

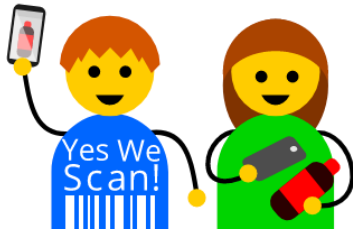


Projet 3 : Concevez une application au service de la santé publique

L'agence "Santé publique France" a lancé un appel à projets pour trouver des idées innovantes d'applications en lien avec l'alimentation. Vous souhaitez y participer et proposer une idée d'application.

Camille BRODIN

Principe général d'Open Food Facts :



Base de données publique

- créée en 2012
- collaboratif
- + de 990 000 produits



Fonctionnalités web et app :

- Décrypter les étiquettes
- Trouver et comparer des produits selon critères
- Explorer, découvrir, contribuer

- **Informers pour aider le consommateur à choisir une meilleure alimentation**
- Eduquer, développer de nouveaux produits et services, et aider la recherche scientifique



Idée d'application

Nettoyage des données

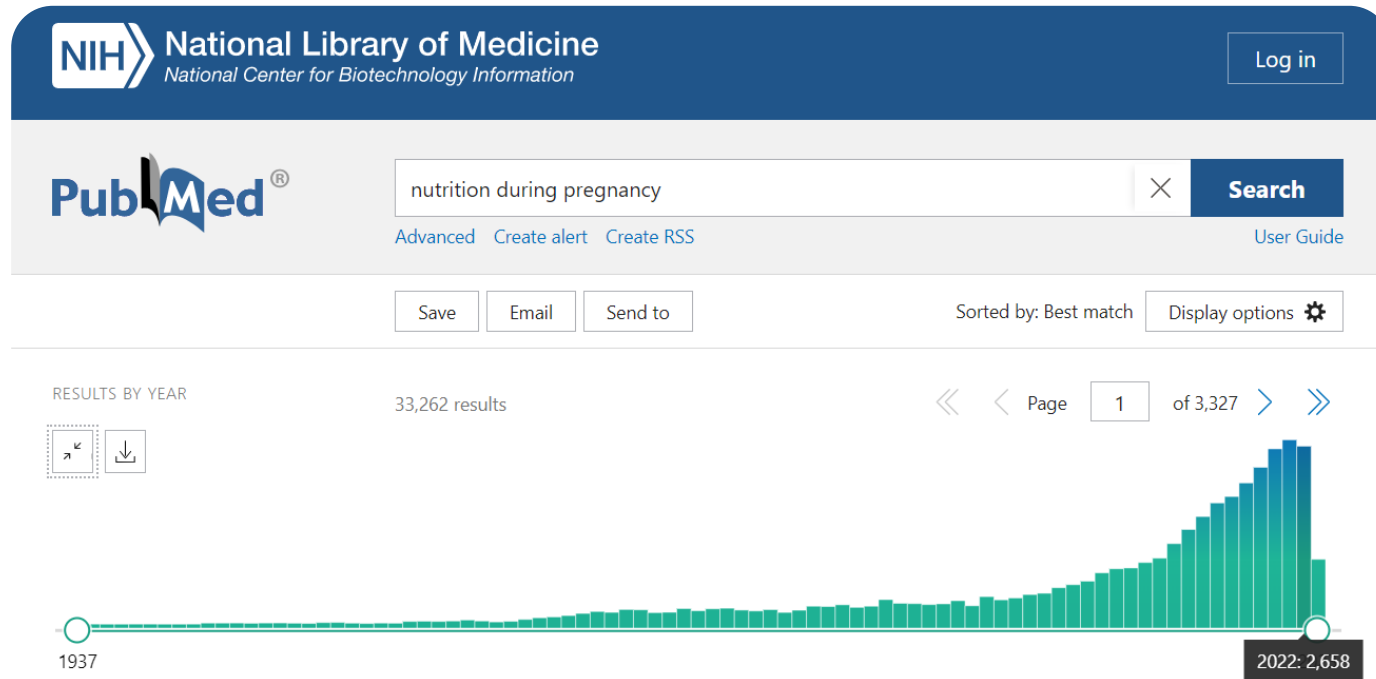
Analyse des données

Conclusions

Une application pour la santé :

Aider les consommatrices
enceintes à identifier les
produits à prioriser et à éviter

1. Présentation de l'idée d'application



risque de troubles de la grossesse
de diabète sucré gestationnel
de naissance prématurée
de complications liées à l'obésité
de prééclampsie
d'hypertension gestationnelle*

*Marshall NE, et al., The importance of nutrition in pregnancy and lactation: lifelong consequences. Am J Obstet Gynecol. 2022 May;226(5):607-632



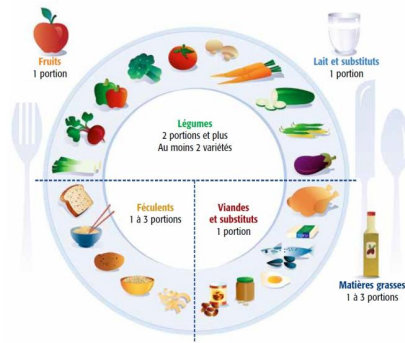
- Fort effort de recherche et meilleure compréhension de la nutrition en période de grossesse
- Toujours pas d'indications sur les aliments à prioriser ou éviter (en dehors des boissons alcoolisées)

1. Présentation de l'idée d'application

Une application pour la santé :

Aider les consommatrices enceintes à identifier les produits à prioriser et à éviter

1. Notification de produits quotidiens



Nos idées menus pour vous aujourd'hui :

- Fruits et légumes :
- Féculents :
- Viandes et volailles :
- produits laitiers :
- produits sucrés :
- Epicerie :

Fonction 1 : Recommander et inspirer

- Notification de liste produits
- Renouvellement chaque jour

2. Classer les produits en 3 niveaux lors des achats



Fonction 2 : Classer et guider

- Diviser les produits en trois groupes
- Améliorer la consommation
- prévenir les risques



Idée d'application

Nettoyage des données

Analyse des données

Conclusions

Une base de données propre

- ☐ sans valeurs aberrantes
- ☐ sans doublons
- ☐ sans valeurs manquantes

Un contenu adapté

- ☐ des produits identifiables
- ☐ des catégories pertinentes
- ☐ des données chiffrées utiles

2. Démarche méthodologique de nettoyage

Le jeu de données Open Food Facts:

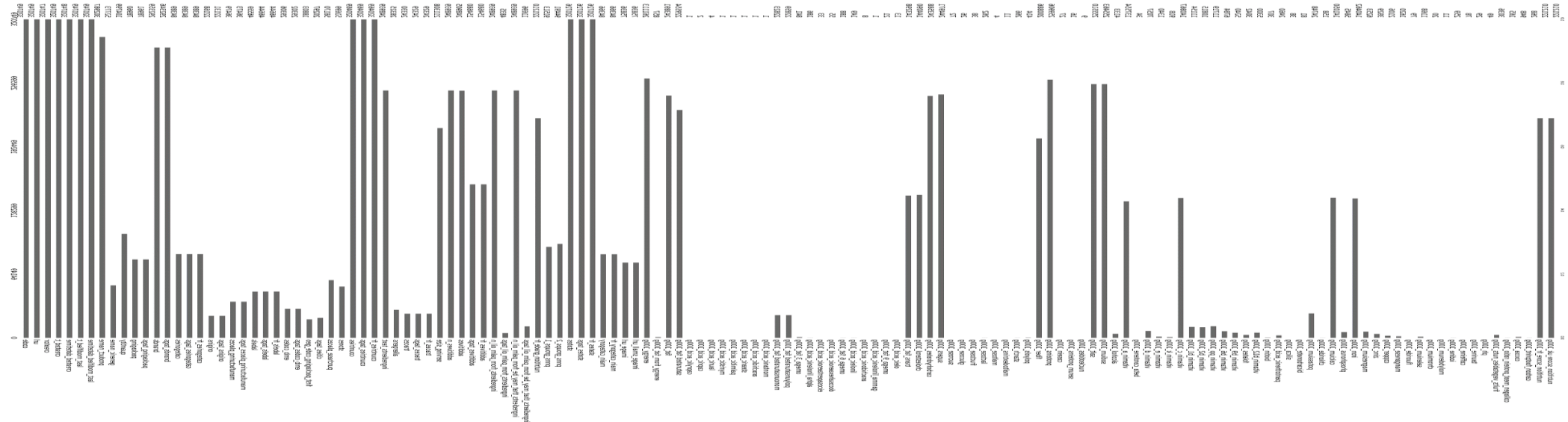


Illustration du % de remplissage/var

Grand jeu de données

- plus de 320 000 lignes
- 162 colonnes
- Autour de 75 % de valeurs manquantes
- Des colonnes presque vides

Plusieurs types de variables

- Infos produit et id.
- Catégories
- Teneurs pour 100g
- Nombre entiers (additifs, ...)
- Ratios et scores

2. Démarche méthodologique de nettoyage

Etapas suivies pour le nettoyage de données :

Etapas nettoyages	Lignes/colonnes supprimées	Méthodes
1.a/Filtrage global	-31 colonnes vides -14 colonnes redondantes	list+df.drop describe+df.drop
1.b/Filtrage orienté projet	-89 colonnes inutiles -261 819 lignes inutiles	df=df[[]] df.loc + df.dropna
2. a/Traitement des valeurs aberrantes	-43 612 lignes aberrantes et -2 colonnes inutiles	df.strftime + df.loc
2. b/Traitement des valeurs outliers	-5276 lignes outliers	df.loc <= 100g
3. Traitement des doublons	/	df['code'].duplicated()



- 320 772 lignes sur le jeu de données brutes, après nettoyage : 10 065
- 162 colonnes sur le jeu de données brutes, après nettoyage : 26

2. Démarche méthodologique de nettoyage

Etapas suivies pour le nettoyage de données :

4. Imputations des valeurs manquantes:

Notre liste de 124 catégories « main_category_fr » avec minimum n=5 produits :

['petit-dejeuners', 'biscuits', 'boissons', 'chocolats', 'epicerie', 'pains', 'snacks sucrés', 'conserves', 'aliments et boissons a base de vegetaux', 'desserts', 'plats prepares', 'gateaux', 'surgeles', 'chips et frites', 'produits a tartiner sales', 'pates alimentaires', 'soupes', 'plats a base de viande', 'fromages', 'confitures', 'snacks sales', 'bonbons', 'laits', 'yaourts', 'jus de fruits', 'jus d'orange 100% pur jus', 'fruits a coques', 'jambons', 'volailles', 'beurres', 'condiments', 'fromages de france', 'sandwichs', 'pates a tartiner', 'sodas', 'jus de fruits 100% pur jus', 'glaces', 'cremes', 'nectars de fruits', 'yaourts aux fruits', 'fromages blancs', 'aliments pour bebe', 'cookies', 'gaufres', 'desserts au chocolat', 'riz', 'fruits secs', 'emmental', 'fromages de vache', 'produits panes', 'saucissons', 'sables', 'yaourts entiers', 'sardines en conserve', 'salades composees', 'poissons', 'huiles', 'produits d'elevages', 'sucres', 'jus de pomme', 'saumons', 'viandes', 'produits labellises', 'sodas light', 'saucisses', 'thons', 'legumes frais', 'tartes', 'huiles d'olive', 'produits de la mer', 'produits a teneur reduite en sel', 'produits laitiers', 'eaux', 'edulcorants', 'produits a tartiner', 'pickles', 'crepes et galettes', 'lardons', 'yaourts brasses', 'yaourts a boire', 'pains de mie complet', 'jus de fruits a base de concentre', 'poissons en conserve', 'pates a tarte', 'petits beurres', 'jus de pamplemousse', 'patisseries', 'jambons de paris', 'farines', 'sodas aux fruits', 'charcuteries', 'fruits', 'jus d'orange a base de concentre', 'farines de ble', 'infusions', 'jambons secs', 'produits deshydrates', 'vinaigres', 'jambons crus', 'poissons fumes', 'olives vertes', 'produits a tartiner sucrés', 'epices', 'jus d'orange', 'taboules', 'thes', 'sels', 'crustaces', 'coquillettes', 'terrines', 'purees', 'pates', 'glaces et sorbets', 'boissons gazeuses', 'legumes tiges', 'quenelles', 'saucissons secs pur porc', 'chocolats-de-degustation', 'pates a pizza', 'compotes pour bebe', 'matieres grasses vegetales', 'poissons et viandes et oeufs', 'thes verts', 'charcuteries diverses']

-> **Création d'une nouvelle variable « general_categ » représentant les 7 familles principales d'aliments :**

- ❖ Les produits sucrés
- ❖ Les corps gras (+épicerie)
- ❖ Les viandes, poissons ou œufs
- ❖ Les produits laitiers
- ❖ Les fruits et légumes
- ❖ Les féculents
- ❖ Les boissons



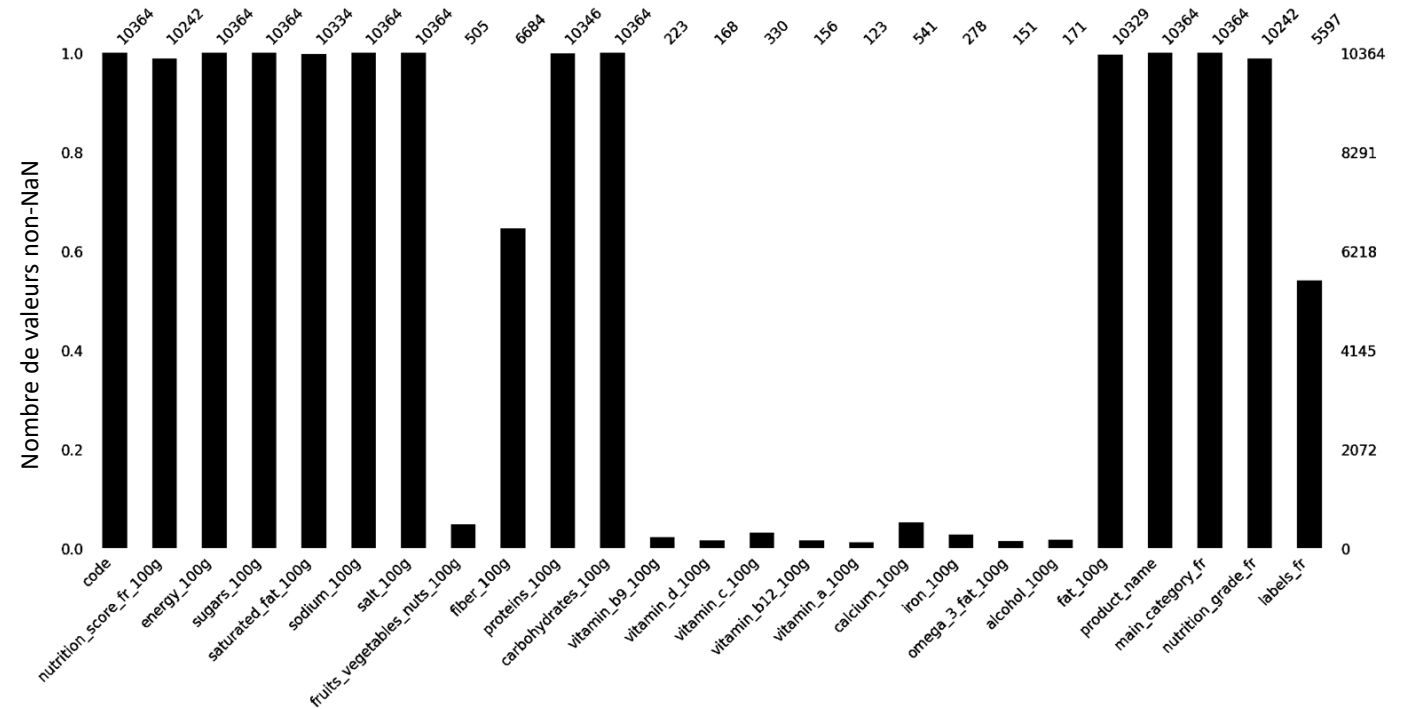
<https://alimentale.fr/sante/les-bases-de-la-nutrition/la-pyramide-alimentaire-tout-lequilibre-en-un-coup-doeil/>

2. Démarche méthodologique de nettoyage

4. Imputations des valeurs manquantes:

Très fort taux de NaN, trois méthodes d'imputation utilisées

- **Méthode 1 : Imputation NaN par moyenne par catégorie**
- **Méthode 2 : Imputation NaN par 0**
- **Méthode 3 : Recalcule des grades (a, b, c, d) à partir des notes**

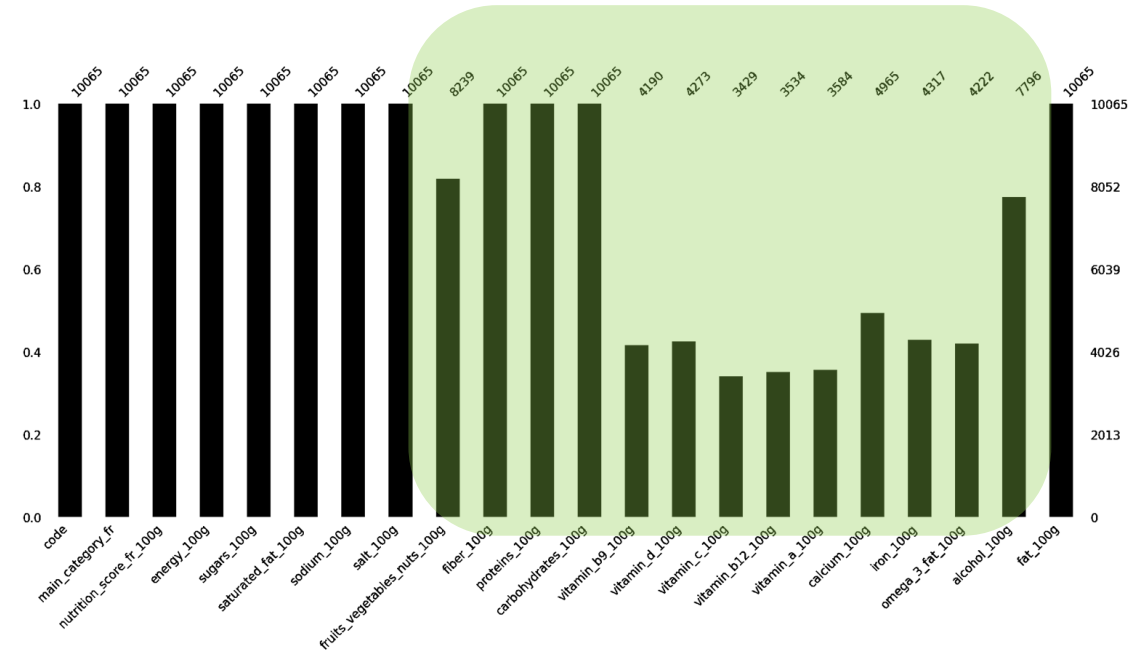


2. Démarche méthodologique de nettoyage

4/Imputations des valeurs manquantes:

- **Méthode 1 : Imputation des NaN par la moyenne**
- **124 groupes de produits**
- **Appliqué sur les données nutritives (_100g)**

```
group = df_100g.groupby('main_category_fr')  
✓ 0.0s  
  
def fill_mean(group):  
    mean = group.mean()  
    group = group.fillna(value=mean)  
    return group  
✓ 0.0s  
  
df_100g = df_100g.groupby('main_category_fr').apply(fill_mean)  
✓ 0.5s
```



➤ **42,34 % de valeurs manquantes avant, 23,53 % après imputation**

2. Démarche méthodologique de nettoyage

4/Imputations des valeurs manquantes:

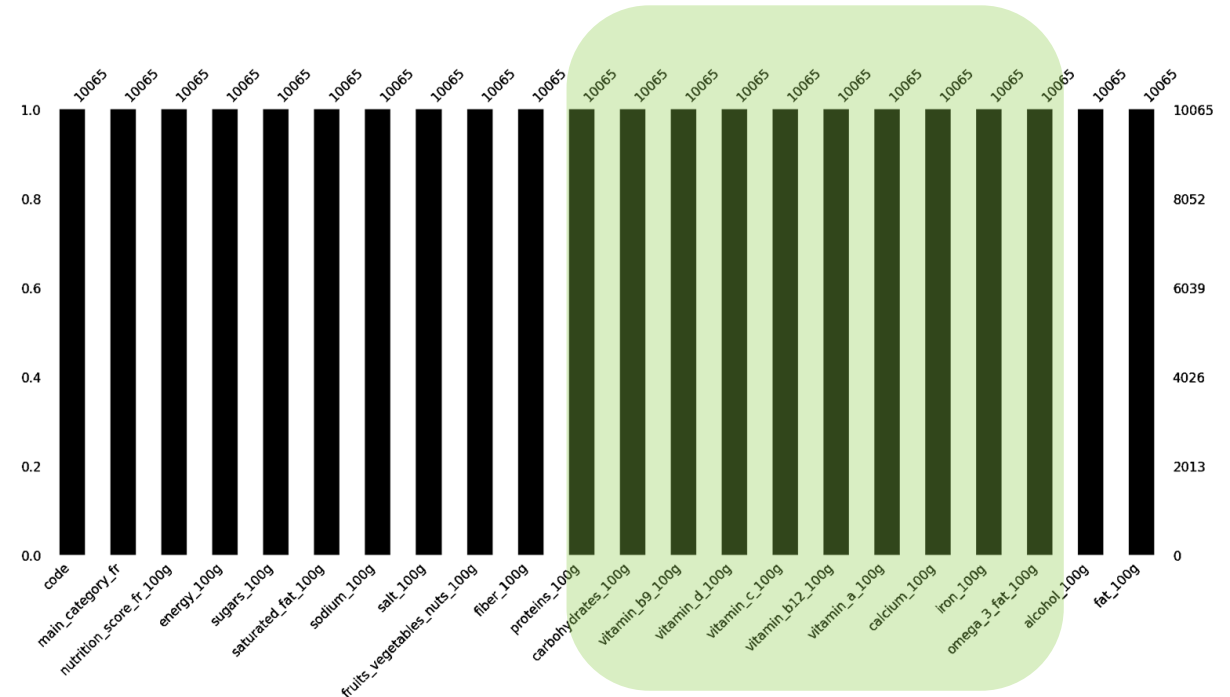
→ **Méthode 2 : Imputation des NaN restants par 0**

→ Appliqué sur les données nutritives (_100g)

```
vitamin_b9_100g    5801
vitamin_d_100g     5638
vitamin_c_100g     6482
vitamin_b12_100g   6457
vitamin_a_100g     6327
calcium_100g       4946
iron_100g          5674
omega_3_fat_100g   5689
```

```
df_100g = df_100g.fillna(0)
```

```
vitamin_b9_100g    0
vitamin_d_100g     0
vitamin_c_100g     0
vitamin_b12_100g   0
vitamin_a_100g     0
calcium_100g       0
iron_100g          0
omega_3_fat_100g   0
```



➤ 23,53 % de valeurs manquantes à 1,79% après cette imputation (avec les catégories, cf diapo suivante)




2. Démarche méthodologique de nettoyage

4/Imputations des valeurs manquantes:

- Méthode 3 : Imputation des NaN restants par recalcule des grades (a, b, c, d) à partir des notes.
- Appliqué sur les données qualitatives

• Attribution des couleurs

Le logo Nutri-Score est ensuite attribué en fonction du score obtenu (cf. tableau ci-dessous).

Points		Logo
Aliments solides	Boissons	
Min à -1	Eaux	
0 à 2	Min à 1	
3 à 10		
11 à 18		
19 à Max		

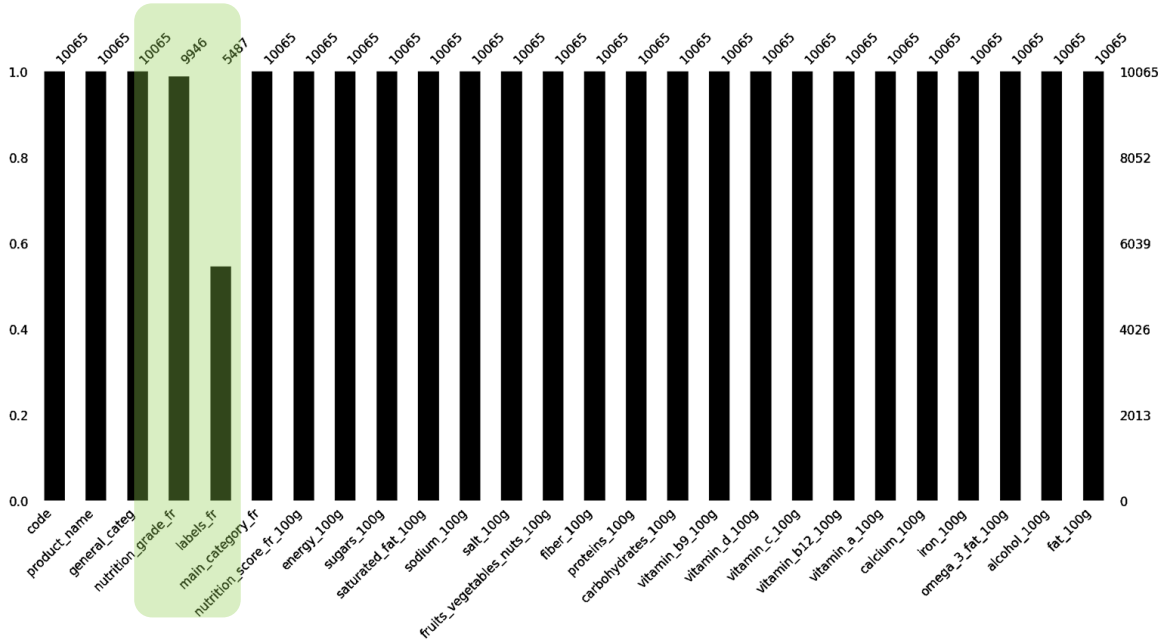
```
df_bio = pd.DataFrame(df_cleaned.loc[df_cleaned["labels_fr"].str.contains("Ab")])
df_bio['AB'] = "oui"
df_bio = df_bio[['code', 'AB']]
df_bio

df_cleaned = pd.merge(df_cleaned, df_bio, on='code', how='left')
df_cleaned['AB'] = df_cleaned['AB'].fillna("non")
df_cleaned['AB'].value_counts()

✓ 0.2s

non      8668
oui       1397
Name: AB, dtype: int64
```

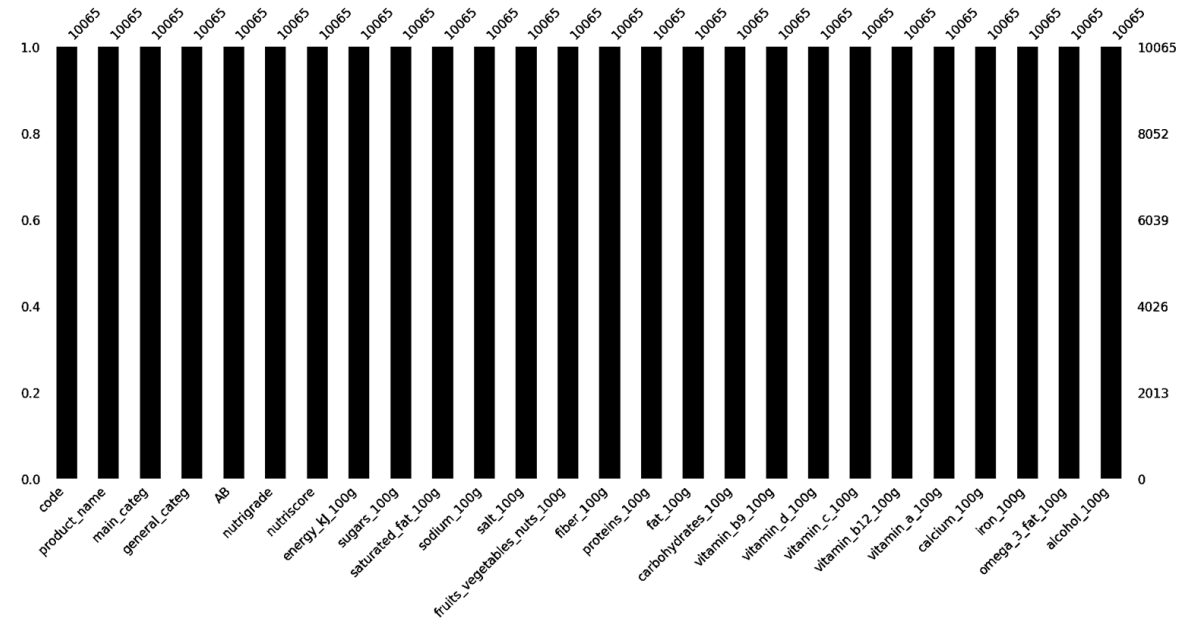
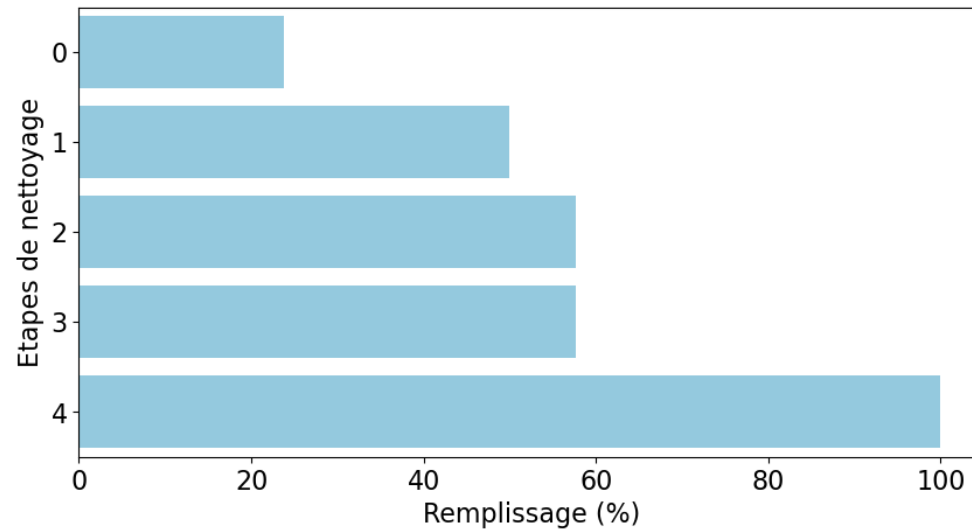
AB -> information manquante -> False



➤ 1,79 % de valeurs manquantes sur le jeu de données brutes, 0% après cette imputation

2. Démarche méthodologique de nettoyage

4/Imputations des valeurs manquantes:



➤ Jeu de données exploitable, 10 065 produits et 26 variables au total



Idée d'application

Nettoyage des données

Analyse des données

Conclusions

Analyses univariées

- ☐ Variables qualitatives
- ☐ Variables quantitatives

Analyses bivariées

- ☐ Quant-quant : Corrélations
- ☐ Quali-quant : Test MW et KW

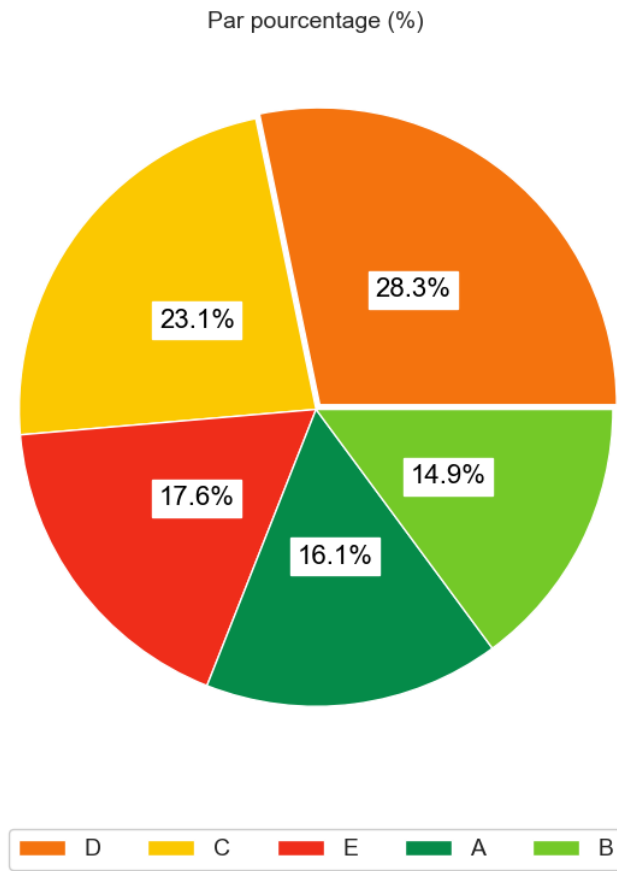
Analyses multivariées

- ☐ k-means et KW
- ☐ Analyse en Composante principale

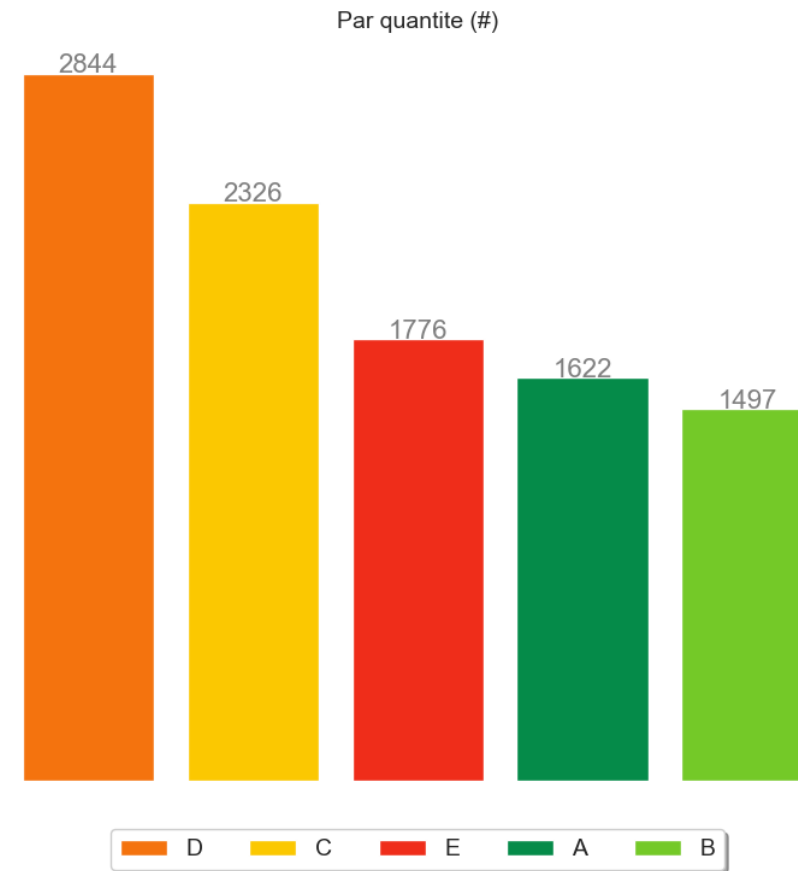
3. Démarche méthodologique d'exploration de données

1. Analyse univariée des différentes variables importantes avec les visualisations associées.

❖ Variables qualitatives : Nutrigrade



Distribution des labels AB dans le dataset

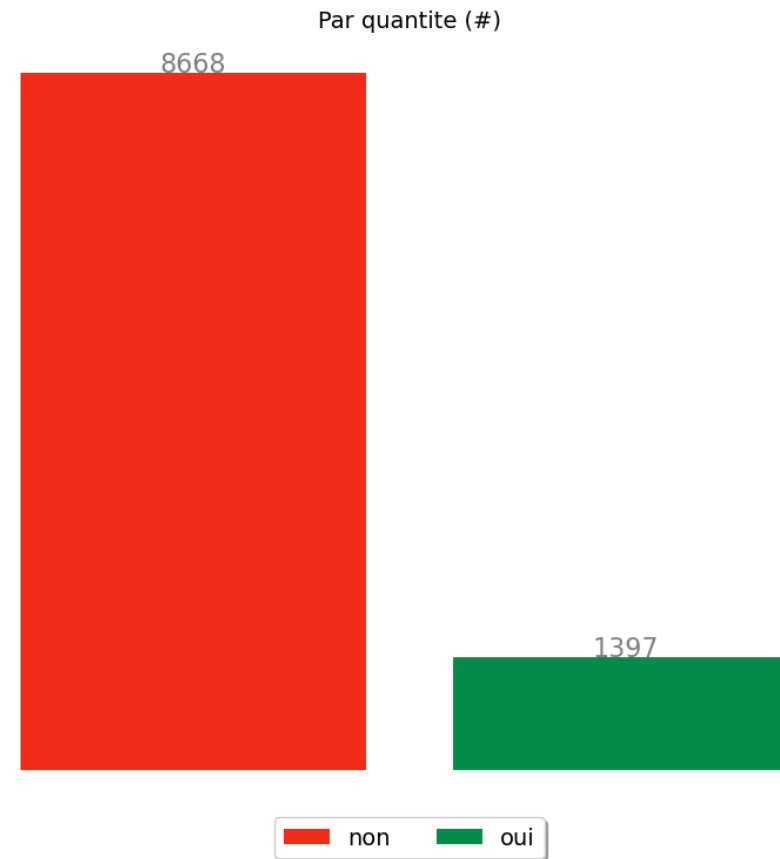
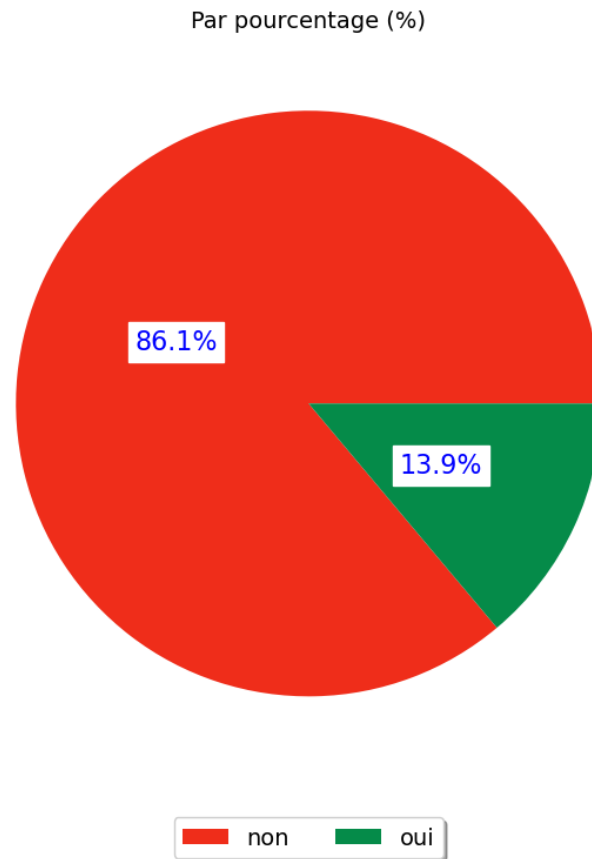


3. Démarche méthodologique d'exploration de données

1. Analyse univariée des différentes variables importantes avec les visualisations associées.

❖ Variables qualitatives : Label biologique AB

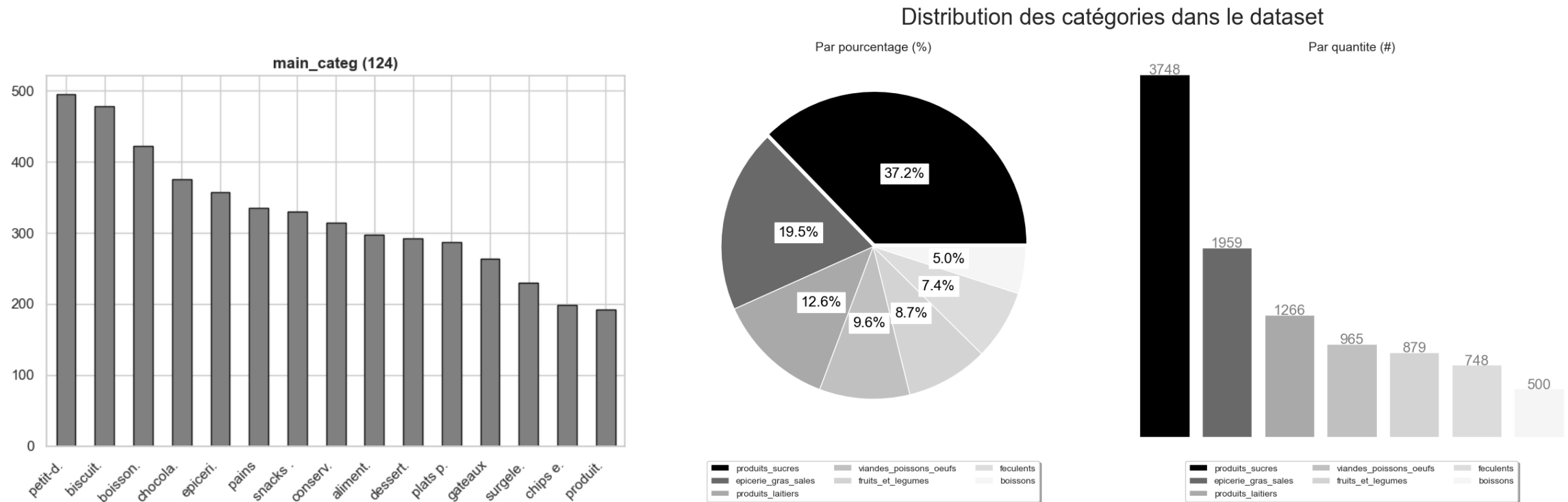
Distribution des labels AB dans le dataset



3. Démarche méthodologique d'exploration de données

1. Analyse univariée des différentes variables importantes avec les visualisations associées.

❖ Variables qualitatives : 124 sous catégories et 7 grandes catégories de produits

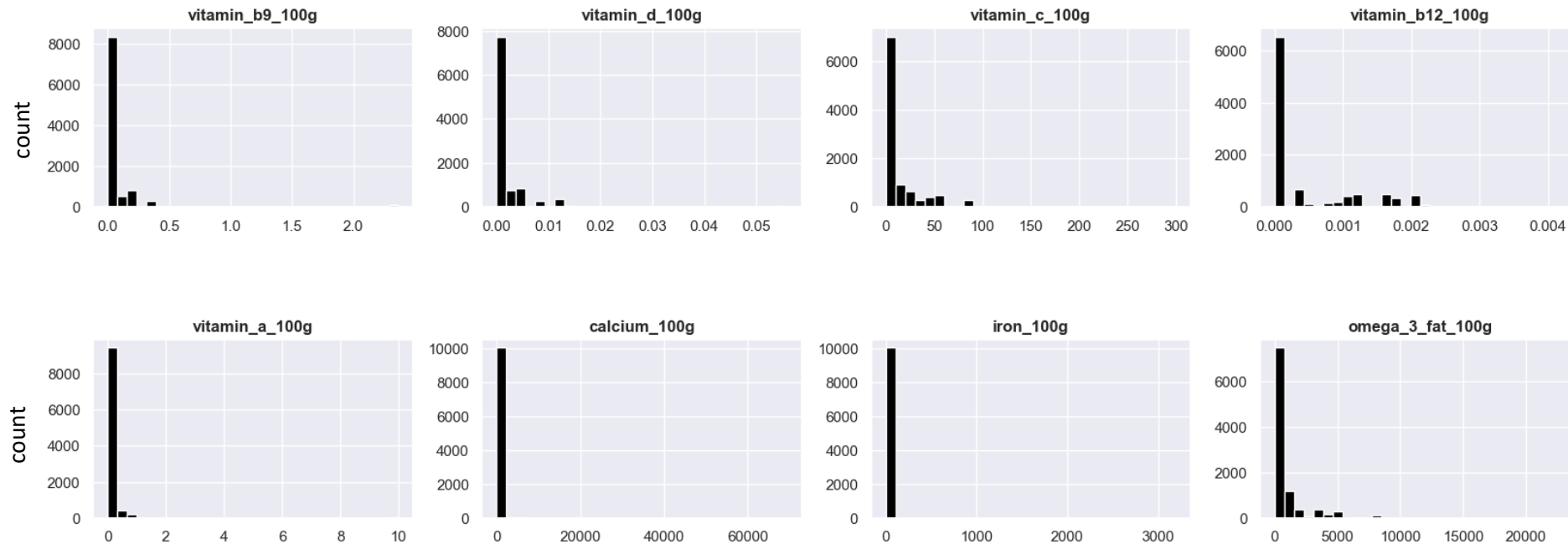


3. Démarche méthodologique d'exploration de données

1. Analyse univariée des différentes variables importantes avec les visualisations associées.

❖ Variables quantitatives :

→ Test de normalité (histogramme, Shapiro-Wilk)



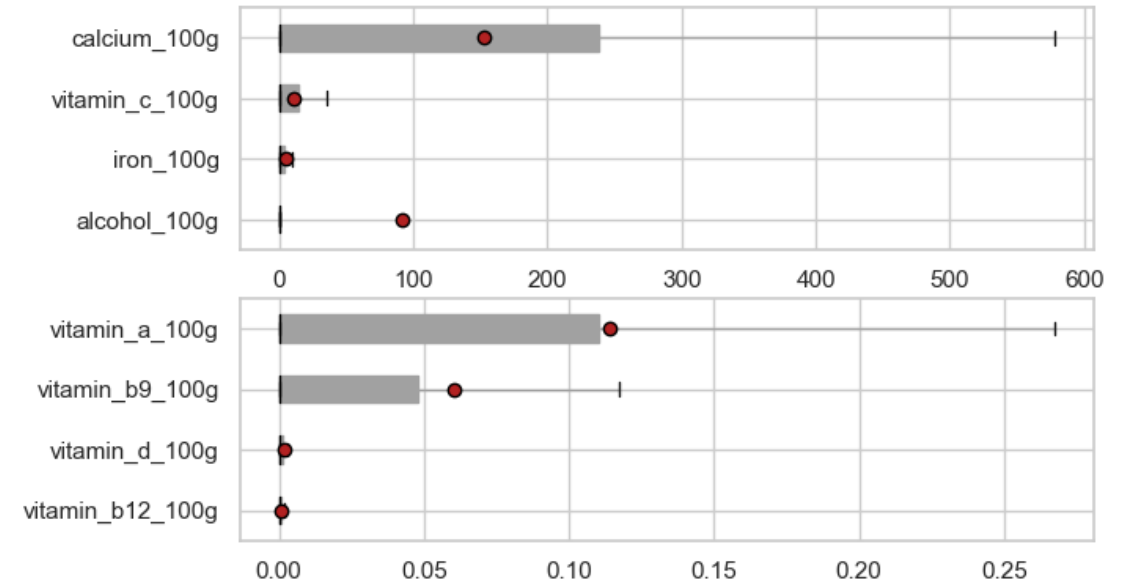
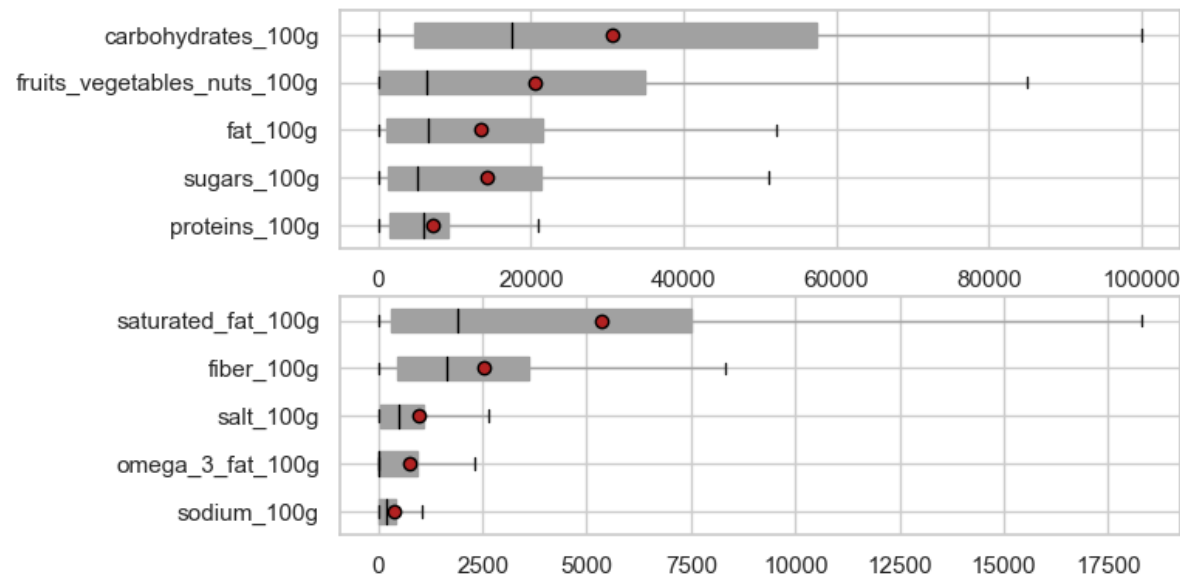
➤ Distributions non gaussiennes très rassemblées sur la gauche dans les variables d'études, (ici 8 principales sont présentées) $p < 0,05$. Nous ferons des tests non-paramétriques par la suite.

3. Démarche méthodologique d'exploration de données

1. Analyse univariée des différentes variables importantes avec les visualisations associées.

❖ **Variables quantitatives** (en mg, ou kJ pour l'énergie) pour 100 g ou 100 ml de produit

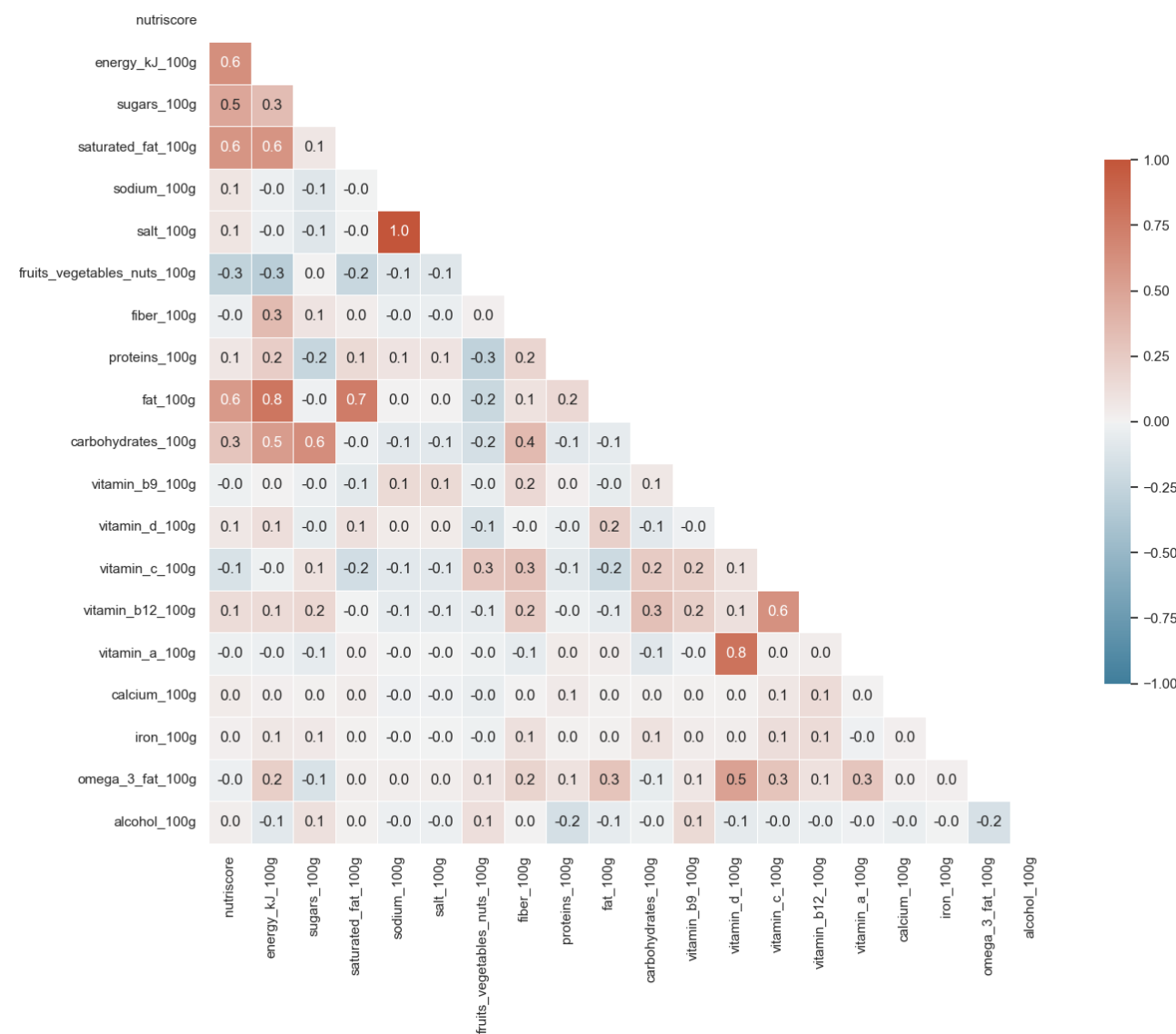
→ **Tendance centrale (boxplot, median, mean)**



- Certaines distributions sont très étalées, d'autres très resserrées
- Nous opérons un centrage puis une réduction de nos données avant de passer aux analyses multivariées

3. Démarche méthodologique d'exploration de données

2. Analyses bivariées entre variables quantitatives: Corrélations

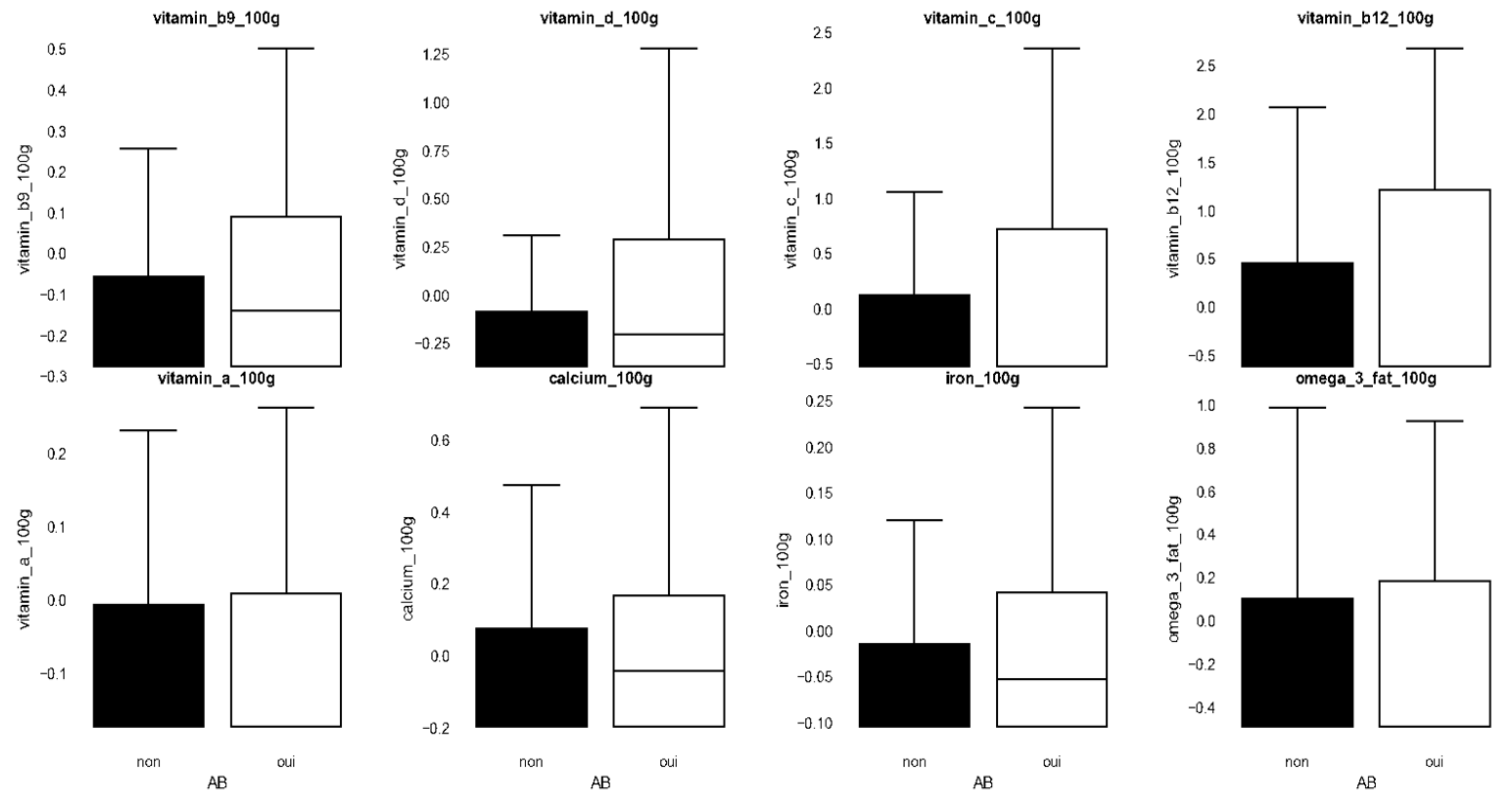


- Relations intuitives (sel-sodium, carbohydrates-sucre-fibres etc)
- L'énergie -> protéines-glucides-lipides
- Le Nutriscore -> lipides, sucres, énergie
- Vitamine B12 et C, Vitamine A et D, Omega3 et vitamine D.
- Aucune variables ne corrèlent avec la vitamine B9, le calcium, le fer

3. Démarche méthodologique d'exploration de données

2. Analyses bivariées entre une variable quantitative et qualitative:

❖ **Test non-paramétrique MW : les produits biologiques sont-ils plus vitaminés ?**



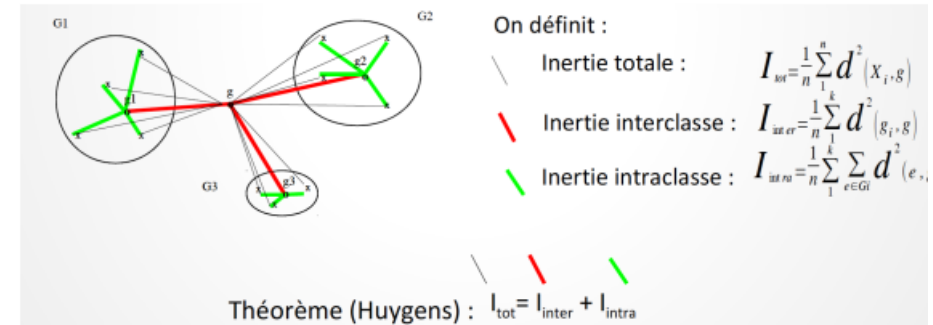
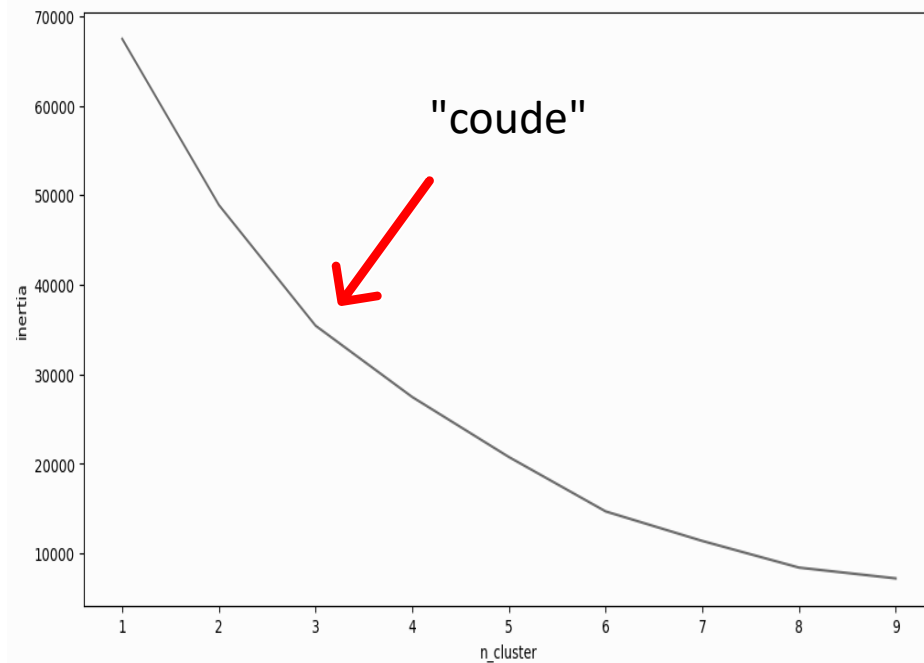
Variables	pvalue two-sided	Conclusion
vitamin_b9_100g	9.49e-21	positif
vitamin_d_100g	8.10e-10	positif
vitamin_c_100g	3.91e-28	positif
vitamin_b12_100g	1.21e-15	positif
vitamin_a_100g	8.59e-03	positif
calcium_100g	5.76e-13	positif
iron_100g	5.31e-25	positif
omega_3_fat_100g	7.32e-07	positif

➤ **Significativement plus de vitamines, minéraux et oméga3 dans les produits bios que dans les produits non-bios**

3. Démarche méthodologique d'exploration de données

3. Analyses multivariées entre variable quantitatives :

❖ Utiliser un algorithme de clustering (k-means).



Licence 3 MIAHS - Université de Bordeaux
Marie Chavent

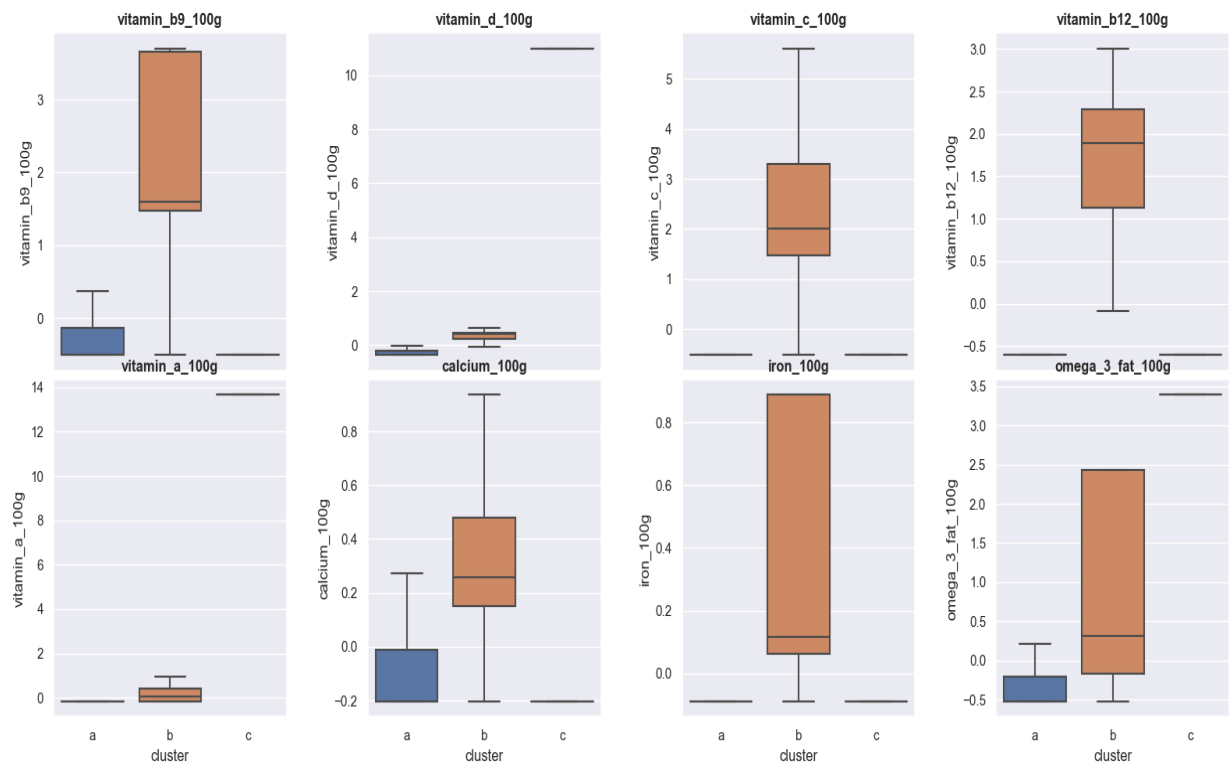
coefficient de silhouette = 0.64878166

- On voit que l'inertie diminue rapidement entre 2 et 3 clusters, et plus lentement après.
- L'algorithme des kmeans est appliqué à ce jeu de données avec : - K = 3 classes, - N = 3 répétitions de l'algorithme.
- Coefficient de silhouette satisfaisant (proche de 1), nous gardons et étudions ces groupes.

3. Démarche méthodologique d'exploration de données

3. Analyses multivariées entre variable quantitatives :

❖ Utiliser un algorithme de clustering (k-means).



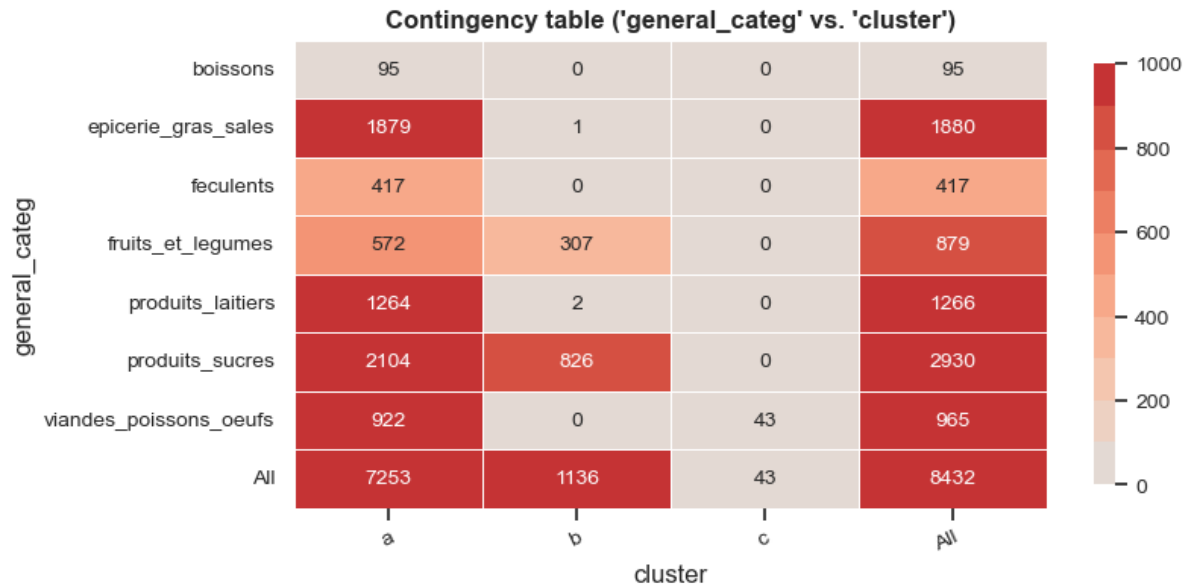
Variables	pvalue KW	pvalue wilc a vs b	pvalue wilc b vs c	pvalue wilc a vs c	Conclusions
vitamin_b9_100g	p<0,05	p<0,05	8.55e-29	0.002	b
vitamin_d_100g	p<0,05	p<0,05	7.66e-29	1.00e-29	c
vitamin_c_100g	p<0,05	p<0,05	1.20e-28	NS	b
vitamin_b12_100g	p<0,05	p<0,05	7.66e-29	0.01	b
vitamin_a_100g	p<0,05	p<0,05	7.66e-29	1.00e-29	c
calcium_100g	p<0,05	p<0,05	1.07e-28	1.56e-06	b
iron_100g	p<0,05	p<0,05	1.07e-28	0.01	b
omega_3_fat_100g	p<0,05	p<0,05	1.69e-25	8.20e-27	c

- Significativement plus de vitamines B, C et minéraux dans le groupe b que dans les autres.
- Significativement plus de vitamine D, A et oméga3 dans le groupe c que dans les autres.

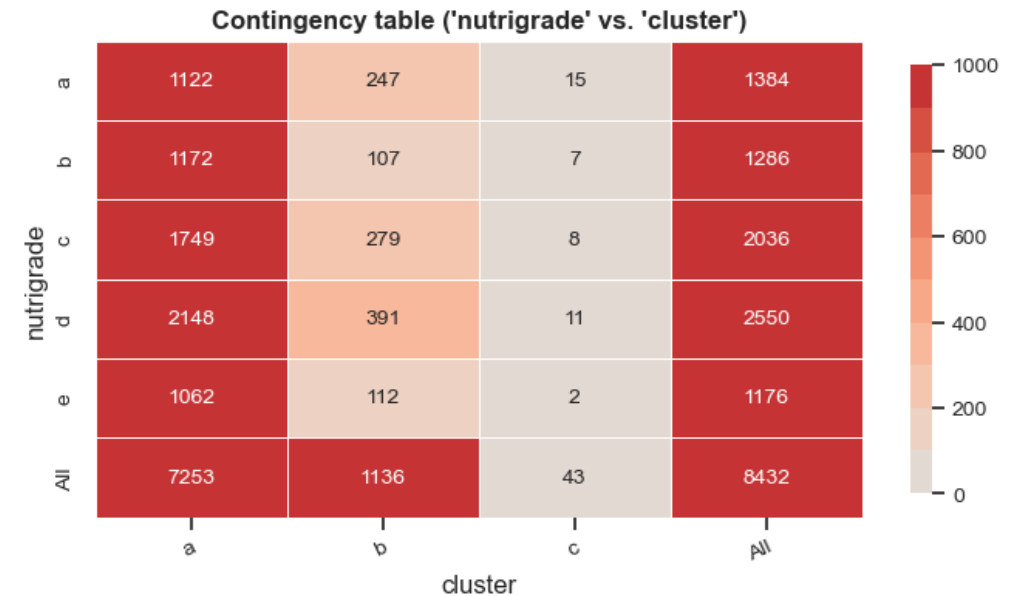
3. Démarche méthodologique d'exploration de données

3. Analyses multivariées entre variable quantitatives :

❖ Utiliser un algorithme de clustering (k-means).



---Chi-squared stat=1927, $p < 0.05$
probably dependent



---Chi-squared stat=89, $p < 0.05$
probably dependent

- Choix possible dans le groupe a , bien représenté dans chaque catégorie d'aliments
- Le groupe b et c beaucoup plus petit et spécifique

3. Démarche méthodologique d'exploration de données

3. Analyses multivariées entre variables quantitatives :

❖ Analyse en Composante principale: variabilité entre les individus



Vitamine B9	0.51	-0.27	-0.08	-0.13	0.00	0.27	0.70	0.28
Vitamine D	0.30	0.59	0.04	0.06	-0.13	-0.10	-0.22	0.70
Vitamine C	0.50	-0.26	-0.11	-0.10	0.01	0.49	-0.64	-0.12
Vitamine B12	0.45	-0.26	0.06	0.01	-0.42	-0.72	-0.07	-0.18
Vitamine A	0.21	0.59	0.07	0.07	-0.39	0.27	0.21	-0.57
Calcium	0.09	-0.08	0.98	0.02	0.16	0.07	-0.01	0.00
Fer	0.13	-0.10	-0.07	0.97	0.14	0.04	0.03	-0.01
Omega3	0.35	0.29	-0.11	-0.13	0.78	-0.29	0.03	-0.26
	F1	F2	F3	F4	F5	F6	F7	F8

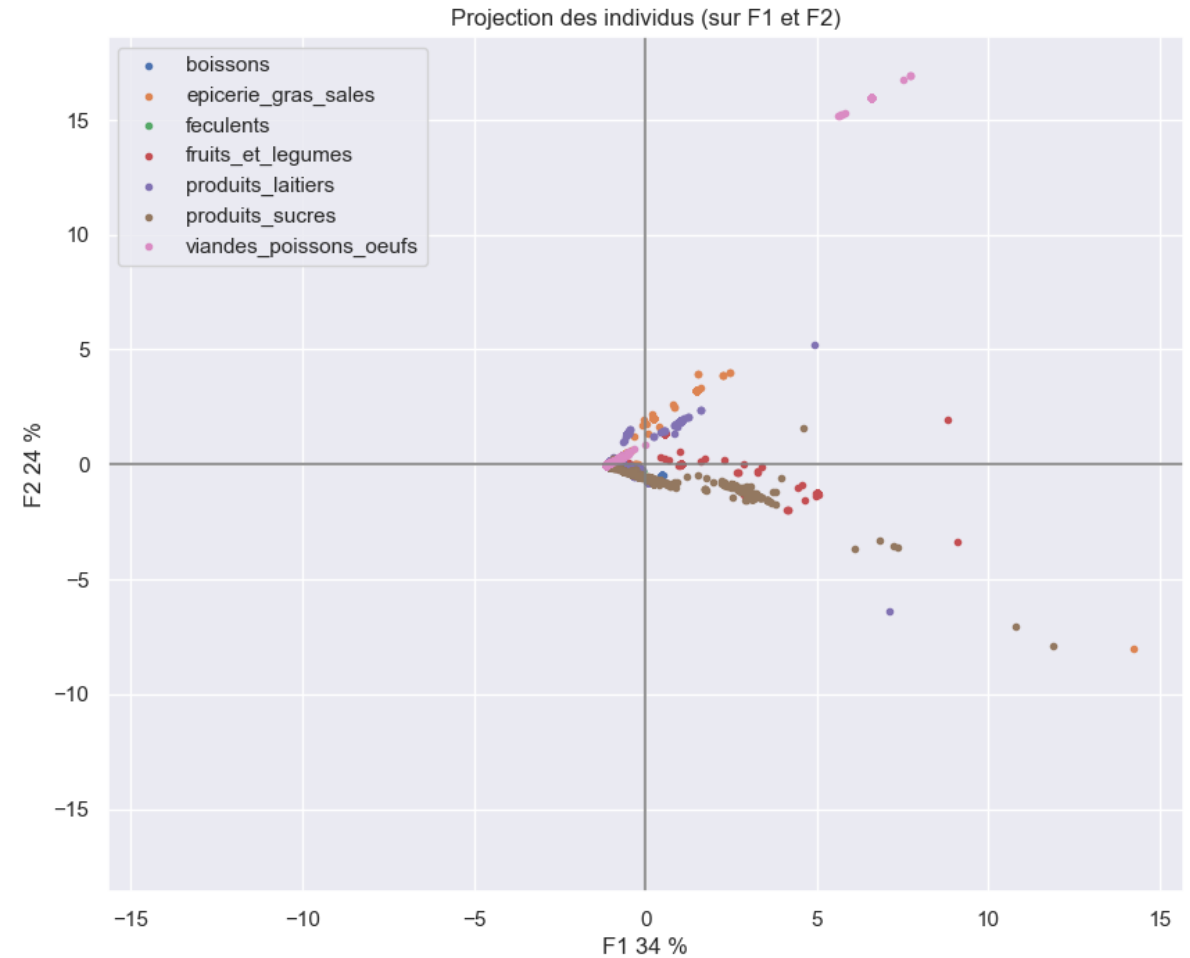
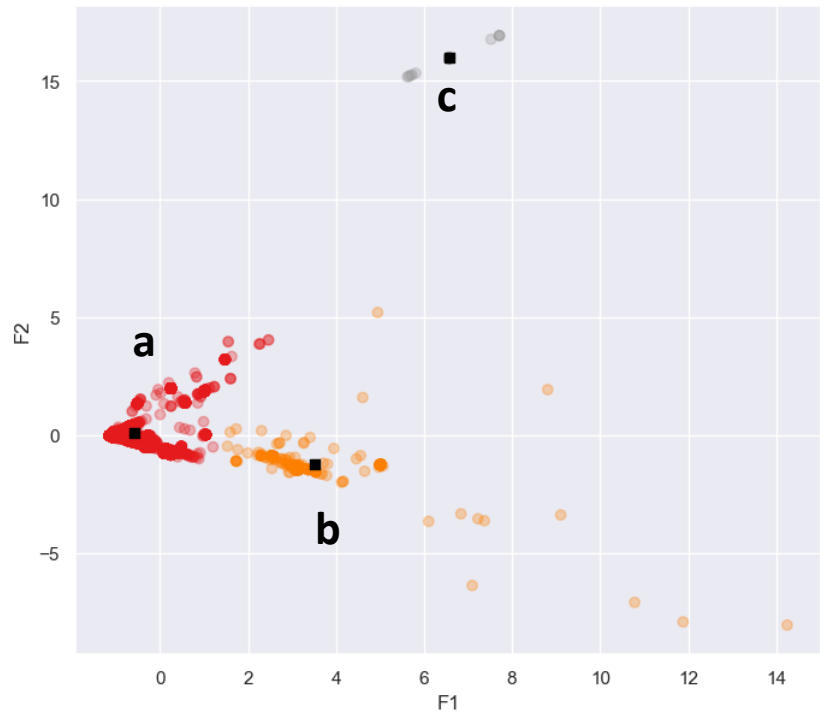
1.00
0.75
0.50
0.25
0.00
-0.25
-0.50
-0.75
-1.00

- On réalise une ACP pour étudier la **variabilité entre les individus**, c'est-à-dire quelles sont les différences et les ressemblances entre individus.
- Déterminer les axes qui absorbent le plus d'inertie possible.

3. Démarche méthodologique d'exploration de données

3. Analyses multivariées entre variables quantitatives :

❖ Analyse en Composante principale:



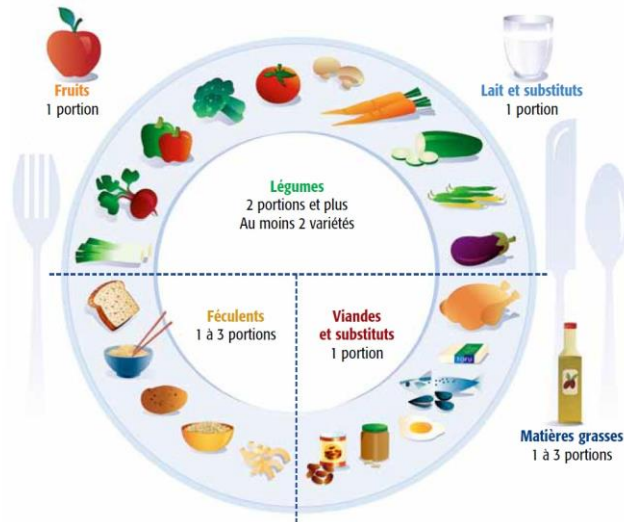
- Les observations sont redondantes avec notre étude k-means et khi2, l'intérêt de l'analyse en composante principale est limité dans notre cadre (mais voir annexes)

3. Démarche méthodologique d'exploration de données

3. Analyses multivariées entre variable quantitatives :

❖ Utiliser un algorithme de clustering (k-means).

1. Notification de produits quotidiens



Pour vous aujourd'hui :

- **Fruits et légumes** : 5
- **Produits sucrés** : 2
- **Féculents** : 3
- **Produits laitiers** : 3
- **Epicerie** : 2
- **Viandes et volailles** : 2

Fonction 1 : Recommander et inspirer

- ✓ Notification de liste produits
- ✓ Renouvellement chaque jour

Cette application est faite pour vous! C'est parti pour les

Notre selection de 5 fruits et légumes :

```
1954      quinoa
1434  galettes de maïs extra fines
1714      boulgour aux fruits secs
1537      quinoa bio
1710      lentilles corail
```

Name: product_name, dtype: object

Notre selection de 2 produits sucrés :

```
8634  cereales coeur fondant  aux noisettes au chocolat
7253      croustillant avoine fruits rouge bio
```

Name: product_name, dtype: object

Notre selection de 3 féculents :

```
921  farine de ble khorasan kamut  complete type 150
643      spaghetti bio
925      riz long complet
```

Name: product_name, dtype: object

Notre selection de 3 laitages :

```
2741  fromage blanc 0% au lait de brebis
2922      boisson au soja calcium
2902      fromage blanc
```

...

Notre selection de 2 viandes ou poissons ou oeufs :

```
3660      brandade de morue a la nimoise
4097  tranches de filets de colin d'alaska
```



Idée d'application

Nettoyage des données

Analyse des données

Conclusions

Pertinence et faisabilité de l'application:

- **Résumé méthodes**
- **Résumé analyses et perspectives**

4. Conclusions

Résumé des méthodes de ce projet 3 :

Étape 1 : Nettoyage de données

- 1 Lire les définitions des variables
- 2 Trouver une idée d'application
- 3 Filtrer votre jeu de données
- 4 Traiter les valeurs aberrantes
- 5 Traiter les valeurs manquantes

Étape 2 : Analyse exploratoire

- 1 Analyses univariées
- 2 Analyses bivariées
- 3 Analyses multivariées

Une base de données propre

- ✓ sans valeurs aberrantes
 - ✓ sans doublons
- ✓ sans valeurs manquantes:
3 méthodes utilisées

Analyses univariées

- ✓ Variables qualitatives
 - ✓ Variables quantitatives
- #### Analyses bivariées
- ✓ Quant-quant : Corrélations
 - ✓ Quali-quant : MW et KW

Un contenu adapté

- ✓ des produits identifiables
- ✓ des catégories pertinentes
- ✓ des données chiffrées utiles

Analyses multivariées

- ✓ k-means et KW
- ✓ Analyses en Composantes principales

4. Conclusions

Résumé des analyses et perspectives de ce projet 3 :

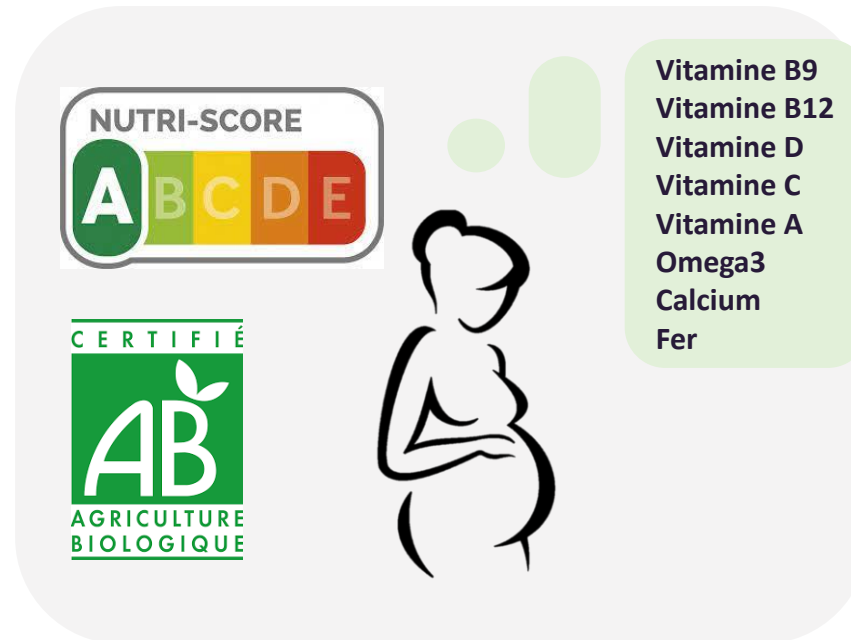
Analyses bivariées

- L'énergie corrélée avec protéines, glucides, lipides pour 100g de produits.
- Le nutriscore corrélé avec lipides, sucre, et l'énergie pour 100g de produits.
- Vitamine B12 et C / Vitamine A et D / Omega3 et vitamine D (ex: poissons)

#Cas des produits difficiles à catégoriser -> imprécisions. #Considérer les quantités et les seuils de nutriments.

- Plus de vitamines dans les produits bios

#Autres labels à considérer



<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9182711/>
<https://www.nhs.uk/pregnancy/keeping-well/vitamins-supplements-and-nutrition/>

Analyses multivariées

- Plus de vitamines B, C et minéraux dans le groupe B. Plus de vitamine D, A et oméga3 dans le groupe C que dans les autres.
- Groupes B et C : spécifiques. Le groupe A laisse plus de choix de produits.
- **Nous incluons les groupes kmeans car ils mettent en lumière des liens intéressants entre produits.**

#Déséquilibre du nombre de produits intergroupes toutefois.

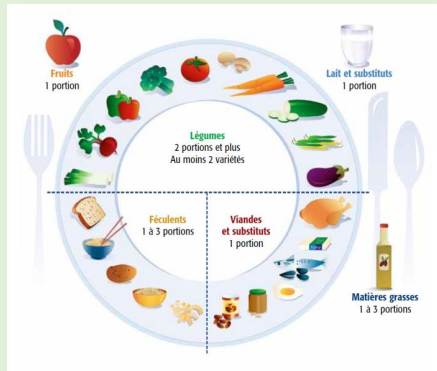
#C ne permet pas une sélection végétarienne.

4. Conclusions

Une application pour la santé :

Aider les consommatrices enceintes à identifier les produits à prioriser et à éviter

1. Notification de produits quotidiens



Pour vous aujourd'hui :

- Fruits et légumes : 5
- Produits sucrés : 2
- Féculents : 3
- Produits laitiers : 3
- Epicerie : 2
- Viandes et volailles : 2

2. Classer les produits en 3 niveaux lors des achats



Fonction 1 : Recommander et inspirer

- Notification de liste produits
- Renouvellement chaque jour

Fonction 2 : Classer et guider

- Diviser les produits en trois groupes
- Améliorer la consommation
- prévenir les risques

✓ Pertinence du projet : **contribuer à réduire les complications/affections en lien avec la grossesse.**

✓ Faisabilité du projet : **conditionnée par un effort d'étiquetage/renseignement sur la teneur en vitamines (minéraux et AG) essentielles sur les produits.**



Projet 3 : Concevez une application au service de la santé publique

L'agence "Santé publique France" a lancé un appel à projets pour trouver des idées innovantes d'applications en lien avec l'alimentation. Vous souhaitez y participer et proposer une idée d'application.

Merci pour votre attention

ANNEXES

2. Démarche méthodologique de nettoyage

1/Filtrage global et orienté projet :

a) Colonnes vides (-31 colonnes vides) =

```
#Nous identifions 31 colonnes vides (%NaN = 100) dans l'analyse ci-dessus
empty_col = ["serum-proteins_100g","gondoic-acid_100g","oleic-acid_100g",
             "molybdenum_100g","maltodextrins_100g","nucleotides_100g",
             "arachidonic-acid_100g","lauric-acid_100g","maltose_100g",
             "capric-acid_100g","myristic-acid_100g","palmitic-acid_100g",
             "stearic-acid_100g","montanic-acid_100g","caprylic-acid_100g",
             "no_nutriments","glycemic-index_100g","lignoceric-acid_100g",
             "chlorophyl_100g","cerotic-acid_100g","melissic-acid_100g",
             "elaidic-acid_100g","ingredients_from_palm_oil","mead-acid_100g",
             "erucic-acid_100g","nervonic-acid_100g","caproic-acid_100g",
             "butyric-acid_100g","nutrition_grade_uk",
             "ingredients_that_may_be_from_palm_oil","water-hardness_100g"]

print("shape avant", df.shape)
df = df.drop(columns=empty_col)
print("shape après", df.shape)
```

[1203] ✓ 0.1s

... shape avant (320772, 162)
shape après (320772, 131)

b) Colonnes redondantes

_tags est la version multilingue normalisée de la variable d'origine et **_fr** la version française des tags, nous exploiterons par la suite les colonnes **_fr**.

par ailleurs on note la redondance de "categories_fr" avec "main_category_fr", que l'on supprime (la 1ere)

```
print("shape avant", df.shape)
df = df.drop(columns=["countries","countries_tags",
                     "created_t", "last_modified_datetime",
                     "categories","categories_fr", "main_category",
                     "categories_tags",
                     "labels","labels_tags",
                     "traces","traces_tags",
                     "additives", "additives_tags"])

print("shape après", df.shape)
```

✓ 0.2s

shape avant (320772, 131)
shape après (320772, 117)

-14 colonnes redondantes

2. Démarche méthodologique de nettoyage

1/Filtrage global et orienté projet :

c) Colonnes pertinentes

- Colonnes liées au nutriscore
- Colonne label qualité AB
- Colonnes liées aux éléments nutritifs importants dans la grossesse

```
print("shape avant", df.shape)
df= df[["code","creator","created_datetime", "product_name",
        "main_category_fr","countries_fr","labels_fr","nutrition_grade_fr",
        "nutrition_score_fr_100g","energy_100g","sugars_100g",
        "saturated_fat_100g","sodium_100g","salt_100g",
        "fruits_vegetables_nuts_100g","fiber_100g","proteins_100g",
        "carbohydrates_100g","vitamin_b9_100g", "vitamin_d_100g",
        "vitamin_c_100g", "vitamin_b12_100g", "vitamin_a_100g",
        "calcium_100g","iron_100g", "omega_3_fat_100g",
        "alcohol_100g","fat_100g"]]
print("shape après", df.shape)
```

✓ 0.2s

shape avant (320772, 117)

shape après (320772, 28)

-89 colonnes inutiles



Vitamine B9
Vitamine B12
Vitamine D
Vitamine C
Vitamine A
Omega3
Calcium
Fer

<https://www.ncbi.nlm.nih.gov/pmc/articles/pMC9182711/>
<https://www.nhs.uk/pregnancy/keeping-well/vitamins-supplements-and-nutrition/>

2. Démarche méthodologique de nettoyage

1/Filtrage global et orienté projet :

d) Lignes pertinentes : pays, date, source, code et catégories

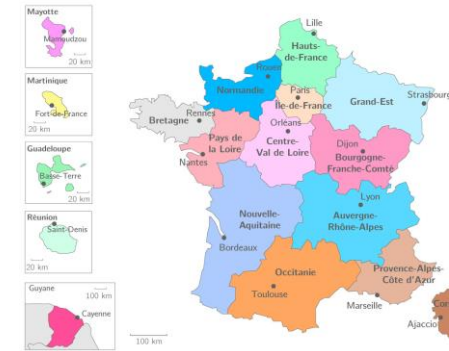
pAYS = Nous aurions pu filtrer avec "france" ou "french" dans _tags mais DOM TOM non inclus

```
print("shape avant", df.shape)
df_fr = df.loc[(df["countries_fr"] == 'France') | (df["countries_fr"] == 'Guadeloupe') | (df["countries_fr"] == 'Guyane') |
(df["countries_fr"] == 'Martinique') | (df["countries_fr"] == 'La Réunion') | (df["countries_fr"] == 'Mayotte') |
(df["countries_fr"] == 'Nouvelle-Calédonie') | (df["countries_fr"] == 'Polynésie française') | (df["countries_fr"] == 'Saint-Martin') |
| (df["countries_fr"] == 'Saint-Pierre-et-Miquelon') | (df["countries_fr"] == 'Wallis-et-Futuna'), :]
print("shape après", df_fr.shape)
```

✓ 0.1s

shape avant (320772, 28)
shape après (94961, 28)

-225 811 lignes inutiles (et -1 colonne)



CODE = l'utilisatrice doit pouvoir scanner le produit, code barre requis

✓ **Chaque produit a un code**

CATEG = Catégories requises pour l'analyse

```
print("shape avant", df_fr.shape)
df_fr = df_fr.dropna(axis=0, subset='main_category_fr')
print("shape après", df_fr.shape)
```

shape avant (94961, 27)
shape après (58953, 27)

-36 008 lignes inutiles

2. Démarche méthodologique de nettoyage

2/Traitement des valeurs aberrantes/outliers:

DATE = suppression des données antérieures au lancement d'Open Food Facts (19-05-2012)

```
print("shape avant", df_fr.shape)
df_fr["created_datetime"] = df_fr["created_datetime"].astype("datetime64")
df_fr = df_fr[df_fr["created_datetime"].dt.strftime("%Y-%m-%d") > "2012-05-19"]
print("shape après", df_fr.shape)
```

shape avant (58953, 27)
shape après (58522, 27)

-431 lignes dates erronées (et -1 colonne)



CREATOR = filtrage sur les contributeurs Open Food Facts

```
print("shape avant", df_fr.shape)
df_fr = df_fr.loc[df_fr["creator"] == "openfoodfacts-contributors", :]
print("shape après", df_fr.shape)
```

shape avant (58522, 26)
shape après (15341, 26)

-43 181 lignes sources (et -1 colonne)

2. Démarche méthodologique de nettoyage

2/Traitement des valeurs aberrantes/outliers:

METADATA (153241, 5) + NUTRINTEREST (15341, 21)

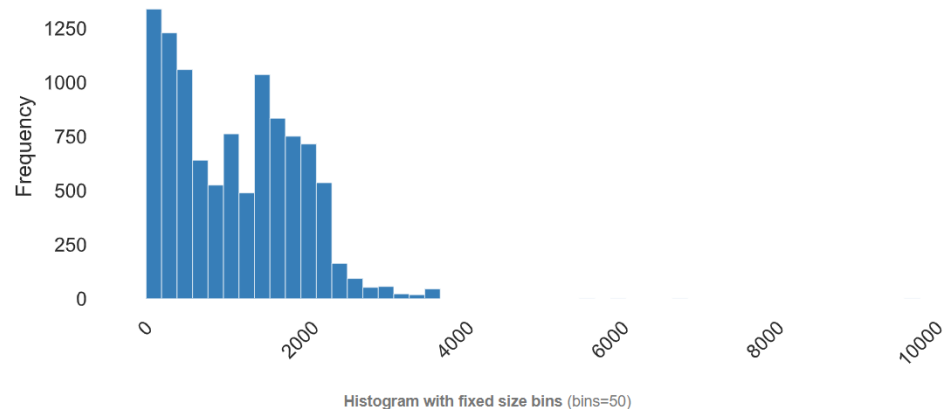
✓ **Nutriscore compris entre -15 et 40 selon la littérature**

```
nutrinterest.describe()
```

✓ 0.1s

	nutrition_score_fr_100g	energy_100g	sugars_100g	saturated_fat_100g	sodium_100g	salt_100g
count	11216.00	11739.00	11433.00	11417.00	11422.00	11422.00
mean	8.72	1421.81	14.40	5.36	0.40	1.02
std	8.89	30008.85	19.16	8.55	1.71	4.34
min	-14.00	0.00	0.00	0.00	0.00	0.00
25%	1.00	398.00	1.30	0.20	0.02	0.05
50%	9.00	1100.00	5.25	1.90	0.19	0.49
75%	15.00	1740.00	21.20	7.50	0.43	1.10
max	40.00	3251373.00	103.50	95.00	49.21	125.00

-> **Cas particulier energy_100g : kJ (descriptif OFF), filtrage <5000kJ_100g**



-> **Suppression des données >à 100, nous nous contentons de ce filtrage pour identifier les outliers.**

```
print("shape avant", nutrinterest.shape)
nutrinterest = nutrinterest.loc[nutrinterest['energy_100g'] < 5000,:]
print("shape apres", nutrinterest.shape)
```

✓ 0.1s

shape avant (15341, 21)
shape apres (11731, 21)

```
print("shape avant", nutrinterest.shape)
nutrinterest = nutrinterest.loc[(nutrinterest["sugars_100g"] <= 100) &
                                (nutrinterest["energy_100g"] < 5000)]
print("shape apres", nutrinterest.shape)
```

✓ 0.0s

shape avant (11731, 21)
shape apres (10364, 21)

- 4 977 lignes

MERGE METADATA + NUTRINTEREST

DF_App = 10364 lignes et 25 colonnes

2. Démarche méthodologique de nettoyage

3/Traitement des doublons: -> pas de doublons

```
df_app.loc[df_app[['code', 'product_name']].duplicated(keep=False),:]
```

✓ 0.0s

code	nutrition_score_fr_100g	energy_100g	sugars_100g	saturated_fat_100g
------	-------------------------	-------------	-------------	--------------------

0 rows × 5 columns

4/Imputations des valeurs manquantes:

-> Suppression des catégories sous représentées (- de 5 produits/ categ)

-> Nous récupérons la liste de 124 catégories et nous filtrons par la formule mask

```
print("NaN avant", df_app['main_category_fr'].isna().sum())
mask = ~df_app['main_category_fr'].isin(lot_cat)
df_app.loc[mask, 'main_category_fr'] = np.NaN
print("NaN après", df_app['main_category_fr'].isna().sum())
```

✓ 0.0s

NaN avant 0
NaN après 299

```
print("shape avant", df_app.shape)
df_app = df_app.dropna(axis= "rows", subset = 'main_category_fr')
print("shape après", df_app.shape)
print("NaN après", df_app['main_category_fr'].isna().sum())
```

✓ 0.0s

shape avant (10364, 25)
shape après (10065, 25)
NaN après 0

- 299 lignes

DF_App = 100065 lignes et 26 colonnes






2. Démarche méthodologique de nettoyage

4/Imputations des valeurs manquantes:

c) Imputation par intervalle : nutrigrade

- Attribution des couleurs

Le logo Nutri-Score est ensuite attribué en fonction du score obtenu (cf. tableau ci-dessous).

Points		Logo
Aliments solides	Boissons	
Min à -1	Eaux	
0 à 2	Min à 1	
3 à 10	2 à 5	
11 à 18	6 à 9	
19 à Max	10 à Max	

```
boi = df_cleaned.groupby('general_categ').get_group('boissons').index
no_boi = pd.Index(set(df_cleaned.index)-set(boi))
eau = df_cleaned[df_cleaned['main_category_fr']=='eaux'].index

nutri_verif = pd.Series(index=df_cleaned.index, dtype='object')

nutri_verif.loc[boi] = pd.cut(df_cleaned.loc[boi,'nutrition_score_fr_100g'],
                              [-15,1,5,9,40], labels=list('bcde')).astype('object')
nutri_verif.loc[eau] = 'a'

nutri_verif.loc[no_boi] = pd.cut(df_cleaned.loc[no_boi,'nutrition_score_fr_100g'],
                              [-15,-1,2,10,18,40], labels=list('abcde')).astype('object')

df_cleaned['nutri_verif'] = nutri_verif

df_cleaned[['nutri_verif','nutrition_grade_fr']]

df_cleaned[['nutri_verif','nutrition_grade_fr']].isna().sum()

nutri_verif          0
nutrition_grade_fr  119
dtype: int64
```

2. Démarche méthodologique de nettoyage

4/Imputations des valeurs manquantes:

c) Variable labels bio AB

Le label **Agriculture biologique (ou label AB)** est un label de qualité français créé en 1985, et fondé sur l'interdiction d'utilisation de produits issus de la chimie de synthèse. Il permet d'identifier les produits issus de l'agriculture biologique.



```
df_bio = pd.DataFrame(df_cleaned.loc[df_cleaned["labels_fr"].str.contains("Agriculture biologique")])
df_bio['AB'] = "oui"
df_bio = df_bio[['code', 'AB']]
df_bio

df_cleaned = pd.merge(df_cleaned, df_bio, on='code', how='left')
df_cleaned['AB'] = df_cleaned['AB'].fillna("non")
df_cleaned['AB'].value_counts()

✓ 0.2s
non      8668
oui       1397
Name: AB, dtype: int64
```

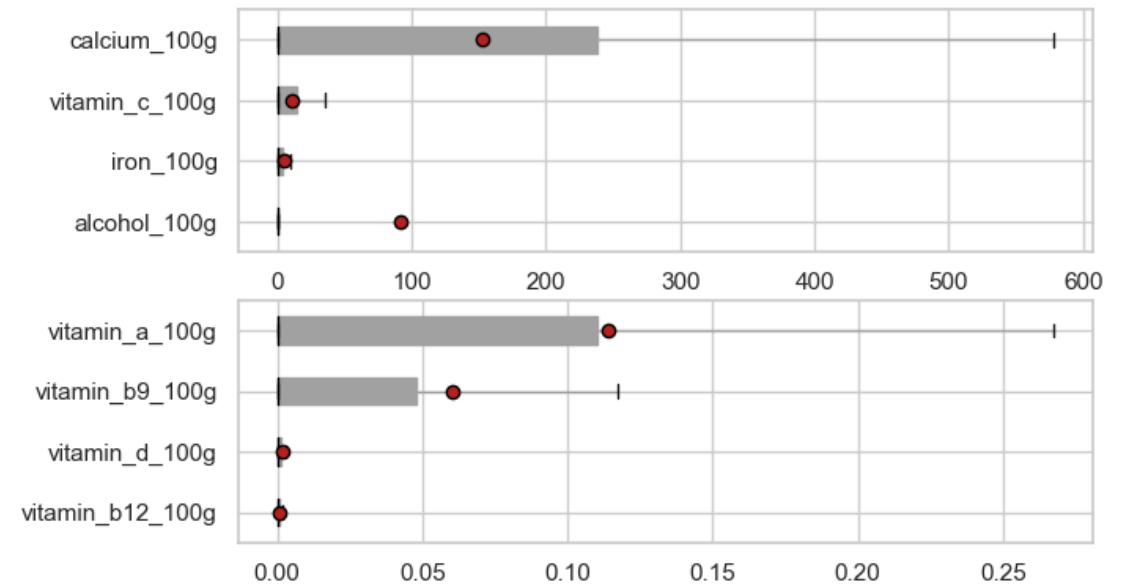
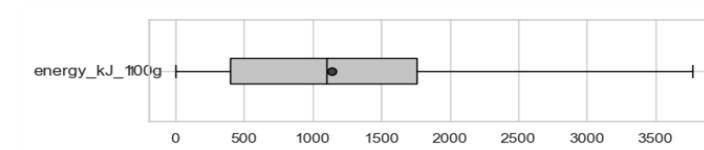
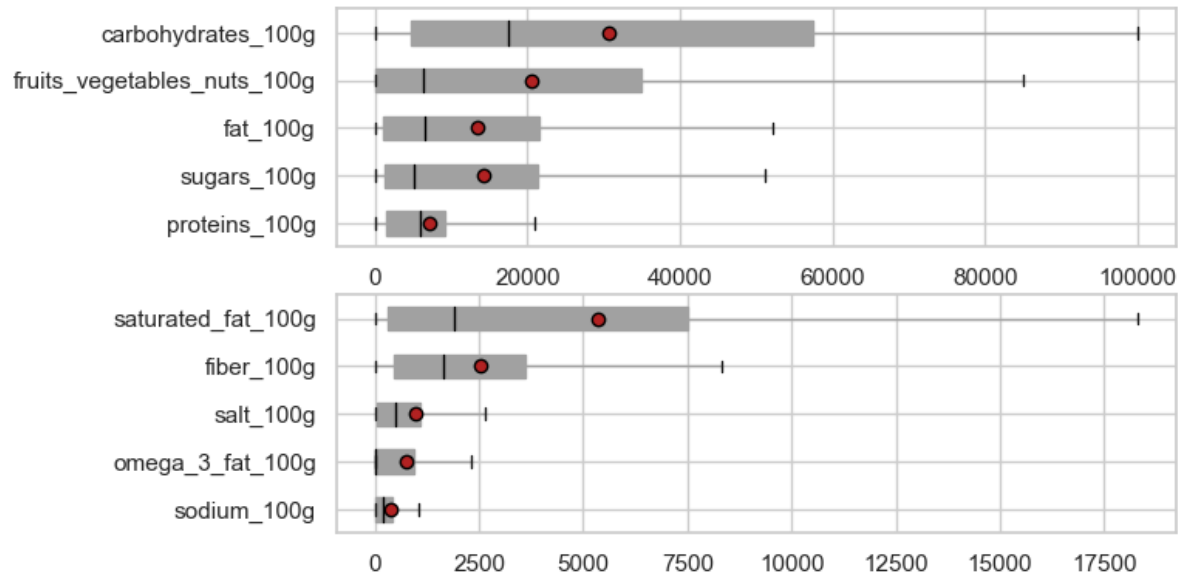
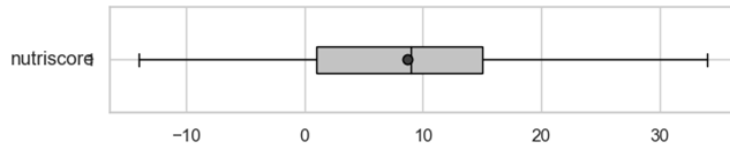
Variables	pvalue two sided
nutriscore	7.50e-42
energy_kJ_100g	7.20e-02
vitamin_b9_100g	9.49e-21
vitamin_d_100g	8.10e-10
vitamin_c_100g	3.91e-28
vitamin_b12_100g	1.21e-15
vitamin_a_100g	8.59e-03
calcium_100g	5.76e-13
iron_100g	5.31e-25
omega_3_fat_100g	7.32e-07

3. Démarche méthodologique d'exploration de données

1. Analyse univariée des différentes variables importantes avec les visualisations associées.

❖ **Variables quantitatives** (en mg, ou kJ pour l'energy) pour 100 g ou 100 ml de produit

→ **Tendance centrale (boxplot, median, mean)**



➤ Certaines distributions très étalées, d'autres très resserrées

➤ Nous opérons un centrage puis une réduction de nos données avant de passer aux analyses multivariées

3. Démarche méthodologique d'exploration de données

3. Analyses multivariées entre variables quantitatives :

❖ Analyse en Composante principale:

