Análisis de Datos - Ventas

Keith Galli - Solving real world data science tasks with Python Pandas!

In [18]:
```
import pandas as pd
import os
import matplotlib.pyplot as plt
```

Extraer data

Unir data de los 12 meses de ventas, en un solo archivo .CSV

In [19]:
```
#df = pd.read_csv("ventas_mensuales/Sales_January_2019.csv")
#df.head(10)

# Crear lista de archivos
files = [file for file in os.listdir("ventas_mensuales")]

#Verificar Listado de Archivos
#for file in files:
    #print(file)

all_months_data = pd.DataFrame()

for file in files:
    df = pd.read_csv("ventas_mensuales/"+file)
    all_months_data = pd.concat([all_months_data, df])

#Verificar
all_months_data.head(10)

#Exportar a archivo excel
all_months_data.to_csv("dataanual.csv", index = False)
```

Leer datos desde archivo consolidado .CSV

In [69]:
```
all_data=pd.read_csv("dataanual.csv")
all_data.head(20)
```

Out[69]:

	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
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001
6	176562	USB-C Charging Cable	1	11.95	04/29/19 13:03	381 Wilson St, San Francisco, CA 94016
7	176563	Bose SoundSport Headphones	1	99.99	04/02/19 07:46	668 Center St, Seattle, WA 98101
8	176564	USB-C Charging Cable	1	11.95	04/12/19 10:58	790 Ridge St, Atlanta, GA 30301
9	176565	Macbook Pro Laptop	1	1700	04/24/19 10:38	915 Willow St, San Francisco, CA 94016
10	176566	Wired Headphones	1	11.99	04/08/19 14:05	83 7th St, Boston, MA 02215
11	176567	Google Phone	1	600	04/18/19 17:18	444 7th St, Los Angeles, CA 90001
12	176568	Lightning Charging Cable	1	14.95	04/15/19 12:18	438 Elm St, Seattle, WA 98101
13	176569	27in 4K Gaming Monitor	1	389.99	04/16/19 19:23	657 Hill St, Dallas, TX 75001
14	176570	AA Batteries (4-pack)	1	3.84	04/22/19 15:09	186 12th St, Dallas, TX 75001
15	176571	Lightning Charging Cable	1	14.95	04/19/19 14:29	253 Johnson St, Atlanta, GA 30301
16	176572	Apple Airpods Headphones	1	150	04/04/19 20:30	149 Dogwood St, New York City, NY 10001
17	176573	USB-C Charging Cable	1	11.95	04/27/19 18:41	214 Chestnut St, San Francisco, CA 94016
18	176574	Google Phone	1	600	04/03/19 19:42	20 Hill St, Los Angeles, CA 90001
19	176574	USB-C Charging Cable	1	11.95	04/03/19 19:42	20 Hill St, Los Angeles, CA 90001

Limpieza de Datos

Visualizar filas con valores NaN

In [21]:
```
nan_df = all_data[all_data.isna().any(axis=1)]
nan_df.head(15)
```

Out[21]:

	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
1571	NaN	NaN	NaN	NaN	NaN	NaN
1992	NaN	NaN	NaN	NaN	NaN	NaN
2265	NaN	NaN	NaN	NaN	NaN	NaN
2798	NaN	NaN	NaN	NaN	NaN	NaN
3024	NaN	NaN	NaN	NaN	NaN	NaN
3098	NaN	NaN	NaN	NaN	NaN	NaN
4279	NaN	NaN	NaN	NaN	NaN	NaN
4562	NaN	NaN	NaN	NaN	NaN	NaN
4958	NaN	NaN	NaN	NaN	NaN	NaN
5565	NaN	NaN	NaN	NaN	NaN	NaN

Eliminar filas con valores NaN

In [22]:
```
# Eliminar filas que al menos presentan un NaN
#all_data = all_data.dropna(how = "any")
all_data = all_data.dropna(how = "all")
```

In [23]:
```
#Verificacion
nan_df = all_data[all_data.isna().any(axis=1)]
nan_df.head(15)
```

Out[23]:

Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address

Visualizar filas con valores OR

In [24]:
```
temp_df = all_data[all_data["Order Date"].str[0:2] == "Or"]
temp_df.head(15)
```

Out[24]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
519	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
1149	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
1155	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
2878	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
2893	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
3036	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
3209	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
3618	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
4138	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
4645	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
4794	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
5303	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
6939	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
7497	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
8635	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address

Eliminar filas con valores OR

In [25]:
```
all_data = all_data[all_data["Order Date"].str[0:2] != "Or"]
all_data.head(15)
```

Out[25]:

	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
6	176562	USB-C Charging Cable	1	11.95	04/29/19 13:03	381 Wilson St, San Francisco, CA 94016
7	176563	Bose SoundSport Headphones	1	99.99	04/02/19 07:46	668 Center St, Seattle, WA 98101
8	176564	USB-C Charging Cable	1	11.95	04/12/19 10:58	790 Ridge St, Atlanta, GA 30301
9	176565	Macbook Pro Laptop	1	1700	04/24/19 10:38	915 Willow St, San Francisco, CA 94016
10	176566	Wired Headphones	1	11.99	04/08/19 14:05	83 7th St, Boston, MA 02215
11	176567	Google Phone	1	600	04/18/19 17:18	444 7th St, Los Angeles, CA 90001

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
12	176568	Lightning Charging Cable	1	14.95	04/15/19 12:18	438 Elm St, Seattle, WA 98101
13	176569	27in 4K Gaming Monitor	1	389.99	04/16/19 19:23	657 Hill St, Dallas, TX 75001
14	176570	AA Batteries (4-pack)	1	3.84	04/22/19 15:09	186 12th St, Dallas, TX 75001
15	176571	Lightning Charging Cable	1	14.95	04/19/19 14:29	253 Johnson St, Atlanta, GA 30301

Convertir formatos de columnas Quantity Ordered and Price Each a valores numéricos

In [26]:
```
all_data["Quantity Ordered"] = pd.to_numeric(all_data["Quantity Ordered"])
all_data["Price Each"] = pd.to_numeric(all_data["Price Each"])
```

Manipulación de datos

Agregar columnas adicionales - Month

In [27]:
```
all_data["Month"] = all_data["Order Date"].str[0:2]
all_data.head(10)
all_data["Month"] = all_data["Month"].astype("int32")
all_data.head(10)
```

Out[27]:

	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
6	176562	USB-C Charging Cable	1	11.95	04/29/19 13:03	381 Wilson St, San Francisco, CA 94016	4
7	176563	Bose SoundSport Headphones	1	99.99	04/02/19 07:46	668 Center St, Seattle, WA 98101	4
8	176564	USB-C Charging Cable	1	11.95	04/12/19 10:58	790 Ridge St, Atlanta, GA 30301	4
9	176565	Macbook Pro Laptop	1	1700.00	04/24/19 10:38	915 Willow St, San Francisco, CA 94016	4

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month
10	176566	Wired Headphones	1	11.99	04/08/19 14:05	83 7th St, Boston, MA 02215	4

Agregar columna - Sales

In [28]:
```
all_data["Sales"] = all_data["Quantity Ordered"] * all_data["Price Each"]
all_data.head()
```

Out[28]:

	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

Agregar columna - City

In [29]:
```
#Uso del metodo apply
#Separar valores de columna Purchase Address y tomar segundo elemento
#all_data["Column"] = all_data["Purchase Address"].apply(lambda x: x.split(",")[1])
#all_data.head()

def get_city(address):
    return address.split(",")[1]
all_data["City"] = all_data["Purchase Address"].apply(lambda x: get_city(x))
#Eliminar Columna Column
#all_data = all_data.drop(columns = "Column")
all_data.head()
```

Out[29]:

	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	4	23.90	Dallas

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City
						75001			
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	4	99.99	Boston
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	600.00	Los Angeles
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	11.99	Los Angeles
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	4	11.99	Los Angeles

Agregar columna - State

In [30]:

```
def get_state(address):
    return address.split(",")[2].split(" ")[1]
all_data["State"] = all_data["Purchase Address"].apply(lambda x:
get_state(x))
#Eliminar Columna Column
all_data.head()
```

Out[30]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	State
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 N° 1: ¿Cuál fue el mejor mes de Ventas? ¿Cuanto fue lo que se ganó?

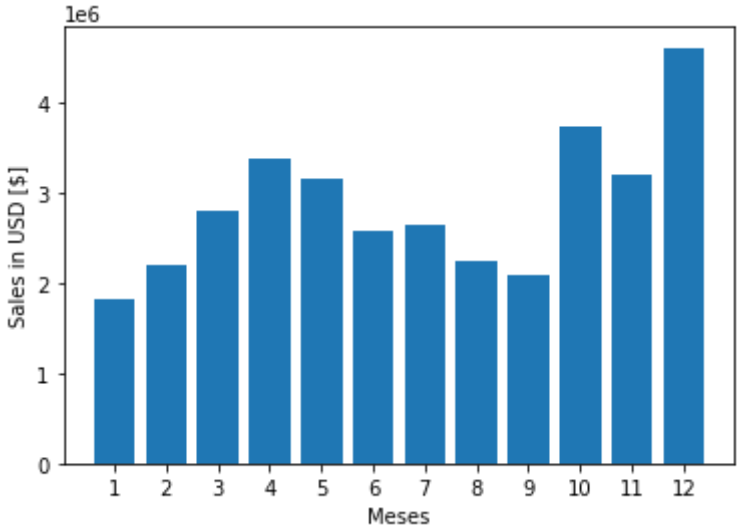
In [31]:
```
all_data.groupby("Month").sum()
```

Out[31]:

Month	Quantity Ordered	Price Each	Sales
1	10903	1.811768e+06	1.822257e+06
2	13449	2.188885e+06	2.202022e+06
3	17005	2.791208e+06	2.807100e+06
4	20558	3.367671e+06	3.390670e+06
5	18667	3.135125e+06	3.152607e+06
6	15253	2.562026e+06	2.577802e+06
7	16072	2.632540e+06	2.647776e+06
8	13448	2.230345e+06	2.244468e+06
9	13109	2.084992e+06	2.097560e+06
10	22703	3.715555e+06	3.736727e+06
11	19798	3.180601e+06	3.199603e+06
12	28114	4.588415e+06	4.613443e+06

In [32]:
```
months = range(1,13)
results = all_data.groupby("Month").sum()

plt.bar(months, results["Sales"])
plt.xticks(months)
plt.ylabel("Sales in USD [$]")
plt.xlabel("Meses")
plt.show()
```



Pregunta N° 2: ¿Qué ciudad tiene el mayor número de ventas?

In [33]:
```
results = all_data.groupby("City").sum()
results
```

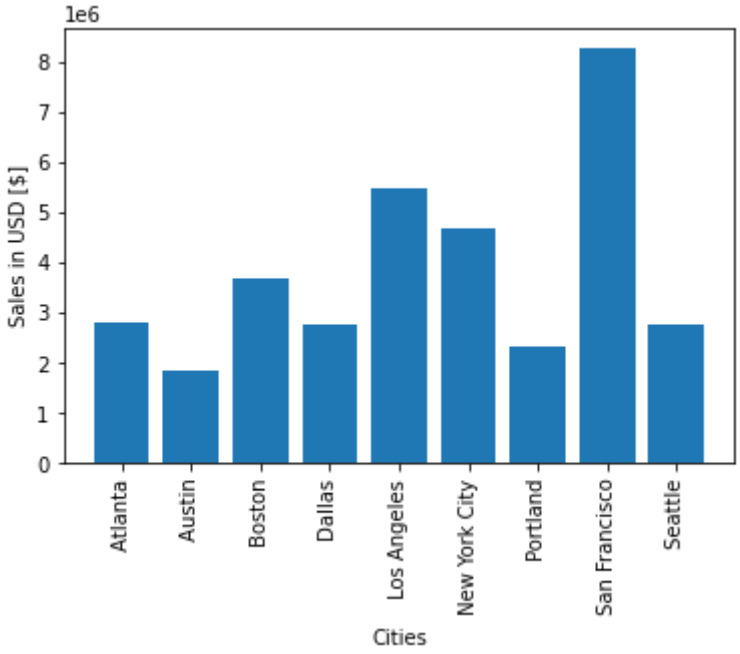
Out[33]:

City	Quantity Ordered	Price Each	Month	Sales
Atlanta	16602	2.779908e+06	104794	2.795499e+06
Austin	11153	1.809874e+06	69829	1.819582e+06
Boston	22528	3.637410e+06	141112	3.661642e+06
Dallas	16730	2.752628e+06	104620	2.767975e+06
Los Angeles	33289	5.421435e+06	208325	5.452571e+06
New York City	27932	4.635371e+06	175741	4.664317e+06
Portland	14053	2.307747e+06	87765	2.320491e+06
San Francisco	50239	8.211462e+06	315520	8.262204e+06
Seattle	16553	2.733296e+06	104941	2.747755e+06

In [34]:
```
#cities = all_data["City"].unique()

#Extraer los valores x de tabla resumen
cities = [city for city, df in all_data.groupby("City")]

plt.bar (cities, results["Sales"])
plt.xticks(cities, rotation = "vertical", size = 10)
plt.ylabel("Sales in USD [$]")
plt.xlabel("Cities")
plt.show()
```



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

In [35]:
```
#Previsualizar el dataframe
all_data.head()
```

Out[35]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	State
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

In [36]:
```
#Convertir columna Order Date a fecha
all_data['Order Date'] = pd.to_datetime(all_data['Order Date'])
```

In [37]:
```
#Previsualizar dataframe
all_data.head()
#Se debe observar que tiene una sintaxis diferente , respecto dataset anterior.
```

Out[37]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	State
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
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
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
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
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

In [38]:
```
#Crear una nueva columna con la hora
all_data['Hour'] = all_data['Order Date'].dt.hour
#Crear una nueva columna con los minutos
all_data['Minute'] = all_data['Order Date'].dt.minute
#Previsualizar dataframe
all_data.head()
```

Out[38]:

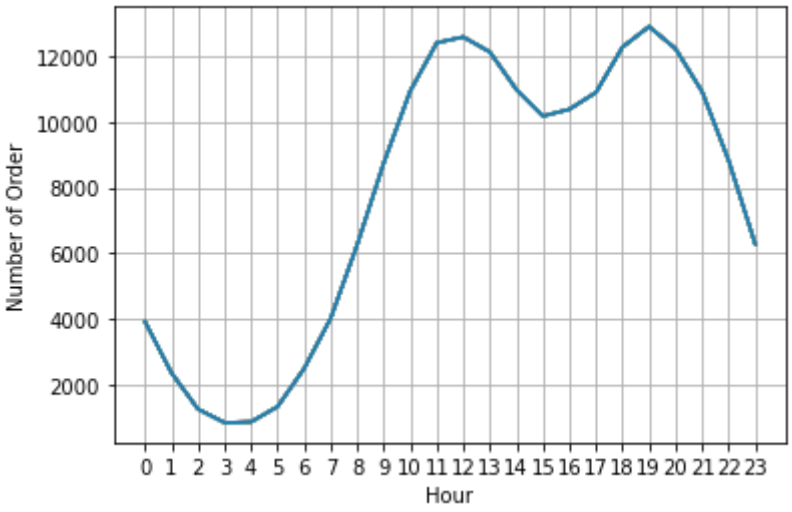
	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	State	Hour
0	176558	USB-C Charging Cable	2	11.95	2019-04-19 08:46:00	917 1st St, Dallas, TX 75001	4	23.90	Dallas	TX	8
2	176559	Bose SoundSport Headphones	1	99.99	2019-04-07 22:30:00	682 Chestnut St, Boston, MA 02215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

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	State	Hour
						CA 90001					
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
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

In [39]:

```
#Crear listado de horas unicas para eje x
hours = [hour for hour, df in all_data.groupby("Hour")]

#Crear gráfico de línea - linechart
plt.plot(hours, all_data.groupby(['Hour']).count())
plt.xticks(hours)
plt.grid()
plt.xlabel('Hour')
plt.ylabel('Number of Order')
plt.show()
#all_data.groupby(['Hour']).count()
```



In [40]:

```
#Respuesta: 11 am / 7 pm
```

Pregunta N° 4: ¿Qué productos son los más frecuentemente vendidos juntos?

In [41]:
```
#Crear un nuevo dataframe que permita ver solo las Ordenes ID duplicadas
df = all_data[all_data['Order ID'].duplicated(keep = False)]
df.head()
```

Out[41]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	State	Hour
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
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
18	176574	Google Phone	1	600.00	2019-04-03 19:42:00	20 Hill St, Los Angeles, CA 90001	4	600.00	Los Angeles	CA	19
19	176574	USB-C Charging Cable	1	11.95	2019-04-03 19:42:00	20 Hill St, Los Angeles, CA 90001	4	11.95	Los Angeles	CA	19
30	176585	Bose SoundSport Headphones	1	99.99	2019-04-07 11:31:00	823 Highland St, Boston, MA 02215	4	99.99	Boston	MA	11

In [42]:
```
#Crear nueva columna , agrupando en una misma celda los productos de la misma Order ID
df['Grouped'] = df.groupby('Order ID')['Product'].transform(lambda x:','.join(x))
df.head()
```

```
<ipython-input-42-0b95cff4c125>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['Grouped'] = df.groupby('Order ID')['Product'].transform(lambda x:','.join(x))
```

Out[42]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	State	Hour
3	176560	Google Phone	1	600.00	2019-04-12 14:38:00	669 Spruce St, Los	4	600.00	Los Angeles	CA	14

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	State	Hour
						Angeles, CA 90001					
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
18	176574	Google Phone	1	600.00	2019-04-03 19:42:00	20 Hill St, Los Angeles, CA 90001	4	600.00	Los Angeles	CA	19
19	176574	USB-C Charging Cable	1	11.95	2019-04-03 19:42:00	20 Hill St, Los Angeles, CA 90001	4	11.95	Los Angeles	CA	19
30	176585	Bose SoundSport Headphones	1	99.99	2019-04-07 11:31:00	823 Highland St, Boston, MA 02215	4	99.99	Boston	MA	11

In [43]:
```
#Crear nuevo dataframe, Eliminar los valores duplicados debido a la misma Order ID
df = df[['Order ID', 'Grouped']].drop_duplicates()
df.head()
```

Out[43]:

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

In [44]:
```
#Contar los pares de productos que mas se repitem

#Importar librería
from itertools import combinations
from collections import Counter
```

In [45]:
```
count = Counter()

for row in df['Grouped']:
    row_list = row.split(',')
    count.update(Counter(combinations(row_list,3)))

#Visualizar los diez mas frecuentes
count.most_common(10)
```

Out[45]:
```
[(('Google Phone', 'USB-C Charging Cable', 'Wired Headphones'), 87),
 (('iPhone', 'Lightning Charging Cable', 'Wired Headphones'), 62),
 (('iPhone', 'Lightning Charging Cable', 'Apple Airpods Headphones'), 47),
 (('Google Phone', 'USB-C Charging Cable', 'Bose SoundSport Headphones'), 35),
 (('Vareebadd Phone', 'USB-C Charging Cable', 'Wired Headphones'), 33),
 (('iPhone', 'Apple Airpods Headphones', 'Wired Headphones'), 27),
 (('Google Phone', 'Bose SoundSport Headphones', 'Wired Headphones'), 24),
 (('Vareebadd Phone', 'USB-C Charging Cable', 'Bose SoundSport Headphones'),
  16),
 (('USB-C Charging Cable', 'Bose SoundSport Headphones', 'Wired Headphones'),
  5),
 (('Vareebadd Phone', 'Bose SoundSport Headphones', 'Wired Headphones'), 5)]
```

In [47]:
```
for key, value in count.most_common(10):
    print(key, value)
```

```
('Google Phone', 'USB-C Charging Cable', 'Wired Headphones') 87
('iPhone', 'Lightning Charging Cable', 'Wired Headphones') 62
('iPhone', 'Lightning Charging Cable', 'Apple Airpods Headphones') 47
('Google Phone', 'USB-C Charging Cable', 'Bose SoundSport Headphones') 35
('Vareebadd Phone', 'USB-C Charging Cable', 'Wired Headphones') 33
('iPhone', 'Apple Airpods Headphones', 'Wired Headphones') 27
('Google Phone', 'Bose SoundSport Headphones', 'Wired Headphones') 24
('Vareebadd Phone', 'USB-C Charging Cable', 'Bose SoundSport Headphones') 16
('USB-C Charging Cable', 'Bose SoundSport Headphones', 'Wired Headphones') 5
('Vareebadd Phone', 'Bose SoundSport Headphones', 'Wired Headphones') 5
```

Pregunta N° 5: ¿Qué producto se vendio más?

In [48]:
```
all_data.head()
```

Out[48]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	State	Hour
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
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
3	176560	Google Phone	1	600.00	2019-04-12 14:38:00	669 Spruce St, Los Angeles,	4	600.00	Los Angeles	CA	14

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	State	Hour
						CA 90001					
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
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

In [49]:
```
#Agrupar segun Product y Cantidad Ordenada
product_group = all_data.groupby('Product')
product_group.sum()
```

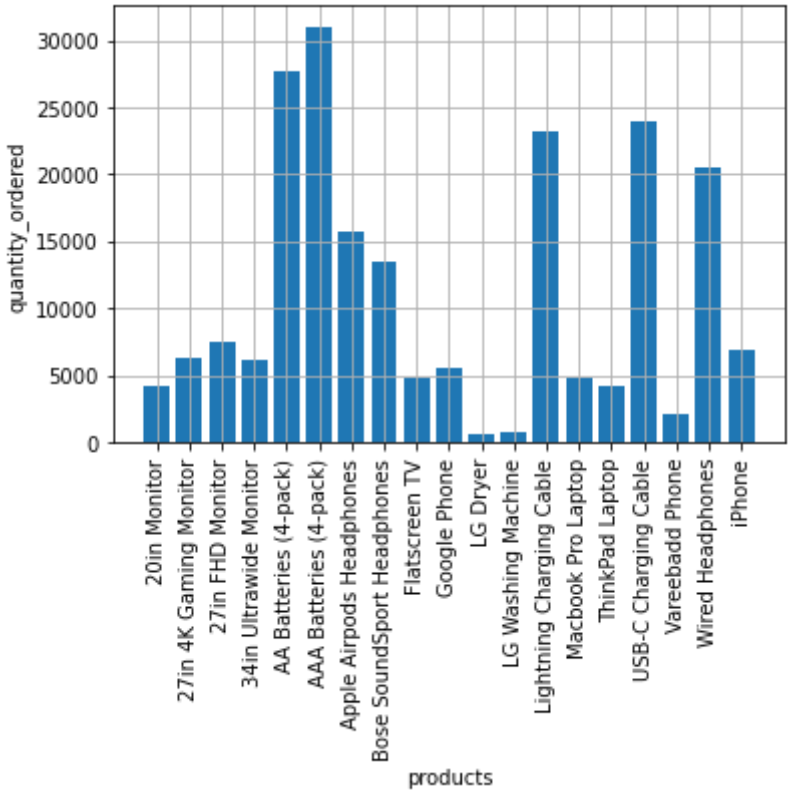
Out[49]:

Product	Quantity Ordered	Price Each	Month	Sales	Hour	Minute
20in Monitor	4129	451068.99	29336	454148.71	58764	122252
27in 4K Gaming Monitor	6244	2429637.70	44440	2435097.56	90916	184331
27in FHD Monitor	7550	1125974.93	52558	1132424.50	107540	219948
34in Ultrawide Monitor	6199	2348718.19	43304	2355558.01	89076	183480
AA Batteries (4-pack)	27635	79015.68	145558	106118.40	298342	609039
AAA Batteries (4-pack)	31017	61716.59	146370	92740.83	297332	612113
Apple Airpods Headphones	15661	2332350.00	109477	2349150.00	223304	455570
Bose SoundSport Headphones	13457	1332366.75	94113	1345565.43	192445	392603
Flatscreen TV	4819	1440000.00	34224	1445700.00	68815	142789
Google Phone	5532	3315000.00	38305	3319200.00	79479	162773
LG Dryer	646	387600.00	4383	387600.00	9326	19043
LG Washing Machine	666	399600.00	4523	399600.00	9785	19462
Lightning Charging Cable	23217	323787.10	153092	347094.15	312529	634442
Macbook Pro Laptop	4728	8030800.00	33548	8037600.00	68261	137574
ThinkPad Laptop	4130	4127958.72	28950	4129958.70	59746	121508
USB-C Charging Cable	23975	261740.85	154819	286501.25	314645	647586
Vareebadd Phone	2068	826000.00	14309	827200.00	29472	61835
Wired Headphones	20557	226395.18	133397	246478.43	271720	554023
iPhone	6849	4789400.00	47941	4794300.00	98657	201688

In [50]:
```
quantity_ordered = product_group.sum()['Quantity Ordered']


#Crear listado de horas unicas para eje x
products = [product for product, df in all_data.groupby("Product")]
```

In [53]:
```
plt.bar(products, quantity_ordered)
plt.xticks(products, rotation = "vertical", size = 10)
plt.ylabel("quantity_ordered")
plt.xlabel("products")
plt.grid()
plt.show()
```



In [54]:
```
#Agregar un segundo eje Y
prices = all_data.groupby('Product').mean()['Price Each']
print(prices)
```

```
Product
20in Monitor                  109.99
27in 4K Gaming Monitor        389.99
27in FHD Monitor              149.99
34in Ultrawide Monitor        379.99
AA Batteries (4-pack)           3.84
AAA Batteries (4-pack)          2.99
Apple Airpods Headphones      150.00
Bose SoundSport Headphones     99.99
Flatscreen TV                 300.00
Google Phone                  600.00
LG Dryer                      600.00
LG Washing Machine            600.00
Lightning Charging Cable       14.95
```

```
Macbook Pro Laptop            1700.00
ThinkPad Laptop                999.99
USB-C Charging Cable            11.95
Vareebadd Phone                400.00
Wired Headphones                11.99
iPhone                         700.00
Name: Price Each, dtype: float64
```

Agregar un segundo eje Y a la grafica

In [68]:

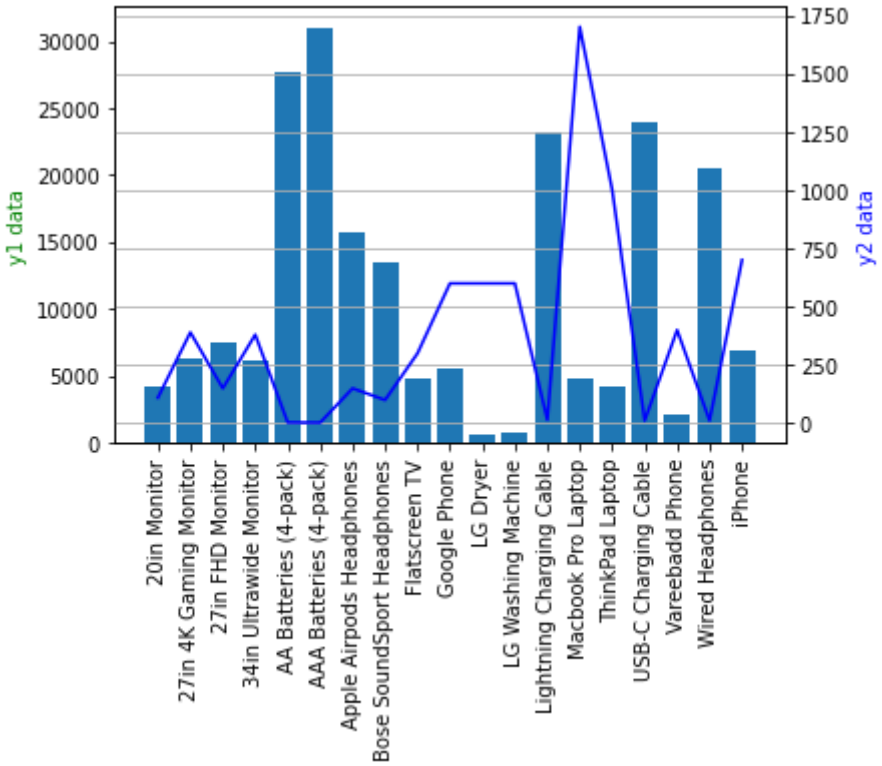
```
fig, ax1 = plt.subplots()


ax2 = ax1.twinx()
ax1.bar(products, quantity_ordered)
ax2.plot(products, prices, 'b')


plt.grid()
ax1.set_ylabel('y1 data', color ='g')
ax2.set_ylabel('y2 data', color = 'b' )
ax1.set_xticklabels(products, rotation = "vertical", size = 10)


plt.show()
```

```
<ipython-input-68-b0cfb551f52f>:10: UserWarning: FixedFormatter should only be used
together with FixedLocator
  ax1.set_xticklabels(products, rotation = "vertical", size = 10)
```



In [ ]: