2020

OPENCLASSROOMS

Parcours Data-Scientist – Projet 3

CONCEVEZ UNE APPLICATION AU SERVICE DE LA SANTÉ PUBLIQUE

PROBLEMATIQUE

L'agence **Santé publique France** (https://www.santepubliquefrance.fr/) a lancé un appel d'offre d'applications en lien avec l'alimentation.

Vous souhaitez y participer et proposer une idée d'application :

1. Traiter le jeu de données de manière automatisée afin de repérer des variables pertinentes
2. Produire des visualisations. Effectuer une analyse univariée pour chaque variable intéressante.
3. Confirmer ou infirmer les hypothèses à l'aide d'une analyse multivariée.
4. Élaborer une idée d'application.

PRESENTATION DES DONNEES

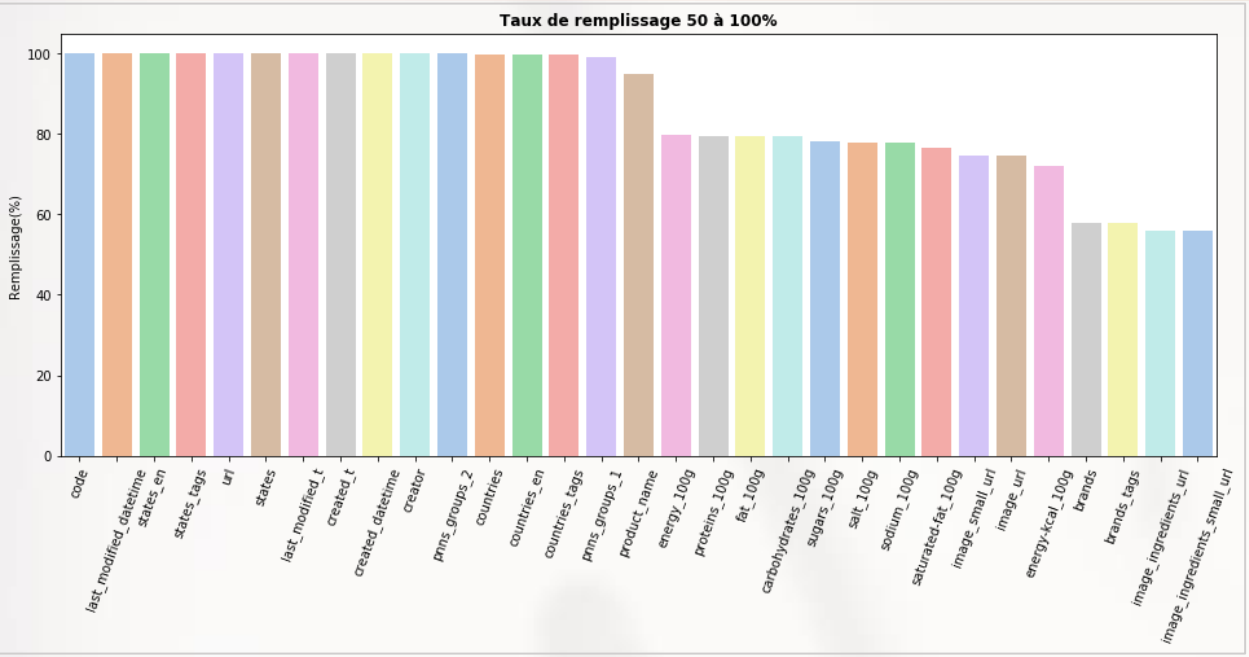
Structure du dataset, d'où proviennent les données ?

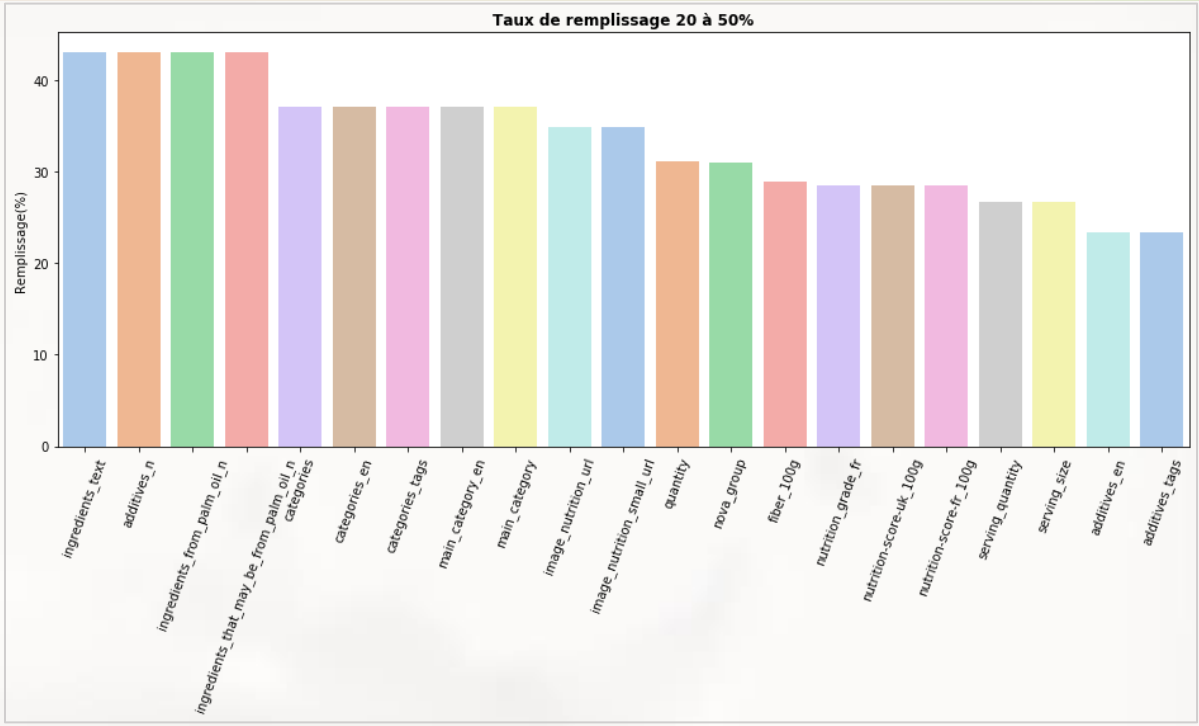
Contenu du dataset, taux de remplissage

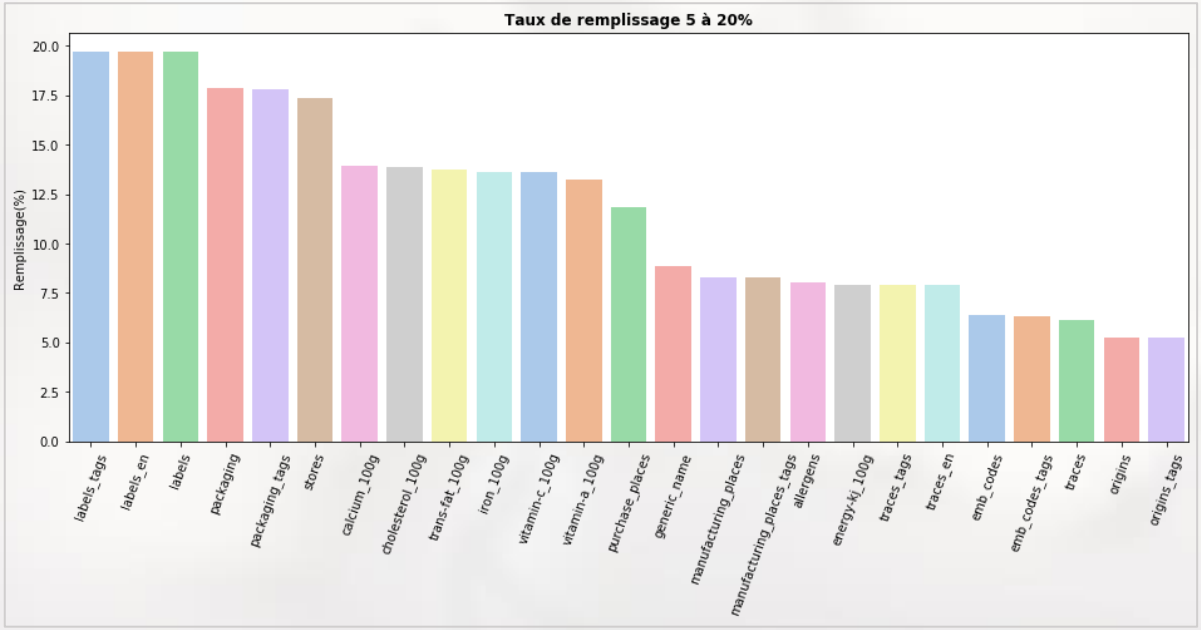
STRUCTURE DU DATASET

- Jeu de données : https://world.openfoodfacts.org/
- Format de fichier : .csv
- Taille : 2,01Go
- Nombre de colonnes : 177
- Nombre de lignes : 1048757

CONTENU DU DATASET





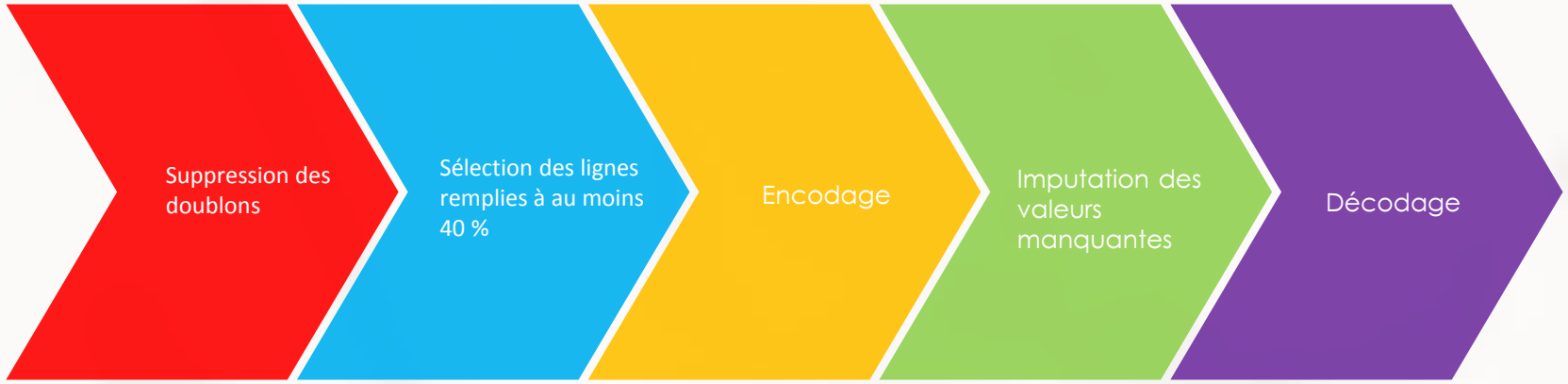


Résumé du taux de remplissage :

- ~ 100 % : mise en ligne, mise à jour, auteur, pays, nom produit
- ~ 80% : valeurs nutritionnelles, URL des photos produits
- ~ 60% : marque, photo des ingrédients
- ~ 40 % : ingrédients, URL des photos des valeurs nutritionnelles
- ~ 30% : nova group, nutriscore, quantité d'une portion
- ~ 20% : additifs, label, emballage (forme, matériaux), magasin
- ~ 15% : vitamines, minéraux, cholestérol, lieu d'achat
- ~ 10% : nom générique, lieu de fabrication, allergènes
- ~ 5% : code-barre, origine

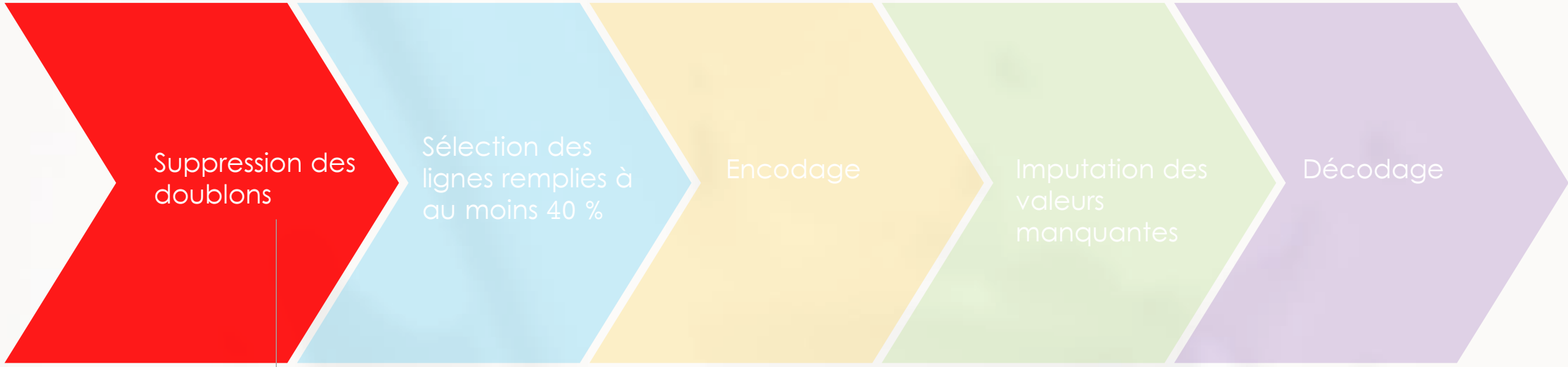
NETTOYAGE DES DONNEES

NETTOYAGE DES DONNEES



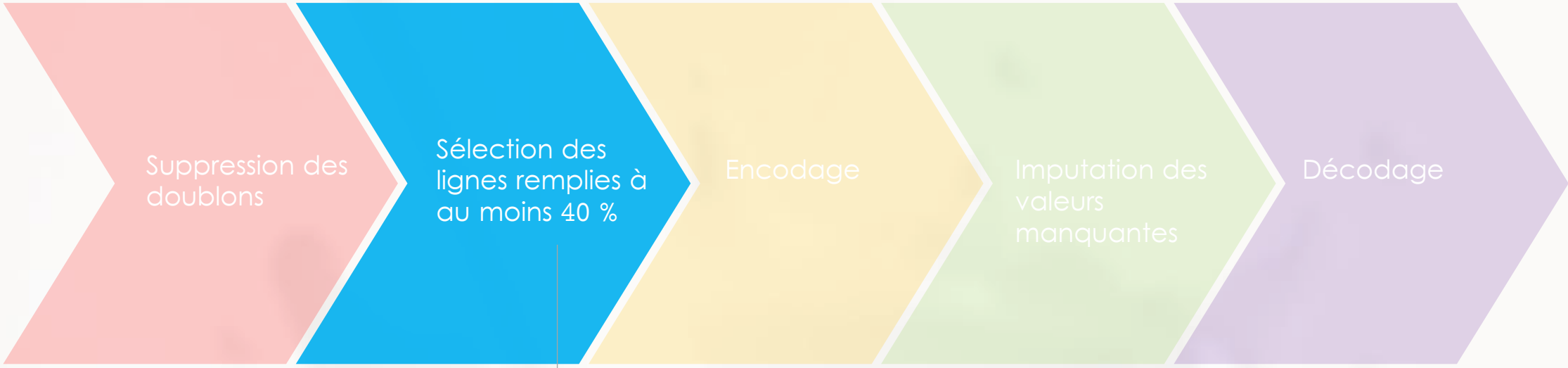
- Processus de nettoyage de données
- 5 étapes principales

NETTOYAGE DES DONNEES



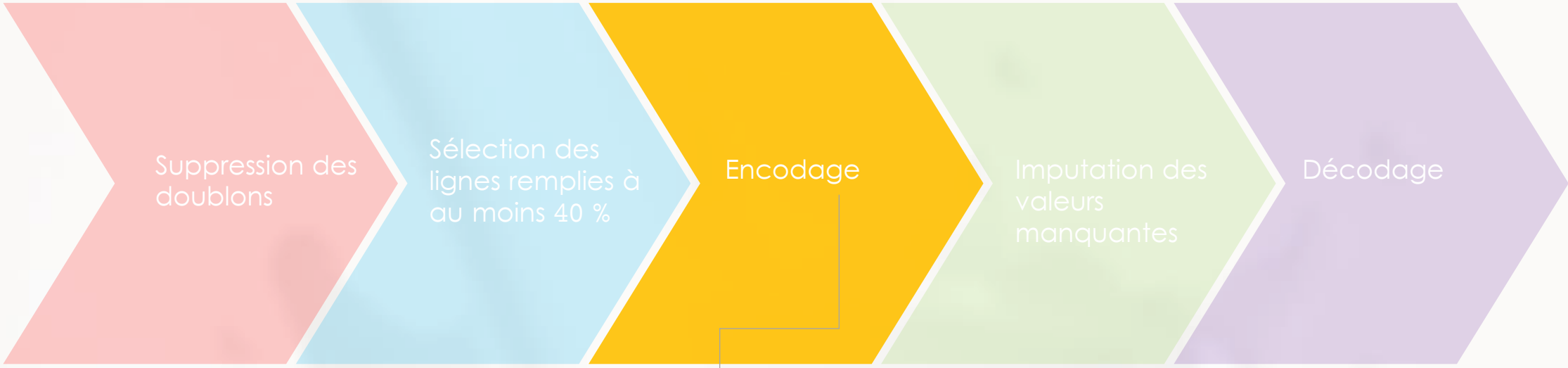

```
# Suppression des doublons :
df = df.drop_duplicates()
```

NETTOYAGE DES DONNEES



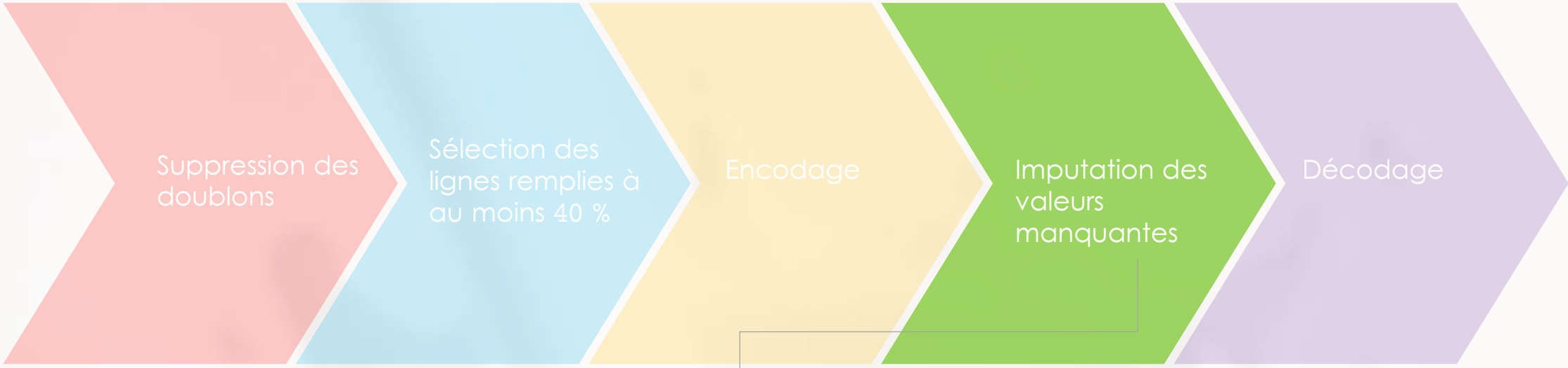

```
# Suppression des lignes dont le ratio de données manquantes dépasse les 40% :
df = df[df['nan_ratio'] > 40]
```

NETTOYAGE DES DONNEES




```
# Remplacement des lettres en valeurs :
mymap = {'a':1, 'b':2, 'c':3, 'd':4, 'e':5}
df['nutrition_grade_fr'] = df['nutrition_grade_fr'].map(lambda s: mymap.get(s) if s in mymap else s)
df.head(5)
```

NETTOYAGE DES DONNEES

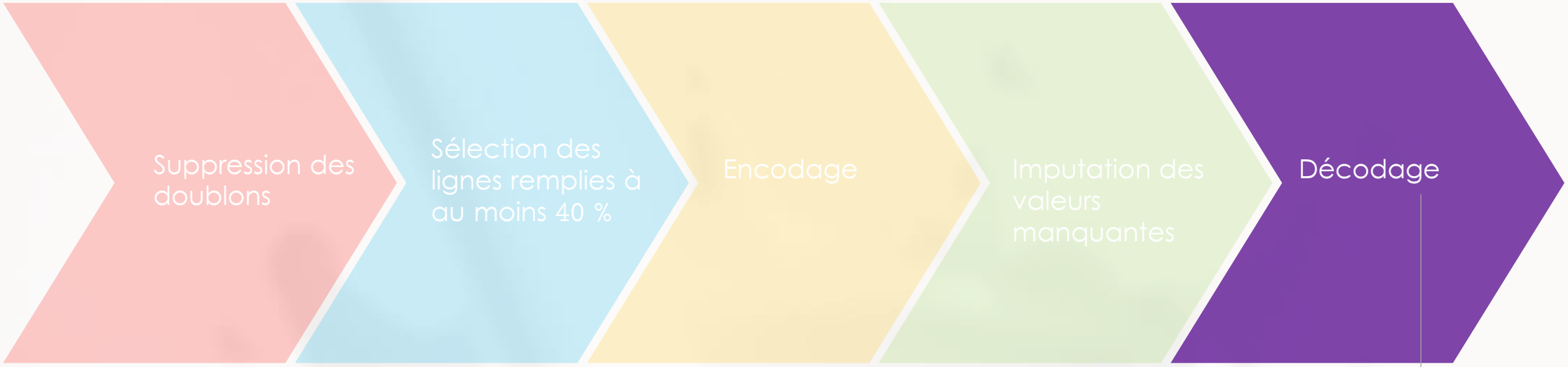



```
# Imputer et réassigner les index/colonnes :
df[cols_knn] = pd.DataFrame(imputer.fit_transform(df[cols_knn]),columns = df[cols_knn].columns)
df.head(5)
```

	energy_100g	proteins_100g	fat_100g
0	NaN	7.8	7.0
1	NaN	NaN	NaN
2	NaN	5.1	8.2
3	NaN	NaN	NaN
4	88.0	0.2	0.0

	energy_100g	proteins_100g	fat_100g
0	4.0	7.8	7.0
1	21.0	45.0	15.0
2	8.0	5.1	8.2
3	21.0	45.0	15.0
4	88.0	0.2	0.0

NETTOYAGE DES DONNEES




```
# Remplacement des lettres en valeurs :
mymap = {1:'a', 2:'b', 3:'c', 4:'d', 5:'e'}
df['nutrition_grade_fr'] = df['nutrition_grade_fr'].map(lambda s: mymap.get(s) if s in mymap else s)
df.head(5)
```

ANALYSE UNIVARIÉE

- Quelques classements
- Des produits Bio ultratransformés !
- Distribution des variables

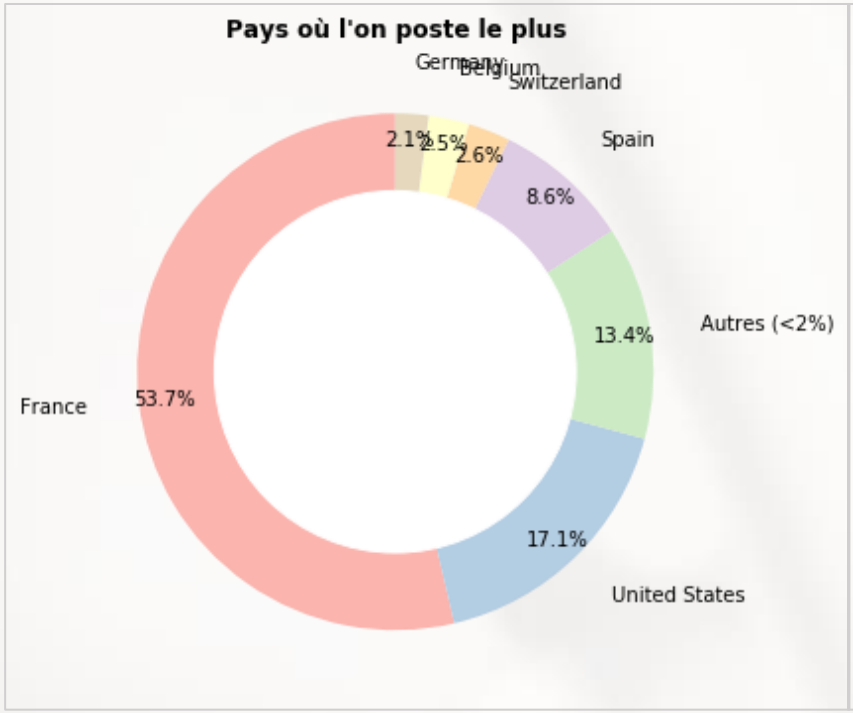
ANALYSE UNIVARIÉE

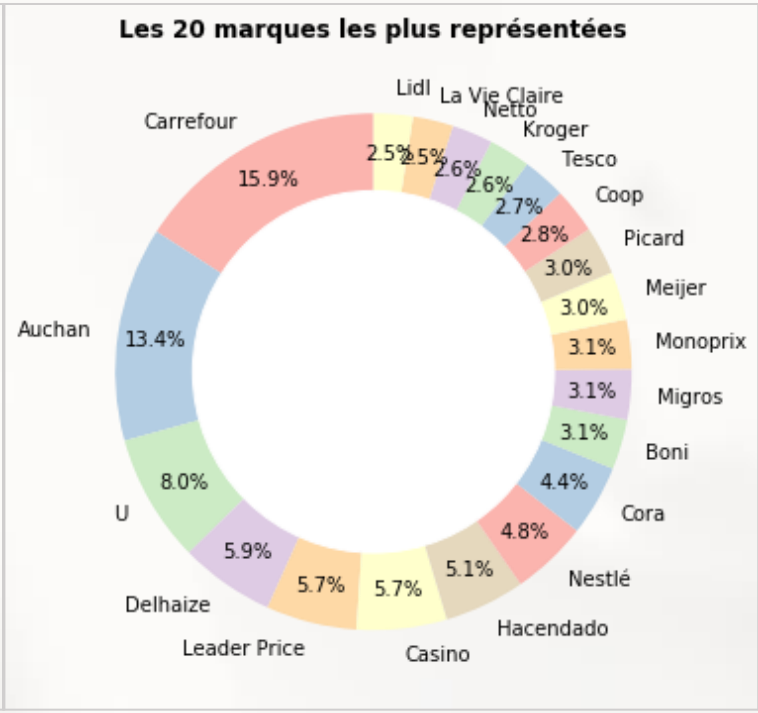
Téléchargement des librairies :

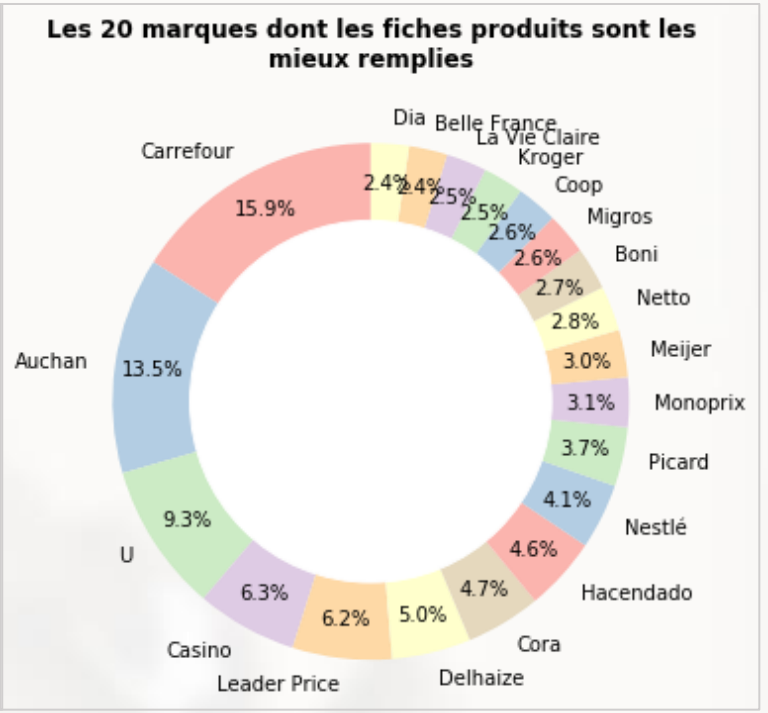
```
# Importation de Pandas
import pandas as pd

# Importation de numpy (utilisé une seule fois pour estimer
import numpy as np

# Importation de la mise en page des titres :
from textwrap import wrap

# Importation de Matplotlib :
import matplotlib.pyplot as plt
%matplotlib inline

# Importation de seaborn :
import seaborn as sns

# Pour les couleurs
from matplotlib import cm
import matplotlib.colors
from matplotlib import colors as mcolors
from palettable.colorbrewer.qualitative import Pastel1_7

# Pour désactiver les alertes :
import warnings

# Importation des étapes de traitement de l'algorythme :

from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import StratifiedKFold
from sklearn.model_selection import GridSearchCV
```

```
# Importation des estimateurs (ici des estimateurs de type classification

from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.svm import LinearSVC
from sklearn.tree import DecisionTreeClassifier

# Importation des calculs de résultats :

from sklearn import preprocessing
from sklearn import metrics

# Importation d'affichage graphique de l'arbre de décision :
from sklearn.tree import DecisionTreeClassifier, export_graphviz
from sklearn import tree
from sklearn.datasets import load_wine
from IPython.display import SVG
from graphviz import Source
from IPython.display import display

# Importation d'affichage des K plus proches voisins :
from mlxtend.plotting import plot_decision_regions
```

ANALYSE UNIVARIÉE

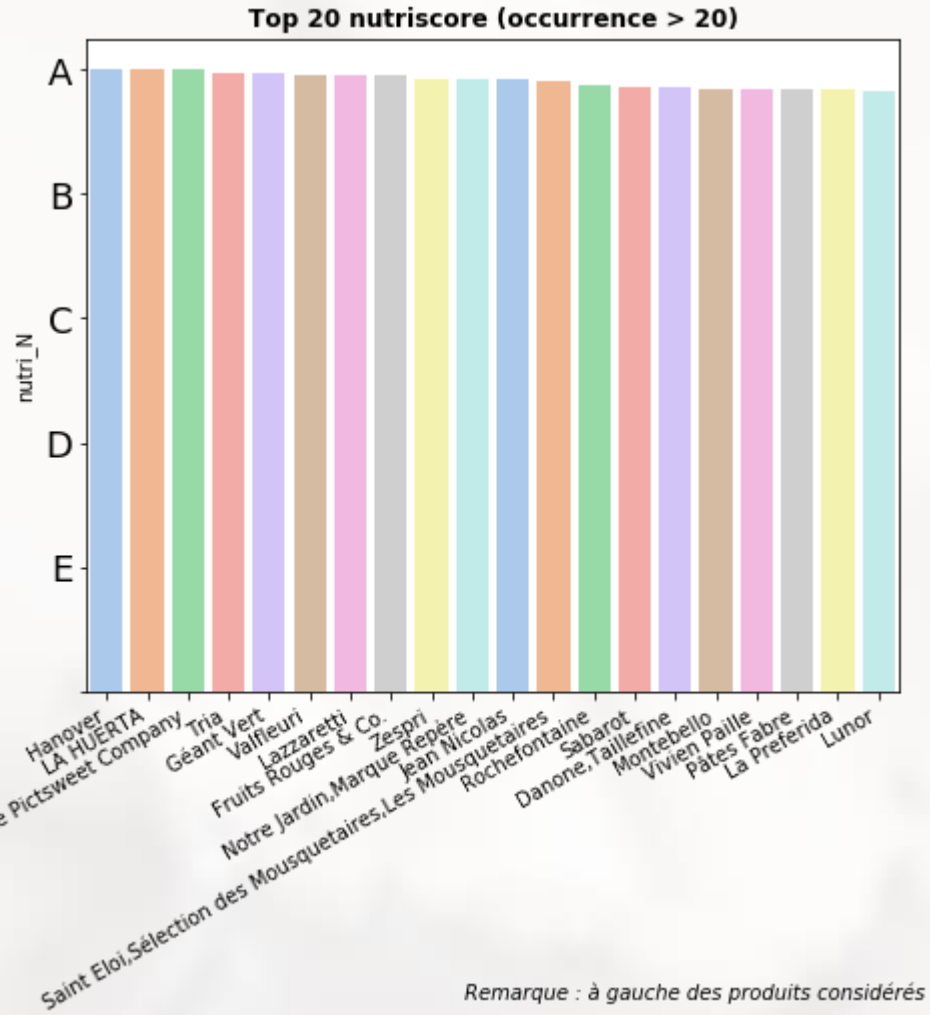


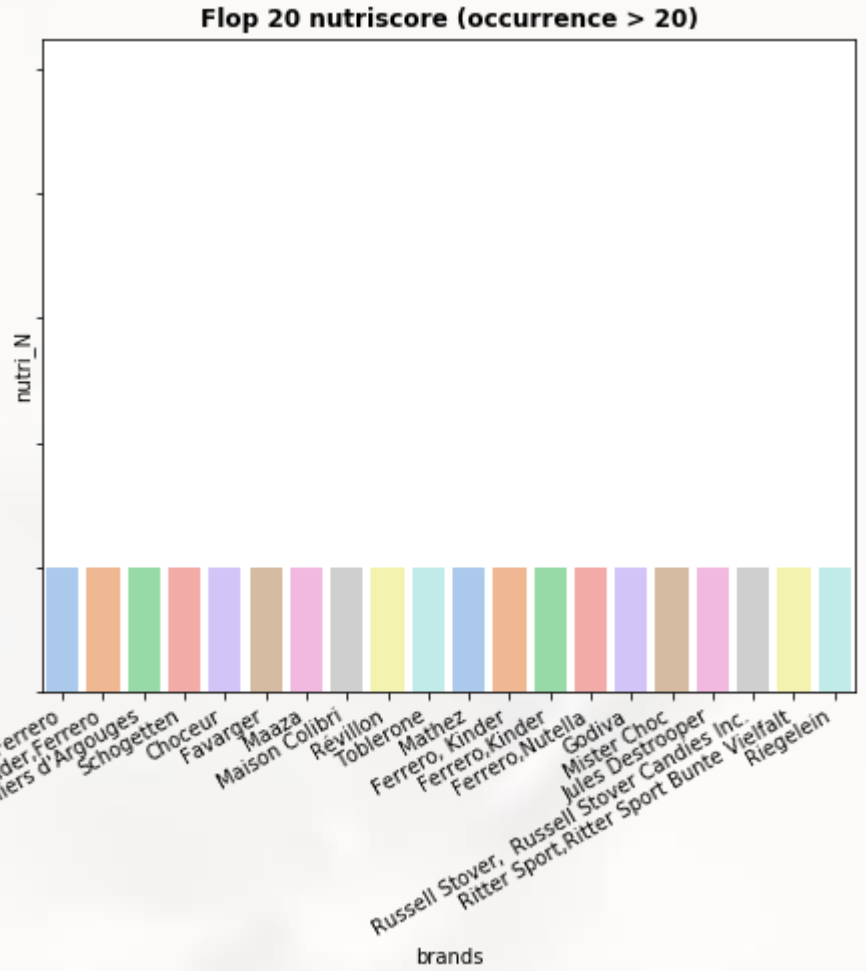




ANALYSE UNIVARIÉE

Marques avec les meilleurs/pires Nutriscore



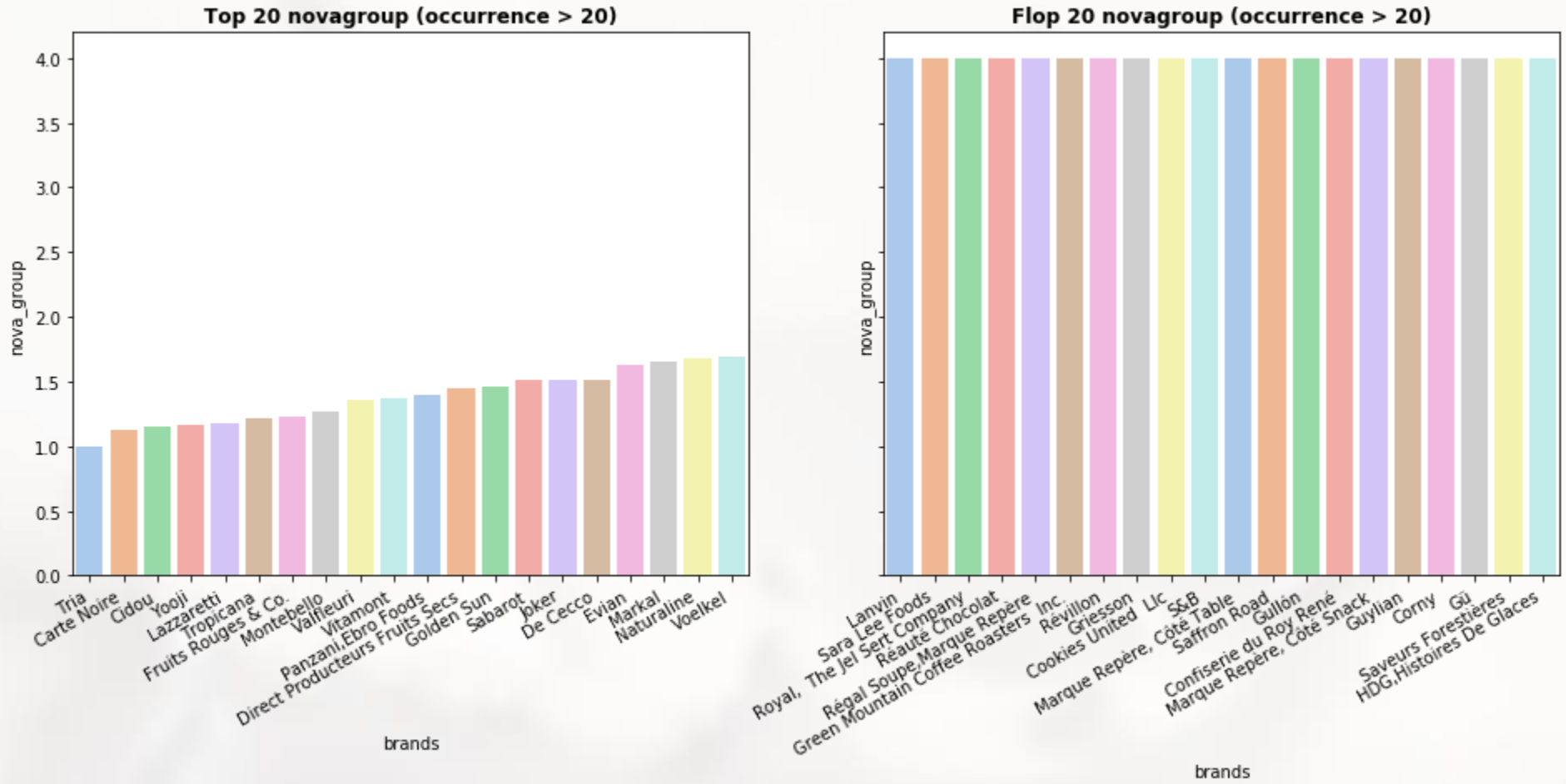


Remarque : à gauche des produits considérés saints, à droite, des produits déconseillés

brands

ANALYSE UNIVARIÉE

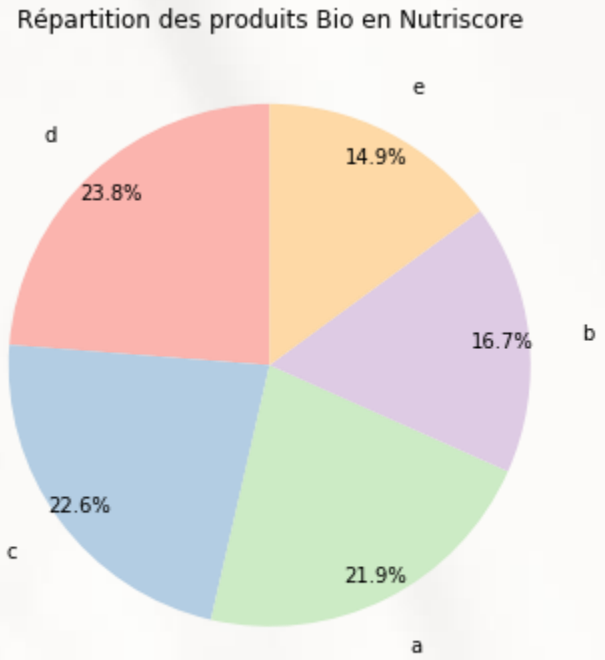
Marques avec les meilleurs/pires Novagroup

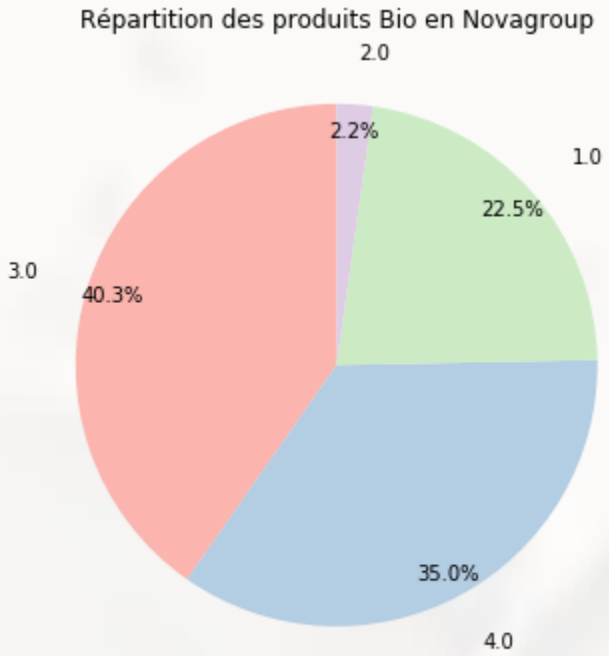


Remarque : à gauche des produits peu transformés, à droite, des produits très transformés

ANALYSE UNIVARIÉE

Répartition des nutriscore / novagroup chez les produits bio

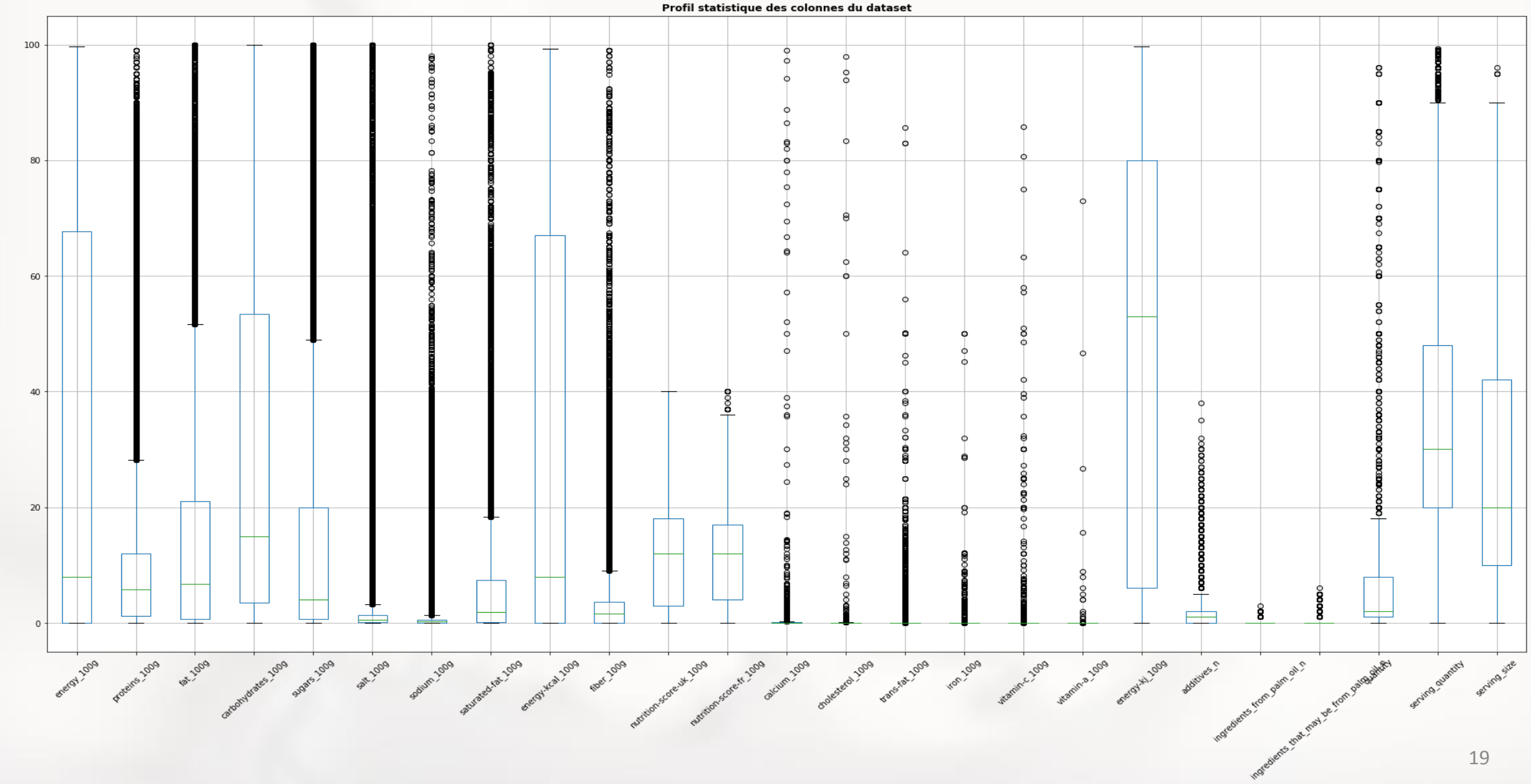




Remarque : Nutriscore : les produits 'Bio' sont uniformément répartis - Novagroup : les produits 'Bio' sont majoritairment très transformés (75%)

ANALYSE UNIVARIÉE

Profil statistique des variables :



ANALYSE UNIVARIÉE

Vue macroscopique des distributions :

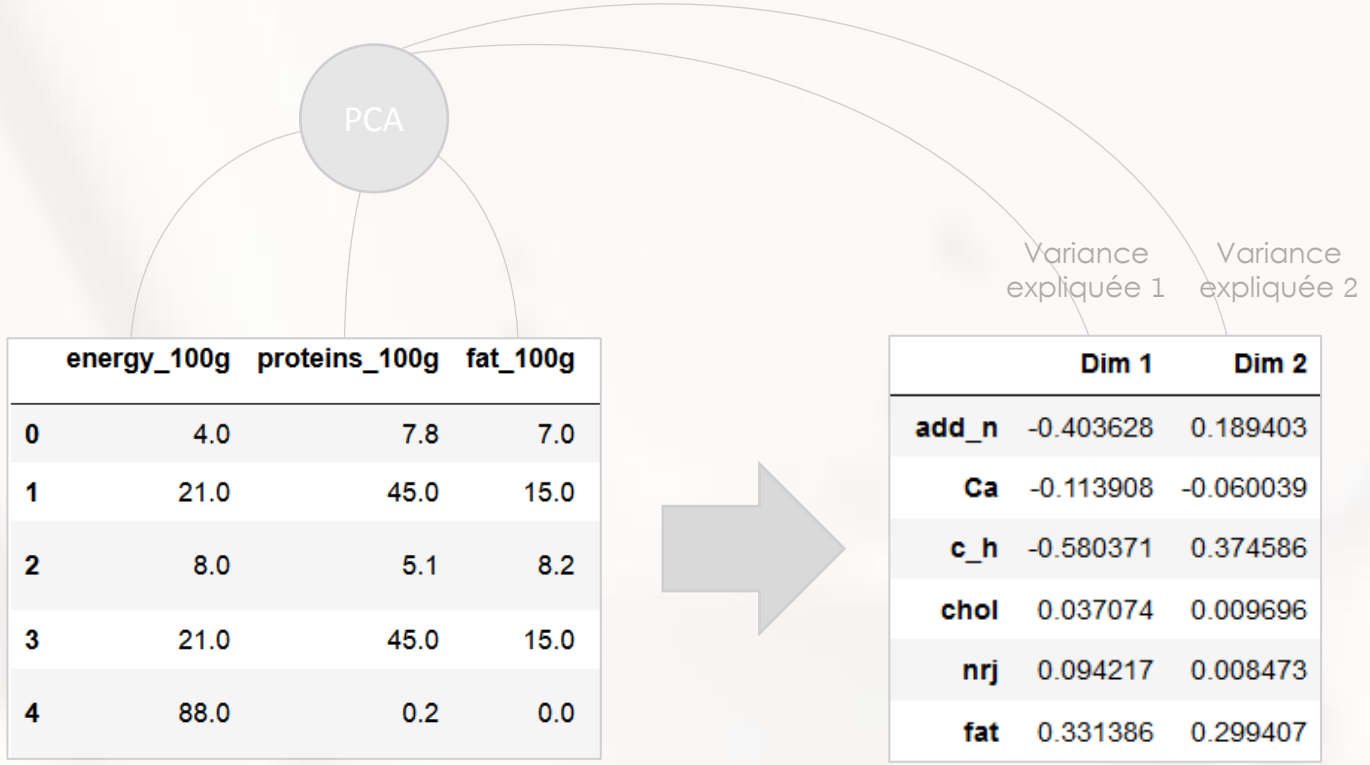
Des patterns similaires

! ≠ Corrélation !

ANALYSE MULTIVARIÉE

ANALYSE MULTIVARIÉE

Principe du PCA appliqué au jeu de données :



	energy_100g	proteins_100g	fat_100g
0	4.0	7.8	7.0
1	21.0	45.0	15.0
2	8.0	5.1	8.2
3	21.0	45.0	15.0
4	88.0	0.2	0.0

	Dim 1	Dim 2
add_n	-0.403628	0.189403
Ca	-0.113908	-0.060039
c_h	-0.580371	0.374586
chol	0.037074	0.009696
nrj	0.094217	0.008473
fat	0.331386	0.299407

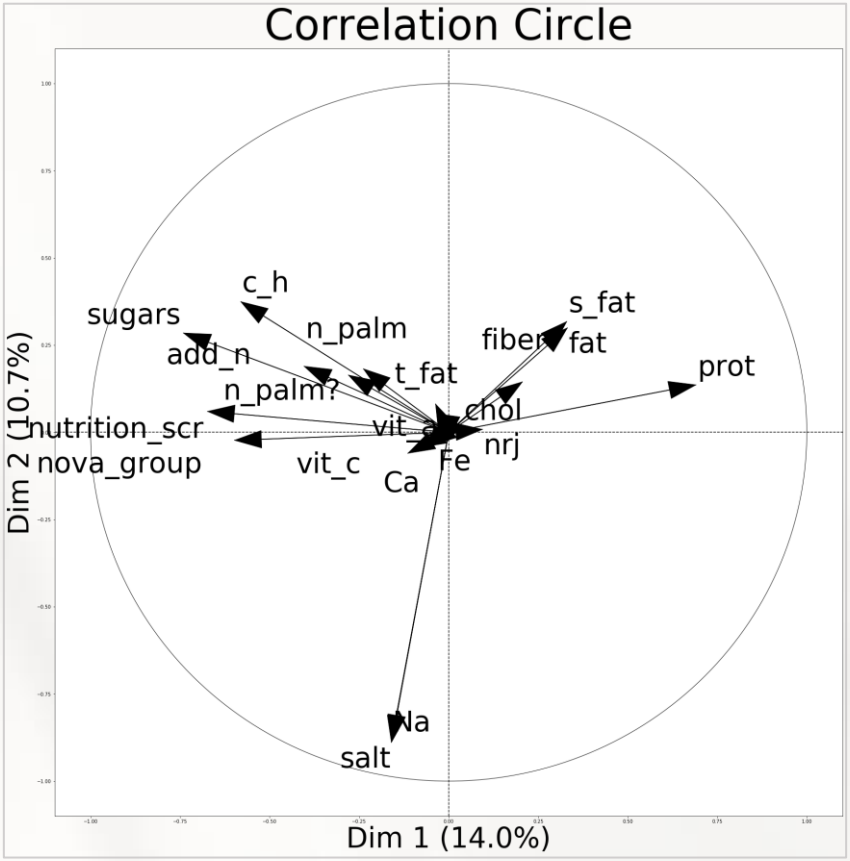
? Principe : transformation des variables corrélées en « composantes principales » (dimensions)

\+ Avantages : facilité de calcul d'un grand dataset, visualisation possible

\- Inconvénients : perte d'information

ANALYSE MULTIVARIÉE

Principe du PCA appliqué au jeu de données :

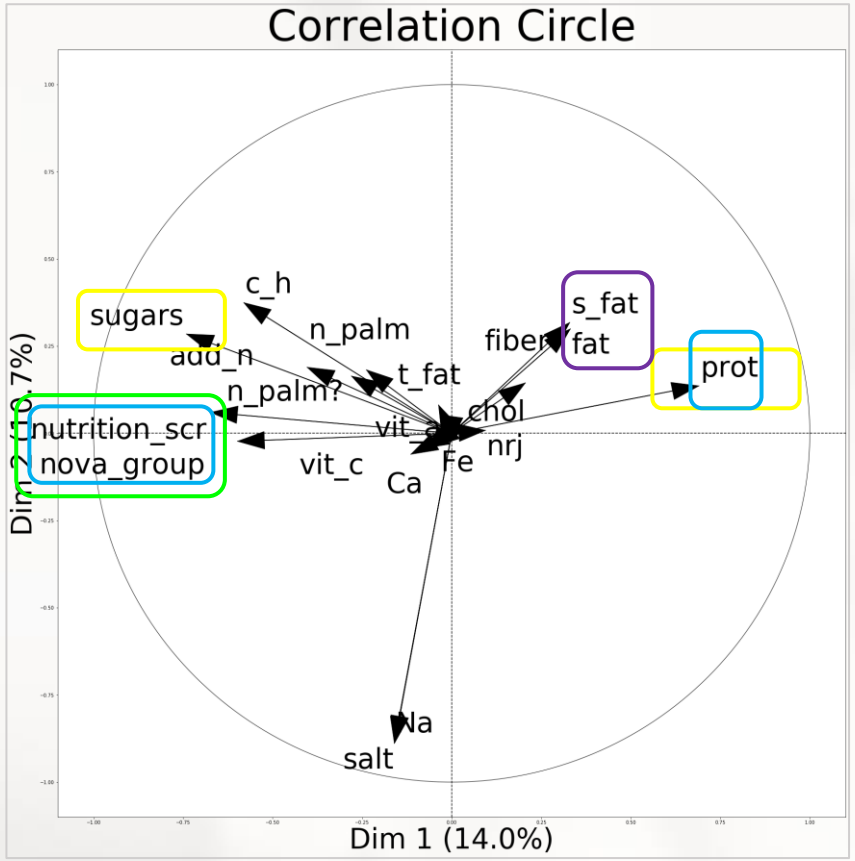


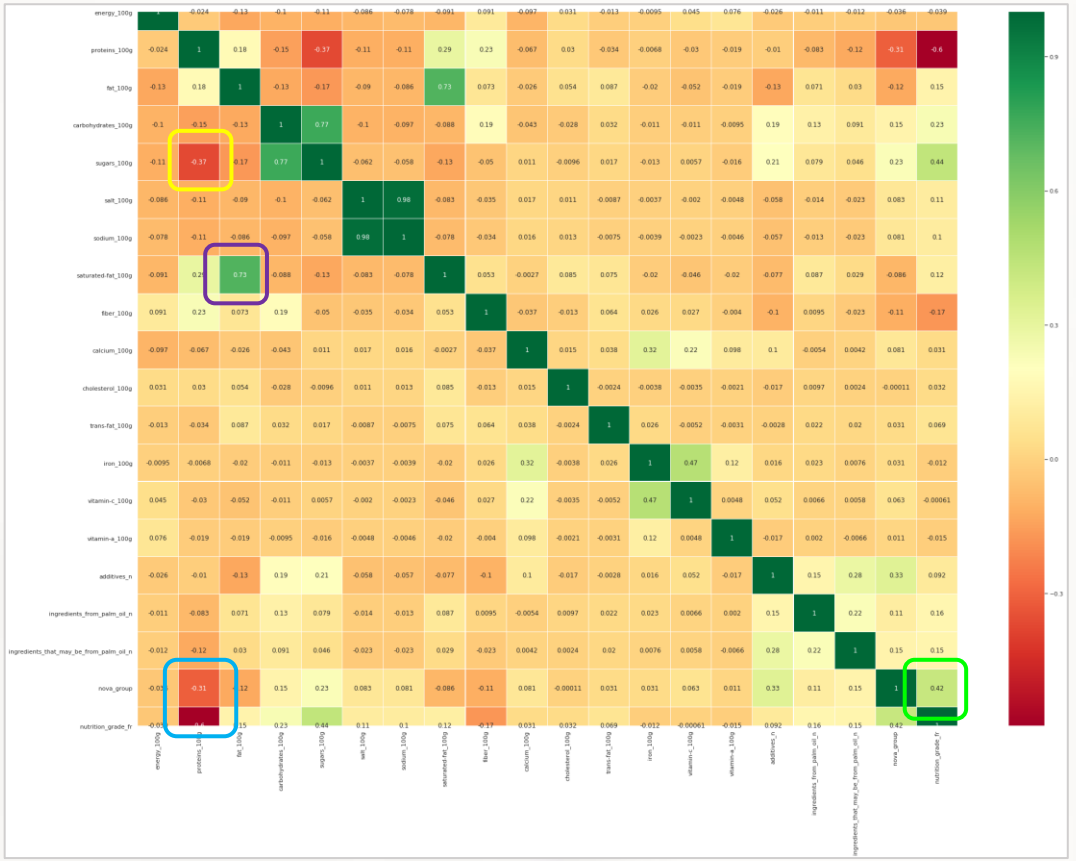
Composante Principale 1 : nutriscore & novagroup inversement proportionnels au protéines / gras / fibres

Composante Principale 2 : sel inversement proportionnel au gras / fibres

ANALYSE MULTIVARIÉE

Vérification de corrélation du PCA avec la méthode de Pearson





Les principales corrélations du cercle de corrélations sont retrouvées et confirmées dans la représentation graphique (heatmap) de Pearson

IDÉE D'APPLICATION

Que proposer à l'agence Santé publique France ?

IDÉE D'APPLICATION

Le consommateur scanne les produits (code-barre)

Prise en compte de la catégorie du produit (utilisation du dataset)

Prise en compte du nombre de portions dans le produit (utilisation du dataset)

Classement du produit (utilisation de l'analyse multivariée)

Recommandation de repas équilibrés en fonction des achats

CONCLUSION ET PERSPECTIVES

CONCLUSION ET PERSPECTIVES

Il a été effectué :

- Nettoyage d'un jeu de données indépendamment des variables
- Analyse univariée de ces données (classements, distributions, etc.)
- Analyse multivariée de ces données (corrélations & confirmation)
- Idée d'application : recommandation de repas équilibrés en fonction des achats.

Perspectives pour l'agence Santé publique France :

- Aller plus loin en proposant des alertes avant le passage à la caisse, des partenariats avec les réseaux de distributeurs (système de points = bons d'achats), etc.