TEAM 4

MARKET TENDENCES

DATA ANALYTICS BOOTCAMP - 1ST PROJECT

QUESTIONS THAT YOU FOUND INTERESTING AND WHAT MOTIVATED YOU TO ANSWER THEM

1.- Distinguish which has been the best year of a specific line of item, compared to the behavior and determine if the change has been set by volume or price.

2.- Which line of items has more sales on a single period?

3.- Which subgroup of an item has more sales?

4.- Which subgroup has increased and decreased?

5.- What is the best selling tendencies over the years?

WHERE AND HOW YOU FOUND THE DATA YOU USED TO ANSWER THESE QUESTIONS?

The database we used is a simulation of the operation of several years of a company dedicated to the commercialization of dried chillies, nuts and spices.

WE STARTED BY MERGING BOTH ITEMS AND TOTAL SALES PER ITEM.

THE DATA EXPLORATION AND CLEANUP PROCESS

```
items = "items.csv"
sales = "sales.csv"
# show = "Show.csv"

items_data = pd.read_csv(items, low_memory=False)
sales_data = pd.read_csv(sales, low_memory=False)
# show_data = pd.read_csv(show, low_memory=False)

combined_sales_items = pd.merge(items_data, sales_data, how="left", on="Item")
combined_sales_items.head()
```

	Item	Description	Dim	Dimension	GDim	Tipo	Unit	Period	Qtty	Sales	Store	TipoC	Customer
0	Item_1	SCRAP BAGS	1605	[M] Comp Empaques	Abarrotes	Venta	PZ	202002.0	-124.0	1488.0	StoreA	V Mostrador	Customer27435
1	Item_1	SCRAP BAGS	1605	[M] Comp Empaques	Abarrotes	Venta	PZ	202007.0	-19.0	228.0	StoreA	V Mostrador	Customer27435
2	Item_1	SCRAP BAGS	1605	[M] Comp Empaques	Abarrotes	Venta	PZ	202010.0	-43.0	516.0	StoreA	V Mostrador	Customer27435
3	Item_1	SCRAP BAGS	1605	[M] Comp Empaques	Abarrotes	Venta	PZ	202104.0	-88.0	1056.0	StoreA	V Mostrador	Customer27435
4	Item_1	SCRAP BAGS	1605	[M] Comp Empaques	Abarrotes	Venta	PZ	202107.0	-155.0	1860.0	StoreA	V Mostrador	Customer27435

THE DATA EXPLORATION AND CLEANUP PROCESS

WE SEPARATED 'YEAR' AND 'MONTH' AND ADDED THEM IN A NEW COLUM.

```
combined_sales_items['Period'] = combined_sales_items['Period'].astype(str)

# Crear las nuevas columnas 'Year' y 'Month'
combined_sales_items['Year'] = combined_sales_items['Period'].str[:4]
combined_sales_items['Month'] = combined_sales_items['Period'].str[4:]
combined_sales_items.head()
```

Dim	Dimension	GDim	Tipo	Unit	Period	Qtty	Sales	Store	TipoC	Customer	Location	Year	Month
1605	[M] Comp Empaques	Abarrotes	Venta	PZ	202002.0	-124.0	1488.0	StoreA	V Mostrador	Customer27435	D. F.	2020	02.0
1605	[M] Comp Empaques	Abarrotes	Venta	PZ	202007.0	-19.0	228.0	StoreA	V Mostrador	Customer27435	D. F.	2020	07.0
1605	[M] Comp Empaques	Abarrotes	Venta	PZ	202010.0	-43.0	516.0	StoreA	V Mostrador	Customer27435	D. F.	2020	10.0
1605	[M] Comp Empaques	Abarrotes	Venta	PZ	202104.0	-88.0	1056.0	StoreA	V Mostrador	Customer27435	D. F.	2021	04.0
1605	[M] Comp Empaques	Abarrotes	Venta	PZ	202107.0	-155.0	1860.0	StoreA	V Mostrador	Customer27435	D. F.	2021	07.0

THE DATA EXPLORATION AND CLEANUP PROCESS

WE DISPLAY EVERY SALE BY SUBGROUP

```
combined_sales_items['Sales'] = combined_sales_items['Sales'].astype(str).replace('-','')

combined_sales_items['Sales'] = combined_sales_items['Sales'] .astype(float).round(2)
combined_sales_items = combined_sales_items[combined_sales_items['Sales'] >= 1]
sales_by_year = combined_sales_items.set_index('GDim')
sales_by_year = combined_sales_items.groupby(['GDim'])
sales_by_year = combined_sales_items.groupby(['Year'])
sales_by_year = combined_sales_items.groupby(['Sales']).sum()
sales_by_year
```

Sales	Item	Description	Dim	Dimension	GDim	Tipo
1.00	Item_4Item_4 File display _1589Item_4055Item_4055I...	other productsother productsother productsSHOE...	32293	NoNoNo[M] UniformesNoNoNoNoNoNoNoNoNoNoPistach...	AbarrotesAbarrotesAbarrotesAbarrotesAbarrotesA...	NoNoNoMaterialesNoNoNoNoNoNoNoNoNoVentaMater...
1.04	Item_4055	OTHER PRODUCTS	1906	No	Abarrotes	No
1.06	Item_4055	OTHER PRODUCTS	1906	No	Abarrotes	No
1.08	Item_4055Item_4097	OTHER PRODUCTSSPECIAL NUGGET CLOVER B-25K	9916	NoPepita	AbarrotesSemillas	NoVenta
1.11	Item_4055	OTHER PRODUCTS	1906	No	Abarrotes	No
...	...	...	...	...	...	...
000.00	Item_599	BLACK CHILE WITH EXTRA LEG	2150	Mora	Chiles Secos	Venta
000.00	Item_2967	DATI-009NoHab	4005	Datiles	Frutas Secas	Venta
000.00	Item_4166	PINON WITH SHELL	4008	Pinon	Frutas Secas	Venta
036.48	Item_1867	ALMOND 25/27 CARMEL 22.68K	6010	Almendra	Nueces y Almendras	Venta
500.00	Item_3315	DEHYDRATED STRAWBERRY	4006	Deshidratadas	Frutas Secas	Venta

THE ANALYSIS PROCESS

WE SHOW THE TOTAL SALES BY YEAR AND SUBGROUP

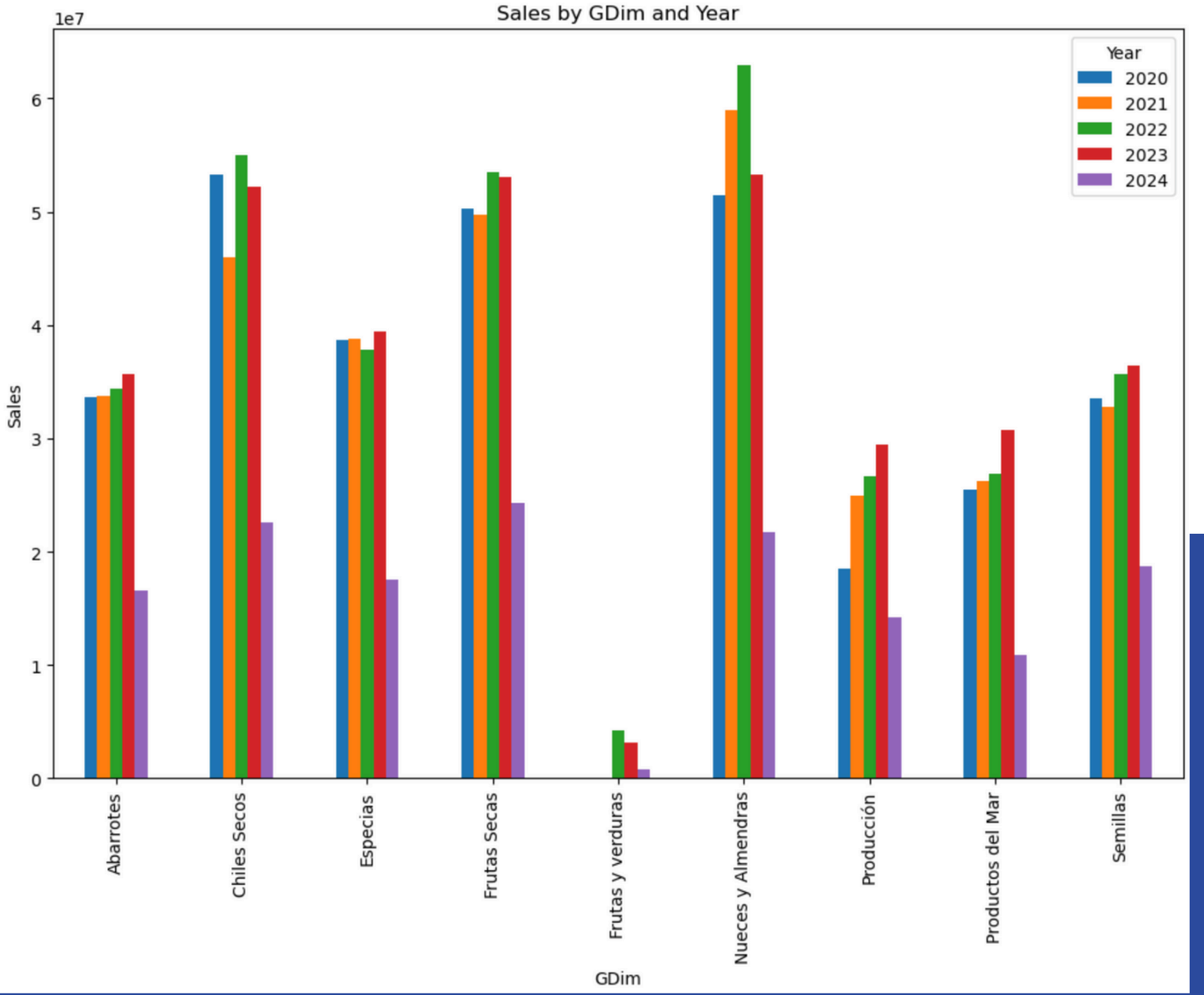
```
pivot_table = pd.pivot_table(combined_sales_items,
                             values='Sales',  # Cambia 'Sales' por el nombre correcto de la columna de ventas
                             index='GDim',
                             columns='Year',
                             aggfunc="sum",  # Función de agregación para sumar las ventas
                             fill_value=0)
pivot_table
```

Year	2020	2021	2022	2023	2024
GDim					
Abarrotes	33695197.68	33769333.19	34394161.06	35642362.08	16582324.75
Chiles Secos	53297977.06	45968490.88	55024702.33	52181460.60	22612518.11
Especias	38675176.36	38792506.90	37798643.38	39445310.22	17558787.92
Frutas Secas	50273260.41	49748280.77	53514533.13	53108275.45	24300819.79
Frutas y verduras	5016.00	2786.19	4267739.40	3098413.60	807411.35
Nueces y Almendras	51512533.53	58949645.43	62986511.23	53315978.87	21704818.51
Producción	18460717.78	24927845.50	26613171.01	29442607.36	14176397.47
Productos del Mar	25526852.47	26248697.29	26912530.08	30744922.29	10887484.19
Semillas	33483406.05	32809354.33	35675649.11	36427036.98	18770759.28

THE ANALYSIS PROCESS

```
pivot_table.plot(kind='bar', figsize=(12, 8))
plt.title('Sales by GDim and Year')
plt.xlabel('GDim')
plt.ylabel('Sales')
plt.legend(title='Year')
plt.show()
```

WE TRANSFORM THE PREVIOUS TABLE INTO A PIVOT TABLE TO VIEW AND ANALYZE THE INFORMATION



THE ANALYSIS PROCESS

WE EXPLORED THE SUBCATEGORY ‘FRUTAS SECAS’ AND DEPLOYED THE TOTAL SALES BY YEAR.

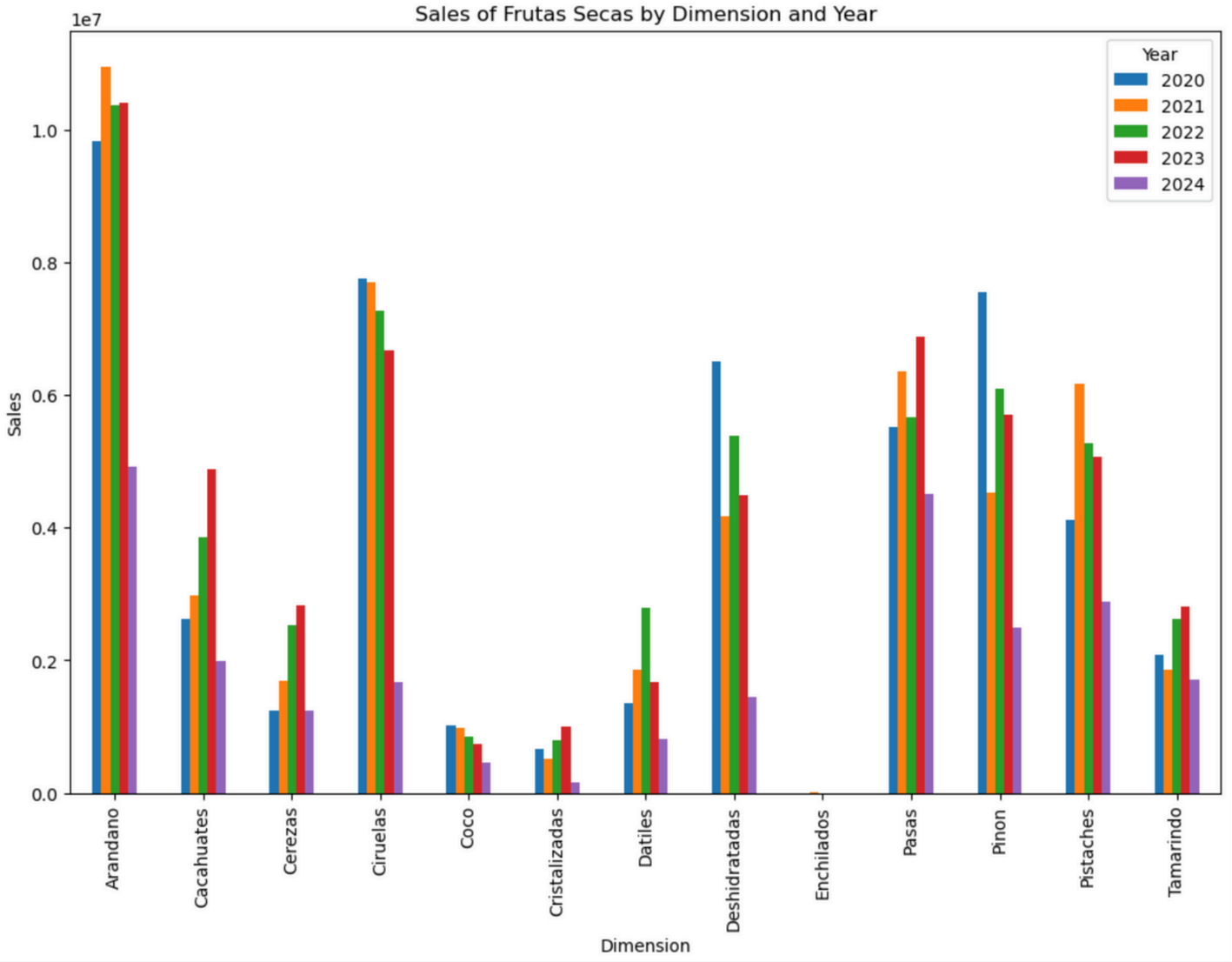
```
# Filtrar los datos para GDim 'Frutas Secas'
frutas_secas_data = combined_sales_items[combined_sales_items['GDim'] == 'Frutas Secas']

# Crear una tabla pivot para analizar las ventas por dimensión y año
pivot_table_frutas_secas = pd.pivot_table(frutas_secas_data,
                                          values='Sales',
                                          index='Dimension',
                                          columns='Year',
                                          aggfunc='sum',
                                          fill_value=0)

# Mostrar la tabla pivot
print(pivot_table_fr                    s)
```

File display

Year	2020	2021	2022	2023	2024
Dimension					
Arandano	9825992.93	10942273.74	10369402.03	10400294.51	4911393.86
Cacahuates	2625790.15	2972198.34	3853061.40	4873848.88	1982943.76
Cerezas	1252101.50	1691465.47	2534558.90	2821689.84	1249335.83
Ciruelas	7759886.41	7697337.13	7261619.24	6667975.69	1667741.27
Coco	1015934.67	979067.06	857163.19	734667.13	462547.14
Cristalizadas	663729.95	525170.69	795044.66	996348.47	161735.97
Datiles	1362713.08	1853284.91	2791040.23	1671112.36	819837.35
Deshidratadas	6499580.51	4164064.02	5385814.39	4498705.31	1453612.91
Enchilados	0.00	5039.10	1508.50	0.00	0.00
Pasas	5511986.42	6355758.70	5668935.63	6869456.55	4510561.97
Pinon	7553461.14	4534391.02	6099599.51	5705887.48	2488197.15
Pistaches	4110419.14	6161015.71	5264978.77	5066174.71	2886218.61
Tamarindo	2091664.51	1867214.88	2631806.68	2802114.52	1706693.97

THE ANALYSIS PROCESS

```
# Graficar la tabla pivot
pivot_table_frutas_secas.plot(kind='bar', figsize=(12, 8))
plt.title('Sales of Frutas Secas by Dimension and Year')
plt.xlabel('Dimension')
plt.ylabel('Sales')
plt.legend(title='Year')
plt.show()
```



WE TRANSFORM THE PREVIOUS TABLE INTO A PIVOT TABLE TO VIEW AND ANALYZE THE INFORMATION

THE ANALYSIS PROCESS

WE EXTRACTED ‘ARANDANOS’ FROM THE PREVIOUS TABLE AND DEPLOYED THE TOTAL SALES BY YEAR.

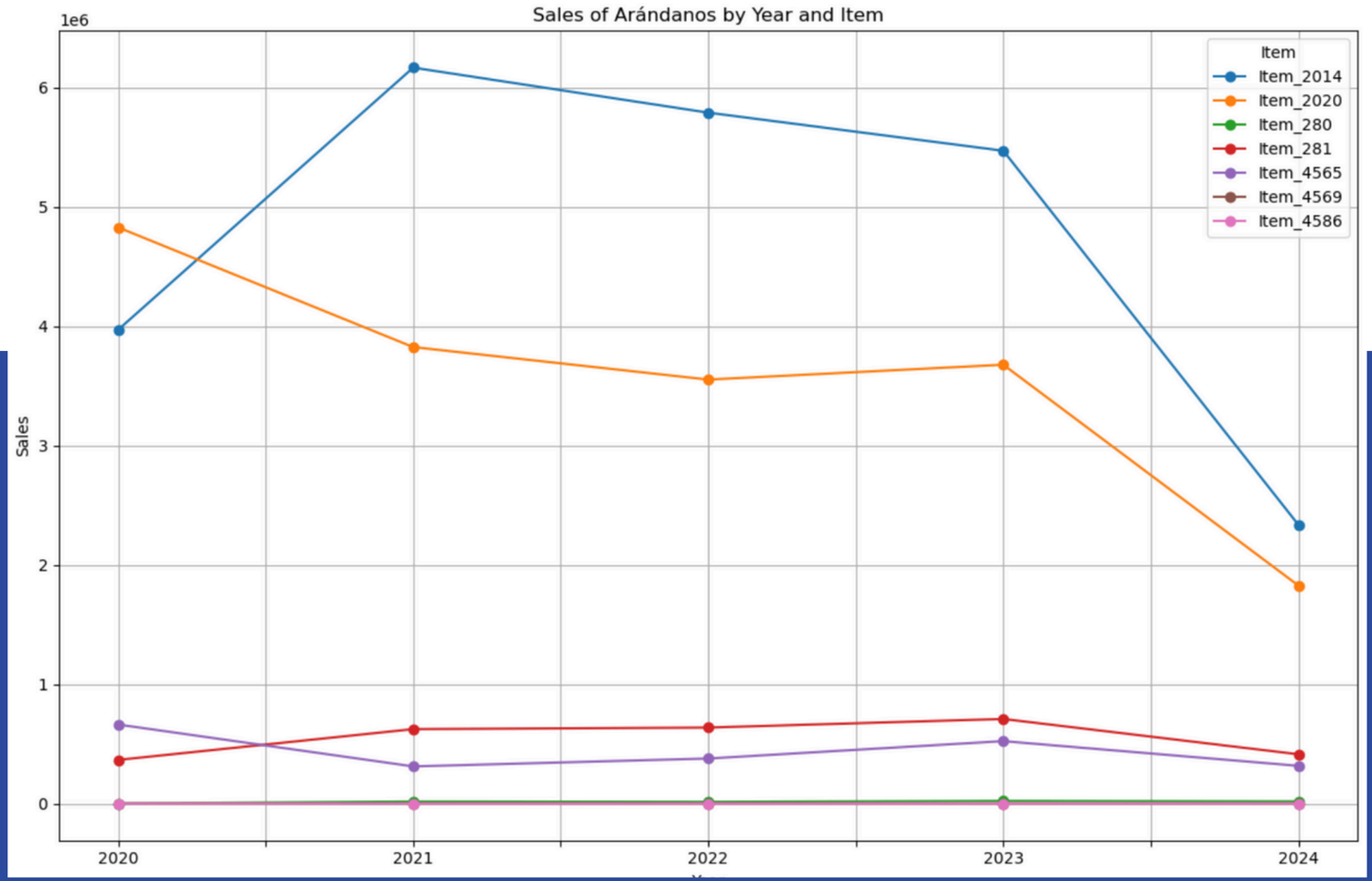
```python
# Filtrar los datos para la subcategoría 'Arándanos'
aran_____a = frutas_secas_data[frutas_secas_data['Dimension'] == 'Arandano']

# Crear una tabla pivot para analizar las ventas por ítem
pivot_table_arandanos = pd.pivot_table(arandanos_data,
                                       values='Sales',
                                       index='Item',
                                       columns='Year',
                                       aggfunc='sum',
                                       fill_value=0)

# Mostrar la tabla pivot
print(pivot_table_arandanos)
```

Year	2020	2021	2022	2023	2024
Item					
Item_2014	3969188.81	6165432.63	5788216.17	5469172.17	2336040.89
Item_2020	4827167.48	3824372.64	3552011.19	3676615.93	1827080.26
Item_280	856.00	16938.10	14222.85	22196.70	18320.80
Item_281	366056.13	623894.11	636988.10	708943.65	413289.77
Item_4565	662302.01	311581.26	377639.72	523366.06	316496.14
Item_4569	272.50	55.00	324.00	0.00	56.00
Item_4586	150.00	0.00	0.00	0.00	110.00

THE ANALYSIS PROCESS

```
pivot_table_transposed = pivot_table_arandanos.T
pivot_table_transposed.plot(kind='line', marker='o', figsize=(12, 8))

# Configurar el título y las etiquetas
plt.title('Sales of Arándanos by Year and Item')
plt.xlabel('Year')
plt.ylabel('Sales')
plt.legend(title='Item')
plt.grid(True)
plt.tight_layout()
# Mostrar la gráfica
plt.show()
```

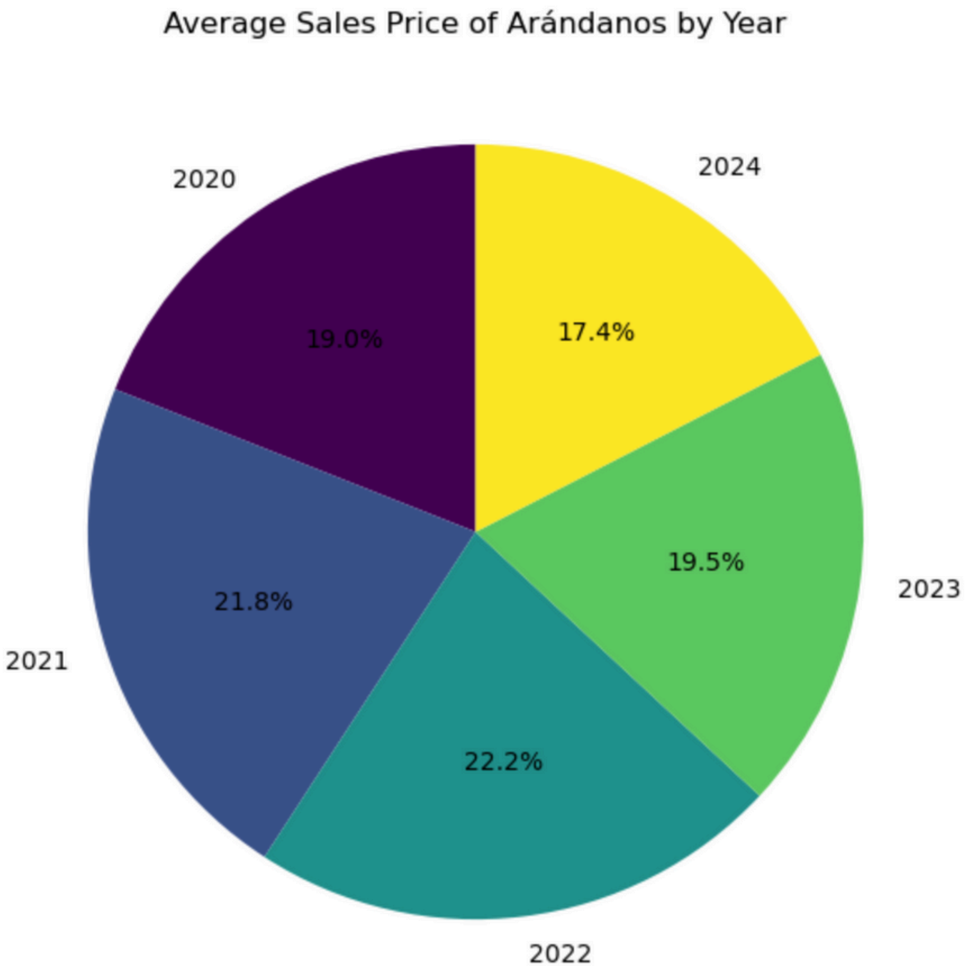
WE TRANSFORM THE PREVIOUS TABLE INTO A PIVOT TABLE TO VIEW AND ANALYZE THE INFORMATION



THE ANALYSIS PROCESS

```
average_price_per_year = arandanos_data.groupby('Year')['Sales'].mean()

# Graficar el precio promedio por año en una gráfica de pie
plt.figure(figsize=(10, 7))
average_price_per_year.plot(kind='pie', autopct='%1.1f%%', startangle=90, colormap='viridis')
plt.title('Average Sales Price of Arándanos by Year')
plt.ylabel('')  # Eliminar la etiqueta del eje y
plt.show()
```

WE CREATED A PIE CHART TO DEMONSTRATE AND APPRECIATE THE AVERAGE SALES PRICE OF 'ARANDANOS' BY YEAR



THE ANALYSIS PROCESS

IN HERE WE CAN SEE THE AVERAGE PRICE OF 'ARANDANOS' BY YEAR

```
item_2014_data = combined_sales_items[combined_sales_items['Item'] == 'Item_2014']

# Filtrar los datos para los años 2021 a 2023
item_2014_data = item_2014_data[item_2014_data['Year'].isin(['2021', '2022', '2023'])]

# Calcular el precio por ítem
item_2014_data['Price'] = item_2014_data['Sales'] / item_2014_data['Qtty']

# Agrupar por año y calcular el precio promedio si es necesario
average_price_per_year = item_2014_data.groupby('Year')['Price'].mean()

# Mostrar el resultado
print(average_price_per_year)
```

```
Year
2021    102.003051
2022    118.373790
2023    100.810450
```

THE ANALYSIS PROCESS

WE DECIDED TO ANALYZE THE AVERAGE PRICE OF 'ARANDANO' BY MONTH

```python
# Crear una columna 'Year-Month' para agrupar por mes y año
item_2014_data['Year-Month'] = item_2014_data['Year'] + '-' + item_2014_data['Month']

# Agrupar por 'Year-Month' y calcular el precio promedio
average_price_per_month = item_2014_data.groupby('Year-Month')['Price'].mean()

# Mostrar el resultado
print(average_price_per_month)
```

```
Year-Month
2021-01.0    101.131408
2021-02.0    100.480806
2021-03.0    101.974460
2021-04.0    101.939835
2021-05.0    100.972191
2021-06.0    100.956936
2021-07.0    100.431370
2021-08.0     99.358214
2021-09.0     97.936353
2021-10.0     98.097979
2021-11.0    102.423369
2021-12.0    114.596544
2022-01.0    114.239179
2022-02.0    114.305650
2022-03.0    118.772213
2022-04.0    119.138180
2022-05.0    119.208209
2022-06.0    119.526615
2022-07.0    118.930565
2022-08.0    119.009236
2022-09.0    118.571431
2022-10.0    117.891086
2022-11.0    118.691591
2022-12.0    120.046878
2023-01.0    118.623429
2023-02.0    115.494286
2023-03.0    115.114859
2023-04.0    109.177056
2023-05.0    107.781312
2023-06.0    101.710017
2023-07.0     96.746630
2023-08.0     93.302261
2023-09.0     89.793771
2023-10.0     94.302889
2023-11.0     90.191823
2023-12.0     88.604665
```

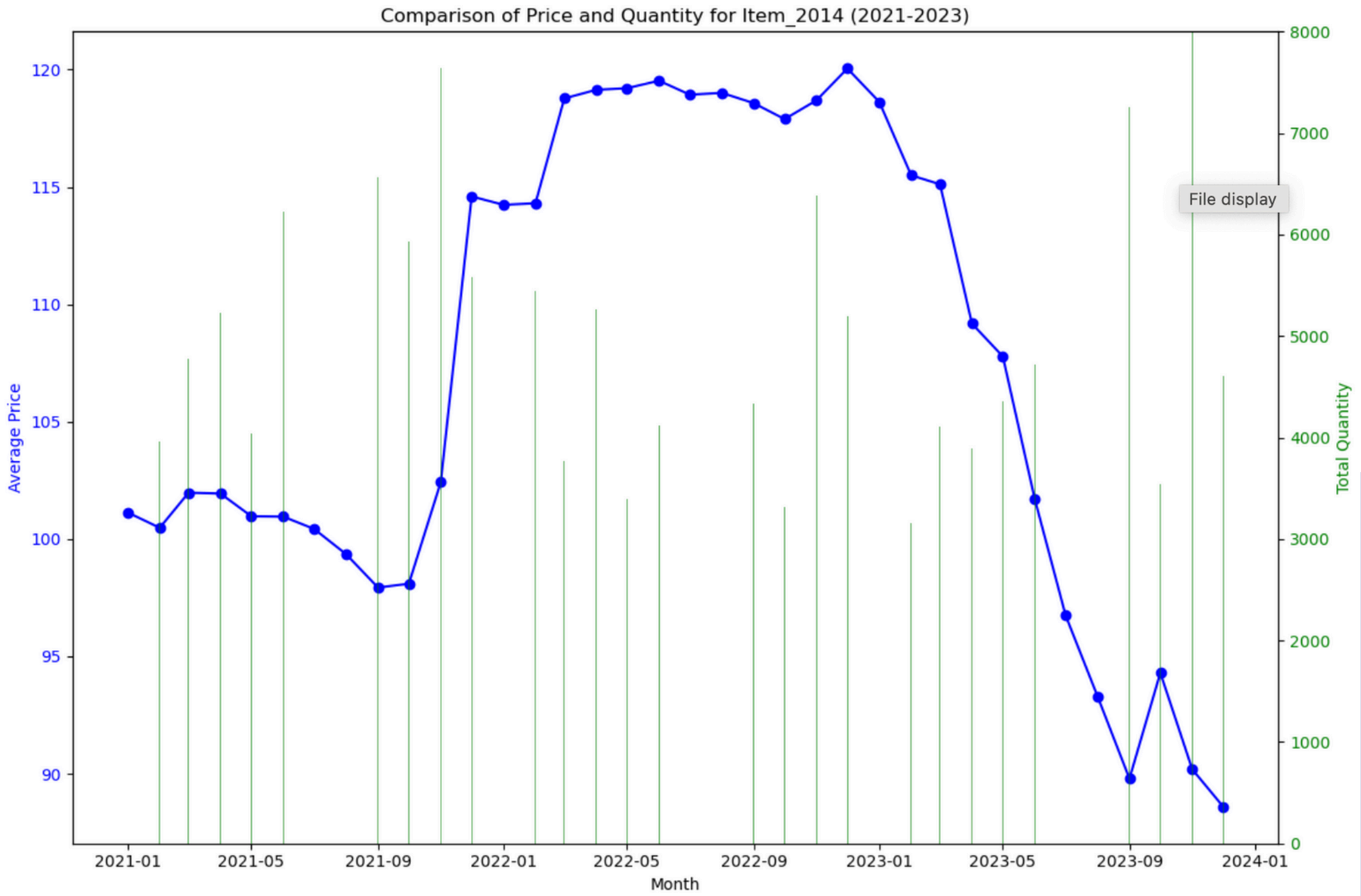
THE ANALYSIS PROCESS

WE COMPARED THE AVERAGE SALES PRICE VS THE SALES QUANTITY

```
# Convertir el índice a un formato de fecha para la gráfica
# Manejar posibles inconsistencias en los datos
grouped_data.index = pd.to_datetime(grouped_data.index, format='%Y-%m', errors='coerce')

# Eliminar filas con fechas inválidas
grouped_data = grouped_data.dropna()

# Crear la gráfica
fig, ax1 = plt.subplots(figsize=(12, 8))

# Graficar el precio en el primer eje y
ax1.plot(grouped_data.index, grouped_data['Price'], marker='o', color='b', label='Average Price')
ax1.set_xlabel('Month')
ax1.set_ylabel('Average Price', color='b')
ax1.tick_params(axis='y', labelcolor='b')

# Crear un segundo eje y para la cantidad
ax2 = ax1.twinx()
ax2.bar(grouped_data.index, grouped_data['Qtty'], color='g', alpha=0.6, label='Total Quantity')
ax2.set_ylabel('Total Quantity', color='g')
ax2.tick_params(axis='y', labelcolor='g')

# Limitar el eje Y derecho a 8,000
ax2.set_ylim(0, 8000)

# Títulos y etiquetas
plt.title('Comparison of Price and Quantity for Item_2014 (2021-2023)')
fig.tight_layout()

# Mostrar la gráfica
plt.show()
```

THE ANALYSIS PROCESS

WE COMPARED THE AVERAGE SALES PRICE VS THE SALES QUANTITY



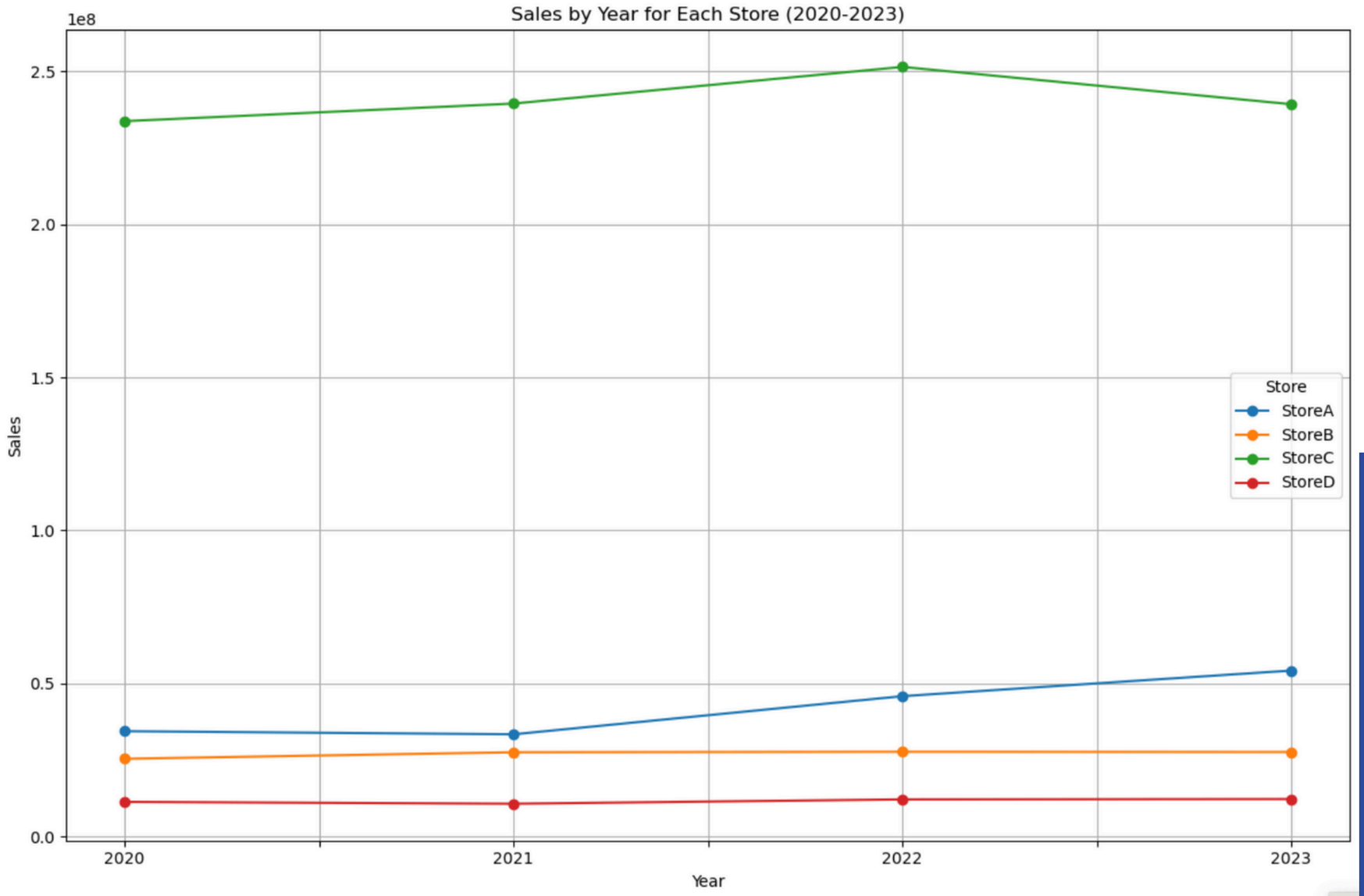
THE ANALYSIS PROCESS

WE ANALYZED THE TOTAL SALES BY STORE IN EVERY YEAR

```
# Filtrar los datos para los años 2020 a 2023
filtered_data = combined_sales_items[combined_sales_items['Year'].isin(['2020', '2021', '2022', '2023'])]

# Agrupar por 'Year' y 'Store' y sumar las ventas
sales_by_year_store = filtered_data.groupby(['Year', 'Store'])['Sales'].sum().unstack()

# Graficar las ventas por año para cada tienda
sales_by_year_store.plot(kind='line', marker='o', figsize=(12, 8))

# Configurar el título y las etiquetas
plt.title('Sales by Year for Each Store (2020-2023)')
plt.xlabel('Year')
plt.ylabel('Sales')
plt.legend(title='Store')
plt.grid(True)
plt.tight_layout()

# Mostrar la gráfica
plt.show()
```

THE ANALYSIS PROCESS

WE ANALYZED THE TOTAL SALES BY STORE IN EVERY YEAR



WHAT DO OUR FINDINGS MEAN?

We were asked to review the data to determine the behavior of sales between 2020 and 2024.

Based on this analysis we find that the company has 4 stores in which 9 large groups of products are handled.

Within the leading groups in sales, we analyzed dried fruits and within this group we saw the behavior of the blueberry, which is the main item.

Although the exchange rate had a gradual decrease from 21.5 annual average in 2020 to 17.7 average in 2023, the price of this item has increased, impacting the volume of consumption in kg.

WHAT DO OUR FINDINGS MEAN?

As a result of this lack of control in prices, it is noted that store C (with the highest historical volume) has decreased in its sales amount, store A has maintained constant growth and the other 2 stores have maintained their volume.

It is important to be attentive to changes in elements external to the market and how they affect us since some item prices moved depending on the availability of freight, for example.

THANK YOU FOR YOUR TIME!