Análisis de Ventas

Importar librerias necesarias

In [ ]:
```
import pandas as pd
import os
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

Tarea 1: Fusionar 12 meses de datos de ventas en un solo archivo CSV

In [ ]:
```
ruta = './Sales_Data/'

archivos = [archivo for archivo in os.listdir(ruta)]

datos_todos_meses = pd.DataFrame()

for archivo in archivos:
    df = pd.read_csv('./Sales_Data/'+archivo)
    datos_todos_meses = pd.concat([datos_todos_meses, df])

datos_todos_meses.to_csv('all_data.csv', index=False)
```

Leer el archivo fusionado en un DataFrame

In [ ]:
```
datos_todos = pd.read_csv('all_data.csv')
datos_todos.head()
```

Out[ ]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001
1	NaN	NaN	NaN	NaN	NaN	NaN
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215
3	176560	Google Phone	1	600	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001

Limpieza de datos

Eliminar las filas con valores `NaN`

In [ ]:
```
df_nan = datos_todos[datos_todos.isna().any(axis=1)]
df_nan.head()
```

Out[ ]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
1	NaN	NaN	NaN	NaN	NaN	NaN
356	NaN	NaN	NaN	NaN	NaN	NaN
735	NaN	NaN	NaN	NaN	NaN	NaN
1433	NaN	NaN	NaN	NaN	NaN	NaN
1553	NaN	NaN	NaN	NaN	NaN	NaN

In [ ]:
```
datos_todos = datos_todos.dropna(how='all')
datos_todos.head()
```

Out[ ]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215
3	176560	Google Phone	1	600	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001

Encontrar 'Or' y borrarlo

In [ ]:
```
datos_todos = datos_todos[datos_todos['Order Date'].str[0:2]!='Or']
datos_todos
```

Out[ ]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215
3	176560	Google Phone	1	600	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001
...	...	...	...	...	...	...
186845	259353	AAA Batteries (4-pack)	3	2.99	09/17/19 20:56	840 Highland St, Los Angeles, CA 90001
186846	259354	iPhone	1	700	09/01/19 16:00	216 Dogwood St, San Francisco, CA 94016
186847	259355	iPhone	1	700	09/23/19 07:39	220 12th St, San Francisco, CA 94016
186848	259356	34in Ultrawide Monitor	1	379.99	09/19/19 17:30	511 Forest St, San Francisco, CA 94016
186849	259357	USB-C Charging Cable	1	11.95	09/30/19 00:18	250 Meadow St, San Francisco, CA 94016

185950 rows × 6 columns

Convertir las columnas al tipo de dato correcto

In [ ]:
```
datos_todos['Quantity Ordered'] = pd.to_numeric(datos_todos['Quantity Ordered'])
datos_todos['Price Each'] = pd.to_numeric(datos_todos['Price Each'])
datos_todos.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 185950 entries, 0 to 186849
Data columns (total 6 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   Order ID          185950 non-null  object 
 1   Product           185950 non-null  object 
 2   Quantity Ordered  185950 non-null  int64  
 3   Price Each        185950 non-null  float64
 4   Order Date        185950 non-null  object 
 5   Purchase Address  185950 non-null  object 
dtypes: float64(1), int64(1), object(4)
memory usage: 9.9+ MB
```

Tarea 2: Agregar una columna de mes (`Month`)

In [ ]:

```
datos_todos['Month'] = datos_todos['Order Date'].str[0:2]
datos_todos['Month'] = datos_todos['Month'].astype('int32')
datos_todos.head()
```

Out[ ]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	4
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	4
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	4

Tarea 3: Agregar una columna de ventas (`Sales`)

In [ ]:

```
datos_todos['Sales'] = datos_todos['Quantity Ordered'] * datos_todos['Price Each']
datos_todos.head()
```

Out[ ]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	4	23.90
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	4	99.99
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	600.00
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	11.99
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	4	11.99

Tarea 4: Agregar una columna de ciudad (`City`)

In [ ]:

```
def obtener_ciudad(direccion):
    return direccion.split(',')[1]

def obtener_estado(direccion):
    return direccion.split(',')[2].split(' ')[1]


datos_todos['City'] = datos_todos['Purchase Address'].apply(lambda x: f'{obtener_ciudad(x)}
datos_todos.head()
```

Out[ ]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	4	23.90	Dallas (TX)
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	4	99.99	Boston (MA)
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	600.00	Los Angeles (CA)
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	11.99	Los Angeles (CA)
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	4	11.99	Los Angeles (CA)

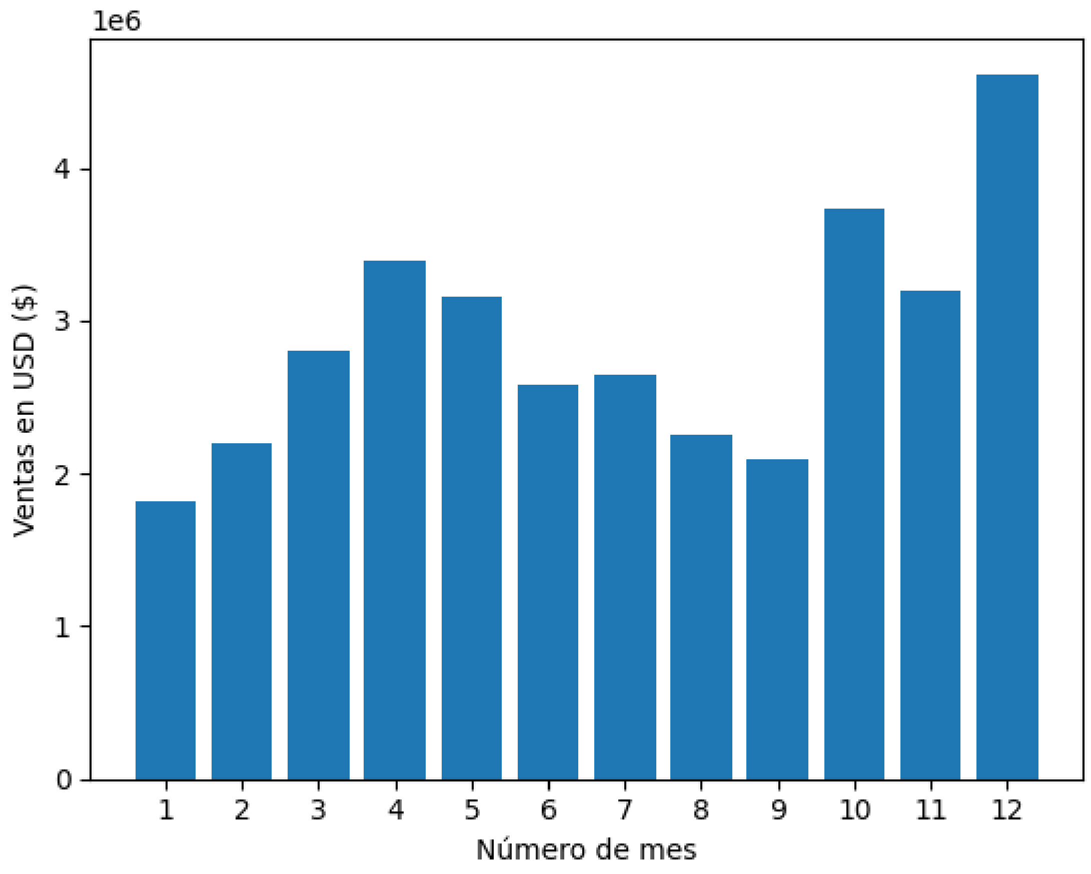
Pregunta 1: ¿Cuál fue el mejor mes para las ventas? ¿Cuánto se ganó ese mes?

In [ ]:
```
resultados = datos_todos.groupby('Month').sum()
resultados
```

Out[ ]:

	Quantity Ordered	Price Each	Sales
Month			
1	10903	1811768.38	1822256.73
2	13449	2188884.72	2202022.42
3	17005	2791207.83	2807100.38
4	20558	3367671.02	3390670.24
5	18667	3135125.13	3152606.75
6	15253	2562025.61	2577802.26
7	16072	2632539.56	2647775.76
8	13448	2230345.42	2244467.88
9	13109	2084992.09	2097560.13
10	22703	3715554.83	3736726.88
11	19798	3180600.68	3199603.20
12	28114	4588415.41	4613443.34

In [ ]:
```
meses = range(1,13)

plt.bar(meses, resultados['Sales'])
plt.xticks(meses)
plt.ylabel('Ventas en USD ($)')
plt.xlabel('Número de mes')
plt.show()
```



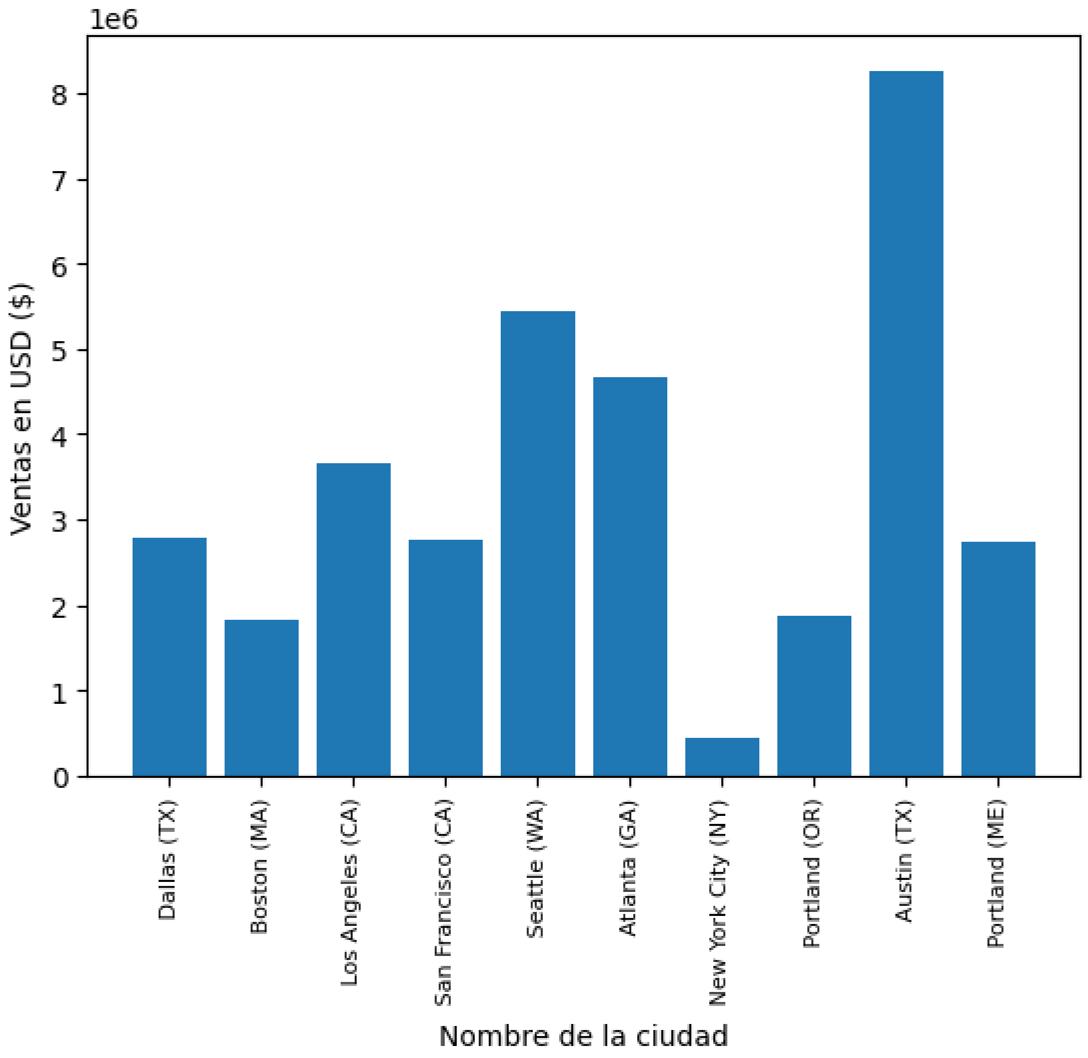
Pregunta 2: ¿Qué ciudad vendió más producto?

```
In [ ]: resultados = datos_todos.groupby('City').sum()
        resultados
```

Out[ ]:

City	Quantity Ordered	Price Each	Month	Sales
Atlanta (GA)	16602	2779908.20	104794	2795498.58
Austin (TX)	11153	1809873.61	69829	1819581.75
Boston (MA)	22528	3637409.77	141112	3661642.01
Dallas (TX)	16730	2752627.82	104620	2767975.40
Los Angeles (CA)	33289	5421435.23	208325	5452570.80
New York City (NY)	27932	4635370.83	175741	4664317.43
Portland (ME)	2750	447189.25	17144	449758.27
Portland (OR)	11303	1860558.22	70621	1870732.34
San Francisco (CA)	50239	8211461.74	315520	8262203.91
Seattle (WA)	16553	2733296.01	104941	2747755.48

```
In [ ]: ciudades = datos_todos['City'].unique()

        plt.bar(ciudades, resultados['Sales'])
        plt.xticks(ciudades, rotation='vertical', size=8)
        plt.ylabel('Ventas en USD ($)')
        plt.xlabel('Nombre de la ciudad')
        plt.show()
```



Pregunta 3: ¿A qué hora debemos mostrar anuncios para maximizar la probabilidad de que el cliente compre el producto?

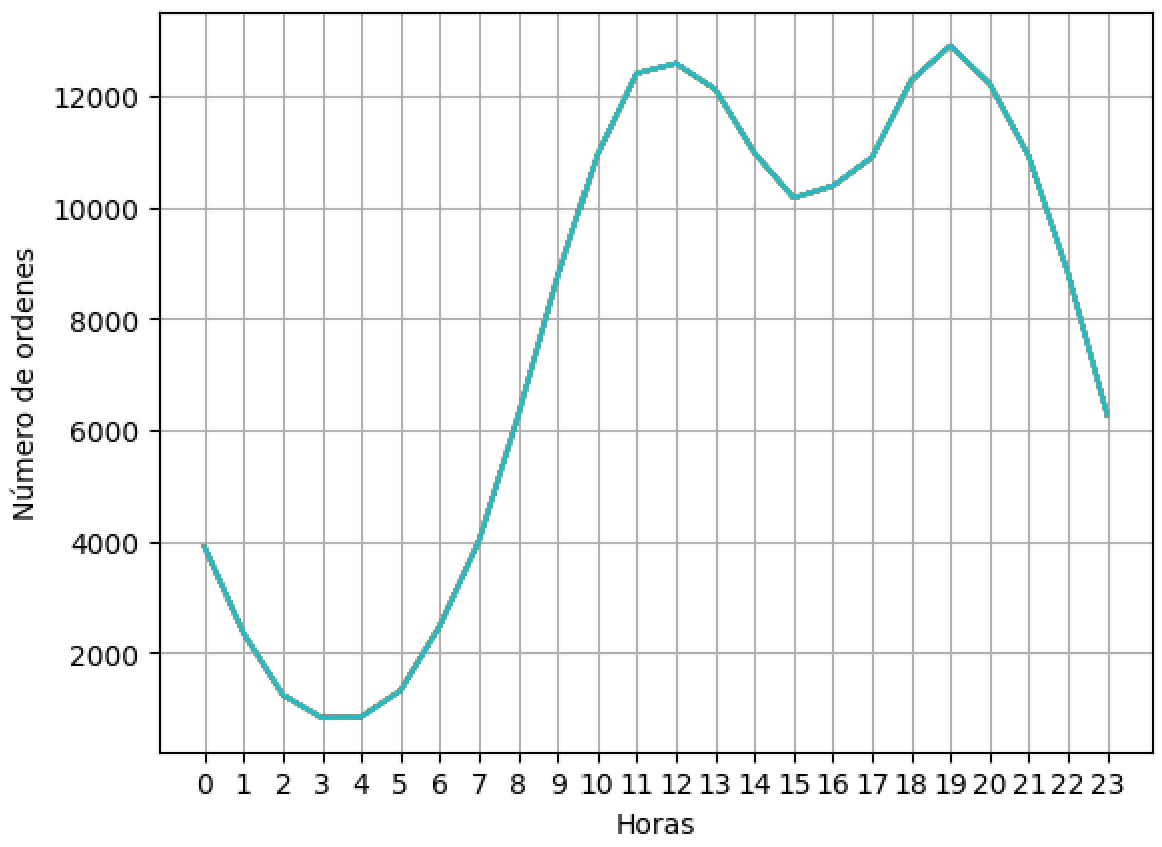
In [ ]:
```
datos_todos['Order Date'] = pd.to_datetime(datos_todos['Order Date'])

datos_todos['Hour'] = datos_todos['Order Date'].dt.hour
datos_todos['Minute'] = datos_todos['Order Date'].dt.minute
datos_todos.head()
```

Out[ ]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	Hour	Minute
0	176558	USB-C Charging Cable	2	11.95	2019-04-19 08:46:00	917 1st St, Dallas, TX 75001	4	23.90	Dallas (TX)	8	46
2	176559	Bose SoundSport Headphones	1	99.99	2019-04-07 22:30:00	682 Chestnut St, Boston, MA 02215	4	99.99	Boston (MA)	22	30
3	176560	Google Phone	1	600.00	2019-04-12 14:38:00	669 Spruce St, Los Angeles, CA 90001	4	600.00	Los Angeles (CA)	14	38
4	176560	Wired Headphones	1	11.99	2019-04-12 14:38:00	669 Spruce St, Los Angeles, CA 90001	4	11.99	Los Angeles (CA)	14	38
5	176561	Wired Headphones	1	11.99	2019-04-30 09:27:00	333 8th St, Los Angeles, CA 90001	4	11.99	Los Angeles (CA)	9	27

In [ ]:

```
horas = [hora for hora, df in datos_todos.groupby('Hour')]

plt.plot(horas, datos_todos.groupby(['Hour']).count())
plt.xticks(horas)
plt.ylabel('Número de ordenes')
plt.xlabel('Horas')
plt.grid()
plt.show()
```



Se recomienda alrededor de las 11 am (11) o las 7 pm (19)

Pregunta 4: ¿Qué productos se venden juntos con mayor frecuencia?

In [ ]:
```
df = datos_todos[datos_todos['Order ID'].duplicated(keep=False)]

df['Grouped'] = df.groupby('Order ID')['Product'].transform(lambda x: ', '.join(x))
df = df[['Order ID', 'Grouped']].drop_duplicates()
df.head()
```

Out[ ]:

	Order ID	Grouped
3	176560	Google Phone, Wired Headphones
18	176574	Google Phone, USB-C Charging Cable
30	176585	Bose SoundSport Headphones, Bose SoundSport He...
32	176586	AAA Batteries (4-pack), Google Phone
119	176672	Lightning Charging Cable, USB-C Charging Cable

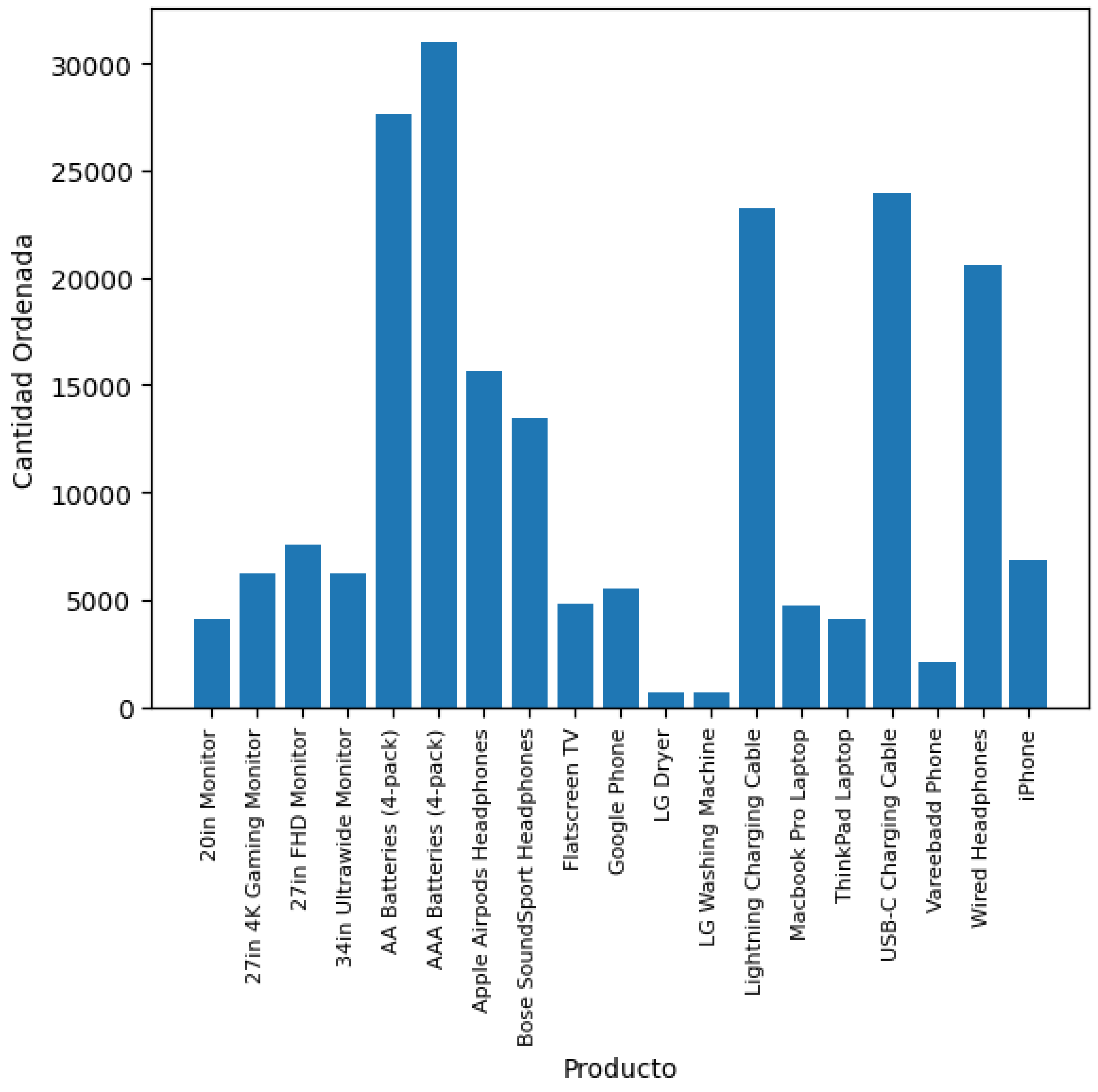
In [ ]:
```
from itertools import combinations
from collections import Counter

contar = Counter()

for fila in df['Grouped']:
    lista_fila = fila.split(', ')
    contar.update(Counter(combinations(lista_fila, 2)))
```

```python
for clave, valor in contar.most_common(10):
    print(clave, valor)
```

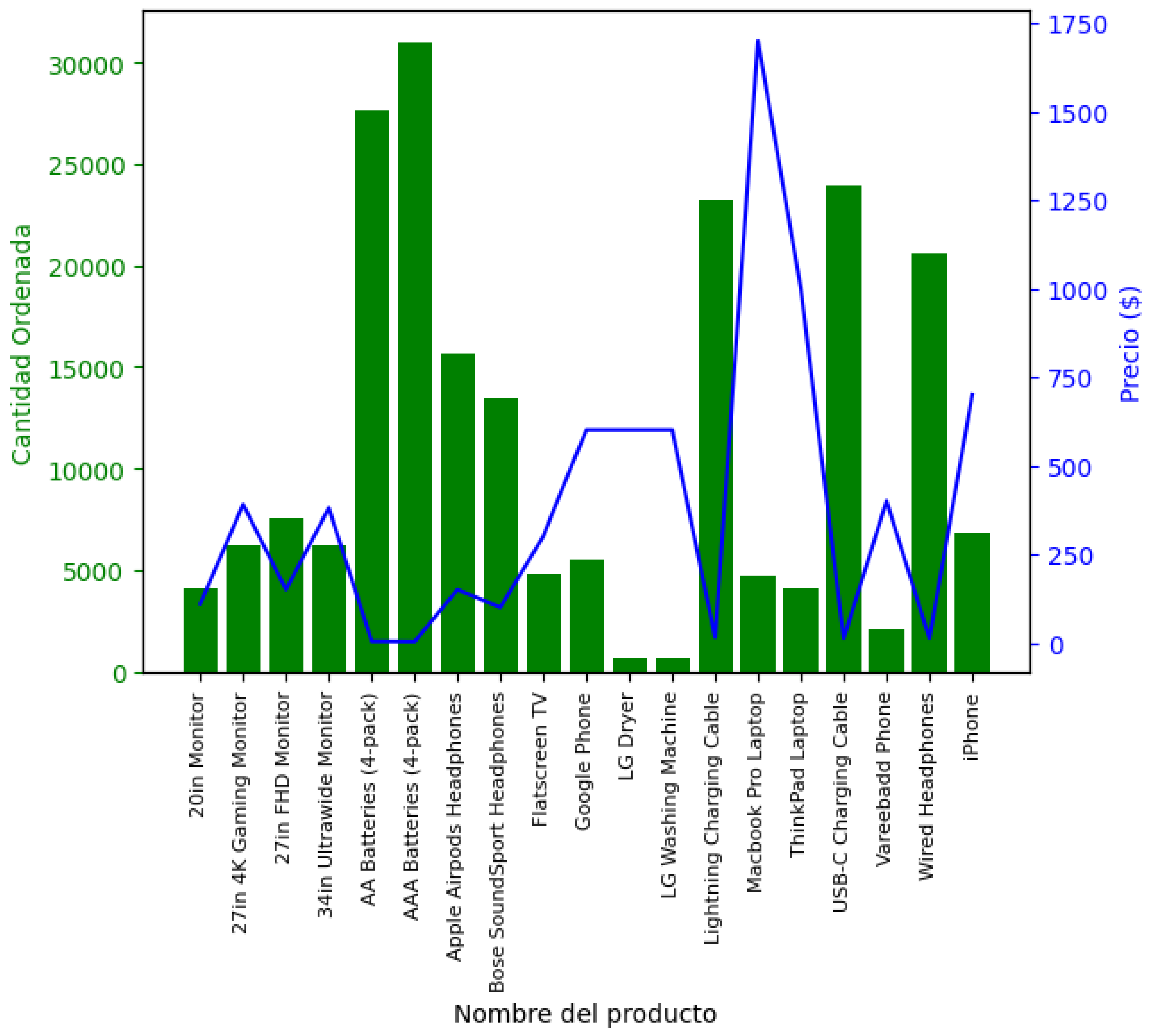
```
('iPhone', 'Lightning Charging Cable') 1005
('Google Phone', 'USB-C Charging Cable') 987
('iPhone', 'Wired Headphones') 447
('Google Phone', 'Wired Headphones') 414
('Vareebadd Phone', 'USB-C Charging Cable') 361
('iPhone', 'Apple Airpods Headphones') 360
('Google Phone', 'Bose SoundSport Headphones') 220
('USB-C Charging Cable', 'Wired Headphones') 160
('Vareebadd Phone', 'Wired Headphones') 143
('Lightning Charging Cable', 'Wired Headphones') 92
```

Pregunta 5: ¿Qué producto vendió más? ¿Por qué crees que vendió más?

In [ ]:
```python
grupo_productos =  datos_todos.groupby('Product')
cantidad_ordenada = grupo_productos.sum()['Quantity Ordered']

productos = [producto for producto, df in grupo_productos]

plt.bar(productos, cantidad_ordenada)
plt.ylabel('Cantidad Ordenada')
plt.xlabel('Producto')
plt.xticks(productos, rotation='vertical', size=8)
plt.show()
```



In [ ]:
```python
precios = datos_todos.groupby('Product').mean()['Price Each']

fig, ax1 = plt.subplots()

ax2 = ax1.twinx()
ax1.bar(productos, cantidad_ordenada, color='g')
ax2.plot(productos, precios, 'b-')

ax1.set_xlabel('Nombre del producto')
ax1.set_ylabel('Cantidad Ordenada', color='g')
ax2.set_ylabel('Precio ($)', color='b')
ax1.set_xticklabels(productos, rotation='vertical', size=8)

# Establecer el color de las etiquetas del eje y
ax1.tick_params(axis='y', colors='g')
ax2.tick_params(axis='y', colors='b')

plt.show()
```


Cantidad Ordenada
0
5000
10000
15000
20000
25000
30000
Precio ($)
0
250
500
750
1000
1250
1500
1750
20in Monitor
27in 4K Gaming Monitor
27in FHD Monitor
34in Ultrawide Monitor
AA Batteries (4-pack)
AAA Batteries (4-pack)
Apple Airpods Headphones
Bose SoundSport Headphones
Flatscreen TV
Google Phone
LG Dryer
LG Washing Machine
Lightning Charging Cable
Macbook Pro Laptop
ThinkPad Laptop
USB-C Charging Cable
Vareebadd Phone
Wired Headphones
iPhone
Nombre del producto