26	21.5	12.0	27.0	-0.0	-0.0
42	شركة كوزموفارم	24	17.1	15.0	20.0	-0.0	-0.0
48	مت能夠ون 3	23	19.0	7.0	23.8	-0.0	-0.0
0	el farouk	22	19.5	10.0	25.0	-0.0	-0.0
65	مخزن كيان فارم لتجارة وتوزيع الادوية	17	25.1	12.0	34.0	-0.0	-0.0
66	مصر ميديكال	16	17.6	3.0	35.0	-0.0	-0.0
13	البدري تو ام للتجارة والتوزيع	14	18.0	3.0	30.0	-0.0	-0.0
54	مخزن الاهرام المطيرية	11	28.3	14.0	41.0	-0.0	-0.0
16	الشركة الوطنية للصيادلة	10	26.9	19.0	36.0	-0.0	-0.0
2	هيثم محمد	8	43.8	8.0	60.0	-0.0	-0.0
31	شركة الفتح شبين الكوم	8	33.0	26.0	41.0	-0.0	-0.0
47	كريم فارما للادوية	5	23.2	11.0	33.0	-0.0	-0.0
46	فارما لاند	4	10.0	10.0	10.0	-0.0	-0.0
21	القصواع	4	42.2	3.8	55.0	-0.0	-0.0

24	جاسمين فارما	3	40.0	40.0	40.0	-0.0	-0.0
27	سيلبيا فارم	3	20.0	3.0	33.0	-0.0	-0.0
11	اكتوبر فارما	2	36.6	20.0	53.3	-0.0	-0.0
51	مجموعة اتش اس فارما	1	13.3	13.3	13.3	-0.0	-0.0
63	مخزن سيف	1	2.3	2.3	2.3	-0.0	-0.0
25	ديفارت	1	40.0	40.0	40.0	-0.0	-0.0
29	شركة أم المصريين	1	33.0	33.0	33.0	-0.0	-0.0
58	مخزن العمودي للتجارة والتوزيع	1	35.1	35.1	35.1	-0.0	-0.0
39	شركة صيادلة المستقبل	1	20.0	20.0	20.0	-0.0	-0.0
71	هنا فارم	1	8.0	8.0	8.0	-0.0	-0.0

In [193...]

```
top_suppliers_anomlies = grouped_anomalies.merge(top_suppliers['Supplier'], left_index=True, right_index=True)
top_suppliers_anomlies
```

Out [193...]

	Supplier	Anomaly Count	Avg Discount %	Min Discount %	Max Discount %	Min Z-Score	Max Z-Score
2	شركة جلوبال فارما	38797	25.5	-76.0	75.0	-0.0	-0.0
1	شركة فارما اوفر سيز	34779	5,638.4	4.4	96,739,055.7	-0.0	191.4
0	ابن سينا فارما	33609	5,377.7	2.1	97,401,460.7	-0.0	192.7
10	متنوعون المعادي	21933	9.0	2.0	510.0	-0.0	-0.0
3	شركة الاهرام شبين الكوم	4930	28.3	3.0	68.0	-0.0	-0.0
8	شركة سوفيكو فارم	4261	30,905.2	6.0	131,568,212.6	-0.0	260.3
6	الفاروق	1416	27.1	2.0	62.0	-0.0	-0.0
7	Wecare	587	27.2	5.0	64.0	-0.0	-0.0
4	مخزن السعيد	264	18.9	4.7	40.6	-0.0	-0.0
9	مخزن داوي	141	19.0	3.3	45.0	-0.0	-0.0
5	AUG PHARMA	63	40.7	10.0	51.0	-0.0	-0.0

In [194...]

```
# discount_map = max_discount.set_index('Product name')['new discount']
# discount_map

# # df_purchased['new discount pot'] = df_purchased['Product name']

# Potential difference in percentage
# # df_purchased['Potential Diff %'] = (df_purchased['new discount'] - df_purchased['original discount']) / df_purchased['original discount'] * 100
# # df_purchased['potential total discount'] = df_purchased['new discount'] - df_purchased['original discount']

# # Value of missed opportunity per transaction
# df_purchased['Potential Value'] = df_purchased['Potential Diff %'] * df_purchased['original discount']

df_purchased['Date'] = pd.to_datetime(df_purchased['Date'])

# Manually create half-year period
df_purchased['Half_Year'] = df_purchased['Date'].apply(lambda x: f'{x.year}-{x.quarter}H-{x.month}H')

# Then group by this column
quarterly_potential = df_purchased.groupby('Half_Year').agg({
    'Total': 'sum',
    'potential value': 'sum',
    'total discount 2': 'sum'}
```

```
}).reset_index()

# Calculate discount percentages
quarterly_potential['Actual Discount %'] = quarterly_potential['total discount 2'] / quarterly_potential['Total'] * 100
quarterly_potential['Potential Extra Discount %'] = quarterly_potential['potential value'] / quarterly_potential['Total'] * 100

# plt.figure(figsize=(10, 6))
# plt.plot(quarterly_potential['Quarter'].astype(str), quarterly_potential['Actual Discount %'])
# plt.plot(quarterly_potential['Quarter'].astype(str), quarterly_potential['Potential Extra Discount %'])
# plt.title('Quarterly Discount Opportunity vs Actual')
# plt.xlabel('Quarter')
# plt.ylabel('Amount (EGP)')
# plt.legend()
# plt.grid(True)
# plt.tight_layout()
# plt.show()
```

Out[194...]

	Half_Year	Total	potential value	total discount 2	Actual Discount %	Potential Extra Discount %
0	2024-H1	110,200,226.5	6,318,035.2	27,128,585.4	24.6	5.7
1	2024-H2	156,312,215.8	10,758,234.5	36,959,190.8	23.6	6.9

In [195...]

```
# Convert to datetime
# df_purchased['Date'] = pd.to_datetime(df_purchased['Date'])

# Extract quarter
df_purchased['Quarter'] = df_purchased['Date'].dt.to_period('Q').asfreq('Q')

# Group by quarter
quarterly_potential = df_purchased.groupby('Quarter').agg({
    'Total': 'sum',
    'potential value': 'sum',
    'total discount 2': 'sum'
}).reset_index()

# Calculate discount percentages
quarterly_potential['Actual Discount %'] = quarterly_potential['total discount 2'] / quarterly_potential['Total'] * 100
quarterly_potential['Potential Extra Discount %'] = quarterly_potential['potential value'] / quarterly_potential['Total'] * 100

quarterly_potential
```

Out[195...]

	Quarter	Total	potential value	total discount 2	Actual Discount %	Potential Extra Discount %
0	2024Q1	58,047,340.2	3,169,604.1	14,296,909.7	24.6	5.5
1	2024Q2	52,152,886.3	3,148,431.1	12,831,675.7	24.6	6.0
2	2024Q3	66,312,472.4	4,457,098.5	15,911,471.7	24.0	6.7
3	2024Q4	89,999,743.4	6,301,135.9	21,047,719.1	23.4	7.0

In [196...]

df_purchased['total discount 2'].sum()

Out[196...]

64087776.23460171

In [197...]

```
plt.figure(figsize=(10, 6)) # Set the figure size here

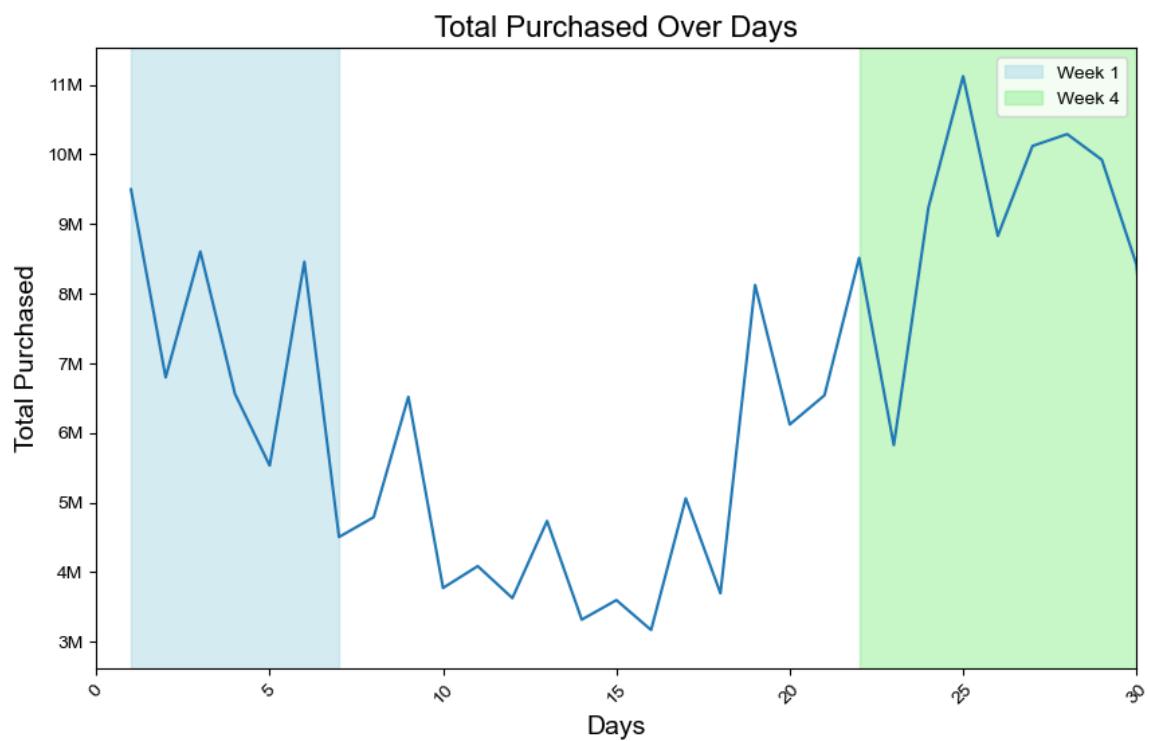
sns.lineplot(data= day_df, x = 'day', y = 'total purchased')
plt.axvspan(1, 7, color='lightblue', alpha=0.5, label='Week 1')
plt.axvspan(22, 30, color='lightgreen', alpha=0.5, label='Week 4')
# plt.text(4, 50, 'Week 1', horizontalalignment='center', fontsize=
# plt.text(25.5, 50, 'Week 4', horizontalalignment='center', fontsize=)

plt.gca().yaxis.set_major_formatter(mticker.FuncFormatter(lambda x,
    plt.title('Total Purchased Over Days', fontsize=16)
plt.xlabel('Days', fontsize=14)
plt.ylabel('Total Purchased', fontsize=14)
plt.xticks(rotation=45)
plt.xlim(0, 30)
plt.savefig('Daily Purchased.jpg', bbox_inches = 'tight', dpi = 200
# sns.despine()

plt.legend()
```

Out[197...]

<matplotlib.legend.Legend at 0x13fcba410>



```
In [198]: df_purchased.head(10)
```

Out [198...]

	Date	Bill No	Store ID	Type	Store Name	Product id	Product name	Si
0	2024-01-01	2023108920907	4	Purchase	وياك صيدلية الهرم	3083	EXFORGE 5MG/160MG 14 F.C. TAB.	
1	2024-01-01	2023108920907	4	Purchase	وياك صيدلية الهرم	10393	GLAPTIVIA PLUS 50/1000MG 30 F.C.TAB.	
2	2024-01-01	2023108920907	4	Purchase	وياك صيدلية الهرم	3080	EXFORGE 10MG/160MG 14 F.C. TAB.	
3	2024-01-01	10253	4	Purchase	وياك صيدلية الهرم	11444	ERALONER 25 MG 20 F.C.TAB.	
4	2024-01-01	10253	4	Purchase	وياك صيدلية الهرم	8683	PANTOLOC 20MG 14 E.C. TAB.	
5	2024-01-01	10253	4	Purchase	وياك صيدلية الهرم	5177	CARDURA 4 MG 14 TAB.	
6	2024-01-01	10253	4	Purchase	وياك صيدلية الهرم	5756	AMARYL 4 MG 30 TAB.	
7	2024-01-01	10253	4	Purchase	وياك صيدلية الهرم	18453	DIAMICRON 60 MG 30 M.R. SCORED TAB.	
8	2024-01-01	10253	4	Purchase	وياك صيدلية الهرم	5312	ATOR 10MG 7 TAB.	
9	2024-01-01	10253	4	Purchase	وياك صيدلية الهرم	3044	ATOR 40MG 10 F.C. TAB.	

In [199...]

```
fig, ax = plt.subplots(3,1, figsize = (10,10))

df_bill_month = df_purchased[df_purchased['month'].isin([1,2,3,4,5,
ax[1].plot(df_bill_month['month'], df_bill_month['Bill Count'], color='blue')
ax[1].set_title('Bills Count Per Month', fontsize=16)
ax[1].set_xlabel('')
ax[1].set_xlim(300,2000)
# ax[1].grid()
sns.despine(ax = ax[1])
ax[1].axhline(y = df_bill_month['Bill Count'].mean(), color='green')
```

```
ax[1].text(x=len(df_bill_month['month']) + 1, y=df_bill_month['Bill Count'].mean(), color='red', verticalalignment='bottom', horizontalalignment='right', fontweight='bold')

max_value = df_bill_month['Bill Count'].max()
min_value = df_bill_month['Bill Count'].min()
max_month = df_bill_month['month'][df_bill_month['Bill Count'].idxmax()]
min_month = df_bill_month['month'][df_bill_month['Bill Count'].idxmin()]

# Mark maximum point
ax[1].plot(max_month, max_value, 'bo', markersize=5) # Red point for max
ax[1].text(max_month, max_value + 100, f'Max: {max_value}', color='red', fontweight='bold')

# Mark minimum point
# ax[1].plot(min_month, min_value, 'ro', markersize=5) # Blue point for min
# ax[1].text(min_month, min_value - 200, f'Min: {min_value}', color='blue', fontweight='bold')
ax[1].axvspan(9, 12, color='lightblue', alpha=0.5, label='Q4')

for i, row in df_bill_month.iterrows():
    if row['Bill Count'] < df_bill_month['Bill Count'].mean():
        ax[1].plot(row['month'], row['Bill Count'], 'o', color='red')
    else:
        ax[1].plot(row['month'], row['Bill Count'], 'o', color='green')

df_supplier_month = df_purchased[df_purchased['month'].isin([1,2,3,4,5,6,7,8,9,10,11,12])]

ax[2].plot(df_supplier_month['month'], df_supplier_month['Supplier Count'])

ax[2].set_title('Supplier Count Per Month', fontsize=16)
ax[2].set_xlabel('')
ax[2].set_ylabel('Supplier Count')
sns.despine(ax = ax[2])
ax[2].axhline(y = df_supplier_month['Supplier Count'].mean(), color='black', linestyle='dashed')
ax[2].text(x=len(df_bill_month['month']) + 1, y=df_supplier_month['Supplier Count'].mean(), color='red', verticalalignment='bottom', horizontalalignment='right', fontweight='bold')

max_value_2 = df_supplier_month['Supplier Count'].max()
min_value_2 = df_supplier_month['Supplier Count'].min()
max_month_2 = df_supplier_month['month'][df_supplier_month['Supplier Count'].idxmax()]
min_month_2 = df_supplier_month['month'][df_supplier_month['Supplier Count'].idxmin()]

# Mark maximum point
ax[2].plot(max_month_2, max_value_2, 'bo', markersize=5) # Red point for max
ax[2].text(max_month_2 + 0.2, max_value_2 + 2, f'Max: {max_value_2}', color='red', fontweight='bold')

# Mark minimum point
ax[2].plot(min_month_2, min_value_2, 'ro', markersize=5) # Blue point for min
ax[2].text(min_month_2, min_value_2 - 4, f'Min: {min_value_2}', color='blue', fontweight='bold')

for i, row in df_supplier_month.iterrows():
    if row['Supplier Count'] < df_supplier_month['Supplier Count'].mean():
        ax[2].plot(row['month'], row['Supplier Count'], 'o', color='red')
    else:
        ax[2].plot(row['month'], row['Supplier Count'], 'o', color='green')
```

```
df_purchasing_month = df_purchased[df_purchased['month'].isin([1,2,3])].copy()
ax[0].plot(df_purchasing_month['month'], df_purchasing_month['Total Purchased'])
ax[0].set_title('Total Purchased Per Month', fontsize=16)
ax[0].set_xlabel('')
sns.despine(ax = ax[0])
ax[0].yaxis.set_major_formatter(mticker.StrMethodFormatter('{x:,.0f}'))
ax[0].axhline(y = df_purchasing_month['Total Purchased'].mean(), color='red')
ax[0].text(x=len(df_bill_month['month']) + 1, y=df_purchasing_month['Total Purchased'].mean(), s=f"AVG: {int(df_purchasing_month['Total Purchased'].mean())}", verticalalignment='bottom', horizontalalignment='right', fontweight='bold')
ax[0].set_ylim(7000000,35000000)
ax[0].axvspan(9, 12, color='lightblue', alpha=0.5, label='Q4')

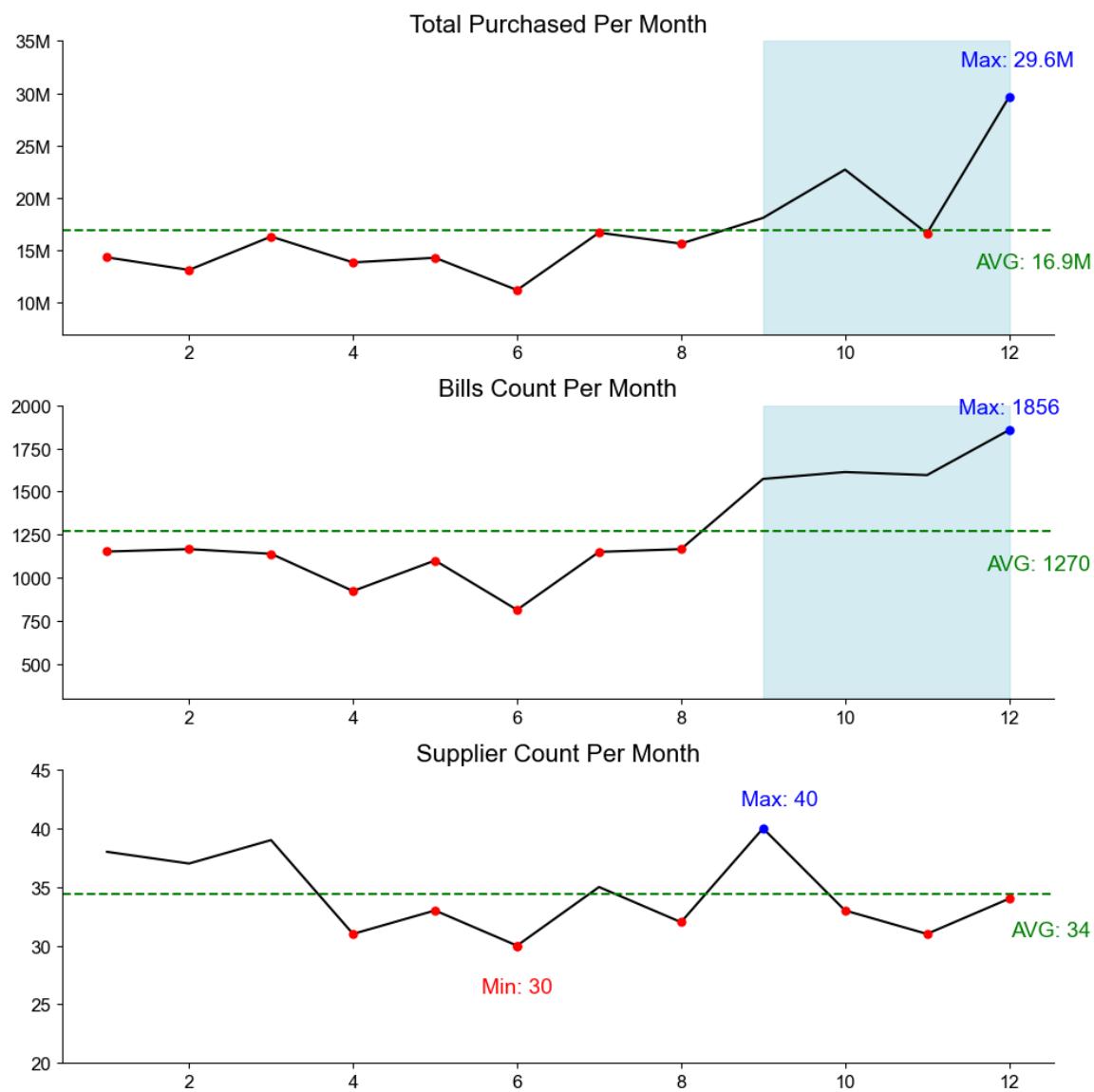
def millions(x, pos):
    return f'{x * 1e-6:.0f}M'
ax[0].yaxis.set_major_formatter(FuncFormatter(millions))

max_value_0 = df_purchasing_month['Total Purchased'].max()
min_value_0 = df_purchasing_month['Total Purchased'].min()
max_month_0 = df_purchasing_month['month'][df_purchasing_month['Total Purchased'].idxmax()]
min_month_0 = df_purchasing_month['month'][df_purchasing_month['Total Purchased'].idxmin()]

# Mark maximum point
ax[0].plot(max_month_0, max_value_0, 'bo', markersize=5) # Red point
ax[0].text(max_month_0 + 0.1, max_value_0 + 3000000, f'Max: {int(max_value_0)}')
# Mark minimum point
# ax[0].plot(min_month_0, min_value_0, 'ro', markersize=5) # Blue point
# ax[0].text(min_month_0, min_value_0 - 1800000, f'Min: {int(min_value_0)}')

for i, row in df_purchasing_month.iterrows():
    if row['Total Purchased'] < df_purchasing_month['Total Purchased'].mean():
        ax[0].plot(row['month'], row['Total Purchased'], 'o', color='red')
    else:
        ax[0].plot(row['month'], row['Total Purchased'], 'o', color='green')

for a in ax:
    a.tick_params(labelsize=12) # Increase tick label size
    a.set_xlabel(a.get_xlabel(), fontsize=12) # X-axis label (if any)
    a.set_ylabel(a.get_ylabel(), fontsize=12) # Y-axis label (if any)
plt.tight_layout()
# plt.grid()
plt.savefig('over view.jpg', bbox_inches = 'tight', dpi = 200)
plt.show()
```



```
In [200]: print(df_purchased['total after discount tax 2'].sum())
print(df_purchased['Total'].sum())
print(df_purchased['total discount 2'].sum())
# print(df_purchased['total after discount tax 2'].sum() / df_purchased['Total'].sum())
print(df_purchased['total discount 2'].sum() / df_purchased['Total'].sum())

202424666.05273923
266512442.2873409
64087776.23460171
24.046823362004748
```

```
In [201]: df_purchased['Supplier'].nunique()
```

```
Out[201]: 72
```

```
In [202]: retail_df = bills_df[bills_df['total after discount tax 2'] < 25000]
retail_count = retail_df['Supplier'].value_counts().reset_index()
retail_supplier_sum = retail_df.groupby(['Supplier', 'Vendor Type'])
    'Total' :
    'total dis
    'potential
retail_supplier_sum['Discount %'] = retail_supplier_sum['total disc
```

```
retail_count_sum = retail_count[retail_count['count'] >= 25].merge(  
    retail_count_sum['potential diff %'] = (retail_count_sum['potential  
  
print(round(retail_count_sum[retail_count_sum['Discount %'] < 20] ['  
print(round(retail_count_sum[retail_count_sum['Discount %'] < 20] ['  
  
# retail_count_sum  
retail_count_sum[retail_count_sum['Discount %'] < 20]  
  
print(round(retail_count_sum[retail_count_sum['Discount %'] < 20] ['  
print(round(retail_count_sum[retail_count_sum['Discount %'] < 20] ['  
  
    retail_count_sum[retail_count_sum['Discount %'] < 20][[['Supplier', '  

```

16179058

6273

2620202

5341671

Out [202...]

	Supplier	count	total after discount tax 2	total discount 2	Discount %	potential total discount	potential diff %
0	متنوعون المعادي	2782	4,428,014.9	455,749.9	9.3	1,352,046.4	18.4
3	متنوعون الهرم	1067	2,856,426.6	139,782.7	4.7	835,039.1	23.2
5	الفاتح	684	2,260,614.1	548,870.0	19.5	865,679.0	11.3
7	مخزن الأخوة المتحدون	555	1,426,686.4	336,933.5	19.1	532,545.6	11.1
8	شركة مالقي ستورز فارما	370	1,207,925.2	245,457.1	16.9	418,978.9	11.9
10	شركة المصري	255	709,829.8	175,385.4	19.8	258,949.1	9.4
12	الشافعي فارما للتجارة والتوزيع	202	905,322.2	194,794.7	17.7	326,015.0	11.9
15	مخزن جلوبال مصطففي	177	776,333.5	180,858.2	18.9	270,483.4	9.4
20	Imported (Maadi)	55	417,729.0	75,399.0	15.3	103,549.7	5.7
21	مخزن السعيد	40	583,050.0	127,087.0	17.9	152,129.9	3.5
24	شركة ماкро فارما	32	65,873.3	15,862.0	19.4	24,829.6	11.0
25	مخزن جلوبال فارم د/ ابراهيم الروبي	29	354,733.0	80,123.6	18.4	128,344.1	11.1
26	شركة شفاء	25	186,519.6	43,899.2	19.1	73,081.5	12.7

In [203...]

5341671 – 2620202

Out [203...]

In [204...]

```

import matplotlib.pyplot as plt
import arabic_reshaper
from bidi.algorithm import get_display

# Arabic reshaping function

```

```
def reshape_arabic(text):
    return get_display(arabic_reshaper.reshape(text))

# Filter and round data
filtered_df3 = retail_count_sum[retail_count_sum['Discount %'] < 20]
filtered_df3 = filtered_df3.round(2)

# Reshape Arabic in Supplier column
filtered_df3['Supplier'] = filtered_df3['Supplier'].apply(reshape_a

# Plot table
fig, ax = plt.subplots(figsize=(20, 6))
ax.axis('off')

# Create the table
table = ax.table(cellText=filtered_df3.values,
                  colLabels=filtered_df3.columns,
                  cellLoc='center',
                  loc='center')

# Format table
table.auto_set_font_size(False)
table.set_fontsize(12) # Base font size
table.scale(1.2, 1.5)

# Increase font size for headers
for (row, col), cell in table.get_celld().items():
    if row == 0:
        cell.set_fontsize(14)
        cell.set_text_props()
    elif col == 0: # Supplier column
        cell.set_fontsize(13)

# Save to PNG
plt.savefig('retail_discount_table.png', bbox_inches='tight', dpi=300)
plt.close()
```

In [205...]

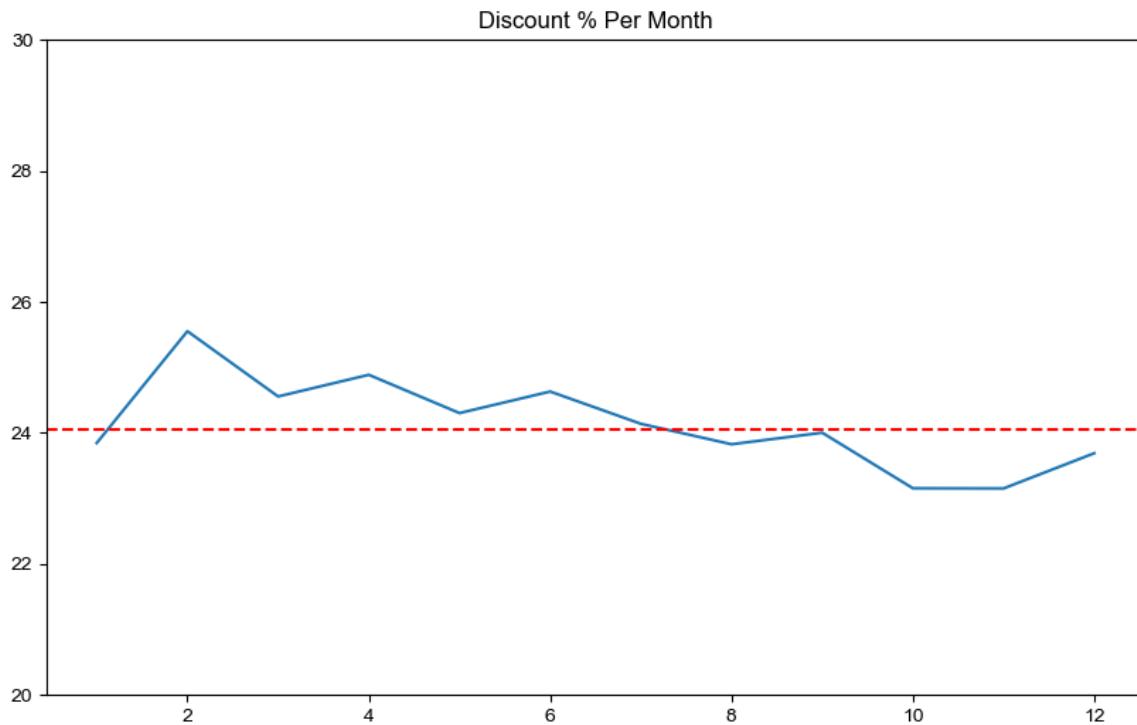
```
plt.figure(figsize=(10, 6)) # Set the figure size here

Discount_month_plot = df_discount_month.plot(kind = 'line', x = 'month')

Discount_month_plot.set_title('Discount % Per Month')
Discount_month_plot.set_xlabel('')
Discount_month_plot.get_legend().remove()
Discount_month_plot.axhline(y = (df_discount_month['total discount']/100).mean(), color='red', linestyle='dashed')
plt.ylim(20,30)
```

Out[205...]

(20.0, 30.0)
<Figure size 1000x600 with 0 Axes>



In [206]:

```
import matplotlib.pyplot as plt

# Plot the line chart
Discount_month_plot = df_discount_month.plot(kind='line', x='month')
Discount_month_plot.set_title('Discount % Per Month', fontsize=16)
Discount_month_plot.set_xlabel('')
Discount_month_plot.get_legend().remove()

# Calculate and plot average line
avg_discount = (df_discount_month['total discount 2'].sum() / df_discount_month.shape[0])
Discount_month_plot.axhline(y=avg_discount, color='red', linestyle='dashed')

#  Add label to average line
plt.text(df_discount_month['month'].iloc[-1], avg_discount + 0.2, f'Avg: {avg_discount:.1f}%', color='red', fontweight='bold', fontstyle='italic')

# Get max and min points
max_idx = df_discount_month['Discount %'].idxmax()
min_idx = df_discount_month['Discount %'].idxmin()

max_month = df_discount_month.loc[max_idx, 'month']
min_month = df_discount_month.loc[min_idx, 'month']
max_value = df_discount_month.loc[max_idx, 'Discount %']
min_value = df_discount_month.loc[min_idx, 'Discount %']

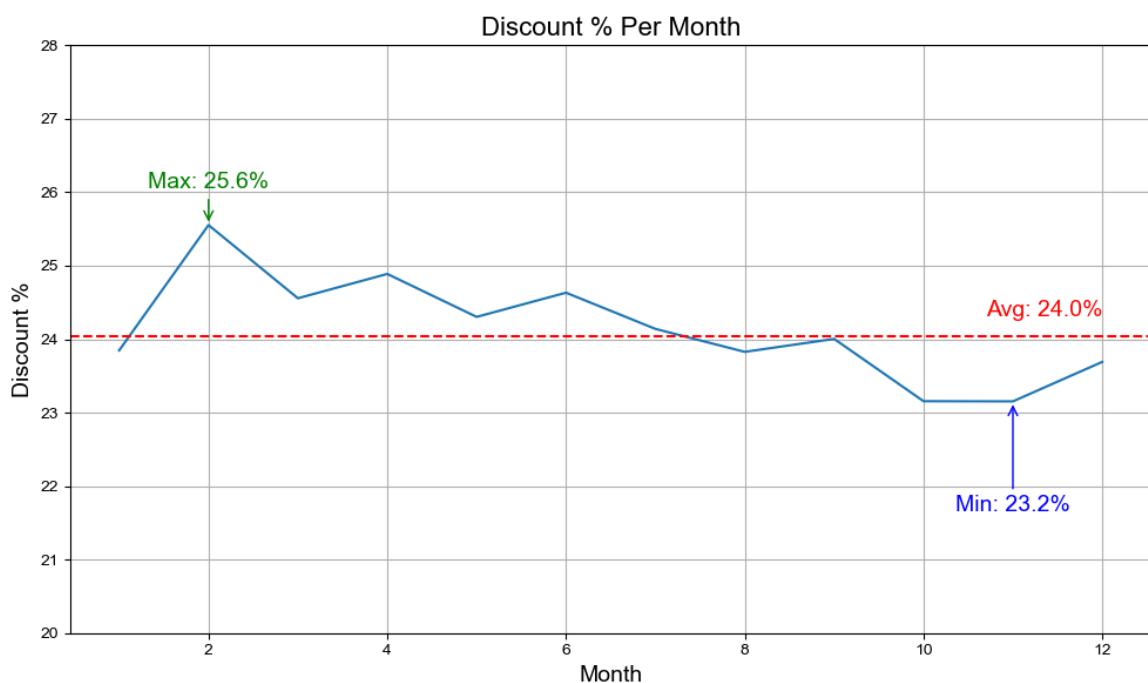
#  Annotate max point
plt.annotate(f'Max: {max_value:.1f}%', xy=(max_month, max_value),
            xytext=(max_month, max_value + 0.5),
            arrowprops=dict(arrowstyle='->', color='green'),
            ha='center', color='green', fontsize=14)

#  Annotate min point
plt.annotate(f'Min: {min_value:.1f}%', xy=(min_month, min_value),
            xytext=(min_month, min_value - 1.5),
```

```
        arrowprops=dict(arrowstyle='->', color='blue'),
        ha='center', color='blue', fontsize=14)

plt.ylim(20, 28)
plt.xlabel('Month', fontsize=14)
plt.ylabel('Discount %', fontsize=14)
plt.grid()
plt.tight_layout()
plt.savefig('Discount %.jpg', bbox_inches = 'tight', dpi = 200)

plt.show()
```



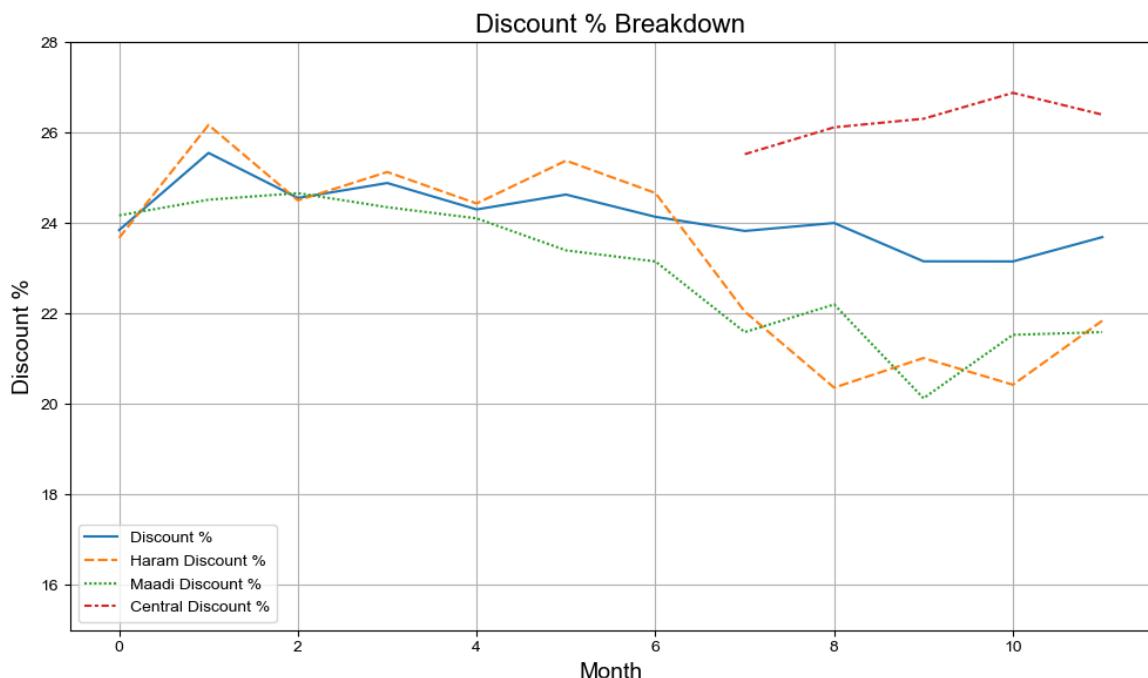
In [207...]

```
plt.figure(figsize=(10, 6)) # Set the figure size here

sns.lineplot(data = monthly_discount[['Discount %', 'Haram Discount']])
# sns.lineplot(data = haram_monthly_discount[['month', 'Haram Discou

plt.ylim(15,28)
plt.title('Discount % Breakdown', fontsize=16)
plt.xlabel('Month', fontsize=14)
plt.ylabel('Discount %', fontsize=14)
plt.grid()

plt.savefig('discount breakdown.jpg', bbox_inches = 'tight', dpi =
plt.tight_layout()
plt.show()
```



```
In [208... vendors = df_purchased.groupby('Vendor Type').agg({'total after dis
                           'Total' : 'sum',
                           'total discount 2' : 'sum'}).re
vendors['Discount %'] = vendors['total discount 2'] / vendors['Total']
vendors
```

Out [208...]

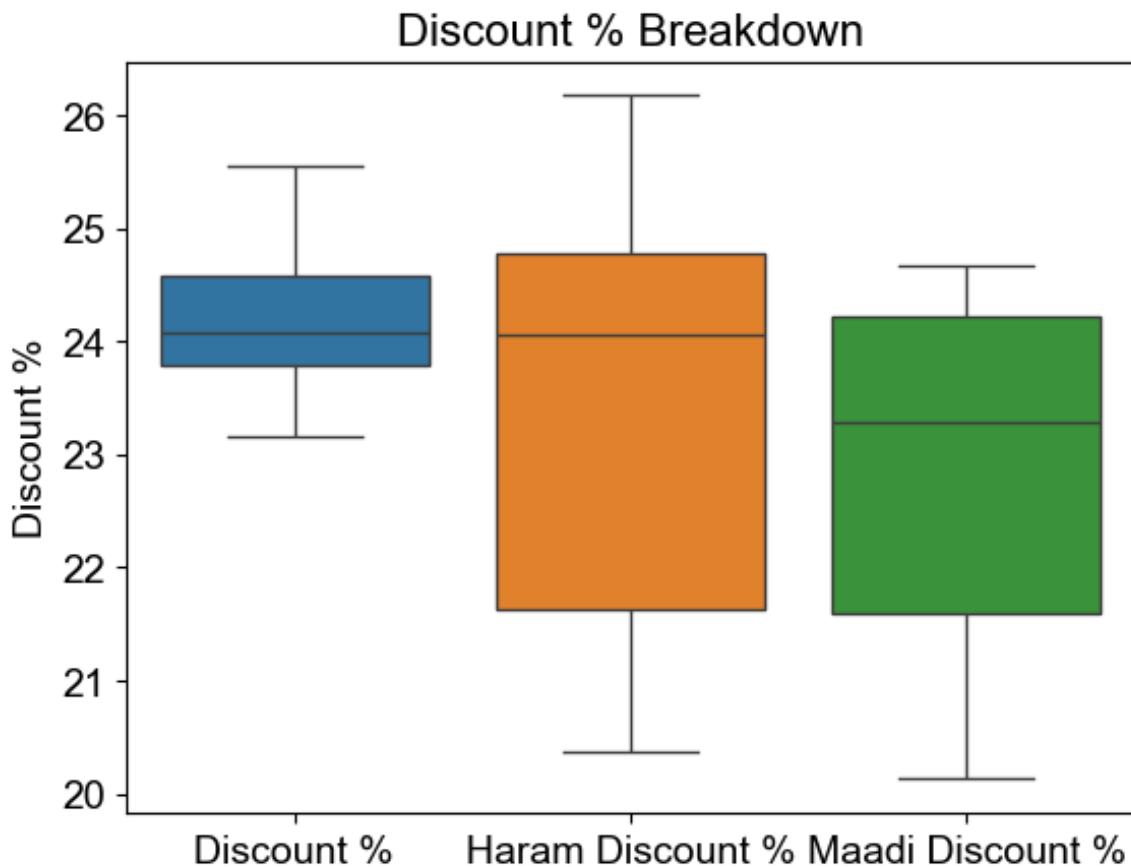
	Vendor Type	total after discount tax 2	Total	total discount 2	Discount %
1	Distribution	105,374,325.0	141,484,579.6	36,110,254.5	25.5
3	Retail	39,827,339.6	49,701,315.5	9,873,975.9	19.9
0	Bulk	32,373,878.9	43,659,183.1	11,285,304.2	25.8
2	NA	24,849,122.5	31,667,364.1	6,818,241.6	21.5

In []:

```
In [209... sns.boxplot(data = monthly_discount[['Discount %', 'Haram Discount %', 'Maadi Discount %', 'Central Discount %']])
plt.title('Discount % Breakdown', fontsize=16)
plt.ylabel('Discount %', fontsize=14)
plt.xticks(fontsize=14)
plt.yticks(fontsize=14)
```

Out [209...]

```
(array([19., 20., 21., 22., 23., 24., 25., 26., 27.]),
 [Text(0, 19.0, '19'),
  Text(0, 20.0, '20'),
  Text(0, 21.0, '21'),
  Text(0, 22.0, '22'),
  Text(0, 23.0, '23'),
  Text(0, 24.0, '24'),
  Text(0, 25.0, '25'),
  Text(0, 26.0, '26'),
  Text(0, 27.0, '27')])
```



```
In [210... store_discount = df_purchased.groupby('Store Name').agg({'total aft
'Total' : 'sum',
'total discount 2' : 'sum'}).re
store_discount['Discount %'] = store_discount['total discount 2'] /
store_discount
```

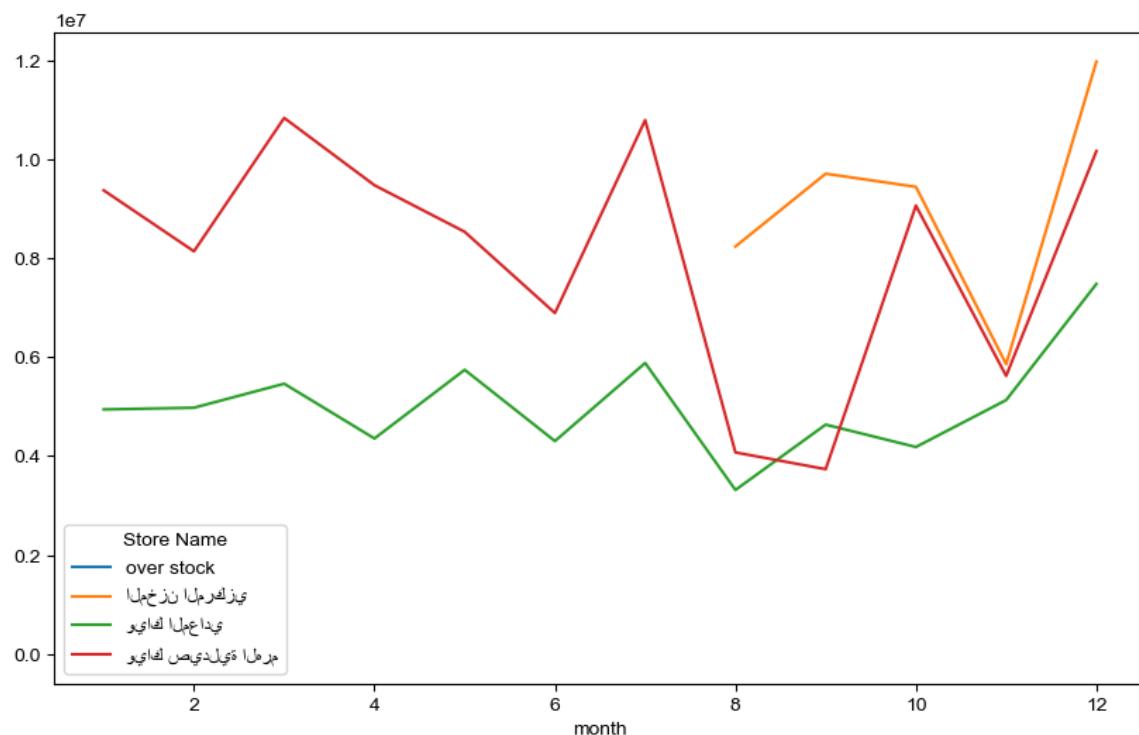
Out [210...]

	Store Name	total after discount tax 2	Total	total discount 2	Discount %
3	وياك صيدلية الهرم	96,744,086.9	126,688,656.8	29,944,570.0	23.6
2	وياك المعادى	60,438,501.7	78,497,814.9	18,059,313.2	23.0
1	المخزن المركزي	45,242,076.1	61,325,970.6	16,083,894.5	26.2
0	over stock	1.3	0.0	-1.3	-inf

In [211...]

```
df_purchased.groupby(['month', 'Store Name']).agg({'total after dis
'Total' : 'sum',
'total discount 2' : 'sum'}).re
```

Out [211... <Axes: xlabel='month'>



In [212]: monthly_discount

Out[212]:

	month	total after discount tax 2	Total	total discount 2	Discount %	Haram Discount %	M Disc
0	1	14,321,341.0	18,806,032.0	4,484,691.0	23.8	23.7	
1	2	13,122,461.0	17,626,735.0	4,504,274.0	25.6	26.2	
2	3	16,306,628.0	21,614,573.0	5,307,945.0	24.6	24.5	
3	4	13,837,457.0	18,422,510.0	4,585,053.0	24.9	25.1	
4	5	14,283,965.0	18,870,100.0	4,586,136.0	24.3	24.4	
5	6	11,199,789.0	14,860,276.0	3,660,488.0	24.6	25.4	
6	7	16,681,366.0	21,989,756.0	5,308,390.0	24.1	24.7	
7	8	15,632,065.0	20,521,993.0	4,889,928.0	23.8	22.0	
8	9	18,087,569.0	23,800,723.0	5,713,154.0	24.0	20.4	
9	10	22,699,714.0	29,540,042.0	6,840,328.0	23.2	21.0	
10	11	16,619,802.0	21,627,328.0	5,007,525.0	23.2	20.4	
11	12	29,632,508.0	38,832,374.0	9,199,866.0	23.7	21.8	

In [213]: df_category

Out [213...]

	Store Name	Total Return	Total Purchased	Return %
0	over stock	9,962.3	1.3	100.0
1	المخزن المركزي	1,652,649.4	45,242,076.1	3.5
2	وياك المعادي	1,543,014.1	60,438,501.7	2.5
3	وياك صيدلية الهرم	4,014,969.7	96,744,086.9	4.0

In [214...]

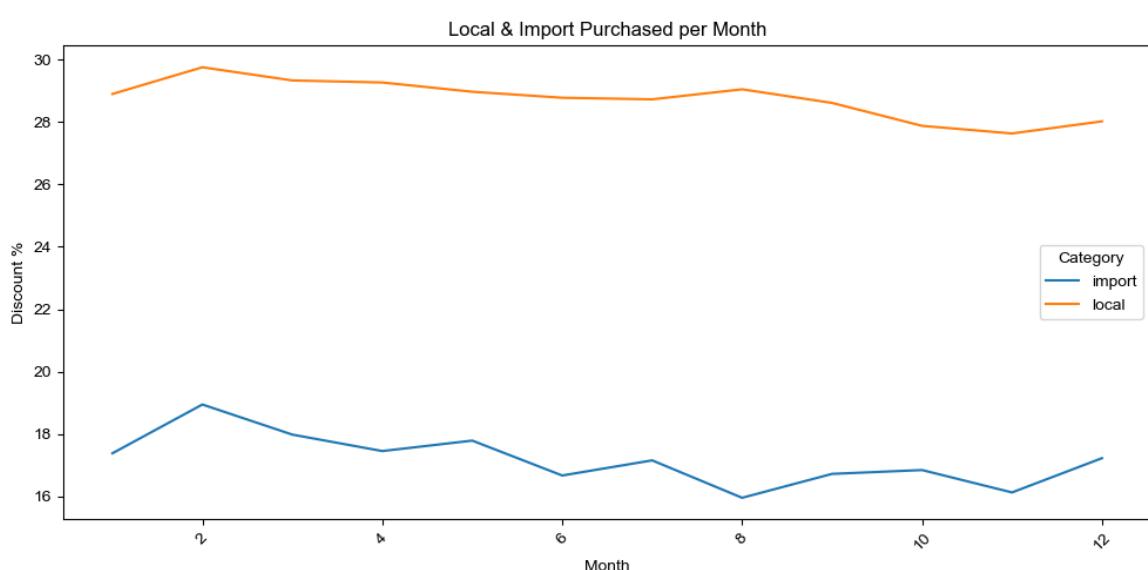
```
df_category_purchased_data = df_purchased[df_purchased['Category'].isin(['Food', 'Clothing', 'Electronics', 'Home & Garden'])].groupby('Category').agg({'Product name' : 'nunique', 'Total' : 'sum', 'total discount 2': 'sum'}).reset_index()
df_category_purchased_data['Discount %'] = df_category_purchased_data['total discount 2'] / df_category_purchased_data['Total'] * 100
df_category_purchased_data
```

Out [214...]

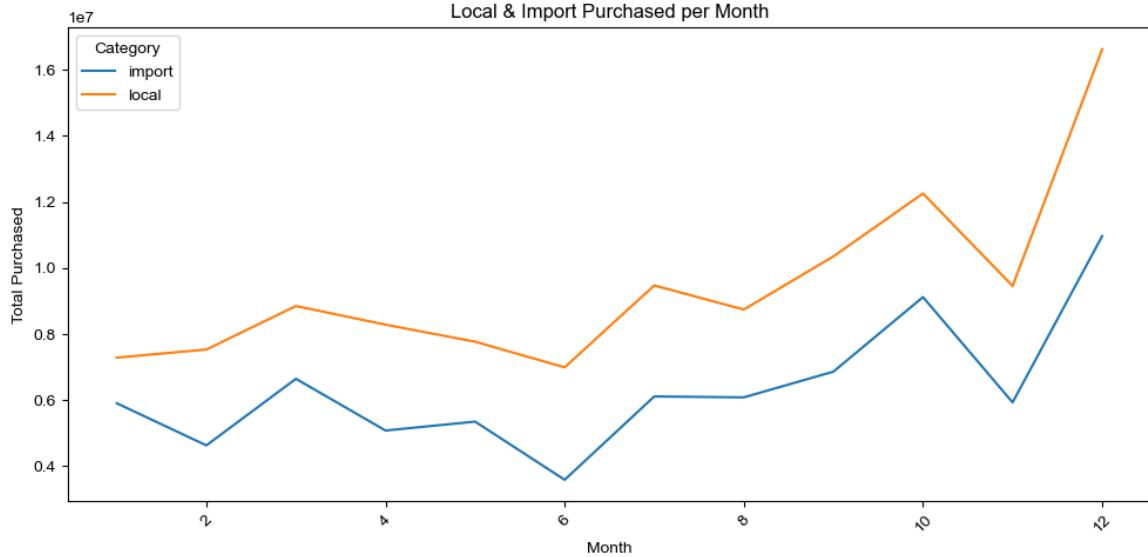
	month	Category	total after discount tax 2	Product name	Total	total discount 2	Discount %
0	1	import	5,906,252.5	431	7,148,861.1	1,242,608.6	
1	1	local	7,287,427.0	2332	10,249,443.3	2,962,016.2	
2	2	import	4,632,054.0	431	5,714,737.3	1,082,683.3	
3	2	local	7,533,212.0	2336	10,724,233.5	3,191,021.5	
4	3	import	6,646,992.3	419	8,104,182.1	1,457,189.8	
5	3	local	8,849,866.3	2291	12,523,302.9	3,673,436.6	
6	4	import	5,081,849.9	408	6,156,321.9	1,074,472.1	
7	4	local	8,284,842.8	2223	11,712,957.6	3,428,114.8	
8	5	import	5,351,139.1	422	6,508,957.2	1,157,818.1	
9	5	local	7,769,804.4	2340	10,938,885.4	3,169,081.0	
10	6	import	3,590,690.7	406	4,308,998.9	718,308.2	
11	6	local	6,993,421.5	2125	9,819,250.9	2,825,829.4	
12	7	import	6,112,988.5	401	7,378,786.3	1,265,797.9	
13	7	local	9,471,733.2	2201	13,289,508.6	3,817,775.4	
14	8	import	6,085,229.1	420	7,240,448.8	1,155,219.6	
15	8	local	8,742,470.5	2156	12,321,868.0	3,579,397.5	
16	9	import	6,863,876.4	447	8,242,090.0	1,378,213.6	
17	9	local	10,351,538.3	2426	14,500,553.7	4,149,015.5	
18	10	import	9,119,656.6	479	10,966,994.3	1,847,337.7	
19	10	local	12,254,674.1	2478	16,991,591.0	4,736,917.0	
20	11	import	5,932,485.8	479	7,073,143.4	1,140,657.5	
21	11	local	9,456,160.7	2496	13,067,074.4	3,610,913.8	
22	12	import	10,964,212.1	485	13,245,783.6	2,281,571.5	
23	12	local	16,618,514.7	2472	23,088,288.4	6,469,773.7	

In [215...]

```
df_category_purchased_data.pivot(index = 'month',
                                    columns = 'Category',
                                    values = 'Discount %').plot(kind =
plt.title('Local & Import Purchased per Month')
plt.xlabel('Month')
plt.ylabel('Discount %')
plt.xticks(rotation=45)
plt.legend(title='Category')
plt.tight_layout()
plt.show()
```



```
In [216]: df_category_purchased_data.pivot(index = 'month',
                                         columns = 'Category',
                                         values = 'total after discount tax')
plt.title('Local & Import Purchased per Month')
plt.xlabel('Month')
plt.ylabel('Total Purchased')
plt.xticks(rotation=45)
plt.legend(title='Category')
plt.tight_layout()
plt.show()
```



```
In [217]: df_t = df_purchased.groupby(['month', 'Category', 'Store Name']).agg(
    'Product name' : 'nunique',
    'Total' : 'sum',
    'total discount 2': 'sum'}).reset_index()
df_t['Discount %'] = df_t['total discount 2'] / df_t['Total'] * 100
df_t
```

month	Category	Store Name	total after discount tax 2	Product name	Total
-------	----------	------------	----------------------------	--------------	-------

0	1	Chemotherapy	وياك المعادي	7,020.0	1	7,800.0
1	1	Chemotherapy	وياك صيدلية الهرم	874,750.0	10	1,121,191.0
2	1	Toll	وياك صيدلية الهرم	7,000.0	1	8,404.0
3	1	import	وياك المعادي	1,863,828.9	309	2,268,210.6
4	1	import	وياك صيدلية الهرم	4,042,423.6	301	4,880,650.5
5	1	import_re_registration	وياك المعادي	39,361.1	4	44,548.0
6	1	import_re_registration	وياك صيدلية الهرم	199,530.5	8	225,785.0
7	1	local	وياك المعادي	3,034,505.5	2043	4,200,422.3
8	1	local	وياك صيدلية الهرم	4,252,921.5	1211	6,049,020.9
9	2	Chemotherapy	وياك صيدلية الهرم	697,750.0	11	902,437.0
10	2	import	وياك المعادي	1,721,185.1	350	2,084,199.6
11	2	import	وياك صيدلية الهرم	2,910,868.9	271	3,630,537.7
12	2	import_re_registration	وياك المعادي	143,461.8	5	158,652.0
13	2	import_re_registration	وياك صيدلية الهرم	115,983.2	5	126,675.0
14	2	local	وياك المعادي	3,114,604.3	2084	4,353,852.2
15	2	local	وياك صيدلية الهرم	4,418,607.8	1146	6,370,381.3
16	3	Chemotherapy	وياك صيدلية الهرم	573,500.0	9	711,924.0
17	3	Medical device	وياك المعادي	0.0	2	0.0

18	3	Medical device	وياك صيدلية الهرم	22,000.0	1	32,800.0
19	3	import	وياك المعادى	2,037,190.4	311	2,471,783.8
20	3	import	وياك صيدلية الهرم	4,609,801.9	295	5,632,398.3
21	3	import_re_registration	وياك المعادى	60,953.2	7	67,415.0
22	3	import_re_registration	وياك صيدلية الهرم	153,316.4	6	174,949.0
23	3	local	وياك المعادى	3,367,328.7	1982	4,715,582.1
24	3	local	وياك صيدلية الهرم	5,482,537.6	1278	7,807,720.8
25	4	Chemotherapy	وياك المعادى	14,040.0	1	15,600.0
26	4	Chemotherapy	وياك صيدلية الهرم	248,000.0	5	302,358.0
27	4	Medical device	وياك المعادى	163.8	2	164.0
28	4	import	وياك المعادى	1,506,689.6	269	1,829,474.3
29	4	import	وياك صيدلية الهرم	3,575,160.3	303	4,326,847.6
30	4	import_re_registration	وياك المعادى	62,869.2	6	70,213.0
31	4	import_re_registration	وياك صيدلية الهرم	145,691.6	5	164,895.5
32	4	local	وياك المعادى	2,773,569.9	1867	3,844,627.8
33	4	local	وياك صيدلية الهرم	5,511,272.9	1265	7,868,329.8
34	5	Chemotherapy	وياك المعادى	13,104.0	1	15,600.0
35	5	Chemotherapy	وياك صيدلية الهرم	856,500.0	9	1,075,660.0

36	5		Toll	وياك صيدلية الهرم	700.0	1	700.0
37	5	import		وياك المعادى	2,035,303.2	291	2,471,089.8
38	5	import		وياك صيدلية الهرم	3,315,835.9	301	4,037,867.5
39	5	import_re_registration		وياك المعادى	97,767.8	8	112,601.5
40	5	import_re_registration		وياك صيدلية الهرم	194,949.3	9	217,696.0
41	5		local	وياك المعادى	3,600,148.5	2072	4,972,264.5
42	5		local	وياك صيدلية الهرم	4,169,655.9	1331	5,966,620.9
43	6	Chemotherapy		وياك المعادى	13,260.0	1	15,600.0
44	6	Chemotherapy		وياك صيدلية الهرم	208,500.0	5	265,613.0
45	6		Toll	وياك المعادى	864.0	1	1,200.0
46	6		Toll	وياك صيدلية الهرم	30,379.3	14	39,111.0
47	6	import		وياك المعادى	1,352,505.5	313	1,617,043.9
48	6	import		وياك صيدلية الهرم	2,238,185.2	259	2,691,955.0
49	6	import_re_registration		وياك المعادى	92,696.8	5	105,951.5
50	6	import_re_registration		وياك صيدلية الهرم	269,976.4	8	304,551.0
51	6		local	وياك المعادى	2,845,277.9	1876	3,879,670.4
52	6		local	وياك صيدلية الهرم	4,148,143.6	1088	5,939,580.5
53	7	Chemotherapy		وياك صيدلية الهرم	587,500.0	8	728,635.0

54	7		Toll	وياك المعادي	2,930.0	4	3,744.0
55	7		Toll	وياك صيدلية الهرم	24,753.2	21	30,156.4
56	7		import	وياك المعادي	2,160,083.6	327	2,599,218.8
57	7		import	وياك صيدلية الهرم	3,952,904.9	297	4,779,567.5
58	7	import_re_registration		وياك المعادي	65,022.6	7	77,952.5
59	7	import_re_registration		وياك صيدلية الهرم	416,438.8	9	480,973.0
60	7		local	وياك المعادي	3,656,235.7	1936	4,976,271.1
61	7		local	وياك صيدلية الهرم	5,815,497.5	1369	8,313,237.5
62	8		Chemotherapy	وياك صيدلية الهرم	381,500.0	9	483,767.0
63	8		Toll	المخزن المركزي	4,828.2	9	6,483.5
64	8		Toll	وياك المعادي	11,488.0	11	13,741.5
65	8		Toll	وياك صيدلية الهرم	24,897.8	15	29,668.5
66	8		import	المخزن المركزي	3,099,386.6	198	3,702,589.1
67	8		import	وياك المعادي	1,232,380.7	285	1,472,879.4
68	8		import	وياك صيدلية الهرم	1,753,461.9	263	2,064,980.3
69	8	import_re_registration		المخزن المركزي	90,326.0	2	101,755.0
70	8	import_re_registration		وياك المعادي	148,919.6	8	164,092.0
71	8	import_re_registration		وياك صيدلية الهرم	142,406.1	7	160,169.0

72	8		local	المخزن المركزي	5,045,998.8	1070	7,254,111.4
73	8		local	وياك المعادى	1,923,094.4	1657	2,578,193.5
74	8		local	وياك صيدلية الهرم	1,773,377.3	1227	2,489,563.2
75	9	Chemotherapy		وياك صيدلية الهرم	517,500.0	12	628,531.0
76	9	Medical device		وياك المعادى	83.6	1	95.0
77	9	Medical device		وياك صيدلية الهرم	95.0	1	95.0
78	9		TABLETS	وياك صيدلية الهرم	206.4	1	258.0
79	9		Toll	المخزن المركزي	3,520.6	14	4,643.5
80	9		Toll	وياك المعادى	6,822.4	12	8,761.8
81	9		Toll	وياك صيدلية الهرم	19,657.5	13	24,023.0
82	9		import	المخزن المركزي	3,605,955.8	239	4,385,887.6
83	9		import	وياك المعادى	1,789,579.0	359	2,121,655.1
84	9		import	وياك صيدلية الهرم	1,468,341.6	266	1,734,547.3
85	9	import_re_registration		المخزن المركزي	38,375.3	6	45,662.0
86	9	import_re_registration		وياك المعادى	107,765.9	5	142,625.0
87	9	import_re_registration		وياك صيدلية الهرم	166,325.1	5	188,718.0
88	9		local	المخزن المركزي	6,065,336.6	1211	8,711,188.7
89	9		local	وياك المعادى	2,728,263.4	2038	3,681,268.8
90	9		local	وياك صيدلية	1,557,938.2	1267	2,108,096.2

الهرم							
91	9	medical supply	وياك المعادي	5,441.8	3	7,153.8	
92	9	medical supply	وياك صيدلية الهرم	6,360.8	7	7,513.5	
93	10	Chemotherapy	وياك المعادي	14,040.0	1	15,600.0	
94	10	Chemotherapy	وياك صيدلية الهرم	589,000.0	7	710,690.0	
95	10	TABLETS	المخزن المركزي	1,741.5	1	2,322.0	
96	10	Toll	المخزن المركزي	1,964.5	5	2,699.0	
97	10	Toll	وياك المعادي	3,489.4	8	4,569.0	
98	10	Toll	وياك صيدلية الهرم	11,826.6	17	14,118.2	
99	10	import	المخزن المركزي	3,364,517.3	163	4,097,800.6	
100	10	import	وياك المعادي	1,459,548.9	378	1,718,397.0	
101	10	import	وياك صيدلية الهرم	4,295,590.3	347	5,150,796.7	
102	10	import_re_registration	المخزن المركزي	195,325.8	6	226,571.0	
103	10	import_re_registration	وياك المعادي	213,762.9	9	266,443.0	
104	10	import_re_registration	وياك صيدلية الهرم	289,792.3	9	332,719.0	
105	10	local	المخزن المركزي	5,883,519.3	817	8,490,592.8	
106	10	local	وياك المعادي	2,493,428.5	2014	3,233,630.3	
107	10	local	وياك صيدلية الهرم	3,877,726.2	1760	5,267,367.9	
108	10	medical supply	وياك المعادي	701.5	4	875.0	
			وياك				

109	10	medical supply	صيدلية الهرم	3,738.5	1	4,850.0
110	10	under license Expired	وياك صيدلية الهرم	0.0	1	0.0
111	11	Chemotherapy	وياك المعادي	46,120.0	1	64,000.0
112	11	Chemotherapy	وياك صيدلية الهرم	560,550.0	11	658,228.0
113	11	Toll	المخزن المركزي	2,897.3	6	3,844.5
114	11	Toll	وياك المعادي	5,712.5	10	7,464.6
115	11	Toll	وياك صيدلية الهرم	18,681.7	19	22,574.2
116	11	import	المخزن المركزي	1,927,890.4	180	2,307,111.5
117	11	import	وياك المعادي	1,850,243.2	384	2,186,831.6
118	11	import	وياك صيدلية الهرم	2,154,352.2	338	2,579,200.3
119	11	import_re_registration	المخزن المركزي	28,521.5	8	33,864.0
120	11	import_re_registration	وياك المعادي	79,381.5	11	95,206.0
121	11	import_re_registration	وياك صيدلية الهرم	481,080.7	17	591,094.5
122	11	local	المخزن المركزي	3,902,415.1	968	5,672,052.6
123	11	local	وياك المعادي	3,149,550.5	2017	4,184,997.4
124	11	local	وياك صيدلية الهرم	2,404,195.1	1847	3,210,024.5
125	11	medical supply	وياك المعادي	2,750.7	6	3,836.0
126	11	medical supply	وياك صيدلية الهرم	3,920.0	7	4,998.0
127	11	under license Expired	وياك المعادي	616.0	1	800.0

128	11	under license Expired	وياك صيدلية الهرم	924.0	1	1,200.0
129	12	Chemotherapy	وياك المعادي	62,748.8	2	84,401.0
130	12	Chemotherapy	وياك صيدلية الهرم	976,781.3	9	1,163,929.0
131	12	Toll	المخزن المركزي	64,751.8	5	76,137.0
132	12	Toll	وياك المعادي	6,873.7	12	9,170.3
133	12	Toll	وياك صيدلية الهرم	16,735.3	14	21,088.5
134	12	import	over stock	1.3	7	0.0
135	12	import	المخزن المركزي	4,039,814.1	152	4,907,974.9
136	12	import	وياك المعادي	2,769,158.0	396	3,280,432.3
137	12	import	وياك صيدلية الهرم	4,155,238.6	381	5,057,376.4
138	12	import_re_registration	المخزن المركزي	86,923.9	4	102,304.0
139	12	import_re_registration	وياك المعادي	141,898.0	11	162,130.5
140	12	import_re_registration	وياك صيدلية الهرم	677,174.4	17	844,225.5
141	12	local	over stock	-0.0	6	0.0
142	12	local	المخزن المركزي	7,788,065.7	825	11,190,376.0
143	12	local	وياك المعادي	4,496,431.0	2093	5,997,888.2
144	12	local	وياك صيدلية الهرم	4,334,018.0	1880	5,900,024.2
145	12	medical supply	المخزن المركزي	0.0	1	0.0
146	12	medical supply	وياك المعادي	5,787.2	5	9,084.0

147	12	medical supply	وياك صيدلية الهرم	9,349.5	6	24,592.3
148	12	under license Expired	وياك المعادي	449.4	1	840.0
149	12	under license Expired	وياك صيدلية الهرم	308.0	1	400.0

In [218]:

```

df_t2 = df_purchased.groupby(['month', 'Store Name']).agg({'total af': 'sum',
                                                               'Product name' : 'nunique',
                                                               'Total' : 'sum',
                                                               'total discount 2': 'sum'}).reset_index()
df_t2['Discount %'] = df_t2['total discount 2'] / df_t2['Total'] * 100
df_t2

import arabic_reshaper
from bidi.algorithm import get_display

# Function to reshape Arabic text
def reshape_arabic(text):
    reshaped = arabic_reshaper.reshape(text)
    return get_display(reshaped)

# Pivot the table
pivot_df = df_t2.pivot(index='month', columns='Store Name', values='Discount %')

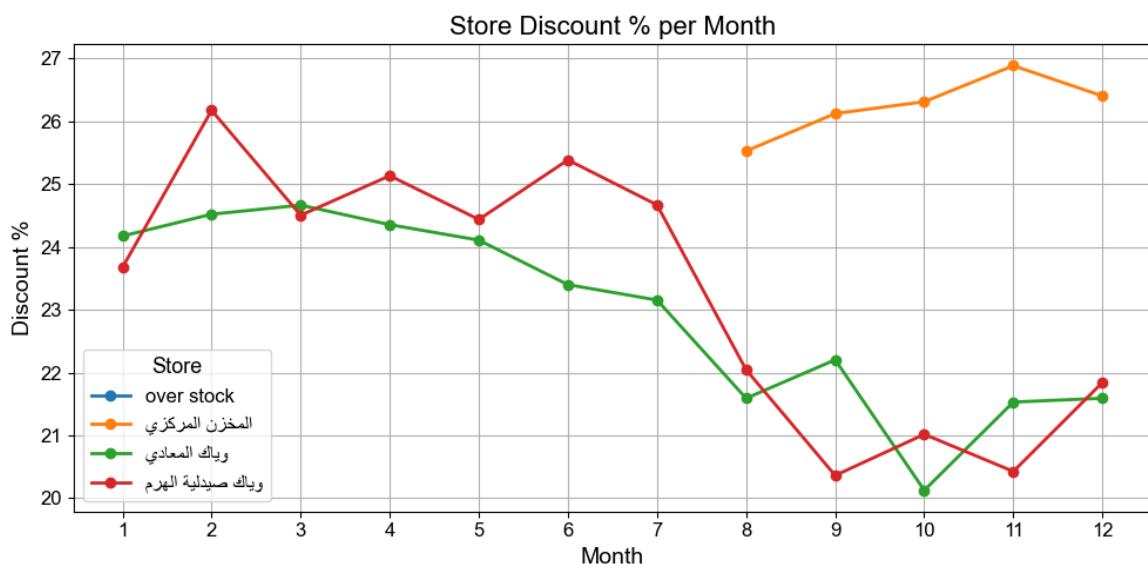
# Plot
ax = pivot_df.plot(kind='line', marker='o', figsize=(10, 5), linewidth=2)

# Reshape legend labels to Arabic
new_labels = [reshape_arabic(label) for label in pivot_df.columns]
ax.legend(title=reshape_arabic('Store'), labels=new_labels, fontsize=14)

plt.title('Store Discount % per Month', fontsize=16)
plt.xlabel('Month', fontsize=14)
plt.ylabel('Discount %', fontsize=14)
plt.xticks(ticks=sorted(df_t2['month'].unique()), fontsize=12)
plt.yticks(fontsize=12)
plt.grid(True)
plt.tight_layout()
plt.savefig('store discount.jpg', bbox_inches = 'tight', dpi = 200)

plt.show()

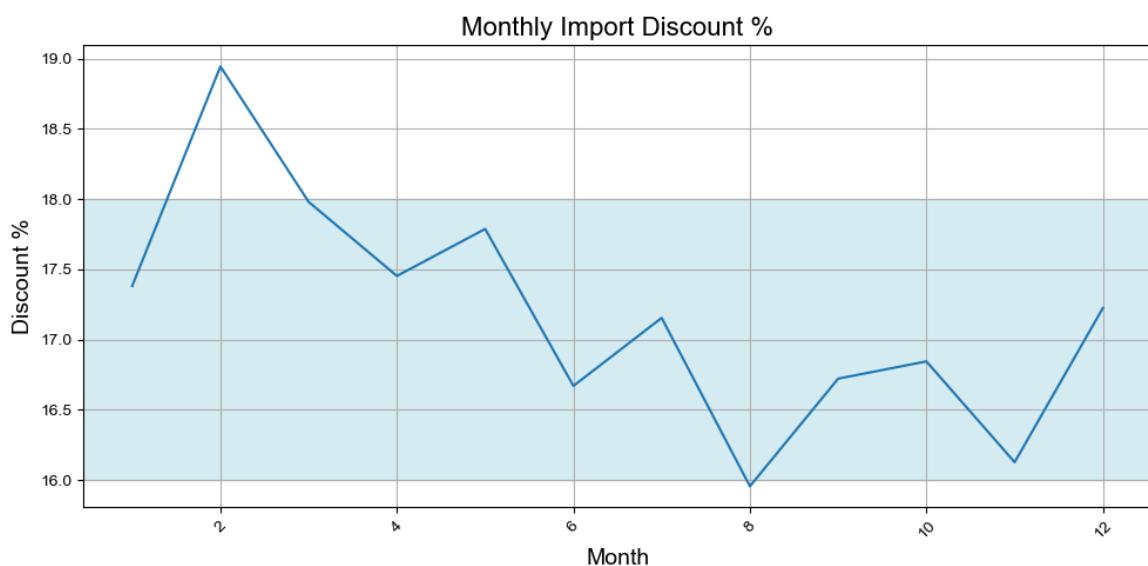
```



```
In [219]: df_category_purchased_data[df_category_purchased_data['Category'] != 'Over Stock'][['Category', 'Discount %']].plot(kind = 'line', title='Monthly Import Discount %', fontsize=16)
plt.xlabel('Month', fontsize=14)
plt.ylabel('Discount %', fontsize=14)
plt.xticks(rotation=45)
plt.axhspan(16, 18, color='lightblue', alpha=0.5, label='Q4')
# plt.legend(title='Category')
plt.grid()
plt.legend().set_visible(False) # Hide legend

plt.tight_layout()
plt.savefig('import discount.jpg', bbox_inches = 'tight', dpi = 200)

plt.show()
```



```
In [220]: df_category_purchased_data[df_category_purchased_data['Category'] != 'Over Stock'][['Category', 'Discount %']].plot(kind = 'line', title='Monthly Local Discount %', fontsize=16)
plt.xlabel('Month', fontsize=14)
plt.ylabel('Discount %', fontsize=14)
```

```

plt.xticks(rotation=45)
plt.legend(title='Category')
plt.ylim(27, 31)
plt.axhspan(27.8, 29.8, color='lightblue', alpha=0.5, label='Q4')
plt.legend().set_visible(False) # Hide legend

plt.grid()
plt.tight_layout()
plt.savefig('local discount.jpg', bbox_inches = 'tight', dpi = 200)

plt.show()

```



In []:

In [221...]: product_purchased_quantity_intersection

Out[221...]:

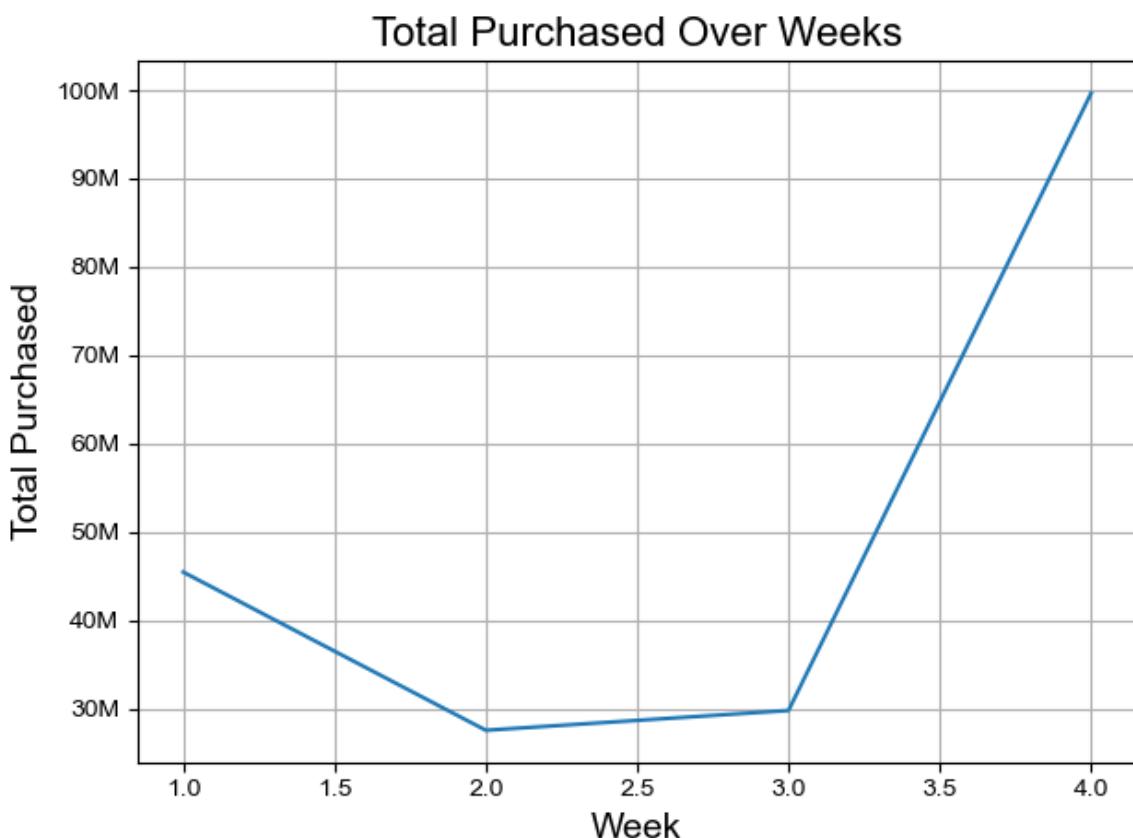
	Product name	Quantity	total after discount tax 2	Total	total discount 2	Disc
0	ZURCAL 40 MG 14 GASTRO RESISTANT TAB.	42,982.0	2,255,545.9	4,001,042.0	1,745,496.1	4
1	ASPIRIN PROTECT 100 MG 30 GASTRO- RESISTANT TAB.	36,370.3	1,644,969.1	2,109,683.0	464,713.8	2
2	PLAVIX 75MG 28 FILM COATED TAB.	22,436.0	4,776,690.6	5,962,951.0	1,186,260.4	1
3	VASTAREL MR 35MG 30 F.C.TAB.	21,335.0	1,978,816.4	2,881,823.0	903,006.6	1
4	DIAMICRON 60 MG 30 M.R. SCORED TAB.	19,823.0	1,228,034.5	1,648,404.0	420,369.5	1
5	MOVINERVE SUB LINGUAL 30 TABLETS.	19,037.0	1,152,235.3	1,770,463.0	618,227.7	1

6	DIBAVALLY PLUS 50/1000 MG 28 TABS.	16,899.0	1,667,992.0	2,743,993.0	1,076,001.0	:
7	CONTROLOC 40MG 14 GASTRORESISTANT TAB.	15,630.0	2,036,507.8	2,496,986.0	460,478.2	
8	MILGA ADVANCE 30 F.C. TAB	15,256.6	1,422,920.1	1,907,545.1	484,625.0	:
9	OMEGA-3 PLUS 30 CAPS	14,513.0	1,391,806.3	1,700,415.0	308,608.7	
10	PEPON PLUS 20 CAPS.	14,475.0	952,438.8	1,329,610.0	377,171.2	:
11	ESMORAP 40MG 14 CAPS.	14,221.0	888,359.3	1,551,736.0	663,376.7	:
12	ATOR 40MG 10 F.C. TAB.	13,950.0	831,280.8	1,152,088.0	320,807.2	
13	TAMSULIN 0.4MG 28 CAPS.	13,519.5	1,085,827.8	1,450,129.5	364,301.7	
14	Laprin 120 gm 10 sachets	12,044.0	1,486,553.4	2,130,030.0	643,476.6	:
15	JANAGLIP PLUS 50/1000MG 28 F.C. TAB.	10,344.0	859,002.5	1,303,344.0	444,341.5	
16	PROCORALAN 5MG 28 F.C. TABS.	10,223.2	1,992,002.4	2,718,532.0	726,529.6	:
17	TRESIBA 100 I.U./ML FLEXTOUCH PRE- FILLED PEN	10,214.0	4,043,688.9	4,794,089.0	750,400.1	
18	CRESTOR 20MG 28 F.C.TAB.	9,652.0	3,662,440.6	4,970,145.0	1,307,704.4	:
19	EXFORGE HCT 10/160/25MG 14 F.C. TAB.	9,458.0	1,473,003.9	1,756,313.0	283,309.1	
20	FORADIL 12 MCG 30 CAPS.+INHALER	9,106.0	1,448,938.1	1,937,816.0	488,877.9	:
21	EXFORGE HCT 5/160/12.5MG 14 F.C. TAB.	8,246.5	1,315,481.4	1,584,389.1	268,907.7	
22	CRESTOR 10MG28F.C. TAB.	8,225.5	1,863,352.8	2,524,056.0	660,703.2	:
23	GLIPTUS PLUS 50/1000MG 30 TABLETS	8,094.0	916,231.9	1,313,013.0	396,781.1	:

		EXFORGE					
24	5MG/160MG 14 F.C. TAB.		7,487.0	927,894.5	1,276,036.5	348,142.0	
		THIOTACID COMPOUND 600 MG 30 F.C.CAPLETS					
25			7,119.0	1,055,944.8	1,497,268.5	441,323.7	

In [222...]

```
sns.lineplot(data= week_df, x = 'week', y = 'total purchased')
plt.gca().yaxis.set_major_formatter(mticker.FuncFormatter(lambda x,
plt.grid()
plt.title('Total Purchased Over Weeks', fontsize=16)
plt.xlabel('Week', fontsize=14)
plt.ylabel('Total Purchased', fontsize=14)
plt.savefig('Weekly purchased.jpg', bbox_inches = 'tight', dpi = 200)
plt.tight_layout()
plt.show()
```



In [223...]

```
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
import arabic_reshaper
from bidi.algorithm import get_display

def reshape_arabic(text):
    return get_display(arabic_reshaper.reshape(text))

# Pivot and reshape Arabic
df_stores_month = df_purchased.pivot_table(
    index='month',
    columns='Store Name',
```

```
    values='total after discount tax 2',
    aggfunc='sum'
).round(0)

# Reshape the column (legend) names to Arabic
df_stores_month.columns = [reshape_arabic(col) for col in df_stores.

# Plot
ax = df_stores_month.plot(kind='line', linestyle='-', linewidth=2,

# Format y-axis with commas
ax.yaxis.set_major_formatter(ticker.FuncFormatter(lambda x, _: '{:,'

# Titles and labels
plt.title('Store Purchased Per Month', fontsize=16)
plt.xlabel('Month', fontsize=14)
plt.ylabel('Total Purchased', fontsize=14)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.legend(title=reshape_arabic('الفرع'), fontsize=12, title_fontsi
plt.tight_layout()
plt.savefig('store purchased.jpg', bbox_inches = 'tight', dpi = 200

plt.show()
```



In [247]: df[df['Product name'].str.contains('cotton', case=False, na=False)]

Out [247...]

	Date	Bill No	Store ID	Type	Store Name	Product id	Product name
61965	2024-05-13	11016771	4	Purchase	وياك صيدلية الهرم	23453	COTTON 100CM
71274	2024-06-01	80505	1	Purchase	وياك المعادى	23453	COTTON 100CM
132751	2024-10-01	59051	1	Purchase	وياك المعادى	23453	COTTON 100CM
138242	2024-10-09	8090671	1	Purchase	وياك المعادى	23453	COTTON 100CM
155822	2024-11-05	601001	1	Purchase	وياك المعادى	23453	COTTON 100CM
160624	2024-11-13	21585	1	Purchase	وياك المعادى	23453	COTTON 100CM
169320	2024-11-26	0000002667365	4	Purchase	وياك صيدلية الهرم	23453	COTTON 100CM
194497	2024-12-26	1234567	1	Purchase	وياك المعادى	23453	COTTON 100CM

In []: