In [1]:
```
#Bienvenue dans l'outil plébiscité par les analystes de données Jupyter
#Etape 1 - Importation des librairies et chargement des fichiers
#1-1-Importation de la librairie Pandas
import pandas as pd
```

In [2]:
```
pip install xlrd
```

```
Requirement already satisfied: xlrd in c:\users\pc\anaconda3\lib\site-packages (2.0.1)
Note: you may need to restart the kernel to use updated packages.
```

In [3]:
```
pip install openpyxl
```

```
Requirement already satisfied: openpyxl in c:\users\pc\anaconda3\lib\site-packages (3.1.2)Note: you may need to restart the kernel to use up
dated packages.

Requirement already satisfied: et-xmlfile in c:\users\pc\anaconda3\lib\site-packages (from openpyxl) (1.1.0)
```

In [4]:
```
#1.2 - Chargements des fichiers<
#Importation du fichier Web.xlsx\
df= pd.read_excel('web.xlsx')
```

```
C:\Users\pc\anaconda3\Lib\site-packages\openpyxl\worksheet\_reader.py:329: UserWarning: Unknown extension is not supported and will be remov
ed
  warn(msg)
```

In [5]:
```
!pip install openpyxl --upgrade
```

```
Requirement already satisfied: openpyxl in c:\users\pc\anaconda3\lib\site-packages (3.1.2)
Requirement already satisfied: et-xmlfile in c:\users\pc\anaconda3\lib\site-packages (from openpyxl) (1.1.0)
```

In [6]:
```
import warnings
```

In [7]:
```
warnings.filterwarnings('ignore')
```

In [8]:
```
df= pd.read_excel('web.xlsx')
```

In [9]:
```
df.head()
```

Out[9]:

	sku	virtual	downloadable	rating_count	average_rating	total_sales	tax_status	tax_class	post_author	post_date	...	post_name	post_modified	post_modified_gmt	post
0	bon-cadeau-25-euros	0	0	0	0.0	10.0	taxable	NaN	1.0	2018-06-01 13:53:46	...	bon-cadeau-de-25-euros	2018-06-01 14:13:57	2018-06-01 12:13:57	
1	15298	0	0	0	0.0	6.0	taxable	NaN	2.0	2018-02-08 12:58:52	...	pierre-jean-villa-saint-joseph-preface-2018	2019-12-30 09:30:29	2019-12-30 08:30:29	
2	15296	0	0	0	0.0	0.0	taxable	NaN	2.0	2018-02-08 13:49:41	...	pierre-jean-villa-saint-joseph-tilde-2017	2019-12-21 09:00:17	2019-12-21 08:00:17	
3	15300	0	0	0	0.0	0.0	taxable	NaN	2.0	2018-02-08 14:08:36	...	pierre-jean-villa-croze-hermitage-accroche-coe...	2020-06-26 18:15:03	2020-06-26 16:15:03	
4	19814	0	0	0	0.0	3.0	taxable	NaN	2.0	2018-02-09 14:01:05	...	pierre-jean-villa-igp-gamine-2018	2020-01-04 16:36:01	2020-01-04 15:36:01	

5 rows × 28 columns

In [10]:
```
#Importation du fichier erp.xlsx
erp_data = pd.read_excel('erp.xlsx')
```

In [11]:
```
print(erp_data)
```

```
     product_id  onsale_web  price  stock_quantity stock_status
0          3847           1   24.2               0   outofstock
1          3849           1   34.3               0   outofstock
2          3850           1   20.8               0   outofstock
3          4032           1   14.1               0   outofstock
4          4039           1   46.0               0   outofstock
..          ...         ...    ...             ...          ...
820        7203           0   45.0              30      instock
821        7204           0   45.0               9      instock
822        7247           1   54.8              23      instock
823        7329           0   26.5              14      instock
824        7338           1   16.3              45      instock

[825 rows x 5 columns]
```

In [12]:
```
#importation du fichier liaison.xlsx
df_liaison = pd.read_excel('C:/Users/pc/OneDrive/Documents/projet 6 Optimisez la gestion du stock d’une boutique en nettoyant ses données/lia
```

In [13]:
```
df_liaison.head()
```

Out[13]:

	product_id	id_web
0	3847	15298
1	3849	15296
2	3850	15300
3	4032	19814
4	4039	19815

In [14]:
```
#2.1 - Analyse exploratoire du fichier erp.xlsx
erp_data = erp_data.drop_duplicates()
```

In [15]:
```
# Gérer les valeurs manquantes dans la colonne 'price'
erp_data['price'] = erp_data['price'].fillna(0)
```

In [16]:
```
erp_data = erp_data[erp_data['price'] >= 0]
```

In [17]:
```
erp_data.tail()
```

Out[17]:

	product_id	onsale_web	price	stock_quantity	stock_status
820	7203	0	45.0	30	instock
821	7204	0	45.0	9	instock
822	7247	1	54.8	23	instock
823	7329	0	26.5	14	instock
824	7338	1	16.3	45	instock

In [18]:
```
#Importation de la bibilothèque chardet
import chardet
```

In [19]:
```
##Afficher les dimensions du dataset
print("Le tableau comporte {} observation(s) ou article(s)".format(erp_data.shape[0]))
```

```
Le tableau comporte 825 observation(s) ou article(s)
```

In [20]:
```
#1.2 - Chargements du fichiers csv
#Identification de l'encodage du fichier
df_caracteristiques = pd.read_csv('C:/Users/pc/OneDrive/Documents/projet 6 Optimisez la gestion du stock d’une boutique en nettoyant ses donn
```

In [21]:
```
print(df_caracteristiques.head())
```

```
                                      post_name   poids Région  \
0       pierre-jean-villa-saint-joseph-preface-2018  1.5 kg  Rhône
1         pierre-jean-villa-saint-joseph-tilde-2017  1.5 kg  Rhône
2  pierre-jean-villa-croze-hermitage-accroche-coe...  1.5 kg  Rhône
3                 pierre-jean-villa-igp-gamine-2018  1.5 kg  Rhône
4          pierre-jean-villa-cote-rotie-carmina-2017  1.5 kg  Rhône

             Domaine            Appellation Couleur      Cépage  Millésime  \
0  Pierre Jean Villa           Saint Joseph   Rouge  100% Syrah     2020.0
1  Pierre Jean Villa           Saint Joseph   Rouge  100% Syrah     2019.0
2  Pierre Jean Villa       Crozes-Hermitage   Rouge  100% Syrah     2020.0
3  Pierre Jean Villa  Collines Rhodaniennes   Rouge  100% Syrah     2020.0
4  Pierre Jean Villa             Côte Rôtie   Rouge  100% Syrah     2019.0

       Garde Contenance Degré d'alcool Température dégustation  \
0    4-7 ans       75cl           13%                    15°C
1    6-8 ans       75cl           13%                    15°C
2    3-5 ans       75cl           13%                    15°C
3    3-5 ans       75cl           13%                    14°C
4  10-20 ans       75cl           13%                    17°C

                            Alliance mets
0  Charcuterie, Lapin, Viande rouge, Volaille
1        Charcuterie, Viande rouge, Volaille
2                    Viande rouge, Volaille
3        Charcuterie, Viande rouge, Volaille
4                      Gibier, Viande rouge
```


In [22]:
```
df_caracteristiques.head()
```

Out[22]:

	post_name	poids	Région	Domaine	Appellation	Couleur	Cépage	Millésime	Garde	Contenance	Degré d'alcool	Température dégustation	Alliance mets
0	pierre-jean-villa-saint-joseph-preface-2018	1.5 kg	Rhône	Pierre Jean Villa	Saint Joseph	Rouge	100% Syrah	2020.0	4-7 ans	75cl	13%	15°C	Charcuterie, Lapin, Viande rouge, Volaille
1	pierre-jean-villa-saint-joseph-tilde-2017	1.5 kg	Rhône	Pierre Jean Villa	Saint Joseph	Rouge	100% Syrah	2019.0	6-8 ans	75cl	13%	15°C	Charcuterie, Viande rouge, Volaille
2	pierre-jean-villa-croze-hermitage-accroche-coe...	1.5 kg	Rhône	Pierre Jean Villa	Crozes-Hermitage	Rouge	100% Syrah	2020.0	3-5 ans	75cl	13%	15°C	Viande rouge, Volaille
3	pierre-jean-villa-igp-gamine-2018	1.5 kg	Rhône	Pierre Jean Villa	Collines Rhodaniennes	Rouge	100% Syrah	2020.0	3-5 ans	75cl	13%	14°C	Charcuterie, Viande rouge, Volaille
4	pierre-jean-villa-cote-rotie-carmina-2017	1.5 kg	Rhône	Pierre Jean Villa	Côte Rôtie	Rouge	100% Syrah	2019.0	10-20 ans	75cl	13%	17°C	Gibier, Viande rouge

In [23]:
```
#Consulter le nombre de colonnes
num_colonnes = erp_data.shape[1]
```

```
print("Le DataFrame a {} colonnes.".format(num_colonnes))
```

```
Le DataFrame a 5 colonnes.
```

In [24]:
```
#La nature des données dans chacune des colonnes
nature_des_donnees = erp_data.dtypes
print("Nature des données dans chaque colonne :")
print(nature_des_donnees)
```

```
Nature des données dans chaque colonne :
product_id          int64
onsale_web          int64
price             float64
stock_quantity      int64
stock_status       object
dtype: object
```

In [25]:
```
#Le nombre de valeurs présentes dans chacune des colonnes
nb_valeurs_non_nulles = erp_data.count()
print("Nombre de valeurs non nulles dans chaque colonne :")
print(nb_valeurs_non_nulles)
```

```
Nombre de valeurs non nulles dans chaque colonne :
product_id        825
onsale_web        825
price             825
stock_quantity    825
stock_status      825
dtype: int64
```

In [26]:
```
#Afficher les 5 premières lignes de la table\
erp_data.head(5)
```

Out[26]:

	product_id	onsale_web	price	stock_quantity	stock_status
0	3847	1	24.2	0	outofstock
1	3849	1	34.3	0	outofstock
2	3850	1	20.8	0	outofstock
3	4032	1	14.1	0	outofstock
4	4039	1	46.0	0	outofstock

In [27]:
```
#Vérification si il y a les lignes en doublons dans la colonne product_id\
doublons = erp_data.duplicated(subset=['product_id'])
```

In [28]:
```
lignes_en_doublons = erp_data.loc[doublons]
```

In [29]:
```
erp_data = erp_data.drop_duplicates(subset=['product_id'], keep='first')
```

In [30]:
```
#À quelle(s) autre(s) colonne(s) sont-elles liées : stock_quantity
#Affichacge :les valeurs distinctes de la colonne stock_status
valeurs_distinctes = erp_data['stock_status'].unique()
print(valeurs_distinctes)
```

```
['outofstock' 'instock']
```

In [31]:
```
#Création d'une colonne \"stock_status_2
erp_data['stock_status_2'] = erp_data['stock_quantity'].apply(lambda x: 'outofstock' if x == 0 else 'instock')
```

In [32]:
```
erp_data.head()
```


	product_id	onsale_web	price	stock_quantity	stock_status	stock_status_2
0	3847	1	24.2	0	outofstock	outofstock
1	3849	1	34.3	0	outofstock	outofstock
2	3850	1	20.8	0	outofstock	outofstock
3	4032	1	14.1	0	outofstock	outofstock
4	4039	1	46.0	0	outofstock	outofstock

In [33]:
```
#Vérifions que les 2 colonnes sont identiques:
identical = erp_data["stock_status"] == erp_data["stock_status_2"]
```

In [34]:
```
differences = erp_data[identical == False]
```

In [35]:
```
identical_count = identical.sum()
```

In [36]:
```
different_count = len(identical) - identical_count
```

In [37]:
```
print(f"Nombre de lignes où les colonnes sont identiques : {identical_count}")
print(f"Nombre de lignes où les colonnes diffèrent : {different_count}")
```

```
Nombre de lignes où les colonnes sont identiques : 824
Nombre de lignes où les colonnes diffèrent : 1
```

In [38]:
```
#Si les colonnes ne sont absolument pas identiques ligne à ligne alors identifier la ligne en écart
diff_filter = erp_data["stock_status"] != erp_data["stock_status_2"]
```

In [39]:
```
differences = erp_data[diff_filter]
```

In [40]:
```
print(differences)
```

```
     product_id  onsale_web  price  stock_quantity stock_status stock_status_2
443        4954           1   25.0               0      instock     outofstock
```

In [41]:
```
#Corriger la ou les données incohérentes\
erp_data.loc[diff_filter, "stock_status"] = erp_data.loc[diff_filter, "stock_status_2"]
```

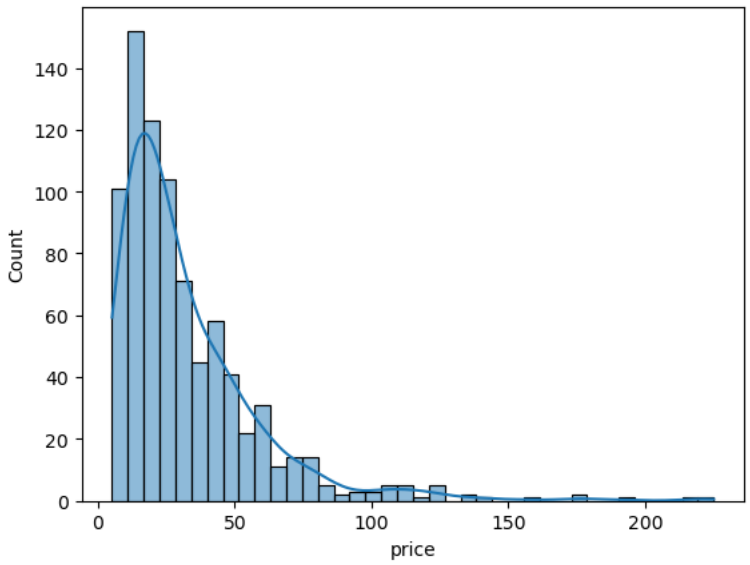
In [42]:
```
erp_data["stock_status_2"] = erp_data["stock_status"]
```

In [43]:
```
#2.1.1 - Analyse exploratoire de chaque variable du fichier erp.xlsx
#Importation de la librairie plotly express
import matplotlib.pyplot as plt
```

In [44]:
```
import seaborn as sns
```

In [45]:
```
sns.histplot(erp_data['price'], kde=True)
```

Out[45]:
```
<Axes: xlabel='price', ylabel='Count'>
```



In [46]:
```
plt.show()
```

In [48]:
```
#2.1.1.1 - Analyse de la variable PRIX
#Vérification des prix: Y a t-il des prix non renseignés
prix_non_renseignes = erp_data[erp_data['price'].isnull()]
```

In [49]:
```
#Vérification des prix: Y a t-il des prix négatif ou nul
prix_negatifs_ou_nuls = erp_data[(erp_data['price'] <= 0)]
```

In [50]:
```
#ffficher le ou les prix négatif ou nul
print("Nombres d'articles avec un prix négatif: {}".format(len(prix_negatifs_ou_nuls)))
```

```
Nombres d'articles avec un prix négatif: 0
```


In [51]:
```
ou les prix non renseignés dans la colonne \"price\
print("Nombres d'articles avec un prix non renseignés: {}".format(len(prix_non_renseignes)))
```

```
Nombres d'articles avec un prix non renseignés: 0
```

In [52]:
```
#Afficher le prix minimum de la colonne \"price
prix_minimum = erp_data['price'].min()
print("Prix minimum :", prix_minimum)
```

```
Prix minimum : 5.2
```

In [53]:
```
#Afficher le prix maximum de la colonne \"price
prix_maximum = erp_data['price'].max()
```

In [54]:
```
print("Prix maximum :", prix_maximum)
```

```
Prix maximum : 225.0
```

In [55]:
```
#2.1.1.2 - Analyse de la variable STOCK<
#Vérification de la colonne stock quantity
distinct_stock_status = erp_data['stock_status'].unique()
```

In [56]:
```
print(distinct_stock_status)
```

```
['outofstock' 'instock']
```

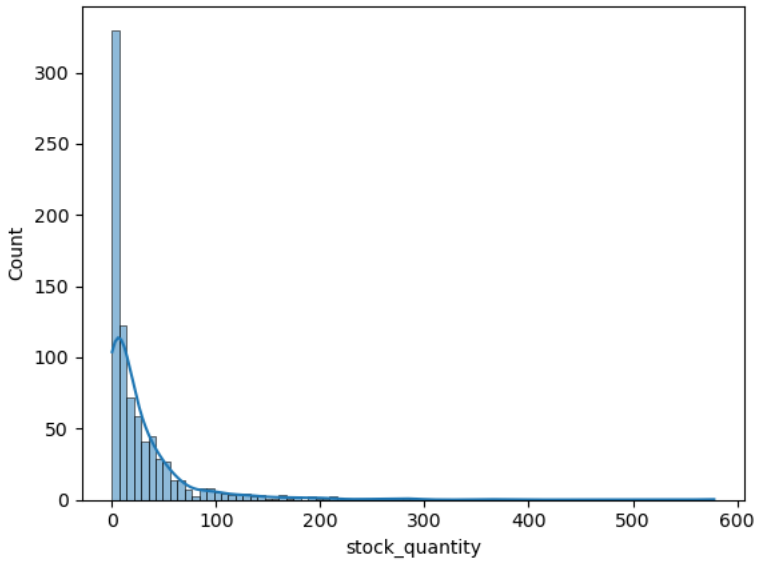
In [57]:
```
stock_quantity_stats = erp_data['stock_quantity'].describe()
```

In [58]:
```
print(stock_quantity_stats)
```

```
count    825.000000
mean      26.496970
std       45.892422
min        0.000000
25%        1.000000
50%       11.000000
75%       34.000000
max      578.000000
Name: stock_quantity, dtype: float64
```

In [59]:
```
sns.histplot(erp_data['stock_quantity'], kde=True)
```

Out[59]:
```
<Axes: xlabel='stock_quantity', ylabel='Count'>
```



In [60]:
```
#Afficher la quantité minimum de la colonne
stock_quantity_min = erp_data['stock_quantity'].min()
print("Quantité minimum de stock_quantity :", stock_quantity_min)
```

```
Quantité minimum de stock_quantity : 0
```

In [61]:
```
#Afficher la quantité maximum de la colonne
stock_quantity_max = erp_data['stock_quantity'].max()
print("Quantité maximum de stock_quantity :", stock_quantity_max)
```

```
Quantité maximum de stock_quantity : 578
```

In [62]:
```
#2.1.1.3 - Analyse de la variable ONSALE_WEB
#Vérification de la colonne onsale_web et des valeurs qu'elle contient
onsale_web_counts = erp_data['onsale_web'].value_counts()
```

In [63]:
```
print(onsale_web_counts)
```

```
1    717
0    108
Name: onsale_web, dtype: int64
```

In [64]:
```
#Supprimer les colonnes comportant le libellé \"stock_status
p_data.drop(columns=['stock_status', 'stock_status_2'])
```


In [65]:
```
erp_data.head()
```

Out[65]:

	product_id	onsale_web	price	stock_quantity
0	3847	1	24.2	0
1	3849	1	34.3	0
2	3850	1	20.8	0
3	4032	1	14.1	0
4	4039	1	46.0	0

In [66]:
```
#2.2 - Analyse exploratoire du fichier web.xlsx</h3
df.head()
```

Out[66]:

	sku	virtual	downloadable	rating_count	average_rating	total_sales	tax_status	tax_class	post_author	post_date	...	post_name	post_modified	post_modified_gmt	post
0	bon-cadeau-25-euros	0	0	0	0.0	10.0	taxable	NaN	1.0	2018-06-01 13:53:46	...	bon-cadeau-de-25-euros	2018-06-01 14:13:57	2018-06-01 12:13:57	
1	15298	0	0	0	0.0	6.0	taxable	NaN	2.0	2018-02-08 12:58:52	...	pierre-jean-villa-saint-joseph-preface-2018	2019-12-30 09:30:29	2019-12-30 08:30:29	
2	15296	0	0	0	0.0	0.0	taxable	NaN	2.0	2018-02-08 13:49:41	...	pierre-jean-villa-saint-joseph-tilde-2017	2019-12-21 09:00:17	2019-12-21 08:00:17	
3	15300	0	0	0	0.0	0.0	taxable	NaN	2.0	2018-02-08 14:08:36	...	pierre-jean-villa-croze-hermitage-accroche-coe...	2020-06-26 18:15:03	2020-06-26 16:15:03	
4	19814	0	0	0	0.0	3.0	taxable	NaN	2.0	2018-02-09 14:01:05	...	pierre-jean-villa-igp-gamine-2018	2020-01-04 16:36:01	2020-01-04 15:36:01	

5 rows × 28 columns

In [67]:
```
#Dimension du dataset ,Nombre d'observations
print("Le tableau comporte {} observation(s) ou article(s)".format(df.shape[0]))
```

```
Le tableau comporte 1513 observation(s) ou article(s)
```

In [68]:
```
#Nombre de caractéristiques
print("Le tableau comporte {} colonne(s)".format(df.shape[1]))
```

```
Le tableau comporte 28 colonne(s)
```

In [69]:
```
df = df.drop_duplicates()
```

In [70]:
```
#La nature des données dans chacune des colonnes
df.describe()
```

Out[70]:

	virtual	downloadable	rating_count	average_rating	total_sales	tax_class	post_author	post_content	post_password	post_content_filtered	post_parent	menu_order	co
count	1431.0	1431.0	1431.0	1430.0	1430.000000	0.0	1430.000000	0.0	0.0	0.0	1430.0	1430.0	
mean	0.0	0.0	0.0	0.0	4.006993	NaN	1.998601	NaN	NaN	NaN	0.0	0.0	
std	0.0	0.0	0.0	0.0	8.510559	NaN	0.037385	NaN	NaN	NaN	0.0	0.0	
min	0.0	0.0	0.0	0.0	0.000000	NaN	1.000000	NaN	NaN	NaN	0.0	0.0	
25%	0.0	0.0	0.0	0.0	0.000000	NaN	2.000000	NaN	NaN	NaN	0.0	0.0	
50%	0.0	0.0	0.0	0.0	1.000000	NaN	2.000000	NaN	NaN	NaN	0.0	0.0	
75%	0.0	0.0	0.0	0.0	4.000000	NaN	2.000000	NaN	NaN	NaN	0.0	0.0	
max	0.0	0.0	0.0	0.0	96.000000	NaN	2.000000	NaN	NaN	NaN	0.0	0.0	

In [71]:
```
#Le nombre de valeurs présentes dans chacune des colonnes
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1431 entries, 0 to 1512
Data columns (total 28 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   sku                   1428 non-null   object
 1   virtual               1431 non-null   int64
 2   downloadable          1431 non-null   int64
 3   rating_count          1431 non-null   int64
 4   average_rating        1430 non-null   float64
 5   total_sales           1430 non-null   float64
 6   tax_status            716 non-null    object
 7   tax_class             0 non-null      float64
 8   post_author           1430 non-null   float64
 9   post_date             1430 non-null   datetime64[ns]
 10  post_date_gmt         1430 non-null   datetime64[ns]
 11  post_content          0 non-null      float64
 12  post_title            1430 non-null   object
 13  post_excerpt          716 non-null    object
 14  post_status           1430 non-null   object
 15  comment_status        1430 non-null   object
 16  ping_status           1430 non-null   object
 17  post_password         0 non-null      float64
 18  post_name             1430 non-null   object
 19  post_modified         1430 non-null   datetime64[ns]
 20  post_modified_gmt     1430 non-null   datetime64[ns]
 21  post_content_filtered 0 non-null      float64
 22  post_parent           1430 non-null   float64
 23  guid                  1430 non-null   object
 24  menu_order            1430 non-null   float64
 25  post_type             1430 non-null   object
 26  post_mime_type        714 non-null    object
 27  comment_count         1430 non-null   float64
dtypes: datetime64[ns](4), float64(10), int64(3), object(11)
memory usage: 324.2+ KB
```

In [72]:
```
nombre_valeurs_non_nulles = df.count()
```

In [73]:
```
print(nombre_valeurs_non_nulles)
```

```
sku                     1428
virtual                 1431
downloadable            1431
rating_count            1431
average_rating          1430
total_sales             1430
tax_status               716
tax_class                  0
post_author             1430
post_date               1430
post_date_gmt           1430
post_content               0
post_title              1430
post_excerpt             716
post_status             1430
comment_status          1430
ping_status             1430
post_password              0
post_name               1430
post_modified           1430
post_modified_gmt       1430
post_content_filtered      0
post_parent             1430
guid                    1430
menu_order              1430
post_type               1430
post_mime_type           714
comment_count           1430
dtype: int64
```

In [74]:
```
#Les colonnes à supprimer, effectuer l'opération\
colonnes_a_supprimer = ['post_mime_type', 'comment_count', 'ping_status', 'tax_class', 'post_content_filtered']
```

In [75]:
```
df = df.drop(colonnes_a_supprimer, axis=1)
```

In [76]:
```
df.head()
```

Out[76]:

	sku	virtual	downloadable	rating_count	average_rating	total_sales	tax_status	post_author	post_date	post_date_gmt	...	post_status	comment_status	post_password
0	bon-cadeau-25-euros	0	0	0	0.0	10.0	taxable	1.0	2018-06-01 13:53:46	2018-06-01 11:53:46	...	publish	closed	NaN
1	15298	0	0	0	0.0	6.0	taxable	2.0	2018-02-08 12:58:52	2018-02-08 11:58:52	...	publish	closed	NaN
2	15296	0	0	0	0.0	0.0	taxable	2.0	2018-02-08 13:49:41	2018-02-08 12:49:41	...	publish	closed	NaN
3	15300	0	0	0	0.0	0.0	taxable	2.0	2018-02-08 14:08:36	2018-02-08 13:08:36	...	publish	closed	NaN
		0	0	0	0.0	3.0	taxable	2.0	2018-02-09 14:01:05	2018-02-09 13:01:05	...	publish	closed	NaN

5 rows × 23 columns

In [77]:
```
#Visualisation des valeurs de la colonne sku
sku_counts = df['sku'].value_counts()
```

In [78]:
```
print(sku_counts)
```

```
bon-cadeau-25-euros    2
14828                  2
14679                  2
15526                  2
16305                  2
                      ..
15145                  2
15801                  2
15452                  2
15038                  2
16230                  2
Name: sku, Length: 714, dtype: int64
```

In [79]:
```
df['sku'] = df['sku'].astype(str)
```

In [80]:
```
# les valeurs qui ne semblent pas respecter la régle de codification
non_conformes = df[~df['sku'].str.match(r'SKU_\d+')]
```

In [81]:
```
print(non_conformes)
```

```
                      sku  virtual  downloadable  rating_count  \
0     bon-cadeau-25-euros        0             0             0
1                   15298        0             0             0
2                   15296        0             0             0
3                   15300        0             0             0
4                   19814        0             0             0
...                   ...      ...           ...           ...
1508                16135        0             0             0
1509                15891        0             0             0
1510                15887        0             0             0
1511              13127-1        0             0             0
1512                16230        0             0             0

      average_rating  total_sales tax_status  post_author           post_date  \
0                0.0         10.0    taxable          1.0 2018-06-01 13:53:46
1                0.0          6.0    taxable          2.0 2018-02-08 12:58:52
2                0.0          0.0    taxable          2.0 2018-02-08 13:49:41
3                0.0          0.0    taxable          2.0 2018-02-08 14:08:36
4                0.0          3.0    taxable          2.0 2018-02-09 14:01:05
...              ...          ...        ...          ...                 ...
1508             0.0          5.0        NaN          2.0 2020-04-25 13:22:38
1509             0.0          0.0        NaN          2.0 2020-05-02 14:53:40
1510             0.0          0.0        NaN          2.0 2020-05-02 15:00:54
1511             0.0          0.0        NaN          2.0 2020-06-09 15:42:04
1512             0.0          0.0        NaN          2.0 2020-07-20 11:00:00

           post_date_gmt  ...  post_status comment_status post_password  \
0    2018-06-01 11:53:46  ...      publish         closed           NaN
1    2018-02-08 11:58:52  ...      publish         closed           NaN
2    2018-02-08 12:49:41  ...      publish         closed           NaN
3    2018-02-08 13:08:36  ...      publish         closed           NaN
4    2018-02-09 13:01:05  ...      publish         closed           NaN
...                  ...  ...          ...            ...           ...
1508 2020-04-25 11:22:38  ...      publish         closed           NaN
1509 2020-05-02 12:53:40  ...      publish         closed           NaN
1510 2020-05-02 13:00:54  ...      publish         closed           NaN
1511 2020-06-09 13:42:04  ...      publish         closed           NaN
1512 2020-07-20 09:00:00  ...      publish         closed           NaN

                                          post_name       post_modified  \
0                           bon-cadeau-de-25-euros 2018-06-01 14:13:57
1     pierre-jean-villa-saint-joseph-preface-2018 2019-12-30 09:30:29
2       pierre-jean-villa-saint-joseph-tilde-2017 2019-12-21 09:00:17
3    pierre-jean-villa-croze-hermitage-accroche-coe... 2020-06-26 18:15:03
4              pierre-jean-villa-igp-gamine-2018 2020-01-04 16:36:01
...                                          ...                 ...
1508             mouthes-le-bihan-aime-chai-2015 2020-08-26 17:35:03
1509   camin-larredya-jurancon-sec-la-virada-2018 2020-08-26 17:35:02
1510       jamet-cote-rotie-fructus-voluptas-2018 2020-08-14 18:15:03
1511  clos-du-mont-olivet-chateauneuf-du-pape-2007-2 2020-07-20 17:09:06
1512  domaine-saint-nicolas-fiefs-vendeens-blanc-les... 2020-08-13 10:45:03

       post_modified_gmt post_parent  \
0    2018-06-01 12:13:57         0.0
1    2019-12-30 08:30:29         0.0
2    2019-12-21 08:00:17         0.0
3    2020-06-26 16:15:03         0.0
4    2020-01-04 15:36:01         0.0
...                  ...         ...
1508 2020-08-26 15:35:03         0.0
1509 2020-08-26 15:35:02         0.0
1510 2020-08-14 16:15:03         0.0
1511 2020-07-20 15:09:06         0.0
1512 2020-08-13 08:45:03         0.0

                                                  guid menu_order   post_type
0     https://www.bottle-neck.fr/?post_type=product&...        0.0     product
1     https://www.bottle-neck.fr/?post_type=product&...        0.0     product
2     https://www.bottle-neck.fr/?post_type=product&...        0.0     product
3     https://www.bottle-neck.fr/?post_type=product&...        0.0     product
4     https://www.bottle-neck.fr/?post_type=product&...        0.0     product
...                                                 ...        ...         ...
1508  https://www.bottle-neck.fr/wp-content/uploads/...        0.0  attachment
1509  https://www.bottle-neck.fr/wp-content/uploads/...        0.0  attachment
1510  https://www.bottle-neck.fr/wp-content/uploads/...        0.0  attachment
1511  https://www.bottle-neck.fr/wp-content/uploads/...        0.0  attachment
1512  https://www.bottle-neck.fr/wp-content/uploads/...        0.0  attachment

[1431 rows x 23 columns]
```

Loading [MathJax]/extensions/Safe.js

In [82]:
```
non_conforme = df[~df['sku'].str.match(r'^\d+')]
```

In [83]:
```
print(non_conforme)
```

```
                     sku  virtual  downloadable  rating_count  \
0     bon-cadeau-25-euros        0             0             0
178                   nan        0             0             0
470                   nan        0             0             0
471                   nan        0             0             0
1209  bon-cadeau-25-euros        0             0             0

      average_rating  total_sales tax_status  post_author           post_date  \
0                0.0         10.0    taxable          1.0 2018-06-01 13:53:46
178              NaN          NaN        NaN          NaN                 NaT
470              0.0          0.0    taxable          2.0 2018-07-31 12:07:23
471              0.0          0.0    taxable          2.0 2018-08-08 11:23:43
1209             0.0         10.0        NaN          1.0 2018-06-01 13:53:46

           post_date_gmt  ...  post_status comment_status post_password  \
0    2018-06-01 11:53:46  ...      publish         closed           NaN
178                  NaT  ...          NaN            NaN           NaN
470  2018-07-31 10:07:23  ...      publish         closed           NaN
471  2018-08-08 09:23:43  ...      publish         closed           NaN
1209 2018-06-01 11:53:46  ...      publish         closed           NaN

                                    post_name       post_modified  \
0                      bon-cadeau-de-25-euros 2018-06-01 14:13:57
178                                       NaN                 NaT
470   pierre-jean-villa-cote-rotie-fongeant-2017 2019-11-02 13:24:15
471     pierre-jean-villa-condrieu-suspendu-2018 2019-11-02 13:24:01
1209                   bon-cadeau-de-25-euros 2018-06-01 14:13:57

       post_modified_gmt post_parent  \
0    2018-06-01 12:13:57         0.0
178                  NaT         NaN
470  2019-11-02 12:24:15         0.0
471  2019-11-02 12:24:01         0.0
1209 2018-06-01 12:13:57         0.0

                                              guid menu_order   post_type
0     https://www.bottle-neck.fr/?post_type=product&...        0.0     product
178                                             NaN        NaN         NaN
470   https://www.bottle-neck.fr/?post_type=product&...        0.0     product
471   https://www.bottle-neck.fr/?post_type=product&...        0.0     product
1209  https://www.bottle-neck.fr/wp-content/uploads/...        0.0  attachment

[5 rows x 23 columns]
```

In [84]:
```
nombre_de_doublons = len(doublons)
```

In [85]:
```
print(f"Il y a {nombre_de_doublons} doublons dans les clés.")
```

```
Il y a 825 doublons dans les clés.
```

In [86]:
```
#Identifier les lignes sans code articles
colonne_code_article = 'sku'
```

In [87]:
```
lignes_sans_code_articl = df[df['sku'].isnull()]
```

In [88]:
```
print(lignes_sans_code_articl)
```

```
Empty DataFrame
Columns: [sku, virtual, downloadable, rating_count, average_rating, total_sales, tax_status, post_author, post_date, post_date_gmt, post_con
tent, post_title, post_excerpt, post_status, comment_status, post_password, post_name, post_modified, post_modified_gmt, post_parent, guid,
menu_order, post_type]
Index: []
```


columns]

In [89]:
```
lignes_sans_code = df[df['sku'].isnull() | df['sku'] == '']
```

In [90]:
```
print(lignes_sans_code)
```

```
Empty DataFrame
Columns: [sku, virtual, downloadable, rating_count, average_rating, total_sales, tax_status, post_author, post_date, post_date_gmt, post_con
tent, post_title, post_excerpt, post_status, comment_status, post_password, post_name, post_modified, post_modified_gmt, post_parent, guid,
menu_order, post_type]
Index: []

[0 rows x 23 columns]
```

In [91]:
```
lignes_non_renseignees = df[df.isna().any(axis=1)]
```

In [92]:
```
print(lignes_non_renseignees)
```

```
                      sku  virtual  downloadable  rating_count  \
0     bon-cadeau-25-euros        0             0             0
1                   15298        0             0             0
2                   15296        0             0             0
3                   15300        0             0             0
4                   19814        0             0             0
...                   ...      ...           ...           ...
1508                16135        0             0             0
1509                15891        0             0             0
1510                15887        0             0             0
1511              13127-1        0             0             0
1512                16230        0             0             0

      average_rating  total_sales tax_status  post_author           post_date  \
0                0.0         10.0    taxable          1.0 2018-06-01 13:53:46
1                0.0          6.0    taxable          2.0 2018-02-08 12:58:52
2                0.0          0.0    taxable          2.0 2018-02-08 13:49:41
3                0.0          0.0    taxable          2.0 2018-02-08 14:08:36
4                0.0          3.0    taxable          2.0 2018-02-09 14:01:05
...              ...          ...        ...          ...                 ...
1508             0.0          5.0        NaN          2.0 2020-04-25 13:22:38
1509             0.0          0.0        NaN          2.0 2020-05-02 14:53:40
1510             0.0          0.0        NaN          2.0 2020-05-02 15:00:54
1511             0.0          0.0        NaN          2.0 2020-06-09 15:42:04
1512             0.0          0.0        NaN          2.0 2020-07-20 11:00:00

           post_date_gmt  ...  post_status comment_status post_password  \
0    2018-06-01 11:53:46  ...      publish         closed           NaN
1    2018-02-08 11:58:52  ...      publish         closed           NaN
2    2018-02-08 12:49:41  ...      publish         closed           NaN
3    2018-02-08 13:08:36  ...      publish         closed           NaN
4    2018-02-09 13:01:05  ...      publish         closed           NaN
...                  ...  ...          ...            ...           ...
1508 2020-04-25 11:22:38  ...      publish         closed           NaN
1509 2020-05-02 12:53:40  ...      publish         closed           NaN
1510 2020-05-02 13:00:54  ...      publish         closed           NaN
1511 2020-06-09 13:42:04  ...      publish         closed           NaN
1512 2020-07-20 09:00:00  ...      publish         closed           NaN

                                          post_name       post_modified  \
0                           bon-cadeau-de-25-euros 2018-06-01 14:13:57
1     pierre-jean-villa-saint-joseph-preface-2018 2019-12-30 09:30:29
2       pierre-jean-villa-saint-joseph-tilde-2017 2019-12-21 09:00:17
3    pierre-jean-villa-croze-hermitage-accroche-coe... 2020-06-26 18:15:03
4              pierre-jean-villa-igp-gamine-2018 2020-01-04 16:36:01
...                                          ...                 ...
1508             mouthes-le-bihan-aime-chai-2015 2020-08-26 17:35:03
1509   camin-larredya-jurancon-sec-la-virada-2018 2020-08-26 17:35:02
1510       jamet-cote-rotie-fructus-voluptas-2018 2020-08-14 18:15:03
1511  clos-du-mont-olivet-chateauneuf-du-pape-2007-2 2020-07-20 17:09:06
1512  domaine-saint-nicolas-fiefs-vendeens-blanc-les... 2020-08-13 10:45:03

       post_modified_gmt post_parent  \
0    2018-06-01 12:13:57         0.0
1    2019-12-30 08:30:29         0.0
2    2019-12-21 08:00:17         0.0
3    2020-06-26 16:15:03         0.0
4    2020-01-04 15:36:01         0.0
...                  ...         ...
1508 2020-08-26 15:35:03         0.0
1509 2020-08-26 15:35:02         0.0
1510 2020-08-14 16:15:03         0.0
1511 2020-07-20 15:09:06         0.0
1512 2020-08-13 08:45:03         0.0

                                                   guid menu_order   post_type
0     https://www.bottle-neck.fr/?post_type=product&...        0.0     product
1     https://www.bottle-neck.fr/?post_type=product&...        0.0     product
2     https://www.bottle-neck.fr/?post_type=product&...        0.0     product
3     https://www.bottle-neck.fr/?post_type=product&...        0.0     product
4     https://www.bottle-neck.fr/?post_type=product&...        0.0     product
...                                                 ...        ...         ...
1508  https://www.bottle-neck.fr/wp-content/uploads/...        0.0  attachment
1509  https://www.bottle-neck.fr/wp-content/uploads/...        0.0  attachment
1510  https://www.bottle-neck.fr/wp-content/uploads/...        0.0  attachment
1511  https://www.bottle-neck.fr/wp-content/uploads/...        0.0  attachment
1512  https://www.bottle-neck.fr/wp-content/uploads/...        0.0  attachment

[1431 rows x 23 columns]
```


In [93]:
```
masque_non_renseignees = df.isna()
```

In [94]:
```
#Créer un dataframe avec uniquement les lignes sans code article
lignes_sans_valeurs_renseignees = masque_non_renseignees.all(axis=1)
```

In [95]:
```
df_toutes_valeurs_non_renseignees = df[lignes_sans_valeurs_renseignees]
```

In [96]:
```
df_toutes_valeurs_non_renseignees.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 0 entries
Data columns (total 23 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   sku                0 non-null      object
 1   virtual            0 non-null      int64
 2   downloadable       0 non-null      int64
 3   rating_count       0 non-null      int64
 4   average_rating     0 non-null      float64
 5   total_sales        0 non-null      float64
 6   tax_status         0 non-null      object
 7   post_author        0 non-null      float64
 8   post_date          0 non-null      datetime64[ns]
 9   post_date_gmt      0 non-null      datetime64[ns]
 10  post_content       0 non-null      float64
 11  post_title         0 non-null      object
 12  post_excerpt       0 non-null      object
 13  post_status        0 non-null      object
 14  comment_status     0 non-null      object
 15  post_password      0 non-null      float64
 16  post_name          0 non-null      object
 17  post_modified      0 non-null      datetime64[ns]
 18  post_modified_gmt  0 non-null      datetime64[ns]
 19  post_parent        0 non-null      float64
 20  guid               0 non-null      object
 21  menu_order         0 non-null      float64
 22  post_type          0 non-null      object
dtypes: datetime64[ns](4), float64(7), int64(3), object(9)
memory usage: 0.0+ bytes
```

In [97]:
```
masque_sans_code_article = df[colonne_code_article].isna()
```

In [98]:
```
#Créer un dataframe avec uniquement les lignes sans code article
lignes_sans_code_article = df[masque_sans_code_article]
```

In [99]:
```
print(lignes_sans_code_article)
```

```
Empty DataFrame
Columns: [sku, virtual, downloadable, rating_count, average_rating, total_sales, tax_status, post_author, post_date, post_date_gmt, post_con
tent, post_title, post_excerpt, post_status, comment_status, post_password, post_name, post_modified, post_modified_gmt, post_parent, guid,
menu_order, post_type]
Index: []

[0 rows x 23 columns]
```

In [100…
```
colonne_code_article = 'sku'
```

In [101…
```
doublons = df.duplicated(subset=colonne_code_article, keep=False)
```

In [102…
```
lignes_avec_doublons = df[doublons]
print("Lignes avec des doublons de codes d'articles :")
print(lignes_avec_doublons)
```

```
Lignes avec des doublons de codes d'articles :
                      sku  virtual  downloadable  rating_count  \
0     bon-cadeau-25-euros        0             0             0
1                   15298        0             0             0
2                   15296        0             0             0
3                   15300        0             0             0
4                   19814        0             0             0
...                   ...      ...           ...           ...
1508                16135        0             0             0
1509                15891        0             0             0
1510                15887        0             0             0
1511              13127-1        0             0             0
1512                16230        0             0             0

      average_rating  total_sales tax_status  post_author           post_date  \
0                0.0         10.0    taxable          1.0 2018-06-01 13:53:46
1                0.0          6.0    taxable          2.0 2018-02-08 12:58:52
2                0.0          0.0    taxable          2.0 2018-02-08 13:49:41
3                0.0          0.0    taxable          2.0 2018-02-08 14:08:36
4                0.0          3.0    taxable          2.0 2018-02-09 14:01:05
...              ...          ...        ...          ...                 ...
1508             0.0          5.0        NaN          2.0 2020-04-25 13:22:38
1509             0.0          0.0        NaN          2.0 2020-05-02 14:53:40
1510             0.0          0.0        NaN          2.0 2020-05-02 15:00:54
1511             0.0          0.0        NaN          2.0 2020-06-09 15:42:04
1512             0.0          0.0        NaN          2.0 2020-07-20 11:00:00

           post_date_gmt  ...  post_status comment_status post_password  \
0    2018-06-01 11:53:46  ...      publish         closed           NaN
1    2018-02-08 11:58:52  ...      publish         closed           NaN
2    2018-02-08 12:49:41  ...      publish         closed           NaN
3    2018-02-08 13:08:36  ...      publish         closed           NaN
4    2018-02-09 13:01:05  ...      publish         closed           NaN
...                  ...  ...          ...            ...           ...
1508 2020-04-25 11:22:38  ...      publish         closed           NaN
1509 2020-05-02 12:53:40  ...      publish         closed           NaN
1510 2020-05-02 13:00:54  ...      publish         closed           NaN
1511 2020-06-09 13:42:04  ...      publish         closed           NaN
1512 2020-07-20 09:00:00  ...      publish         closed           NaN

                                          post_name       post_modified  \
0                             bon-cadeau-de-25-euros 2018-06-01 14:13:57
1          pierre-jean-villa-saint-joseph-preface-2018 2019-12-30 09:30:29
2            pierre-jean-villa-saint-joseph-tilde-2017 2019-12-21 09:00:17
3     pierre-jean-villa-croze-hermitage-accroche-coe... 2020-06-26 18:15:03
4                   pierre-jean-villa-igp-gamine-2018 2020-01-04 16:36:01
...                                               ...                 ...
1508                  mouthes-le-bihan-aime-chai-2015 2020-08-26 17:35:03
1509        camin-larredya-jurancon-sec-la-virada-2018 2020-08-26 17:35:02
1510            jamet-cote-rotie-fructus-voluptas-2018 2020-08-14 18:15:03
1511    clos-du-mont-olivet-chateauneuf-du-pape-2007-2 2020-07-20 17:09:06
1512  domaine-saint-nicolas-fiefs-vendeens-blanc-les... 2020-08-13 10:45:03

       post_modified_gmt post_parent  \
0    2018-06-01 12:13:57         0.0
1    2019-12-30 08:30:29         0.0
2    2019-12-21 08:00:17         0.0
3    2020-06-26 16:15:03         0.0
4    2020-01-04 15:36:01         0.0
...                  ...         ...
1508 2020-08-26 15:35:03         0.0
1509 2020-08-26 15:35:02         0.0
1510 2020-08-14 16:15:03         0.0
1511 2020-07-20 15:09:06         0.0
1512 2020-08-13 08:45:03         0.0

                                                   guid menu_order   post_type
0     https://www.bottle-neck.fr/?post_type=product&...        0.0     product
1     https://www.bottle-neck.fr/?post_type=product&...        0.0     product
2     https://www.bottle-neck.fr/?post_type=product&...        0.0     product
3     https://www.bottle-neck.fr/?post_type=product&...        0.0     product
4     https://www.bottle-neck.fr/?post_type=product&...        0.0     product
...                                                 ...        ...         ...
1508  https://www.bottle-neck.fr/wp-content/uploads/...        0.0  attachment
1509  https://www.bottle-neck.fr/wp-content/uploads/...        0.0  attachment
1510  https://www.bottle-neck.fr/wp-content/uploads/...        0.0  attachment
1511  https://www.bottle-neck.fr/wp-content/uploads/...        0.0  attachment
1512  https://www.bottle-neck.fr/wp-content/uploads/...        0.0  attachment

[1431 rows x 23 columns]
```

In [103…
```
nombre_codes_articles_uniques = df[colonne_code_article].nunique()
```

In [104…
```
print("Nombre de codes d'articles uniques :", nombre_codes_articles_uniques)
```

```
Nombre de codes d'articles uniques : 715
```

In [105…
```
lignes_sans_code_article = df[df[colonne_code_article].isna()]
```

In [106…
```
print(lignes_sans_code_article)
```

```
Empty DataFrame
Columns: [sku, virtual, downloadable, rating_count, average_rating, total_sales, tax_status, post_author, post_date, post_date_gmt, post_con
tent, post_title, post_excerpt, post_status, comment_status, post_password, post_name, post_modified, post_modified_gmt, post_parent, guid,
menu_order, post_type]
Index: []

[0 rows x 23 columns]
```

In [107…
```
lignes_sans_code_article.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 0 entries
Data columns (total 23 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   sku                0 non-null      object
 1   virtual            0 non-null      int64
 2   downloadable       0 non-null      int64
 3   rating_count       0 non-null      int64
 4   average_rating     0 non-null      float64
 5   total_sales        0 non-null      float64
 6   tax_status         0 non-null      object
 7   post_author        0 non-null      float64
 8   post_date          0 non-null      datetime64[ns]
 9   post_date_gmt      0 non-null      datetime64[ns]
 10  post_content       0 non-null      float64
 11  post_title         0 non-null      object
 12  post_excerpt       0 non-null      object
 13  post_status        0 non-null      object
 14  comment_status     0 non-null      object
 15  post_password      0 non-null      float64
 16  post_name          0 non-null      object
 17  post_modified      0 non-null      datetime64[ns]
 18  post_modified_gmt  0 non-null      datetime64[ns]
 19  post_parent        0 non-null      float64
 20  guid               0 non-null      object
 21  menu_order         0 non-null      float64
 22  post_type          0 non-null      object
dtypes: datetime64[ns](4), float64(7), int64(3), object(9)
memory usage: 0.0+ bytes
```

In [116…
```
df_sans_doublons = df.drop_duplicates(subset='sku')
```

In [117…
```
print(df_sans_doublons)
```

```
                    sku  virtual  downloadable  rating_count  average_rating  \
0    bon-cadeau-25-euros        0             0             0             0.0
1                  15298        0             0             0             0.0
2                  15296        0             0             0             0.0
3                  15300        0             0             0             0.0
4                  19814        0             0             0             0.0
..                   ...      ...           ...           ...             ...
762                16135        0             0             0             0.0
767                15891        0             0             0             0.0
768                15887        0             0             0             0.0
797              13127-1        0             0             0             0.0
798                16230        0             0             0             0.0

     total_sales tax_status  post_author           post_date  \
0           10.0    taxable          1.0 2018-06-01 13:53:46
1            6.0    taxable          2.0 2018-02-08 12:58:52
2            0.0    taxable          2.0 2018-02-08 13:49:41
3            0.0    taxable          2.0 2018-02-08 14:08:36
4            3.0    taxable          2.0 2018-02-09 14:01:05
..           ...        ...          ...                 ...
762          5.0    taxable          2.0 2020-04-25 13:22:38
767          0.0    taxable          2.0 2020-05-02 14:53:40
768          0.0    taxable          2.0 2020-05-02 15:00:54
797          0.0    taxable          2.0 2020-06-09 15:42:04
798          0.0    taxable          2.0 2020-07-20 11:00:00

          post_date_gmt  ...  post_status comment_status post_password  \
0   2018-06-01 11:53:46  ...      publish         closed           NaN
1   2018-02-08 11:58:52  ...      publish         closed           NaN
2   2018-02-08 12:49:41  ...      publish         closed           NaN
3   2018-02-08 13:08:36  ...      publish         closed           NaN
4   2018-02-09 13:01:05  ...      publish         closed           NaN
..                  ...  ...          ...            ...           ...
762 2020-04-25 11:22:38  ...      publish         closed           NaN
767 2020-05-02 12:53:40  ...      publish         closed           NaN
768 2020-05-02 13:00:54  ...      publish         closed           NaN
797 2020-06-09 13:42:04  ...      publish         closed           NaN
798 2020-07-20 09:00:00  ...      publish         closed           NaN

                                        post_name       post_modified  \
0                        bon-cadeau-de-25-euros 2018-06-01 14:13:57
1          pierre-jean-villa-saint-joseph-preface-2018 2019-12-30 09:30:29
2            pierre-jean-villa-saint-joseph-tilde-2017 2019-12-21 09:00:17
3    pierre-jean-villa-croze-hermitage-accroche-coe... 2020-06-26 18:15:03
4                    pierre-jean-villa-igp-gamine-2018 2020-01-04 16:36:01
..                                                 ...                 ...
762                  mouthes-le-bihan-aime-chai-2015 2020-08-26 17:35:03
767         camin-larredya-jurancon-sec-la-virada-2018 2020-08-26 17:35:02
768            jamet-cote-rotie-fructus-voluptas-2018 2020-08-14 18:15:03
797     clos-du-mont-olivet-chateauneuf-du-pape-2007-2 2020-07-20 17:09:06
798  domaine-saint-nicolas-fiefs-vendeens-blanc-les... 2020-08-13 10:45:03

      post_modified_gmt post_parent  \
0   2018-06-01 12:13:57         0.0
1   2019-12-30 08:30:29         0.0
2   2019-12-21 08:00:17         0.0
3   2020-06-26 16:15:03         0.0
4   2020-01-04 15:36:01         0.0
..                  ...         ...
762 2020-08-26 15:35:03         0.0
767 2020-08-26 15:35:02         0.0
768 2020-08-14 16:15:03         0.0
797 2020-07-20 15:09:06         0.0
798 2020-08-13 08:45:03         0.0

                                                  guid menu_order  post_type
0    https://www.bottle-neck.fr/?post_type=product&...        0.0    product
1    https://www.bottle-neck.fr/?post_type=product&...        0.0    product
2    https://www.bottle-neck.fr/?post_type=product&...        0.0    product
3    https://www.bottle-neck.fr/?post_type=product&...        0.0    product
4    https://www.bottle-neck.fr/?post_type=product&...        0.0    product
..                                                 ...        ...        ...
762  https://www.bottle-neck.fr/?post_type=product&...        0.0    product
767  https://www.bottle-neck.fr/?post_type=product&...        0.0    product
768  https://www.bottle-neck.fr/?post_type=product&...        0.0    product
797  https://www.bottle-neck.fr/?post_type=product&...        0.0    product
798  https://www.bottle-neck.fr/?post_type=product&...        0.0    product

[715 rows x 23 columns]
```

Loading [MathJax]/extensions/Safe.js

In [108…
```
#2.3 - Analyse exploratoire du fichier liaison.xlsx</
df_liaison = pd.read_excel('C:/Users/pc/OneDrive/Documents/projet 6 Optimisez la gestion du stock d'une boutique en nettoyant ses données/lia
```

In [109…
```
df_liaison.head()
```

Out[109]:

	product_id	id_web
0	3847	15298
1	3849	15296
2	3850	15300
3	4032	19814
4	4039	19815

In [110…
```
#"#Nombre d'observations
print("Nombre d'observations : ", df_liaison.shape[0])
```

```
Nombre d'observations :  825
```

In [111…
```
#Nombre de caractéristiques
print("Nombre de caractéristiques : ", df_liaison.shape[1])
```

```
Nombre de caractéristiques :  2
```

In [112…
```
df_liaison.isnull().sum()
```

Out[112]:
```
product_id     0
id_web        91
dtype: int64
```

In [113…
```
#La nature des données dans chacune des colonnes
df_liaison.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 825 entries, 0 to 824
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   product_id  825 non-null    int64 
 1   id_web      734 non-null    object
dtypes: int64(1), object(1)
memory usage: 13.0+ KB
```

In [118…
```
#Le nombre de valeurs présentes dans chacune des colonnes
nombre_de_valeurs_non_nulles = df_liaison.count()
```

In [119…
```
print(nombre_de_valeurs_non_nulles)
```

```
product_id    825
id_web        734
dtype: int64
```

In [120…
```
#Les valeurs de la colonne \"product_id\" sont elles toutes uniques
sont_uniques = df_liaison['product_id'].nunique() == len(df_liaison)
print("Les valeurs de la colonne 'product_id' sont-elles toutes uniques ?", sont_uniques)
```

```
Les valeurs de la colonne 'product_id' sont-elles toutes uniques ? True
```

In [121…
```
#Les valeurs de la colonne \"id_web\" sont-elles toutes uniques
df_liaison['id_web'].nunique()
```

Out[121]:
```
734
```

In [122…
```
#des articles sans correspondances
articles_sans_correspondance = erp_data[~erp_data['product_id'].isin(df_liaison['id_web'])]
```

In [123…
```
articles_sans_correspondance
```

Out[123]:

	product_id	onsale_web	price	stock_quantity
0	3847	1	24.2	0
1	3849	1	34.3	0
2	3850	1	20.8	0
3	4032	1	14.1	0
4	4039	1	46.0	0
	...	...	...	...
820	7203	0	45.0	30
821	7204	0	45.0	9
822	7247	1	54.8	23
823	7329	0	26.5	14
824	7338	1	16.3	45

822 rows × 4 columns

In [124…
```
articles_sans_correspondance_entre_erp_et_df_liaison = erp_data[~erp_data['product_id'].isin(df_liaison['product_id'])]
```

In [125…
```
articles_sans_correspondance_entre_erp_et_df_liaison
```

Out[125]:

	product_id	onsale_web	price	stock_quantity

In [126…
```
#2.4-Analyse exploratoire du fichier caracteristiques_vins.xlsx
df_caracteristiques.head()
```

Out[126]:

	post_name	poids	Région	Domaine	Appellation	Couleur	Cépage	Millésime	Garde	Contenance	Degré d'alcool	Température dégustation	Alliance mets
0	pierre-jean-villa-saint-joseph-preface-2018	1.5 kg	Rhône	Pierre Jean Villa	Saint Joseph	Rouge	100% Syrah	2020.0	4-7 ans	75cl	13%	15°C	Charcuterie, Lapin, Viande rouge, Volaille
1	pierre-jean-villa-saint-joseph-tilde-2017	1.5 kg	Rhône	Pierre Jean Villa	Saint Joseph	Rouge	100% Syrah	2019.0	6-8 ans	75cl	13%	15°C	Charcuterie, Viande rouge, Volaille
2	pierre-jean-villa-croze-hermitage-accroche-coe...	1.5 kg	Rhône	Pierre Jean Villa	Crozes-Hermitage	Rouge	100% Syrah	2020.0	3-5 ans	75cl	13%	15°C	Viande rouge, Volaille
3	pierre-jean-villa-igp-gamine-2018	1.5 kg	Rhône	Pierre Jean Villa	Collines Rhodaniennes	Rouge	100% Syrah	2020.0	3-5 ans	75cl	13%	14°C	Charcuterie, Viande rouge, Volaille
4	pierre-jean-villa-cote-rotie-carmina-2017	1.5 kg	Rhône	Pierre Jean Villa	Côte Rôtie	Rouge	100% Syrah	2019.0	10-20 ans	75cl	13%	17°C	Gibier, Viande rouge

In [127…
```
#Nombre d'observations\
nombre_observations, nombre_caracteristiques = df_caracteristiques.shape
```

In [128…
```
nombre_observations = len(df_caracteristiques)
```

In [129…
```
nombre_caracteristiques = len(df_caracteristiques.columns)
```

In [130…
```
print("Le dataset comporte {} observation(s) et {} colonne(s).".format(nombre_observations, nombre_caracteristiques))
```

```
Le dataset comporte 611 observation(s) et 13 colonne(s).
```

In [131…
```
#La nature des données dans chacune des colonnes
types_de_donnees = df_caracteristiques.dtypes
print("Nature des données dans chaque colonne :")
print(types_de_donnees)
```

```
Nature des données dans chaque colonne :
post_name                  object
poids                      object
Région                     object
Domaine                    object
Appellation                object
Couleur                    object
Cépage                     object
Millésime                 float64
Garde                      object
Contenance                 object
Degré d'alcool             object
Température dégustation    object
Alliance mets              object
dtype: object
```

In [132…
```
#Le nombre de valeurs présentes dans chacune des colonnes
nombre_de_valeurs_non_nulles = df_caracteristiques.count()
print("Nombre de valeurs non nulles dans chaque colonne :")
print(nombre_de_valeurs_non_nulles)
```

```
Nombre de valeurs non nulles dans chaque colonne :
post_name                  611
poids                      611
Région                     586
Domaine                    577
Appellation                559
Couleur                    566
Cépage                     571
Millésime                  541
Garde                      569
Contenance                 611
Degré d'alcool             586
Température dégustation    574
Alliance mets              574
dtype: int64
```


In [133…
```
#les produits avec des informations manquantes
produits_manquants = df_caracteristiques[df_caracteristiques.isna().any(axis=1)]
```

In [134…
```
print(produits_manquants)
```

```
                                        post_name   poids  \
60                    gosset-champagne-grande-reserve  1.5 kg
62                        champagne-gosset-grand-rose  1.5 kg
63                   champagne-mailly-gc-brut-reserve  1.5 kg
64          champagne-mailly-grand-cru-extra-brut-2012  1.5 kg
65                champagne-mailly-grand-cru-brut-rose  1.5 kg
..                                                 ...     ...
580   saumaize-michelin-pouilly-fuisse-ampelopsis-2016  1.5 kg
581  la-preceptorie-cotes-du-roussillon-coume-marie...  1.5 kg
582  la-preceptorie-cotes-du-roussillon-blanc-terre...  1.5 kg
601   emile-boeckel-cremant-chardonnay-extra-brut-2016  1.5 kg
610  domaine-saint-nicolas-fiefs-vendeens-blanc-les...  1.5 kg

                  Région            Domaine       Appellation Couleur  \
60             Champagne             Gosset         Champagne   Blanc
62             Champagne             Gosset         Champagne     NaN
63             Champagne   Mailly Grand Cru         Champagne   Blanc
64             Champagne   Mailly Grand Cru         Champagne   Blanc
65             Champagne   Mailly Grand Cru         Champagne     NaN
..                   ...                ...               ...     ...
580            Bourgogne  Saumaize-Michelin    Pouilly-Fuissé   Blanc
581  Languedoc-Roussillon     La Preceptorie               NaN   Blanc
582  Languedoc-Roussillon     La Preceptorie               NaN   Blanc
601               Alsace      Emile Boeckel  Crémant d'Alsace   Blanc
610    Vallée de la Loire      Saint-Nicolas               NaN   Blanc

                                              Cépage  Millésime    Garde  \
60   10% Pinot Meunier, 45% Chardonnay, 45% Pinot Noir        NaN    4 ans
62                     50% Chardonnay, 50% Pinot Noir        NaN    4 ans
63                     25% Chardonnay, 75% Pinot Noir        NaN    4 ans
64                     25% Chardonnay, 75% Pinot Noir        NaN    4 ans
65                     10% Chardonnay, 90% Pinot Noir        NaN    4 ans
..                                                ...        ...      ...
580                                       Chardonnay     2018.0      NaN
581  Carignan Blanc, Grenache Blanc, Grenache Gris,...     2020.0  6-8 ans
582                           Grenache Gris, Macabeu     2020.0  6-8 ans
601                                       Chardonnay        NaN  3-5 ans
610                                Chardonnay, Chenin     2019.0  3-5 ans

    Contenance Degré d'alcool Température dégustation  \
60        75cl            12%                   10°C
62        75cl            12%                   10°C
63        75cl            12%                   10°C
64        75cl            12%                   10°C
65        75cl            12%                   10°C
..         ...            ...                    ...
580       75cl            13%                   12°C
581       75cl            15%                   12°C
582       75cl            15%                   12°C
601       75cl            12%                    9°C
610       75cl            14%                   12°C

                                  Alliance mets
60          Apéritif, Fruits cuits, Sucré-salé, Tajine
62             Apéritif, Desserts, Foie gras, Poissons
63   Apéritif, Chaource, Légumes croquants, Suprême...
64               Apéritif, Carpaccios, Caviar, Huîtres
65                     Apéritif, Tapas, Viande Blanche
..                                                 ...
580             Apéritif, Fromages, Poissons, Volaille
581         Poisson en sauce, Viande Blanche, Volaille
582           Crustacés, Fruits de mer, Poisson grillé
601              Apéritif, Fromages, Huîtres, Poissons
610                     Crustacés, Fromages, Poissons
```



```
[118 rows x 13 columns]
```

In [135…
```
#Etape 3 - Jonction des fichiers<
#Etape 3.1 - Jonction du fichier df_erp et df_liaison</h3>
#Fusion des fichiers df_erp et df_liaison\
result = erp_data.merge(df_liaison, on='product_id', how='inner')
```

In [136…
```
#es lignes ne \"matchant\" entre les 2 fichiers?
merged = erp_data.merge(df_liaison, left_on='product_id', right_on='product_id', how='outer', indicator=True)

non_matching_rows = merged[merged['_merge'] != 'both']
count_non_matching = len(non_matching_rows)
```

In [137…
```
print("Nombre de lignes non correspondantes : ", count_non_matching)
print("Lignes non correspondantes :")
print(non_matching_rows)
```

```
Nombre de lignes non correspondantes :  0
Lignes non correspondantes :
Empty DataFrame
Columns: [product_id, onsale_web, price, stock_quantity, id_web, _merge]
Index: []
```

In [138…
```
#Etape 3.2 - Jonction du fichier df_merge et df_web</h3
df_web=df
```

In [139…
```
df_web = df_web[df_web['sku'].str.isnumeric()]
```

In [140…
```
import re
```

In [141…
```
df_web['sku'] = df_web['sku'].apply(lambda x: re.sub(r'\D', '', str(x))).astype('int64')
```

In [142…
```
##Fusionnez les datasets df_merge et df_web
df_fusion = result.merge(df_web, left_on='product_id', right_on='sku', how='inner')
```

In [143…
```
df_fusion = df_fusion.drop('sku', axis=1)
```

In [146…
```
df_pin = result.merge(df_web, left_on='id_web', right_on='sku', how='inner')
```

In [147…
```
df_officiel = df_pin.merge(df_caracteristiques, on='post_name', how='left')
```

In [148…
```
#Etape 3.3 - Jonction du fichier df_merge et df_caracteristiques</
#Fusion de la table df_merge et df_caracteristiques \
df_final = df_fusion.merge(df_caracteristiques, on='post_name', how='left')
```

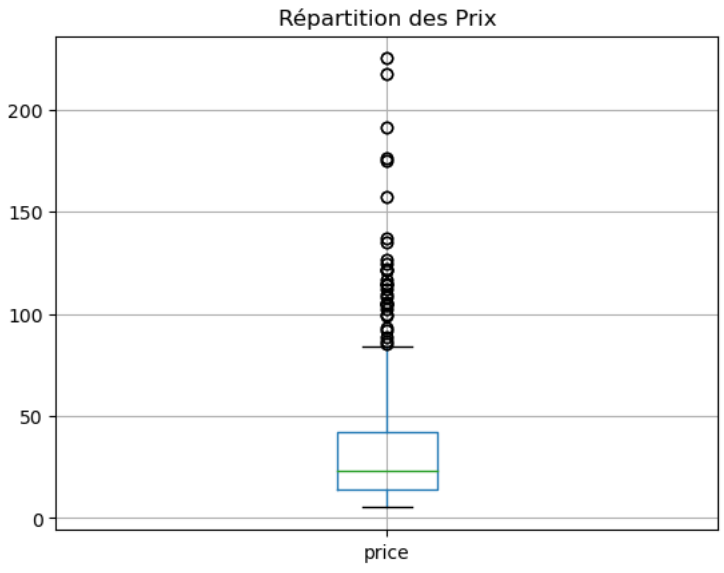
In [150…
```
print(df_final)
```

```
   product_id  onsale_web  price  stock_quantity id_web  virtual  \
0        4679           1   13.2              37  15283        0
1        4679           1   13.2              37  15283        0
2        6616           1   15.4               8  15781        0
3        6616           1   15.4               8  15781        0
4        7086           0   26.0               2    NaN        0
5        7086           0   26.0               2    NaN        0

   downloadable  rating_count  average_rating  total_sales  ...  \
0             0             0             0.0          0.0  ...
1             0             0             0.0          0.0  ...
2             0             0             0.0          0.0  ...
3             0             0             0.0          0.0  ...
4             0             0             0.0          0.0  ...
5             0             0             0.0          0.0  ...

            Domaine  Appellation Couleur                          Cépage  \
0  Mailly Grand Cru    Champagne   Blanc  25% Chardonnay, 75% Pinot Noir
1  Mailly Grand Cru    Champagne   Blanc  25% Chardonnay, 75% Pinot Noir
2             Huet      Vouvray   Blanc                          Chenin
3             Huet      Vouvray   Blanc                          Chenin
4  Mailly Grand Cru    Champagne     NaN  40% Chardonnay, 60% Pinot Noir
5  Mailly Grand Cru    Champagne     NaN  40% Chardonnay, 60% Pinot Noir

   Millésime        Garde Contenance Degré d'alcool Température dégustation  \
0        NaN      6-8 ans       75cl            12%                    10°C
1        NaN      6-8 ans       75cl            12%                    10°C
2     2008.0  15 ans et +       75cl         12,50%                    11°C
3     2008.0  15 ans et +       75cl         12,50%                    11°C
4        NaN  10 ans et +       75cl            12%                    10°C
5        NaN  10 ans et +       75cl            12%                    10°C

                                  Alliance mets
0  Coquilles Saint Jacques, Foie gras, Poissons, ...
1  Coquilles Saint Jacques, Foie gras, Poissons, ...
2  Cuisine Exotique, Desserts, Foie gras, Sucré-salé
3  Cuisine Exotique, Desserts, Foie gras, Sucré-salé
4     Apéritif, Fruits, Noix de St Jacques, Poissons
5     Apéritif, Fruits, Noix de St Jacques, Poissons

[6 rows x 39 columns]
```

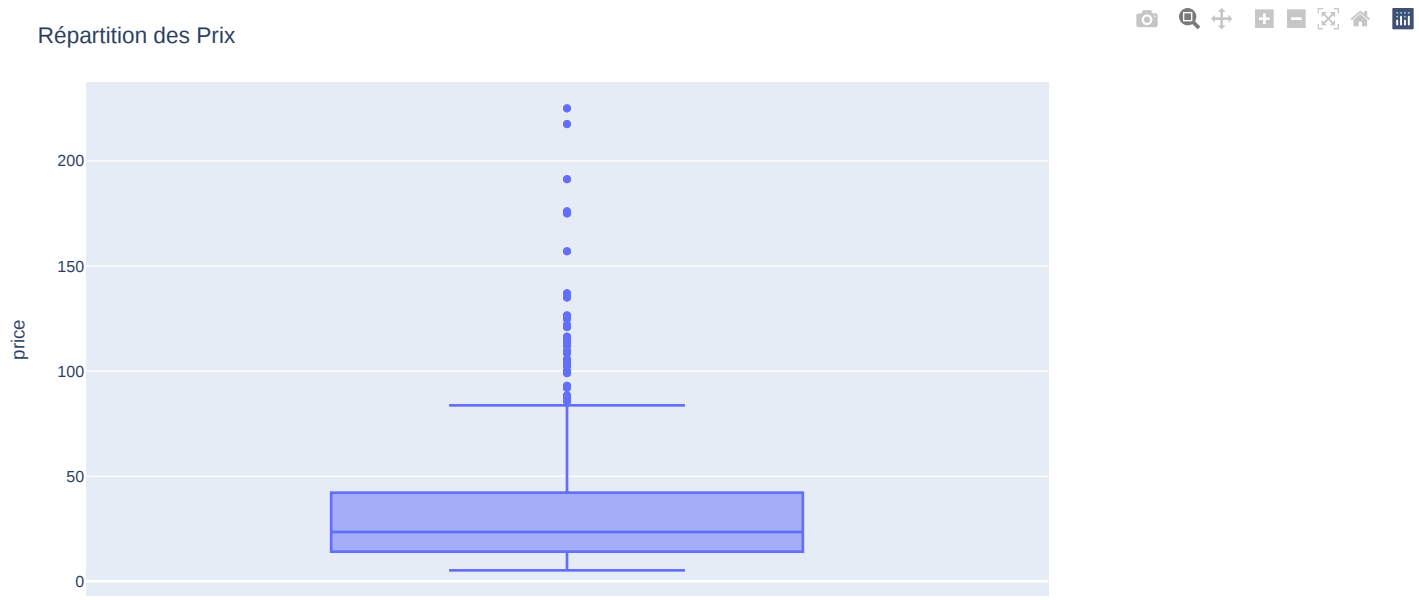
In [151…
```
import plotly.express as px
```

Loading [MathJax]/extensions/Safe.js

```
Exploration par la visualisation de données<
#Création d'une Boite à moustache de la répartition des prix grâce à Pandas
df_officiel.boxplot(column='price')
plt.title('Répartition des Prix')
plt.show()
```



In [153…
```
#Autre méthode avec plotly express
fig = px.box(df_officiel, y='price')
fig.update_layout(title='Répartition des Prix')

fig.show()
```



In [155…
```
#Etape 4.2 - Exploration par l'utisation de méthodes statistique<
#Calculer la moyenne du prix\
moyenne_prix = df_officiel['price'].mean()
print("La moyenne des prix est de :", moyenne_prix)
```

```
La moyenne des prix est de : 32.47233146067416
```

In [156…
```
#Calculer l'écart-type du prix
ecart_type_prix = df_officiel['price'].std()
print("L'écart-type des prix est de :", ecart_type_prix)
```

```
L'écart-type des prix est de : 27.82583583195121
```

In [157…
```
#Calculer le Z-score
df_officiel['Z-score'] = (df_officiel['price'] - moyenne_prix) / ecart_type_prix
```

In [158…
```
print(df_officiel['Z-score'])
```

```
0      -0.297290
1      -0.297290
2       0.065682
3       0.065682
4      -0.419478
          ...
1419   -0.173695
1420    1.312725
1421    1.312725
1422   -0.581198
1423   -0.581198
Name: Z-score, Length: 1424, dtype: float64
```


In [159…
```
#le seuil prix dont z-score est supérieur à 3?
seuil_prix = df_officiel['price'][df_officiel['Z-score'] > 3]

print("Seuil de prix pour Z-score > 3 :")
print(seuil_prix)
```

```
Seuil de prix pour Z-score > 3 :
398     225.0
399     225.0
402     126.5
403     126.5
436     176.0
437     176.0
442     157.0
443     157.0
762     137.0
763     137.0
850     217.5
851     217.5
1020    124.8
1021    124.8
1104    175.0
1105    175.0
1172    191.3
1173    191.3
1204    122.0
1205    122.0
1282    135.0
1283    135.0
1294    116.4
1295    116.4
1306    121.0
1307    121.0
1312    121.0
1313    121.0
Name: price, dtype: float64
```

In [160…
```
#Utilisation de la fonction describe de Pandas pour l'etude des mesures de dispersions
df_officiel.describe()
```

Out[160]:

	product_id	onsale_web	price	stock_quantity	sku	virtual	downloadable	rating_count	average_rating	total_sales	post_author	post_content	post_pa
count	1424.000000	1424.0	1424.000000	1424.000000	1424.000000	1424.0	1424.0	1424.0	1424.0	1424.000000	1424.0	0.0	
mean	5029.557584	1.0	32.472331	28.794944	14479.633427	0.0	0.0	0.0	0.0	4.009831	2.0	NaN	
std	786.965739	0.0	27.825836	48.050601	3008.523217	0.0	0.0	0.0	0.0	8.522875	0.0	NaN	
min	3847.000000	1.0	5.200000	0.000000	38.000000	0.0	0.0	0.0	0.0	0.000000	2.0	NaN	
25%	4279.250000	1.0	14.087500	2.000000	14369.750000	0.0	0.0	0.0	0.0	0.000000	2.0	NaN	
50%	4794.500000	1.0	23.450000	12.000000	15380.000000	0.0	0.0	0.0	0.0	1.000000	2.0	NaN	
75%	5709.500000	1.0	42.125000	35.000000	15880.250000	0.0	0.0	0.0	0.0	4.000000	2.0	NaN	
max	7338.000000	1.0	225.000000	578.000000	19822.000000	0.0	0.0	0.0	0.0	96.000000	2.0	NaN	

In [161…
```
#Etape 4.2.2 - Identification par l'interval interquartile<
from scipy import stats
```

In [162…
```
z_scores = stats.zscore(df_officiel['price'])
```

In [163…
```
#Définissez un seuil pour les articles \"outliers\" en prix
seuil = 3
articles_outliers = df_officiel[df_officiel['price'] > seuil]
```

```
s_outliers)
```

```
      product_id  onsale_web  price  stock_quantity id_web    sku  virtual  \
0           3847           1   24.2               0  15298  15298        0
1           3847           1   24.2               0  15298  15298        0
2           3849           1   34.3               0  15296  15296        0
3           3849           1   34.3               0  15296  15296        0
4           3850           1   20.8               0  15300  15300        0
...          ...         ...    ...             ...    ...    ...      ...
1419        7023           1   27.5              15  15891  15891        0
1420        7025           1   69.0               2  15887  15887        0
1421        7025           1   69.0               2  15887  15887        0
1422        7338           1   16.3              45  16230  16230        0
1423        7338           1   16.3              45  16230  16230        0

      downloadable  rating_count  average_rating  ...       Appellation  \
0                0             0             0.0  ...      Saint Joseph
1                0             0             0.0  ...      Saint Joseph
2                0             0             0.0  ...      Saint Joseph
3                0             0             0.0  ...      Saint Joseph
4                0             0             0.0  ...  Crozes-Hermitage
...            ...           ...             ...  ...               ...
1419             0             0             0.0  ...          Jurançon
1420             0             0             0.0  ...        Côte Rôtie
1421             0             0             0.0  ...        Côte Rôtie
1422             0             0             0.0  ...               NaN
1423             0             0             0.0  ...               NaN

     Couleur                                  Cépage Millésime  \
0      Rouge                              100% Syrah    2020.0
1      Rouge                              100% Syrah    2020.0
2      Rouge                              100% Syrah    2019.0
3      Rouge                              100% Syrah    2019.0
4      Rouge                              100% Syrah    2020.0
...      ...                                     ...       ...
1419   Blanc  Gros Manseng, Petit Courbu, Petit Manseng    2018.0
1420   Rouge                                   Syrah    2020.0
1421   Rouge                                   Syrah    2020.0
1422   Blanc                    Chardonnay, Chenin     2019.0
1423   Blanc                    Chardonnay, Chenin     2019.0

            Garde  Contenance Degré d'alcool Température dégustation  \
0         4-7 ans        75cl            13%                    15°C
1         4-7 ans        75cl            13%                    15°C
2         6-8 ans        75cl            13%                    15°C
3         6-8 ans        75cl            13%                    15°C
4         3-5 ans        75cl            13%                    15°C
...           ...         ...            ...                     ...
1419     8-10 ans        75cl            14%                    11°C
1420  10 ans et +        75cl            13%                    16°C
1421  10 ans et +        75cl            13%                    16°C
1422      3-5 ans        75cl            14%                    12°C
1423      3-5 ans        75cl            14%                    12°C

                                  Alliance mets   Z-score
0          Charcuterie, Lapin, Viande rouge, Volaille -0.297290
1          Charcuterie, Lapin, Viande rouge, Volaille -0.297290
2                Charcuterie, Viande rouge, Volaille  0.065682
3                Charcuterie, Viande rouge, Volaille  0.065682
4                             Viande rouge, Volaille -0.419478
...                                              ...       ...
1419  Charcuterie, Fromages, Poissons, Viande Blanche -0.178695
1420             Agneau, Gibier, Pigeon, Viande rouge  1.312725
1421             Agneau, Gibier, Pigeon, Viande rouge  1.312725
1422                   Crustacés, Fromages, Poissons -0.581198
1423                   Crustacés, Fromages, Poissons -0.581198

[1424 rows x 41 columns]
```


In [165…
```
seuil = 75
article_outlier = df_officiel[df_officiel['price'] > seuil]
```

In [166…
```
print(article_outlier )
```

```
      product_id  onsale_web  price  stock_quantity id_web    sku  virtual  \
20          4046           1   80.0               0  15269  15269        0
21          4046           1   80.0               0  15269  15269        0
70          4073           1   77.8              11  13074  13074        0
71          4073           1   77.8              11  13074  13074        0
126         4115           1  100.0              11  15382  15382        0
...          ...         ...    ...             ...    ...    ...      ...
1311        6215           1  115.0               4  12790  12790        0
1312        6216           1  121.0               6  15070  15070        0
1313        6216           1  121.0               6  15070  15070        0
1332        6299           1   78.0              10  15710  15710        0
1333        6299           1   78.0              10  15710  15710        0

      downloadable  rating_count  average_rating  ...  \
20               0             0             0.0  ...
21               0             0             0.0  ...
70               0             0             0.0  ...
71               0             0             0.0  ...
126              0             0             0.0  ...
...            ...           ...             ...  ...
1311             0             0             0.0  ...
1312             0             0             0.0  ...
1313             0             0             0.0  ...
1332             0             0             0.0  ...
1333             0             0             0.0  ...

                    Appellation Couleur      Cépage Millésime        Garde  \
20                   Côte Rôtie   Rouge  100% Syrah    2019.0    10-20 ans
21                   Côte Rôtie   Rouge  100% Syrah    2019.0    10-20 ans
70          Châteauneuf-du-Pape   Rouge    Grenache    2015.0  15 ans et +
71          Châteauneuf-du-Pape   Rouge    Grenache    2015.0  15 ans et +
126                         NaN     NaN         NaN       NaN          NaN
...                         ...     ...         ...       ...          ...
1311             Volnay 1er Cru   Rouge  Pinot Noir    2014.0    10-12 ans
1312             Volnay 1er Cru   Rouge  Pinot Noir    2016.0    10-12 ans
1313             Volnay 1er Cru   Rouge  Pinot Noir    2016.0    10-12 ans
1332  Gevrey-Chambertin 1er Cru   Rouge  Pinot Noir    2019.0  15 ans et +
1333  Gevrey-Chambertin 1er Cru   Rouge  Pinot Noir    2019.0  15 ans et +

      Contenance Degré d'alcool Température dégustation  \
20          75cl            13%                   17°C
21          75cl            13%                   17°C
70          75cl            15%                   17°C
71          75cl            15%                   17°C
126          NaN            NaN                    NaN
...          ...            ...                    ...
1311        75cl          13,5%                   15°C
1312        75cl          13,5%                   15°C
1313        75cl          13,5%                   15°C
1332        75cl          13,5%                   15°C
1333        75cl          13,5%                   15°C

                                  Alliance mets   Z-score
20                          Gibier, Viande rouge  1.708041
21                          Gibier, Viande rouge  1.708041
70                          Gibier, Viande rouge  1.628978
71                          Gibier, Viande rouge  1.628978
126                                          NaN  2.426797
...                                          ...       ...
1311  Côtes de Veau, Lapin, Viande rouge, Volaille  2.965865
1312  Côtes de Veau, Lapin, Viande rouge, Volaille  3.181492
1313  Côtes de Veau, Lapin, Viande rouge, Volaille  3.181492
1332                 Lapin, Tourte, Viande rouge  1.636165
1333                 Lapin, Tourte, Viande rouge  1.636165
```



```
[94 rows x 41 columns]
```

In [167…]

```
unique_id_web_count = df_officiel['id_web'].nunique()

total_rows = len(df_officiel)

if unique_id_web_count == total_rows:
    print("Toutes les valeurs de la colonne 'id_web' sont uniques.")
else:
    print("Il y a des valeurs en double dans la colonne 'id_web'.")
```

```
Il y a des valeurs en double dans la colonne 'id_web'.
```

In [168…]

```
doublons = df_officiel[df_officiel.duplicated(subset='id_web', keep=False)]
print(doublons)
```

```
      product_id  onsale_web  price  stock_quantity id_web    sku  virtual  \
0           3847           1   24.2               0  15298  15298        0
1           3847           1   24.2               0  15298  15298        0
2           3849           1   34.3               0  15296  15296        0
3           3849           1   34.3               0  15296  15296        0
4           3850           1   20.8               0  15300  15300        0
...          ...         ...    ...             ...    ...    ...      ...
1419        7023           1   27.5              15  15891  15891        0
1420        7025           1   69.0               2  15887  15887        0
1421        7025           1   69.0               2  15887  15887        0
1422        7338           1   16.3              45  16230  16230        0
1423        7338           1   16.3              45  16230  16230        0

      downloadable  rating_count  average_rating  ...       Appellation  \
0                0             0             0.0  ...      Saint Joseph
1                0             0             0.0  ...      Saint Joseph
2                0             0             0.0  ...      Saint Joseph
3                0             0             0.0  ...      Saint Joseph
4                0             0             0.0  ...  Crozes-Hermitage
...            ...           ...             ...  ...               ...
1419             0             0             0.0  ...          Jurançon
1420             0             0             0.0  ...        Côte Rôtie
1421             0             0             0.0  ...        Côte Rôtie
1422             0             0             0.0  ...               NaN
1423             0             0             0.0  ...               NaN

     Couleur                                    Cépage Millésime  \
0      Rouge                                 100% Syrah    2020.0
1      Rouge                                 100% Syrah    2020.0
2      Rouge                                 100% Syrah    2019.0
3      Rouge                                 100% Syrah    2019.0
4      Rouge                                 100% Syrah    2020.0
...      ...                                        ...       ...
1419   Blanc  Gros Manseng, Petit Courbu, Petit Manseng    2018.0
1420   Rouge                                     Syrah    2020.0
1421   Rouge                                     Syrah    2020.0
1422   Blanc                     Chardonnay, Chenin     2019.0
1423   Blanc                     Chardonnay, Chenin     2019.0

            Garde  Contenance Degré d'alcool Température dégustation  \
0         4-7 ans        75cl            13%                    15°C
1         4-7 ans        75cl            13%                    15°C
2         6-8 ans        75cl            13%                    15°C
3         6-8 ans        75cl            13%                    15°C
4         3-5 ans        75cl            13%                    15°C
...           ...         ...            ...                     ...
1419     8-10 ans        75cl            14%                    11°C
1420  10 ans et +        75cl            13%                    16°C
1421  10 ans et +        75cl            13%                    16°C
1422      3-5 ans        75cl            14%                    12°C
1423      3-5 ans        75cl            14%                    12°C

                                    Alliance mets   Z-score
0          Charcuterie, Lapin, Viande rouge, Volaille -0.297290
1          Charcuterie, Lapin, Viande rouge, Volaille -0.297290
2                Charcuterie, Viande rouge, Volaille  0.065682
3                Charcuterie, Viande rouge, Volaille  0.065682
4                             Viande rouge, Volaille -0.419478
...                                              ...       ...
1419  Charcuterie, Fromages, Poissons, Viande Blanche -0.178695
1420             Agneau, Gibier, Pigeon, Viande rouge  1.312725
1421             Agneau, Gibier, Pigeon, Viande rouge  1.312725
1422                   Crustacés, Fromages, Poissons -0.581198
1423                   Crustacés, Fromages, Poissons -0.581198

[1424 rows x 41 columns]
```

Loading [MathJax]/extensions/Safe.js

In [169…
```
df_officiel_sans_doublons = df_officiel.drop_duplicates(subset='id_web')
```

In [170…
```
#Définissez le nombre d'articles et la proportion de l'ensemble du catalogue \"outliers
nombre_outliers = len(articles_outliers)

proportion_outliers = nombre_outliers / len(df_officiel)

print("Nombre d'articles outliers : ", nombre_outliers)
print("Proportion d'outliers par rapport à l'ensemble du catalogue : {:.2%}".format(proportion_outliers))
```

```
Nombre d'articles outliers :  1424
Proportion d'outliers par rapport à l'ensemble du catalogue : 100.00%
```

In [176…
```
df_web = df_web.drop_duplicates(subset='sku')
```

In [177…
```
df_web.drop_duplicates(subset=['sku'], inplace=True)
```

In [178…
```
df_web = df_web.reset_index(drop=True)
```

In [179…
```
df_offociel = df_pin.merge(df_caracteristiques, on='post_name', how='left')
```

In [180…
```
df_merge = df_officiel
```

In [183…
```
print(df_merge)
```

```
      product_id  onsale_web  price  stock_quantity id_web    sku  virtual  \
0           3847           1   24.2               0  15298  15298        0
1           3847           1   24.2               0  15298  15298        0
2           3849           1   34.3               0  15296  15296        0
3           3849           1   34.3               0  15296  15296        0
4           3850           1   20.8               0  15300  15300        0
...          ...         ...    ...             ...    ...    ...      ...
1419        7023           1   27.5              15  15891  15891        0
1420        7025           1   69.0               2  15887  15887        0
1421        7025           1   69.0               2  15887  15887        0
1422        7338           1   16.3              45  16230  16230        0
1423        7338           1   16.3              45  16230  16230        0

      downloadable  rating_count  average_rating  ...  Couleur  \
0                0             0             0.0  ...    Rouge
1                0             0             0.0  ...    Rouge
2                0             0             0.0  ...    Rouge
3                0             0             0.0  ...    Rouge
4                0             0             0.0  ...    Rouge
...            ...           ...             ...  ...      ...
1419             0             0             0.0  ...    Blanc
1420             0             0             0.0  ...    Rouge
1421             0             0             0.0  ...    Rouge
1422             0             0             0.0  ...    Blanc
1423             0             0             0.0  ...    Blanc

                                      Cépage  Millésime        Garde  \
0                                  100% Syrah     2020.0      4-7 ans
1                                  100% Syrah     2020.0      4-7 ans
2                                  100% Syrah     2019.0      6-8 ans
3                                  100% Syrah     2019.0      6-8 ans
4                                  100% Syrah     2020.0      3-5 ans
...                                       ...        ...          ...
1419  Gros Manseng, Petit Courbu, Petit Manseng     2018.0     8-10 ans
1420                                    Syrah     2020.0  10 ans et +
1421                                    Syrah     2020.0  10 ans et +
1422                        Chardonnay, Chenin     2019.0      3-5 ans
1423                        Chardonnay, Chenin     2019.0      3-5 ans

     Contenance  Degré d'alcool Température dégustation  \
0          75cl             13%                   15°C
1          75cl             13%                   15°C
2          75cl             13%                   15°C
3          75cl             13%                   15°C
4          75cl             13%                   15°C
...         ...             ...                    ...
1419       75cl             14%                   11°C
1420       75cl             13%                   16°C
1421       75cl             13%                   16°C
1422       75cl             14%                   12°C
1423       75cl             14%                   12°C

                                    Alliance mets   Z-score CA_par_article
0          Charcuterie, Lapin, Viande rouge, Volaille -0.297290          145.2
1          Charcuterie, Lapin, Viande rouge, Volaille -0.297290          145.2
2                Charcuterie, Viande rouge, Volaille  0.065682            0.0
3                Charcuterie, Viande rouge, Volaille  0.065682            0.0
4                           Viande rouge, Volaille -0.419478            0.0
...                                             ...       ...            ...
1419  Charcuterie, Fromages, Poissons, Viande Blanche -0.178695            0.0
1420             Agneau, Gibier, Pigeon, Viande rouge  1.312725            0.0
1421             Agneau, Gibier, Pigeon, Viande rouge  1.312725            0.0
1422                   Crustacés, Fromages, Poissons -0.581198            0.0
1423                   Crustacés, Fromages, Poissons -0.581198            0.0

[1424 rows x 42 columns]
```


In [189…
```
df_merge.drop('sku', axis=1, inplace=True)
```

In [190…
```
df_merge_sans_doublons = df_merge.drop_duplicates(subset=['product_id', 'id_web'])
```

In [200…
```
df_merge = df_merge.drop_duplicates(subset=['product_id', 'id_web'])
```

In [201…
```
print(df_merge )
```

```
      product_id  onsale_web  price  stock_quantity id_web  virtual  \
0           3847           1   24.2               0  15298        0
2           3849           1   34.3               0  15296        0
4           3850           1   20.8               0  15300        0
6           4032           1   14.1               0  19814        0
8           4039           1   46.0               0  19815        0
...          ...         ...    ...             ...    ...      ...
1414        6928           1   19.0              20  15741        0
1416        6930           1    8.4              83  16135        0
1418        7023           1   27.5              15  15891        0
1420        7025           1   69.0               2  15887        0
1422        7338           1   16.3              45  16230        0

      downloadable  rating_count  average_rating  total_sales  ... Couleur  \
0                0             0             0.0          6.0  ...   Rouge
2                0             0             0.0          0.0  ...   Rouge
4                0             0             0.0          0.0  ...   Rouge
6                0             0             0.0          3.0  ...   Rouge
8                0             0             0.0          0.0  ...   Rouge
...            ...           ...             ...          ...  ...     ...
1414             0             0             0.0          2.0  ...     NaN
1416             0             0             0.0          5.0  ...   Rouge
1418             0             0             0.0          0.0  ...   Blanc
1420             0             0             0.0          0.0  ...   Rouge
1422             0             0             0.0          0.0  ...   Blanc

                                      Cépage Millésime        Garde  \
0                                  100% Syrah    2020.0      4-7 ans
2                                  100% Syrah    2019.0      6-8 ans
4                                  100% Syrah    2020.0      3-5 ans
6                                  100% Syrah    2020.0      3-5 ans
8                                  100% Syrah    2019.0    10-20 ans
...                                       ...       ...          ...
1414                                      NaN       NaN          NaN
1416        Cabernet Sauvignon, Malbec, Merlot    2016.0      4-7 ans
1418  Gros Manseng, Petit Courbu, Petit Manseng    2018.0     8-10 ans
1420                                    Syrah    2020.0  10 ans et +
1422                     Chardonnay, Chenin      2019.0      3-5 ans

      Contenance Degré d'alcool Température dégustation  \
0          75cl            13%                    15°C
2          75cl            13%                    15°C
4          75cl            13%                    15°C
6          75cl            13%                    14°C
8          75cl            13%                    17°C
...         ...            ...                     ...
1414        NaN            NaN                     NaN
1416       75cl          13,5%                    14°C
1418       75cl            14%                    11°C
1420       75cl            13%                    16°C
1422       75cl            14%                    12°C

                                    Alliance mets   Z-score  \
0           Charcuterie, Lapin, Viande rouge, Volaille -0.297290
2                 Charcuterie, Viande rouge, Volaille  0.065682
4                       Viande rouge, Volaille -0.419478
6                 Charcuterie, Viande rouge, Volaille -0.660262
8                          Gibier, Viande rouge  0.486155
...                                          ...       ...
1414                                       NaN -0.484166
1416  Boeuf aux carottes, Pommes de terre sarladaise... -0.865107
1418    Charcuterie, Fromages, Poissons, Viande Blanche -0.178695
1420              Agneau, Gibier, Pigeon, Viande rouge  1.312725
1422                    Crustacés, Fromages, Poissons -0.581198

      CA_par_article
0              145.2
2                0.0
4                0.0
6               42.3
8                0.0
...              ...
1414            38.0
1416            42.0
1418             0.0
1420             0.0
1422             0.0

[712 rows x 41 columns]
```

Loading [MathJax]/extensions/Safe.js

In [202…

```
#Etape 5 - Analyse univarié du CA et des quantités vendues<
#Etape 5.1 - Analyse des ventes en CA
#Créez une colonne calculant le CA par article
df_merge['CA_par_article'] = df_merge['total_sales'] * df_merge['price']
```

In [174…

```
print(df_merge['CA_par_article'])
```

```
0       145.2
1       145.2
2         0.0
3         0.0
4         0.0
        ...
1419      0.0
1420      0.0
1421      0.0
1422      0.0
1423      0.0
Name: CA_par_article, Length: 1424, dtype: float64
```

In [207…
```
#Calculez la somme de la colonne \"ca_par_article
somme_CA = df_merge['CA_par_article'].sum()
print("Somme du chiffre d'affaires par article : {:.2f} €".format(somme_CA))
```

```
Somme du chiffre d'affaires par article : 70318.60 €
```

In [204…
```
#le tri dans l'ordre décroissant du CA du dataset
df_merge = df_merge.sort_values(by='CA_par_article', ascending=False)
```

In [205…
```
print(df_merge[['product_id', 'CA_par_article']].sort_values(by='CA_par_article', ascending=False))
```

```
     product_id  CA_par_article
388        4334          4704.0
142        4144          4263.0
436        4402          2288.0
140        4142          1590.0
138        4141          1560.0
...         ...             ...
1390       6663             0.0
1392       6664             0.0
1396       6666             0.0
1400       6751             0.0
1422       7338             0.0

[712 rows x 2 columns]
```

In [208…
```
#Réinitialiser l'index du dataset par un reset_index
df_merge = df_merge.reset_index(drop=True)
```

In [209…
```
#Afficher les 20 premier articles en CA
top_20_articles = df_merge.sort_values(by='CA_par_article', ascending=False).head(20)
print(top_20_articles)
```

	product_id	onsale_web	price	stock_quantity	id_web	virtual
0	4334	1	49.0	0	7818	0
1	4144	1	49.0	11	1662	0
2	4402	1	176.0	8	3510	0
3	4142	1	53.0	8	11641	0
4	4141	1	39.0	1	304	0
5	4355	1	126.5	2	12589	0
6	4352	1	225.0	0	15940	0
7	4153	1	29.0	0	16237	0
8	6206	1	25.2	120	16580	0
9	4068	1	16.6	157	16416	0
10	4053	1	44.3	16	13127	0
11	4596	1	43.9	0	15476	0
12	4891	1	27.9	0	15807	0
13	5067	1	59.9	0	15346	0
14	6207	1	25.2	363	16077	0
15	4918	1	37.2	0	15533	0
16	5922	1	48.5	0	15343	0
17	4054	1	71.6	0	19816	0
18	4286	1	69.8	4	16066	0
19	4904	1	137.0	13	14220	0

	downloadable	rating_count	average_rating	total_sales	...	Couleur
0	0	0	0.0	96.0	...	Blanc
1	0	0	0.0	87.0	...	NaN
2	0	0	0.0	13.0	...	NaN
3	0	0	0.0	30.0	...	Blanc
4	0	0	0.0	40.0	...	Blanc
5	0	0	0.0	11.0	...	Blanc
6	0	0	0.0	5.0	...	Blanc
7	0	0	0.0	36.0	...	Rouge
8	0	0	0.0	41.0	...	Blanc
9	0	0	0.0	62.0	...	Rouge
10	0	0	0.0	23.0	...	Rouge
11	0	0	0.0	23.0	...	Blanc
12	0	0	0.0	36.0	...	Blanc
13	0	0	0.0	16.0	...	NaN
14	0	0	0.0	37.0	...	Rouge
15	0	0	0.0	24.0	...	Rouge
16	0	0	0.0	17.0	...	NaN
17	0	0	0.0	10.0	...	Rouge
18	0	0	0.0	10.0	...	Rouge
19	0	0	0.0	5.0	...	Blanc

	Cépage	Millésime	Garde
0	Chardonnay	NaN	4 ans
1	50% Chardonnay, 50% Pinot Noir	NaN	4 ans
2	NaN	NaN	NaN
3	44% Chardonnay, 56% Pinot Noir	2015.0	8-10 ans
4	10% Pinot Meunier, 45% Chardonnay, 45% Pinot Noir	NaN	4 ans
5	Pinot Noir	NaN	5-10 ans
6	30% Chardonnay, 70% Pinot Noir	NaN	10 ans et +
7	Abouriou, Cabernet Franc, Cabernet Sauvignon, ...	2014.0	10-15 ans
8	Vermentino	2020.0	3-5 ans
9	100% Syrah	2021.0	3-5 ans
10	Cinsault, Grenache, Mourvèdre, Syrah	2007.0	15 ans et +
11	Chardonnay	2018.0	4-7 ans
12	Chardonnay	2020.0	4-7 ans
13	NaN	NaN	NaN
14	Nielluccio	2018.0	6-8 ans
15	100% Syrah	2018.0	10-15 ans
16	NaN	NaN	NaN
17	Grenache, Mourvèdre, Syrah	2018.0	15 ans et +
18	Carignan, Cinsault, Grenache	2018.0	10-15 ans
19	Chardonnay	2019.0	15 ans et +

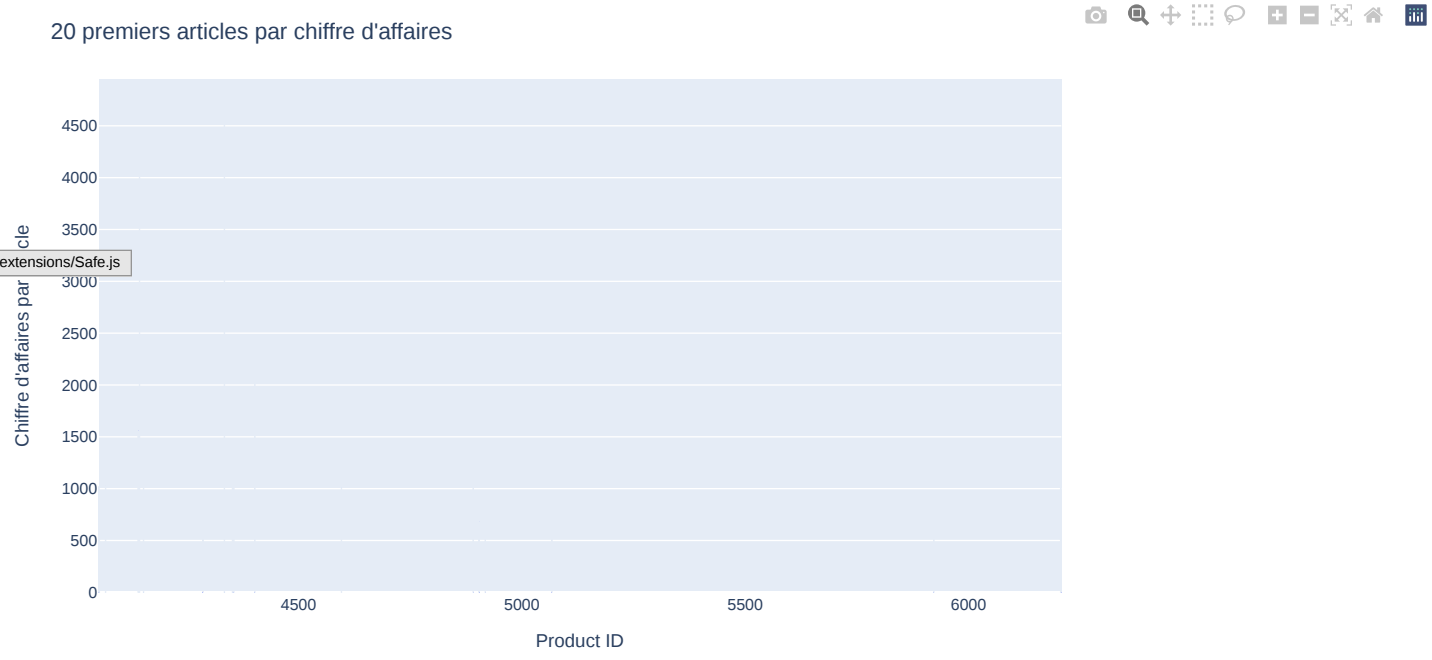
	Contenance	Degré d'alcool	Température dégustation
0	75cl	12%	10°C
1	75cl	12%	10°C
2	70cl	40%	NaN
3	75cl	12%	10°C
4	75cl	12%	10°C
5	75cl	12,50%	10°C
6	75cl	12,50%	10°C
7	75cl	13%	16°C
8	75cl	13,5%	11°C
9	75cl	13,5%	15°C
10	75cl	15%	16°C
11	75cl	13%	12°C
12	75cl	13,5%	12°C
13	NaN	NaN	NaN
14	75cl	14%	15°C
15	75cl	13%	15°C
16	NaN	NaN	NaN
17	75cl	15%	16°C
18	75cl	12,50%	16°C
19	75cl	13%	12°C

	Alliance mets	Z-score
0	Apéritif, Coquilles Saint Jacques, Huîtres, Po...	0.593968
1	Apéritif, Desserts, Foie gras, Poissons	0.593968
2	NaN	5.158072
3	Agneau, Apéritif, Fromage de chèvre, Noix de S...	0.737720
4	Apéritif, Fruits cuits, Sucré-salé, Tajine	0.234590
5	Apéritif, Desserts, Tartes aux fruits	3.379150
6	Apéritif, Crustacés, Desserts, Poissons	6.919026
7	Gibier, Grillades, Viande rouge	-0.124788

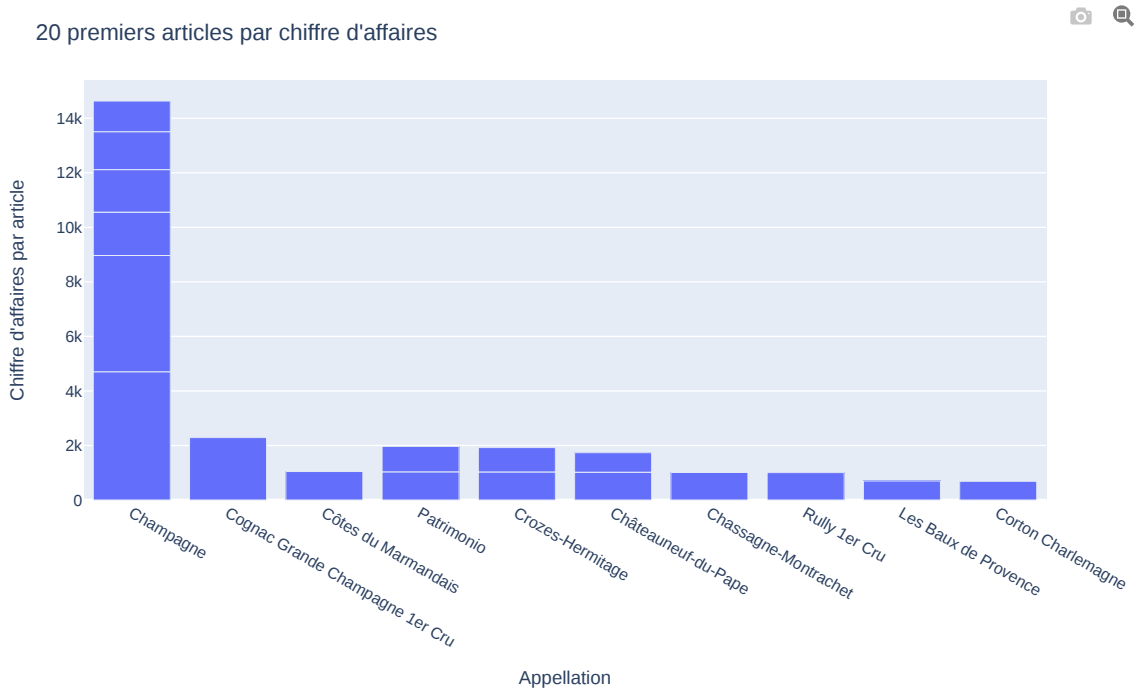
```
8        Fruits de mer, Langoustes, Poissons, Risotto -0.261352
9       Apéritif, Charcuterie, Viande rouge, Volaille -0.570417
10                              Gibier, Viande rouge  0.425061
11  Bouchées à la Reine, Ris de veau, Risotto, Vol...  0.410686
12  Fromages, Poisson en sauce, Viande Blanche, Vo... -0.164320
13                                             NaN  0.985691
14  Cuisine méditerranéenne, Gibiers à plumes, Pig... -0.261352
15            Agneau, Gibier, Grillades, Viande rouge  0.169902
16                                             NaN  0.576000
17                              Gibier, Viande rouge  1.406163
18  Grillades, Légumes du Soleil, Mouton, Viande r...  1.341475
19          Poissons nobles, Viande Blanche, Volaille  3.756497

    CA_par_article
0           4704.0
1           4263.0
2           2288.0
3           1590.0
4           1560.0
5           1391.5
6           1125.0
7           1044.0
8           1033.2
9           1029.2
10          1018.9
11          1009.7
12          1004.4
13           958.4
14           932.4
15           892.8
16           824.5
17           716.0
18           698.0
19           685.0

[20 rows x 41 columns]
```

In [ ]:
```
top_20_articles = df_merge.sort_values(by='CA_par_article', ascending=False).head(20)[['product_id', 'Appellation', 'CA_par_article']]
print(top_20_articles)
```

In [210…
```
import plotly.express as px

fig = px.bar(top_20_articles, x='product_id', y='CA_par_article')


fig.update_xaxes(title_text='Product ID')
fig.update_yaxes(title_text='Chiffre d\'affaires par article')
fig.update_layout(title_text='20 premiers articles par chiffre d\'affaires')

# Affichez le graphique
fig.show()
```



In [211…
```
import plotly.express as px

fig = px.bar(top_20_articles, x='Appellation', y='CA_par_article')


fig.update_xaxes(title_text='Appellation')
fig.update_yaxes(title_text='Chiffre d\'affaires par article')
fig.update_layout(title_text='20 premiers articles par chiffre d\'affaires')

# Affichez le graphique
fig.show()
```



In [212…
```
#Créer une colonne calculant la part du CA de la ligne dans le dataset\
df_merge['Part_du_CA'] = df_merge['CA_par_article'] / df_merge['CA_par_article'].sum()
```

In [213…
```
#Créer une colonne réalisant la somme cumulative de la colonne précedemment créée
df_merge['Somme_Cumulative_CA'] = df_merge['Part_du_CA'].cumsum()
```

In [214…
```
#calculer le nombre d'articles représentant 80% du CA\n
total_ca = df_merge['CA_par_article'].sum()  # Calcule le CA total
objectif_ca = 0.8 * total_ca  # Objectif : 80% du CA total
articles_requis = 0  # Initialisation du nombre d'articles requis
ca_cumulatif = 0  # Initialisation du CA cumulatif
```

In [215…
```
for index, row in df_merge.iterrows():
    ca_cumulatif += row['CA_par_article']
    articles_requis += 1
    if ca_cumulatif >= objectif_ca:
        break
```

In [216…
```
print("Nombre d'articles représentant 80% du CA : ", articles_requis)
```

```
Nombre d'articles représentant 80% du CA :  131
```

In [217…
```
print(df_merge['Somme_Cumulative_CA'])
```

```
0      0.066896
1      0.127520
2      0.160057
3      0.182669
4      0.204853
707    1.000000
708    1.000000
709    1.000000
710    1.000000
711    1.000000
Name: Somme_Cumulative_CA, Length: 712, dtype: float64
```


In [218…
```
print(df_merge)
```

```
     product_id  onsale_web  price  stock_quantity id_web  virtual  \
0          4334           1   49.0               0   7818        0
1          4144           1   49.0              11   1662        0
2          4402           1  176.0               8   3510        0
3          4142           1   53.0               8  11641        0
4          4141           1   39.0               1    304        0
..          ...         ...    ...             ...    ...      ...
707        4653           1   36.2              18  13515        0
708        5483           1   17.9              22  15753        0
709        5481           1   11.5              46  15138        0
710        4654           1   33.4               0  13514        0
711        7338           1   16.3              45  16230        0

     downloadable  rating_count  average_rating  total_sales  ... Millésime  \
0               0             0             0.0         96.0  ...       NaN
1               0             0             0.0         87.0  ...       NaN
2               0             0             0.0         13.0  ...       NaN
3               0             0             0.0         30.0  ...    2015.0
4               0             0             0.0         40.0  ...       NaN
..            ...           ...             ...          ...  ...       ...
707             0             0             0.0          0.0  ...    2016.0
708             0             0             0.0          0.0  ...    2018.0
709             0             0             0.0          0.0  ...    2018.0
710             0             0             0.0          0.0  ...       NaN
711             0             0             0.0          0.0  ...    2019.0

        Garde Contenance Degré d'alcool  Température dégustation  \
0       4 ans       75cl            12%                     10°C
1       4 ans       75cl            12%                     10°C
2         NaN       70cl            40%                      NaN
3    8-10 ans       75cl            12%                     10°C
4       4 ans       75cl            12%                     10°C
..        ...        ...            ...                      ...
707  5-10 ans       75cl            13%                     15°C
708   6-8 ans       75cl            15%                     15°C
709   4-7 ans       75cl            13%                     15°C
710       NaN        NaN            NaN                      NaN
711   3-5 ans       75cl            14%                     12°C

                                      Alliance mets   Z-score  \
0    Apéritif, Coquilles Saint Jacques, Huîtres, Po...  0.593968
1              Apéritif, Desserts, Foie gras, Poissons  0.593968
2                                                 NaN  5.158072
3    Agneau, Apéritif, Fromage de chèvre, Noix de S...  0.737720
4           Apéritif, Fruits cuits, Sucré-salé, Tajine  0.234590
..                                                 ...       ...
707              Gibier, Tartare de Boeuf, Viande rouge  0.133964
708   Fromages, Viande Blanche, Viande rouge, Volaille -0.523698
709  Charcuterie, Salade, Viande Blanche, Viande ro... -0.753700
710                                               NaN  0.033338
711                    Crustacés, Fromages, Poissons -0.581198

    CA_par_article Part_du_CA  Somme_Cumulative_CA
0           4704.0   0.066896              0.066896
1           4263.0   0.060624              0.127520
2           2288.0   0.032538              0.160057
3           1590.0   0.022611              0.182669
4           1560.0   0.022185              0.204853
..             ...        ...                   ...
707            0.0   0.000000              1.000000
708            0.0   0.000000              1.000000
709            0.0   0.000000              1.000000
710            0.0   0.000000              1.000000
711            0.0   0.000000              1.000000
```



```
[712 rows x 43 columns]
```

In [219…
```python
# Étape 1 : Calcul du chiffre d'affaires total
ca_total = df_merge['CA_par_article'].sum()

# Étape 2 : Sélection des articles représentant 80 % du chiffre d'affaires total ou moins
seuil_80_percent = 0.8 * ca_total
articles_80_percent = df_merge[df_merge['Somme_Cumulative_CA'] <= seuil_80_percent]

# Étape 3 : Calcul du chiffre d'affaires de ces articles
ca_articles_80_percent = articles_80_percent['CA_par_article'].sum()

# Étape 4 : Calcul de la proportion
proportion = ca_articles_80_percent / ca_total

print("Proportion d'articles représentant 80 % du CA total : {:.2%}".format(proportion))
```

```
Proportion d'articles représentant 80 % du CA total : 100.00%
```

In [220…
```python
#Etape 5.2 - Analyse des ventes en Quantités<
#le tri dans l'ordre décroissant de quantités vendues du dataset df_last
df_merge = df_merge.sort_values(by='total_sales', ascending=False)
```

In [221…
```python
#Réinitialiser l'index du dataset par un reset_index
df_merge = df_merge.reset_index(drop=True)
```

In [222…
```python
#Afficher les 20 premier articles en quantité
top_20_articles_quantity = df_merge.sort_values(by='total_sales', ascending=False).head(20)
```

In [223…
```python
top_20_articles_quantity
```

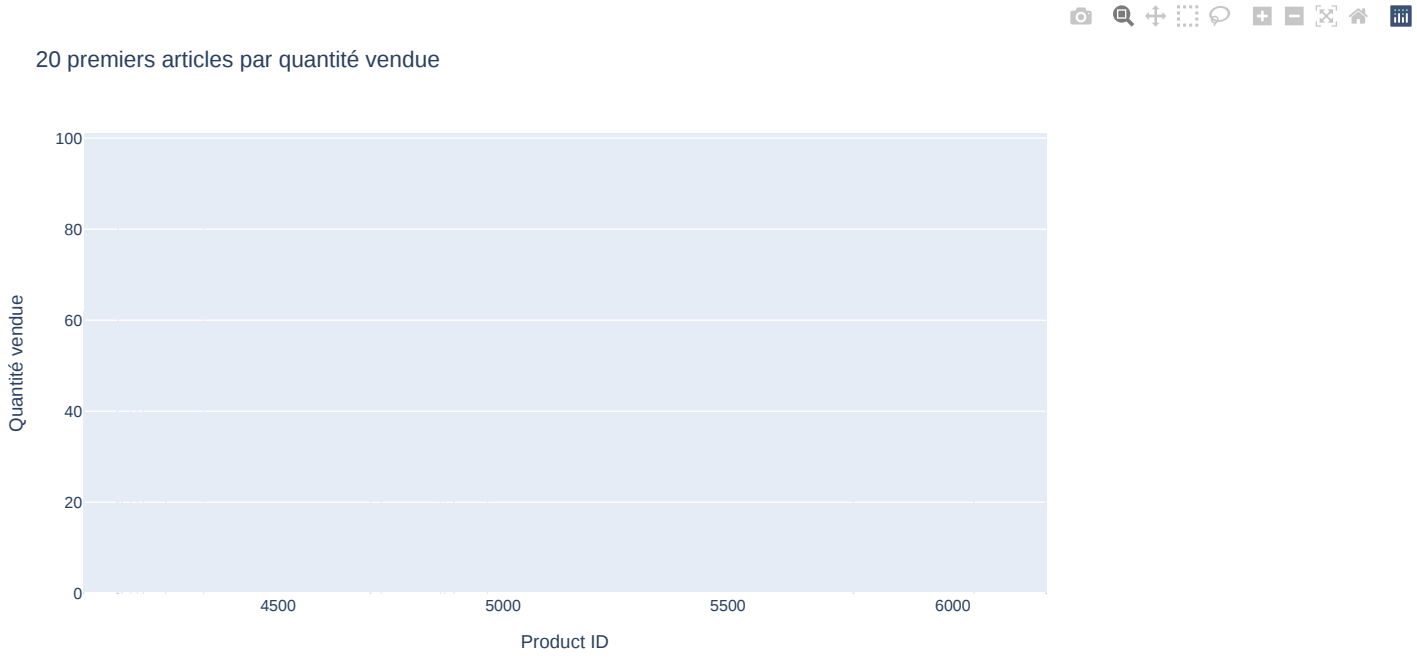
Out[223]:

	product_id	onsale_web	price	stock_quantity	id_web	virtual	downloadable	rating_count	average_rating	total_sales	...	Millésime	Garde	Contenance	Degré d'alcool	Températ dégustat
0	4334	1	49.0	0	7818	0	0	0	0.0	96.0	...	NaN	4 ans	75cl	12%	1
1	4144	1	49.0	11	1662	0	0	0	0.0	87.0	...	NaN	4 ans	75cl	12%	1
2	4068	1	16.6	157	16416	0	0	0	0.0	62.0	...	2021.0	3-5 ans	75cl	13,5%	1
3	4200	1	5.8	190	16295	0	0	0	0.0	46.0	...	2021.0	2 ans	75cl	12%	1
4	4172	1	5.7	167	16210	0	0	0	0.0	43.0	...	2021.0	3 ans	75cl	12,50%	1
5	4187	1	13.3	90	16189	0	0	0	0.0	42.0	...	2020.0	3-5 ans	75cl	13,5%	1
6	6206	1	25.2	120	16580	0	0	0	0.0	41.0	...	2020.0	3-5 ans	75cl	13,5%	1
7	4141	1	39.0	1	304	0	0	0	0.0	40.0	...	NaN	4 ans	75cl	12%	1
8	6047	1	10.9	46	16264	0	0	0	0.0	38.0	...	2020.0	3 ans	75cl	14%	1
9	4729	1	8.6	151	38	0	0	0	0.0	38.0	...	NaN	3-5 ans	75cl	12%	
10	6207	1	25.2	363	16077	0	0	0	0.0	37.0	...	2018.0	6-8 ans	75cl	14%	1
12	4891	1	27.9	0	15807	0	0	0	0.0	36.0	...	2020.0	4-7 ans	75cl	13,5%	1
11	4153	1	29.0	0	16237	0	0	0	0.0	36.0	...	2014.0	10-15 ans	75cl	13%	1
13	4870	1	9.3	0	16149	0	0	0	0.0	33.0	...	2021.0	3 ans	75cl	13%	1
14	4706	1	16.8	23	15349	0	0	0	0.0	32.0	...	NaN	NaN	NaN	NaN	N
15	4142	1	53.0	8	11641	0	0	0	0.0	30.0	...	2015.0	8-10 ans	75cl	12%	1
16	4250	1	19.5	14	16317	0	0	0	0.0	30.0	...	2019.0	3-5 ans	75cl	13%	1
17	4861	1	8.5	284	15307	0	0	0	0.0	29.0	...	2018.0	3-5 ans	75cl	13,5%	1
18	4965	1	7.1	203	16586	0	0	0	0.0	26.0	...	2020.0	3 ans	75cl	13%	1
19	5778	1	5.8	36	15561	0	0	0	0.0	24.0	...	2020.0	3-5 ans	75cl	13%	1

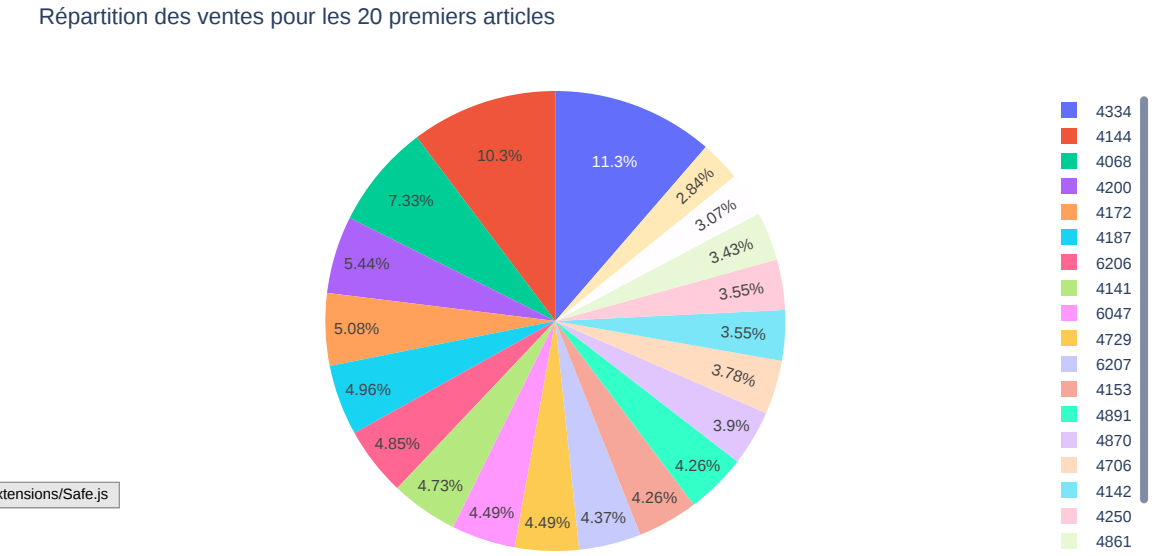
20 rows × 43 columns

In [224…
```
top_20_articles_by_quantity = df_merge[['product_id', 'Appellation', 'CA_par_article', 'total_sales']].head(20)
```

In [225…
```
fig = px.bar(top_20_articles_by_quantity, x='product_id', y='total_sales', text='Appellation', title='20 premiers articles par quantité vendu
fig.update_traces(texttemplate='%{text}', textposition='outside')
fig.update_layout(xaxis_title='Product ID', yaxis_title='Quantité vendue')
fig.show()
```

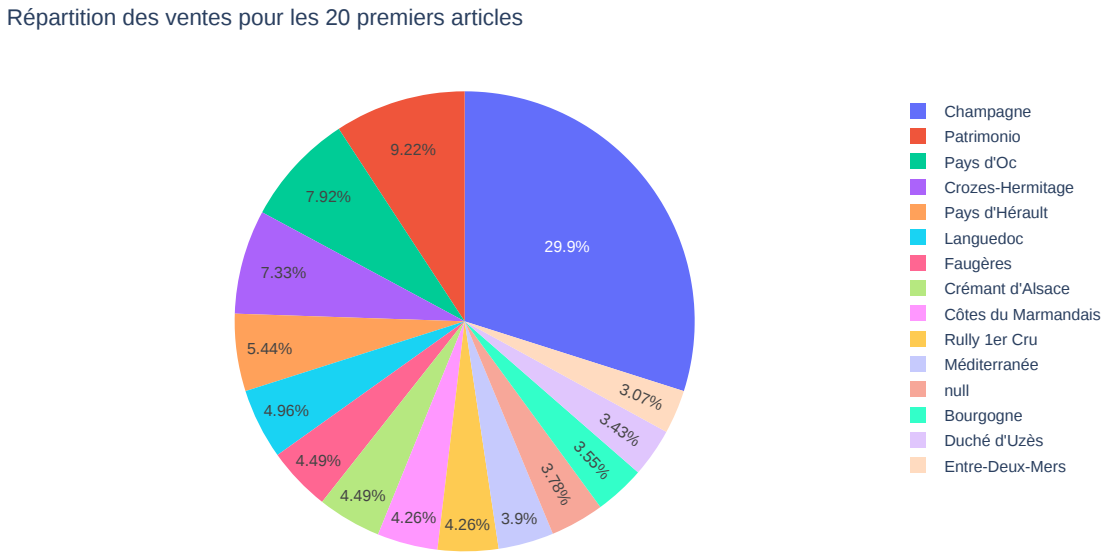


In [226…
```
top_20_articles = df_merge.nlargest(20, 'total_sales')[['product_id', 'total_sales']]

# Créez un graphique à secteurs
fig = px.pie(top_20_articles, names='product_id', values='total_sales', title='Répartition des ventes pour les 20 premiers articles')
fig.show()
```



Loading [MathJax]/extensions/Safe.js

In [227…
```
top_20_articles = df_merge.nlargest(20, 'total_sales')[['Appellation', 'total_sales']]

# Créez un graphique à secteurs
fig = px.pie(top_20_articles, names='Appellation', values='total_sales', title='Répartition des ventes pour les 20 premiers articles')
fig.show()
```



In [228…
```
#"#Créer une colonne calculant la part en quantité de la ligne dans le dataset\n",
df_merge['Part_en_Quantite'] = df_merge['total_sales'] / df_merge['total_sales'].sum()
```

In [229…
```
#Créer une colonne réalisant la somme cumulative de la colonne précedemment créée\n",
df_merge['Somme_Cumulative_Quantite'] = df_merge['Part_en_Quantite'].cumsum()
```

In [230…
```
#Grâce au deux colonnes créées précedemment, calculer le nombre d'articles représentant 80% des ventes en quantité
# Tri du DataFrame par ordre décroissant de Somme_Cumulative_Quantite
df_merge = df_merge.sort_values(by='Somme_Cumulative_Quantite', ascending=False)
```

In [231…
```
# Calculez le seuil de 80% des ventes en quantité
seuil_80_percent = df_merge['total_sales'].sum() * 0.8
total_quantite = 0
nombre_articles_80_percent = 0
```

In [232…
```
#Parcourez le DataFrame pour calculer le nombre d'articles
for index, row in df_merge.iterrows():
    total_quantite += row['total_sales']
    nombre_articles_80_percent += 1
    if total_quantite >= seuil_80_percent:
        break

# Affichez le résultat
print("Nombre d'articles représentant 80% des ventes en quantité : ", nombre_articles_80_percent)
```

```
Nombre d'articles représentant 80% des ventes en quantité :  701
```

In [233…
```
nombre_total_articles = df_merge.shape[0]  # Nombre de lignes dans le DataFrame
```

Loading [MathJax]/extensions/Safe.js

```
print("Nombre total d'articles :", nombre_total_articles)
```

```
Nombre total d'articles : 712
```

In [235…
```
# Nombre total d'articles dans le catalogue
nombre_total_articles = 1424

# Nombre d'articles représentant 80% des ventes en quantité
nombre_articles_80_percent = 1402

# Calcul de la proportion
proportion = (nombre_articles_80_percent / nombre_total_articles) * 100

# Afficher la proportion
print("Proportion d'articles représentant 80% des ventes en quantité par rapport au catalogue : {:.2f}%".format(proportion))
```

```
Proportion d'articles représentant 80% des ventes en quantité par rapport au catalogue : 98.46%
```

In [237…
```
nom_fichier_excel = "df_merge.xlsx"

df_merge.to_excel(nom_fichier_excel, index=False)


print("Le DataFrame a été sauvegardé dans", nom_fichier_excel)
```

```
Le DataFrame a été sauvegardé dans df_merge.xlsx
```

In [238…
```
import os
print(os.getcwd())
```

```
C:\Users\pc
```

In [ ]:
```

```