



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

Raphaël GIRAUDOT

Sommaire :



1. Contexte de la mission
2. Présentation de l'idée d'application
3. Opérations de nettoyage effectuées
4. Récapitulatif des variables sélectionnées
5. Analyse univariée et description des variables
6. Analyse Bivariée
 - a. Quantitatives x Quantitatives : Corrélation de Pearson
 - b. Quantitatives x Qualitatives : ANOVA
 - c. Qualitatives x Qualitatives : Chi2
7. Analyse Multivariée - ACP
8. Prototypage de l'Application
9. Pertinence & Faisabilité
10. Conclusion

1. Contexte et objectifs de la mission

- Contexte :
 - Participation à l'appel à projet de l'agence "Santé Publique France"
 - Proposer une idée d'application innovante en lien avec l'alimentation
 - En utilisant les données d'Open Food Facts



2. Présentation de l'idée d'application 1/2

Alimentation & Santé :

D'après une [infographie de l'UFC Que Choisir de 2015](#) sur les chiffres clés du déséquilibre alimentaire :

- Près d'un adulte sur deux sont en **surpoids** ou **obèse**
- $\frac{1}{3}$ des cas de **cancers** seraient **évitables** par une **alimentation plus équilibrée**
- 17,6 % des Français sont traités pour l'**hypertension artérielle**
- Les **maladies cardio-vasculaires** sont la **2^{ème} cause de mortalité en France**
- La cause principale est une surconsommation de **sucres**, d'**acides gras saturés** et de **sel**.

Aussi afin de rester en bonne santé en faisant attention à son alimentation l'OMS a émis des [recommandations](#).

Seulement, de nos jours il n'est pas simple faire attention à la composition des aliments que l'on consomme par manque de temps, de connaissance et par la quantité de produits à notre disposition.

2. Présentation de l'idée d'application 2/2

Le visuel :

Mes Macros

Bienvenue sur Mes Macros !
Pour votre première visite
veuillez renseigner les
informations afin de calculer
vos besoins énergétiques :

Âge : ans

Sexe :

Poids : kg

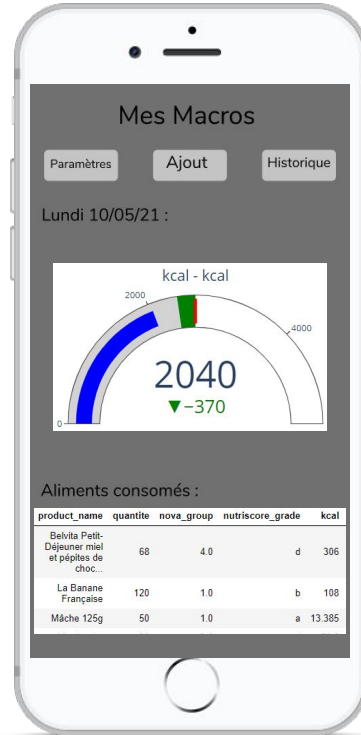
Taille : m

Niveau d'Activité :

Mes Macros

Afin de maintenir votre poids
actuel vous devez consommer,
au quotidien :

kcal



Mes Macros

Ajouter un aliment

Scanner un code barre :



Mes Macros

Ajouter un aliment

Prévisualisation



Quantité : g

Barilla pâtes
intégrale
spaghetti n°5
au ble c...

nova_group	nutriscore_grade	kcal
1.0	a	260.25

3. Opérations de nettoyage effectuées 1/2



- **Sélection des variables 'intéressantes' :**
 - Informations "Générales": 'code', 'last_modified_datetime', 'product_name', 'brands', 'countries', 'countries_tags', 'countries_en'
 - Groupe d'aliments: 'categories', 'main_category', 'main_category_en', 'pnns_groups_1', 'pnns_groups_2', 'nova_group', 'glycemic-index_100g'
 - Nutriments pour 100g: 'fiber_100g', 'fat_100g', 'saturated-fat_100g', 'carbohydrates_100g', 'sugars_100g', 'proteins_100g', 'salt_100g', 'sodium_100g'
 - Energie pour 100g: 'energy-kcal_100g', 'energy_100g', 'energy-kj_100g'
 - Scores de nutrition: 'nutriscore_score', 'nutriscore_grade', 'nutriscore-score-fr_100g'
- **Sélection de produits vendus en France :**

Filtre sur les colonnes 'countries', 'countries_tags', 'countries_en' qui contiennent 'fr' ou 'Fr'
- **Retrait des produits non-alimentaire :**

Filtre sur la colonne 'categories', qui contient une catégorie 'non-alimentaire' (Non alimentaire, Nourriture pour chats, Nourriture sèche pour animaux, Livres, etc...)
- **Uniformisation des colonnes relatives à l'énergie, 'energy-kcal_100g', 'energy_100g', 'energy-kj_100g' :**

Valorisation des données manquantes d'une colonne par une autre colonne sachant que 1 kcal = 4,1868 kJ

3. Opérations de nettoyage effectuées 2/2

- **Retrait des produits présentant des valeurs aberrantes :**
 - Colonnes Nutriments : les nutriments calculés pour 100g de produits donc les valeurs aberrantes sont :
 - un nutriments avec une valeur > 100g ou <0g
 - la **somme** des nutriments dépasse 100g, avec une marge d'erreur de 10%
 - **tous** les nutriments présentent des valeurs manquantes
 - Colonnes Énergie : Les lipides, 'fat', sont les nutriments qui dégagent le plus d'énergie pour 1g avec 9kcal :
 - Aliment le plus énergétique pour 100g contient 100g de lipides et donc dégage 900kcal.
 - Retrait des produits qui dépassent ce seuil avec une marge d'erreur à 1000kcal.
 - Retrait des produits dont **toutes** les colonnes d'énergies présente des valeurs manquantes.

Après ces opérations on a constitué un fichier de **599.428 individus**, ce qui représente **34,8%** des individus du fichier initial.

Aussi, avant de créer le fichier de travail final, on va retirer du tableau les colonnes qui ne vont pas nous servir dans l'analyse :
'last_modified_datetime', 'countries', 'countries_tags', 'countries_en', 'categories', 'main_category', 'main_category_en', 'sodium_100g',
'energy_100g', 'energy-kj_100g', 'nutrition-score-fr_100g'.

4. Récapitulatif des variables sélectionnées

“Générales” :

- **code** : code barre du produit
- **delta_jours_actu** : nombre de jours écoulés entre 'last_modified_datetime' et la date du jour
- **product_name** : nom du produit
- **brands** : marque du produit

“Catégories d'aliments” :

- **pnns_groups_1** : catégorie principale du produit selon le PNNS
- **pnns_groups_2** : catégorie secondaire du produit selon le PNNS
- **nova_group** : groupe selon le degré de transformation du produit
- **glycemic-index_100g** : index glycémique
- **classe_IG** : classe d'index glycémique

Nutriments (pour 100g) :

- **fiber_100g** : Fibres
- **fat_100g** : Lipides (graisses)
- **saturated-fat_100g** : Graisses Saturées (quantité incluse dans 'fat_100g')
- **carbohydrates_100g** : Glucides
- **sugars_100g** : Sucres (quantité incluse dans 'carbohydrates_100g')
- **proteins_100g** : Protéines
- **salt_100g** : Sel

Score Nutritionnel :

- **nutriscore_score** : score nutriscore
- **nutriscore_grade** : note nutriscore

Energie (pour 100g) :

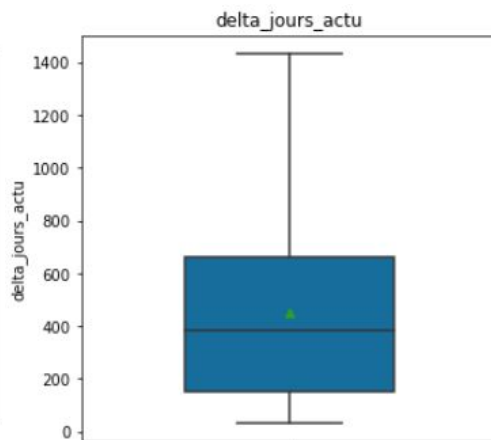
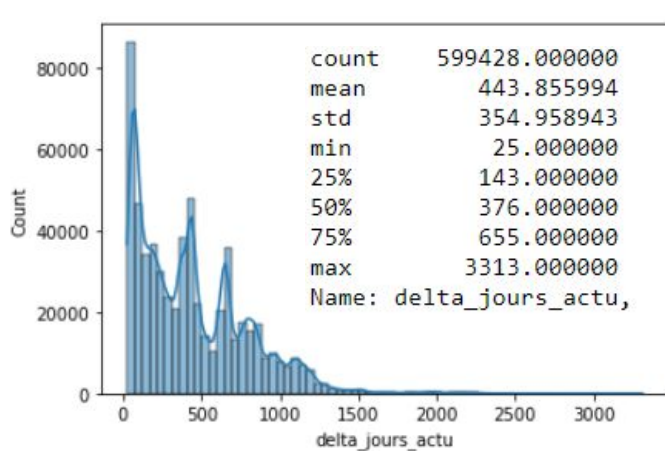
- **energy-kcal_100g** : nombre de kilocalories

5. Analyse univariée et description des variables

Variables "Générales" :

- **delta_jours_actu** : nombre de jours écoulés depuis la dernière modification, variable *quantitative discrète*
- **brands** : marque des produits, variable *qualitative nominale* à 58.081 modalités

Distribution 'delta jours actu' :



On observe qu'une très grande partie des produits ont un delta_jours_actu proche de 0, et cette valeur baisse plus delta_jours_actu augmente.

Les 10 marques les + présentes:

	brands	n	f
0	Auchan	8156	2.336983
1	Carrefour	7388	2.116924
2	U	4855	1.391129
3	Casino	4041	1.157890
4	Leader Price	3712	1.063619
5	Cora	2882	0.825795
6	Picard	2237	0.640980
7	Nestlé	2200	0.630378
8	Monoprix	1974	0.565621
9	Thiriet	1871	0.536108

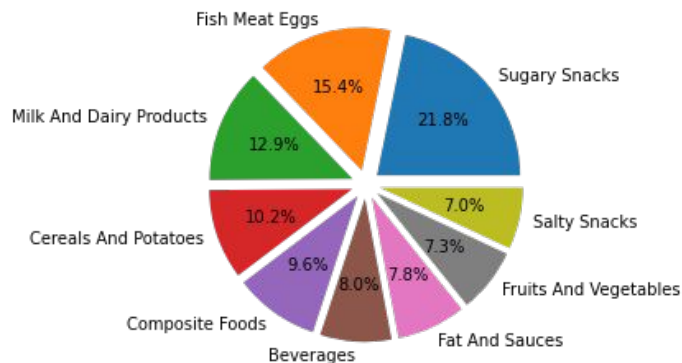
5. Analyse univariée et description des variables

Variables “Catégories d’Aliments” :

- **pnns_groups_1** : catégorie principale du produit selon le PNNS, variable *qualitative nominale* à (9+1) modalités
- **nova_group** : groupe selon le degré de transformation du produit, variable *qualitative ordinale*, à 4 modalités

pnns_groups_1	n	f
Unknown	334680	55.971890

55.97% des individus n’ont pas de groupe pnns_1 identifié.

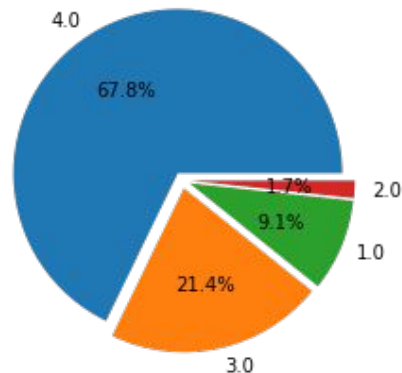


Parmi les 44.03% des individus restant, les ‘Sugary Snacks’ sont les plus représentés avec 21.8% des individus.

Répartition ‘nova_group’ :

69.16% des individus n’ont pas de nova_group de renseigné

```
nbNaCol(data, 'nova_group')  
(414547, 69.15709643193178)
```



Parmi les 30.84% des individus restant, le groupe 4.0 est le plus représenté avec 67.8% des individus.

5. Analyse univariée et description des variables

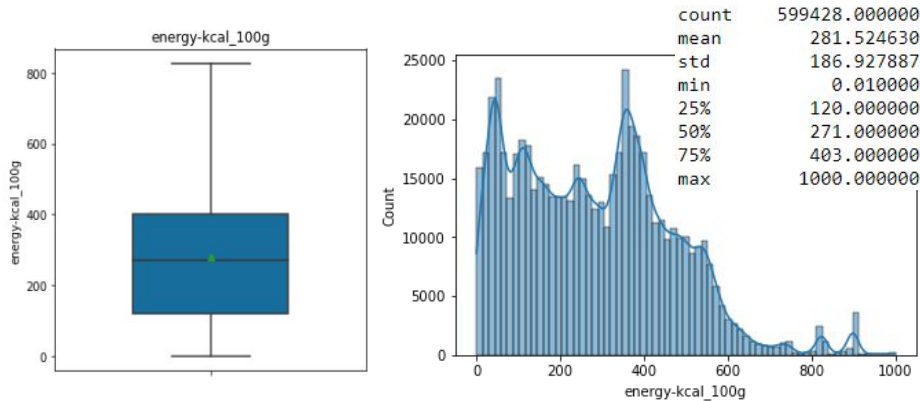
Variables Energy (pour 100g):

- **energy-kcal_100g** : nombre de calories pour 100g de produit, en kilocalories, variable *quantitative continue*

Variables Score Nutritionnel :

- **nutriscore_grade** : note du nutriscore, variable *qualitative ordinale*

energy-kcal 100g :



Un produit dégage, en moyenne, 282 kcal.

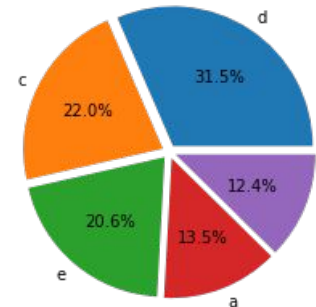
25% des individus dégagent plus de 403kcal.

On observe que le nombre de produit chute un peu avant 600kcal.

Par contre, on remarque des pics légèrement au dessus de 800kcal et aux alentours de 900kcal.

Distribution :

```
nbNaCol(data, 'nutriscore_grade')  
(327176, 54.58136757041713)
```

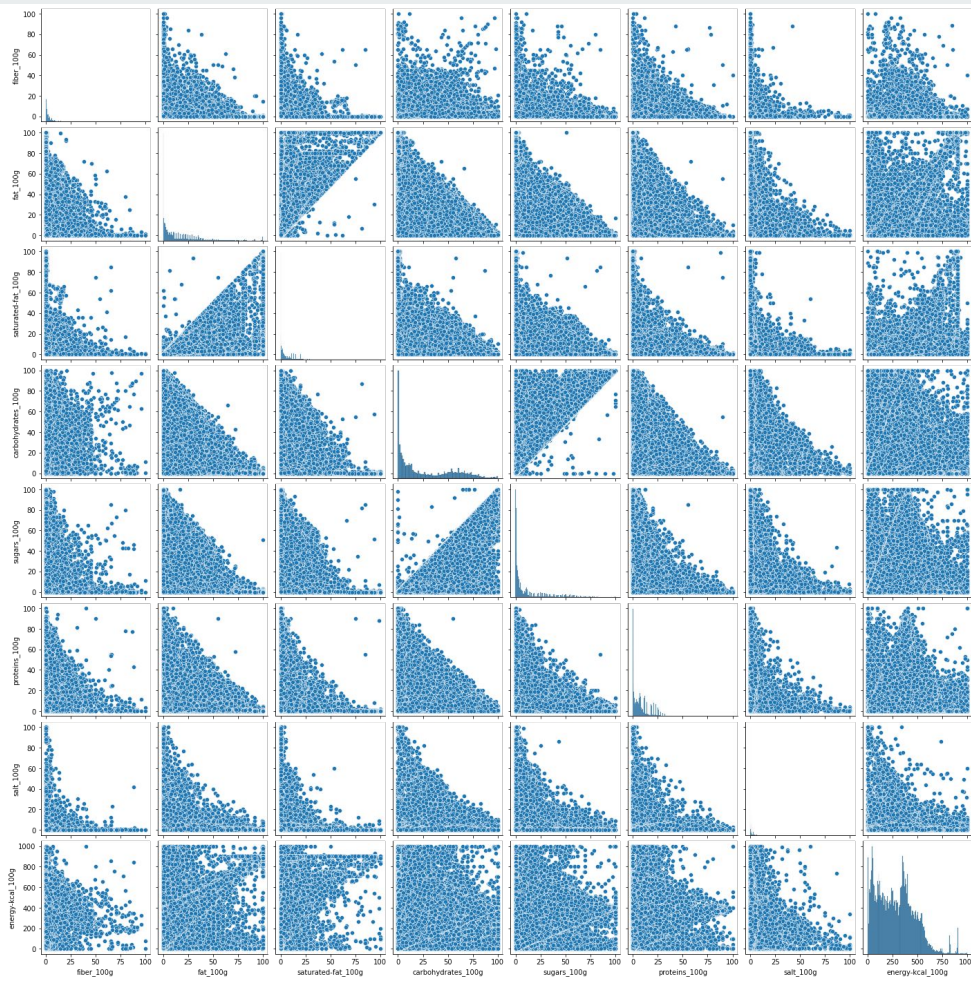


Seulement 45,42% des aliments présentent une classe du nutriscore.

La classe la plus représentée est la classe 'd' avec plus de 31,5% des individus.

Les classes 'a' et 'b' réunies ne représentent que 25,9% des produits.

6. Analyse Bivariée - Corrélations de Pearson



Pour visualiser les corrélations nous allons utiliser la fonction **pairplot** de *Seaborn* qui nous permet de créer une matrice de **scatterplot**.

On observe pas droite qui se dessine clairement et qui pourrait sous entendre une relation linéaire entre 2 variables.

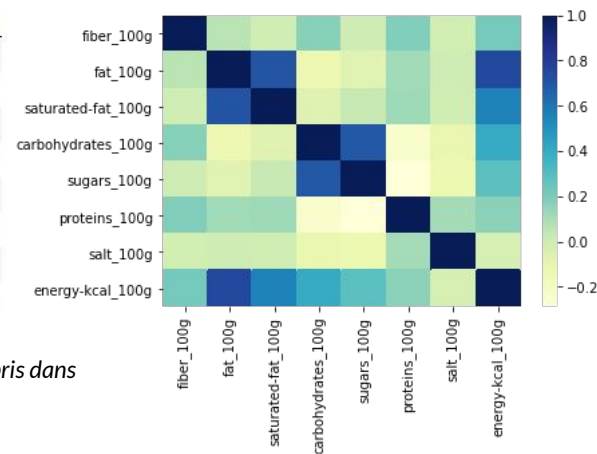
De plus, vu que l'on a beaucoup d'individus, il peut y avoir du "bruit" fausse la lecture.

6. Analyse Bivariée - Corrélations de Pearson

quantitative x quantitative :

Est-ce que des variables quantitatives sont corrélées ?

	fiber_100g	fat_100g	saturated-fat_100g	carbohydrates_100g	sugars_100g	proteins_100g	salt_100g	energy-kcal_100g
fiber_100g	1.000000	0.061484	-0.002386	0.179092	0.006232	0.190000	-0.016243	0.220780
fat_100g	0.061484	1.000000	0.710954	-0.136776	-0.072241	0.119310	0.007913	0.755497
saturated-fat_100g	-0.002386	0.710954	1.000000	-0.066627	0.031372	0.127492	-0.001492	0.563734
carbohydrates_100g	0.179092	-0.136776	-0.066627	1.000000	0.702530	-0.234911	-0.107615	0.406120
sugars_100g	0.006232	-0.072241	0.031372	0.702530	1.000000	-0.288207	-0.124892	0.283257
proteins_100g	0.190000	0.119310	0.127492	-0.234911	-0.288207	1.000000	0.109489	0.167781
salt_100g	-0.016243	0.007913	-0.001492	-0.107615	-0.124892	0.109489	1.000000	-0.032364
energy-kcal_100g	0.220780	0.755497	0.563734	0.406120	0.283257	0.167781	-0.032364	1.000000



Corrélation de Pearson : Fait ressortir une relation linéaire entre 2 variables quantitatives. Le coefficient de corrélation est compris dans l'intervall [-1;1]:

- 0 : relation nulle entre les variables,
- >0 : signifie que les variables varient dans le même sens,
- <0 : signifie que lorsqu'une variable augmente l'autre diminue

Il en ressort les coefficient de corrélation suivant :

1. **energy-kcal_100g x fat_100g** : 0,755497, les lipides sont les nutriments les plus énergétiques (9kcal)
2. **saturated-fat_100g x fat_100g** : 0,710954, la quantité de saturated_fat est incluse dans fat_100g
3. **sugars_100g x carbohydrates_100g** : 0,702530, la quantité de sucres sont inclus dans les carbohydrates
4. **energy-kcal_100g x saturated-fat_100g** : 0.563734, les graisses saturées sont des lipides, mais présent en plus petites quantité que les graisses non-saturées

6. Analyse Bivariée - ANOVA

Introduction à l'ANOVA & Hypothèses sous-jacentes

l'ANOVA va nous permettre de voir si il y a des liens entre des variables qualitatives et des variables quantitatives.

Avant de réaliser l'ANOVA il va falloir que l'on vérifie que les populations que l'on étudies sont i.i.d (indépendamment et identiquement distribués)

1. Les populations étudiées sont-elles indépendantes ?

Ici on s'intéresse aux variables **qualitatives** : **pnns_groups_1**, **nova_group** et **nutriscore_grade**

Pour chaque variable qualitative les populations sont effectivement indépendantes.

Chaque produit présente un **unique pnns_groups_1**, un **unique nova_group** et un **unique nutriscore_grade**.

2. Les populations étudiées suivent-elle une Loi Normale ?

Ici on s'intéresse aux variables **quantitatives**.

Lors de l'analyse on a pas vu de distribution se rapprochant d'une distribution normale.

On va donc vérifier cette hypothèse en réalisant des tests de normalité.

```
stats.mstats.normaltest(data['kcal'])
```

```
NormaltestResult(statistic=28791.11579465828, pvalue=0.0)
```

```
stats.mstats.normaltest(data['fiber_100g'])
```

```
NormaltestResult(statistic=992553.9814421922, pvalue=0.0)
```

```
stats.mstats.normaltest(data['saturated_fat'])
```

```
NormaltestResult(statistic=429537.1551308122, pvalue=0.0)
```

```
stats.mstats.normaltest(data['sugars_100g'])
```

```
NormaltestResult(statistic=187347.69535608418, pvalue=0.0)
```

```
stats.mstats.normaltest(data['salt_100g'])
```

```
NormaltestResult(statistic=1148968.9167780522, pvalue=0.0)
```

Tous les tests présentent donc une pvalue < 0.05, on rejette donc l'hypothèse nulle que l'échantillon provient d'une distribution Normale.

-> ANOVA non-paramétrique :

- Kruskal-Wallis
- Mann-Whitney (paires)

6. Analyse Bivariée - ANOVA

Synthèse des résultats:

Toutes les p-values des tests de Kruskal-Wallis sont inférieures à 0.05.

Les moyennes de variables quantitatives sont donc significativement différentes suivant les groupes de variables qualitatives.

Sauf pour **Sugary Snacks X Salty Snacks** avec la quantité moyenne de graisses saturées :

- Resultats test Mann-Whitney : `Sugary Snacks / Salty Snacks`
`stat : 523534446.0 p-value : 0.2273084524027439 rejet : False`

Et aussi pour **Fat & Sauces X Salty Snacks** pour la quantité moyenne de **sucres** :

- Resultats test Mann-Whitney : `Fat And Sauces / Salty Snacks`
`stat : 187298442.0 p-value : 0.19125388114176878 rejet : False`

Le Nutriscore remplit son rôle, en effet on peut voir que le classement du Nutriscore est respectée pour les taux de nutriments à limiter : **saturated-fat, sugars**.

On a pu voir que le NOVA group '2' est le groupe qui contient, en moyenne, le plus de **sucres**, de **graisses saturées** et le plus **énergétique**.

	nova_group	mean kcal
0	2.0	617.773685
1	0.0	284.768270
2	4.0	279.092781
3	3.0	261.424702
4	1.0	204.002349

	nova_group	mean saturated_fat
0	2.0	21.675664
1	3.0	6.392273
2	0.0	5.533693
3	4.0	5.354984
4	1.0	1.480082

	nova_group	mean sugars_100g
0	2.0	24.507120
1	4.0	17.888365
2	0.0	13.604139
3	3.0	8.073996
4	1.0	6.858046

6. Analyse Bivariée - Chi-2

Introduction au test du Chi2 & Prérequis

Le test du Chi2 va nous permettre de tester l'indépendance entre 2 variables qualitatives.

Ici nous avons 3 variables qualitatives à évaluer nous devons donc réaliser 3 tests du Chi2 :

- nova_group x pnns_groups_1
- nutriscore_grade x pnns_groups_1
- nutriscore_grade x nova_group

Prérequis au Chi2 : Chaque classe doit présenter un effectif théorique non-nul et qu'au moins 80% des classes présentent un effectif théorique supérieur ou égal à 5

Vérification des effectifs théoriques :

pnns_groups_1	0.0	1.0	2.0	3.0	4.0	Total
Beverages	1.5e+04	5.9e+02	1.1e+02	1.4e+03	4.4e+03	2.1e+04
Cereals And Potatoes	1.9e+04	7.5e+02	1.4e+02	1.8e+03	5.6e+03	2.7e+04
Composite Foods	1.8e+04	7.1e+02	1.3e+02	1.7e+03	5.3e+03	2.5e+04
Fat And Sauces	1.4e+04	5.8e+02	1.1e+02	1.4e+03	4.3e+03	2.1e+04
Fish Meat Eggs	2.8e+04	1.1e+03	2.1e+02	2.7e+03	8.5e+03	4.1e+04
Fruits And Vegetables	1.3e+04	5.4e+02	1e+02	1.3e+03	4e+03	1.9e+04
Milk And Dairy Products	2.4e+04	9.5e+02	1.8e+02	2.2e+03	7.1e+03	3.4e+04
Salty Snacks	1.3e+04	5.1e+02	96	1.2e+03	3.8e+03	1.8e+04
Sugary Snacks	4e+04	1.6e+03	3e+02	3.8e+03	1.2e+04	5.7e+04
Unknown	2.3e+05	9.4e+03	1.8e+03	2.2e+04	7e+04	3.4e+05
Total	4.1e+05	1.7e+04	3.1e+03	4e+04	1.3e+05	6e+05

pnns_groups_1	NA	a	b	c	d	e	Total
Beverages	1.2e+04	1.3e+03	1.2e+02	1e+03	3e+03	2e+03	2.1e+04
Cereals And Potatoes	1.5e+04	1.6e+03	1.5e+02	7e+02	3.8e+03	5e+03	2.7e+04
Composite Foods	1.4e+04	1.6e+03	1.4e+02	5e+02	3.6e+03	4e+03	2.5e+04
Fat And Sauces	1.1e+04	1.3e+03	1.2e+02	1e+03	2.9e+03	1.9e+03	2.1e+04
Fish Meat Eggs	2.2e+04	5e+03	2.3e+03	1e+03	3.8e+03	8e+03	4.1e+04
Fruits And Vegetables	1e+04	1.2e+03	1.1e+02	9e+02	2.7e+03	1.8e+03	1.9e+04
Milk And Dairy Products	1.9e+04	1e+03	1.9e+03	3.4e+03	3.9e+03	3.2e+03	3.4e+04
Salty Snacks	1e+04	1.1e+03	1e+03	1.8e+02	2.6e+03	1.7e+03	1.8e+04
Sugary Snacks	3.1e+04	5e+03	2e+03	7e+03	3.2e+03	4e+03	5.7e+04
Unknown	1.8e+05	1e+04	9e+03	4e+04	8e+04	1e+05	3.4e+05
Total	3.3e+05	7e+04	4e+04	6e+04	8.6e+04	5.6e+04	6e+05

nova_group	NA	a	b	c	d	e	Total
0.0	2.3e+05	5e+04	2.3e+04	2e+04	5.9e+04	4.9e+04	1e+05
1.0	9.2e+03	1e+03	9.4e+02	1.7e+03	2.4e+03	1.6e+03	1.7e+04
2.0	1.7e+03	1.9e+02	1.8e+02	2.2e+02	5e+02	2.9e+02	3.1e+03
3.0	2.2e+04	4e+03	2e+03	4e+03	5.7e+03	3.7e+03	3.4e+04
4.0	6.8e+04	7e+03	7e+03	3.1e+04	4.8e+04	1.2e+04	1.3e+05
Total	3.3e+05	7e+04	4e+04	6e+04	8.6e+04	5.6e+04	6e+05

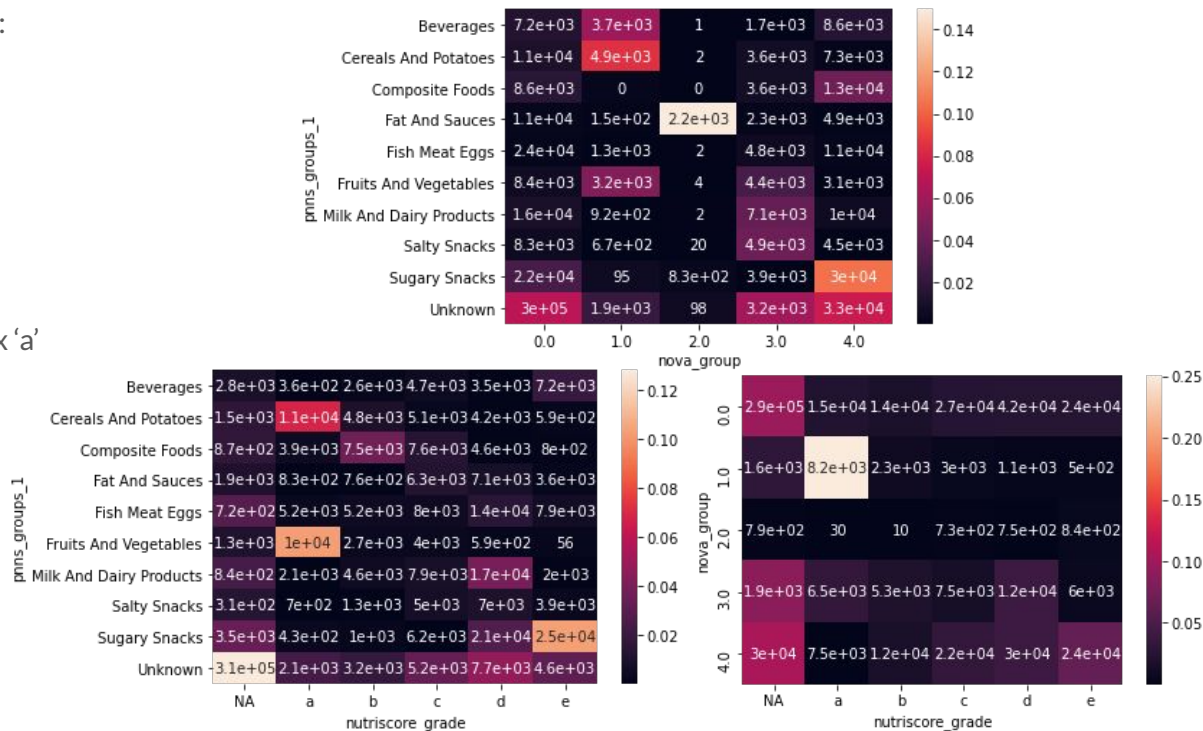
Tous les effectifs théoriques des classes sont supérieur à 5 on peut donc réaliser les tests du Chi2.

6. Analyse Bivariée - Chi-2

Synthèse des résultats remarquables :

Les 3 tests du chi-2 renvoient une p-value inférieure à 0.05. On rejette donc l'hypothèse d'indépendance des variables entre-elle et en particulier pour les paires suivantes :

- PNNS 1 x NOVA:
 1. 'Fat And Sauces' x '2.0'
 2. 'Sugary Snacks' x '4.0'
- PNNS 1 x Nutriscore:
 3. 'Fruits And Vegetables' x 'a'
 4. 'Sugary Snacks' x 'e'
- NOVA x Nutriscore:
 5. '1.0' x 'a'



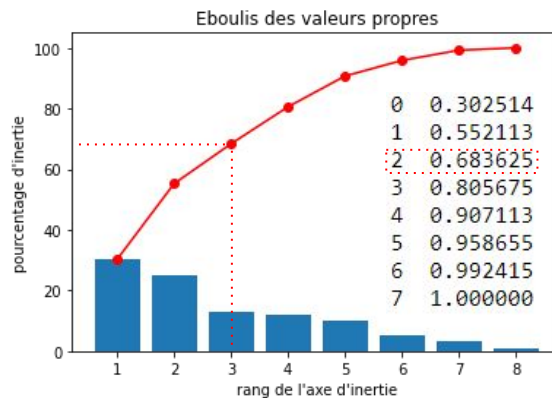
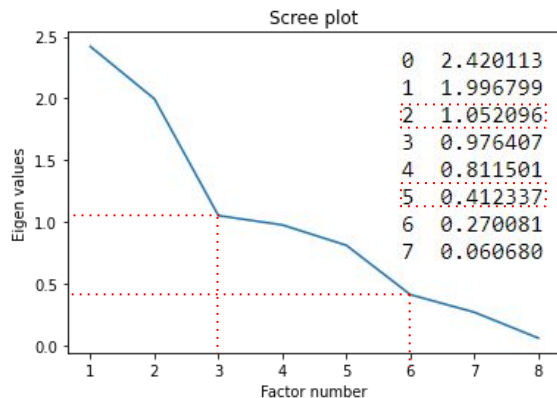
7. Analyse Multivariée - ACP

Choix du nombres d'axes pour l'analyse :

- On va réaliser une ACP pour voir si des groupes d'individus se détachent.
- Nombre de valeurs propres = nombre de variables = 8



- Visualisation de la variance expliquée :

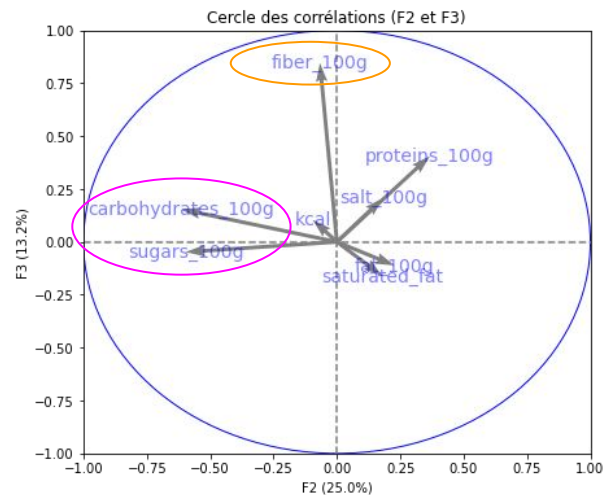
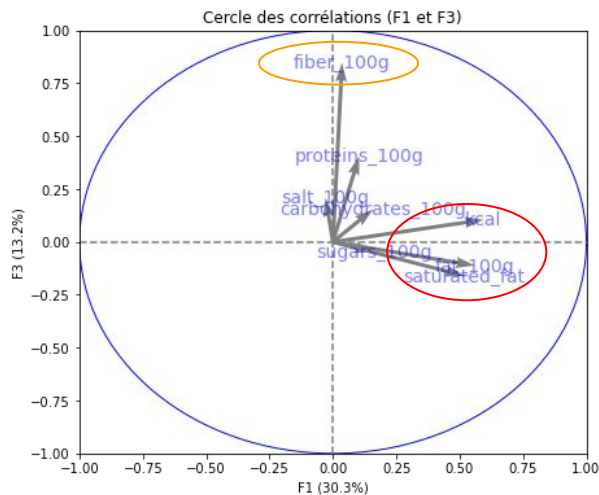
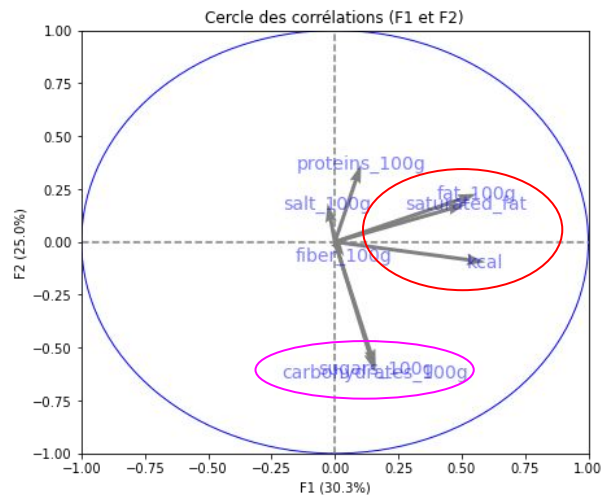


- Règle de Kaiser : On ne garde que les axes pour lesquels on a une valeur propre > 1
 - Ici on gardera donc les 3 premiers axes
- Règle du coude : On garde les axes jusqu'au 'coude', changement de pente significatif
 - Sur la figure de gauche on observe 2 coudes, pour l'axe 3 et pour l'axe 6

En combinant ces 2 règles, on va retenir 3 axes pour 68% de variance expliquée

7. Analyse Multivariée - ACP

Analyse en composantes principales :



Axe 1:

id	COS2_1
kcal	0.855782
fat_100g	0.757421
saturated_fat	0.652150
sugars_100g	0.062609
carbohydrates_100g	0.060898
proteins_100g	0.026088
fiber_100g	0.003155
salt_100g	0.002011

Axe 2:

id	COS2_2
carbohydrates_100g	0.753539
sugars_100g	0.712965
proteins_100g	0.269343
fat_100g	0.103293
salt_100g	0.066600
saturated_fat	0.063782
kcal	0.018284
fiber_100g	0.008993

Axe 3:

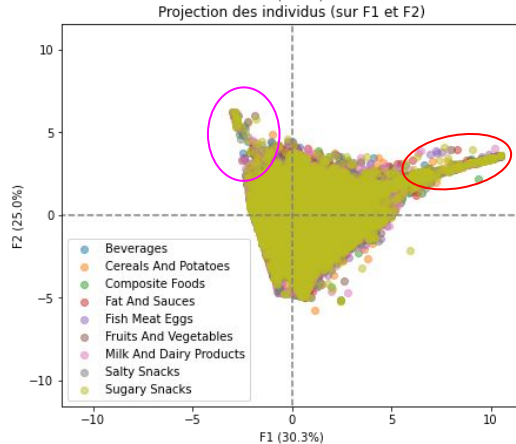
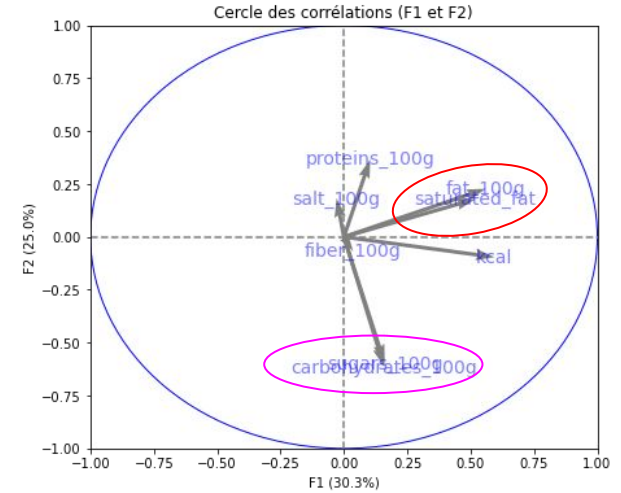
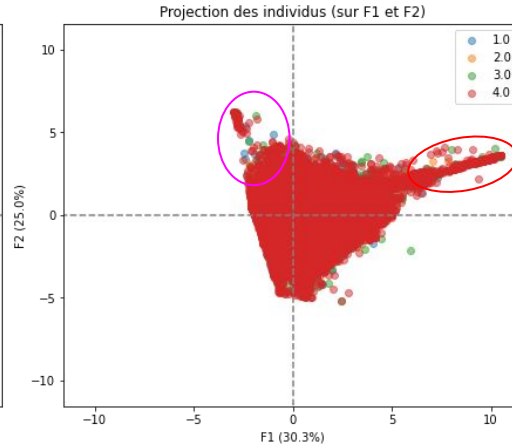
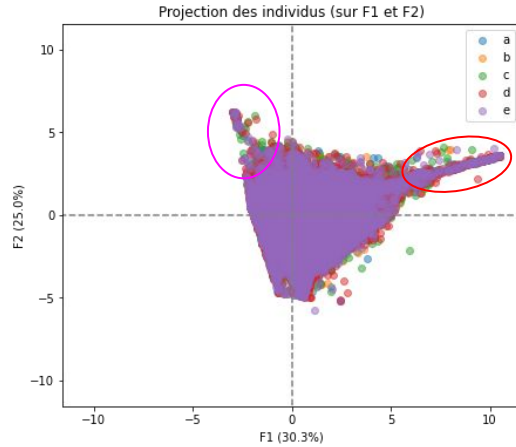
id	COS2_3
fiber_100g	0.754248
proteins_100g	0.173349
salt_100g	0.045375
saturated_fat	0.027080
carbohydrates_100g	0.025013
fat_100g	0.013102
kcal	0.011520
sugars_100g	0.002410

Variables les mieux représentées sur les axes :

- Axe 1:
kcal, fat_100g et saturated_fat
- Axe 2:
carbohydrates_100g et sugars_100g
- Axe 3:
fiber_100g

7. Analyse Multivariée - ACP

Position des individus : Interprétation de la position des individus par rapports à l'analyse des variables des axes (Axe 1 & 2)



Axe 1 :

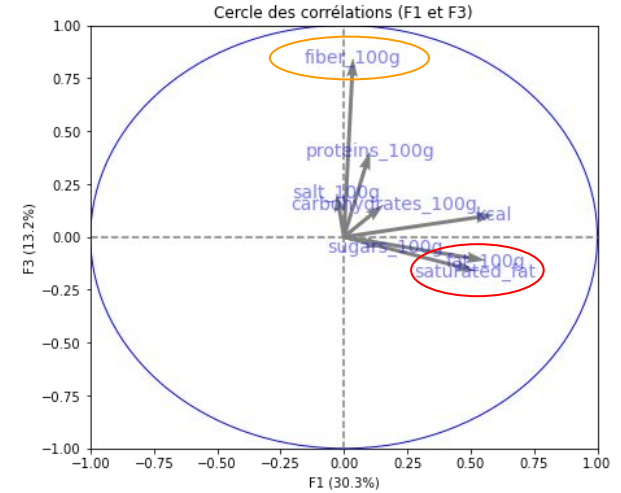
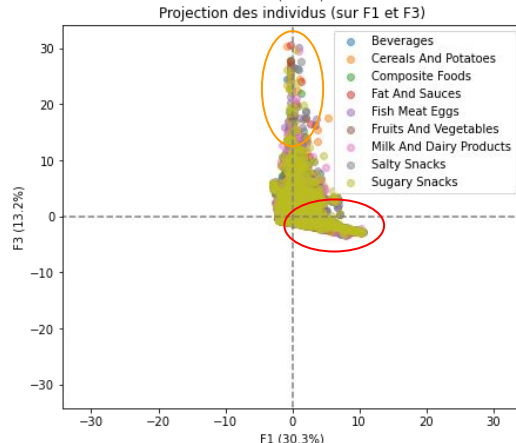
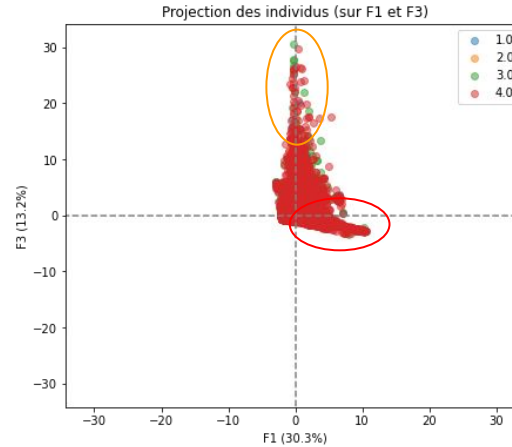
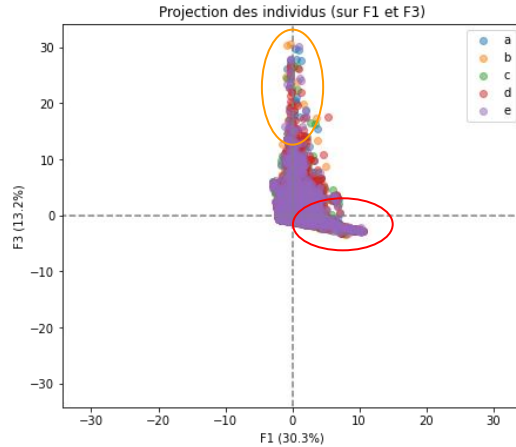
Étalement des individus suivant les vecteurs 'fat_100g', 'saturated-fat_100g'

Axe 2 :

Étalement d'individus à l'opposée des vecteurs 'carbohydrates_100g' et 'sugars_100g'

7. Analyse Multivariée - ACP

Position des individus : Interprétation de la position des individus par rapports à l'analyse des variables des axes (Axe 1 & 3)



Axe 1 :

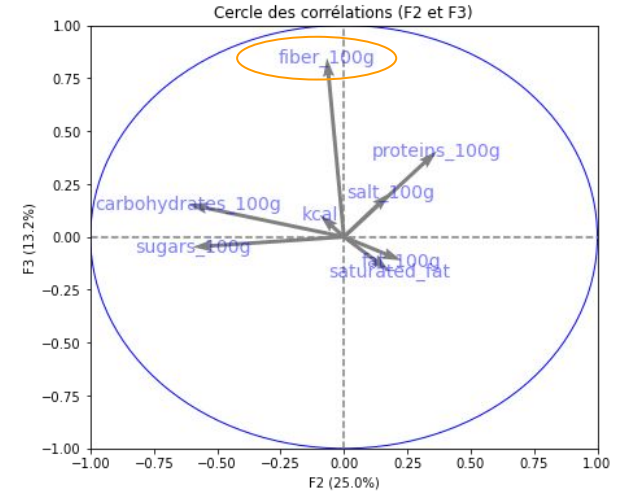
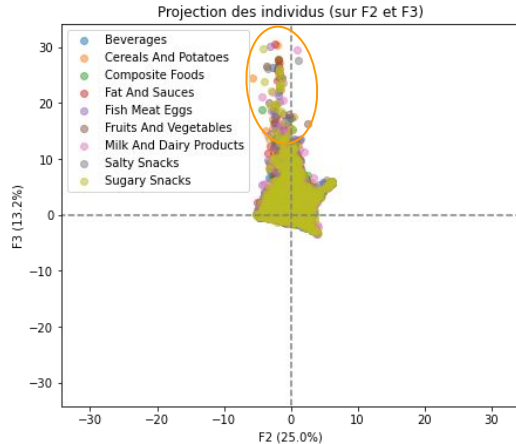
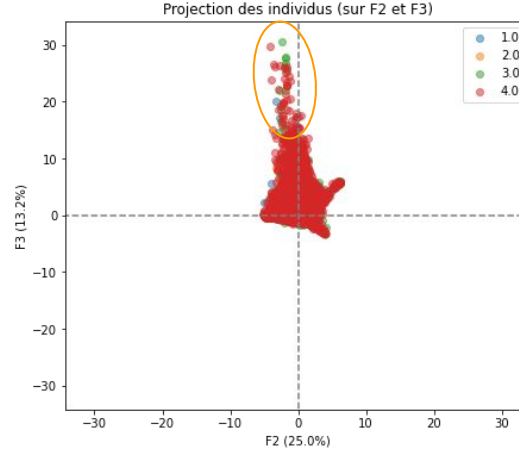
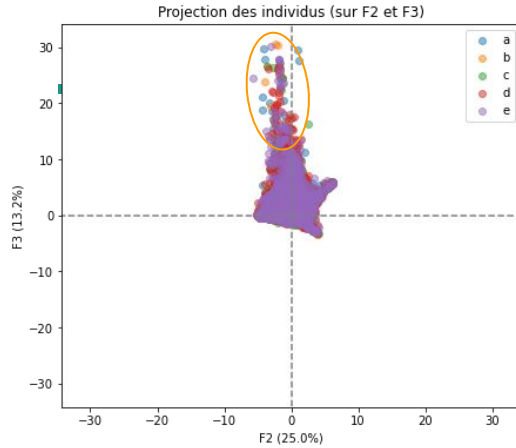
Étalement des individus suivant les vecteurs 'fat_100g', 'saturated-fat_100g'

Axe 3 :

Étalement des individus suivant les vecteurs 'fiber_100g'

7. Analyse Multivariée - ACP

Position des individus : Interprétation de la position des individus par rapports à l'analyse des variables des axes (Axe 2 & 3)



Axe 3 :

Étalement des individus suivant les vecteurs 'fiber_100g'

7. Analyse Multivariée - ACP



Synthèse ACP :

L'interprétation de l'ACP nous a permis d'observer des tendances des produits à s'aligner sur les axes des vecteurs des variables :

- Glucides (dont sucres)
- Lipides (dont graisses saturées)
- Fibres
- Energie, en kcal

Pas de regroupement d'individus en utilisant des variables illustratives tels que le nutriscore, le groupe pnns_1 ou le degré de transformation NOVA.

Cette analyse nous permet donc de voir que le nutriscore et/ou de son degré de transformation ne permettent pas de prendre totalement en compte les nutriments et l'énergie d'un produit.

8. Prototypage de l'Application

On va définir des fonctions dans le but de réaliser un prototype de l'application :



Fonction qui définit les besoins quotidiens, en kcal :

```
#fonction qui calcule le besoin en calories + répartition
#coef_act : sedentaire:1.37 / actif:1.55 / sportif:1.8 - 2.0
def kcalJourna(sexe, poids, taille, age, coef_act):
    kcal=(poids**0.48)*(taille**0.5)*(age**(-0.13))
    if sexe=='F':
        kcal=kcal*230
    elif sexe=='M':
        kcal=kcal*259
    kcal=kcal*coef_act
    return np.around(kcal,0)
```

Fonction qui ajoute les aliments consommés dans un tableau:

```
#fonction qui ajoute des aliments consommés dans un tableau
def AddAlimentConso(source, df_quoti, code, quantite):
    df_quoti=df_quoti.append({'code':code}, ignore_index=True) #ajout ligne et valorisation du code barre

    #ajout nom produit
    valeur=source['product_name'][source['code']==code].reset_index(drop=True)
    df_quoti['product_name'][df_quoti['code']==code]=valeur[0]

    #ajout quantite
    df_quoti['quantite'][df_quoti['code']==code]=quantite

    #ajout nova groupe
    valeur=source['nova_group'][source['code']==code].reset_index(drop=True)
    df_quoti['nova_group'][df_quoti['code']==code]=valeur[0]

    #ajout nutriscore
    valeur=source['nutriscore_grade'][source['code']==code].reset_index(drop=True)
    df_quoti['nutriscore_grade'][df_quoti['code']==code]=valeur[0]

    #ajout des fibres&kcal pour la qtte sélectionnée
    valeur=source['fiber_100g'][source['code']==code].reset_index(drop=True)
    valcalc=valeur[0]*quantite/100
    df_quoti['fibres'][df_quoti['code']==code]=valcalc
    df_quoti['kcal_fibres'][df_quoti['code']==code]=valcalc*2
```


8. Prototypage de l'Application

On va définir des fonctions dans le but de réaliser un prototype de l'application :



Fonction qui renvoie l'axe sur lequel l'aliment contribue le plus :

```
#fonction qui renvoie l'axe sur lequel l'aliment contribue le plus
def maxContrib(data,df_ctrInd,code):
    idx=data[data['code']==code].index[0]#on récupère l'index du produit
    ctrib=0

    for i in range(3):
        ncol=i+1
        if df_ctrInd.iloc[idx,ncol] > ctrib:
            ctrib=df_ctrInd.iloc[idx,ncol]
            maxcol=ncol

    nomcol=df_ctrInd.columns[maxcol]
    tabsupmoy=df_ctrInd[df_ctrInd[nomcol]>(sum(df_ctrInd[nomcol])/len(df_ctrInd[nomcol]))]
    if len(tabsupmoy[tabsupmoy.index.isin([idx])==True]) == 1:
        classee=tabsupmoy.sort_values(by=[nomcol],ascending=False).reset_index(drop=True)
        rang=classee[classee['id']==idx].index[0]
        top=np.around(rang/len(classee)*100)
        print("Contribue le plus à l'axe :",maxcol,"\n C'est le",rang,"ième produit participant")
    else:
        print("Le produit ne se détache pas particulièrement")
```

Fonction qui calcule la somme des nutriments et d'énergie des aliments consommés :

```
def totalJour(df):
    df_res = pd.DataFrame(columns=['kcal','kcal_fibres','kcal_lipides','kcal_graisses_saturees','kcal_glucides','kcal_sucres'],'kcal':df['kcal'].sum(),
    df_res=df_res.append({'kcal':df['kcal'].sum(), 'kcal_fibres':df['kcal_fibres'].sum(), 'kcal_lipides':df['kcal_lipides'].sum()})
    return df_res
```

8. Prototypage de l'Application

Simulation :

- Homme de 26 ans, 75 kg, 1m87 avec une “activité” sédentaire :



```
kcal_test=kcalJourna('M',75,1.87,26,1.37)
kcal_test
```

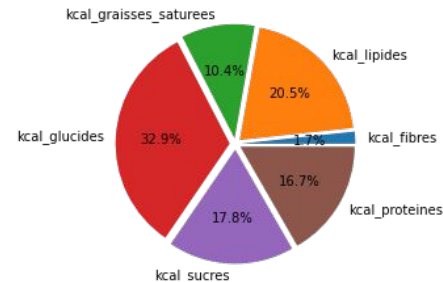
2524.0

Repas :

- Petit-Déjeuner : 5 belvita + 1 banane
- Midi :
 - Salade & Vinaigrette
 - Pâtes carbonara
 - 75g spaghetti
 - 50g lardons
 - 5g beurre
 - 6,5cl creme
 - 1oeuf
 - 30g parmesan
 - 1g sel(pincée)
 - 1 Compote de Pomme
- Goûter :
 - 2 tranche Pain de mie
 - Nutella
- Dîner :
 - Taboulé
 - Riz
 - Saumon
 - Yaourt

	code	product_name	quantite	nova_group	nutriscore_grade	kcal	fibres	kcal_fibres	lipides
0	7622210713889	Belvita Petit-Déjeuner miel et pépites de choc...	68	4.0	d	306	3.944	7.888	8.432
1	3347761000663	La Banane Française	120	1.0	b	108	0	0	0
2	3280220892006	Mâche 125g	50	1.0	a	13.385	1.2	2.4	0.245
3	5400101036734	Vinaigrette	30	0.0	d	58.8	0	0	5.55
4	8076809529419	Barilla pates integrale spaghetti n°5 au ble c...	75	1.0	a	260.25	6	12	1.5

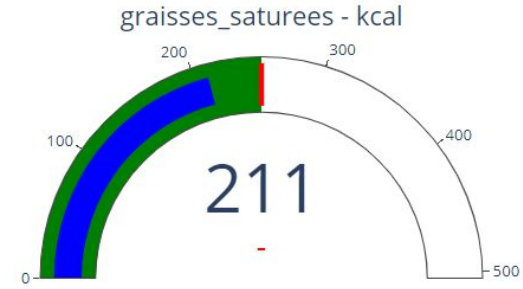
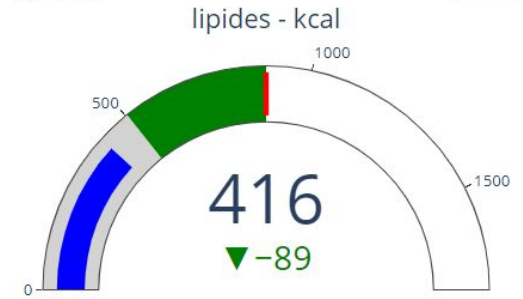
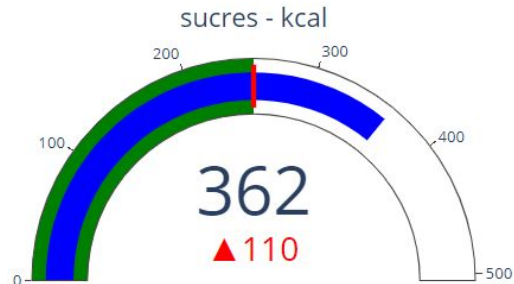
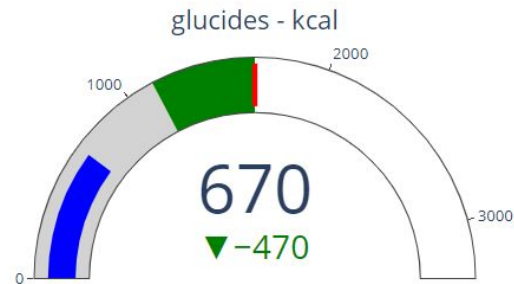
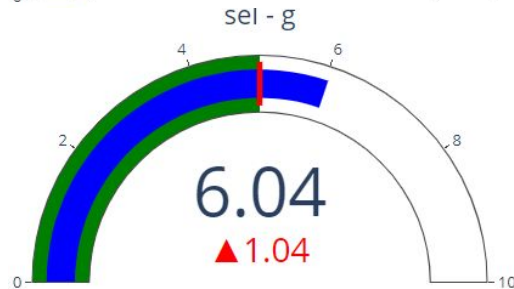
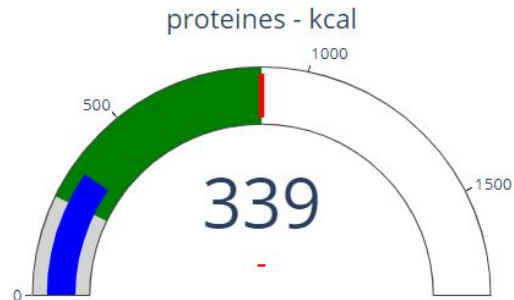
kcal	kcal_fibres	kcal_lipides	kcal_graisses_saturees	kcal_glucides	kcal_sucres	kcal_proteines	sel
2035.325	34.348	415.773	210.717	667.120001	361.965999	338.706	6.042



8. Prototypage de l'Application

Visualisations :

- Les jauges permettent une lecture facile des calories consommées et leur répartitions



9. Pertinence & Faisabilité

Compléments au Nutriscore :

Seulement, comment choisir entre 2 produit ayant le même nutriscore ?

```
maxContrib(data,df_ctrInd,'3017620422003')#nutella
```

Contribue le plus à l'axe : 1
C'est le 57594 ième produit participant le plus à cet axe,
il fait partie des 30.0 % des produits participant le plus à l'axe,
(195205 produits participant plus que la moyenne)

```
maxContrib(data,df_ctrInd,'3330720237361')#pate a tartiner du Lot et garonne, sans huile
```

Contribue le plus à l'axe : 1
C'est le 48267 ième produit participant le plus à cet axe,
il fait partie des 25.0 % des produits participant le plus à l'axe,
(195205 produits participant plus que la moyenne)

```
maxContrib(data,df_ctrInd,'20968663')#pate a tartiner bio, sans huile de palme
```

Contribue le plus à l'axe : 1
C'est le 77140 ième produit participant le plus à cet axe,
il fait partie des 40.0 % des produits participant le plus à l'axe,
(195205 produits participant plus que la moyenne)

Pâtes à tartiner de nutriscore D :

On remarques que les pâtes à tartiner contribuent le plus à l'Axe 1,
qui est représentatif de la quantité de graisses dans le produit.

Classement suivant la contribution à l'Axe 1:

1. Pâte à tartiner du Lot et Garonne, sans huile de palme
2. Nutella
3. Pâte à tartiner Bio, sans huile de palme

La contribution à l'axe peut donc être un un indicateur intéressant pour différencier deux produits similaires.

Par contre, cette méthode à ses limites :

- L'ACP présente seulement 3 axes qui ne représentent pas la totalité des variables des nutriments
- Si on compare deux produits qui ne contribuent pas plus que la moyenne à un axe on ne pourras pas en tirer plus de conclusion

10. Conclusion

On dispose de tous les éléments nous permettant de réaliser l'application

Le but est que le consommateur puisse comprendre comment équilibrer son alimentation en prenant en considération le contenu des produits qu'il consomme sans qu'il ait besoin de lire systématiquement les tableaux des nutriments.

L'idée est aussi de pouvoir prévenir le consommateur lorsqu'il ajoute un produit si il dépasse les recommandations afin qu'il puisse choisir un autre aliment.

Concernant les données d'Open Food Facts, on a remarqué que l'on a du nettoyer beaucoup de données. En effet, le fait que cette base de données soit collaborative elle permet à tout le monde de pouvoir la compléter. C'est un avantage car cela permet d'avoir une maximum de produits en base de données, par contre il y a beaucoup d'erreur dans ces données, ce qui n'aide pas pour avoir une application précise. Par contre, on peut imaginer que comme la base de données est collaborative plus les gens l'utiliseront plus les erreurs vont être corrigées.

Pour pallier à ces erreurs on pourrait aussi intégrer une fonctionnalité permettant d'ajouter et de corriger les informations nutritionnelles des produits.