In [2]:
```
#Importation de la librairie Pandas
import pandas as pd
import numpy as np
import datetime as dt
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
import missingno as msno
import scipy.stats as st
```

In [3]:
```
# import et lecture export ERP
erp_df = pd.read_excel('Fichier_erp.xlsx')
erp_df.head()
```

```
c:\Python311\Lib\site-packages\openpyxl\worksheet\_reader.py:329: UserWarning: Unknown e
xtension is not supported and will be removed
  warn(msg)
```

Out[3]:

	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 [4]:
```
# import et lecture export BDD Web
web_df = pd.read_excel('Fichier_web.xlsx')
web_df.head()
```

```
c:\Python311\Lib\site-packages\openpyxl\worksheet\_reader.py:329: UserWarning: Unknown e
xtension is not supported and will be removed
  warn(msg)
```

Out[4]:

	sku	virtual	downloadable	rating_count	average_rating	total_sales	tax_status	tax_class	post_author
0	16004	0	0	0	0.0	5.0	NaN	NaN	2.0
1	NaN	0	0	0	NaN	NaN	NaN	NaN	NaN
2	15075	0	0	0	0.0	3.0	taxable	NaN	2.0
3	16209	0	0	0	0.0	6.0	taxable	NaN	2.0
4	15763	0	0	0	0.0	1.0	NaN	NaN	2.0

5 rows × 28 columns

In [5]:
```
# lecture du dataset liaisons
liaisons_ = pd.read_excel('fichier_liaison.xlsx')
liaisons_
```

```
c:\Python311\Lib\site-packages\openpyxl\worksheet\_reader.py:329: UserWarning: Unknown e
xtension is not supported and will be removed
  warn(msg)
```

Out[5]:

	product_id	id_web
0	3847	15298
1	3849	15296
2	3850	15300
3	4032	19814
4	4039	19815
...	...	...
820	7203	NaN
821	7204	NaN
822	7247	13127-1
823	7329	14680-1
824	7338	16230

825 rows × 2 columns

In [6]:
```
# dimensions du fichier ERP
print(f"The ERP dataset has {erp_df.shape[0]} rows and {erp_df.shape[1]} columns")
```

```
The ERP dataset has 825 rows and 5 columns
```

In [7]:
```
# dimensions du fichier WEB
print(f"The WEB dataset has {web_df.shape[0]} rows and {web_df.shape[1]} columns")
```

```
The WEB dataset has 1513 rows and 28 columns
```

In [8]:
```
# dimensions du fichier ERP
print(f"The liaisons dataset has {liaisons_.shape[0]} rows and {liaisons_.shape[1]} colu
```

```
The liaisons dataset has 825 rows and 2 columns
```

In [9]:
```
web_df = web_df.rename(columns={'sku': 'id_web'})
web_df.sample(20)
```

Out[9]:

	id_web	virtual	downloadable	rating_count	average_rating	total_sales	tax_status	tax_class	post_auth
526	16057	0	0	0	0.0	0.0	taxable	NaN	2
818	15440	0	0	0	0.0	0.0	NaN	NaN	2
1012	12585	0	0	0	0.0	12.0	taxable	NaN	2
1155	13754	0	0	0	0.0	0.0	taxable	NaN	2

1116	15688	0	0	0	0.0	0.0	taxable	NaN	2
176	14944	0	0	0	0.0	3.0	taxable	NaN	2
998	15677	0	0	0	0.0	0.0	NaN	NaN	2
813	12641	0	0	0	0.0	0.0	NaN	NaN	2
773	16296	0	0	0	0.0	0.0	taxable	NaN	2
1146	15668	0	0	0	0.0	3.0	NaN	NaN	2
364	15185	0	0	0	0.0	0.0	taxable	NaN	2
1017	16256	0	0	0	0.0	8.0	NaN	NaN	2
230	15240	0	0	0	0.0	0.0	NaN	NaN	2
1377	15399	0	0	0	0.0	0.0	taxable	NaN	2
259	15428	0	0	0	0.0	7.0	NaN	NaN	2
193	NaN	0	0	0	NaN	NaN	NaN	NaN	Na
564	14596	0	0	0	0.0	0.0	taxable	NaN	2
565	11996	0	0	0	0.0	0.0	taxable	NaN	2
1199	15704	0	0	0	0.0	0.0	taxable	NaN	2

493	NaN	0	0	0	NaN	NaN	NaN	NaN	Na

20 rows × 28 columns

In [10]:
```
# types des colonnes
erp_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 825 entries, 0 to 824
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   product_id      825 non-null    int64
 1   onsale_web      825 non-null    int64
 2   price           825 non-null    float64
 3   stock_quantity  825 non-null    int64
 4   stock_status    825 non-null    object
dtypes: float64(1), int64(3), object(1)
memory usage: 32.4+ KB
```

In [11]:
```
# types des colonnes
web_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1513 entries, 0 to 1512
Data columns (total 28 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   id_web                 1428 non-null   object
 1   virtual                1513 non-null   int64
 2   downloadable           1513 non-null   int64
 3   rating_count           1513 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: 331.1+ KB
```

In [12]:
```
liaisons_.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
```

Vérifier les doublons

In [13]:
```
#liaisons_.loc[liaisons_['id_web'].duplicated(keep=False),:]
len(liaisons_.loc[liaisons_['product_id'].duplicated(keep=False),:])
```

Out[13]:
```
0
```

In [14]:
```
#liaisons_.loc[liaisons_['id_web'].duplicated(keep=False),:]
len(liaisons_.loc[liaisons_['id_web'].duplicated(keep=False),:])
```

Out[14]:
```
91
```

In [15]:
```
len(erp_df.loc[erp_df['product_id'].duplicated(keep=False),:])
```

Out[15]:
```
0
```

In [16]:
```
len(web_df.loc[web_df['id_web'].duplicated(keep=False),:])
```

Out[16]:
```
1513
```

In [17]:
```
comparison = web_df.loc[web_df['id_web'] == 16209]
comparison[['id_web','total_sales','post_name','tax_status','post_type','post_mime_type'
```

Out[17]:

	id_web	total_sales	post_name	tax_status	post_type	post_mime_type
3	16209	6.0	maurel-cabardes-tradition-2017	taxable	product	NaN
13	16209	6.0	maurel-cabardes-tradition-2017	NaN	attachment	image/jpeg

In [18]:
```
# Vérifier si les valeurs 'string' sont dans les 2 exports
differentes_valeurs_liaisons = liaisons_['id_web'].unique()

# Affichez les différentes valeurs
print(differentes_valeurs_liaisons)
```

```
[15298 15296 15300 19814 19815 15303 14975 16042 14980 16041 15269 14977
 16044 16043 16449 16045 16030 13127 19816 nan 16029 16039 16318 16275
 16498 16320 16319 15966 15022 15967 15490 16416 11862 15444 15953 12045
 13074 15941 16069 13072 15440 13435 13078 13117 16296 16014 16462 16013
 16180 15676 16120 15564 15675 15378 15813 13416 14905 15767 16505 15683
 16504 15787 14800 15353 15382 15339 11668 13209 15341 13217 304 11641
 1662 1360 15648 1364 7086 1366 15140 16238 16237 15141 14944 14941 14751
 16093 15668 15373 15375 14474 15482 13453 15075 16124 15785 15784 15786
 14332 16210 16211 16209 15629 15583 16160 16166 15783 16560 15747 15746
 16190 16189 16265 16191 16263 15605 16529 15441 13032 16256 16322 16295
 15656 15655 15415 15414 15413 16023 16024 15720 15714 15717 15718 15480
 15213 15672 12599 15758 15829 15759 16585 15306 16497 15261 12657 15403
 15461 16269 13905 16567 15436 14725 15310 15770 16097 15428 15033 16317
 15032 6616 12203 14253 12476 14485 14945 15662 15663 15664 15665 15136
 16537 16307 16244 15839 13460 13089 12942 14864 14527 14865 15690 16330
 16154 16153 16066 16065 15292 13771 16246 16501 16578 15567 16553 13172
 15120 15949 15946 7818 13599 4679 12586 12588 15940 12587 12589 12585
 9562 13854 13853 11585 11467 11586 13765 13766 11587 9636 12639 12641
 12640 14768 3506 3510 3507 13230 7819 3509 15426 15621 15457 15065 13604
 12857 14785 15476 14000 15478 15475 16151 15659 15147 15660 15148 15149
 15146 15145 15801 15452 15038 15030 15875 16186 14371 10459 14372 11049
 15850 15849 812 807 805 802 2534 793 791 2179 804 41 798 2361 15848 16525
```

```
16262 16261 15206 11849 13515 13514 13516 10814 11847 13517 16081 15402
15404 13647 14657 16053 15525 15527 14676 16057 16056 13762 15280 15282
15281 15283 15934 15933 15575 16239 14451 16324 15582 13736 13659 15465
15004 14699 15349 15466 14700 10775 16119 15667 14746 15361 15196 15657
15658 15670 16527 16513 15880 15879 16010 14950 16540 15729 38 5646 8344
15576 16138 14366 13412 12601 14632 15315 13627 14184 15429 16132 14680
15859 16229 14302 16072 14300 13096 16564 13754 15734 15448 15881 15731
15316 15732 14599 15733 15730 12771 3568 14506 15811 16342 16292 15307
16047 16255 15154 16274 16148 14360 16149 16289 14981 15773 15776 16037
16038 15807 15952 15808 16062 16063 14802 13052 14805 14220 14374 14395
15614 13809 15612 13814 15613 15615 15533 15531 15530 15608 15586 15928
16276 16277 15456 15425 15047 15927 16155 16280 9937 16281 15554 15106
16283 13379 15338 15337 'bon-cadeau-25-euros' 15737 15958 16515 16586
11225 16004 14756 16005 14930 13313 15229 14507 14509 14508 15868 14581
14580 15869 15871 15870 12791 11602 15073 14839 15272 14696 15630 11996
13914 13913 11997 531 13531 15711 15713 15715 15346 15345 15344 15755
15677 14561 16022 16011 3383 14149 13904 14141 12494 15462 15095 14626
12496 12315 15649 14809 15155 12194 16328 14469 16034 14679 15526 16305
16306 15138 15753 15756 16131 16130 16129 14712 15481 16146 14648 14192
15860 15863 15861 15862 15864 14819 14828 14827 15202 13959 13965 13958
13957 13520 13969 14715 19820 19821 15748 19822 16192 14730 14729 8463
13982 15944 15930 14912 15945 14915 14855 14856 15923 14845 14844 15921
15922 12366 8365 12365 14647 15812 14661 16304 15797 16094 14736 11736
15036 15360 15674 13557 15035 16121 14241 14982 15026 15116 15369 15566
16003 15127 15125 14323 15631 16147 7033 11258 13849 15818 15179 15185
15183 15254 15178 15184 15180 15264 14338 15561 16213 14692 13291 13895
15688 14461 14689 11277 15399 13572 14955 13567 15471 15080 14429 15238
15237 14600 15241 11933 15240 15325 15328 15329 15775 15774 14983 13910
16539 15910 12339 12869 14095 14099 15856 12881 15857 12882 15227 10014
14265 14774 14775 14773 15343 15351 16323 523 15432 16472 14379 15609
14377 15895 13577 15577 15766 15892 16326 15574 13662 11669 13215 13211
15342 15318 13073 16159 16264 14899 15134 16133 16028 15951 15487 15486
15489 15529 14089 14100 14092 14090 14106 14101 14797 15201 14923 14573
14569 14570 15834 14596 15126 14604 16565 16580 16077 13996 15072 11601
12790 15070 16096 7032 15324 15162 15161 15163 16273 16247 15654 15710
15745 15678 15810 15779 15707 15705 15706 15704 15473 15479 15647 15769
15434 15764 16071 15781 16031 15539 16046 15204 15205 15790 15791 15792
15793 15795 15794 15763 16152 15661 16068 16067 8193 16144 15256 15735
14897 15736 15740 15845 15741 16135 15891 15887 '13127-1' '14680-1' 16230]
```

In [19]:
```python
# Filtrer les chaînes de caractères
chaines = [str(val) for val in web_df['id_web'].unique() if isinstance(val, str)]

# Afficher la liste des chaînes de caractères
print(chaines)
```

```
['bon-cadeau-25-euros', '13127-1']
```

In [20]:
```python
# Filtrer les chaînes de caractères
chaines = [str(val) for val in liaisons_['id_web'].unique() if isinstance(val, str)]

# Afficher la liste des chaînes de caractères
print(chaines)
```

```
['bon-cadeau-25-euros', '13127-1', '14680-1']
```

In [21]:
```python
# Vérifier si les valeurs 'string' sont dans les 2 exports
differentes_valeurs_id_web = web_df['id_web'].unique()

# Affichez les différentes valeurs
print(len(differentes_valeurs_id_web))
```

```
715
```

In [22]:
```python
differentes_valeurs_product_id = erp_df['product_id'].unique()
```

```
# Affichez les différentes valeurs
print(len(differentes_valeurs_product_id))
```

```
825
```

In [23]:
```
# méthode .isna() pour obtenir la somme des valeurs manquantes pour chaque colonne
web_df.isna().sum()
```

Out[23]:
```
id_web                   85
virtual                   0
downloadable              0
rating_count              0
average_rating           83
total_sales              83
tax_status              797
tax_class              1513
post_author              83
post_date                83
post_date_gmt            83
post_content           1513
post_title               83
post_excerpt            797
post_status              83
comment_status           83
ping_status              83
post_password          1513
post_name                83
post_modified            83
post_modified_gmt        83
post_content_filtered  1513
post_parent              83
guid                     83
menu_order               83
post_type                83
post_mime_type          799
comment_count            83
dtype: int64
```

In [24]:
```
# pour afficher uniquement les variables qui ont des valeurs manquantes
nb_na = web_df.isnull().sum()
nb_na[nb_na>0]
```

Out[24]:
```
id_web                   85
average_rating           83
total_sales              83
tax_status              797
tax_class              1513
post_author              83
post_date                83
post_date_gmt            83
post_content           1513
post_title               83
post_excerpt            797
post_status              83
comment_status           83
ping_status              83
post_password          1513
post_name                83
post_modified            83
post_modified_gmt        83
post_content_filtered  1513
post_parent              83
guid                     83
menu_order               83
post_type                83
post_mime_type          799
```

```
comment_count              83
dtype: int64
```

In [25]:
```
# méthode .isna() pour obtenir la somme des valeurs manquantes pour chaque colonne
erp_df.isna().sum()
```

Out[25]:
```
product_id        0
onsale_web        0
price             0
stock_quantity    0
stock_status      0
dtype: int64
```

In [26]:
```
# méthode .isna() pour obtenir la somme des valeurs manquantes pour chaque colonne
liaisons_.isna().sum()
```

Out[26]:
```
product_id     0
id_web        91
dtype: int64
```

In [27]:
```
# pour afficher uniquement les variables qui ont des valeurs manquantes
nb_na = liaisons_.isnull().sum()
nb_na[nb_na>0]
```

Out[27]:
```
id_web    91
dtype: int64
```

In [28]:
```
# filtrer web_df sur 'product'
web_df = web_df[web_df['post_type'] == 'product']
web_df.head()
```

Out[28]:

	id_web	virtual	downloadable	rating_count	average_rating	total_sales	tax_status	tax_class	post_author
2	15075	0	0	0	0.0	3.0	taxable	NaN	2.0
3	16209	0	0	0	0.0	6.0	taxable	NaN	2.0
5	13895	0	0	0	0.0	0.0	taxable	NaN	2.0
6	12857	0	0	0	0.0	0.0	taxable	NaN	2.0
9	14106	0	0	0	0.0	0.0	taxable	NaN	2.0

5 rows × 28 columns

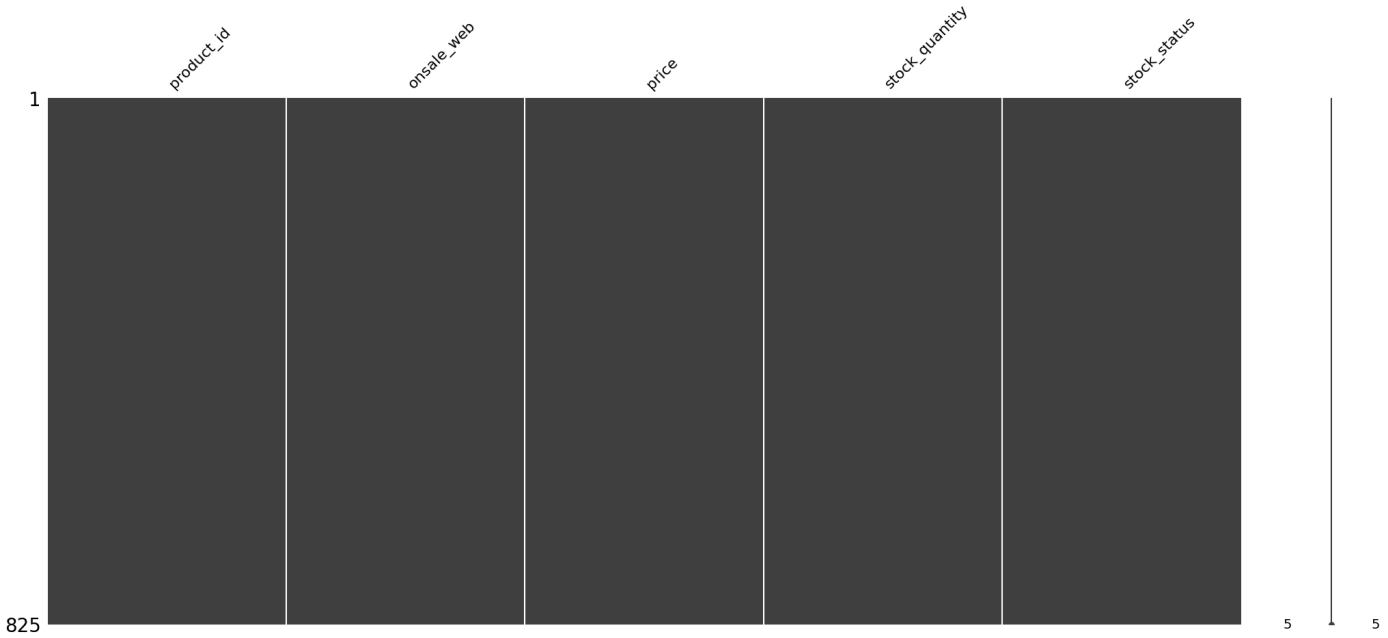
In [29]:
```
# pour afficher uniquement les variables qui ont des valeurs manquantes
nb_na = web_df.isnull().sum()
nb_na[nb_na>0]
```

Out[29]:
```
id_web                   2
tax_class              716
```

```
post_content             716
post_password            716
post_content_filtered    716
post_mime_type           716
dtype: int64
```

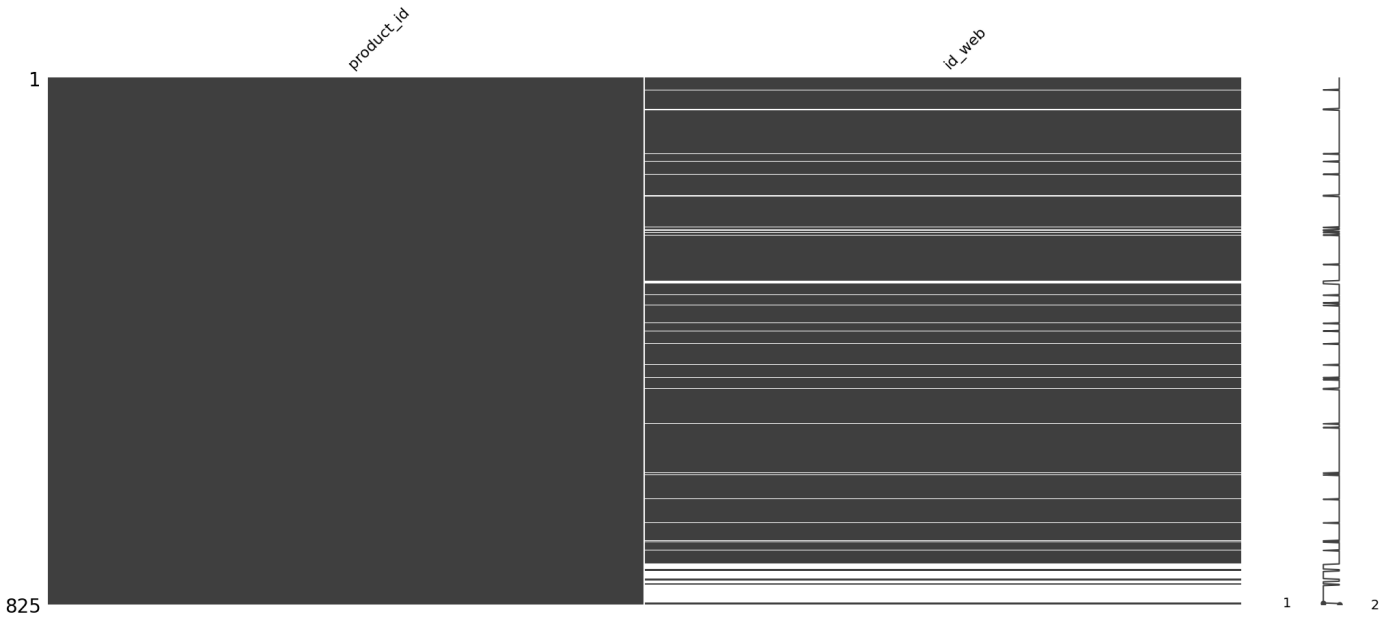
In [30]:
```
msno.matrix(erp_df)
```

Out[30]:
```
<Axes: >
```



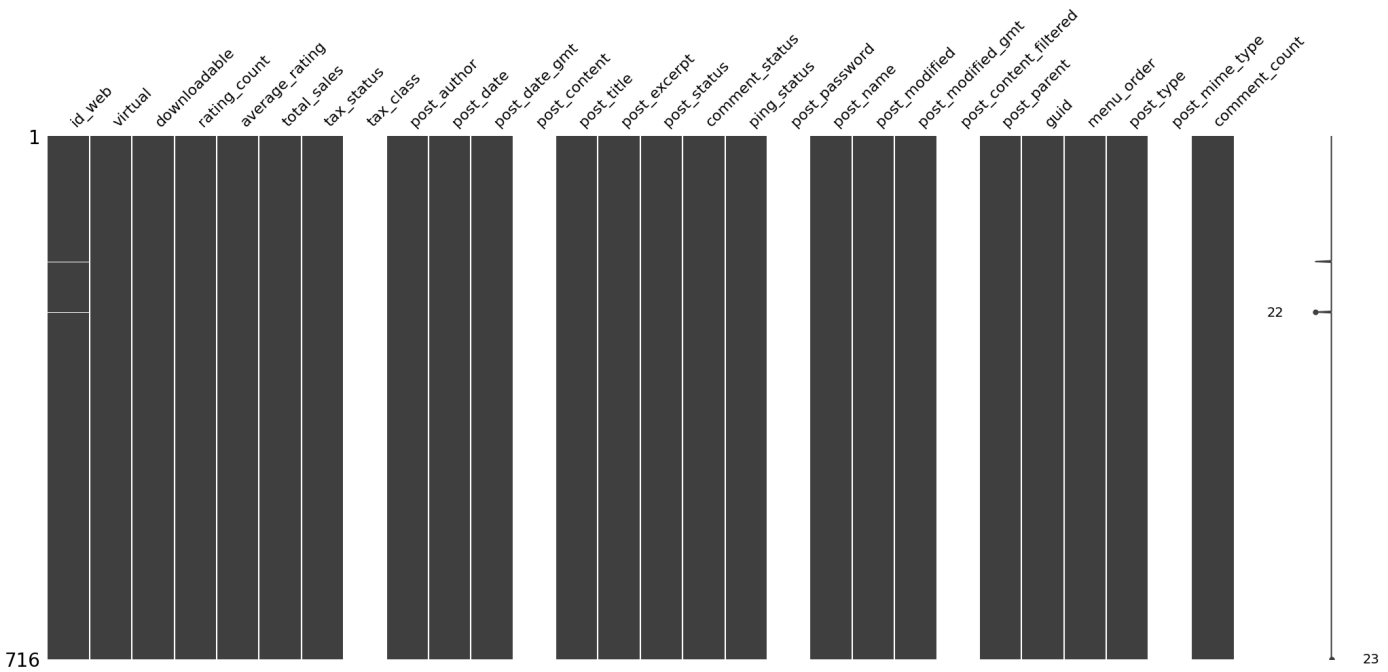
In [31]:
```
msno.matrix(liaisons_)
```

Out[31]:
```
<Axes: >
```



In [32]:
```
msno.matrix(web_df)
```

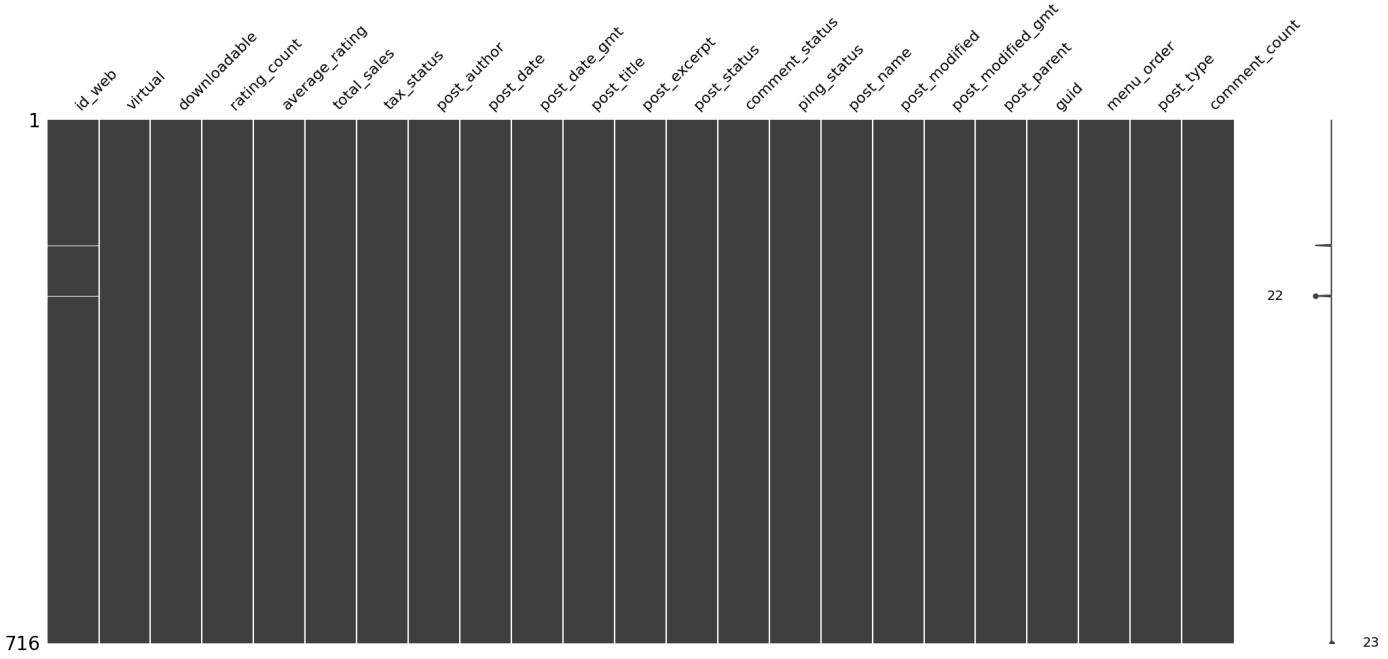
Out[32]:
```
<Axes: >
```



In [33]:
```
# suppression des colonnes vides
web_df = web_df.drop(['tax_class','post_content','post_password','post_content_filtered'
```

In [34]:
```
# Vérificiation visuelle avec le package missingno
msno.matrix(web_df)
```

Out[34]:
```
<Axes: >
```



In [35]:
```
web_df.loc[web_df['id_web'].isnull(),:]
```

Out[35]:

	id_web	virtual	downloadable	rating_count	average_rating	total_sales	tax_status	post_author	post_dat
353	NaN	0	0	0	0.0	3.0	taxable	2.0	2018-0 0 11:23:4
488	NaN	0	0	0	0.0	10.0	taxable	2.0	2018-0 3 12:07:2

2 rows × 23 columns

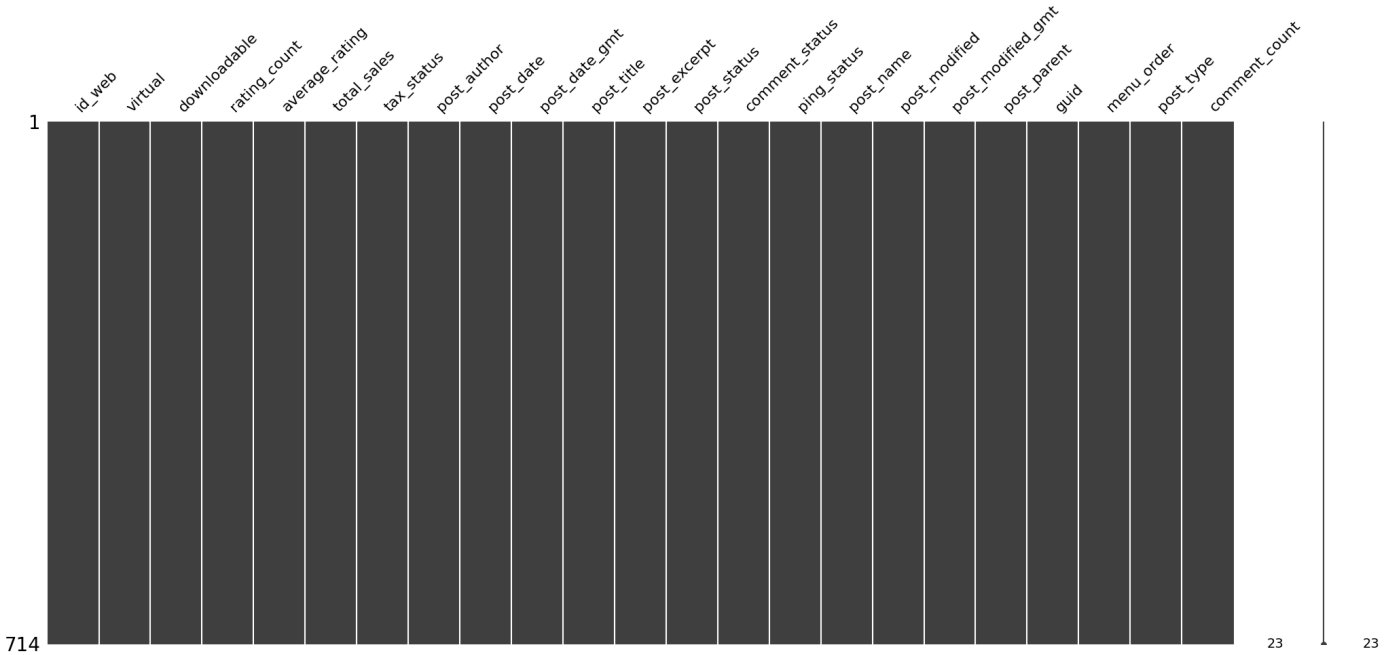
In [36]:
```
web_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 716 entries, 2 to 1510
Data columns (total 23 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id_web             714 non-null    object
 1   virtual            716 non-null    int64
 2   downloadable       716 non-null    int64
 3   rating_count       716 non-null    int64
 4   average_rating     716 non-null    float64
 5   total_sales        716 non-null    float64
 6   tax_status         716 non-null    object
 7   post_author        716 non-null    float64
 8   post_date          716 non-null    datetime64[ns]
 9   post_date_gmt      716 non-null    datetime64[ns]
 10  post_title         716 non-null    object
 11  post_excerpt       716 non-null    object
 12  post_status        716 non-null    object
 13  comment_status     716 non-null    object
 14  ping_status        716 non-null    object
 15  post_name          716 non-null    object
 16  post_modified      716 non-null    datetime64[ns]
 17  post_modified_gmt  716 non-null    datetime64[ns]
 18  post_parent        716 non-null    float64
 19  guid               716 non-null    object
 20  menu_order         716 non-null    float64
 21  post_type          716 non-null    object
 22  comment_count      716 non-null    float64
dtypes: datetime64[ns](4), float64(6), int64(3), object(10)
memory usage: 134.2+ KB
```

In [37]:
```
web_df = web_df.dropna(subset=['id_web'])
```

In [38]:
```
msno.matrix(web_df)
```

Out[38]:
```
<Axes: >
```



Jointure des 2 exports : ERP et CMS

In [39]:
```
merging_erp_web = erp_df.merge(liaisons_, how='inner', on='product_id')
```

In [40]:
```
merging_erp_web2 = merging_erp_web.merge(web_df, how='inner', on='id_web')
```

In [41]:
```
merging_erp_web2.head()
```

Out[41]:

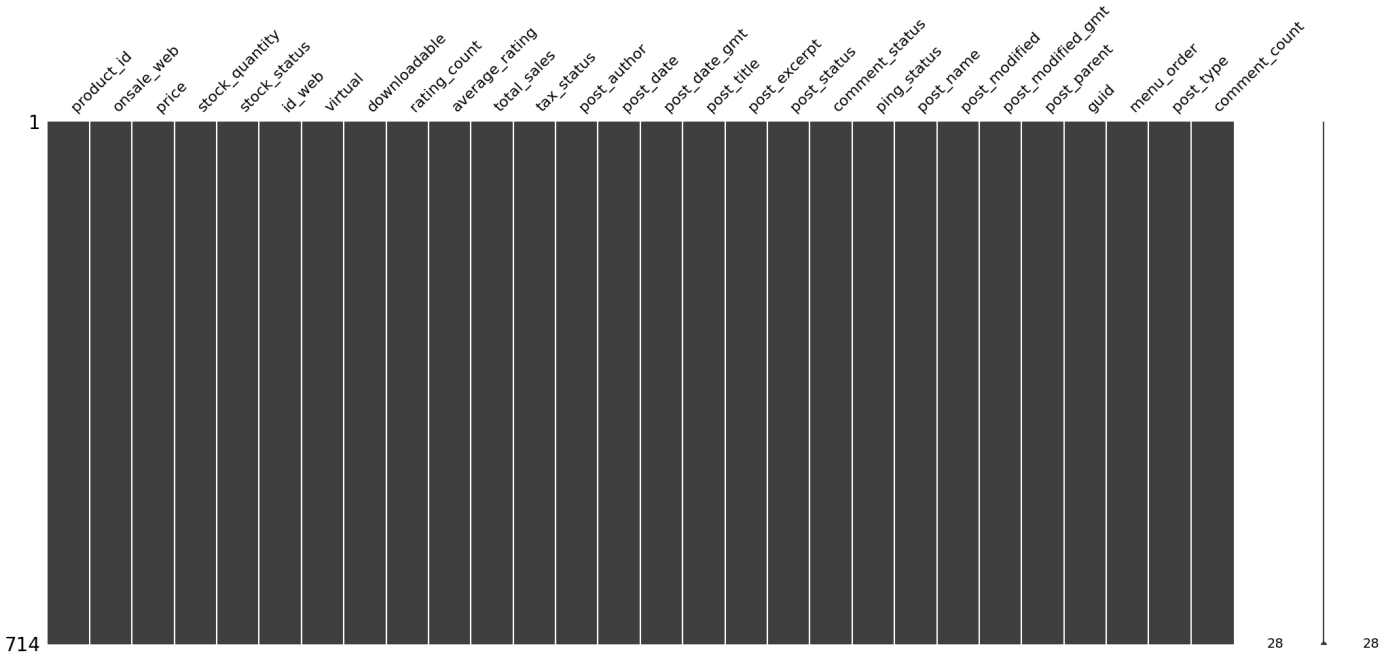
	product_id	onsale_web	price	stock_quantity	stock_status	id_web	virtual	downloadable	rating_count	av
0	3847	1	24.2	0	outofstock	15298	0	0	0	
1	3849	1	34.3	0	outofstock	15296	0	0	0	
2	3850	1	20.8	0	outofstock	15300	0	0	0	
3	4032	1	14.1	0	outofstock	19814	0	0	0	
4	4039	1	46.0	0	outofstock	19815	0	0	0	

5 rows × 28 columns

Visuel du dataframe nettoyé avec jointure

In [42]:
```
msno.matrix(merging_erp_web2)
```

Out[42]:
```
<Axes: >
```



In [43]:
```
merging_erp_web2.shape
```

Out[43]:
```
(714, 28)
```

Calcul du chiffres d'affaires = 70 568,60 EUR

In [44]:
```
merging_erp_web2["C.A."] = (merging_erp_web2['price'] * merging_erp_web2['total_sales'])
```

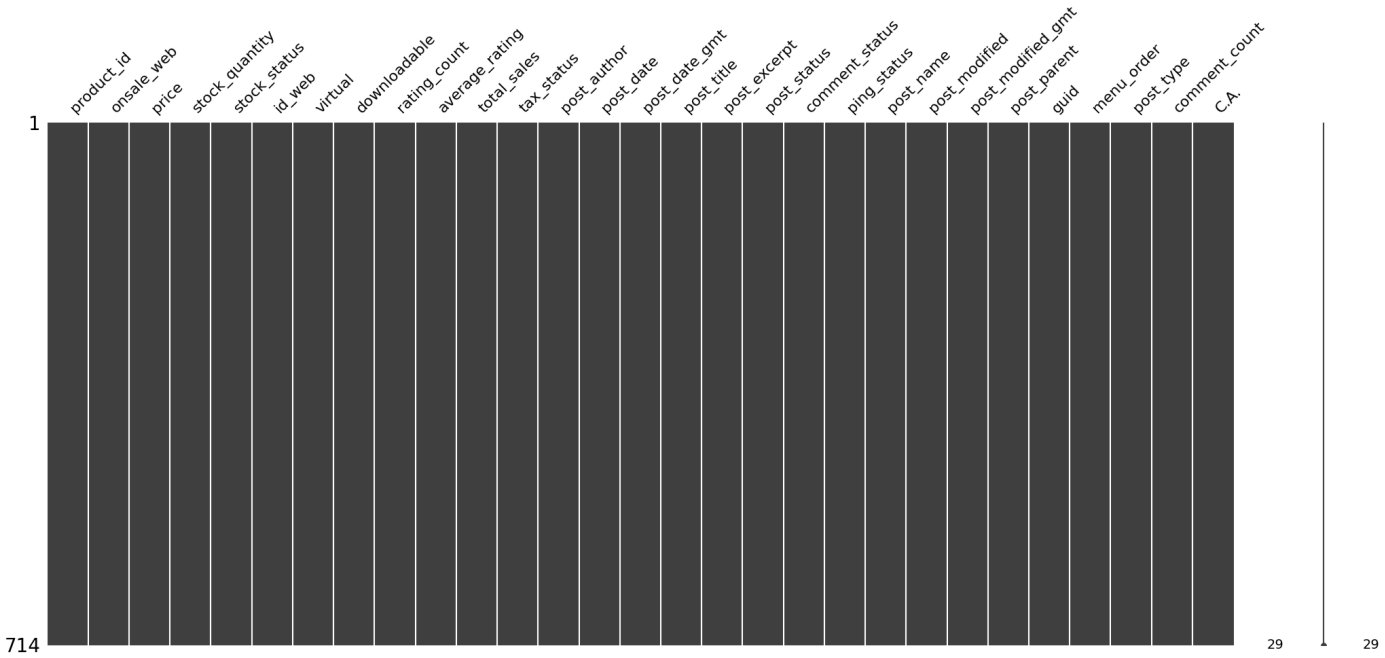
In [45]:
```
CA_total = merging_erp_web2['C.A.'].sum().round(1)
CA_total
```

Out[45]:
```
70568.6
```

Dataframe avec la colonne CA

In [46]:
```
msno.matrix(merging_erp_web2)
```

Out[46]:
```
<Axes: >
```



Chiffres d'affaires par produit trié par ordre décroissant

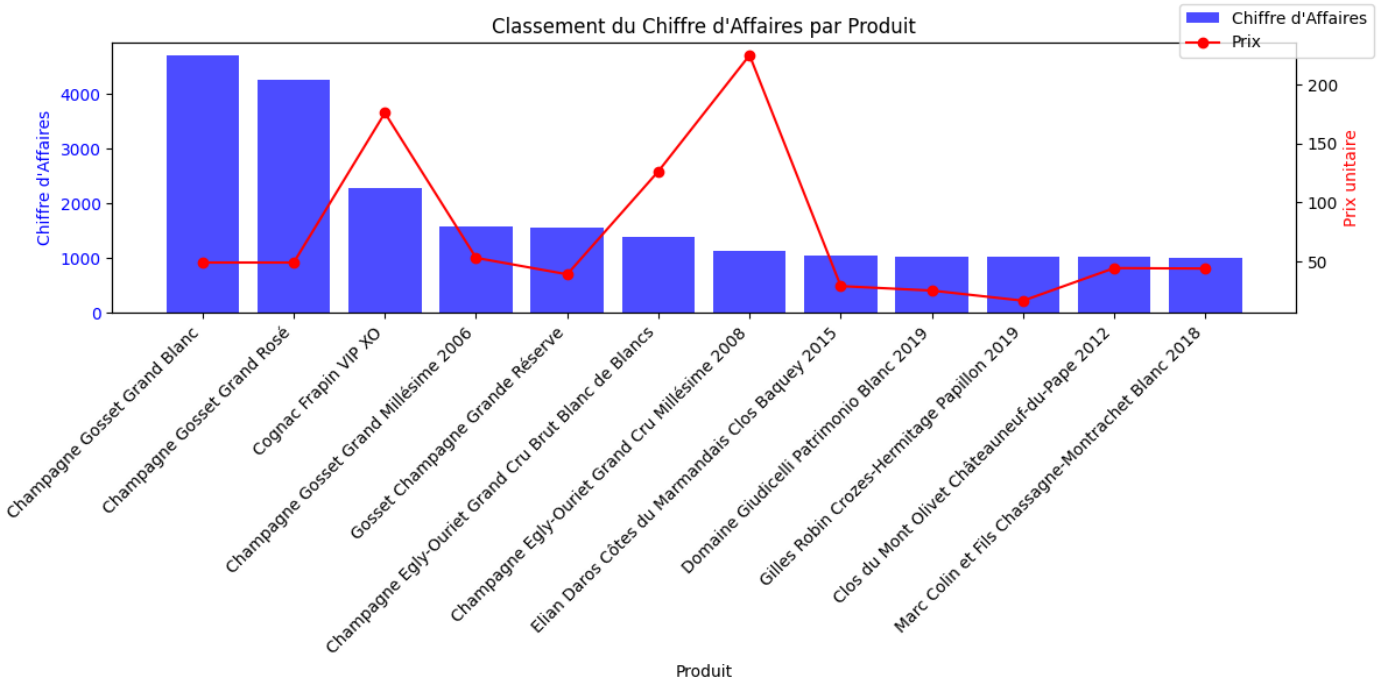
In [47]:
```
ca_par_produit = merging_erp_web2[['post_name','price','C.A.']]
ca_par_produit.sort_values(by='C.A.', ascending=False).head(20)
```

Out[47]:

	post_name	price	C.A.
194	champagne-gosset-grand-blanc-de-blanc	49.0	4704.0
71	champagne-gosset-grand-rose	49.0	4263.0
218	cognac-frapin-vip-xo	176.0	2288.0
70	champagne-gosset-grand-millesime-2006	53.0	1590.0
69	gosset-champagne-grande-reserve	39.0	1560.0
201	champagne-egly-ouriet-grand-cru-brut-blanc-de-...	126.5	1391.5
199	champagne-egly-ouriet-grand-cru-millesime-2008	225.0	1125.0
79	elian-daros-cotes-du-marmandais-clos-baquey-2015	29.0	1044.0
651	domaine-giudicelli-patrimonio-blanc-2019	25.2	1033.2
30	gilles-robin-crozes-hermitage-papillon-2019	16.6	1029.2

17	clos-du-mont-olivet-chateauneuf-du-pape-2012	44.3	1018.9
228	marc-colin-et-fils-chassagne-montrachet-blanc-...	43.9	1009.7
373	jacqueson-rully-blanc-1er-cru-la-pucelle-2018	27.9	1004.4
445	albert-mann-pinot-noir-grand-h-2017	59.9	958.4
652	domaine-giudicelli-patrimonio-rouge-2016	25.2	932.4
390	alain-graillot-crozes-hermitage-rouge-la-guira...	37.2	892.8
605	albert-mann-pinot-noir-clos-de-la-faille-2017	48.5	824.5
18	clos-du-mont-olivet-chateauneuf-du-pape-papet-...	71.6	716.0
182	hauvette-baux-provence-amethyste-2017	69.8	698.0
381	domaine-des-croix-corton-charlemagne-grand-cru...	137.0	685.0

In [48]:
```
# Données
produits = [
    "Champagne Gosset Grand Blanc",
    "Champagne Gosset Grand Rosé",
    "Cognac Frapin VIP XO",
    "Champagne Gosset Grand Millésime 2006",
    "Gosset Champagne Grande Réserve",
    "Champagne Egly-Ouriet Grand Cru Brut Blanc de Blancs",
    "Champagne Egly-Ouriet Grand Cru Millésime 2008",
    "Elian Daros Côtes du Marmandais Clos Baquey 2015",
    "Domaine Giudicelli Patrimonio Blanc 2019",
    "Gilles Robin Crozes-Hermitage Papillon 2019",
    "Clos du Mont Olivet Châteauneuf-du-Pape 2012",
    "Marc Colin et Fils Chassagne-Montrachet Blanc 2018"
]
prix = [49.0, 49.0, 176.0, 53.0, 39.0, 126.5, 225.0, 29.0, 25.2, 16.6, 44.3, 43.9]
chiffre_affaires = [4704.0, 4263.0, 2288.0, 1590.0, 1560.0, 1391.5, 1125.0, 1044.0, 1033

# Créer un DataFrame
df = pd.DataFrame({'Produit': produits, 'Prix': prix, 'Chiffre d\'Affaires': chiffre_aff

# Faire pivoter les noms de produits pour les rendre lisibles
fig, ax = plt.subplots(figsize=(12, 6))
ax.bar(df.index, df['Chiffre d\'Affaires'], color='b', alpha=0.7, label='Chiffre d\'Affa
ax.set_xlabel('Produit')
ax.set_ylabel('Chiffre d\'Affaires', color='b')
ax.tick_params(axis='y', labelcolor='b')
ax.set_xticks(df.index)
ax.set_xticklabels(df['Produit'], rotation=45, ha='right')

# Ajouter les prix comme des points sur le graphique
ax2 = ax.twinx()
ax2.plot(df.index, df['Prix'], color='r', marker='o', label='Prix')
ax2.set_ylabel('Prix unitaire', color='r')

# Titre et légende
plt.title('Classement du Chiffre d\'Affaires par Produit')
fig.tight_layout()
fig.legend(loc='upper right')

# Afficher le graphique
plt.show()
```



In [49]:
```
ca_par_produit.dtypes
```

Out[49]:
```
post_name     object
price        float64
C.A.         float64
dtype: object
```

In [50]:
```
gosset_grand_blanc = ca_par_produit[ca_par_produit['C.A.'] == 4704]
gosset_grand_rose = ca_par_produit[ca_par_produit['C.A.'] == 4263]

# Convertir la colonne 'C.A.' de gosset_grand_rose en float
gosset_grand_rose['C.A.'] = gosset_grand_rose['C.A.'].astype(float)

# Calcule de la part des champagnes gosset par rapport au CA total
part_du_CA_total = (((gosset_grand_blanc['C.A.'].values + gosset_grand_rose['C.A.'].valu

# Afficher la part du CA total
print(f"la part du chiffres d\'affaires des champagnes Gosset Grand Blanc et Grand Rosé
```

```
la part du chiffres d'affaires des champagnes Gosset Grand Blanc et Grand Rosé est de [1
2.7] %.
C:\Users\nbous\AppData\Local\Temp\ipykernel_18332\2820164503.py:5: SettingWithCopyWarnin
g:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  gosset_grand_rose['C.A.'] = gosset_grand_rose['C.A.'].astype(float)
```

In [51]:
```
# Trier par ordre décroissant de prix
best_ca = ca_par_produit.sort_values(by='C.A.', ascending=False).head(10)


# Création du graphique à barres avec la palette de couleurs bordeaux
fig = px.bar(best_ca, x='post_name', y='C.A.', color='C.A.',
             labels={'post_name': 'Vin', 'C.A.': 'chiffres d\'affaires'},
             color_discrete_sequence=px.colors.qualitative.Set3)  # Utilisation de la pa

# Personnalisation du graphique
fig.update_layout(
    title='Chiffres d\'affaires classé par produit',
    xaxis_title='vin',
```

```
    yaxis_title='CA',
    legend_title='Légende',
    xaxis_tickangle=-45,
    showlegend=False
)

# Ajustement de la taille du graphique et des marges
fig.update_layout(
    autosize=False,
    width=1000,
    height=600,
    margin=dict(l=50, r=50, b=100, t=100)  # Ajustement des marges
)

# Affichage du graphique
fig.show()
```

Analyse univariée : Prix et Outliers

Statistiques descriptives

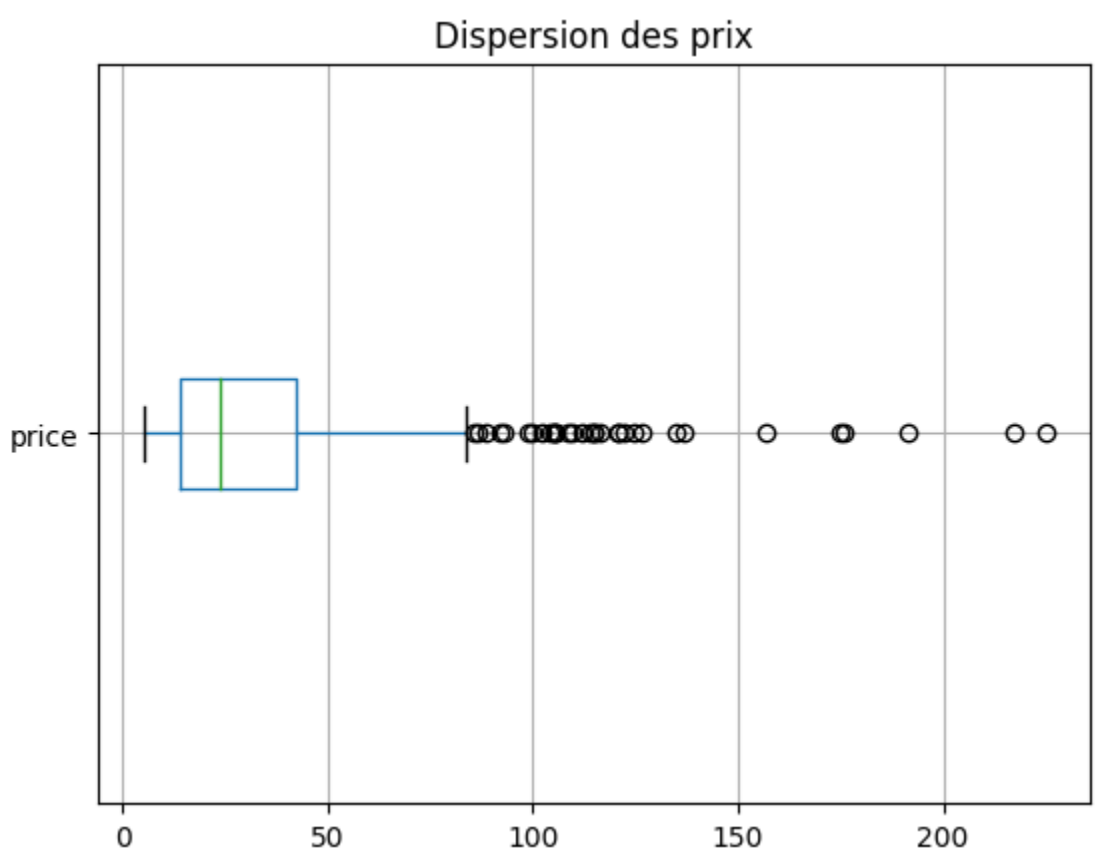
In [52]:
```
merging_erp_web2[['price']].describe().round(2)
```

Out[52]:

	price
count	714.00
mean	32.49
std	27.81
min	5.20
25%	14.10
50%	23.55
75%	42.18
max	225.00

Distribution des prix

In [84]:
```
merging_erp_web2.boxplot(column="price", vert=False)
plt.title('Dispersion des prix')
plt.show()
```



Outliers

1/ Méthode Interquartile (IQR) - NumPy

In [53]:
```
# Calculer les quartiles
Q1 = np.percentile(merging_erp_web2['price'], 25)
Q3 = np.percentile(merging_erp_web2['price'], 75)
IQR = Q3 - Q1

# Calculer les bornes des outliers
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Trouver les outliers
outliers = merging_erp_web2[(merging_erp_web2['price'] < lower_bound) | (merging_erp_web

print(f"IQR = {IQR:.2f}")
print(f"Seuil inférieur pour les outliers = {lower_bound:.2f}")
print(f"Seuil supérieur pour les outliers = {upper_bound:.2f}")
print("Outliers :")
print(outliers) # aucun prix en-dessous du seuil inférieur
```

```
IQR = 28.08
Seuil inférieur pour les outliers = -28.01
Seuil supérieur pour les outliers = 84.29
Outliers :
     product_id  onsale_web  price  stock_quantity stock_status id_web
63         4115           1  100.0              11      instock  15382  \
65         4132           1   88.4               5      instock  11668
199        4352           1  225.0               0   outofstock  15940
201        4355           1  126.5               2      instock  12589
205        4359           1   85.6               0   outofstock  13853
218        4402           1  176.0               8      instock   3510
219        4404           1  108.5               2      instock   3507
221        4406           1  157.0               3      instock   7819
222        4407           1  104.0               6      instock   3509
```

227	4582	1	109.6	7	instock	12857	
380	4903	1	102.3	20	instock	14805	
381	4904	1	137.0	13	instock	14220	
426	5001	1	217.5	20	instock	14581	
431	5007	1	105.0	17	instock	12791	
432	5008	1	105.0	10	instock	11602	
437	5025	1	112.0	0	outofstock	13914	
438	5026	1	86.8	2	instock	13913	
502	5565	1	92.0	0	outofstock	19822	
511	5612	1	124.8	12	instock	14915	
553	5767	1	175.0	12	instock	15185	
587	5892	1	191.3	10	instock	14983	
602	5916	1	93.0	3	instock	14774	
603	5917	1	122.0	4	instock	14775	
604	5918	1	114.0	8	instock	14773	
642	6126	1	135.0	10	instock	14923	
647	6201	1	105.6	7	instock	14596	
648	6202	1	116.4	14	instock	15126	
653	6212	1	115.0	2	instock	13996	
654	6213	1	121.0	7	instock	15072	
655	6214	1	99.0	7	instock	11601	
656	6215	1	115.0	4	instock	12790	
657	6216	1	121.0	6	instock	15070	

	virtual	downloadable	rating_count	average_rating	...	ping_status	
63	0	0	0	0.0	...	closed	\
65	0	0	0	0.0	...	closed	
199	0	0	0	0.0	...	closed	
201	0	0	0	0.0	...	closed	
205	0	0	0	0.0	...	closed	
218	0	0	0	0.0	...	closed	
219	0	0	0	0.0	...	closed	
221	0	0	0	0.0	...	closed	
222	0	0	0	0.0	...	closed	
227	0	0	0	0.0	...	closed	
380	0	0	0	0.0	...	closed	
381	0	0	0	0.0	...	closed	
426	0	0	0	0.0	...	closed	
431	0	0	0	0.0	...	closed	
432	0	0	0	0.0	...	closed	
437	0	0	0	0.0	...	closed	
438	0	0	0	0.0	...	closed	
502	0	0	0	0.0	...	closed	
511	0	0	0	0.0	...	closed	
553	0	0	0	0.0	...	closed	
587	0	0	0	0.0	...	closed	
602	0	0	0	0.0	...	closed	
603	0	0	0	0.0	...	closed	
604	0	0	0	0.0	...	closed	
642	0	0	0	0.0	...	closed	
647	0	0	0	0.0	...	closed	
648	0	0	0	0.0	...	closed	
653	0	0	0	0.0	...	closed	
654	0	0	0	0.0	...	closed	
655	0	0	0	0.0	...	closed	
656	0	0	0	0.0	...	closed	
657	0	0	0	0.0	...	closed	

	post_name	post_modified	
63	zind-humbrecht-riesling-gc-rangen-thann-clos-s...	2020-02-08 11:45:02	\
65	zind-humbrecht-pinot-gris-grand-cru-rangen-de-...	2020-02-20 09:55:02	
199	champagne-egly-ouriet-grand-cru-millesime-2008	2020-03-07 11:18:45	
201	champagne-egly-ouriet-grand-cru-brut-blanc-de-...	2020-08-13 10:15:02	
205	champagne-larmandier-bernier-grand-cru-vieille...	2019-12-23 09:30:11	
218	cognac-frapin-vip-xo	2020-08-22 11:35:03	
219	cognac-frapin-fontpinot-xo	2020-08-12 09:30:16	

```
221     cognac-frapin-chateau-de-fontpinot-1989-20-ans 2020-03-14 16:05:04
222                            cognac-frapin-cigar-blend 2020-07-04 09:45:03
227  chateau-de-puligny-montrachet-1cru-champ-canet... 2020-02-06 16:35:02
380  domaine-des-croix-corton-grand-cru-les-greves-... 2020-06-27 09:00:07
381  domaine-des-croix-corton-charlemagne-grand-cru... 2020-05-19 17:15:02
426     david-duband-charmes-chambertin-grand-cru-2014 2020-05-16 09:00:05
431  domaine-des-comtes-lafon-volnay-1er-cru-santen... 2020-07-02 09:30:03
432  domaine-des-comtes-lafon-volnay-1er-cru-santen... 2020-06-23 15:35:02
437  champagne-agrapart-fils-lavizoise-grand-cru-20... 2020-07-09 17:05:02
438  champagne-agrapart-fils-mineral-extra-brut-bla... 2020-05-11 14:35:02
502                       tempier-bandol-cabassaou-2017 2020-01-04 13:57:04
511  domaine-weinbach-gewurztraminer-gc-furstentum-... 2019-01-23 09:33:57
553                camille-giroud-clos-de-vougeot-2016 2020-06-11 15:25:04
587  coteaux-champenois-egly-ouriet-ambonnay-rouge-... 2020-04-01 09:30:09
602       wemyss-malts-single-cask-chocolate-moka-cake 2019-12-23 09:30:21
603  wemyss-malts-single-cask-scotch-whisky-choc-n-... 2020-03-11 09:30:09
604  wemyss-malts-single-cask-scotch-whisky-chai-ca... 2020-07-31 18:25:03
642              champagne-gosset-celebris-vintage-2007 2020-08-27 11:45:02
647  david-duband-chambolle-musigny-1er-cru-les-sen... 2020-02-29 15:25:02
648           domaine-clerget-echezeaux-en-orveaux-2015 2020-06-06 15:45:01
653  domaine-des-comtes-lafon-volnay-1er-cru-santen... 2020-06-16 09:30:16
654  domaine-des-comtes-lafon-volnay-1er-cru-santen... 2020-06-25 09:30:06
655  domaine-des-comtes-lafon-volnay-1er-cru-champa... 2020-07-04 11:35:02
656  domaine-des-comtes-lafon-volnay-1er-cru-champa... 2019-11-04 09:30:25
657  domaine-des-comtes-lafon-volnay-1er-cru-champa... 2020-07-30 09:30:08

      post_modified_gmt post_parent
63  2020-02-08 10:45:02         0.0  \
65  2020-02-20 08:55:02         0.0
199 2020-03-07 10:18:45         0.0
201 2020-08-13 08:15:02         0.0
205 2019-12-23 08:30:11         0.0
218 2020-08-22 09:35:03         0.0
219 2020-08-12 07:30:16         0.0
221 2020-03-14 15:05:04         0.0
222 2020-07-04 07:45:03         0.0
227 2020-02-06 15:35:02         0.0
380 2020-06-27 07:00:07         0.0
381 2020-05-19 15:15:02         0.0
426 2020-05-16 07:00:05         0.0
431 2020-07-02 07:30:03         0.0
432 2020-06-23 13:35:02         0.0
437 2020-07-09 15:05:02         0.0
438 2020-05-11 12:35:02         0.0
502 2020-01-04 12:57:04         0.0
511 2019-01-23 08:33:57         0.0
553 2020-06-11 13:25:04         0.0
587 2020-04-01 07:30:09         0.0
602 2019-12-23 08:30:21         0.0
603 2020-03-11 08:30:09         0.0
604 2020-07-31 16:25:03         0.0
642 2020-08-27 09:45:02         0.0
647 2020-02-29 14:25:02         0.0
648 2020-06-06 13:45:01         0.0
653 2020-06-16 07:30:16         0.0
654 2020-06-25 07:30:06         0.0
655 2020-07-04 09:35:02         0.0
656 2019-11-04 08:30:25         0.0
657 2020-07-30 07:30:08         0.0

                                                 guid menu_order post_type
63   https://www.bottle-neck.fr/?post_type=product&...        0.0   product  \
65   https://www.bottle-neck.fr/?post_type=product&...        0.0   product
199  https://www.bottle-neck.fr/?post_type=product&...        0.0   product
201  https://www.bottle-neck.fr/?post_type=product&...        0.0   product
205  https://www.bottle-neck.fr/?post_type=product&...        0.0   product
```

```
218  https://www.bottle-neck.fr/?post_type=product&...        0.0   product
219  https://www.bottle-neck.fr/?post_type=product&...        0.0   product
221  https://www.bottle-neck.fr/?post_type=product&...        0.0   product
222  https://www.bottle-neck.fr/?post_type=product&...        0.0   product
227  https://www.bottle-neck.fr/?post_type=product&...        0.0   product
380  https://www.bottle-neck.fr/?post_type=product&...        0.0   product
381  https://www.bottle-neck.fr/?post_type=product&...        0.0   product
426  https://www.bottle-neck.fr/?post_type=product&...        0.0   product
431  https://www.bottle-neck.fr/?post_type=product&...        0.0   product
432  https://www.bottle-neck.fr/?post_type=product&...        0.0   product
437  https://www.bottle-neck.fr/?post_type=product&...        0.0   product
438  https://www.bottle-neck.fr/?post_type=product&...        0.0   product
502  https://www.bottle-neck.fr/?post_type=product&...        0.0   product
511  https://www.bottle-neck.fr/?post_type=product&...        0.0   product
553  https://www.bottle-neck.fr/?post_type=product&...        0.0   product
587  https://www.bottle-neck.fr/?post_type=product&...        0.0   product
602  https://www.bottle-neck.fr/?post_type=product&...        0.0   product
603  https://www.bottle-neck.fr/?post_type=product&...        0.0   product
604  https://www.bottle-neck.fr/?post_type=product&...        0.0   product
642  https://www.bottle-neck.fr/?post_type=product&...        0.0   product
647  https://www.bottle-neck.fr/?post_type=product&...        0.0   product
648  https://www.bottle-neck.fr/?post_type=product&...        0.0   product
653  https://www.bottle-neck.fr/?post_type=product&...        0.0   product
654  https://www.bottle-neck.fr/?post_type=product&...        0.0   product
655  https://www.bottle-neck.fr/?post_type=product&...        0.0   product
656  https://www.bottle-neck.fr/?post_type=product&...        0.0   product
657  https://www.bottle-neck.fr/?post_type=product&...        0.0   product

    comment_count    C.A.
63            0.0     0.0
65            0.0     0.0
199           0.0  1125.0
201           0.0  1391.5
205           0.0    85.6
218           0.0  2288.0
219           0.0   217.0
221           0.0     0.0
222           0.0   104.0
227           0.0     0.0
380           0.0     0.0
381           0.0   685.0
426           0.0     0.0
431           0.0     0.0
432           0.0     0.0
437           0.0     0.0
438           0.0     0.0
502           0.0     0.0
511           0.0     0.0
553           0.0     0.0
587           0.0   573.9
602           0.0     0.0
603           0.0     0.0
604           0.0     0.0
642           0.0   270.0
647           0.0     0.0
648           0.0     0.0
653           0.0   230.0
654           0.0     0.0
655           0.0     0.0
656           0.0     0.0
657           0.0     0.0

[32 rows x 29 columns]
```

2/ Calcul Interquartile classique

In [56]:
```
# Données fournies
Q1 = merging_erp_web2['price'].quantile(0.25).round(2)
Q3 = merging_erp_web2['price'].quantile(0.75).round(2)
IQR = Q3 - Q1

# Seuils pour les outliers
lower_bound = Q1 - 1.5 * IQR # juste pour information, mais nous n'avons pas de prix nég
seuil = Q3 + 1.5 * IQR

# Données
data = merging_erp_web2['price']

# Trouver les outliers
outliers = [val for val in data if val > seuil]
not_outliers = [val for val in data if val < seuil]

print(f"Seuil supèrieur = {seuil :.2f}")
print(f"Seuil infèrieur = {lower_bound :.2f}")
print(f"Outliers : {outliers}")
print(f"not_outliers: {not_outliers}")
```

```
Seuil supèrieur = 84.30
Seuil infèrieur = -28.02
Outliers : [100.0, 88.4, 225.0, 126.5, 85.6, 176.0, 108.5, 157.0, 104.0, 109.6, 102.3, 1
37.0, 217.5, 105.0, 105.0, 112.0, 86.8, 92.0, 124.8, 175.0, 191.3, 93.0, 122.0, 114.0, 1
35.0, 105.6, 116.4, 115.0, 121.0, 99.0, 115.0, 121.0]
not_outliers: [24.2, 34.3, 20.8, 14.1, 46.0, 34.3, 32.7, 31.2, 60.0, 42.6, 80.0, 18.3, 2
2.8, 19.3, 21.8, 7.7, 33.7, 44.3, 71.6, 12.7, 8.7, 8.7, 8.7, 11.9, 11.9, 14.5, 14.4, 19.
5, 20.8, 22.0, 16.6, 60.0, 23.4, 33.2, 32.0, 77.8, 12.7, 14.7, 14.05, 22.9, 44.0, 37.0,
39.0, 17.0, 23.2, 19.0, 16.4, 14.4, 13.7, 12.6, 22.8, 12.8, 22.1, 12.8, 15.8, 15.8, 16.
3, 16.3, 9.7, 6.8, 12.6, 35.0, 31.7, 23.0, 29.8, 25.7, 77.4, 39.0, 53.0, 49.0, 29.5, 33.
0, 37.5, 69.0, 59.0, 13.7, 19.2, 29.0, 9.8, 14.5, 20.35, 12.0, 18.5, 9.3, 9.3, 11.6, 14.
3, 10.8, 7.6, 12.0, 20.5, 14.3, 18.2, 9.0, 7.8, 5.7, 5.7, 5.7, 13.5, 13.5, 11.5, 24.0, 2
4.0, 11.9, 16.7, 21.4, 16.6, 13.3, 9.5, 12.1, 9.3, 17.8, 13.5, 10.8, 27.2, 9.4, 5.8, 5.
8, 38.0, 38.0, 9.9, 11.3, 23.0, 6.7, 7.6, 79.8, 48.5, 39.8, 58.8, 26.5, 13.4, 17.1, 16.
7, 11.6, 12.8, 8.9, 9.7, 17.2, 16.9, 12.6, 29.9, 9.6, 11.1, 11.1, 17.1, 28.0, 28.0, 8.9,
8.6, 13.5, 8.9, 15.3, 14.8, 19.5, 14.1, 59.6, 26.9, 24.4, 31.7, 32.1, 12.2, 9.9, 15.8, 1
5.8, 17.8, 9.6, 19.0, 15.2, 10.2, 15.5, 16.6, 9.2, 12.9, 14.9, 17.6, 24.8, 18.9, 11.6, 2
7.0, 41.0, 69.8, 38.6, 26.7, 19.0, 23.2, 39.1, 44.0, 17.5, 30.0, 8.1, 10.7, 10.9, 49.0,
35.5, 83.0, 59.0, 79.5, 79.5, 51.6, 39.0, 77.0, 49.5, 49.5, 49.5, 57.0, 59.8, 27.5, 62.
0, 59.0, 59.0, 59.0, 44.0, 62.5, 68.1, 28.1, 21.7, 28.5, 67.2, 43.9, 61.6, 41.8, 26.5, 1
6.1, 31.5, 31.5, 49.0, 32.2, 50.1, 13.4, 11.8, 13.1, 26.2, 20.6, 16.9, 19.0, 24.0, 57.0,
67.5, 30.6, 59.0, 11.9, 16.5, 52.4, 52.9, 58.3, 39.6, 52.4, 62.4, 76.8, 50.0, 52.4, 41.
0, 62.4, 50.0, 21.5, 28.5, 24.3, 16.5, 25.3, 49.0, 36.2, 33.4, 40.2, 43.0, 43.0, 48.8, 2
0.8, 16.4, 14.4, 21.0, 12.3, 20.2, 17.0, 21.9, 17.8, 19.8, 19.0, 10.7, 12.9, 9.5, 29.8,
13.2, 6.3, 7.1, 9.1, 9.1, 18.1, 14.0, 30.1, 12.8, 12.8, 19.8, 18.2, 31.6, 16.8, 22.8, 3
2.6, 44.0, 55.4, 15.8, 18.4, 13.3, 11.1, 18.6, 23.4, 18.2, 12.5, 15.9, 13.7, 29.0, 23.4,
12.7, 26.0, 29.5, 8.6, 14.3, 22.0, 16.8, 15.3, 7.4, 9.7, 14.5, 11.9, 16.4, 27.9, 12.0,
7.4, 26.5, 24.3, 16.9, 6.8, 13.9, 7.8, 13.7, 9.8, 29.5, 28.5, 10.1, 12.1, 12.3, 11.1, 1
1.1, 13.6, 21.0, 18.7, 41.6, 12.0, 78.0, 14.9, 6.5, 8.7, 8.5, 9.9, 8.2, 9.8, 9.9, 9.3, 2
2.8, 18.7, 28.4, 27.9, 25.3, 17.0, 27.9, 20.1, 27.9, 21.2, 20.8, 41.0, 46.0, 22.9, 53.2,
25.9, 17.3, 25.9, 28.0, 25.9, 25.9, 37.2, 24.4, 24.4, 7.0, 12.8, 23.2, 7.9, 6.5, 7.9, 1
6.7, 17.5, 27.8, 25.7, 18.4, 22.2, 11.1, 9.9, 12.5, 12.9, 20.5, 25.0, 11.3, 7.0, 12.1,
7.1, 49.5, 23.0, 23.7, 16.45, 16.3, 18.5, 26.5, 78.0, 78.0, 78.0, 27.3, 64.9, 48.7, 59.
4, 48.7, 55.6, 9.3, 19.8, 45.0, 62.1, 22.5, 7.5, 52.6, 45.0, 67.0, 59.9, 59.9, 65.0, 15.
2, 19.0, 11.1, 18.0, 22.8, 19.5, 28.8, 16.1, 24.2, 35.3, 10.7, 12.7, 17.1, 24.0, 39.0, 1
3.2, 23.6, 15.5, 16.3, 16.2, 7.2, 54.8, 42.0, 19.8, 10.2, 10.4, 11.5, 17.9, 21.6, 33.4,
49.5, 43.5, 43.5, 26.5, 13.8, 18.2, 12.9, 38.6, 48.4, 60.4, 38.6, 12.0, 61.6, 65.9, 15.
2, 24.6, 48.8, 34.3, 36.3, 57.7, 38.0, 58.0, 58.0, 30.8, 27.5, 34.7, 59.6, 83.7, 30.5, 3
8.6, 18.0, 63.4, 19.2, 19.2, 56.4, 38.4, 27.8, 71.3, 71.3, 25.0, 10.3, 28.0, 44.6, 13.0,
12.7, 6.5, 17.5, 29.9, 44.5, 29.4, 16.9, 19.8, 10.3, 10.8, 31.7, 25.0, 57.6, 13.7, 7.1,
14.9, 11.0, 14.6, 10.7, 73.3, 42.1, 57.0, 24.5, 10.1, 42.2, 13.1, 19.5, 12.9, 35.6, 35.
6, 33.6, 34.4, 38.4, 29.7, 32.8, 5.7, 5.8, 5.8, 21.7, 23.0, 12.5, 17.2, 40.2, 24.0, 23.
8, 17.1, 25.0, 17.4, 27.3, 17.9, 24.0, 16.6, 16.9, 57.6, 63.5, 56.0, 63.5, 41.2, 55.0, 5
7.0, 19.3, 19.0, 26.6, 15.4, 24.7, 28.1, 18.25, 35.1, 27.3, 18.8, 43.9, 19.8, 17.7, 57.
```

```
0, 36.0, 36.0, 48.5, 49.5, 14.1, 59.9, 46.0, 74.5, 17.2, 8.7, 15.4, 12.7, 30.0, 13.5, 1
6.3, 56.3, 71.5, 69.0, 17.9, 48.5, 71.7, 8.5, 10.9, 21.8, 38.5, 9.3, 13.6, 24.5, 12.6, 1
3.4, 29.8, 36.9, 40.7, 33.2, 34.8, 74.8, 62.4, 46.0, 39.2, 10.6, 10.6, 5.2, 46.0, 31.0,
20.2, 25.2, 25.2, 23.5, 26.4, 26.7, 20.4, 20.4, 40.2, 9.0, 45.9, 10.4, 78.0, 40.5, 27.9,
22.4, 28.4, 72.0, 29.0, 29.2, 44.0, 68.3, 41.8, 40.0, 13.5, 19.0, 24.4, 32.8, 15.4, 9.9,
13.5, 51.0, 35.2, 42.2, 33.2, 41.8, 32.2, 37.7, 47.2, 52.7, 22.4, 50.4, 35.6, 27.7, 48.
5, 15.4, 46.5, 46.5, 46.5, 42.0, 21.8, 50.5, 49.9, 19.0, 8.4, 27.5, 69.0, 54.8, 16.3]
```

3/ Méthode interquartile avec package scipy

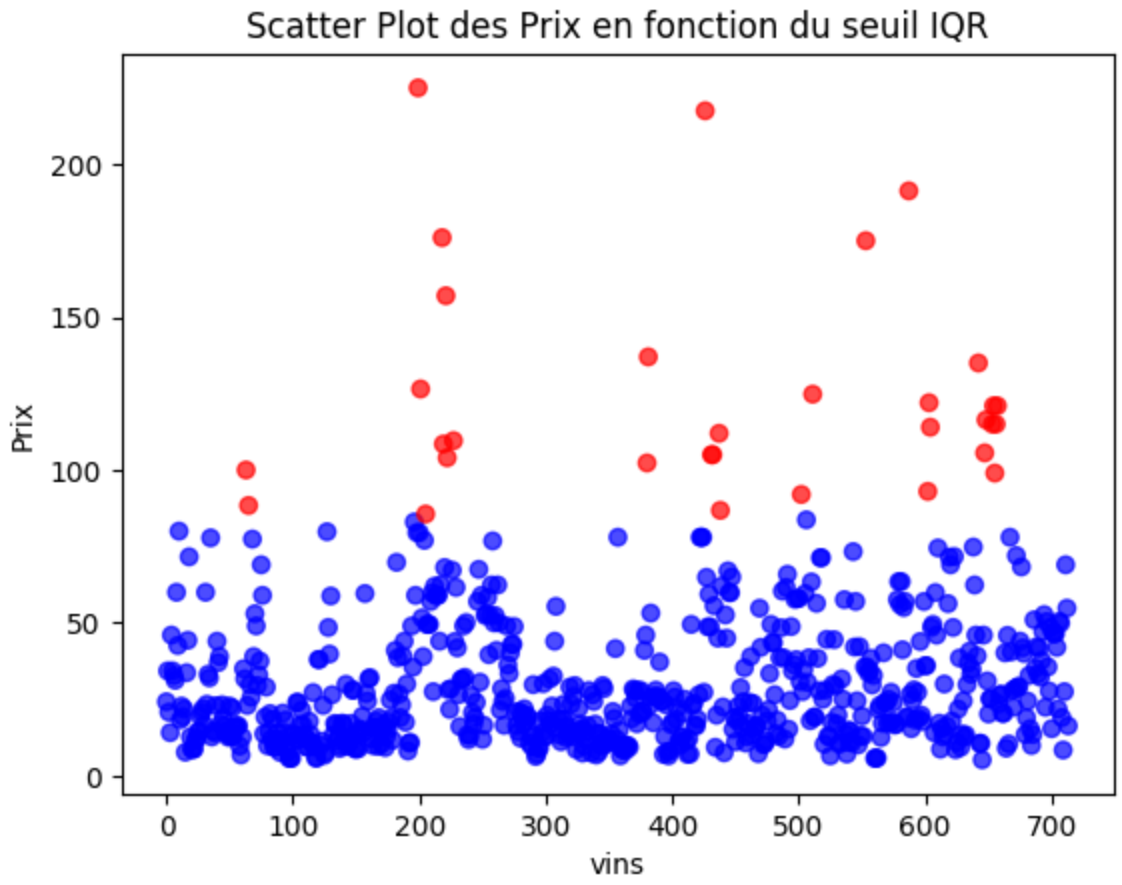
In [59]:
```python
from scipy.stats import iqr

# Calculer l'IQR
data = merging_erp_web2['price']
iqr_value = iqr(data)

# Calculer les bornes des outliers
Q1 = data.quantile(0.25)
Q3 = data.quantile(0.75)
lower_bound = Q1 - 1.5 * iqr_value
upper_bound = Q3 + 1.5 * iqr_value

# Trouver les outliers
outliers = data[(data < lower_bound) | (data > upper_bound)]

print(f"IQR = {iqr_value:.2f}")
print(f"Borne inférieure pour les outliers = {lower_bound:.2f}")
print(f"Borne supérieure pour les outliers = {upper_bound:.2f}")
print(f"Outliers : {outliers.tolist()}")
```

```
IQR = 28.08
Borne inférieure pour les outliers = -28.01
Borne supérieure pour les outliers = 84.29
Outliers : [100.0, 88.4, 225.0, 126.5, 85.6, 176.0, 108.5, 157.0, 104.0, 109.6, 102.3, 1
37.0, 217.5, 105.0, 105.0, 112.0, 86.8, 92.0, 124.8, 175.0, 191.3, 93.0, 122.0, 114.0, 1
35.0, 105.6, 116.4, 115.0, 121.0, 99.0, 115.0, 121.0]
```

Visuel avec Matplotlib

In [60]:
```python
# Données fournies
Q1 = merging_erp_web2['price'].quantile(0.25).round(2)
Q3 = merging_erp_web2['price'].quantile(0.75).round(2)
IQR = Q3 - Q1

# Seuils pour les outliers
seuil = Q3 + 1.5 * IQR

# Données (remplacez cela par vos données)
data = merging_erp_web2['price']

# Initialisation des listes pour les couleurs
colors = []

# Attribution des couleurs en fonction de la position par rapport au seuil
for val in data:
    if val > seuil:
        colors.append('red')  # Au-dessus du seuil
    else:
        colors.append('blue')  # En-dessous du seuil

# Création du scatter plot
plt.scatter(range(len(data)), data, color=colors, alpha=0.7)

# Ajout des étiquettes
plt.xlabel('vins')
```

```
plt.ylabel('Prix')
plt.title('Scatter Plot des Prix en fonction du seuil IQR')

# Affichage du graphique
plt.show()
```



Visuel avec Plotly

In [61]:

```
# Données fournies
Q1 = merging_erp_web2['price'].quantile(0.25).round(2)
Q3 = merging_erp_web2['price'].quantile(0.75).round(2)
IQR = Q3 - Q1

# Seuils pour les outliers
seuil = Q3 + 1.5 * IQR

print(IQR)
print(seuil)


# Données (remplacez cela par vos données)
data = merging_erp_web2

# Attribution des couleurs en fonction de la position par rapport au seuil
colors = ['> IQR' if val > seuil else '< IQR' for val in data['price']]

# Création du scatter plot avec Plotly
fig = px.scatter(data, x=data.index, y='price', color=colors, labels={'color': 'Position

# Personnalisation du graphique
fig.update_traces(marker=dict(size=5, opacity=0.7))
fig.update_layout(
    title='Dispersion des Prix en fonction du seuil Interquartile',
    xaxis_title='Index',
    yaxis_title='Prix'
)
```

```
# Affichage du graphique
fig.show()
```

```
28.08
84.3
```

In [62]:
```
# Données fournies
Q1 = merging_erp_web2['price'].quantile(0.25).round(2)
Q3 = merging_erp_web2['price'].quantile(0.75).round(2)
IQR = Q3 - Q1

# Seuils pour les outliers
seuil = Q3 + 1.5 * IQR

# Données
data = merging_erp_web2

# Sélectionnez les vins au-dessus du seuil
above_threshold = data[data['price'] > seuil]

# Trier par ordre décroissant de prix
above_threshold = above_threshold.sort_values(by='price', ascending=False).head(10)

# Création du graphique à barres avec la palette de couleurs bordeaux
fig = px.bar(above_threshold, x='post_name', y='price', color='price',
             labels={'post_name': 'Nom du Vin', 'price': 'Prix'},
             color_discrete_sequence=px.colors.qualitative.Set3)  # Utilisation de la pa

# Personnalisation du graphique
fig.update_layout(
    title='Vins de prestige au-dessus du seuil IQR (Triés par Prix décroissant)',
    xaxis_title='Nom du Vin',
    yaxis_title='Prix',
    legend_title='Vin',
    xaxis_tickangle=-45,
    showlegend=False
)

# Ajustement de la taille du graphique et des marges
fig.update_layout(
    autosize=False,
    width=1000,
    height=600,
    margin=dict(l=50, r=50, b=100, t=100)  # Ajustement des marges
)

# Affichage du graphique
fig.show()
```

In [63]:
```
above_threshold = len(data[data['price'] > seuil])
print(f"Il y a {above_threshold} outliers (valeurs atypiques).")
```

```
Il y a 32 outliers (valeurs atypiques).
```

A/ Z-SCORE

1/ Méthode classique

In [65]:
```
# Calculer le Z-score
mean = merging_erp_web2['price'].mean()
std_ = merging_erp_web2['price'].std()
```

```
merging_erp_web2['Z-Score'] = (merging_erp_web2['price'] - mean) / std_

# Définir un seuil de 2 pour identifier les valeurs aberrantes
threshold = 2
outliers = merging_erp_web2[abs(merging_erp_web2['Z-Score']) > threshold]

top_outliers_ = outliers[['post_name', 'price', 'Z-Score']].sort_values(by='Z-Score', as
top_outliers_
```

Out[65]:

	post_name	price	Z-Score
199	champagne-egly-ouriet-grand-cru-millesime-2008	225.0	6.922087
426	david-duband-charmes-chambertin-grand-cru-2014	217.5	6.652405
587	coteaux-champenois-egly-ouriet-ambonnay-rouge-...	191.3	5.710315
218	cognac-frapin-vip-xo	176.0	5.160164
553	camille-giroud-clos-de-vougeot-2016	175.0	5.124206
221	cognac-frapin-chateau-de-fontpinot-1989-20-ans	157.0	4.476969
381	domaine-des-croix-corton-charlemagne-grand-cru...	137.0	3.757817
642	champagne-gosset-celebris-vintage-2007	135.0	3.685902
201	champagne-egly-ouriet-grand-cru-brut-blanc-de-...	126.5	3.380262
511	domaine-weinbach-gewurztraminer-gc-furstentum-...	124.8	3.319134
603	wemyss-malts-single-cask-scotch-whisky-choc-n-...	122.0	3.218453
657	domaine-des-comtes-lafon-volnay-1er-cru-champa...	121.0	3.182495
654	domaine-des-comtes-lafon-volnay-1er-cru-santen...	121.0	3.182495
648	domaine-clerget-echezeaux-en-orveaux-2015	116.4	3.017090
656	domaine-des-comtes-lafon-volnay-1er-cru-champa...	115.0	2.966750
653	domaine-des-comtes-lafon-volnay-1er-cru-santen...	115.0	2.966750
604	wemyss-malts-single-cask-scotch-whisky-chai-ca...	114.0	2.930792
437	champagne-agrapart-fils-lavizoise-grand-cru-20...	112.0	2.858877
227	chateau-de-puligny-montrachet-1cru-champ-canet...	109.6	2.772578
219	cognac-frapin-fontpinot-xo	108.5	2.733025
647	david-duband-chambolle-musigny-1er-cru-les-sen...	105.6	2.628748
432	domaine-des-comtes-lafon-volnay-1er-cru-santen...	105.0	2.607173
431	domaine-des-comtes-lafon-volnay-1er-cru-santen...	105.0	2.607173
222	cognac-frapin-cigar-blend	104.0	2.571216
380	domaine-des-croix-corton-grand-cru-les-greves-...	102.3	2.510088
63	zind-humbrecht-riesling-gc-rangen-thann-clos-s...	100.0	2.427385
655	domaine-des-comtes-lafon-volnay-1er-cru-champa...	99.0	2.391428
602	wemyss-malts-single-cask-chocolate-moka-cake	93.0	2.175682
502	tempier-bandol-cabassaou-2017	92.0	2.139725
65	zind-humbrecht-pinot-gris-grand-cru-rangen-de-...	88.4	2.010277

In [85]:
```
len(top_outliers_)
```

Out[85]:
```
30
```

30 vins représentent des valeurs atypiques

2/ Méthode package Scikit Learn

In [66]:
```
from sklearn.preprocessing import StandardScaler
# Créer un objet StandardScaler
scaler = StandardScaler()
# Calculer les Z-scores
merging_erp_web2['Z-Score'] = scaler.fit_transform(merging_erp_web2[['price']])

# Définir un seuil de 2 pour identifier les valeurs aberrantes
threshold = 2
outliers = merging_erp_web2[abs(merging_erp_web2['Z-Score']) > threshold]

top_outliers = outliers[['post_name', 'price', 'Z-Score']].sort_values(by='Z-Score', asc
top_outliers
```

Out[66]:

	post_name	price	Z-Score
199	champagne-egly-ouriet-grand-cru-millesime-2008	225.0	6.926939
426	david-duband-charmes-chambertin-grand-cru-2014	217.5	6.657068
587	coteaux-champenois-egly-ouriet-ambonnay-rouge-...	191.3	5.714318
218	cognac-frapin-vip-xo	176.0	5.163781
553	camille-giroud-clos-de-vougeot-2016	175.0	5.127798
221	cognac-frapin-chateau-de-fontpinot-1989-20-ans	157.0	4.480108
381	domaine-des-croix-corton-charlemagne-grand-cru...	137.0	3.760451
642	champagne-gosset-celebris-vintage-2007	135.0	3.688486
201	champagne-egly-ouriet-grand-cru-brut-blanc-de-...	126.5	3.382632
511	domaine-weinbach-gewurztraminer-gc-furstentum-...	124.8	3.321461
603	wemyss-malts-single-cask-scotch-whisky-choc-n-...	122.0	3.220709
657	domaine-des-comtes-lafon-volnay-1er-cru-champa...	121.0	3.184726
654	domaine-des-comtes-lafon-volnay-1er-cru-santen...	121.0	3.184726
648	domaine-clerget-echezeaux-en-orveaux-2015	116.4	3.019205
656	domaine-des-comtes-lafon-volnay-1er-cru-champa...	115.0	2.968829
653	domaine-des-comtes-lafon-volnay-1er-cru-santen...	115.0	2.968829
604	wemyss-malts-single-cask-scotch-whisky-chai-ca...	114.0	2.932846
437	champagne-agrapart-fils-lavizoise-grand-cru-20...	112.0	2.860881
227	chateau-de-puligny-montrachet-1cru-champ-canet...	109.6	2.774522
219	cognac-frapin-fontpinot-xo	108.5	2.734941
647	david-duband-chambolle-musigny-1er-cru-les-sen...	105.6	2.630591
432	domaine-des-comtes-lafon-volnay-1er-cru-santen...	105.0	2.609001
431	domaine-des-comtes-lafon-volnay-1er-cru-santen...	105.0	2.609001
222	cognac-frapin-cigar-blend	104.0	2.573018
380	domaine-des-croix-corton-grand-cru-les-greves-...	102.3	2.511848
63	zind-humbrecht-riesling-gc-rangen-thann-clos-s...	100.0	2.429087
655	domaine-des-comtes-lafon-volnay-1er-cru-champa...	99.0	2.393104
602	wemyss-malts-single-cask-chocolate-moka-cake	93.0	2.177207

502	tempier-bandol-cabassaou-2017	92.0	2.141225
65	zind-humbrecht-pinot-gris-grand-cru-rangen-de-...	88.4	2.011686

3/ Méthode avec le package scipy

In [67]:
```
from scipy.stats import zscore

# Calculer les Z-scores
merging_erp_web2['Z-Score'] = zscore(merging_erp_web2['price'])

# Définir un seuil de 2 pour identifier les valeurs aberrantes
threshold = 2
outliers = merging_erp_web2[abs(merging_erp_web2['Z-Score']) > threshold]

top_32_outliers = outliers[['post_name', 'price', 'Z-Score']].sort_values(by='Z-Score',
top_32_outliers
```

Out[67]:

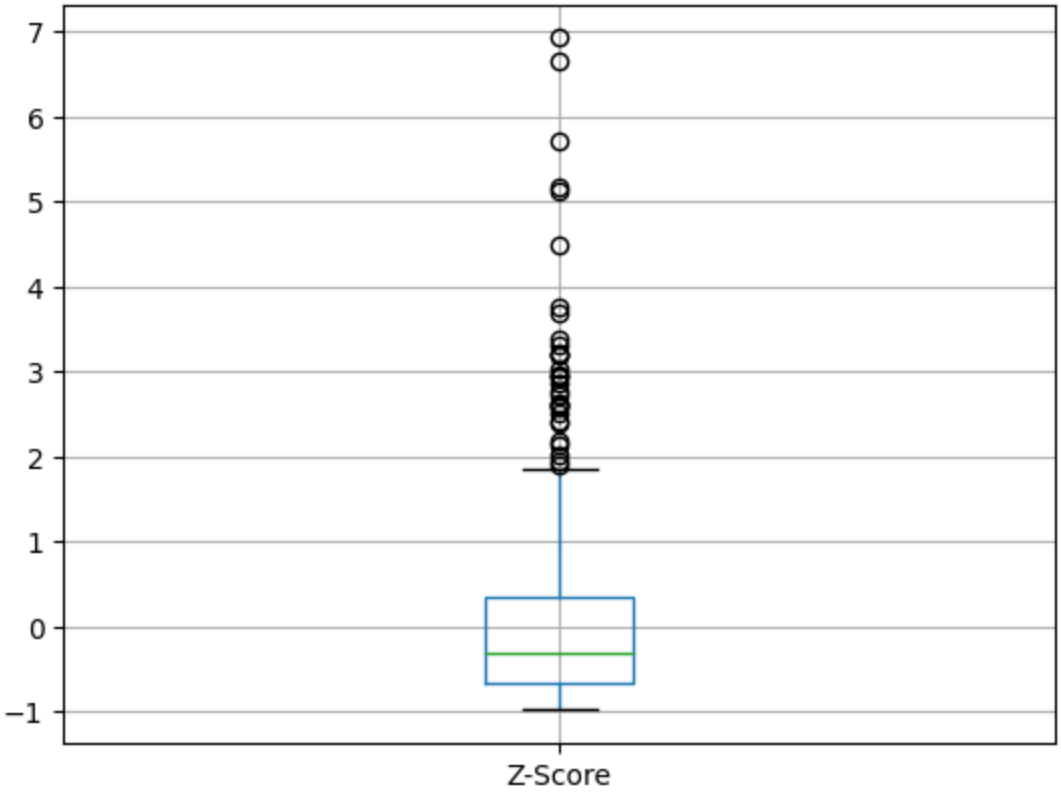
	post_name	price	Z-Score
199	champagne-egly-ouriet-grand-cru-millesime-2008	225.0	6.926939
426	david-duband-charmes-chambertin-grand-cru-2014	217.5	6.657068
587	coteaux-champenois-egly-ouriet-ambonnay-rouge-...	191.3	5.714318
218	cognac-frapin-vip-xo	176.0	5.163781
553	camille-giroud-clos-de-vougeot-2016	175.0	5.127798
221	cognac-frapin-chateau-de-fontpinot-1989-20-ans	157.0	4.480108
381	domaine-des-croix-corton-charlemagne-grand-cru...	137.0	3.760451
642	champagne-gosset-celebris-vintage-2007	135.0	3.688486
201	champagne-egly-ouriet-grand-cru-brut-blanc-de-...	126.5	3.382632
511	domaine-weinbach-gewurztraminer-gc-furstentum-...	124.8	3.321461
603	wemyss-malts-single-cask-scotch-whisky-choc-n-...	122.0	3.220709
657	domaine-des-comtes-lafon-volnay-1er-cru-champa...	121.0	3.184726
654	domaine-des-comtes-lafon-volnay-1er-cru-santen...	121.0	3.184726
648	domaine-clerget-echezeaux-en-orveaux-2015	116.4	3.019205
656	domaine-des-comtes-lafon-volnay-1er-cru-champa...	115.0	2.968829
653	domaine-des-comtes-lafon-volnay-1er-cru-santen...	115.0	2.968829
604	wemyss-malts-single-cask-scotch-whisky-chai-ca...	114.0	2.932846
437	champagne-agrapart-fils-lavizoise-grand-cru-20...	112.0	2.860881
227	chateau-de-puligny-montrachet-1cru-champ-canet...	109.6	2.774522
219	cognac-frapin-fontpinot-xo	108.5	2.734941
647	david-duband-chambolle-musigny-1er-cru-les-sen...	105.6	2.630591
432	domaine-des-comtes-lafon-volnay-1er-cru-santen...	105.0	2.609001
431	domaine-des-comtes-lafon-volnay-1er-cru-santen...	105.0	2.609001
222	cognac-frapin-cigar-blend	104.0	2.573018
380	domaine-des-croix-corton-grand-cru-les-greves-...	102.3	2.511848
63	zind-humbrecht-riesling-gc-rangen-thann-clos-s...	100.0	2.429087
655	domaine-des-comtes-lafon-volnay-1er-cru-champa...	99.0	2.393104
602	wemyss-malts-single-cask-chocolate-moka-cake	93.0	2.177207

502	tempier-bandol-cabassaou-2017	92.0	2.141225
65	zind-humbrecht-pinot-gris-grand-cru-rangen-de-...	88.4	2.011686

Z-Score Boxplot - Détermination du seuil

In [68]:
```
merging_erp_web2[['Z-Score']].boxplot()
```

Out[68]:
```
<Axes: >
```



In [70]:
```
# Sélectionnez les colonnes nécessaires
outliers_data = outliers[['post_name', 'Z-Score']]

# Trier les données en ordre décroissant selon le Z-Score
outliers_data = outliers_data.sort_values(by='Z-Score', ascending=False).head(15)

# Créer un graphique interactif
fig = px.bar(outliers_data, x='Z-Score', y='post_name', orientation='h', text='Z-Score',
             labels={'post_name': 'Vins', 'Z-Score': 'Z-Score'},
             title='Classement du Z-Score des Outliers')

# Ajuster la taille de la police
fig.update_layout(font=dict(size=10))

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

Analyses complémentaires

In [82]:
```
best_sales = merging_erp_web2.groupby(['post_title','price']).agg({'total_sales':'sum'})
best_sales
```

Out[82]:

		total_sales
post_title	**price**	

Champagne Gosset Grand Blanc de Blancs	**49.0**	96.0
Champagne Gosset Grand Rosé	**49.0**	87.0
Gilles Robin Crozes-Hermitage Rouge Papillon 2019	**16.6**	62.0
Moulin de Gassac IGP Pays d'Hérault Guilhem Rosé 2019	**5.8**	46.0
Maurel Pays d'Oc Chardonnay 2019	**5.7**	43.0
Le Pas de l'Escalette Languedoc Les Petits Pas 2019	**13.3**	42.0
Domaine Giudicelli Patrimonio Blanc 2019	**25.2**	41.0
Champagne Gosset Grande Réserve	**39.0**	40.0
Emile Boeckel Crémant Brut Blanc de Blancs	**8.6**	38.0
Château de La Liquière Faugères L'Ampoule 2019	**10.9**	38.0
Domaine Giudicelli Patrimonio Rouge 2016	**25.2**	37.0
Jacqueson Rully Blanc 1er Cru La Pucelle 2018	**27.9**	36.0
Elian Daros Côtes du Marmandais Clos Baquey 2015	**29.0**	36.0
Triennes IGP Méditerranée Rosé 2019	**9.3**	33.0
Albert Mann Muscat 2018	**16.8**	32.0
Champagne Gosset Grand Millésime 2006	**53.0**	30.0
Domaine Saint-Denis Bourgogne Rouge Clos de la Coque 2018	**19.5**	30.0
Philippe Nusswitz Duché d'Uzès Orénia Rouge 2017	**8.5**	29.0
Château Turcaud Entre-Deux-Mers 2019	**7.1**	26.0
Maurel Pays d'Oc Merlot 2018	**5.8**	24.0

In [90]:
```
# Trier les données par total_sales pour obtenir les meilleures ventes des valeurs atypi
vins_les_plus_chers = merging_erp_web2[['post_title','price','total_sales']].sort_values
vins_les_plus_chers = vins_les_plus_chers[vins_les_plus_chers['price'] > 84]
vins_les_plus_chers.sort_values(by='total_sales', ascending=False)
```

Out[90]:

	post_title	price	total_sales
218	Cognac Frapin VIP XO	176.0	13.0
201	Champagne Egly-Ouriet Grand Cru Blanc de Noirs	126.5	11.0
199	Champagne Egly-Ouriet Grand Cru Millésimé 2008	225.0	5.0
381	Domaine Des Croix Corton Charlemagne Grand Cru...	137.0	5.0
587	Coteaux Champenois Egly-Ouriet Ambonnay Rouge ...	191.3	3.0
653	Domaine des Comtes Lafon Volnay 1er Cru Santen...	115.0	2.0
219	Cognac Frapin Château de Fontpinot XO	108.5	2.0
642	Champagne Gosset Célébris Vintage 2007	135.0	2.0
222	Cognac Frapin Cigar Blend	104.0	1.0
205	Champagne Larmandier-Bernier Grand Cru Les Che...	85.6	1.0
603	Wemyss Malts Single Cask Scotch Whisky Choc 'n...	122.0	0.0
654	Domaine des Comtes Lafon Volnay 1er Cru Santen...	121.0	0.0
438	Champagne Agrapart & Fils Minéral Extra Br...	86.8	0.0
65	Zind-Humbrecht Pinot Gris Grand Cru Rangen De ...	88.4	0.0
502	Tempier Bandol Cabassaou 2017	92.0	0.0
602	Wemyss Malts Single Cask Scotch Whisky Chocola...	93.0	0.0

655	Domaine des Comtes Lafon Volnay 1er Cru Champa...	99.0	0.0
63	Zind-Humbrecht Riesling Grand Cru Rangen De Th...	100.0	0.0
380	Domaine Des Croix Corton Grand Cru Les Grèves ...	102.3	0.0
553	Camille Giroud Clos de Vougeot 2016	175.0	0.0
432	Domaine des Comtes Lafon Volnay 1er Cru Santen...	105.0	0.0
431	Domaine des Comtes Lafon Volnay 1er Cru Santen...	105.0	0.0
647	David Duband Chambolle-Musigny 1er Cru Les Sen...	105.6	0.0
221	Cognac Frapin Château de Fontpinot 1989 20 Ans...	157.0	0.0
227	Château de Meursault Puligny-Montrachet 1er Cr...	109.6	0.0
437	Champagne Agrapart & Fils L'Avizoise Extra...	112.0	0.0
426	David Duband Charmes-Chambertin Grand Cru 2014	217.5	0.0
511	Domaine Weinbach Gewurztraminer Grand Cru Furs...	124.8	0.0
656	Domaine des Comtes Lafon Volnay 1er Cru Champa...	115.0	0.0
648	Domaine Clerget Echezeaux Grand Cru En Orveaux...	116.4	0.0
657	Domaine des Comtes Lafon Volnay 1er Cru Champa...	121.0	0.0
604	Wemyss Malts Single Cask Scotch Whisky Chai Ca...	114.0	0.0

In [91]:
```
# compter les lignes dont les valeurs ne sont pas des zéros
np.count_nonzero(vins_les_plus_chers['total_sales'])
```

Out[91]:
```
10
```

10 vins prestigieux représentent des ventes

In [97]:
```
import plotly.graph_objects as go

# Sélectionner les 10 meilleures ventes
top_10_sales = vins_les_plus_chers.sort_values(by='total_sales', ascending=False).head(1

# Créer le graphique
fig = go.Figure()

# Créer une liste de couleurs en fonction de la meilleure vente
colors = ['red' if row.equals(top_10_sales.iloc[0]) else 'green' for _, row in top_10_sa

# Ajouter les barres pour les meilleures ventes
fig.add_trace(go.Bar(
    x=top_10_sales['post_title'],
    y=top_10_sales['total_sales'],
    marker_color=colors,
    textposition='outside'
))

# Mise en page du graphique
fig.update_layout(
    title='Classement des meilleures Ventes parmi les Outliers',
    xaxis_title='Nom du Vin',
    yaxis_title='Ventes Totales',
    xaxis_tickangle=-45,  # Inclinaison des étiquettes pour une meilleure lisibilité
    font=dict(size=10)  # Réduire la taille de la police
)
```

```
# Affichage du graphique
fig.show()
```