Université de Nantes --- UFR Sciences et Techniques

Master Informatique parcours Apprentissage et Traitement Automatique de la Langue ATAL

Année 2021-2022

Analyse de Données

DIALLO Mamadou

27 Novembre 2020

Sommaire

I. Introduction

Ce projet consiste à faire une Analyse en Composantes Principales du sous-groupe alimentaire « fromages et assimilés » du fichier Table Ciqual 2020_FR_2020 07 07SsInf0.csv.

Une Analyse en Composantes Principales (ACP) est une des méthodes d'analyse factorielle qui sont des méthodes linéaires destinées à fournir des représentations plus synthétiques de l'information contenue dans des jeux de données volumineux.

L'Analyse en Composantes Principales permet de résumer l'information qu'on trouve dans le nuage de points : c'est-à-dire de percevoir les tendances, les relations entre les individus et les variables.

Son but est de trouver des espaces de dimensions plus petites minimisant la déformation du graphique obtenu lorsqu'on projette les données. L'intérêt étant entre autres :

- Localiser les regroupements d'individus ou de variables
- Détecter des individus exceptionnels
- Construire des variables synthétiques non corrélées

Nous allons étudier le contenu du fichier dans sa globalité (structure, dimension, …), nous effectuerons ensuite une étude générale du sous-groupe alimentaire « fromages et assimilés », puis nous effectuerons une étude approfondie à l'aide d'une Analyse en Composantes Principales sur le même jeu de données à savoir « fromages et assimilés » puis nous dégagerons de cette étude une interprétation sur le résultat obtenu.

II. Description du fichier

Le fichier Table Ciqual 2020_FR_2020 07.xls fournit la composition nutritionnelle des aliments consommés en France de la table Ciqual 2020. Ce fichier est un tableau croisé incluant 3186 aliments et 67 constituants : les aliments en ligne et les constituants en colonne. La teneur des données de compositions nutritionnelle est fourni pour 100 grammes de la partie comestible de l'aliment (c'est-à-dire sans les os pour la viande). Les valeurs

des teneurs manquantes sont représentées par « un tiret » : ces valeurs ne sont pas et ne doivent pas être assimilées à des zéro.

Voici un bref aperçu du fichier à l'aide de la fonction ***str()*** qui nous renvoi les différentes variables, indique leur type ainsi qu'un échantillon des premières valeurs.

Note : L'image suivante est une image tronquée pour des raisons de place

```
> str(Data)
'data.frame':   3186 obs. of  76 variables:
 $ alim_grp_code                                 : int  0 1 1 1 1 1 1 1 1 1 ...
 $ alim_ssgrp_code                               : int  0 101 101 101 101 101 101 101 101 101 ...
 $ alim_sssgrp_code                              : int  0 0 0 0 0 0 0 0 0 0 ...
 $ alim_grp_nom_fr                               : chr  "" "entrées et plats composés" "entrées et
 $ alim_ssgrp_nom_fr                             : chr  "" "salades composées et crudités" "salade
 $ alim_sssgrp_nom_fr                            : chr  "" "-" "-" "-" ...
 $ alim_code                                     : int  24999 25601 25602 25605 25606 25608 25609
 $ alim_nom_fr                                   : chr  "Dessert (aliment moyen)" "Salade de thon
nons à la grecque, appertisés" ...
 $ alim_nom_sci                                  : chr  "" "" "" "" ...
 $ Energie..Règlement.UE.N..1169.2011..kJ.100.g. : chr  "" "-" "-" "-" ...
 $ Energie..Règlement.UE.N..1169.2011..kcal.100.g.: chr  "" "-" "-" "-" ...
 $ Energie..N.x.facteur.Jones..avec.fibres...kJ.100.g.: chr  "" "-" "-" "-" ...
 $ Energie..N.x.facteur.Jones..avec.fibres...kcal.100.g.: chr  "" "-" "-" "-" ...
 $ Eau..g.100.g.                                 : chr  "45,4" "76,5" "76,7" "85,2" ...
 $ Protéines..N.x.facteur.de.Jones..g.100.g.     : chr  "4,63" "9,15" "8,06" "2,08" ...
 $ Protéines..N.x.6.25..g.100.g.                 : chr  "4,61" "9,15" "8,06" "2,08" ...
 $ Glucides..g.100.g.                            : chr  "36,6" "7,74" "6,4" "3,95" ...
 $ Lipides..g.100.g.                             : chr  "12,9" "4,7" "5,3" "3,55" ...
 $ Sucres..g.100.g.                              : chr  "23,7" "3,08" "1,9" "2,38" ...
 $ Fructose..g.100.g.                            : chr  "1,81" "0,82" "0,7" "0,7" ...
 $ Galactose..g.100.g.                           : chr  "" "-" "-" "-" ...
 $ Glucose..g.100.g.                             : chr  "2,18" "0,78" "0,6" "1" ...
 $ Lactose..g.100.g.                             : chr  "1,89" "-" "0,2" "0,3" ...
 $ Maltose..g.100.g.                             : chr  "1,07" "0,3" "0,2" "0,3" ...
 $ Saccharose..g.100.g.                          : chr  "15,7" "0,97" "0,6" "0,3" ...
 $ Amidon..g.100.g.                              : chr  "9,53" "4,1" "3,3" "0,99" ...
 $ Fibres.alimentaires..g.100.g.                 : chr  "1,54" "2,7" "2" "2,35" ...
 $ Polyols.totaux..g.100.g.                      : chr  "" "-" "-" "-" ...
 $ Cendres..g.100.g.                             : chr  "0,92" "1,79" "1,5" "1,65" ...
 $ Alcool..g.100.g.                              : chr  "0,081" "0" "0" "0" ...
 $ Acides.organiques..g.100.g.                   : chr  "0,083" "-" "-" "-" ...
 $ AG.saturés..g.100.g.                          : chr  "5,18" "0,56" "0,16" "0,23" ...
 $ AG.monoinsaturés..g.100.g.                    : chr  "4,74" "1,83" "3,27" "0,2" ...
 $ AG.polyinsaturés..g.100.g.                    : chr  "1,68" "1,76" "1,54" "1,67" ...
 $ AG.4.0..butyrique..g.100.g.                   : chr  "0,14" "0,05" "0,01" "-" ...
 $ AG.6.0..caproïque..g.100.g.                   : chr  "0,19" "0,05" "0,01" "-" ...
 $ AG.8.0..caprylique..g.100.g.                  : chr  "0,083" "0,05" "0,01" "-" ...
 $ AG.10.0..caprique..g.100.g.                   : chr  "0,14" "0,05" "0,01" "-" ...
 $ AG.12.0..laurique..g.100.g.                   : chr  "0,35" "0,05" "0,01" "-" ...
 $ AG.14.0..myristique..g.100.g.                 : chr  "0,55" "0,008" "0,01" "-" ...
 $ AG.16.0..palmitique..g.100.g.                 : chr  "2,47" "0,38" "0,12" "-" ...
 $ AG.18.0..stéarique..g.100.g.                  : chr  "0,99" "0,12" "0,04" "-" ...
 $ AG.18.1.9c..n.9...oléique..g.100.g.           : chr  "4,62" "-" "2,24" "-" ...
 $ AG.18.2.9c.12c..n.6...linoléique..g.100.g.    : chr  "1,32" "1,15" "1,08" "-" ...
 $ AG.18.3.c9.c12.c15..n.3...alpha.linolénique..g.100.g.: chr  "0,37" "0,056" "0,28" "0,026" ...
 $ AG.20.4.5c.8c.11c.14c..n.6...arachidonique..g.100.g.: chr  "0,012" "-" "0,016" "-" ...
 $ AG.20.5.5c.8c.11c.14c.17c..n.3..EPA..g.100.g. : chr  "0,0041" "0,008" "0,016" "-" ...
 $ AG.22.6.4c.7c.10c.13c.16c.19c..n.3..DHA..g.100.g.: chr  "0,0051" "0,039" "0,15" "-" ...
 $ Cholestérol..mg.100.g.                        : chr  "56,7" "19,2" "15,2" "0,11" ...
 $ Sel.chlorure.de.sodium..g.100.g.              : chr  "0,38" "1,11" "0,95" "1,26" ...
 $ Calcium..mg.100.g.                            : chr  "75" "20,7" "22" "27" ...
 $ Chlorure..mg.100.g.                           : chr  "178" "731" "584" "-" ...
 $ Cuivre..mg.100.g.                             : chr  "0,15" "0,1" "0,07" "0,17" ...
 $ Fer..mg.100.g.                                : chr  "1,45" "1,1" "0,7" "1,06" ...
 $ Iode..µg.100.g.                               : chr  "11" "2" "20" "2,46" ...
```

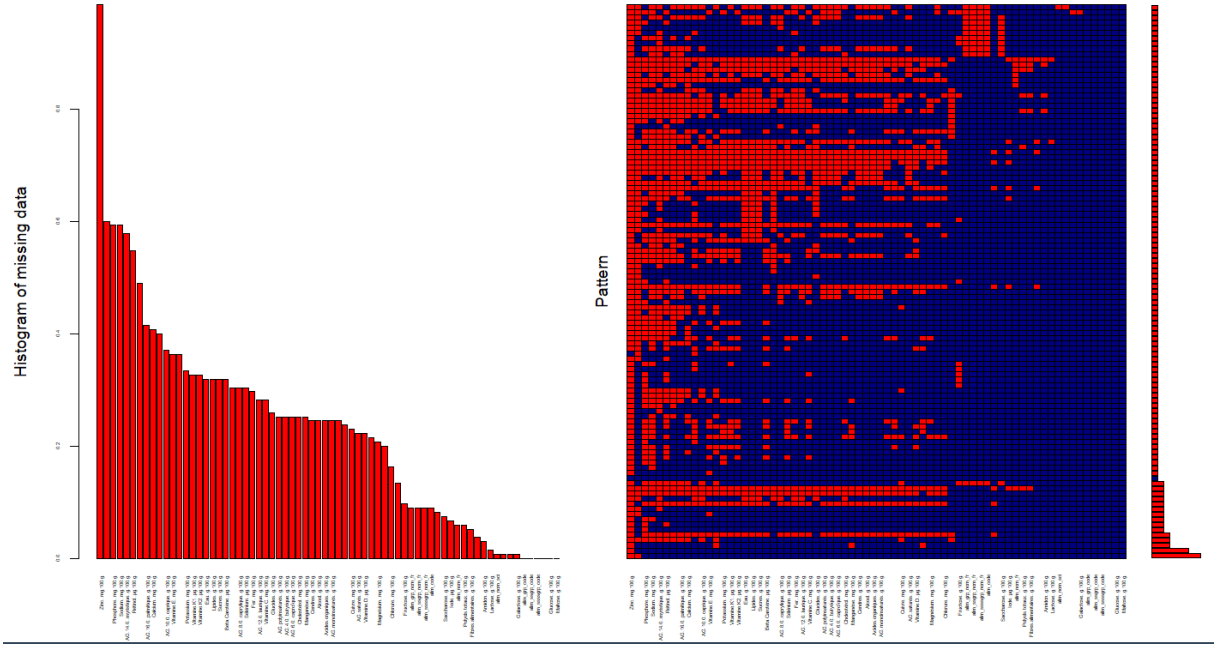
Dans certains cas, un constituant donné est détecté analytiquement, sans pouvoir être précisément quantifié. Le résultat analytique peut alors être communiqué comme « traces ».

Le terme « traces » peut aussi être utilisé en l'absence d'analyse quand un compilateur de données estime que la teneur d'un aliment en un constituant est très faible, mais ne peut être considérée nulle. La mention « traces » apparait alors.

III. Analyse « Fromages et assimilés » :

1. **Pré-traitement** : L'analyse commence par un tri préliminaire : Extraction du sous-groupe « Fromage et assimilés » du fichier global ceci nous donne un DataFrame de taille 135 * 76.

A ce niveau nous remarquons que nous avons plusieurs valeurs manquantes en ligne et en colonne comme le montre la figure suivante.



Ou encore observons ce résultat retourné par la fonction ***df_status()*** qui nous donne clairement le nombre de données manquantes et le pourcentage correspondant pour chaque variable(***Fichier tronqué mais suffisant pour prendre connaissance de l'information importante***)

```
> df_status(Frame)
                                                variable q_zeros p_zeros q_na  p_na q_inf p_inf    type unique
1          Energie..Règlement.UE.N..1169.2011..kJ.100.g.       0    0.00   12  8.89     0     0 numeric     69
2        Energie..Règlement.UE.N..1169.2011..kcal.100.g.       0    0.00   12  8.89     0     0 numeric     92
3     Energie..N.x.facteur.Jones..avec.fibres...kJ.100.g.       0    0.00   12  8.89     0     0 numeric     72
4   Energie..N.x.facteur.Jones..avec.fibres...kcal.100.g.       0    0.00   12  8.89     0     0 numeric     94
5                                       Eau..g.100.g.       0    0.00    8  5.93     0     0 numeric    105
6               Protéines..N.x.facteur.de.Jones..g.100.g.       0    0.00    1  0.74     0     0 numeric     99
7                         Protéines..N.x.6.25..g.100.g.       0    0.00    1  0.74     0     0 numeric    100
8                                   Glucides..g.100.g.     108   80.00    1  0.74     0     0 numeric     24
9                                     Lipides..g.100.g.       0    0.00    0  0.00     0     0 numeric     95
10                                    Sucres..g.100.g.      63   46.67   18 13.33     0     0 numeric     25
11                                 Fructose..g.100.g.       3    2.22   43 31.85     0     0 numeric      9
12                                Galactose..g.100.g.       2    1.48   81 60.00     0     0 numeric      9
13                                  Glucose..g.100.g.       4    2.96   41 30.37     0     0 numeric      9
14                                  Lactose..g.100.g.       3    2.22   35 25.93     0     0 numeric     24
15                                  Maltose..g.100.g.       3    2.22   43 31.85     0     0 numeric      8
16                                Saccharose..g.100.g.      3    2.22   43 31.85     0     0 numeric      8
17                                   Amidon..g.100.g.     116   85.93   13  9.63     0     0 numeric      7
18                        Fibres.alimentaires..g.100.g.     129   95.56    1  0.74     0     0 numeric      6
19                             Polyols.totaux..g.100.g.     134   99.26    0  0.00     0     0 numeric      2
20                                    Cendres..g.100.g.       0    0.00    2  1.48     0     0 numeric    114
21                                    Alcool..g.100.g.     135  100.00    0  0.00     0     0 numeric      1
22                          Acides.organiques..g.100.g.      76   56.30   10  7.41     0     0 numeric     45
23                                 AG.saturés..g.100.g.       0    0.00    4  2.96     0     0 numeric     76
24                           AG.monoinsaturés..g.100.g.       0    0.00    7  5.19     0     0 numeric    120
25                           AG.polyinsaturés..g.100.g.       0    0.00    8  5.93     0     0 numeric     72
26                          AG.4.0..butyrique..g.100.g.       1    0.74   33 24.44     0     0 numeric     64
27                          AG.6.0..caproïque..g.100.g.       1    0.74   33 24.44     0     0 numeric     48
28                         AG.8.0..caprylique..g.100.g.       1    0.74   33 24.44     0     0 numeric     53
29                          AG.10.0..caprique..g.100.g.       1    0.74   30 22.22     0     0 numeric     74
30                          AG.12.0..laurique..g.100.g.       0    0.00   33 24.44     0     0 numeric     64
31                        AG.14.0..myristique..g.100.g.       0    0.00   34 25.19     0     0 numeric     84
32                         AG.16.0..palmitique..g.100.g.      0    0.00   34 25.19     0     0 numeric     96
33                          AG.18.0..stéarique..g.100.g.       0    0.00   34 25.19     0     0 numeric     80
34                  AG.18.1.9c..n.9...oléique..g.100.g.       1    0.74   41 30.37     0     0 numeric     85
35             AG.18.2.9c.12c..n.6...linoléique..g.100.g.      1    0.74   50 37.04     0     0 numeric     46
36 AG.18.3.c9.c12.c15..n.3...alpha.linolénique..g.100.g.       1    0.74   38 28.15     0     0 numeric     37
37  AG.20.4.5c.8c.11c.14c..n.6...arachidonique..g.100.g.       3    2.22   78 57.78     0     0 numeric     12
38          AG.20.5.5c.8c.11c.14c.17c..n.3..EPA..g.100.g.      14   10.37   56 41.48     0     0 numeric     20
39      AG.22.6.4c.7c.10c.13c.16c.19c..n.3..DHA..g.100.g.      17   12.59   66 48.89     0     0 numeric     11
40                              Cholestérol..mg.100.g.       0    0.00   29 21.48     0     0 numeric     87
41                     Sel.chlorure.de.sodium..g.100.g.      0    0.00    5  3.70     0     0 numeric     92
42                                  Calcium..mg.100.g.       0    0.00   11  8.15     0     0 numeric    105
43                                 Chlorure..mg.100.g.       0    0.00   54 40.00     0     0 numeric     72
44                                   Cuivre..mg.100.g.       0    0.00   33 24.44     0     0 numeric     34
45                                      Fer..mg.100.g.       0    0.00   32 23.70     0     0 numeric     47
46                                     Iode..µg.100.g.       0    0.00   34 25.19     0     0 numeric     46
47                                Magnésium..mg.100.g.       0    0.00   27 20.00     0     0 numeric     78
48                                Manganèse..mg.100.g.       4    2.96   55 40.74     0     0 numeric     26
49                                Phosphore..mg.100.g.       0    0.00   22 16.30     0     0 numeric     94
50                                Potassium..mg.100.g.       0    0.00   31 22.96     0     0 numeric     78
51                                 Sélénium..µg.100.g.       0    0.00   40 29.63     0     0 numeric     54
52                                   Sodium..mg.100.g.       0    0.00    9  6.67     0     0 numeric    114
53                                     Zinc..mg.100.g.       0    0.00   28 20.74     0     0 numeric     74
54                                   Rétinol..µg.100.g.      0    0.00   34 25.19     0     0 numeric     84
55                               Beta.Carotène..µg.100.g.      5    3.70   80 59.26     0     0 numeric     44
56                                  Vitamine.D..µg.100.g.      0    0.00   45 33.33     0     0 numeric     26
```

Dans cette analyse j'ai décidé de supprimer en premier les variables atteignant un certain seuil de valeurs manquantes ici soit toutes variables ayant plus de 10 variables ce qui se justifie par n'être plus représentatif de l'élément avec ce taux de pourcentage en données manquantes(environs 8%). On se retrouve dans ce cas avec des variables plus représentatifs : celles qui ont le plus de données à pouvoir être exploité.

Ceci ne résolvant pas complètement la problématique de données manquantes je me penche vers les individus en supprimant tous les individus dans le jeu de données ayant des valeurs manquantes. Ceci a été réalisé avec la fonction ***na.omit()*** . Toutes ces étapes me ramènent à un jeu de données de taille : 106 lignes et 15 variables.

Note importante : *A noter qu'afin de réaliser une analyse correcte des données, nous supprimons l'aliment moyen qui correspond ici à la 1ère ligne*

du jeu de données. Il est aussi important à noter que parmi les colonnes retenues dans le dataFrame nous supprimons une colonne qui provoque l'affichage de NAN dans la matrice d'inertie lors des calculs. Cette opération est effectuée avant le passage de la matrice de données à la fonction ACP.

Donc la taille du jeu de données final pour l'analyse devient : 105 lignes et 14 variables.

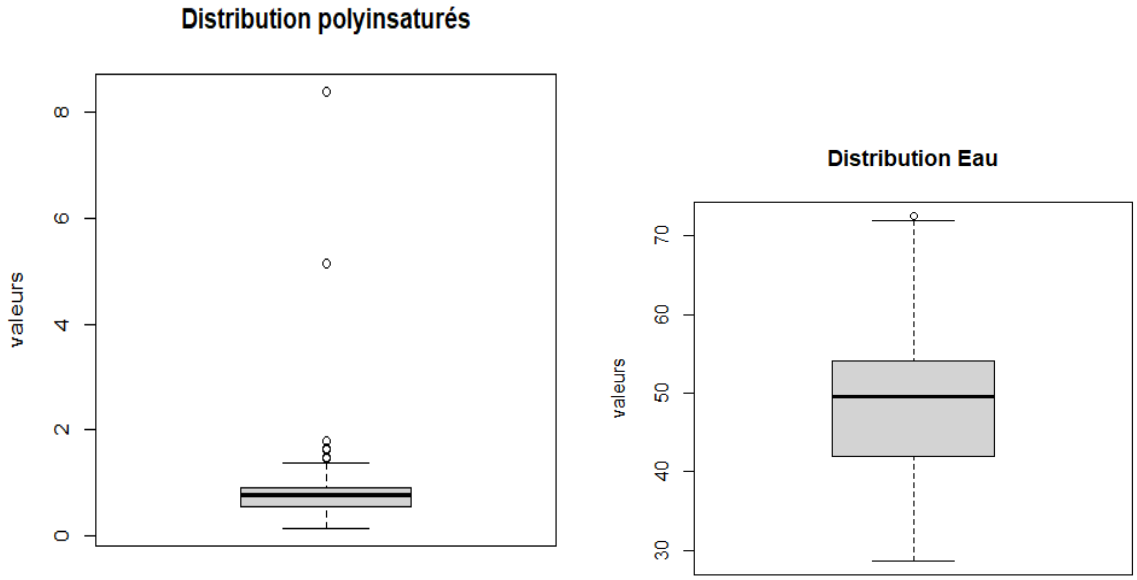
2. **Analyse Univariée** :

Voici quelques statistiques des variables à l'aide de la fonction ***summary()***

```
> summary(APICleaned)
Eau..g.100.g.    Protéines..N.x.facteur.de.Jones..g.100.g. Protéines..N.x.6.25..g.100.g. Glucides..g.100.g. Lipides..g.100.g.
Min.   : 29.0   Min.   :  5.0                            Min.   :  5.0                 Min.   :  0.00     Min.   : 10.0
1st Qu.:396.0   1st Qu.:178.0                            1st Qu.:165.0                 1st Qu.:  0.00     1st Qu.:225.0
Median :489.0   Median :207.0                            Median :202.0                 Median :  0.00     Median :254.0
Mean   :444.1   Mean   :220.1                            Mean   :201.2                 Mean   : 28.51     Mean   :236.4
3rd Qu.:534.0   3rd Qu.:239.0                            3rd Qu.:234.0                 3rd Qu.:  0.00     3rd Qu.:294.0
Max.   :725.0   Max.   :989.0                            Max.   :969.0                 Max.   :638.00     Max.   :384.0
Fibres.alimentaires..g.100.g. Polyols.totaux..g.100.g. Cendres..g.100.g. Alcool..g.100.g. Acides.organiques..g.100.g.
Min.   : 0.0000               Min.   :0.000000         Min.   :  3.0     Min.   :0        Min.   :  0.00
1st Qu.: 0.0000               1st Qu.:0.000000         1st Qu.:171.0     1st Qu.:0        1st Qu.:  0.00
Median : 0.0000               Median :0.000000         Median :267.0     Median :0        Median :  0.00
Mean   : 0.5143               Mean   :0.009524         Mean   :250.9     Mean   :0        Mean   : 31.73
3rd Qu.: 0.0000               3rd Qu.:0.000000         3rd Qu.:361.0     3rd Qu.:0        3rd Qu.: 61.00
Max.   :26.0000               Max.   :1.000000         Max.   :508.0     Max.   :0        Max.   :188.00
AG.saturés..g.100.g. AG.monoinsaturés..g.100.g. AG.polyinsaturés..g.100.g. Sel.chlorure.de.sodium..g.100.g.
Min.   : 15.0        Min.   :  5.0              Min.   :  4.00             Min.   :  1
1st Qu.:155.0        1st Qu.:309.0              1st Qu.: 45.00             1st Qu.:108
Median :175.0        Median :526.0              Median : 74.00             Median :145
Mean   :180.3        Mean   :481.4              Mean   : 79.91             Mean   :139
3rd Qu.:198.0        3rd Qu.:687.0              3rd Qu.: 87.00             3rd Qu.:184
Max.   :844.0        Max.   :955.0              Max.   :838.00             Max.   :368
Sodium..mg.100.g.
Min.   : 197.0
1st Qu.: 500.0
Median : 624.0
Mean   : 636.7
3rd Qu.: 755.0
Max.   :1470.0
```

Il est important de notifier ici que nous somme en absence de valeur NAN confirmation qui a été effectuée avec la fonction ***describe()*** . Ce résultat nous montre que nous avons des variables dont la distribution est proche de la normale(avec quelques variables qui se démarquent) et des variables

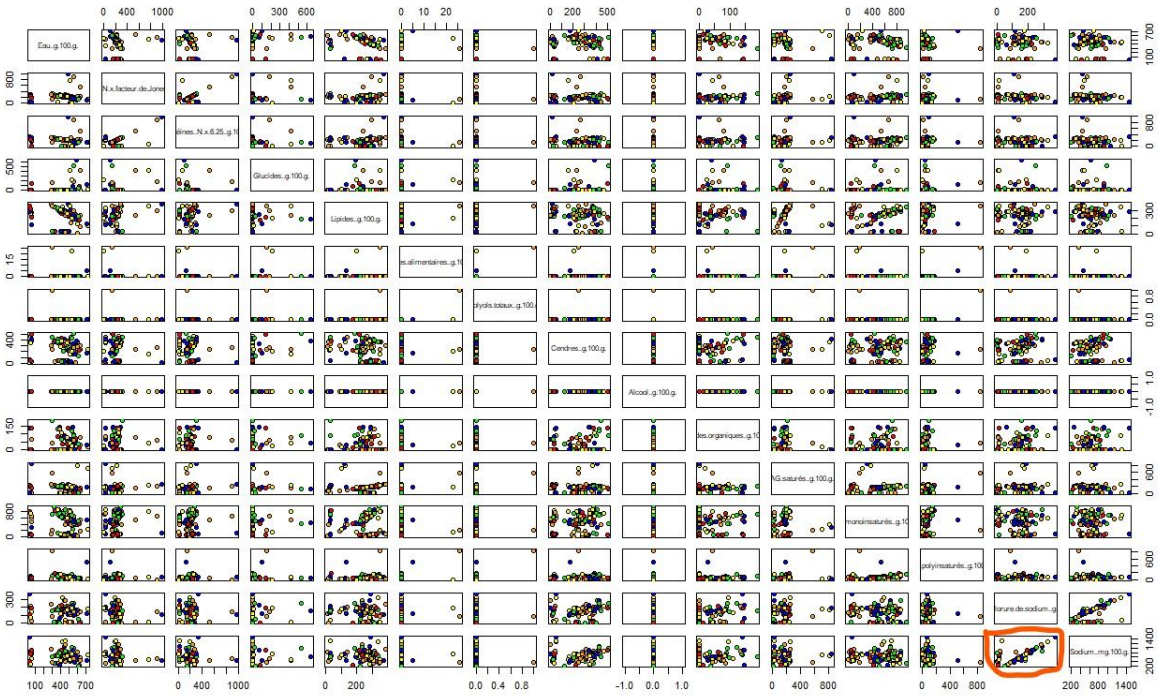
avec des valeurs extrêmes comme le montre les deux figures suivantes



3. **Analyse Bivariée** :

L'analyse univarié se concentrant uniquement sur les colonnes individuellement il est important d'avoir un bref aperçu sur les relations qui pourraient exister entre les variables.

Ci-dessous la figure de la description bivariée avec la fonction ***pairs()*** de R. On y voit des graphiques (symétriques par rapport à la diagonale) de chaque variable en relation avec toutes les autres variables. De ce fait on peut remarquer que des variables semblent évoluer ensemble (*exemple entouré sur la figure*)



4. **Analyse à l'aide d'une Analyse en Composantes Principales** :

Notre Analyse en Composantes Principales se réalise sur une matrice de données après traitement préalables : l'utilisateur de ma fonction doit manuellement analyser et traiter ses données afin d'en tirer le jeu de données qui lui convient qu'il passera à la fonction ACP.

a. **Choix du type d'Analyse en Composantes Principales** : Le choix effectué est une ACP Normé et donc en découle les valeurs de la matrice D et de la matrice Q et les étapes de la fonction sont :

- Centrage et Réduction de la matrice
- Calcul de la matrice d'inertie qui dans ce cas correspond à la matrice de corrélation
- Calcul de l'inertie du nuage de point Ig
- Calcul des valeurs propres et des vecteurs propres
- Demande du nombre d'axe à garder à l'utilisateur
- Calcul des coordonnées des individus et des variables en fonction de l'espace de représentation choisi
- Calcul de la contribution des individus et des variables pour chaque axe
- Calcul de la qualité de représentation des individus et des variables sur chaque axe

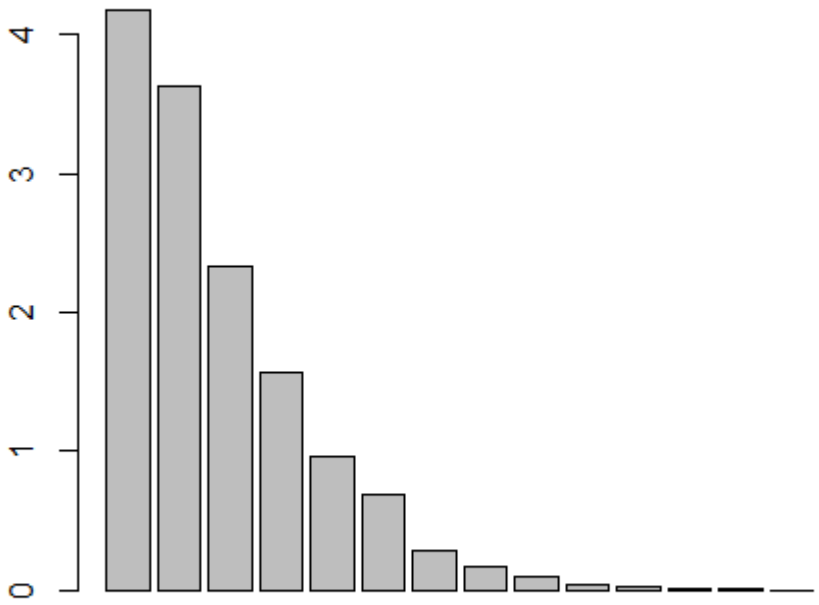
Ensuite j'invoque la fonction ACP pour un premier temps sans l'aliment moyen
ACP(dataMatrix)
Puis avec l'aliment moyen
ACP(dataMatrix2)

b. **Choix du nombre d'axe à retenir** :
Les méthodes servant dans le choix du nombre d'axe à retenir n'étant pas entièrement satisfaisantes, faisons une analyse afin de choisir le mieux dans notre cas :

i. **La règle de Kaiser** : préconise de conserver les axes correspondant aux valeurs propres supérieurs à 1. Avec la liste des valeurs propres obtenus affiché ci-dessous nous retiendrons 4 axes avec cette règle

```
> Eigvalues
[1] 4.1760685139 3.6293720313 2.3362417570 1.5723010510 0.9668606291 0.6817994329 0.2873333985 0.1679456248
[9] 0.0944837938 0.0376774853 0.0314039103 0.0122318934 0.0062236073 0.0000568713
```

ii. **La règle de coude** : Analysons le graphique ci-dessous des valeurs propres



Nous pouvons voir une cassure entre la 2ème valeur propre et la 3ème valeur propre, ce qui nous amènerait à garder les deux premiers axes.

iii. Pourcentage de l'inertie cumulé :

```
[1] "Pourcentages cumulées en fonction du nombre d'axe"
[1]  29.83  55.75  72.44  83.67  90.58  95.45  97.50  98.70  99.37  99.64  99.87  99.96 100.00 100.00
```

Cette méthode nous donne le pourcentage de l'information capté pour chaque nombre d'axe choisi.

Cette dernière méthode est plus parlante car elle nous montre le pourcentage de l'information que nous captons en choisissant un nombre d'axe.

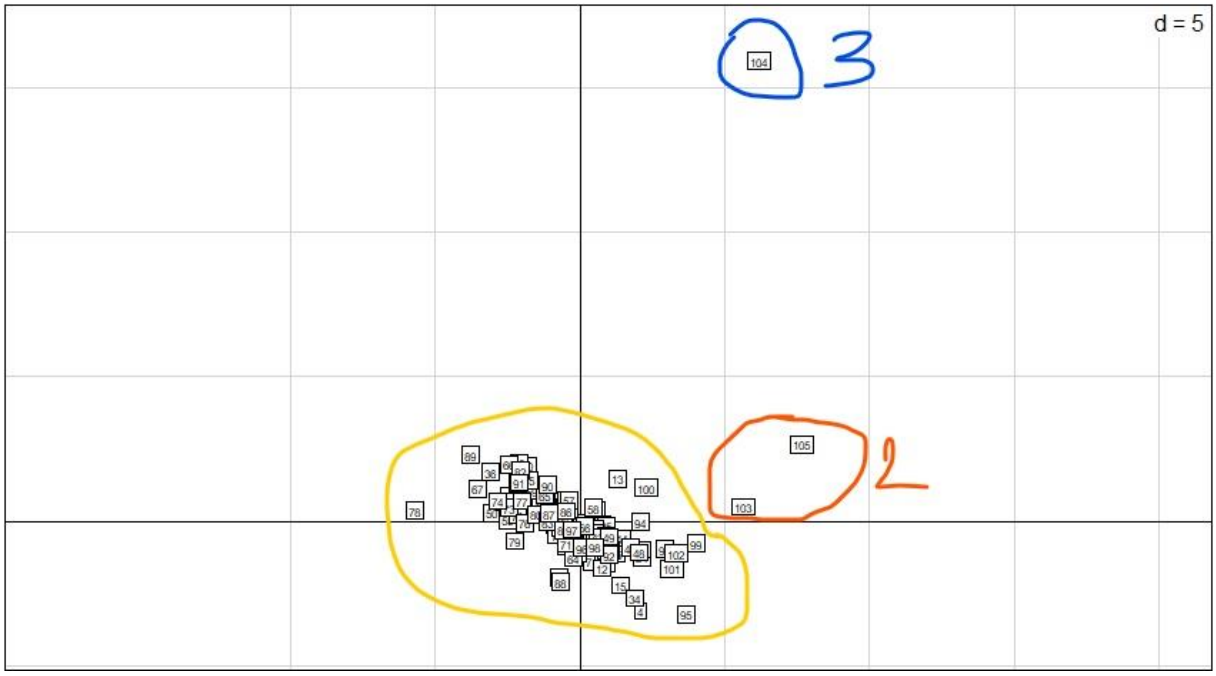
Les représentations suivantes le nombre d'axe choisi est 3 : 55,75% est peu et d'après les graphiques effectués le 4ème axe n'apporte que peu à la représentation.

IV. Interprétation et Vérification :

1. Interprétation :

a. Représentation des individus selon l'axe 1-2

On peut dénoter trois groupes sans pour autant en tirer de conclusions solides.

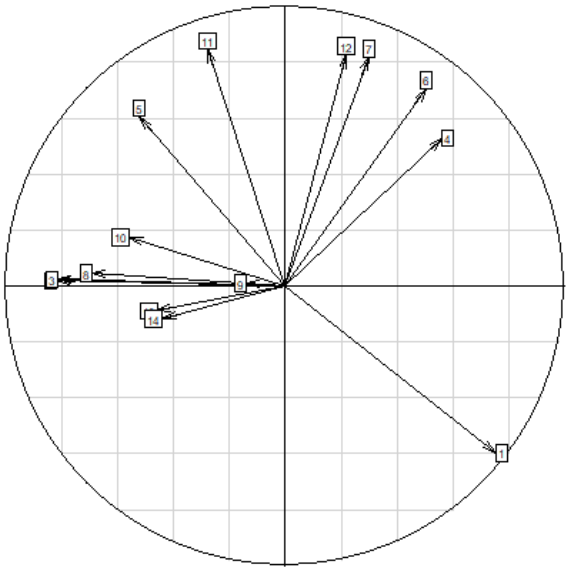


b. Représentation des variables : selon l'axe 1-2

On peut observer que les variables son plus ou moins bien représenté. On pourrait en tirer quantres groupes :

- La variable 1
- Les variables 12, 7, 6 et 4
- Les variables 11 et 5
- Et tout le reste

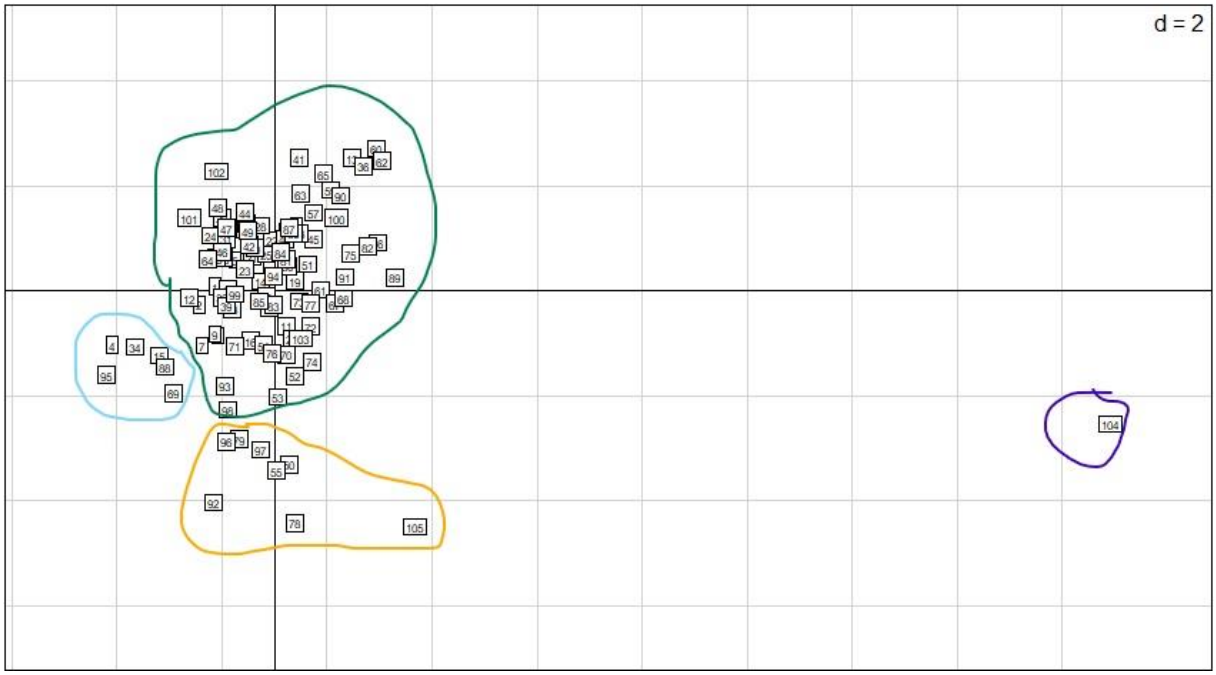
On peut observer que la variable 1 qui correspond à la variable **Eau** est négativement corrélée aux variables 12, 7, 6, 4 qui correspondent aux: **AG polyinsaturés, Polyols,Fibres alimentaires et Glucides**



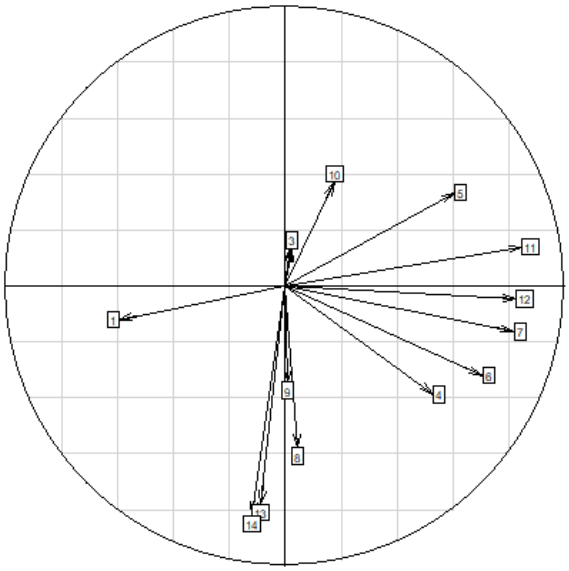
c. Représentation des individus : selon l'axe 2-3

On peut en ressortir quatres groupes comme montré sur la figure ci-dessous

Les individus entourés en bleu-ciel (4, 34, 69, 95, 88, 15) et l'individu 104 sont négativement corrélés

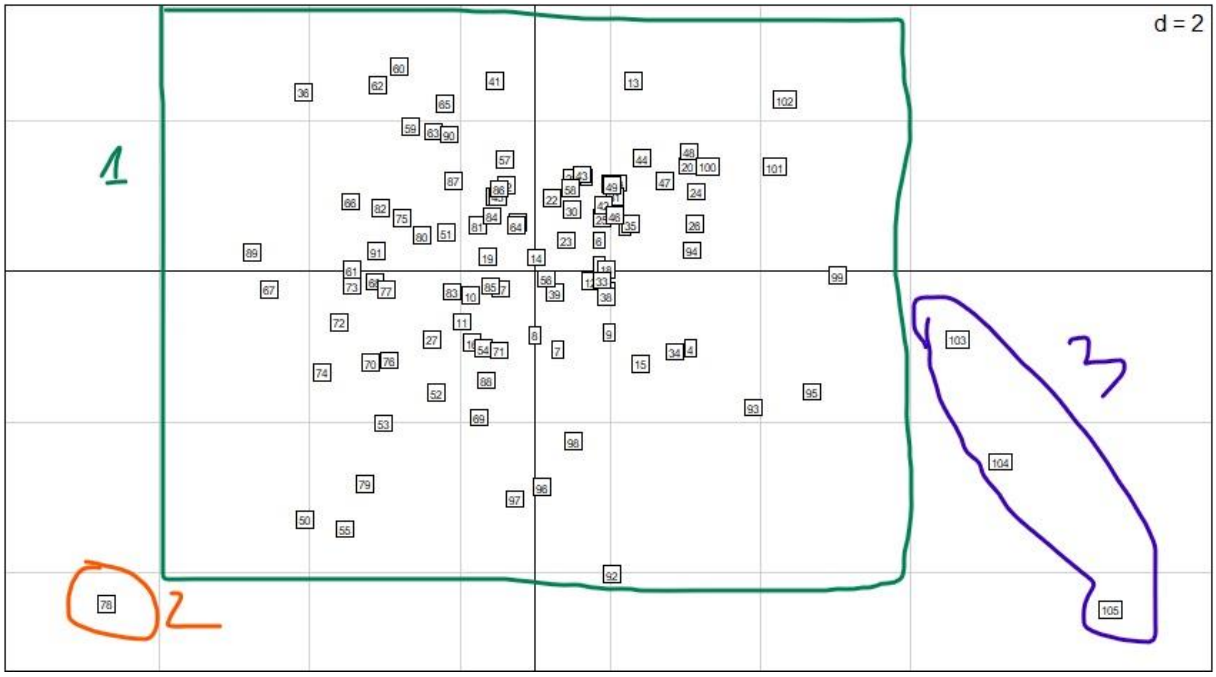


d. Représentation des variables selon l'axe 2-3 : on peut en tirer quatre groupes mais les variables étant éloignées du cercle on ne peut en dégager une interprétation conséquente. Cependant on peut encore remarqué que la variable 1 est toujours négativement corrélés par rapport aux variables 12,7,4,6

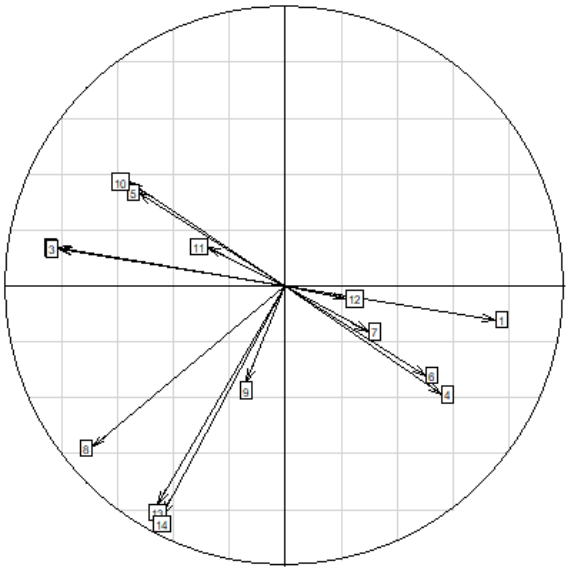


e. Représentation des individus selon l'axe 1-3 :

Nous pouvons en dégager 3 groupes et nous remarquerons que le groupe 3 est négativement corrélé par rapport au groupe 2



f. Représentation des variables selon l'axe 1-3 : Il y a peu de variables qui sont proche du cercle donc nous ne pouvons en tirer une interprétation conséquente

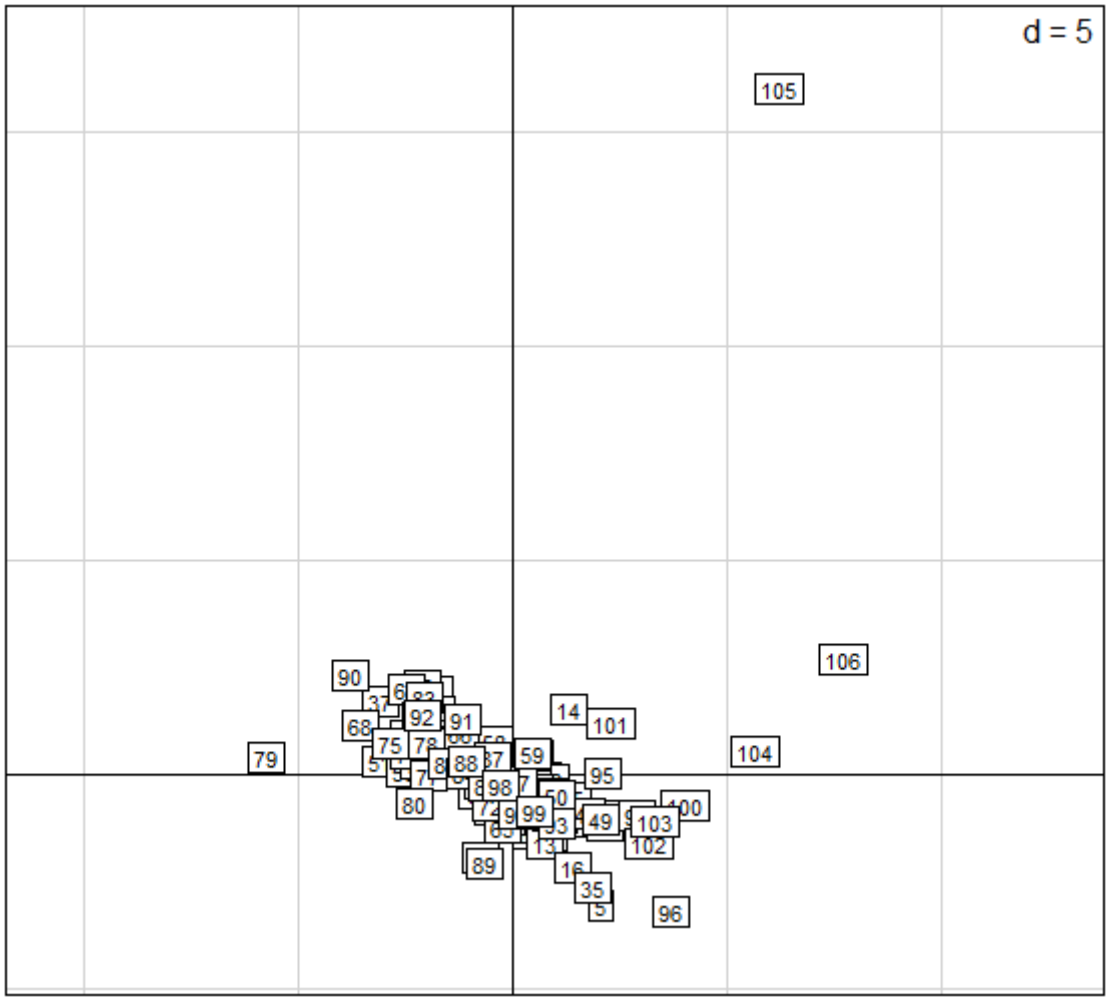


2. **Vérification** : On peut remarquer après calcul que la somme des contributions des individus sur chaque axe est égale à 1 ce qui vérifie bien la condition et nous permet de dégager des conclusions comme l'individu 105 est celui qui contribue le plus sur l'axe 1, l'individu 104 sur l'axe 2 et ensuite l'individu 105 sur l'axe 3(*Voir Variable indContrib retourné par la matrice*)

```
Combien d'axes souhaitez-vous gardé ? 3
La somme des contributions des individus sur l'axe1
[1] 1
La somme des contributions des individus sur l'axe2
[1] 1
La somme des contributions des individus sur l'axe3
[1] 1
```

V. Prise en compte du fromage moyen :

Après l'Analyse Composantes Principales effectuée, il nous faut rajouter le fromage moyen dans l'étude et le projeter et projeter. Nous ne voyons pas un grand changement car le point moyen est noyé dans l'amas de point autour du centre(voir figure suivante)



Et nous remarquons toujours que la somme des contributions sur les différents axes retenus est égale à 1 comme le montre la figure suivante :

```
Combien d'axes souhaitez-vous gardé ? 3
La somme des contributions des individus sur l'axe1
[1] 1
La somme des contributions des individus sur l'axe2
[1] 1
La somme des contributions des individus sur l'axe3
[1] 1
```

VI. Comparaison avec la fonction existante ACP :

- L'une des différences majeures et flagrantes entre la fonction et celle existante sur R est que ma fonction ACP fait une ACP Normée. Ce choix est du au fait du vouloir d'avoir une fonction générique qui marche et donne des résultats correctes lorsque les variables ont été mesurées dans des unités différentes.
- Ma fonction ACP permet à l'utilisateur de voir le pourcentage cumulée des axes avant son choix du nombre d'axe là ou la fonction existante demande simplement le nombre d'axe souhaité.

VII. Améliorations possibles :

- Avoir la possibilité de faire de l'ACP non Normé
- Affichage plus élégante des valeurs intermédiaires (les coordonnées des variables, les coordonnées des individus, la contribution des variables aux axes, …)

VIII. Conclusion :

L'Analyse en Composantes est une méthode puissante appartenant à la famille des analyses factorielles. Elle permet de transformer un tableau de données en un autre tableau plus facilement interprétable puisque de dimension réduite par rapport à l'initial. Ce nouveau tableau nous permet de représenter les variables et les observations graphiquement.

IX. Bibligraphie :

R documentation https://www.rdocumentation.org/

R Manual https://cran.r-project.org/manuals.html

Delladata https://delladata.fr/

X. Annexe :

Code R

```
install.packages("VIM")
install.packages("funModeling")
install.packages("summarytools")
library(funModeling)
library(VIM)

Data = read.csv("C:/Users/Mamadou/Documents/Cours/M1 ATAL/Analyse de données/projet/Analysis/Table Ciqual 2020_FR_2020 07 07SsInf0.csv", sep=";")
Data
str(Data)
API = subset(Data, Data$alim_ssgrp_nom_fr=="fromages et assimilés")
API
dim(API)
typeof(API)
is.data.frame(API)
#Df = API  %>% select(-(alim_grp_code:alim_nom_sci))
Frame = subset(API, select=(-c(alim_grp_code:alim_nom_sci)))

for ( iter in 1:ncol(Frame)){Frame[,iter] = as.numeric(gsub(",", ".", gsub("\\.", "", Frame[,iter])))}
FrameWithName = cbind(API$alim_nom_fr, API$alim_ssssgrp_nom_fr, API$alim_code, Frame) #we add useful columns
```

```
str(FrameWithName) #liste les variables, indique leur type ainsi qu'un echantillon des 1ères valeurs
NA_Studies = aggr(FrameWithName,
                  col=c('navyblue','red'),
                  numbers=TRUE,
                  sortVars=TRUE,
                  labels=names(data),
                  cex.axis=.4, gap=3,
                  ylab=c("Histogram of missing data","Pattern"))

summary(FrameWithName)

df_status(Frame) #connaitre le nombre de données manquantes et le pourcentage correspondant pour chaque variable

#columnToDelete = subset(Frame, df_status(Frame)$p_na > 10)
#remove column with more than 10 NA's values which represent
removenacol = function(mat){
 nacols = colSums(is.na(mat))
 for(i in ncol(mat):1){
  if(nacols[i] > 10){
   mat = mat[-c(i)]
  }
 }
 return(mat)
}
```

```
newDataFrame = removenacol(Frame)


newDataFrame

dim(newDataFrame)

#A ce niveau on décide de supprimer les colonnes avec %NA > 8

#Cette fonction est dans le rapport à citer mais à ne pas mettre dans le code

summarytools::descr(FrameWithName,

                    headings = FALSE,

                    transpose = TRUE)


#Analyse UNIDIMENSIONNELS

#Graphique: permettent de savoir si les suppositions(faites sur les variables) sont plus ou moins bien respectés c'est ce qu'on appelle analyse exploratoire des données EDA(Exploratory Data ANalysis) en anglais


boxplot(Frame$Energie..Règlement.UE.N..1169.2011..kJ.100.g., main="Analyse de la distribution de la variable Energie Règlement UE N1169")

boxplot(Frame$Beta.Carotène..µg.100.g.) #dissimetrie avec un <<fort>> étalement vers les grandes valeurs

boxplot(Frame$Sélénium..µg.100.g.)

boxplot(Frame$AG.polyinsaturés..g.100.g., main="Distribution de la variable AG Polyinsaturés")

boxplot(Frame$AG.monoinsaturés..g.100.g., main="Analyse de la distribution de la variable AG Monoinsaturés")

boxplot(Frame)
```

```
cleanData = na.omit(newDataFrame)

View(cleanData)

dim(cleanData) #maintenant on travaille avec une matrice de 106 lignes et 15 variables



APICleaned = cleanData[-1,] #suppression de la 1ère ligne qui correspond aux fromages moyens

dim(APICleaned)

View(APICleaned)



# ********** PART 2 ********** #

# ********** ANALYSE UNIDIMENSIONNELLE ********** #


summary(APICleaned)

describe(APICleaned) # a ce niveau on voit avec la colonne distinct  qu'on a des doublon et on peut voir la fréquence d'apparition avec freq

freq(APICleaned$Eau..g.100.g.)

plot(density(APICleaned$Eau..g.100.g.), main = "Eau")


boxplot(APICleaned, main="Ensemble des données", ylab="valeurs")

boxplot(APICleaned$Glucides..g.100.g.)

boxplot(APICleaned$AG.monoinsaturés..g.100.g.) #distribution proche de la normale
```

```
boxplot(Frame$AG.monoinsaturés..g.100.g., main="Distribution de la variable AG Monoinsaturés")

boxplot(APICleaned$AG.polyinsaturés..g.100.g., main="Distribution polyinsaturés", ylab="valeurs")

boxplot(APICleaned$Eau..g.100.g., main= "Distribution Eau", ylab="valeurs") #valeurs extrêmes avec des valeurs < 100 environs 5

boxplot(APICleaned$Fibres.alimentaires..g.100.g., main="Fibre alimentaires")

hist(APICleaned$Eau..g.100.g., )



# ********** PART 2 ********** #

# ********* ANALYSE BIDIMENSIONNELLE ********** #


pairs(APICleaned, c(APICleaned$Eau..g.100.g., APICleaned$Protéines..N.x.facteur.de.Jones..g.100.g., APICleaned$Protéines..N.x.6.25..g.100.g.,

        APICleaned$Glucides..g.100.g., APICleaned$Lipides..g.100.g., APICleaned$Fibres.alimentaires..g.100.g., APICleaned$Polyols.totaux..g.100.g.,

        APICleaned$Cendres..g.100.g., APICleaned$Alcool..g.100.g., APICleaned$Acides.organiques..g.100.g., APICleaned$AG.saturés..g.100.g.,

        APICleaned$AG.monoinsaturés..g.100.g., APICleaned$AG.polyinsaturés..g.100.g., APICleaned$Sel.chlorure.de.sodium..g.100.g., APICleaned$Sodium..mg.100.g.))


#quoi de mieux pour une analyse bidimensionnelle qu'un scatterplot via la fonction pairs de R

pairs(APICleaned, pch = 21, bg = c("red", "green", "blue"))
```

```
# ********************** PART 3 ************* #
# ******* DESCRIPTION MULTIVARIEE: ACP ********** #

#Center and Reduct
CenterReduct = function(Mat){
  return (scale(Mat, center = TRUE,
scale=TRUE)*sqrt(nrow(Mat)/(nrow(Mat)-1)))
}

APICleaned =  subset(APICleaned, select = -c(Alcool..g.100.g.)) #delete of alcool column it provoques a NAN values
cleanData = subset(cleanData, select = -c(Alcool..g.100.g.)) #Data with the mean chease

dim(cleanData)

dataMatrix = data.matrix(APICleaned) #transform APICleaned to Matrix
dataMatrix2 = data.matrix(cleanData)

Inertia = function(Mat){
  Q=matrix(0,nrow=ncol(Mat),ncol=ncol(Mat))
  diag(Q)=1
  D=matrix(0,nrow=nrow(Mat),ncol=nrow(Mat))
  diag(D)=1/nrow(Mat)
  return(t(Mat)%*%D%*%Mat%*%Q)
}
```

```
#valeurs propres
EigenValues = function(Mat){
 return(eigen(Mat)$values)
}

#vecteurs propres
EigenVectors = function(Mat){
 return(eigen(Mat)$vectors)

}

#Calcul des coordonnées des individus
Fcoordonate = function(Mat,vec){
 coord = matrix(0,nrow=nrow(Mat),ncol=ncol(vec))
 for(i in 1:ncol(vec)){
  coord[,i]=Mat%*%vec[,i]
 }
 return(coord)
}

#Calcul des coordonnées des variables
Gcoordonate = function(val,vec){
 coord=matrix(0,nrow=nrow(vec),ncol=length(val))
 for(i in 1:ncol(vec)){
  coord[,i]=sqrt(val[i])*vec[,i]
 }
```

```
  return(coord)
}

#contribution individu
indContribution = function(Mat, D, EigValues, Fco, axesNb){

  result = matrix(0, nrow = nrow(Mat), ncol = axesNb)
  for(i in 1:dim(Mat)[1]){
    for(k in 1:axesNb){
      result[i,k] = (D[i,i]*Fco[i,k]**2)/EigValues[k]
      #result[i,k] = (Fco[i,k]**2)/dim(Mat)[1]*EigValues[k]
    }
  }
  return(result)
}

#contribution variable
varContribution = function(Mat, Gco, EigValues, axesNb){
  result = matrix(0, nrow = dim(Mat)[2], ncol = axesNb)
  for(j in 1:dim(Mat)[2]){
    for(k in 1:axesNb){
      result[j,k] = (Gco[j,k]**2)/EigValues[k]
      #result[j,k] = (Gco[j,k]**2)/EigValues[k]
    }
  }
  return(result)
}
```

```
#calcul qualité representation des individus
indRepresentationQuality = function(Mat, Fco, axesNb){
 result = matrix(0, nrow = dim(Mat)[1], ncol = axesNb)
 for(i in 1:dim(Mat)[1]){
  somme = sum(Fco[i,]**2)

  for(k in 1:axesNb){
   result[i,k] = (Fco[i,k]**2)/somme
  }
 }
 return(result)
}

#calcul qualité representation des variables
varRepresentationQuality = function(Mat, Gco, axesNb){
 result = matrix(0, nrow = dim(Mat)[2], ncol = axesNb)
 for(j in 1:dim(Mat)[2]){
  somme = sum(Gco[j,]**2)

  for(k in 1:axesNb){
   result[j,k] = (Gco[j,k]**2)/somme
  }
 }
 return(result)
}
```

```
library("ade4")
ACP = function(Mat){

  Q=matrix(0,nrow=ncol(Mat),ncol=ncol(Mat)) #Qp Matrix
  diag(Q) = 1

  D=matrix(0,nrow=nrow(Mat),ncol=nrow(Mat)) #D = 1/n * In Matrix
  diag(D) = 1/nrow(Mat)



  Mcr = CenterReduct(Mat)
  View(Mcr)
  Mi = Inertia(Mcr)
  View(Mi)
  Ig = sum(diag(Mi)) #trace de la matrice d'inertie = trace matrice de
corrélation = p

  EigValues = EigenValues(Mi)
  View(EigValues)
  EigValues
  barplot(EigValues)
  plot(EigValues)

  EigVects = EigenVectors(Mi)
  View(EigVects)
```

```
print("Pourcentages cumulées en fonction du nombre d'axe")
print(round(cumsum((EigValues/sum(diag(Mi))) * 100), 2))

axes = readline("Combien d'axes souhaitez-vous gardé ? ")
axes = as.numeric(axes)

if( !(1 <= axes && axes <= dim(Mcr)[2]) ){
  print("Le nombre d'axes ne peut être inférieur à 1 ou supérieur au
nombre de colonnes")
  #print("Le nombre d'axe est fixé de ce pas à 2 par défaut")

}else{
  #Coordonnées Individu
  Fco = Fcoordonate(Mcr, EigVects)
  View(Fco)

  #Coordonnées Variables
  Gco = Gcoordonate(EigValues, EigVects)
  View(Gco)

  #Contribution individus
  indContrib = indContribution(Mat, D, EigValues, Fco, axes)
  View(indContrib)

  #Contribution variables
  varContrib  = varContribution(Mat, Gco, EigVects, axes)
```

```
View(varContrib)

#Qualité de la representation for les individus
indQuality = indRepresentationQuality(Mat, Fco, axes)
View(indQuality)

#Qualité de la representaiton pour les variables
varQuality = varRepresentationQuality(Mat, Gco, axes)
View(varQuality)


### Calcul de la somme des contributions des individus pour chaque axe retenu

for(i in 1:axes){
  message("La somme des contributions des individus sur l'axe",i)
  print(sum(indContrib[,i]))
}

#Graphique
#Visualisation des individus
s.label(Fco, xax=1, yax=2, label = 1:dim(Fco)[1], clabel = 0.7)

#Visualisation des variables
s.corcircle(Gco, xax =1, yax=2, label = 1:dim(Gco)[1], clabel = 0.7)

# plot(Fco,xlab="F1",ylab="F2")
```

```
    # text(Fco,cex=0.65,pos=3,labels=1:nrow(Mat))


  }


}

ACP(dataMatrix)

ACP(dataMatrix2) #ACP after adding the average food to the dataset
```