

# ANALYSE DU STOCK ET DES VENTES DU SITE BOTTLENECK

Simon Doussin  
Business Intelligence Analyst  
Projet 6

# Analyses Exploratoires des Données

---

- *Fichier : df\_erp*
- *825 lignes / 6 colonnes*
- *Traitement réalisés*
  - *Nettoyages des données*
    - *Doublon : aucun*
    - *Prix inférieur à 0 : supprimé*
    - *Stock inférieur à 0 supprimé*

	product_id	onsale_web	price	stock_quantity	purchase_price
count	825.00	825.00	825.00	825.00	825.00
mean	5162.60	0.87	32.19	21.59	16.94
std	902.64	0.34	26.71	21.93	14.56
min	3847.00	0.00	-20.00	-10.00	2.74
25%	4348.00	1.00	14.50	7.00	7.59
50%	4907.00	1.00	24.30	18.00	12.71
75%	5805.00	1.00	42.00	30.00	22.02
max	7338.00	1.00	225.00	145.00	137.81

# Analyses Exploratoires des Données

- *Fichier : df\_web*
- *1513 lignes / 29 colonnes*
- *Traitement réalisés*
  - *Nettoyages des données*
    - *Colonne supprimée car aucune valeur : 'tax\_class', 'post\_content', 'post\_password', 'post\_content\_filtered'*
    - *Ligne supprimée si NaN : colonne 'sku'*

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	product_type	1429	non-null	object
13	post_title	1430	non-null	object
14	post_excerpt	716	non-null	object
15	post_status	1430	non-null	object
16	comment_status	1430	non-null	object
17	ping_status	1430	non-null	object
18	post_password	0	non-null	float64
19	post_name	1430	non-null	object
20	post_modified	1430	non-null	datetime64[ns]
21	post_modified_gmt	1430	non-null	datetime64[ns]
22	post_content_filtered	0	non-null	float64
23	post_parent	1430	non-null	float64
24	guid	1430	non-null	object
25	menu_order	1430	non-null	float64
26	post_type	1430	non-null	object
27	post_mime_type	714	non-null	object
28	comment_count	1430	non-null	float64

# Analyses Exploratoires des Données

---

- *Fichier : df\_liaison*
- *825 lignes / 2 colonnes*
- *Traitement réalisés*
  - *Nettoyages des données*
    - *Ligne supprimée si NaN : 'id\_web'*

```
0  id_web  734 non-null  object
1  product_id  825 non-null  int64
```

# Fusion ou consolidations des données

- *Choix des attributs :*

Table A	Table B	Fonction	Type de jointure	Colonne
df_erp	df_liaison	.merge()	LEFT	product_id

- *Vigilances particulières au cours du traitements*

*S'assurer que le nombre de ligne du df\_merge est bien égal au nombre de ligne de la table de gauche : c'est le cas.*

	product_id	onsale_web	price	stock_quantity	stock_status	purchase_price	id_web	merge
0	3847	1	24.2	16	instock	12.88	15298	both
1	3849	1	34.3	10	instock	17.54	15296	both
2	3850	1	20.8	0	outofstock	10.64	15300	both
3	4032	1	14.1	26	instock	6.92	19814	both
4	4039	1	46.0	3	instock	23.77	19815	both
...	...	...	...	...	...	...	...	...
815	7203	0	45.0	30	instock	23.48	NaN	both
816	7204	0	45.0	9	instock	24.18	NaN	both
817	7247	1	54.8	6	instock	27.18	13127-1	both
818	7329	0	26.5	14	instock	13.42	14680-1	both
819	7338	1	16.3	40	instock	8.00	16230	both

820 rows x 8 columns

# Fusion ou consolidations des données

- *Choix des attributs :*

Table A	Table B	Fonction	Type de jointure	Colonne
df_merge	df_web	.merge()	INNER	Table A : 'id_web' Table B : 'sku'

- *Vigilances particulières au cours du traitements*

Après le merge (outer) :

```
both      1426
left_only    20
right_only    2
Name: count, dtype: int64
```

-> puis utilisation du « inner »

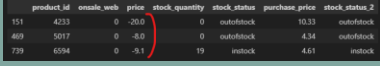
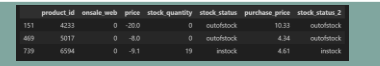
```
both      1426
left_only    0
right_only    0
Name: count, dtype: int64
```

Supprimer articles ayant un prix inférieur à 0 :

	product_id	onsale_web	price	stock_quantity	stock_status	purchase_price	stock_status_2
151	4233	0	-20.0	0	outofstock	10.33	outofstock
469	5017	0	-8.0	0	outofstock	4.34	outofstock
739	6594	0	-9.1	19	instock	4.61	instock

# Analyses univariées du prix

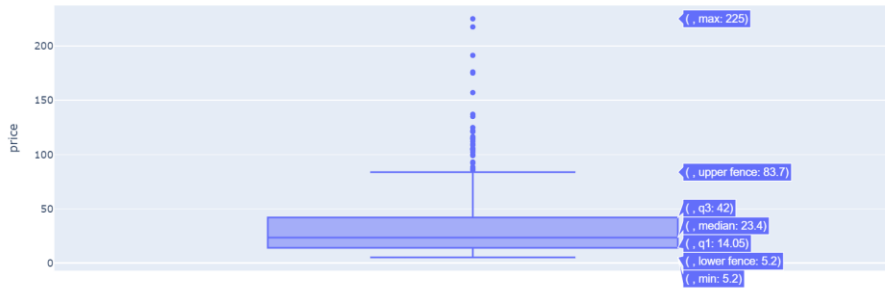
- *Méthodes statistiques employés*

Ce qu'on veut observer	Méthode	Résultat
Le prix minimum	.min()	
Le prix maximum	.max()	225 €
Les prix inférieurs à 0	.loc()	

- *Graphique avec commentaire des résultats*
- *Limites éventuelles de l'analyse*

# Analyses univariées du prix

- Graphique avec commentaire des résultats



Observation :

- Min = 5,2
- Max = 225
- Médiane = 23,4
- Outliers > 83,7
- Moyenne = 32,32
- Ecart-type = 27,6
- IQR = 28,1

Calcul de l'intervalle interquartile :

```
#### Q1 = médiane des prix inférieur à la médiane
#### Q2 = médiane des prix
#### Q3 = médiane des prix supérieur à la médiane
median_prix_q2 = df_merge2['price'].median()
print("La médiane de df_erp['price'] est de", median_prix_q2)
ligne_q1 = df_merge2[df_merge2['price'] < median_prix_q2]
display(ligne_q1)
mediane_q1 = ligne_q1['price'].median()
print('La médiane de Q1 est de', mediane_q1)
ligne_q3 = df_merge2[df_merge2['price'] > median_prix_q2]
display(ligne_q3)
mediane_q3 = ligne_q3['price'].median()
print('La médiane de Q3 est de', mediane_q3)
iqr_price = mediane_q3 - mediane_q1
print("L'IQR est de :", round(iqr_price,2))
```



## Analyses complémentaires CA, quantités, stocks, taux de marge et correlations

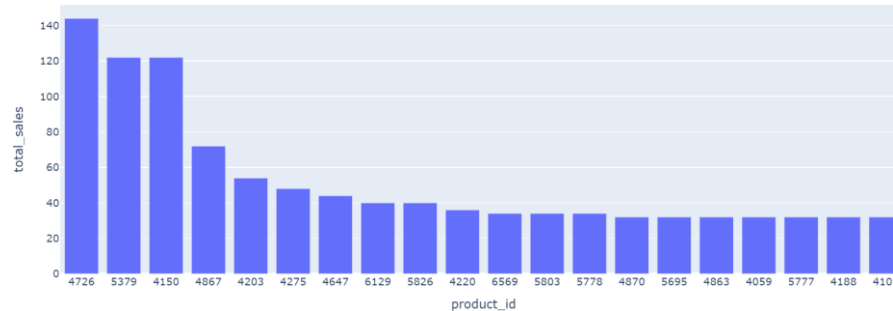
---

- *Méthodes statistiques employés*

Ce qu'on veut observer	Méthode	Résultat
Le chiffre d'affaires	Prix * quantité vendue	153392,1€
La proportion d'article représentant 80% du CA	Somme cumulée de la part du CA. Nb d'article représentant 80% du CA. Proportion	Nb article @80% CA : 853 Proportion : 59,8% des articles permettent d'atteindre 80% du CA

## Analyses complémentaires CA, quantités, stocks, taux de marge et correlations

- *Graphique avec commentaire des résultats*



Résultat :

Le produit n°4726 (vin) a vendu 144 articles, le n°5379 et 4150 en ont vendu 122 respectivement.

	product_id	product_type	stock_quantity	id_web
0	4726	Vin	0	14950
9	4726	Vin	0	14950

## Analyses complémentaires CA, quantités, stocks, taux de marge et correlations

- *Méthodes statistiques employés*

Ce qu'on veut observer	Méthode	Résultat
La proportion d'article représentant 80% des quantités vendues	Somme cumulée de la part du CA. Nb d'article représentant 80% du CA. Proportion	Nb article @80% des quantités : 856 Proportion : 60,03% des articles permettent d'atteindre 80% des quantités vendues.
Quantité d'article en stock	.sum()	Il y a 33 480 articles en stock dont 1 article qui n'est pas vendu sur internet *.

\* Article qui n'est pas vendu sur internet :

product_id	product_type	onsale_web	price	stock_quantity
4200	Vin	0	5.8	33

## Analyses complémentaires CA, quantités, stocks, taux de marge et correlations

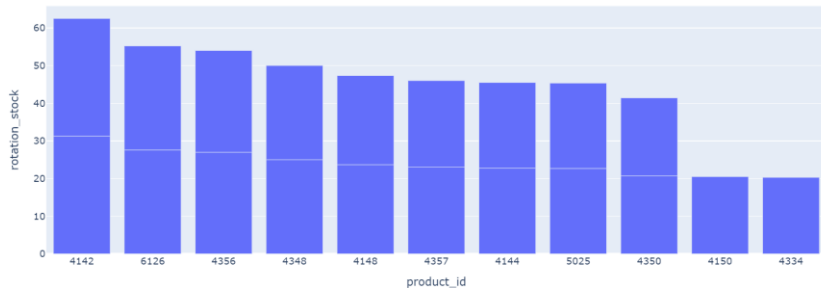
---

- *Méthodes statistiques employés*

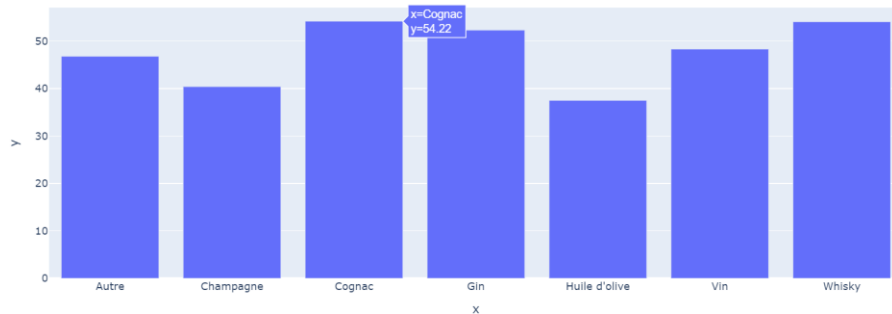
Ce qu'on veut observer	Méthode	Résultat
Rotation de stock	Quantité en stock / Ventes totales	Cf graphique

## Analyses complémentaires CA, quantités, stocks, taux de marge et correlations

- Graphique avec commentaire des résultats



Le produit n°4142 a une capacité de 31 mois encore en stock.  
Le produit n°6126 a une capacité de 26 mois encore en stock.  
En moyenne il y a 2,96 mois de stock sur l'ensemble des produits.



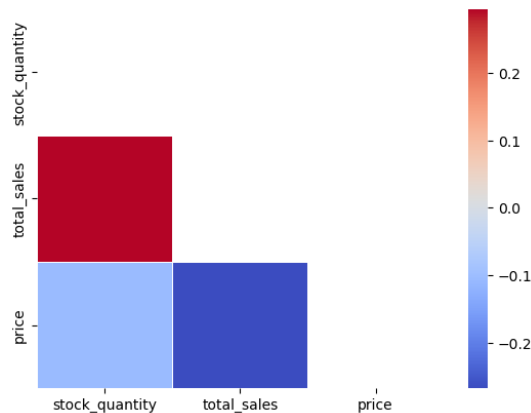
Le cognac a le taux de marge moyen le plus élevé (54,2).  
L'huile d'olive possède le taux de marge moyen le moins élevé avec 37,5.  
Le taux de marge positif le plus élevé est le whisky avec 56,5 (product\_id = 5760) tandis que l'huile d'olive possède le taux de marge positif le moins élevé avec 35,7 (product\_id = 5916).  
Parmi tous les produits le champagne a le taux de marge le plus faible avec -512,6 (product\_id = 4355).

product_type	product_id	price	taux_marge
705	Champagne	4355	12.65
			-512.618596

## Analyses complémentaires CA, quantités, stocks, taux de marge et correlations



- *Graphique avec commentaire des résultats*



Il semble y avoir une faible corrélation positive entre les ventes et la quantité en stock, c'est-à-dire que quand les ventes augmentent le stock augmente également.  
La corrélation est plus faible entre le prix et la quantité en stock.

## Analyses complémentaires CA, quantités, stocks, taux de marge et correlations

---

- *Méthodes statistiques employés*

Ce qu'on veut observer	Méthode	Résultat
Valorisation du stock	Quantité en stock * Prix	Le montant total des articles encore en stock est de 989 275,8 €.
Nombre d'article en stock	.sum()	Il y a 33480 articles en stock.

# Actions pour la suite

Action	Résultat	Ce qu'on peut observer
Table.info()	<pre>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</pre>	<ul style="list-style-type: none"><li>- Type de colonne</li><li>- Colonne avec des valeurs manquantes</li><li>- Nb de colonne et de ligne.</li></ul>
Table.describe()	<pre>product_id  onsale_web  price  stock_quantity  purchase_price count      825.00      825.00  825.00      825.00      825.00 mean       5162.60      0.87   32.19      21.59      16.94 std        902.64      0.34   26.71      21.93      14.56 min        3847.00      0.00  -20.00     -10.00      2.74 25%        4348.00      1.00   14.50       7.00       7.59 50%        4907.00      1.00   24.30      18.00     12.71 75%        5805.00      1.00   42.00      30.00     22.02 max        7338.00      1.00  225.00     145.00     137.81</pre>	<ul style="list-style-type: none"><li>- Statistique descriptive sur toutes les colonnes numériques du DataFrame</li></ul>
Table['colonne_str'].value_counts()	<pre>stock_status instock      732 outofstock   88 Name: count, dtype: int64</pre>	<ul style="list-style-type: none"><li>- Faute de frappe par exemple.</li><li>- Valeur aberrantes (des « int » alors que la colonne attend des « str »)</li></ul>



# Actions pour la suite

*Automatisation de la tâche de vérification de la qualité des données.*

```
def analyse_table(table):  
    detection = {}  
  
    print("\n Informations générales sur la table :\n")  
    print(table.info())  
  
    print("\n Statistiques descriptives pour les colonnes numériques :\n")  
    print(table.describe())  
  
    # Valeurs manquantes  
    valeurs_manquantes = table.isnull().sum()  
    detection['Valeurs manquantes'] = valeurs_manquantes[valeurs_manquantes > 0]  
    print("\n Valeurs manquantes :")  
    print(detection['Valeurs manquantes'])  
  
    # Doublons  
    doublons = table.duplicated().sum()  
    detection['Nb de doublons'] = doublons  
    print(f"\n Nombre de doublons : {doublons}")  
  
    # Value_counts() pour colonnes de type texte  
    colonne_texte = table.select_dtypes(include='object')  
    detection["Value_counts"] = {}  
    for col in colonne_texte.columns:  
        detection["Value_counts"][col] = table[col].value_counts()  
        print(f"\n Value_counts pour '{col}' :\n", table[col].value_counts())  
  
    #détection des outliers :  
    colonnes_numeriques = table.select_dtypes(include=['number'])  
    for col in colonnes_numeriques.columns:  
        mediane_q2 = colonnes_numeriques[col].median()  
        print(f"\n Analyse de la colonne '{col}':")  
        print(f"\n Médiane (Q2) : {mediane_q2}")  
  
        ligne_q1 = colonnes_numeriques[col][colonnes_numeriques[col] < mediane_q2]  
        mediane_q1 = ligne_q1.median()  
        print(f"\n Médiane de Q1 : {mediane_q1}")  
  
        ligne_q3 = colonnes_numeriques[col][colonnes_numeriques[col] > mediane_q2]  
        mediane_q3 = ligne_q3.median()  
        print(f"\n Médiane de Q3 : {mediane_q3}")
```

```
    iqr_col = mediane_q3 - mediane_q1  
    print(f"\n IQR : {round(iqr_col, 2)}")  
  
    # Sélection des lignes où la valeur est supérieure à l'IQR  
    valeurs_sup_iqr = table[table[col] > iqr_col]  
  
    print(f"\n Nombre de valeurs supérieures à l'IQR de la colonne {col}: {valeurs_sup_iqr.shape[0]}")  
    # Sauvegarde des lignes dans le dictionnaire  
    detection[col] = valeurs_sup_iqr  
  
    return detection
```

# Point sur les compétences apprises

---

- *Qu'est-ce qui s'est bien passé pour vous dans ce travail de nettoyage ?*

*L'importation des 3 fichiers s'est bien passé.*

*La suppression des valeurs abberantes et des valeurs nulles s'est bien passé avec pandas.*

- *Qu'est-ce que vous avez trouvé le plus difficile ?*

*Le plus difficile a été de comprendre les calculs spécifiques de valorisation de stock par exemple.*

*Comprendre certaines incohérences dans les données et déterminer les valeurs aberrantes à conserver ou à supprimer*

- *Sur quelles tâches est-ce que vous pensez avoir besoin de plus d'entraînement ?*

*Sur les tâches de nettoyage de donnée, représentation graphique en python me semble être l'axe d'amélioration le plus important.*