In [ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import missingno as msno
import plotly.express as px
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.neighbors import KNeighborsClassifier
from sklearn.decomposition import PCA
pd.options.display.max_rows = 100
pd.options.mode.copy_on_write = True
```

In [ ]:
```
df = pd.read_csv('fr.openfoodfacts.org.products.csv',sep = '\t',low_memory=False
df.describe()
```

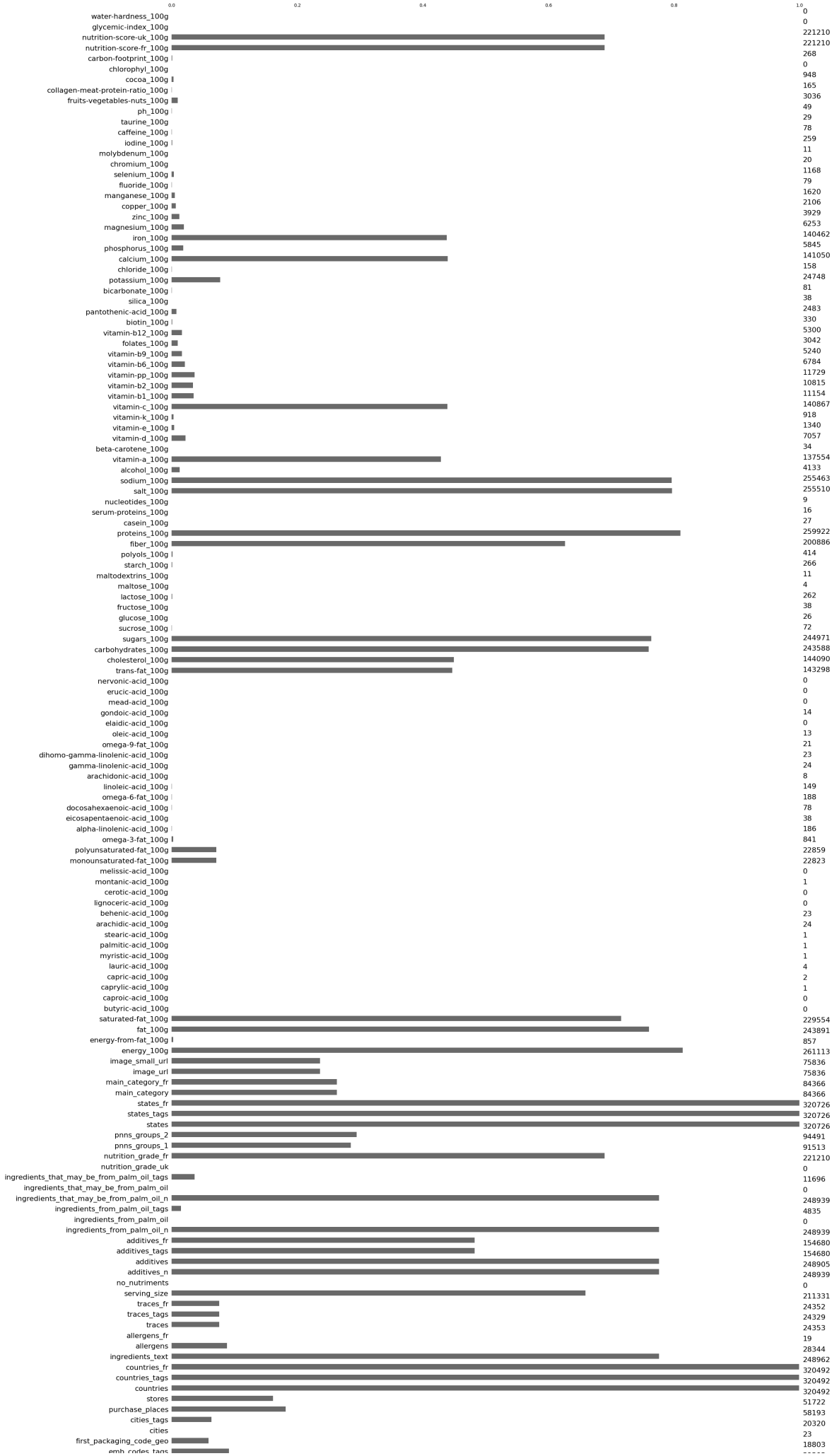
Out[ ]:

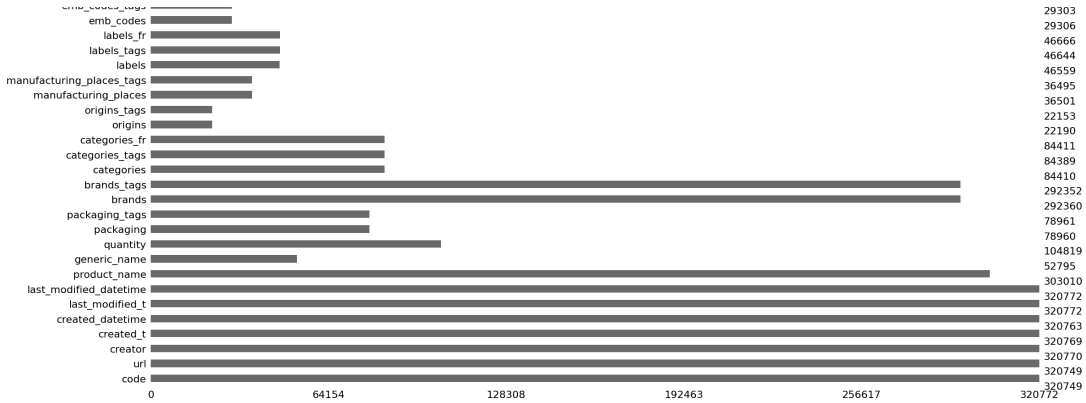
	no_nutriments	additives_n	ingredients_from_palm_oil_n	ingredients_from_palr
count	0.0	248939.000000	248939.000000	
mean	NaN	1.936024	0.019659	
std	NaN	2.502019	0.140524	
min	NaN	0.000000	0.000000	
25%	NaN	0.000000	0.000000	
50%	NaN	1.000000	0.000000	
75%	NaN	3.000000	0.000000	
max	NaN	31.000000	2.000000	

8 rows × 106 columns

In [ ]:
```
%matplotlib inline
msno.bar(df)
```

Out[ ]: <Axes: >


0.0
0.2
0.4
0.6
0.8
1.0
water-hardness_100g 0
glycemic-index_100g 0
nutrition-score-uk_100g 221210
nutrition-score-fr_100g 221210
carbon-footprint_100g 268
chlorophyl_100g 0
cocoa_100g 948
collagen-meat-protein-ratio_100g 165
fruits-vegetables-nuts_100g 3036
ph_100g 49
taurine_100g 29
caffeine_100g 78
iodine_100g 259
molybdenum_100g 11
chromium_100g 20
selenium_100g 1168
fluoride_100g 79
manganese_100g 1620
copper_100g 2106
zinc_100g 3929
magnesium_100g 6253
iron_100g 140462
phosphorus_100g 5845
calcium_100g 141050
chloride_100g 158
potassium_100g 24748
bicarbonate_100g 81
silica_100g 38
pantothenic-acid_100g 2483
biotin_100g 330
vitamin-b12_100g 5300
folates_100g 3042
vitamin-b9_100g 5240
vitamin-b6_100g 6784
vitamin-pp_100g 11729
vitamin-b2_100g 10815
vitamin-b1_100g 11154
vitamin-c_100g 140867
vitamin-k_100g 918
vitamin-e_100g 1340
vitamin-d_100g 7057
beta-carotene_100g 34
vitamin-a_100g 137554
alcohol_100g 4133
sodium_100g 255463
salt_100g 255510
nucleotides_100g 9
serum-proteins_100g 16
casein_100g 27
proteins_100g 259922
fiber_100g 200886
polyols_100g 414
starch_100g 266
maltodextrins_100g 11
maltose_100g 4
lactose_100g 262
fructose_100g 38
glucose_100g 26
sucrose_100g 72
sugars_100g 244971
carbohydrates_100g 243588
cholesterol_100g 144090
trans-fat_100g 143298
nervonic-acid_100g 0
erucic-acid_100g 0
mead-acid_100g 0
gondoic-acid_100g 14
elaidic-acid_100g 0
oleic-acid_100g 13
omega-9-fat_100g 21
dihomo-gamma-linolenic-acid_100g 23
gamma-linolenic-acid_100g 24
arachidonic-acid_100g 8
linoleic-acid_100g 149
omega-6-fat_100g 188
docosahexaenoic-acid_100g 78
eicosapentaenoic-acid_100g 38
alpha-linolenic-acid_100g 186
omega-3-fat_100g 841
polyunsaturated-fat_100g 22859
monounsaturated-fat_100g 22823
melissic-acid_100g 0
montanic-acid_100g 1
cerotic-acid_100g 0
lignoceric-acid_100g 0
behenic-acid_100g 23
arachidic-acid_100g 24
stearic-acid_100g 1
palmitic-acid_100g 1
myristic-acid_100g 1
lauric-acid_100g 4
capric-acid_100g 2
caprylic-acid_100g 1
caproic-acid_100g 0
butyric-acid_100g 0
saturated-fat_100g 229554
fat_100g 243891
energy-from-fat_100g 857
energy_100g 261113
image_small_url 75836
image_url 75836
main_category_fr 84366
main_category 84366
states_fr 320726
states_tags 320726
states 320726
pnns_groups_2 94491
pnns_groups_1 91513
nutrition_grade_fr 221210
nutrition_grade_uk 0
ingredients_that_may_be_from_palm_oil_tags 11696
ingredients_that_may_be_from_palm_oil 0
ingredients_that_may_be_from_palm_oil_n 248939
ingredients_from_palm_oil_tags 4835
ingredients_from_palm_oil 0
ingredients_from_palm_oil_n 248939
additives_fr 154680
additives_tags 154680
additives 248905
additives_n 248939
no_nutriments 0
serving_size 211331
traces_fr 24352
traces_tags 24329
traces 24353
allergens_fr 19
allergens 28344
ingredients_text 248962
countries_fr 320492
countries_tags 320492
countries 320492
stores 51722
purchase_places 58193
cities_tags 20320
cities 23
first_packaging_code_geo 18803
emb_codes_tags



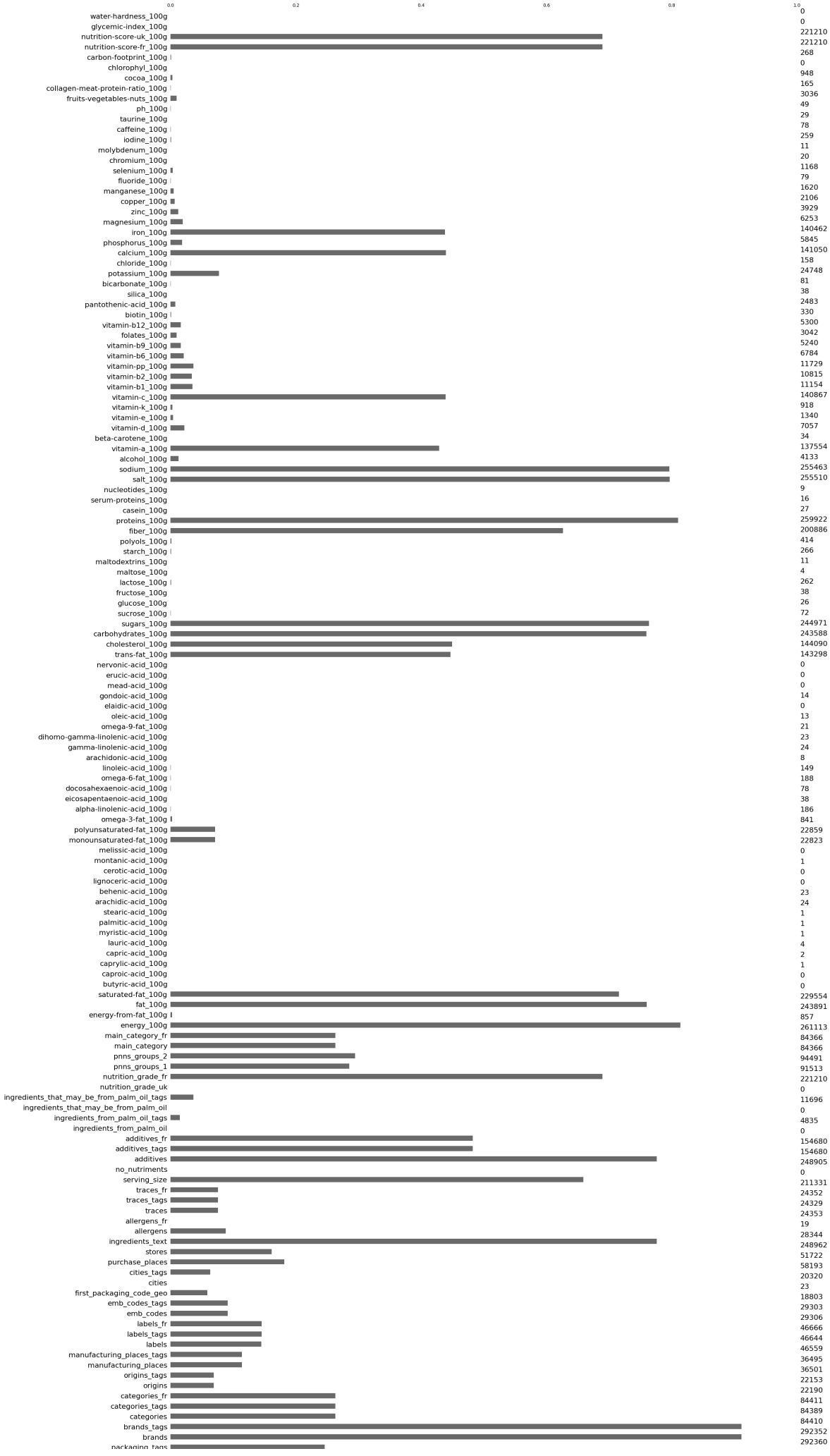
In [ ]:
```
df.columns
```

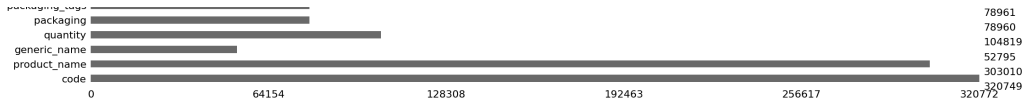
Out[ ]:
```
Index(['code', 'url', 'creator', 'created_t', 'created_datetime',
       'last_modified_t', 'last_modified_datetime', 'product_name',
       'generic_name', 'quantity',
       ...
       'ph_100g', 'fruits-vegetables-nuts_100g',
       'collagen-meat-protein-ratio_100g', 'cocoa_100g', 'chlorophyl_100g',
       'carbon-footprint_100g', 'nutrition-score-fr_100g',
       'nutrition-score-uk_100g', 'glycemic-index_100g',
       'water-hardness_100g'],
      dtype='object', length=162)
```

In [ ]:
```
df.drop(columns=['url', 'creator', 'created_t', 'created_datetime','last_modifie
cols_to_drop = df.filter(regex='_n$').columns
df.drop(columns=cols_to_drop, inplace=True)
cols_to_drop = df.filter(regex='^state').columns

df.drop(columns=cols_to_drop, inplace=True)
```

In [ ]:
```
msno.bar(df)
```

Out[ ]:
```
<Axes: >
```


0.0
0.2
0.4
0.6
0.8
1.0
water-hardness_100g 0
glycemic-index_100g 0
nutrition-score-uk_100g 221210
nutrition-score-fr_100g 221210
carbon-footprint_100g 268
chlorophyl_100g 0
cocoa_100g 948
collagen-meat-protein-ratio_100g 165
fruits-vegetables-nuts_100g 3036
ph_100g 49
taurine_100g 29
caffeine_100g 78
iodine_100g 259
molybdenum_100g 11
chromium_100g 20
selenium_100g 1168
fluoride_100g 79
manganese_100g 1620
copper_100g 2106
zinc_100g 3929
magnesium_100g 6253
iron_100g 140462
phosphorus_100g 5845
calcium_100g 141050
chloride_100g 158
potassium_100g 24748
bicarbonate_100g 81
silica_100g 38
pantothenic-acid_100g 2483
biotin_100g 330
vitamin-b12_100g 5300
folates_100g 3042
vitamin-b9_100g 5240
vitamin-b6_100g 6784
vitamin-pp_100g 11729
vitamin-b2_100g 10815
vitamin-b1_100g 11154
vitamin-c_100g 140867
vitamin-k_100g 918
vitamin-e_100g 1340
vitamin-d_100g 7057
beta-carotene_100g 34
vitamin-a_100g 137554
alcohol_100g 4133
sodium_100g 255463
salt_100g 255510
nucleotides_100g 9
serum-proteins_100g 16
casein_100g 27
proteins_100g 259922
fiber_100g 200886
polyols_100g 414
starch_100g 266
maltodextrins_100g 11
maltose_100g 4
lactose_100g 262
fructose_100g 38
glucose_100g 26
sucrose_100g 72
sugars_100g 244971
carbohydrates_100g 243588
cholesterol_100g 144090
trans-fat_100g 143298
nervonic-acid_100g 0
erucic-acid_100g 0
mead-acid_100g 0
gondoic-acid_100g 14
elaidic-acid_100g 0
oleic-acid_100g 13
omega-9-fat_100g 21
dihomo-gamma-linolenic-acid_100g 23
gamma-linolenic-acid_100g 24
arachidonic-acid_100g 8
linoleic-acid_100g 149
omega-6-fat_100g 188
docosahexaenoic-acid_100g 78
eicosapentaenoic-acid_100g 38
alpha-linolenic-acid_100g 186
omega-3-fat_100g 841
polyunsaturated-fat_100g 22859
monounsaturated-fat_100g 22823
melissic-acid_100g 0
montanic-acid_100g 1
cerotic-acid_100g 0
lignoceric-acid_100g 0
behenic-acid_100g 23
arachidic-acid_100g 24
stearic-acid_100g 1
palmitic-acid_100g 1
myristic-acid_100g 1
lauric-acid_100g 4
capric-acid_100g 2
caprylic-acid_100g 1
caproic-acid_100g 0
butyric-acid_100g 0
saturated-fat_100g 229554
fat_100g 243891
energy-from-fat_100g 857
energy_100g 261113
main_category_fr 84366
main_category 84366
pnns_groups_2 94491
pnns_groups_1 91513
nutrition_grade_fr 221210
nutrition_grade_uk 0
ingredients_that_may_be_from_palm_oil_tags 11696
ingredients_that_may_be_from_palm_oil 0
ingredients_from_palm_oil_tags 4835
ingredients_from_palm_oil 0
additives_fr 154680
additives_tags 154680
additives 248905
no_nutriments 0
serving_size 211331
traces_fr 24352
traces_tags 24329
traces 24353
allergens_fr 19
allergens 28344
ingredients_text 248962
stores 51722
purchase_places 58193
cities_tags 20320
cities 23
first_packaging_code_geo 18803
emb_codes_tags 29303
emb_codes 29306
labels_fr 46666
labels_tags 46644
labels 46559
manufacturing_places_tags 36495
manufacturing_places 36501
origins_tags 22153
origins 22190
categories_fr 84411
categories_tags 84389
categories 84410
brands_tags 292352
brands 292360
packaging_tags

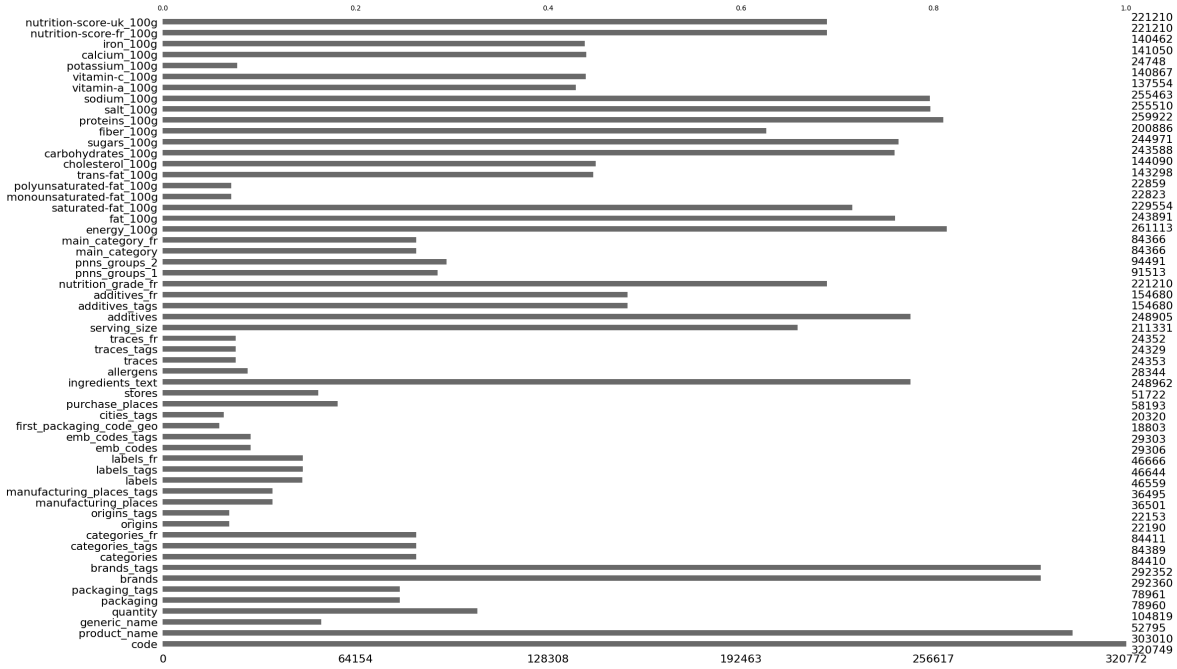


In [ ]:
```
# Calculer le nombre total de lignes
total_rows = len(df)

# Calculer le nombre de valeurs non nulles dans chaque colonne
non_null_counts = df.notna().sum()

# Calculer le taux de remplissage de chaque colonne
filling_rates = non_null_counts / total_rows * 100

# Filtrer les colonnes avec un taux de remplissage entre 20% et 50%
selected_columns = filling_rates[(filling_rates >= 5) ].index

# Conserver uniquement les colonnes sélectionnées
df_cleaned = df[selected_columns]
len(df_cleaned)
```

Out[ ]: 320772

In [ ]:
```
msno.bar(df_cleaned)
```

Out[ ]: <Axes: >



In [ ]:
```
df_cleaned.drop_duplicates(subset=['code'])
len(df_cleaned)
```

Out[ ]: 320772

In [ ]:
```
df_cible=df_cleaned.dropna(subset=['pnns_groups_1'])
df_cible_cleaned = df_cible[df_cible['pnns_groups_1']!= 'unknown']
len(df_cible)
```

Out[ ]: 91513

In [ ]:
```
df_pnns1 = df_cible_cleaned['pnns_groups_1']
df_pnns1
```

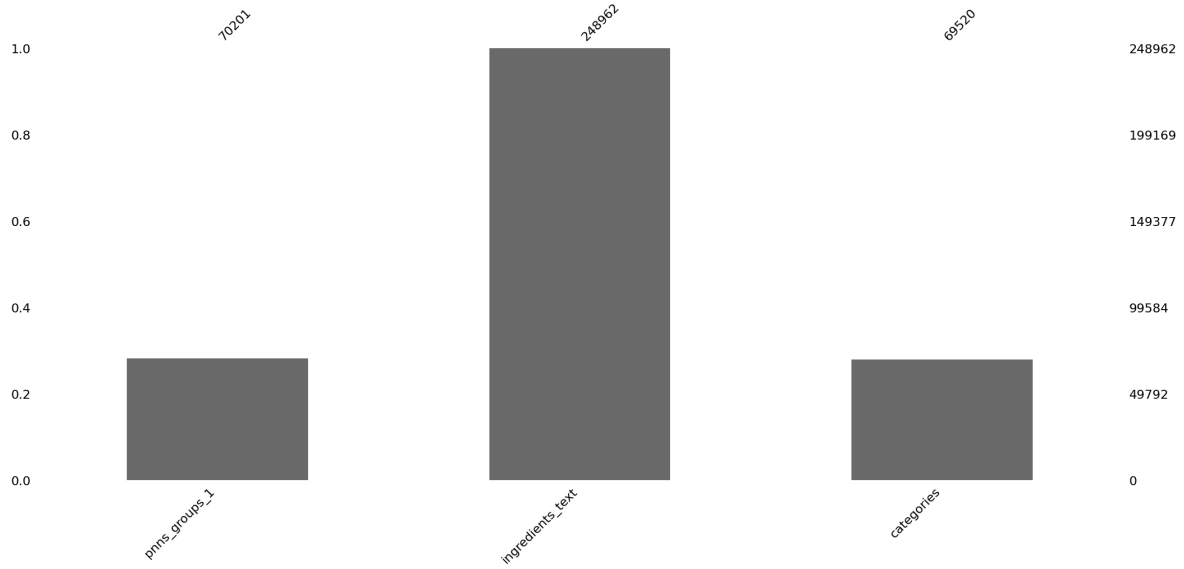
Out[ ]:
```
174       Fruits and vegetables
175               Sugary snacks
177        Cereals and potatoes
180               Sugary snacks
182        Cereals and potatoes
                   ...
320759    Fruits and vegetables
320763                Beverages
320765           Fish Meat Eggs
320766             Salty snacks
320769             Salty snacks
Name: pnns_groups_1, Length: 68889, dtype: object
```

In [ ]:
```
counts = df_pnns1.value_counts().reset_index()
counts.columns = ['pnns_groups_1', 'Counts']
print(df_pnns1.value_counts())
# Créer le graphique avec Plotly
fig = px.bar(counts, x='pnns_groups_1', y='Counts', title='Value Counts de la va

# Afficher le graphique
fig.show()
```

```
pnns_groups_1
Sugary snacks              12368
Beverages                   9033
Milk and dairy products     8825
Cereals and potatoes        8442
Fish Meat Eggs              8041
Composite foods             6747
Fruits and vegetables       5908
Fat and sauces              5216
Salty snacks                2809
fruits-and-vegetables        987
sugary-snacks                496
cereals-and-potatoes          16
salty-snacks                   1
Name: count, dtype: int64
```

In [ ]:
```
colonnes = ['pnns_groups_1','ingredients_text','categories']
df_select = df_cleaned.dropna(subset=['ingredients_text'])
df_travail = df_select[colonnes]

msno.bar(df_travail)
```

Out[ ]:
```
<Axes: >
```



remplacer les valeurs incohérentes

In [ ]:
```
df_travail['pnns_groups_1']=df_travail.pnns_groups_1.replace({"-":" "},regex=Tru
df_travail['pnns_groups_1']=df_travail['pnns_groups_1'].str.capitalize()
df_travail['pnns_groups_1'].value_counts()
```

Out[ ]:
```
pnns_groups_1
Unknown                  12366
Sugary snacks            10865
Milk and dairy products   7336
Cereals and potatoes      7152
Fish meat eggs            6980
Beverages                 6825
Fruits and vegetables     6032
Composite foods           6025
Fat and sauces            4226
Salty snacks              2394
Name: count, dtype: int64
```

enlever les unknown

In [ ]:
```
df_travail.drop(df_travail[df_travail['pnns_groups_1'] == 'Unknown'].index, inpl
df_travail['pnns_groups_1'].value_counts()
```

Out[ ]:
```
pnns_groups_1
Sugary snacks            10865
Milk and dairy products   7336
Cereals and potatoes      7152
Fish meat eggs            6980
Beverages                 6825
Fruits and vegetables     6032
Composite foods           6025
Fat and sauces            4226
Salty snacks              2394
Name: count, dtype: int64
```

In [ ]:
```
#df d'entrainement avec pnns group
df_train = df_travail.dropna(subset=['pnns_groups_1'])
#DF sans pnns group
df_missing = df_travail[df['pnns_groups_1'].isna()]
```

```
C:\Users\devil\AppData\Local\Temp\ipykernel_5268\1862913704.py:4: UserWarning:

Boolean Series key will be reindexed to match DataFrame index.
```

In [ ]:
```
vectorizer = CountVectorizer(tokenizer=lambda x: x.split(','), max_features=1000

X_train = vectorizer.fit_transform(df_train['ingredients_text'])
X_missing = vectorizer.transform(df_missing['ingredients_text'])

# Encodage des labels
y_train = df_train['pnns_groups_1']
# Affichage de la matrice
```

```
c:\Users\devil\anaconda3\envs\PROJET3\Lib\site-packages\sklearn\feature_extractio
n\text.py:523: UserWarning:

The parameter 'token_pattern' will not be used since 'tokenizer' is not None'
```

In [ ]:
```
# Initialisation du classifieur KNN
knn = KNeighborsClassifier(n_neighbors=3)

# Entraînement du modèle
knn.fit(X_train, y_train)
```

Out[ ]:
```
KNeighborsClassifier
KNeighborsClassifier(n_neighbors=3)
```

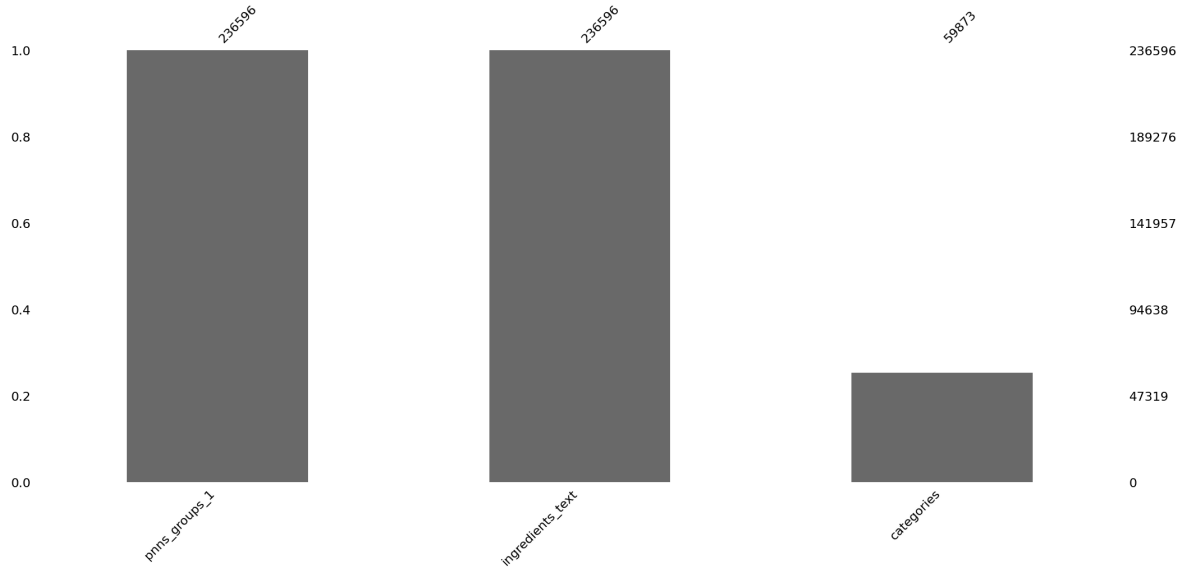
In [ ]:
```
# Prédictions sur les données manquantes
y_pred_missing = knn.predict(X_missing)

# Remplir les valeurs manquantes dans le DataFrame original
df_travail.loc[df_travail['pnns_groups_1'].isna(), 'pnns_groups_1'] = y_pred_mis

print("\nDataFrame avec valeurs manquantes prédites:")
print(df)
```

```
DataFrame avec valeurs manquantes prédites:
                code                                       product_name  \
0       0000000003087                                Farine de blé noir
1       0000000004530                 Banana Chips Sweetened (Whole)
2       0000000004559                                            Peanuts
3       0000000016087                          Organic Salted Nut Mix
4       0000000016094                                 Organic Polenta
...               ...                                             ...
320767  9948282780603                                  Tomato & ricotta
320768      99567453  Mint Melange Tea A Blend Of Peppermint, Lemon ...
320769  9970229501521                                          乐吧泡菜味薯片
320770  9980282863788                        Tomates aux Vermicelles
320771   999990026839                 Sugar Free Drink Mix, Peach Tea

                        generic_name quantity  packaging packaging_tags  \
0                                NaN      1kg        NaN            NaN
1                                NaN      NaN        NaN            NaN
2                                NaN      NaN        NaN            NaN
3                                NaN      NaN        NaN            NaN
4                                NaN      NaN        NaN            NaN
...                              ...      ...        ...            ...
320767                           NaN        1  Plastique      plastique
320768                           NaN      NaN  Plastique      plastique
320769  Leba pickle flavor potato chips     50 g  Plastique      plastique
320770                           NaN      67g        NaN            NaN
320771                           NaN      NaN  Plastique      plastique

                brands      brands_tags    categories  \
0       Ferme t'y R'nao  ferme-t-y-r-nao           NaN
1                   NaN              NaN           NaN
2        Torn & Glasser     torn-glasser           NaN
3             Grizzlies        grizzlies           NaN
4        Bob's Red Mill   bob-s-red-mill           NaN
...                 ...              ...           ...
320767         Panzani           panzani           NaN
320768     Trader Joe's     trader-joe-s           NaN
320769               乐吧               乐吧  Potato chips
320770            Knorr            knorr           NaN
320771    Market Pantry    market-pantry           NaN

                                       categories_tags  ... ph_100g  \
0                                                  NaN  ...     NaN
1                                                  NaN  ...     NaN
2                                                  NaN  ...     NaN
3                                                  NaN  ...     NaN
4                                                  NaN  ...     NaN
...                                                ...  ...     ...
320767                                             NaN  ...     NaN
320768                                             NaN  ...     NaN
320769  en:salty-snacks,en:appetizers,en:chips-and-fri...  ...     NaN
320770                                             NaN  ...     NaN
320771                                             NaN  ...     NaN

       fruits-vegetables-nuts_100g collagen-meat-protein-ratio_100g  \
0                              NaN                              NaN
1                              NaN                              NaN
2                              NaN                              NaN
3                              NaN                              NaN
4                              NaN                              NaN
...                            ...                              ...
```

```
320767                        NaN                                NaN
320768                        NaN                                NaN
320769                        NaN                                NaN
320770                        NaN                                NaN
320771                        NaN                                NaN

       cocoa_100g chlorophyl_100g carbon-footprint_100g  \
0             NaN            NaN                   NaN
1             NaN            NaN                   NaN
2             NaN            NaN                   NaN
3             NaN            NaN                   NaN
4             NaN            NaN                   NaN
...           ...            ...                   ...
320767        NaN            NaN                   NaN
320768        NaN            NaN                   NaN
320769        NaN            NaN                   NaN
320770        NaN            NaN                   NaN
320771        NaN            NaN                   NaN

       nutrition-score-fr_100g nutrition-score-uk_100g glycemic-index_100g  \
0                          NaN                    NaN                 NaN
1                         14.0                   14.0                 NaN
2                          0.0                    0.0                 NaN
3                         12.0                   12.0                 NaN
4                          NaN                    NaN                 NaN
...                        ...                    ...                 ...
320767                     NaN                    NaN                 NaN
320768                     0.0                    0.0                 NaN
320769                     NaN                    NaN                 NaN
320770                     NaN                    NaN                 NaN
320771                     NaN                    NaN                 NaN

       water-hardness_100g
0                      NaN
1                      NaN
2                      NaN
3                      NaN
4                      NaN
...                    ...
320767                 NaN
320768                 NaN
320769                 NaN
320770                 NaN
320771                 NaN

[320772 rows x 145 columns]
```

In [ ]:
```
msno.bar(df_travail)
```

Out[ ]:
```
<Axes: >
```



In [ ]:
```
df_travail['pnns_groups_1'].value_counts()
```

Out[ ]:
```
pnns_groups_1
Cereals and potatoes       75723
Beverages                  37575
Fat and sauces             36028
Sugary snacks              32117
Fruits and vegetables      13614
Milk and dairy products    12176
Fish meat eggs             11740
Salty snacks                9718
Composite foods             7905
Name: count, dtype: int64
```