

# Project1\_\_Analysis

February 7, 2024

```
[1]: import pandas as pd #importer pandas
```

```
[43]: #chargement de dataset
df = pd.read_csv('./Sales Data.csv' , index_col=0)
df.head() #affichage des 5 premier lignes
```

```
[43]:
```

	Order ID	Product	Quantity Ordered	Price Each	\
0	295665	Macbook Pro Laptop	1	1700.00	
1	295666	LG Washing Machine	1	600.00	
2	295667	USB-C Charging Cable	1	11.95	
3	295668	27in FHD Monitor	1	149.99	
4	295669	USB-C Charging Cable	1	11.95	

	Order Date	Purchase Address	Month	\
0	2019-12-30 00:01:00	136 Church St, New York City, NY 10001	12	
1	2019-12-29 07:03:00	562 2nd St, New York City, NY 10001	12	
2	2019-12-12 18:21:00	277 Main St, New York City, NY 10001	12	
3	2019-12-22 15:13:00	410 6th St, San Francisco, CA 94016	12	
4	2019-12-18 12:38:00	43 Hill St, Atlanta, GA 30301	12	

	Sales	City	Hour
0	1700.00	New York City	0
1	600.00	New York City	7
2	11.95	New York City	18
3	149.99	San Francisco	15
4	11.95	Atlanta	12

```
[44]: df.tail() #affichage des 5 dernier lignes
```

```
[44]:
```

	Order ID	Product	Quantity Ordered	Price Each	\
13617	222905	AAA Batteries (4-pack)	1	2.99	
13618	222906	27in FHD Monitor	1	149.99	
13619	222907	USB-C Charging Cable	1	11.95	
13620	222908	USB-C Charging Cable	1	11.95	
13621	222909	AAA Batteries (4-pack)	1	2.99	

	Order Date	Purchase Address	Month	\
13617	2019-06-07 19:02:00	795 Pine St, Boston, MA 02215	6	

13618	2019-06-01 19:29:00	495 North St, New York City, NY 10001	6
13619	2019-06-22 18:57:00	319 Ridge St, San Francisco, CA 94016	6
13620	2019-06-26 18:35:00	916 Main St, San Francisco, CA 94016	6
13621	2019-06-25 14:33:00	209 11th St, Atlanta, GA 30301	6

	Sales	City	Hour
13617	2.99	Boston	19
13618	149.99	New York City	19
13619	11.95	San Francisco	18
13620	11.95	San Francisco	18
13621	2.99	Atlanta	14

```
[45]: df.dtypes #afficher les type des colonnes
```

```
[45]: Order ID          int64
      Product         object
      Quantity Ordered int64
      Price Each      float64
      Order Date      object
      Purchase Address object
      Month           int64
      Sales           float64
      City            object
      Hour            int64
      dtype: object
```

```
[46]: df.isnull().sum() #check if theres missing data #no missing data
      #dataframe.dropna() #on l'utilise pour remplacer les données manquants ppar nan
```

```
[46]: Order ID          0
      Product          0
      Quantity Ordered  0
      Price Each       0
      Order Date       0
      Purchase Address  0
      Month            0
      Sales            0
      City             0
      Hour             0
      dtype: int64
```

```
[47]: df.loc[:, 'Sales'].describe() #decrire le dataframe
```

```
[47]: count    185950.000000
      mean      185.490917
      std       332.919771
      min       2.990000
```

```

25%          11.950000
50%          14.950000
75%          150.000000
max          3400.000000
Name: Sales, dtype: float64

```

```
[48]: df.info() #affiche des info sur dataframe
```

```

<class 'pandas.core.frame.DataFrame'>
Index: 185950 entries, 0 to 13621
Data columns (total 10 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Order ID              185950 non-null  int64
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
6   Month                 185950 non-null  int64
7   Sales                 185950 non-null  float64
8   City                  185950 non-null  object
9   Hour                  185950 non-null  int64
dtypes: float64(2), int64(4), object(4)
memory usage: 15.6+ MB

```

```
[24]: df.shape #affiche les dimmension de dataframe
```

```
[24]: (185950, 10)
```

```

[28]: #let's add Month column
def month(x):
    return x.split('-')[1]

df['Month'] = df['Order Date'].apply(month)
df.head()

```

```

[28]:
   Order ID      Product  Quantity Ordered  Price Each  \
0   295665  Macbook Pro Laptop             1    1700.00
1   295666   LG Washing Machine             1     600.00
2   295667  USB-C Charging Cable             1     11.95
3   295668    27in FHD Monitor              1    149.99
4   295669  USB-C Charging Cable             1     11.95

      Order Date      Purchase Address  Month  Sales  \
0  2019-12-30 00:01:00  136 Church St, New York City, NY 10001    12  1700.00
1  2019-12-29 07:03:00    562 2nd St, New York City, NY 10001    12    600.00
2  2019-12-12 18:21:00   277 Main St, New York City, NY 10001    12     11.95

```

3	2019-12-22 15:13:00	410 6th St, San Francisco, CA 94016	12	149.99
4	2019-12-18 12:38:00	43 Hill St, Atlanta, GA 30301	12	11.95

	City	Hour
0	New York City	0
1	New York City	7
2	New York City	18
3	San Francisco	15
4	Atlanta	12

```
[31]: #verifier la colonne sales en la recalcullant
df['Quantity Ordered'] = df['Quantity Ordered'].astype(int)
df['Price Each'] = df['Price Each'].astype(float)

df['Sales'] = df['Quantity Ordered'] * df['Price Each']
```

```
[72]: #nombre de ventes par mois
pd.DataFrame(df.groupby('Month')['Sales'].sum())
```

```
[72]:
```

	Sales
Month	
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

```
[56]: pd.DataFrame(df.groupby('City')['City'].count()) #calculer le nombre des
      ↪commande par city
```

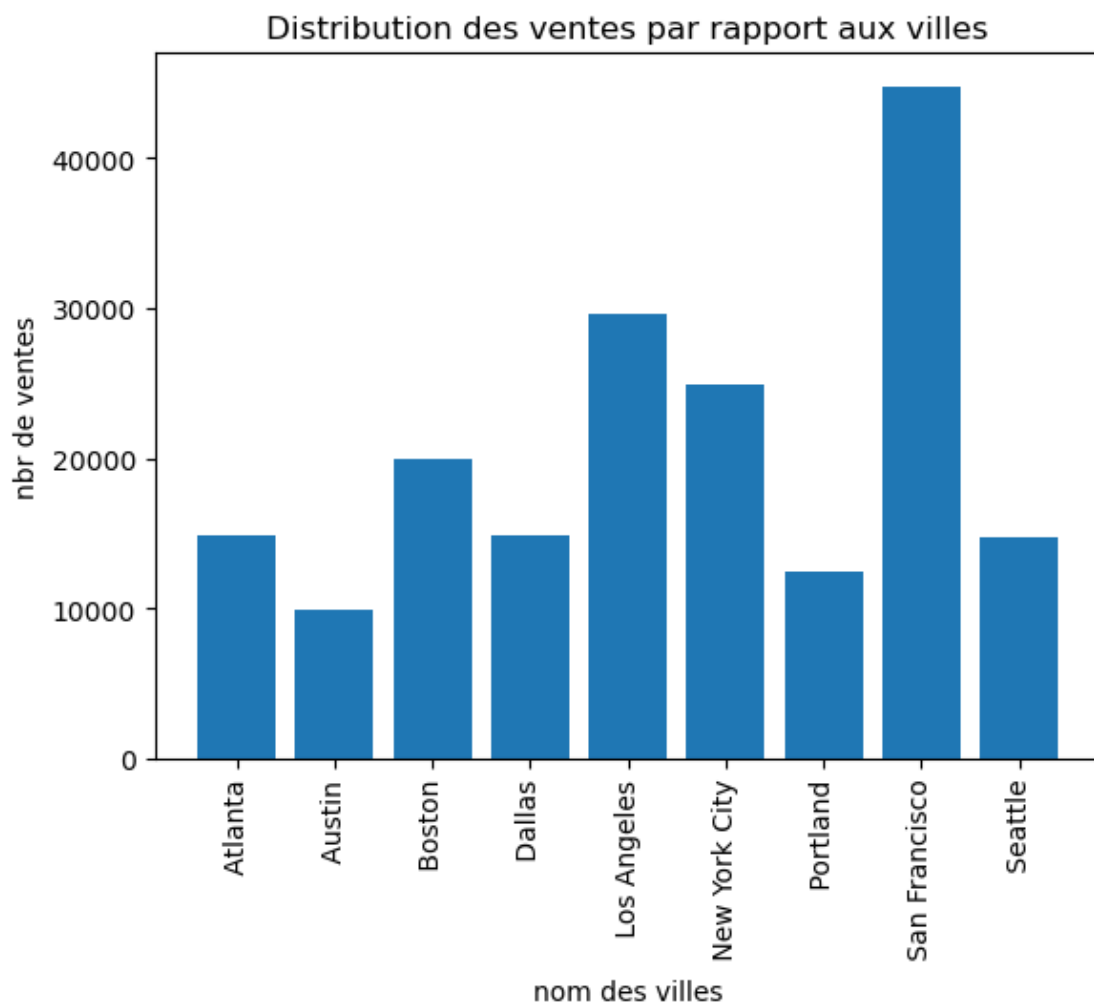
```
[56]:
```

	City
City	
Atlanta	14881
Austin	9905
Boston	19934
Dallas	14820
Los Angeles	29605
New York City	24876
Portland	12465
San Francisco	44732

Seattle 14732

```
[49]: import matplotlib.pyplot as plt #importer plot pour les graphique
```

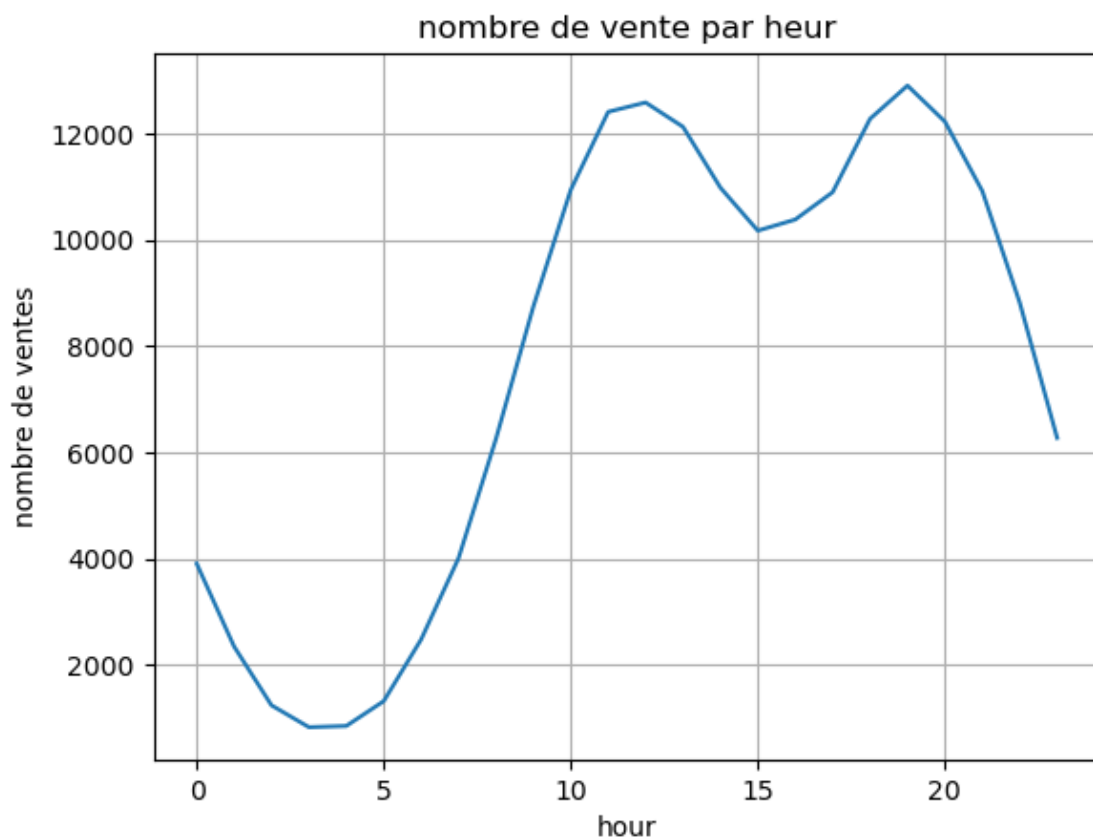
```
[73]: #illustrer Distribution des ventes par rapport aux villes
plt.bar(df.groupby('City')['City'].count().index , df.groupby('City')['City'].
        ↪count() )
plt.xticks(rotation='vertical')
plt.xlabel('nom des villes')
plt.ylabel('nbr de ventes')
plt.title('Distribution des ventes par rapport aux villes')
plt.show()
```



```
[58]: keys = []
      hours= []
```

```
#chaque heure dans key et le nombre de ventes convenable dans hours
for key, hour in df.groupby('Hour'):
    keys.append(key)
    hours.append(len(hour))
```

```
[61]: #illustrer nombre de vente par heure
plt.grid()
plt.plot(keys, hours)
plt.title('nombre de vente par heure')
plt.xlabel('hour')
plt.ylabel('nombre de ventes')
plt.show()
```



```
[71]: #moyenne des prix de chaque produit
pd.DataFrame(df.groupby('Product')['Price Each'].mean())
```

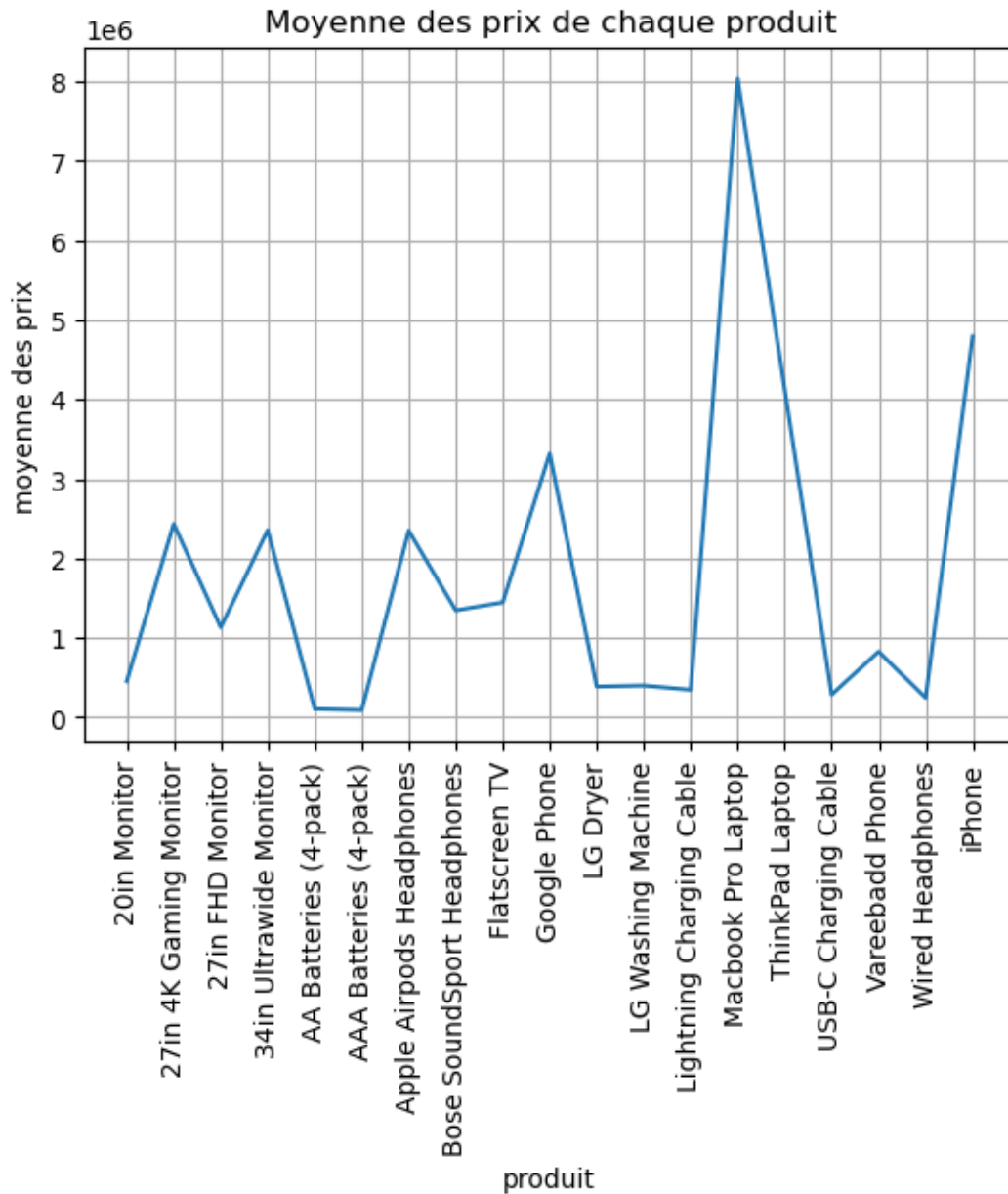
```
[71]:
```

	Price Each
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

```
[85]: #illustrer Moyenne des prix de chaque produit
prix = df.groupby('Product')['Price Each'].mean()
produit = df.groupby('Product')['Quantity Ordered'].sum().index
quantite = df.groupby('Product')['Quantity Ordered'].sum()

plt.plot(produit , prix*quantite)
plt.grid()
plt.xticks(rotation='vertical')
plt.title('Moyenne des prix de chaque produit')
plt.xlabel('produit')
plt.ylabel('moyenne des prix')
plt.show()
```



```
[83]: #illustrer Moyenne des prix de chaque produit
plt.figure()
fig, ax1 = plt.subplots()
ax2 = ax1.twinx()
ax1.bar(produit, quantite , color = 'g')
ax2.plot(produit, prix)
ax1.set_xticklabels(produit, rotation='vertical')
```

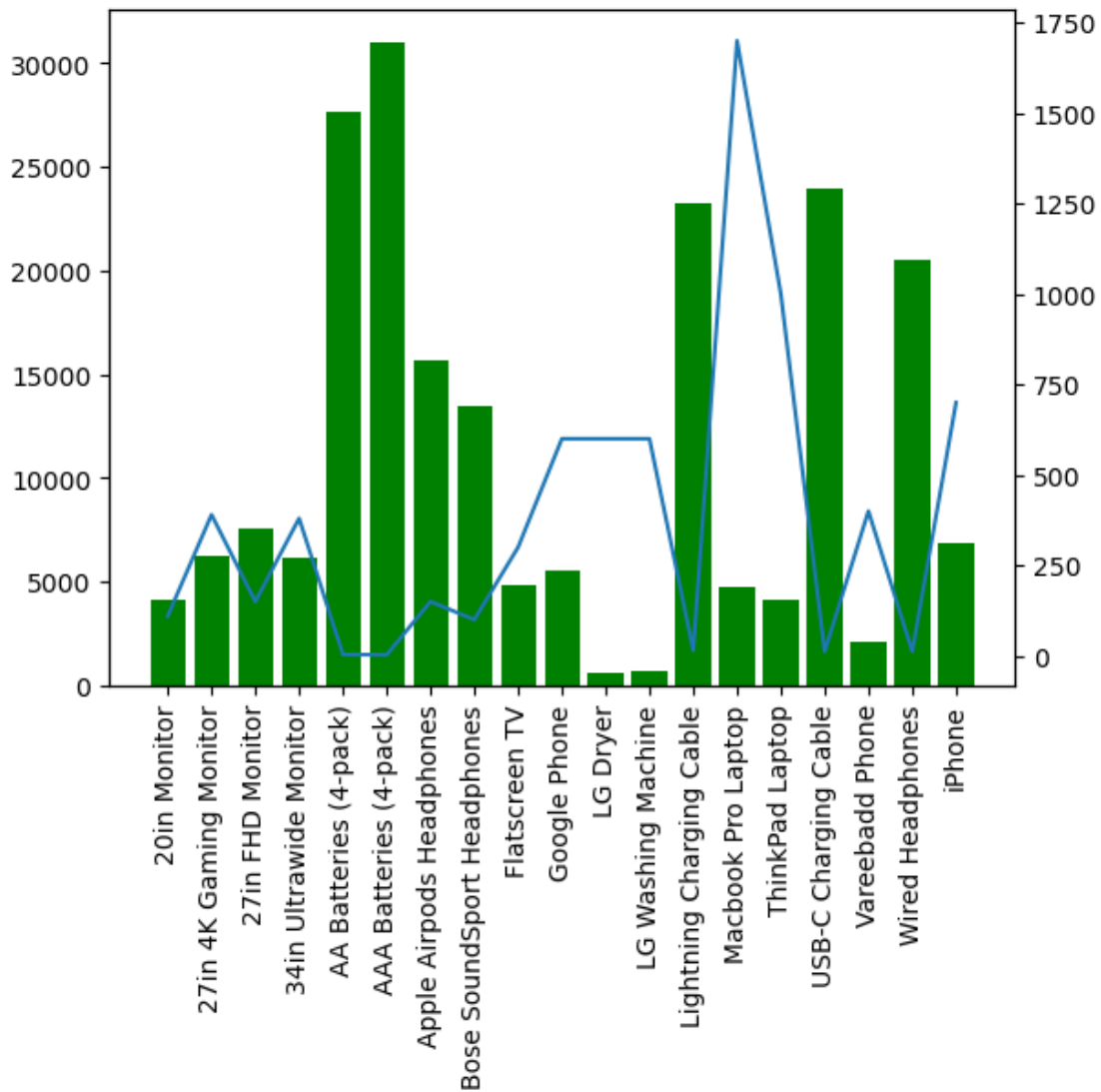
C:\Users\HP\AppData\Local\Temp\ipykernel\_21112\3166665475.py:6: UserWarning:  
FixedFormatter should only be used together with FixedLocator



```
ax1.set_xticklabels(produit, rotation='vertical')
```

```
[83]: [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 640x480 with 0 Axes>



```
[89]: df['commande'] = df.groupby('Order ID')['Product'].transform(lambda x : ",".
↪join(x))
```

```
[90]:
```

```
[90]:
```

	Order ID	Product	Quantity Ordered	Price Each \
0	295665	Macbook Pro Laptop	1	1700.00
1	295666	LG Washing Machine	1	600.00
2	295667	USB-C Charging Cable	1	11.95
3	295668	27in FHD Monitor	1	149.99
4	295669	USB-C Charging Cable	1	11.95

	Order Date	Purchase Address	Month \
--	------------	------------------	---------

0	2019-12-30 00:01:00	136 Church St, New York City, NY 10001	12
1	2019-12-29 07:03:00	562 2nd St, New York City, NY 10001	12
2	2019-12-12 18:21:00	277 Main St, New York City, NY 10001	12
3	2019-12-22 15:13:00	410 6th St, San Francisco, CA 94016	12
4	2019-12-18 12:38:00	43 Hill St, Atlanta, GA 30301	12

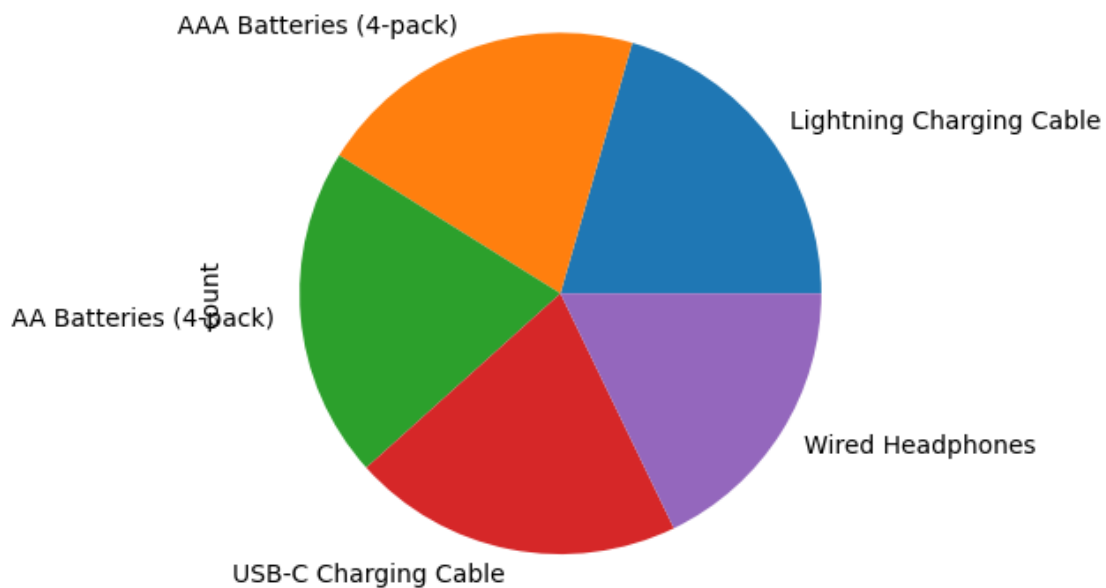
	Sales	City	Hour	commande
0	1700.00	New York City	0	Macbook Pro Laptop
1	600.00	New York City	7	LG Washing Machine
2	11.95	New York City	18	USB-C Charging Cable
3	149.99	San Francisco	15	27in FHD Monitor
4	11.95	Atlanta	12	USB-C Charging Cable

```
[100]: df['commande'].value_counts()[:5]
```

```
[100]: commande
Lightning Charging Cable    19831
AAA Batteries (4-pack)     19826
AA Batteries (4-pack)      19809
USB-C Charging Cable       19792
Wired Headphones           17208
Name: count, dtype: int64
```

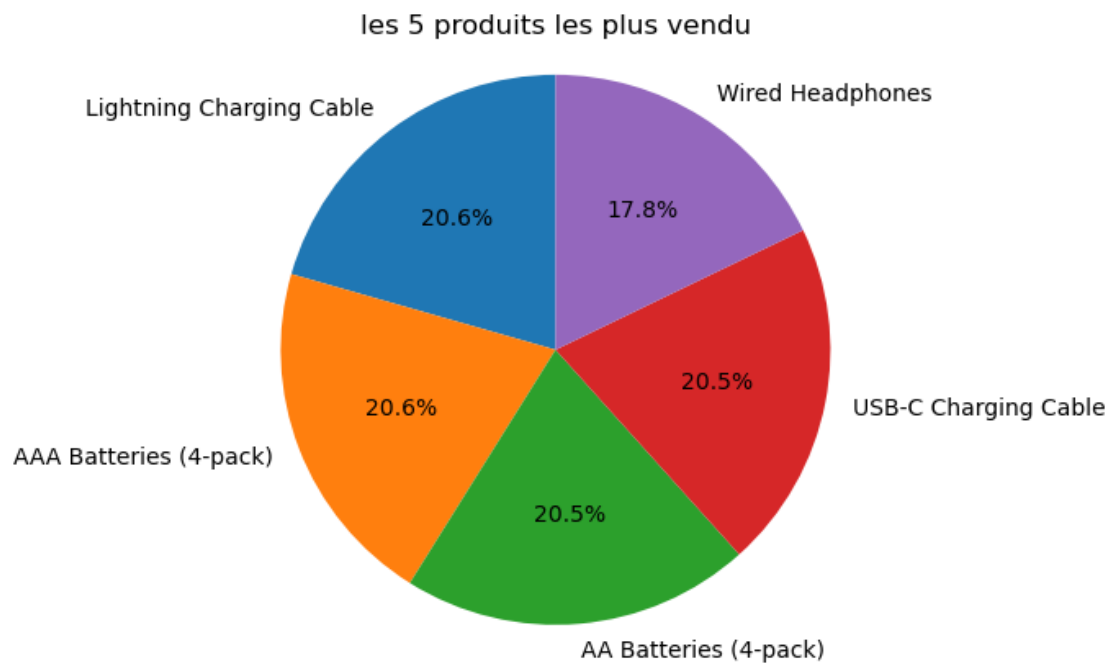
```
[97]: df['commande'].value_counts()[:5].plot.pie()
```

```
[97]: <Axes: ylabel='count'>
```



```
[103]: # Create a pie chart++
plt.pie(df['commande'].value_counts()[:5], labels=df['commande'].
        value_counts()[:5].index, autopct='%1.1f%%', startangle=90)

# Equal aspect ratio ensures that pie is drawn as a circle.
plt.axis('equal')
plt.title('les 5 produits les plus vendu')
plt.show()
```



```
[ ]:
```