Optimisez la gestion des données d’une boutique avec Python

Bardey Rémi

Data Analyst

XXXXXXXXXXXXX

Analyses Exploratoires des Données

Le datasets, contexte de mission

ERP
➢ Références produits, prix HT, prix d'achat, stock
➢ Source : logiciel de gestion interne

Web
➢ Nom des produits, quantités vendues, prix TTC
➢ Source : site de vente en ligne

Liaison
➢ Table de correspondance entre identifiants ERP et Web
➢ Source : fichier mis à jour manuellement

Période concernée :

Ventes : du 1er au 31 octobre
État du stock : au 31 octobre

Objectif de la mission :

- Optimiser l'analyse des ventes, des marges et des stocks pour la boutique BottleNeck
- Fusionner les différentes sources de données internes, encore peu exploitées
- Identifier les erreurs, les écarts et proposer des axes d'amélioration pour la gestion

Analyses :

- Calcul du chiffre d'affaires (total et par produit)
- Identification des top ventes (20/80)
- Détection des erreurs de saisie (prix aberrants, marges négatives)
- Analyse du stock (quantités restantes, valeur, rotation)
- Étude des marges (unitaires et par type de produit)
- Recherche de corrélations (prix, ventes, marges, stock)

Analyses Exploratoires des Données

Première vue des données du fichier ERP

CONTENU

824 produits – 6 colonnes :

- product_id > identifiant produit
- onsale_web > vente en ligne ou non
- price, > prix de vente
- purchase_price > prix d'achat
- stock_quantity > Quantité en stock
- stock_status > Status du stock

Observations :

product_id = int → converti en string
onsale_web = int → converti en booléen
stock_status = object → converti en string
Pas de valeur manquante

```
RangeIndex: 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   purchase_price  825 non-null    float64
```

product_id	onsale_web	price	stock_quantity	stock_status	purchase_price
3847	1	24.2	16	instock	12.88
3849	1	34.3	10	instock	17.54
3850	1	20.8	0	outofstock	10.64
4032	1	14.1	26	instock	6.92
4039	1	46.0	3	outofstock	23.77

Analyses Exploratoires des Données

Nettoyage des stocks - ficher ERP

	product_id	onsale_web	price	stock_quantity	stock_status	purchase_price	stock_status_2
4	4039	1	46.0	3	outofstock	23.77	instock
398	4885	1	18.7	0	instock	9.66	outofstock
449	4973	0	10.0	-10	outofstock	4.96	instock
573	5700	1	44.5	-1	outofstock	22.30	instock

Valeur de stock négatif = impossible. Considérer qu'il s'agit d'une erreur d'export ou de saisie > modifier en valeur positive

Données incohérentes puisque nous avons des produits en stock, référencé comme en rupture et vice versa.

Analyses Exploratoires des Données

Nettoyage des prix - ficher ERP

```
Prix manquant (NaN) : 0
Prix négatif : 3
Prix nul : 0
```

L'analyse des prix ne montre aucune valeur manquante ou nulle. En revanche, il y a des valeurs négatives, incohérente dans ce cas.

	product_id	onsale_web	price
151	4233	0	-20.0
469	5017	0	-8.0
739	6594	0	-9.1

Comme pour les autres valeurs négatives, je pars du principe qu'il s'agit d'une erreur de saisie ou durant l'export, mais que les valeurs sont justes.

Je les convertis en nombre positif grâce à ce code :

```
df_erp['price'] = df_erp['price'].abs()
```

Analyses Exploratoires des Données

Première vue des données du fichier WEB

```
RangeIndex: 1513 entries, 0 to 1512
Data columns (total 29 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  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
```

CONTENU

1 513 lignes, 29 colonnes

Observations :

Colonnes vides ou inutiles (tax_class, post_content…)

Données partiellement remplies, voire absentes

Erreur de syntaxe (valeurs, nom colonne)

RGPD : Suppression des colonnes post_author et post_password.

Analyses Exploratoires des Données

Nettoyage SKU, valeurs null - fichier WEB

	sku	virtual	downloadable	rating_count	average_rating	total_sales	tax_status	tax_class
0	11862	0	0	0	0.0	3.0	NaN	NaN
1	16057	0	0	0	0.0	5.0	NaN	NaN

Beaucoup de colonnes et lignes soit remplie de 0 ou de NaN. Le manque d'information peut fausser l'analyse et surcharger les ressources informatiques. J'ai choisi de supprimer ces colonnes.

85 lignes sans code article (sku) ou vides trouvées

	sku	total_sales	post_date	product_type
8	<NA>	NaN	NaT	NaN
20	<NA>	NaN	NaT	NaN
30	<NA>	NaN	NaT	NaN

85 SKU sans valeur, ne permettant pas d'identifier clairement les produits. Meme raisonnement que précédemment, ces lignes sont supprimées

```
SKUs non conformes : <StringArray>
[<NA>, '13127-1', 'bon-cadeau-25-euros']
Length: 3, dtype: string
```

3 SKU ont des erreurs de syntaxes.

Analyses Exploratoires des Données

Résultat final - fichier WEB

```
RangeIndex: 714 entries, 0 to 713
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   id_web        714 non-null    string
 1   total_sales   714 non-null    float64
 2   post_date     714 non-null    object
 3   product_type  714 non-null    string
 4   post_title    714 non-null    string
dtypes: float64(1), object(1), string(3)
```

	id_web	total_sales	post_date	product_type	post_title
0	14692	5.0	2019-03-19	Vin	Château Fonréaud Bordeaux Blanc Le Cygne 2016
1	15328	2.0	2019-03-27	Vin	Agnès Levet Côte Rôtie Maestria 2017
2	16515	10.0	2018-06-02	Vin	Château Turcaud Bordeaux Rouge Cuvée Majeure 2018
3	16585	15.0	2018-02-16	Vin	Xavier Frissant Touraine Sauvignon 2019
4	12869	7.0	2019-03-28	Vin	Stéphane Tissot Arbois D.D. 2016

Résumé du nettoyage :

Syntaxe, valeurs négative ou incohérente modifiées

Valeurs manquantes, supprimées ou complétées

Données incohérentes corrigées

Suppression des colonnes et lignes vides

Gestion des doublons

Modification des types de données

Fusion ou consolidations des données

Data Frame final

```
df_final = pd.merge((df_erp), (df_liaison), on = ['product_id'], how = 'inner')
df_final.head()
```

```
RangeIndex: 734 entries, 0 to 733
Data columns (total 7 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   product_id      734 non-null    string
 1   onsale_web      734 non-null    bool
 2   price           734 non-null    float64
 3   stock_quantity  734 non-null    int64
 4   stock_status    734 non-null    string
 5   purchase_price  734 non-null    float64
 6   id_web          734 non-null    string
dtypes: bool(1), float64(2), int64(1), string(3)
```

- Fusion entre erp et liaison
- Jointure interne pour exclure les articles sans correspondances
- Clé : Product_id
- Aucune donnée manquante

Fusion ou consolidations des données

Data Frame final

```
df_final = pd.merge((df_final), df_web2, on='id_web', how='inner')
```

```
RangeIndex: 714 entries, 0 to 713
Data columns (total 11 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   product_id      714 non-null    string
 1   onsale_web      714 non-null    bool
 2   price           714 non-null    float64
 3   stock_quantity  714 non-null    int64
 4   stock_status    714 non-null    string
 5   purchase_price  714 non-null    float64
 6   id_web          714 non-null    string
 7   total_sales     714 non-null    float64
 8   post_date       714 non-null    object
 9   product_type    714 non-null    string
 10  post_title      714 non-null    string
dtypes: bool(1), float64(3), int64(1), object(1), string(5)
```

- Fusion entre df_final et web2
- Jointure interne pour exclure les articles sans correspondances
- Clé : id_web
- Aucune donnée manquante

Fusion ou consolidations des données

Data Frame final

	product_id	onsale_web	price	stock_quantity	stock_status	purchase_price	id_web	total_sales	post_date	product_type	post_title
0	3847	True	24.2	16	instock	12.88	15298	6.0	2018-02-08	Vin	Pierre Jean Villa Saint-Joseph Préface 2018
1	3849	True	34.3	10	instock	17.54	15296	9.0	2018-02-08	Vin	Pierre Jean Villa Saint-Joseph Rouge Tildé 2017
2	3850	True	20.8	0	outofstock	10.64	15300	0.0	2018-02-08	Vin	Pierre Jean Villa Crozes-Hermitage Accroche Co...
3	4032	True	14.1	26	instock	6.92	19814	12.0	2018-02-09	Vin	Pierre Jean Villa IGP Collines Rhodaniennes Ga...
4	4039	True	46.0	3	instock	23.77	19815	3.0	2018-02-12	Vin	Pierre Jean Villa Côte Rôtie Carmina 2017

Le résultat de la fusion des 3 fichiers, contient toutes les variables nécessaires à l'analyse.

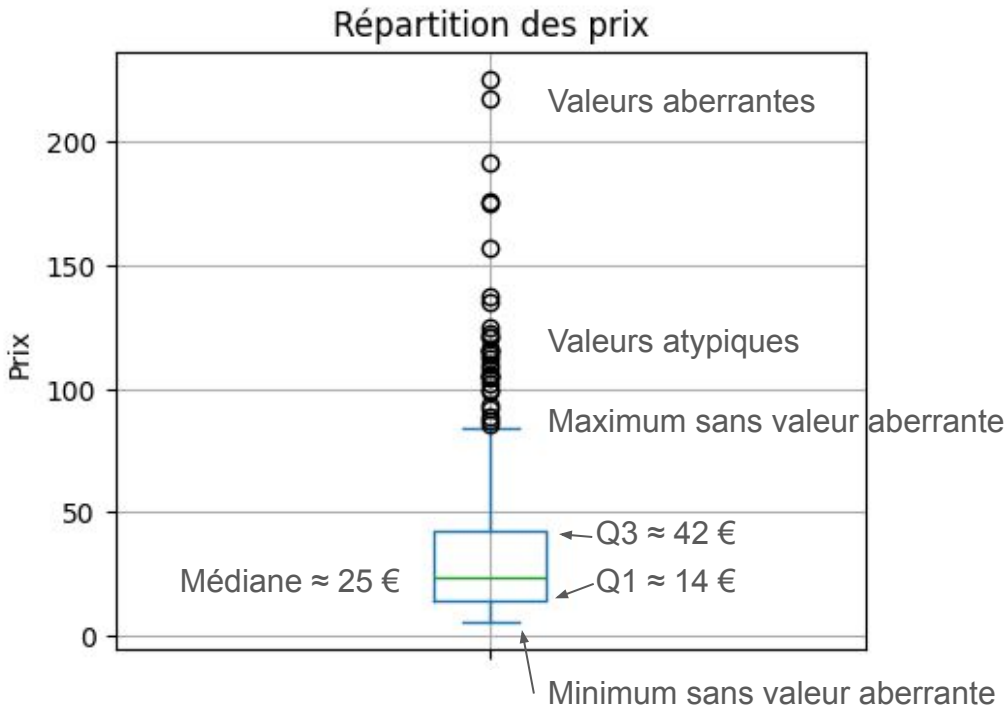
On y trouve les identifiants produits, le prix, le cout, les quantités de stock, le volume de vente, le nom des produits.

Être vigilent lors des manipulations des fichers pour ne pas compromettre les données.

Prendre le temps de vérifier chaque étape pour valider les décisions.

Analyses univariées du prix

Distribution des prix



On observe une majorité de prix situés entre 10€ et 45€, et une médiane à 25€, ainsi que de nombreuses valeurs atypiques au-delà de 60€, dont certaines dépassent 200€.

Ces valeurs extrêmes peuvent représenter :

Des produits haut de gamme, ou d'éventuelles erreurs de saisie, à analyser au cas par cas.

Méthode statistique dites de Tukey

```
Q1 = df_final['price'].quantile(0.25)
Q3 = df_final['price'].quantile(0.75)
IQR = Q3 - Q1

seuil_min = Q1 - 1.5 * IQR
seuil_max = Q3 + 1.5 * IQR

print(f"IQR : {IQR:.2f}")
print(f"Seuil minimum : {seuil_min:.2f} €")
print(f"Seuil maximum : {seuil_max:.2f} €")

print(f'Q1 : {Q1}')
print(f'Q3 : {Q3}')
```

Résultats :

```
IQR : 28.01
Seuil minimum : -27.96 €
Seuil maximum : 84.09 €
Q1 : 14.0625
Q3 : 42.075
```

Analyses univariées du prix

Identification des outliers

Outliers cohérent puisque prix d'achat proportionnellement élevé par rapport au prix de vente.

```
outliers_prix = df_final[df_final['price'] > seuil_max]
nb_outliers = outliers_prix.shape[0]
nb_total = df_final.shape[0]
pct_outliers = (nb_outliers / nb_total) * 100

print(f"Nombre d'articles outliers : {nb_outliers}")
print(f"Nombre total d'articles : {nb_total}")
print(f"Proportion d'outliers : {pct_outliers:.2f} %")
```

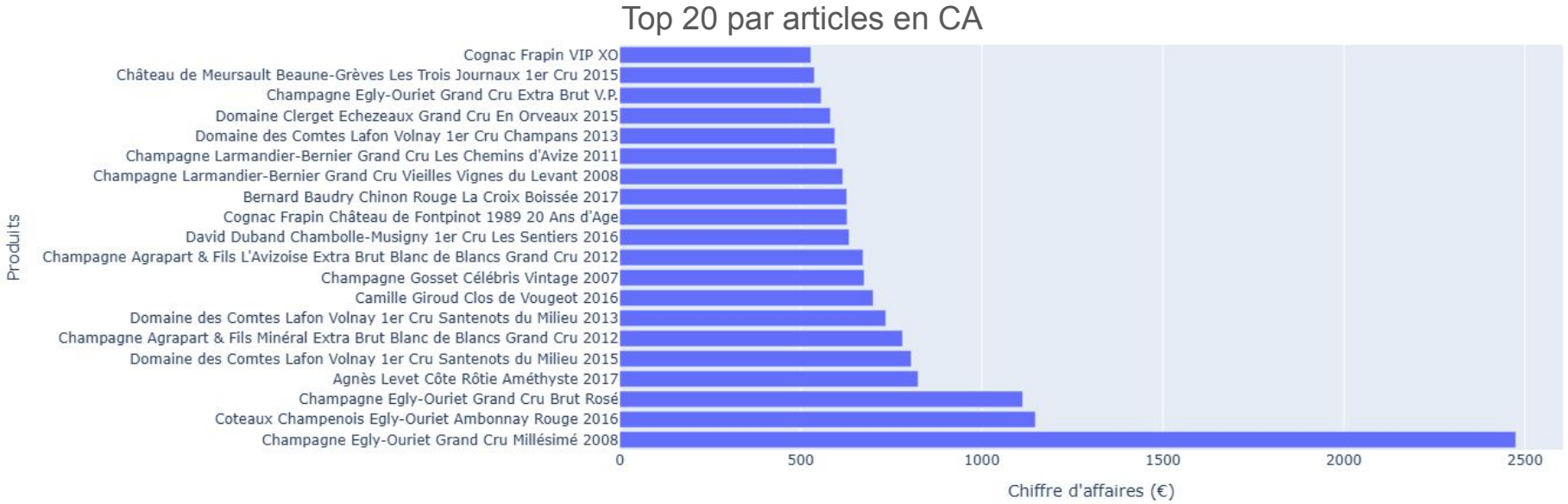
```
Nombre d'articles outliers : 31
Nombre total d'articles : 714
Proportion d'outliers : 4.34 %
```

	purchase_price	price	product_id
63	52.70	100.0	4115
65	44.30	88.4	4132
199	137.81	225.0	4352
205	51.93	85.6	4359
218	78.25	176.0	4402
219	52.22	108.5	4404
221	69.08	157.0	4406

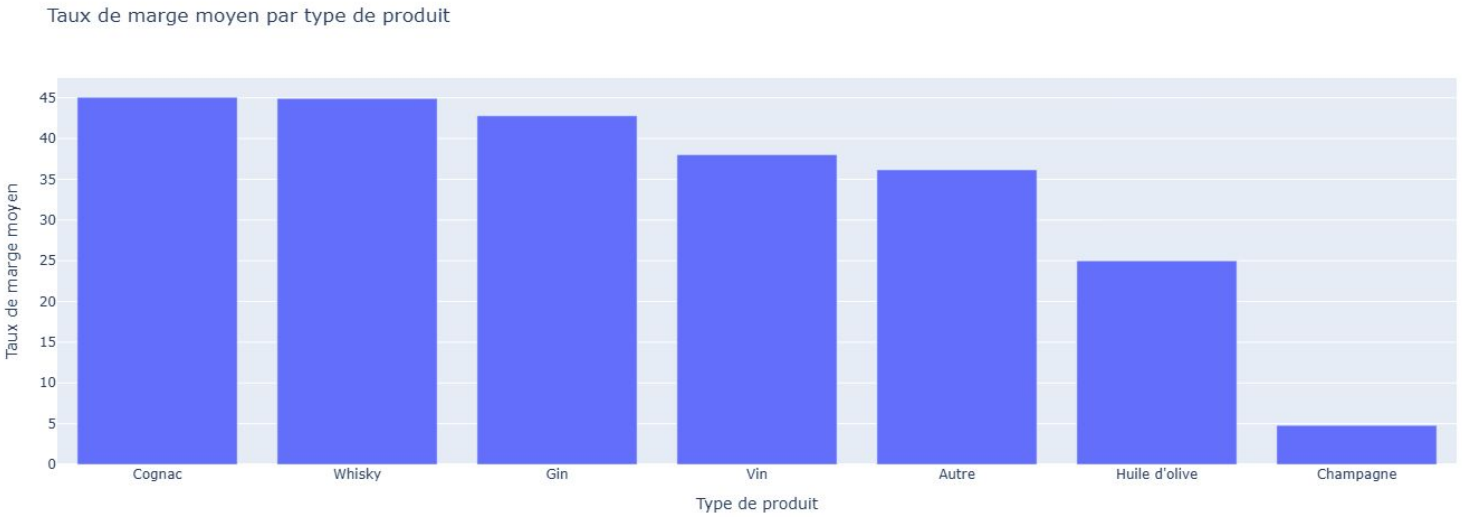
Analyses complémentaires
CA, quantités, stocks, taux de marge et corrélations

Un chiffre d'affaires global de 143 680 € sur la période

La loi de Pareto ne s'applique pas à notre cas. 80% du CA est généré par 60% du catalogue. *(434 produits sur 714)*

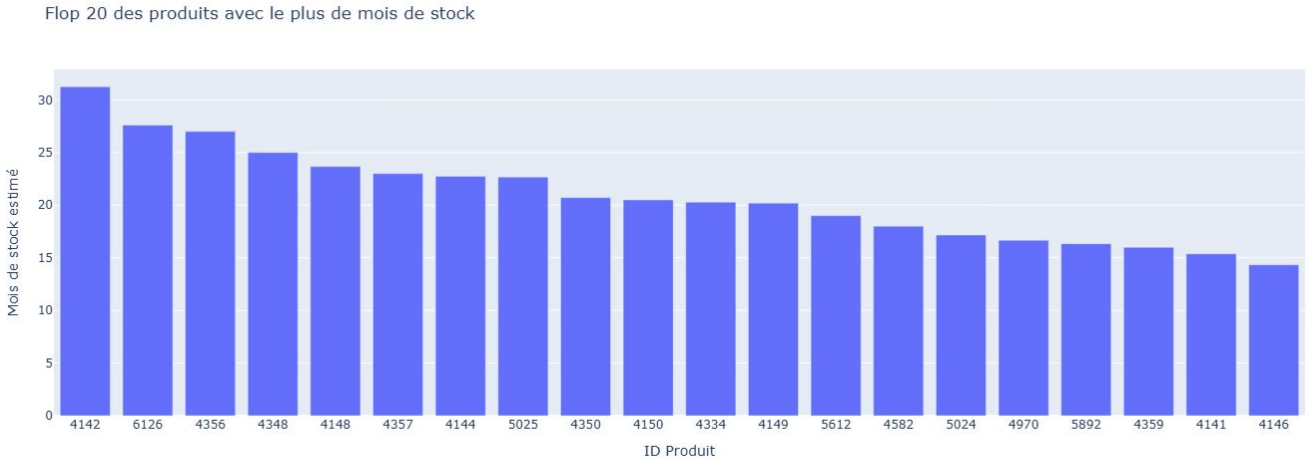


Analyses complémentaires
CA, quantités, stocks, taux de marge et corrélations



L’analyse des taux marges montre que bien que ce soit la catégorie vin qui soit la plus vendue, ce sont les Cognac, Whisky et le Gin qui sont les plus rentables.

Analyses complémentaires
CA, quantités, stocks, taux de marge et corrélations



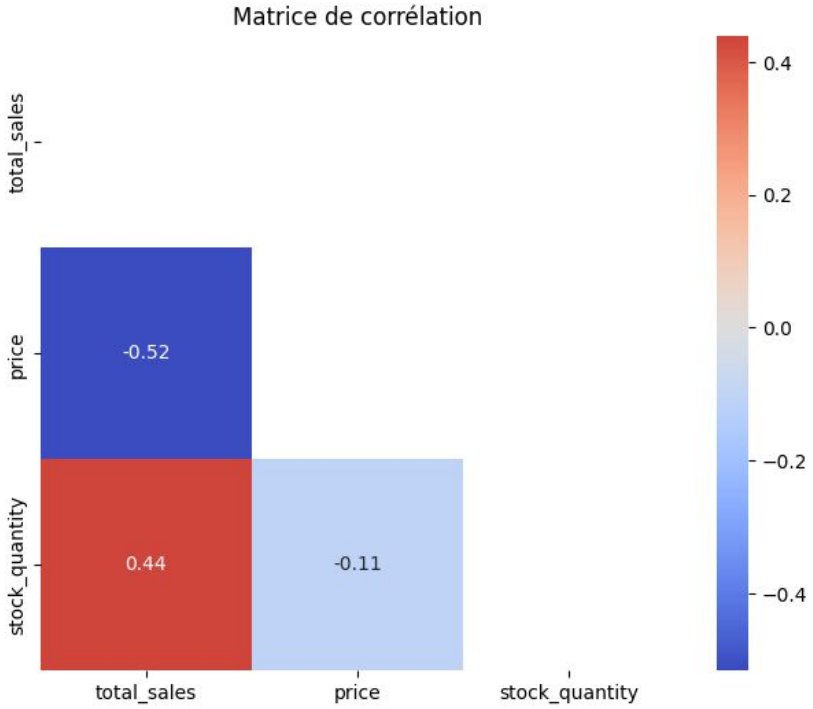
L'analyse des stocks mets en évidence nous montre que :

- 16 741 articles en stock
- Contre 5751 articles vendu sur la période
- Pour une valeur estimé à 277 350€ immobilisé

Cela indique qu'environ 64 % du stock est encore immobilisé, ce qui peut poser des questions sur :

la rotation des stocks,
ou la pertinence des réapprovisionnements.

Analyses complémentaires
CA, quantités, stocks, taux de marge et corrélations



Résultats observés :

Prix ↔ Ventes : −0.52 → corrélation négative modérée
➤ Plus le prix est élevé, moins le produit se vend

Stock ↔ Ventes : +0.44 → corrélation positive modérée
➤ Les produits bien approvisionnés se vendent mieux

Prix ↔ Stock : −0.11 → corrélation très faible
➤ Pas de lien significatif entre le prix et la quantité en stock

À noter :

Les corrélations ne prouvent pas la causalité, mais elles suggèrent des tendances générales utiles pour orienter les décisions.

Un coefficient est considéré :
fort > 0.7
modéré ≈ 0.4 à 0.7
faible < 0.3

Actions pour la suite

Amélioration des données

- Mettre en place des règles de validation automatique (prix, marges, stock)
- Uniformiser les formats et identifiants entre ERP et site Web
- Automatiser la mise à jour de la table de liaison

Suivi & pilotage

- Créer un tableau de bord dynamique
- Suivre les KPI clés : CA, marges, stocks, produits à faible rotation
- Intégrer un système d'alerte pour les valeurs aberrantes

Limites de l'analyse

Méthodes statistiques à relativiser
Les outils comme le boxplot, les outliers statistiques (IQR, Z-score) ou les corrélations ne suffisent pas à évaluer la pertinence métier des prix ou des ventes.

Certaines valeurs extrêmes détectées comme aberrantes sont en réalité justifiées commercialement (ex : bouteilles rares ou premium)

Spécificités du secteur

Le monde du vin implique une large gamme de prix : de quelques euros à plusieurs centaines
Les "anomalies" statistiques doivent toujours être interprétées à la lumière du métier.
Ce travail est un point de départ, mais nécessite un regard expert du terrain pour valider certaines interprétations.

Point sur les compétences apprises

Ce qui s’est bien passé

L’analyse exploratoire, le nettoyage et les jointures se sont déroulés de manière fluide.
J’ai trouvé cela très motivant de plonger dans un projet concret, avec des données réelles et un objectif métier clair.

Ce qui a été le plus difficile

L’analyse statistique a été un vrai défi : je ne connaissais pas toutes les méthodes, il a donc fallu les comprendre et les appliquer rapidement.
J’ai aussi dû veiller à la cohérence globale du projet, en suivant l’évolution des fichiers et des traitements.
Certaines analyses ont été complexes à interpréter : j’ai déjà travaillé sur les marges ou les coûts, mais avec d’autres approches métier, ce qui a rendu la lecture de ces résultats plus délicate.

Points à renforcer

- Consolider mes bases en statistiques et en mathématiques
- Mieux rédiger du code Python sans erreurs, sans dépendre d’un correcteur
- Continuer à pratiquer pour gagner en aisance dans l’analyse de données complexes