CONCEVEZ UNE APP AU SERVICE DE LA SANTÉ PUBLIQUE

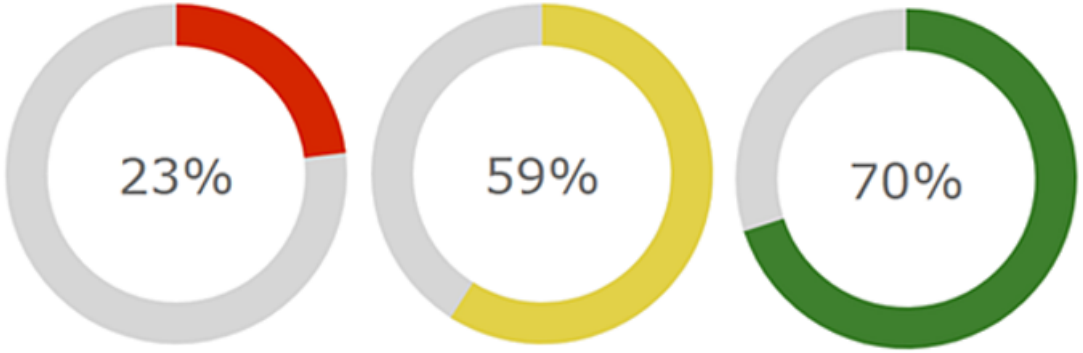
Santé publique France

IDÉE D'APPLICATION

SCAN

Code Barre

BILAN APPORTS + RECO

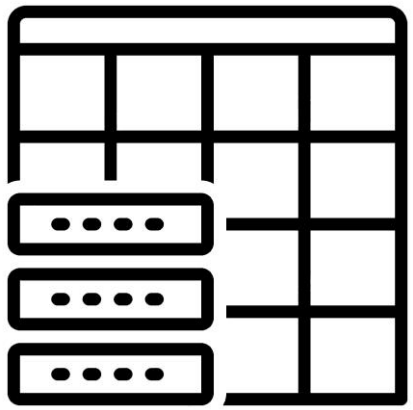


Lipides **Glucides** **Proteines**

Santé publique France

EXPLORATION

_DATASET



2M PRODUITS
183 CATEGORIES

- **Origine: Pays – Marque…**
- **Produit: Packaging – Catégorie – Nom – Text – Image…**
- **Composition: Lipides – Additifs – Huile Palme…**
- **Scoring: Nutriscore – Nitrigrade – Nova…**
- **Suivi: Date – Url – Id….**

Santé publique France

NETTOYAGE

_INDICATEURS

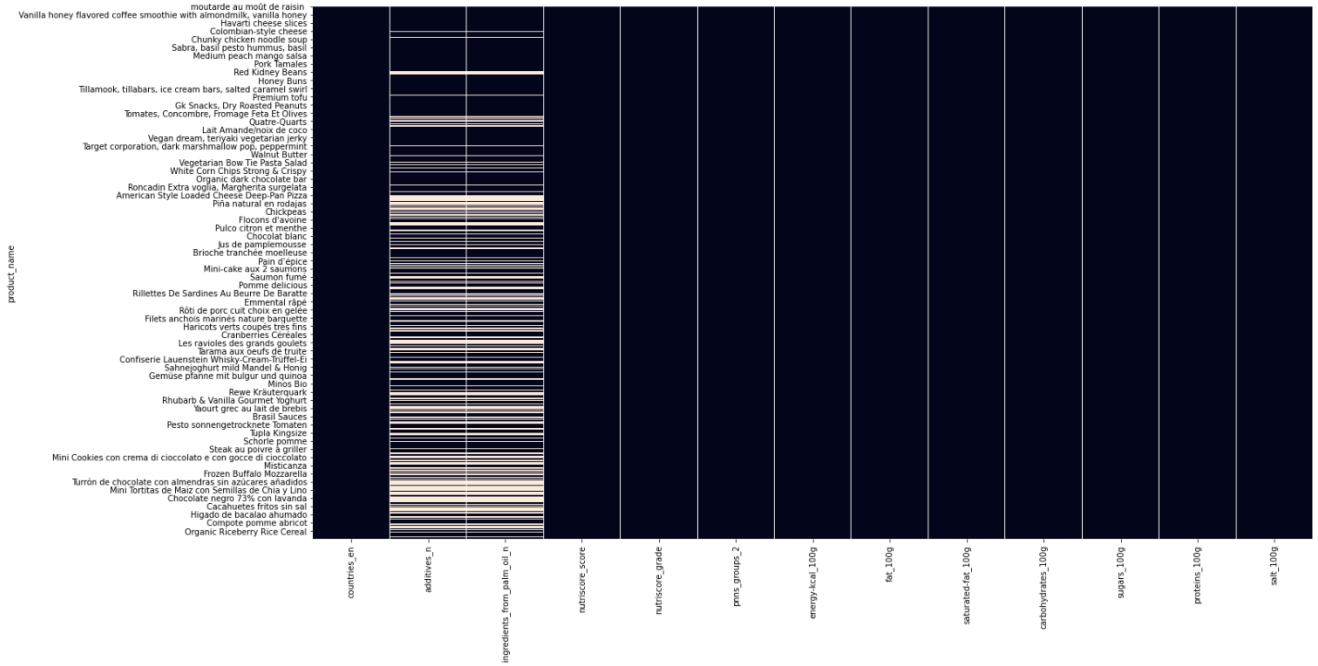
183 INDICATEURS

60 Qualitatifs -126 Quantitatifs

13 INCATEURS

f(NaN, Redondance, Pertinence)

64% > 90% de NaN



Santé publique France

NETTOYAGE

_PRODUITS

2M PRODUITS

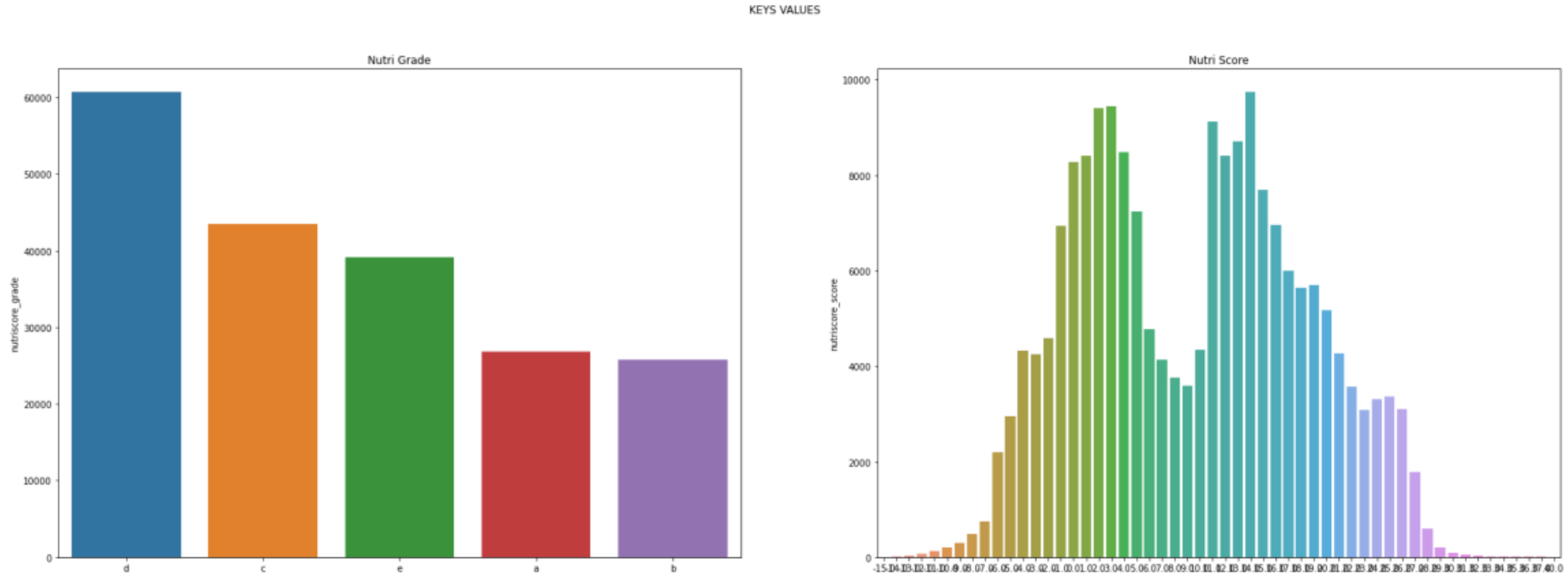
- **Valeurs Abérrantes: 200k**
  - > 100 pour 100g et < 0
  - Inclut > Groupe (ex: Sucre dans Glucides)
  - Valeurs Energetiques incohérentes
- **Redondants: 202k**
- **NaN:**
  - dropna() sur la Target
  - Remplacement NaN produits par median pnns_groups_2
  - dropna(thresh=11) (2 NàN ou moins)
- **Filtre: Produits Francais (plus grand nombre de produits)**

Santé publique France

ANALYSE
_UNIVARIÉ

VARIABLE QUALITATIVES

Target

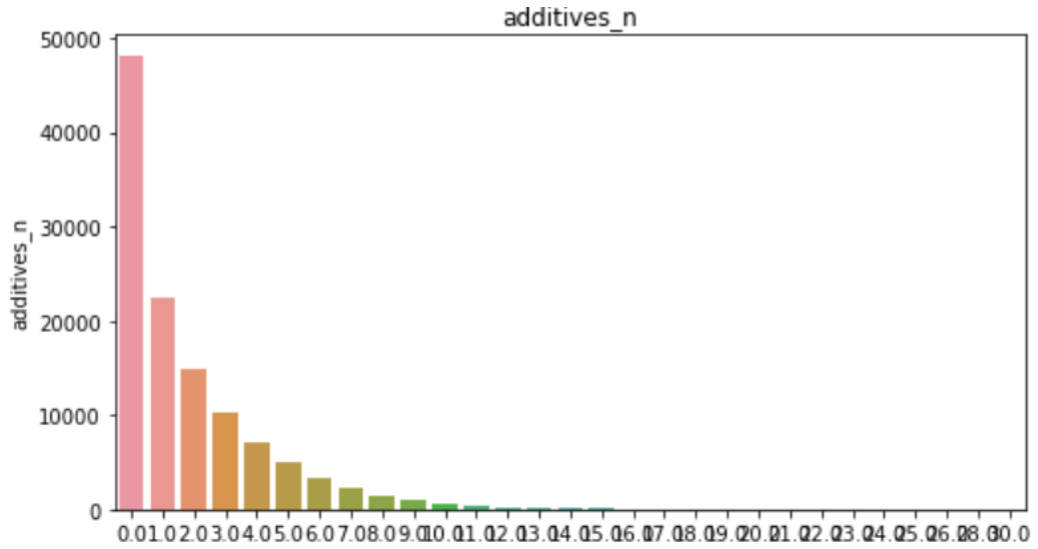


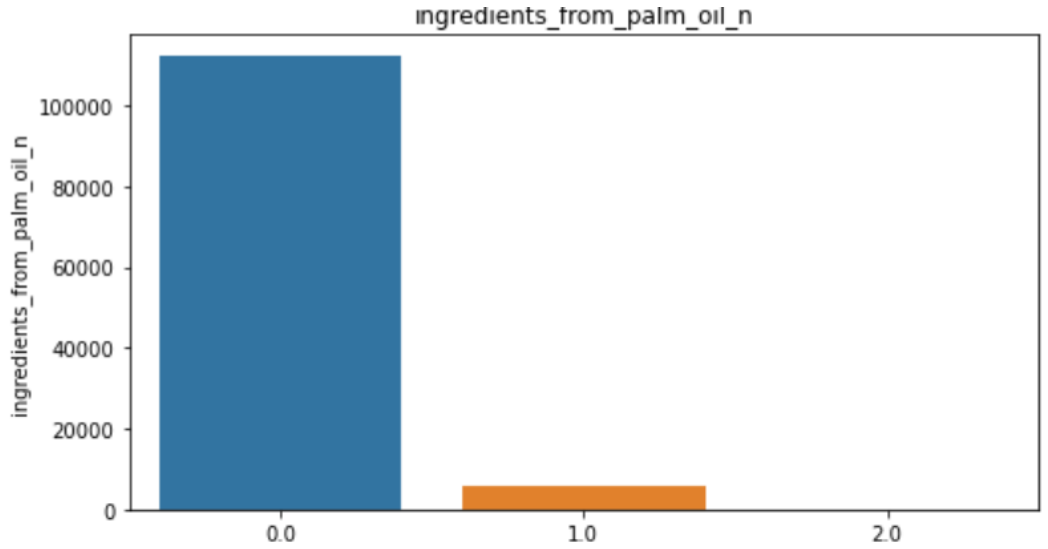
Santé publique France

ANALYSE
_UNIVARIÉ

VARIABLE QUALITATIVES

Additifs – Huile Palme





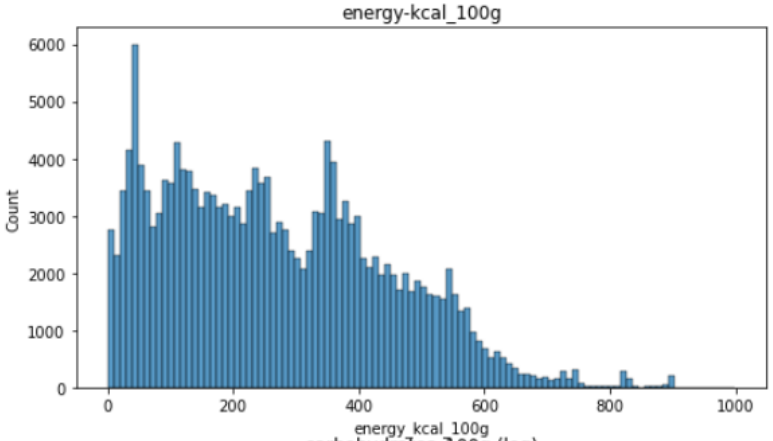
37% sans additifs – 18% en ont 1

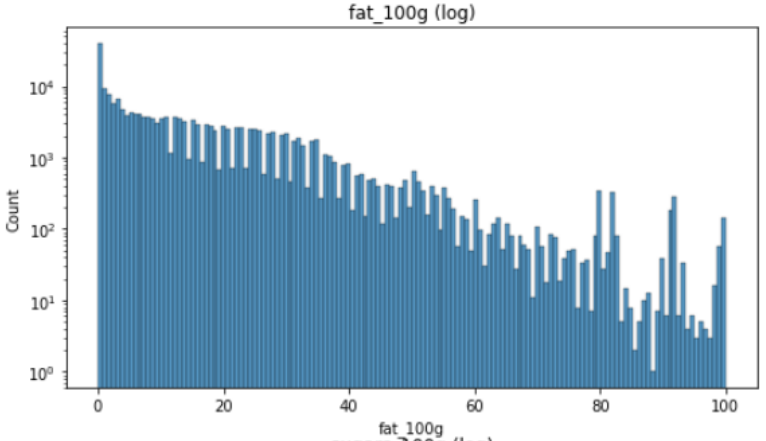
+ de 90% sans Huile de Palme

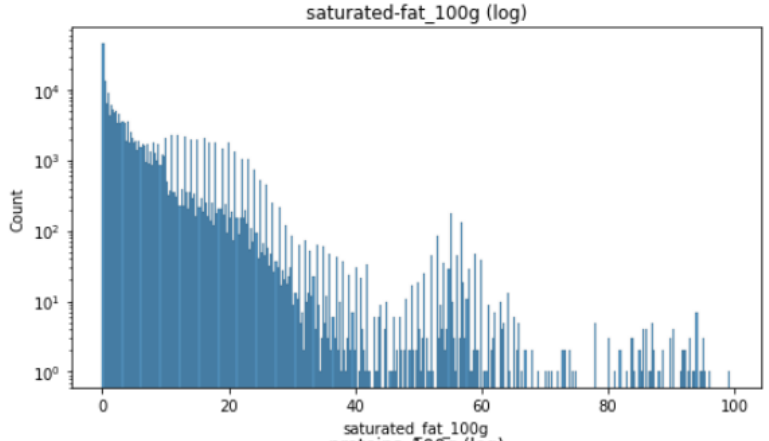
Santé publique France

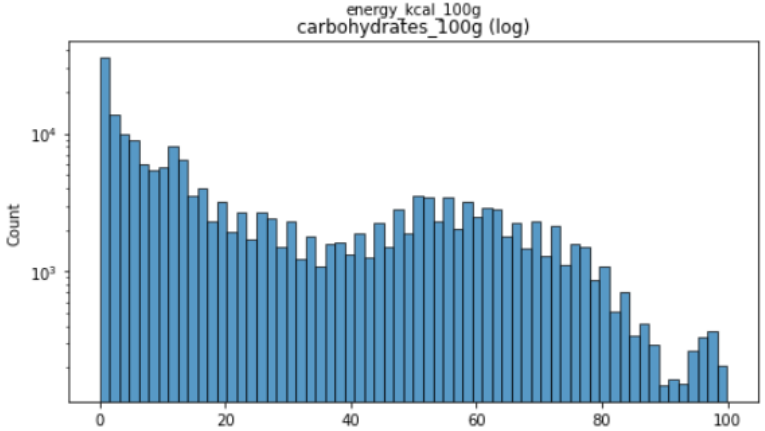
ANALYSE
_UNIVARIÉ

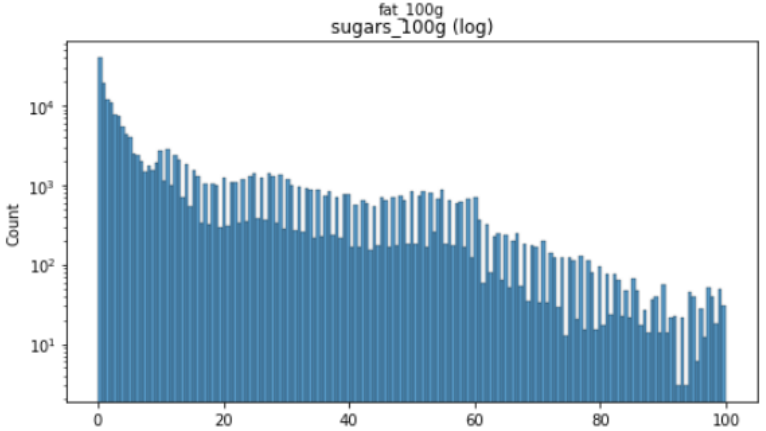
VARIABLE QUANTITATIVES

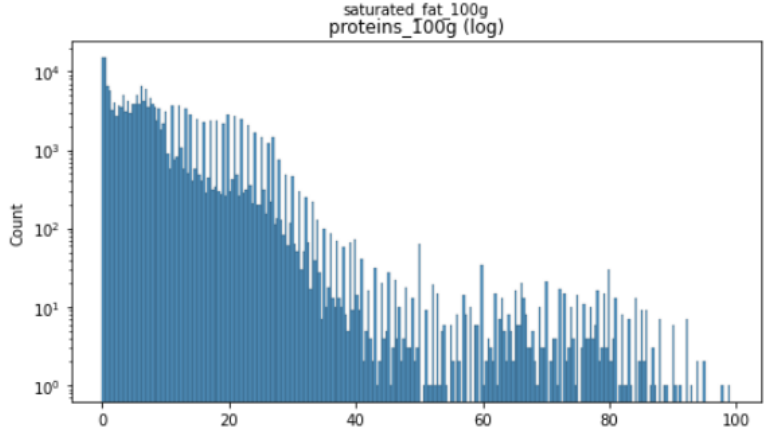










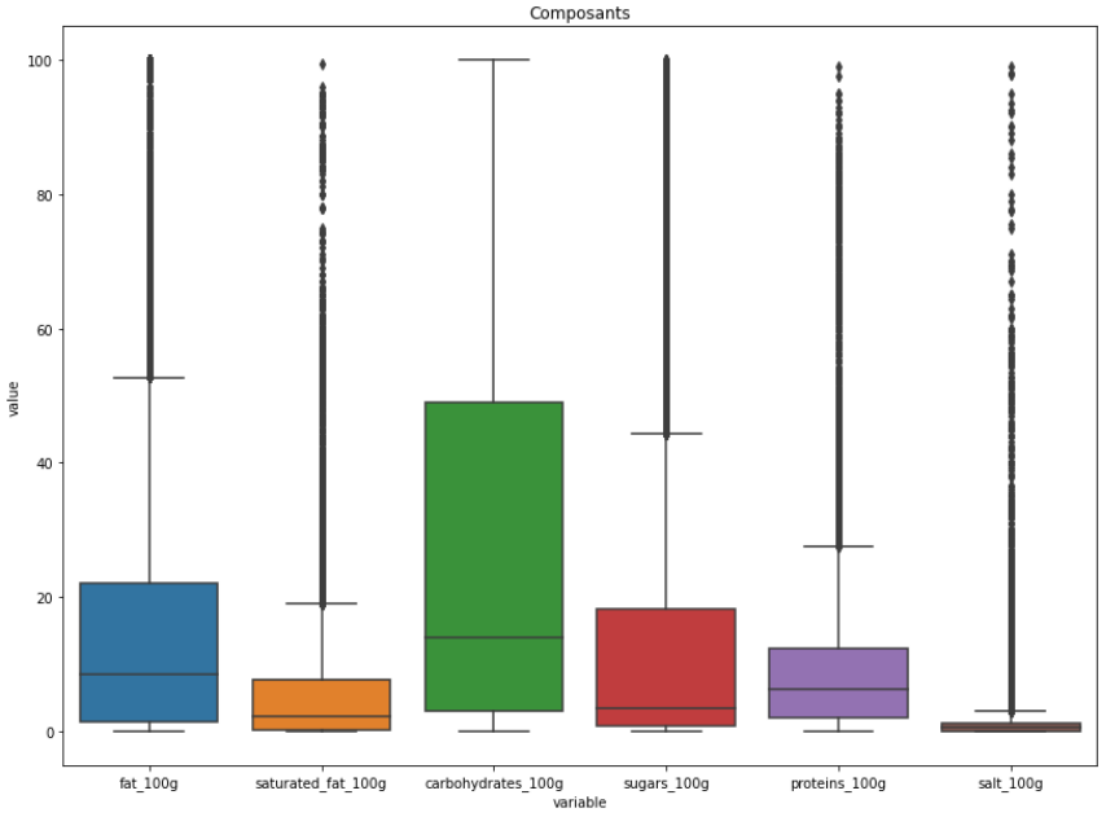


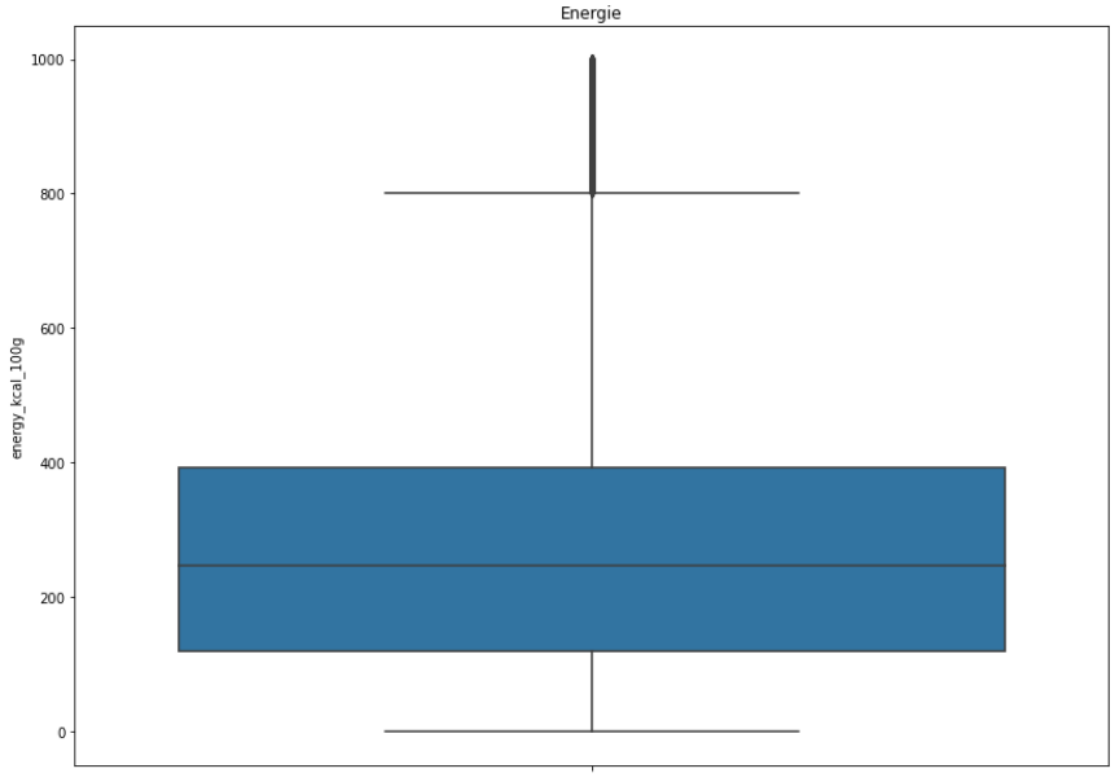
Santé publique France

ANALYSE
_UNIVARIÉ

VARIABLE QUANTITATIVES

BOXPLOTS



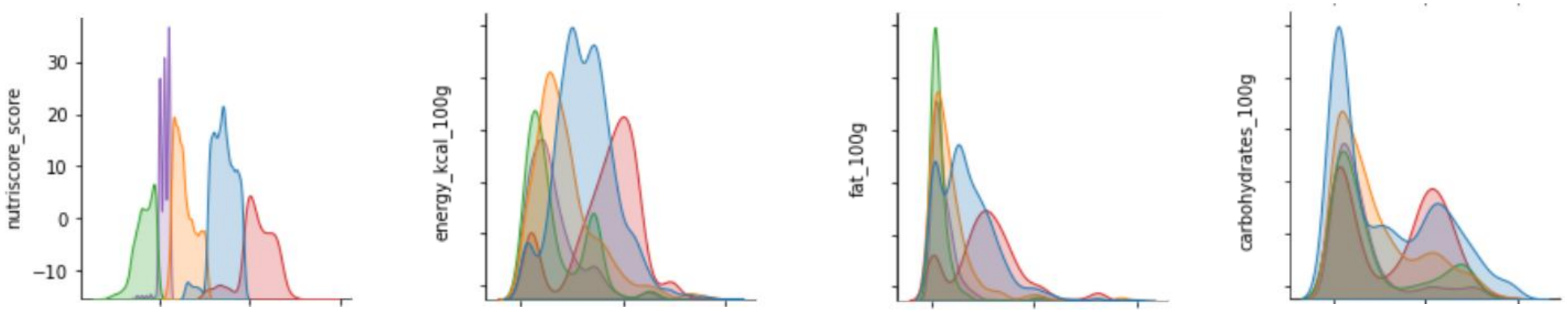


Santé publique France

ANALYSE
_MULTIVARIÉ

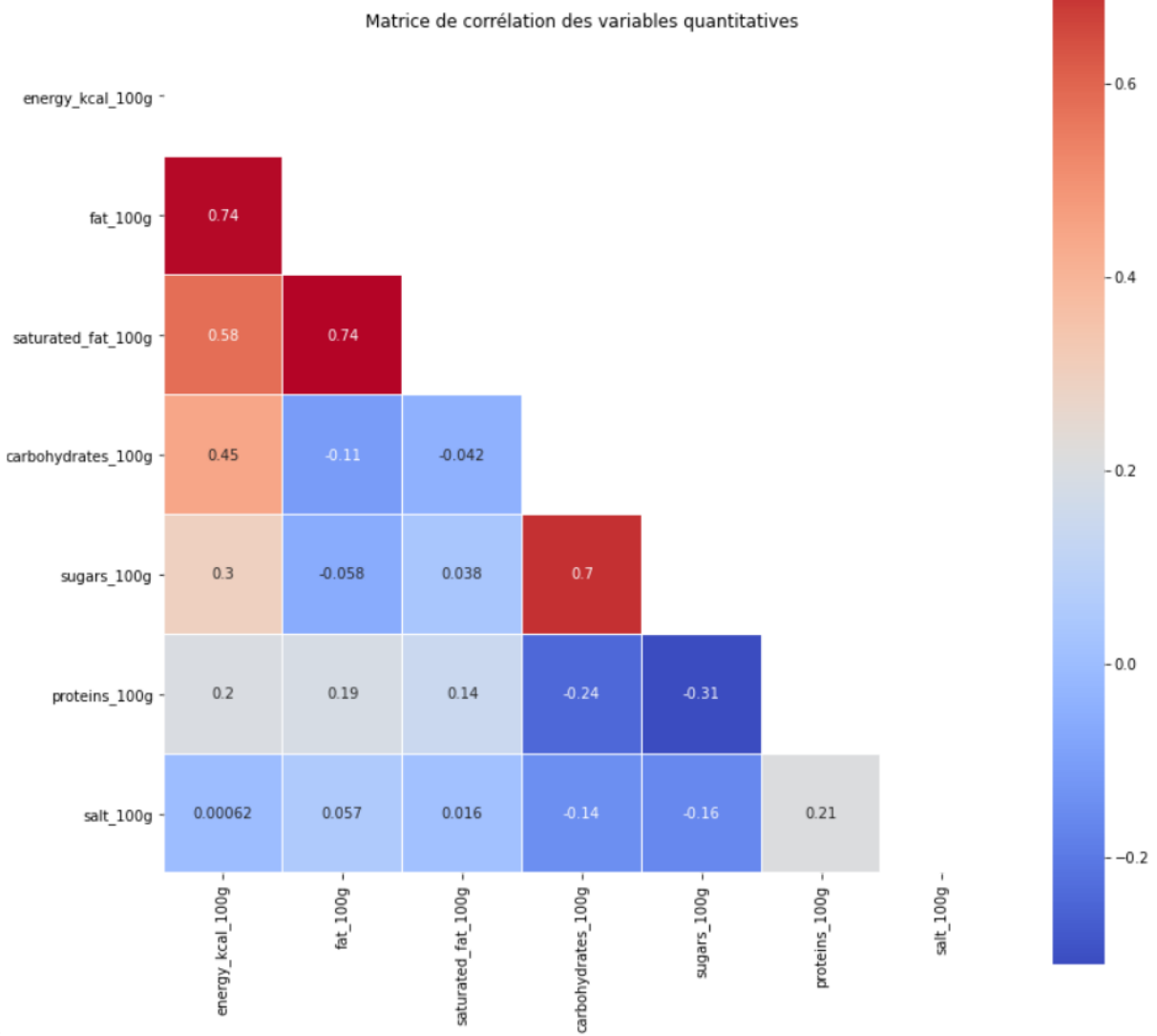
RELATION AU NUTRI-GRADE

Pairplot



Santé publique France

ANALYSE
_MULTIVARIÉ



RELATION QUANT-QUANT

Matrice de Corrélation

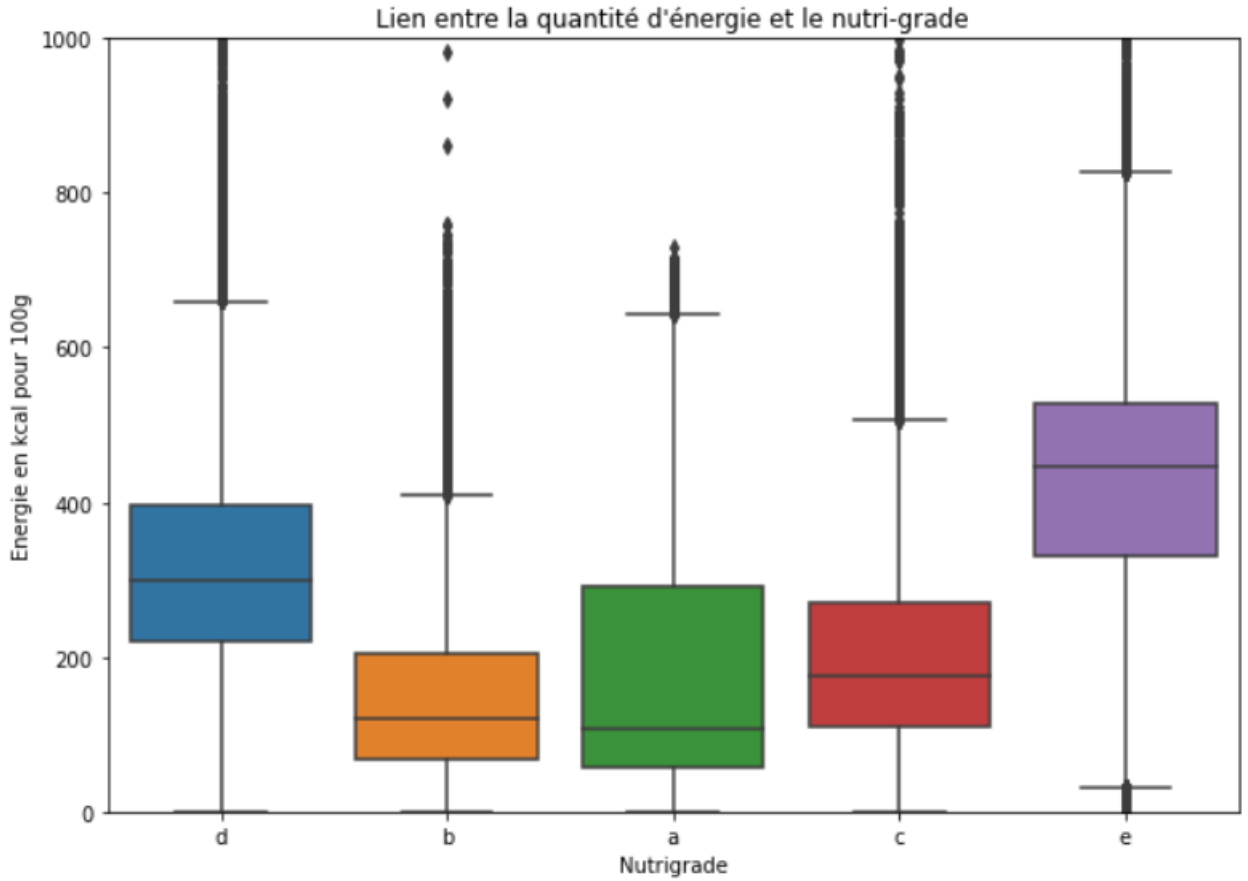
- **Energie:**
  - Lipides
  - Acides gras saturés
  - Glucides
- **Lipides:**
  - Acides gras saturés
- **Glucides:**
  - Sucres

Sante publique France

ANALYSE
_MULTIVARIÉ

RELATION QUANT-QUAL

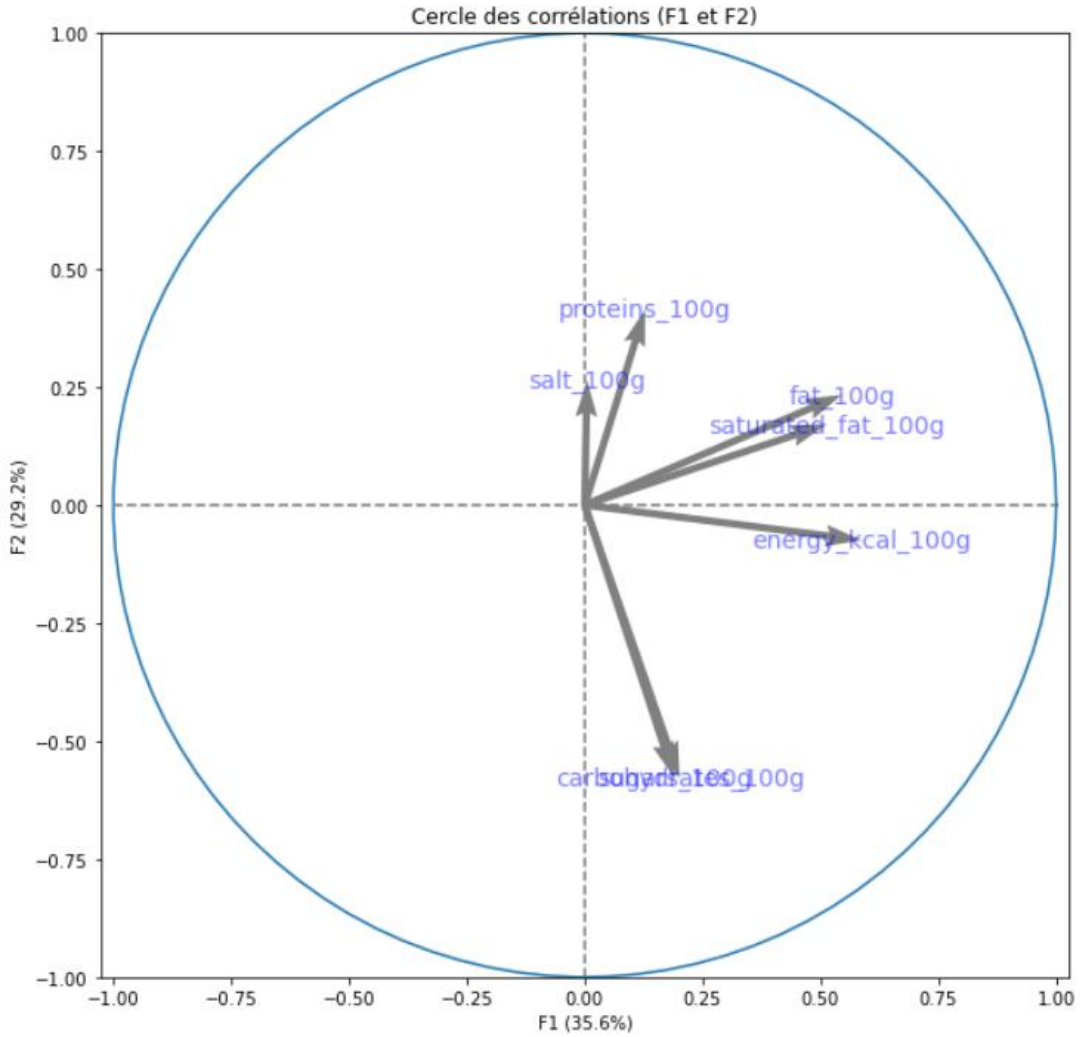
Anova



	Source	SS	DF	MS	F	p-unc	np2
0	nutriscore_grade	1.486602e+09	4	3.716505e+08	15883.074531	0.0	0.261117
1	Within	4.206630e+09	179777	2.339916e+04	NaN	NaN	NaN

Santé publique France

ANALYSE
_REDUCTION DE DIMENSION



PCA

Reduction de 7 à 2 dimensions

- **F1 (35,6%): Energie et Lipides (+Acides Gras)**
- **F2 (29,2%): Glucides (+Sucres) – Proteines -Sel**

Santé publique France

ANALYSE
_MACHINE LEARNING

KNN

Prévision du Nutri-grade

- Sample: 20000
- Train/Test: 80/20
- Accuracy: 75%

```
[[ 440   81   31    4    3]
 [  86  316   80   11    3]
 [  48  115  577  131   15]
 [  11   18  127 1011   66]
 [  23    7   30  100  666]]
              precision    recall  f1-score   support

           a       0.72      0.79      0.75       559
           b       0.59      0.64      0.61       496
           c       0.68      0.65      0.67       886
           d       0.80      0.82      0.81      1233
           e       0.88      0.81      0.84       826

    accuracy                           0.75      4000
   macro avg       0.74      0.74      0.74      4000
weighted avg       0.76      0.75      0.75      4000
```


Santé publique France

CONCLUSION
_FAISABILITÉ

- **Dataset: Nombre de produit élévé**
- **Composition: Accès aux valeurs nutritives**
- **Classements: Par categories, score, composition…**
- **Algorithme: Peut compléter les nutri-grade manquants**
- **Amélioration: Auto-completion des Nan (KNN)**

ANNEXES
_BOUCLE - FONCTION

```
# On supprime tous les produits ayant des valeurs pour 100g > 100 ou < 0
index3_ = ['fat_100g', 'saturated-fat_100g', 'carbohydrates_100g', 'sugars_100g', 'proteins_100g', 'salt_100g']
for i in index3_:
    df_clean = df_clean.drop(df_clean[(df_clean[i] < 0) | (df_clean[i] >= 100)].index)
```

```
df_clean3_cols = df_clean3.loc[:,['energy-kcal_100g','fat_100g', 'saturated-fat_100g', 'carbohydrates_100g', 'sugars_100g', 'proteins_100g', 'salt_100g']]
for cols in df_clean3_cols:
        df_clean3[cols].fillna(df_clean3.groupby('pnns_groups_2')[cols].transform('median'), inplace=True)
```


Santé publique France

ANNEXES
_FONCTION

```
def display_circles(pcs, n_comp, pca, axis_ranks, labels=None, label_rotation=0, lims=None):
    for d1, d2 in axis_ranks: # On affiche les 3 premiers plans factoriels, donc les 6 premières composantes
        if d2 < n_comp:

            # initialisation de la figure
            fig, ax = plt.subplots(figsize=(10,10))

            # détermination des limites du graphique
            if lims is not None :
                xmin, xmax, ymin, ymax = lims
            elif pcs.shape[1] < 30 :
                xmin, xmax, ymin, ymax = -1, 1, -1, 1
            else :
                xmin, xmax, ymin, ymax = min(pcs[d1,:]), max(pcs[d1,:]), min(pcs[d2,:]), max(pcs[d2,:])

            # affichage des flèches
            # s'il y a plus de 30 flèches, on n'affiche pas le triangle à leur extrémité
            if pcs.shape[1] < 30 :
                plt.quiver(np.zeros(pcs.shape[1]), np.zeros(pcs.shape[1]),
                   pcs[d1,:], pcs[d2,:],
                   angles='xy', scale_units='xy', scale=1, color="grey")
                # (voir la doc : https://matplotlib.org/api/_as_gen/matplotlib.pyplot.quiver.html)
            else:
                lines = [[[0,0],[x,y]] for x,y in pcs[[d1,d2]].T]
                ax.add_collection(LineCollection(lines, axes=ax, alpha=.1, color='black'))

            # affichage des noms des variables
            if labels is not None:
                for i,(x, y) in enumerate(pcs[[d1,d2]].T):
                    if x >= xmin and x <= xmax and y >= ymin and y <= ymax :
                      plt.text(x, y, labels[i], fontsize='14', ha='center', va='center', rotation=label_rotation, color="blue", alpha=0.5)

            # affichage du cercle
            an = np.linspace(0, 2 * np.pi, 100)  # Add a unit circle for scale
            plt.plot(np.cos(an), np.sin(an))
            plt.axis('equal')

            # définition des limites du graphique
            plt.xlim(xmin, xmax)
            plt.ylim(ymin, ymax)

            # affichage des lignes horizontales et verticales
            plt.plot([-1, 1], [0, 0], color='grey', ls='--')
            plt.plot([0, 0], [-1, 1], color='grey', ls='--')

            # nom des axes, avec le pourcentage d'inertie expliqué
```

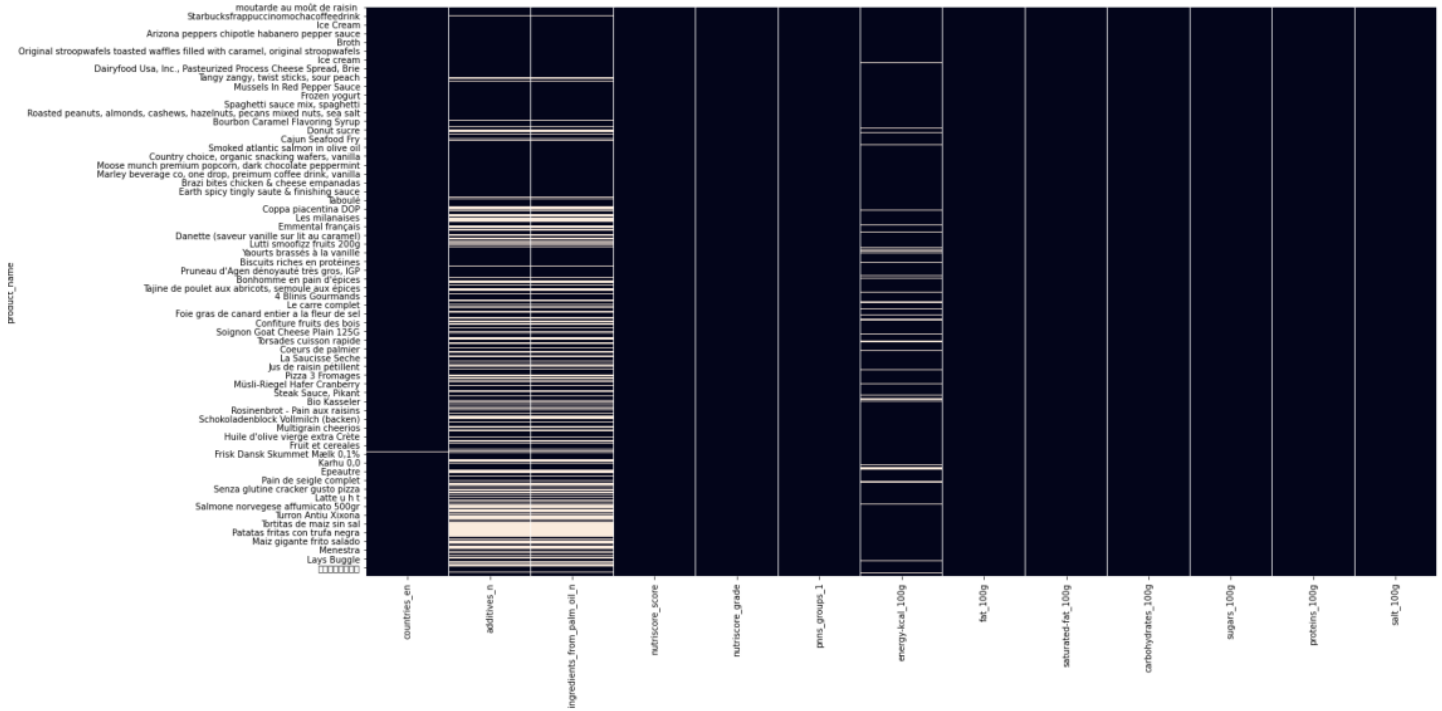

Santé publique France

ANNEXES
_COMMANDES UTILISÉES

- **df = pd.read_csv()** # importation de fichier csv
- **df.columns** # nom colonnes
- **df.dtypes** # types données
- **df.shape** # taille
- **df.sort_values()** # trie par ordre
- **df.isna().mean()** # taux valeurs manquantes
- **sns.heatmap / sns.histplot()** # graphiques sns
- **df.boxplot()** # graphique python
- **df.isin()** # présence élt dans dataframe (boolean)
- **df.drop() / df.dropna()** # Supprime () / Supprimer Na ()
- **df.interpolate()** # méthode pour remplir NaN
- **df.describe()** # statistiques dataframes
- **df.head** # affiche 15 premières lignes
- **df[df[]>x]** # mask de sélection
- **df.loc[]** # sélection []
- **df.duplicates()** # identification des duplicatas

ANNEXES

_NETTOYAGE




```
nutriscore_score                0.000000
```

```
nutriscore_grade                0.000000
pnns_groups_1                   0.000000
fat_100g                        0.000002
proteins_100g                   0.000002
sugars_100g                     0.000015
salt_100g                       0.000039
saturated-fat_100g              0.000048
carbohydrates_100g              0.000325
countries_en                    0.001096
energy-kcal_100g                0.060155
additives_n                     0.289007
ingredients_from_palm_oil_n     0.289007
dtype: float64
```

Après nettoyage

Santé publique France

ANNEXES

_INDICATEURS: STATISTIQUES

	additives_n	ingredients_from_palm_oil_n	nutriscore_score	energy-kcal_100g	fat_100g	saturated-fat_100g	carbohydrates_100g	sugars_100g	proteins_100g	salt_100g
count	118002.000000	118002.000000	195743.000000	179782.000000	195742.000000	195742.000000	195682.000000	195742.000000	195743.000000	195740.000000
mean	1.845096	0.050347	9.464405	269.672612	13.948252	5.410349	25.541238	12.883045	8.634954	1.051574
std	2.469573	0.221624	8.786371	177.953920	15.843627	7.700983	25.912209	18.308370	8.630960	2.309968
min	0.000000	0.000000	-15.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	2.000000	119.000000	1.500000	0.300000	3.100000	0.800000	2.100000	0.100000
50%	1.000000	0.000000	10.000000	248.000000	8.555000	2.200000	14.000000	3.500000	6.400000	0.610000
75%	3.000000	0.000000	16.000000	392.000000	22.000000	7.800000	49.000000	18.200000	12.300000	1.300000
max	30.000000	2.000000	40.000000	999.000000	99.990000	99.300000	99.950000	99.950000	99.000000	99.000000

Santé publique France