SY09 - Compte Rendu de projet 1

Anh Tu Nguyen et Marie Valmori

Jeudi 10 mai 2018

Table des matières

Introduction

Pour ce premier rendu de projet de l'UV SY09, nous avons appliqué plusieurs méthodes vues en cours, qui concernent la classification non supervisée. La classification non supervisée désigne un ensemble de méthodes dont l'objectif est de dresser et/ou de retrouver une typologie existante caractérisant un ensemble de n observations, à partir de p caractéristiques. Ainsi, pour cette première partie de l'UV SY09 et pour ce premier rendu de projet, nous avons procédé à une phase dite exploratoire. Les principaux outils utilisés sont donc les représentations graphiques, la construction de variables synthétiques et l'identification de groupes homogènes.

I- Cuisine

Dans cette première partie du projet, nous nous sommes intéressés à un jeu de données dans lequel était répartie l'utilisation d'ingrédients selon différentes zones géographiques. Une méthode exploratoire est nécessaire pour comprendre et s'approprier les données fournies. Plusieurs outils ont été utilisés pour leur analyse.

I.1- Analyse exploratoire des données

Le tableau de données est constitué de 26 observations et 50 variables. Ces variables étant toutes quantitatives, elles correspondent à la proportion utilisée de chacun des ingrédients dans les zones géographiques : plus les données sont proches de 1, plus l'ingrédient est utilisé dans la zone géographique. Dans le cas où le pays n'utilise pas l'ingrédient, la variable quantitative pour ce pays est de 0.

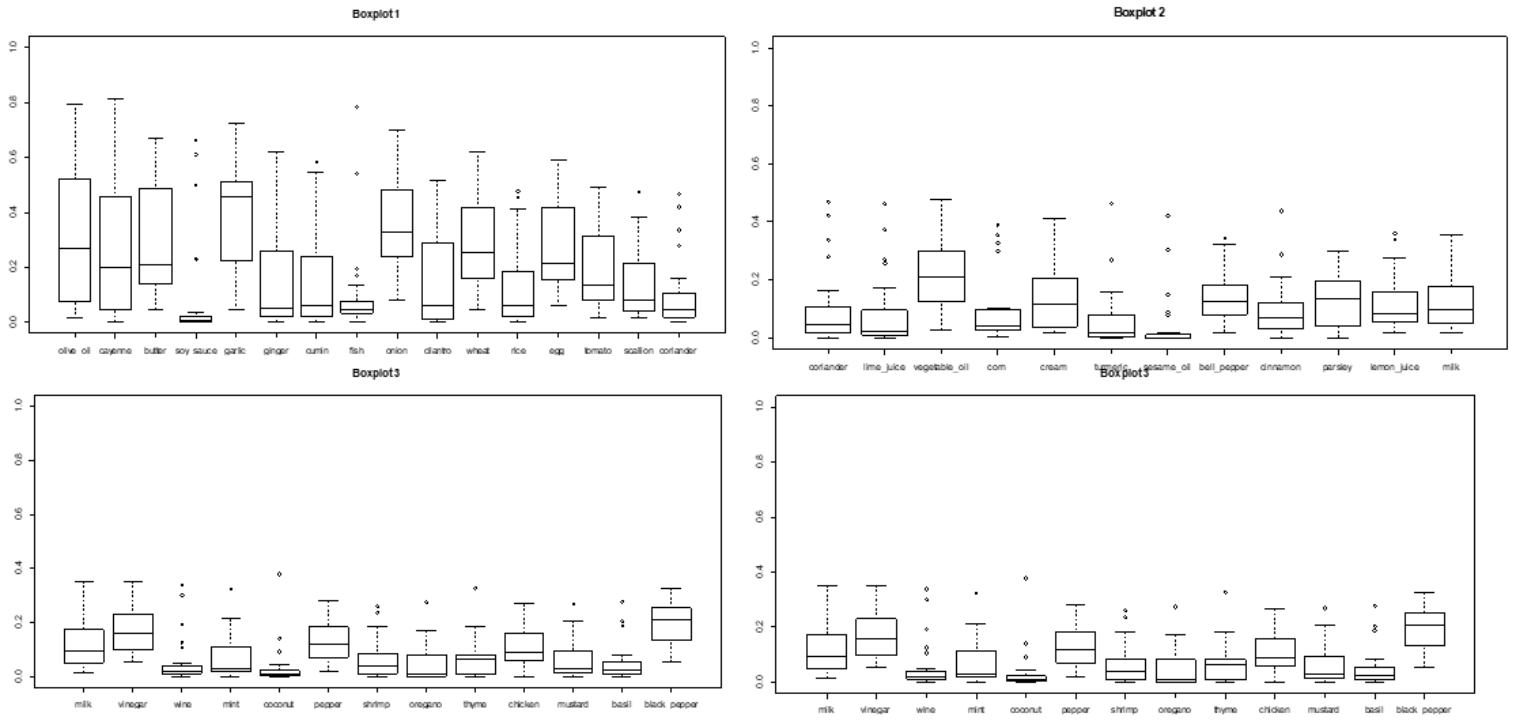


Figure 1- Boîtes à moustaches des données recettes

A partir des boites à moustaches figurées ci-dessus, on peut voir que l'utilisation des ingrédients demeure différente selon les pays. Par exemple pour l'ingrédient ail, (traduit garlic), on retrouve une médiane de 0.46, signifiant donc que 50% des individus, ou 50% des pays en l'occurrence, utilisent l'ail à hauteur de 0.46 sur une échelle de 1. D'autre part, dans le but d'étudier une quelconque corrélation entre les ingrédients, nous avons réalisé un graphique croisant chacun des ingrédients. Etant donné qu'il n'est pas ressortie un graphique exploitable, nous avons éalisé des tests de corrélations entre chacun des ingrédients, par la méthode de Pearson, puis par la suite nous avons réalisé un corrélogramme, présenté en Figure 2. Son interprétation se base sur la grosseur de la pastille présente dans le tableau croisé et sa couleur:

- plus le test est significatif plus la pastille est grosse,
- la couleur de la pastille varie selon la valeur du coefficient de corrélation R^2
- les cases vides symbolisent une corrélation non significative entre les ingrédients.

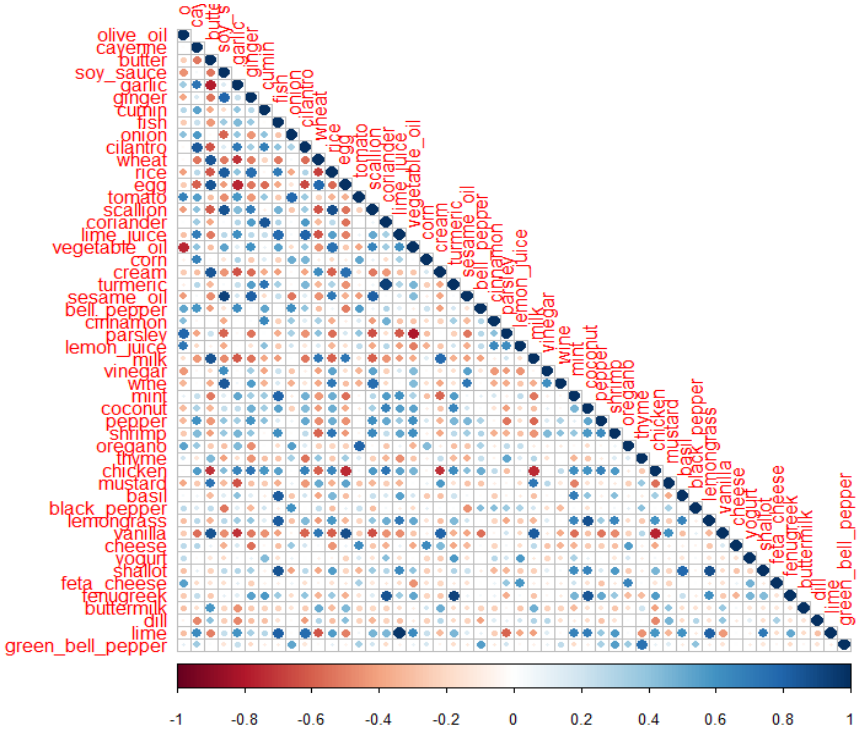


Figure 2- Corrélogramme des données recttes pour p-value=0.05

Par exemple, les pays utilisant l'huile d'olive dans leur recette n'utilisent pas d'huile végétale venant d'autres oléagineux et pour les pays utilisant le lait, du beurre est retrouvé dans la recette. Cette analyse met en évidence des familles d'ingrédients et des principales transformations des matières premières (avec du lait, les industries peuvent en faire du beurre). L'étude de cette corrélation est poursuivie par la réalisation d'analyse en composante principales dans la question 2.

Enfin, une information peu importante mais bonne à savoir les ingrédients, dans le tableau de données brutes, ont été rangés selon leur variance, de la plus grande à la plus petite.

I.2- Analyse en composantes principales

L'analyse en composantes principales (ACP) est un outil permettant d'analyser et de visualiser un jeu de données contenant des individus décrits par plusieurs variables.

C'est une méthode de visualisation d'une table multivariée. Dans notre cas, les individus sont les différentes origines et les variables sont les ingrédients. L'analyse en composantes principales demande certaines conditions, notamment d'avoir un nombre d'observations supérieur au nombre de variables. Cette condition n'est dès lors pas respectée car nous avons à faire à 50 variables pour 26 observations. La transformations du jeu de données à la main ainsi que l'application de la fonction *princomp*() n'est donc pas possible. C'est pourquoi, nous avons fait le choix d'utiliser la

fonction *PCA*() de la bibliothèque `FactoMineR`. A partir de cette bibliothèque, nous pouvons ainsi tracer le cercle de corrélations et étudier le pourcentage d'inertie des valeurs propres associées, montré en Figure 3. Dans le cadre de notre analyse, nous pouvons observer que les trois premières composantes principales expliquent 60% de la variation et 70% pour les quatre premières.

Le cercle de corrélations des variables expliqué par les deux premières dimensions nous montre ainsi la corrélation entre les variables. L'analyse ACP se fait ainsi : un cos2 élevé indique une bonne représentation de la variable, et est d'autant plus importante que sa couleur vire au rouge, des variables sont d'autant plus corrélées que leur valeur de cos2 est importante et que leur direction sont les mêmes. Ainsi, pour le jeu de données recettes, on observe une corrélation entre les variables lait et beurre, qui sont des variables dites contributives, car ont une valeur importante de cos2. On retrouve donc les corrélations établies en question 1 dans le corrélogramme (Figure 2). Un autre exemple : les ingrédients *lime* et *lime juice* sont fortement corrélés. Enfin, en affichant les individus sur ce cercle de corrélations, on peut voir la répartition des pays selon l'utilisation des ingrédients dans leurs recettes. Par exemple, pour les pays asiatiques, on va avoir une utilisation assez massive de la sauce soja.

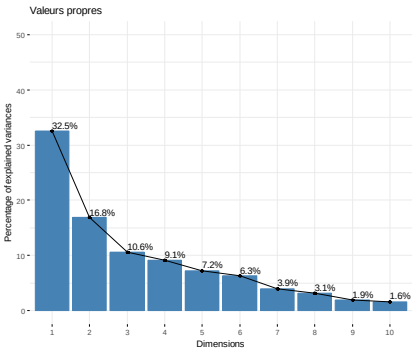


Figure 3- Inertie des valeurs propres pour les 10 première dimensions

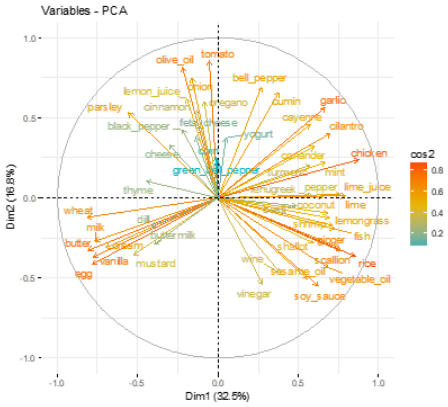


Figure 4- Cercle des corrélations

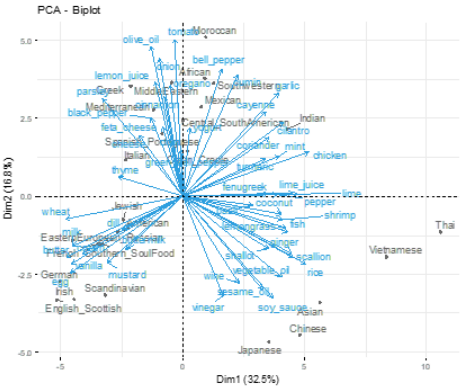


Figure 5-Cercles des corrélations et répartion des individus

I.3- Analyse ascendante hiérarchique (CAH) 1

L'analyse ascendante hiérarchique vise à partitionner une population en différentes partitions, c'est une méthode parmi de nombreuses techniques. Le principe de la CAH est de rassembler des individus selon un critère de vraisemblance. L'analyse est dite ascendante car on part de la partition où chaque classe est un singleton. On procède par la suite par fusion successive des classes qui se « ressemblent » jusqu'à obtenir une seule classe. La notion de ressemblance entre des observations est évaluée par une distance entre individus, nous avons donc déterminé dans un premier temps une matrice de distances entre les individus, en utilisant la distance de Manhattan : $L_1 : d(x,y) = \sum_{i=1}^{p} |x_i - y_i|$.

Une fois que nous avons déterminé la matrice de distances, il faut par la suite déterminer un critère d'agrégation pour la réalisation du dendrogramme. Dans le cas de notre étude, nous avons dès lors éliminé les méthodes centroïdes et single, de par un chevauchement des branches, rendant difficile la compréhension. La méthode de ward semble la mieux adaptée pour figurer la classification ascendante hiérarchique.

Le graphique présentant les sauts d'inertie nous indique quel serait le nombre de classes à retenir. Dans le cas de l'agrégation par la méthode de Ward, nous retenons 3 classes et non 8 classes.

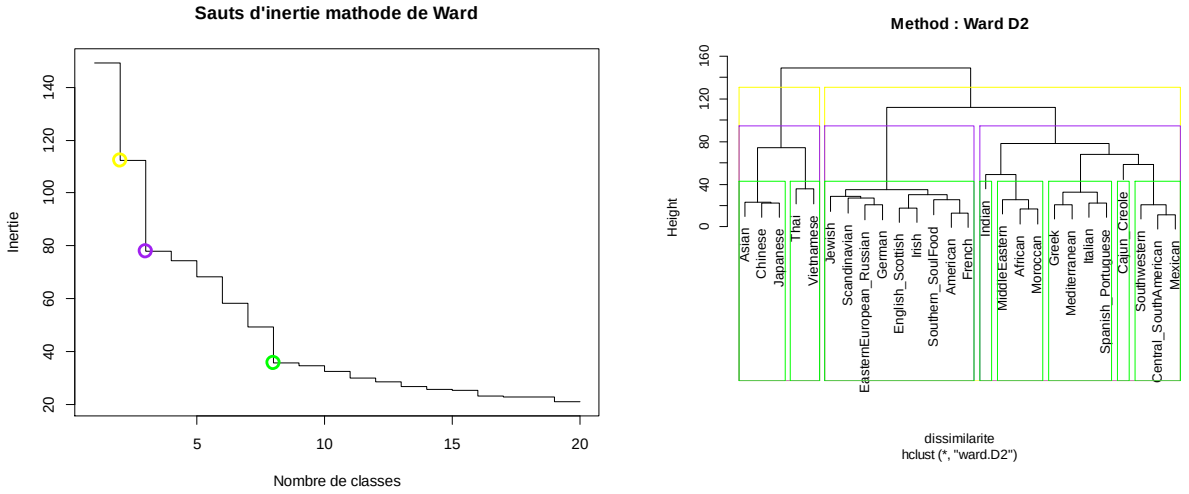


Figure 6- (à gauche) Inertie d'après le nombre de classes (à droite) Dendrogramme obtenu d'après les dissimilarités entre les individus.

I.4- Algorithme des K-means

L'algorithme des k-means est une méthode permettant de regrouper les individus selon un nombre de classe imposée par l'utilisateur. L'algorithme est programmé de manière à affecter chaque point au centre le plus proche. Dans un premier temps, nous avons testé pour k=3, nombre de classes qui nous parait le plus judicieux par comparaison à la CAH. Mais nous ne retrouvons pas les mêmes individus dans les mêmes groupes. En effet, par exemple Italian, se retrouve avec le pays Scandinavian, la répartition demeure donc différente par rapport à la CAH.

Pour la réalisation d'un clustering efficace, on cherche à minimiser l'inertie intra-classe (minimiser la distance entre les individus d'un même groupe) et maximiser l'inertie interclasse (maximiser la distance entre les individus n'appartenant pas au même groupe). C'est deux inerties peuvent donc être optimisées en faisant varier le nombre de classe k. En analysant la Figure 9, déterminant la proportion de l'inertie intra-classe, on observe une augmentation significative jusque k=8 mais ceci reste approximatif.

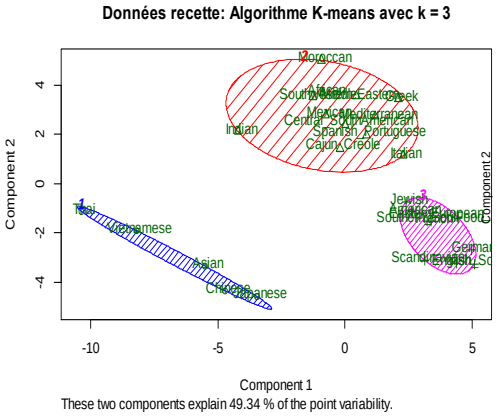


Figure 7- Algorithme des k-means pour k=3

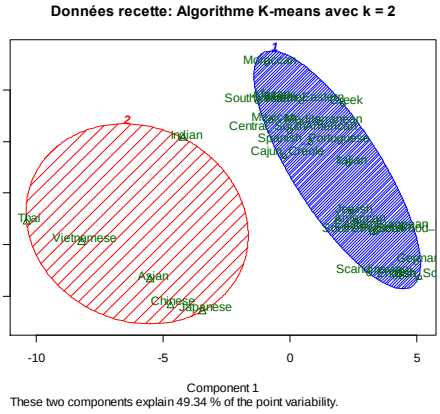


Figure 8- Algorithme des k-means pour k=2

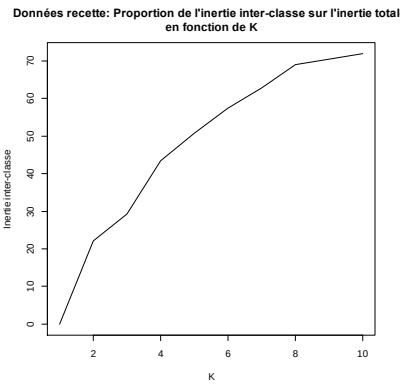


Figure 9- Proportion de l'inertie inter-classe pour différentes valeurs de k

I.5- Analyse factorielle sur un tableau de distance (AFTD)

Une analyse factorielle (AFTD) permet de complémenter l'analyse en composantes principales. Le tableau de données Recette a été transformé en matrice de dissimilarités entre les différentes zones géographiques. Elle calcule une représentation multidimensionnelle de ces individus (dont le tableau de dissimilarités ne donne qu'une description implicite) dans un espace euclidien. Par la méthode du coude, on observe la décroissance des valeurs propres et visuellement, nous pouvons dès lors affirmer que l'on peut se limiter aux deux premières valeurs propres pour la réalisation de l'AFTD dans R.

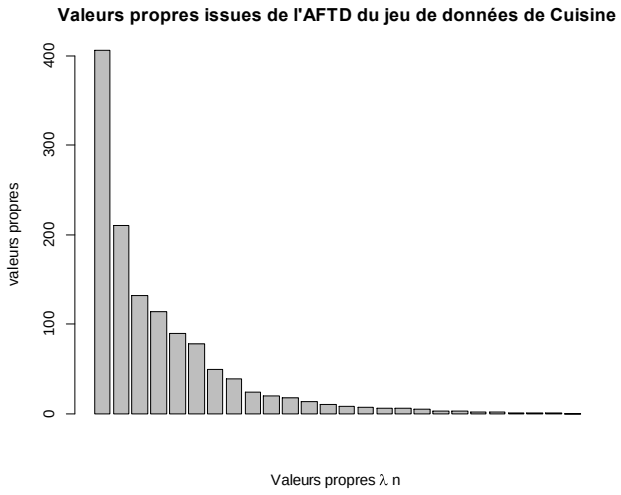


Figure 10- Valeurs propres de l'AFTD

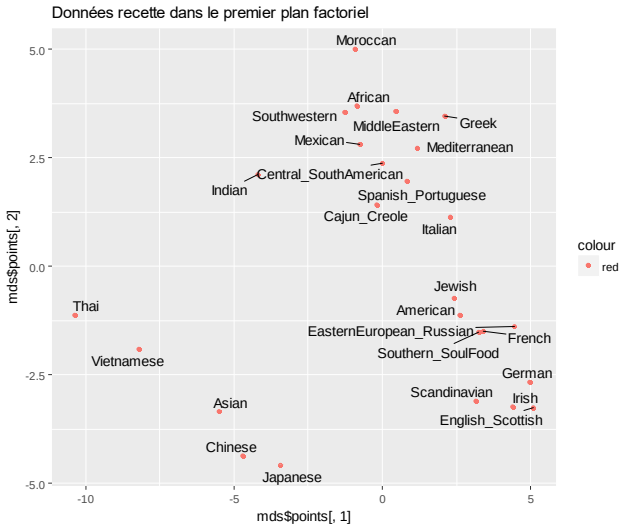


Figure 11- Répartition des origines

Le diagramme de Shepard réalisé montre une représentation des données correcte: avec R^2 de 0.87 demeure assez proches des données réelles. On retrouve bien ce que l'on avait observé dans l'analyse en composantes principales réalisée précédemment.

I.6- Analyse descriptive des données

Pour ce jeu de données sous format binaire (les observations sont soit de 0 ou de 1), il est répertorié pour différentes recettes venant d'un même pays l'utilisation ou non d'un ingrédient : s'il y a utilisation de l'ingrédient dans la recette, la valeur 1 est attribuée sinon, 0.

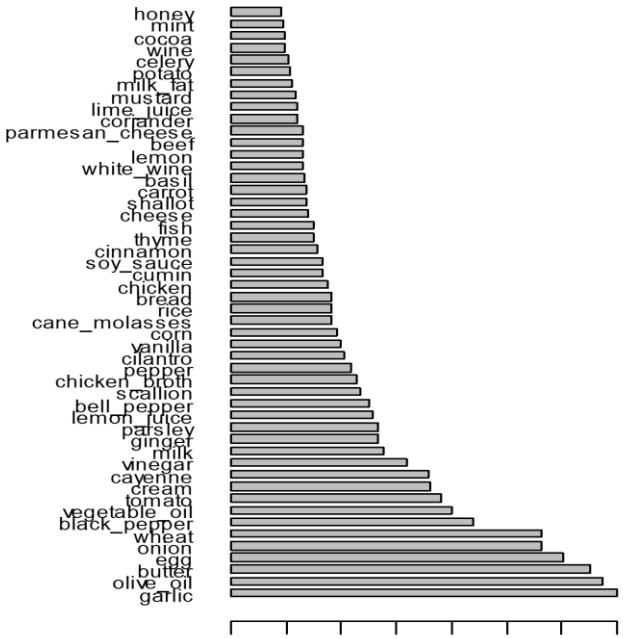


Figure 12- Barplot sur les données cuisine

Il est présenté dans ce jeu de données 2000 recettes et 50 variables.

Le graphique nous donne l'utilisation des ingrédients pour tous les pays. L'ingrédient sera d'autant plus utilisé que la barre est grande. On peut voir l'ingrédient ail, (traduit garlic) est beaucoup utilisé dans les différentes recettes.

I.7- Données transformées et calcul des dissimilarités

La transformation des données du tableau de données *recettes_echantillon*, s'est basée sur le nombre de 0 et de 1 pour chaque ingrédient et pour chaque origine géographique. L'objectif de cette transformation était d'obtenir le nombre d'utilisation d'un ingrédient pour toutes les recettes venant d'une seule origine. A partir de la matrice obtenue, les dissimilarités ont été déterminées à partir de la méthode de *Manhattan*.

Le tableau obtenu après la transformation des données est donc celui-ci :

	garlic	olive_oil	butter	egg	onion
French	40	42	80	72	26
Italian	111	166	65	75	63
EasternEuropean_Russian	5	2	12	9	8
Moroccan	6	11	4	2	5
Indian	16	3	7	6	14

I.8- Classification ascendante hiérarchique (CAH) 2

D'après le tableau de dissimilarités nouvellement obtenu, la classification ascendante hiérarchique présente dans ci-dessus, montre une répartition des ingrédients en 4 classes.

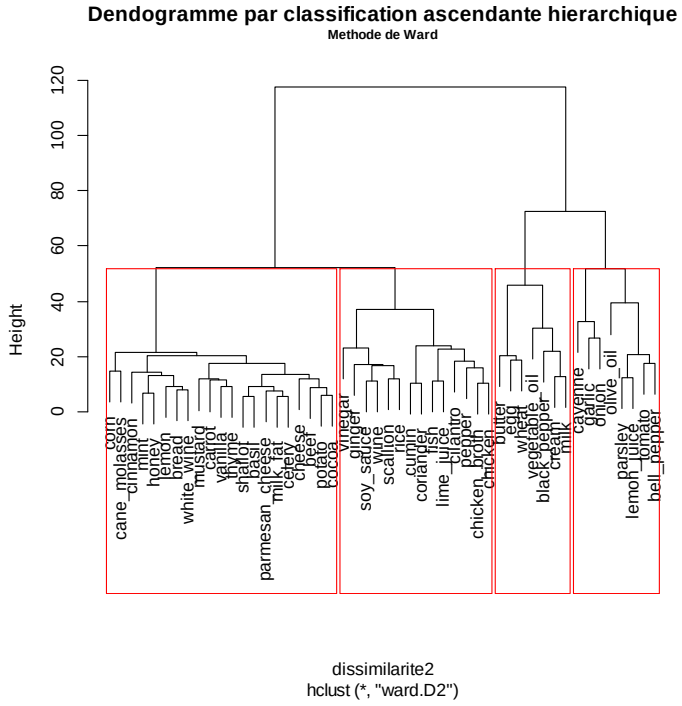


Figure 13- Dendrogramme classification ascendante hierarchique

I.9- Algorithme des k-médoïdes

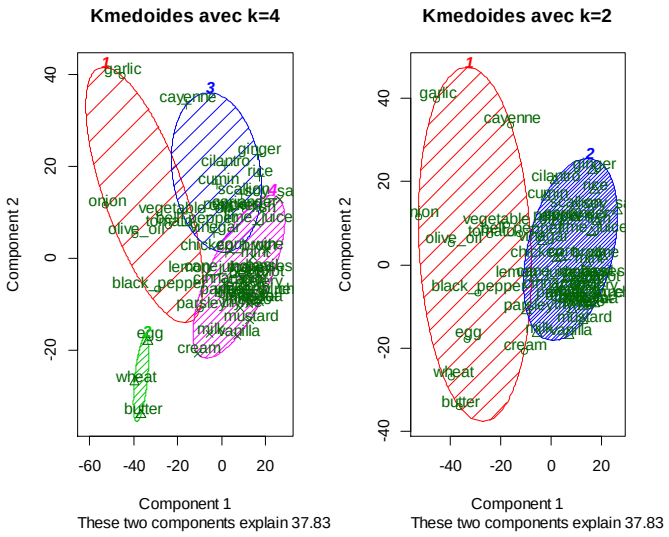


Figure 14- Execution de l'algorithme des k-médoïdes

Le terme médoïde se réfère à un objet au sein d'un cluster pour lequel la dissimilarité moyenne entre lui et tous les autres membres du cluster est minimale. Il correspond au point le plus central du cluster, alors que pour l'algorithme des k-means le centre d'un cluster donné était calculé comme la valeur moyenne de tous les points de données du cluster. L'algorithme des k-médoides est moins sensible au bruit et aux valeurs aberrantes, par rapport à l'algorithme des k-means. Lorsque l'on execute l'algorithme pour k=4, les ingrédients rprésentatifs sont: toamato-egg-chicken-celery. Dans le cas où l'on execute l'algorithme pour k=2, les ingrédients représentatifs sont black_pepper et celery. Ces ingrédients représentent les médoïdes de chacune des classes.

II- Classification par k-means avec distance adaptative

L'objectif de cette seconde partie du projet est de tester et de comparer l'algorithme des k- means avec l'algorithme réalisé par nos soins que l'on nomme l'algorithme des k-means adaptatif. Nous allons donc exécuter ces deux types d'algorithme sur 3 jeux de données dits « synthétiques » et dans un deuxième temps sur un jeu de données, que nous avons déjà exploité en td : le jeu de données Iris.

II.1- Comparaison et application de l'algorithme K-means classique et K-means adaptatif

Dans un premier temps, l'algorithme des k-means classique est exécuté sur chacun des jeux de données. Par la représentation graphique des clusters obtenus, nous avons comparé à la distribution originelle de chacun des groupes par la réalisation d'ACP.

De manière à observer la qualité du clustering obtenu par l'algorithme des k-means, les inerties inter-classe et intra-classe ont été observées, ainsi que l'indice de Rand. Pour ce dernier, celui-ci est maximale pour k=2 en raison de la répartition du jeu de données en deux groupes.

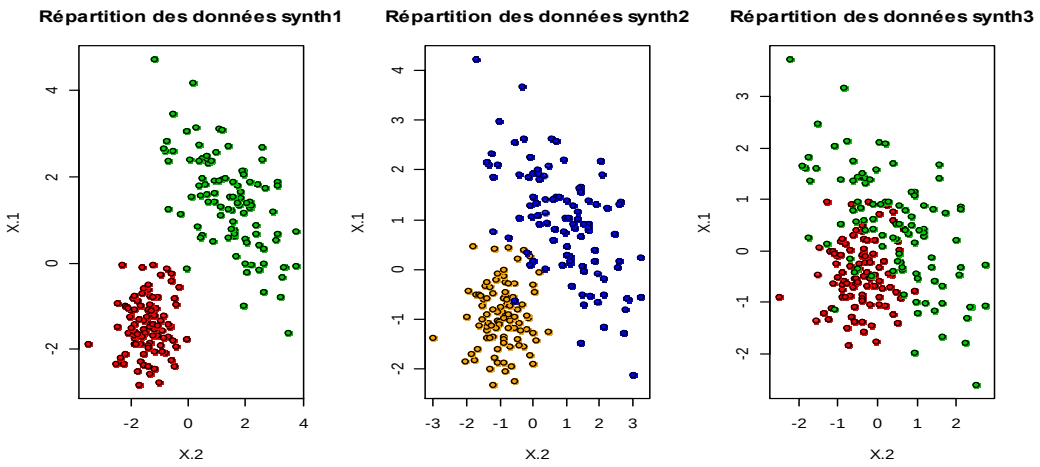


Figure 15- Répartition des données et répartions selon les groupes

Jeu de données synth1

Figure 16- ACP pour la connaissance de la partition des données selon la répartition réelle des données synth1

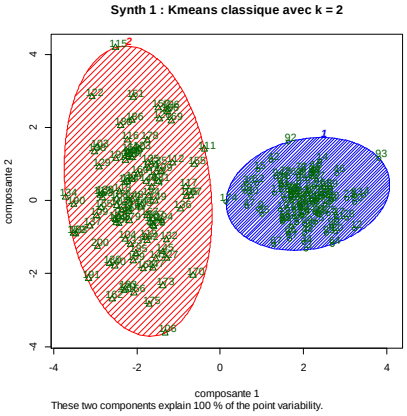


Figure 17- K-means classique synth1

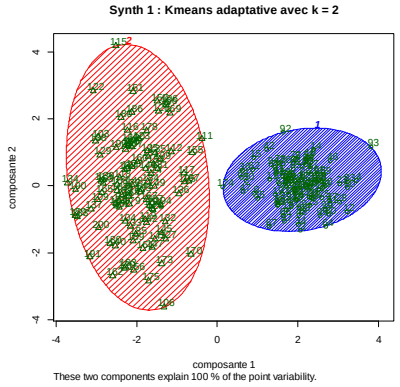


Figure 18- K-means adaptatif synth1

L'indice de Rand rend compte de la qualité de la répartition des données d'après la répartition réelle, on peut alors comparer ces deux partitions en comparant les deux indices de Rand, qui sont:

$$I_{k-means\ ada} = 0.9800 \qquad I_{k-means\ class} = 0.9800$$

Jeu de données synth2

Pour ce deuxième jeu de données, nous observons une intersection non vide entre les deux groupes qui constituent le jeu de données. Dans ce cas présent, il est judicieux d'observer la proportion de l'inertie intraclasse sur l'inertie totale pour différentes valeurs de k.

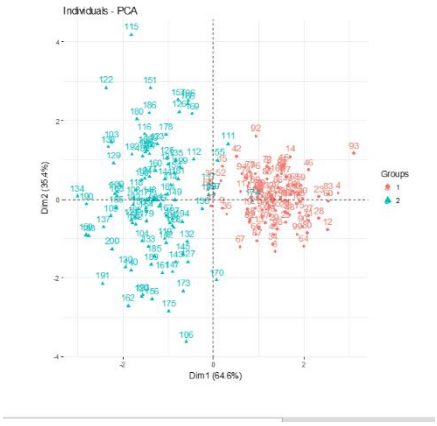


Figure 19- ACP pour la connaissance de la partition des données selon la répartition réelle des données synth2

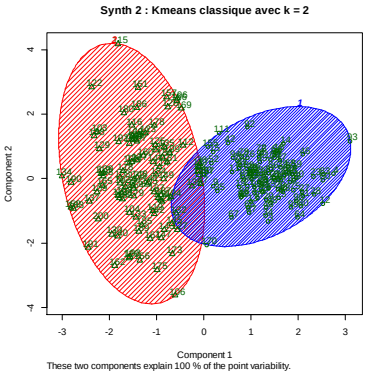


Figure 20- K-means classique synth2

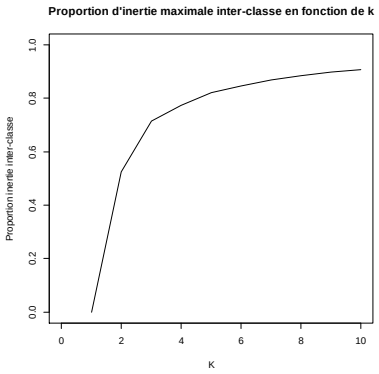


Figure 21- Proportion d'inertie inter-classe sur l'inertie totale synth2

Dans le cas présent, la proportion de l'inertie inter-classe augmente considérablement jusque k=3. D'autre part, pour déterminer le nombre de groupes maximisant l'inertie inter-classe, nous avons eu recourt à l'indice de *Calinski Harabasz*. Ainsi, nous pouvons dès lors affirmer que le nombre de groupes réduisant l'inertie intra-classe est de 3.

Pour le cas des k-maans on retrouve une partition des données en cluster plus représentatif des données originelles. Ceci est également révélé par l'indice de Rand pour K=2.

$$I_{k-means\ class} = 0.8456 \qquad I_{k-means\ ada} = 0.8642.$$

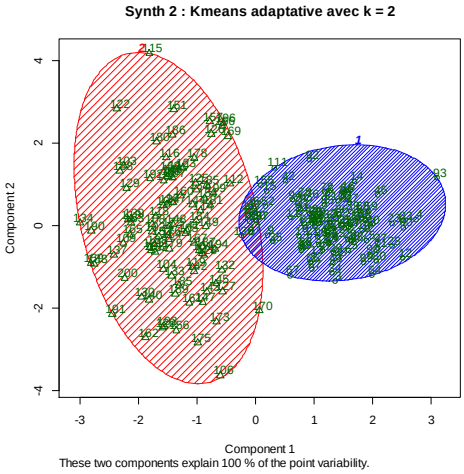


Figure 22- K-means adaptatif synth2

Jeu de données synth3

Pour ce troisième jeu de données, les individus ne sont pas distinctement séparés. La représentation des clusters opérée par l'algorithme des k-means classique, montre un clustring bien différent des donnés originelles. Ceci vient alors se repercuter sur la valeur de l'indice de Rand qui est de $I_{2-means\ class} = 0.1809$ et est de $I_{2-means\ ada} = 0.2991$

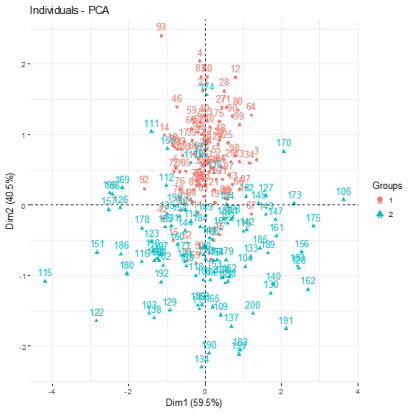


Figure 23- ACP pour la connaissance de la partition des données selon la répartition réelle des données synth3

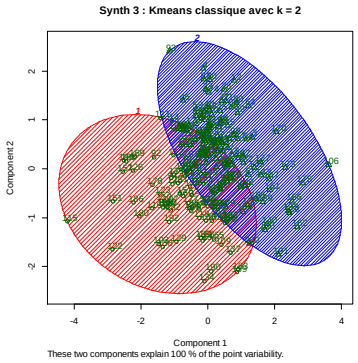


Figure 24- K-means classique synth3

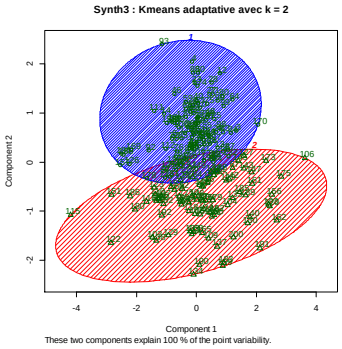


Figure 25- K-means adaptatif synth3

En conclusion, nous pouvons dire que l'algorithme des k-means adaptatif, fourni une partition des données plus vraisemblable que les données réelles. On peut alors affirmer que l'algorithme prend en compte la forme de la répartition des données.

Données Iris

Dans la continuité de notre analyse par application du k-means adaptatif, nous nous intéressons au jeu de données Iris. L'ACP présenté ci-dessous nous donne la répartition des individus selon les espèces d'après les deux premières composantes principales. Pour K=1, c'est à dire, sans partition des données Iris, la valeur de l'inertie vaut 681.3706 pour le k-means classique et la distance totale, déterminée par l'algorithme k-means adapatif, vaut 124.64.

Figure 26- Répartition des données iris selon les espèces

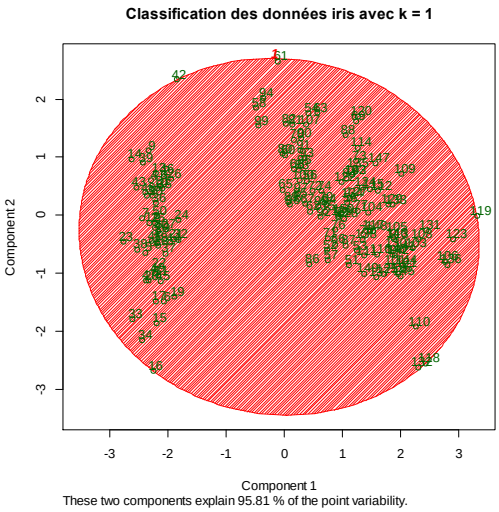
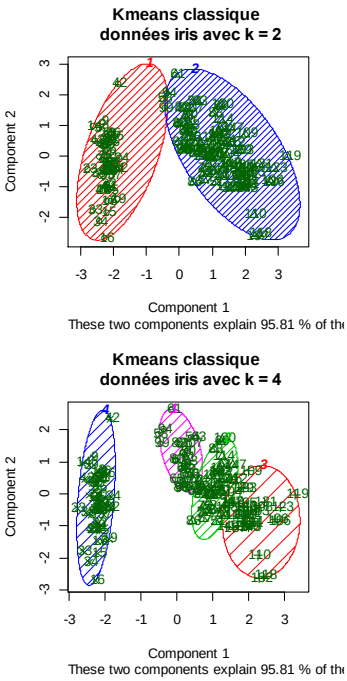


Figure 27- Classification des données pour K=1

L'execution des deux algortithmes a été réalisée pour différentes valeur de k. Nous comparons la partition des données obtenue pour ces différentes valeurs de k. Pour k=3, nous avons déterminé l'indice de Rand:

$$I_{k-means\ class} = 0.7302383 \quad et \quad I_{k-means\ ada} = 0.8514569$$



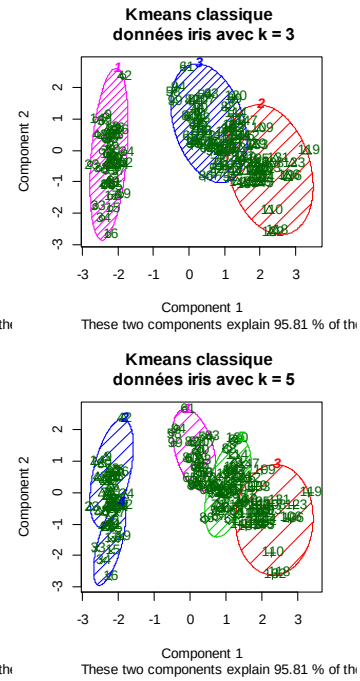
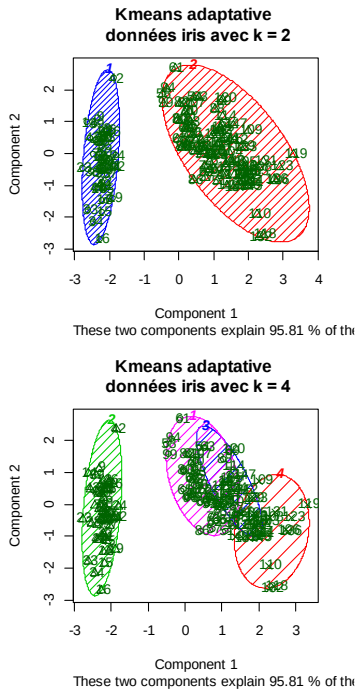


Figure 28- Répartition ds données Iris par K-means classique



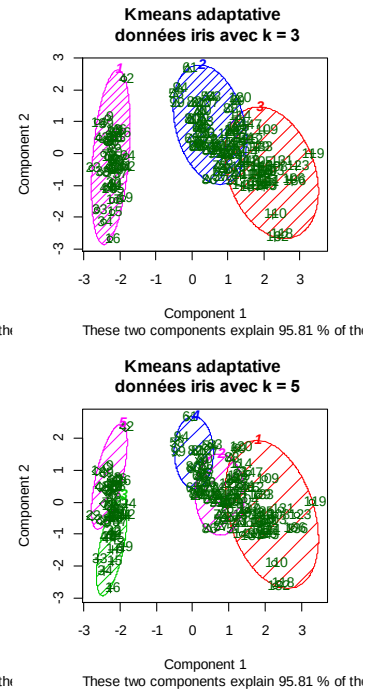


Figure 29- Répartition ds données Iris par K-means adaptatif

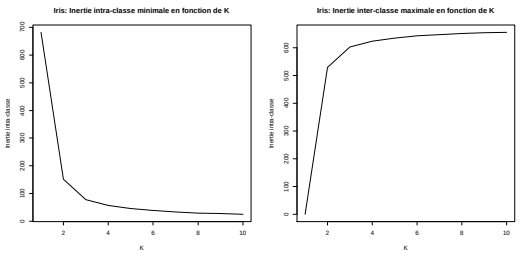


Figure 30- Inertie intra-classe et inter-classe des données Iris

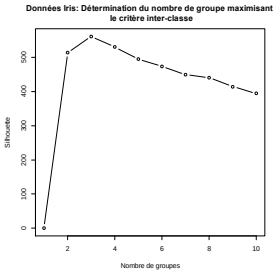


Figure 31- Recherche de l'indice de Calinski Harabasz pour maximiser l'inertie inter-classe

En observant la valeur du critère, nous pouvons affirmer que pour l'execution du k-means, le nombre de classes le plus probables correspond à 3. Pour ce qui est de l'algorithme adaptatif, nous retrouvons une certaine instabilité mais nous puvons d'ores et déjà conclure que d'après la valeur du critère, diminuant en fonction de K, il semblrait que le meilleur nombre de classes serait de 2 pour minimiser la valeur du critère, car on retrouve une pente importante.

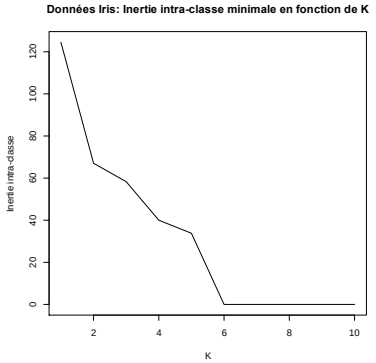


Figure 32- Valeur du critère de K-means adaptatif pour différentes valeurs de K

II.2- Données Spam

Comme la dimension de jeux de données est très grande et il existe beaucoup de 0, nous pensons d'abord à faire un pré-traitement avant d'appliquer l'algorithme. Afin de diminuer la dimension de données, nous réalisons l'analyse de composantes principales sur les données.

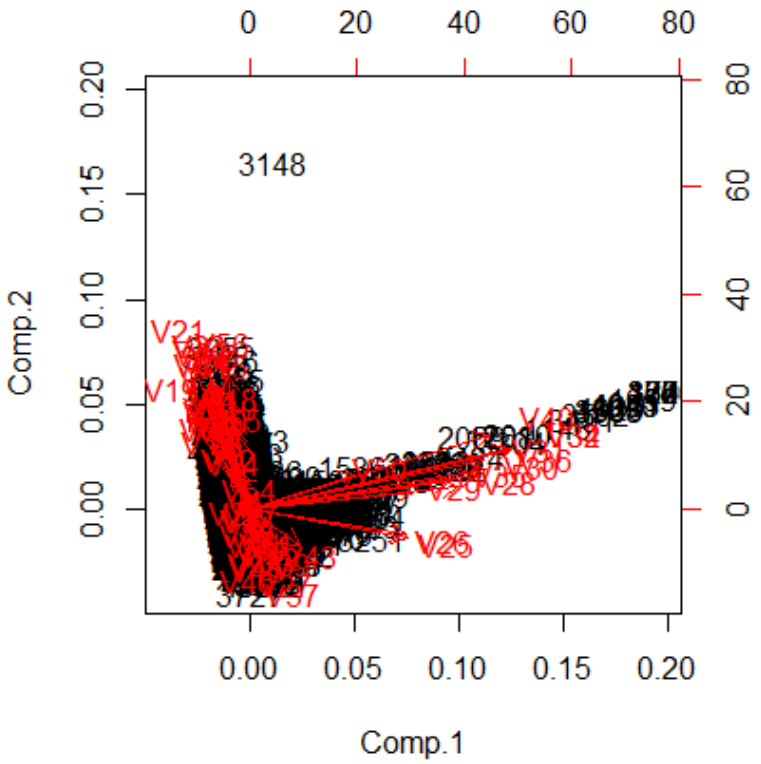


Figure 33- ACP données Spam

Selon notre observation, des données sont très concentrées au point (0,0) et les variables sont corrélées. Pour simplifier ce cas, nous remplaçons toutes les valeurs supérieures à 0 en 1 afin d'améliorer la performance d'agrégation et diminuer la complexité des données. Ensuite, nous utilisons la fonction ***predict*** pour obtenir les ordonnées des individus dans les nouvelles composantes principales.

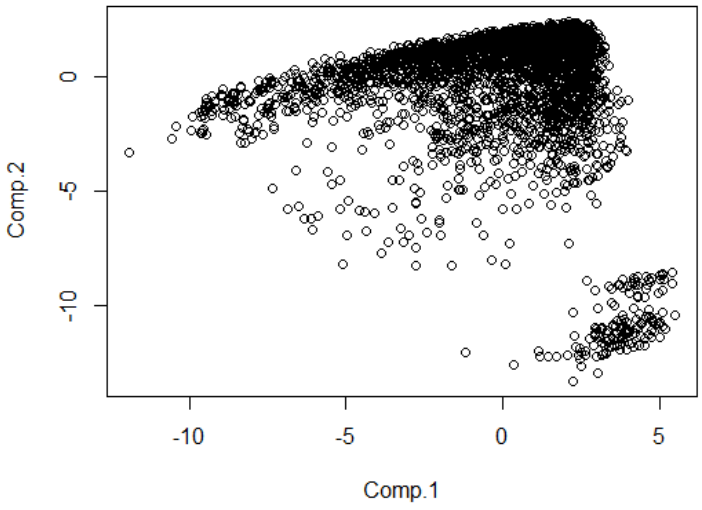


Figure 34- ACP après transformation des données Spam

Selon le graphique ci-dessus, nous observons que la présentation des données est beaucoup meilleure et des données se groupent à 2 groupes clairement. Puis nous réalisons la classification par l'algorithme k-means classique et par l'algorithme par kmeans adaptatif pour k = 2. Nous avons déterminé l'indice de Rand pour k = 2 :

$I_{2-means\ class} = 0.3439405 \quad et \quad I_{2-means\ ada} = 0.5808958$

En conclusion, nous pouvons dire que l'algorithme des k-means adaptatif, fourni une partition des données plus vraisemblable que l'algorithme des k-means classique. Cependant, l'algorithme des k-means adaptatif perd plus de temps pour finir la classification donc nous pensons à une autre méthode. Nous avons choisi une méthode supervisée parce que nous pourrons apprendre un modèle qui permettra de déterminer la classe à partir des observations des variables étiquetées explicatives pour de nouveaux individus. Nous avons décidé d'utiliser la méthode ***K plus proches voisins*** (PPV) suite à ses qualités :

- La complexité de *training* est égale à 0
- Le prévu est simple (après avoir déterminé les voisins)
- PPV peut être améliorée par le calcul de distance Mahalanobis au lieu de la distance Euclidienne

Nous avons fait l'algorithme de K plus proches voisins en codant une fonction pour le *training* et *testing* des données, une autre fonction pour le calcul des erreurs. Nous avons réalisé la classification par PPV avec k = 2, 3, 4 et déterminé des erreurs :

Par des différences entre la partition trouvée et la partition réelle

K	Erreur d'apprentissage	Erreur de test
2	0.09735898	0.2142112
3	0.10231497	0.2053455
4	0.13022498	0.2099087

Par l'indice de Rand

K	Erreur d'apprentissage	Erreur de test
2	0.6579222	0.2925423
3	0.6264480	0.3496981
4	0.5470356	0.3218716

Nous pouvons conclure que la classification par la méthode K plus proches voisins avec la distance Mahalanobis fournit une meilleure partition de données.

II.3- Justification

Relation entre la mise à jour des prototypes et le calcul des centres de gravité des classes

Comme $\det M_k = \rho_k$, nous avons :

$$\frac{\partial L(\{v_k, M_k, \lambda_k\})}{\partial v_k} = \frac{\partial J(\{v_k, M_k\})_{k=1,..,k}}{\partial v_k} = \frac{\partial(\sum_{k=1}^{K}\sum_{i=}^{n} z_{ik}.\, d_{ik}^2)}{\partial v_k}$$
$$= \frac{\partial\left(\sum_{k=1}^{K}\sum_{i=1}^{n} z_{ik}.\ \ x_i - v_k\ ^T M_k\ x_i - v_k\ \right)}{\partial v_k}$$

D'après la propriété de matrice Single-entry, nous obtenons :

$$\frac{\partial L(\{v_k, M_k, \lambda_k\})}{\partial v_k}$$
$$= \sum_{k=1}^{K}\sum_{i=1}^{n} -\ M_k + M_k^T\quad x_i - v_k\ \frac{\partial\ x_i - v_k}{\partial v_k}$$
$$= -\sum_{k=1}^{K}\ M_k + M_k^T\ .\sum_{i=1}^{n} z_{ik}\ x_i - v_k\ = -\sum_{k=1}^{K}\ M_k + M_k^T\ .\left(\sum_{i=1}^{n} z_{ik}\, x_i - \ \overline{n_k} v_{ik}\right)$$

Pour minimiser le critère, il faut que $\frac{\partial L(\{v_k, M_k, \lambda_k\})}{\partial v_k} = 0$. Après la mise à jour des prototypes, on a $v_{ik} = \frac{1}{n_k}\sum_{i=1}^{n} z_{ik}\, x_i$ ou bien $\sum_{i=1}^{n} z_{ik}\, x_i - \ \overline{n_k} v_{ik} = 0$ et cela revient $\frac{\partial L(\{v_k, M_k, \lambda_k\})}{\partial v_k} = 0$. La mise à jour des prototypes donc revient à calculer les centres des classes définis par l'équation $v_{ik} = \frac{1}{n_k}\sum_{i=1}^{n} z_{ik}\, x_i$.