1 Objectif

Analyse de l'algorithme de classification automatique « TwoStep Cluster » de SPSS. Détection automatique du nombre de classes. Comparaison avec d'autres méthodes mixtes implémentées dans SPAD, Tanagra et R.

Au gré de mes pérégrinations sur le web, mon attention a été attirée par l'algorithme de classification automatique (clustering, apprentissage non supervisé) **TwoStep Cluster** décrit sur le site de SPSS. Il présente des caractéristiques très intéressantes de prime abord : il sait traiter les variables actives quantitatives et qualitatives, il peut appréhender des très grandes bases de données avec des temps de traitements ébouriffants, il sait détecter automatiquement le nombre adéquat de classes. En fait, la méthode répond à bien des préoccupations pratiques des data scientist dans ce type de démarche. Pourtant, très étrangement, elle est peu connue, et il n'existe pas de package pour R ou pour Python qui ait essayé de l'implémenter.

J'ai donc regardé de plus près. Je me suis rendu compte qu'il s'agissait d'une approche mixte (ou approche combinée) (Lebart et al., 1995 ; section 2.3) popularisée par le logiciel SPAD. On comprend la notion de « two step ». Elle consiste à créer rapidement un pré-regroupement des observations en un nombre élevé de classes (de l'ordre de quelques dizaines voire des centaines) à l'aide d'une méthode de partitionnement rapide (ex. K-Means avec peu d'itérations), puis à réaliser une agrégation hiérarchique à partir de ces pre-cluster pour bénéficier de la lisibilité des résultats associée au dendrogramme. Cet aspect de TwoStep Cluster n'est pas vraiment original. De même, la capacité à traiter des variables mixtes n'est pas révolutionnaire. Il suffit d'utiliser une mesure de distance capable d'arbitrer correctement entre les influences des variables quantitatives et qualitatives comme celle basée sur la similarité de Grower (1971) par exemple. Finalement, et c'est ce qui a réellement éveillé ma curiosité, c'est la stratégie mise en place pour identifier le bon nombre de classes dans une structure hiérarchique qui se révèle inédite. On dépasse les astuces usuelles basées sur l'analyse de la courbe l'inertie intra-classes ou encore la recherche du « saut » entre les paliers du dendrogramme. Elle s'appuie sur une mesure de la vraisemblance, peu utilisée dans le contexte du regroupement hiérarchique, qui permet de développer des critères de type BIC ou AIC pour identifier le nombre de classes. Et, surprise,

plutôt que de s'en tenir à leur simple optimisation, une mécanique assez complexe est rajoutée pour peaufiner la détection.

Ce tutoriel s'attachera surtout dans un premier temps à décrypter ce processus d'identification du nombre « optimal » de classes dans la structure hiérarchique. Pour une analyse plus globale et approfondie de la méthode « TwoStep Cluster » et de son comportement, je conseille l'article de Bacher et al. (2004) qui était arrivé à la conclusion (pour la version qu'ils ont étudiée tout du moins) que l'algorithme était assez efficace tant que l'on ne manipulait que des variables actives quantitatives.

Dans un second temps, nous comparerons sur une base de taille relativement importante l'efficacité (identification du bon nombre de classes) et les temps de calcul de l'algorithme par rapport à des implémentations classiques de l'approche mixte combinant méthode des centres mobiles et classification ascendante hiérarchique, sous SPAD, Tanagra et R.

2 L'algorithme TwoStep Cluster de SPSS

2.1 Etapes de l'algorithme

La description de l'algorithme est accessible sur le site de SPSS. Il comporte 2 principales étapes. Bacher et al. (2004) en propose un décryptage assez pointu.

Pre-cluster. Dans un premier temps, un pré-regroupement des observations est opéré à l'aide de la méthode de classification BIRCH (Zhang et al., 1996). La technique est connue et ne se révèle pas décisive dans ce contexte. Nous nous bornerons à remarquer qu'elle est particulièrement rapide car ne nécessite pas d'accès répétés aux données, qu'elle est capable d'isoler les points relevant « du bruit » c.-à-d. des points atypiques se détachant des autres mais ne constituant pas une classe à part entière. Par défaut sous SPSS, l'arbre CF-TREE de BIRCH est configurée avec un nombre maximal de branches par feuille égale à MXBRANCH = 8, et une profondeur maximale MXLEVEL = 3, ce qui nous mène à un nombre maximal de pre-clusters égal à $\text{MXBRANCH}^{\text{MXLEVEL}} = 8^3 = 512$ (le nombre maximal de nœuds est lui égal à $585 = 1 + \sum_{b=1}^{MXLEVEL} MXBRANCH^b$).

Agrégation hiérarchique. Dans un second temps, les classes sont fusionnées au fur et à mesure jusqu'à obtenir un seul groupe. Une distance euclidienne peut être utilisée si les variables sont toutes quantitatives. Une mesure basée sur la log-vraisemblance est mise en œuvre dans le cas des variables mixtes. Elle reste valable lorsque les variables sont

exclusivement quantitatives ou qualitatives. Elle constitue une des originalités de l'algorithme. Elle est basée sur la notion de dispersion, quoi de plus normal dans le contexte d'une classification automatique, et s'appuie : sur la variance intra-classe pour les variables quantitatives, sur l'entropie pour les variables qualitatives.

Détection du nombre de classes. Une fois la hiérarchie élaborée, chaque découpage peut être évalué à l'aide du critère BIC [Bayesian Information Criterion] (ou AIC [Akaike]), et un procédé ad hoc que nous détaillerons plus loin permet d'identifier le nombre adéquat de classes. Pour moi, la véritable innovation réside ici.

2.2 Distance entre classes basée sur la log-vraisemblance

Distance entre classes. La mesure de distance est primordiale dans un processus de classification automatique. La distance euclidienne est bien connue des data scientists. La distance log-vraisemblance est autrement plus originale. La distance entre deux classes C_a et C_b est définie comme suit :

$$d(a,b) = \xi_a + \xi_b - \xi_{<a,b>}$$

Dispersion des classes. ξ_a et ξ_b correspondent à la dispersion des classes Ca et Cb ; $\xi_{<a,b>}$ à la dispersion de la classe issue de la fusion de C_a et C_b ; d(a,b) – mesure de proximité entre deux classes - peut se lire comme la perte d'inertie inter-classes consécutive à leur agrégation. Lors de la construction de la hiérarchie, l'algorithme réunit itérativement les paires de classes qui minimisent d(a,b) jusqu'à obtenir un groupe unique. La démarche est conforme avec les principes usuels de la classification ascendante hiérarchique (CAH).

La dispersion pour la classe C_k est définie comme suit :

$$\xi_k = -n_k \left(\sum_{j=1}^{p} \frac{1}{2} \ln\left(\hat{\sigma}_{kj}^2 + \hat{\sigma}_j^2\right) - \sum_{j=1}^{q} \sum_{l=1}^{m_j} \hat{\pi}_{kjl} \ln \hat{\pi}_{kjl} \right)$$

Où

- p est le nombre de variable quantitatives ;
- n_k correspond à l'effectif de la classe C_k ;
- $\hat{\sigma}_j^2$ est la variance de la variable X_j dans l'ensemble de l'échantillon ;
- $\hat{\sigma}_{kj}^2$ est la variance de la variable X_j dans la classe C_k ;

L'introduction de la grandeur $\hat{\sigma}_j^2$ permet de résoudre l'écueil du $ln(0)$ lorsque la classe est constituée d'un singleton ou lorsque la valeur de la variable est constante dans la classe. *Remarque : si la variable est une constante d'emblée c.-à-d. $\hat{\sigma}_j^2 = 0$ il ne faut pas l'introduire dans l'étude de toute manière*.

Et pour les variables qualitatives :

- q est le nombre de variables qualitatives ;
- m_j est le nombre de modalités de la variable X_j ;
- $\hat{\pi}_{kjl}$ est la fréquence relative de la modalité l de la variable Xj au sein de la classe C_k.

Lecture en termes de log-vraisemblance. Dixit la documentation de SPSS, si l'on omet la quantité $\hat{\sigma}_j^2$qui est introduite pour la raison invoquée ci-dessus, l'expression ξ_k de la dispersion des classes correspond à une log-vraisemblance. Et la distance *d(a,b)* s'interprète comme une diminution de la log-vraisemblance lorsque les classes C_a et C_b sont fusionnées.

2.3 Evaluation d'une partition – Critères BIC et AIC

La log-vraisemblance d'une partition en K classes peut être exprimée comme suit :

$$LL(K) = \sum_{k=1}^{K} \xi_k$$

Il est dès lors possible de dégager les critères BIC :

$$BIC(K) = -2LL(K) + \ln(n) \times r_K$$

Et AIC :

$$AIC(K) = -2LL(K) + 2 \times r_K$$

Où :

- n est la taille de l'échantillon ;
- $r_K$est le degré de liberté associé à la partition, avec $r_K = K\left[2p + \sum_{j=}^{q}(m_j - 1)\right]$

Remarque : Notons que l'expression $-2LL(k)$ [2 fois la log-vraisemblance] correspond à la déviance, bien connue des statisticiens.

2.4 Stratégie de détection des classes

Très étrangement, SPSS n'exploite pas directement le critère BIC ou AIC pour détecter le bon nombre de classes. Il aurait été intuitif de choisir comme meilleure partition la valeur K qui minimise ce critère comme on le ferait dans d'autres contextes telle que la sélection de variables pour la régression. Cette stratégie ne donne pas satisfaction apparemment (SPSS, 2001), SPSS préfère une procédure à plusieurs étages. Nous l'explicitons à partir d'un exemple.

2.4.1 Un exemple didactique

Nous avons généré un ensemble de données de n = 300 observations dans le plan c.-à-d. décrites par p = 2 variables quantitatives (X_1, X_2). Il n'y a donc pas de variables qualitatives dans notre exemple. Représentés dans le plan, nous distinguons relativement aisément la présence de 3 groupes de points, un assez distinct, dans la partie sud-est, deux autres relativement proches au nord (Figure 1).

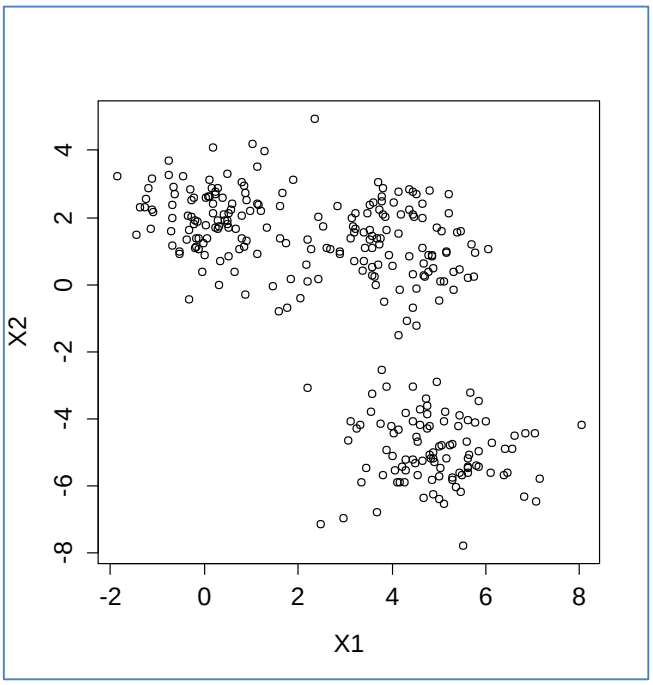


Figure 1 - Exemple didactique - Disposition des points dans le plan (*n = 300 ; p = 2*)

2.4.2 Le ratio de modification BIC

Après avoir lancé les traitements (nous détaillerons la mise en œuvre sous **IBM SPSS Statistics 24** [Trial version] dans la section 3), nous obtenons un tableau retraçant les

valeurs du critère BIC pour les partitions en K classes allant de 1 à K_{max} = 15 (cette valeur est un paramètre de l'algorithme, elle peut être modifiée dans le logiciel) (Tableau 1).

Nombre de clusters	Critère bayésien de Schwartz (BIC)	Modification BIC[a]	Rapport des modifications BIC[b]	Rapport des mesures de distance[c]
1	1634.531			
2	1418.372	-216.158	1.000	2.501
3	1345.632	-72.740	0.337	7.155
4	1355.092	9.460	-0.044	1.282
5	1367.490	12.399	-0.057	1.280
6	1382.170	14.679	-0.068	1.446
7	1399.359	17.189	-0.080	1.089
8	1417.007	17.647	-0.082	1.172
9	1435.411	18.404	-0.085	1.294
10	1454.818	19.407	-0.090	1.219
11	1474.839	20.021	-0.093	1.132
12	1495.184	20.345	-0.094	1.298
13	1516.096	20.912	-0.097	1.020
14	1537.045	20.949	-0.097	1.303
15	1558.428	21.383	-0.099	1.358

Tableau 1 - Table pour identification des clusters

Remarque : Nous nous focalisons sur le BIC dans ce qui suit mais la démarche est transposable au critère AIC.

K = 3 classes minimise le critère BIC [BIC(3) = 1345.632] (Tableau 1, fond orange).

La modification BIC (a). Mais ce résultat n'emporte pas la décision. SPSS procède à des calculs supplémentaires. La modification BIC est égale à l'écart entre deux valeurs successibles du BIC, c.-à-d. lors du passage à une partition plus grossière avec un groupe en moins[1].

$$dBIC(K) = BIC(K+1) - BIC(K)$$

Dans notre exemple (Tableau 1), dBIC(1) = BIC(2)-BIC(1) = 1418.372 – 1634.531 = -216.158 ; dBIC(3) = BIC(3) – BIC(2) = 1345.632-1418.372 = -72.740 ; etc. *Remarque : Attention, SPSS place la valeur dBIC(1) = -216.158 au niveau de la 2ème ligne du tableau*.

[1] Ce que dit la documentation en ligne ne correspond pas toujours avec ce que sort le logiciel, j'ai en partie déduit les formules à partir des résultats fournis par SPSS.

Si dBIC(1) > 0, on décide que le nombre adéquat de classes est K* = 1. Et on arrête les frais.

Dans notre cas, dBIC(1) = -216.158. Nous passons à l'étape suivante.

Le rapport des modifications BIC (b). Nous calculons le rapport des modifications BIC :

$$R_1(K) = \frac{dBIC(K)}{dBIC(1)}$$

On identifie alors la valeur K_0 qui détermine le nombre de clusters à partir duquel nous cherchons la solution dans la phase suivante.

K_0 correspond à la plus petite valeur de K pour laquelle

$$R_1(K) < 0.04$$

Remarque : Le seuil 0.04 paraît arbitraire. Il est vraisemblablement issu d'expérimentations à grande échelle sur des configurations diverses et variées.

Dans notre exemple, **$K_0 = 3$** puisque $R_1(3)$ = -0.044 (Tableau 1, en fond bleu en 4ème ligne, n'oublions pas qu'il y a un décalage dans les indices pour le calcul dBIC(K) – c'est vraiment très étrange que SPSS présente son tableau de calcul avec cette organisation).

Remarque : Si aucune des valeurs $R_1(K)$ n'est en dessous du seuil, nous partons à partir du bas du tableau (K_{max}).

Rapport des mesures de distance (c). Pour calculer le rapport de distances entre clusters, nous devons tout d'abord les distances d(K) entre deux solutions successives en (K-1) et (K) classes, en partant de K_0. Fort heureusement, il n'est pas nécessaire de revenir sur les données pour obtenir ces distances, nous pouvons les déduire du BIC.

La log-vraisemblance associée à une partition en K classes est liée au BIC de la manière suivante (voir formule du BIC plus haut) :

$$LL(K) = \frac{1}{2}\left[\ln(n) \times r_K - BIC(K)\right]$$

Nous avons successivement LL(1) = 0.5 x [ln(300) x (1 x 2 x 2) – 1634.531] = -805.858 ; LL(2) = 0.5 x [ln(300) x (2 x 2 x 2) – 1418.372] = -686.371 ; etc.

Nous pouvons déduire maintenant la distance d(K) :

$$d(K) = LL(K-1) - LL(K)$$

Ainsi, pour d(2) = LL(1) – LL(2) = -805.858 – (-686.371) = -119.487 ; d(3) = -686.371 – (-638.593) = -47.778 ; etc. On remarquera que d(1) n'est pas défini dans ce contexte.

Le rapport de distance est alors le rapport entre deux distances successives, soit :

$$R_2(K) = \frac{d(K)}{d(K+1)}$$

Dans notre exemple, $R_2(2)$ =d(2) / d(3) = -119.487 / (-47.778) = 2.501 ; $R_2(3)$ = d(3) / d(4) = -47.778 / (-6.678) = 7.155. Et on devrait s'en tenir à ces calculs puisque nous avons vu précédemment que $K_0 = 3$. SPSS affiche pourtant le reste de la colonne.

Ici non plus, $R_2(1)$ n'est pas défini.

Nous avons reproduit les calculs à l'aide d'un tableur (Tableau 2) en partant des seules valeurs du BIC (en fond vert). Nos résultats concordent en tous points avec ceux de SPSS.

Nombre de clusters	Critère bayésien de Schwartz (BIC)	Modification BIC[a]	Rapport des modifications BIC[b]	LL(K)	d(K)	Rapport des mesures de distance[c]
1	1634.531			-805.858		
2	1418.372	-216.158	1	-686.371	-119.487	2.501
3	1345.632	-72.740	0.337	-638.593	-47.778	7.155
4	1355.092	9.460	-0.044	-631.916	-6.678	1.282
5	1367.490	12.399	-0.057	-626.707	-5.208	1.280
6	1382.170	14.679	-0.068	-622.64	-4.068	1.446
7	1399.359	17.189	-0.080	-619.827	-2.813	1.089
8	1417.007	17.647	-0.082	-617.243	-2.584	1.172
9	1435.411	18.404	-0.085	-615.037	-2.205	1.294
10	1454.818	19.407	-0.090	-613.333	-1.704	1.219
11	1474.839	20.021	-0.093	-611.936	-1.397	1.132
12	1495.184	20.345	-0.094	-610.701	-1.235	1.298
13	1516.096	20.912	-0.097	-609.75	-0.952	1.020
14	1537.045	20.949	-0.097	-608.816	-0.933	1.303
15	1558.428	21.383	-0.099	-608.1	-0.716	

Tableau 2 - Reproduction des calculs de SPSS à l'aide d'un tableur

Pour détecter le bon nombre de classes, SPSS identifie les deux configurations K_1 et K_2 correspondant aux deux plus grandes valeurs de $R_2(K)$. Dans notre cas, il s'agit de $K_1 = 3$ avec $R_2(K_1) = 7.155$ et $K_2 = 2$ avec $R_2(K_2) = 2.501$. La règle de gestion suivante est alors appliquée :

- Si le ratio $\frac{R_2(K1)}{R_2(K2)}$ est supérieur à 1.15, alors nous fixons K* = K_1.
- Dans le cas contraire, nous prenons K* = max (K_1, K_2).

Dans notre exemple, puisque $\frac{R_2(K1)}{R_2(K2)} = \frac{7.155}{2.501} = 2.86 > 1.15$, SPSS choisit **K* = 3**. Et c'est heureux parce qu'il correspond au bon nombre de classes.

3 Traitement des grandes bases

Dans cette section est détaillée la mise en œuvre pas à pas de la méthode TwoStep Cluster sous SPSS 24. De nouveau, nous travaillons sur des données générées artificiellement, dans un espace à p = 4 dimensions cette fois-ci (X1, X2, X3, X4). Nous serons confrontés à deux difficultés : (1) nous travaillons sur une « grande » base de n = 1.000.000 d'observations ; (2) seules les deux premières variables (X1, X2) sont pertinentes pour la définition de classes. La bonne solution correspond à K* = 4. Nous représentons un échantillon de 500 observations dans l'espace défini par chaque paire de variables (Figure 2).

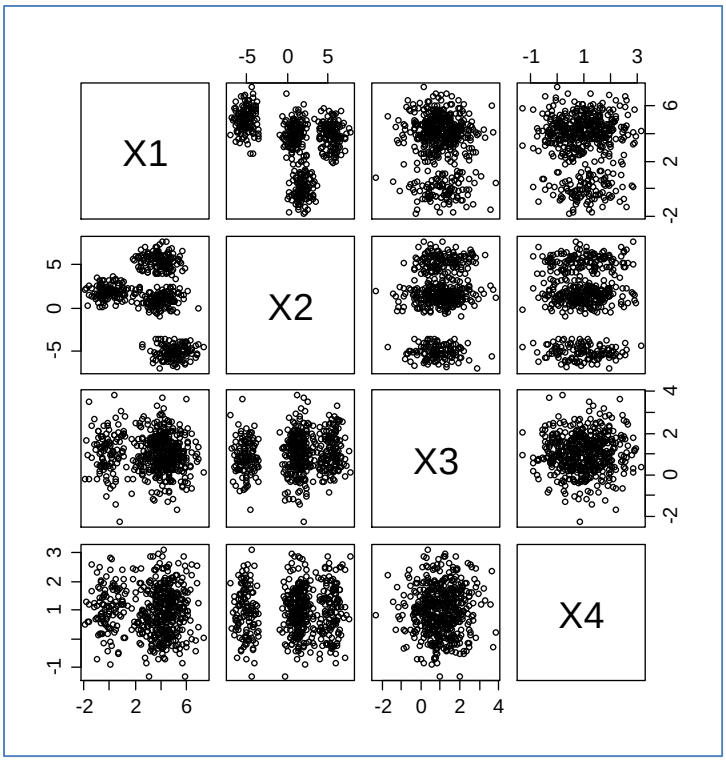
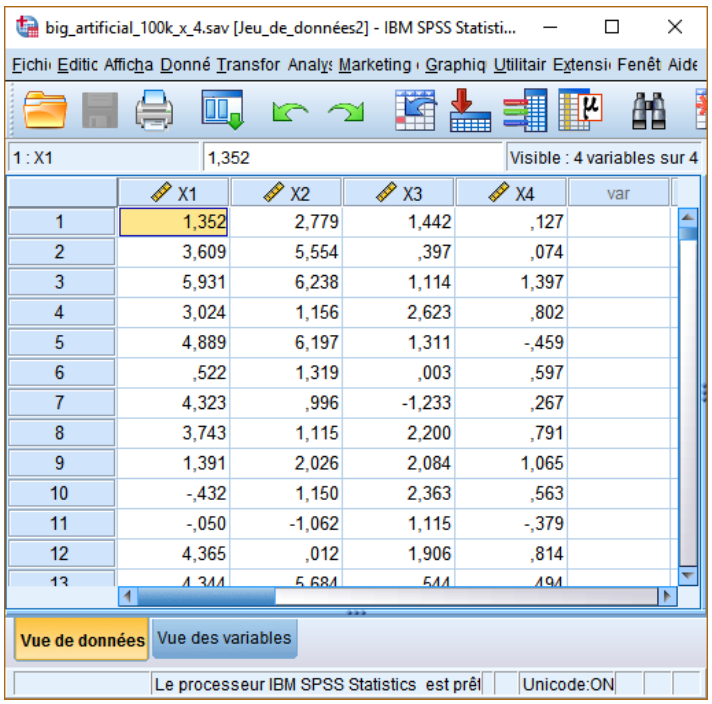


Figure 2 - Disposition des points (p = 4)

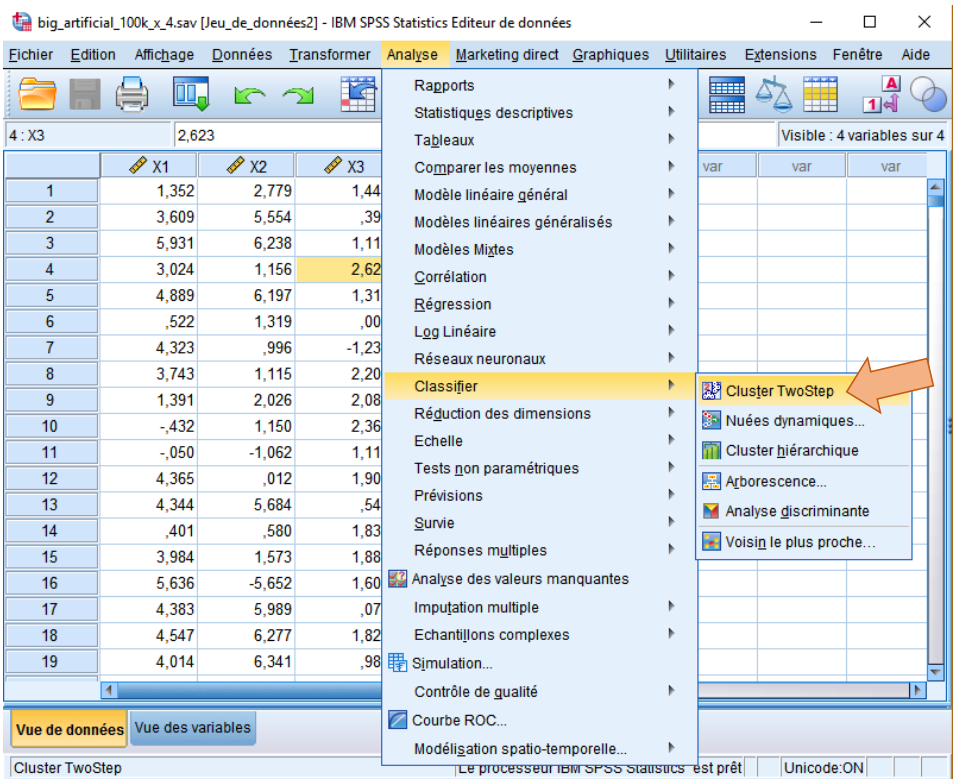
Deux critères seront importants pour juger de l'efficacité de l'approche : le temps de traitement, la capacité à détecter le bon nombre de classes compte tenu du rôle perturbateur des variables « bruit » X_3 et X_4.

3.1 Chargement des données

L'importation du fichier « big_artificial.txt » ne pose aucune difficulté sous SPSS. Nous pouvons le visualiser dans l'éditeur de données.



3.2 Analyse TwoStep Cluster



Nous actionnons le menu ANALYSE / CLASSIFIER / CLUSTER TWOSTEP. Dans la boîte de paramétrage qui apparaît, nous sélectionnons les variables actives de l'étude.

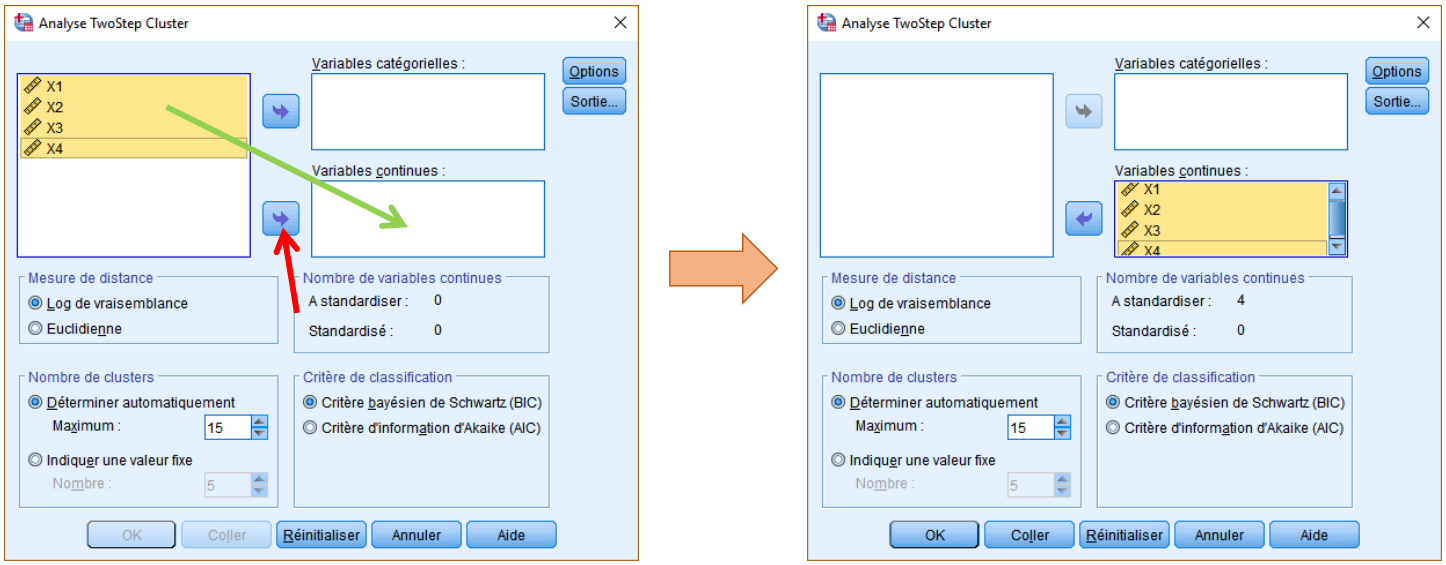
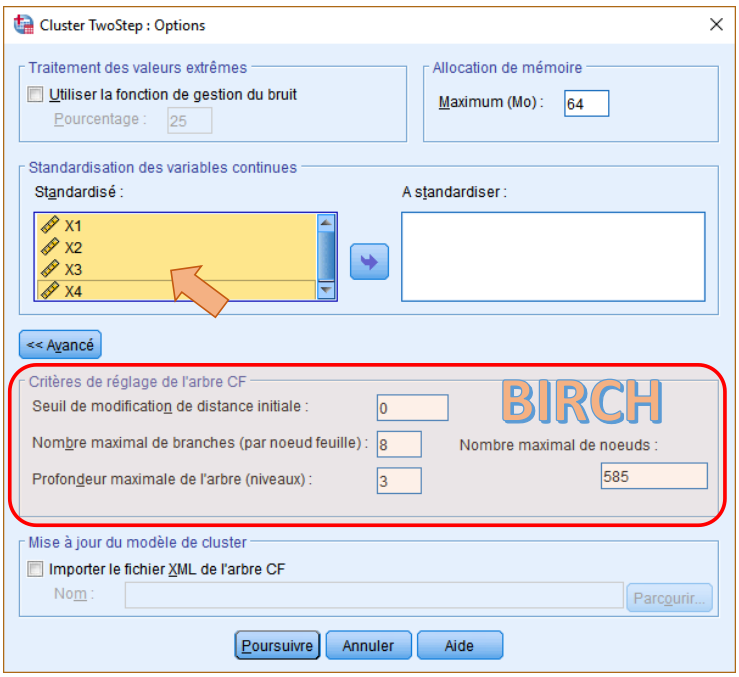


Figure 3 - Fenêtre initiale de paramétrage - SPSS

Nous remarquons que la mesure de distance est « Log de vraisemblance », et que le nombre maximum de clusters est fixé à $K_{max} = 15$.

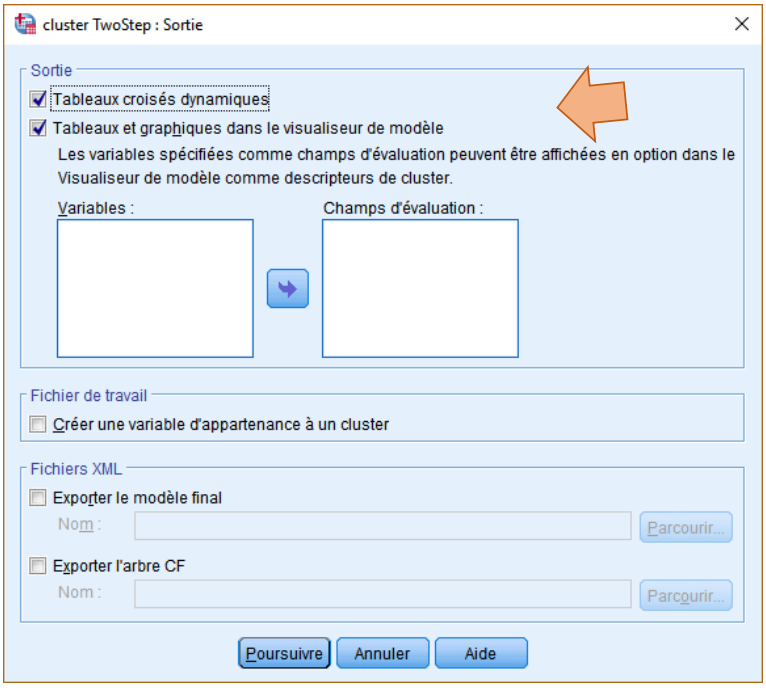
Il nous faut affiner le paramétrage, nous cliquons sur le bouton OPTIONS.



Il n'est pas nécessaire de standardiser les variables. Dans la partie basse de la boîte de dialogue, nous observons les paramètres de l'algorithme BIRCH destiné à générer les pre-

clusters. Augmenter le « seuil de modification de distance initiale » permet de réduire le nombre de groupes initiaux, mais identifier sa valeur adéquate en rapport avec le problème traité est difficile. On laisse « Seuil = 0 » généralement.

En revenant dans la fenêtre initiale (Figure 3), nous actionnons le bouton SORTIE. Dans la nouvelle fenêtre qui apparaît, nous indiquons les éléments à inclure dans le rapport, essentiellement les tableaux.



Il ne reste plus qu'à lancer les calculs en cliquant sur OK dans la fenêtre initiale (Figure 3).

Quelques **11 secondes** suffisent pour produire les résultats. C'est positivement impressionnant, n'oublions pas que notre fichier comporte n = 1.000.000 d'observations.

Voyons le rapport SPSS dans le détail.

DEBUT DU RAPPORT

SPSS traduit la commande utilisée dans sa syntaxe interne.

```
TWOSTEP CLUSTER
  /CONTINUOUS VARIABLES=X1 X2 X3 X4
  /DISTANCE LIKELIHOOD
  /NUMCLUSTERS AUTO 15 BIC
  /NOSTANDARDIZE VARIABLES=X1 X2 X3 X4
  /HANDLENOISE 0
  /MEMALLOCATE 64
  /CRITERIA INITHRESHOLD(0) MXBRANCH(8) MXLEVEL(3)
  /VIEWMODEL DISPLAY=YES
  /PRINT IC COUNT SUMMARY.
```

Les caractéristiques de l'analyse sont résumées. Nous disposons des indications sur le temps de traitement.

TwoStep Cluster

Remarques

Sortie obtenue		22-OCT-2016 09:34:10
Commentaires		
Entrée	Données	D:\DataMining\Databases_for_mining\dataset_for_soft_dev_and_comparison\clustering\two_step_clustering\big_artificial_100k_x_4.sav
	Jeu de données actif	Jeu_de_données2
	Filtre	<sans>
	Pondération	<sans>
	Fichier scindé	<sans>
Gestion des valeurs manquantes	Définition de la valeur manquante	Les valeurs manquantes définies par l'utilisateur sont traitées comme étant manquantes.
	Observations utilisées	Les statistiques sont basées sur toutes les observations comportant des données valides pour l'ensemble des variables utilisées dans l'analyse.
Syntaxe		TWOSTEP CLUSTER /CONTINUOUS VARIABLES=X1 X2 X3 X4 /DISTANCE LIKELIHOOD /NUMCLUSTERS AUTO 15 BIC /NOSTANDARDIZE VARIABLES=X1 X2 X3 X4 /HANDLENOISE 0 /MEMALLOCATE 64 /CRITERIA INITHRESHOLD(0) MXBRANCH(8) MXLEVEL(3) /VIEWMODEL DISPLAY=YES /PRINT IC COUNT SUMMARY.
Ressources	Temps de processeur	00:00:10.97
	Temps écoulé	00:00:11.10
Fichiers enregistrés	Modèle	C:\Users\Ricco\AppData\Local\Temp\spss9372\tsctempm.5

Le nœud de l'affaire est dans le tableau de « création automatique de clusters » ci-dessous. Nous disposons du BIC pour chaque solution de partition. **Le mécanisme conclut à une subdivision « optimale » en K* = 3 classes**. Ce qui est erroné. Nous savons que le bon nombre est K = 4.

Remarque : Notons que le critère BIC baisse constamment – *Modification BIC est toujours négatif* – à mesure que l'on augmente K, jusqu'à K_{max} = 15 en tous les cas. Il est inadapté sur cet exemple pour désigner directement le bon nombre de classes. SPSS a raison de ne pas le prendre pour argent comptant et d'introduire un dispositif additionnel.

Création automatique de clusters

Nombre de clusters	Critère bayésien de Schwartz (BIC)	Modification BIC[a]	Rapport des modifications BIC[b]	Rapport des mesures de distance[c]
1	6317655,892			
2	5682168,416	-635487,476	1,000	1,543
3	5270446,170	-411722,246	,648	**2,831**
4	5125085,094	-145361,076	,229	1,247
5	5008524,361	-116560,734	,183	1,349
6	4922142,907	-86381,453	,136	1,093
7	4843106,668	-79036,239	,124	1,257
8	4780273,598	-62833,070	,099	1,108
9	4723577,154	-56696,444	,089	1,026
10	4668297,756	-55279,399	,087	1,047
11	4615516,913	-52780,842	,083	1,102
12	4567619,418	-47897,495	,075	1,105
13	4524287,431	-43331,987	,068	1,284
14	4490576,939	-33710,492	,053	1,038
15	4458115,846	-32461,093	,051	1,101

a. Les modifications correspondent au nombre précédent de clusters dans la table.

b. Les rapports de modification sont fonction de la modification de la solution à deux clusters.

c. Les rapports des mesures de distance sont basés sur le nombre actuel de clusters, comparé au nombre précédent de clusters.

Le ratio entre les deux plus grandes valeurs de rapport des mesures de distance(c) est égal à $\frac{2.831}{1.543} = 1.834 > 1.15$, on conclut bien à une partition en K* = 3 classes.

La distribution de clusters indique les effectifs dans chaque classe.

Distribution de clusters

		N	% des combinés	% du total
Cluster	1	252263	25,2%	25,2%
	2	252369	25,2%	25,2%
	3	495368	49,5%	49,5%
	Combiné	1000000	100,0%	100,0%
Total		1000000		100,0%

Le profil de cluster indique – entre autres - les coordonnées des barycentres conditionnels.

Profils de cluster

Centroïdes

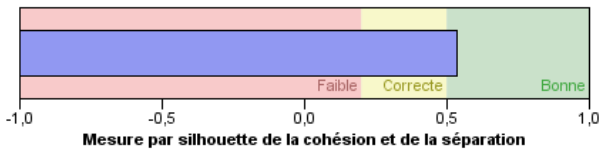
		X1		X2		X3		X4	
		Moyenne	Ecart type	Moyenne	Ecart type	Moyenne	Ecart type	Moyenne	Ecart type
Cluster	1	4,99877	,848746	-4,96724	,922581	1,00004	,850970	,99985	,849652
	2	,01111	,851594	1,99208	,878274	1,00317	,851379	1,00165	,848850
	3	4,00884	,831076	3,28047	2,389629	1,00039	,849147	,99998	,851189
	Combiné	3,24966	2,100254	,87473	3,876797	1,00100	,850171	1,00037	,850211

Enfin, la qualité des clusters correspond à l'indice silhouette. La partition est juste « bonne ».

Récapitulatif du modèle

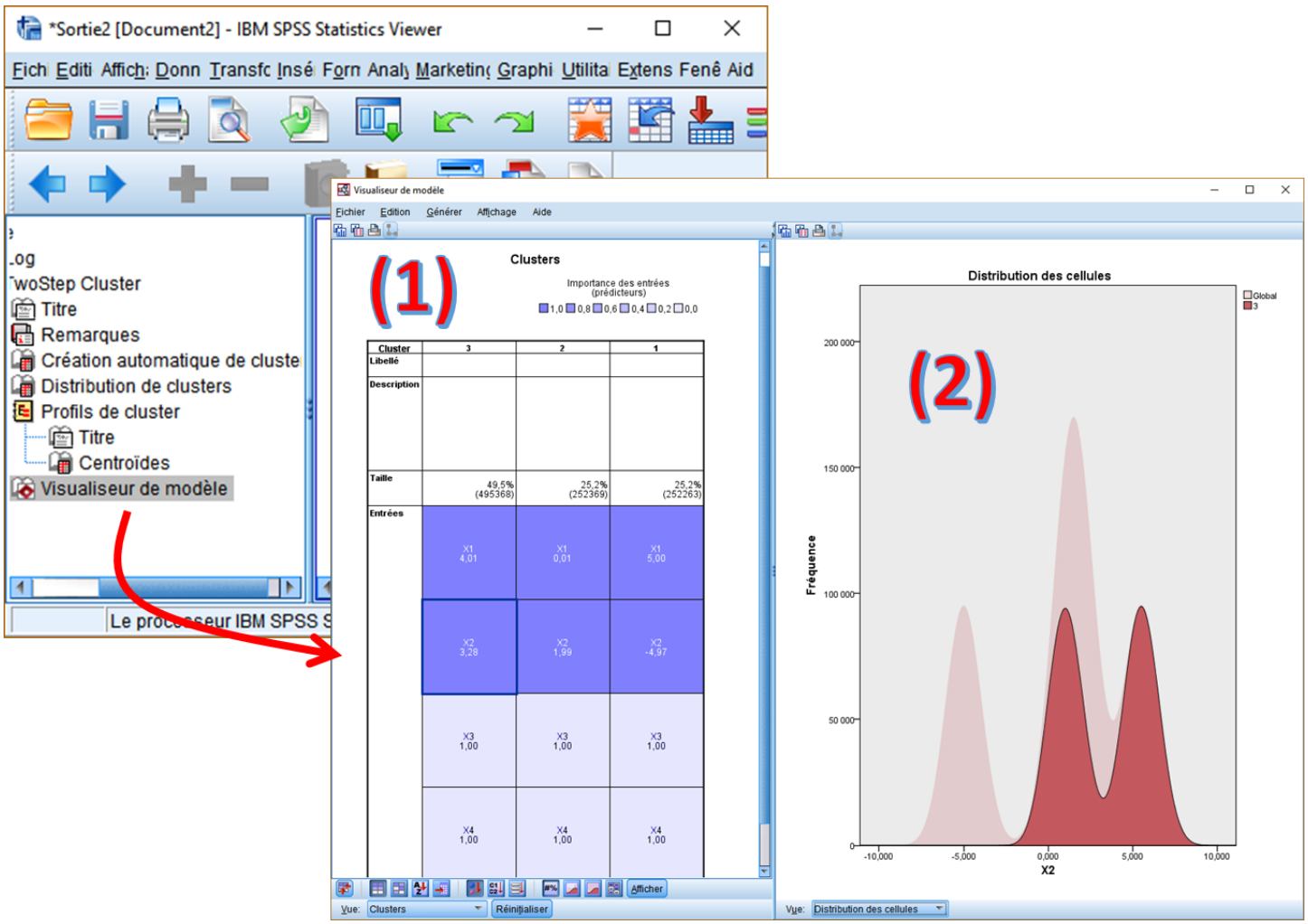
Algorithme	TwoStep
Entrées	4
Clusters	3

Qualité des clusters



FIN DU RAPPORT

En double-cliquant sur cette dernière partie du rapport dans **IBM SPSS Statistics Viewer**, nous avons accès à une nouvelle fenêtre « Visualisateur de modèle » où nous pouvons sélectionner de nouveaux indicateurs.



Nous y observons notamment (1) l'importance des variables dans la définition des classes, plus la couleur est foncée, plus la variable est déterminante ; (2) les distributions conditionnelles.

De très nombreuses options sont disponibles dans le « Visualisateur de modèle ». L'outil est très riche, et les multiples points e vue permettent de cerner la nature des classes. Mais d'un autre côté, il est vite facile de s'y perdre à force de faire joujou.

4 Autres implémentations d'approches mixtes

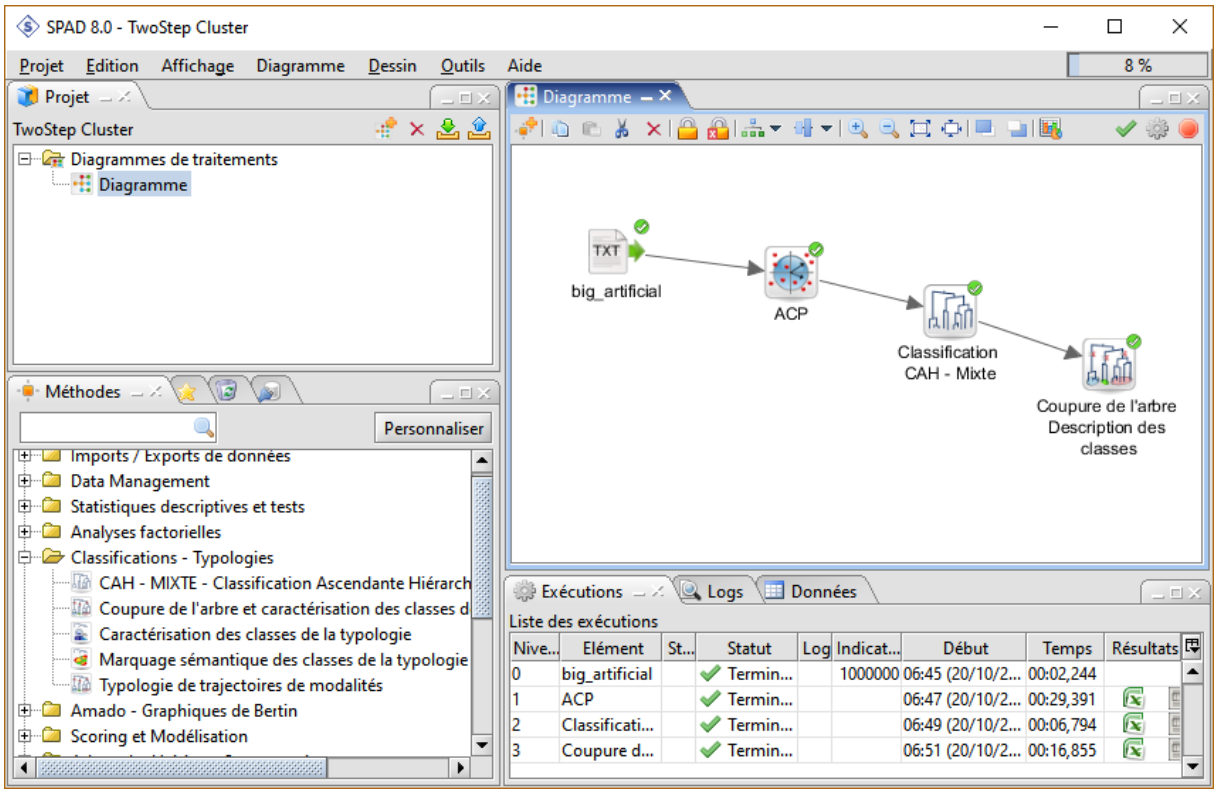
SPSS n'a pas l'exclusivité de l'approche mixte. Elle est disponible dans d'autres outils. Nous pouvons aussi la programmer relativement facilement sous R par exemple. Dans cette

section, nous évaluons les procédures sous SPAD, TANAGRA et R, avec toujours les deux critères clés que sont la rapidité de traitement et la détection du nombre de classes qui est égal à K = 4 rappelons-le.

4.1 Traitements sous SPAD

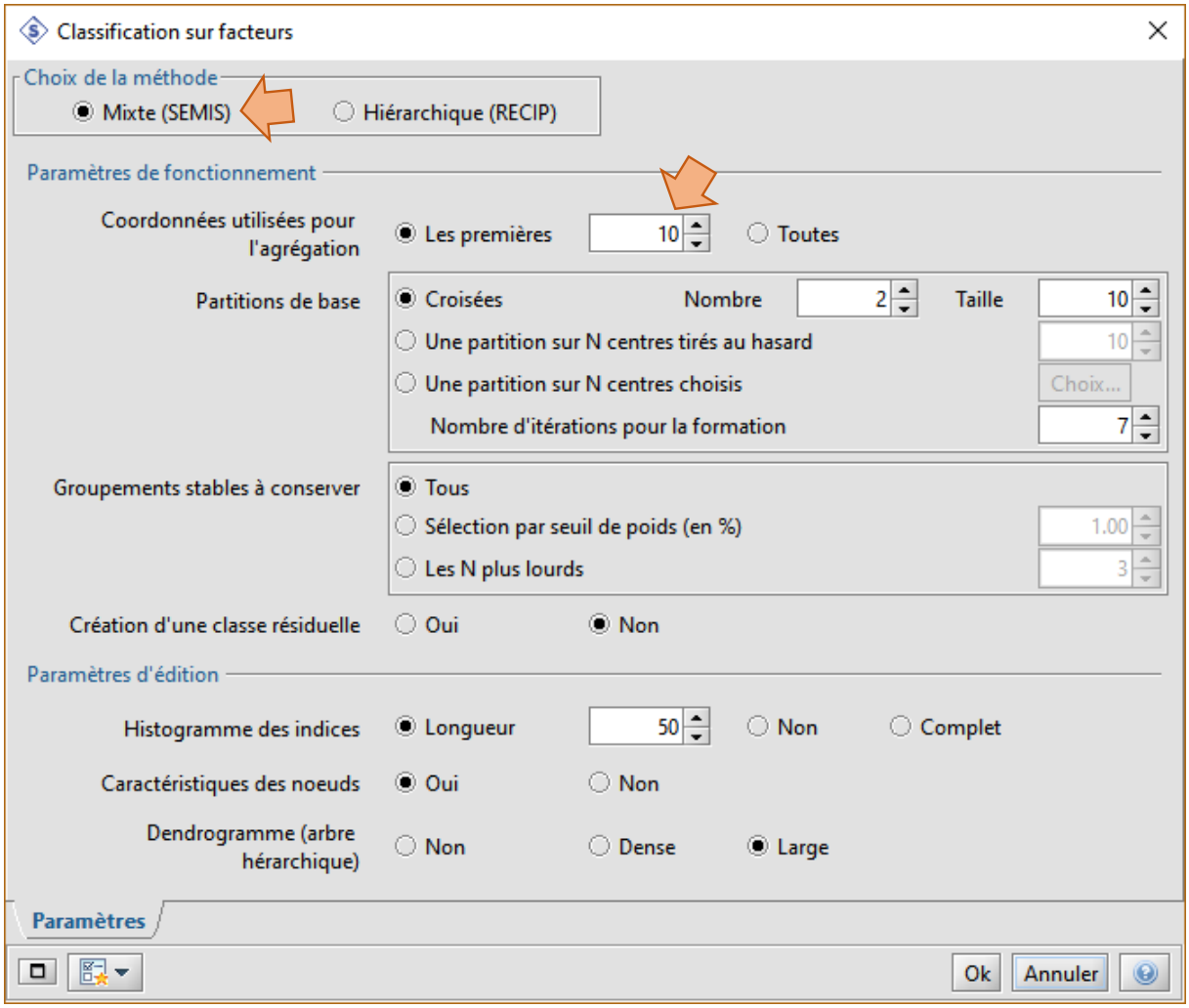
Nous utilisons SPAD version 8.0 dans cette section. Autant que je me rappelle (mon premier contact avec le logiciel doit dater du début des années 90), la classification mixte a toujours tenu une bonne place dans le logiciel SPAD car il permet de traiter des grandes bases de données avec des temps de calculs raisonnables. La méthode SEMIS est ainsi déjà présente dans l'antique SPAD.N (sans interface graphique encore) (Morineau, 1991).

Le processus global est visible dans la copie d'écran suivante :



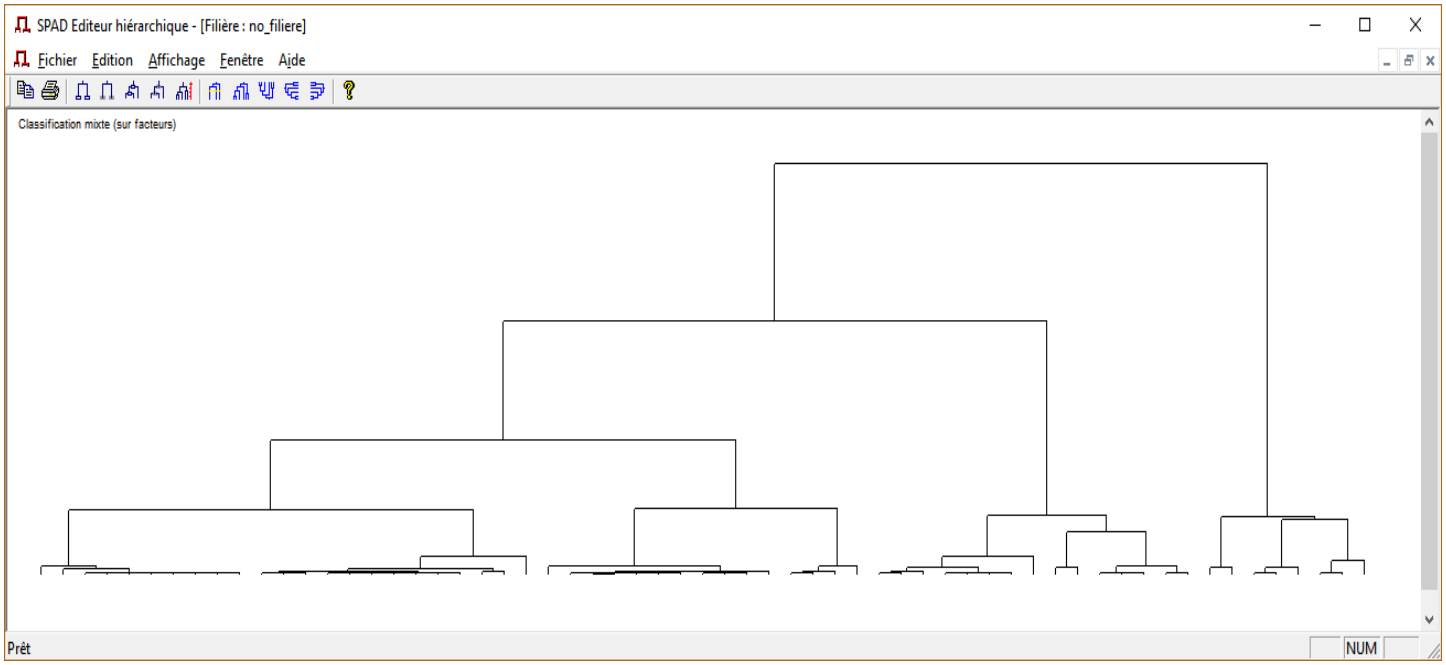
L'importation des données ne pose aucun problème (2 secondes pour 1.000.000 d'observations). La CAH MIXTE opère sur les axes factoriels issus d'une ACP (analyse en composantes principales) sur les variables initiales. Cette étape préalable se justifie par le fait que les facteurs sont deux à deux orthogonaux, la distance euclidienne devient pleinement légitime pour comptabiliser les proximités entre les individus.

Nous plaçons ensuite la CAH MIXTE dans l'espace de travail. Le paramétrage n'appelle pas de commentaires particuliers, si ce n'est que nous utilisons l'ensemble des facteurs de l'ACP dans notre exemple (les 10 premières sont spécifiées, or l'ACP ne produit que 4 facteurs sur notre exemple). Nous utilisons bien la méthode SEMIS. Selon la documentation du logiciel, la construction des pre-clusters est basée sur un algorithme K-means. Dans notre cas, 2 partitions de taille 10 sont générées, nous partons donc de 10 x 10 = 100 groupes pour démarrer la CAH (classification ascendante hiérarchique) subséquente. SPAD sait également construire des classes résiduelles pour isoler les données « bruit » correspondant à des *outliers* (points atypiques).



Le temps de traitement est de 6 secondes, presque moitié moins que sous SPSS. Après, la question est de savoir s'il faut intégrer la constitution des axes factoriels (16 secondes) dans le décompte de la durée des calculs. Quoiqu'il en soit, la rapidité du dispositif est tout aussi impressionnante que celle de SPSS.

Le dendrogramme est visible dans l'éditeur hiérarchique.



Il n'y a pas vraiment de solution tranchée. Nous utilisons l'outil COUPURE DE l'ARBRE pour identifier automatiquement le bon nombre de classes.

SPAD propose des scénarios de solutions, triées par ordre de pertinence. Sa préférence va vers un découpage en K* = 4 classes, qui correspond à la solution correcte (les autres pistes étaient K = 9 et K = 10). Le rapport montre qu'elles sont de taille homogène.

```
Coupure 'a' de l'arbre en  4 classes
FORMATION DES CLASSES (INDIVIDUS ACTIFS)
DESCRIPTION SOMMAIRE
+--------+----------+-----------+-----------+
| CLASSE | EFFECTIF |   POIDS   |  CONTENU  |
+--------+----------+-----------+-----------+
|  aa1a  | 264790   |264790.00  |   1 A  23 |
|  aa2a  | 234930   |234930.00  |  24 A  38 |
|  aa3a  | 247989   |247989.00  |  39 A  53 |
|  aa4a  | 252291   |252291.00  |  54 A  61 |
+--------+----------+-----------+-----------+
```

SPAD procède ensuite à une consolidation autour des centres de classes pour améliorer la qualité des clusters. Dans notre exemple, le gain obtenu à l'issue de ce post-traitement est négligeable.

```
CONSOLIDATION DE LA PARTITION
AUTOUR DES   4 CENTRES DE CLASSES, REALISEE PAR 10 ITERATIONS A CENTRES MOBILES
PROGRESSION DE L'INERTIE INTER-CLASSES
+-----------+------------+------------+------------+
| ITERATION |   I.TOTALE |    I.INTER |  QUOTIENT  |
+-----------+------------+------------+------------+
```

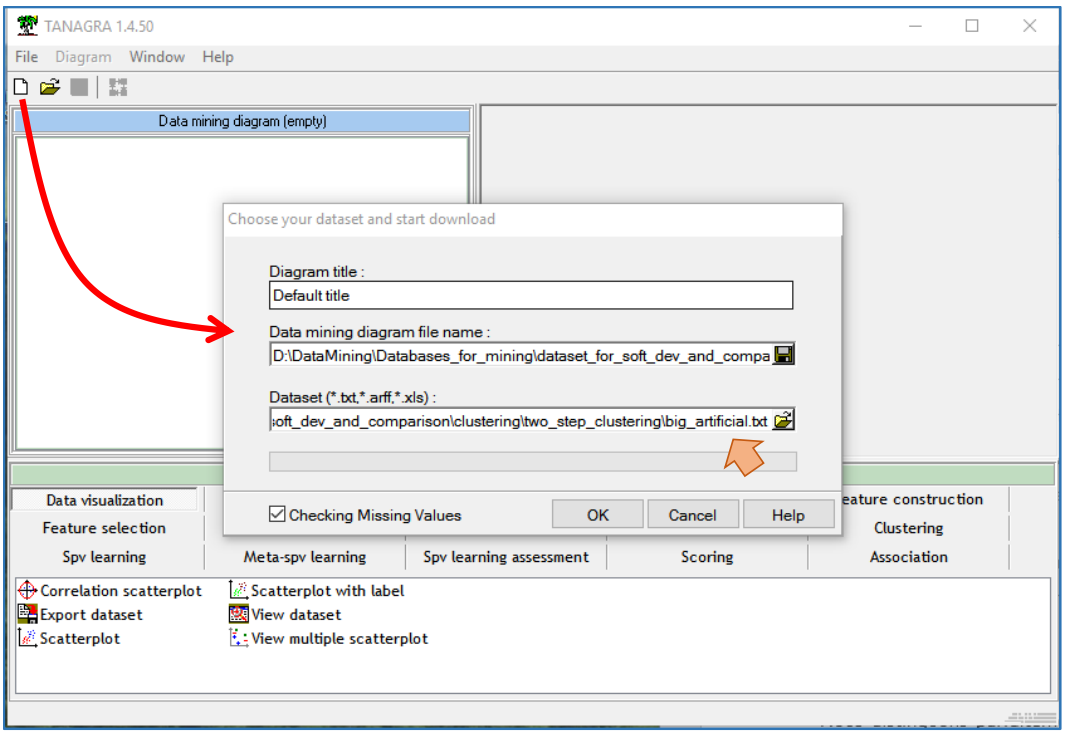
```
|     0     |    4.00224 |    1.93602 |   0.48373  |
|     1     |    4.00223 |    1.95289 |   0.48795  |
|     2     |    4.00223 |    1.95310 |   0.48800  |
|     3     |    4.00223 |    1.95315 |   0.48802  |
+-----------+------------+------------+------------+
ARRET APRES L'ITERATION  3 L'ACCROISSEMENT DE L'INERTIE INTER-CLASSES
PAR RAPPORT A L'ITERATION PRECEDENTE N'EST QUE DE 0.002 %.
```

SPAD propose de nombreux outils pour mieux apprécier la teneur et la qualité des résultats. La description des classes notamment permet de positionner les groupes dans l'espace de représentation. La notion de valeur-test permet de situer les contributions des variables.

4.2 Traitements sous Tanagra

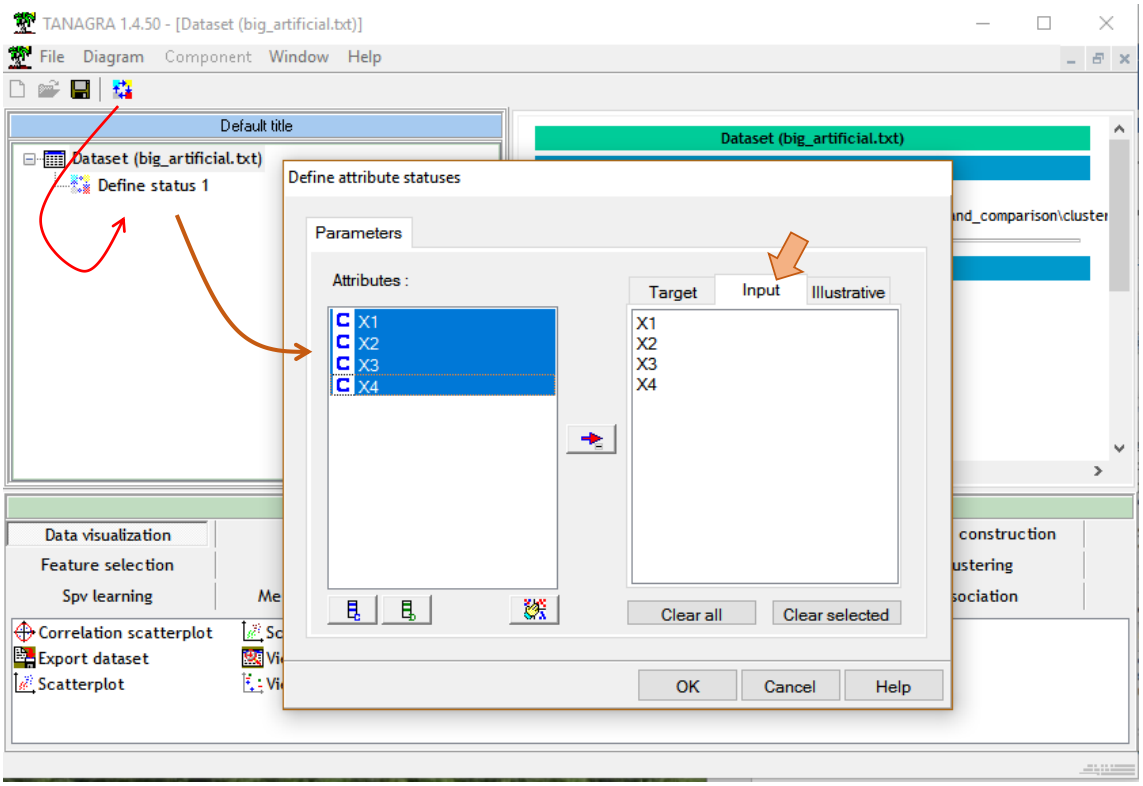
Tanagra 1.4.50 ne dispose pas de composant spécifiquement dédié à la classification mixte (pour l'instant…). Mais il est possible de la reproduire en enchaînant les outils idoines dans le diagramme de traitements. La mise en œuvre sous Tanagra a déjà été détaillée dans un précédent tutoriel (Traitement des gros volumes – CAH Mixte, octobre 2008). Nous irons donc à l'essentiel dans cette section.

Importation des données. Pour importer un fichier texte, nous lançons Tanagra et nous créons un nouveau diagramme. Nous indiquons le fichier à charger « big_artificial.txt ». *Attention, si le point décimal est autre que « . » sur votre système, vous devez modifier le fichier en conséquence avant de l'importer (avec un éditeur de texte quelconque)*.

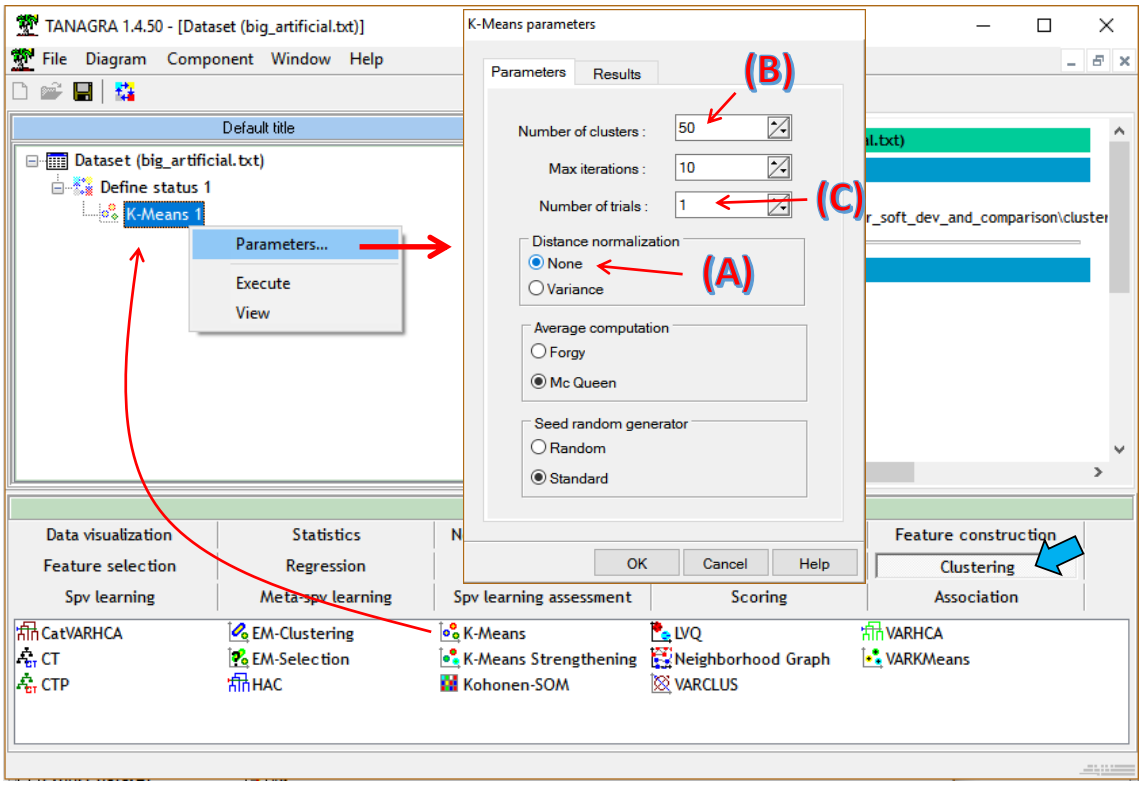


L'importation est très rapide, moins d'une seconde.

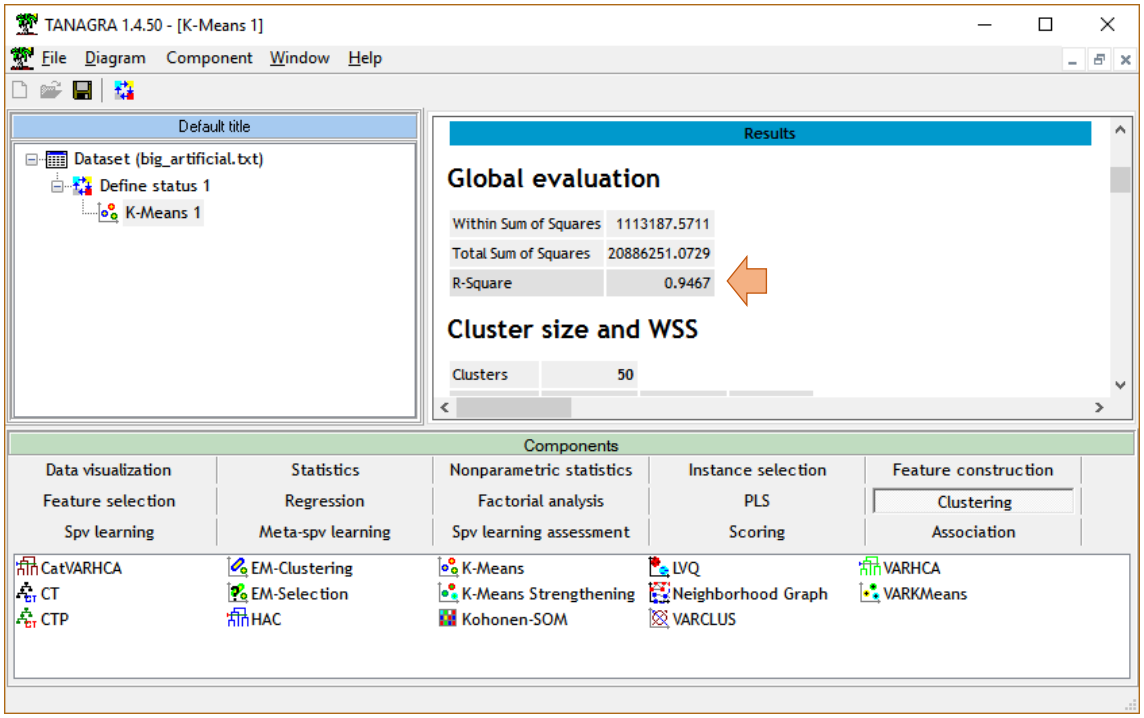
K-Means. Nous utilisons la méthode des K-Means pour construire les pre-clusters. Les variables actives (X_1 à X_4) sont placées en INPUT à l'aide de l'outil DEFINE STATUS.



Puis nous insérons l'outil K-MEANS (onglet CLUSTERING) avec le paramétrage suivant :

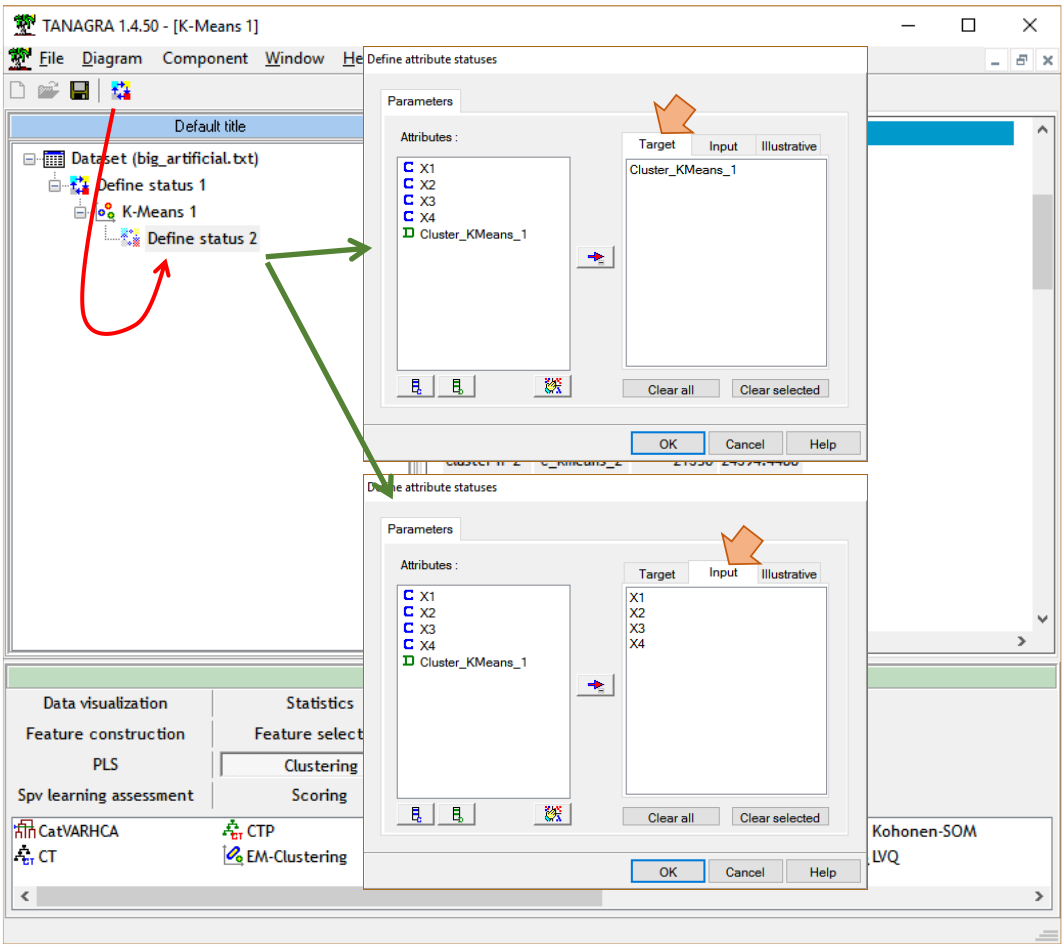


(A) Il n'est pas nécessaire de standardiser les données (Distance normalization = None) ; (B) nous demandons 50 groupes (Number of Clusters = 50) ; nous sommes dans une étape préalable, il n'est pas nécessaire de construire la meilleure partition possible, un seul essai (Number of Trials = 1) (C) suffira dans cette première phase (*habituellement, la méthode est lancée plusieurs fois et la meilleure partition au sens de l'inertie expliquée est retenue*). Nous lançons les calculs (menu contextuel VIEW) après avoir validé les paramètres.

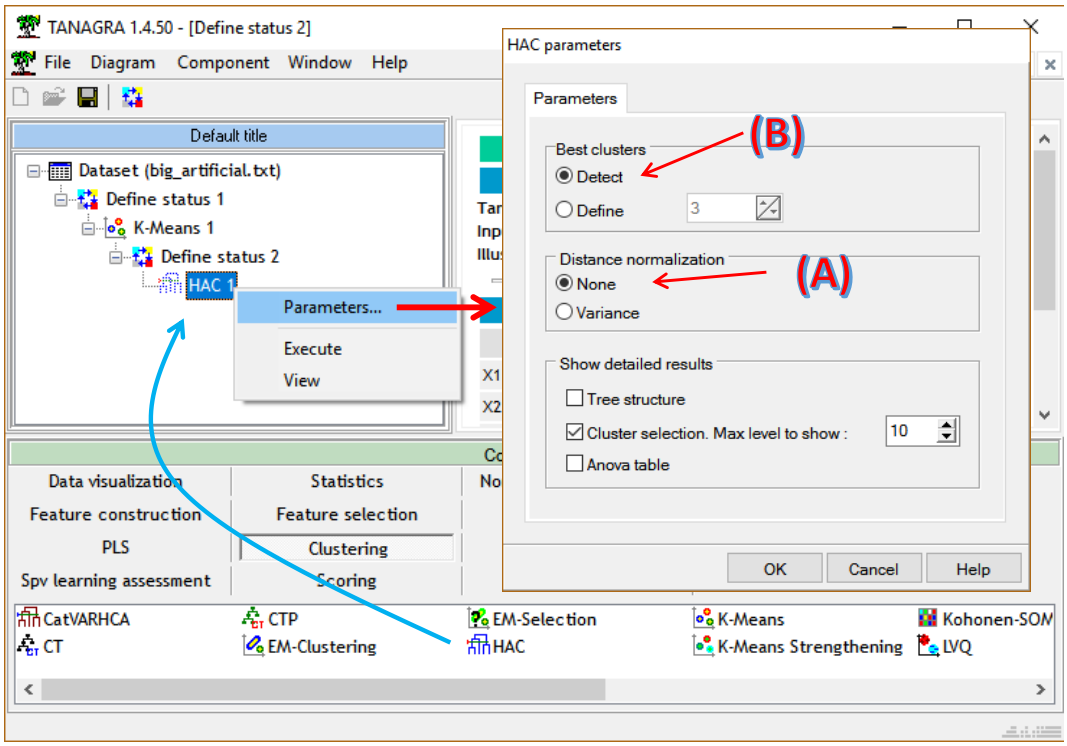


La partition en 50 classes explique 94.67% de l'inertie. Cette valeur ne veut pas dire grand-chose à ce stade. Du côté de la vélocité des calculs, notons que les résultats sont venus après 23 secondes. Ça reste raisonnable pour un échantillon de 1.000.000 d'observations. Mais l'implémentation des K-Means est assez basique dans Tanagra, il y a certainement un travail d'optimisation à faire pour les prochaines versions. On doit pouvoir faire mieux.

CAH à partir des groupes issus des K-Means. Pour instancier la classification ascendante hiérarchique, nous insérons de nouveau DEFINE STATUS. Nous plaçons : en TARGET les classes issues des K-Means (CLUSTER_KMEANS_1), en INPUT les variables actives (X_1 à X_4).

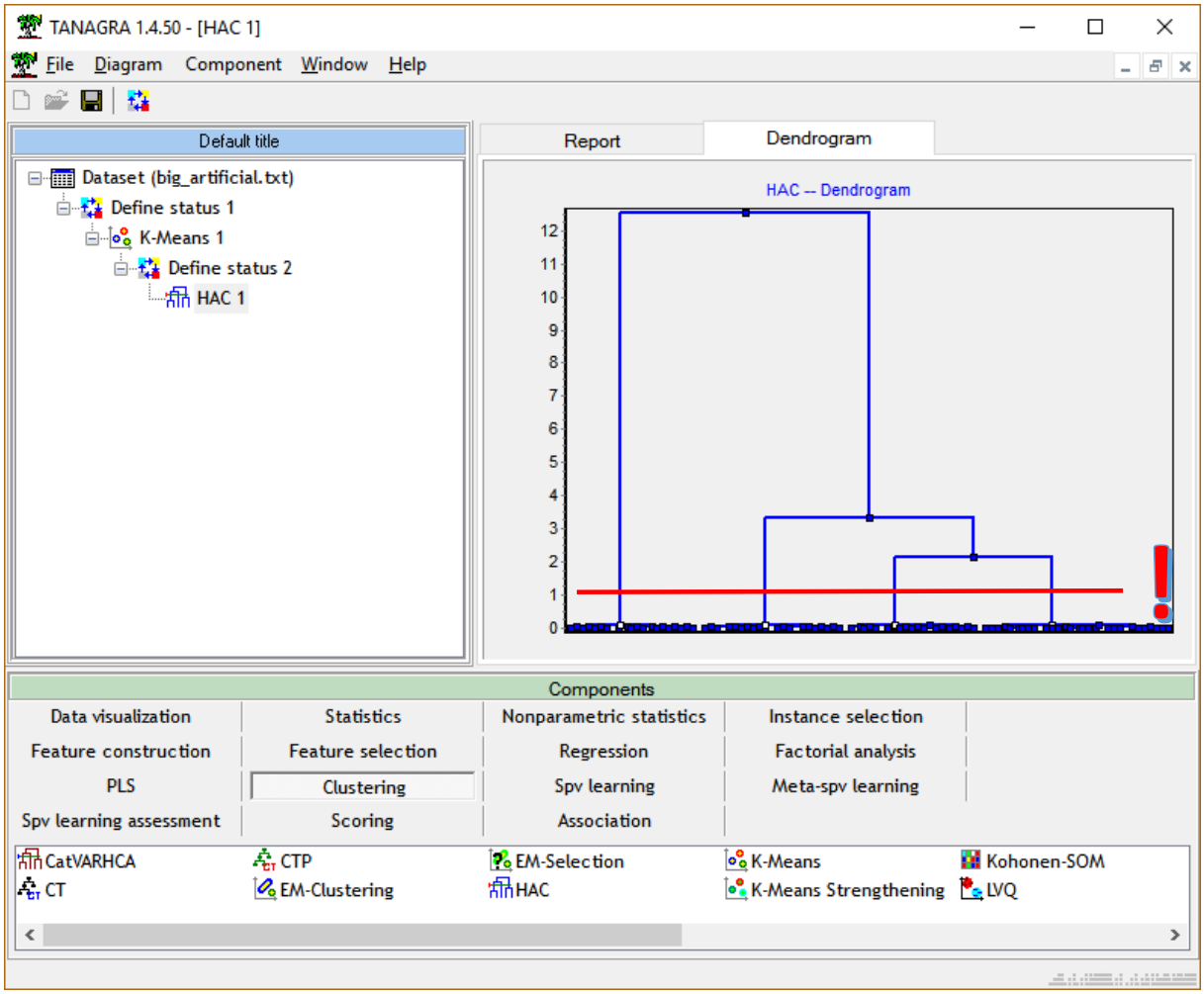


Nous insérons ensuite le composant HAC (onglet CLUSTERING).

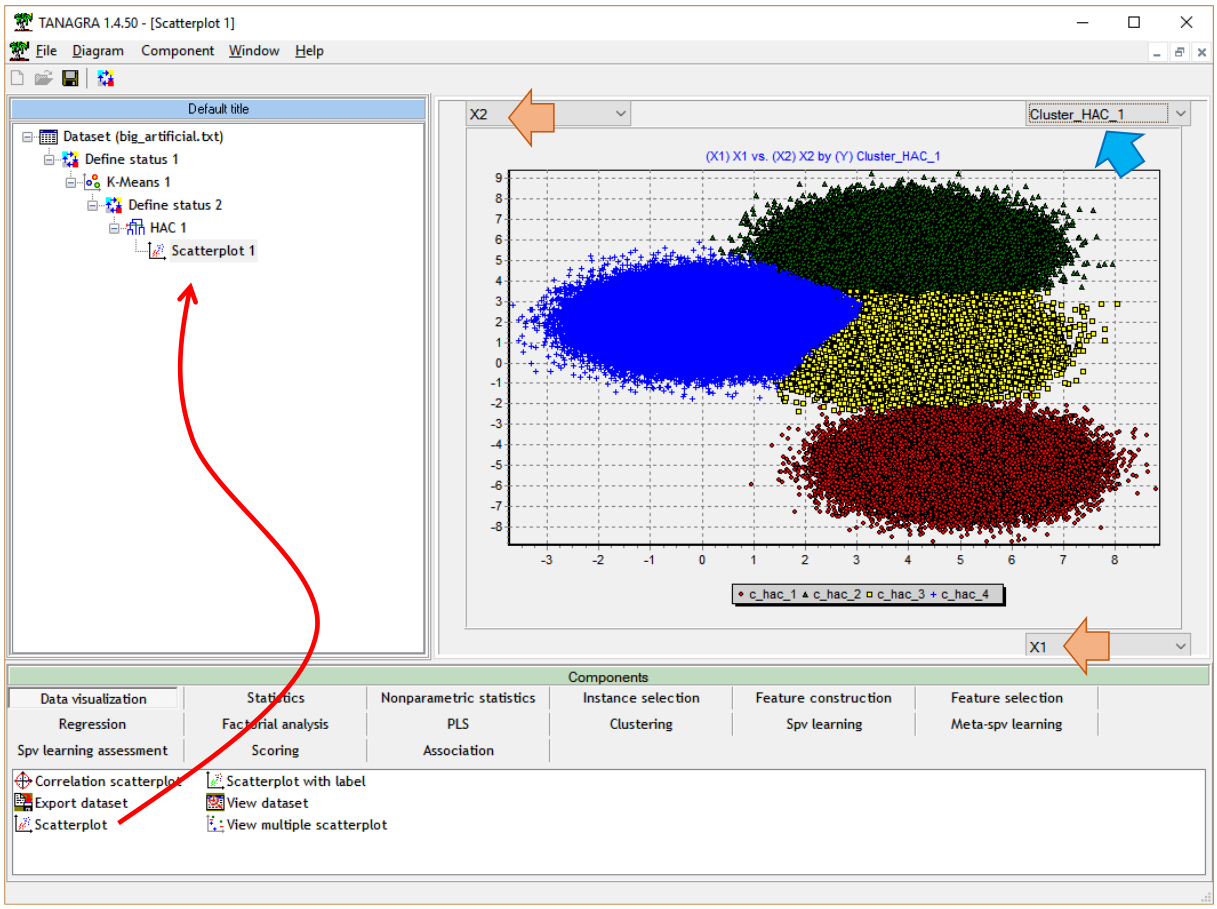


Dans la fenêtre de paramétrage, nous désactivation la normalisation des données (A), nous veillons à ce que l'outil sélectionne automatique le nombre de classes (B).

Il reste à lancer les calculs. En moins d'une demi-seconde, Tanagra construit la hiérarchie et choisit une partition en K* = 4 classes en se basant sur les écarts entre les hauteurs des paliers de regroupement. Sachant que Tanagra ignore délibérément la partition en 2 classes qui paraît quasiment toujours évidente dans la CAH, la solution en 4 classes paraît s'imposer à la vue du dendrogramme.



Positionnement des classes dans le plan (X1, X2). K* = 4 est, certes, la bonne solution. Mais est- ce que les groupes produits matchent avec la position réelle des classes dans l'espace de représentation (X1, X2) ? Pour le savoir, nous insérons l'outil SCATTERPLOT (onglet DATA VISUALISATION) dans le diagramme. Nous plaçons X1 en abscisses, X2 en ordonnée, nous illustrons les points par la variable « classe d'appartenance » construite automatiquement par la CAH (CLUSTER_HAC_1).



Nous distinguons parfaitement les classes dans des positions conformes à ce que nous avions noté dans les graphiques croisant les variables deux à deux (Figure 2). En plus de détecter le bon nombre de classes, Tanagra les a correctement identifiées.

4.3 Traitements sous R

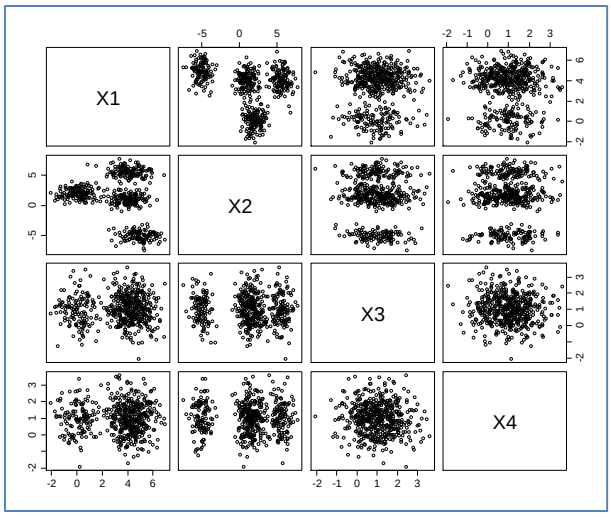
Les manipulations sous R ont été décrites dans le même précédent tutoriel (Traitement des gros volumes – CAH Mixte, octobre 2008). Nous montrons directement le code commenté.

```
#importation des données the dataset, en moins de 2 secondes
X <- read.table(file="big_artificial.txt",header=T,sep="\t")
print(head(X))

#initialisation du générateur de nombre aléatoire
#pour rendre reproductible l'expérimentation
set.seed(100)

#identifiant des individus pour l'échantillonnage
ids <- sample(1:nrow(X),500,replace=F)

#paires de nuages de point sur l'échantillon
#cf. graphique de la Figure 2
pairs(X[ids,])
```




```
#lancement de la méthode k-means
#50 pre-clusters (sous-groupes) demandés
#comme sous Tanagra, un essai (nstart), et un maximum de 10 itérations (iter.max)
#opération réalisée en 6 secondes
km <- kmeans(X,centers=50,iter.max=10,nstart=1)

#effectif dans chaque sous-groupe
print(table(km$cluster))
```

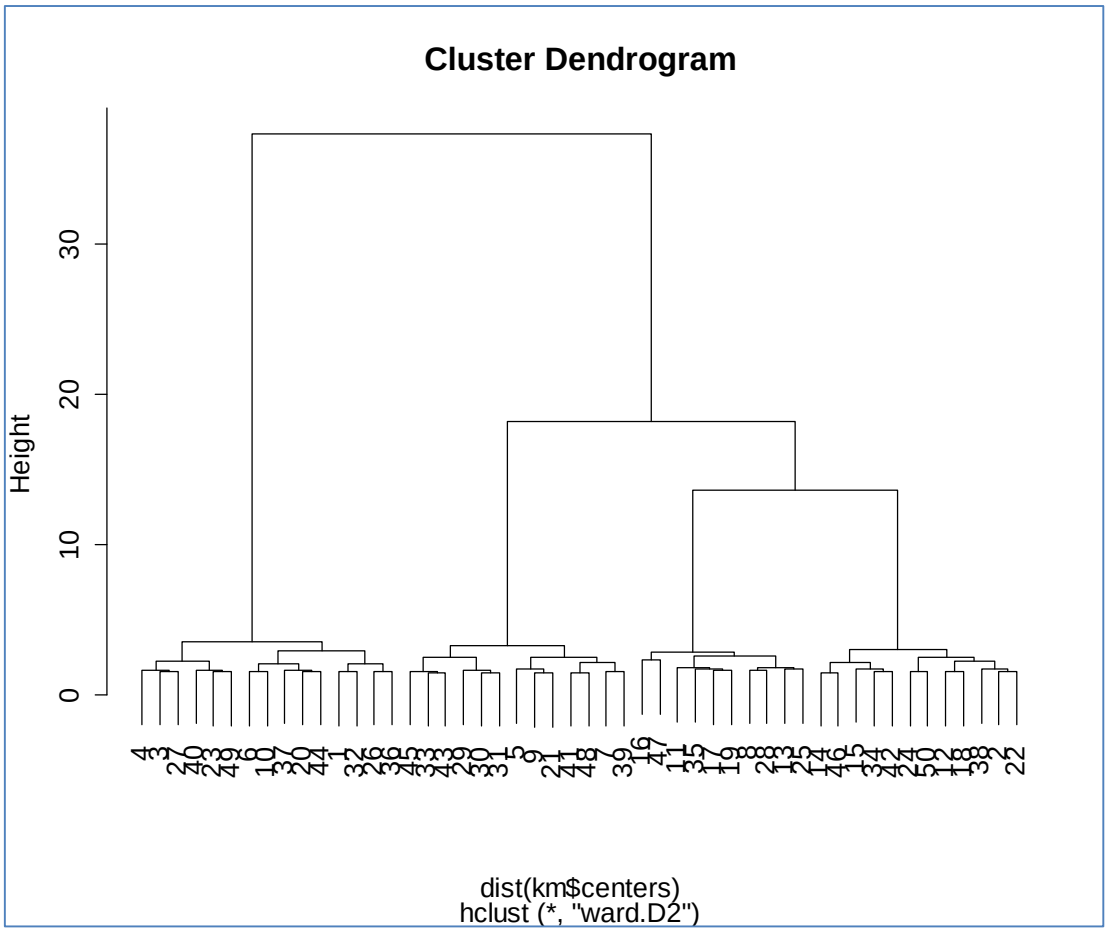
Nous obtenons 50 sous-groupes relativement équilibrés avec des effectifs allant de 15000 à 20000 observations.

```
> print(table(km$cluster))
    1     2     3     4     5     6     7     8     9    10    11    12    13    14    15
16    17    18    19    20    21
16698 20947 17028 16945 19010 15813 18099 31120 22561 17357 27713 22461 28391 21529 19750
13464 30802 19124 26346 16602 16929
   22    23    24    25    26    27    28    29    30    31    32    33    34    35    36
37    38    39    40    41    42
19861 16191 21492 31232 16946 16466 26282 19590 17393 20773 16666 19224 18495 28495 16137
16632 20336 16586 16976 20424 21743
   43    44    45    46    47    48    49    50
19826 16741 19434 19251 14583 16447 16911 20178
```

```
#réalisation de la CAH à partir des sous-groupes
#matrice de distances entre les barycentres conditionnels des 50 groupes
#méthode de Ward
#les feuilles du dendrogramme correspondent à des sous-groupes
#il faut fournir les effectifs associés
#calcul en moins d'une demi-seconde
cah <- hclust(dist(km$centers),method="ward.D2",members=table(km$cluster))

#affichage du dendrogramme
plot(cah)
```

Ici également, le regroupement en 4 classes semble évident (si l'on ignore la solution K = 2).




```
#découpage en 4 classes
intermed.g <- cutree(cah,k=4)

#nombre de sous-groupes dans chaque groupe
print(table(intermed.g))
```

Les 50 sous-groupes issus des K-Means ont été réunis en 4 grands groupes.

```
> print(table(intermed.g))
intermed.g
 1  2  3  4
15 12 13 10
```

On lit : 15 sous-groupes issus des K-Means ont été réunis dans la classe n°1 de la CAH, etc.

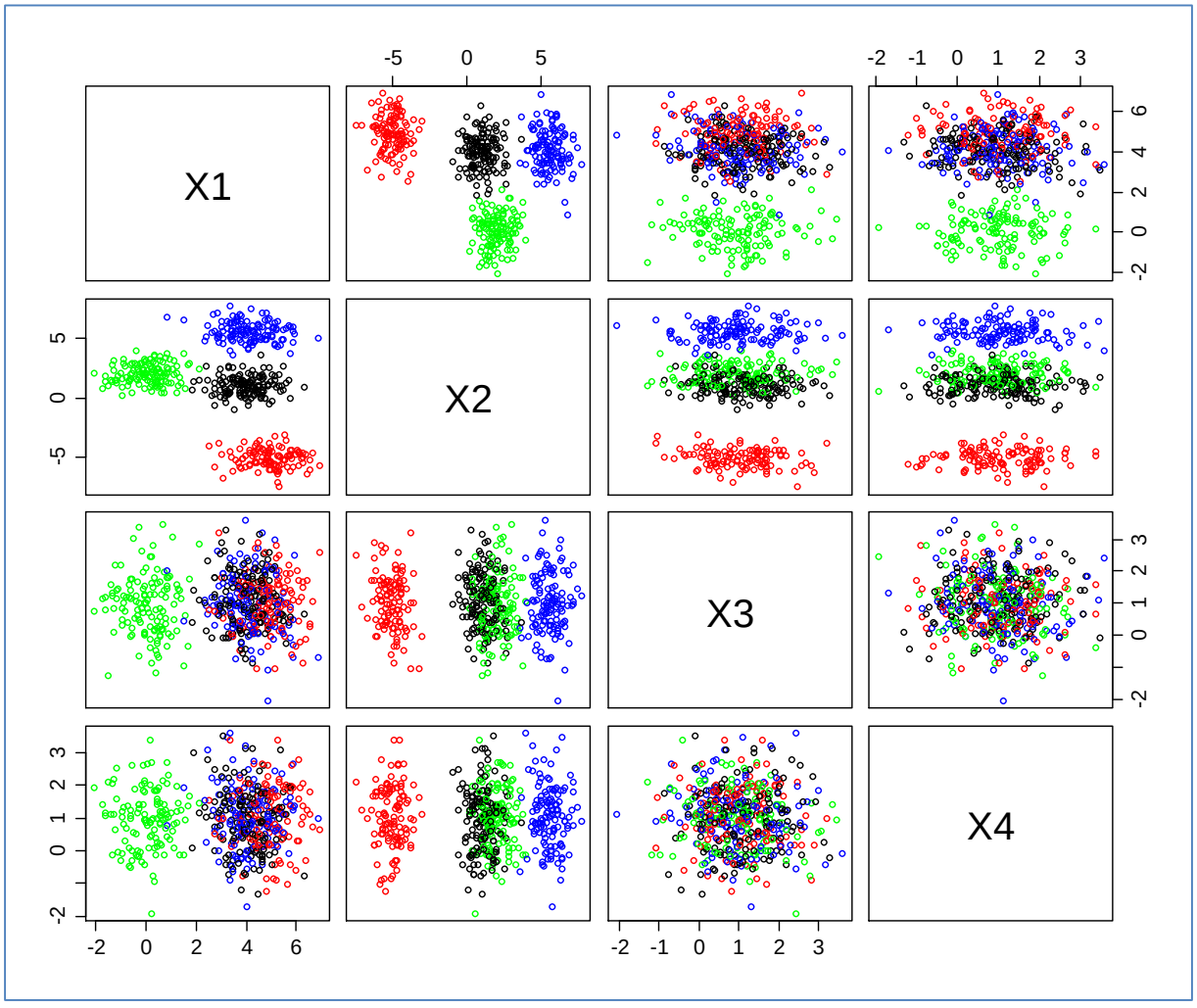
```
#affecter à chaque individu son groupe final CAH
final.g <- rep(0,nrow(X))
#boucle pour chaque cluster des k-means
for (i in unique(km$cluster)){
  final.g[which(km$cluster==i)] <- intermed.g[i] #associer son groupe CAH
}

#nombre d'observations dans chaque groupe CAH, la somme doit faire 1000000
table(final.g)
```

250109 observations se retrouvent dans la classe n°1, etc. Les classes sont relativement équilibrées :

```
> table(final.g)
final.g
     1      2      3      4
250109 245167 246296 258428
```

```
#représentation de l'échantillon dans les différents plans
#les individus sont coloriés selon leur classe d'appartenance
pairs(X[ids,],col=c("red","green","blue","black")[final.g[ids]])
```



R a parfaitement identifié les classes qui se démarquent dans le plan (X_1, X_2).

5 Conclusion

La classification mixte est une solution performante pour appréhender les grandes bases de données. La méthode TwoStep Cluster de SPSS ne déroge pas à la règle. Mais nous observons dans ce tutoriel que d'autres implémentations sont possibles, tout aussi rapides.

Finalement, la principale originalité de TwoStep Cluster est sa technique, un peu alambiquée quand même, de détection automatique du nombre de classes. La documentation n'est pas très précise. J'ai dû opérer parfois par déductions à partir des sorties du logiciel. Il reste que TwoStep Cluster a le mérite de bien fonctionner semble-t-il dans la majorité des cas (Bacher et al., 2004). Par rapport à cette référence – qui est une des très rares assez complètes que j'aie pu trouver sur le sujet, et qui date de 2004 – j'ai noté que l'algorithme de SPSS version 24 (2016) a été modifié, les ingrédients de base restant les mêmes. J'imagine qu'elles (les modifications) vont dans le sens de l'amélioration du dispositif.

6 Références

(Bacher et al., 2004) Bacher J., Wenzig K., Vogler M., "SPSS TwoStep Cluster – A first evaluation", 2004.

(Lebart et al., 1995) Lebart L., Morineau A., Piron M., "Statistique exploratoire multidimensionnelle", Dunod, 1995.

(Morineau, 1991) Morineau A., "SPAD.N – Logiciel pour l'Analyse Statistique des Données", Revue Modulad, n°6, pp. 27-60, 1991.

(SPSS 2001) SPSS Inc., "The SPSS TwoStep Cluster Component", SPSS White Paper, Technical Report TSCPWP-0101, 2001.

(Zhang et al., 1996) Zhang, T., Ramakrishnan, R., Livny, M., "BIRCH: an efficient data clustering method for very large databases". Proceedings of the 1996 ACM SIGMOD international conference on Management of data - SIGMOD '96. pp. 103–114, 1996.