Concevez une application au service de la santé publique

PLAN

1/ Présentation de la problématique
2/ Présentation du projet d'application
3/ Présentation du jeu de données
4/ Nettoyage du jeu de données et des données manquantes
5/ Exploration et analyse du jeu de données
6/ Statistiques
7/ Conclusion
8/ Suite du projet

Santé publique France

Appel à projets de ***Santé Publique France***

Trouver des idées innovantes d'applications en lien avec **l'alimentation**

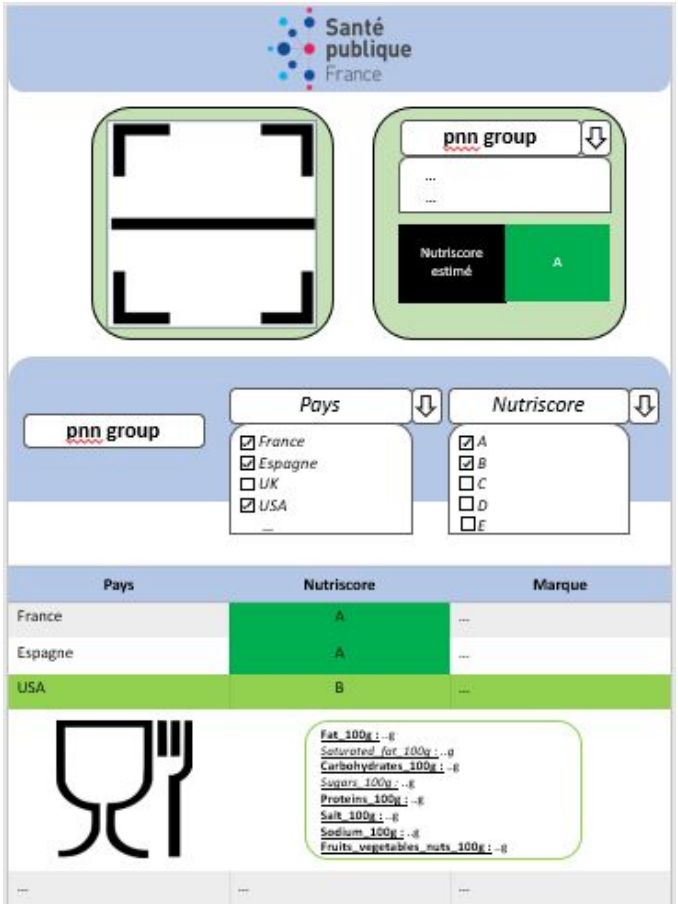
Objectifs :

Participer à l'appel
Proposer une solution d'application

Mission :

Démontrer la faisabilité de l'application à l'aide des données à disposition

- Scanner les valeurs nutritionnelles d'un produit et choisir le groupe (pnn) du produit
- Estimation du nutriscore
- Liste de produits du même groupe (pnn) avec filtres (pays et nutriscore)



Fichier CSV très volumineux (>6.0Go)

Lignes :

Produits enregistrés dans le jeu de données

Colonnes :

Colonnes textuelles (code produit, pays, catégorie, nutriscore, nutrigrade etc.)
Colonnes numériques (valeurs des indicateurs nutritionnelles, nutrition score etc.)

Beaucoup de données manquantes

- **<u>Première étape</u>**

  Conserver les colonnes avec taux de valeurs manquantes inférieures à un seuil (70%) : faciliter le chargement du jeu de données

  Calculer du taux de données manquantes des colonnes du jeu de données en plusieurs paquets : résumé dans un tableau avec en index les colonnes et chaque colonne le taux pour chaque paquet

  Calculer la moyenne de tous les paquets

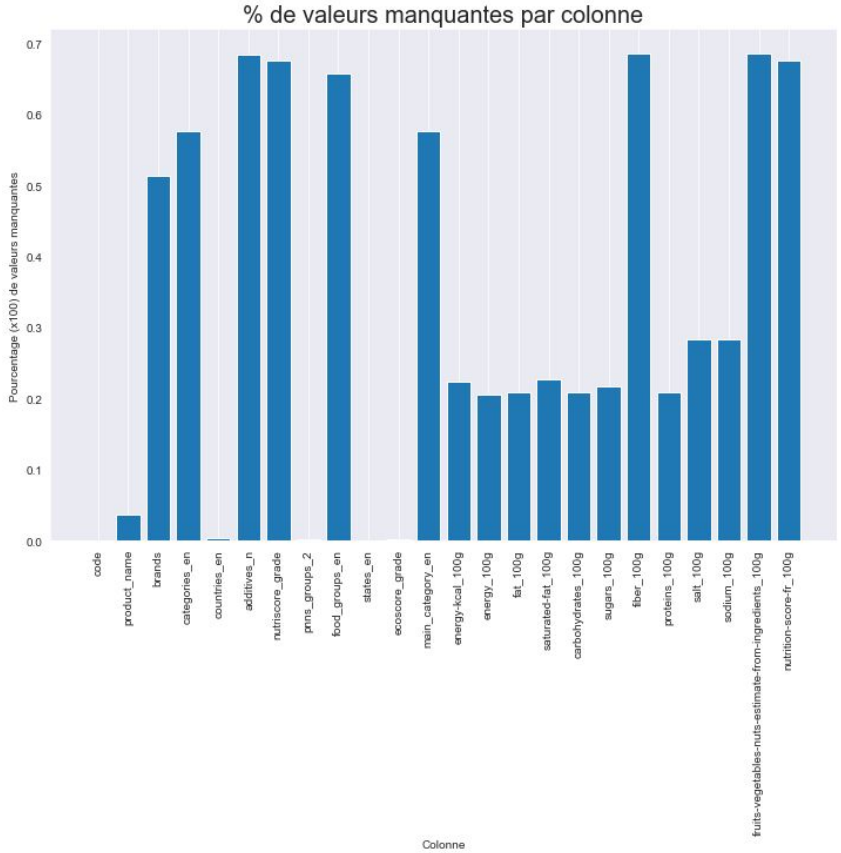
  Filtrer les index avec le seuil

Suppression des colonnes identiques (conservation de la mieux remplie)

24 colonnes restantes

```
Index(['code', 'url', 'creator', 'created_t', 'created_datetime',
       'last_modified_t', 'last_modified_datetime', 'product_name', 'brands',
       'brands_tags', 'categories', 'categories_tags', 'categories_en',
       'countries', 'countries_tags', 'countries_en', 'ingredients_text',
       'ingredients_tags', 'additives_n', 'nutriscore_score',
       'nutriscore_grade', 'pnns_groups_1', 'pnns_groups_2', 'food_groups',
       'food_groups_tags', 'food_groups_en', 'states', 'states_tags',
       'states_en', 'ecoscore_grade', 'main_category', 'main_category_en',
       'image_url', 'image_small_url', 'image_ingredients_url',
       'image_ingredients_small_url', 'image_nutrition_url',
       'image_nutrition_small_url', 'energy-kcal_100g', 'energy_100g',
       'fat_100g', 'saturated-fat_100g', 'carbohydrates_100g', 'sugars_100g',
       'fiber_100g', 'proteins_100g', 'salt_100g', 'sodium_100g',
       'fruits-vegetables-nuts-estimate-from-ingredients_100g',
       'nutrition-score-fr_100g'],
      dtype='object')
```

Nettoyage du jeu de données et des données manquantes



Taux de valeurs manquantes pour chaque colonne

- **Deuxième partie :**

  Supprimer les lignes sans *'nutrition-score'*

  Attribuer 0 aux sous catégories si la catégorie a 0 et sous catégorie est manquante

  Supprimer les lignes où la somme valeurs nutritionnelles > 100g

  Supprimer les lignes sans catégorie de produits

  Supprimer les dupliqués (conserver le mieux rempli)

  Supprimer les lignes où valeurs nutritionnelles < 0g et > 100g

- **<u>Deuxième partie (suite) :</u>**

  Supprimer les lignes où sous catégorie > catégorie

  Supprimer les lignes où valeurs nutritionnelles < 0g et > P99

  Supprimer les lignes où *'energy-kcal_100g'* > 900

  Supprimer les lignes où *'nutrition-score'* < -15 et > 40

- **<u>Troisième partie :</u>**

  Imputer les valeurs nutritionnelles manquantes avec IterativeImputer

  Refaire le cycle de nettoyage de la deuxième partie

  Calculer *'energy_kcal_100g'* et convertir pour obtenir *'energy_kJ_100g'*

  Calculer le nutriscore à partir du *'nutrition_score'*

- Utilisation d'IterativeImputer : les valeurs nutritionnelles sont liées entre elles

Relation 'fat' - 'proteins'

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	8.7741	0.026	338.263	0.000	8.723	8.825
proteins_100g	0.5115	0.002	222.708	0.000	0.507	0.516

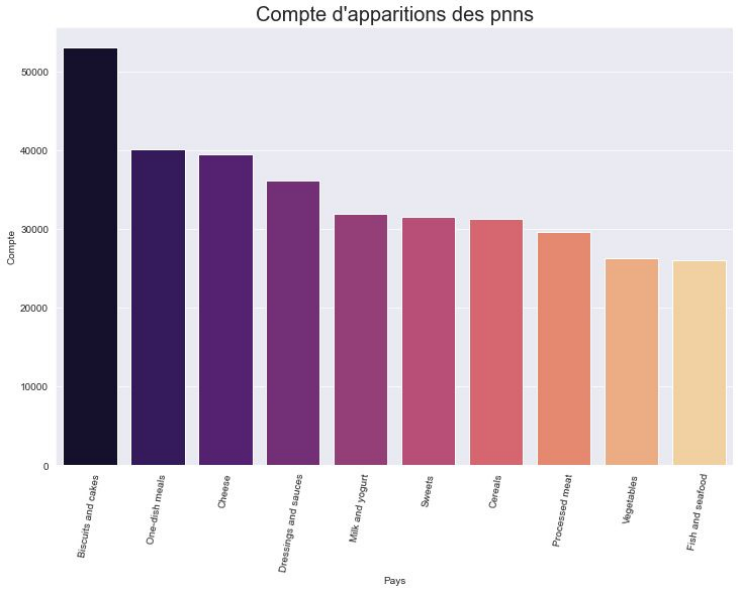
Relation 'carbohydrates' - 'fiber'

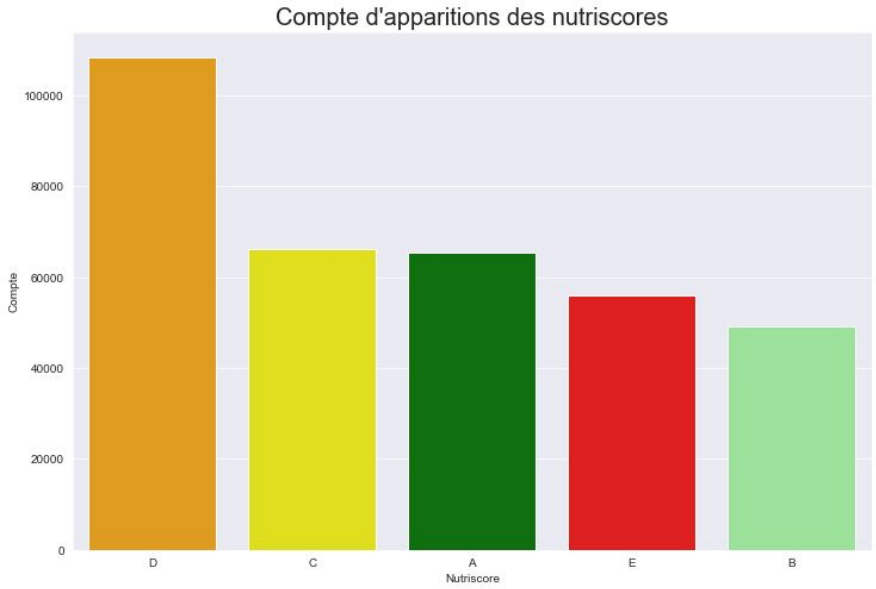
	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	21.2483	0.051	417.754	0.000	21.149	21.348
fiber_100g	3.6633	0.015	248.576	0.000	3.634	3.692

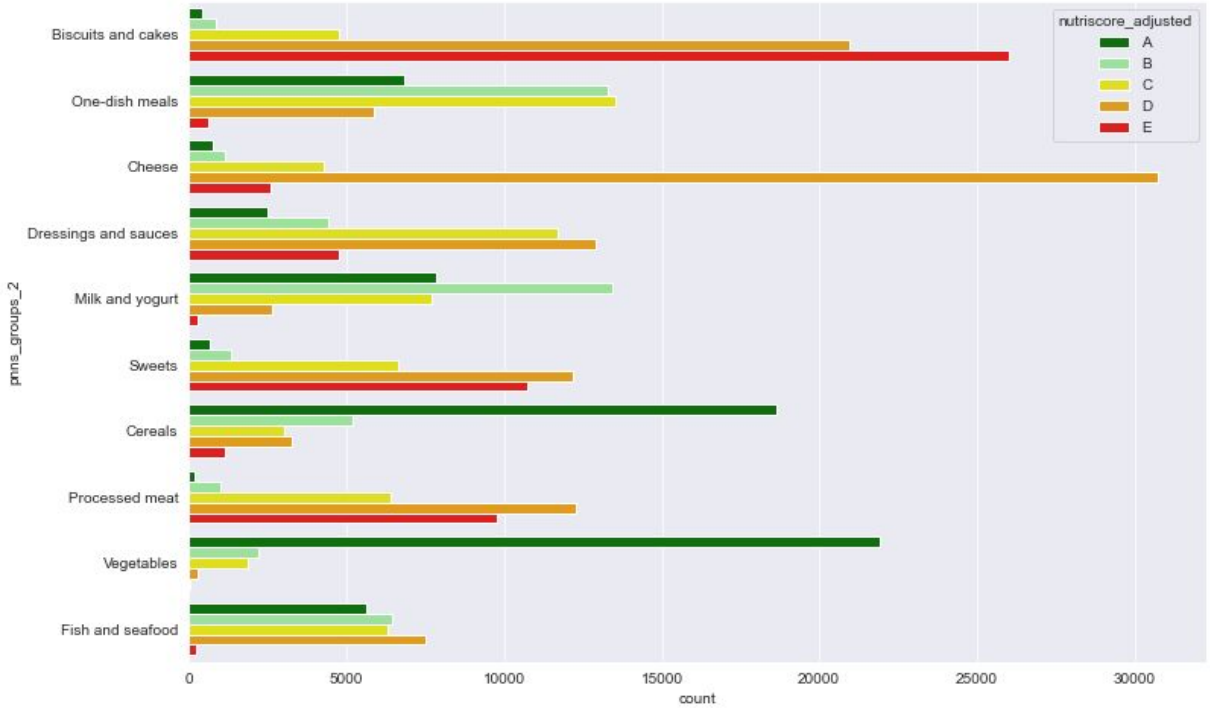
Relation 'sugar' - 'salt"

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	15.5712	0.025	632.678	0.000	15.523	15.619
salt_100g	-4.9473	0.019	-256.716	0.000	-4.985	-4.910

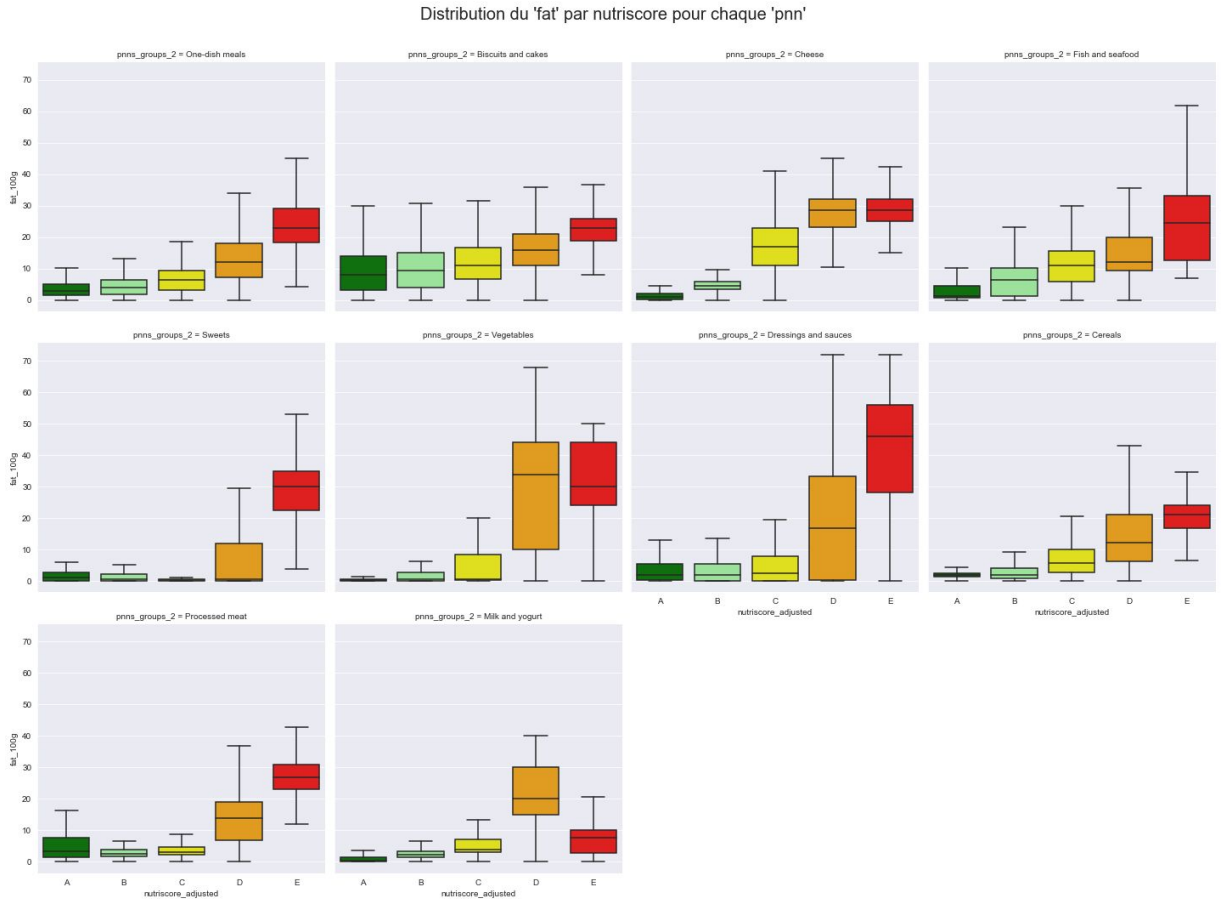
- Réalisons une exploration et analyse macroscopique sur les 10 groupes ‘pnn’ les plus représentés



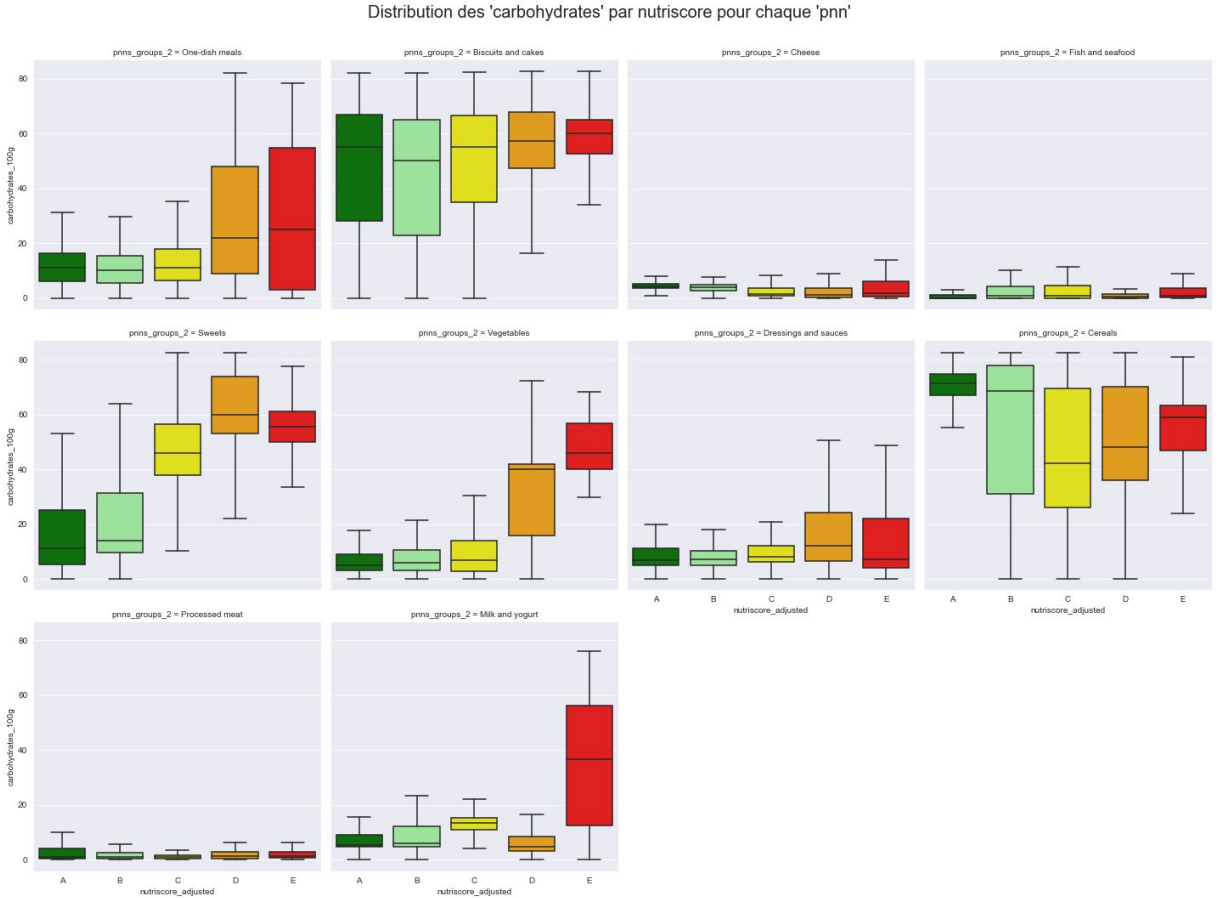



nutriscore_adjusted
A
B
C
D
E
Biscuits and cakes
One-dish meals
Cheese
Dressings and sauces
Milk and yogurt
Sweets
Cereals
Processed meat
Vegetables
Fish and seafood
pnns_groups_2
0
5000
10000
15000
20000
25000
30000
count

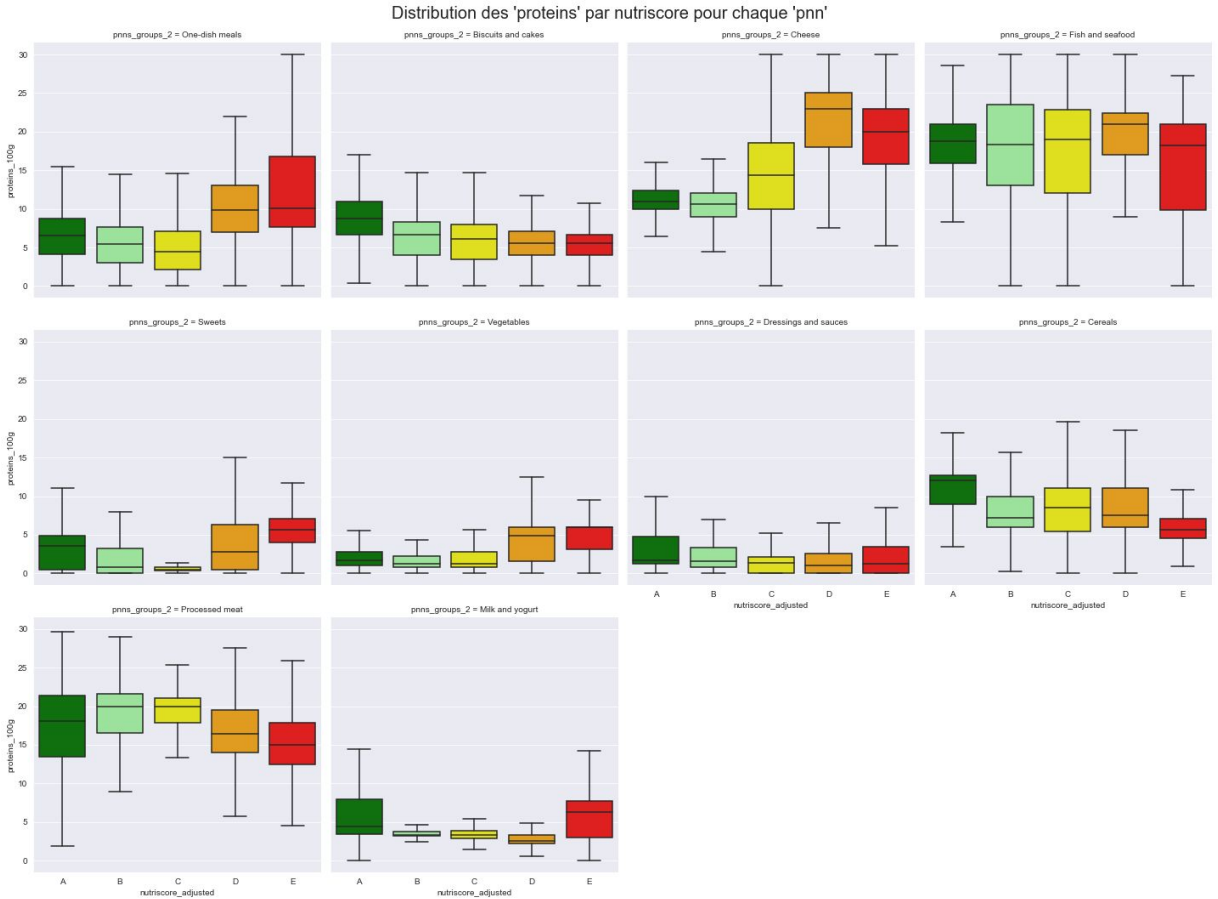
Exploration et analyse du jeu de données



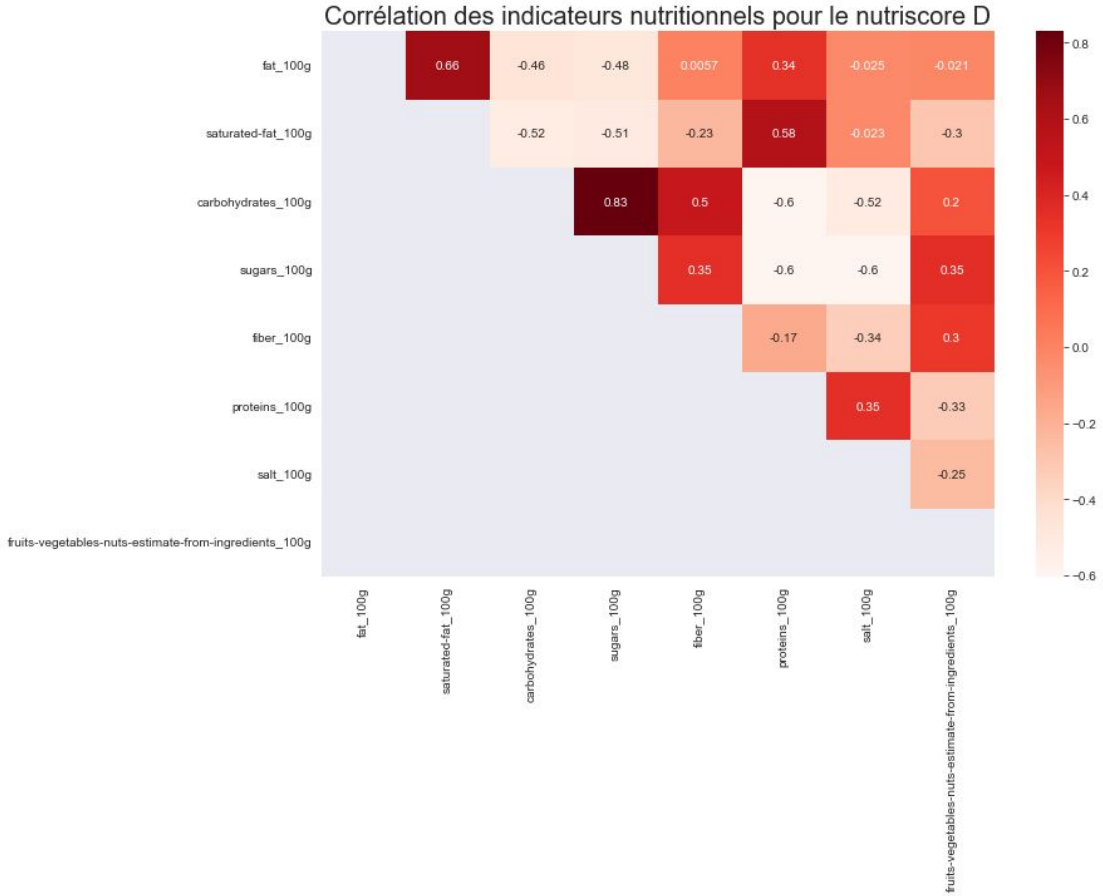
Exploration et analyse du jeu de données

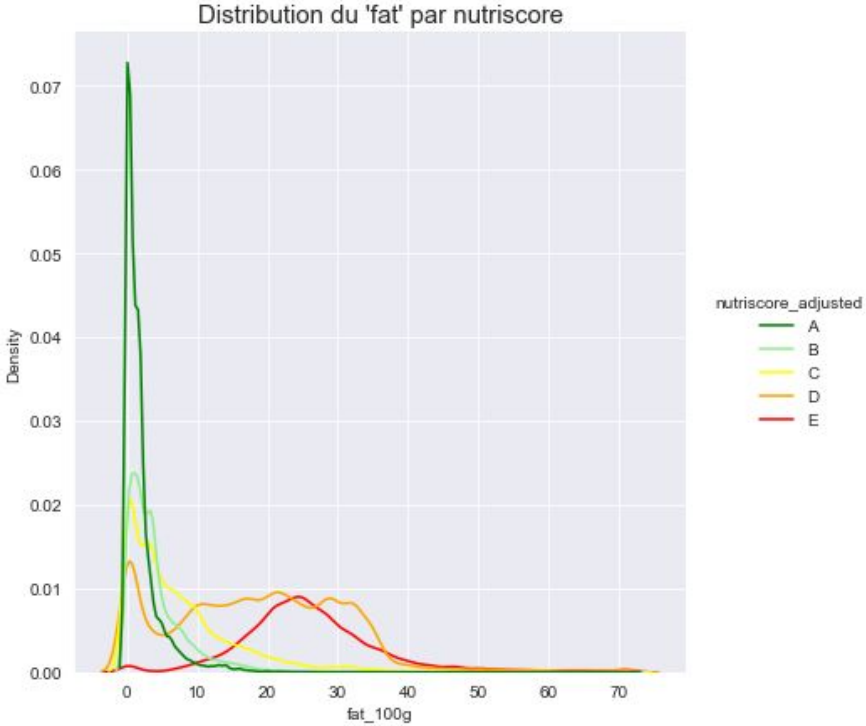


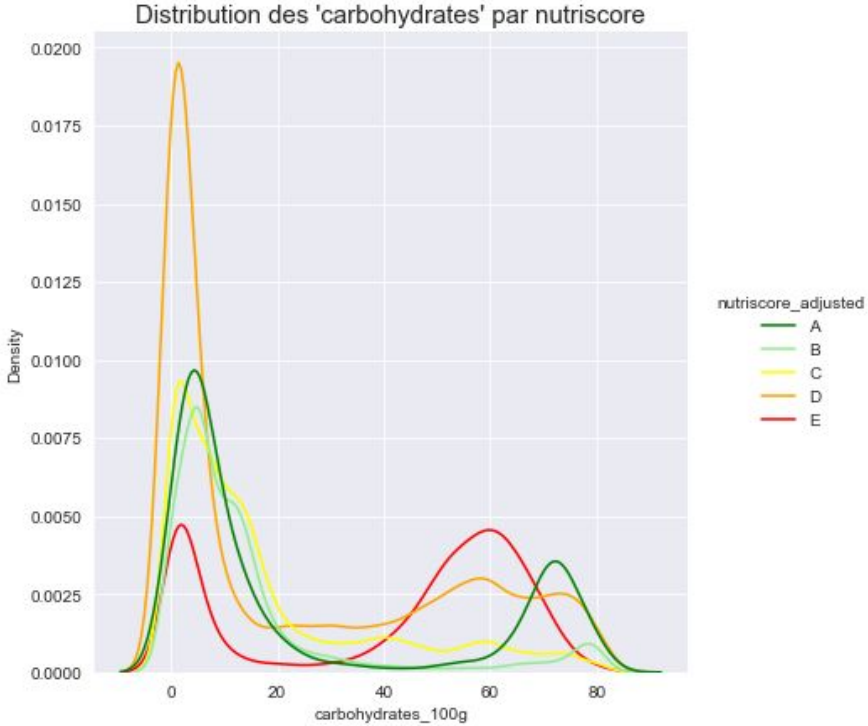
Exploration et analyse du jeu de données

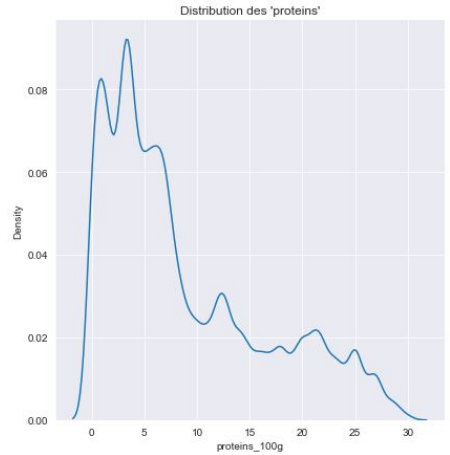


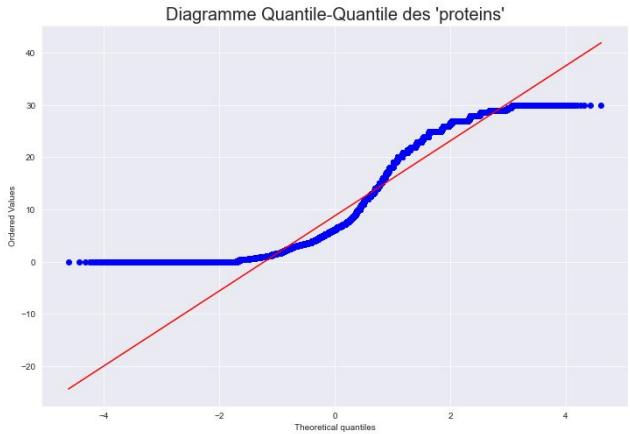
Exploration et analyse du jeu de données




Distribution du 'fat' par nutriscore
Density
fat_100g
nutriscore_adjusted
A
B
C
D
E


Distribution des 'carbohydrates' par nutriscore
Density
carbohydrates_100g
nutriscore_adjusted
A
B
C
D
E





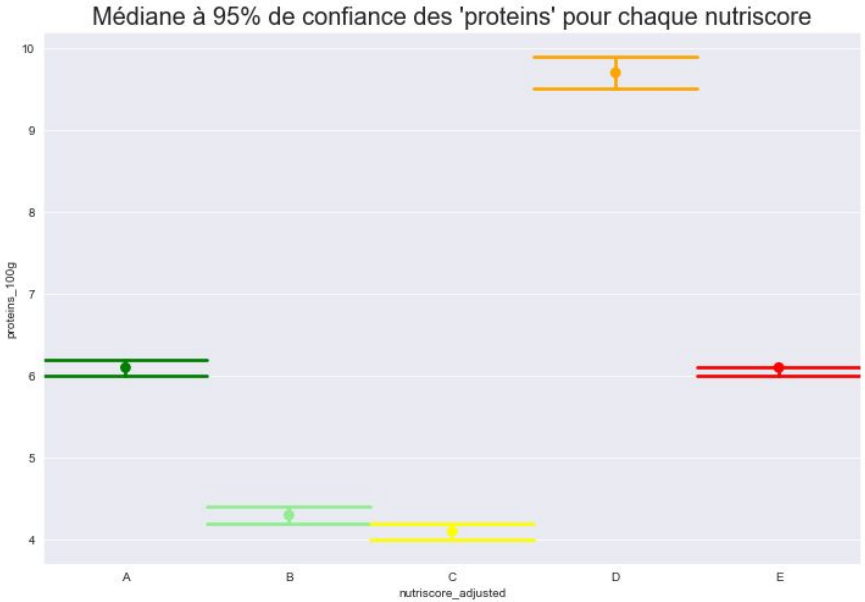
- Test de normalité (test de Jarque-Bera) sous l'hypothèse H0, la distribution suit une loi normale, à un niveau de confiance de 95%

  p-value < 0.05, on rejette donc H0

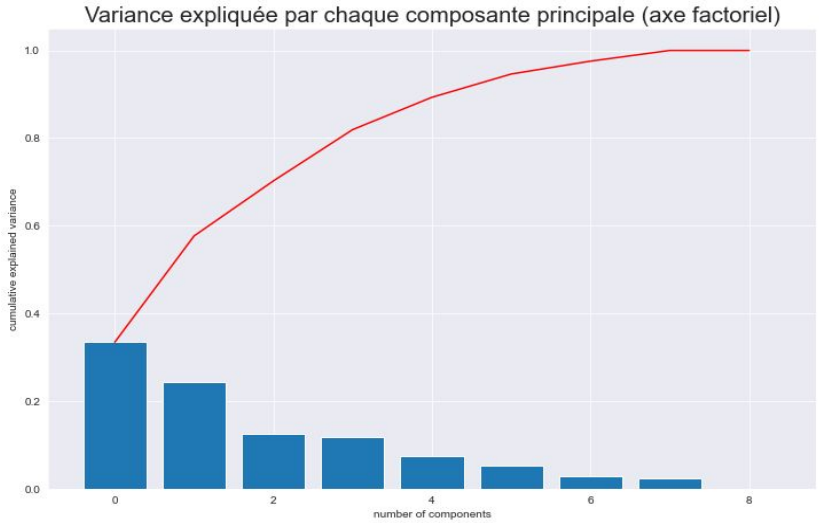
- Distribution dans plusieurs groupes (nutriscores)

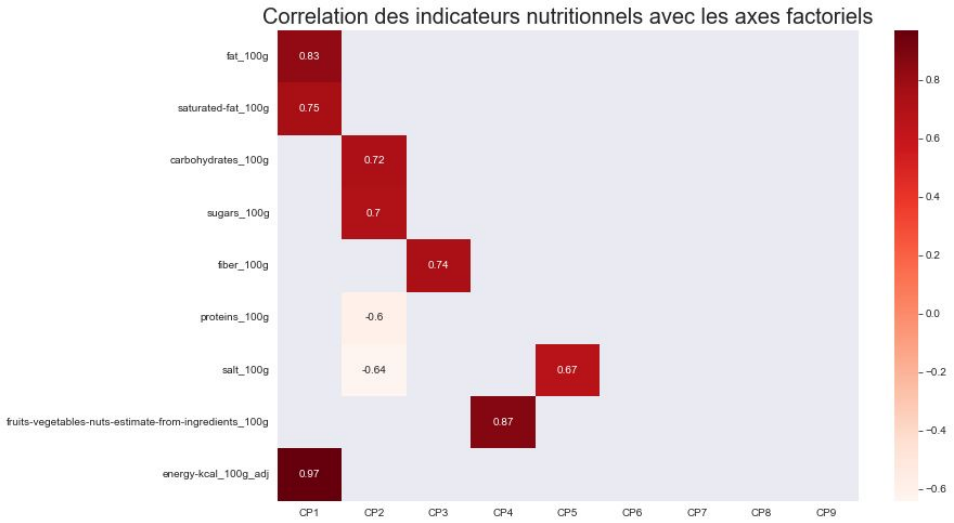
- Test non paramétrique de Kruskal-Wallis sous l'hypothèse H0, la distribution est la même dans tous les groupes, à un niveau de confiance de 95%

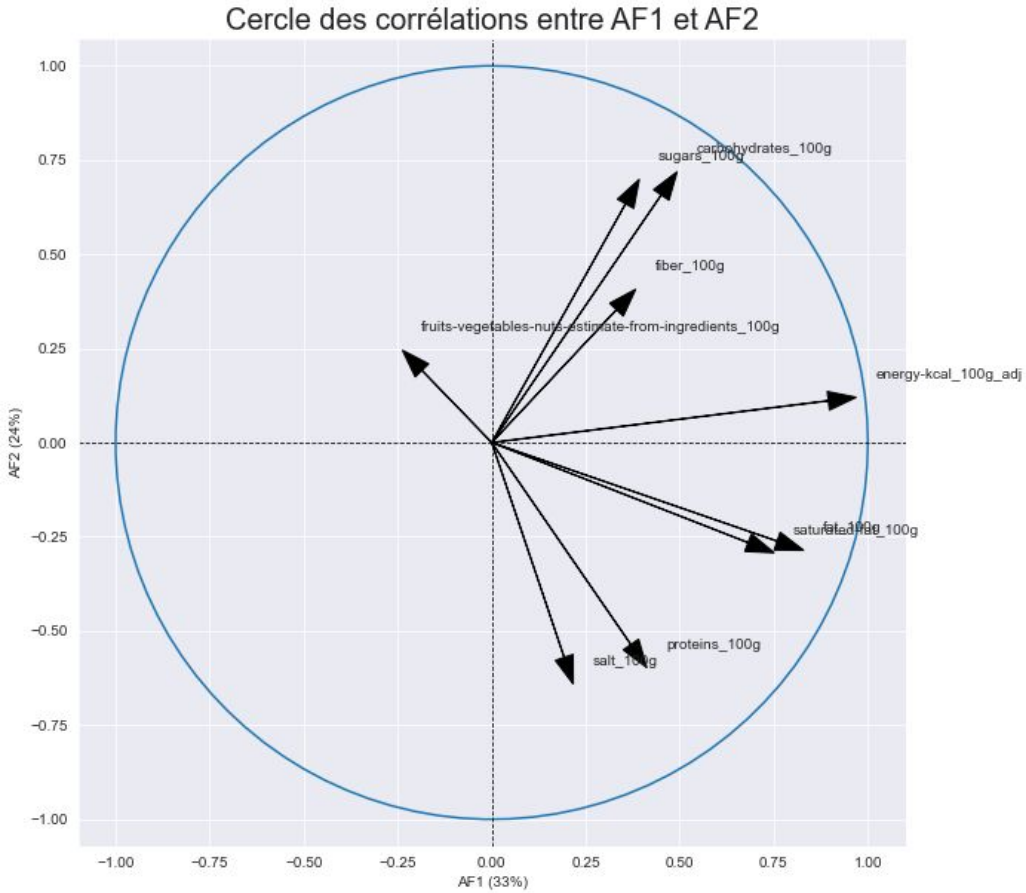
  p-value < 0.05, on rejette donc H0



- Réaliser une ACP (Analyse en composantes principales) afin de réduire les dimensions du jeu de données et obtenir une meilleure visualisation





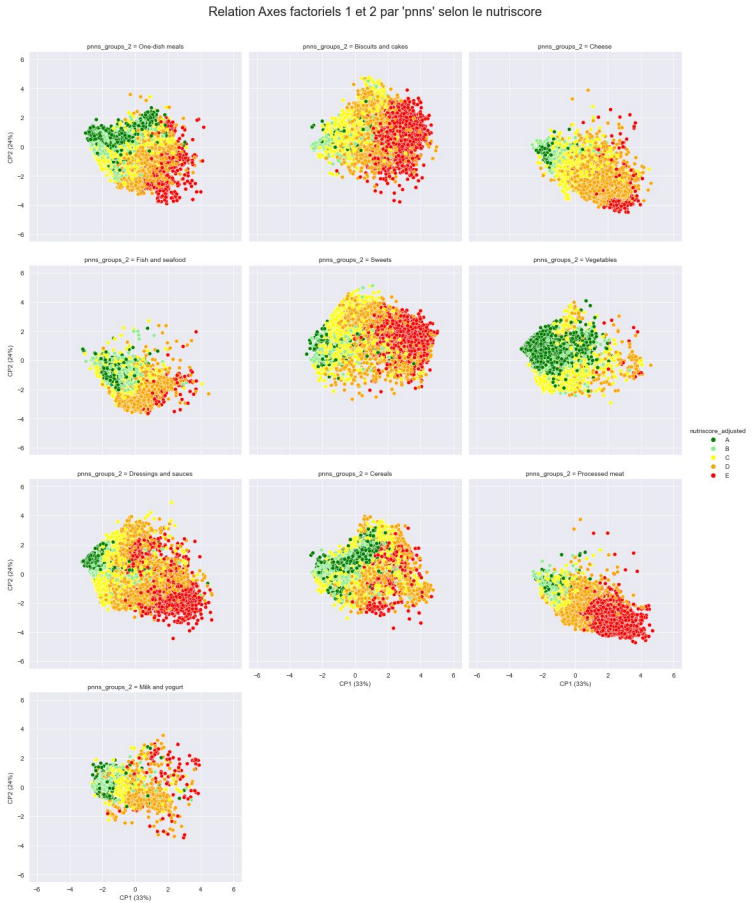


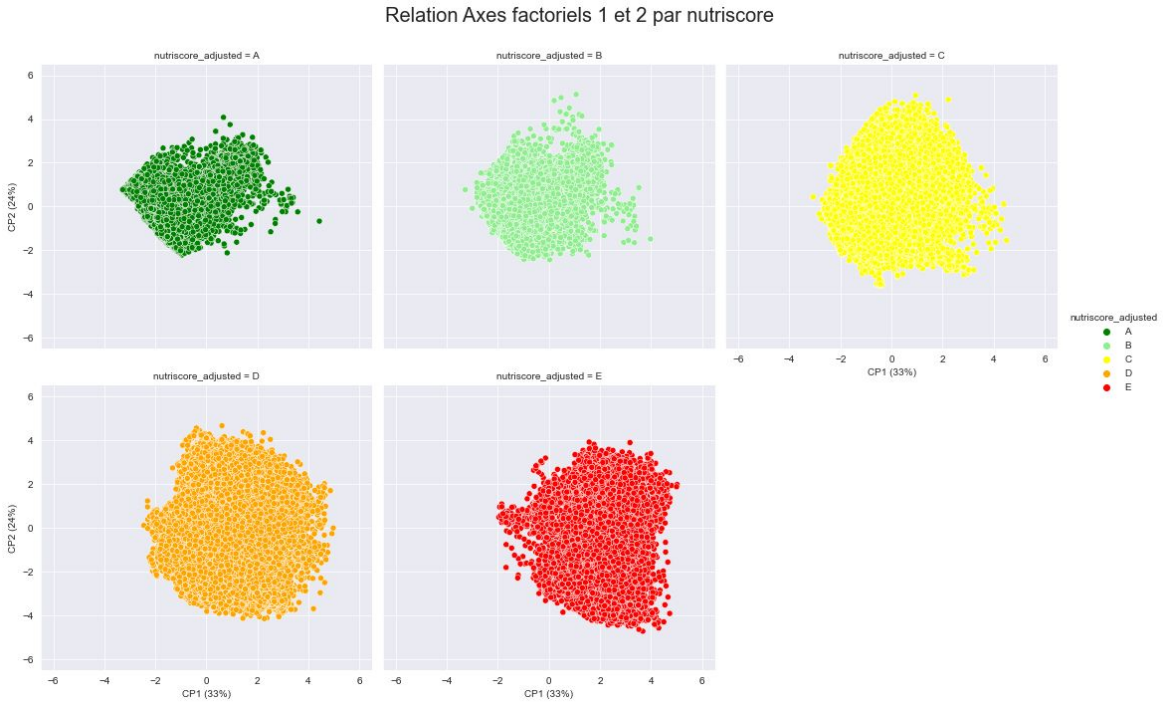
- **AF 1** est caractérisé par :
  'energy_kcal' (+)
  ' fat' (+)
  'saturated_fat' (+)

- **AF 2** est caractérisé par :
  'carbohydrates' (+)
  'sugars' (+)
  'proteins' (-)
  'salt' (-)

- **AF 1** est caractérisé par :
  ‘energy_kcal’ (+)
  ‘ fat’ (+)
  ‘saturated_fat’ (+)

- **AF 2** est caractérisé par :
  ‘carbohydrates’ (+)
  ‘sugars’ (+)
  ‘proteins’ (-)
  ‘salt’ (-)





- **AF 1** est caractérisé par :
  'energy_kcal' (+)
  ' fat' (+)
  'saturated_fat' (+)

- **AF 2** est caractérisé par :
  'carbohydrates' (+)
  'sugars' (+)
  'proteins' (-)
  'salt' (-)

- Nous avons vu avec plusieurs méthodes que les nutriscores sont “influencés” par les valeurs des indicateurs nutritionnels
- L’information supplémentaire du groupe ‘pnn’ vient affiner l’attribution du nutriscore
- L’idée d’application semble donc réalisable à partir du jeu de données

- Développer une interface d'application semblable pour tester la réalisation et l'utilisation de celle-ci

- **Bonus :**
  Réaliser un modèle de prédiction de classe (nutriscore) à l'aide d'un modèle KNN

  Paramètres : n_neighbors = 7, distance de Manhattan
  Score : 84% (soit 16% d'erreur)