Projet 3 : Préparez les données

pour un organisme de Santé Publique

Santé publique France

open FOOD facts

Introduction

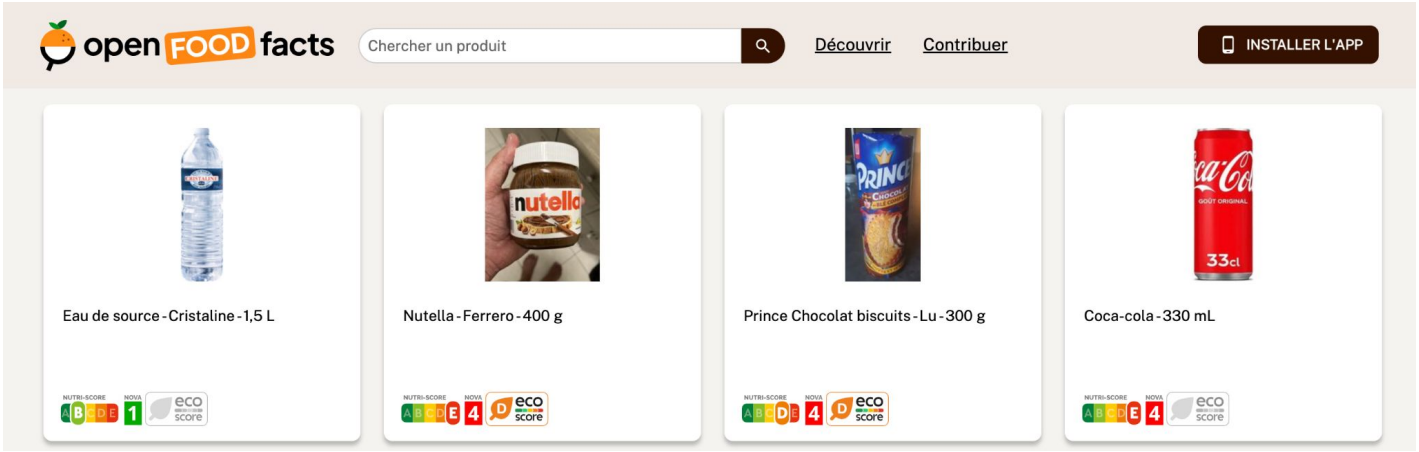
OPENCLASSROOMS

Gael Delescluse

Table des matières

OPENCLASSROOMS

Présentation



NUTRI-SCORE

A B C D E

+ 800 millions d'adultes

+ 2,8 millions décès

OPENCLASSROOMS

Analyse métier

Dataset

320.772 individus

162 variables

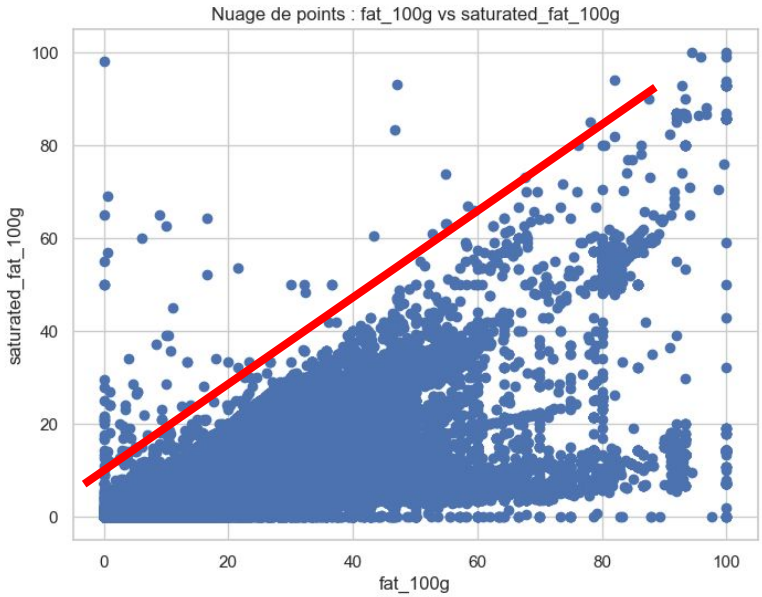
données manquantes

Séparateur \t

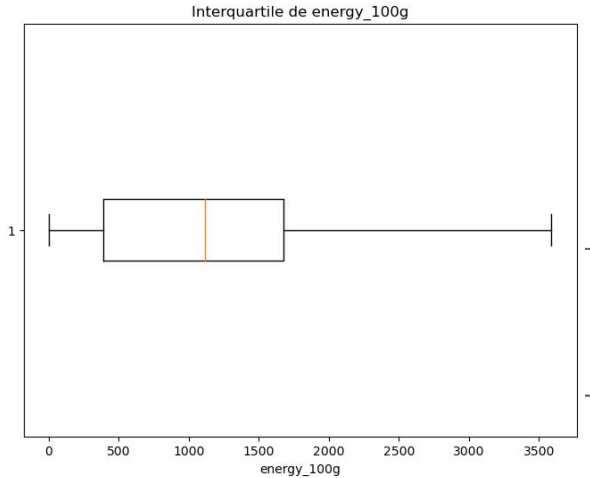
```
product_name                    object
nutrition_grade_fr              object
energy_100g                    float64
proteins_100g                  float64
sugars_100g                    float64
fat_100g                       float64
saturated-fat_100g             float64
salt_100g                      float64
sodium_100g                    float64
fiber_100g                     float64
fruits-vegetables-nuts_100g    float64
dtype: object
```

Nettoyage du Dataset

Valeurs aberrantes



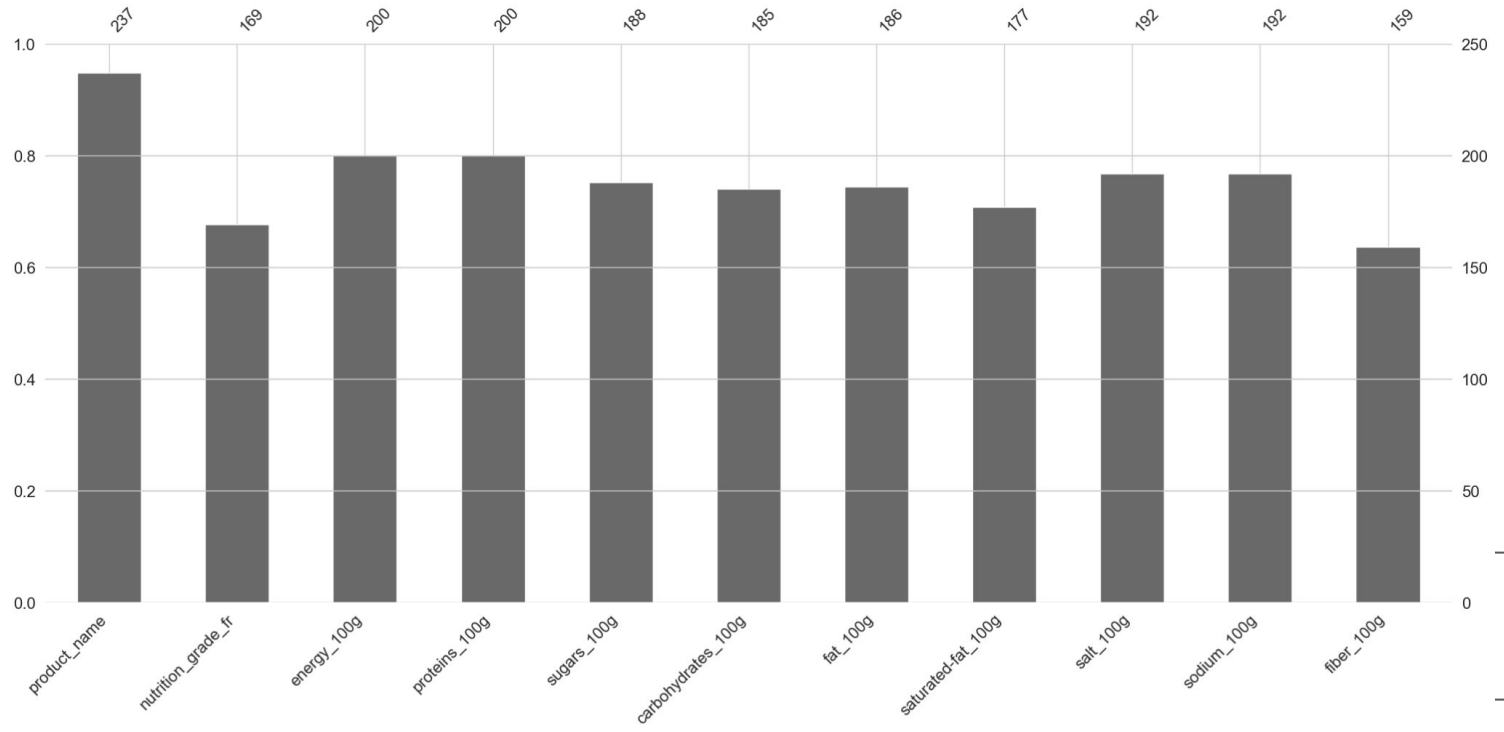

```
index_drop_rogue_value = df[df['proteins_100g'] > 100].index
df.drop(index_drop_rogue_value, inplace=True)
index_drop_rogue_value = df[df['sugars_100g'] > 100].index
df.drop(index_drop_rogue_value, inplace=True)
index_drop_rogue_value = df[df['fat_100g'] > 100].index
df.drop(index_drop_rogue_value, inplace=True)
index_drop_rogue_value = df[df['saturated_fat_100g'] > 100].index
df.drop(index_drop_rogue_value, inplace=True)
index_drop_rogue_value = df[df['salt_100g'] > 100].index
df.drop(index_drop_rogue_value, inplace=True)
index_drop_rogue_value = df[df['fiber_100g'] > 100].index
df.drop(index_drop_rogue_value, inplace=True)
index_drop_rogue_value = df[df['proteins_100g'] < 0].index
df.drop(index_drop_rogue_value, inplace=True)
index_drop_rogue_value = df[df['sugars_100g'] < 0].index
df.drop(index_drop_rogue_value, inplace=True)
index_drop_rogue_value = df[df['fiber_100g'] < 0].index
df.drop(index_drop_rogue_value, inplace=True)
```



Huile d'avocat 3765 kJ

OPENCLASSROOMS

Données manquantes



Corrélation Pearson

fat_100g

salt_100g

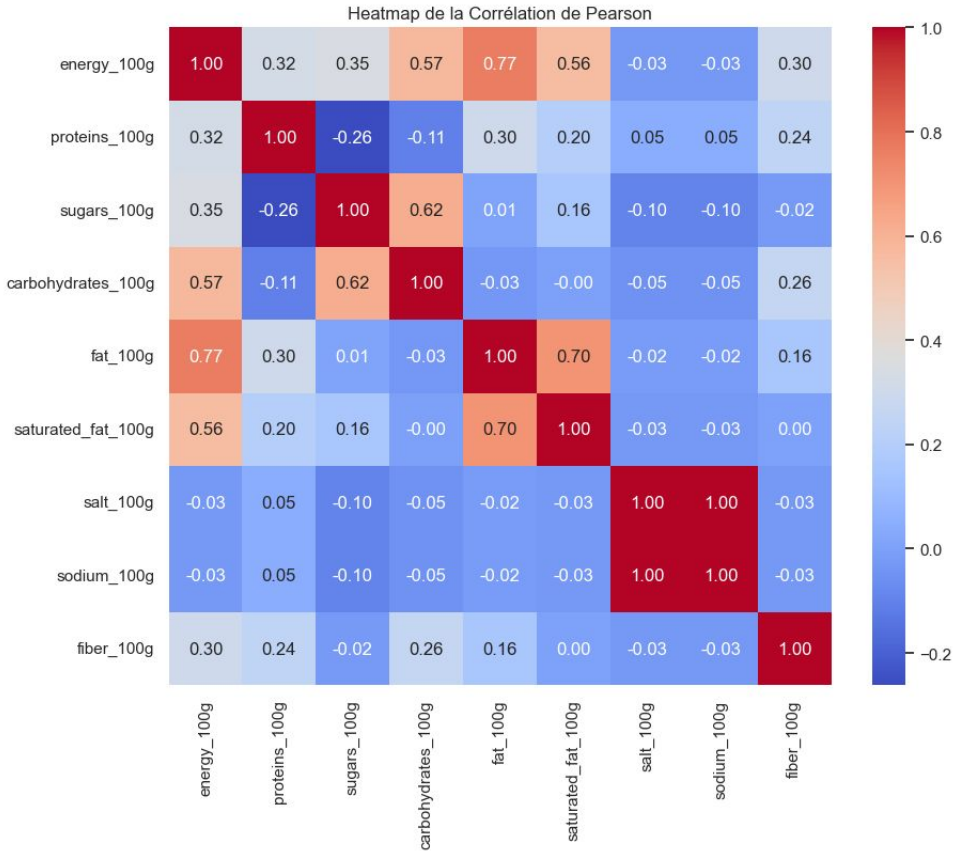
sodium_100g

energy_100g

satured_fat_100g

sugars_100g

carbohydrates_100g



IterativeImputer

proteins_100g -> 1%

salt_100g -> 3%

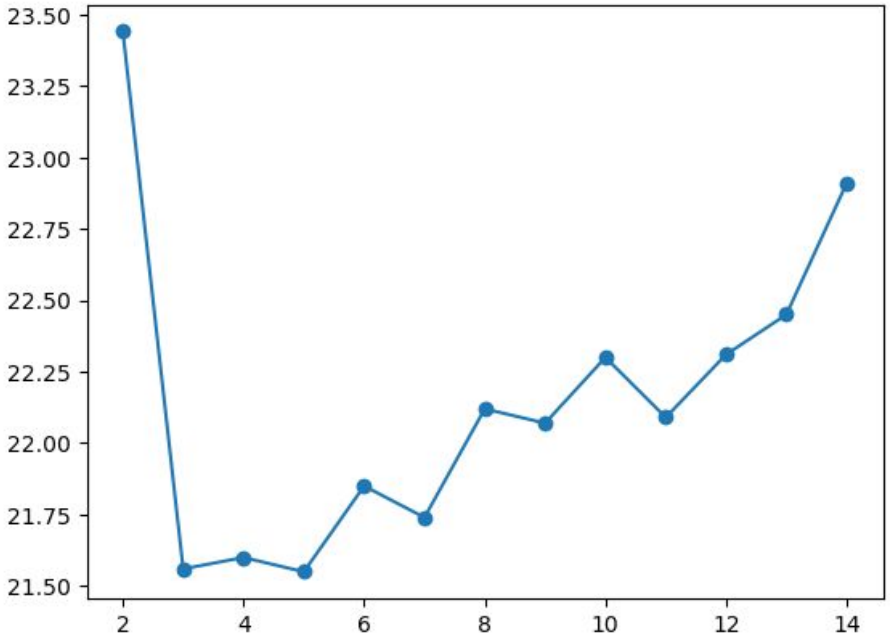
sodium_100g -> 3%

```
product_name            0.00
nutrition_grade_fr     13.04
energy_100g             0.00
proteins_100g           0.94
sugars_100g             0.00
carbohydrates_100g      0.00
fat_100g                0.00
saturated_fat_100g      0.00
salt_100g               2.31
sodium_100g             2.33
fiber_100g             21.21
dtype: float64
```

```
imputer = IterativeImputer()
df['fat_100g'] = imputer.fit_transform(df[['fat_100g', 'saturated_fat_100g',\
                                           'energy_100g']])[:, 0]
df['saturated_fat_100g'] = imputer.fit_transform(df[['saturated_fat_100g', 'fat_100g',\
                                                     'energy_100g']])[:, 0]
df['energy_100g'] = imputer.fit_transform(df[['energy_100g', 'fat_100g', 'saturated_fat_100g',\
                                              'carbohydrates_100g']])[:, 0]
df['carbohydrates_100g'] = imputer.fit_transform(df[['carbohydrates_100g', 'sugars_100g',\
                                                     'energy_100g']])[:, 0]
df['sugars_100g'] = imputer.fit_transform(df[['sugars_100g', 'carbohydrates_100g']])[:, 0]
```

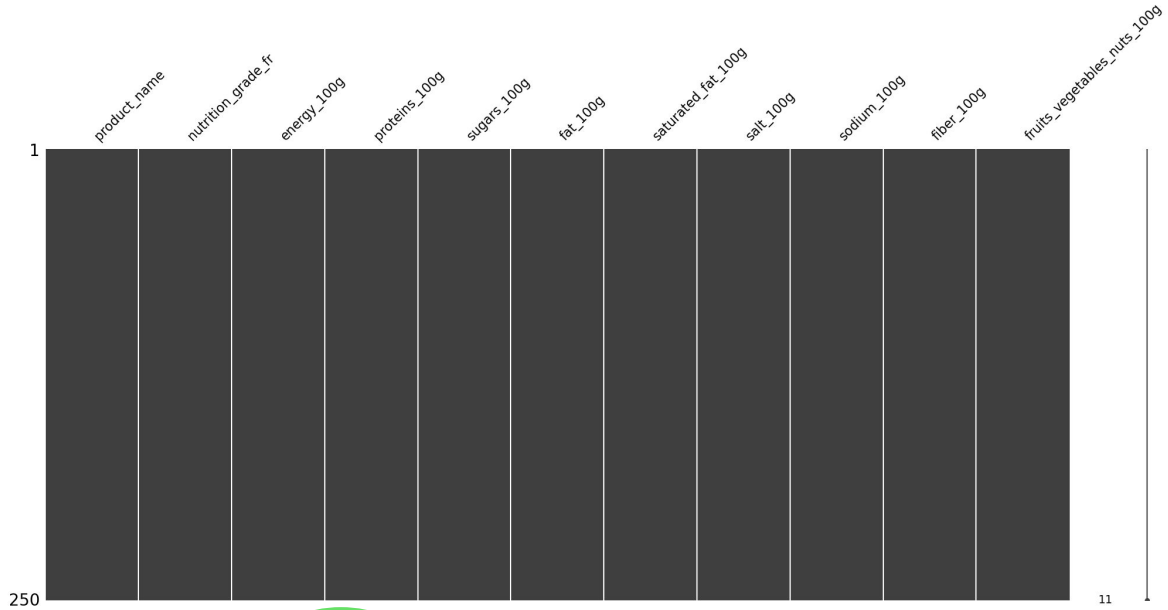
K nearest neighbor

```
df_sample = df_without_Nan.sample(50000, replace=False)
```




```
n_neighbors=3
```

K nearest neighbor



Accuracy

84%

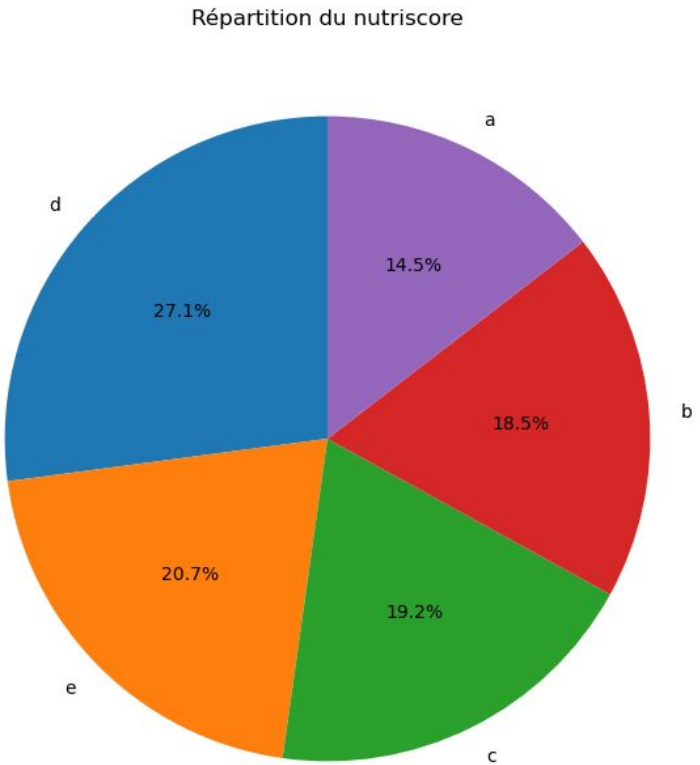
29%

suppression d’individus

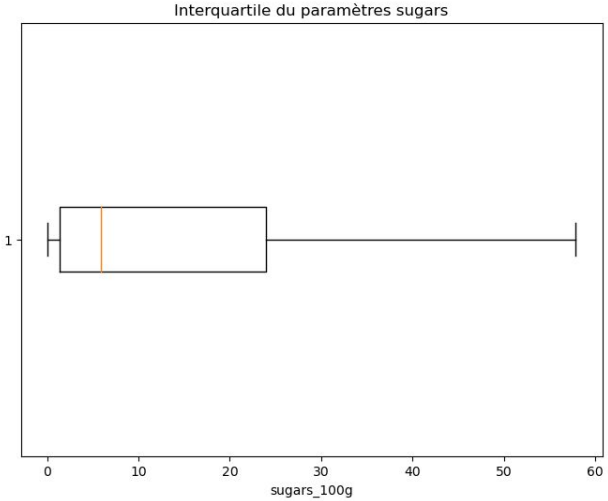
OPENCLASSROOMS

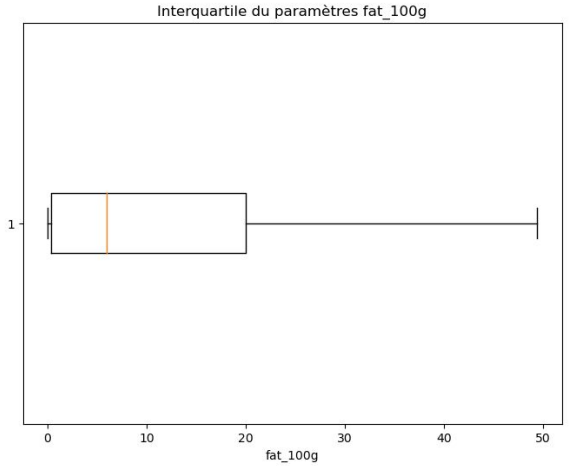
Analyse exploratoire du Dataset

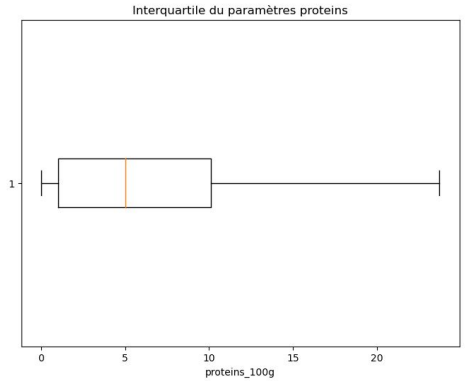
Analyse univariée



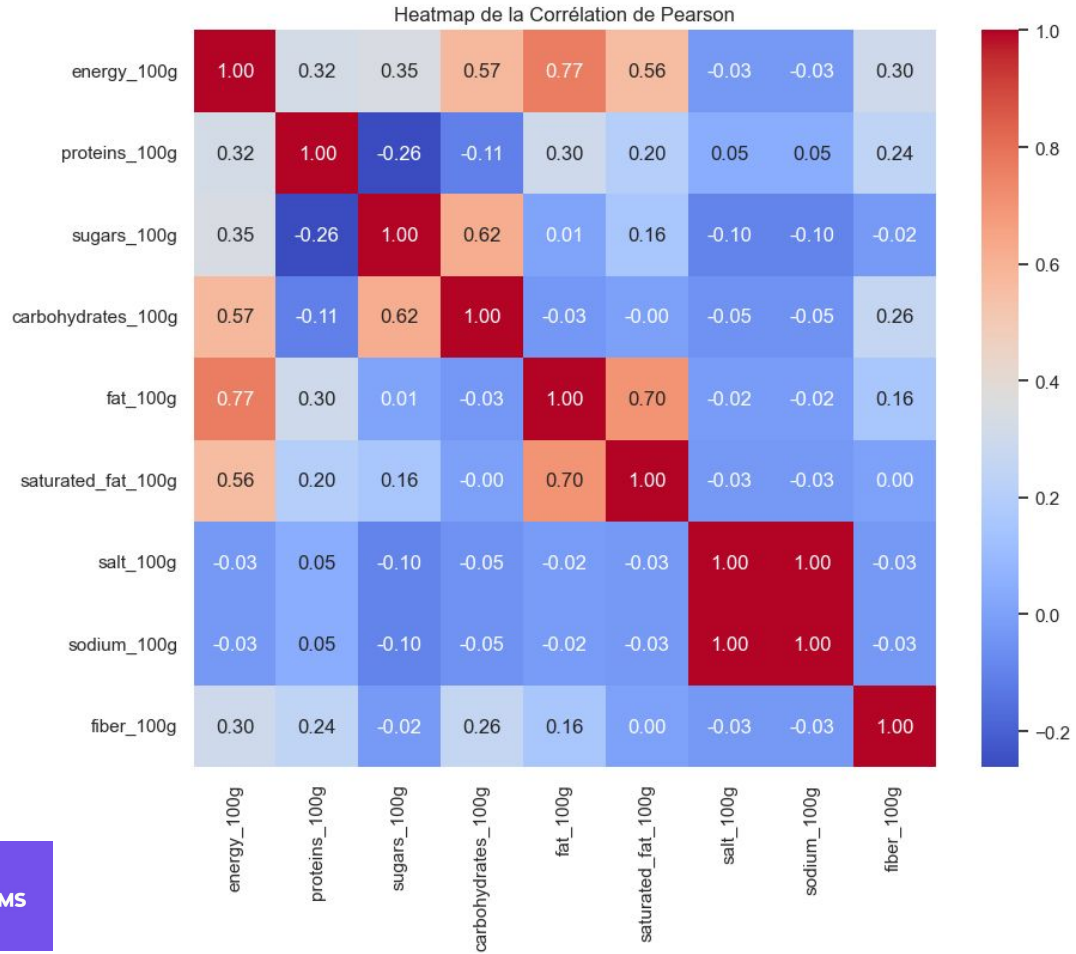
Analyse univariée

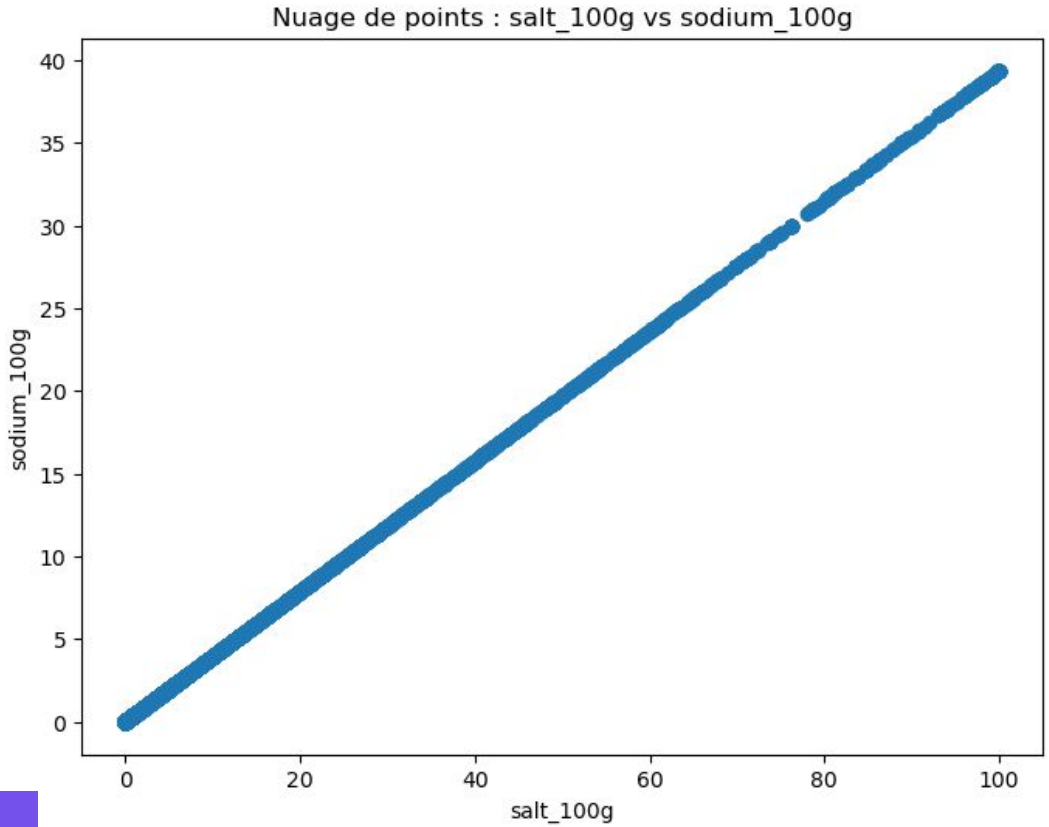




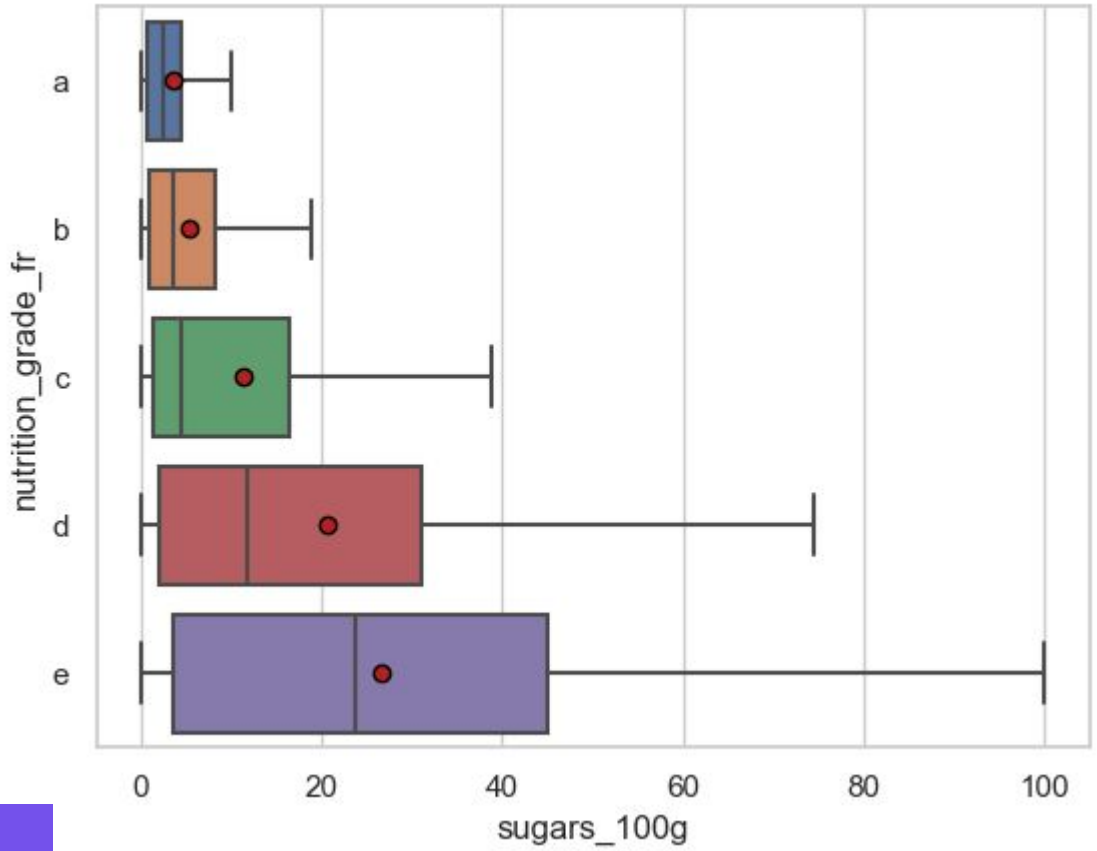


Analyse bivariée



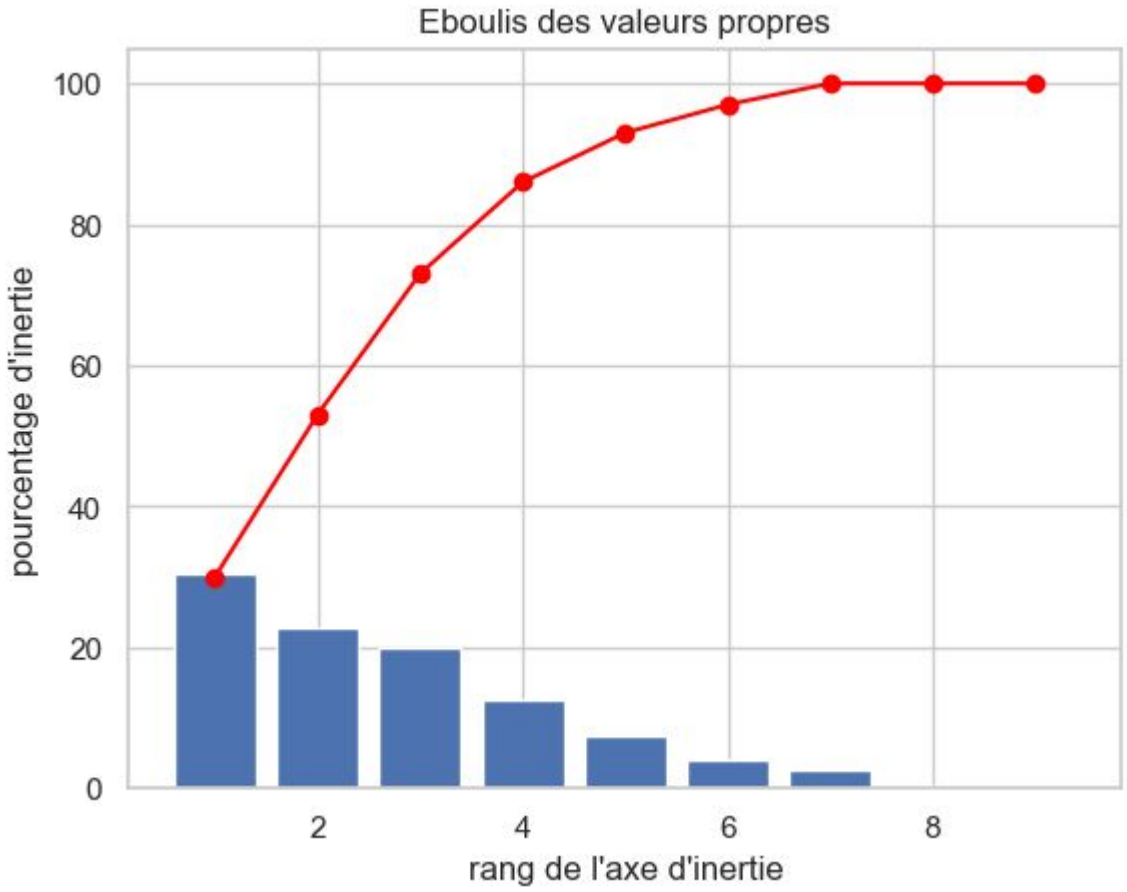


Analyse bivariée



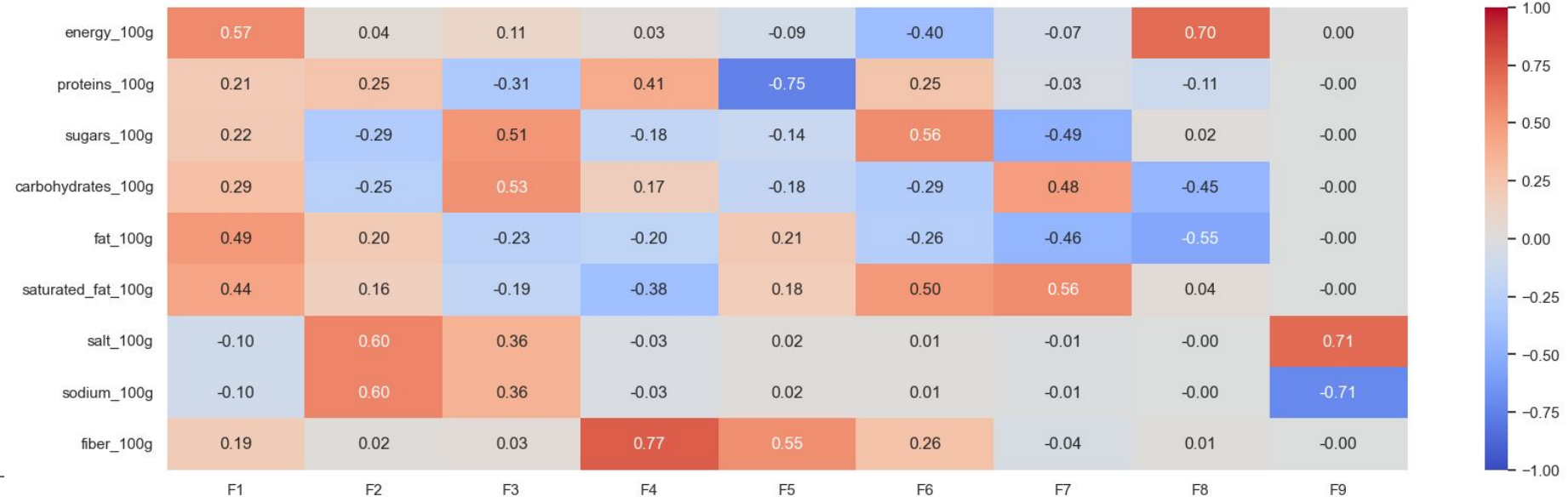
Analyse bivariée

OPENCLASSROOMS

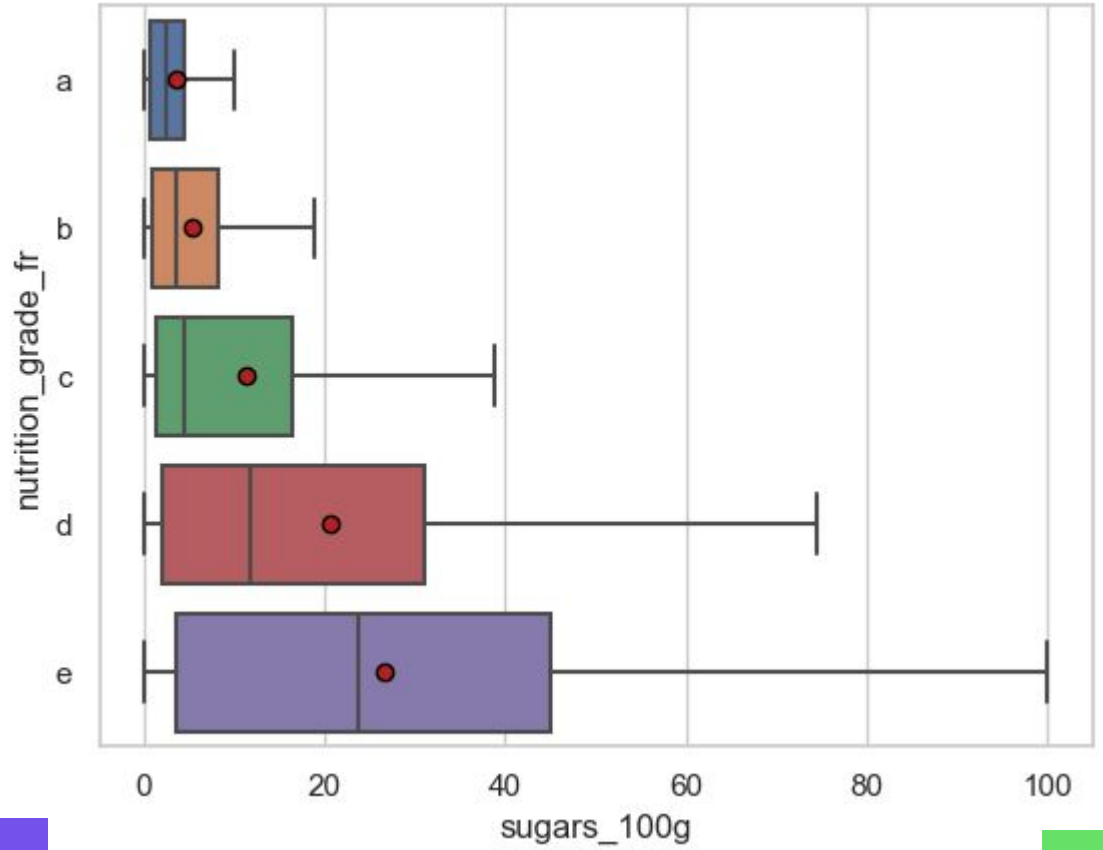


ACP

OPENCLASSROOMS


energy_100g 0.57 0.04 0.11 0.03 -0.09 -0.40 -0.07 0.70 0.00
proteins_100g 0.21 0.25 -0.31 0.41 -0.75 0.25 -0.03 -0.11 -0.00
sugars_100g 0.22 -0.29 0.51 -0.18 -0.14 0.56 -0.49 0.02 -0.00
carbohydrates_100g 0.29 -0.25 0.53 0.17 -0.18 -0.29 0.48 -0.45 -0.00
fat_100g 0.49 0.20 -0.23 -0.20 0.21 -0.26 -0.46 -0.55 -0.00
saturated_fat_100g 0.44 0.16 -0.19 -0.38 0.18 0.50 0.56 0.04 -0.00
salt_100g -0.10 0.60 0.36 -0.03 0.02 0.01 -0.01 -0.00 0.71
sodium_100g -0.10 0.60 0.36 -0.03 0.02 0.01 -0.01 -0.00 -0.71
fiber_100g 0.19 0.02 0.03 0.77 0.55 0.26 -0.04 0.01 -0.00
F1 F2 F3 F4 F5 F6 F7 F8 F9
1.00 0.75 0.50 0.25 0.00 −0.25 −0.50 −0.75 −1.00

ACP Heatmap


a
b
c
d
e
nutrition_grade_fr
0
20
40
60
80
100
sugars_100g

```python
from scipy.stats import kruskal

groups = [group['sugars_100g'] for name, group in df.groupby('nutrition_grade_fr')]
h_statistic, p_value = kruskal(*groups)
print(f"Statistique H : {h_statistic}")
print(f"p-valeur : {p_value}")

alpha = 0.05  # Niveau de signification
if p_value < alpha:
    print("Les distributions des groupes sont différentes.")
else:
    print("Nous ne pouvons pas conclure que les distributions des groupes sont différentes.")
```

```
Statistique H : 31056.540053574132
p-valeur : 0.0
Les distributions des groupes sont différentes.
```

Protection des données personnelles (RGPD)

Licéité, loyauté et transparence

Limitation des finalités

Minimisation des données

Exactitude des données

Limitation de la conservation

Conclusion de l’étude

Corrélation entre différentes variables

Distribution des valeurs impactent le résultat du nutriscore

Synthétisation de variable

Imputation valeurs manquantes