PROJET 05 DATA ANALYST

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

PARTIE NETTOYAGE DES DONNEES </h2>

In [628…
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

In [629…
```
df_liaison=pd.read_csv('Fichier_liaison.csv', delimiter=';',encoding='utf-8')
df_erp=pd.read_csv('Fichier_erp.csv', delimiter=';',encoding='utf-8')
df_web=pd.read_csv('Fichier_web.csv', delimiter=';',encoding='utf-8')
```

1 - Preparation du fichier_web</h3>

In [630…
```
df_web.head()
```

Out[630]:

	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 [631…
```
#renommer sku en id_web
df_web.rename(columns={"sku":"id_web"}, inplace=True)
```

In [632…
```
df_web.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   object 
 10  post_date_gmt          1430 non-null   object 
 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   object 
 20  post_modified_gmt      1430 non-null   object 
 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: float64(10), int64(3), object(15)
memory usage: 331.1+ KB
```

In [633…
```
#supression des valeurs manquantes dans la colonne id_web
df_web.dropna(subset=['id_web'],inplace=True)
df_web.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1428 entries, 0 to 1512
Data columns (total 28 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   id_web                 1428 non-null   object 
 1   virtual                1428 non-null   int64  
 2   downloadable           1428 non-null   int64  
 3   rating_count           1428 non-null   int64  
 4   average_rating         1428 non-null   float64
 5   total_sales            1428 non-null   float64
 6   tax_status             714 non-null    object 
 7   tax_class              0 non-null      float64
 8   post_author            1428 non-null   float64
 9   post_date              1428 non-null   object 
 10  post_date_gmt          1428 non-null   object 
 11  post_content           0 non-null      float64
 12  post_title             1428 non-null   object 
 13  post_excerpt           714 non-null    object 
 14  post_status            1428 non-null   object 
```

```
 15  comment_status         1428 non-null   object
 16  ping_status            1428 non-null   object
 17  post_password          0 non-null      float64
 18  post_name              1428 non-null   object
 19  post_modified          1428 non-null   object
 20  post_modified_gmt      1428 non-null   object
 21  post_content_filtered  0 non-null      float64
 22  post_parent            1428 non-null   float64
 23  guid                   1428 non-null   object
 24  menu_order             1428 non-null   float64
 25  post_type              1428 non-null   object
 26  post_mime_type         714 non-null    object
 27  comment_count          1428 non-null   float64
dtypes: float64(10), int64(3), object(15)
memory usage: 323.5+ KB
```

In [634…

```
df_web['id_web'].unique()
```

Out[634]:

```
array(['16004', '15075', '16209', '15763', '13895', '12857', '15740',
       '14253', '14106', '13996', '16146', '15895', '15861', '15428',
       '15310', '14372', '812', '7033', '7032', '16077', '16237', '16028',
       '1364', '13913', '15202', '15576', '19815', '15148', '15774',
       '14982', '15339', '15382', '15325', '15945', '14941', '15241',
       '12641', '16269', '12942', '15344', '15661', '16274', '14661',
       '15718', '14395', '15254', '16056', '14839', '13957', '15476',
       '13515', '15004', '15070', '15032', '16042', '16063', '11602',
       '16005', '16283', '15615', '15206', '13520', '19821', '15296',
       '11847', '14371', '15678', '16416', '16130', '15921', '15378',
       '15141', '14805', '16044', '16238', '16129', '15280', '15038',
       '16081', '16281', '16505', '15880', '14101', '16527', '15649',
       '16072', '15621', '14604', '15329', '15812', '15733', '15237',
       '16191', '14099', '15711', '15261', '15790', '12194', '15414',
       '15482', '15030', '14950', '15892', '13412', '15425', '16567',
       '13762', '16246', '807', '15316', '15933', '15868', '15478',
       '15444', '15269', '14000', '16186', '13904', '16043', '16247',
       '793', '14302', '14192', '14508', '10775', '15448', '15736',
       '15648', '15342', '16148', '12476', '15735', '15196', '16071',
       '13531', '3568', '13969', '14632', '12869', '16564', '15670',
       '15306', '15047', '14975', '11668', '13435', '16263', '14506',
       '15527', '804', '14580', '15490', '16189', '15714', '13089',
       '14802', '14800', '16053', '14944', '16264', '11586', '14573',
       '15303', '15213', '15345', '15461', '14746', '13959', '15657',
       '16151', '14092', '15479', '15887', '8193', '15891', '15928',
       'bon-cadeau-25-euros', '12585', '15033', '12791', '13910', '16210',
       '15526', '14323', '16342', '13567', '15850', '3506', '11258',
       '14338', '12599', '15665', '3383', '13211', '14657', '14899',
       '16244', '15818', '15328', '15361', '2179', '16318', '2361',
       '14095', '16328', '7086', '16586', '15240', '16024', '16013',
       '11601', '13052', '15658', '3509', '16153', '15315', '16067',
       '15318', '13514', '15360', '16030', '14855', '15436', '16230',
       '16305', '15179', '16066', '15205', '15729', '15402', '16262',
       '15860', '16578', '14509', '15473', '16138', '15829', '16144',
       '14332', '16304', '15715', '16295', '13849', '13765', '15204',
       '15462', '15944', '15776', '14945', '16160', '16147', '14626',
       '15863', '16154', '14485', '15811', '15934', '15155', '15813',
       '15629', '16553', '15413', '15434', '15662', '14581', '13914',
       '16166', '19820', '16135', '12339', '15134', '15952', '12366',
       '15706', '12882', '14768', '16317', '16525', '14149', '15704',
       '12657', '16324', '16322', '15147', '15930', '16093', '16096',
       '14865', '11669', '13854', '805', '16537', '14930', '16239',
       '15672', '15404', '13460', '12365', '13291', '15441', '16229',
       '15337', '15655', '15612', '12588', '15300', '15185', '19814',
       '14897', '15756', '14915', '12203', '15577', '15292', '14856',
       '14241', '15140', '15791', '15795', '3510', '16155', '15845',
       '15567', '15753', '15797', '15282', '15574', '15922', '15958',
       '14374', '15927', '15530', '16529', '15415', '16023', '15720',
```

```
'13209', '12771', '14679', '12640', '13096', '15429', '16062',
'15487', '11736', '11585', '15614', '16539', '14570', '15870',
'15631', '15264', '14366', '15647', '16037', '14265', '12587',
'14955', '13215', '16498', '15486', '15183', '15656', '13627',
'19822', '11049', '15794', '15946', '14696', '5646', '14100',
'14729', '15881', '16255', '16565', '16211', '15862', '13074',
'16585', '15145', '13736', '13453', '14220', '7819', '16256',
'16014', '523', '16192', '14775', '14844', '13127-1', '13073',
'2534', '15713', '13965', '15748', '13032', '4679', '16540',
'15533', '15676', '14864', '15138', '13217', '13072', '13230',
'15525', '15338', '14774', '15136', '15737', '16261', '15298',
'15664', '15792', '16057', '14912', '15178', '16472', '15561',
'15281', '16580', '15675', '10014', '15426', '802', '16159',
'15403', '15859', '15073', '16022', '15674', '13958', '15663',
'15775', '14797', '16047', '15781', '15783', '14596', '11996',
'15683', '14809', '15808', '14089', '15481', '13809', '15707',
'15201', '15583', '13313', '15767', '14692', '15731', '15613',
'16330', '14819', '13853', '16190', '14905', '11467', '12315',
'15849', '15127', '14725', '14599', '15554', '8365', '798',
'14828', '16121', '16131', '14700', '16280', '16094', '15848',
'13662', '15349', '13557', '14561', '14773', '16497', '14184',
'16031', '14980', '15839', '15690', '15688', '15072', '14300',
'16307', '15475', '15910', '12496', '15539', '15452', '15785',
'13814', '14569', '15705', '41', '15793', '15184', '13117',
'13766', '11849', '14461', '16029', '11587', '15125', '14469',
'16273', '16306', '15162', '16326', '15766', '16120', '13379',
'15036', '13572', '15582', '16045', '15095', '16010', '15149',
'13172', '7818', '15732', '15834', '15787', '304', '15341', '8344',
'15755', '15779', '15369', '8463', '15745', '15375', '12586',
'16289', '15810', '15967', '16296', '16515', '16119', '14474',
'13647', '9562', '15801', '11862', '16152', '15373', '16319',
'531', '14756', '15864', '12790', '12639', '15869', '15660',
'14680', '15161', '15163', '13517', '11933', '15440', '15784',
'15126', '15324', '14429', '15471', '11641', '14699', '19816',
'15080', '16320', '9636', '11277', '791', '15256', '15531',
'16504', '16097', '15238', '16275', '16265', '13127', '14090',
'15346', '6616', '15875', '16124', '16501', '16213', '15654',
'15489', '15480', '14981', '16560', '15951', '15466', '15807',
'15769', '15353', '14507', '16180', '14751', '15035', '15741',
'15773', '15717', '16034', '16039', '15399', '16449', '15283',
'15668', '15759', '15307', '15747', '16132', '16038', '13078',
'16041', '15564', '14141', '15677', '12045', '15026', '15730',
'15879', '15605', '1366', '15120', '14527', '14647', '15229',
'16277', '15856', '16323', '15953', '16046', '15786', '12494',
'15871', '13604', '16276', '16292', '15746', '1662', '15966',
'13982', '15949', '15022', '15734', '12589', '14600', '15146',
'15227', '15667', '15457', '16513', '38', '14451', '15566', '9937',
'15923', '16149', '11225', '13754', '13905', '14736', '14977',
'11997', '15659', '16133', '16003', '14983', '16069', '15764',
'15941', '14923', '15770', '15710', '15343', '3507', '10459',
'15432', '13516', '16462', '10814', '13659', '15106', '14712',
'14676', '13416', '15758', '15351', '15465', '15116', '15940',
'15456', '14845', '16065', '14827', '1360', '13599', '12881',
'16068', '15575', '15857', '16011', '15180'], dtype=object)
```

In [651…
```
#selection de la vcaleur id_web = bon-cadeau-25-euros
mask=df_web['id_web']=='bon-cadeau-25-euros'
df_web[mask].index
```

Out[651]:
```
Int64Index([], dtype='int64')
```

In [652…
```
#suppression id_web = bon-cadeau-25-euros
df_web.drop(df_web[mask].index, inplace=True)
df_web[mask]
```

Out[652]:

id_web	virtual	downloadable	rating_count	average_rating	total_sales	tax_status	tax_class	post_author

0 rows × 28 columns

In [578…
```
#ranger par ordre decroissant les id_web
df_web.sort_values('id_web').head()
```

Out[578]:

	id_web	virtual	downloadable	rating_count	average_rating	total_sales	tax_status	tax_class	post_auth
541	10014	0	0	0	0.0	0.0	NaN	NaN	
955	10014	0	0	0	0.0	0.0	taxable	NaN	
1503	10459	0	0	0	0.0	0.0	taxable	NaN	
1230	10459	0	0	0	0.0	0.0	NaN	NaN	
905	10775	0	0	0	0.0	0.0	NaN	NaN	

5 rows × 28 columns

In [579…
```
#selection de post_type = attachment et supression de ces lignes
masc=df_web['post_type']=='attachment'
df_web.drop(df_web[masc].index,inplace=True)
df_web.sort_values('id_web').head()
```

Out[579]:

	id_web	virtual	downloadable	rating_count	average_rating	total_sales	tax_status	tax_class	post_auth
955	10014	0	0	0	0.0	0.0	taxable	NaN	
1503	10459	0	0	0	0.0	0.0	taxable	NaN	
140	10775	0	0	0	0.0	0.0	taxable	NaN	

1466	10814	0	0	0	0.0	0.0	taxable	NaN
451	11049	0	0	0	0.0	0.0	taxable	NaN

5 rows × 28 columns

In [580…
```
#Vérification des doublons
df_web.duplicated(subset=['id_web']).sum()
```

Out[580]:
```
0
```

In [581…
```
#affichage des noms de colonnes de df_web
colonnes_web = df_web.columns
colonnes_web
```

Out[581]:
```
Index(['id_web', 'virtual', 'downloadable', 'rating_count', 'average_rating',
       'total_sales', 'tax_status', 'tax_class', 'post_author', 'post_date',
       'post_date_gmt', 'post_content', 'post_title', 'post_excerpt',
       'post_status', 'comment_status', 'ping_status', 'post_password',
       'post_name', 'post_modified', 'post_modified_gmt',
       'post_content_filtered', 'post_parent', 'guid', 'menu_order',
       'post_type', 'post_mime_type', 'comment_count'],
      dtype='object')
```

In [582…
```
# Calculer le nombre unique pour chaque colonne
df_unique = pd.DataFrame(columns=['Colonne', 'Nombre Unique'])
for colonne in colonnes_web:
    nb_valeurs_uniques = df_web[colonne].nunique()
    df_unique =pd.concat([df_unique,pd.DataFrame({'Colonne':[colonne],'Nombre Unique':
df_unique
```

Out[582]:

	Colonne	Nombre Unique
0	id_web	714
1	virtual	1
2	downloadable	1
3	rating_count	1
4	average_rating	1
5	total_sales	41
6	tax_status	1
7	tax_class	0
8	post_author	2
9	post_date	90

10	post_date_gmt	90
11	post_content	0
12	post_title	711
13	post_excerpt	677
14	post_status	1
15	comment_status	1
16	ping_status	1
17	post_password	0
18	post_name	714
19	post_modified	160
20	post_modified_gmt	160
21	post_content_filtered	0
22	post_parent	1
23	guid	714
24	menu_order	1
25	post_type	1
26	post_mime_type	0
27	comment_count	1

In [583…
```
colonnes_a_supprimer = df_unique[(df_unique['Nombre Unique'] == 0) | (df_unique['Nombr
colonnes_a_supprimer
```

Out[583]:
```
['virtual',
 'downloadable',
 'rating_count',
 'average_rating',
 'tax_status',
 'tax_class',
 'post_content',
 'post_status',
 'comment_status',
 'ping_status',
 'post_password',
 'post_content_filtered',
 'post_parent',
 'menu_order',
 'post_type',
 'post_mime_type',
 'comment_count']
```

In [584…
```
# Suppression des colonnes non pertinentes
df_web = df_web.drop(columns=colonnes_a_supprimer)
```

In [585…
```
df_web.head()
```

Out[585]:

	id_web	total_sales	post_author	post_date	post_date_gmt	post_title	post_excerpt	post_name	post_n
2	15075	3.0	2.0	14/02/2018	14/02/2018	Parés Baltà Penedès Indigena 2017	Des couleurs et aromes intenses où le fruit et...	pares-balta-penedes-indigena-2017	20/

3	16209	6.0	2.0	14/02/2018	14/02/2018	Maurel Cabardès Tradition 2017	Un joli nez aux arômes de fruits rouges, de ca...	maurel-cabardes-tradition-2017	05/
5	13895	0.0	2.0	19/03/2019	19/03/2019	Château Saransot-Dupré Bordeaux Blanc 2016	<span style="display: inline !important; float...	chateau-saransot-dupre-bordeaux-blanc-2016	25/
6	12857	0.0	2.0	12/04/2018	12/04/2018	Château de Meursault Puligny-Montrachet 1er Cr...	Il présente une grande fraîcheur minérale au n...	chateau-de-puligny-montrachet-1cru-champ-canet...	06/
9	14106	0.0	2.0	08/06/2019	08/06/2019	Stéphane Tissot Château-Chalon 2011	Ce vin peut-être dégusté sur sa jeunesse mais ...	stephane-tissot-chateau-chalon-2011	29/

In [586…
```
df_web.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 714 entries, 2 to 1510
Data columns (total 11 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   id_web             714 non-null    object 
 1   total_sales        714 non-null    float64
 2   post_author        714 non-null    float64
 3   post_date          714 non-null    object 
 4   post_date_gmt      714 non-null    object 
 5   post_title         714 non-null    object 
 6   post_excerpt       714 non-null    object 
 7   post_name          714 non-null    object 
 8   post_modified      714 non-null    object 
 9   post_modified_gmt  714 non-null    object 
 10  guid               714 non-null    object 
dtypes: float64(2), object(9)
memory usage: 66.9+ KB
```

2 - Preparation du fichier_erp</h3>

In [587…
```
df_erp.head()
```

Out[587]:

	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	outofstock

df_erp.info()

In [588…
```
#caster la variable price en float
df_erp.price = df_erp.price.str.replace(',', '.').astype(float)
```

In [589…
```
#caster la variable product_id en str
df_erp['product_id']=df_erp['product_id'].astype(str)
df_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    object 
 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(2), object(2)
memory usage: 32.4+ KB
```

In [590…
```
#selection  des lignes avec stock_quantity = 0
stock_zero= df_erp[df_erp['stock_quantity'] == 0 ]
stock_zero.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 181 entries, 0 to 775
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   product_id      181 non-null    object 
 1   onsale_web      181 non-null    int64  
 2   price           181 non-null    float64
 3   stock_quantity  181 non-null    int64  
 4   stock_status    181 non-null    object 
dtypes: float64(1), int64(2), object(2)
memory usage: 8.5+ KB
```

In [591…
```
stock_zero['stock_status'].unique()
```

Out[591]:
```
array(['outofstock', 'instock'], dtype=object)
```

In [592…
```
# Identifier les produits avec un statut 'instock' alors que la quantité en_stock est
stock0=stock_zero[stock_zero['stock_status'] == 'instock']
stock0
```

Out[592]:

	product_id	onsale_web	price	stock_quantity	stock_status
443	4954	1	25.0	0	instock

In [593…
```
stock_zero = df_erp[df_erp['stock_quantity'] == 0]
stock_zero = stock_zero[stock_zero['stock_status'] != 'instock']
```

In [594…
```
#suppression des lignes stock_status =instock et stock_quantity = 0
condition = (stock_zero['stock_status'] == 'instock') & (stock_zero['stock_quantity'
indices_a_supprimer = stock_zero[condition].index
stock_zero.drop(indices_a_supprimer, inplace=True)
stock_zero['stock_status'].unique()
```

Out[594]:
```
array(['outofstock'], dtype=object)
```

In [595…
```
price_list=df_erp.price.unique()
price_list
```

Out[595]:
```
array([ 24.2 ,  34.3 ,  20.8 ,  14.1 ,  46.  ,  32.7 ,  31.2 ,  60.  ,
        42.6 ,  80.  ,  18.3 ,  22.8 ,  19.3 ,  21.8 ,   7.7 ,  33.7 ,
        44.3 ,  71.6 ,  86.1 ,  12.7 ,   8.7 ,  11.9 ,  14.5 ,  14.4 ,
        19.5 ,  22.  ,  16.6 ,  23.4 ,  33.2 ,  32.  ,  77.8 ,  14.7 ,
        14.05,  22.9 ,  44.  ,  37.  ,  39.  ,  17.  ,  23.2 ,  19.  ,
        16.4 ,  73.  ,  47.  ,  13.7 ,  12.6 ,  12.8 ,  22.1 ,  15.8 ,
        16.3 ,   9.7 ,   6.8 ,  35.  ,  31.7 , 100.  ,  23.  ,  88.4 ,
        29.8 ,  25.7 ,  77.4 ,  53.  ,  49.  ,  29.5 ,  33.  ,  37.5 ,
        69.  ,  59.  ,  19.2 ,  29.  ,   9.8 ,  20.35,  12.  ,  18.5 ,
         9.3 ,  11.6 ,  14.3 ,  10.8 ,   7.6 ,  20.5 ,  18.2 ,   9.  ,
         7.8 ,   5.7 ,  13.5 ,  11.5 ,  24.  ,  16.7 ,  21.4 ,  13.3 ,
         9.5 ,  12.1 ,  17.8 ,  27.2 ,   9.4 ,   5.8 ,  38.  ,   9.9 ,
        11.3 ,   6.7 ,  73.5 ,  79.8 ,  48.5 ,  39.8 ,  58.8 ,  26.5 ,
        13.4 ,  17.1 ,   8.9 ,  17.2 ,  16.9 ,  29.9 ,   9.6 ,  11.1 ,
        20.  ,  28.  ,   8.6 ,  15.3 ,  14.8 ,  59.6 ,  26.9 ,  24.4 ,
        32.1 ,  12.2 ,  15.2 ,  10.2 ,  15.5 ,   9.2 ,  12.9 ,  14.9 ,
        17.6 ,  24.8 ,  21.5 ,  18.9 ,  27.  ,  41.  ,  69.8 ,  38.6 ,
        26.7 ,  39.1 ,  17.5 ,  30.  ,   8.1 ,  10.7 ,  10.9 ,  35.5 ,
        83.  ,  79.5 , 225.  , 126.5 ,  51.6 ,  77.  ,  85.6 ,  49.5 ,
        57.  ,  59.8 ,  27.5 ,  62.  ,  62.5 , 176.  , 108.5 ,  68.1 ,
       157.  , 104.  ,  28.1 ,  21.7 ,  30.5 ,  28.5 ,  67.2 ,  40.  ,
       109.6 ,  32.3 , 144.  ,  43.9 ,  61.6 ,  41.8 ,  36.9 ,  16.1 ,
        31.5 ,  32.2 ,  50.1 ,  11.8 ,  13.1 ,  26.2 ,  20.6 ,  67.5 ,
        30.6 ,  16.5 ,  52.4 ,  52.9 ,  58.3 ,  39.6 ,  62.4 ,  76.8 ,
        50.  ,  24.3 ,  25.3 ,  36.2 ,  33.4 ,  40.2 ,  43.  ,  48.8 ,
        23.6 ,  21.  ,  12.3 ,  20.2 ,  21.9 ,  19.8 ,  13.2 ,   6.3 ,
         7.1 ,   9.1 ,  18.1 ,  14.  ,  30.1 ,  34.5 ,  23.8 ,  31.6 ,
        16.8 ,  32.6 ,  55.4 ,  18.4 ,  18.6 ,  12.5 ,  15.9 ,  26.  ,
         7.4 ,  12.4 ,  13.8 ,  27.9 ,  13.9 ,  10.1 ,  13.6 ,  18.7 ,
        41.6 ,  78.  ,   6.5 ,   8.5 ,   8.2 ,   8.3 ,  14.6 ,  28.4 ,
        20.1 ,  21.2 , 102.3 , 137.  ,  53.2 ,  25.9 ,  17.3 ,  37.2 ,
         7.  ,   7.9 ,  27.8 ,  22.2 ,  25.  ,  10.  ,  23.7 ,  16.45,
        27.3 , 217.5 ,  64.9 ,  48.7 ,  59.4 , 105.  ,  55.6 ,  -8.  ,
        15.4 ,  45.  , 112.  ,  86.8 ,  62.1 ,  22.5 ,   7.5 ,  52.6 ,
        67.  ,  59.9 ,  65.  ,  84.7 ,  43.3 ,  18.  ,  28.8 ,  35.3 ,
        16.2 ,   7.2 ,  54.8 ,  42.  ,  10.4 ,  17.9 ,  21.6 ,  43.5 ,
        48.4 ,  60.4 ,  65.9 ,  24.6 ,  36.3 ,  57.7 ,  58.  ,  30.8 ,
        92.  ,  19.9 ,  34.7 ,  83.7 ,  63.4 , 124.8 ,  56.4 ,  38.4 ,
        71.3 ,  10.3 ,  44.6 ,  13.  ,  44.5 ,  29.4 ,  57.6 ,  11.  ,
        73.3 ,  42.1 ,  24.5 ,  42.2 ,  35.6 , 175.  ,  33.6 ,  34.4 ,
        29.7 ,  32.8 ,  29.2 ,  17.4 ,  34.2 ,  63.5 ,  56.  ,  41.2 ,
        55.  , 191.3 ,  26.6 ,  24.7 ,  18.25,  35.1 ,  18.8 ,  17.7 ,
        36.  ,  93.  , 122.  , 114.  ,  74.5 ,  42.5 ,  47.5 ,  56.3 ,
        71.5 ,  71.7 ,  38.5 ,  40.7 ,  34.8 ,  74.8 ,  39.2 ,  14.2 ,
       135.  ,  10.6 ,   5.2 , 105.6 , 116.4 ,  31.  ,  25.2 , 115.  ,
       121.  ,  99.  ,  23.5 ,  26.4 ,  20.4 ,  45.9 ,  40.5 ,  22.4 ,
        72.  ,  68.3 ,  -1.  ,  51.  ,  35.2 ,  37.7 ,  47.2 ,  52.7 ,
        50.4 ,  27.7 ,  46.5 ,  50.5 ,  49.9 ,   8.4 ])
```

In [596…
```
valeurs_negatives = [x for x in price_list if x < 0]
valeurs_negatives
```

Out[596]:
```
[-8.0, -1.0]
```

In [604…
```
df_erp = df_erp[df_erp['price'] >= 0]
df_erp
```

Out[604]:

	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

823 rows × 5 columns

3 - Preparation du fichier_liaison</h3>

In [605…
```
df_liaison.head()
```

Out[605]:

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

In [606…
```
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 [607…
```
#suppression des valeurs manquantes
df_liaison.dropna(inplace=True)
df_liaison.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 734 entries, 0 to 824
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   product_id  734 non-null    int64 
```

```
 1   id_web       734 non-null    object
dtypes: int64(1), object(1)
memory usage: 17.2+ KB
```

In [608…
```
# caster product_id en str(chaine de caractére)
df_liaison['product_id']=df_liaison['product_id'].astype(str)
df_liaison.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 734 entries, 0 to 824
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   product_id  734 non-null    object
 1   id_web      734 non-null    object
dtypes: object(2)
memory usage: 17.2+ KB
```

PROJET 05 DATA ANALYST

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

PARTIE ANALYSE DES DONNEES </h2>

In [609…
```
#jointure externe entre df_erp et df_liaison suivant la colonne product_id
erp_liaison = pd.merge(df_erp, df_liaison, on='product_id', how='outer',indicator=
erp_liaison.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 823 entries, 0 to 822
Data columns (total 7 columns):
 #   Column           Non-Null Count  Dtype   
---  ------           --------------  -----   
 0   product_id       823 non-null    object  
 1   onsale_web       823 non-null    int64   
 2   price            823 non-null    float64 
 3   stock_quantity   823 non-null    int64   
 4   stock_status     823 non-null    object  
 5   id_web           734 non-null    object  
 6   merge_indicator  823 non-null    category
dtypes: category(1), float64(1), int64(2), object(3)
memory usage: 45.9+ KB
```

In [610…
```
#verification s'il y'a des valeurs de product_id existent uniquement dans le jeu d
erp_liaison[erp_liaison['merge_indicator']=='right_only']
```

Out[610]:

	product_id	onsale_web	price	stock_quantity	stock_status	id_web	merge_indicator

In [611…
```
#verification s'il y'a des valeurs de product_id existent uniquement dans le jeu d
erp_liaison[erp_liaison['merge_indicator']=='left_only']
```

Out[611]:

	product_id	onsale_web	price	stock_quantity	stock_status	id_web	merge_indicator

19	4055	0	86.1	1	outofstock	NaN	left_only
49	4090	0	73.0	6	outofstock	NaN	left_only
50	4092	0	47.0	6	outofstock	NaN	left_only
119	4195	0	14.1	0	outofstock	NaN	left_only
131	4209	0	73.5	0	outofstock	NaN	left_only
...	...	...	...	...	...	...	...
815	7196	0	31.0	55	instock	NaN	left_only
816	7200	0	31.0	6	instock	NaN	left_only
817	7201	0	31.0	18	instock	NaN	left_only
818	7203	0	45.0	30	instock	NaN	left_only
819	7204	0	45.0	9	instock	NaN	left_only

89 rows × 7 columns

In [612…
```
#suppression des ligne merge_indicator = left_only
erp_liaison.drop(erp_liaison[erp_liaison['merge_indicator']=='left_only'].index,ir
erp_liaison[erp_liaison['merge_indicator']=='left_only']
```

Out[612]:

product_id	onsale_web	price	stock_quantity	stock_status	id_web	merge_indicator

In [613…
```
erp_liaison.head()
```

Out[613]:

	product_id	onsale_web	price	stock_quantity	stock_status	id_web	merge_indicator
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

In [614…
```
#jointure externe entre erp_liaison et df_web suivant la colonne id_web
df_jointure = pd.merge(erp_liaison,df_web,on='id_web', how='outer',indicator='merg
df_jointure.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 734 entries, 0 to 733
Data columns (total 18 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   product_id        734 non-null    object
 1   onsale_web        734 non-null    int64
 2   price             734 non-null    float64
 3   stock_quantity    734 non-null    int64
 4   stock_status      734 non-null    object
 5   id_web            734 non-null    object
 6   merge_indicator   734 non-null    category
 7   total_sales       714 non-null    float64
 8   post_author       714 non-null    float64
 9   post_date         714 non-null    object
 10  post_date_gmt     714 non-null    object
 11  post_title        714 non-null    object
```

```
 12  post_excerpt        714 non-null    object
 13  post_name           714 non-null    object
 14  post_modified       714 non-null    object
 15  post_modified_gmt   714 non-null    object
 16  guid                714 non-null    object
 17  merge_indicator_2   734 non-null    category
dtypes: category(2), float64(3), int64(2), object(11)
memory usage: 99.2+ KB
```

In [615…]
```
df_jointure.head()
```

Out[615]:

	product_id	onsale_web	price	stock_quantity	stock_status	id_web	merge_indicator	total_sales	post
0	3847	1	24.2	0	outofstock	15298	both	6.0	
1	3849	1	34.3	0	outofstock	15296	both	0.0	
2	3850	1	20.8	0	outofstock	15300	both	0.0	
3	4032	1	14.1	0	outofstock	19814	both	3.0	
4	4039	1	46.0	0	outofstock	19815	both	0.0	

In [616…]
```
#verification s'il y'a des valeurs de id_web existent uniquement dans le jeu de do
df_jointure[df_jointure['merge_indicator_2'] == 'right_only' ]
```

Out[616]:

	product_id	onsale_web	price	stock_quantity	stock_status	id_web	merge_indicator	total_sales	post_

In [617…]
```
#verification s'il y'a des valeurs de id_web existent uniquement dans le jeu de do
df_jointure[df_jointure['merge_indicator_2'] == 'leftt_only' ]
```

Out[617]:

	product_id	onsale_web	price	stock_quantity	stock_status	id_web	merge_indicator	total_sales	post_

Chiffre d'affaire par produit

In [618…]
```
#Chiffre d'affaire par produit en ordre decroissant
df_jointure['cf_product']=df_jointure['total_sales'] * df_jointure['price']
df_jointure.sort_values(by='id_web', ascending=False).head()
```

Out[618]:

	product_id	onsale_web	price	stock_quantity	stock_status	id_web	merge_indicator	total_sales	p
418	4954	1	25.0	0	instock	bon-cadeau-25-euros	both	10.0	
410	4932	1	25.7	0	outofstock	9937	both	4.0	
213	4396	1	62.0	7	instock	9636	both	0.0	
204	4357	1	39.0	0	outofstock	9562	both	0.0	
518	5574	1	59.6	9	instock	8463	both	0.0	

Chiffre d'affaire total

In [619…

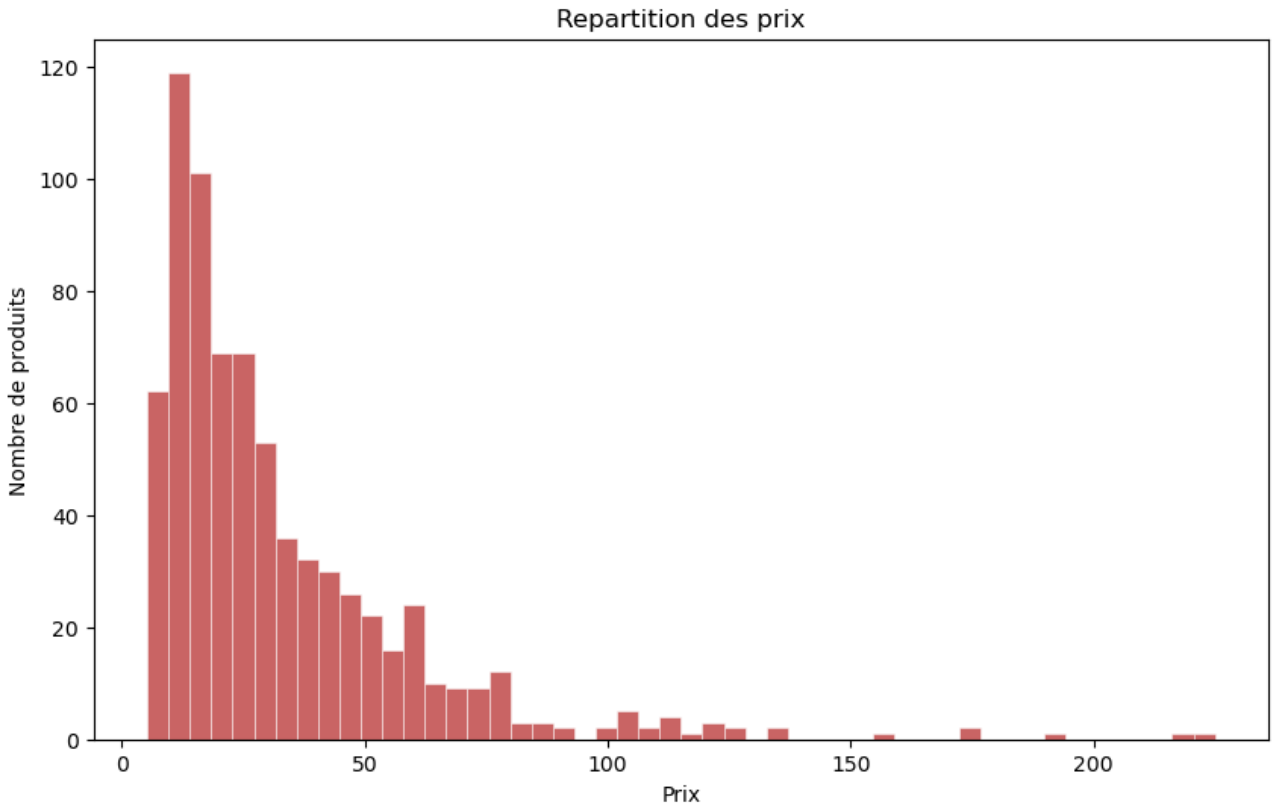
```
#Chiffre d'affaire total
cf_total= df_jointure['cf_product'].sum()
print(f" le chiffre d'affaires total est de {cf_total:,.2f} €")
```

```
 le chiffre d'affaires total est de 70,568.60 €
```

Repartition des prix par produit

In [620…

```
plt.figure(figsize=(10, 6))
plt.hist(df_jointure['price'], bins=50, color='firebrick', alpha=0.7,edgecolor='wh
plt.xlabel("Prix")
plt.ylabel("Nombre de produits")
plt.title("Repartition des prix")
plt.show()
```



Vleurs abérantes des prix

In [621…

```
# identifier les outliers:les valeurs en dehors de la plage [Q1- 1.5 * IQR,Q3+ 1.5

q1 = df_jointure['price'].quantile(0.25)
q3 = df_jointure['price'].quantile(0.75)

# Calcul de la plage interquartile (IQR)
iqr = q3 - q1

# Définition des seuils pour détecter les valeurs aberrantes
seuil_inf = q1 - 1.5 * iqr
seuil_sup = q3 + 1.5 * iqr

# Filtrer les valeurs aberrantes
outliers =df_jointure[( df_jointure['price'] < seuil_inf) | (df_jointure['price']
outliers
```

Out[621]:

	product_id	onsale_web	price	stock_quantity	stock_status	id_web	merge_indicator	total_sales	p
63	4115	1	100.0	11	instock	15382	both	0.0	
65	4132	1	88.4	5	instock	11668	both	0.0	
200	4352	1	225.0	0	outofstock	15940	both	5.0	

202	4355	1	126.5	2	instock	12589	both	11.0
206	4359	1	85.6	0	outofstock	13853	both	1.0
219	4402	1	176.0	8	instock	3510	both	13.0
220	4404	1	108.5	2	instock	3507	both	2.0
222	4406	1	157.0	3	instock	7819	both	0.0
223	4407	1	104.0	6	instock	3509	both	1.0
229	4582	1	109.6	7	instock	12857	both	0.0
386	4903	1	102.3	20	instock	14805	both	0.0
387	4904	1	137.0	13	instock	14220	both	5.0
434	5001	1	217.5	20	instock	14581	both	0.0
439	5007	1	105.0	17	instock	12791	both	0.0
440	5008	1	105.0	10	instock	11602	both	0.0
447	5025	1	112.0	0	outofstock	13914	both	0.0
448	5026	1	86.8	2	instock	13913	both	0.0

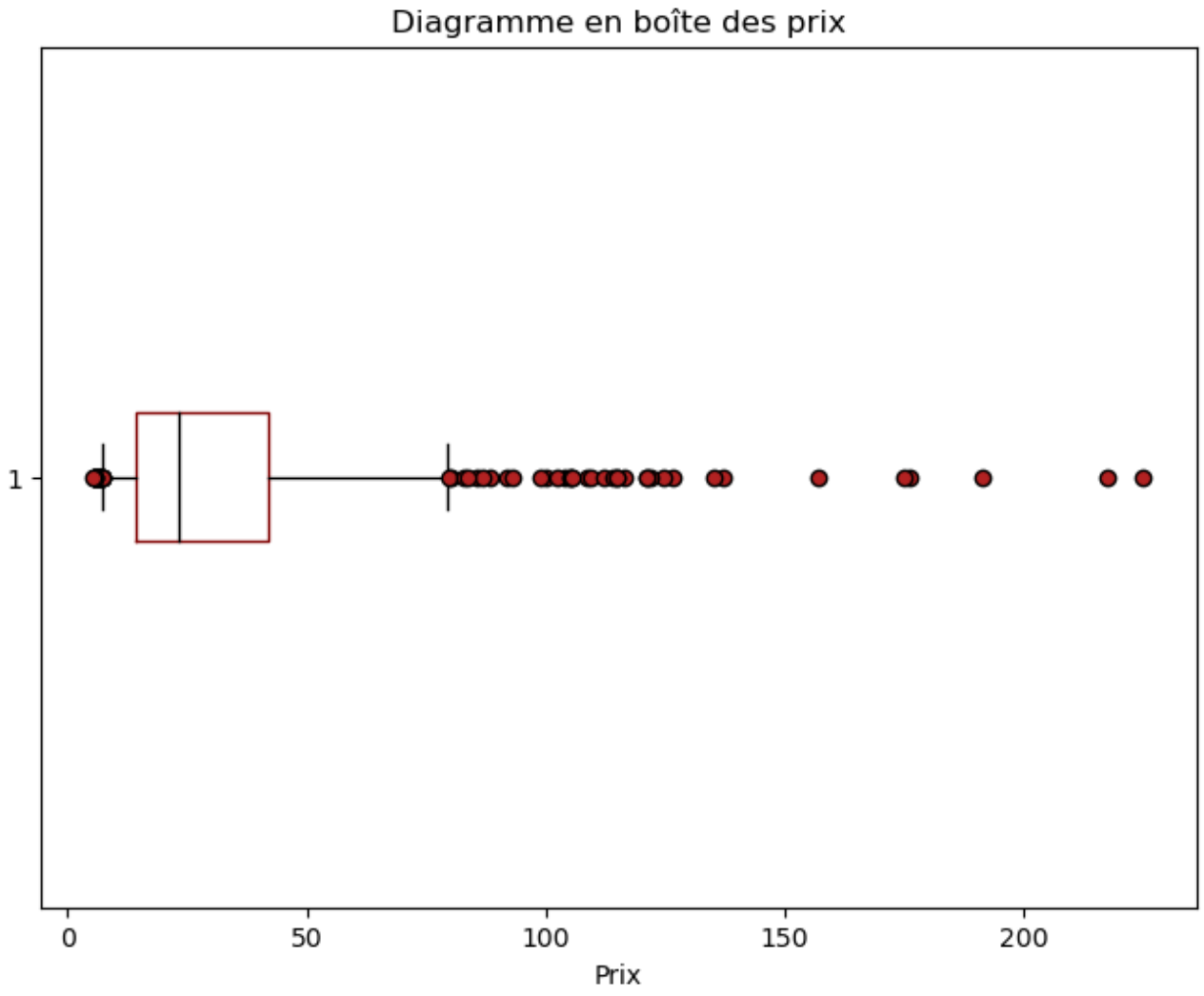
514	5565	1	92.0	0	outofstock	19822	both	0.0
519	5580	1	83.7	18	instock	13982	both	0.0
524	5612	1	124.8	12	instock	14915	both	0.0
566	5767	1	175.0	12	instock	15185	both	0.0
601	5892	1	191.3	10	instock	14983	both	3.0
616	5916	1	93.0	3	instock	14774	both	0.0
617	5917	1	122.0	4	instock	14775	both	0.0
618	5918	1	114.0	8	instock	14773	both	0.0
661	6126	1	135.0	10	instock	14923	both	2.0
666	6201	1	105.6	7	instock	14596	both	0.0
667	6202	1	116.4	14	instock	15126	both	0.0
672	6212	1	115.0	2	instock	13996	both	2.0
673	6213	1	121.0	7	instock	15072	both	0.0
674	6214	1	99.0	7	instock	11601	both	0.0

675	6215	1	115.0	4	instock	12790	both	0.0
676	6216	1	121.0	6	instock	15070	both	0.0

In [622…
```
plt.figure(figsize=(8, 6))
box_color = 'maroon'
median_color = 'black'
outlier_color = 'firebrick'

prices = df_jointure['price']

plt.boxplot(prices, vert=False, boxprops=dict(color=box_color), medianprops=dict(c
plt.title('Diagramme en boîte des prix')
plt.xlabel('Prix')
plt.show()
```



In [623…
```
#analyse des valeurs abérantes des prix
outliers.describe()
```

Out[623]:

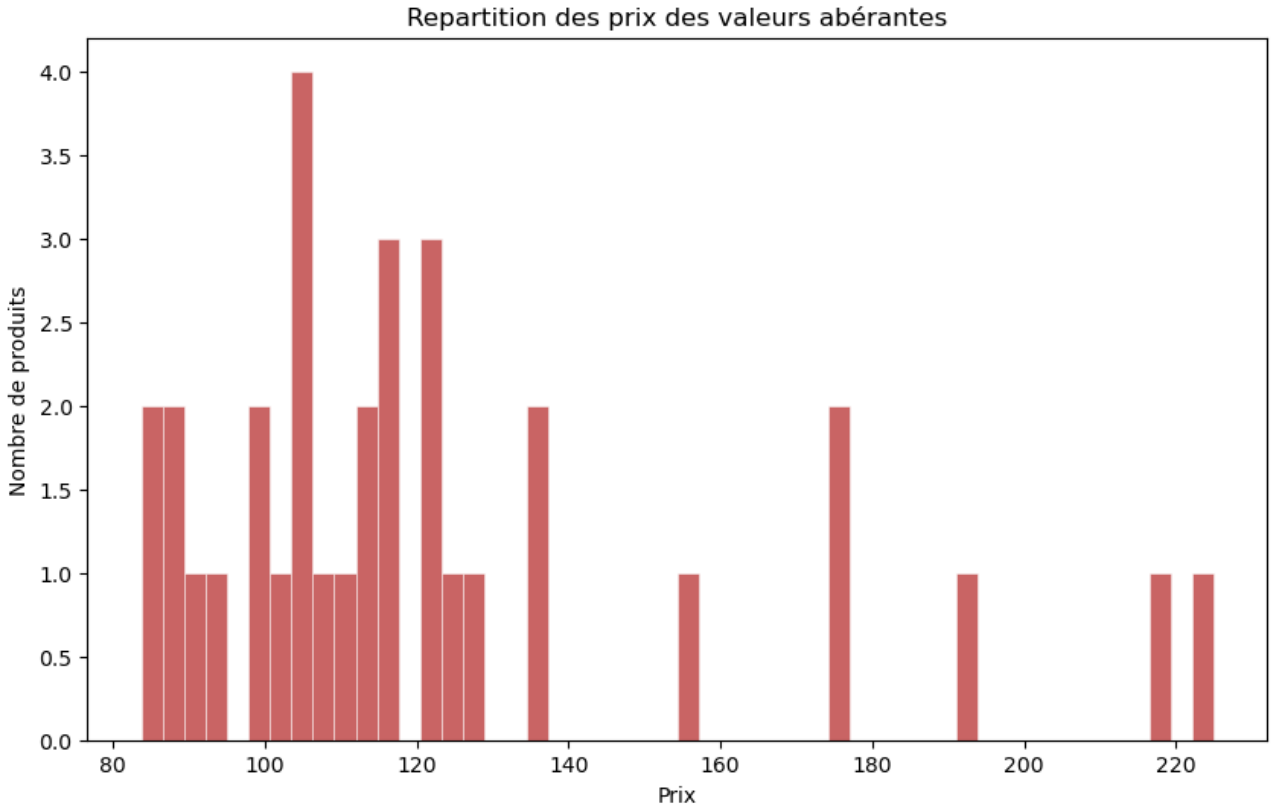
	onsale_web	price	stock_quantity	total_sales	post_author	cf_product
count	33.0	33.000000	33.000000	33.000000	33.0	33.000000

mean	1.0	123.333333	7.575758	1.363636	2.0	211.212121
std	0.0	36.206626	5.809638	3.070201	0.0	498.570185
min	1.0	83.700000	0.000000	0.000000	2.0	0.000000
25%	1.0	102.300000	3.000000	0.000000	2.0	0.000000
50%	1.0	114.000000	7.000000	0.000000	2.0	0.000000
75%	1.0	126.500000	11.000000	1.000000	2.0	104.000000
max	1.0	225.000000	20.000000	13.000000	2.0	2288.000000

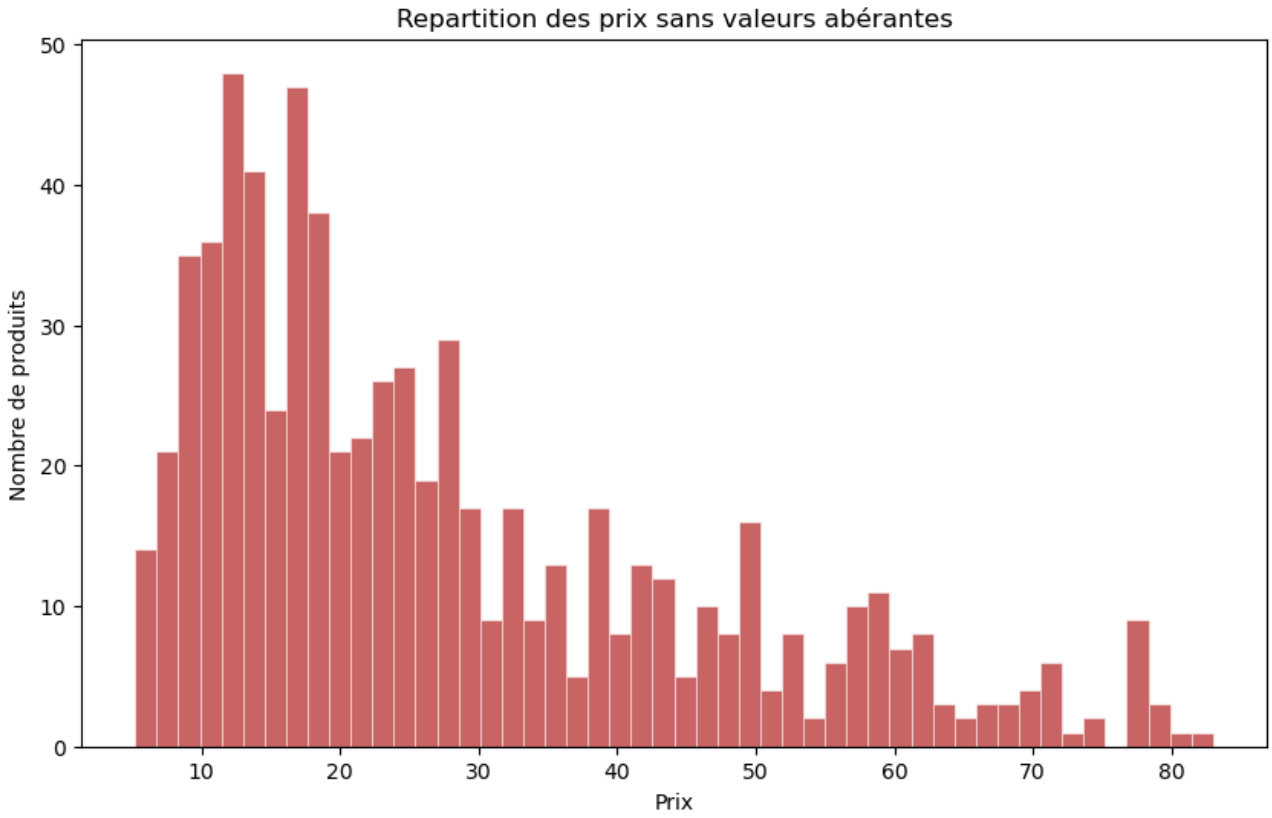
In [624…
```
outliers.price.unique()
```

Out[624]:
```
array([100. ,  88.4, 225. , 126.5,  85.6, 176. , 108.5, 157. , 104. ,
       109.6, 102.3, 137. , 217.5, 105. , 112. ,  86.8,  92. ,  83.7,
       124.8, 175. , 191.3,  93. , 122. , 114. , 135. , 105.6, 116.4,
       115. , 121. ,  99. ])
```

In [625…
```
#Repartition des prix des valeurs abérantes
plt.figure(figsize=(10, 6))
plt.hist(df_jointure[(( df_jointure['price'] < seuil_inf) | (df_jointure['price']
plt.xlabel("Prix")
plt.ylabel("Nombre de produits")
plt.title("Repartition des prix des valeurs abérantes")
plt.show()
```



In [626…
```
#Repartition des prix sans valeurs abérantes
plt.figure(figsize=(10, 6))
plt.hist(df_jointure[~(( df_jointure['price'] < seuil_inf) | (df_jointure['price']
plt.xlabel("Prix")
plt.ylabel("Nombre de produits")
plt.title("Repartition des prix sans valeurs abérantes")
plt.show()
```


Repartition des prix sans valeurs abérantes
Nombre de produits
Prix
0
10
20
30
40
50
10
20
30
40
50
60
70
80

In [ ]:

In [ ]:

In [ ]:

In [ ]: