

J.-P. BENZÉCRI

F. BENZÉCRI

**Programmes de création de tableaux :
notice d'utilisation**

Les cahiers de l'analyse des données, tome 14, n° 1 (1989),
p. 35-54

http://www.numdam.org/item?id=CAD_1989__14_1_35_0

© Les cahiers de l'analyse des données, Dunod, 1989, tous droits réservés.

L'accès aux archives de la revue « Les cahiers de l'analyse des données » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques

<http://www.numdam.org/>

PROGRAMMES DE CRÉATION DE TABLEAUX: NOTICE D'UTILISATION

[NOT. CRÉ. TAB.]

J.-P. & F. BENZÉCRI

1 Introduction: les tableaux de données

1.1 Tableau de base et tableau transformé

Dans [NOT. CORR. CAH] nous nous sommes placés dans la situation la plus simple: l'utilisateur part d'un tableau apte à être soumis tel quel à l'analyse des correspondances, par exemple un tableau de contingence; le tableau est saisi en utilisant un éditeur de texte, et, sitôt la saisie faite, on entre dans le programme 'qori', (puis, éventuellement dans 'CAH2').

Cependant, d'importants ensembles de données, dignes d'être analysés, ne se présentent pas sous la forme de tableaux de contingence: ils doivent être soumis à un codage préalable. Sans les règles du codage, échapperaient à l'analyse des données des domaines entiers, où ont été obtenus des résultats essentiels: enquêtes par questionnaires, expérimentation biologique...

Un des cas les plus simples est celui des tableaux de réponses à un questionnaire clos, où chaque réponse est initialement notée comme un numéro de modalité: un tel tableau doit être éclaté, sous forme binaire, en (zéro, un), en attribuant à chaque modalité une colonne. Et l'analyse porte, le plus souvent, non sur ce tableau binaire, mais sur le tableau de Burt, (ou tableau de cooccurrence des modalités), qui lui est associé; le tableau binaire étant adjoint en supplémentaire au tableau de Burt.

Ce modèle du questionnaire clos est important, non seulement en lui-même, mais parce que s'y ramène le traitement des ensembles de variables numériques hétérogènes. En effet, en découpant en p sous-intervalles successifs l'intervalle de variation d'une variable continue, on peut coder celle-ci comme une variable discrète à p modalités.

1.2 Codage et création de tableaux

On ne saurait concevoir un programme apte à réaliser toutes les formes de codage qui ont été utilisées avec succès, et encore moins toutes celles qu'on

pourra désirer expérimenter dans l'avenir. Les tableurs, dont l'usage est aujourd'hui si répandu parmi les gestionnaires et les économistes, peuvent jouer un grand rôle dans la saisie et l'élaboration préalable des données en vue d'une analyse de correspondance et d'une CAH. Mais il restera nécessaire d'écrire des programmes de codage particuliers, dans un langage de programmation universel, tel que Pascal, FORTRAN, Basic...

Cependant, quelques programmes de création de tableaux suffisent à l'utilisateur non programmeur pour dépouiller par lui-même une enquête par questionnaire, et analyser un ensemble de variables hétérogènes recueillies au laboratoire, ou dans une expérimentation clinique. La présente notice est consacrée à un ensemble de tels programmes.

Le plus important de ces programmes, 'zrang', sert à ranger, observer et découper les variables d'un tableau: *ranger*, c'est-à-dire ordonner par valeurs croissantes; *observer*, en affichant des tableaux de valeurs, des histogrammes, des croisements, voire des cartes; *découper*, en fixant un ensemble de modalités et créant un tableau de numéros de modalités ou un tableau binaire.

Le programme 'zBurt' crée le tableau de Burt à partir du tableau des numéros de modalités; 'qorlsup' adjoint à une analyse factorielle un tableau externe de lignes ou de colonnes supplémentaires; 'soustab' crée un sous-tableau d'un tableau donné, en retenant des sous-ensembles spécifiés de lignes et de colonnes; 'discr1' à partir de deux tableaux donnant les coordonnées sur les axes factoriels de deux ensembles d'éléments appelés respectivement ensemble des individus et ensemble des centres, attache chaque *individu* au *centre* dont il est le plus proche; 'décale', crée à partir d'un ensemble de séries temporelles un ensemble de séries décalées...

Tous ces programmes ont servi à des études publiées dans les Cahiers de l'Analyse des Données, en sorte que les exemples ne manquent pas à l'utilisateur qui désire se livrer à des essais.

1.3 Formats de tableaux de données

Le chercheur, sociologue, biologiste, économiste, géographe..., qui aborde le codage des données, doit avoir, au préalable, analysé tels quels des tableaux de contingence: avec l'expérience, sa curiosité s'est avivée. Sans être informaticien, il a avantage à se représenter sous quelles formes diverses les tableaux sont créés, lus ou transformés par les programmes qui participent successivement à leur élaboration.

Par tableau de données, nous entendons un tableau susceptible d'être traité directement par le programme 'qori' ou, plus généralement, (notamment s'il comporte des nombres négatifs), d'être lu par la procédure 'litab' (de l'unité 'ulire') utilisée dans de nombreux programmes et notamment dans le programme 'zrang' destiné à l'observation (histogrammes, tris croisés,...) et au recodage

des données (découpage en classes, etc...). Outre le format texte de base, où les nombres sont sous la forme d'entiers longs positifs, ce qui permet d'introduire dans le texte le maximum de commentaires, on utilise actuellement 3 autres formats désignés par les suffixes 'yy', 'zz' et 'ww'.

Le format 'yy' est un format de texte: les données sont écrites au clavier comme dans le format de base, mais les nombres peuvent être sous un format quelconque: entier, réel avec point décimal ou réel avec exposant, éventuellement avec le signe négatif. Le nom du fichier des données est alors constitué d'un nom de base suivi du suffixe 'yy': par exemple le tableau 'gazole' peut sans recevoir aucune modification, être enregistré sous le nom 'gazoleyy', et accepté par 'qori', 'litab'... L'intérêt du format 'yy' est surtout qu'il permet de saisir des données quelconques, impropres à être traitées telles quelles par l'analyse des correspondances, mais destinées à être recodées par 'zrang'.

Les formats 'zz' et 'ww' sont des formats de fichiers numériques qui ne peuvent être relus directement par un éditeur de texte (mais peuvent l'être après traitement par des programmes utilitaires de mise en format de texte). Des tableaux sous ce format sont créés par divers programmes. En voici quelques exemples. Le programme 'qori', outre le listage usuel d'analyse des correspondances désigné par le suffixe cortx, crée pour chacun des ensembles i et j (ainsi que pour les ensembles supplémentaires éventuels is et js) des tableaux, 'iFacww' et 'jFacww', contenant, en bref, pour chaque individu, son sigle, la liste des premiers facteurs (en nombre fixé par l'utilisateur) la masse et le carré de la distance au centre de gravité. Dans 'CAH2', la procédure 'Facor' crée, de même, des tableaux 'iqFacww', 'jqFacww' pour les centres des classes. Mais pour l'utilisateur, il importe essentiellement de savoir que les programmes de codage tel que 'zrang', ou des programmes *ad hoc* écrits pour traiter des données reçues sous un format étranger à nos normes, ainsi que le programme 'zBurt' (qui crée un tableau de Burt à partir d'un tableau de numéros de modalités créé par 'zrang'), créent tous des tableaux sous les formats 'zz' ou 'ww'.

Il n'est pas indispensable d'expliquer en toute précision en quoi consistent les formats 'zz' et 'ww'. Nous nous bornerons à des indications sur les informations codées et le mode de codage. En bref, le tableau contient ce que contient un tableau usuel, sous forme de texte: un titre, le nombre des colonnes (ainsi que le nombre des lignes), les sigles attribués à chaque colonne; puis des enregistrements de lignes formés d'un sigle suivi de valeurs numériques qui sont soit des entiers longs (de signe quelconque) pour le format 'zz' (allemand 'zahl') soit des réels courts pour le format 'ww' (allemand 'wirklich').

Quant au codage, la procédure 'litab', ou tout autre procédure lisant un tableau en format 'zz' ou 'ww', considère ce tableau comme un tableau d'entiers, c'est-à-dire, en somme, comme une suite de paires d'octets quelconques; mais, grâce à des pointeurs, ces octets sont relus soit comme des

caractères soit comme des quantités: 3 entiers, ou 6 octets, pour un sigle; 2 entiers, ou 4 octets, pour un entier long ou un réel court; etc...

Lorsqu'on donne un nom de base à la procédure 'litab' (ou à un programme, tel que 'gori', utilisant celle-ci), la procédure cherche d'abord un tableau sous ce nom; puis, à défaut, cherche successivement un tableau avec les suffixes 'yy', 'zz' ou 'ww'; si la recherche est infructueuse, s'affiche un commentaire d'erreur. Mais certains programmes, tels que 'soustab', n'acceptent pas de tableau en format de texte, mais seulement en 'zz' ou 'ww': ces tableaux sont en effets plus faciles à lire. La restriction n'est toutefois pas essentielle, car 'zrang' offre une option C (copier: §2.3.2) qui crée de tout tableau, quel qu'en soit le format initial, une copie en 'ww'.

2 Le programme 'zrang', pour ranger, observer, découper les données

2.0 Accès au programme 'zrang'

Le programme demande premièrement à l'utilisateur le nom du tableau de base; ce nom doit être donné sans suffixe: par exemple: 'disq:gazole', même s'il n'existe pas de fichier nommé 'disq:gazole' mais seulement un fichier 'disq:gazolezz'. Le programme cherche d'abord un fichier dont le nom est sans suffixe; puis, à défaut, tente successivement d'ajouter les suffixes 'yy', 'zz' et 'ww', comme le fait le programme d'analyse des correspondances, lequel utilise, comme 'zrang', la procédure de lecture 'litab'. Sauf erreur de nom, 'zrang' lit le tableau des données, ligne après ligne, en l'affichant à l'écran, sous forme arrondie, en entiers.

On offre alors à l'utilisateur plusieurs options, (comportant éventuellement des variantes): T (Transposer), C (Copier), R (Ranger), D (Découper en classes), Z (observer sans créer de tableaux). Nous commencerons par l'option 'Z', à laquelle l'utilisateur consciencieux est généralement heureux de recourir pour se familiariser avec les données sans créer inutilement des fichiers et encore moins des listages de papiers! Nous passerons ensuite à 'D', outil indispensable de l'analyse des questionnaires et du codage. Nous terminerons sur les utilitaires, 'T', 'C', 'R'.

2.1 'Z': observer les données

En entrant dans l'option 'Z', l'utilisateur doit répondre à la question:

'les individus i sont-ils des unités territoriales O ou N '

Nous suposerons d'abord que la réponse est 'Non', réservant à une note (§2.1.4) la production de cartes associées aux histogrammes.

Les variables peuvent être affichées dans un ordre quelconque, autant de fois qu'on le désire, avec, éventuellement un croisement entre la variable qu'on

termine d'afficher et celle affichée immédiatement avant celle-ci. C'est pourquoi l'on voit d'abord sur l'écran:

'ci dessous sont rappelés les num et sigles des variables'

(Le nombre des variables pouvant aller jusqu'à 300 (dans la version actuelle) il se peut que la liste ne tienne pas sur l'écran: l'affichage se fait alors en deux listes, l'utilisateur passant de l'une à l'autre en tapant '*', ou tout autre caractère, selon la consigne: 'pour afficher la suite des sigles entrer *').

Vient alors la question:

'le numéro de la variable à afficher est '

et, après qu'on a choisi, la mention:

'on affichera successivement, pour la variable... le tableau ordonné des valeurs, puis des histogrammes'

Nous décrirons successivement le tableau des valeurs rangées, les histogrammes et le croisement.

2.1.1 Le tableau ordonné des valeurs

Il faut afficher en une ou plusieurs colonnes, depuis la plus petite jusqu'à la plus grande, et en les répétant s'il y a lieu, les valeurs prises, sur l'ensemble des individus, par la variable choisie; en d'autres termes, afficher l'ensemble des valeurs constituant une colonne.

Avec un tableau de 100 lignes, le tri semble instantané, le tableau s'affiche sans délai apparent; avec 1000 individus, on attend 1s.

Les nombres sont affichés en format flottant, avec un exposant: par exemple '1.00e+0' pour l'entier 1. Ce format n'est pas adapté aux valeurs entières petites, notamment aux numéros de modalités; mais il a l'avantage d'être universel: '5.73e-6' pouvant exprimer utilement une concentration, etc...; et on n'a pas voulu imposer à l'utilisateur de répondre à une question de plus pour fixer le format d'affichage!

L'écran du Macintosh ne permet pas d'afficher ensemble plus de 200 valeurs; il ne nous a pas paru opportun d'afficher en pages successives, dans la mesure où le but principal est de prendre une vue d'ensemble de l'étalement des données. Donc, si on ne peut pas afficher toutes les valeurs, on affiche les valeurs de 2 en 2, ou de 3 en 3...

L'utilisateur sait à quel rang correspond chaque valeur, additionnant les nombres entiers écrits en marge gauche et en marge supérieure; par exemple, à l'intersection de la ligne commençant par le nombre 17 et de la colonne commençant par 80, se trouve la valeur dont le rang est 97.

2.1.2 Les histogrammes

Un programme créant des histogrammes doit être conversationnel ou ne pas être. Il suffit d'un individu isolé très écarté pour contraindre tous les autres à s'enfermer dans un seul créneau. D'autre part, il vaut la peine de connaître les noms des individus rentrant dans chaque créneau. On s'est efforcé d'utiliser au mieux la place disponible.

Vient d'abord la question:

'nombre de créneaux (de 1 à 25) = '

On se limite à 25 créneaux afin de pouvoir écrire, en haut de chacun de ceux-ci, le rang de la plus forte valeur qui y est comprise (information essentielle pour fixer un découpage en classes).

Il n'est pas vain de demander un seul créneau, parce que, dans ce cas, on peut lire dans ce créneau qui n'est autre qu'un tableau rectangulaire emplissant l'écran, les sigles des individus triés suivant la variable considérée; avec le plus fort en haut de la dernière colonne à droite, ceux qui le suivent immédiatement en dessous de lui; et le reste sur les colonnes précédentes.

En demandant 18 créneaux, on a des créneaux assez larges pour écrire dans chacun de ceux-ci une colonne de sigles; ce qui est utile, même si la place manque pour écrire tous les individus de chaque créneau. Avec 9 créneaux, on a deux colonnes de sigles par créneau... Le lecteur expérimentera.

Vient alors le dialogue de cadrage de l'histogramme:

'on rangera les valeurs comprises entre deux rangs ia et ib'
'le rang ia de la première valeur prise est '
'le rang ib de la dernière valeur prise est '

Dans un premier essai, on peut demander l'affichage de toutes les valeurs (ce que permet le raccourci: ia=ib=0); puis, d'après les rangs écrits en haut des créneaux, fixer la longueur de la tête ou de la queue qu'on désire éliminer.

On peut tracer successivement autant d'histogrammes qu'on le veut; et éventuellement, demander un nouvel affichage du tableau des valeurs triées, pour tracer encore des histogrammes. Quand, finalement, (en entrant 'N'), on répond 'Non' à la question:

'faut il réafficher la variable... O ou N '

on entre dans le croisement avec la variable précédente (à condition qu'on n'en soit pas à la première variable affichée!)

2.1.3 Croisement de deux variables

Si l'on a répondu 'Oui', (c'est-à-dire 'O'), à la question 'faut il croiser...', apparaît la mention:

'NB on affichera successivement un plan et un tableau'
'pour appeler ou quitter le plan, entrer * '

Ce plan ne diffère guère d'un graphique usuel d'analyse factorielle produit par le programme 'planF': les individus sont représentés par leurs sigles, avec pour ordonnée la variable qu'on vient d'afficher et pour abscisse la variable précédente. Il faut préciser que le plan est cadré en se bornant aux individus dont les rangs sont compris dans l'intervalle (ia, ib) du dernier histogramme affiché pour chacune des deux variables concernées; autrement dit, les queues écartées comme nuisibles au bon étalement des histogrammes sont aussi éliminées tacitement du plan, pour la même raison.

Le plan est d'une lecture agréable s'il s'agit de variables continues; avec des variables discrètes, on a une accumulation de sigles écrasés aux points dont les abscisses et ordonnées correspondent aux numéros des modalités; on peut seulement déceler les combinaisons absentes ou rares...

Sous le plan on lit:

'pour quitter le plan croisant ... (axe horiz) et ... (axe vertic) entrer * '

En entrant un caractère, on voit s'afficher un tableau usuel de tri croisé: la seule contrainte est le manque de place, sensible notamment dans l'écriture des bornes supérieures des classes. Il n'y a pas place pour plus de 14 colonnes; le nombre des lignes pouvant aller jusqu'à 18.

De façon précise, les bornes des classes du tri sont celles-là mêmes des créneaux du dernier histogramme affiché. Si on avait éliminé une tête ou une queue, celle-ci se retrouve comme une classe dans le tableau; (respectivement en première et dernière ligne ou colonne, selon qu'il s'agit de la variable ordonnée ou de la variable abscisse). C'est pourquoi on demandera 12 créneaux dans l'histogramme, afin de ne pas dépasser 14, tête et queue comprises.

2.1.4 Note: Cartographie et histogrammes

Si l'on a répondu 'Oui' (i.e. 'O') à la question

'les individus i sont ils des unités territoriales O ou N '

on doit satisfaire à deux demandes

'le nom du disque utilisé est '
'le nom du pays est '

On peut donner le nom du disque avec celui d'un dossier, 'ddur:atlas', où est contenu le fichier dont le nom est suffixé en 'reg', fichier qui sert pour tracer la carte du pays subdivisé en unités territoriales (ou de la ville subdivisée en arrondissements ou quartiers...): e.g. pour la France on aura le fichier 'ddur:atlas:Francereg'. La création et la structure de ce fichier font l'objet d'une partie de la notice du programme 'carthage' et de ses annexes. Nous n'en dirons rien ici.

Le programme 'zrang' vérifie que le fichier nécessaire existe, et que le nombre des unités territoriales coïncide avec celui des individus du tableau principal. Si la vérification échoue, 'zrang' annule tacitement la demande de cartes et se borne à afficher des tableaux, histogrammes et croisements.

Une carte s'affiche après chaque histogramme dont elle constitue la traduction. Les unités territoriales de chaque créneau reçoivent une même trame, d'autant plus foncée que la valeur de la variable est plus élevée. S'il y a lieu, la tête (précédant l'histogramme: rang<ia) est en blanc; et la queue (rang>ib) est en noir. Comme le nombre des trames de gris est limité à 9, le programme calcule pour utiliser au mieux l'étendue de sa palette, quel que soit le nombre des créneaux.

La taille de l'écran étant réduite, la carte s'affiche par moitiés, suivant les instructions ci-dessous:

'pour finir l'affichage de la carte de... entrer la lettre N'

'pour afficher la carte ou passer alternativement
du Nord au Sud, entrer une autre lettre '

2.2 'D': découper en classes

Cette option crée soit un tableau de numéros de modalités, soit un tableau éclaté binaire en (0,1); le tableau a pour nom le nom de base suivi respectivement du suffixe 'Mzz' ou 'Bbzz'; le format est donc 'zz' (cf. §1.3).

2.2.1 Dialogue d'entrée dans 'D'

Si l'on a pris l'option 'D' de 'zrang', s'affichent deux questions

'faut il découper les variables
avec des bornes préétablies(P), ou non(N) '

'faut il créer un tableau booléen(B)
ou un tableau des numéros des modalités(M) '

Il est facile d'expliquer l'intérêt de la deuxième question: nous avons dit que pour construire rapidement le tableau de Burt, (cf. §3), on se sert du tableau donnant à l'intersection de la ligne i et de la colonne j (une colonne par variable initiale) le numéro de la modalité de j dans laquelle rentre l'individu i; tandis que

le tableau éclaté en (0,1) est adjoint en supplémentaire à l'analyse du tableau de Burt, (cf. §4); (ou peut éventuellement être analysé tel quel).

Quant à la première question, nous dirons qu'en général l'utilisateur préfère découper les variables continues en mode conversationnel, (d'après des tableaux et histogrammes que l'option 'D' affiche comme le fait l'option 'Z'); mais que si le tableau des données primaires ne comprend que des numéros de modalités de réponse, le découpage s'impose sans discussion, l'examen des histogrammes servant seulement à découvrir les modalités vides. D'autre part, après une première analyse, on peut vouloir modifier des détails du découper, sans reprendre le dialogue; ou encore utiliser pour un nouveau jeu de données des bornes déjà choisies pour un fichier de base. Voilà l'intérêt de bornes préétablies.

Si l'on répond 'P', (bornes préétablies), le programme cherche s'il existe bien un fichier de bornes, (fichier désigné par le nom de base suivi du suffixe 'Dcodx'); et il ouvre ce fichier pour vérifier l'exactitude du nombre des variables et des sigles de celles-ci; pour conclure, si tout va bien, en affichant:

'le fichier Dcodx semble correct'

Il faut prendre garde que la vérification n'est pas superflue, car l'utilisateur est sujet à bien des erreurs quand il prépare un tel fichier. En fait, on part souvent d'un fichier créé lors d'un premier découpage en mode conversationnel, et les risques d'erreur sont moindres que si le fichier est créé de toutes pièces. Nous considérerons au §2.2.2 le dialogue de découpage; et au §2.2.3 le fichier Dcodx.

2.2.2 Le dialogue de découpage des variables en classes

Dans ce dialogue, les variables sont traitées l'une après l'autre dans l'ordre des colonnes du tableau des données. Pour chaque variable, le découpage proprement dit est précédé d'une phase d'observation fondée sur les mêmes procédures d'affichage que dans l'option 'Z' (observation sans créer de tableau). Ceci nous permettra de renvoyer le lecteur aux §§2.1.1 et 2.1.2.

Plaçons-nous au début du dialogue. Sans demander le numéro de la variable à afficher, le programme affiche la première variable (ou colonne 1); et une fois celle-ci découpée, la deuxième; et ainsi de suite.

Comme au §2.1, on a, après le tableau ordonné des valeurs, un dialogue d'affichage d'histogrammes qui se poursuit jusqu'à ce que l'utilisateur réponde 'Non' à la question:

'faut il tracer un autre histogramme(H) pour... ou non(N) '

On entre alors dans le dialogue de découpage proprement dit.

Vient d'abord la question:

'le nombre de modalités (de 2 a 10) pour..., sera '

En fait, l'utilisateur peut répondre 0 ('zéro') s'il ne désire pas découper la variable; ce nombre est même imposé par le programme si l'on est dans l'option 'B' (créer un tableau en 0,1) et que le nombre total des modalités des variables déjà découpées atteint le nombre maximum de colonnes admis pour un tableau (soit 300, dans la présente version).

On a ensuite pour chaque modalité à répondre à deux questions:

'le sigle choisi pour la modalité n° ... est '

'le RANG de la plus forte valeur de... est '

(cette deuxième question manque toutefois pour la dernière modalité qui s'étend tacitement jusqu'à englober l'individu réalisant le maximum de la variable considérée). Le choix des sigles est libre: on peut utiliser de 1 à 4 signes, lettres ou chiffres.

On notera qu'à la différence de ce qui est le cas dans le tableau 'Dcodx' des bornes, (cf. infra, §2.2.3), les limites des classes sont ici spécifiées par RANG et non par valeur. Il est en effet plus simple d'écrire un nombre entier qu'une valeur de format quelconque; et le programme procède, en fait, comme si on lui avait donné une borne supérieure (égale à la valeur prise par la variable pour l'individu dont on lui a donné le rang); en ce sens que si, au delà du rang fixé, se trouvent des individus pour lesquels la variable considérée prend la même valeur, ceux-ci sont englobés dans la classe.

Quand on a fini de spécifier les modalités, apparaît la question:

'faut il réafficher la variable... O ou N '

Si l'on répond 'Oui', le découpage de la variable est tenu pour nul, et le dialogue complet recommence, à partir de l'affichage du tableau des valeurs triées. Si l'on répond 'Non', on passe à la variable suivante (s'il y a lieu).

2.2.3 Le fichier 'Dcodx' des bornes de découpage des variables

Nous avons dit que le dialogue de découpage implique la création par le programme d'un fichier 'Dcodx', de format texte, i.e. lisible à l'écran par un éditeur de textes. Nous considérerons donc un exemple de fichier 'Dcodx' tel qu'il est ainsi créé, en indiquant les éléments essentiels de sa structure, qui doivent être respectés par l'utilisateur; soit que celui-ci modifie 'Dcodx', soit qu'il le crée de toutes pièces.

Le texte commence par deux lignes de titre, dont le contenu est indifférent, mais qui doivent obligatoirement exister et se terminer par un 'aller à la ligne'.

Variations relatives des variables hémodynam. et des concentrations
 B:nx:xTn*Dcodx: bornsup pour le découpage des variables
 le nombre des variables est 5

IC a 3 modalités dont les sigles et bornes sup sont			
IC-	IC=	IC+	
4.848484993e-1	5.208333135e-1	6.129032373e-1	
pcp a 3 modalités dont les sigles et bornes sup sont			
pcp-	pcp=	pcp+	
4.615384638e-1	5.000000000e-1	7.058823705e-1	
SVR a 3 modalités dont les sigles et bornes sup sont			
SVR-	SVR=	SVR+	
4.779411852e-1	5.236220360e-1	5.913978219e-1	
TE a 4 modalités dont les sigles et bornes sup sont			
TE-	TE=	TE+	TE++
4.140969217e-1	5.221675038e-1	5.888888836e-1	8.658536673e-1
TS a 4 modalités dont les sigles et bornes sup sont			
TS--	TS-	TS=	TS+
4.352159500e-1	4.747953415e-1	5.398482084e-1	9.181173444e-1

§2.2.3: Exemple de tableau 'Dcodx'

Vient ensuite la phrase:

'le nombre des variables est 5'

où seul compte le nombre (ici 5), mais qui doit, elle aussi, se terminer par aller à la ligne.

Suivent des alinéas de format identique propres chacun à une variable. D'abord une ligne telle que:

'IC a 3 modalités dont les sigles et bornes sup sont'

Il faut obligatoirement commencer par le sigle de la variable suivi d'un blanc; écrire ensuite le nombre de modalités et terminer par 'aller à la ligne'. Quant au texte d'accompagnement il n'est assujéti qu'à deux conditions essentielles:

ne comporter que des lettres (à l'exclusion de chiffres)
 comporter au moins un mot (voire une lettre) après que le nombre des modalités a été écrit.

Par exemple on peut écrire en abrégé 'IC 3 mod'; ou, en espagnol, 'IC tiene 4 m'; etc...

Après cette première ligne, on écrit les sigles puis les valeurs des bornes, sans inclure aucun texte; le format des nombres est entièrement libre: -15 convient aussi bien que -15.000, ou -.15e+2; on écrit indifféremment sur une ou plusieurs lignes; on termine l'alinéa en allant à la ligne. Si le nombre de modalités spécifié est 0 (zéro), c'est-à-dire si on a éliminé la variable du découpage, il n'y a évidemment ni sigles ni valeurs à écrire: on se contente de passer une ligne blanche.

La nécessité de toutes ces contraintes, d'ailleurs légères, se comprend si l'on considère que le programme exécute successivement des ordres de lecture soit de nombre, soit de mot (sigle), soit de chaîne de caractères jusqu'à une fin de ligne.

NB Si l'on est dans l'option 'B', (créer un tableau en 0,1), et que le nombre total des modalités demandées dans 'Dcodx' dépasse le maximum permis, (300 dans la présente version), le programme supprime tacitement les modalités surnuméraires de la dernière variable qu'on peut découper, et attribue 0 modalité aux variables suivantes.

2.3 Les options 'T', 'C', 'R' du programme 'zrang'

L'utilisation de ces options étant simple, nous nous bornerons à en expliquer l'effet et l'utilité

2.3.1 'T': transposer le tableau donné

À partir d'un tableau 'tab' ayant cari lignes et carj colonnes, cette option crée un tableau transposé ayant carj lignes et cari colonnes. Le tableau est créé même si cari dépasse la valeur maxima (300) permise pour le nombre de colonnes d'un tableau destiné à être soumis aux programmes 'zrang', 'qori', ou tout autre programme lisant le tableau par la procédure 'litab' pour le loger en mémoire centrale; avant d'entrer dans un tel programme, il faut donc supprimer des colonnes, ce que permet le programme 'soustab' (cf. §5).

Le tableau transposé est créé avec pour nom le nom de base suivi du suffixe 'Tww'; c'est donc un fichier numérique en format réel, quel que soit le format du tableau de départ.

Il est utile de transposer un tableau quand, les données initiales étant hétérogènes, on préfère les saisir colonne après colonne, plutôt que ligne par ligne.

Compte tenu de la souplesse de la procédure de lecture 'litab' qui, une fois la ligne de titre écrite, laisse l'utilisateur libre d'aller à la ligne quand il le juge bon, entrer les données colonne par colonne permet de disposer les nombres en blocs rectangulaires sur lesquels la vérification peut être facilitée.

2.3.2 'C': copier le tableau donné

À partir d'un tableau 'tab', cette option crée un tableau contenant les mêmes données, semblablement disposées (mêmes nombres de lignes et de colonnes), mais dans le format 'ww' (réels). Le nom du tableau créé n'est autre que le nom de base, suivi du suffixe 'ww'.

Si le tableau initial était déjà en format 'ww', l'effet du programme est nul: après avoir lu un tableau, on le récrit à la même place, sous le même format et le même nom.

Cette option sert à recopier en format 'ww' un tableau de données saisi par éditeur de texte, (nom sans suffixe, ou avec le suffixe 'yy'). Ainsi, le tableau copié peut être soumis au programme 'soustab', pour en créer un ou plusieurs sous-tableaux.

2.3.3 'R': créer un tableau numérique des rangs

Quant à l'affichage des variables avec tableau des valeurs triées et histogrammes, l'option 'R' ne diffère pas de l'option 'D'. Quant au tableau créé, elle se rapproche le plus de la variante 'M' de 'D', en ce sens que le tableau créé a même nombre de colonnes que le tableau de base; mais le contenu est différent: au lieu d'un numéro de modalité, (dépendant d'un découpage, spécifié par 'Dcodx' ou en dialogue), on inscrit, à l'intersection de la ligne i et de la colonne j , le rang qu'occupe la valeur initiale $k(i,j)$ dans l'ensemble des nombres de la colonne j , triés par valeurs croissantes.

(On signalera au passage que, dans un tableau de numéros de modalité, 'Mzz', le programme 'zrang' inscrit des rangs dans la colonne j , si on a demandé de découper en zéro modalité la variable j . En ce cas, du tableau éclaté en (0,1), la variable j disparaît sans laisser de trace).

Comme il est naturel le tableau des rangs est créé, avec le suffixe 'Rzz', en format entier.

Dans la pratique, l'option 'R' pourrait être une étape intermédiaire dans la création d'un tableau de rangs dédoublés, ou dans un codage plus complexe à base de rangs.

3 Le programme 'zBurt' de création du tableau de Burt

À partir d'un tableau 'Mzz', qui lui fournit, pour les individus successifs, les numéros des modalités de chaque variable, et du fichier 'Dcodx', qui lui fournit les sigles des modalités, le programme 'zBurt' crée un tableau de Burt, ou tableau de cooccurrence des modalités ($k(m,m)$ = nombre des individus rentrant, à la fois, dans la modalité m et la modalité m').

Le programme 'zBurt' vérifie l'existence des deux fichiers 'Mzz' et 'Dcodx' qui lui sont nécessaires. Il s'assure que les fichiers s'accordent quant au nombre total des variables et aux sigles de celles-ci. De plus, si on a procédé à plusieurs découpages successifs et que les fichiers 'Mzz' et 'Dcodx' utilisés ont été créés lors de deux essais différents, il se peut que le tableau 'Mzz' contienne des numéros de modalité excédant le nombre de modalités inscrit dans 'Dcodx' pour la variable correspondante. Dès qu'il rencontre une telle discordance, 'zrang' affiche un commentaire d'erreur:

'ERREUR la variable ... a ... modalités et pour ... le numéro est ...'

(i.e. en précisant de quelle variable et de quel individu il s'agit); et le tableau de Burt n'est pas créé.

Les individus sont lus un par un, sans que le tableau initial soit gardé en mémoire centrale, où seul est gardé le tableau de Burt en cours d'élaboration. S'il y a carj variables retenues, (avec un nombre non nul de modalités), cela fait, pour chaque individu, $\text{carj} * (\text{carj}+1)/2$ cooccurrences, que l'on peut prendre en compte par une double boucle en carj , sans effectuer de multiplications, (et non avec une double boucle en carm , nombre total des modalités, comme on devrait le faire si le tableau de Burt était créé à partir du tableau éclaté 'Bbzz'). Au fur et à mesure de la lecture des individus, s'affichent à l'écran les sigles de ceux-ci; ce qui permet à l'utilisateur de vérifier le bon déroulement du programme.

Le tableau de Burt est créé en format entier, avec le suffixe 'Bzz' ajouté au nom du tableau de base. Ainsi, pour le programme 'qorlsup', le tableau éclaté 'Bbzz' peut constituer un tableau de colonnes supplémentaires pour le tableau de Burt, le sigle de l'ensemble étant 'b': cf. *infra*, §4.

4 Le programme 'qorlsup' pour adjoindre des tableaux externes de lignes et colonnes supplémentaires

4.1 Fonction du programme 'qorlsup'

En analyse des correspondances, on appelle élément supplémentaire une ligne ou colonne dont on ne tient pas compte pour déterminer les axes factoriels mais qui est projetée sur ceux-ci. Le programme 'qori' permet de mettre en supplémentaire un sous-ensemble de lignes ou de colonnes du tableau soumis à l'analyse.

Il peut s'agir d'éléments apportant à un axe une contribution telle que l'analyse en est perturbée; ou encore d'éléments pour lesquels certaines données sont absentes ou imprécises. Souvent, il est préférable de faire totalement disparaître ces éléments, comme permet de le faire le programme 'soustab' (cf. §5).

Parfois, on désire projeter en supplémentaire des éléments qui n'appartiennent pas au tableau de base analysé, mais proviennent (par exemple) d'une expérience identique faite sur un autre groupe d'individus; ou d'une expérience différente faite sur les mêmes individus. Si (comme il est d'usage) on place en lignes les individus et en colonne les variables, on aura, dans le premier cas, un tableau externe de lignes supplémentaires; et, dans le deuxième cas, un tableau externe de colonnes supplémentaires. Nous disons *tableau externe* parce qu'il ne s'agit pas d'un sous-tableau du tableau de base soumis à l'analyse.

Le programme 'qorlsup' sert à créer, pour un ou plusieurs tableaux supplémentaires externes, (de lignes ou de colonnes), un fichier numérique de facteurs (fichier 'Facww') et un listage usuel (fichier 'corsutx').

4.2 Déroulement du programme 'qorlsup'

Le programme 'qorlsup' commence par demander le nom du tableau de base, qui doit déjà avoir été soumis à l'analyse des correspondances; et qorlsup vérifie l'existence de quatre fichiers: 'iFacww', 'jFacww', 'ipr' et 'jpr'. Puisque l'adjonction des éléments supplémentaires se fait en utilisant la formule de transition, l'utilité des fichiers numériques de facteurs est évidente: 'iFacww' et 'jFacww' servent respectivement à adjoindre des colonnes et des lignes supplémentaires. Mais les fichiers numériques 'ipr' et 'jpr', qui donnent la liste des lignes et colonnes retenues pour principales dans l'analyse du tableau de base, sont également indispensables; car, en bref, dans la formule de transition, la sommation porte sur les seuls éléments principaux.

Le programme demande ensuite le nombre (de 1 à 20) de tableaux à mettre en supplémentaire. Ces tableaux doivent être dans l'un des formats usuels (présentés au §1.3); et leur nom est celui du tableau de base suivi d'un sigle (de 1 à 3 lettres) ainsi que du suffixe éventuel 'yy', 'zz' ou 'ww' indiquant le format. Dans la version actuelle, le sigle doit obligatoirement se terminer par 'a' pour un tableau de lignes supplémentaires, et par 'b' pour un tableau de colonnes supplémentaires.

Il importe de préciser qu'un tableau de colonnes supplémentaires doit, en fait, être présenté transposé, avec un nombre de colonnes égal au nombre des lignes du tableau de base. Cette disposition permet d'abord de vérifier, en ouvrant le tableau supplémentaire, si la liste des sigles de ses colonnes est bien identique à celle des sigles des lignes (individus) du tableau principal, (ces derniers sigles nous étant connus par le fichier 'iFacww'). Elle permet ensuite de lire les unes après les autres les colonnes supplémentaires et de leur appliquer immédiatement la formule de transition.

La remarque précédente suffit à faire entendre que 'qorlsup' garde en mémoire centrale les facteurs et les sigles afférents au tableau de base (plus précisément à ses éléments principaux); tandis que les tableaux d'éléments supplémentaires sont traités élément par élément; les fichiers 'Facww' et 'corsutx' étant ouvert en écriture en même temps que le fichier du tableau des éléments supplémentaires est ouvert en lecture.

Après le dialogue initial, le programme 'qorlsup' procède à la lecture des sigles et facteurs afférents au tableau de base. Puis il passe au traitement des tableaux d'éléments supplémentaires. Les facteurs calculés pour ceux-ci s'affichent à l'écran, au fur et à mesure qu'ils s'inscrivent sur fichier, comme c'est le cas avec le programme 'qori'.

4.3 Exercice pour utiliser 'qorlsup'

Soit un tableau de données, que, pour fixer les notations, nous désignerons par 'disq:tabyy'. Créons une copie de ce tableau à laquelle nous donnons pour

nom 'disq:tab2ayy'; et créons, par l'option 'T' de 'zrang', un tableau transposé, dont le nom est 'disq:tabTww', mais que nous renommerons 'disq:tabTbww'. On peut utiliser le programme 'qorlsup' en prenant pour tableau de base 'disq:tab' (dont le format est 'yy'); et donnant deux sigles de tableaux supplémentaires: '2a' et 'Tb'; le premier étant un tableau de lignes supplémentaires et le second, un tableau de colonnes supplémentaires.

Il est clair que le listage 'disq:tab2acorsutx' doit contenir les mêmes résultats que le tableau du listage 'disq:tabcortx' afférent à l'ensemble des i; tandis que le listage 'disq:tabTbcorsutx' s'identifie au tableau des j du listage 'disq:tabcortx'. En effet, les lignes et colonnes supplémentaires des tableaux externes que nous avons créés par copie ou transposition, sont identiques aux lignes et colonnes du tableau de base 'disq:tabyy'.

Quant aux tableaux de facteurs, il importe de préciser qu'ils sont créés avec pour noms:

'disq:tab2iaFacww' et: 'disq:tabTjbFacww',

où l'on a inséré respectivement, dans les sigles des ensembles, la lettre 'i' avant 'a', et la lettre 'j' avant 'b'; ceci afin de se protéger contre des confusions possibles entre noms de fichiers.

5 Le programme 'soustab' de création de sous-tableaux

Partant d'un tableau de base en format 'zz' ou 'ww', le programme a pour fonction de créer un sous-tableau de même format, en retenant les sous-ensembles de lignes et de colonnes qui sont spécifiés par l'utilisateur.

5.1 Utilisation du programme 'soustab'

Le programme demande premièrement le nom du tableau de base: ce nom doit lui être donné sans suffixe; soustab cherche d'abord un tableau en 'zz' et, s'il n'en trouve pas, un tableau en 'ww'. Le programme lit le tableau afin de garder en mémoire les sigles des colonnes et des lignes, sigles qui s'affichent au fur et à mesure de la lecture.

En vue de créer un sous-tableau, le programme demande alors à l'utilisateur d'attribuer à celui-ci un sigle. Si, par exemple, le nom de base est 'B:phos', le format 'ww' et le sigle '3A', le sous-tableau créé aura pour nom 'B:phos3Aww' et sera en format 'ww'. Il y a toutefois à cela une exception: celle des tableaux de facteurs, dont le nom se termine par 'Facww'. Pour de tels tableaux, le programme constate la présence de 'Fac', et introduit le sigle avant cette désinence. Par exemple, si le tableau de base est 'B:phosiFacww' et le sigle est '2', le tableau créé sera 'B:phosi2Facww'. Moyennant quelques précautions, dont nous ne parlerons pas tout de suite, on peut ainsi créer des sous-fichiers permettant de faire une classification d'après les facteurs 3, 4 et 5 sans tenir compte des facteurs 1 et 2; ou de ne placer sur un graphique (créé par 'planF'

ou 'planX') que certains individus constituant un sous-ensemble appelé 'i2', etc...

Le nom du sous-tableau étant donné, reste à spécifier les sous-ensembles de lignes et colonnes que l'on retient. Pour cela deux voies s'offrent: le dialogue à l'écran, tout analogue à celui du choix des éléments supplémentaires dans 'qori' (puisqu'il utilise la même procédure 'ensembler' avec affichage des sigles et choix des éléments un par un ou par blocs); ou la donnée d'un ensemble par fichier. De façon précise, avant d'interroger l'utilisateur sur le choix des éléments j (colonnes) à garder, 'soustab' cherche s'il existe un fichier ayant pour nom le nom de base suivi du suffixe et de la désinence 'ensi': par exemple, 'B:phos3Aensi'.

Ce fichier, qui est un texte, doit contenir, après une ligne de titre obligatoirement terminée par un aller à la ligne, la suite des numéros des individus à retenir: cette suite pouvant être dans un ordre quelconque, avec autant d'aller à la ligne qu'on voudra, mais sans aucune adjonction de texte. Il est particulièrement commode de créer un tel fichier 'ensj' (ou 'ensi') si la liste des numéros des individus existe déjà, étant celle d'une classe de partition extraite d'une CAH. Même si la liste n'existe pas toute faite, il est souvent plus facile de créer le fichier 'ensi', que d'entrer les numéros par dialogue.

Après avoir créé un sous-tableau, (en relisant et recopiant partiellement le tableau de base), le programme demande si l'on doit en créer un autre du même tableau; et, si la réponse est négative, demande s'il y a lieu de relire un autre tableau de base pour en créer des sous-tableaux.

Sans décrire l'algorithme de 'soustab' disons qu'il s'exécute rapidement, parce qu'il lit les données *numériques* du tableau de base sans les afficher ni même les traduire, se contentant de tenir compte de ce qu'une donnée, réelle ou entier long occupe 4 octets; et qu'il ne requiert que peu de place en mémoire, parce que les seules données interprétées et gardées sont les données alphanumériques: titre et sigles; les données numériques étant seulement lues pour être sautées ou copiées sur le sous-tableau en cours de création.

5.2 Cas des tableaux de facteurs

On a plusieurs fois fait allusion aux tableaux 'iFacww', 'isFacww', 'iqFacww', 'inFacww'... (individus principaux, supplémentaires, centres des classes d'une partition, nœuds de la sous-hiérarchie retenue...) Dans la mesure du possible, afin de pouvoir les examiner par le programme 'zrang', et aussi, précisément, en extraire des sous-tableaux par 'soustab', nous nous sommes efforcés de donner aux tableaux de facteurs le même format qu'aux tableaux de données en 'ww'.

Mais comme il est indispensable de garder les valeurs propres et la trace, on a eu recours à un artifice: dans la place de 255 octets prévue pour le titre selon

l'usage du Pascal, on a logé, après le titre limité à 127 caractères, les informations relatives à un maximum de 29 facteurs. Dans chaque ligne, ou enregistrement individuel, figurent après le sigle de l'individu et les facteurs proprement dits (désignés par les sigles de colonnes 'axe1', 'axe2', ...) la masse et le carré de la distance à l'origine (notés 'poid' et 'dis2'; comme on peut s'en assurer en relisant un fichier 'Facww' par 'zrang').

Il serait d'ailleurs facile d'assouplir le format de tableaux 'zz' et 'ww', en spécifiant au début du fichier la longueur d'une en-tête, contenant outre le titre, toutes informations jugées opportunes. Matériellement en effet, en mémoire centrale comme sur le support magnétique (ou optique) les données représentent des chiffres binaires groupés en octets. Le bon usage des pointeurs permet à un programme Pascal de décoder ces octets sous un format quelconque, sans qu'il soit même nécessaire de recourir à la définition de types; et, en tout cas, le format de transfert universel des données de toute nature sémantique (les textes mis à part) est : 'fichier d'entier'!

Si (comme on l'a suggéré plus haut) on désire faire une CAH d'après les facteurs 3, 4 et 5, il faut garder les 5 colonnes 'axe3', 'axe4', 'axe5', 'poid' et 'dis2'. Il est facile de spécifier par blocs les colonnes conservées. Quant aux individus, il importe de signaler que, si l'on crée un fichier 'ensi', la place du sigle dans le nom de ce fichier est la même que dans le nom du tableau à créer, c'est-à-dire, immédiatement avant 'Fac'; soit, dans l'exemple ci-dessus, 'B:phosi2Facensi'.

6 Le programme 'discri' d'affectation d'un ensemble d'individus à un ensemble de centres

En bref, le programme 'discri' part de deux fichiers numériques de facteurs, le fichier des centres et le fichier des individus; et il affecte chaque individu au centre dont il est le plus proche. Dans la mesure où chaque individu est censé relever d'une classe, il s'agit de ce qu'on appelle classiquement une procédure de discrimination; d'où le nom du programme. Afin de présenter le déroulement du programme et d'en montrer l'effet, nous partons d'un exemple typique. L'ensemble des individus est l'ensemble des *i* usuel; et les centres sont les *i*_q, centres de gravité des classes d'une partition des *i* obtenue par CAH.

Dans la pratique, 'discri' servira particulièrement à affecter à des centres, résultant de l'analyse d'un premier ensemble de données, un ensemble d'individus adjoints en supplémentaire à cette analyse. D'où le lien de 'discri' avec 'qorlsup' (§4).

6.1 Déroulement du programme 'discri'

S'affiche d'abord la phrase de commentaire:

'ce programme affecte un ensemble d'individus à un ensemble de centres'

puis les questions:

'le nom de base des fichiers est '
 'le sigle de l'ensemble des centres est '
 'le sigle de l'ensemble des individus est '

auxquelles nous répondons respectivement: 'disq:tab', 'iq' et 'i'. Le programme vérifie alors qu'existent les deux fichiers de facteurs correspondant à ces ensembles, soit: 'disq:tabiqFacww' et 'disq:tabiFacww'; et il s'assure que les fichiers sont compatibles, en ce sens qu'ils comportent le même nombre de facteurs.

Après avoir donné le nombre des facteurs disponibles par la phrase

'nombre de facteurs disponibles =...'

'discri' s'enquiert du nombre de facteurs à utiliser (si, e.g., la réponse est 4, on procédera dans l'espace engendré par les 4 premiers axes) et de la largeur choisie pour le listage de présentation des résultats:

'nombre de facteurs a utiliser = '
 'largeur, en caractères, du tableau des résultats = '

Ce dialogue terminé, le programme lit les fichiers de facteurs et crée le listage de résultats, sous la forme d'un texte qu'on peut afficher à l'écran grâce à un éditeur; et, éventuellement, imprimer.

6.2 Présentation des résultats

Le listage des résultats a un nom qui en évoque clairement le contenu:

'disq:tabaffi->iq'

(nom où les sigles 'i' et 'iq' varieront selon les exemples traités).

Le listage comprend d'une part un premier tableau énumérant les individus, (dans l'ordre du fichier où ont été lues leurs coordonnées sur les axes factoriels), en donnant pour chacun d'eux, après une flèche (->) le centre auquel il a été affecté. Puis vient une succession de tableaux donnant, pour chaque centre, la liste des individus qui lui ont été affectés.

Sur les extraits de listage donnés ici, on constate notamment que:

Les 7 individus affectés par 'discri' à i224, centre de la classe 224, sont précisément ceux qui, par la CAH, ont constitué cette classe;

Les 6 individus affectés par 'discri' à i226, centre de la classe 226 de la CAH, sont d'une part les 5 individus de cette classe et d'autre part l'individu Fa2;

```

disq:tab ; nombre de facteurs utilises = 4
affectation des i aux iq
( Be1->j237)( Be3->j230)( Be4->j237)( Be5->j228)( Bh1->j209)( Bh2->j239)
( Bh3->j237)( Bh4->j230)( Bh5->j226)( Bh6->j237)( Cc1->j209)( Cc2->j239)
( Cc3->j232)( Cc4->j237)( Cc5->j228)( Cc6->j239)( Ce1->j229)( Ce2->j222)
( Ce3->j230)( Ce4->j239)( Ce5->j228)( Ce6->j235)( Ea1->j209)( Ea2->j230)
( Ea3->j237)( Ea4->j236)( Ea6->j239)( Eb2->j237)( Eb3->j229)( Eb4->j229)
.....
liste des i affectes a j226
  Bh5 Fa2 Fc3 Lc1 Ld2 Oc4
.....
liste des i affectes a j224
  Fa6 Hb4 Hc5 Oa2 Ob2 Ra2 Ua2

```

```

=====
c | Partition en 13 classes : Sigles des individus de la classe numero c
-----
.....
226| Bh5 Ld2 Fc3 Lc1 Oc4
.....
235| Ld1 Ja3 Ce6 Fa2 Qa2 La2 Ma3 Fb2
224| Hb4 Hc5 Fa6 Ra2 Ua2 Oa2 Ob2
-----

```

```

237 _____ 243 _____ 248 _____ 250 _____
232 _____| _____| _____| _____|
222 _____ 240 _____| _____|
229 _____| _____|
228 _____ 245 _____| _____|
236 _____ 242 _____| _____|
230 _____| _____|
238 _____ 246 _____ 249 _____|
239 _____| _____|
226 _____ 244 _____ 247 _____| _____|
209 _____| _____|
235 _____ 241 _____| _____|
224 _____| _____|

```

ci dessus 1 arbre de la partition en 13 classes

Cet individu Fa2 est rangé par la CAH dans une classe i235; laquelle se place avec i226 dans une même branche, sous le nœud 247; cette proximité pouvant expliquer un échange d'individu dans l'affectation.

N.B. Les programmes décrits dans la présente notice font partie du Logiciel Mac-SAIF, ('Système d'Analyse des InFormations'); les lecteurs désireux d'acquérir ce Logiciel s'adresseront à la

Société STATMATIC: 4, rue de Fécamp 75012 Paris;
Téléphone: (16.1) 43.42.48.19 / (16.1) 47.98.77.39