

Préparez des données
pour un organisme de
santé publique





Sommaire

PARTIE 1 - PRÉSENTATION JEU DE DONNÉES & PROPOSITION D'APPLICATION

1. Caractéristiques générales
2. Une idée d'application
3. Sélection des variables

PARTIE 2 - NETTOYAGE

1. Doublons / Formatage / Valeurs aberrantes
2. Imputation des valeurs manquantes

PARTIE 3 – EXPLORATION

1. Analyses univariées
2. Analyses bivariées – pairplot & ANOVA
3. Analyses multivariées – ACP

CONCLUSION



PARTIE 1 - PRÉSENTATION JEU DE DONNÉES & PROPOSITION D'APPLICATION

1.1. Caractéristiques générales

code	url	creator	created_t	created_datetime	last_modified_t	last_modified_datetime	product_name	generic_name	quantity	...	ph_100g	fruits-vegetables-nuts_100g	collagen-meat-protein-ratio_100g	cocoa_100g	chlorophyl_100g	carbon-footprint_100g	nutrition-score_fr_100g	nutrition-score_uk_100g	glycemic-index_100g	water-hardness_100g
0	0000000003087	http://world-fr.openfoodfacts.org/produit/0000...	openfoodfacts-contributors	1474103866	2016-09-17T09:17:46Z	1474103893	2016-09-17T09:18:13Z	Farine de blé noir	NaN	1kg	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	0000000004530	http://world-fr.openfoodfacts.org/produit/0000...	usda-ndb-import	1489069957	2017-03-09T14:32:37Z	1489069957	2017-03-09T14:32:37Z	Banana Chips Sweetened (Whole)	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	14.0	14.0	NaN	NaN
2	0000000004559	http://world-fr.openfoodfacts.org/produit/0000...	usda-ndb-import	1489069957	2017-03-09T14:32:37Z	1489069957	2017-03-09T14:32:37Z	Peanuts	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	0.0	0.0	NaN	NaN
3	0000000016087	http://world-fr.openfoodfacts.org/produit/0000...	usda-ndb-import	1489055731	2017-03-09T10:35:31Z	1489055731	2017-03-09T10:35:31Z	Organic Salted Nut Mix	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	12.0	12.0	NaN	NaN
4	0000000016094	http://world-fr.openfoodfacts.org/produit/0000...	usda-ndb-import	1489055653	2017-03-09T10:34:13Z	1489055653	2017-03-09T10:34:13Z	Organic Polenta	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

↪ 320 000 lignes
 ⇔ 162 colonnes

4 types de variables :

- Informations générales : nom, date de modification, etc.
- Ensemble de tags : catégorie du produit, localisation, origine, etc.
- Ingrédients composant les produits et leurs additifs éventuels
- Informations nutritionnelles :
 - quantité en grammes d'un nutriment pour 100 grammes du produit
 - des scores nutritionnels (comme le nutriscore)

1.2. Proposition d'application

Que pourrait-on faire de ces données ?

- Aider malades et médecins



Flot de maladies chroniques :

- Hypertension
- Diabète
- Cholestérol
- Obésité

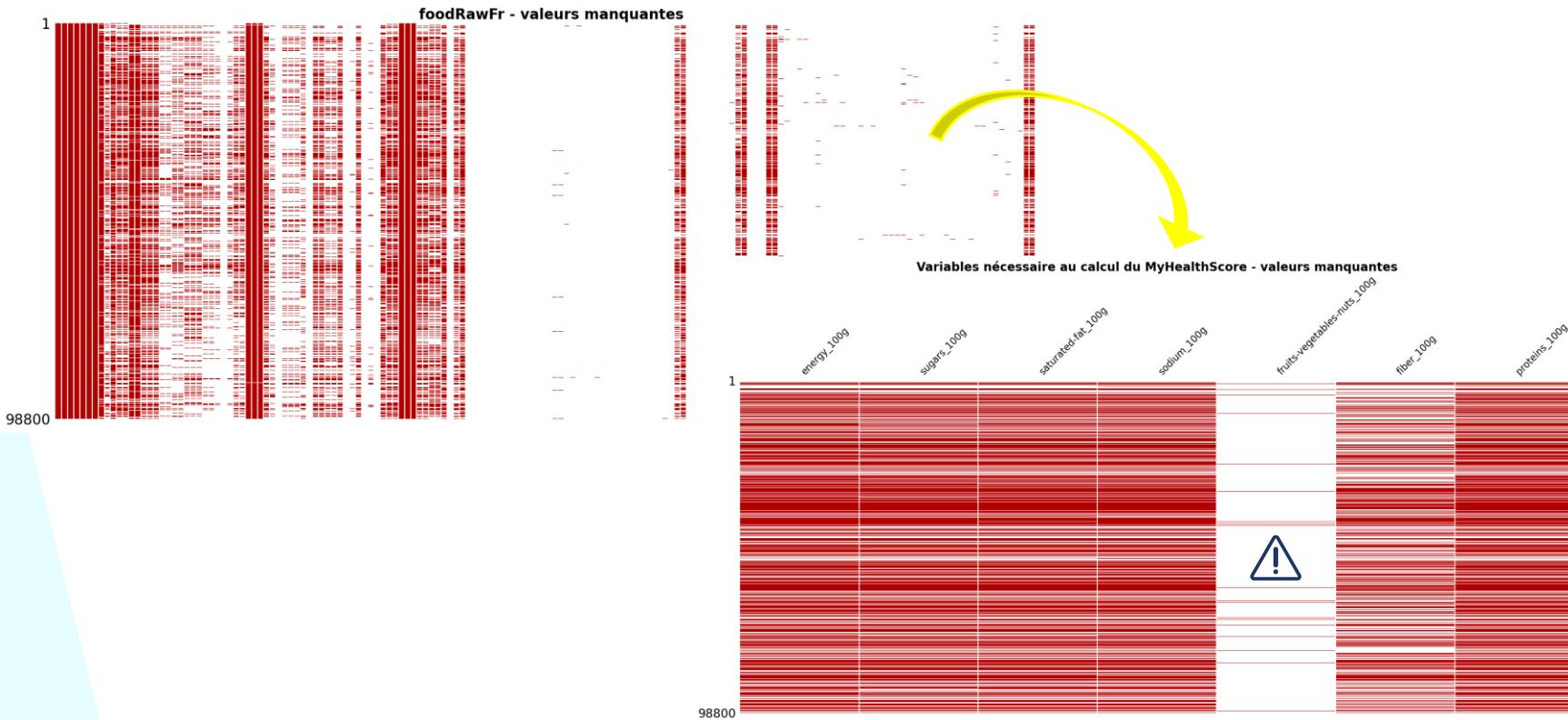
Cause/Solution ➔ régime alimentaire

Adapter le fameux score nutritionnel.
Le personnalisé pour chaque patient :

MyHealthScore



1.3. Sélection des variables

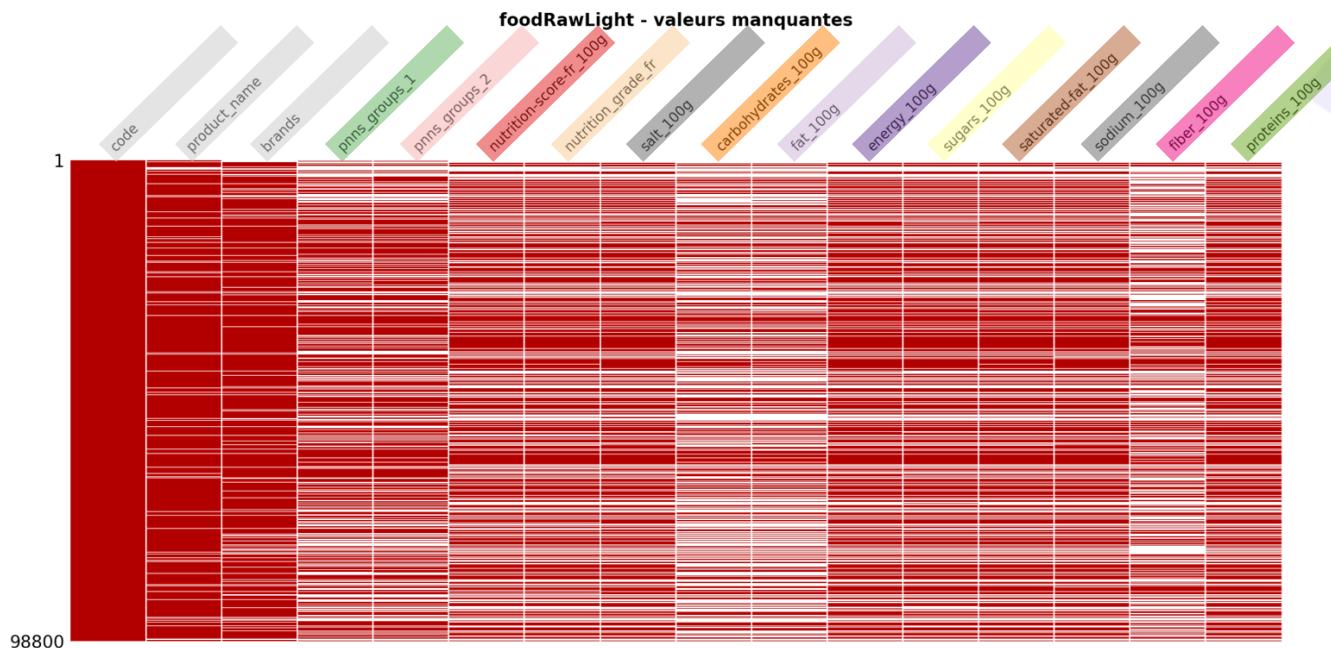


Colonne « Fruits et Légumes » vide !!

➡ Obligé de se tourner vers d'autres données.

1.3. Sélection des variables

1. Nous baser sur le taux de remplissage de nos colonnes
2. Utiliser le score nutritionnel `nutrition-score-fr_100g` et ses composantes connues ... `_100g` pour calculer un score différent, notre MyHealthScore, avec des pondérations personnalisées





PARTIE 2 - NETTOYAGE

2.1. Doublons / Formatage / Valeurs aberrantes

Colonne code

- Formatage : produit sous système EAN 8 ou EAN 13
- Pas de doublon

Colonnes product_name et brands

- On trouve des doublons :

product_name	brands	code
Coca-Cola	Coca-Cola	20
Pom'potes	Materne	15
Nutella	Ferrero,Nutella	14
Orangina	Orangina	14
Nesquik	Nestlé,Nesquik	11
Perrier	Perrier	11
Auchan	Auchan	10
Nutella	Ferrero	10
Mousline	Maggi	9

2.1. Doublons / Formatage / Valeurs aberrantes

Colonnes pnns_groups_1 et pnns_groups_2

- Formatage :

```
Index(['Beverages', 'Cereals and potatoes',  
       'Composite foods', 'Fat and sauces',  
       'Fish Meat Eggs', 'Fruits and vegetables',  
       'Milk and dairy products', 'Salty snacks',  
       'Sugary snacks', 'cereals-and-potatoes',  
       'fruits-and-vegetables', 'sugary-snacks',  
       'unknown', 'nan'],  
      dtype='object')
```

- Imputations logiques :

```
'Beverages', <-- 'Alcoholic beverages'  
'Cereals and potatoes',  
'Composite foods',  
'Fat and sauces',  
'Fish meat eggs', <-- 'Tripe dishes'  
'Fruits and vegetables',  
'Milk and dairy products',  
'Salty snacks',  
'Sugary snacks'
```

2.1. Doublons / Formatage / Valeurs aberrantes

	nutrition_score_fr_100g	salt_100g	carbohydrates_100g	fat_100g	energy_100g	sugars_100g	saturated_fat_100g	sodium_100g	fiber_100g	proteins_100g
count	59076.000000	60197.000000	45154.000000	45585.000000	6.215600e+04	60145.000000	60007.000000	60194.000000	44059.000000	61881.000000
mean	8.676366	1.162005	27.839665	13.293629	1.174376e+03	13.396842	5.451226	0.457503	2.576606	7.787648
std	9.038369	4.261100	27.348395	16.752498	1.308400e+04	19.010655	8.554641	1.677578	4.682310	7.889584
min	-15.000000	0.000000	0.000000	0.000000	0.000000e+00	-0.100000	0.000000	0.000000	0.000000	0.000000
25%	1.000000	0.080000	4.100000	1.400000	4.310000e+02	1.000000	0.300000	0.031496	0.000000	1.900000
50%	9.000000	0.560000	14.900000	7.000000	1.037000e+03	4.000000	2.000000	0.220472	1.400000	6.000000
75%	15.000000	1.250000	53.000000	21.000000	1.648000e+03	17.860000	7.500000	0.492126	3.200000	11.000000
max	40.000000	211.000000	190.000000	380.000000	3.251373e+06	105.000000	210.000000	83.000000	178.000000	100.000000

- Sugars_100g : valeurs négatives
- ..._100g : valeurs maximales > 100 ...
- energy_100g : valeurs maximales > 3700 ...

➡ valeurs manquantes NaN

2.1. Doublons / Formatage / Valeurs aberrantes

Contexte métier : \sum macro-constituants > 100g

alcohol_100g

proteins_100g

carbohydrates_100g

fat_100g

salt_100g (+ cendres)

eau

→ corrections des valeurs pour cas où on dépasse légèrement 100g

→ NaN pour les autres

Contexte métier : sous-constituant > macro-constituants

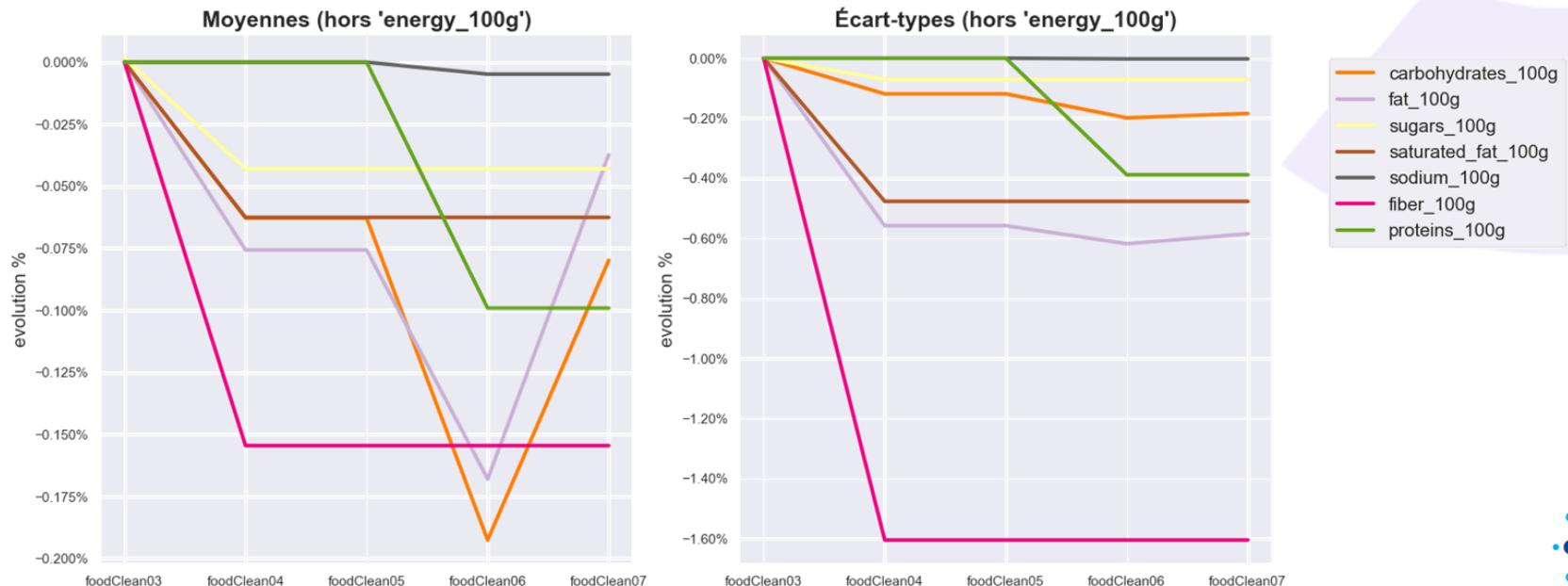
- sugars_100g > carbohydrates_100g
- saturated_fat_100g > fat_100g

→ NaN

2.1. Doublons / Formatage / Valeurs aberrantes

Traitement valeurs aberrantes : impact sur moyenne et écart-type

Traitements successifs des outliers - Évolution de la moyenne et de l'écart type



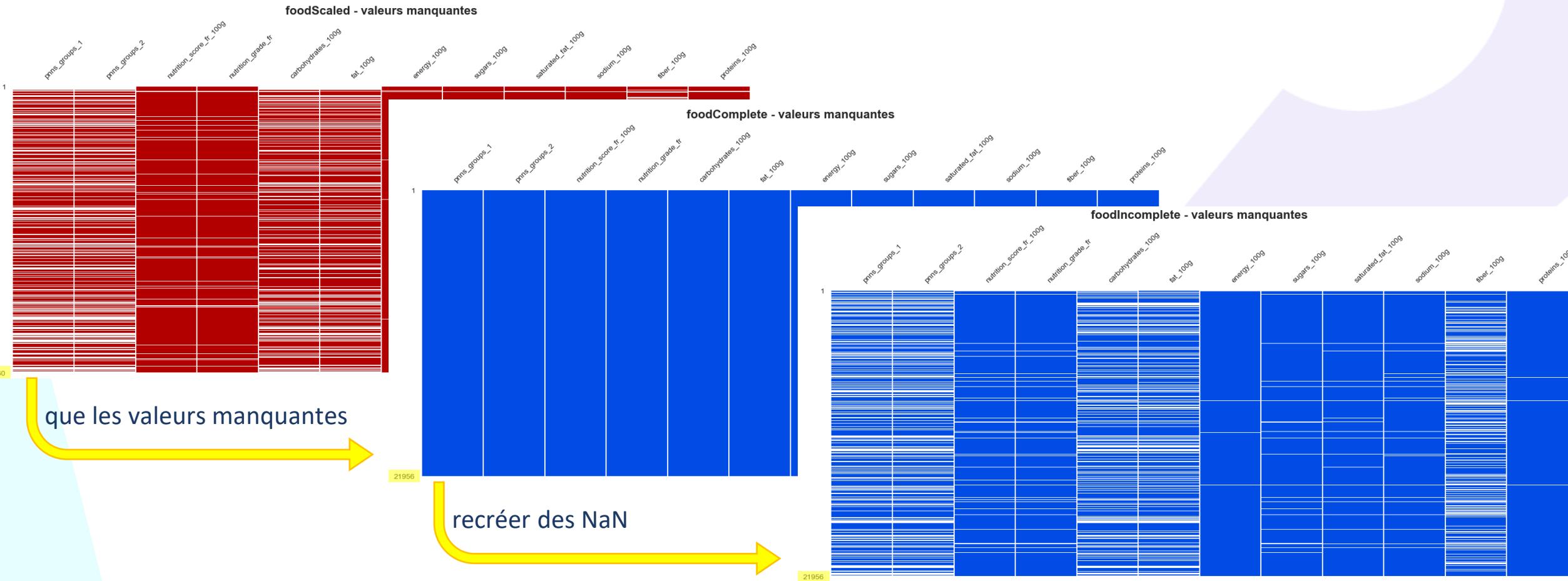
2.2. Imputation des valeurs manquantes

Différentes étapes :

- Suppression des produits trop mal renseignés (toutes les masses inconnues)
- Uniformiser les échelles de 0 à 100
- Tester différents algorithmes à notre disposition :
 - Variables numériques :
 - KNN imputer
 - Iterative Imputer
 - Variables catégorielles :
 - Créer un algorithme
- ...
- ... sur un jeu de données test
- Déterminer la méthode ou le paramètre le plus performant grâce à ces valeurs manquantes créées artificiellement

2.2. Imputation des valeurs manquantes

Jeu de données test

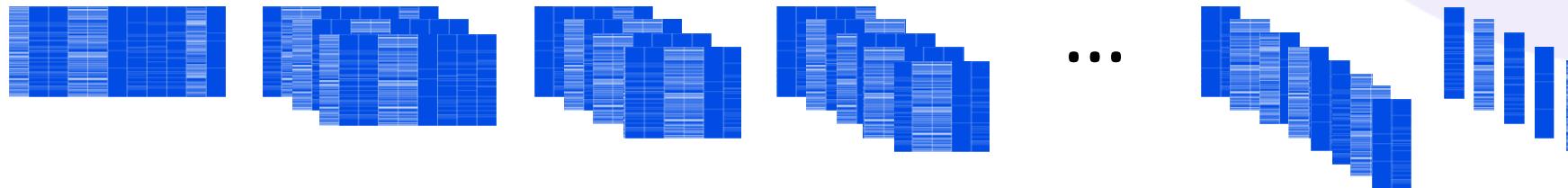


2.2. Imputation des valeurs manquantes

Imputation des colonnes catégorielles

Création d'un imputeur basé sur l'algorithme de classification kNN :

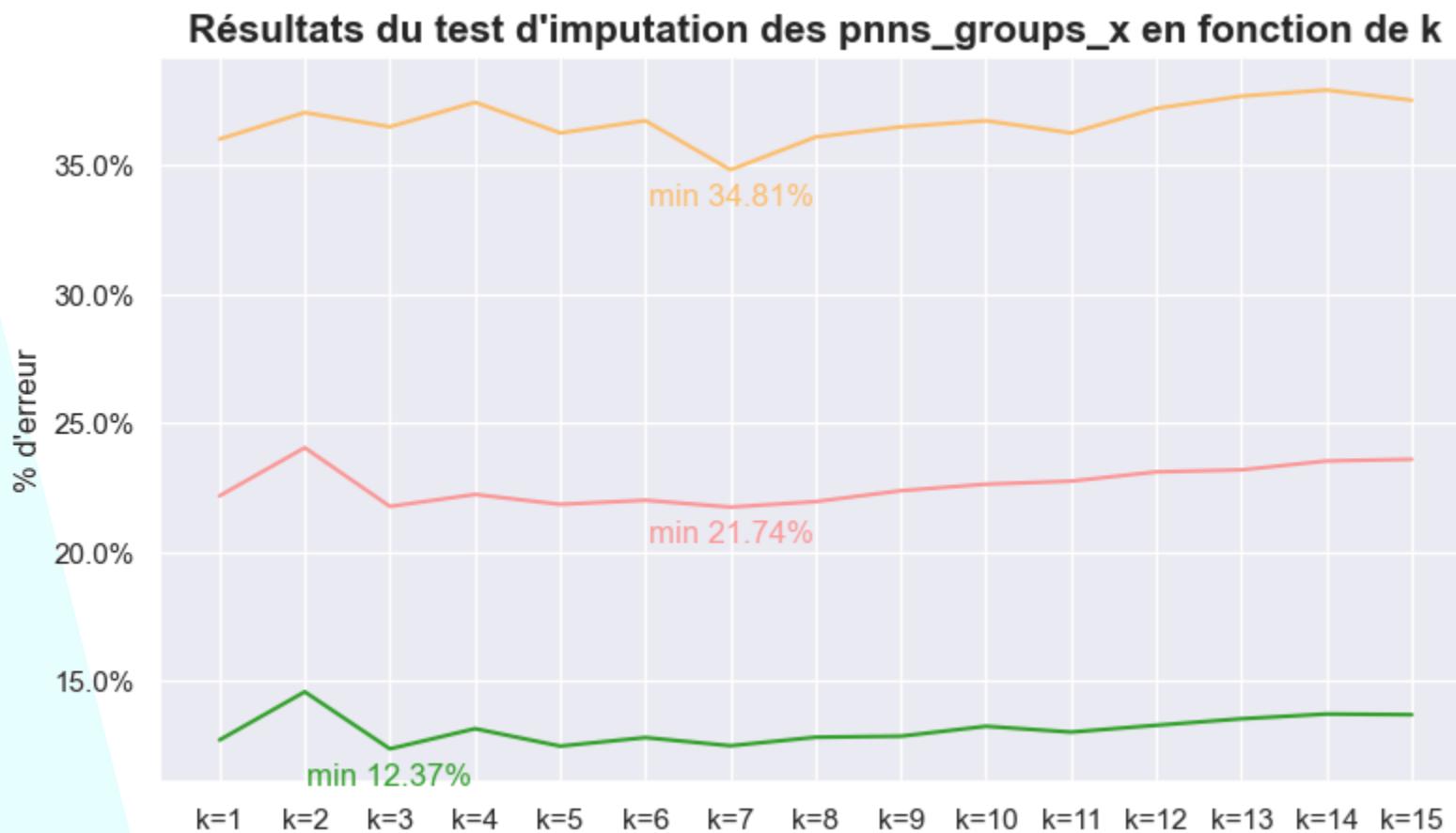
1. Séparer tableau en 2 :
 - A : là où l'on connaît la catégorie (pour entraîner le kNN),
 - B : là où l'on doit imputer
2. itérer sur toutes les **combinaisons** possibles de colonnes numériques



3. filtrer A et B : où toutes les valeurs numériques de la **combinaison** sont **NON vides**
4. si A filtré a suffisamment de lignes, **on entraîne un kNN et on l'utilise sur B filtré pour imputer**, sinon, on impute avec le mode de la colonne catégorielle
5. On **enregistre les valeurs imputées**, on **supprime les lignes** en question de A et B, et on **passe à la combinaison suivante**

2.2. Imputation des valeurs manquantes

Imputation des colonnes catégorielles



pas très précis sur
nutrition_grade_fr

Solution : filtrage préalable
sur pnns_groups_1

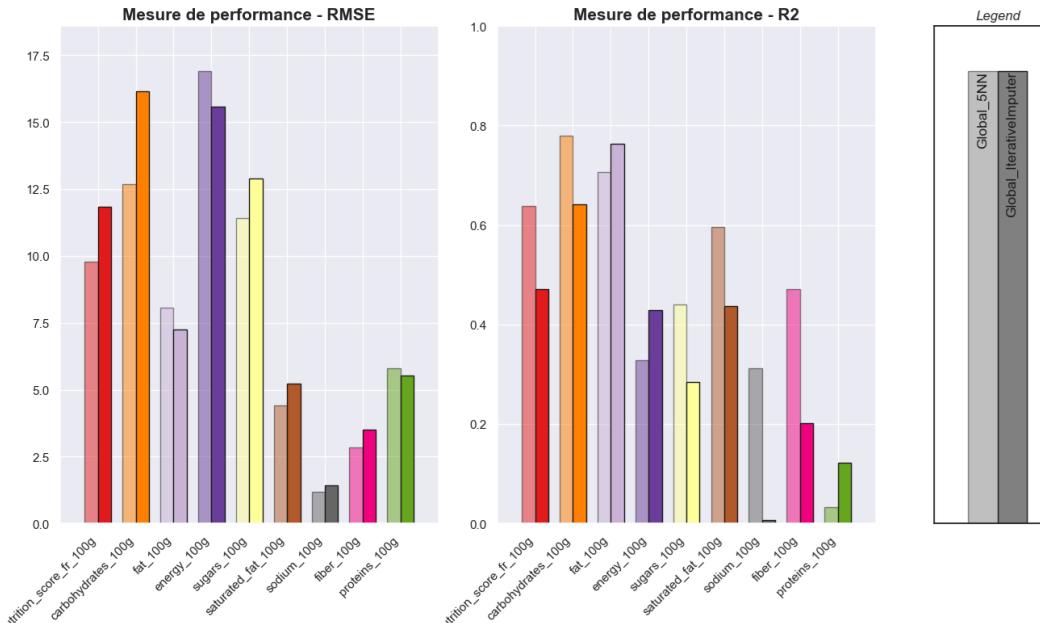
→ 34,81 27,44%

2.2. Imputation des valeurs manquantes

Imputation des colonnes numériques

1. Test du **kNN Imputer** avec **k=5**, sur **toutes les variables numériques simultanément**
2. Tentative d'optimisation de **k** pour chaque colonne → temps de calcul **trop long**
3. Test de l'**Iterative Imputer** sur toutes les variables numériques simultanément → beaucoup plus rapide, mais **performance moins bonne**

Résultats des tests d'imputation avec chaque méthode



2.2. Imputation des valeurs manquantes

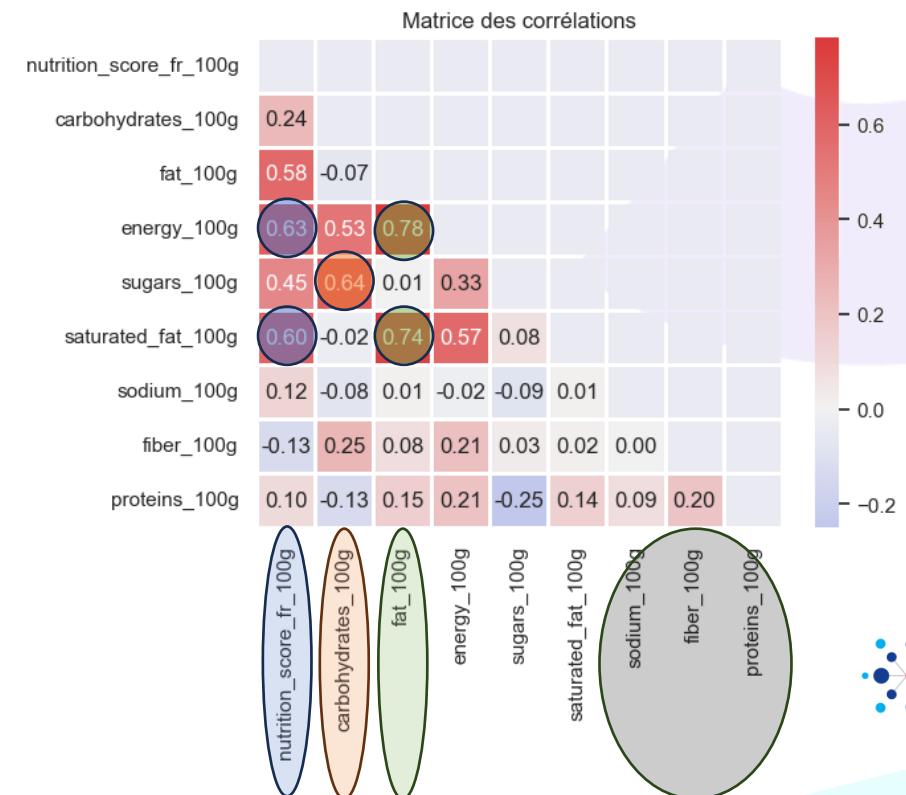
Imputation des colonnes numériques

4. Améliorer nos imputations :

- a) Étudier les corrélations
- b) Créer des groupes d'imputation

```
[{'G1': ['nutrition_score_fr_100g', 'energy_100g', 'saturated_fat_100g'],
 'G2': ['carbohydrates_100g', 'sugars_100g'],
 'G3': ['fat_100g', 'energy_100g', 'saturated_fat_100g'],
 'others': ['sodium_100g', 'fiber_100g', 'proteins_100g']}
```

- c) Le faire pour chaque catégorie de pnns_groups_1
- d) Pour les Gx : IterativeImputer
- e) Pour les others : kNN Imputer avec k=4

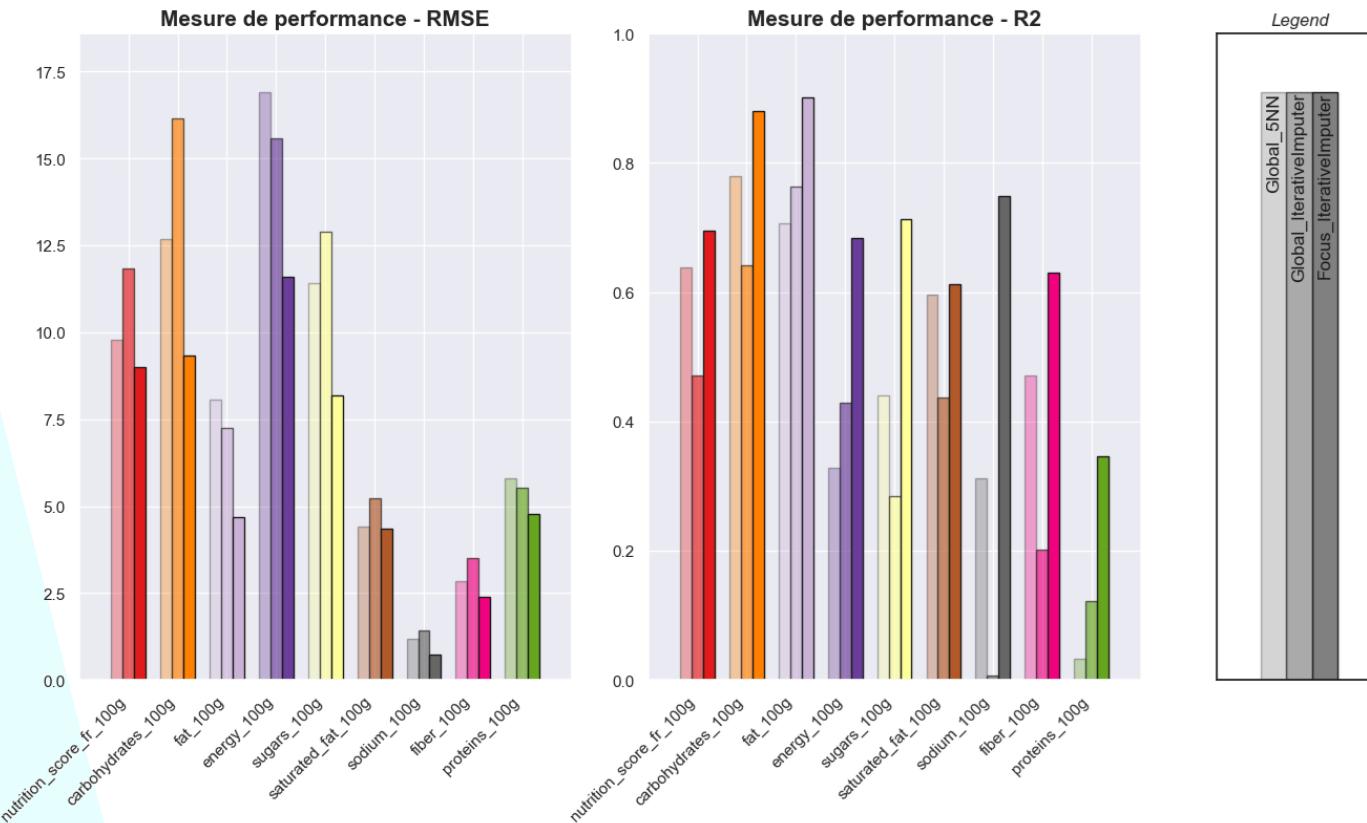


2.2. Imputation des valeurs manquantes

Imputation des colonnes numériques

4. Améliorer nos imputations :

Résultats des tests d'imputation avec chaque méthode

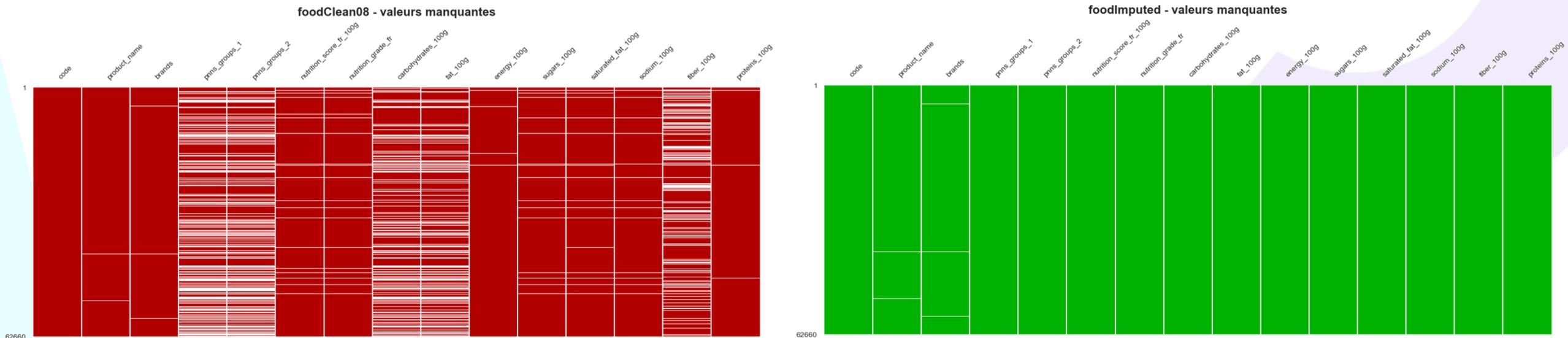


Par souci de cohérence : imputations <0 ou >100 passées à 0 et 100.

Puis application de cette méthode, plus performante en précision et temps de calcul, sur notre jeu de données.

2.2. Imputation des valeurs manquantes

Imputation des valeurs manquantes - Résultats



PARTIE 3 - EXPLORATION



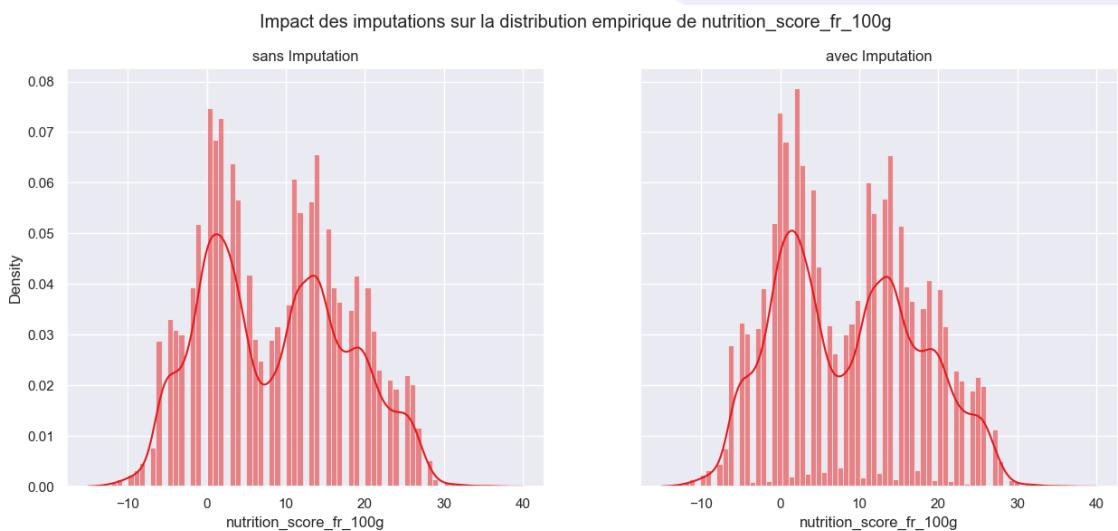
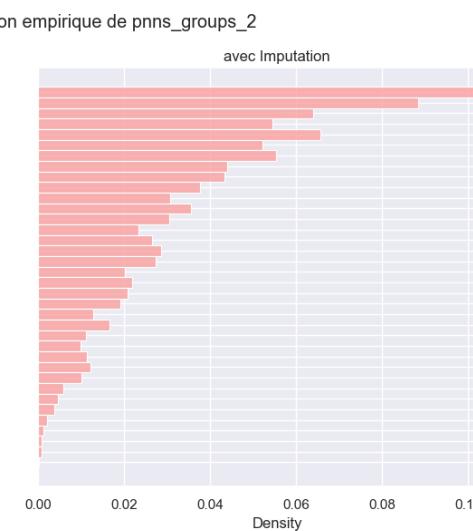
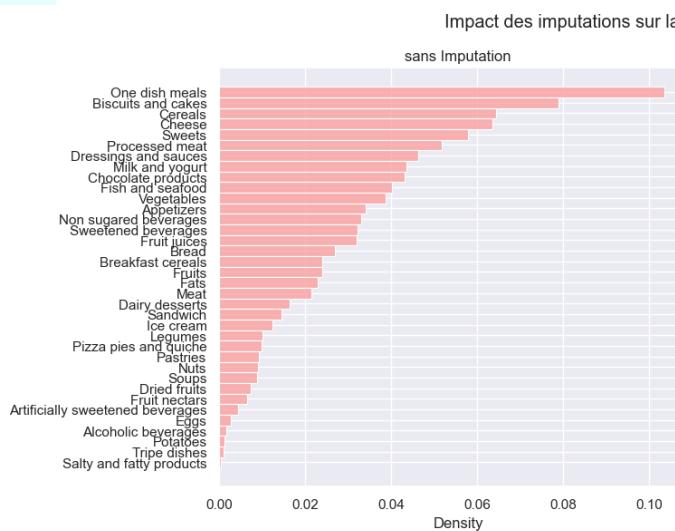
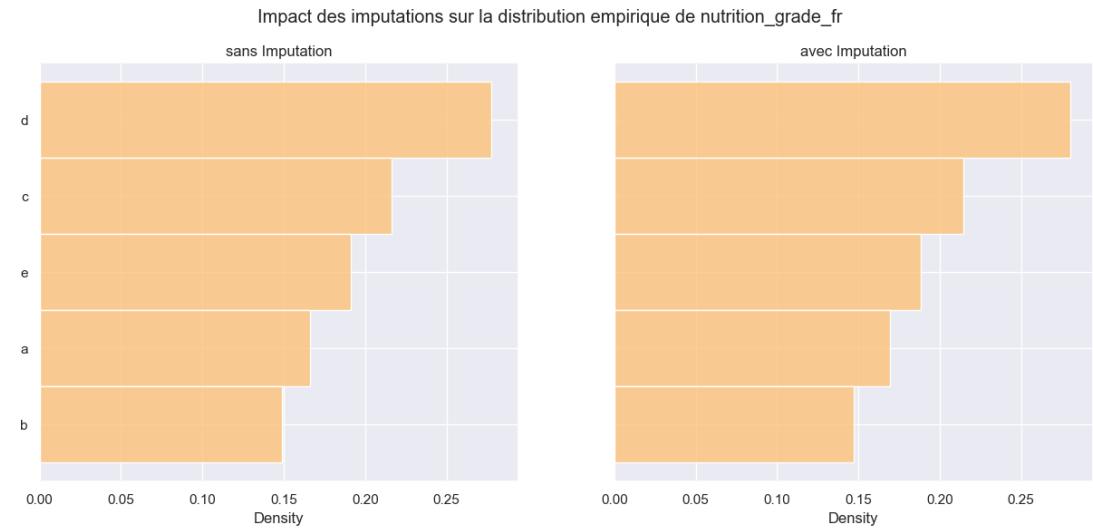
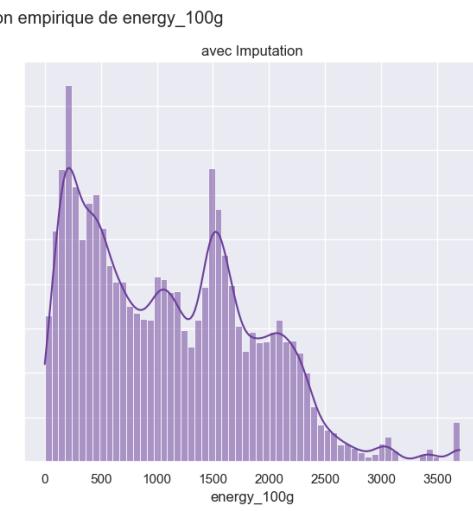
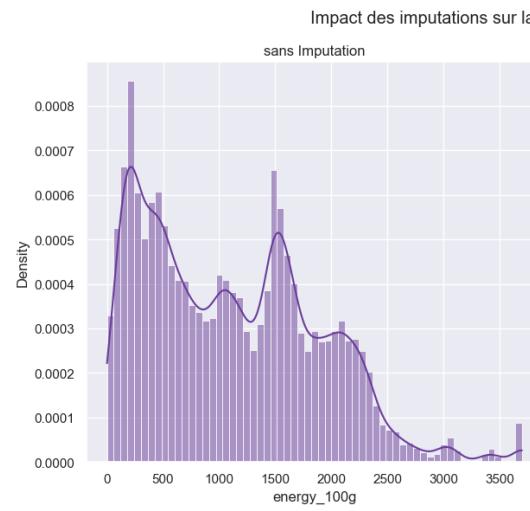
3.1. Analyses univariées

	nutrition_score_fr_100g	carbohydrates_100g	fat_100g	energy_100g	sugars_100g	saturated_fat_100g	sodium_100g	fiber_100g	proteins_100g
count	62660.000000	62660.000000	62660.000000	62660.000000	62660.000000	62660.000000	62660.000000	62660.000000	62660.000000
mean	8.662014	28.502025	13.438469	1113.166833	13.206042	5.440283	0.457700	2.093254	7.769236
std	8.950269	27.465244	16.769059	767.156301	18.779868	8.443086	1.653692	4.116904	7.834819
min	-15.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	1.000000	4.448136	1.550000	433.000000	1.000000	0.300000	0.031496	0.000000	1.900000
50%	8.000000	15.586146	7.000000	1039.000000	4.000000	2.000000	0.222441	0.775000	6.000000
75%	15.000000	55.000000	21.000000	1644.000000	17.000000	7.500000	0.492126	2.600000	10.900000
max	40.000000	100.000000	100.000000	3700.000000	100.000000	100.000000	83.000000	100.000000	100.000000

Pas de valeurs aberrantes

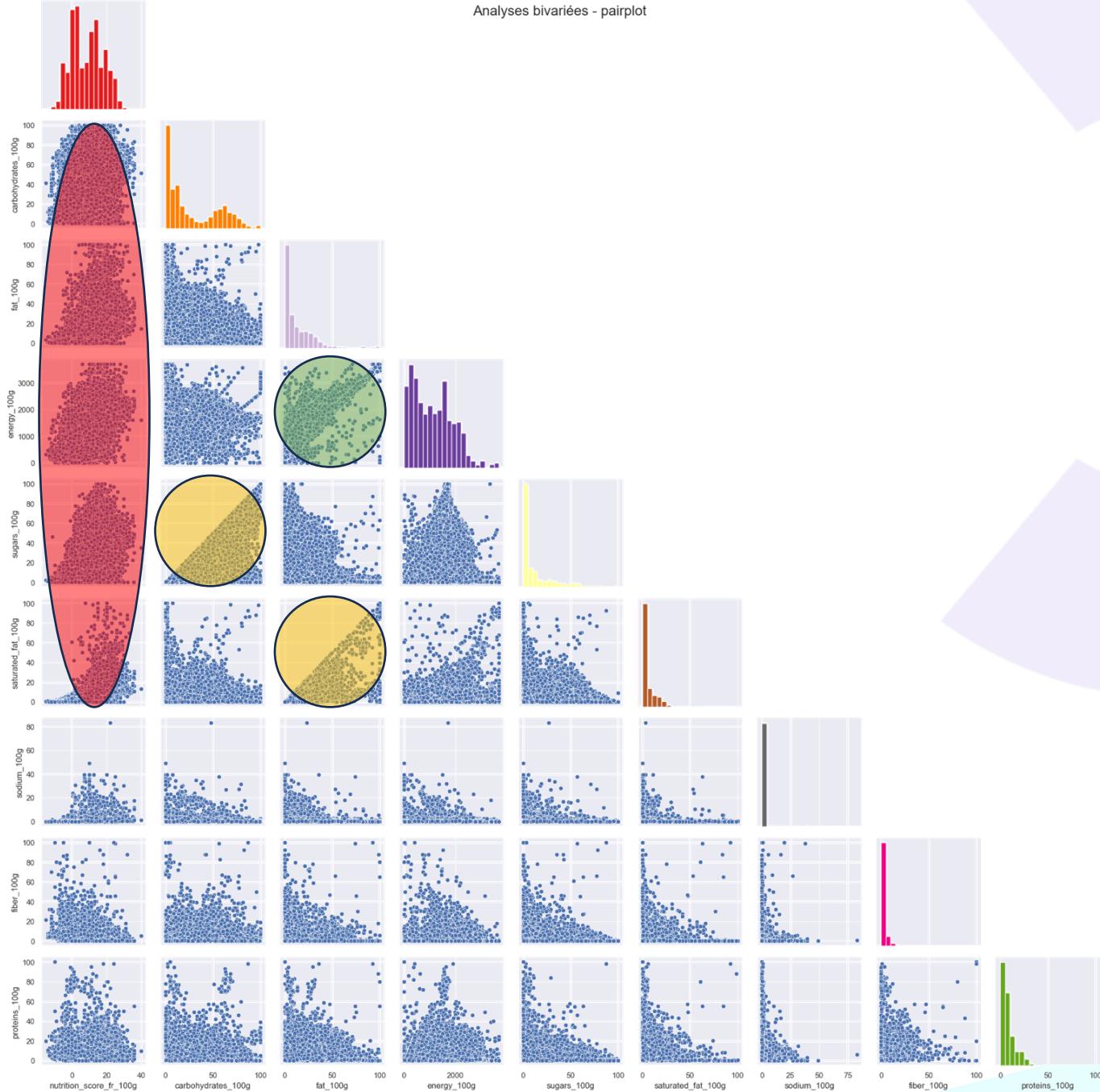
Max, min, écarts types, médianes : semblent cohérents

3.1. Analyses univariées



3.2. Analyses bivariées

Analyses bivariées - pairplot



3.2. Analyses bivariées – One-way ANOVA

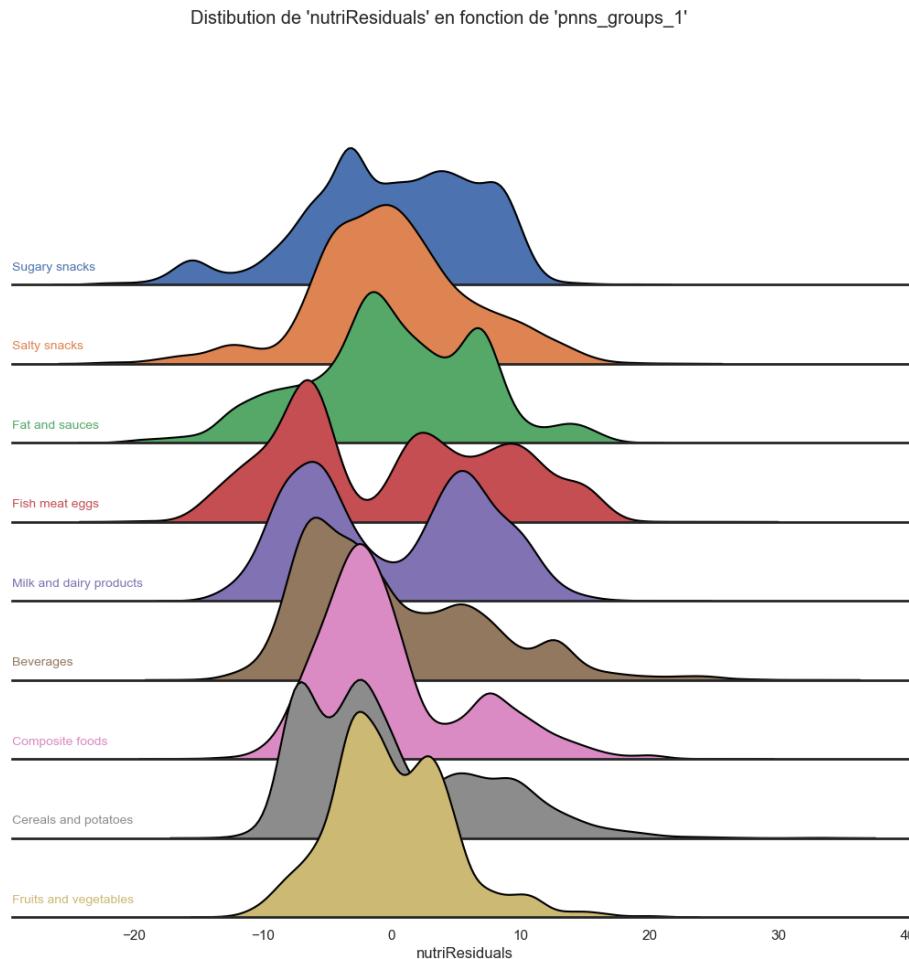
Dans le cadre de l'appli, opportun d'utiliser `pnns_groups_1` pour mieux prédire `nutrition_score_fr_100g` ?

→ Test statistique ANOVA (Analysis of Variance) :

- Hypothèse à tester :
 - H_0 : distributions de `nutrition_score_fr_100g` pour **chaque catégorie** de `pnns_groups_1` **similaires**
→ i.e. les moyennes sont toutes identiques
 - H_1 : **au moins** une distribution **est différente**
→ i.e. au moins une moyenne se distingue
- Postulats, présupposés :
 - a) Distributions des résiduels (écart autour des moyennes de chaque catégorie) sont **normales**
 - b) Homogénéité des **variances**
 - c) Données **indépendantes**

3.2. Analyses bivariées – One-way ANOVA

Vérification des postulats : distributions suivent la loi normale ?



Non,

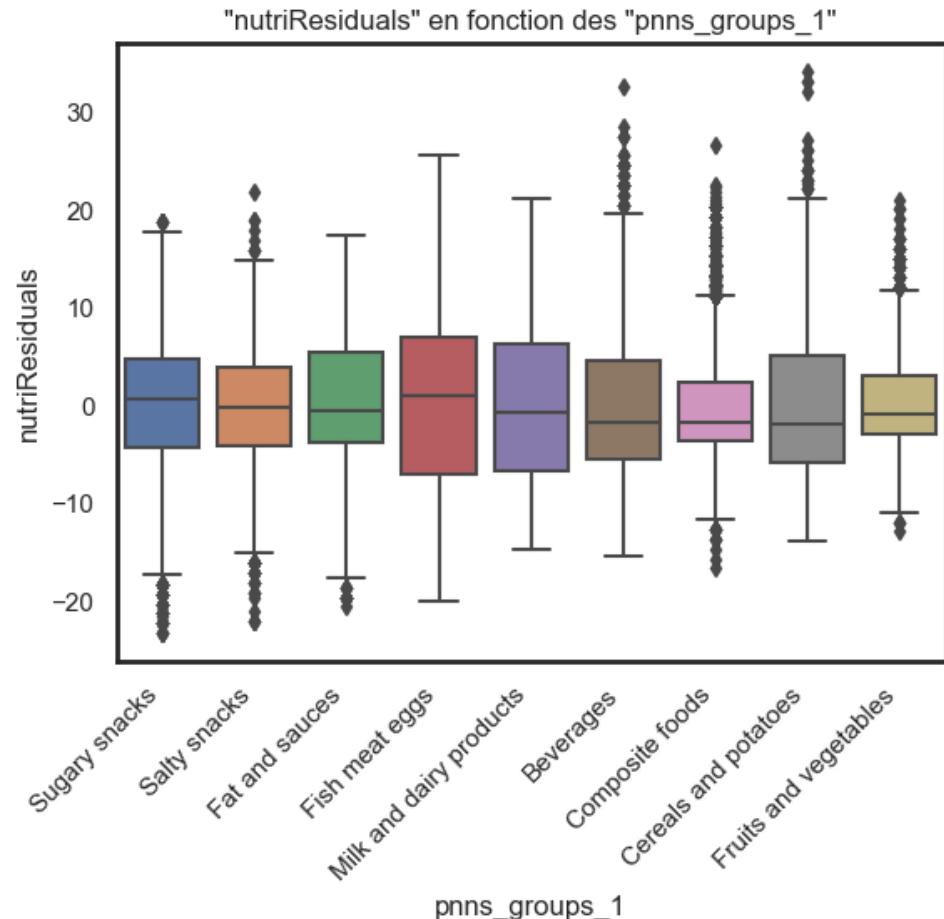
Mais grand nombre d'individus (de produits)



Violation critère normalité ne remet pas en cause la validité du test

3.2. Analyses bivariées – One-way ANOVA

Vérification des postulats : homogénéité des variances



Non,

Solution pour remédier à cette violation du postulat d'homogénéité des variances



Choisir des échantillons de mêmes tailles

3.2. Analyses bivariées – One-way ANOVA

Interprétation des résultats

	Sum_of_Squares	Degrees_Freedom	Mean_Square	F_statistic	Critical_F	F_test_p_value	Eta_Square
Model	8.474331e+05	8.0	105929.139486	2413.397366	1.938771	0.0	0.428086
Residual_Error	1.132153e+06	25794.0	43.892125	NaN	NaN	NaN	NaN
Total	1.979587e+06	25802.0	NaN	NaN	NaN	NaN	NaN

P-value = 0.0 < 0.05 (notre niveau de signification)

Statistique F > sa valeur critique

→ on rejette l'hypothèse nulle selon laquelle il n'y avait pas de différence de moyennes

Nota : échantillons très grands → on avait de grandes chances de trouver des différences

Taille de l'effet : $\eta^2 = 0,43$ → réduction variation de nutrition_score_fr_100g de 43% lorsqu'on prend en compte pnns_groups_1

3.3. Analyses multivariées – ACP

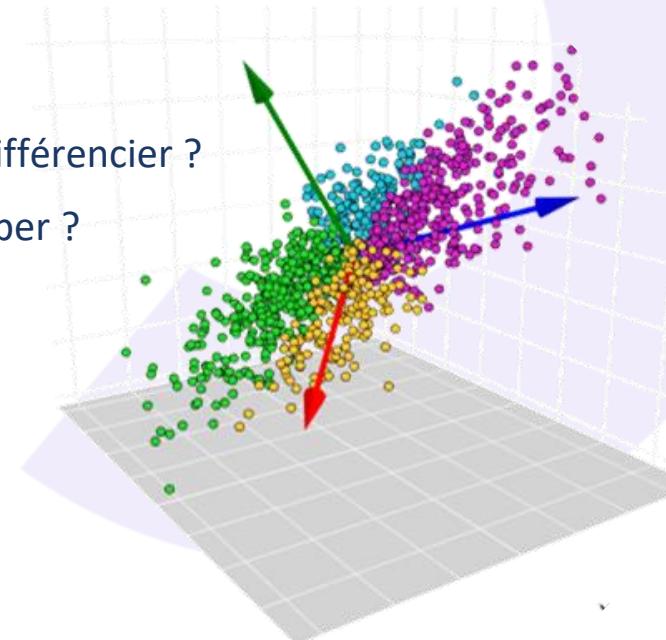
Analyse en Composantes Principales

1. Variabilité entre nos différents produits ? Quelles tendances peuvent nous permettre de les différencier ?
2. Relations entre les variables ? Quelles variables très corrélées entre elles ? Peut-on les regrouper ?

Objectif final pour développement appli

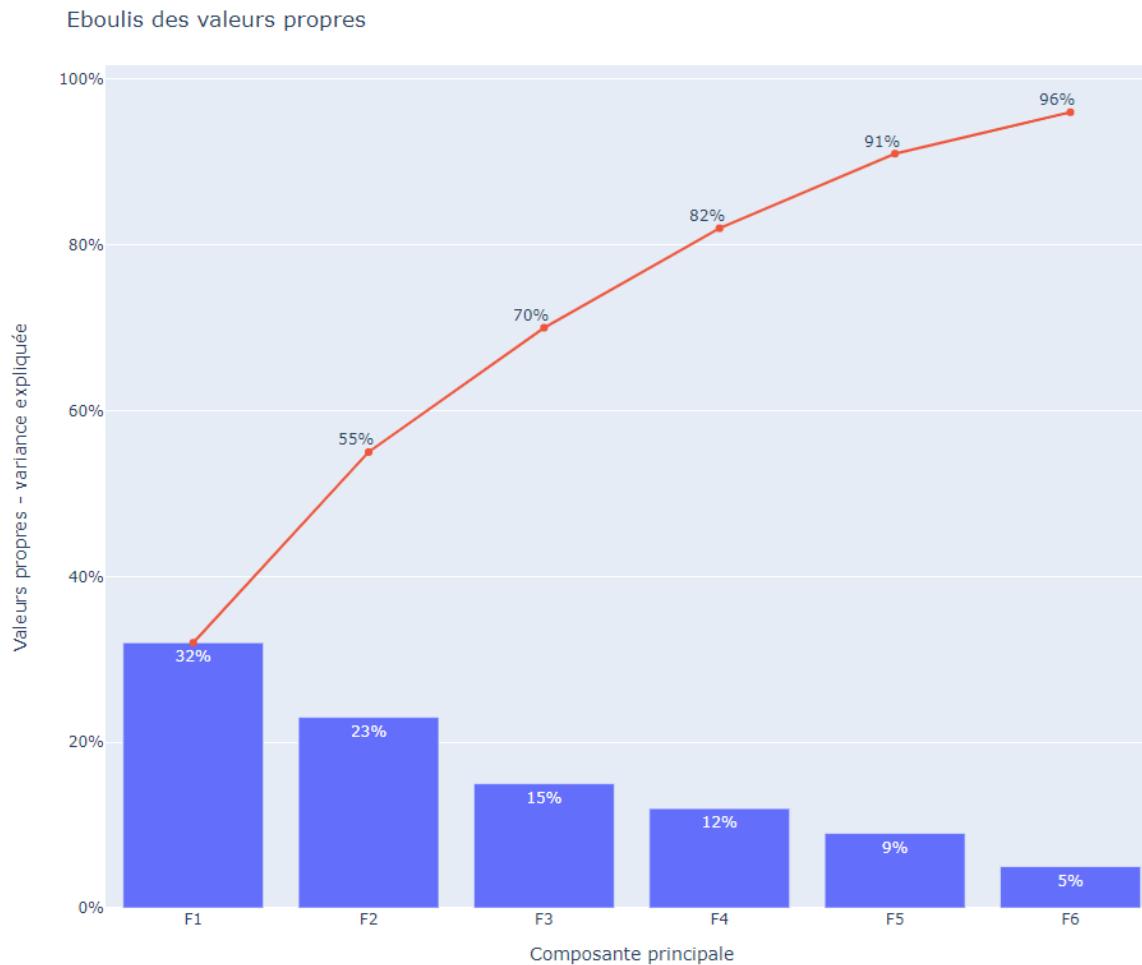
Regrouper les variables :

- Multicolinéarité ↴
- Temps de calcul ↴
- Overfitting ↴



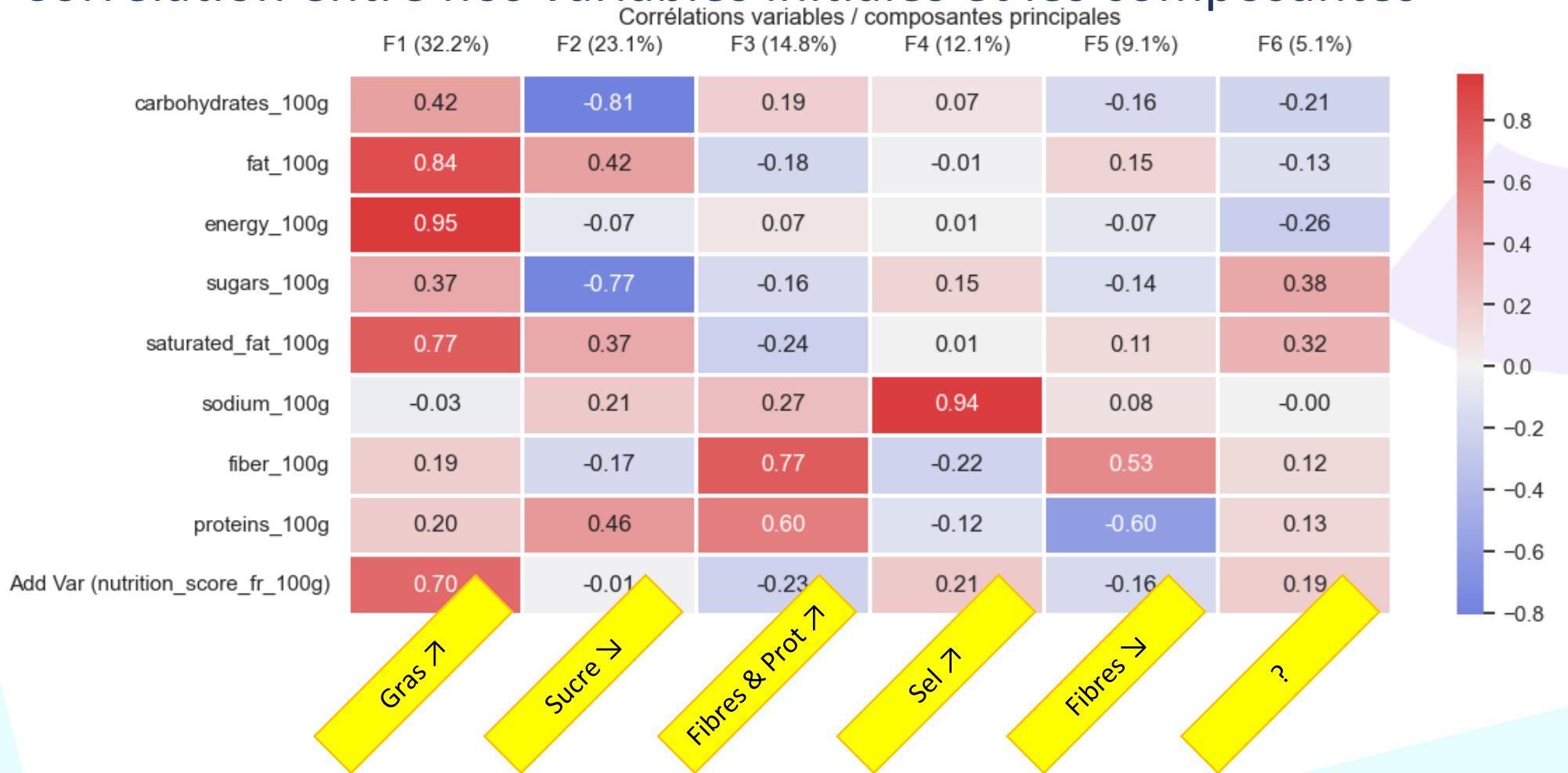
3.3. Analyses multivariées – ACP

Part dispersion expliquée par la réduction



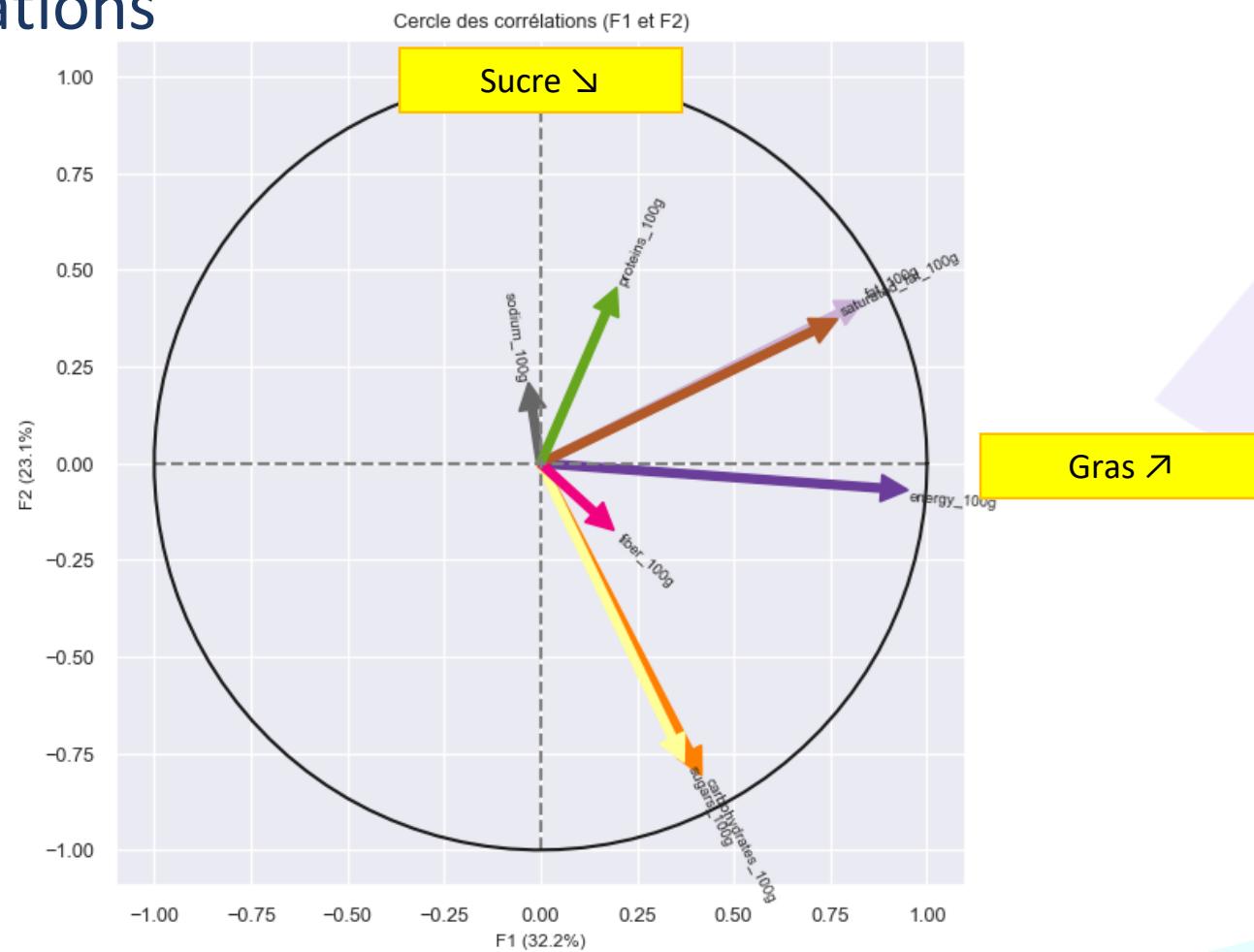
3.3. Analyses multivariées – ACP

Corrélation entre nos variables initiales et les composantes



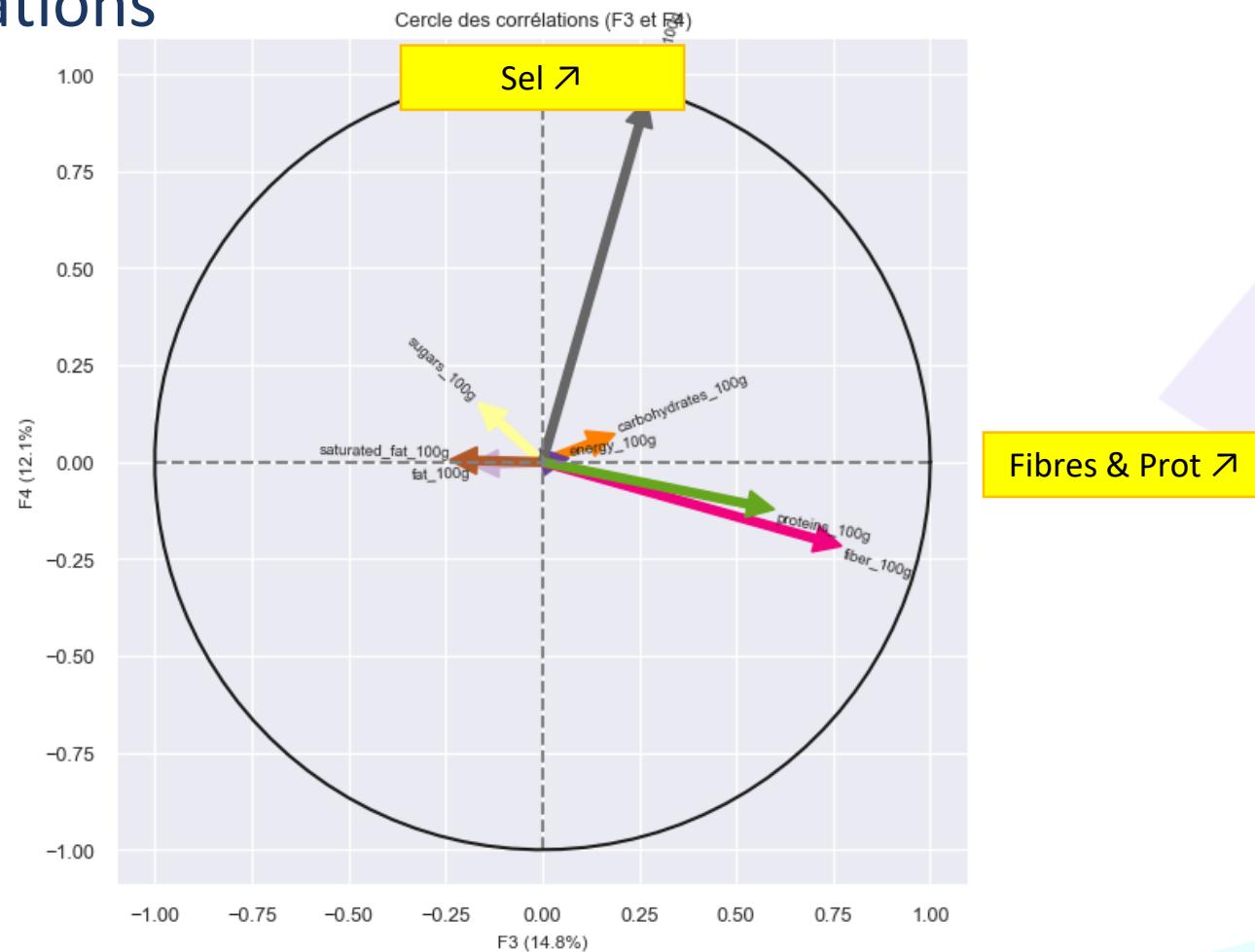
3.3. Analyses multivariées – ACP

Cercle des corrélations



3.3. Analyses multivariées – ACP

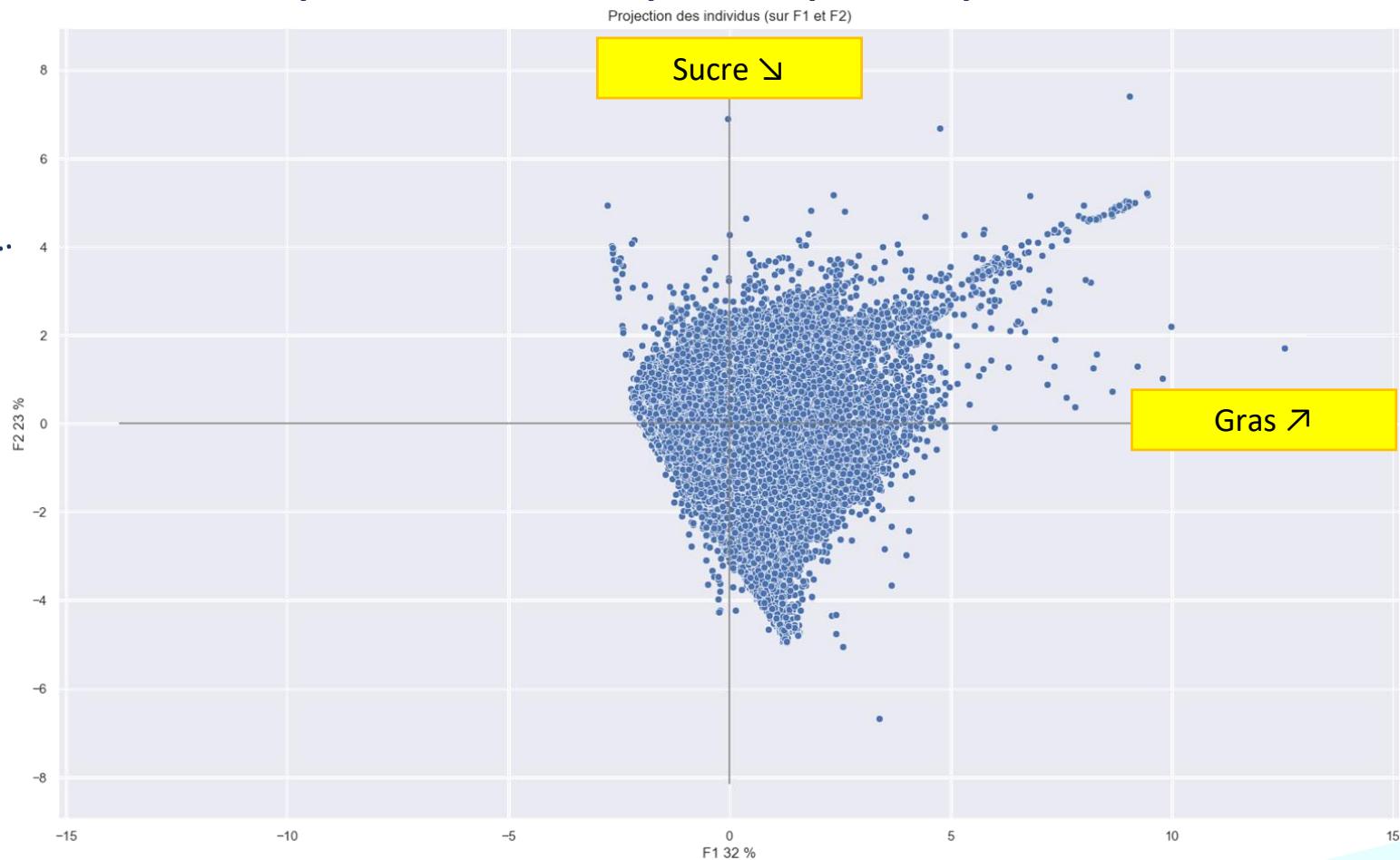
Cercle des corrélations



3.3. Analyses multivariées – ACP

Représentation des produits sur plans principaux

Pas facile à interpréter ...



3.3. Analyses multivariées – ACP

Représentation des produits sur plans principaux

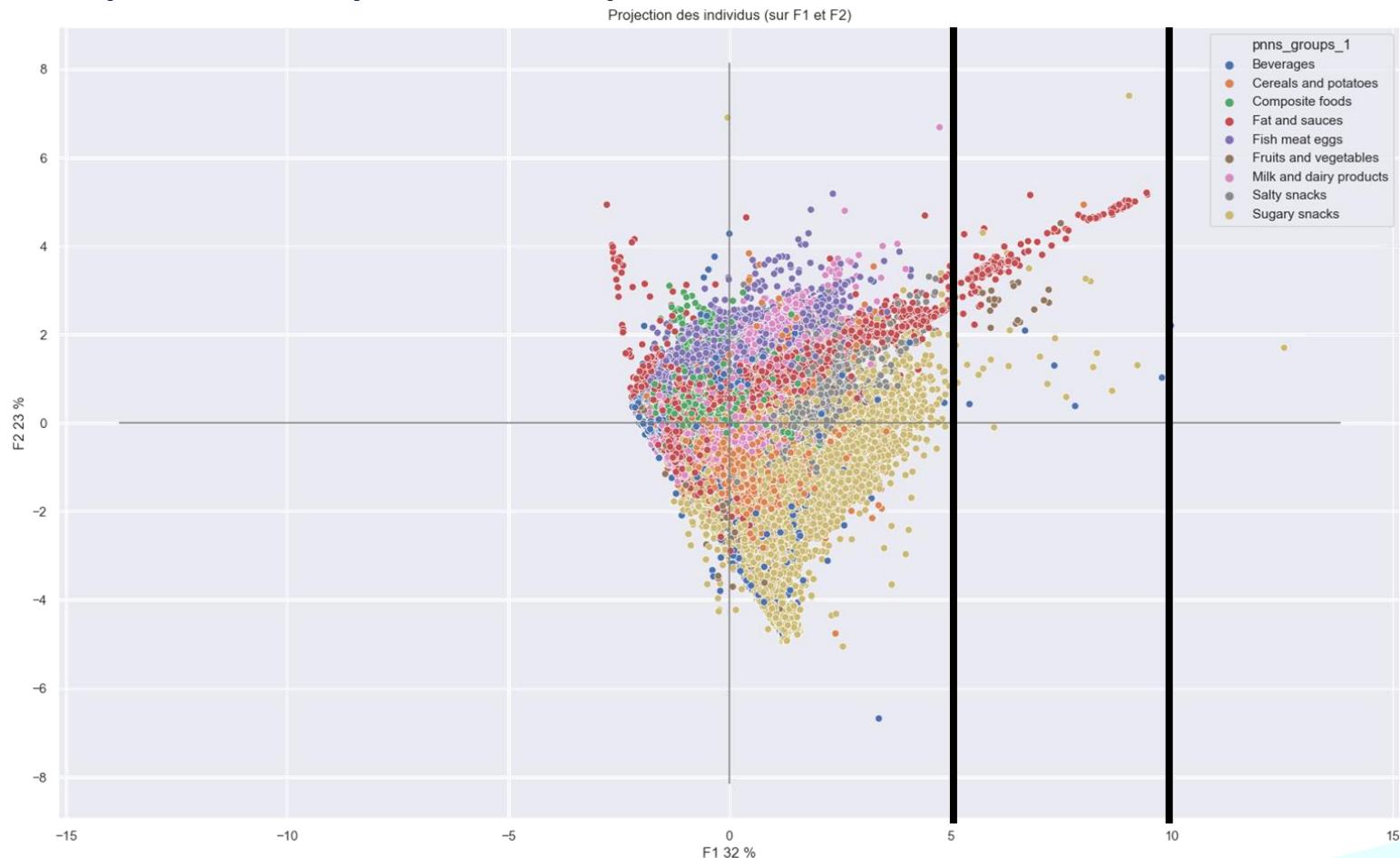


CONCLUSION



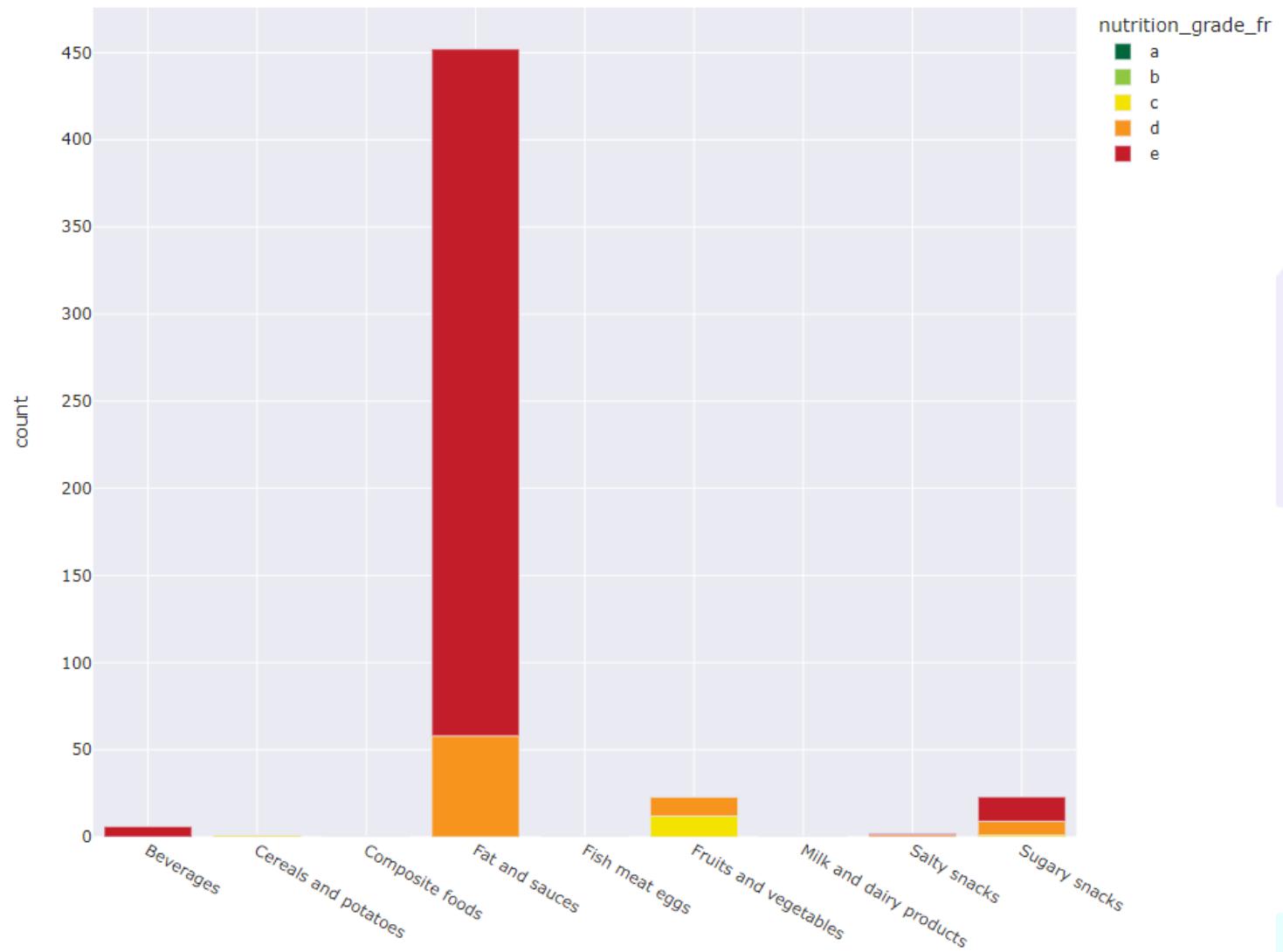
Améliorer la lisibilité et l'interprétabilité

Regarder les produits présents pour certaines valeurs de F1, F2, F3, etc.



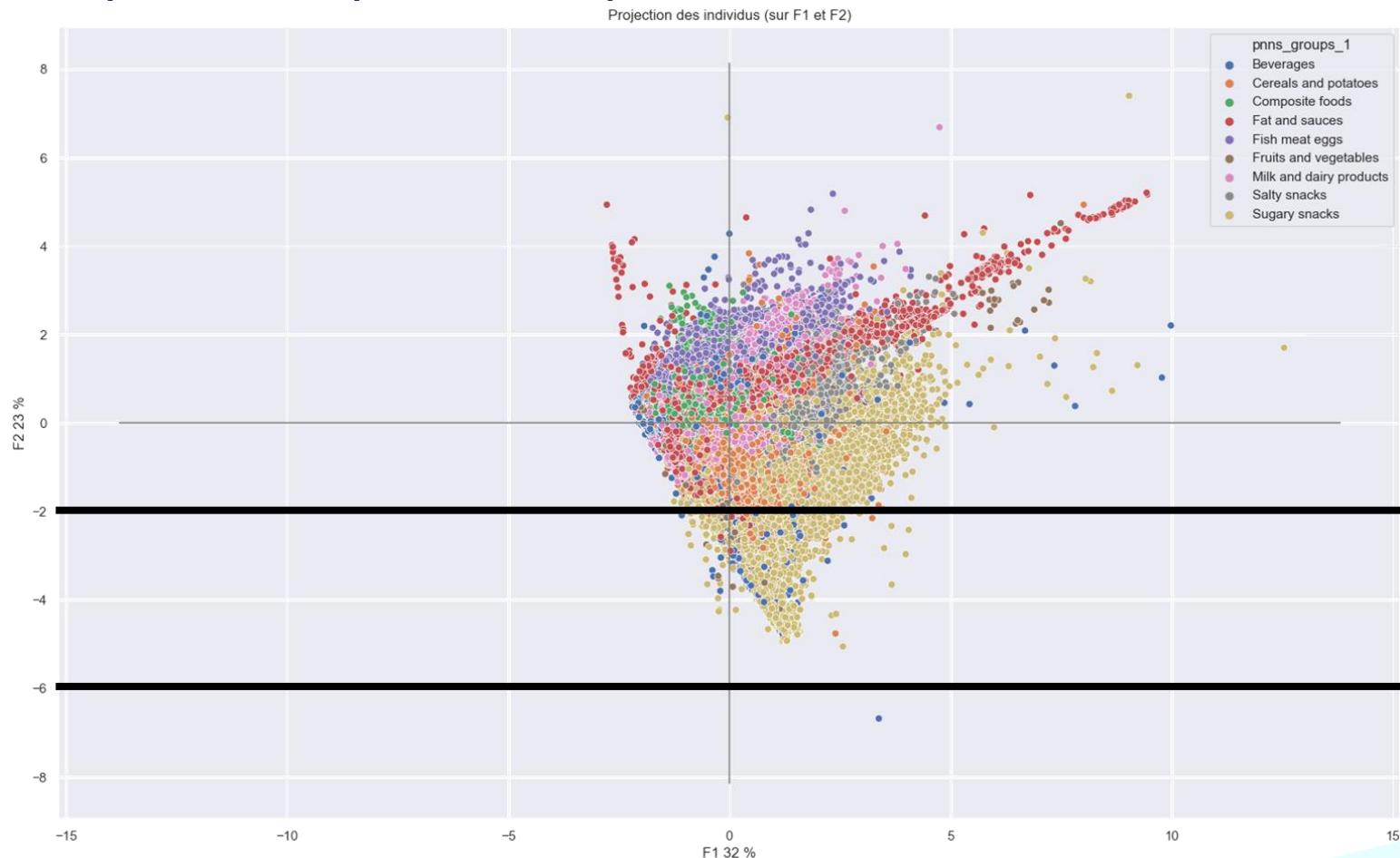
Améliorer la lisibilité et l'interprétabilité

Distribution de pnns_groups_1 pour F1 compris entre 5.0 et 10.0



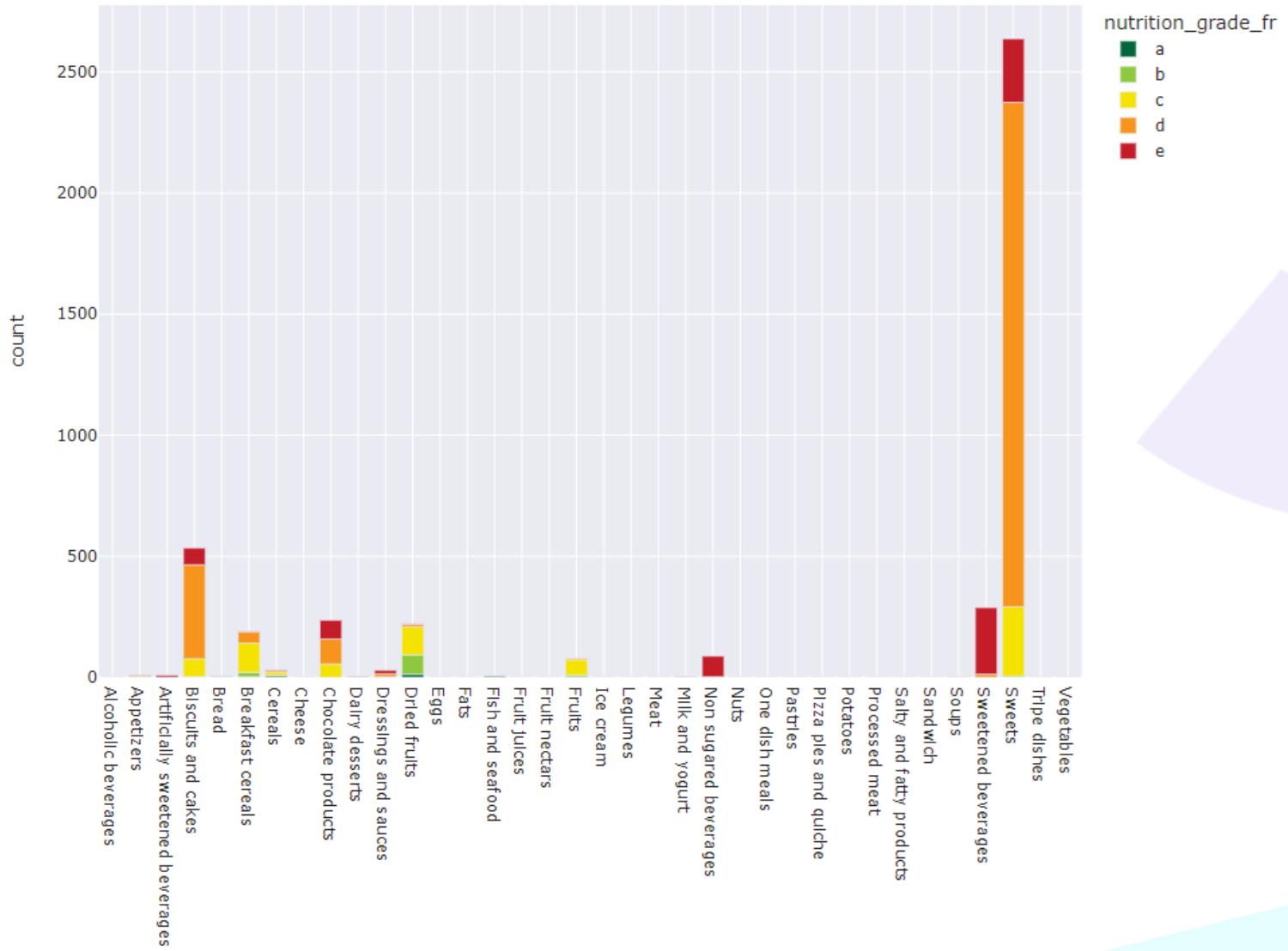
Améliorer la lisibilité et l'interprétabilité

Regarder les produits présents pour certaines valeurs de F1, F2, etc.



Améliorer la lisibilité et l'interprétabilité

Distribution de pnns_groups_2 pour F2 compris entre -6.0 et -2.0



Une meilleure vision de vos données ?

Nous espérons que cette analyse permettra **une meilleure compréhension** de vos données.

Si l'on revient sur l'exemple d'exploitation de vos données qu'est le développement d'une app, on a montré que :

- Score nutritionnel corrélé aux catégories
- Il était possible de réduire le nombre de colonnes grâce à l'ACP



Optimisation du futur algorithme de prédiction du score nutritionnel associé à l'App