Table des matières

Table des figures

Chapitre 1

Introduction

L'étude suivante a pour objet le traitement d'une base de données sociologiques grâce notamment aux arbres de classification (et de régression). Ce sont des méthodes exploratoires qui peuvent guider l'utilisateur vers la mise en place de traitements plus complets et plus précis. L'apparition de ces méthodes date de 1963 avec les travaux de Morgan et Sonquist. En 1984, Brieman, Friedman, Olshen et Stone ont renouvelé cette approche en développant la méthode la plus connue : la méthode CART (Classification And Regression Trees).
C'est dans ce contexte que nous traiterons ces données sociologiques grâce au logiciel Splus. Nous effectuerons une première approche des données qui proviennent d'un questionnaire destiné aux personnes âgées (normalité,...). Puis, nous traiterons à proprement dit le sujet grâce aux arbres de classification : à savoir déterminer les variables qui "expliquent" le mieux la variable lieu de vie. Enfin, une dernière étude aura pour but d'examiner les arbres de régression sous Splus et les arbres de classification lorsque les variables "explicatives" sont des variables qualitatives à plus de 2 modalités.

Chapitre 2

Présentation du CICT : Centre Inter-universitaire de Calcul de Toulouse

2.1 Présentation de l'entreprise

2.1.1 Présentation générale

Le Centre Inter-universitaire de Calcul de Toulouse est un centre de ressources informatiques (C.R.I.), service commun aux établissements universitaires toulousains suivants :

* U.T.1 : Toulouse I, Université des Sciences Sociales
* U.T.M. : Toulouse II, Université de Toulouse Le Mirail
* U.P.S. : Toulouse III, Université Paul Sabatier
* I.N.P.T. : Institut National Polytechnique de Toulouse
* I.N.S.A. : Institut National des Sciences Appliquées

Destiné principalement aux équipes de recherche scientifique, il est à l'écoute des besoins des universitaires pour jouer un rôle fédérateur dans le domaine de l'équipement informatique. Il analyse ces besoins pour fournir, soit des moyens en complément de leur équipement, soit l'ensemble de la solution informatique.
Créé en 1972 avec une douzaine de personnes, il compte aujourd' hui 35 personnes.

2.1.2 Organisation

Le CICT est administré par un conseil de 39 membres représentant les établissements co-contractants et certains laboratoires et organismes régionaux.
Rattaché administrativement à l'université Paul Sabatier, il est placé sous la responsabilité d'un directeur nommé pour 5 ans, assisté d'un directeur technique. Le directeur est **Yves Raynaud**, professeur d'informatique à l'université Paul Sabatier et le directeur technique est **Jean-Pierre Sylvain**.
Le CICT comprend un service administratif et comptable, et des services techniques qui travaillent en étroite collaboration.

2.1.3 Les services offerts

Le CICT offre des services dans de nombreux domaines de l'informatique :

* Mise à disposition de moyens matériels et logiciels

- des réseaux informatiques, des serveurs, de nombreux logiciels (le CICT est un centre de diffusion de logiciels dans le cadre d'une convention passée entre le ministère de l'Education Nationale et Microsoft)
- des imprimantes, un numériseur (scanner)
- plusieurs parcs de terminaux X
- des PC et Macintosh en libre service pour réaliser des transferts de fichiers

* Formation

Le CICT organise des stages de formation pour ses utilisateurs. Il conçoit et donne des formations spécifiques à ses logiciels et matériels.
Pour répondre à une demande importante, le CICT propose également des stages de formation à l'utilisation de logiciels pour micro-informatique.
Ouvert sur l'extérieur, le CICT organise des formations dispensées par des intervenants extérieurs et peut également concevoir des formations pour des organismes extérieurs.

* Documentation, Assistance

En plus des manuels disponibles à la permanence, de nombreuses documentations sont accessibles "en ligne" sur les ordinateurs, pour consultation ou impression. Il s'agit de documentations fournies par les constructeurs ou écrites par le CICT.
Le CICT assure une assistance aux utilisateurs de micros et de réseaux de micros comprenant 5 axes : conseil, achat, installation, formation, maintenance.

* Développement

Dans certains cas, le CICT peut prendre en charge le développement d'une application informatique.
Il est également associé à un projet d'enseignement multimédia à distance avec le service de la formation continue de l'UPS, l'Aérospatiale et Hewlett Packard : le projet FUDMIP (Formation Universitaire à Distance en Midi-Pyrénées) et un projet européen d'enseignement multimédia à distance : le projet ARIADNE.

* Exploitation, sauvegardes

Un des rôles les plus importants d'un centre informatique consiste à assurer la sécurité des données qui lui sont confiées (sécurité vis à vis des accidents et vis à vis des pirates). Elle est assurée par des matériels, des logiciels et des procédures de travail.

2.2 Présentation du service

Le stage a été réalisé au sein du service *Applications et Logiciels* dirigé par Mr. Thouzellier. Ce service a pour but d'installer et de suivre les logiciels d'application dans de nombreux domaines (graphiques, bases de données, statistiques, ...). Il s'occupe de la formation et de l'assistance des utilisateurs.
L'étude statistique suivante a été menée en étroite collaboration avec Mr. Saint Pierre, ingénieur, responsable de l'utilisation des logiciels de statistiques au sein de ce groupe.
En outre, Mr. saint Pierre est également chargé de traiter des données statistiques provenant de la faculté Paul Sabatier, mais aussi de l'extérieur (données sociologiques de la faculté du Mirail par exemple).

Chapitre 3

Présentation des données et méthodologie

Il s'agit d'un questionnaire : les mêmes questions ont été posées à des personnes âgées vivant soit à domicile, soit en maison de retraite (le lieu de domicile étant codé par la variable qualitative v4 à 2 modalités : "1" représente les individus en institut de retraite et "2" les individus à domicile).
Ces questions ont porté pour l'essentiel sur les impressions personnelles des personnes interrogées (stress, soutien social, ...). 5 réponses au choix étaient possibles. A chaque réponse était associé un "nombre de points" spécifique :
* pas du tout d'accord (1)
* un peu d'accord (2)
* accord moyen (3)
* accord (4)
* tout à fait d'accord (5)

10 variables représentent ainsi au final des scores :
  ⋆ les 3 ve.. concernent l'estime (de soi ...)
  ⋆ les 3 vs.. concernent le stress
  ⋆ les 4 co.. concernent le coping (stratégie d'adaptation à des situations difficiles)

D'autres variables sont entrées en ligne de compte :
* l'âge : variable quantitative (v5)
* le sexe : variable qualitative (v8)

Le but de l'étude est d'expliquer la variable lieu de vie par les 10 scores sachant que les variables âge et sexe peuvent jouer un rôle important.
On utilisera notamment une méthode statistique de plus en plus répandue : les arbres de classification et de régression. Ce sont des méthodes qui apparaissent comme une alternative à des méthodes plus classiques comme l'analyse factorielle discriminante ou la régression linéaire. Mais elles présentent l'avantage de nécessiter moins d'hypothèses (notamment de linéarité). Elles sont également très facile à lire.
Ainsi, dans un premier temps, une étude préliminaire (qui aura pour but de se familiariser avec les variables) sera réalisée. Puis, nous en viendrons au sujet proprement-dit : à savoir l'étude de la variable v4 au moyen des arbres de classification. Enfin, en complément nous verrons un exemple d'arbre de régression et un exemple d'arbre de classification où les variables explicatives possèdent plus de 2 modalités.
Les traitements sont réalisés avec les logiciels Splus (pour l'essentiel) et SPSS.

Chapitre 4

Etude préliminaire

Le but de ce chapitre est de se faire une première idée des variables contenues dans le fichier de données. Ainsi, nous examinerons d'abord les distributions des variables (normalité et boîtes à moustaches). Puis nous réaliserons une analyse en composantes principales sur les 10 scores afin de mettre en évidence des possibles interactions entre ces variables.

4.1 Normalité des variables

Notre première préoccupation est de valider la normalité des variables. Un moyen de la vérifier est de tracer les histogrammes de chaque variable. Grâce à la forme, nous pouvons avoir une idée assez précise de la normalité de ces variables (Cf. Figure 4.1). Un autre moyen est de tracer les "qqnorm" puis les "qqline" (Cf. annexes).
Dans l'ensemble, les résultats semblent convaincants, mis à part pour les variables vs36 et co32. Malgré tout, nous les prendrons en compte dans nos études ultérieures.

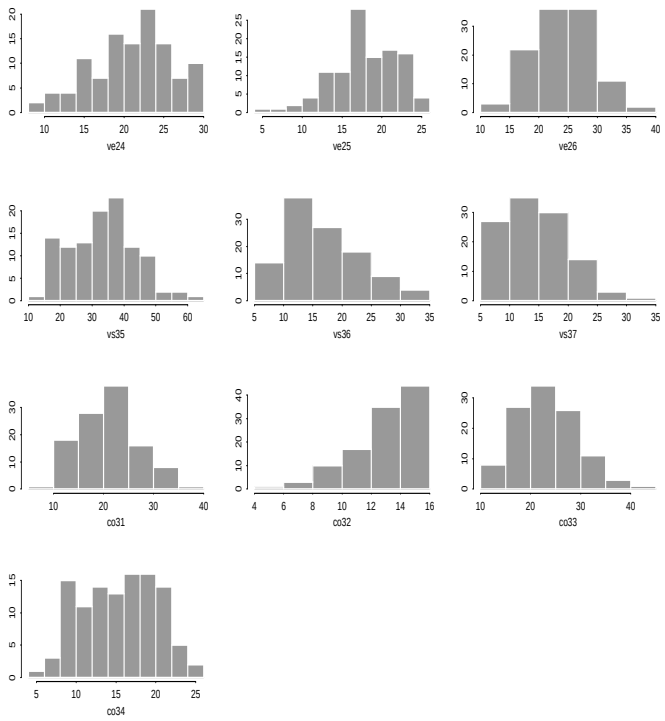


FIG. 4.1 – *Histogrammes*

4.2 Boxplots des 10 scores suivant le lieu de vie

On remarque (Cf. Figure 4.2) pour toutes les boîtes à moustaches, que la médiane se situe entre 15 et 25, mis à part pour la variable vs35 qui est particulière. En effet, cette variable possède l'écart-type le plus important de toutes (valeurs comprises entre $\simeq$ 15 et $\simeq$ 60). Au contraire, la variable co32 est celle qui est la plus homogène (valeurs comprises entre 5 et 15 avec une majorité de valeurs égales à 15 et 14).
Pour les variables d'estime, les médianes sont toujours un peu plus fortes lorsqu'elles concernent les personnes vivant à domicile. On ne retrouve pas ce phénomène pour les variables de stress et les variables de coping (pour ces dernières, ce serait d'ailleurs plutôt le processus inverse : les médianes sont plus fortes pour des personnes vivant dans une institution).

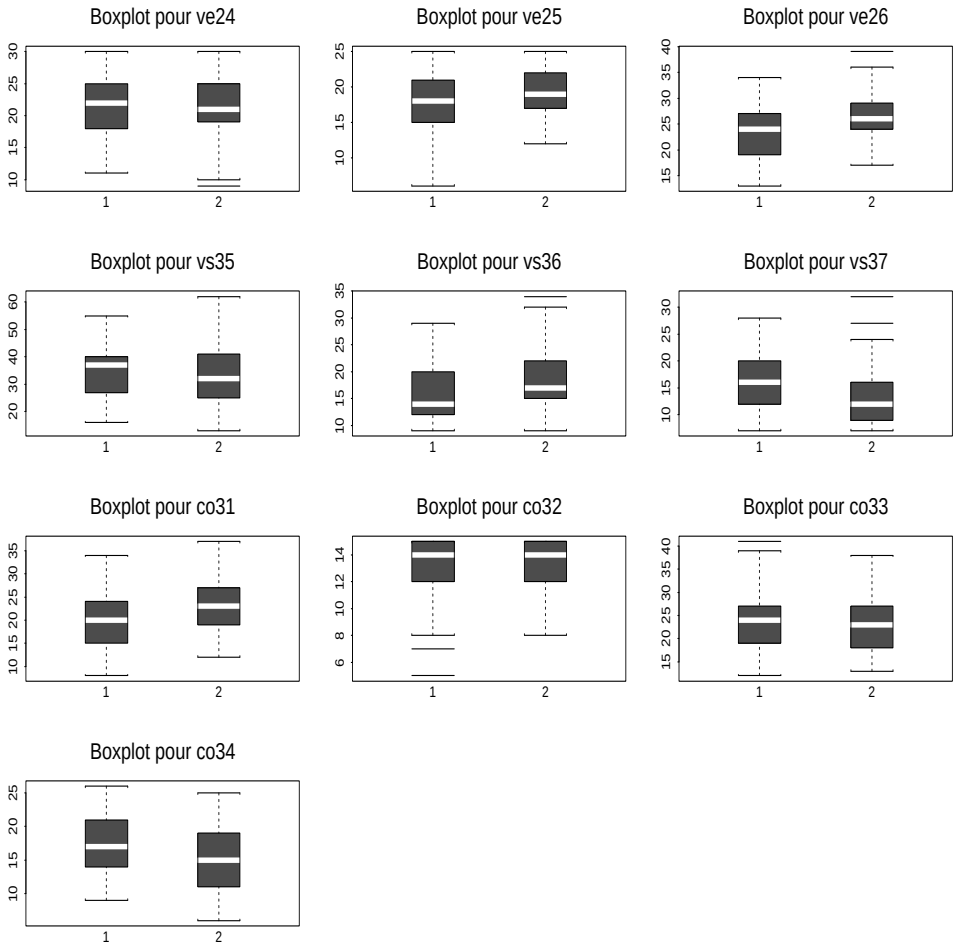


FIG. 4.2 – *Boîtes à moustaches*

4.3 Analyse en composantes principales des 10 scores

Nous avons réalisé une ACP centrée réduite sur les 10 scores.
D'après le graphe des valeurs propres ("screeplot") (Cf. annexe), nous décidons de ne retenir que les 4 premières composantes principales.
D'après la représentation des variables (Cf. Figure 4.3), le 1^{er} axe principal apparaît comme un axe d'opposition entre les variables d'estime et les variables de stress. Au niveau du 2^{nd} axe, quasiment toutes les variables ont des coordonnées positives. L'axe 3 est caractérisé par l'opposition : co31 contre ve24, co32 et co34. Quant à l'axe 4, il est quasiment exclusivement illustré par la variable co33.
Nous avons ensuite rajouté la variable illustrative "lieu de vie" aux résultats de l'ACP (Cf. Figure 4.3). Pour cela nous avons représenté sur un graphique les barycentres des individus définis par la variable qualitative v4.

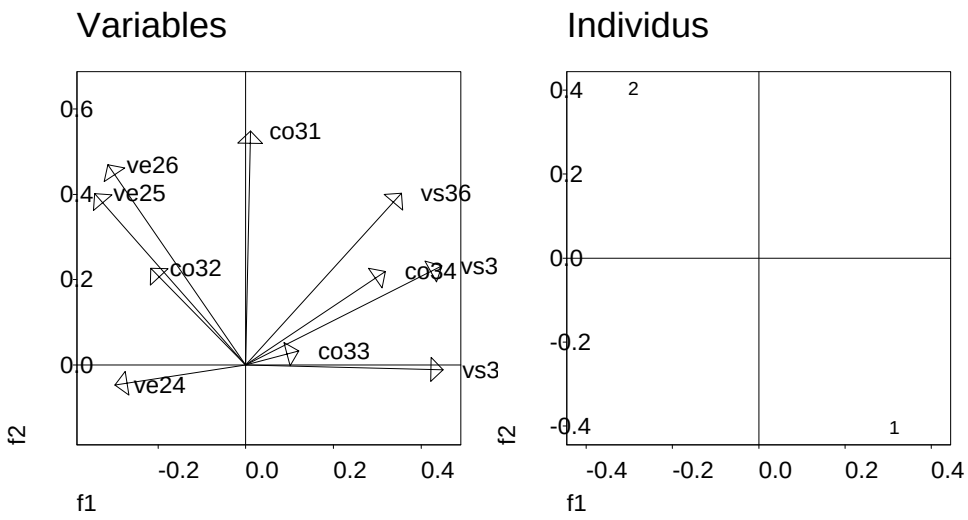


FIG. 4.3 – *Représentation des var., et des barycentres des ind. suivant le lieu de vie*

On remarque ainsi que l'axe 1 (celui qui a, par définition, le plus de poids) oppose le domicile à l'institut. Pour l'axe 2, c'est encore plus flagrant.
Il semble ainsi que les individus vivant en institution sont liés positivement aux variables de stress.

Chapitre 5

Explication de v4 grâce aux arbres de classification

Ce chapitre s'attache essentiellement à décrire le modèle d'explication de la variable lieu de vie grâce à la méthode des arbres de classification. Nous essaierons de déterminer le modèle de meilleure qualité.
Nous parlerons souvent de classe 1 et 2 : la "1" correspond à la modalité "Institution" et la "2" à la modalité "Domicile" de la variable lieu de vie.

5.1 Modèle : $v4 \sim 10\, scores + v5 + v8$

Le but est de comprendre l'utilisation de la méthode des arbres de classification. Tous les traitements suivants sont réalisés avec le logiciel Splus.
Notre première étude porte sur l'explication du "lieu de vie" par les 10 scores, la variable "âge" et la variable "sexe". Il s'agit donc d'un arbre de classification puisque la variable à expliquer est une variable qualitative. Ce premier traitement est largement détaillé.

∗ La commande de base est : **tree**
Elle donne les différentes étapes de la construction de l'arbre. Ainsi, nous obtenons pour le modèle considéré, les résultats suivants :

```
node), split, n, deviance, yval, (yprob)
      * denotes terminal node

 1) root 110 152.500 1 ( 0.5000 0.50000 )
   2) v5<80.5 54  65.630 2 ( 0.2963 0.70370 )
     4) ve26<20.5 7   5.742 1 ( 0.8571 0.14290 ) *
     5) ve26>20.5 47  48.650 2 ( 0.2128 0.78720 )
      10) ve25<18.5 18   0.000 2 ( 0.0000 1.00000 ) *
      11) ve25>18.5 29  37.360 2 ( 0.3448 0.65520 )
        22) co31<27.5 24  32.600 2 ( 0.4167 0.58330 )
          44) co33<24 17  18.550 2 ( 0.2353 0.76470 )
            88) v8:1 8  11.090 1 ( 0.5000 0.50000 ) *
            89) v8:2 9   0.000 2 ( 0.0000 1.00000 ) *
          45) co33>24 7   5.742 1 ( 0.8571 0.14290 ) *
        23) co31>27.5 5   0.000 2 ( 0.0000 1.00000 ) *
   3) v5>80.5 56  68.750 1 ( 0.6964 0.30360 )
     6) vs37<12.5 15  20.190 2 ( 0.4000 0.60000 )
      12) ve26<27.5 10  13.460 1 ( 0.6000 0.40000 )
        24) ve24<23.5 5   5.004 1 ( 0.8000 0.20000 ) *
        25) ve24>23.5 5   6.730 2 ( 0.4000 0.60000 ) *
      13) ve26>27.5 5   0.000 2 ( 0.0000 1.00000 ) *
     7) vs37>12.5 41  40.470 1 ( 0.8049 0.19510 )
      14) vs36<26 33  20.110 1 ( 0.9091 0.09091 )
        28) vs37<16.5 13  14.050 1 ( 0.7692 0.23080 )
          56) ve25<17 5   6.730 1 ( 0.6000 0.40000 ) *
          57) ve25>17 8   6.028 1 ( 0.8750 0.12500 ) *
        29) vs37>16.5 20   0.000 1 ( 1.0000 0.00000 ) *
      15) vs36>26 8  10.590 2 ( 0.3750 0.62500 ) *
```

Chaque noeud possède donc un numéro. Au niveau de tous les noeuds, on distingue :
– la partition de laquelle il est issu
– le nombre d'individus qu'il contient
– la déviance au sein de chaque noeud : c'est à dire l'hétérogénéité de la variable v4
– la classe dans laquelle la majorité des individus sont regroupés
– entre parenthèses, les probabilités pour que les individus appartiennent aux différentes classes
Les "$*$" marquent les noeuds terminaux (au nombre de 13 dans notre cas).

Ainsi, au niveau du noeud numéro 2, la partition fait intervenir la variable quantitative v5. La partition est donc de la forme $v5 < t$ (contre $v5 > t$) avec t=80.5. Au sein de ce noeud, on trouve 54 individus. La déviance est égale à 65.63. Les individus sont majoritairement rassemblés dans la classe 2. En effet, la proportion d'individus dans la classe 2 est de 0.7037.
Au niveau du noeud numéro 88, la partition fait intervenir la variable qualitative v8. Les niveaux sont donc séparés en 2 classes. Ici, la variable v8 ne possède que 2 modalités. Donc la séparation est uniforme (c'est différent lorsque la variable possède plus de 2 modalités). Dans ce noeud, ce sont les individus ayant 1 comme modalité de v8 qui sont sélectionnés. Ils sont 8. La déviance est égale à 11.09. Ces 8 individus sont répartis équitablement dans les 2 classes.

∗ Grâce à la commande **summary**, nous obtenons des éléments essentiels sur l'arbre :

```
Classification tree:
tree(formula = v4 ~ ve24 + ve25 + ve26 + vs35 + vs36 + vs37 + co31 + co32 +
                    co33 + co34 + v5 + v8, data = etude3)
Variables actually used in tree construction:
[1] "v5"   "ve26" "ve25" "co31" "co33" "v8"   "vs37" "ve24" "vs36"
Number of terminal nodes:  13
Residual mean deviance:  0.5943 = 57.65 / 97
Misclassification error rate: 0.1364 = 15 / 110
```

Les variables jouant un rôle dans la construction de l'arbre sont indiquées. On remarque ainsi que les variables vs35, vs37, co32, co34 ne sont pas présentes.
Le "Residual mean deviance" correspond à la déviance totale de l'arbre. C'est la somme des résidus de chaque individu. Elle peut aussi s'obtenir en tapant la commande *deviance.tree(tr)*. Le but est qu'elle soit la plus faible possible.
Le "Misclassification error rate" correspond à la proportion d'individus mal classés. Elle est ici de 15 sur 110 (ce qui semble correct). Il y a donc 15 individus qui ont "migré" d'une classe vers l'autre.

∗ Il existe des commandes pour représenter ces arbres. Il s'agit de :
– **plot.tree** et **text.tree** : l'arbre est visible directement dans une fenêtre graphique.
– **post.tree** : un fichier "postscript" (visionnable sur une fenêtre externe) est créé. Ce graphique est plus soigné et plus complet. Les graphiques seront donc réalisés sous ce format (Cf. Figure 5.1).

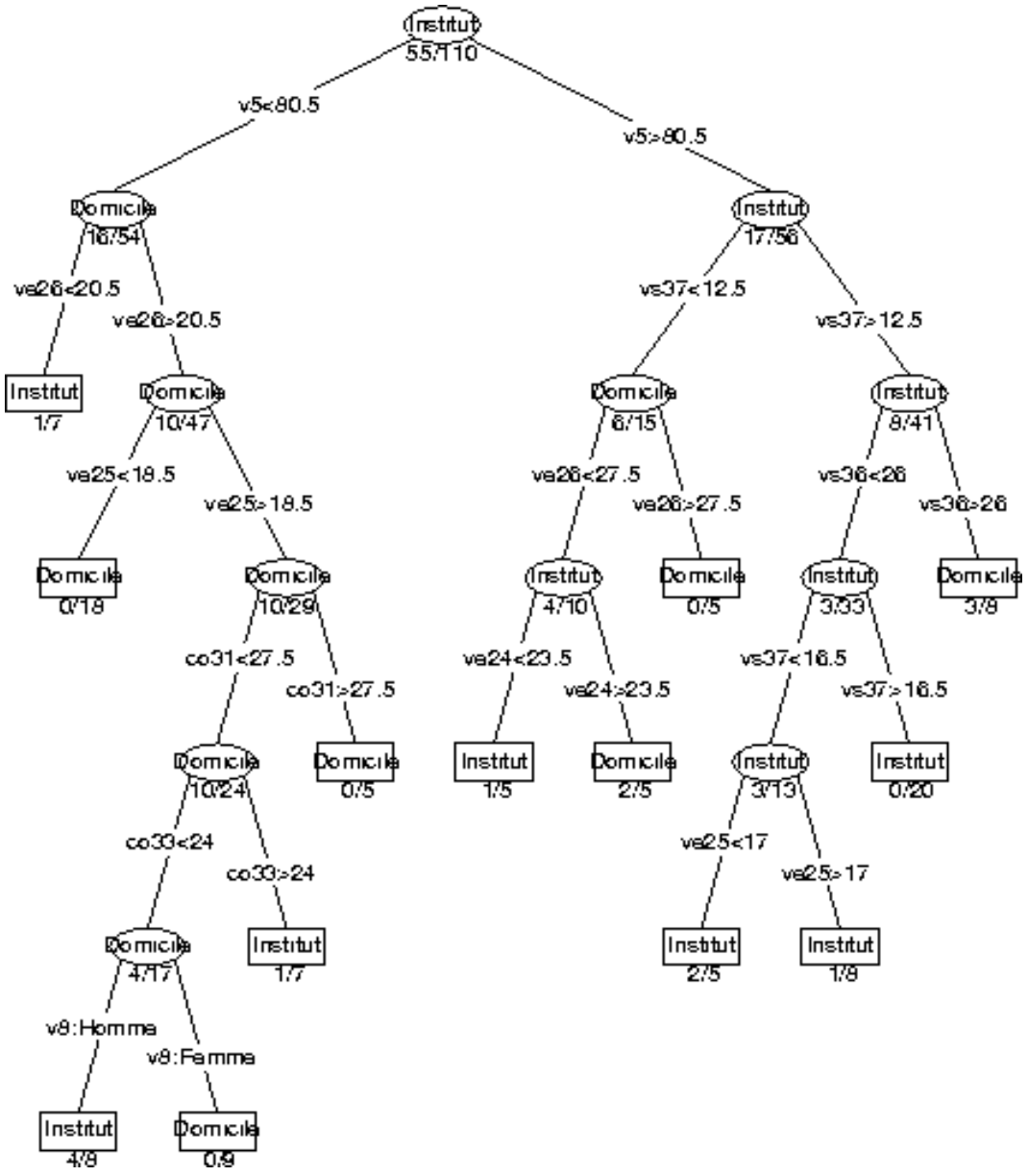


FIG. 5.1 – *Arbre de classification pour v4*

Comment lire le graphique ?

Les noeuds de forme ovale sont les noeuds intermédiaires. Les noeuds de forme rectangulaire sont les noeuds terminaux. Les branches de l'arbre correspondent à des partitions.
Le noeud initial ("root node") est "scindé" en 2 nouveaux noeuds par la variable âge. Lorsque la variable âge est inférieure à 80.5, la majorité des individus est regroupée dans la classe 2 (38 sur 54). Quand la variable âge est supérieure à 80.5, la majorité des individus est regroupée dans la classe 1 (39 sur 56).
Les variables "discriminantes" qui partitionnent les premières dans l'arbre sont considérées comme les variables ayant le plus de poids dans la discrimination de la variable lieu de vie. Ainsi, dans l'ordre d'importance, on trouve les variables suivantes : v5, ve26, vs37, ...

On peut penser qu'il existe une interaction entre ve26 et v5, et entre vs37 et v5. En effet, ve26 et vs37 sont des variables partitionnant deux noeuds issus d'un même noeud parent (qui est lui divisé par la variable v5).
La lecture du graphique ne peut pas se suffire à elle même car ce serait alors une interprétation incomplète voire érronée.

∗ Les commandes **identify.tree** et **browser.tree** fonctionnent de façon interactive en cliquant sur les noeuds du graphique qui intéressent l'utilisateur. La première permet d'identifier les individus au sein du noeud sélectionné. La seconde permet d'obtenir les mêmes renseignements que la fonction **tree**, mais uniquement pour le noeud considéré.

∗ La commande **prune.tree** nous indique un élagage possible de l'arbre. Car le but est d'obtenir un arbre le plus court possible (ie : avec le moins de prédicteurs) tout en minimisant la proportion d'individus mal classés.
Il existe 2 méthodes à travers cette commande :

- Méthode "déviance" : c'est la méthode par défaut sous Splus:

```
$size:
[1] 13 12 11 10  9  6  4  2  1

$dev:
[1]  57.65120  58.93807  60.66416  66.72484  73.45496  93.98575 111.85673
[8] 134.38322 152.49238

$k:
[1]      -Inf  1.286868  1.726092  6.060675  6.730117  6.843597  8.935492
[8] 11.263243 18.109163

$method:
[1] "deviance"

attr(, "class"):
[1] "prune"         "tree.sequence"
```

Elle est basée sur le principe de minimiser la déviance totale de l'arbre.
Il est donc possible d'élaguer l'arbre de façon à ce qu'il n'ait plus que 12, 11, 10, 9, 6, 4, ... noeuds terminaux. Ce qui signifie que l'arbre qui ne contiendrait que 7 noeuds terminaux n'existe pas.
La déviance augmente de façon importante selon que l'arbre possède 9 où 6 noeuds terminaux (on passe respectivement de $\simeq 73$ à $\simeq 94$).
Ce phénomène est remarquable sur un simple graphique obtenu par la commande *plot*. On serait donc tenté de choisir l'arbre qui possède 9 noeuds terminaux.

- **Méthode "misclass"** : elle est utilisée uniquement pour les arbres de classification :

```
$size:
[1] 13 10  9  6  4  3  2  1

$dev:
[1] 15 15 16 21 25 28 33 55

$k:
[1]      -Inf  0.000000  1.000000  1.666667  2.000000  3.000000  5.000000
[8] 22.000000

$method:
[1] "misclass"

attr(, "class"):
[1] "prune"         "tree.sequence"
```

On remarque pour cette méthode, que la meilleure taille correspond à 9 noeuds terminaux (on passe de 16 à 21 au niveau des déviances).

Afin de visualiser cet arbre, il suffit de taper la commande **prune.tree** en rajoutant en argument "best=9" pour indiquer le nombre de noeuds terminaux choisis.
On peut aussi indiquer l'argument "k=6.730117" qui produira le même arbre.
k représente une pénalité pour chaque noeud terminal. Ce qui permet de calculer le "coût-complexité" :

$$\text{coût-complexité} = \text{coût global de l'arbre} + \text{k} * \text{nombre de noeuds terminaux}$$

A partir de ce calcul, Splus définit le meilleur sous arbre possible en minimisant la valeur du coût-complexité (Cf. Figure 5.2).

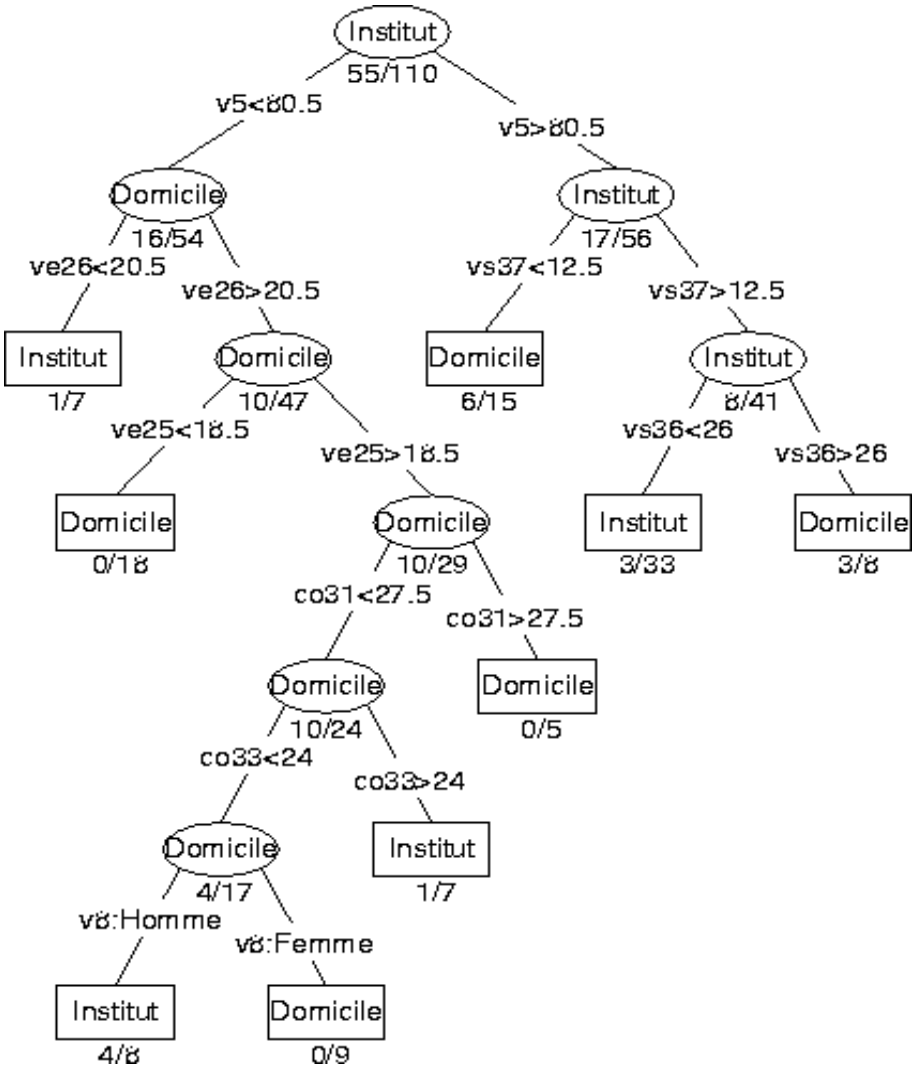


FIG. 5.2 – *Arbre de classification élagué pour v4*

Le résumé de cet arbre nous indique que 18 individus sur 110 sont mal classés, soit 3 individus de plus que pour l'arbre initial. Mais en contre partie, il ne reste plus que 9 noeuds terminaux (contre 13). On perd donc en qualité, mais on gagne en lisibilité.

Une autre manière classique d'élaguer un arbre de classification est de suivre la règle suivante : donner à la pénalité k la valeur $2*(K-1)$ où K est le nombre de modalités de la variable à expliquer. C'est la m'ethode visant à minimiser le critère d'Akaike.
Dans notre cas, $K = 2$ car v4 possède 2 modalités.
Par cette méthode, il reste 11 noeuds terminaux et 16 individus sont mal classés. Ce sont les mêmes variables qui participent à l'élaboration de l'arbre.

Ces méthodes d'élagage conduisent à des résultats convenables. La première aboutit à moins de noeuds terminaux (mais à une déviance un peu forte), et la seconde présente une proportion d'individus mal classés un peu moins importante (mais 2 noeuds terminaux supplémentaires). Tout dépend donc des prérogatives de l'utilisateur.

∗ La commande **cv.tree** permet de valider le modèle sélectionné précédemment (par la fonction **prune.tree**). Elle réalise de la validation croisée. Le principe est de diviser le fichier de données en α parties. L'une d'entre elle est mise de côté. Toutes les autres sont utilisées pour la construction de l'arbre et l'échantillon mis à part sert au final d'échantillon de contrôle. Ce processus est réitéré jusqu'à ce que chacune des α parties ait servie d'échantillon de contrôle. L'arbre final est alors une "moyenne" de tous les arbres obtenus.
C'est l'argument "rand" qui fixe le nombre de division du fichier de données. C'est un vecteur de dimension égale au nombre d'individus du fichier de données. En général, on lui donne des valeurs allant de 1 à α où α correspond à l'ordre de la validation croisée.
Par défaut, Splus réalise une validation croisée en divisant le fichier en 10 parties.
Cette division en α parties étant aléatoire, plusieurs tentatives ne donnent pas forcément les mêmes résultats.
Prenons un exemple sur l'arbre tr :

```
$size:
[1] 13 12 11 10  9  6  4  2  1

$dev:
[1] 339.4534 339.0071 339.0071 272.2507 252.1162 243.2260 195.9506 149.5512
[9] 163.0723

$k:
[1]      -Inf  1.286868  1.726092  6.060675  6.730117  6.843597  8.935492
[8] 11.263243 18.109163

$method:
[1] "deviance"

attr(, "class"):
[1] "prune"         "tree.sequence"
```

Ainsi, les 3 premières déviances sont quasiment égales. Puis la quatrième chute (elle vaut $\simeq 272$). Donc, on serait tenté de relever l'arbre qui possède 11 noeuds terminaux. Mais ce résultat est plutôt indicatif : tout dépend des desiderata de l'utilisateur.

Grâce à cette première étude largement détaillée, on s'aperçoit que la variable "âge" a une forte influence sur le lieu de vie et "cache" en quelque sorte l'influence des autres variables. Ainsi, lorsque la variable âge est inférieure à 80.5, les individus se regroupent essentiellement dans la classe 2 et quand l'âge est supérieur à 80.5, les individus se regroupent essentiellement dans la classe 1.
C'est pourquoi nous avons décidé de partager le fichier de données en 2 : d'un côté, les données concernant les personnes âgées de moins de 80.5 ans et de l'autre les données concernant les personnes âgées de plus de 80.5 ans.
Sur ces 2 fichiers, nous avons réalisé un arbre de classification de v4 suivant les 10 scores et v8 (qui contrairement à l'âge ne prend pas toute l'information).

5.2 Modèles : explication du lieu de vie suivant les 10 scores et v8 selon l'âge

5.2.1 Pour âge < 80.5

Pour construire l'arbre correspondant à âge < 80.5, nous avons d'abord construit le fichier "age1" regroupant uniquement les données pour lesquelles la variable v5 était $<$ à 80.5.
Les résultats sont alors :

```
node), split, n, deviance, yval, (yprob)
      * denotes terminal node

 1) root 54 65.630 2 ( 0.2963 0.7037 )
   2) ve26<20.5 7  5.742 1 ( 0.8571 0.1429 ) *
   3) ve26>20.5 47 48.650 2 ( 0.2128 0.7872 )
     6) ve25<18.5 18  0.000 2 ( 0.0000 1.0000 ) *
     7) ve25>18.5 29 37.360 2 ( 0.3448 0.6552 )
      14) co31<27.5 24 32.600 2 ( 0.4167 0.5833 )
        28) co33<24 17 18.550 2 ( 0.2353 0.7647 )
          56) v8:1 8 11.090 1 ( 0.5000 0.5000 ) *
          57) v8:2 9  0.000 2 ( 0.0000 1.0000 ) *
        29) co33>24 7  5.742 1 ( 0.8571 0.1429 ) *
      15) co31>27.5 5  0.000 2 ( 0.0000 1.0000 ) *
```

Seulement 5 variables sont utilisées. il y a 6 noeuds terminaux et la déviance de l'arbre est égale à 0.4703 et 6 individus sont mal classés (sur 54).
Les variables ayant le plus de poids sont ve26, ve25 et co31 .

5.2.2 Pour âge > 80.5

On considère cette fois-ci le fichier âge2 (qui rassemble toutes les observations pour lesquelles la variable "âge" est supérieure à 80.5).
On obtient alors :

```
node), split, n, deviance, yval, (yprob)
      * denotes terminal node

 1) root 56 68.750 1 ( 0.6964 0.30360 )
   2) vs37<12.5 15 20.190 2 ( 0.4000 0.60000 )
     4) ve26<27.5 10 13.460 1 ( 0.6000 0.40000 )
       8) ve24<23.5 5  5.004 1 ( 0.8000 0.20000 ) *
       9) ve24>23.5 5  6.730 2 ( 0.4000 0.60000 ) *
     5) ve26>27.5 5  0.000 2 ( 0.0000 1.00000 ) *
   3) vs37>12.5 41 40.470 1 ( 0.8049 0.19510 )
     6) vs36<26 33 20.110 1 ( 0.9091 0.09091 )
      12) vs37<16.5 13 14.050 1 ( 0.7692 0.23080 )
        24) ve25<17 5  6.730 1 ( 0.6000 0.40000 ) *
        25) ve25>17 8  6.028 1 ( 0.8750 0.12500 ) *
      13) vs37>16.5 20  0.000 1 ( 1.0000 0.00000 ) *
     7) vs36>26 8 10.590 2 ( 0.3750 0.62500 ) *
```

De nouveau, seulement 5 variables figurent dans l'arbre. Il y a 9 individus mal classés sur 56 et 7 noeuds terminaux. Les variables les plus importantes sont vs37, ve26 et vs36 (avec une possible interaction entre vs37 et ve26 et une autre entre vs37 et vs36).
Il est à noter que ve25 et ve26 se retrouvent dans les 2 arbres.

5.3 Modèles : explication du lieu de vie suivant les 10 scores, l'âge, le sexe et 4 nouvelles variables

Nous examinerons dans ce paragraphe, comment l'ajout de 4 nouvelles variables influence le modèle d'explication de v4.

5.3.1 Description des 4 nouvelles variables

Ces 4 nouvelles variables sont issues d'une analyse factorielle des correspondances qui concernait un autre jeu de données. Ces 4 nouvelles variables sont davantage liées à la vie pratique des personnes interrogées (et pas à leur sentiment comme jusqu'à présent). Elles ont reçu les noms suivants :
⋆ factsou1 : "hors famille-proximité amis voisins"
⋆ factsou2 : "hors famille-famille soignants"
⋆ factsou3 : "spiritualité"
⋆ factsou4 : "désaffection"
Ces variables présentent des écarts-types beaucoup plus importants que les variables étudiées jusqu'ici (Cf. Figure 5.3).

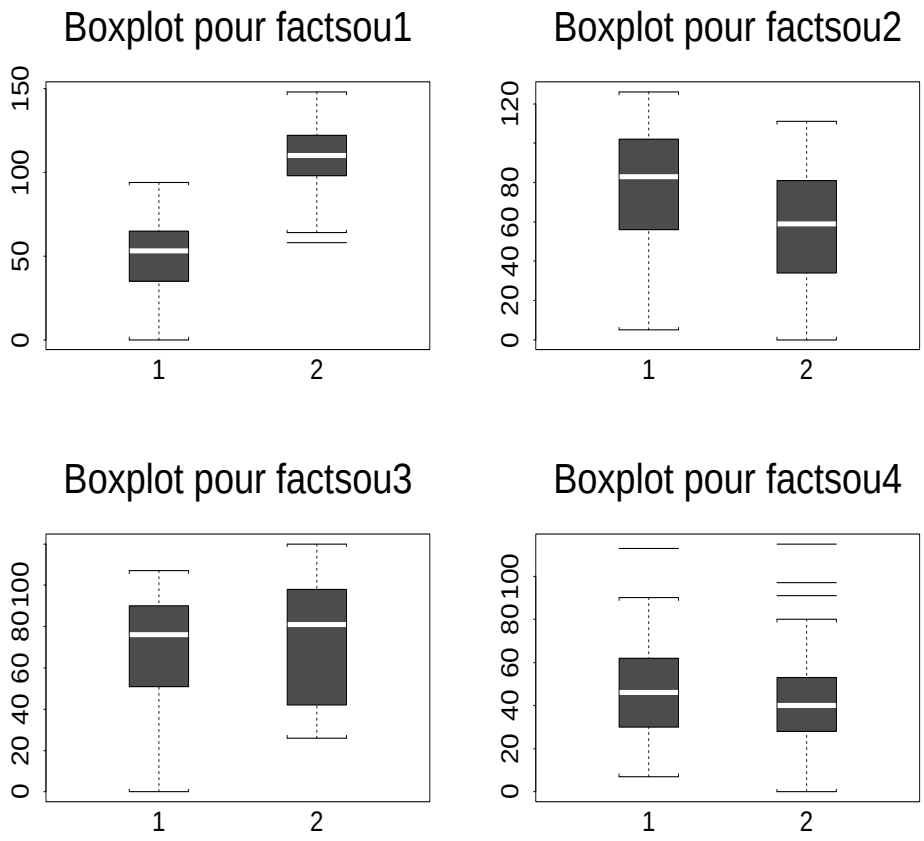


FIG. 5.3 – *Boîtes à moustaches*

Si on rajoute en variables illustratives à l'ACP réalisée dans le 3.3. ces 4 nouvelles variables, on remarque que :
– En dimension 1 et 2, la variable factsou1 semble corrélée avec les variables d'estime (notamment ve26) et co32. Elle serait donc en opposition aux variables de stress (Cf. Figure 5.4).
– En dimension 3 et 4, la variable factsou3 semble corrélée avec les variables co34 et vs35 et en opposition avec les variables factsou4 et factsou1 (Cf. Figure 5.4).
Mais aucune de ces 4 nouvelles variables n'apparaît comme une variable exceptionnelle qui prendrait toute l'information.

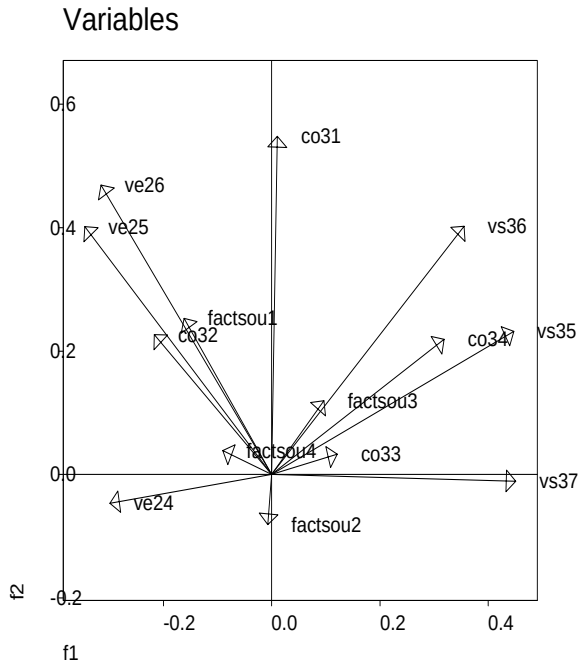


FIG. 5.4 – *Représentation des scores et des 4 variables illustratives en dim. 1 et 2*

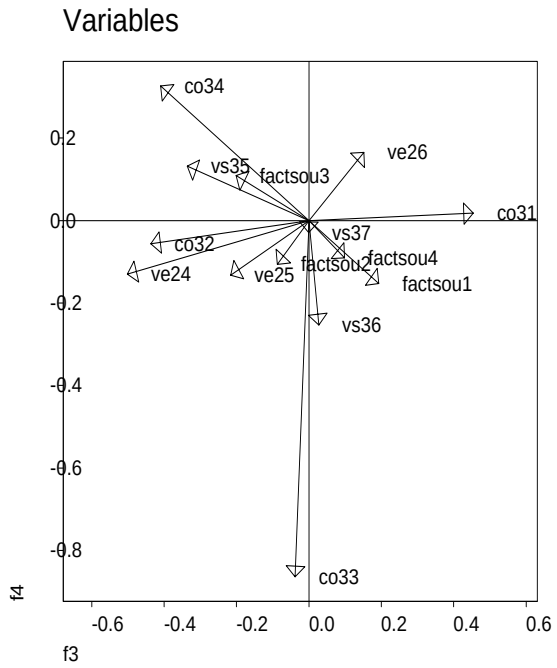


FIG. 5.5 – *Représentation des scores et des 4 variables illustratives en dim. 3 et 4*

Regardons maintenant si l'arbre de classification est modifié lorsque l'on ajoute au modèle ces 4 variables.

5.3.2 Arbre de classification de v4 suivant les 10 scores, v5, v8 et les 4 variables "factsou."

On réalise donc un arbre de classification sur le même modèle que le précédent mais en rajoutant les 4 variables factsou. . On obtient alors les résultats suivants :

```
node), split, n, deviance, yval, (yprob)
      * denotes terminal node

 1) root 110 152.500 1 ( 0.5000 0.5000 )
   2) factsou1<71.5 50  16.790 1 ( 0.9600 0.0400 )
     4) vs35<48.5 45   0.000 1 ( 1.0000 0.0000 ) *
     5) vs35>48.5 5   6.730 1 ( 0.6000 0.4000 ) *
   3) factsou1>71.5 60  43.230 2 ( 0.1167 0.8833 )
     6) factsou1<94.5 17  23.030 2 ( 0.4118 0.5882 )
      12) factsou2<57.5 9   0.000 2 ( 0.0000 1.0000 ) *
      13) factsou2>57.5 8   6.028 1 ( 0.8750 0.1250 ) *
     7) factsou1>94.5 43   0.000 2 ( 0.0000 1.0000 ) *
```

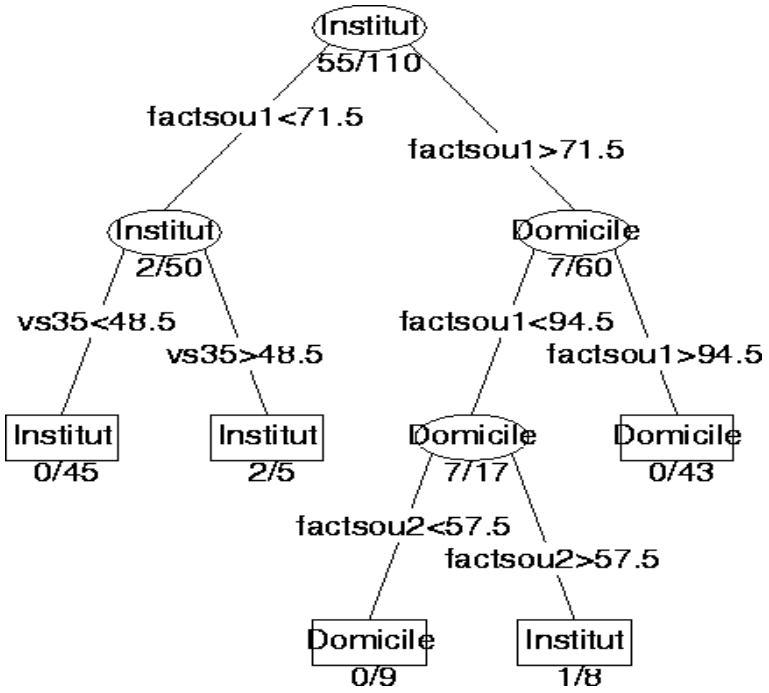


FIG. 5.6 – *Arbre de classification en fonction des "factsou."*

Cette fois-ci, la variable âge n'apparaît même pas parmi les variables discriminantes. En fait, aucune des variables présentes dans l'arbre de départ (Cf. Figure 5.1) ne se retrouvent dans celui-ci. Seulement 3 variables ont été nécessaires pour construire l'arbre: les variables factsou1, factsou2 et vs35. L'arbre est donc très limpide (3 variables discriminantes et 5 noeuds terminaux).
L'arbre est d'excellente qualité puisque 3 individus seulement sont malclassés (sur 110).
Ce résultats pourraient provenir de la nature même des variables rajoutées. En effet, comme on l'a dit, ces variables concernent la vie pratique et quotidienne des personnes (sur l'entourage, les voisins, la famille, les soignants, ...). Par conséquent, on peut penser que ces variables sont indubitablement

liées au lieu de vie et donc à la variable v4. Ce qui expliquerait pourquoi, ces variables prennent une telle place dans les résultats finaux.

Chapitre 6

Etude complémentaire

L'objet de ce chapitre est d'examiner comment fonctionne d'une part un arbre de régression, et d'autre part les arbres de classification lorsque les variables explicatives possèdent plus de 2 modalités.

6.1 Arbre de régression

La variable à expliquer est quantitative. Nous prendrons la variable âge en tant que telle. Ce sont les 10 scores qui joueront le rôle de variables explicatives. Sous Splus, les commandes permettant de réaliser un arbre de régression ne diffèrent pas vraiment de celles utilisées pour un arbre de classification.

* L'arbre obtenu est alors le suivant :

```
node), split, n, deviance, yval
      * denotes terminal node

 1) root 110 6167.00 81.07
   2) vs35<33.5 55 2052.00 78.25
     4) vs36<15.5 27 1299.00 80.41
       8) ve25<18.5 11  124.50 84.36
        16) co31<22.5 6   21.50 86.50 *
        17) co31>22.5 5   42.80 81.80 *
       9) ve25>18.5 16  883.40 77.69
        18) ve24<25.5 10  159.60 74.20
          36) ve25<20 5   16.80 72.20 *
          37) ve25>20 5  102.80 76.20 *
         .     .      .     .     .
         .     .      .     .     .
      14) vs36<16.5 21  827.00 84.62
        28) co33<25.5 13  468.30 81.77
          56) vs36<12 5  154.80 77.80 *
          57) vs36>12 8  185.50 84.25 *
        29) co33>25.5 8   81.50 89.25 *
      15) vs36>16.5 20  984.80 79.60
        30) ve25<16.5 6   96.83 75.17 *
        31) ve25>16.5 14  719.50 81.50
          62) ve25<19 7  207.40 87.29 *
          63) ve25>19 7   43.43 75.71 *
```

Le principe de lecture est le même que pour un arbre de classification. La formule de calcul de la déviance n'y est évidemment pas identique et au niveau de chaque noeud, on ne trouve non plus des proportions mais des valeurs prédites.
Ainsi, le noeud numéro 36 (qui est terminal) est issu de la partition $ve25 < 20$ (la dernière après bien d'autres). Il contient 5 individus. La déviance est relativement faible (égale à 16.8) comparée à celle des autres noeuds terminaux. Ce noeud paraît donc stable. La valeur prédite de v5 est alors 72.2.
Le résumé de cet arbre montre qu'il possède 17 noeuds terminaux pour une déviance égale à 22.54.

∗ L'élagage, grâce à la commande *prune.tree*, donne les résultats suivants :

```
$size:
 [1] 17 16 15 14 13 12 11 10  9  6  3  2  1

$dev:
 [1] 2095.837 2135.837 2181.850 2237.898 2298.144 2426.152 2599.606 2784.179
 [9] 3061.324 3922.008 4817.170 5293.782 6167.418

$k:
 [1]      -Inf  40.00000  46.01345  56.04848  60.24545 128.00769 173.45474
 [8] 184.57302 277.14469 286.89459 298.38734 476.61201 873.63636

$method:
[1] "deviance"

attr(, "class"):
[1] "prune"         "tree.sequence"
```

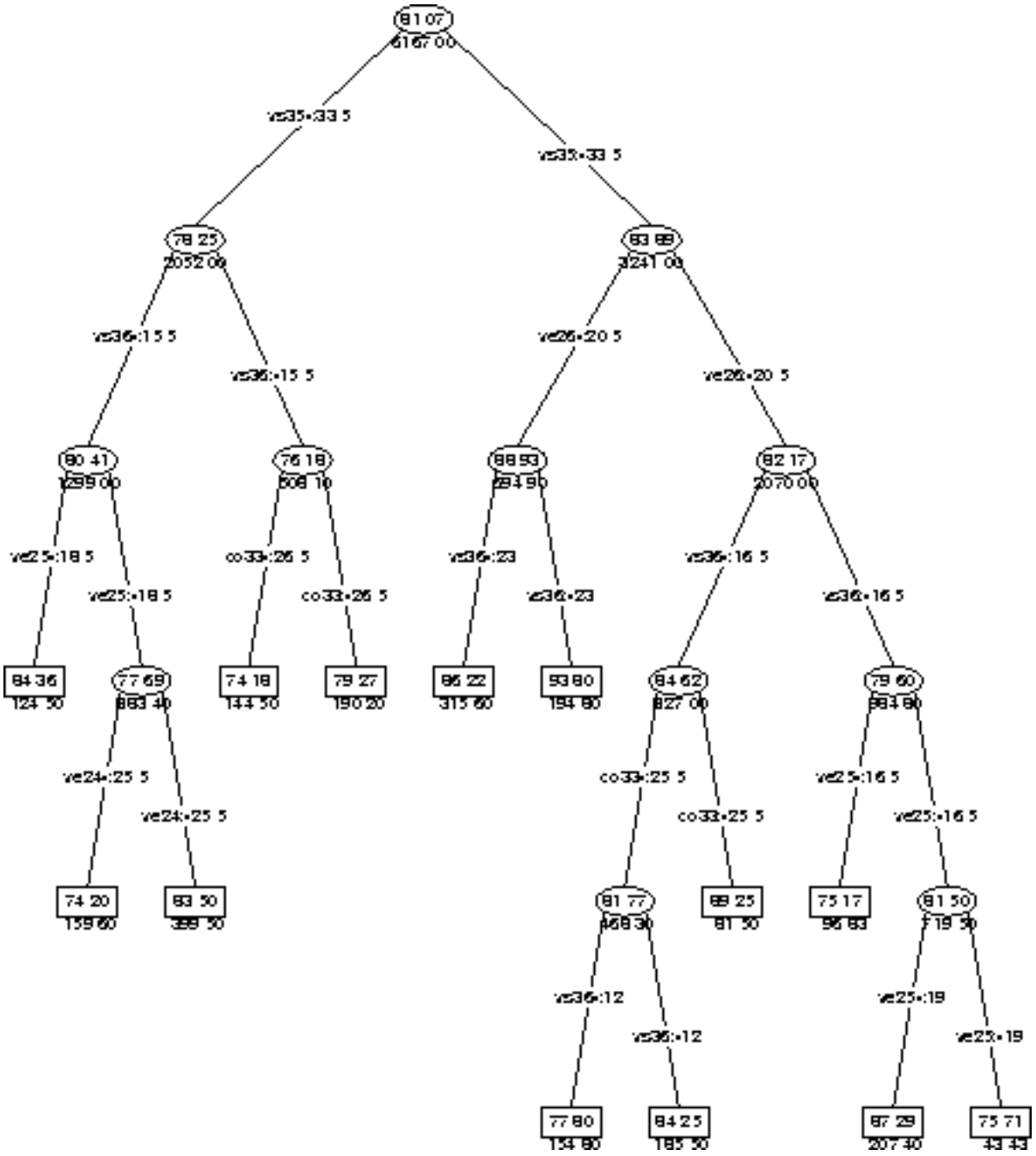
Selon le même principe que pour l'arbre de classification, on décide de réduire l'arbre de régression initial à un arbre ne possédant plus que 13 noeuds terminaux.
Ce choix nous paraît d'autant plus pertinent que la déviance de ce nouvel arbre n'est pas démesurément supérieure à celle de l'arbre initial puisque elle vaut 23.69.

∗ L'étape de la validation croisée semble confirmer notre choix :

```
$size:
 [1] 17 16 15 14 13 12 11 10  9  6  3  2  1

$dev:
 [1] 8625.936 8552.302 8552.302 8560.189 8510.758 7848.914 7885.807 7878.985
 [9] 7454.597 7513.484 7222.169 6678.335 6630.739

$k:
 [1]      -Inf  40.00000  46.01345  56.04848  60.24545 128.00769 173.45474
 [8] 184.57302 277.14469 286.89459 298.38734 476.61201 873.63636

$method:
[1] "deviance"

attr(, "class"):
[1] "prune"         "tree.sequence"
```

En effet, lorsque l'on passe de 13 à 12 noeuds terminaux, la déviance tombe de $\simeq 8511$ à $\simeq 7849$.
C'est encore un encouragement à retenir l'arbre possédant 13 noeuds terminaux.

Cet arbre de régression est présenté sur la page suivante :
cette fois-ci, les valeurs dans les noeuds sont les valeurs prédites de v5. Au dessous de chaque noeud, on trouve la valeur de la déviance au sein de ces noeuds.
Lorsque vs35 est $<$ à 33.5 et vs36 est $>$ à 15.5, la valeur prédite de l'âge est 76.18. La déviance à ce noeud vaut 508.1.
Comme pour un arbre de classification, ce sont les variables qui partitionnent les premières qui sont jugées les plus "prédictives". Ici, ce serait donc les variables vs35, vs36 et ve26.
Des interactions entre vs35 et vs36 d'une part, et entre vs35 et ve26 d'autre part sont plausibles.


81 07
6167 00
vs35<33 5
vs35>=33 5
78 25
2052 00
83 88
3241 00
vs36<15 5
vs36>=15 5
ve26<20 5
ve26>=20 5
80 41
1299 00
76 18
608 10
88 93
884 90
82 17
2070 00
ve25<18 5
ve25>=18 5
co33<26 5
co33>=26 5
vs36<23
vs36>=23
vs36<16 5
vs36>=16 5
84 36
124 50
77 69
883 40
74 18
144 50
79 27
190 20
86 22
315 60
53 80
194 80
84 62
827 00
79 60
883 80
ve24<25 5
ve24>=25 5
co33<25 5
co33>=25 5
ve25<16 5
ve25>=16 5
74 20
159 60
83 50
388 50
81 77
668 30
88 25
81 50
75 17
96 83
81 50
vs36<12
vs36>=12
ve25<19
ve25>=19
77 80
154 80
84 25
185 50
87 29
207 40
75 71
43 43

6.2 Arbre de classification pour des variables explicatives à plus de 2 niveaux

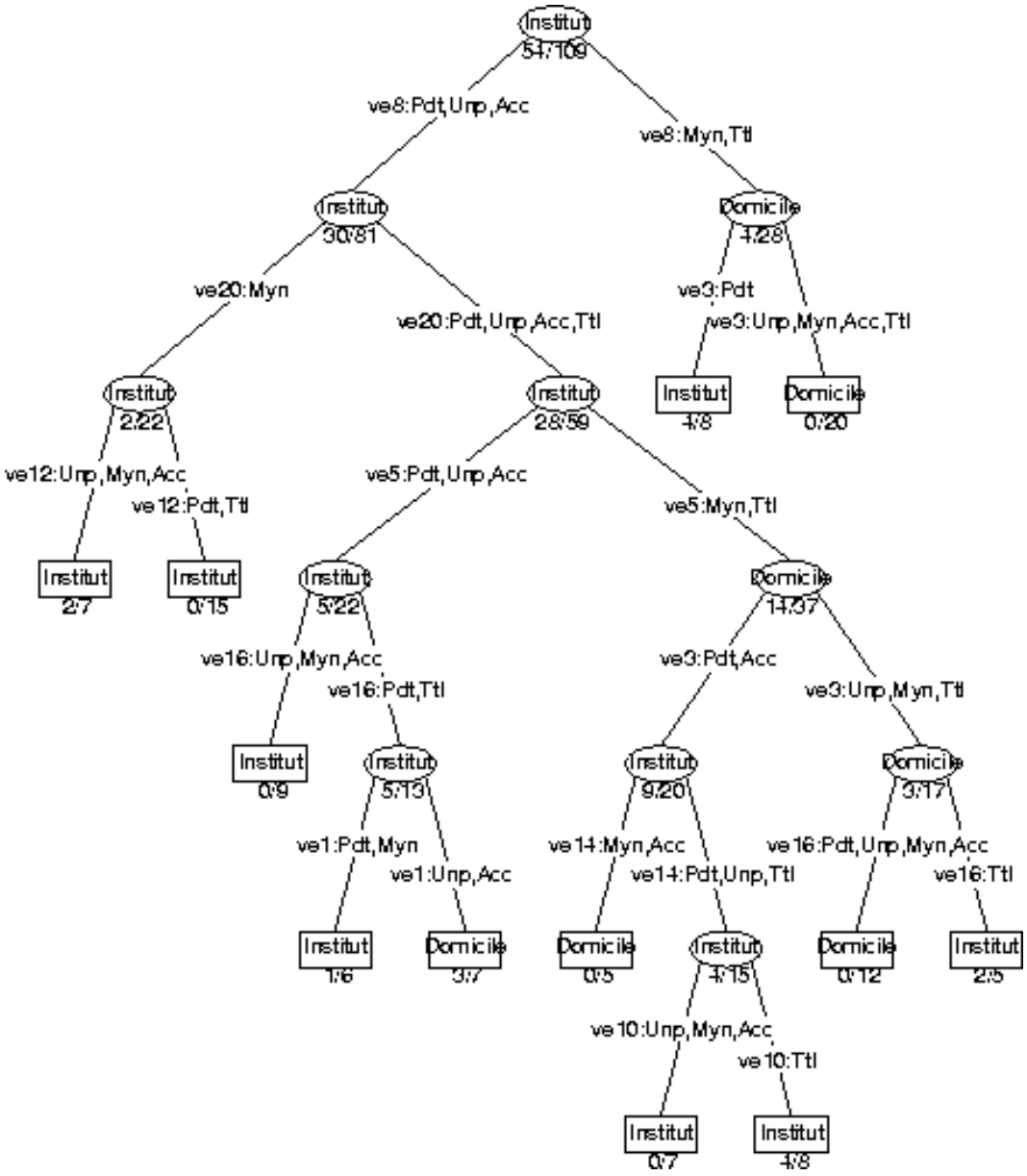
Ce traitement est réalisé à partir de données qui ont permis de construire la base de données sur laquelle nous avons travaillé jusqu'à présent.
Ce sont les données brutes du questionnaire: elles regroupent notamment des variables de stress (ve1,...,ve20). Pour chaque question (sous forme d'affirmation), les individus (toujours au nombre de 110) devaient choisir un item parmi 5 (pas du tout d'accord, un peu d'accord, ...).
Ces variables peuvent donc être considérées comme des variables qualitatives à 5 modalités.
Le but est d'expliquer toujours la variable lieu de vie par ces 20 variables en utilisant l'arbre de classification. Ce qui nous importe est d'examiner comment les variables explicatives à plus de 2 modalités sont gérées.

L'arbre de classification est le suivant :

```
node), split, n, deviance, yval, (yprob)
      * denotes terminal node

 1) root 109 151.100 1 ( 0.5046 0.49540 )
   2) ve8:1,2,4 81 106.800 1 ( 0.6296 0.37040 )
     4) ve20:3 22  13.400 1 ( 0.9091 0.09091 )
       8) ve12:2,3,4 7   8.376 1 ( 0.7143 0.28570 ) *
       9) ve12:1,5 15   0.000 1 ( 1.0000 0.00000 ) *
     5) ve20:1,2,4,5 59  81.640 1 ( 0.5254 0.47460 )
      10) ve5:1,2,4 22  23.580 1 ( 0.7727 0.22730 )
        20) ve16:2,3,4 9   0.000 1 ( 1.0000 0.00000 ) *
        21) ve16:1,5 13  17.320 1 ( 0.6154 0.38460 )
          42) ve1:2,4 6   5.407 1 ( 0.8333 0.16670 ) *
          43) ve1:3,5 7   9.561 2 ( 0.4286 0.57140 ) *
      11) ve5:3,5 37  49.080 2 ( 0.3784 0.62160 )
        22) ve3:1,4 20  27.530 1 ( 0.5500 0.45000 )
          44) ve14:3,4 5   0.000 2 ( 0.0000 1.00000 ) *
          45) ve14:1,2,5 15  17.400 1 ( 0.7333 0.26670 )
            90) ve10:2,3,4 7   0.000 1 ( 1.0000 0.00000 ) *
            91) ve10:5 8  11.090 1 ( 0.5000 0.50000 ) *
        23) ve3:2,3,5 17  15.840 2 ( 0.1765 0.82350 )
          46) ve16:1,2,3,4 12   0.000 2 ( 0.0000 1.00000 ) *
          47) ve16:5 5   6.730 1 ( 0.6000 0.40000 ) *
   3) ve8:3,5 28  22.970 2 ( 0.1429 0.85710 )
     6) ve3:1 8  11.090 1 ( 0.5000 0.50000 ) *
     7) ve3:2,3,4,5 20   0.000 2 ( 0.0000 1.00000 ) *
```

Il n'y a plus que 109 individus pris en compte. En effet, la variable ve12 contient une valeur manquante. Une solution pour contourner le problème des valeurs manquantes est de supprimer l'individu pour lequel une telle valeur a été détectée. C'est ce que nous avons fait ici; c'est pourquoi, il ne reste plus que 109 individus.
Chaque partition aboutit à une séparation en 2 classes. Il y a donc regroupement de certains niveaux de classes.
Ainsi, la première partition fait intervenir la variable ve8. A gauche, sont placés les individus ayant soit le niveau 1, soit le niveau 2, soit le niveau 4 (de ve8). Ils sont regroupés dans la classe 1.

A droite, sont regroupés (dans la classe 2, cette fois-ci) les individus ayant le niveau 3 ou le niveau 5 de ve8.
La représentation graphique de l'arbre est alors :



On voit bien comment ces variables qualitatives à 5 modalités sont "coupées" en 2. Il y a séparation des variables suivant des regroupements de niveaux.
Ce phénomène ne se remarque pas pour des variables qualitatives à 2 modalités. C'est pourquoi, nous avons traité cet exemple à titre illustratif.

Chapitre 7

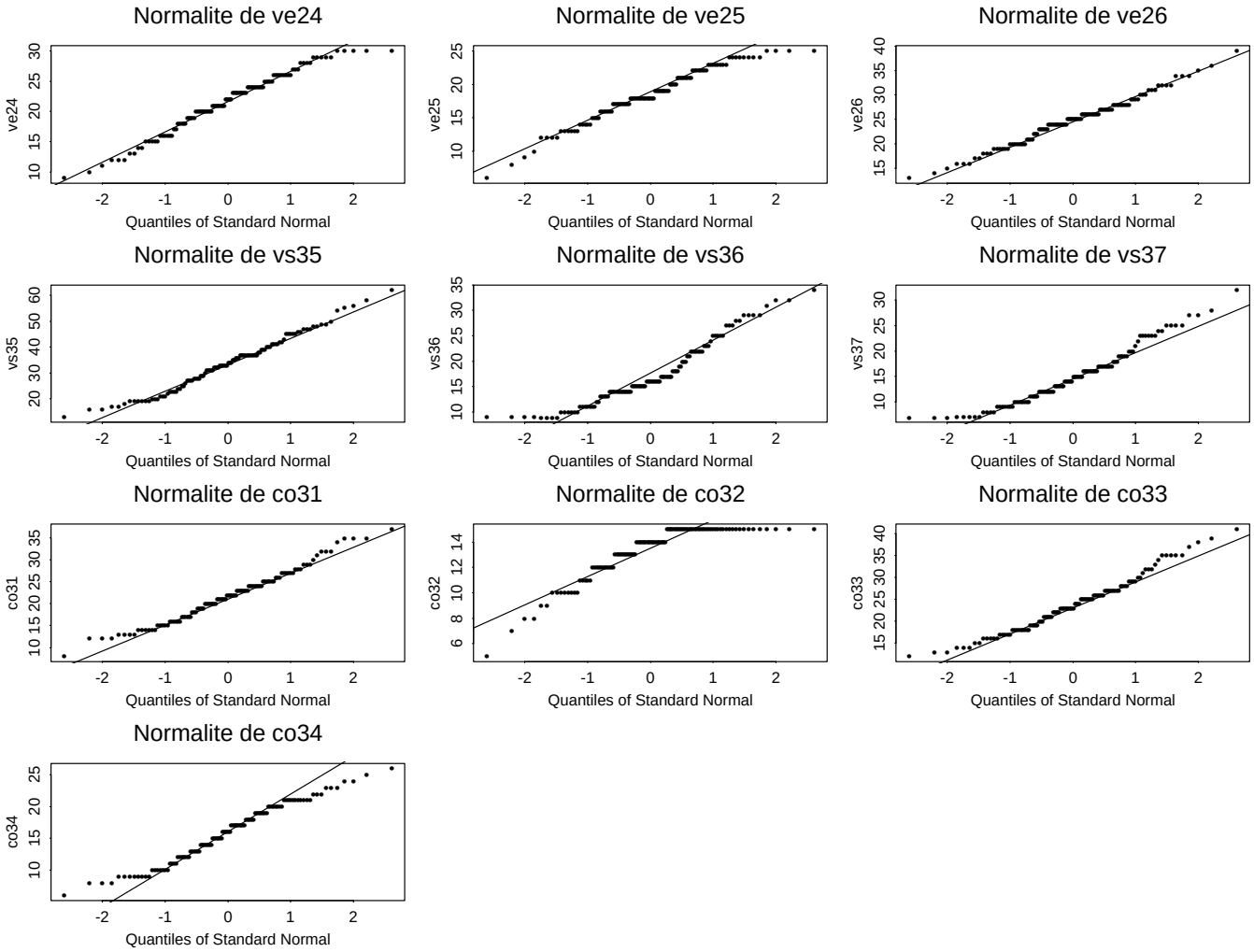
Conclusion

Ainsi, l'utilisation des arbres de classification a montré, dans un premier temps, que la variable lieu de vie était surtout sensible à l'âge, aux variables d'estime (ve26 et ve25) et aux variables de stress (vs37 et vs36). Les variables de coping (faire face) et la variable sexe interviennent à un moindre degrés.
Cependant, il semble que des variables concernant la vie quotidienne des personnes interrogées (factsou1 et factsou2) soient encore plus "explicatives" que l'âge ou que les variables de stress et d'estime. Nous avons expliqué ce phénomène par le lien logique entre ces 2 variables et la variable lieu de vie.
Enfin, le dernier chapitre (qui ne traite pas réellement du sujet de départ) nous a permis d'illustrer les arbres de régression (la variable dépendante étant quantitative) et les arbres de classification dans le cas où les variables indépendantes sont qualitatives à plus de 2 modalités.
Ces méthodes d'arbres de classification (et de régression) permettent une lisibilité facile des résultats (avec la possibilité de plus ou moins valider les résultats obtenus). Cela dit, l'utilisateur a un rôle actif pendant de tels traitements (notamment au niveau de l'élagage des arbres) : c'est lui qui doit prendre les décisions finales. Enfin, il nous a semblé que ces traitements devaient être considérés comme des traitements préalables à d'autres traitements (plans factoriels, ...), La mise en oeuvre de ces traitements (par exemple au niveau du choix des variables du modèle) découlerait alors des résultats obtenus grâce aux arbres de classification ou de régression.

Annexes

enquete

QQnorm et QQlines des 10 scores



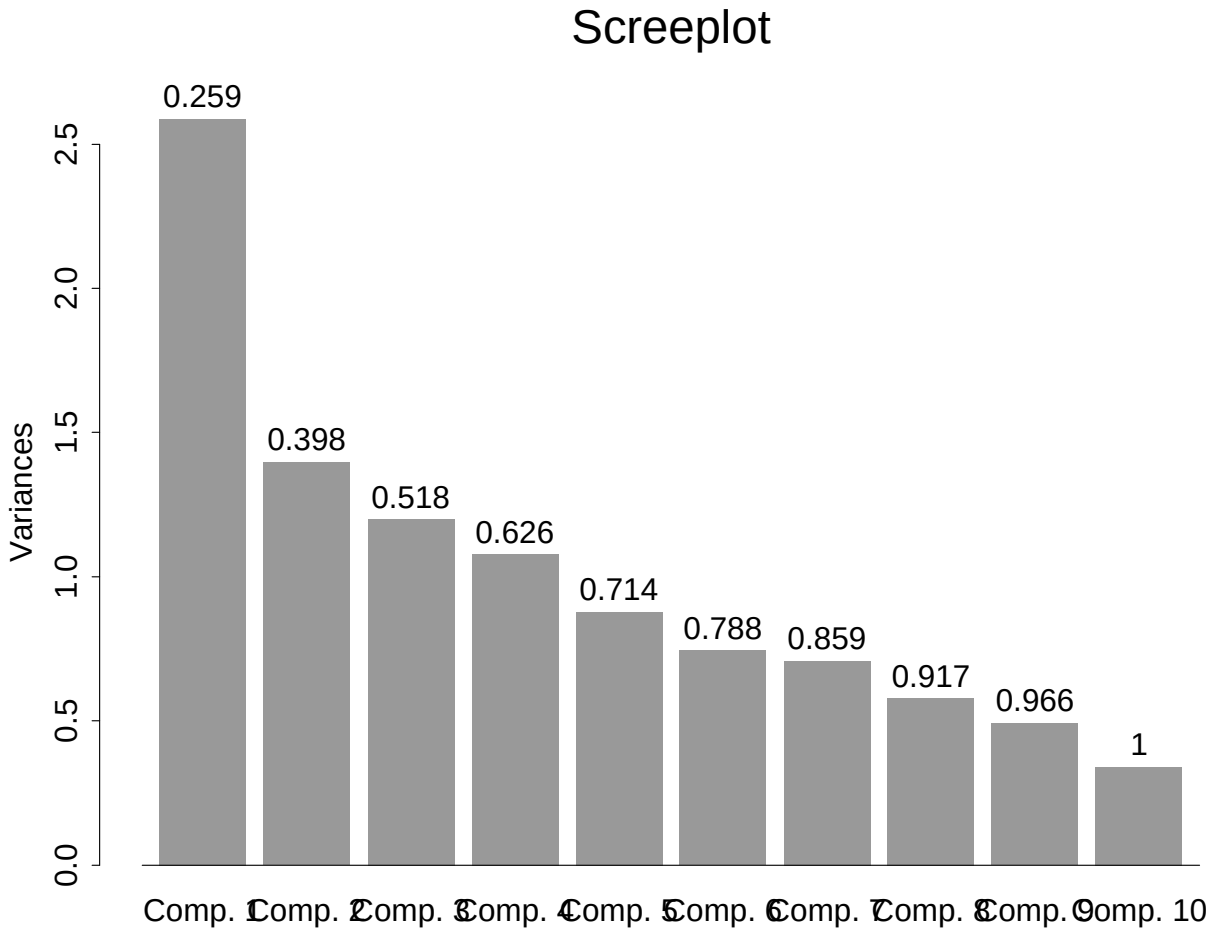
ACP centrée réduite des 10 scores

```
** vectors **

             f1          f2           f3          f4          f5          f6
ve24 -0.2986931 -0.04703535 -0.5013628924 -0.13022612 -0.1991143  0.60824590
ve25 -0.3449052  0.40158275 -0.2159395274 -0.13222002 -0.2811708 -0.30524807
ve26 -0.3149619  0.46904391  0.1517985567  0.16506636 -0.2210604  0.14733047
vs35  0.4470313  0.23202167 -0.3356030198  0.13200934 -0.2284924  0.03756388
vs36  0.3554062  0.40228104  0.0271076489 -0.25298182 -0.3511229  0.09994400
vs37  0.4506909 -0.01098724 -0.0009470122 -0.02932140  0.1515700  0.51489453
co31  0.0106758  0.54795358  0.4540020559  0.01804575  0.3822156  0.20650660
co32 -0.2168977  0.22671624 -0.4363200229 -0.05613312  0.6632602  0.07441308
co33  0.1211780  0.03271067 -0.0383005536 -0.86381626  0.1260718 -0.20401074
co34  0.3186005  0.21910866 -0.4094355857  0.32614569  0.1790592 -0.38579190
               f7          f8          f9         f10
ve24  0.330395219  0.11020186  0.31559741 -0.09863939
ve25  0.149961082  0.42156629 -0.50491274  0.16817533
ve26  0.007700062 -0.73023963 -0.12731297 -0.10205540
vs35 -0.135886860 -0.13686340  0.17253768  0.70405057
vs36 -0.442294740  0.21832256  0.15219120 -0.49738733
vs37  0.181997566 -0.02079788 -0.68633743 -0.05649584
co31  0.308769410  0.29804646  0.29385336  0.18421618
co32 -0.505459752 -0.04112354 -0.07264754 -0.01091930
co33  0.243924196 -0.33215890  0.05930785  0.07576282
co34  0.459076540 -0.09594914  0.10058552 -0.40682188


** Pourcentage de inertie expliquee **

            f1 f2 f3 f4 f5 tot
inertie exp 26 14 12 11  9  71
inertie cum 26 40 52 63 71  71


** Carre des correlations variables facteurs **

      f1  f2  f3  f4  f5 f>5  TOT
ve24 231   3 301  18  35 412 1000
ve25 308 225  56  19  69 323 1000
ve26 257 307  28  29  43 336 1000
vs35 517  75 135  19  46 208 1000
vs36 327 226   1  69 108 269 1000
vs37 526   0   0   1  20 453 1000
co31   0 419 247   0 128 205 1000
co32 122  72 228   3 386 189 1000
co33  38   1   2 804  14 141 1000
co34 263  67 201 115  28 326 1000
```

```
 TOT 259 140 120 108  88 286 1000
```

** Contribution des variables aux facteurs **

	f1	f2	f3	f4	f5	f>5	TOT	VARI
ve24	89	2	251	17	40	144	100	73
ve25	119	161	47	17	79	113	100	47
ve26	99	220	23	27	49	117	100	71
vs35	200	54	113	17	52	73	100	317
vs36	126	162	1	64	123	94	100	111
vs37	203	0	0	1	23	158	100	90
co31	0	300	206	0	146	71	100	102
co32	47	51	190	3	440	66	100	13
co33	15	1	1	746	16	49	100	114
co34	102	48	168	106	32	114	100	62
	1000	1000	1000	1000	1000	1000	1000	1000



Conversion de fichiers de données d'un logiciel à un autre

* SPSS (portable) ⇒ SAS

Soit le fichier portable SPSS fic.por. Le petit programme SAS qui transforme ce fichier en un fichier SAS est :

```
filename spssfile "fic.por";
proc convert spss=spssfile out=sasuser.fic;
run;
```

* SAS ⇒ Splus

Soient les données SAS sasuser.fic (le fichier fic.ssd01 doit exister dans le répertoire sasuser). La commande Splus transformant ces données en des données utilisables sous Splus est :

```
cf <- sas.get(library="$HOME/sasuser",member="fic")
```

Bibliographie

* *Modern applied statistics with S-Plus*, W.N. Venables, B.D. Ripley, SPRINGER-VERLAG, 1994

* *Classification And Regression Trees*, Breiman, Friedman, Olshen, Stone, WADSWORTH, 1984

* *Les arbres de régression*, J.R. Mathieu, cours universitaire, Toulouse (P. Sabatier)

* *Statistique exploratoire multidimensionnelle*, L. Lebart, A. Morineau, M. Piron, DUNOD, 1995

* *Analyse discriminante sur variables qualitatives*, G. Celeux, J.P. Nakache, POLYTECHNICA, 1994

Index

* Commandes de construction et de représentation

browser.tree : *browser.tree(nom-arbre,nodes,...)*
où "nodes" est un argument optionnel qui représente un vecteur contenant les numéros des noeuds qui intéressent l'utilisateur.

identify.tree : *identify.tree(nom-arbre,nodes,...)*

plot.tree : *plot.tree(nom-arbre,type="")*
où "type" est un argument optionnel. Si type="u", l'espace séparant chaque noeud est uniforme.

post.tree : *post.tree(nom-arbre,title=...)*
où "title" est un argument optionnel.

summary.tree : *summary.tree(nom-arbre)*

text.tree : *text.tree(nom-arbre,splits=T,label="yval",FUN=text,all=F,pretty=NULL)*
– où "splits" est un argument logique. Si sa valeur est T (par défaut), les partitions sont marquées.
– où "label" est un argument qui prend la valeur d'une colonne de l'objet "nom-arbre$frame". Il permet de donner telle ou telle valeur au noeud ("yval" par défaut). La valeur "dev" donne par exemple les déviances au niveau de chaque noeud.
– où "all" est un argument logique (par défaut all=F). Si sa valeur est T, tous les noeuds ont une valeur, sinon seuls les noeuds terminaux ont une valeur.
– où "pretty" est un argument qui par défaut prend la valeur NULL. Les niveaux du facteur sont alors représentés par des lettres. Cet argument reçoit une valeur entière. S'il prend la valeur 0, les niveaux du facteur sont représentés tels quels. Si sa valeur est 1, les niveaux du facteur sont représentés par leur initiale. Au delà de 1, le logiciel sélectionne autant de consonnes appartenant au label du niveau du facteur, que "pretty" l'indique.

tree : *tree(formula,data=nom-fichier,na.action=na.fail)*
– où "formula" est un argument qui correspond au modèle.
Ex. : $v1\ v2 + v3 + v4$ (la variable v1 est expliquée par v2, v3, v4)
– où "data" est un argument qui indique le fichier de données à partir duquel l'arbre est construit.
– où "na.action" est l'argument qui traite des valeurs manquantes. Par défaut, sa valeur est na.fail; ce qui crée un message d'erreur si une valeur manquante est détectée. Une alternative est de choisir l'argument "na.omit" qui supprime les observations qui contiennent une valeur manquante (ou plus).

* Commandes d'élagage

cv.tree : *cv.tree(nom-arbre,rand,FUN=...)*
– où "rand" est un vecteur d'entiers de longueur égale au nombre d'observations dans le noeud "sommet" de l'arbre. Par défaut, Splus crée un vecteur aléatoire contenant 10 éléments distincts (de 1 à 10). Autrement dit, il fait de la validation croisée d'ordre 10.
– où "FUN" est égal au nom de la fonction qui produit un objet de type séquence d'arbre.

prune.tree : *prune.tree(nom-arbre,k=...,best=...,method=...)*
– où "k" est le paramètre du coût-complexité
– où "best" est un entier indiquant la taille de l'arbre (en fait le nombre de noeuds terminaux)
– où "method" représente la méthode utilisée pour mesurer l'hétérogénéité des noeuds. Par défaut c'est la déviance (seul choix possible pour les arbres de régression). Pour les arbres de classification, la méthode misclass est une alternative.

* Commandes de transformation des variables

factor : *factor(x,levels=...,labels=...)*
– où "x" est le vecteur à transformer en facteur
– où "levels" correspond à un vecteur des niveaux du facteur
– où "labels" correspond à un vecteur des appellations que l'on veut donner aux niveaux du facteur.
Ex. : soit x un vecteur numérique correspondant au sexe (possédant des 1 et des 2). La commande suivante transforme x en facteur de niveaux "Homme" et "Femme" :

$$x\_factor(x, levels = c("1","2"), labels = c("Homme","Femme"))$$

as.numeric et **as.vector** : ces commandes retournent des objets de type numerique (vecteur de mode numerique) pour la première et des objets de type vecteur pour la seconde.
Ex. : $x\_as.numeric(x)$ et $x\_as.vector(x)$