

OpenClassrooms_5ème projet de la formation Data Analyst

Optimisez la gestion des données d'une boutique avec R ou Python!

REZAE_Mortaza_1_notebook_022023

Importer des bibliothèques

Entrée [1]:

```
import pandas as pd
import numpy as np
```

Importer des données

Entrée [2]:

```
web = pd.read_excel('web.xlsx')
erp = pd.read_excel('erp.xlsx')
liaison = pd.read_excel('liaison.xlsx')
```

```
C:\Users\REZAE\anaconda3\lib\site-packages\openpyxl\worksheet\_read_only.py:79: UserWarning: Unknown extension is not supported and will be removed
  for idx, row in parser.parse():
C:\Users\REZAE\anaconda3\lib\site-packages\openpyxl\worksheet\_read_only.py:79: UserWarning: Unknown extension is not supported and will be removed
  for idx, row in parser.parse():
C:\Users\REZAE\anaconda3\lib\site-packages\openpyxl\worksheet\_read_only.py:79: UserWarning: Unknown extension is not supported and will be removed
  for idx, row in parser.parse():
```

Afficher la table de web

Entrée [3]:

```
web.head()
```

Out[3]:

	sku	virtual	downloadable	rating_count	average_rating	total_sales	tax_status	tax_class	post_a
0	bon-cadeau-25-euros	0	0	0	0.0	10.0	taxable	NaN	
1	15298	0	0	0	0.0	6.0	taxable	NaN	
2	15296	0	0	0	0.0	0.0	taxable	NaN	

Afficher le nombre de ligne et colonne de web

Entrée [4]:

```
web.shape
```

Out[4]:

```
(1513, 28)
```

Afficher le nombre de ligne et colonne de erp

Entrée [5]:

```
erp.shape
```

Out[5]:

```
(825, 5)
```

Afficher le nombre de ligne et colonne de liaison

Entrée [6]:

```
liaison.shape
```

Out[6]:

```
(825, 2)
```

Verifier les infos de chaque colonne

Entrée [7]:

```
web.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1513 entries, 0 to 1512
Data columns (total 28 columns):
#   Column                Non-Null Count  Dtype
---  -
0   sku                    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
```

Le nombre totale de valeur null de colonne sku

Entrée [8]:

```
web['sku'].isnull().sum()
```

Out[8]:

Counter et afficher les valeurs null dans le colonne de sku

Entrée [9]:

```
web['sku'].isnull().value_counts()
```

Out[9]:

```
False    1428
True       85
Name: sku, dtype: int64
```

Créer et afficher la table de web_null

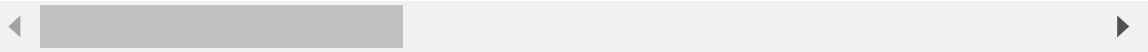
Entrée [10]:

```
web_null=web[web['sku'].isnull()]
web_null.head()
```

Out[10]:

	sku	virtual	downloadable	rating_count	average_rating	total_sales	tax_status	tax_c
178	NaN	0	0	0	NaN	NaN	NaN	
179	NaN	0	0	0	NaN	NaN	NaN	
227	NaN	0	0	0	NaN	NaN	NaN	
230	NaN	0	0	0	NaN	NaN	NaN	
231	NaN	0	0	0	NaN	NaN	NaN	

5 rows × 28 columns



Afficher le nombre de ligne et colonne de web_null

Entrée [11]:

```
web_null.shape
```

Out[11]:

```
(85, 28)
```

Afficher info de web_null

Entrée [12]:

```
web_null.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 85 entries, 178 to 796
Data columns (total 28 columns):
#   Column                Non-Null Count  Dtype
---  -
0   sku                   0 non-null     object
1   virtual              85 non-null    int64
2   downloadable         85 non-null    int64
3   rating_count         85 non-null    int64
4   average_rating       2 non-null     float64
5   total_sales          2 non-null     float64
6   tax_status           2 non-null     object
7   tax_class            0 non-null     float64
8   post_author          2 non-null     float64
9   post_date            2 non-null     datetime64[ns]
10  post_date_gmt        2 non-null     datetime64[ns]
11  post_content         0 non-null     float64
12  post_title           2 non-null     object
13  post_excerpt         2 non-null     object
14  post_status          2 non-null     object
15  comment_status       2 non-null     object
16  ping_status          2 non-null     object
17  post_password        0 non-null     float64
18  post_name            2 non-null     object
19  post_modified        2 non-null     datetime64[ns]
20  post_modified_gmt    2 non-null     datetime64[ns]
21  post_content_filtered 0 non-null     float64
22  post_parent          2 non-null     float64
23  guid                2 non-null     object
24  menu_order           2 non-null     float64
25  post_type            2 non-null     object
26  post_mime_type       0 non-null     object
27  comment_count        2 non-null     float64
dtypes: datetime64[ns](4), float64(10), int64(3), object(11)
memory usage: 19.3+ KB
```

Afficher les lignes que total_sales sont pas nulls

Entrée [13]:

```
web_null[web_null['total_sales'].notnull()]
```

Out[13]:

	sku	virtual	downloadable	rating_count	average_rating	total_sales	tax_status	tax_c
470	NaN	0	0	0	0.0	0.0	taxable	
471	NaN	0	0	0	0.0	0.0	taxable	

2 rows × 28 columns

Créer la table web2 (sku n'a accune valeur null)

Entrée [14]:

```
web2=web[web['sku'].notnull()]
```

Afficher web2

Entrée [15]:

```
web2.head()
```

Out[15]:

	sku	virtual	downloadable	rating_count	average_rating	total_sales	tax_status	tax_class	post_a
0	bon-cadeau-25-euros	0	0	0	0.0	10.0	taxable	NaN	
1	15298	0	0	0	0.0	6.0	taxable	NaN	
2	15296	0	0	0	0.0	0.0	taxable	NaN	

Afficher les lignes avec les valeurs nulls

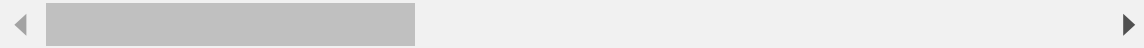
Entrée [16]:

```
web2[web2['sku'].isnull()]
```

Out[16]:

sku	virtual	downloadable	rating_count	average_rating	total_sales	tax_status	tax_class
-----	---------	--------------	--------------	----------------	-------------	------------	-----------

0 rows × 28 columns



Nombre de valeur unique/ doublon dans colonne sku

Entrée [17]:

```
web2.sku.nunique()
```

Out[17]:

714

Changer la type de sku à str

Entrée [18]:

```
web2.sku=web2.sku.astype('str')
```

C:\Users\REZAE\AppData\Local\Temp\ipykernel_9600\1847724746.py:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
web2.sku=web2.sku.astype('str')
```

Trier par colonne sku

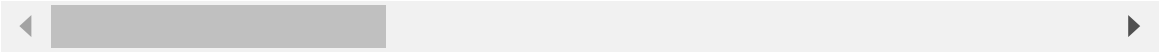
Entrée [19]:

```
web2.sort_values(by='sku')
```

Out[19]:

	sku	virtual	downloadable	rating_count	average_rating	total_sales	tax_status	t
1399	10014	0	0	0	0.0	0.0	NaN	
628	10014	0	0	0	0.0	0.0	taxable	
255	10459	0	0	0	0.0	0.0	taxable	
1046	10459	0	0	0	0.0	0.0	NaN	
1106	10775	0	0	0	0.0	0.0	NaN	
...	
1011	9636	0	0	0	0.0	0.0	NaN	
1201	9937	0	0	0	0.0	4.0	NaN	
422	9937	0	0	0	0.0	4.0	taxable	
1209	bon-cadeau-25-euros	0	0	0	0.0	10.0	NaN	
0	bon-cadeau-25-euros	0	0	0	0.0	10.0	taxable	

1428 rows × 28 columns



Afficher le colonne post_type

Entrée [20]:

```
web2.post_type.value_counts()
```

Out[20]:

```
product      714
attachment   714
Name: post_type, dtype: int64
```

Supprimer l'attachment et garder que le product

Entrée [21]:

```
web2=web2[web2['post_type']!='product'].copy()
web2.post_type.value_counts()
```

Out[21]:

```
product      714
Name: post_type, dtype: int64
```

Afficher info de la table erp

Entrée [22]:

```
erp.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
```

Afficher erp

Entrée [23]:

```
erp.head()
```

Out[23]:

	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

Il n'y pas des valeurs manquants dans erp

Trouver les doublons dans erp

Entrée [24]:

```
erp.duplicated().sum()
```

Out[24]:

0

Trouver valeur unique dans erp

Entrée [25]:

```
erp.product_id.nunique()
```

Out[25]:

825

Il n'y a pas des doublons dans erp

Entrée [26]:

```
erp[erp['product_id'].isnull()]
```

Out[26]:

	product_id	onsale_web	price	stock_quantity	stock_status
--	------------	------------	-------	----------------	--------------

Afficher la table de liaison

Entrée [27]:

```
liaison.head()
```

Out[27]:

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

Affiche info de liaison

Entrée [28]:

```
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
```

Merger erp avec liaison

Entrée [29]:

```
merge1= pd.merge(erp, liaison, on='product_id',how='outer', indicator=True)
```

Afficher merge1

Entrée [30]:

```
merge1.head()
```

Out[30]:

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

Changer la type de id_web à str

Entrée [31]:

```
merge1.id_web=merge1.id_web.astype('str')
```

Afficher info de merge1

Entrée [32]:

```
merge1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 825 entries, 0 to 824
Data columns (total 7 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
5   id_web          825 non-null    object
6   _merge          825 non-null    category
dtypes: category(1), float64(1), int64(3), object(2)
memory usage: 46.1+ KB
```

Le nombre de 'product id' est identique dans deux tables de liaison et erp

Entrée [33]:

```
merge1._merge.value_counts()
```

Out[33]:

```
both          825
left_only      0
right_only     0
Name: _merge, dtype: int64
```

Supprimer le colonne de _merge

Entrée [34]:

```
merge1.drop('_merge',axis=1, inplace=True)
```

Afficher info de merge1

Entrée [35]:

```
merge1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 825 entries, 0 to 824
Data columns (total 6 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
5   id_web          825 non-null   object
dtypes: float64(1), int64(3), object(2)
memory usage: 45.1+ KB
```

Afficher merge1

Entrée [36]:

```
merge1.head()
```

Out[36]:

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

Merger les tables merge1 et web2

Entrée [37]:

```
merge2=pd.merge(merge1, web2, left_on='id_web', right_on='sku', how='outer', indicator=1)
```

Afficher info de merge2

Entrée [38]:

```
merge2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 825 entries, 0 to 824
Data columns (total 35 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
5   id_web                825 non-null    object
6   sku                   714 non-null    object
7   virtual               714 non-null    float64
8   downloadable          714 non-null    float64
9   rating_count          714 non-null    float64
10  average_rating        714 non-null    float64
11  total_sales           714 non-null    float64
12  tax_status            714 non-null    object
13  tax_class             0 non-null      float64
14  post_author           714 non-null    float64
15  post_date             714 non-null    datetime64[ns]
16  post_date_gmt         714 non-null    datetime64[ns]
17  post_content          0 non-null      float64
18  post_title            714 non-null    object
19  post_excerpt          714 non-null    object
20  post_status           714 non-null    object
21  comment_status        714 non-null    object
22  ping_status           714 non-null    object
23  post_password         0 non-null      float64
24  post_name             714 non-null    object
25  post_modified          714 non-null    datetime64[ns]
26  post_modified_gmt     714 non-null    datetime64[ns]
27  post_content_filtered 0 non-null      float64
28  post_parent           714 non-null    float64
29  guid                 714 non-null    object
30  menu_order            714 non-null    float64
31  post_type             714 non-null    object
32  post_mime_type        0 non-null      object
33  comment_count         714 non-null    float64
34  _merge               825 non-null    category
dtypes: category(1), datetime64[ns](4), float64(14), int64(3), object(13)
memory usage: 226.5+ KB
```

Afficher les valeurs _merge

Entrée [39]:

```
merge2._merge.value_counts()
```

Out[39]:

```
both          714
left_only     111
right_only      0
Name: _merge, dtype: int64
```

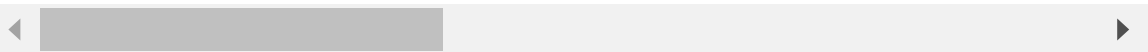
Describe left only

Entrée [40]:

```
merge2[merge2['_merge']=='left_only'].describe()
```

Out[40]:

	product_id	onsale_web	price	stock_quantity	virtual	downloadable	rating_c
count	111.000000	111.000000	111.000000	111.000000	0.0	0.0	
mean	5999.072072	0.027027	31.917117	12.027027	NaN	NaN	
std	1112.203699	0.162898	19.107908	24.302809	NaN	NaN	
min	4055.000000	0.000000	8.000000	0.000000	NaN	NaN	
25%	4866.500000	0.000000	20.350000	0.000000	NaN	NaN	
50%	6324.000000	0.000000	30.000000	3.000000	NaN	NaN	
75%	7082.500000	0.000000	40.000000	14.000000	NaN	NaN	
max	7329.000000	1.000000	144.000000	192.000000	NaN	NaN	



Vérifier les produits qui sont vendu en web mais il n y a pas dans le stock

Entrée [41]:

```
merge2[(merge2['_merge']=='left_only') & (merge2['onsale_web']==1)]
```

Out[41]:

	product_id	onsale_web	price	stock_quantity	stock_status	id_web	sku	virtual	dow
30	4594	1	144.0	0	outofstock	nan	NaN	NaN	
47	5070	1	84.7	0	outofstock	nan	NaN	NaN	
48	5075	1	43.3	0	outofstock	nan	NaN	NaN	

3 rows × 35 columns

Garder uniquement 'both' dans le colonne de _merge (les autres sont pas vendus)

Entrée [42]:

```
merge3=merge2[merge2['_merge']=='both'].copy()  
merge3._merge.value_counts()
```

Out[42]:

```
both          714  
left_only      0  
right_only     0  
Name: _merge, dtype: int64
```

Créer et afficher le colonne de CA

Entrée [43]:

```
merge3['CA']=merge3['total_sales']*merge3['price']
merge3.head()
```

Out[43]:

	product_id	onsale_web	price	stock_quantity	stock_status	id_web	sku	virtual	dow
0	3847	1	24.2	0	outofstock	15298	15298	0.0	
1	3849	1	34.3	0	outofstock	15296	15296	0.0	
2	3850	1	20.8	0	outofstock	15300	15300	0.0	
3	4032	1	14.1	0	outofstock	19814	19814	0.0	
4	4039	1	46.0	0	outofstock	19815	19815	0.0	

5 rows × 36 columns

Afficher les 10 produits qui ont les CA plus importants

Entrée [44]:

```
merge3[['product_id','price', 'total_sales','CA']].sort_values(by='CA',ascending=False).
```

Out[44]:

	product_id	price	total_sales	CA
286	4334	49.0	96.0	4704.0
162	4144	49.0	87.0	4263.0
310	4402	176.0	13.0	2288.0
161	4142	53.0	30.0	1590.0
160	4141	39.0	40.0	1560.0
293	4355	126.5	11.0	1391.5
291	4352	225.0	5.0	1125.0
170	4153	29.0	36.0	1044.0
761	6206	25.2	41.0	1033.2
121	4068	16.6	62.0	1029.2

Chiffre d'affaires totale

Entrée [45]:

```
merge3.CA.sum()
```

Out[45]:

```
70568.6
```

Un outlier ou valeur aberrante correspond à une valeur éloignée de la distribution de la variable.

Cela pourra être dû à une erreur de typographie ou à une erreur de mesure mais cela pourra également être une valeur extrême.

On parle couramment de valeur extrême, pour désigner une valeur non erronée qui s'éloigne néanmoins fortement du reste des valeurs de la variable.

Importe la bibliotheque de seaborn

Entrée [46]:

```
import seaborn as sns
```

Afficher le boxplot associé à la price

Entrée [47]:

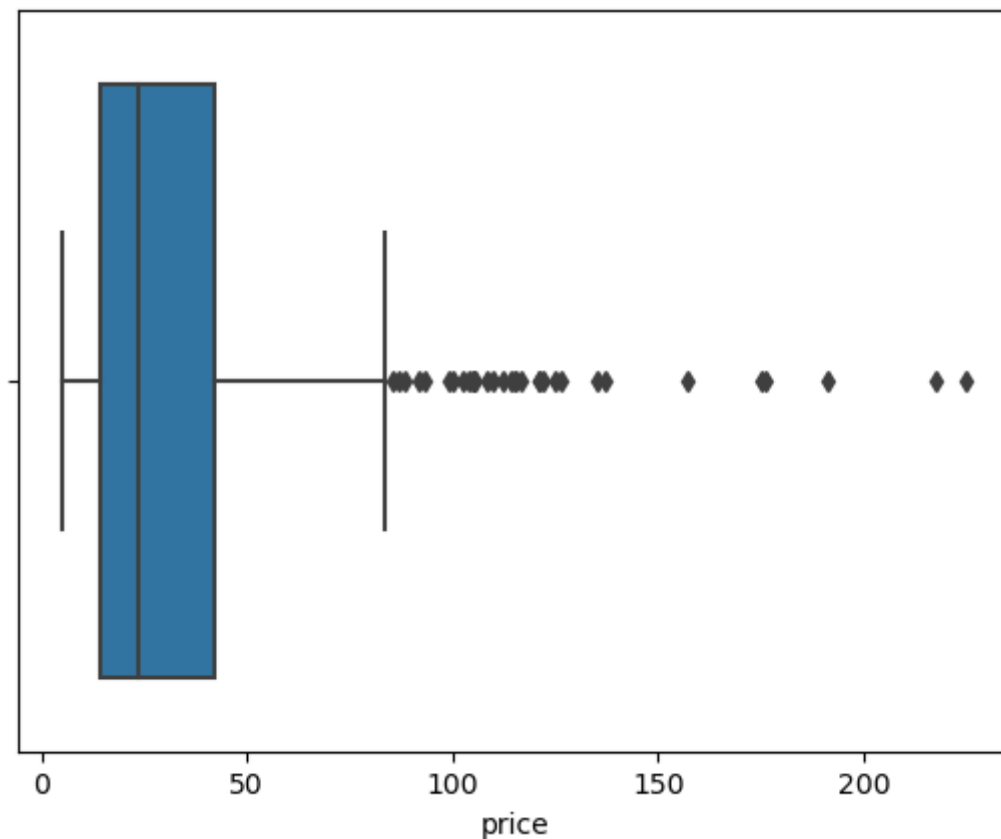
```
sns.boxplot(merge3['price'])
```

C:\Users\REZAE\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```

Out[47]:

```
<AxesSubplot:xlabel='price'>
```



Importer la librairie de numpy

Entrée [48]:

```
import numpy as np
```

les methodes pour traiter les outliers : np.quantile() et Zscores

Entrée [49]:

```
Q1=np.quantile(merge3.price,0.25)  
Q1
```

Out[49]:

14.1

Entrée [50]:

```
Q3=np.quantile(merge3.price,0.75)  
Q3
```

Out[50]:

42.175000000000004

Entrée [51]:

```
upper_limit=Q3+(1.5*(Q3-Q1))  
upper_limit
```

Out[51]:

84.28750000000001

Trier merge3 par price qui est superieur à upper_limit

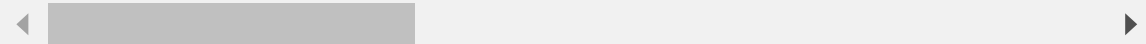
Entrée [52]:

```
merge3[merge3['price']>upper_limit].sort_values('price', ascending=False)  
merge3.head()
```

Out[52]:

	product_id	onsale_web	price	stock_quantity	stock_status	id_web	sku	virtual	dow
0	3847	1	24.2	0	outofstock	15298	15298	0.0	
1	3849	1	34.3	0	outofstock	15296	15296	0.0	
2	3850	1	20.8	0	outofstock	15300	15300	0.0	
3	4032	1	14.1	0	outofstock	19814	19814	0.0	
4	4039	1	46.0	0	outofstock	19815	19815	0.0	

5 rows × 36 columns



Nombre de produit avec les valeurs aberrante

Entrée [53]:

```
merge3[merge3['price']>upper_limit].shape
```

Out[53]:

(32, 36)

Importer la bibliotheque de zscore

Entrée [54]:

```
import scipy.stats as stats
```

Créer et afficher le colonne de zscore

Entrée [55]:

```
merge3['zscores'] = stats.zscore(merge3['price'])
merge3.head()
```

Out[55]:

	product_id	onsale_web	price	stock_quantity	stock_status	id_web	sku	virtual	down
0	3847	1	24.2	0	outofstock	15298	15298	0.0	
1	3849	1	34.3	0	outofstock	15296	15296	0.0	
2	3850	1	20.8	0	outofstock	15300	15300	0.0	
3	4032	1	14.1	0	outofstock	19814	19814	0.0	
4	4039	1	46.0	0	outofstock	19815	19815	0.0	

5 rows × 37 columns



- 1. Un produit dont zscore est égale à 0, c'est à dire un produit dans le prix égal à la moyenne
- 2. Un produit dont zscore est supérieur à 0, c'est à dire un produit dans le prix supérieur à moyenne
- 3. un produit dont zscore est inférieur à 0, c'est à dire un produit dans le prix inferieur à moyenne

Afficher les produits avec les zscores superier à 3 (14 produits)

Entrée [56]:

```
merge3[merge3['zscores']>3].sort_values('price', ascending=False)
```

Out[56]:

	product_id	onsale_web	price	stock_quantity	stock_status	id_web	sku	virtual	dc
291	4352	1	225.0	0	outofstock	15940	15940	0.0	
525	5001	1	217.5	20	instock	14581	14581	0.0	
692	5892	1	191.3	10	instock	14983	14983	0.0	
310	4402	1	176.0	8	instock	3510	3510	0.0	
657	5767	1	175.0	12	instock	15185	15185	0.0	
313	4406	1	157.0	3	instock	7819	7819	0.0	
478	4904	1	137.0	13	instock	14220	14220	0.0	
752	6126	1	135.0	10	instock	14923	14923	0.0	
293	4355	1	126.5	2	instock	12589	12589	0.0	
615	5612	1	124.8	12	instock	14915	14915	0.0	
708	5917	1	122.0	4	instock	14775	14775	0.0	
764	6213	1	121.0	7	instock	15072	15072	0.0	
767	6216	1	121.0	6	instock	15070	15070	0.0	
758	6202	1	116.4	14	instock	15126	15126	0.0	

14 rows × 37 columns



Afficher les produits avec les zscores superier à 2 (30 produits)

Entrée [57]:

```
merge3[merge3['zscores']>2].sort_values('price', ascending=False)
```

Out[57]:

	product_id	onsale_web	price	stock_quantity	stock_status	id_web	sku	virtual	dc
291	4352	1	225.0	0	outofstock	15940	15940	0.0	
525	5001	1	217.5	20	instock	14581	14581	0.0	
692	5892	1	191.3	10	instock	14983	14983	0.0	
310	4402	1	176.0	8	instock	3510	3510	0.0	
657	5767	1	175.0	12	instock	15185	15185	0.0	
313	4406	1	157.0	3	instock	7819	7819	0.0	
478	4904	1	137.0	13	instock	14220	14220	0.0	
752	6126	1	135.0	10	instock	14923	14923	0.0	
293	4355	1	126.5	2	instock	12589	12589	0.0	
615	5612	1	124.8	12	instock	14915	14915	0.0	
708	5917	1	122.0	4	instock	14775	14775	0.0	
767	6216	1	121.0	6	instock	15070	15070	0.0	
764	6213	1	121.0	7	instock	15072	15072	0.0	
758	6202	1	116.4	14	instock	15126	15126	0.0	
766	6215	1	115.0	4	instock	12790	12790	0.0	
763	6212	1	115.0	2	instock	13996	13996	0.0	
709	5918	1	114.0	8	instock	14773	14773	0.0	
538	5025	1	112.0	0	outofstock	13914	13914	0.0	

	product_id	onsale_web	price	stock_quantity	stock_status	id_web	sku	virtual	dc
320	4582	1	109.6	7	instock	12857	12857	0.0	
311	4404	1	108.5	2	instock	3507	3507	0.0	
757	6201	1	105.6	7	instock	14596	14596	0.0	
531	5008	1	105.0	10	instock	11602	11602	0.0	
530	5007	1	105.0	17	instock	12791	12791	0.0	
314	4407	1	104.0	6	instock	3509	3509	0.0	
477	4903	1	102.3	20	instock	14805	14805	0.0	
154	4115	1	100.0	11	instock	15382	15382	0.0	
765	6214	1	99.0	7	instock	11601	11601	0.0	
707	5916	1	93.0	3	instock	14774	14774	0.0	
605	5565	1	92.0	0	outofstock	19822	19822	0.0	
156	4132	1	88.4	5	instock	11668	11668	0.0	

30 rows × 37 columns



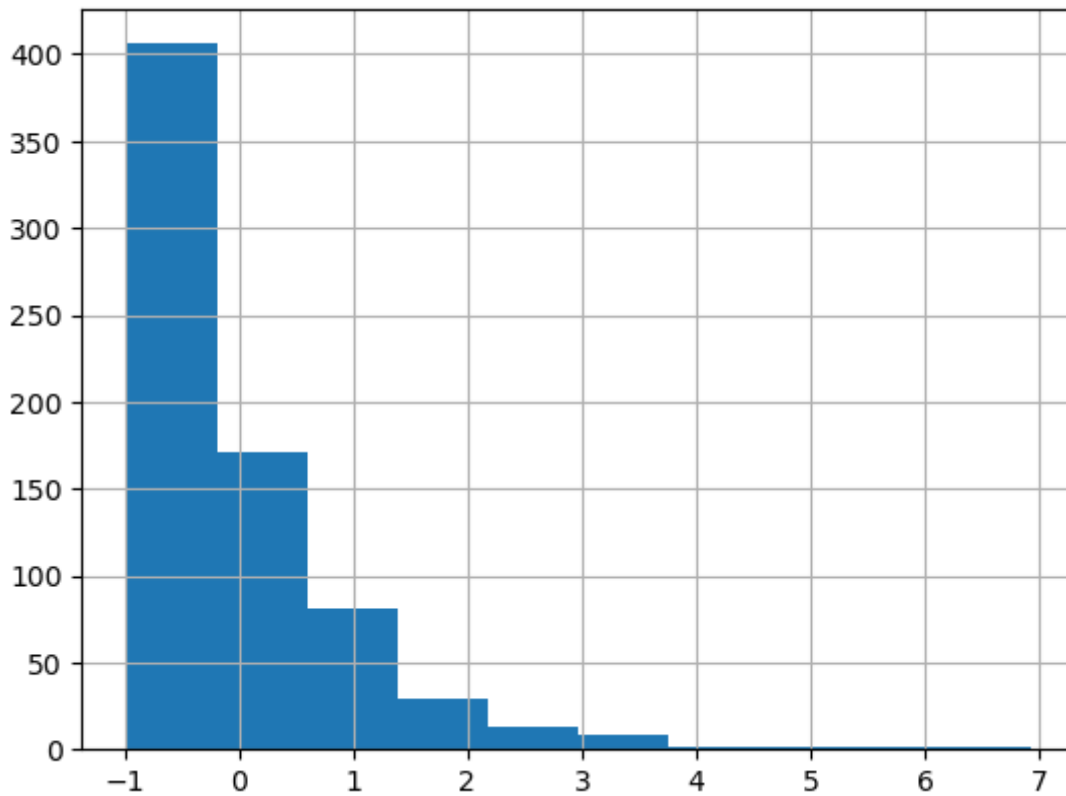
Afficher histogram de colonne zscores

Entrée [58]:

```
merge3['zscores'].hist()
```

Out[58]:

<AxesSubplot:>



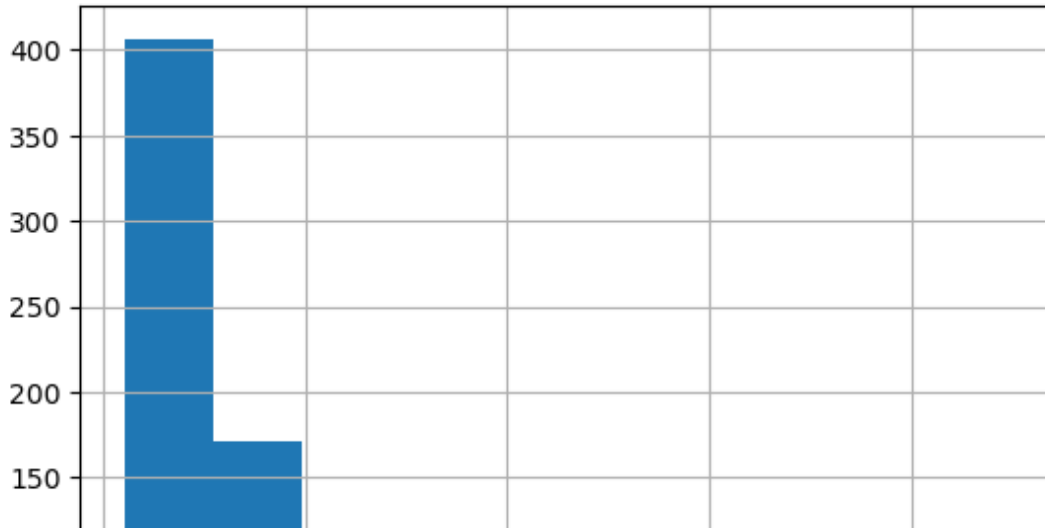
Afficher histogram de colonne price

Entrée [59]:

```
merge3['price'].hist()
```

Out[59]:

<AxesSubplot:>



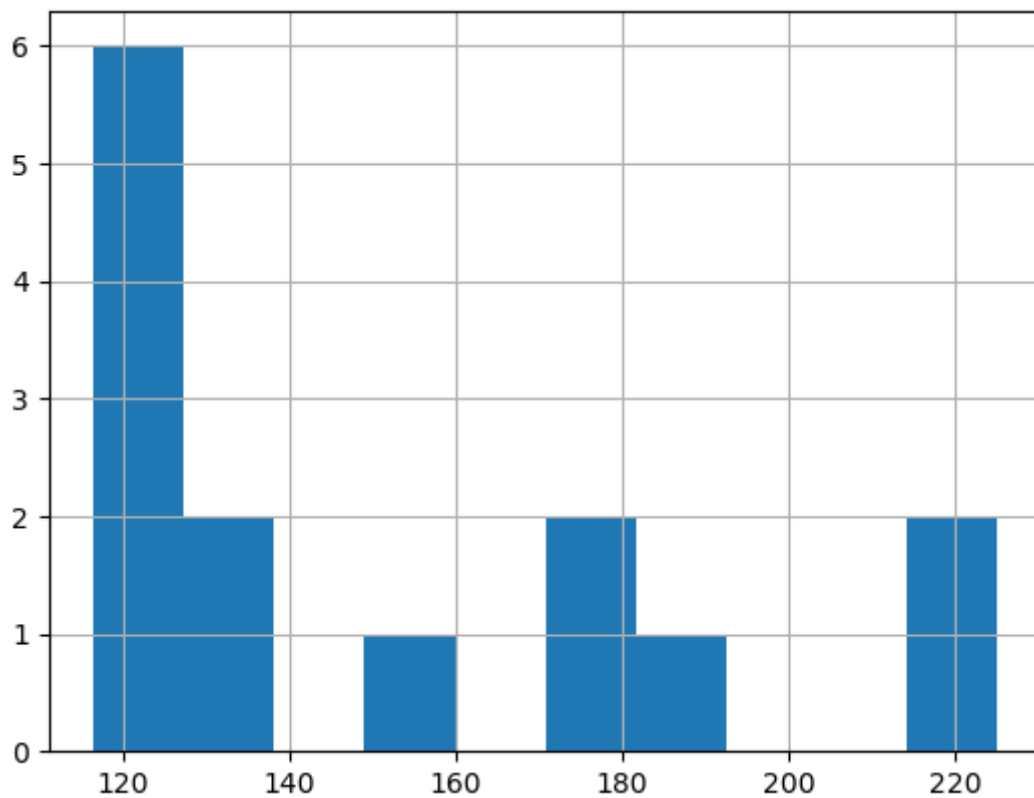
Afficher les prix pour lesquelles zscores sont supérieur à 3

Entrée [60]:

```
merge3['price'][merge3['zscores']>3].hist()
```

Out[60]:

<AxesSubplot:>



Afficher les prix pour lesquelles zscores sont supérieur à 2

Entrée [61]:

```
merge3['price'][merge3['zscores']>2].hist()
```

Out[61]:

<AxesSubplot:>

