

# Analyse des données de ventes d'une entreprise

October 24, 2024

## 1 Découverte des données

### 1.1 Importation des bibliothèques

```
[1]: # importer les packages nécessaires
```

```
import os
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
[3]: #Collecte des noms de fichiers de dataset
```

```
files = [file for file in os.listdir(r"C:/Users/ordinateur/Desktop/dossier/
python/Analyse de données de ventes d'une entreprise/data-sales-main")]
for file in files :
    print(file)
```

```
Sales_April_2019.csv
Sales_August_2019.csv
Sales_December_2019.csv
Sales_February_2019.csv
Sales_January_2019.csv
Sales_July_2019.csv
Sales_June_2019.csv
Sales_March_2019.csv
Sales_May_2019.csv
Sales_November_2019.csv
Sales_October_2019.csv
Sales_September_2019.csv
```

```
[186]: #Afficher les données d'un seul fichier
```

```
path = "C:/Users/ordinateur/Desktop/dossier/python/Analyse de données de ventes_
d'une entreprise/data-sales-main/Sales_April_2019.csv"
April_data = pd.read_csv(path, delimiter=',')
display(April_data)
```

	Order ID	Product	Quantity Ordered	Price Each	\
0	176558	USB-C Charging Cable	2	11.95	

1	NaN	NaN	NaN	NaN
2	176559	Bose SoundSport Headphones	1	99.99
3	176560	Google Phone	1	600
4	176560	Wired Headphones	1	11.99
...	...	...	...	...
18378	194090	Google Phone	1	600
18379	194091	AA Batteries (4-pack)	1	3.84
18380	194092	AAA Batteries (4-pack)	2	2.99
18381	194093	AA Batteries (4-pack)	1	3.84
18382	194094	Lightning Charging Cable	1	14.95

	Order Date	Purchase Address
0	04/19/19 08:46	917 1st St, Dallas, TX 75001
1	NaN	NaN
2	04/07/19 22:30	682 Chestnut St, Boston, MA 02215
3	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001
4	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001
...	...	...
18378	04/08/19 17:11	177 Jackson St, Los Angeles, CA 90001
18379	04/15/19 16:02	311 Forest St, Austin, TX 73301
18380	04/28/19 14:36	347 Sunset St, San Francisco, CA 94016
18381	04/14/19 15:09	835 Lake St, Portland, OR 97035
18382	04/18/19 11:08	354 North St, Boston, MA 02215

[18383 rows x 6 columns]

```
[5]: #Fusionner tous les fichiers en un seul

path = "C:/Users/ordinateur/Desktop/dossier/python/Analyse de données de ventes_
      ↪d'une entreprise/data-sales-main"

#créer une base de données vide

all_data = pd.DataFrame()

for file in files :
    current_data = pd.read_csv(path+('/')+file)
    all_data = pd.concat([all_data, current_data])
```

```
[7]: all_data.shape
```

```
[7]: (186850, 6)
```

```
[9]: #supprimer les valeurs manquantes
all_data = all_data.dropna(how='all')
print(all_data)
```

Order ID	Product	Quantity Ordered	Price Each	\
----------	---------	------------------	------------	---

0	176558	USB-C Charging Cable	2	11.95
2	176559	Bose SoundSport Headphones	1	99.99
3	176560	Google Phone	1	600
4	176560	Wired Headphones	1	11.99
5	176561	Wired Headphones	1	11.99
...	...	...	...	...
11681	259353	AAA Batteries (4-pack)	3	2.99
11682	259354	iPhone	1	700
11683	259355	iPhone	1	700
11684	259356	34in Ultrawide Monitor	1	379.99
11685	259357	USB-C Charging Cable	1	11.95

	Order Date	Purchase Address
0	04/19/19 08:46	917 1st St, Dallas, TX 75001
2	04/07/19 22:30	682 Chestnut St, Boston, MA 02215
3	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001
4	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001
5	04/30/19 09:27	333 8th St, Los Angeles, CA 90001
...	...	...
11681	09/17/19 20:56	840 Highland St, Los Angeles, CA 90001
11682	09/01/19 16:00	216 Dogwood St, San Francisco, CA 94016
11683	09/23/19 07:39	220 12th St, San Francisco, CA 94016
11684	09/19/19 17:30	511 Forest St, San Francisco, CA 94016
11685	09/30/19 00:18	250 Meadow St, San Francisco, CA 94016

[186305 rows x 6 columns]

```
[11]: # Renommer les colonnes
all_data = all_data.rename(columns={
    'Order ID': 'ID Commande',
    'Product': 'Produit',
    'Quantity Ordered': 'Quantité Commandée',
    'Price Each': 'Prix Unitaire',
    'Order Date': 'Date de Commande',
    'Purchase Address': 'Adresse d\'Achat'
})
```

```
[13]: #générer en fichier csv
all_data.to_csv(path+'/all_data.csv', index=False)
```

```
[15]: #afficher les données du fichier global de ventes
all_data = pd.read_csv(path+'/all_data.csv')
all_data.head(200)
```

[15]:	ID Commande	Produit	Quantité Commandée	Prix Unitaire \
0	176558	USB-C Charging Cable	2	11.95
1	176559	Bose SoundSport Headphones	1	99.99
2	176560	Google Phone	1	600

3	176560	Wired Headphones	1	11.99
4	176561	Wired Headphones	1	11.99
..	...	...	...	...
195	176745	Wired Headphones	1	11.99
196	176746	27in 4K Gaming Monitor	1	389.99
197	176747	Macbook Pro Laptop	1	1700
198	176748	Flatscreen TV	1	300
199	176749	ThinkPad Laptop	1	999.99

	Date de Commande	Adresse d'Achat
0	04/19/19 08:46	917 1st St, Dallas, TX 75001
1	04/07/19 22:30	682 Chestnut St, Boston, MA 02215
2	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001
3	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001
4	04/30/19 09:27	333 8th St, Los Angeles, CA 90001
..	...	...
195	04/24/19 09:23	461 Adams St, Los Angeles, CA 90001
196	04/08/19 19:04	638 Park St, Seattle, WA 98101
197	04/27/19 15:04	645 7th St, San Francisco, CA 94016
198	04/24/19 15:28	339 13th St, Boston, MA 02215
199	04/26/19 07:55	322 Adams St, Seattle, WA 98101

[200 rows x 6 columns]

```
[17]: all_data.dtypes
```

```
[17]: ID Commande      object
      Produit         object
      Quantité Commandée object
      Prix Unitaire    object
      Date de Commande object
      Adresse d'Achat  object
      dtype: object
```

```
[19]: all_data.isnull().sum()
```

```
[19]: ID Commande      0
      Produit         0
      Quantité Commandée 0
      Prix Unitaire    0
      Date de Commande 0
      Adresse d'Achat  0
      dtype: int64
```

```
[21]: all_data.shape
```

```
[21]: (186305, 6)
```

```
[23]: all_data.describe()
```

```
[23]:
```

	ID Commande	Produit	Quantité Commandée	Prix Unitaire \
count	186305	186305	186305	186305
unique	178438	20	10	24
top	Order ID	USB-C Charging Cable	1	11.95
freq	355	21903	168552	21903

	Date de Commande	Adresse d'Achat
count	186305	186305
unique	142396	140788
top	Order Date	Purchase Address
freq	355	355

## 2 Quel est le mois durant lequel il y'a le meilleur chiffre d'affaires

```
[25]: def month(x) :
        return x.split('/')[0]

all_data['Mois'] = all_data['Date de Commande'].apply(month)
display(all_data)
```

	ID Commande	Produit	Quantité Commandée \
0	176558	USB-C Charging Cable	2
1	176559	Bose SoundSport Headphones	1
2	176560	Google Phone	1
3	176560	Wired Headphones	1
4	176561	Wired Headphones	1
...	...	...	...
186300	259353	AAA Batteries (4-pack)	3
186301	259354	iPhone	1
186302	259355	iPhone	1
186303	259356	34in Ultrawide Monitor	1
186304	259357	USB-C Charging Cable	1

	Prix Unitaire	Date de Commande \
0	11.95	04/19/19 08:46
1	99.99	04/07/19 22:30
2	600	04/12/19 14:38
3	11.99	04/12/19 14:38
4	11.99	04/30/19 09:27
...	...	...
186300	2.99	09/17/19 20:56
186301	700	09/01/19 16:00
186302	700	09/23/19 07:39
186303	379.99	09/19/19 17:30
186304	11.95	09/30/19 00:18

	Adresse d'Achat	Mois
0	917 1st St, Dallas, TX 75001	04
1	682 Chestnut St, Boston, MA 02215	04
2	669 Spruce St, Los Angeles, CA 90001	04
3	669 Spruce St, Los Angeles, CA 90001	04
4	333 8th St, Los Angeles, CA 90001	04
...	...	...
186300	840 Highland St, Los Angeles, CA 90001	09
186301	216 Dogwood St, San Francisco, CA 94016	09
186302	220 12th St, San Francisco, CA 94016	09
186303	511 Forest St, San Francisco, CA 94016	09
186304	250 Meadow St, San Francisco, CA 94016	09

[186305 rows x 7 columns]

```
[27]: # Vérification avant l'exclusion pour voir si toutes les colonnes sont présentes
print(all_data.columns)

# Exclure les lignes où la colonne 'Mois' est égale à 'Order Date'
all_data = all_data[all_data['Mois'] != 'Order Date']

# Vérifier à nouveau les colonnes après la modification
print(all_data.columns)

# Afficher les éléments uniques de la colonne 'Mois'
mois_uniques = all_data['Mois'].unique()

# Afficher les mois uniques
print(mois_uniques)

# Afficher un échantillon des premières lignes du DataFrame pour voir toutes
↳ les colonnes
print(all_data.head())
```

```
Index(['ID Commande', 'Produit', 'Quantité Commandée', 'Prix Unitaire',
      'Date de Commande', 'Adresse d'Achat', 'Mois'],
      dtype='object')
Index(['ID Commande', 'Produit', 'Quantité Commandée', 'Prix Unitaire',
      'Date de Commande', 'Adresse d'Achat', 'Mois'],
      dtype='object')
['04' '05' '08' '09' '12' '01' '02' '03' '07' '06' '11' '10']
ID Commande      Produit  Quantité Commandée  Prix Unitaire \
0      176558      USB-C Charging Cable          2         11.95
1      176559  Bose SoundSport Headphones          1         99.99
2      176560          Google Phone              1           600
3      176560      Wired Headphones              1         11.99
4      176561      Wired Headphones              1         11.99
```

	Date de Commande	Adresse d'Achat	Mois
0	04/19/19 08:46	917 1st St, Dallas, TX 75001	04
1	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	04
2	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	04
3	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	04
4	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	04

```
[29]: all_data['Mois'] = all_data['Mois'].astype(int)
all_data['Prix Unitaire'] = all_data['Prix Unitaire'].astype(float)
all_data['Quantité Commandée'] = all_data['Quantité Commandée'].astype(int)
```

```
[31]: all_data.dtypes
```

```
[31]: ID Commande      object
Produit              object
Quantité Commandée   int32
Prix Unitaire        float64
Date de Commande     object
Adresse d'Achat      object
Mois                 int32
dtype: object
```

```
[33]: all_data["Ventes"] = all_data['Quantité Commandée'] * all_data['Prix Unitaire']
all_data
```

```
[33]:
```

	ID Commande	Produit	Quantité Commandée \
0	176558	USB-C Charging Cable	2
1	176559	Bose SoundSport Headphones	1
2	176560	Google Phone	1
3	176560	Wired Headphones	1
4	176561	Wired Headphones	1
...	...	...	...
186300	259353	AAA Batteries (4-pack)	3
186301	259354	iPhone	1
186302	259355	iPhone	1
186303	259356	34in Ultrawide Monitor	1
186304	259357	USB-C Charging Cable	1

	Prix Unitaire	Date de Commande \
0	11.95	04/19/19 08:46
1	99.99	04/07/19 22:30
2	600.00	04/12/19 14:38
3	11.99	04/12/19 14:38
4	11.99	04/30/19 09:27
...	...	...
186300	2.99	09/17/19 20:56
186301	700.00	09/01/19 16:00

```
186302      700.00   09/23/19 07:39
186303      379.99   09/19/19 17:30
186304       11.95   09/30/19 00:18
```

	Adresse d'Achat	Mois	Ventes
0	917 1st St, Dallas, TX 75001	4	23.90
1	682 Chestnut St, Boston, MA 02215	4	99.99
2	669 Spruce St, Los Angeles, CA 90001	4	600.00
3	669 Spruce St, Los Angeles, CA 90001	4	11.99
4	333 8th St, Los Angeles, CA 90001	4	11.99
...	...	...	...
186300	840 Highland St, Los Angeles, CA 90001	9	8.97
186301	216 Dogwood St, San Francisco, CA 94016	9	700.00
186302	220 12th St, San Francisco, CA 94016	9	700.00
186303	511 Forest St, San Francisco, CA 94016	9	379.99
186304	250 Meadow St, San Francisco, CA 94016	9	11.95

[185950 rows x 8 columns]

```
[35]: mois_uniques = all_data['Mois'].unique()
      print(mois_uniques)
```

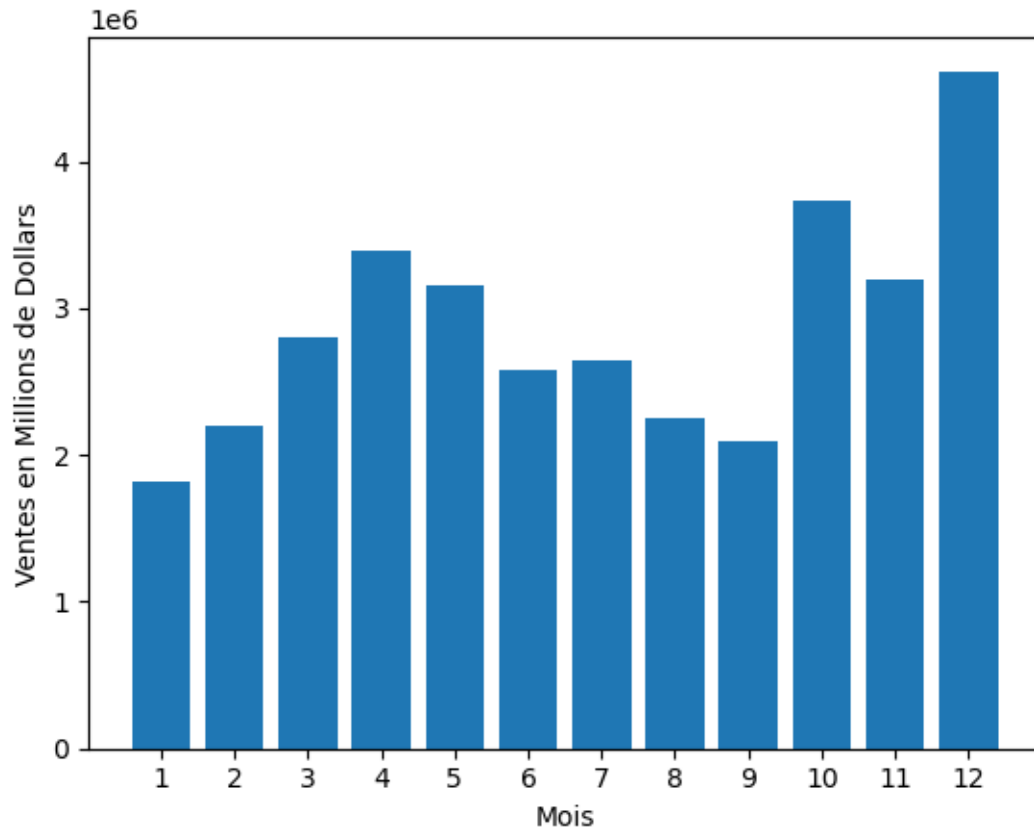
```
[ 4  5  8  9 12  1  2  3  7  6 11 10]
```

```
[37]: all_data.groupby('Mois')['Ventes'].sum()
```

```
[37]: Mois
1      1822256.73
2      2202022.42
3      2807100.38
4      3390670.24
5      3152606.75
6      2577802.26
7      2647775.76
8      2244467.88
9      2097560.13
10     3736726.88
11     3199603.20
12     4613443.34
Name: Ventes, dtype: float64
```

```
[45]: Mois = range(1,13)
      plt.bar(Mois, all_data.groupby('Mois')['Ventes'].sum())
      plt.xticks(Mois)
      plt.ylabel('Ventes en Millions de Dollars')
      plt.xlabel('Mois')
      plt.show()
```





### 3 Dans quelle ville avons nous enregistré un maximum de commandes ?

```
[47]: def city(x) :
      return x.split(',')[1]

all_data['Ville'] = all_data["Adresse d'Achat"].apply(city)
display(all_data)
```

	ID Commande	Produit	Quantité Commandée	\
0	176558	USB-C Charging Cable	2	
1	176559	Bose SoundSport Headphones	1	
2	176560	Google Phone	1	
3	176560	Wired Headphones	1	
4	176561	Wired Headphones	1	
...	...	...	...	
186300	259353	AAA Batteries (4-pack)	3	
186301	259354	iPhone	1	
186302	259355	iPhone	1	
186303	259356	34in Ultrawide Monitor	1	

186304	259357	USB-C Charging Cable	1
--------	--------	----------------------	---

	Prix Unitaire	Date de Commande	\
0	11.95	04/19/19	08:46
1	99.99	04/07/19	22:30
2	600.00	04/12/19	14:38
3	11.99	04/12/19	14:38
4	11.99	04/30/19	09:27
...	...	...	...
186300	2.99	09/17/19	20:56
186301	700.00	09/01/19	16:00
186302	700.00	09/23/19	07:39
186303	379.99	09/19/19	17:30
186304	11.95	09/30/19	00:18

	Adresse d'Achat	Mois	Ventes	Ville
0	917 1st St, Dallas, TX 75001	4	23.90	Dallas
1	682 Chestnut St, Boston, MA 02215	4	99.99	Boston
2	669 Spruce St, Los Angeles, CA 90001	4	600.00	Los Angeles
3	669 Spruce St, Los Angeles, CA 90001	4	11.99	Los Angeles
4	333 8th St, Los Angeles, CA 90001	4	11.99	Los Angeles
...	...	...	...	...
186300	840 Highland St, Los Angeles, CA 90001	9	8.97	Los Angeles
186301	216 Dogwood St, San Francisco, CA 94016	9	700.00	San Francisco
186302	220 12th St, San Francisco, CA 94016	9	700.00	San Francisco
186303	511 Forest St, San Francisco, CA 94016	9	379.99	San Francisco
186304	250 Meadow St, San Francisco, CA 94016	9	11.95	San Francisco

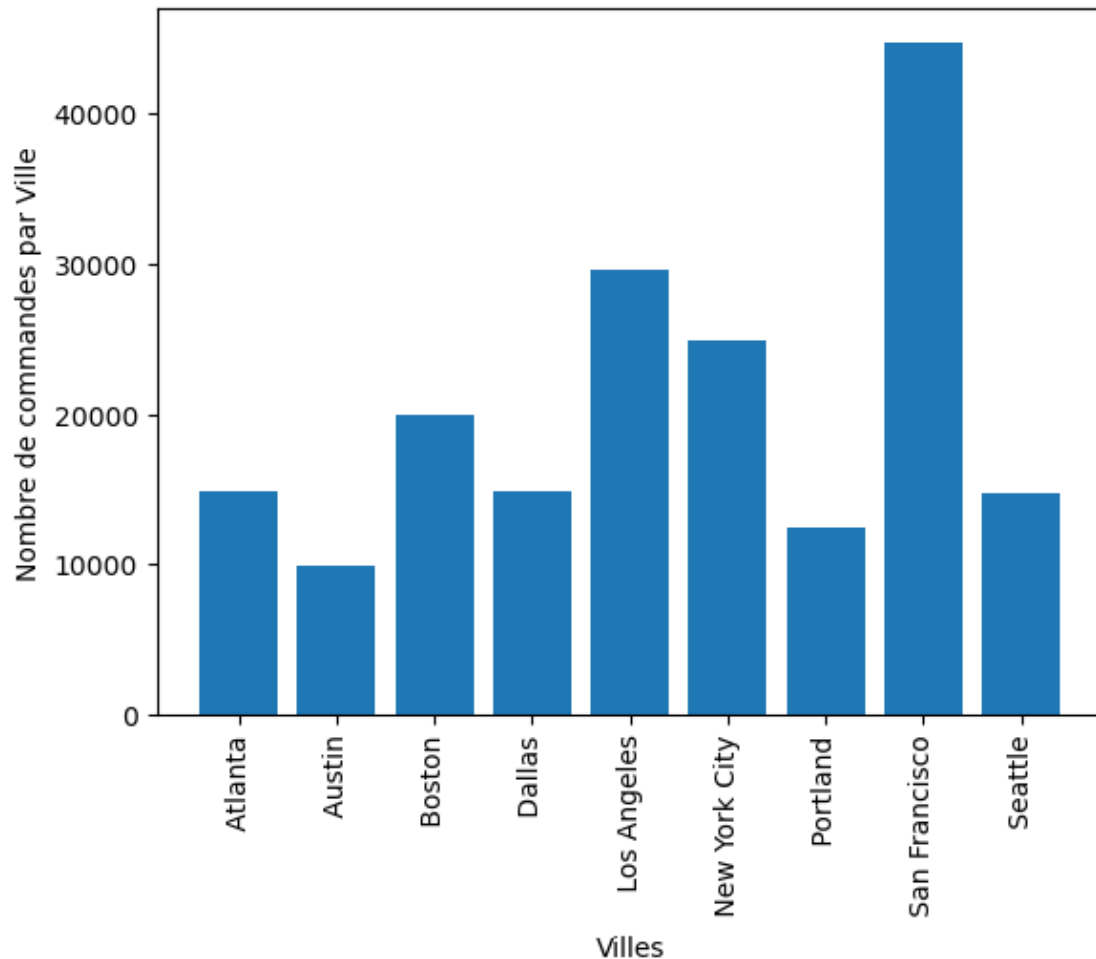
[185950 rows x 9 columns]

```
[49]: # Afficher les éléments uniques de la colonne 'Mois'
      villes_uniques = all_data['Ville'].unique()

      # Afficher les mois uniques
      print(villes_uniques)
```

```
[' Dallas' ' Boston' ' Los Angeles' ' San Francisco' ' Seattle' ' Atlanta'
 ' New York City' ' Portland' ' Austin']
```

```
[51]: plt.bar(all_data.groupby('Ville')['Ville'].count().index, all_data.
      ↪groupby('Ville')['Ville'].count())
      plt.xticks(rotation='vertical')
      plt.ylabel('Nombre de commandes par Ville')
      plt.xlabel('Villes')
      plt.show()
```



#### 4 À quel moment doit on faire des campagnes publicitaires pour faire plus de ventes ?

```
[53]: all_data['Heure']=pd.to_datetime(all_data['Date de Commande']).dt.hour
```

```
C:\Users\ordinateur\AppData\Local\Temp\ipykernel_16220\1412383529.py:1:
UserWarning: Could not infer format, so each element will be parsed
individually, falling back to `dateutil`. To ensure parsing is consistent and
as-expected, please specify a format.
```

```
all_data['Heure']=pd.to_datetime(all_data['Date de Commande']).dt.hour
```

```
[54]: all_data
```

```
[54]:
```

	ID Commande	Produit	Quantité Commandée	\
0	176558	USB-C Charging Cable	2	
1	176559	Bose SoundSport Headphones	1	

2	176560	Google Phone	1
3	176560	Wired Headphones	1
4	176561	Wired Headphones	1
...	...	...	...
186300	259353	AAA Batteries (4-pack)	3
186301	259354	iPhone	1
186302	259355	iPhone	1
186303	259356	34in Ultrawide Monitor	1
186304	259357	USB-C Charging Cable	1

	Prix Unitaire	Date de Commande \
0	11.95	04/19/19 08:46
1	99.99	04/07/19 22:30
2	600.00	04/12/19 14:38
3	11.99	04/12/19 14:38
4	11.99	04/30/19 09:27
...	...	...
186300	2.99	09/17/19 20:56
186301	700.00	09/01/19 16:00
186302	700.00	09/23/19 07:39
186303	379.99	09/19/19 17:30
186304	11.95	09/30/19 00:18

	Adresse d'Achat	Mois	Ventes	Ville \
0	917 1st St, Dallas, TX 75001	4	23.90	Dallas
1	682 Chestnut St, Boston, MA 02215	4	99.99	Boston
2	669 Spruce St, Los Angeles, CA 90001	4	600.00	Los Angeles
3	669 Spruce St, Los Angeles, CA 90001	4	11.99	Los Angeles
4	333 8th St, Los Angeles, CA 90001	4	11.99	Los Angeles
...	...	...	...	...
186300	840 Highland St, Los Angeles, CA 90001	9	8.97	Los Angeles
186301	216 Dogwood St, San Francisco, CA 94016	9	700.00	San Francisco
186302	220 12th St, San Francisco, CA 94016	9	700.00	San Francisco
186303	511 Forest St, San Francisco, CA 94016	9	379.99	San Francisco
186304	250 Meadow St, San Francisco, CA 94016	9	11.95	San Francisco

	Heure
0	8
1	22
2	14
3	14
4	9
...	...
186300	20
186301	16
186302	7
186303	17

186304        0

[185950 rows x 10 columns]

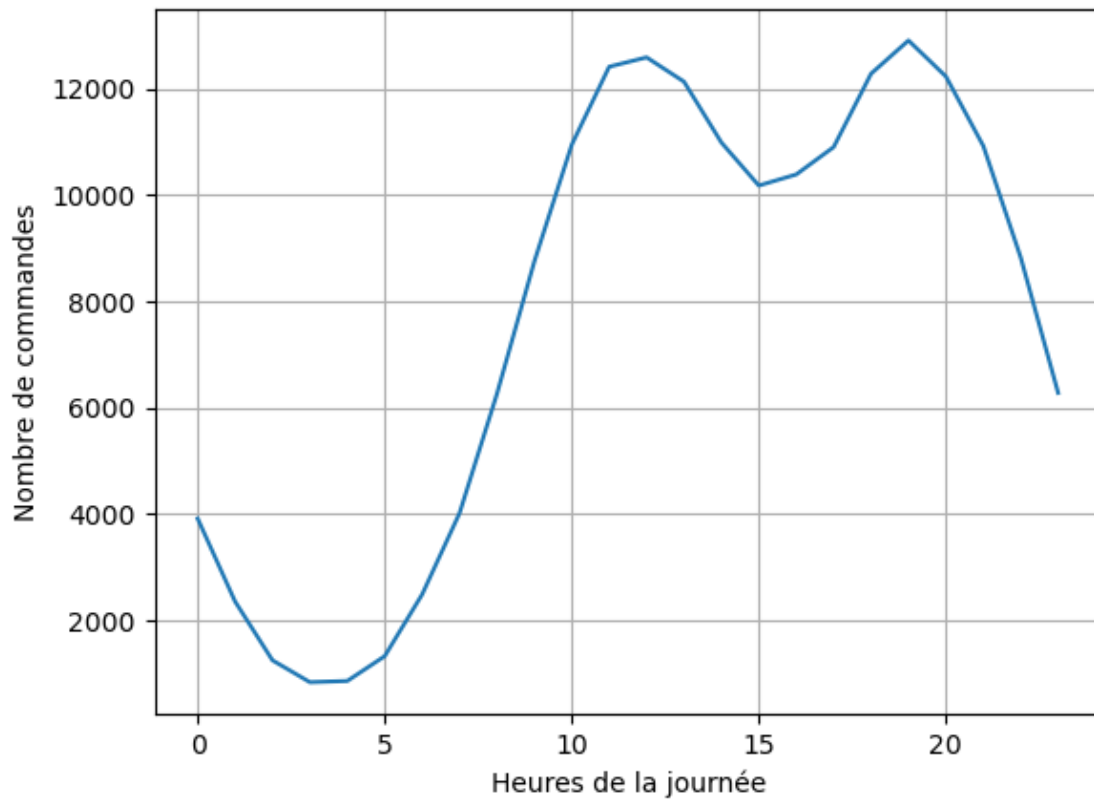
```
[55]: keys=[]
      hours=[]
      for key, hour in all_data.groupby('Heure'):
          keys.append(key)
          hours.append(len(hour))

      print(hours)
```

```
[3910, 2350, 1243, 831, 854, 1321, 2482, 4011, 6256, 8748, 10944, 12411, 12587,
12129, 10984, 10175, 10384, 10899, 12280, 12905, 12228, 10921, 8822, 6275]
```

```
[56]: plt.grid()
      plt.plot(keys, hours)
      plt.xlabel('Heures de la journée')
      plt.ylabel('Nombre de commandes')
```

```
[56]: Text(0, 0.5, 'Nombre de commandes')
```

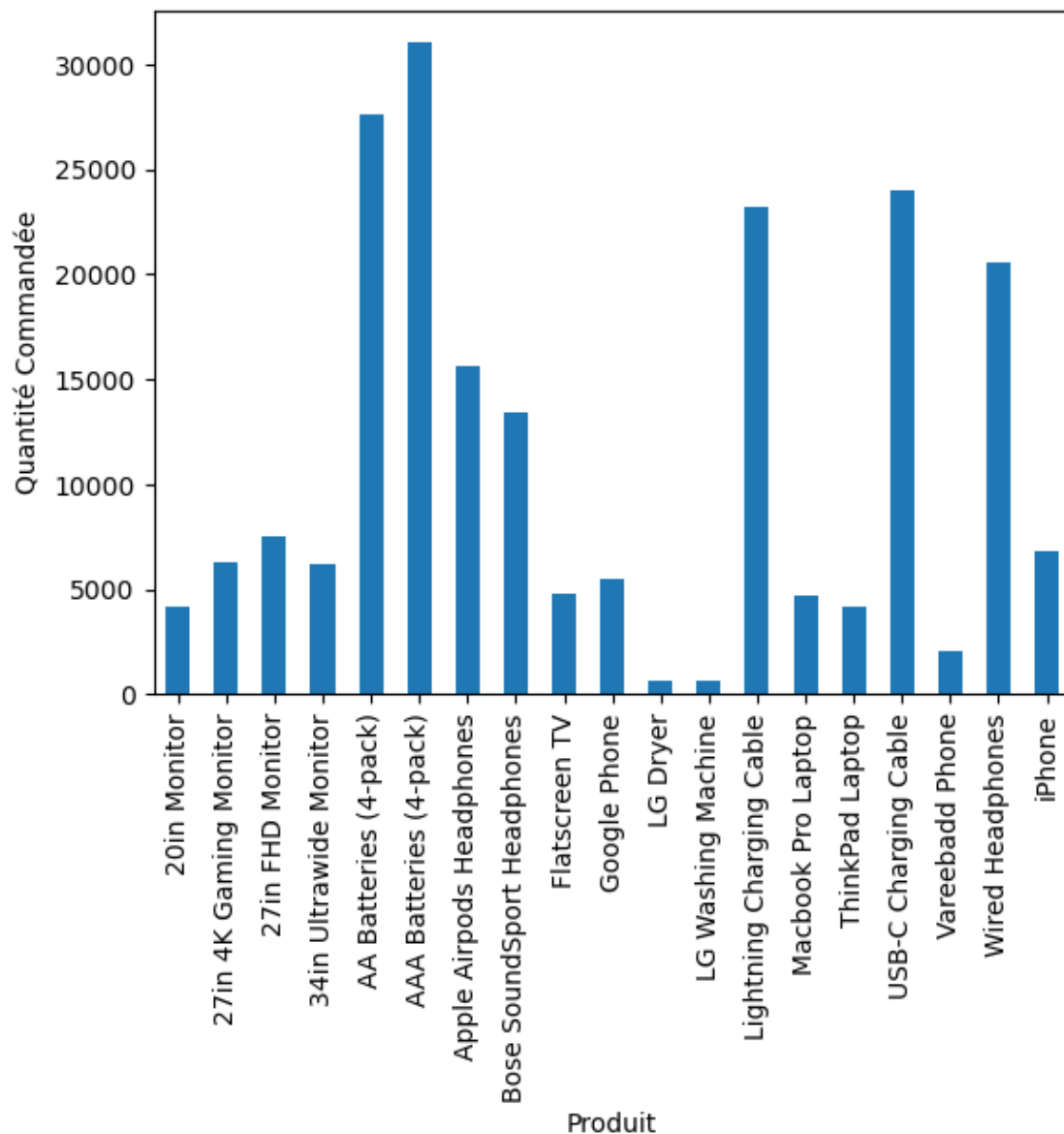


## 5 Quel produit se vend le plus, et pourquoi?

### 5.0.1 Quel produit se vend le plus

```
[57]: all_data.groupby('Produit')['Quantité Commandée'].sum().plot(kind='bar')
plt.xlabel('Produit')
plt.ylabel('Quantité Commandée')
```

```
[57]: Text(0, 0.5, 'Quantité Commandée')
```



Le produit qui se vend le plus est la batterie suivie par les cables de chargeur lumineux, les cable de chargeur type USB-C et les écouteurs sans fils

### 5.0.2 Et pourquoi ?

```
[58]: all_data.groupby('Produit')['Prix Unitaire'].mean()
```

```
[58]: Produit
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: Prix Unitaire, dtype: float64
```

```
[59]: #Afficher les données sur un visuel avec les trois variables Produit, Quantité
      ↪ et le Prix
produit = all_data.groupby('Produit')['Quantité Commandée'].sum().index
quantite = all_data.groupby('Produit')['Quantité Commandée'].sum()
prix = all_data.groupby('Produit')['Prix Unitaire'].mean()
```

```
[60]: plt.figure(figsize=(40,24))
fig,ax1=plt.subplots()
ax2=ax1.twinx()
ax1.bar(produit,quantite,color='g')
ax2.plot(produit,prix,'b-')
ax1.set_xticklabels(produit,rotation='vertical',size=8)
```

```
C:\Users\ordinateur\AppData\Local\Temp\ipykernel_16220\792966097.py:6:
UserWarning: set_ticklabels() should only be used with a fixed number of ticks,
i.e. after set_ticks() or using a FixedLocator.
    ax1.set_xticklabels(produit,rotation='vertical',size=8)
```

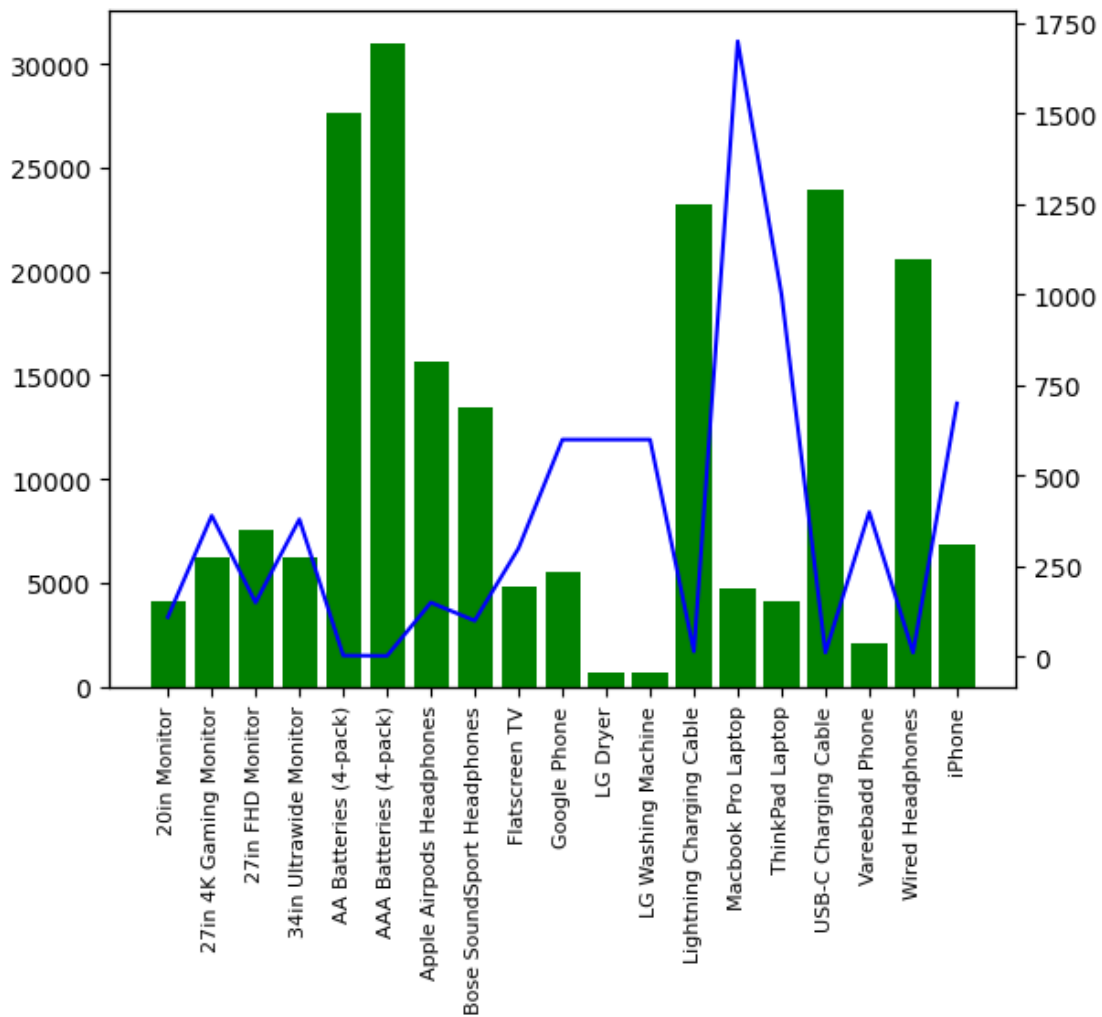
```
[60]: [Text(0, 0, '20in Monitor'),
      Text(1, 0, '27in 4K Gaming Monitor'),
      Text(2, 0, '27in FHD Monitor'),
      Text(3, 0, '34in Ultrawide Monitor'),
```

```

Text(4, 0, 'AA Batteries (4-pack)'),
Text(5, 0, 'AAA Batteries (4-pack)'),
Text(6, 0, 'Apple Airpods Headphones'),
Text(7, 0, 'Bose SoundSport Headphones'),
Text(8, 0, 'Flatscreen TV'),
Text(9, 0, 'Google Phone'),
Text(10, 0, 'LG Dryer'),
Text(11, 0, 'LG Washing Machine'),
Text(12, 0, 'Lightning Charging Cable'),
Text(13, 0, 'Macbook Pro Laptop'),
Text(14, 0, 'ThinkPad Laptop'),
Text(15, 0, 'USB-C Charging Cable'),
Text(16, 0, 'Vareebadd Phone'),
Text(17, 0, 'Wired Headphones'),
Text(18, 0, 'iPhone')]

```

<Figure size 4000x2400 with 0 Axes>





## 6 Quelles sont les combinaisons de produits qui se vendent le plus ?

```
[61]: df = all_data[all_data['ID Commande'].duplicated(keep=False)]
df['Groupés'] = df.groupby('ID Commande')['Produit'].transform(lambda x : ','.join(x))
```

C:\Users\ordinateur\AppData\Local\Temp\ipykernel\_16220\4005855878.py: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](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
df['Groupés'] = df.groupby('ID Commande')['Produit'].transform(lambda x : ','.join(x))
```

```
[69]: df
```

```
[69]:
```

	ID Commande	Produit	Quantité Commandée	\
2	176560	Google Phone	1	
3	176560	Wired Headphones	1	
17	176574	Google Phone	1	
18	176574	USB-C Charging Cable	1	
29	176585	Bose SoundSport Headphones	1	
...	...	...	...	
186248	259303	AA Batteries (4-pack)	1	
186259	259314	Wired Headphones	1	
186260	259314	AAA Batteries (4-pack)	2	
186296	259350	Google Phone	1	
186297	259350	USB-C Charging Cable	1	

	Prix Unitaire	Date de Commande	Adresse d'Achat	\
2	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	
3	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	
17	600.00	04/03/19 19:42	20 Hill St, Los Angeles, CA 90001	
18	11.95	04/03/19 19:42	20 Hill St, Los Angeles, CA 90001	
29	99.99	04/07/19 11:31	823 Highland St, Boston, MA 02215	
...	...	...	...	
186248	3.84	09/20/19 20:18	106 7th St, Atlanta, GA 30301	
186259	11.99	09/16/19 00:25	241 Highland St, Atlanta, GA 30301	
186260	2.99	09/16/19 00:25	241 Highland St, Atlanta, GA 30301	
186296	600.00	09/30/19 13:49	519 Maple St, San Francisco, CA 94016	
186297	11.95	09/30/19 13:49	519 Maple St, San Francisco, CA 94016	

	Mois	Ventes	Ville	Heure	\
2	4	600.00	Los Angeles	14	
3	4	11.99	Los Angeles	14	
17	4	600.00	Los Angeles	19	
18	4	11.95	Los Angeles	19	
29	4	99.99	Boston	11	
...	...	...	...	...	
186248	9	3.84	Atlanta	20	
186259	9	11.99	Atlanta	0	
186260	9	5.98	Atlanta	0	
186296	9	600.00	San Francisco	13	
186297	9	11.95	San Francisco	13	

		Groupés
2		Google Phone,Wired Headphones
3		Google Phone,Wired Headphones
17		Google Phone,USB-C Charging Cable
18		Google Phone,USB-C Charging Cable
29	Bose SoundSport Headphones,Bose SoundSport Hea...	
...		...
186248	34in Ultrawide Monitor,AA Batteries (4-pack)	
186259	Wired Headphones,AAA Batteries (4-pack)	
186260	Wired Headphones,AAA Batteries (4-pack)	
186296	Google Phone,USB-C Charging Cable	
186297	Google Phone,USB-C Charging Cable	

[14649 rows x 11 columns]

```
[73]: df2 = df.drop_duplicates(subset=['ID Commande'])
```

```
[75]: df2
```

```
[75]:
```

	ID Commande	Produit	Quantité Commandée	\
2	176560	Google Phone	1	
17	176574	Google Phone	1	
29	176585	Bose SoundSport Headphones	1	
31	176586	AAA Batteries (4-pack)	2	
118	176672	Lightning Charging Cable	1	
...	...	...	...	
186237	259296	Apple AirPods Headphones	1	
186239	259297	iPhone	1	
186247	259303	34in Ultrawide Monitor	1	
186259	259314	Wired Headphones	1	
186296	259350	Google Phone	1	

	Prix Unitaire	Date de Commande	\
2	600.00	04/12/19 14:38	

17	600.00	04/03/19 19:42
29	99.99	04/07/19 11:31
31	2.99	04/10/19 17:00
118	14.95	04/12/19 11:07
...	...	...
186237	150.00	09/28/19 16:48
186239	700.00	09/15/19 18:54
186247	379.99	09/20/19 20:18
186259	11.99	09/16/19 00:25
186296	600.00	09/30/19 13:49

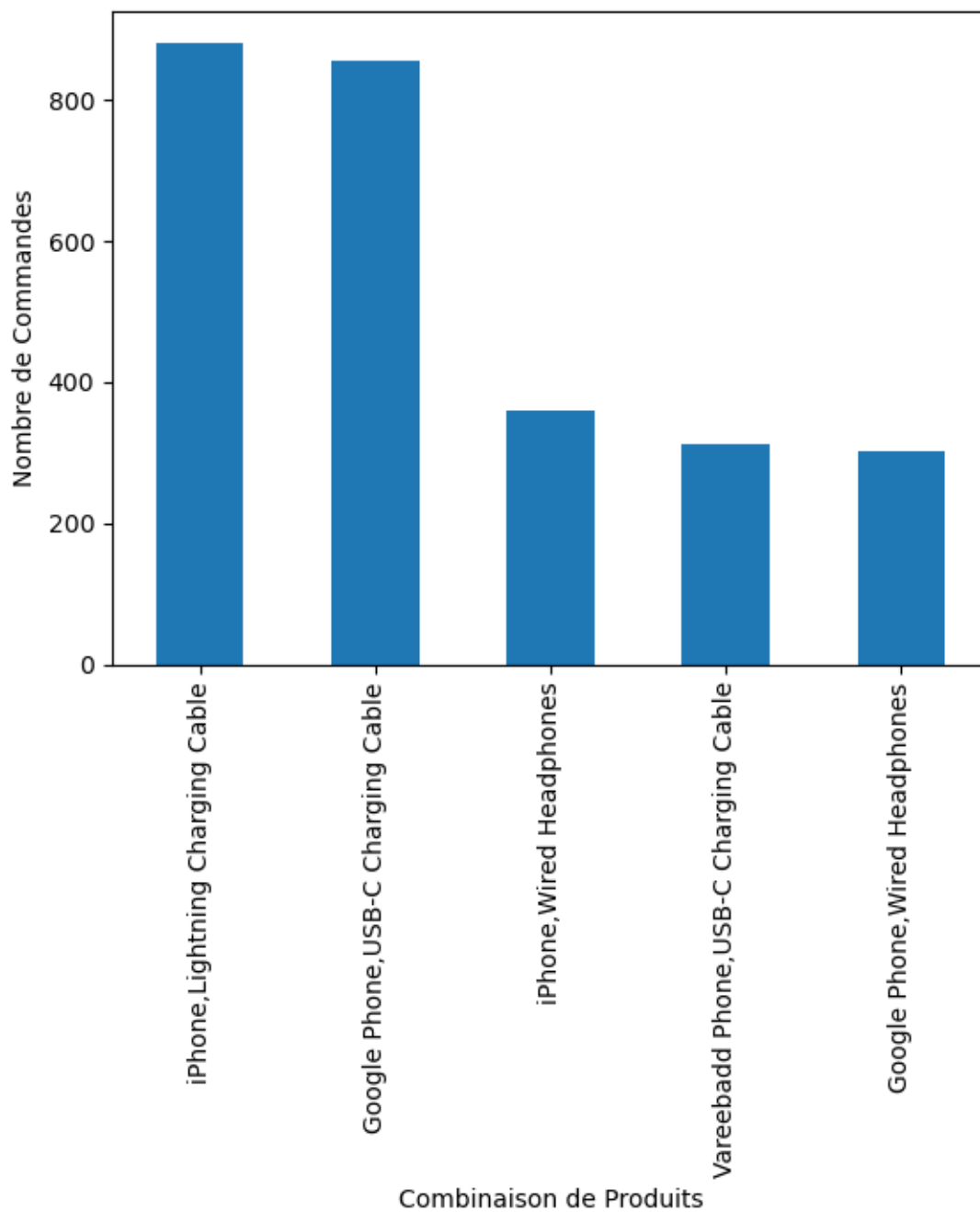
	Adresse d'Achat	Mois	Ventes	Ville \
2	669 Spruce St, Los Angeles, CA 90001	4	600.00	Los Angeles
17	20 Hill St, Los Angeles, CA 90001	4	600.00	Los Angeles
29	823 Highland St, Boston, MA 02215	4	99.99	Boston
31	365 Center St, San Francisco, CA 94016	4	5.98	San Francisco
118	778 Maple St, New York City, NY 10001	4	14.95	New York City
...	...	...	...	...
186237	894 6th St, Dallas, TX 75001	9	150.00	Dallas
186239	138 Main St, Boston, MA 02215	9	700.00	Boston
186247	106 7th St, Atlanta, GA 30301	9	379.99	Atlanta
186259	241 Highland St, Atlanta, GA 30301	9	11.99	Atlanta
186296	519 Maple St, San Francisco, CA 94016	9	600.00	San Francisco

	Heure	Groupés
2	14	Google Phone,Wired Headphones
17	19	Google Phone,USB-C Charging Cable
29	11	Bose SoundSport Headphones,Bose SoundSport Hea...
31	17	AAA Batteries (4-pack),Google Phone
118	11	Lightning Charging Cable,USB-C Charging Cable
...	...	...
186237	16	Apple AirPods Headphones,Apple AirPods Headphones
186239	18	iPhone,Lightning Charging Cable,Lightning Char...
186247	20	34in Ultrawide Monitor,AA Batteries (4-pack)
186259	0	Wired Headphones,AAA Batteries (4-pack)
186296	13	Google Phone,USB-C Charging Cable

[7136 rows x 11 columns]

```
[77]: df2['Groupés'].value_counts()[0:5].plot.bar()
plt.xlabel('Combinaison de Produits')
plt.ylabel('Nombre de Commandes')
```

```
[77]: Text(0, 0.5, 'Nombre de Commandes')
```



[ ]: