

Concevez une application au service de la santé publique

Etudiant: Etienne NGENZI
Mentor : Julia Wabant

January 27, 2023

Jeux de données

La qualité de jeu de données

Importation des données

Chargeons les données.

```
: data.head()  
:
```

	code	url	creator	created_t	created_datetime	last_modified_t	last_modified_datetime	product_name	generic_name
0	3087	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
1	4530	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
2	4559	http://world-fr.openfoodfacts.org/produit/0000...	usda-ndb-import	1489069957	2017-03-09T14:32:37Z	1489069957	2017-03-09T14:32:37Z	Peanuts	NaN
3	16087	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
4	16094	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

5 rows x 162 columns

- 1 Le fichier contient 320772 rangées:
- 2 et 162 colonnes
- 3 Le fichier utilise la mémoire de 396.5+ MB

La qualité de jeu de données

Distribution statistique des données

	no_nutriments	additives_n	ingredients_from_palm_oil_n	ingredients_from_palm_oil	ingredients_that_may_be_from_palm_oil_n	ingredients_that_may_be
count	0.0	248939.000000	248939.000000	0.0	248939.000000	
mean	NaN	1.936024	0.019659	NaN	0.055246	
std	NaN	2.502019	0.140524	NaN	0.269207	
min	NaN	0.000000	0.000000	NaN	0.000000	
25%	NaN	0.000000	0.000000	NaN	0.000000	
50%	NaN	1.000000	0.000000	NaN	0.000000	
75%	NaN	3.000000	0.000000	NaN	0.000000	
max	NaN	31.000000	2.000000	NaN	6.000000	

8 rows × 106 columns

- 1 A partir de cette indication globale, on voit clairement que le fichier contient beaucoup de défauts.
- 2 cela peut être aussi confirmer en regardant le skweness des données

code	541.194383
no_nutriments	NaN
additives_n	2.175374
ingredients_from_palm_oil_n	7.174753
ingredients_from_palm_oil	NaN
	...
carbon-footprint_100g	2.771240
nutrition-score-fr_100g	0.114836
nutrition-score-uk_100g	0.132006
glycemic-index_100g	NaN
water-hardness_100g	NaN
Length: 107, dtype: float64	

Les défauts dans les données

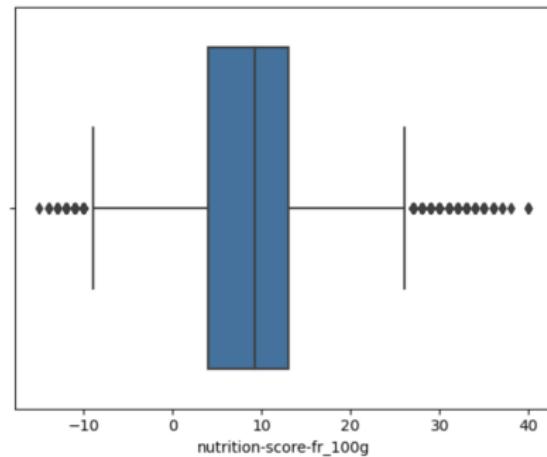
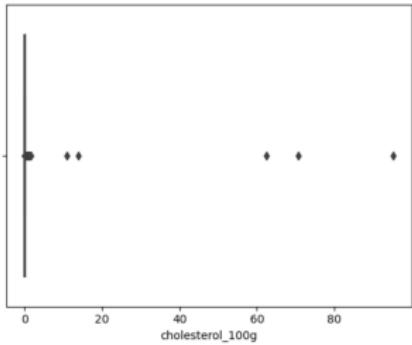
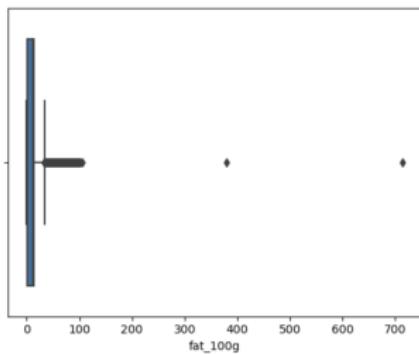
- 1 Le fichier contient de nombreuses valeurs manquantes mais pas de doublons.
- 2 Ici, nous montrons la moyenne des valeurs manquantes par colonne

code	0.000072
url	0.000072
creator	0.000006
created_t	0.000009
created_datetime	0.000028
	...
carbon-footprint_100g	0.999165
nutrition-score-fr_100g	0.310382
nutrition-score-uk_100g	0.310382
glycemic-index_100g	1.000000
water-hardness_100g	1.000000

- 3 Non seulement les valeurs manquantes mais le fichier contient également les valeurs aberrantes.

La qualité de jeu de données

Les valeurs aberrantes



- 1 Ici, nous pouvons voir que les variables contiennent beaucoup de valeurs aberrantes.
- 2 Cela nécessite de nettoyer le fichier pour ne rester qu'avec les colonnes très pertinentes

Nettoyage des données

Nous avons utilisé plusieurs méthodes de nettoyage du fichier.

- 1** Nous avons commencé par supprimer les colonnes non pertinentes pour notre étude.

- 2** Pour traiter les valeurs manquantes nous avons au moins utilisé quatre méthodes:
 - Remplaçons les valeurs manquantes par la moyenne
 - Remplaçons les valeurs manquantes par la médiane
 - Remplaçons les valeurs manquantes par le mode
 - Remplaçons les valeurs manquantes par la zero
 - effaçons les valeurs manquantes

- 3** Pour identifier et traiter les valeurs aberrantes, nous avons utilisé au moins deux méthodes que je montrerai en détail dans ce qui suit.

La qualité de jeu de données

Une nouvelle distribution statisque des données

Le nouveau fichier contient maintenant:

1 Le nombre de rangées= 320772

2 Le nombre de colonnes= 25

	energy_100g	fat_100g	cholesterol_100g	sugars_100g	fiber_100g
count	3.207720e+05	320772.000000	320772.000000	320772.000000	320772.000000
mean	1.141915e+03	12.730379	0.020071	16.003484	2.862111
std	5.816797e+03	15.328066	0.239980	19.511686	10.182936
min	0.000000e+00	0.000000	0.000000	-17.860000	-6.700000
25%	5.020000e+02	0.900000	0.000000	2.500000	0.800000
50%	1.141915e+03	12.730379	0.020071	13.330000	2.862111
75%	1.569000e+03	14.290000	0.020071	16.003484	2.862111
max	3.251373e+06	714.290000	95.238000	3520.000000	5380.000000

8 rows x 22 columns

Valeur manquante par colonne:

energy_100g	0
fat_100g	0
cholesterol_100g	0
sugars_100g	0
fiber_100g	0
proteins_100g	0
sodium_100g	0
vitamin-a_100g	0
vitamin-d_100g	0
vitamin-e_100g	0
vitamin-k_100g	0
vitamin-c_100g	0
vitamin-b1_100g	0
vitamin-b2_100g	0
vitamin-b6_100g	0
vitamin-b9_100g	0
vitamin-b12_100g	0
potassium_100g	0
calcium_100g	0
phosphorus_100g	0
iron_100g	0
nutrition-score-fr_100g	0

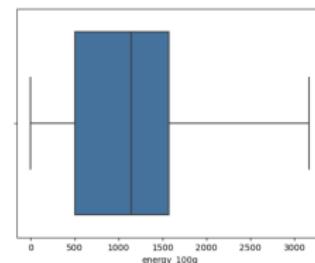
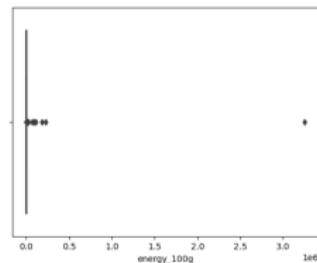
dtype: int64

Ici, on a traité que les valeurs manquantes, maintenant on va traiter les valeurs aberrantes.

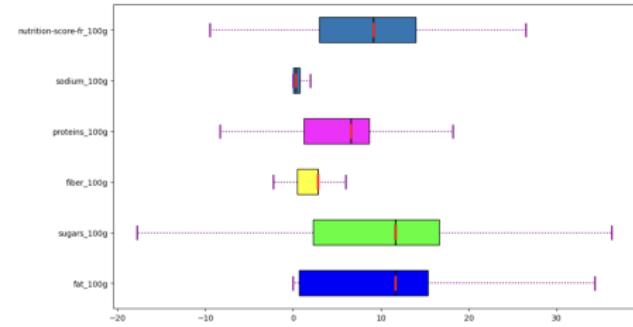
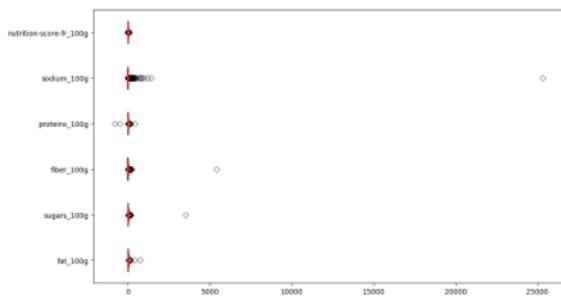
La qualité de jeu de données

Traitons les valeurs aberrantes

Ici nous avons utilisé au moins deux méthodes pour identifier les valeurs aberrantes (méthodes visuelles et méthodes statistiques basées sur les quartiles et la déviation standard):



Ici nous comparons les fichiers contenant les valeurs aberrantes et l'autre nettoyé (sans valeurs aberrantes)



La qualité de jeu de données

Passons en revue l'amélioration de la distribution statistique

Maintenant, après que tous les défauts ont été traités, nous constatons une forte amélioration de la distribution statistique des variables

Distribution statistique des données

	energy_100g	fat_100g	cholesterol_100g	sugars_100g	fiber_1
count	302759.000000	302759.000000	302759.000000	302759.000000	302759.000000
mean	1122.924925	11.239035	0.016373	13.257899	2.365
std	723.953154	10.925574	0.013734	12.251439	1.803
min	0.000000	0.000000	0.000000	-17.755225	-2.293
25%	466.000000	0.700000	0.000000	2.300000	0.500
50%	1141.914605	11.670000	0.020071	11.670000	2.862
75%	1594.000000	15.380000	0.020071	16.670000	2.862
max	3169.500000	34.375000	0.050178	36.258709	5.958

8 rows x 22 columns

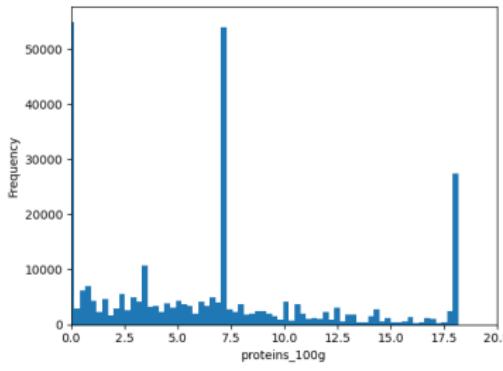
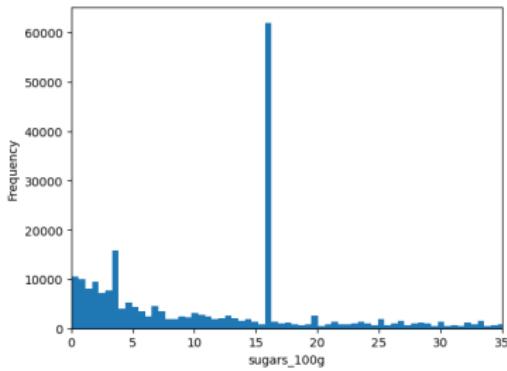
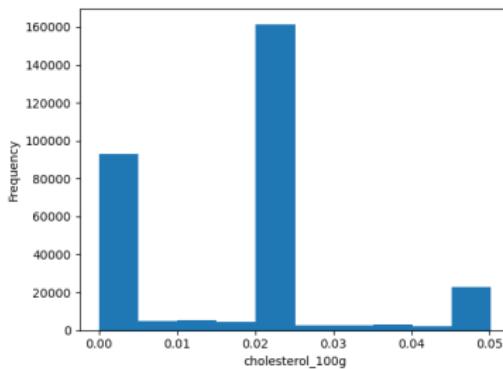
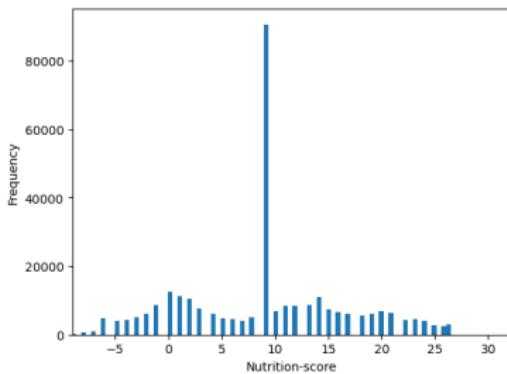
Reverfions le skewness des variables

energy_100g	0.375593
fat_100g	0.804812
cholesterol_100g	0.714842
sugars_100g	0.715821
fiber_100g	0.351027
proteins_100g	0.753593
sodium_100g	1.300894
vitamin-a_100g	0.224577
vitamin-d_100g	0.000000
vitamin-e_100g	0.000000
vitamin-k_100g	0.000000
vitamin-c_100g	0.229787
vitamin-b1_100g	0.000000
vitamin-b2_100g	0.000000
vitamin-b6_100g	0.000000
vitamin-b9_100g	0.000000
vitamin-b12_100g	0.000000
potassium_100g	0.000000

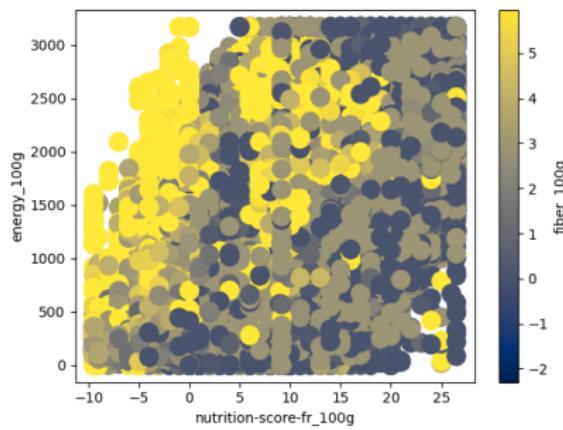
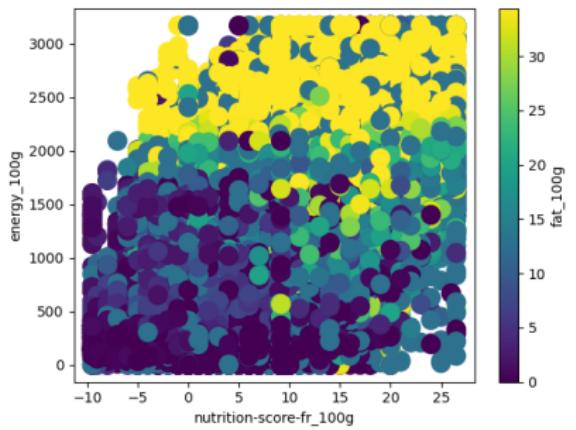
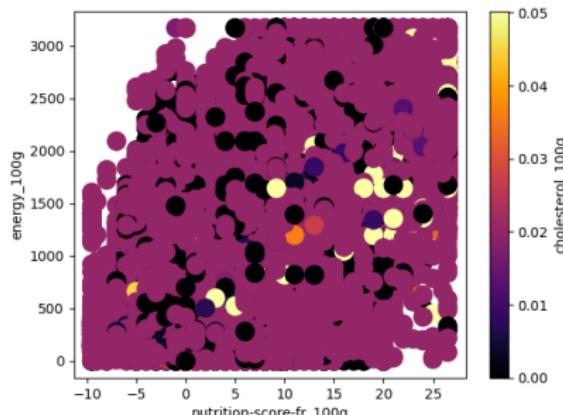
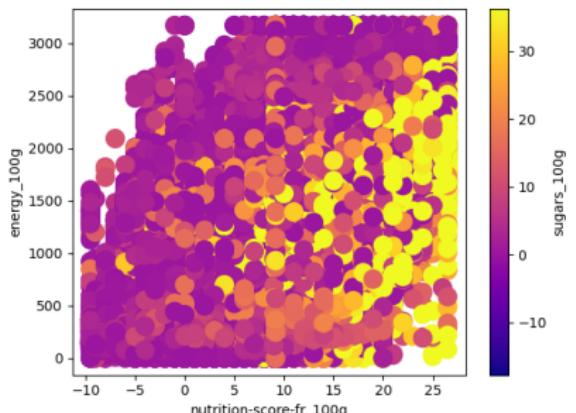
=> Nous pouvons maintenant passer à l'analyse des variables pertinentes.

La qualité de jeu de données

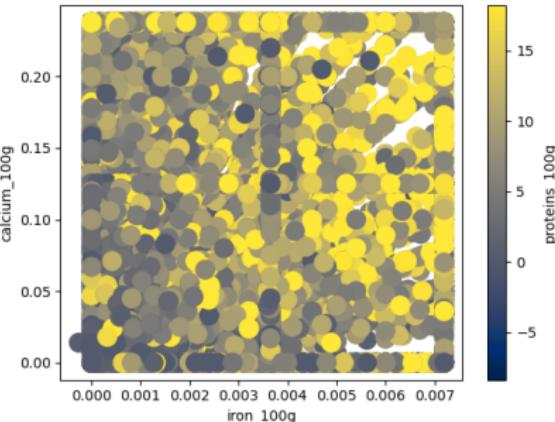
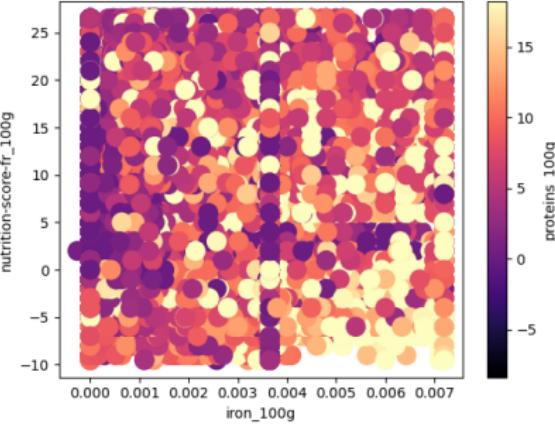
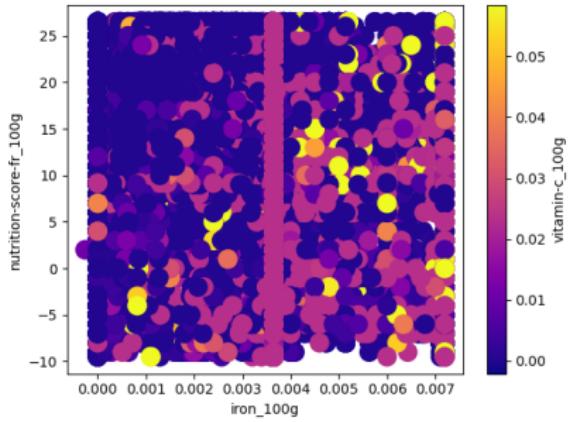
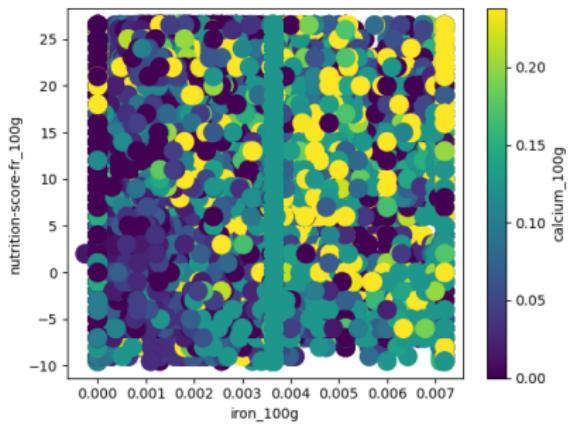
Histogrammes des variables représentatifs



La qualité de jeu de données



La qualité de jeu de données



Application

Scanner un produit

- 1 Le code commence par scanner un produit
- 2 Deuxièmement, il vous indique si le produit du scanner est bon pour lutter contre les maladies cardiaques

```
In [104]: Nutriscore=Data_exploration.Scanning_product(Data_Mean,4530)
```

The product has nutriscore value of = 14.0

2. Grouping data set based on nutriscore

```
In [105]: Grouped_data=Data_exploration.Grouping_data(Data_Mean)
```

3. Get suggestion of products similar to the scanned item

```
In [*]: Nutriscore_data=Data_exploration.Get_similar_product(Data_Mean,Nutriscore)
```

Vous avez choisi un bon produit contre les maladies cardiaques

Veuillez entrer le nombre d'articles similaires que vous souhaitez voir:

- 3 Troisièmement, il vous demande combien de produits similaires vous souhaitez voir.
- 4 Dans la dernière étape, il vous donne la liste.

Fonctionnement

Get suggestion of products similar to the scanned item

- Voici un exemple des produit de suggestion avec le taux de **cholesterol=0g et le nutrition score de 14**.

In [106]: `Nutriscore_data=Data_exploration.Get_similar_product(Data_Mean,Nutriscore)`

Veuillez entrer le nombre d'articles similaires que vous souhaitez voir:5

index	code	product_name	countries_fr	energy_100g	fat_100g	cholesterol_100g	sugars_100g	fiber_100g	proteins_100g	vitamin-b1_100g	vitamin-b2_100g	vitamin-b6_100g	vitamin-b9_100g	
0	80	33688	Peanuts, Mixed Nuts	2389.0	34.375	0.0	14.290000	5.955277	18.20	...	0.325574	0.259007	0.023378	0.00689
1	208	20046576	Freshly Baked Apple Pie	1155.0	13.820	0.0	23.580000	1.600000	1.63	...	0.325574	0.259007	0.023378	0.00689
2	364	790350187	Hib Candy, Jelly Fish Candy	1464.0	0.000	0.0	36.258709	0.000000	0.00	...	0.325574	0.259007	0.023378	0.00689
3	373	790400363	Freeze-Dried Grapes	1569.0	0.000	0.0	36.258709	0.000000	0.00	...	0.325574	0.259007	0.023378	0.00689
4	395	790520603	Fruit Snacks, Cherry, Lemon, Raspberry, Apple,...	1360.0	0.000	0.0	36.258709	0.000000	5.00	...	0.325574	0.259007	0.023378	0.00689

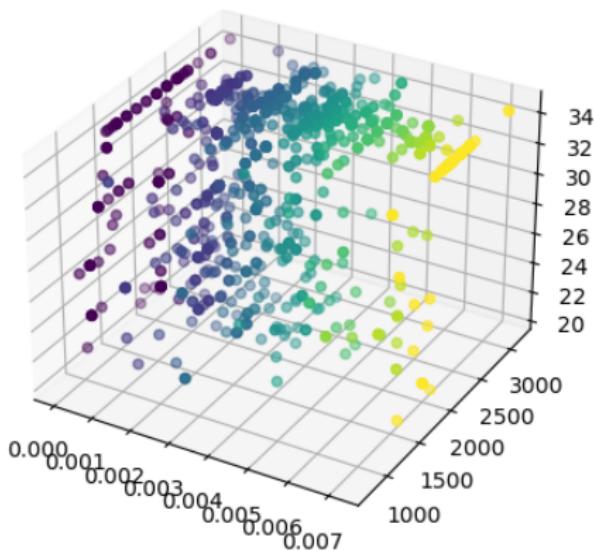
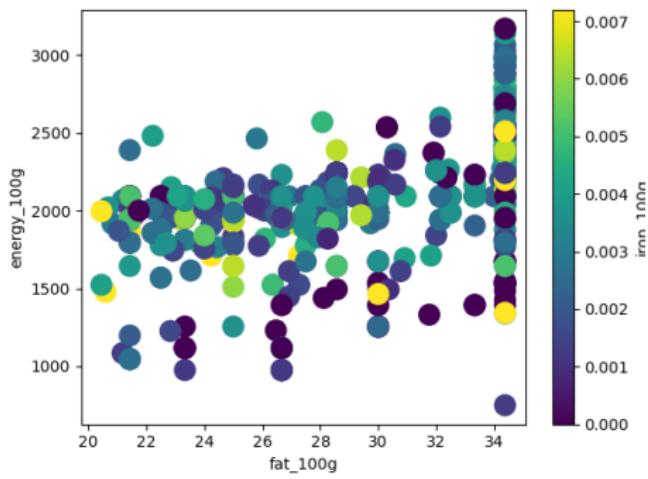
- Après avoir vu que le code fonctionne, nous commençons maintenant à travailler sur les données qu'il génère.
- On va se baser sur exploration des données cleanée comportant une analyse univariée, multivariée et une réduction dimensionnelle

Machine Learning

Regression linéaire

Projection des données

Mêmes données tracées en 3D

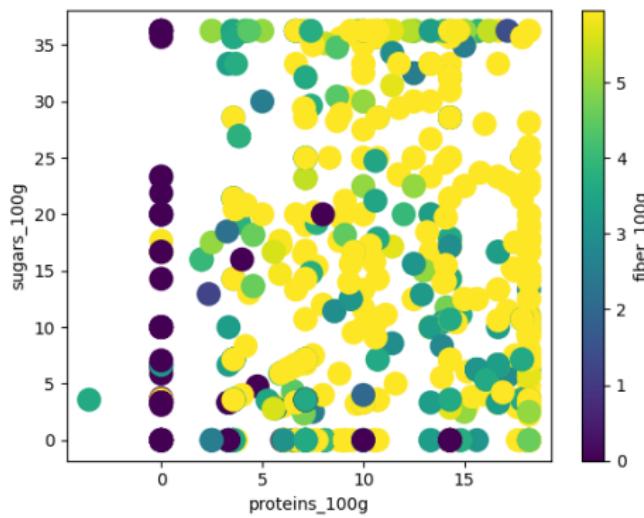
Energie fonction de graisse, coloré par la teneur en fer

- On voit bien que en 3 dimensions, on peut distinguer 3 couche en se referant sur la teneur en Fer

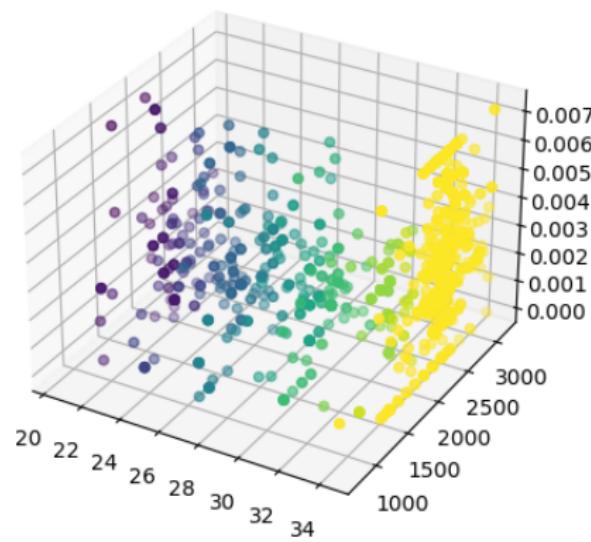
Regression linéaire

Projection des données

Energie fonction de graise, coloré par la teneur en fer



Mêmes données tracées en 3D

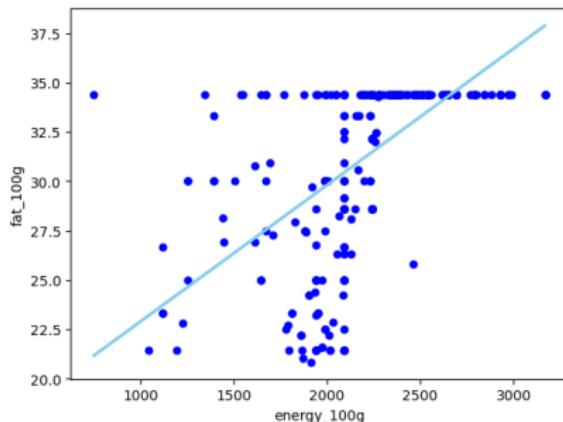
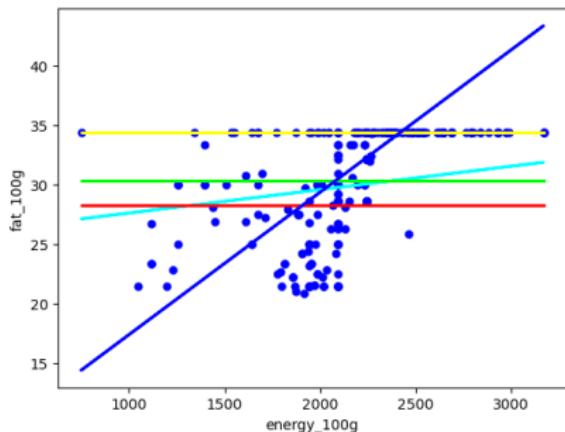


- On voit bien que en 3dimensions, on peut distinguer 3 couche en se référant sur la teneur en Fer

Regression linéaire

Fitter les données à l'aide de la régression linéaire

- On va effectuer une régression spécifique sur chaque variable
- Pour avoir plus de précision. D'abord, on sépare jeu de données d'entraînement et jeu de données test



- On effectue la prédiction finale sur le jeu de donnée test avec notre nouveau modèle.

Mise à l'échelle des données

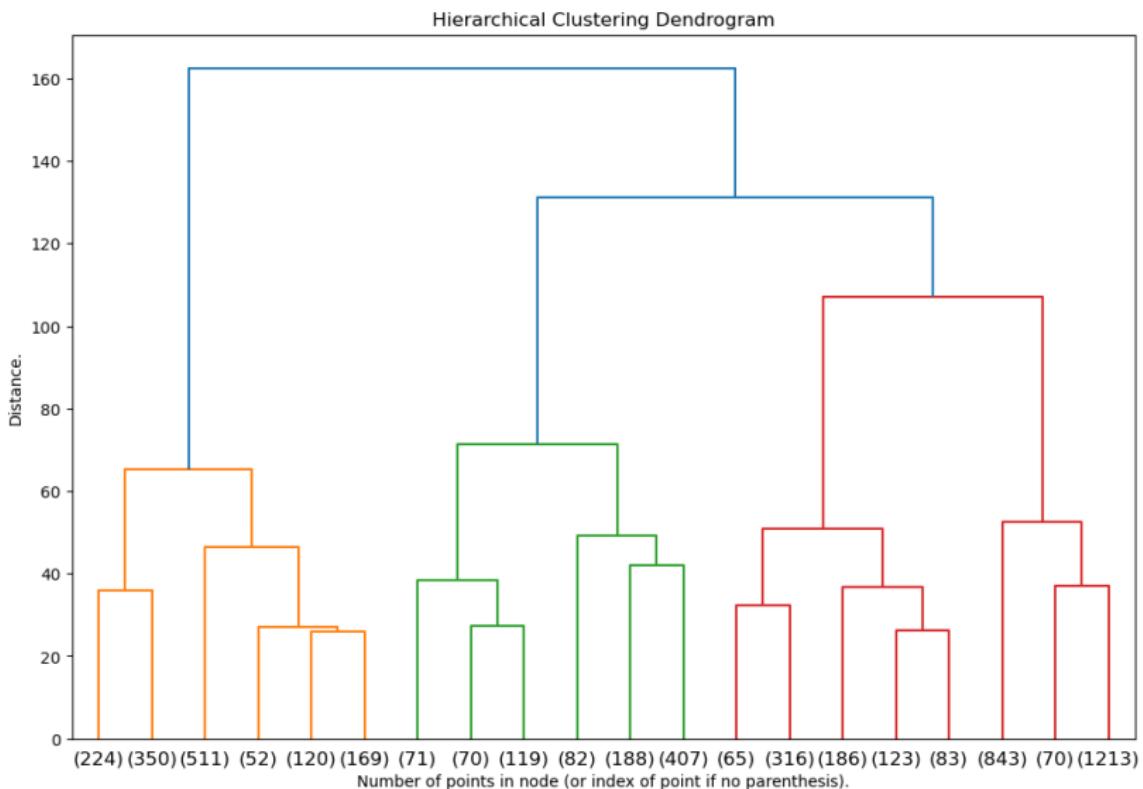
- La Classification Ascendante Hiérarchique (CAH). c'est un algorithme non supervisé très connu en matière de Clustering.
- En premier lieu, nous avons commencé par scaler nos données.

	0	1	2	3	4	5	6	7	8	9	...	13	14	15	16	17	18	19	20	21	22	
mean	-0.0	0.0	-0.0	0.0	0.0	0.0	-0.0	0.0	0.0	0.0	...	0.0	0.0	-0.0	-0.0	0.0	0.0	-0.0	-0.0	-0.0	0.0	
std	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0

2 rows × 23 columns

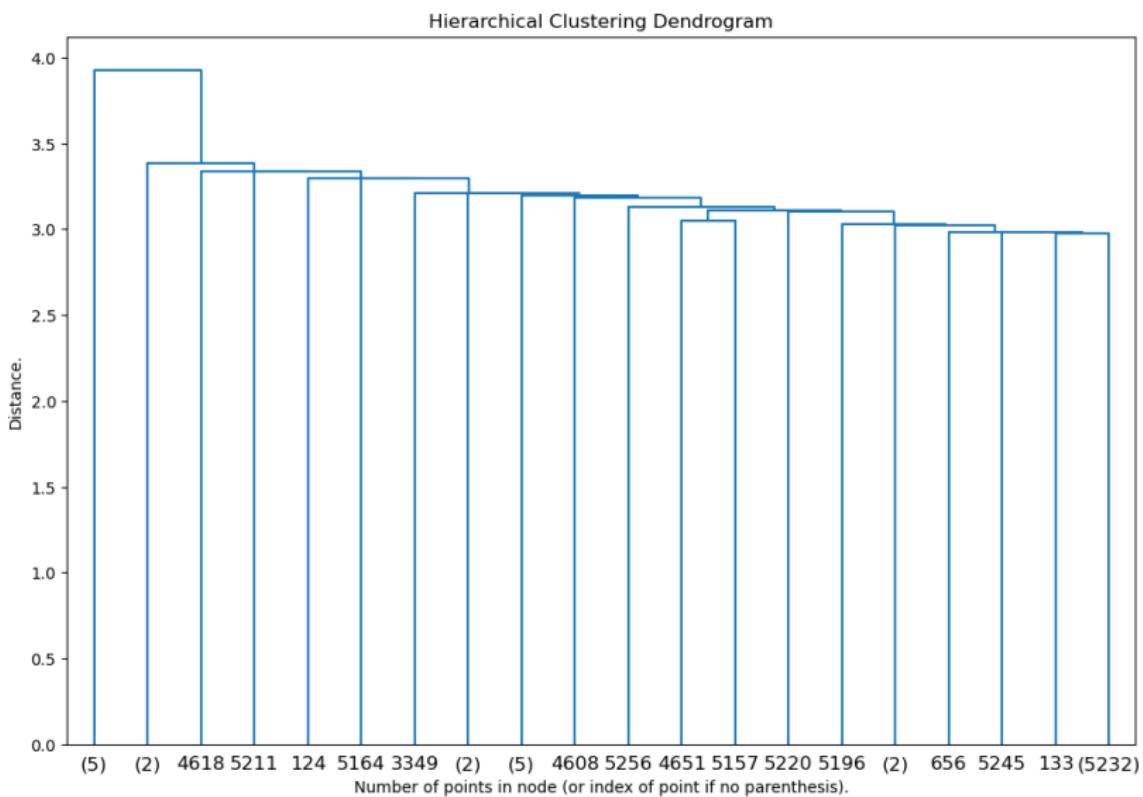
- La standardisation d'un ensemble de données est une exigence commune pour de nombreux estimateurs d'apprentissage automatique.
- Ils peuvent mal se comporter si les caractéristiques individuelles ne ressemblent pas plus ou moins à des données standard normalement distribuées (par exemple, gaussiennes avec une moyenne de 0 et une variance unitaire).

Classification par la méthode de Ward



CAH

Classification par la méthode de Single



Clusters

- Nous avons calculé les distances en utilisant la méthode de Ward

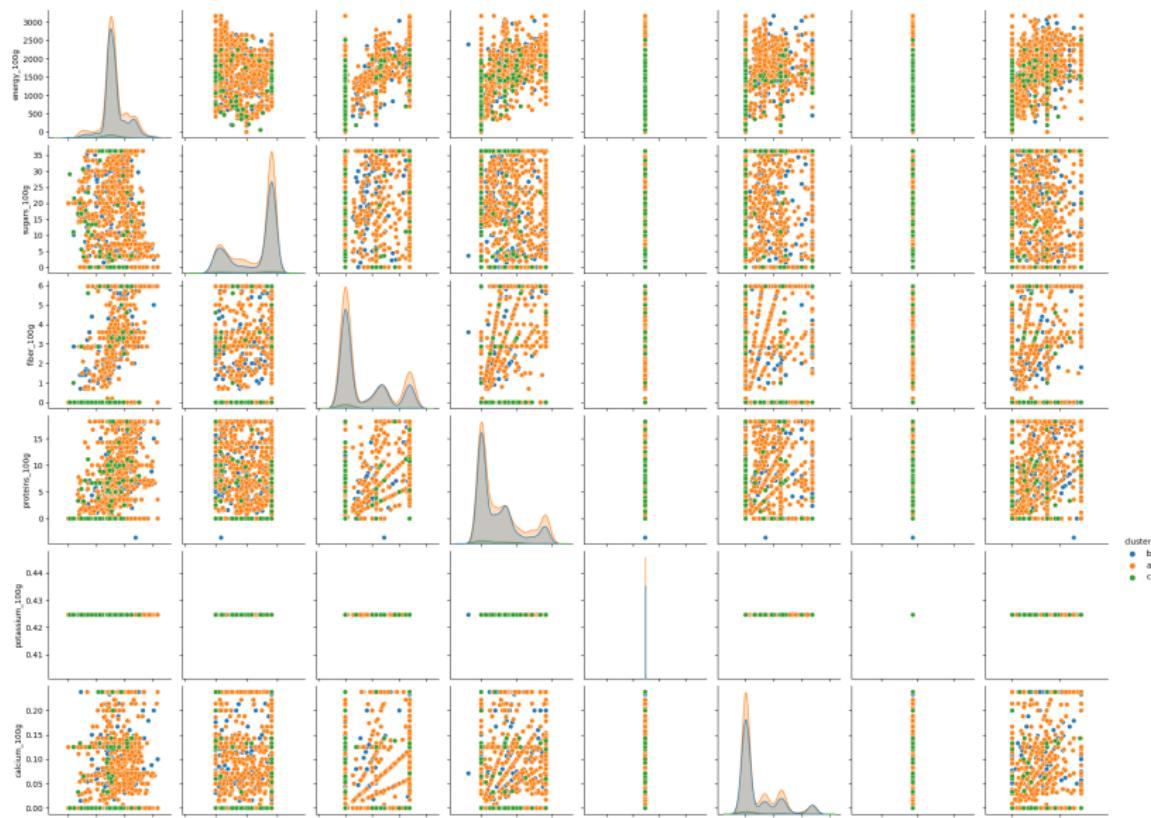
```
In [132]: _CAH_.printing_class_(clusters,names,themes)
```

	name	theme	cluster
0	États-Unis	Peanuts, Mixed Nuts	3
1	États-Unis	Freshly Baked Apple Pie	11
2	États-Unis	Hfb Candy, Jelly Fish Candy	8
3	États-Unis	Freeze-Dried Grapes	13
4	États-Unis	Fruit Snacks, Cherry, Lemon, Raspberry, Apple,...	9
	cluster : 1		
	États-Unis (Leila Bay Trading Co., Apple Cinnamon Granola, Apple, Cinnamon) / États-Unis (Leila Bay Trading Co., Banana Nut Granola) / États-Unis (Leila Bay Trading Co., Cranberry Almond Granola) / États-Unis (Leila Bay Trading Co., Chocolate Huckleberry Quinoa Granola) / États-Unis (Rare Fare Foods, Fruit & Seed Blend) / États-Unis (Classic Trail Mix) / États-Unis (Greek Yogurt Protein Chewy Bars, Mixed Berry) / États-Unis (Raisin Bran, Whole Grain Wheat And Bran Cereal With Raisins) / États-Unis (Pudding & Pie Filling, Chocolate Fudge) / États-Unis (Trail Mix) / États-Unis (Berry Chocolate Trail Mix) / États-Unis (Holiday Gingerbread Cookies) / États-Unis (Chewy Protein Bars) / États-Unis (Toffee Peanuts) / États-Unis (Yin & Yang Peanuts) / États-Unis (Raincoast Crisps, Fig And Olive Crackers) / États-Unis (Organically Yours, Feelin' Energized Trek Mix) / États-Unis (Chewy Protein Granola Bars) / États-Unis (Chewy Protein Bars) / Etats-Unis (Gluten Free Brownie Mix) / Etats-Unis (Coconut Flour) / Etats-Unis (Maui Wowi Trail Mix) / États-Unis (Peanuts) / États-Unis (Dark Chocolate Cranberry Trail) / États-Unis (Smokey Style Butter Toffee Peanuts) / États-Unis (Butter Toffee Peanuts, Chili-Lime) / États-Unis (Low Fat Chocolate Mousse Mix) / États-Unis (Sans Sucre, Macho Cappuccino Mousse Mix) / États-Unis (Organic Nuts N' Berries Mix With Coconut) / États-Unis (Dark Chocolate Cherry) / Etats-Unis (Nutrition Bar) / Etats-Unis (Nekot, Chocolate Sandwich Cookies, Real Peanut		

- Ici nous montons que voici une capture d'écran pour montrer ce que l'algorithme peut faire.

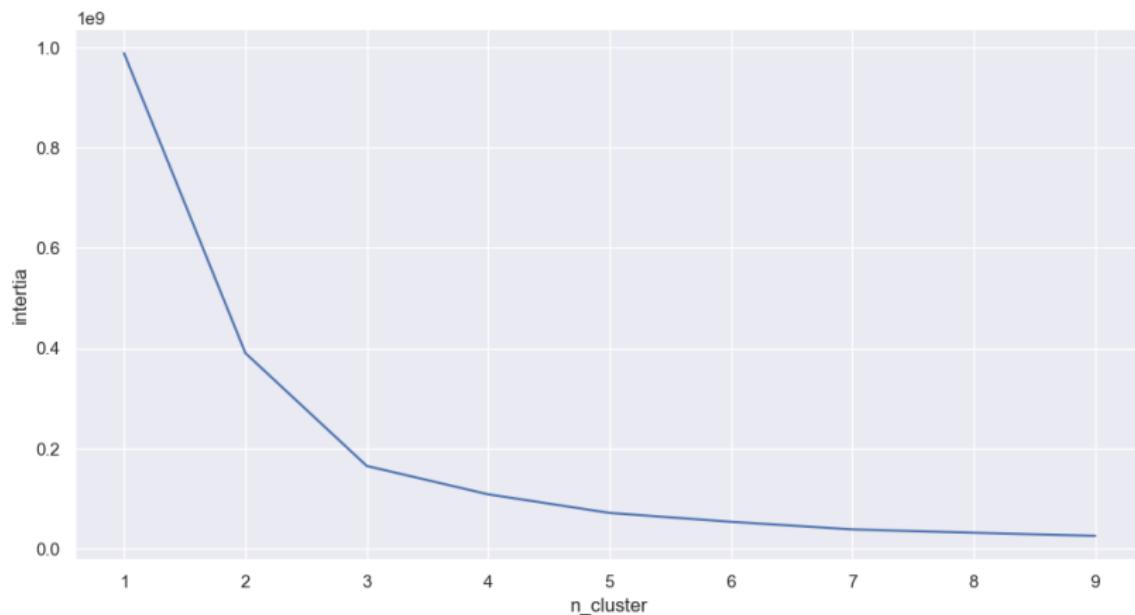
CAH

Clusters



La méthode de Elbow

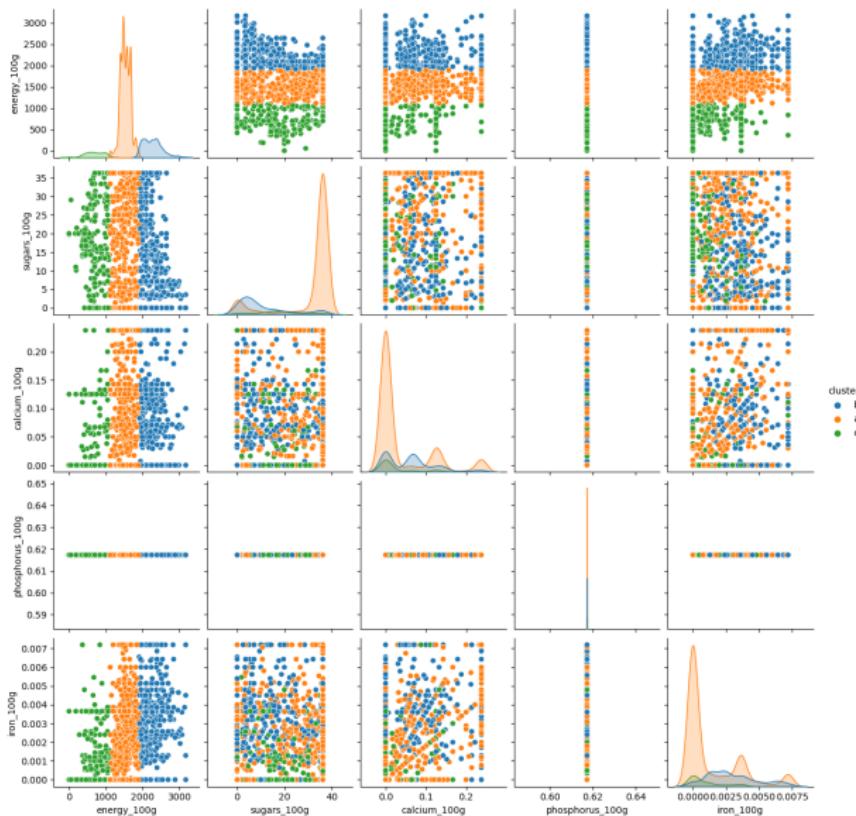
- Pour déterminer le nombre de clusters à étudier, on s'intéresse au graphique qui trace l'inertie intraclasse en fonction du nombre de clusters.



- On cherche plus particulièrement une "cassure" dans la courbe. Cette **cassure** nous indique à partir de quel nombre de clusters nous "allons trop loin".
- Regardons maintenant les clusters que nous avons obtenus.

K-Mean

Les clusters par K mean

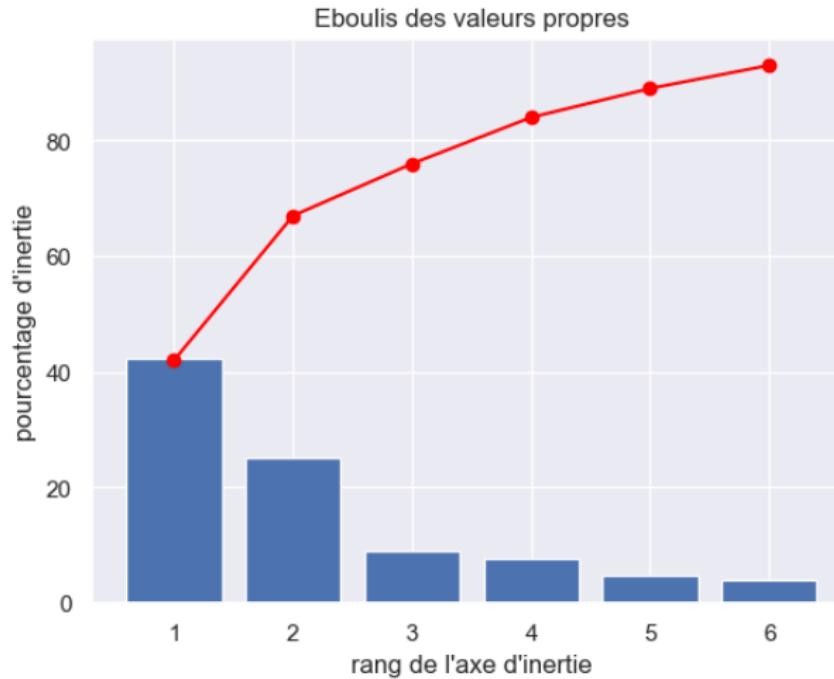


Analyse exploratoire des données

PCA analysis

Eboulis des valeurs propres

- On a en bleu la variance de chaque nouvelle composante

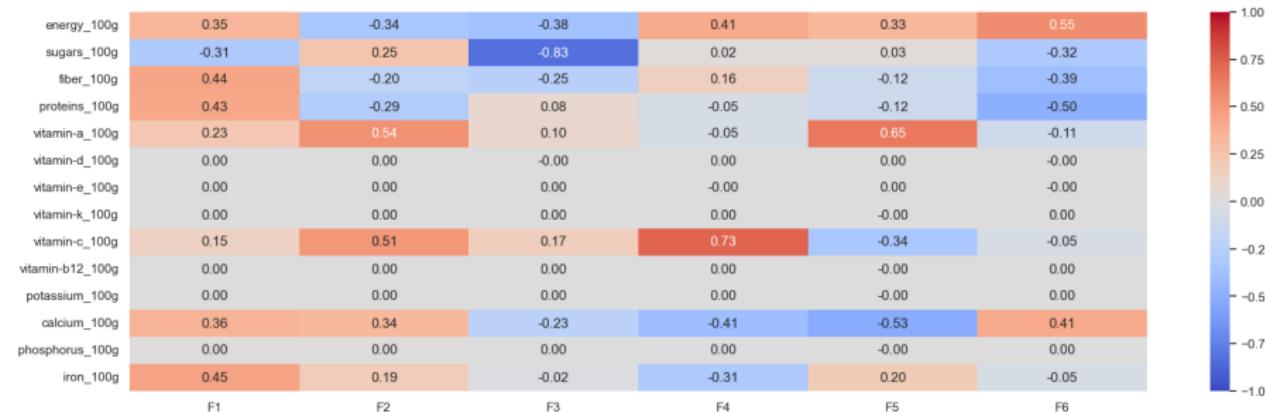


- Et en rouge la variance cumulée.

PCA analysis

Heat map

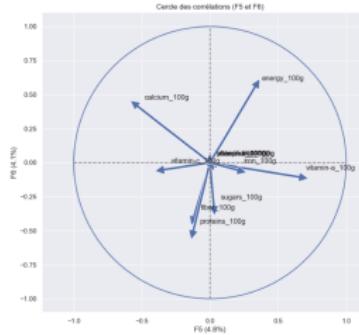
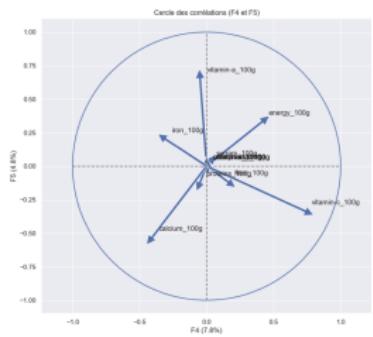
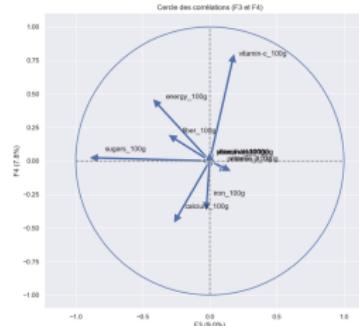
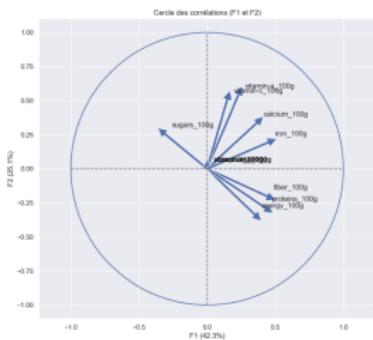
- Pour une représentation plus visuelle des composantes (F1, F2, F3,...,F6)



- On voit ici et dans la diapo précédente que près de 80% de la variance est comprise dans les 3 premières composantes.

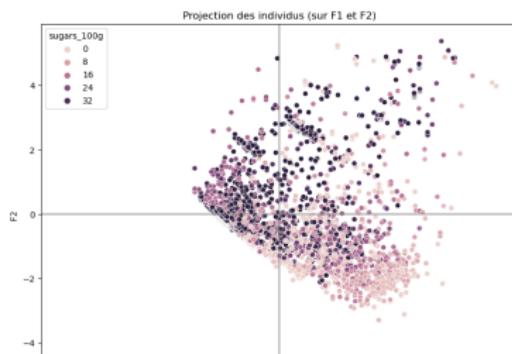
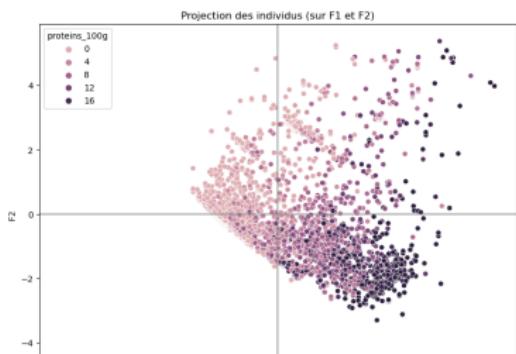
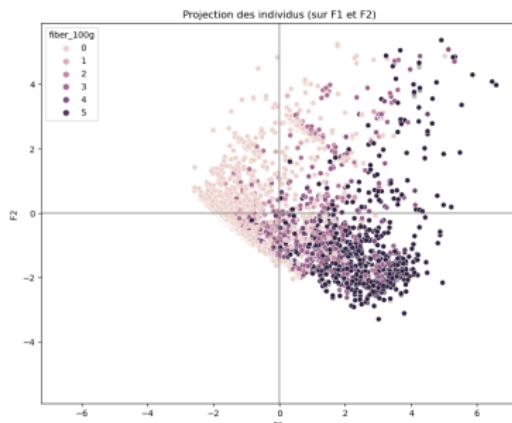
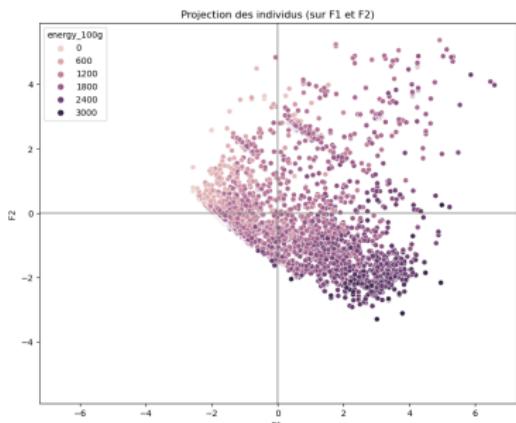
PCA analysis

Cercle de corrélation



PCA analysis

La projection de dimensions



Conclusions

Conclusions

- ① Nous avons chargé les données avec succès.
- ② Les données contiennent de nombreux défauts, nous avons alors procédé au nettoyage du jeu de données
- ③ Nous avons mené avec succès une analyse univariée des variables.
- ④ Après, nous avons mené avec succès une analyse multi-variée.
- ⑤ Nous avons développé une application qui dit avec succès aux clients si le produit est bon pour la santé lorsque le taux de cholestérol est nul.
- ⑥ Nous avons mis en place l'algorithme de machine learning en place sur les données retournée par l'application pour un produit scanné.