MASTER EN SCIENCES INFORMATIQUES

Analyse de données de quartiers urbains

Delperdange, Marc

Award date:
2000

Facultés Universitaires
Notre-Dame de la Paix
Namur
Institut d'Informatique

Analyse de données de quartiers urbains

Marc DELPERDANGE

Mémoire présenté en vue de l'obtention du grade de Licencié en Informatique

Promoteur : M. Noirhomme

Année académique 1999 - 2000

Résumé

Des données concernant les quartiers urbains de Namur ont été récoltées. Le mémoire consiste en l'analyse de ces données à l'aide de différentes méthodes et de logiciels spécialisés, en particulier de l'étoile zoom. Les différents outils, qui sont utilisés pour faire ces analyses sont présentés, et leurs fonctionnalités les plus importantes sont expliquées.

Une deuxième partie de ce travail porte sur le développement d'un logiciel. Ce logiciel crée une liaison entre deux moyens différents de visualiser des données, à savoir la représentation avec l'étoile zoom et une représentation sur cartes géographiques.

Abstract

Information has been gathered about a number of urban districts of Namur. This essay will make an analysis of those figures with the help of some specialized methods and programs, and in particular with the zoom star. The programs that are used to make those analyses will be introduced first, and an explanation of the most important functions will be given.

A second part of this work is devoted to the development of a program. This program will make a connection between two different ways of visualizing data, namely the representation with the zoom star and a representation on geographical maps.

Remerciements

J'aimerais remercier tous ceux qui, de près ou de loin, m'ont aidé à réaliser ce mémoire.

Ces remerciements sont adressés d'abord à mon promoteur, le professeur Monique NOIRHOMME. Je la remercie pour la manière dont elle m'a accompagné dans ce travail, pour ses conseils et l'aide précieuse qu'elle m'a fournie tout au long de la réalisation de ce mémoire. Je remercie aussi Madame An DE BAENST pour les différents renseignements qu'elle m'a donnés.

Je remercie également Madame BOUDART, des services administratifs de la ville de Namur, pour les renseignements qu'elle m'a communiqués, et les explications fournies.

Je remercie Françoise ORBAN et Jehan DECROP, du Département de Géographie des Facultés Universitaires Notre-Dame de la Paix, qui m'ont aidé à comprendre les aspects cartographiques traités dans ce mémoire.

Je remercie finalement mes parents. Ils m'ont permis d'entreprendre ces études, et m'ont donné le soutien moral et financier indispensable pour les mener à terme.

TABLE DES MATIÈRES

Introduction

Ces deux dernières décennies ont été caractérisées par une augmentation drastique de la quantité d'informations accessibles ou de données stockées sous forme électronique. Cette croissance continue à croître à une vitesse explosive. On estime en effet, que la quantité d'information dans le monde double tous les 20 mois. Cette croissance est supportée par l'évolution des techniques dans le domaine de la collection et du stockage des données. La collecte des données est facilitée par un nombre croissant de moyens électroniques reliés entre eux par des réseaux de communications de plus en plus performants. Le stockage et le traitement des données deviennent de plus en plus facile grâce à la puissance augmentée des ordinateurs.

Les données qui sont collectées de cette façon sont certainement des ressources précieuses. Elles permettent par exemple aux entreprises de disposer d'un grand nombre de données concernant leurs ventes, leurs clients, le cycle de production de leurs produite. Les systèmes traditionnels pour traiter des données électroniquement conviennent parfaitement pour remplir des bases de données de manière rapide, efficace et sans risque. Mais à cause de l'augmentation du nombre de données disponibles, ces systèmes ont de plus en plus de difficultés pour fournir des analyses significatives de ces données.

C'est là que les techniques et les analyses du Data Mining sont apparues. Dans la littérature, on trouve plusieurs définitions de Data Mining. Elles mentionnent presque toutes la puissance augmentée des ordinateurs et les techniques d'analyse avancées nécessaires pour découvrir des relations utiles dans les grandes bases de données. La définition de Data Mining, que nous proposons ici, a été dérivée de définitions proposées par Fayyad, Berry et d'autres encore [NOIRHOMME99] :

> *' Data mining is the exploration and analysis, by automatic and semiautomatic means, of huge databases for identifying valid, novel, potentially useful, and ultimately understandable correlation, patterns and trends in data. '*

La visualisation de données complexes est également un domaine de recherche qui a gagné beaucoup en popularité ces dernières années. Vu la taille des bases de données et les instruments de calculs de plus en plus importants, les relations étudiées sont de plus en plus complexes et difficiles à exprimer. D'où l'intérêt croissant pour des méthodes de visualisation ou de présentation nouvelles, permettant de synthétiser en quelques dessins ou graphiques les résultats d'une analyse. Une tendance linéaire dans une banque de données, par exemple, peut être difficile à voir dans un tableau de données, alors qu'un graphique simple permet déjà une compréhension immédiate de cette relation. Cela est vrai à fortiori lorsque les relations étudiées deviennent complexes. Dans ce cas, l'utilisation d'outils de visualisation permet souvent à l'utilisateur de remarquer des anomalies dans les données, ou de découvrir des relations significatives nouvelles, qui lui permettent d'orienter ses analyses dans de nouvelles directions.

Nous disposons actuellement d'outils de visualisation qui vont plus loin que les limites des graphiques à trois dimensions. Ces représentations utilisent en général d'autres caractéristiques comme des couleurs, la forme des points et même des angles pour exprimer les informations des autres dimensions.

Dans ce travail, nous expliquerons dans la première partie quelques outils de visualisation existants. Il s'agit du logiciel SODAS, qui a été développé dans le cadre du projet SODAS. Un de ces modules sera analysé plus en détails, à savoir le SOEditor. Ces programmes sont basés sur un type de données élargi, les données symboliques, qui vont également être présentées dans le premier chapitre. A la fin du premier chapitre, un autre outil de visualisation, basé sur des données géographiques, sera également expliqué.

La deuxième partie de ce travail comporte une explication des différentes parties du programme que nous avons crée. On essaiera d'abord d'effectuer l'analyse en spécifiant, et en l'axant sur les utilisateurs futurs, et en se basant sur quelques paramètres descriptifs. Ensuite, une interface sera proposé à l'aide de quelques images pour montrer le fonctionnement du programme.

Le troisième chapitre sera surtout une description de la phase d'implémentation du programme. On donnera quelques détails sur les différentes classes du programme et une explication de son fonctionnement. Les problèmes rencontrés dans la phase de l'implémentation seront également mentionnés.

Finalement, le quatrième chapitre montrera l'analyse des données de certains quartiers urbains de Namur. A l'aide d'une subdivision en différentes régions, on essaiera de trouver les relations possibles entre les données. Les logiciels d'analyse qui sont utilisés dans cette partie correspondent à ceux qui auront été expliqués dans la première partie. Les analyses seront donc faites à l'aide du logiciel SODAS et le module SOEditor, et seront comparées avec des analyses faites au moyen du logiciel Mapinfo, développé spécialement pour effectuer des analyses sur des données cartographiques. Chacune de ces analyses sera illustrée au moyen de représentations graphiques pour faciliter l'analyse et pour donner une meilleure visualisation des résultats.

1. Description de l'existant

Dans cette première partie du texte, la notion "d'analyse symbolique", basée sur les données symboliques et les objets symboliques, sera d'abord expliquée brièvement. Après cela, une description du projet et du logiciel SODAS sera donnée. La description du logiciel SODAS se concentrera sur le module SOEditor qui sera utilisé dans notre programme.

SODAS est un logiciel construit dans le cadre du projet SODAS, pour analyser et pour évaluer des données symboliques et des objets symboliques. Le module SOEditor (Symbolic Objects Editor) est un moyen de visualiser ces objets symboliques. Des représentations graphiques aident en effet beaucoup à découvrir les relations qui existent entre les données et à trouver les informations intéressantes dans les grandes bases de données.

1.1. Analyse symbolique

1.1.1. Données symboliques

Classiquement, les données sont rangées dans un tableau de n x m éléments. La matrice de données est donc constituée de la relation de n individus avec un certain nombre de variables $Y_1, \ldots, Y_m$. Chaque élément de la matrice de données contient une seule valeur. Parfois cependant, les relations sont trop complexes pour être représentées dans un tableau simple. Dans ce cas, il devient nécessaire de représenter les variables par des données symboliques, on pourrait les appeler dès lors des variables symboliques.

Des variables symboliques peuvent prendre différentes formes. Les trois types les plus fréquents sont les variables multi-valuées, les variables intervalles et les variables modales.

Une variable est une variable multi-valuée, si chaque élément de la matrice de données correspondant à cette variable est représenté par plusieurs données. On peut encore faire une distinction entre des variables **multi-valuées catégorielles** et des variables **multi-**

valuées quantitatives. On parle de variables **catégorielles** quand les données proviennent d'un nombre fini de catégories. Les différentes catégories, pour un élément de la matrice de données, peuvent par exemple être des couleurs, Y = {rouge, vert, bleu, jaune, noir} ou encore Y = {matin, après-midi, soir, nuit}. On parle par contre de variables **quantitatives** quand les données sont des nombres réels. On pourrait par exemple représenter les salaires de différents individus par Y(k) = {36.000, 65.000, 29.000, 46.000, 51.000}.

Une variable est une variable intervalle, si sa valeur s'exprime sous forme d'un intervalle. Un exemple d'une variable intervalle est la moyenne d'âge d'une population qui peut avoir un résultat de Y(k) = [35, 40].

Les variables modales sont des variables pour lesquelles il existe différentes modalités pour chaque élément de la matrice de données. On ne donne pas seulement toutes les catégories possibles, mais pour chaque élément on donne aussi le poids ou la probabilité de toutes ces catégories. L'interprétation des différents poids dépend du modèle et pourra porter sur des probabilités, des nombres, des fréquences, des possibilités, ... Si on regarde par exemple la proportion des enfants dans le genre d'enseignement suivi pour une certaine région, on pourrait trouver une distribution de Y(k) = ((primaire , 0.35) , (secondaire , 0.45) , (supérieur , 0.2)).

Dans SODAS, on utilise aussi des types de données plus générales comme les relations de dépendance et les relations hiérarchiques (taxonomies). Deux variables sont dans une relation de dépendance si une des catégories d'une variable catégorielle est elle-même réutilisée pour une autre variable. Une variable peut n'avoir de sens que si une autre variable catégorielle prend telle ou telle valeur. Par exemple la variable enceinte oui/non n'a de sens que si la variable sexe prend la valeur "femme".

Dans le cas de statistiques sur les accidents de la route, il se peut que la valeur de la variable vitesse du véhicule ne soit apparente que si la cause de l'accident est la vitesse. La modalité "vitesse" de la variable qui explique la cause de l'accident peut alors elle-même être représentée par un histogramme ou comme une variable intervalle si la vitesse était la cause. Dans le cas où elle est représentée par une variable intervalle, cette variable donne alors l'intervalle de confiance dans lequel 95 % des individus seront

situés (par exemple [120, 150]). Dans ces deux cas, les variables sont dépendantes. Il s'agit donc d'une dépendance logique et non d'une corrélation.

On parle d'une relation hiérarchique si toutes les catégories d'une variable peuvent être représentées comme une hiérarchie ou une échelle de valeurs. Une variable avec une découpe hiérarchique exige souvent une connaissance a priori sur les valeurs possibles. La variable "âge" peut par exemple être découpée en deux catégories "mineur" et "adulte". Chacune des deux catégories peut encore être divisée pour arriver aux feuilles de l'arbre qui représentent les différents groupes d'âges. L'image 1.1 ci-dessous montre une relation hiérarchique de la variable "âge".

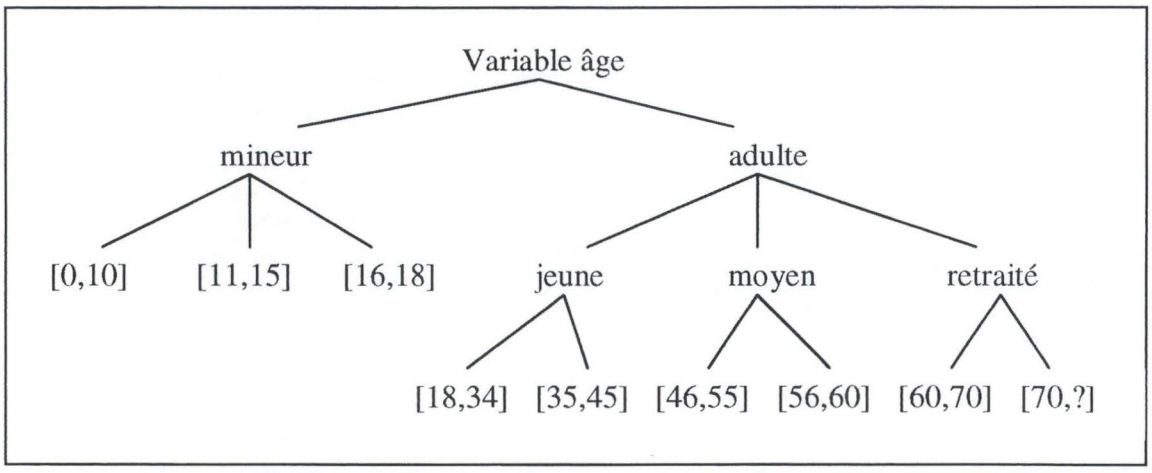


IMAGE 1.1 : Exemple d'une représentation hiérarchique.
Source : European Commission Directorate General III

1.1.2. Objets symboliques

Le concept "objets symboliques" a été introduit par E. Diday [DIDAY93] et est utilisé dans un sens beaucoup plus général que les objets dans les langages de programmation. Un objet symbolique ne correspond pas à un objet réel qui est décrit par des variables symboliques. Un objet symbolique est une description d'un ensemble d'individus ou d'un individu pour lequel on n'a pas de connaissance certaine des variables le décrivant. L'exemple [âge > 40] $\wedge$ [sexe = féminin] représente la catégorie des femmes âgées de plus de 40 ans.

Le but principal de l'approche symbolique dans l'analyse de données est d'étendre les problèmes, les méthodes et les algorithmes utilisés sur les données standards. Ils vont

être étendus vers des données plus complexes, de manière à être capables de représenter une connaissance, où les unités sont appelées les objets symboliques.

D'un point de vue informatique, un problème fondamental est de trouver tous les éléments dans une base de données qui remplissent certaines conditions par rapport aux variables symboliques. Il y a différentes possibilités pour spécifier ces exigences. Un objet symbolique donne une spécification formelle pour pouvoir trouver les éléments qui remplissent ces conditions à l'aide d'une fonction. Dans le cas d'un booléen, ce sera une fonction binaire qui exprime si la condition est satisfaite ou pas. Dans le cas de variables symboliques modaux, ce sera une fonction qui spécifie jusqu'à quel degré la condition est satisfaite.

[DIDAY93] présente plusieurs structures de données pour les objets symboliques. D'abord il y a les événements élémentaires booléens qui sont les éléments de base dans la construction des objets symboliques. Ils aident à exprimer le fait qu'un objet appartient à une des catégories d'une liste sans préciser de laquelle il s'agit exactement. On peut par exemple exprimer qu'une voiture appartient à l'une des marques suivantes, sans que l'on précise laquelle : Peugeot, Opel, Ford, Fiat. En outre, il y a aussi un moyen de faire une conjonction entre les différents événements élémentaires booléens pour obtenir des objets assertions booléens. Cela permet d'exprimer plusieurs conditions en même temps dans une seule expression. On peut par exemple exprimer qu'une voiture n'appartient pas seulement à une des marques, précisées ci-avant, mais aussi que sa couleur est rouge, verte ou noire.

Mais les objets booléens sont quand même assez restreints. On ne peut exprimer que deux valeurs : vrai ou faux (ou dans l'exemple précédent : l'appartenance ou non à une liste de valeurs possibles). Une représentation booléenne est insuffisante pour une description acceptable de la réalité multidimensionnelle.

A côté de ces types booléens, il existe alors également d'autres types comme des objets symboliques modaux. Ces objets modaux permettent de décrire plusieurs modes sur les événements élémentaires pour avoir des jugements plus nuancés. Ainsi, dans le cas de la sémantique de la certitude, on peut par exemple avoir les quatre modes suivantes : il est

absolument vrai, il est vrai sous conditions, il est faux sous conditions, il est absolument faux.

Dans les objets symboliques modaux, il existe plusieurs variétés. Les objets possibilistes expriment l'aptitude à réaliser une tâche (matérielle ou par rapport aux connaissances). Les objets probabilistes par contre modélisent plusieurs sortes de connaissances, comme la chance, la fréquence ou l'incertitude. L'origine des probabilités de chance se situe dans les calculs des chances au jeu. Pour la fréquence, la probabilité d'un événement est la limite de la fréquence de ce résultat lorsqu'on répète l'expérience un grand nombre de fois. L'incertitude mesure un degré de certitude d'un événement qui ne se passe qu'une seule fois.

On peut également faire des conjonctions entre les différents événements élémentaires modaux. Les assertions sont donc des conjonctions d'événements élémentaires booléens et/ou modaux (probabilistes).

[MATHOT97] donne une définition formelle des objets symboliques. Dans ce qui suit, on va reprendre la définition qu'elle donne d'un objet modal de l'intérieur. Un objet modal de l'intérieur se distingue d'un objet modal de l'extérieur parce que les différents modes ne portent pas sur l'événement élémentaire tout entier, mais simplement sur l'ensemble de valeurs de la variable qui décrit l'événement.

On commence la définition à partir de quelques hypothèses :

- M^x est un ensemble de noms ou de nombres qui traduit les modes associés à une sémantique notée x. On a par exemple $M^x = [0,1]$ ou $M^x = \{$jamais, rarement, parfois, souvent, toujours$\}$.
- $Q_i = \{q_i^j\}_j$[2] est l'ensemble des applications $q_i^j : Q_i \rightarrow M^x$.
- O P_x désigne les trois opérations 'ensemblistes' définies dans Q_i : les opérations d'union $\cup_x$, les opérations d'intersection $\cap_x$, et les opérations de complémentation c_x.
- g_x est une application de comparaison : $g_x : Q_i \times Q_i \rightarrow L^x$ où L^x est l'espace d'interprétation qui est ordonné et parfois identique à M^x.

- f_x est une application symétrique d'agrégation : $f_x : P(L^x) \rightarrow L^x$ où $P(L^x)$ sont les parties de L^x.
- $Y = \{y_i\}$ est un ensemble des descripteurs et $V = \{V_i\} = \{(q_i^j)\}_j \subseteq Q_i$ est un ensemble de parties V_i de Q_i.

La définition d'un objet modal de l'intérieur est alors :

Etant donnés O P_x, g_x, f_x, *un objet modal de l'intérieur est une application* $a_{YV} : \Omega \rightarrow L^x$ *notée* $a_{YV} = \wedge_{i,x}[y_i = \{q_i^j\}_j]$ *tel que si* $\omega \in \Omega$ *est décrit pour chaque i par* $y_i(\omega) = \{r_i^j\}$ *alors,*

$$a_{YV}(\omega) = f_x(\{g_x(\cup_{j,x} q_i^j, \cup_{j,x} r_i^j)\}_i).$$

[BOCK&00] donne une autre définition des objets symboliques. Dans ce qui suit, on va reprendre la définition formelle qu'il donne pour les objets symboliques généraux.

Un objet symbolique est constitué de différents événements du type :

$$[Y_j R_j z_j] \quad \text{ou} \quad [Y_j R_j D_j]$$

par exemple [longueur = 100] ou [couleur ∈ {rouge,vert}]

avec Y_j = les variables définies pour les individus (j = 1, …, p)

R_j = la relation du produit cartésien

z_j = un vecteur de description d'une variable Y_j

D_j = un système de description d'une variable Y_j

Une combinaison de vecteurs et de systèmes de description sera appelée une description d'une variable.

Un objet symbolique est alors une conjonction d'événements analogues :

$$q = \bigwedge_{v=1}^{r} [Y_{jv} R_{jv} z_v] \quad \text{ou} \quad q = \bigwedge_{v=1}^{r} [Y_{jv} R_{jv} D_v]$$

Par exemple : q = [longueur ≤ 100] ∧ [longueur ≥ 50] ∧ [couleur ∈ {rouge,vert}].

L'exemple suivant peut préciser ces concepts. Le tableau 1.1 sur la page suivante contient la matrice de données. Les variables Y_1 et Y_2 sont des variables quantitatives avec un domaine = R_+, la variable Y_3 est une variable catégorielle avec un domaine =

{Eco, Info, Math, Med} et la variable Y_4 est une variable binaire avec un domaine = {homme, femme} = {0, 1}.

Personne	Salaire Y_1	Age Y_2	Etudes Y_3	Sexe Y_4
Individu 1	36.000	25	Eco	1
Individu 2	65.000	29	Info	0
Individu 3	29.000	34	Math	1
Individu 4	46.000	38	Med	0
Individu 5	51.000	45	Med	1

TABLEAU 1.1 : Matrice de données.
Source : Bock (1999)

Si maintenant on était intéressé de trouver tous les individus dans la base de données avec un salaire au dessus de 45.000 et ayant suivi des études de médecine, on peut formuler une question de la forme :

$$q_1 = [Y_1 > 45.000] \wedge [Y_3 = \text{Med}]$$

On obtiendra comme réponse les individus 4 et 5 qui remplissent cette condition. On pourrait donc dire que q_1 définit l'objet symbolique: "une personne avec un salaire au dessus de 45.000 qui a suivi des études de médecine". La relation q_1 va donc faire une comparaison entre les variables Y_1 , Y_3 et les valeurs de norme 45.000 et Med, normes qui sont souvent spécifiées par l'utilisateur.

De la même manière, $q_2 = [Y_1 < 40.000] \wedge [Y_2 < 35] \wedge [Y_4 = 1]$ donne une description de l'objet symbolique: "une femme avec un salaire en dessous de 40.000 et qui a moins de 35 ans" donne les individus 1 et 3 comme réponse.

Les objets symboliques sont introduits parce qu'il est parfois impossible de décrire des systèmes de données complexes avec les modèles classiques. C'est ce concept généralisé des objets symboliques qui est l'élément de base du logiciel SODAS et du module SOEditor.

1.1.3. Analyse symbolique

Les méthodes classiques d'analyse de données statistiques travaillent avec un tableau de données d'individus pour certaines variables. Comme mentionné dans l'explication des données symboliques, chaque cellule de ce tableau contient une valeur atomique, dans le sens que ce n'est pas une liste ou plusieurs valeurs.

Dans les méthodes d'analyse symbolique introduites par E. Diday, les entrées sont également représentées par un tableau d'individus et de variables, mais elles sont étendues. Chaque cellule du tableau de données peut être non-atomique. Les données pourraient par exemple prendre le forme d'un intervalle de valeurs, d'une distribution de probabilités ou d'une liste de valeurs. Avant, on avait par exemple une variable 'âge moyen' pour représenter l'âge d'une population. Avec les nouvelles méthodes symboliques, on peut représenter cela avec une variable sous la forme d'un intervalle ou des probabilités.

Au fond, dans le monde réel, les données sont toujours complexes. L'utilisation de méthodes standards pour faire des analyses de données nécessite une transformation de ces données pour les faire correspondre au format des entrées demandées. Cette phase de transformation amène toujours une perte importante d'information et cela montre l'utilité des analyses symboliques.

[MATHOT97] distingue quatre types d'analyse. Le tableau 1.2 ci-dessous montre ces différents types. Une distinction est faite entre les données et les analyses qui peuvent toutes les deux avoir un caractère symbolique ou pas.

	Données classiques	**Données symboliques**
Analyse numérique (classique)	Type 1	Type 2
Analyse symbolique	Type 3	Type 4

TABLEAU 1.2 : Différents types d'analyse.
Source : Mathot (1997)

Une analyse de type 1 signifie l'utilisation des outils statistiques usuels et l'algèbre linéaire sur des données quantitatives et qualitatives. L'analyse de type 2 sera faite sur des données symboliques. On pourrait par exemple calculer une distance entre différents objets symboliques pour faire une classification ou encore une analyse en composantes principales. Les types 3 et 4 utilisent les méthodes de l'analyse symbolique. De cette façon, il serait possible de tenir compte des connaissances supplémentaires si on travaille avec des données symboliques.

1.2. Le projet SODAS

Le but du projet SODAS (Symbolic Official Data Analysis System) est de faciliter l'emploi de techniques d'analyse de données symboliques. Ils veulent montrer ainsi aux instituts de statistiques et aux entreprises dans les différents pays que ces nouvelles méthodes sont utiles pour eux et utilisables en pratique. Ils veulent également utiliser le feed-back des applications réelles pour améliorer la précision des méthodes d'analyse de données symboliques.

Les données et les méthodes pour analyser les objets symboliques proviennent de différents pays d'Europe. Le projet SODAS a été développé par différents instituts de statistiques, d'entreprises industrielles et d'universités d'Europe. La liste de tous les partenaires du projet est reproduite en Annexe 1.

Les instituts nationaux de statistique (mais aussi quelques entreprises industrielles) ont des bases de données très larges mais rencontrent souvent des problèmes lorsqu'ils veulent les exploiter. Les différents instituts de statistique travaillent presque toujours avec des concepts équivalents, comme par exemple le chômage ou les accidents de route. Les objets symboliques peuvent résoudre le problème qui naît quand ces concepts sont décrits par des variables différentes ou des noms officiels différents. Les objets symboliques offrent également un moyen ergonomique de présenter les données quand elles doivent être diffusées, une des tâches principales des instituts de statistique.

Différents moyens ont été mis en oeuvre pour atteindre ces buts. D'abord, un logiciel a été développé, afin de pouvoir faire des analyses sur des données symboliques. Le

résultat est le logiciel SODAS, qui est expliqué plus en détails dans le prochain paragraphe. D'autres moyens pour atteindre ces buts ont été la construction de systèmes qui étaient orientés concepts pour les statistiques officielles, ou encore la stipulation d'une approche avec points de référence proposés par les instituts de statistiques et les entreprises industrielles.

En même temps que le développement des méthodes d'analyse de données, le problème de la stabilité des méthodes était examiné, avec vérification de leur robustesse. La validation de l'approche sera faite avec des points de référence des instituts de statistiques.

1.3. Description du logiciel SODAS

1.3.1. Exigences d'un logiciel d'analyse de données symboliques

Avec un logiciel standardisé pour faire leurs analyses, les instituts de statistiques des différents pays pourraient facilement s'échanger leurs données entre eux. Dans le projet SODAS, différentes exigences ont été spécifiées. L'environnement du logiciel devrait contenir :

- Des outils pour stocker, interroger et faire des mises à jour des objets symboliques qui seront les entrées et les sorties des différentes méthodes d'analyse. Ces objets symboliques doivent former la structure de base pour représenter les données complexes.
- Des outils pour extraire les objets symboliques. Cela peut être réalisé ou bien directement par des experts, ou bien à partir des grandes bases de données statistiques existantes.
- Une collection de méthodes d'analyse de données permettant de faire des analyses avec des objets symboliques. Parmi ces méthodes d'analyse, on doit par exemple trouver des méthodes de classification, des méthodes de construction d'arbres de décision, d'analyse discriminante, d'analyse factorielle, des méthodes descriptives, ...
- Des possibilités de transformer les objets symboliques en objets 'standards' et d'exécuter des méthodes d'analyse 'standards'.

- Des outils ergonomiques pour présenter les résultats des méthodes aux utilisateurs.

1.3.2. Explication du logiciel SODAS

1.3.2.1. Chaîne de méthodes

La description du logiciel SODAS est basée sur la version 1.032 de ce logiciel. L'élément de base pour faire des analyses en SODAS est la chaîne de méthodes. Une chaîne est une suite de méthodes d'analyse appliquée aux bases de données numériques. Les méthodes sont des modules statistiques avec des paramètres de configuration qui peuvent être spécifiés par l'utilisateur. Un paramètre que l'utilisateur peut définir est le choix des différentes variables qu'il veut inclure dans l'analyse. D'autres paramètres spécifient le fonctionnement des différentes méthodes ou le format des résultats des méthodes.

L'ordre d'exécution d'une suite de méthodes d'une chaîne est de haut en bas. L'éditeur des chaînes permet la création, l'exécution, la suppression ou la modification d'une chaîne. Au lieu de devoir créer le même genre de suite de méthodes à chaque fois, il est possible d'enregistrer cette suite comme un modèle de chaîne. A noter que bien que le mot chaîne soit utilisé, les différentes analyses s'exécutent sur le même fichier de base et non sur la sortie de la méthode précédente.

L'image 1.2 sur la page suivante montre un exemple de la fenêtre de l'éditeur des chaînes. On voit une suite de trois méthodes d'analyse, à savoir DIV (classification divisive), SOE (représentation graphique des objets symboliques avec l'étoile zoom) et STAT (méthodes de statistique élémentaire). Si on double-clique sur l'icône du livre à côté de chaque méthode, un fichier est ouvert en WordPad avec plus d'information sur la méthode. En général, ce fichier contient plus de détails sur l'exécution et sur les résultats de cette méthode. Au cas où il y aurait eu des problèmes pendant l'exécution des différentes méthodes, les messages d'erreurs qui sont inscrits dans ce fichier peuvent aider l'utilisateur à trouver les erreurs.

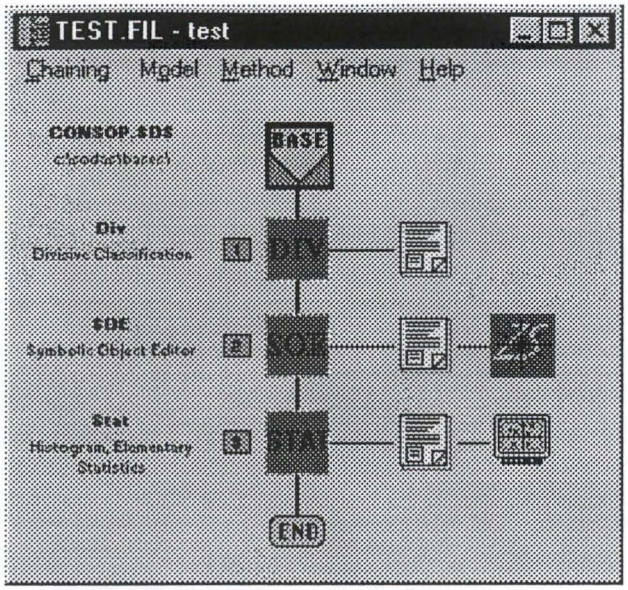


IMAGE 1.2 : Editeur de chaînes.
Source : SODAS (1998)

Si une autre icône apparaît encore à côté, cela signifie que la méthode a aussi généré une représentation graphique. C'est le cas dans notre exemple pour les méthodes SOE et STAT. Si on double-clique sur l'icône du graphique, on est renvoyé vers d'autres modules pour les représentations graphiques, comme par exemple le SOEditor pour la méthode SOE.

1.3.2.2. Type de données

Le format des fichiers qui est utilisé dans SODAS est le format ".sds". C'est un format de données spécifique pour SODAS pour travailler avec des objets symboliques. Il existe un module dans SODAS pour faire une transformation en fichiers ".sds", qui s'appelle DB2SO. Cette transformation peut être faite à partir de fichiers des programmes qui ont leurs données sous la forme de tableaux, comme Excel ou MSAccess. Cette transformation est réalisée à l'aide des fichiers standards de configuration ODBC de Microsoft.

Le module DB2SO [HEBRAIL99] peut être appelé à partir du menu d'importation de données dans SODAS. Ce module offre à l'utilisateur l'occasion de créer des assertions d'objets symboliques à partir de données qui proviennent des bases de données relationnelles. Une base de données relationnelle est formée de plusieurs tableaux ou

'relations'. Des liaisons entre les différents tableaux peuvent également être représentées dans un même modèle relationnel.

Ces bases de données doivent être construites à partir de plusieurs individus qui sont répartis en groupes. DB2SO peut alors créer une assertion par groupe d'individus. Des relations de dépendance ou des relations hiérarchiques peuvent également être associées aux différentes assertions. Une propriété intéressante des bases de données relationnelles est qu'on peut définir des opérations sur les tableaux, dont le résultat se trouve dans de nouveaux tableaux. Des requêtes sur des bases de données relationnelles sont créées d'une combinaison de ces opérations et le résultat a alors également la forme d'un tableau. Le langage standard pour faire ces requêtes sur des bases de données relationnelles est SQL. C'est donc aussi avec SQL qu'on doit créer les assertions pour les différents groupes dans DB2SO.

1.3.2.3. Différentes méthodes

SODAS offre dix méthodes pour effectuer des analyses sur des objets symboliques. Dans ce paragraphe, chacune de ces méthodes sera brièvement expliquée [LEPRINCE&99]. La méthode SOE quant à elle, sera expliquée dans le prochain paragraphe avec la description du module SOEditor.

La méthode DIV fait une classification divisive sur un fichier SODAS avec des données de type quantitatif, ordinal, intervalle symbolique, ou des probabilités symboliques. Il faut être prudent lorsque l'on effectue des analyses avec un mélange de ces quatre types de données, parce que tous les mélanges ne sont pas supportés par DIV. Le fichier de sortie donne un rapport sur l'analyse avec les résultats après chaque étape de la classification (le nombre de classes voulu peut être spécifié avant de commencer la division). On obtient également un arbre de classification qui montre les variables qui sont retenues pour faire toutes les divisions, avec en plus des informations sur chacune des classes de la division finale.

Le but de la méthode DKS (Symbolic Kernel Discriminant Analysis) est d'étendre la méthode d'analyse discriminante non-paramétrée de Bayes à des données symboliques. Cette analyse fonctionne à partir d'un ensemble d'objets symboliques. A l'aide de ces

objets, on va essayer de faire une distinction entre un certain nombre de classes. Une fois que les classes ont été construites, les autres objets peuvent être attribués à une des classes selon la règle de Bayes.

La méthode DI calcule la matrice de distance ou une fonction similaire entre deux objets symboliques qui sont spécifiées dans le fichier. Cette méthode exige comme entrée un fichier SODAS et un nombre de paramètres. La sortie de cette méthode est un fichier de rapport sur la méthode et un autre fichier SODAS. Le fichier de sortie SODAS sera une modification du fichier d'entrée, auquel on a ajouté un tableau des valeurs de distances entre les objets symboliques.

Il existe aussi une méthode STAT, qui exécute des analyses de statistiques élémentaires. Les différents types d'analyse possibles sont les fréquences relatives pour des variables multinominales et pour des variables intervalles, un biplot des variables intervalles ou des analyses avec variables probabilistes. Le fichier de sortie est un graphique qui peut être visualisé avec l'éditeur graphique, qui est également incorporé dans le logiciel SODAS.

La méthode PCM exécute une analyse en composantes principales sur des variables classiques quantitatives et/ou sur des variables avec un intervalle symbolique. Avec la méthode de composantes principales, on tente de réduire le nombre de variables à deux mesures qui expliquent la plus grande partie des informations et qui discriminent le mieux possible les données du modèle. Et en plus, parce qu'on aura obtenu un graphique à deux dimensions, on pourra analyser la proximité entre les différents individus pour ces variables, et la proximité entre les différentes variables originelles du modèle. La méthode PCM utilise également l'éditeur graphique (comme la méthode STAT) pour représenter les données.

Le but de la méthode FDA est de généraliser l'analyse discriminante factorielle pour les objets symboliques. Cette méthode s'occupe principalement de représenter des classes les plus distinctes possibles, d'un point de vue géométrique. On va chercher une série de variables discriminantes (ou facteurs) qui sont obtenues comme des combinaisons linéaires des variables originelles. Les différents groupes sont représentés dans un

espace factoriel avec des groupes homogènes, mais le plus différent possible entre eux par rapport aux nouvelles variables factorielles.

La méthode DSD consiste à chercher des descriptions qui généralisent des classes données pour un certain nombre d'observations. Ce processus construit une ou plusieurs descriptions pour chaque classe. Le but est d'obtenir des règles pour avoir une discrimination entre les différentes classes. Ces descriptions vont caractériser les différentes classes, et elles sont formalisées comme des objets symboliques avec des probabilités. La sortie de cette méthode est un autre fichier SODAS où les individus sont les objets symboliques avec des probabilités, qui sont utilisées pour généraliser et pour discriminer les classes.

Les arbres de décision offrent un moyen efficace pour construire des fonctions de discrimination dépendant de plusieurs variables. Le but de la méthode SDT (Strata Decision Tree) est de généraliser ces méthodes de simplification pour les objets symboliques. Chaque nœud d'un arbre de décision est composé d'une série de couches et offre une règle pour les individus dans ces couches pour expliquer la variable dépendante.

Finalement, la méthode TREE propose également un algorithme pour des arbres de décision, mais qui est appliquée aux données explicitement imprécises. Elles sont décrites formellement par des assertions probabilistes dans le cadre d'analyses des objets symboliques. L'utilisateur doit alors spécifier deux groupes de variables, pour indiquer quelles variables vont être sélectionnées. D'une part, il y a la liste de variables de prédiction et d'autre part la variable qualitative qui représente la série de classes 'a priori'.

1.3.3. Description du module SOEditor

La description faite du SOEditor (Symbolic Objects Editor) est basée sur la version 2.1 de ce module. Dans la description du SOEditor [NOIRHOMME&97], les trois vues de base qui existent dans ce programme pour faire une analyse d'un objet symbolique seront successivement présentées. La première vue est une vue globale du tableau de données. Celui-ci représente simplement toutes les données pour les différents objets

symboliques. La deuxième vue est l'étoile zoom qu'on peut obtenir pour chaque individu ou objet symbolique. C'est cette vue qui est la plus importante dans le SOEditor pour faire des analyses des objets symboliques. C'est ici aussi que l'on peut demander plus de détails sur les distributions des différentes variables. Et finalement, la troisième vue est une description de chaque individu ou objet symbolique dans le SOL (Symbolic Object Language). Après la description de chacune de ces vues, les différentes options qui sont offertes par le programme SOEditor pour faire une analyse seront traitées.

Cette description sera surtout utile par après, lorsque l'on spécifiera le programme à réaliser et les modifications à apporter au SOEditor. Notre programme va devoir en effet appeler le module de SOEditor et devra également modifier ou ajouter quelques options. C'est dans cette optique que l'analyse sera faite et pourra être intéressante.

1.3.3.1. La vue "tableau"

Le tableau est normalement la première vue que l'on obtient lorsque l'on lance le programme après l'ouverture d'un fichier. Toutes les données sont affichées à l'écran dans un tableau qui reprend les différentes variables dans les colonnes et les différents individus dans les lignes. Les individus sont ici représentés par les objets symboliques et les variables contiennent des valeurs complexes comme expliqué dans la première partie. L'idée fondamentale de l'interface "tableau" est la même que pour les tableaux en Excel.

L'image sur la page suivante (image 1.3) montre un exemple d'un tableau de données. Le fichier qui est représenté analyse la consommation des ménages au Royaume-Uni. Le fichier contient 25 objets symboliques à savoir les différentes régions du Royaume-Uni que l'on voit dans les différentes lignes. Chacun de ces objets symboliques est spécifié à l'aide de 15 variables symboliques qui apparaissent dans les colonnes.

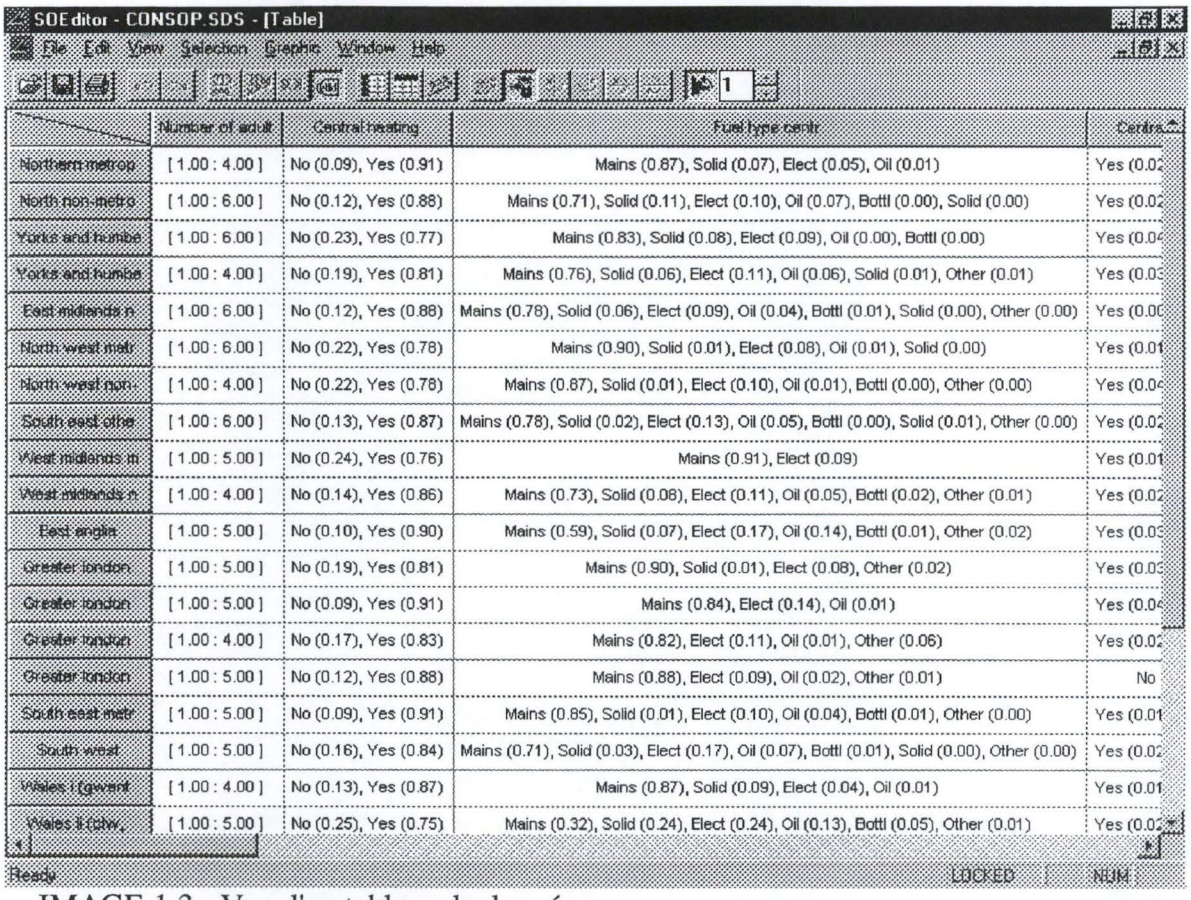


IMAGE 1.3 : Vue d'un tableau de données.
Source : SOEditor (1998)

Comme le montre ce tableau, la première variable ("number of adult") est une variable intervalle. Toutes les autres variables sont des variables modales et sont donc représentées par les différentes catégories possibles et par la probabilité de chaque catégorie pour cette région.

Dans un tableau, il est toujours possible de modifier les valeurs d'un objet symbolique existant. L'utilisateur doit simplement double-cliquer sur l'élément qu'il veut changer et introduire les nouvelles valeurs dans la syntaxe du langage des objets symboliques. Une autre fenêtre sera affichée qui contient des méta-données sur la variable sélectionnée. Cette fenêtre sert à simplifier la tâche de l'utilisateur et à réduire le risque de faire des erreurs lors de l'introduction de données.

Dans le tableau de données, l'utilisateur peut toujours sélectionner une ou plusieurs lignes et colonnes. La sélection va déterminer par après quels objets symboliques l'on va

représenter (les lignes sélectionnées) et quelles variables seront prises en compte pour cette représentation (les colonnes sélectionnées). L'utilisateur est alors obligé de sélectionner un ou plusieurs objets symboliques avant de pouvoir passer à la représentation de ces objets. Cette représentation pourra être faite sous forme d'une étoile zoom, ou encore par une description en SOL. Ces deux termes seront expliqués dans les paragraphes suivants.

1.3.3.2. Etoile zoom

Dans SODAS, on manipule des objets symboliques, qui sont caractérisés par leur complexité. Ils ont par exemple différentes valeurs pour une variable ou ils sont décrits par des variables de différents types en même temps (des variables quantitatives et des variables en catégories). Il était de ce fait très difficile de représenter les objets symboliques par des méthodes de visualisation courantes. On avait donc besoin de nouvelles méthodes pour représenter les objets symboliques, sans surcharger les représentations graphiques.

L'étoile zoom (Zoom Star) est une représentation sur plusieurs axes placés en forme d'étoile. Chaque axe donne la visualisation d'une variable. Il est possible de représenter des variables en intervalles, des variables multi-valuées, des variables en catégories et même des dépendances et des hiérarchies. Il est également possible d'avoir un graphique à deux ou à trois dimensions, ce qui offre la possibilité de faire des analyses sur différents niveaux.

Pour les graphiques en deux dimensions, les différents axes sont reliés par des lignes. Sur chaque axe on prend le point qui a la probabilité la plus élevée et on le relie avec ceux des autres axes. Dans le cas d'un intervalle, on le relie avec les deux bornes de l'intervalle et on colorie toute la surface qui se trouve entre les lignes. Les axes des graphiques en trois dimensions ne sont par contre pas reliés. On montre simplement l'intervalle de valeurs ou la distribution des probabilités entre les différentes catégories sur chaque axe.

Le but de l'étoile zoom est d'offrir une visualisation des objets symboliques, tout en gardant l'option de montrer différents niveaux de détail. Cette option est importante

pour améliorer l'apprentissage de l'utilisateur de l'objet présenté. Le graphique à deux dimensions offre une première vue globale de l'objet symbolique. Cette image peut être détaillée si on choisit l'option de montrer la distribution des probabilités sur un axe, ou de montrer les relations de dépendance ou les hiérarchies, s'il sont présents pour un des axes.

Une autre façon d'avoir plus de détails est de passer à une représentation à trois dimensions. Cette représentation reprend pour chaque variable les distributions des probabilités, sans devoir choisir cette option comme dans la représentation à deux dimensions. Ces différents niveaux de détail offrent à l'utilisateur le choix de changer la vue comme il le veut.

Prenons maintenant un petit exemple pour préciser les représentations à deux et à trois dimensions de l'étoile zoom. L'image ci-dessous (image 1.4) montre une analyse d'un objet symbolique (Northern Metropolitan) avec l'étoile zoom. Dans la représentation à deux dimensions (gauche), on voit que tous les axes sont reliés par une ligne. Les points qui sont reliés correspondent à la proportion la plus élevée de chaque catégorie. Si l'on clique sur un des axes, on obtient un graphique sous forme d'histogramme qui montre la distribution des probabilités pour la variable sélectionnée. Mais si l'on choisit l'option d'un graphique à trois dimensions (droite), on obtient directement les distributions des probabilités sur le graphique, ce qui offre plus de détails pour chacune des variables représentées.

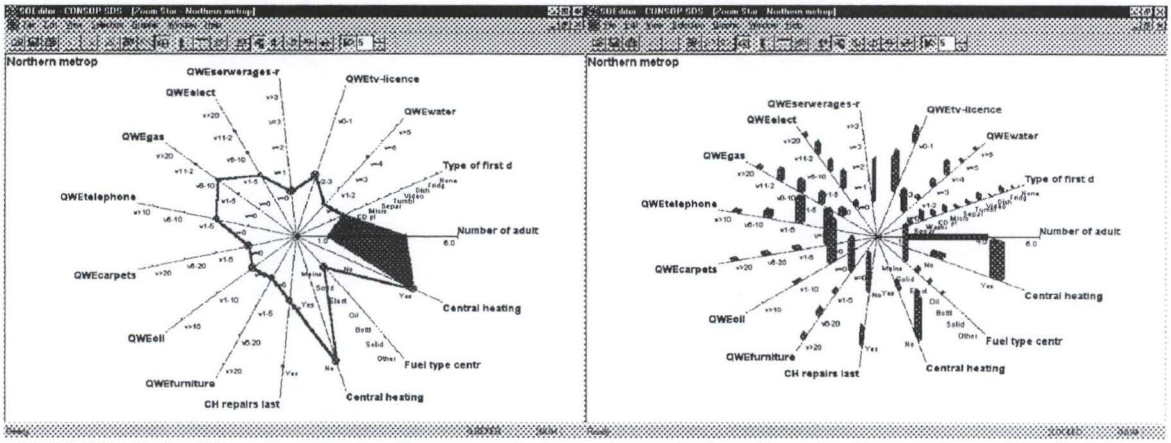


IMAGE 1.4 : Une vue en étoile zoom à deux (gauche) et à trois dimensions (droite).
Source : SOEditor (1998)

Une autre possibilité d'analyse est de comparer les représentations graphiques en étoile de différentes régions ou objets symboliques. La représentation à deux dimensions offre ici un moyen intéressant d'analyse. Ainsi, on peut mettre deux ou plusieurs étoiles l'une à côté de l'autre et la forme des lignes qui relient les différents axes montre les différences qu'il y a entre les objets symboliques. C'est une grande différence avec les méthodes d'analyse classiques comme les méthodes qui représentent des nuages de points. Ces méthodes veulent interpréter les interactions qui existent entre les variables (p.ex. analyse en composantes principales, brushing, hyperslice, hyperbox, ...). La représentation de l'étoile zoom pourra donc être classifiée dans les représentations par icônes.

1.3.3.3. Le langage des objets symboliques

Dans le logiciel SODAS, les données sont représentées dans le langage des objets symboliques (SOL). Ce langage a été choisi parce que c'est une représentation qui est facilement lisible pour les utilisateurs et parce que la syntaxe de ce langage est clairement définie. La représentation ressemble fort à ce qu'on a montré dans la première partie de l'explication des objets symboliques.

Dans l'image sur la page suivante (image 1.5), on peut ainsi voir une représentation en langage SOL qui vient du SOEditor. Une description de la région "Northern Metropolitan" y est donnée en SOL.

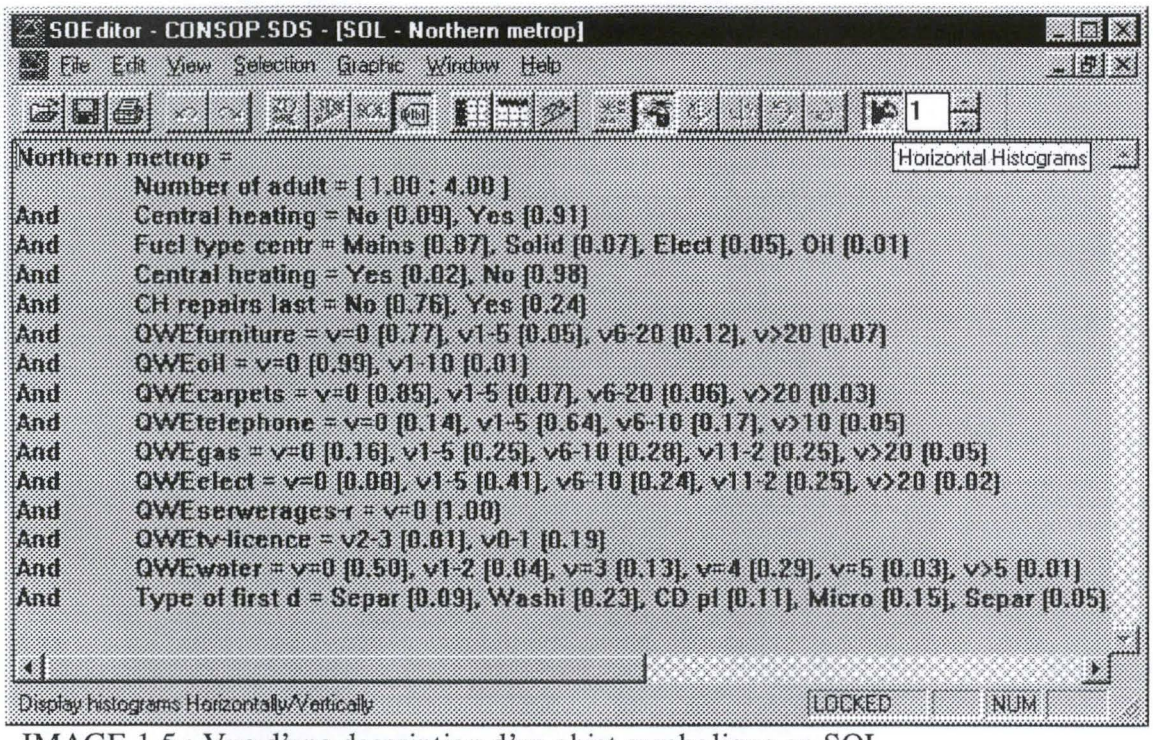


IMAGE 1.5 : Vue d'une description d'un objet symbolique en SOL.
Source : SOEditor (1998)

1.3.3.4. Options d'analyse du SOEditor

Dans cette partie, toutes les fonctions qui se trouvent dans le programme SOEditor pour faire l'analyse d'un objet symbolique, vont être décrites. Cette explication sera faite selon l'ordre de la barre d'outils, qui se trouve sur le bord supérieur dans le SOEditor, comme affiché ci-dessous (image 1.6). Ce paragraphe pourra également être utile par après, lorsque l'on devra changer ou ajouter quelques fonctions.

IMAGE 1.6 : Barre d'outils du SOEditor.
Source : SOEditor (1998)

Les cinq premières icônes correspondent aux options des menus "File" et "Edit". Ce sont des options qui sont standards dans la plupart des applications sous Windows. Elles servent respectivement à ouvrir un fichier du format SODAS, à enregistrer le fichier en cours, à imprimer la vue ou la représentation actuelle, à revenir en arrière pour annuler la dernière action et à refaire une action annulée.

Les quatre icônes suivantes correspondent aux options du menu "View". Ce sont des options qui vont invoquer une autre forme de représentation à savoir la représentation en deux ou trois dimensions, ou la représentation en langage des objets symboliques (SOL). La quatrième icône sert à afficher ou cacher les libellés des variables pour les colonnes et des objets symboliques pour les lignes du tableau. Il faut aussi noter que pour pouvoir exécuter les trois premières options, il faut avoir sélectionné au moins un objet symbolique ou une ligne du tableau. Sinon les trois icônes sont grisées et impossible à sélectionner. La sélection de différentes variables ou colonnes du tableau sera optionnel et sert à choisir seulement quelques variables à représenter sur le graphique.

Les trois icônes suivantes correspondent aux options du menu "Selection" et servent à sélectionner les colonnes et les lignes que l'on veut garder dans le tableau. On peut même aussi sélectionner les catégories que l'on veut afficher sur l'axe de chaque variable.

Toutes les autres icônes sont des options pour essayer d'améliorer la visualisation du graphique, et correspondent aux options du menu "Graphic". Ces options servent vraiment à aider l'utilisateur dans son analyse graphique des objets symboliques. Avec la première icône, on peut ajouter quatre boutons sur le graphique. Les deux premiers boutons font tourner le graphique à gauche ou à droite afin d'obtenir une meilleure présentation des différents axes. Les deux autres boutons font tourner le graphique en haut ou en bas. Les fonctions de ces quatre boutons peuvent aussi être appelées par les icônes avec les flèches. Tout à droite sur la barre d'outils se trouve encore une icône pour changer l'orientation des histogrammes et le champ, avec un nombre qui indique la hauteur voulue pour les élévations. Cette dernière option peut bien sûr seulement être utilisée pour les graphiques à trois dimensions.

Les menus "Window" et "Help" contiennent à nouveau des options standards pour des applications Windows. Le menu "Window" sert à gérer l'emplacement et la sélection des différentes fenêtres de l'application et le menu "Help" fournit de l'aide (si on a installé cette option) pour le SOEditor.

1.4. GIS

Les GIS (Geographic Information Systems) sont des logiciels pour faire des analyses avec des données cartographiques. Ils combinent différentes couches d'informations d'un endroit pour essayer de mieux comprendre cet endroit. Le but de l'utilisateur qui fait l'analyse va déterminer quelles couches d'informations doivent être combinées. Ces buts peuvent être multiples, comme par exemple chercher le meilleur emplacement pour un nouveau magasin, analyser les dégâts de l'environnement, analyser le crime dans différentes villes, ...

La différence avec une carte en papier est que toutes ces informations sont stockées dans une base de données et qu'elles sont seulement affichées à l'écran quand l'utilisateur le demande. Chaque information est stockée dans une couche différente et l'utilisateur peut choisir de l'afficher ou pas selon ses besoins. L'image 1.7 ci-dessous montre différentes couches d'un endroit qui montrent différentes informations, à savoir les rues, les bâtiments et les clients.

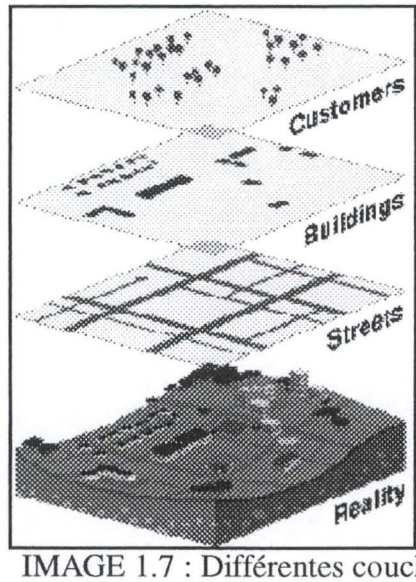


IMAGE 1.7 : Différentes couches d'une carte
Source : ESRI

L'utilisation d'un logiciel GIS peut avoir plusieurs avantages. Les deux avantages principaux sont la facilité de visualiser et d'analyser des données. Des cartes sont des moyens intéressants pour visualiser les résultats d'une analyse. Il y a une grande différence entre les données qu'on voit dans un tableau et une présentation sous la forme d'une carte. Un GIS permet de créer des cartes à un certain niveau de détail, à la

demande de l'utilisateur. Une visualisation sur carte, où l'on peut choisir ce qui est représenté, peut aider à découvrir des tendances dans les données, qui sont peut-être difficile à voir sans l'utilisation de cartes.

L'utilisation des GIS peut également aider à faire des analyses. Un GIS n'est pas un système qui prend des décisions, mais il permet d'analyser des données et de faire des requêtes sur des données, et aider ainsi aux prises de décision. On peut réunir différentes séries de données d'un endroit. Si on augmente la qualité de l'information, on peut prendre de meilleures décisions. Il sera alors plus facile par exemple, de gérer les ressources nécessaires d'une entreprise.

2. Représentation graphique d'un objet symbolique en liaison avec une carte géographique

Dans ce deuxième chapitre, nous allons expliquer le fonctionnement du logiciel que nous allons créer. Une partie de ce logiciel fera appel au module SOEditor de SODAS dont les fonctions les plus importantes ont été expliquées dans le premier chapitre, et auquel nous allons apporter quelques modifications.

Le premier paragraphe explique les objectifs globaux du logiciel. Dans ce paragraphe, nous ferons appel à plusieurs concepts qui sont expliqués dans le cours d'Interfaces Homme-Machine, lorsque seront décrites la tâche et les utilisateurs. Le deuxième paragraphe présentera alors la solution retenue pour le logiciel. A la fin de ce chapitre, on montrera aussi quelques images des écrans du logiciel pour préciser l'interface qui va être construite.

2.1. Objectifs globaux

2.1.1. But

Le but de ce travail est de développer un logiciel qui fait une combinaison de deux types différents de représentations. Ce logiciel va offrir la possibilité de relier les représentations existantes des objets symboliques avec des représentations sur une carte géographique.

Le programme qui va être développé contiendra un module qui servira à analyser des données sur des cartes. Il sera implémenté de manière à offrir une possibilité de faire une analyse de différentes variables sur la carte. L'utilisateur pourra par exemple choisir de visualiser l'effet démographique d'une des variables. Les régions seront alors coloriées en différentes couleurs selon la valeur de cette variable. La deuxième option du programme sera d'analyser toutes les variables d'un objet symbolique plus en détails. Cela sera réalisé en faisant appel au module SOEditor de SODAS.

2.1.2. Le stéréotype des utilisateurs

Le programme à développer fournit un support multimédia qui pourrait par exemple être utilisé par les étudiants d'un département de géographie. Autre utilisateur potentiel de ce programme pourrait être un employé d'un service de statistique. Nous allons commencer cette analyse en prenant l'hypothèse d'un étudiant comme utilisateur. Après cela, les caractéristiques seront mentionnées dans le cas d'un employé d'un service de statistique. La description du stéréotype des utilisateurs peut se faire à partir des paramètres suivants (cf. le cours d'Interfaces Homme-Machine [BODART98]). Dans le cas d'un étudiant :

- Expérience de la tâche : moyenne. L'utilisation du logiciel pour faire des analyses sur des données cartographiques suppose que l'étudiant ait acquis ou est en train d'acquérir les connaissances de la partie théorique du cours. Mais les connaissances théoriques de l'étudiant ne font pas encore de lui un spécialiste, il lui manque une expérience pratique.

- Expérience de systèmes informatiques : moyenne. L'étudiant du département de géographie a certainement déjà travaillé sur une station de travail. Mais on ne peut pas supposer que l'étudiant maîtrise entièrement le fonctionnement et le comportement des objets d'une interface graphique.

- Motivation : élevée. L'étudiant est vu comme voulant maîtriser son cours. Avec le logiciel, il a maintenant un moyen pratique et intéressant qui lui permet de travailler de manière autonome. Un facteur important qui justifie sa motivation pour utiliser le logiciel est le fait que ses connaissances seront évaluées à la fin de l'année.

- Expérience d'un moyen d'interaction : riche. L'ordinateur ne forme peut-être pas un outil quotidien pour les étudiants en géographie. Mais on peut quand même supposer que la manipulation du clavier et de la souris, n'est pas étrange pour ces étudiants, qui ont sûrement au moins déjà rédigé des documents avec un traitement de texte.

Le stéréotype de l'utilisateur sera légèrement différent pour un employé d'un service de statistique :

- Expérience de la tâche : riche. On peut supposer qu'un employé d'un service de statistique travaille souvent avec des logiciels pour faire des analyses de données. Il a donc, contrairement aux étudiants, également l'expérience pratique de ces logiciels en plus de ses connaissances théoriques. Cela justifie donc une expérience riche de la tâche.

- Expérience de systèmes informatiques : moyenne. Pour faire ses analyses, l'employé du service de statistique utilise sûrement une station de travail. Il ne maîtrise cependant peut-être pas entièrement le fonctionnement et le comportement de tous les objets d'une interface graphique.

- Motivation : élevée. En général, on peut supposer qu'un employé sera motivé pour faire le travail qu'il fait. D'abord il y a la motivation intrinsèque de l'employé qui revient à un intérêt pour le travail, ou à la volonté de trouver des résultats significatifs maintenant qu'il dispose d'un moyen pratique et intéressant pour le faire. En outre, il y a toujours la motivation extrinsèque qui provient du salaire versé en fin de mois.

- Expérience d'un moyen d'interaction : riche. Comme l'ordinateur est un outil quotidien pour un employé d'un service de statistique, on peut sûrement supposer qu'il maîtrise la manipulation des éléments nécessaires comme le clavier et la souris.

2.1.3. Les paramètres descriptifs

Ce paragraphe sur les paramètres descriptifs part de l'hypothèse que le stéréotype des utilisateurs correspond aux étudiants en géographie. L'explication des paramètres descriptifs suivra également la découpe qui est faite dans le cours d'Interfaces Homme-Machine [BODART98] :

- Pré-requis : minimaux. Les pré-requis d'une tâche expriment la quantité de connaissances du système que l'utilisateur doit posséder en vue de l'utilisation efficace du système. Dans ce cas, le système correspond au logiciel d'analyse sur des données cartographiques. L'interface du logiciel sera très explicite en ce qui concerne son utilisation, mais il faut quand même un minimum de temps pour s'habituer au programme. Une connaissance de base d'un environnement de type Windows en ce qui concerne les éléments interactifs qui composent l'interface (listes déroulantes, boutons, ...) est supposée connue.

- Productivité de la tâche : faible. La productivité d'une tâche exprime la fréquence d'exécution. Les étudiants vont utiliser le logiciel dans le cadre d'un cours. Ils le feront avec une fréquence que nous pouvons qualifier comme faible parce que cela ne prendra pas plus que deux heures par semaine. On voit bien la différence si on le compare par exemple avec un employé qui utilise un logiciel pour enregistrer des bons de commande pendant toute la journée.

- Environnement objectif : non existant. L'environnement objectif représente la manipulation effective ou non d'objets spécifiques indépendants de l'interface. L'étudiant n'a pas besoin d'utiliser d'autres objets pendant qu'il est en train de faire une analyse sur des données cartographiques.

- Reproductibilité de l'environnement : praticable. La reproductibilité de l'environnement est praticable si elle peut être transposée dans le cadre du système. L'utilisateur manipule directement l'interface par lequel il veut faire ses analyses.

- Organisation / structuration de la tâche : faible. La structuration de la tâche explique le degré de liberté ou de contrainte que l'utilisateur a dans son accomplissement. L'utilisateur est libre de choisir l'option d'analyse qu'il veut prendre. En outre, dans chaque option d'analyse il a le choix de demander plus de détails sur quelque chose qui l'intéresse ou de retourner en arrière pour commencer une autre analyse.

- Importance de la tâche : modérée. L'importance de la tâche informe quant au caractère fondamental, crucial ou vital d'une tâche. Une analyse des données cartographiques sert à mieux comprendre les données et à trouver les tendances éventuelles qui existent. Si ces tendances sont trouvées, il sera peut-être possible d'anticiper sur des conséquences.

- Complexité de la tâche : modérée. La complexité de la tâche manifeste le degré de complexité cognitif, manipulatoire, intellectuel, quant aux sous-tâches et actions mises en jeu en vue d'accomplir la tâche. L'étudiant a parfois besoin de faire référence à son cours pour utiliser le programme. Un utilisateur qui connaît son cours aura beaucoup plus facile de faire des analyses sur des données cartographiques que celui qui n'a pas encore revu son cours.

2.1.4. Les critères d'utilité et d'utilisabilité

Le but d'une interface est de permettre à l'utilisateur du logiciel de réaliser efficacement la tâche qu'il veut accomplir. L'évaluation de l'interface peut se faire par rapport à son utilité et son utilisabilité. Pour évaluer ces caractéristiques, nous allons prendre en considération les critères suivants (comme vu dans le cours d'Interfaces Homme-Machine [BODART98]) :

- Temps d'apprentissage : Le temps nécessaire pour apprendre à travailler avec le programme nous paraît assez court. Comme l'utilisateur fait partie du département de géographie, il sera habitué de travailler avec des données cartographiques. Il connaît les principes des différentes méthodes d'analyse et l'apprentissage devrait se passer alors assez vite.

- Rapidité d'exécution : On peut dire que la rapidité d'exécution de la tâche sera assez élevée. L'utilisateur est familiarisé avec les différentes méthodes d'analyse. Il pourra alors vite se rendre compte comment il devra procéder pour faire l'analyse souhaitée.

- Fréquence d'erreurs : On pourra distinguer deux types d'erreurs différents, à savoir les erreurs d'intention et les erreurs d'exécution. La fréquence des erreurs d'intention va dépendre de la qualité de l'interface. Comme l'interface sera assez claire et comme l'utilisateur sera toujours bien guidé dans ses actions, on pourra supposer que la fréquence des erreurs d'intention sera limitée. La fréquence des erreurs d'exécution sera principalement liée au niveau de l'utilisateur et à ses défauts de manipulation.

- Temps de correction : L'interface du logiciel permettra de revenir en arrière quand on se rend compte de l'existence d'une erreur dans une des étapes précédentes. On considère donc que le temps nécessaire pour réaliser une correction est court.

- Période de rémanence : Il est difficile d'estimer la période de conservation des connaissances et des techniques acquises par l'utilisateur. On peut seulement évaluer l'ordre de grandeur de cette période de rémanence. Etant donné que l'effort cognitif requis pour utiliser le programme est faible et que le temps d'apprentissage est assez court, on peut conclure que la période de rémanence sera assez importante.

- Satisfaction subjective à utiliser le système : Le logiciel offre un moyen intéressant qui permet à l'étudiant de maîtriser son cours. Il existe donc une certaine satisfaction de pouvoir utiliser dans le pratique les connaissances acquises pendant les cours théoriques.

- Degré de couverture : Le degré de couverture des dispositifs de l'interface par rapport aux actions qui font partie de la tâche de l'utilisateur nous semble moyen. Il existe en effet beaucoup d'autres techniques pour faire des analyse sur des données cartographiques. L'utilisateur est cependant limité aux méthodes proposées dans le programme. Il ne peut donc pas passer à une autre méthode qui n'est pas inclue dans le programme et qui pourrait sembler être intéressant à un moment donné.

2.2. Solution retenue et justification

Deux options principales seront proposées pour faire une analyse de la carte de Namur ou d'une autre ville pour laquelle on possède les données cartographiques et les régions. La première consiste à se concentrer sur une région en particulier et à regarder toutes les variables pour cette région sur un graphique. On aura un lien vers le module SOEditor de SODAS pour avoir la représentation sur un graphique. Ce sera donc une représentation en forme d'une étoile pour visualiser en même temps toutes les variables du modèle pour cette région. Dans SOEditor, on pourra exécuter toutes les fonctions habituelles, qui ont été expliquées dans la première partie.

La deuxième option que l'on peut choisir pour faire une analyse consiste à prendre une variable pour toutes les régions. On regarde alors les différences qui existent entre les différentes régions pour cette variable et on va les visualiser également sur la carte. La visualisation sera faite en coloriant toutes les régions selon la valeur qu'elles ont pour cette variable.

2.3. Proposition de l'interface

Dans ce qui suit, des spécifications plus détaillées des différentes parties du logiciel seront données. D'abord, le menu principal avec ses options sera expliqué, après quoi une explication de l'option de l'étoile des différentes régions avec le lien vers SOEditor sera donnée. Pour finir, la deuxième option d'analyse du programme sera expliquée, c'est-à-dire l'option pour colorier une carte selon une variable particulière.

Dans cette partie, les différents écrans du logiciel seront également visualisés. Les images de ces écrans ont été créées en Borland Delphi 3.0 et sont montrées seulement comme illustration des écrans et des liens entre eux. Les écrans montrés ici peuvent donc encore changer dans l'implémentation du programme, mais l'idée derrière les écrans restera la même.

Dans le programme, on aura besoin de deux genres de fichiers. Le premier est un fichier qui stocke tous les noms des villes. Le deuxième sert à stocker pour chaque ville tous

les noms des régions. Ici, on aura donc un fichier pour chaque ville. Les fichiers peuvent être dans le format ".txt" ou bien dans le format ".ini".

2.3.1. Menu principal

Avant d'arriver sur le menu principal, l'utilisateur devra choisir la ville dont il veut faire une analyse. Un combobox sera montré à l'écran avec une liste des villes possibles. L'utilisateur peut ou bien choisir une des villes de la liste, ou bien il peut choisir d'ajouter une ville qui n'est pas encore dans la liste. S'il choisit une ville de la liste, il procède au menu principal après avoir appuyé sur le bouton 'OK'. Le bouton 'Annuler' sert à retourner au menu principal. On ne peut donc pas utiliser ce bouton quand on vient de lancer le programme, mais seulement après qu'on a choisi l'option de changer de ville dans le menu principal. L'image 2.1 ci-dessous montre un exemple de la fenêtre qui sert à choisir la ville.

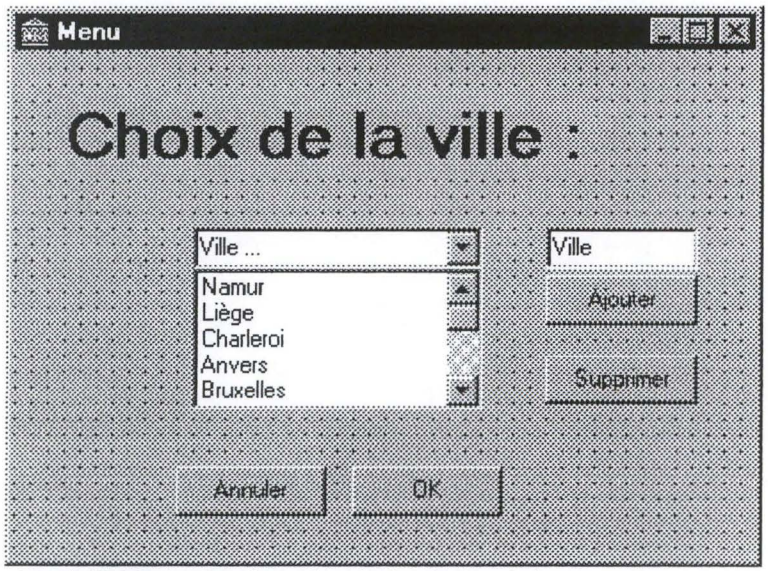


IMAGE 2.1 : Fenêtre du choix de la région

Si l'utilisateur veut ajouter une ville dans la liste, il écrit le nom dans la fenêtre et appuie sur le bouton 'Ajouter'. Une recherche sera automatiquement lancée pour retrouver les fichiers nécessaires pour cette ville, comme par exemple le fichier de l'écran de fond avec la carte et le fichier des données. Si les fichiers nécessaires sont trouvés, l'utilisateur arrive sur le menu principal. Sinon, un menu de configuration de données sera montré et l'utilisateur peut alors chercher les fichiers manuellement. Le bouton

'Supprimer' sert à enlever une ville de la liste. Si l'utilisateur ne choisit pas une ville, il ne pourra pas procéder au menu principal.

Le menu principal sera ouvert en même temps que la fenêtre du choix de la ville et sera partiellement visible derrière celle-ci. Dès que l'utilisateur a choisi une ville pour faire ses analyses, la fenêtre du choix de la ville disparaît et l'utilisateur peut faire son choix sur le menu principal. Avant d'avoir fait un choix de ville, tous les boutons seront affichés comme inutilisables et l'utilisateur ne pourra donc pas les sélectionner.

Le menu principal permet à l'utilisateur de choisir les options qui seront à la base de son analyse. Ce menu contiendra cinq options. Les deux premières sont les deux moyens d'analyse qui sont possibles pour l'utilisateur, à savoir l'analyse avec l'étoile zoom d'une région et l'analyse d'une ville en coloriant la carte. Ces options seront expliquées dans les paragraphes suivants. La troisième option est la configuration de données, la quatrième donne une possibilité de changer de ville, et la cinquième option sert simplement à quitter le programme.

Il sera donc possible pour l'utilisateur de remplacer les données par une configuration de nouvelles données, par exemple une mise à jour des données existantes. L'utilisateur sera déjà renvoyé à cet écran de configuration quand les fichiers nécessaires ne sont pas trouvés après l'ajout d'une ville. L'écran pour faire cette configuration ou cette importation n'est pas montré ici, mais sera semblable aux écrans standards Windows pour l'importation de données ou de fichiers. Ces données devront déjà être dans le format des données Sodas (.sds). Il est possible de créer des données dans ce format SODAS à l'aide de DB2SO.

Si l'utilisateur choisit l'option de changer de ville, la fenêtre de début avec le choix de la ville sera à nouveau ouverte. Il peut alors exécuter toutes les fonctions comme avant. Et c'est ici que le bouton 'Annuler' sur la fenêtre du choix de la ville prouve son utilité. Si l'utilisateur désire quand même ne pas choisir une autre ville et de retourner au menu principal, il peut appuyer sur le bouton 'Annuler' pour simplement être renvoyé au menu principal sans changer de ville.

Le menu principal comporte aussi une carte avec comme image de fond les différentes régions de la ville. Cette carte n'est pas cliquable, comme dans une autre partie du programme, mais montre simplement une image avec les différentes régions qui existent pour cette ville. L'image du menu principal est montrée ci-dessous (image 2.2).

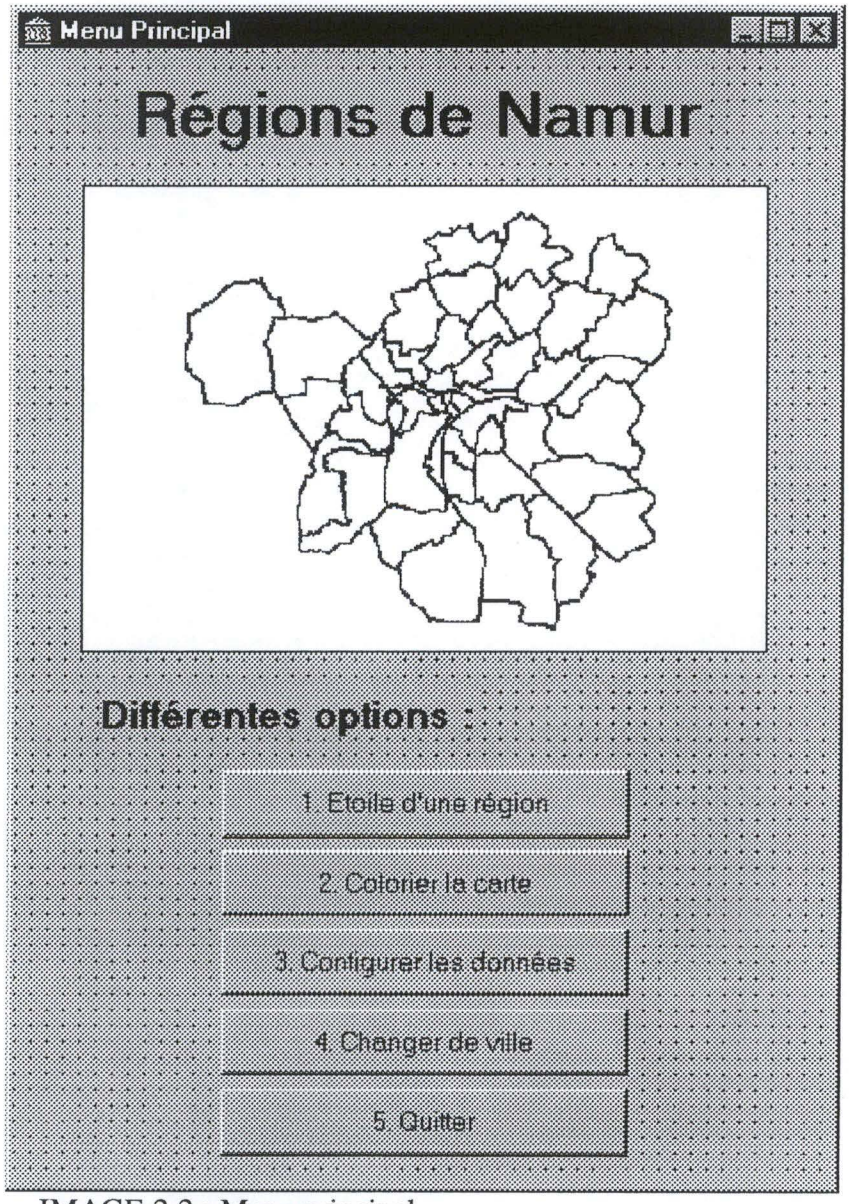


IMAGE 2.2 : Menu principal.

2.3.2. Etoile d'une région

Si on clique sur le bouton qui représente la première option donnant l'étoile d'une région, on arrive au premier sous-menu. Une image de ce sous-menu est montrée sur la page suivante (image 2.3).

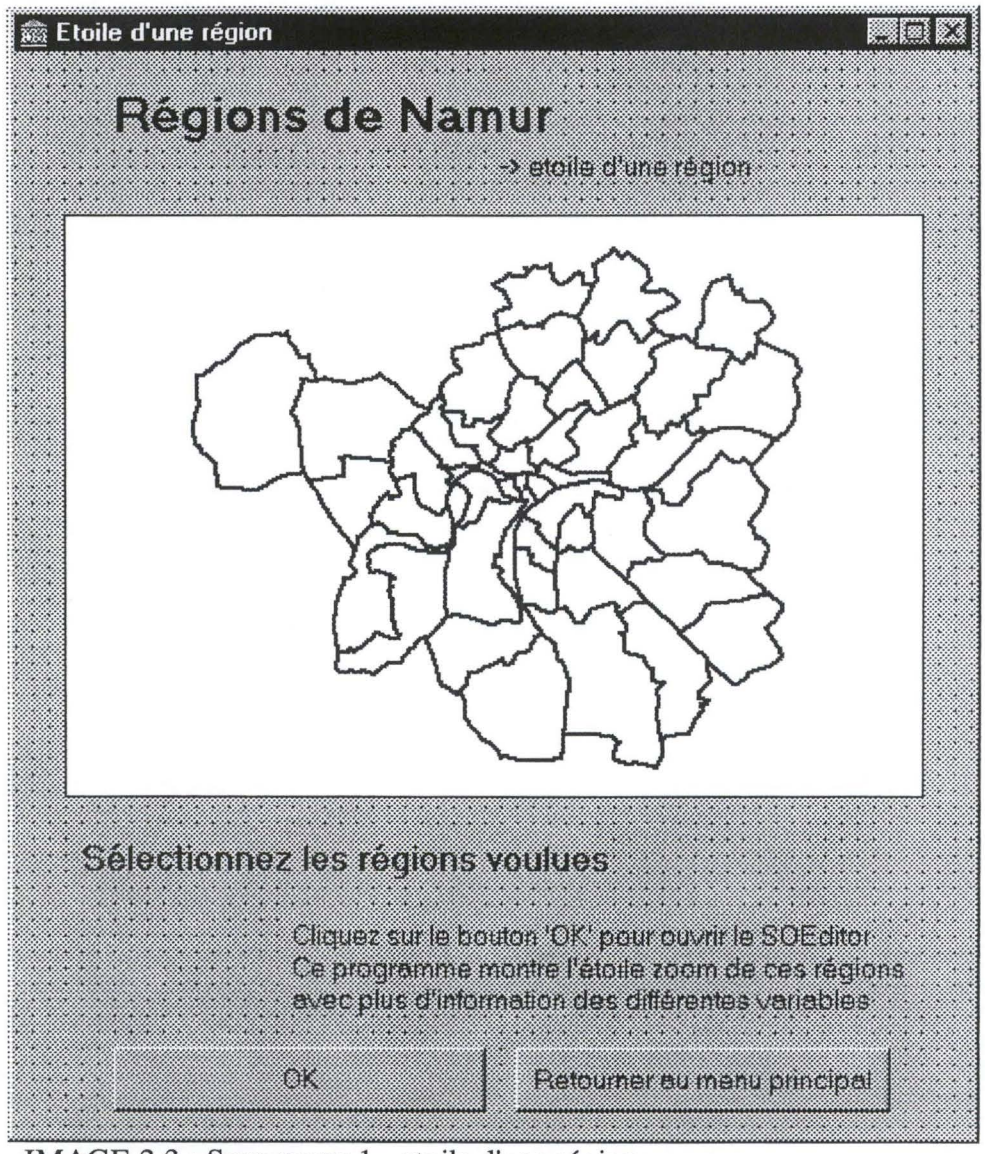


IMAGE 2.3 : Sous-menu 1 - etoile d'une région

Dans ce sous-menu, la carte sera une carte cliquable. Si on clique sur une des régions qui sont représentées sur la carte, cette région sera sélectionnée pour une analyse avec le SOEditor après. On voit qu'une région est sélectionnée quand elle change de couleur. Le module SOEditor sera ouvert automatiquement après que l'utilisateur clique sur le bouton 'OK'.

Quand le module SOEditor est ouvert, le fichier de données sera chargé automatiquement et on peut choisir d'afficher les étoiles des régions sélectionnées à l'écran. Cela peut être réalisé à l'aide des paramètres que l'on fournit au module SOEditor qui règlent l'ouverture du fichier.

Après l'ouverture de SOEditor, les fenêtres du programme principal vont disparaître de l'écran. Elles ne seront pas simplement minimisées en bas de l'écran sur la barre de tâche, mais elles vont complètement disparaître. Si on ne procède pas de cette façon l'utilisateur pourrait encore rentrer dans le programme sans fermer le SOEditor et ouvrir une deuxième version du SOEditor. Quand l'utilisateur a fini l'analyse des représentations graphiques dans le SOEditor, il peut fermer cet écran ou sortir du SOEditor, et il sera automatiquement transporté vers le programme principal dans le premier sous-menu, la fenêtre de l'étoile d'une région.

Sur la fenêtre du premier sous-menu, on affiche aussi quelques explications qui guident l'utilisateur dans ses actions. En bas de l'écran il y a encore un bouton pour retourner au menu principal sans passer au SOEditor.

2.3.3. Colorier la carte

Si on choisit le bouton qui représente la deuxième action du menu principal, on arrive sur un deuxième sous-menu. Ce sous-menu sera la deuxième analyse que l'utilisateur peut faire. Il peut analyser les données d'une ville en coloriant la carte selon une variable sélectionnée. Le sous-menu contient un combobox dans lequel l'utilisateur peut choisir la variable pour faire l'analyse. La carte sera coloriée après que l'utilisateur a appuyé sur le bouton 'Afficher'. Ce sous-menu contient également un bouton pour retourner au menu principal. L'image 2.4 sur la page suivante montre la fenêtre du deuxième sous-menu.

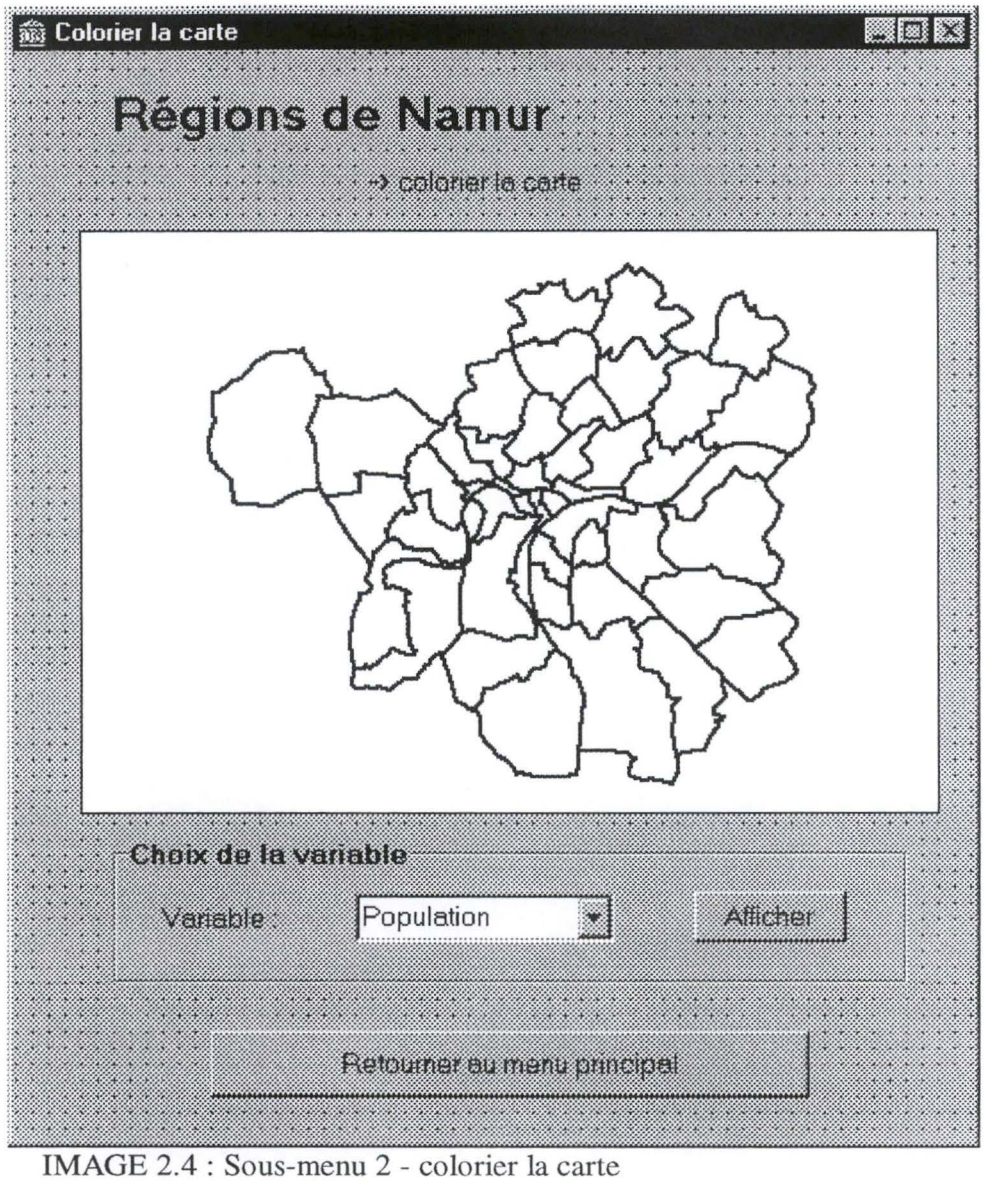


IMAGE 2.4 : Sous-menu 2 - colorier la carte

3. Implémentation

Dans ce troisième chapitre, nous allons parler du programme que nous avons réalisé en décrivant les différentes classes. Il y aura également quelques explications supplémantaires pour spécifier le fonctionnement du programme. Après cela, les problèmes que nous avons rencontré pendant l'implémentation sont expliqués, avec, à la fin de ce chapitre une introduction sur le format ".dxf".

3.1. Visual C++

3.1.1. Différentes classes

Au total, sept classes ont été créées, à savoir les six classes des fenêtres plus une classe de paramètres. Dans le programme, il existe encore beaucoup d'autres classes, mais ce sont toutes des classes standards qui sont disponibles dans la bibliothèque de Visual C++. Nous ne parlerons ici donc pas des classes correspondantes aux combobox et aux listes, dont on a utilisé les fonctions dans notre programme.

Une liste complète de toutes les classes importantes est donnée dans le tableau 3.1 ci-dessous.

CChoisirRegions	Classe qui sert à choisir la ville, elle est ouverte dès le début du programme.
CColorer	Classe qui contient la deuxième option du menu principal.
CConfigRegion	Configuration des villes (différentes régions + les cartes).
CEtoile_ProdDlg	Classe qui sert à montrer l'étoile d'une région, faire le lien vers SOEditor.
CSodas_ProdApp	Classe de l'application, qui forme le début du programme.
CSodas_ProdDlg	Cette classe va définir le comportement du

	menu principal de l'application, classe qui sert à appeler les autres classes.
CParamFile	Classe avec le fichier des paramètres.

TABLEAU 3.1 : Liste des différentes classes du programme.

3.1.2. Explication

Le principe de l'implémentation est qu'il y a une classe derrière chaque fenêtre. Une classe importante correspond dès lors au menu principal, les autres fenêtres étant gerées par des sous-classes.

Toutes les sous-fenêtres contiennent deux variables. La première variable ('id') est une variable pour l'identification de la fenêtre. C'est une variable qui est utilisée automatiquement par le constructeur de la classe et qui sert à l'utilisation interne du programme. Il est important d'avoir créé une variable de cette forme selon les documents du MicroSoft Developer Network (MSDN) pour avoir plus de sécurité dans le programma. La deuxième variable ('parent') est un pointeur qui construit une référence vers la classe au-dessus. C'est à l'aide de ces pointeurs qu'on pourra naviguer entre les différentes fenêtres.

La fenêtre avec le menu principal sera donc appelée dès que l'on commence le programme par la classe principale. Et ce sera cette classe avec le menu principal qui va immédiatement appeler la fenêtre du choix de la ville à l'aide de la méthode 'oninitdialog'.

Il y aura un grand nombre d'événements à travers tout le programme. Chaque combobox, chaque bouton ou chaque liste qui est présente sur une des fenêtres va donner lieu à un événement quand l'utilisateur a choisi de le sélectionner. Pour chaque événement, on a alors spécifié le code qui devait être exécuté par le programme après la sélection de l'item en question.

Il existe encore une dernière classe 'cparamfile', qui est une classe dérivée d'une classe standard en Visual C++, à savoir 'cstdiofile'. Cette classe de paramètres sert à écrire un

objet quand on veut ajouter une région. La valeur de la région et la valeur du fichier seront alors automatiquement écrites dans le fichier correct.

3.2. Problèmes rencontrés dans l'implémentation

Les problèmes que nous avons rencontrés dans l'implémentation ont surtout à voir avec une connaissance modérée du Visual C++. N'ayant jamais suivi des cours en C++, et n'ayant jamais utilisé C++ auparavant pour faire des développements, il me manquait les connaissances théoriques nécessaires. Le programme a alors été construit pas à pas, après avoir acquis les connaissances nécessaires par auto-apprentissage, en étudiant les différentes sources que l'on peut trouver sur Internet ou dans des livres.

Nous devons mentionner quelque chose qui a changé dans le programme. C'est le fait qu'il n'y a pas de carte du tout dans le menu principal. Cette carte était prévue dans la spécification du deuxième chapitre mais n'aurait fait que rendre les choses confuses. Nous avons préféré inclure les cartes dans les différents sous-menus pour faire les analyses. Les différents écrans du programme n'auront dès lors pas exactement la même mise en page de celle qui était présentée dans le deuxième chapitre, mais l'idée derrière les écrans est restée inchangée.

Dans l'implémentation du programme, quelques problèmes ont également été rencontrés. Il y a donc, par conséquent, quelques aspects du programme qui ne correspondent pas entièrement à ce qui a été décrit dans la deuxième partie de ce travail. Un changement important est que l'utilisateur ne peut pas cliquer sur la carte pour sélectionner une région pour passer au SOEditor. La sélection doit être faite dans une liste avec les noms de toutes les régions. Après que l'utilisateur a choisi la région dont il veut avoir la représentation avec l'étoile, il doit simplement cliquer sur un bouton pour être renvoyé au module SOEditor.

Ce problème est dû au format de fichier des cartes. Originellement, on voulait avoir les représentations des cartes de la ville en format ".dxf". C'est un format de fichiers qui contient de l'information supplémentaire sur chaque pixel de l'image. A l'aide de cette

information, on aurait pu alors extraire les données nécessaires pour faire savoir au programme dans quelle région de l'image l'utilisateur se trouve.

Une autre conséquence de ce problème est que la deuxième option d'analyse était également devenue plus difficile à développer. Cette option consistait à colorier les différentes régions de la carte dans différentes couleurs, selon les valeurs d'une variable. Comme on ne pouvait pas travailler avec les différents pixels des cartes, on a dû résoudre ce problème autrement. La solution qui a été retenue finalement est l'importation d'un fichier par variable. Ces fichiers contiennent alors les différentes cartes coloriées des variables de la ville.

Malgré ces problèmes avec les fichiers ".dxf", il serait intéressant de faire plus de recherches dans ce domaine. Le format de fichiers ".dxf" est un format riche qui doit être envisagé pour des analyses de ce genre. C'est dans cette optique que l'on a inclus le dernier paragraphe de ce chapitre. Ce paragraphe donne une analyse de base du format de fichiers ".dxf".

3.3. Format ".dxf"

Le format des fichiers ".dxf" (drawing interchange and file formats) est développé et est maintenu par Autodesk. C'est un format de fichier ASCII indépendant de la plate-forme, qui est utilisé pour échanger des fichiers graphiques basés sur des vecteurs. Le format de fichiers de AutoCAD (.dwg) est un format qui change assez régulièrement avec des nouvelles versions du logiciel. Le format .dxf a alors été développé originellement pour échanger des dessins entre AutoCAD et les autres applications CAD (computer-aided design).

Un fichier ".dxf" est donc un fichier texte, mais il a été organisé en différentes parties pour faciliter l'utilisation. Le tableau 3.2 ci-dessous montre l'organisation des fichiers standards ".dxf".

1. Section en-tête	Cette section contient des informations générales sur le dessin. Chaque paramètre a un nom variable et une valeur associée.

2. Section des classes	Cette section contient l'information des classes, pour lesquelles il existe des instances dans les autres sections.
3. Section des tableaux	Cette section contient différentes définitions de tableaux :
Tableau de l'identification de l'application (APPID)	
Tableau de référence aux blocks (BLOCK_RECORD)	
Tableau du style de dimension (DIMSTYLE)	
Tableau des couches (LAYER)	
Tableau du type de ligne (LTYPE)	
Tableau du style du texte (STYLE)	
Tableau du système des coordonnées de l'utilisateur (UCS)	
Tableau des vues (VIEW)	
Tableau de la configuration du viewport (VPORT)	
4. Section des blocks	Cette section contient de l'information sur les entités qui forment les blocks du dessin.
5. Section des entités	Cette section contient les entités (= objets graphiques) du dessin avec les références vers les blocks.
6. Section des objets	Cette section contient les objets non-graphiques du dessin
7. Section de l'image (thumbnail)	Une vue du dessin est gardée dans cette section (section optionnelle)
8. Section fin du fichier	-

TABLEAU 3.2 : Organisation des fichiers .dxf.
Source : Autodesk, Inc.

Un fichier ".dxf" est composé d'un grand nombre de groupes. Chaque groupe prend la place de deux lignes du fichier. La première ligne du groupe est le code du groupe et la deuxième ligne contient la valeur pour ce groupe. Ces groupes vont indiquer le type des valeurs qui suivent. En utilisant des groupes spéciaux de séparation, les fichiers ".dxf" sont divisé en différentes sections et tableaux. Il existe également des groupes spéciaux pour indiquer le début et la fin du fichier. Comme on peut voir ci-dessus, un fichier ".dxf" contient sept sections avec de l'information plus la partie de la fin du fichier. Nous allons expliquer plus en détails quelques-unes de ces sections.

La section en-tête du fichier ".dxf" contient différentes variables associées au dessin. Ces variables sont définies à l'aide de plusieurs opérations. Elles peuvent par exemple

représenter la version du logiciel AutoCAD, le date et l'heure de création du dessin, le type des lignes, la taille des textes, le nom de la couche et encore beaucoup d'autres paramètres.

La section des tableaux contient plusieurs tableaux qui peuvent se trouver dans un ordre quelconque, sauf le tableau LTYPE qui doit précéder le tableau LAYER. Chaque tableau commence avec un délimiteur (TABLE), avec de l'information qui suit pour identifier le type du tableau. Après cela, on trouve les entrées du tableau. La fin du tableau est indiquée avec un délimiteur (ENDTAB).

La section des blocks du fichier ".dxf" contient toutes les définitions des blocks et également les entités qui forment les blocks qui sont utilisés dans le dessin. Le format des entités dans cette section est identique que celui de la section des entités. Toutes les entités sont définies entre les délimiteurs BLOCK et ENDBLK, qui se trouvent seulement dans cette section-ci.

La section des entités contient également les définitions des entités. Certains groupes de cette section sont toujours présents pour définir les entités, tandis que d'autres groupes sont seulement présents s'ils diffèrent de leurs valeurs par défaut. Chaque entité peut avoir des paramètres qui spécifient l'élévation, l'épaisseur, le type des lignes ou de l'information sur les couleurs associées.

La section des objets contient de l'information sur les objets non-graphique du dessin. Cela veut dire que tous les objets qui ne sont pas des entités ou des tableaux de symboles sont stockés dans cette section. Un exemple d'une entrée de cette section est le dictionnaire qui contient les groupes et quelques styles.

Pour économiser de l'espace dans la base de données des dessins et dans le fichier ".dxf", les points qui sont associés aux entités sont exprimés dans le système des coordonnées de chaque entité (ECS). Ce système permet à AutoCAD d'utiliser des représentations beaucoup plus compactes pour les différentes entités. La seule information additionnelle dont on a besoin pour décrire la position d'une entité dans l'espace, est le vecteur décrivant l'axe pour la troisième dimension du système des coordonnées et la valeur d'élévation.

Le format ".dxf" qui est décrit ci-dessus, est une représentation complète d'un dessin en AutoCAD. C'est un fichier sous format ASCII qui est facilement utilisable par d'autres programmes. Mais il existe en outre une autre représentation sous une forme binaire. Un fichier pareil contient toutes les informations des autres fichiers ASCII ".dxf", mais il est beaucoup plus compact. Typiquement, un fichier binaire prend 25 % d'espace en moins que les fichiers ".dxf". Parce que les fichiers ".dxf" ASCII sont beaucoup plus utilisés que les fichiers binaires, le terme "fichier .dxf" est utilisé pour les fichiers ASCII, et le terme ".dxf binaire" pour le format binaire.

4. Analyse des données de la ville de Namur

Dans ce quatrième chapitre nous donnerons d'abord quelques explications concernant les données. Nous expliquerons d'où elles viennent, quelles transformations nous avons dû faire pour arriver au modèle définitif, et nous donnerons la liste des variables qui ont été retenues pour faire les analyses. Après cela, nous procèderons aux analyses des données. Nous ferons ces analyses avec le programme SODAS et le module SOEditor, mais également avec un logiciel pour analyser des données cartographiques.

4.1. Les données

4.1.1. Provenance des données

Les données qui sont utilisées dans cette analyse sont les données récoltées par le professeur Poulain. Il est professeur de démographie à l'UCL et dirige le Gédap (Centre d'étude de gestion démographique pour les administrations publiques), qui étudie la situation à Namur depuis 1992. Il a surtout utilisé les données fournies par le recensement en 1991, mais aussi celles du registre national de la population et quelques autres sources comme le CPAS, l'enseignement primaire et des images satellites du paysage pour faire la division des quartiers.

La région du grand Namur a été divisée en 46 quartiers. Ces frontières des quartiers ne correspondent pas aux limites des anciennes communes, puisqu'elles ont été repensées en fonction des habitudes des habitants ou de modifications physiques, comme par exemple une autoroute qui a coupé une région en deux. Il n'y a pas beaucoup d'anciennes communes qui se retrouvent intégralement dans le nouveau découpage. L'ancien Namur par exemple, est partagé en huit quartiers et Jambes en cinq. Selon la découpe faite par le professeur Poulain, un quartier coïncide avec une population confrontée au même environnement, au même quotidien. Ce sont donc des espaces de vie qui ont été définis en allant sur le terrain. La liste complète des 46 régions définies par le professeur Poulain se trouve en Annexe 2. A côté du nom de la région, on trouvera aussi les abréviations qui sont utilisées dans les analyses.

4.1.2. Variables retenues

Le modèle retenu pour faire cette analyse comporte un fichier de huit variables pour les 46 régions. Le tableau 4.1 ci-dessous montre les différentes variables avec leurs modalités.

1. Groupes d'âges de la population	
a)	< 20 ans
b)	20 - 60 ans
c)	60 - 80 ans
d)	> 80 ans
2. Confort des logements privés	
a)	Sans petit confort
b)	Petit confort (= eau courante, wc dans le logement, salle de bain)
c)	Moyen confort (= petit confort + chauffage central dans le logement)
d)	Grand confort (= moyen confort + cuisine > 4m² + téléphone + auto)
3. Année de construction des logements privés	
a)	< 1919
b)	1919 - 1945
c)	1946 - 1961
d)	1962 - 1970
e)	1971 - 1980
f)	> 1981
4. Superficie du logement en m²	
a)	< 35 m²
b)	35 - 44 m²
c)	45 - 54 m²
d)	55 - 64 m²
e)	65 - 84 m²

f)	85 - 104 m²
g)	105 - 124 m²
h)	> 125 m²
5. Personne habitant dans le logement	
a)	< 2 ans
b)	2 - 5 ans
c)	6 - 11 ans
d)	> 80 ans
6. Population active par groupe d'âge	
a)	< 24 ans
b)	25 - 34 ans
c)	35 - 44 ans
d)	45 - 54 ans
e)	> 55 ans
7. Population active par statut professionnel	
a)	Employé du service public
b)	Employé du secteur privé
c)	Ouvrier du secteur privé
d)	Ouvrier du secteur public
e)	Indépendant
f)	Employeur
g)	Autre
8. Population scolaire par genre d'enseignement suivi	
a)	Enseignement primaire
b)	Enseignement secondaire
c)	Enseignement supérieur

TABLEAU 4.1 : Variables retenues dans l'analyse

Ces variables ont été choisi afin de garder quelques mesures démographiques bien différentes pour décrire la population dans les régions. Ainsi, nous avons retenu des variables sur le logement, sur l'âge des individus, sur la population active et sur l'enseignement.

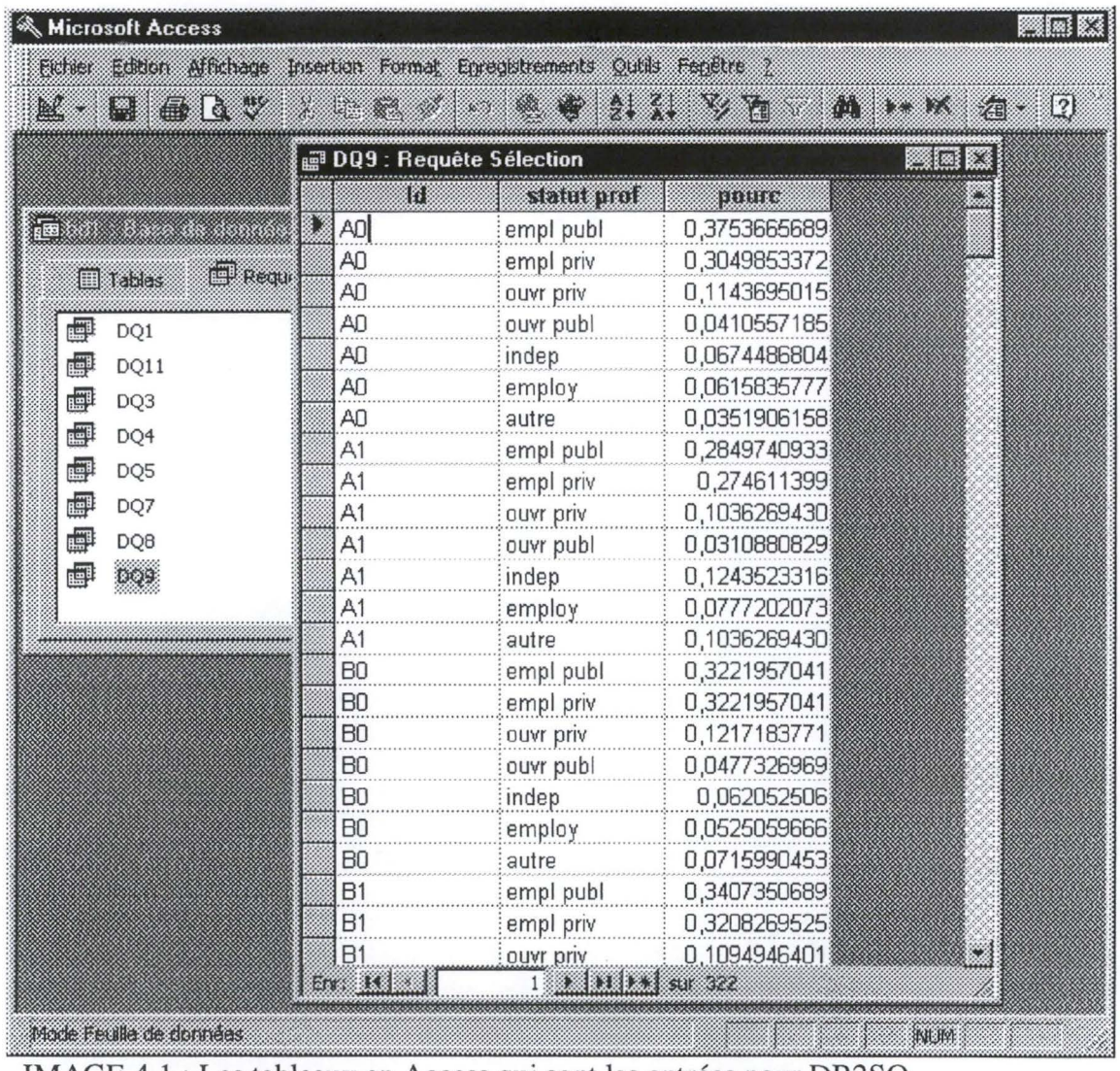
4.1.3. Transformation des données en objets symboliques

Toutes les variables sont sous la forme d'objets symboliques avec des proportions. Cela veut dire que la somme de toutes les catégories est égale à 1 pour toutes les variables et pour chaque région. Mais nous avons dû faire quelques transformations avant de pouvoir travailler avec des objets symboliques sous cette forme.

D'abord, une transformation a été faite pour obtenir les proportions. Les données du recensement sont toutes des données absolues qui représentent des unités, comme par exemple des individus ou des ménages. Pour chaque région, nous avons pris les données de chacune des catégories. La proportion a été calculée en divisant la valeur de chaque catégorie par le total des observations de la région correspondante. Les totaux des régions ont été calculés en faisant la somme de toutes les catégories. Les observations inconnues qui se présentaient pour certaines régions n'ont pas été prises en considération pour calculer la somme, de sorte que la somme des proportions sera égale à 1 pour chaque variable de chaque région.

Les tableaux en Excel qui ont été obtenus de cette manière sont les données d'entrée pour Access. Etant donné que l'on va faire une transformation des données en objets symboliques, les tableaux et les requêtes d'Access doivent avoir une forme spéciale. Comme les différentes catégories des variables vont former des variables multiples, on est obligé de fournir des tableaux à DB2SO qui ont une forme comme montré dans l'image 4.1 sur la page suivante.

L'image 4.1 montre le tableau pour la septième variable "population active par statut professionnel" qui sera l'entrée sous DB2SO. On distingue la première partie du tableau avec les données pour les régions Andoy (A0), Wierde (A1), Bouge (B0) et Moulin à Vent (B1). Pour chaque région, la valeur de chaque catégorie est écrite sur une nouvelle ligne. La première colonne du tableau doit contenir l'identifiant du groupe ou de la région, la deuxième colonne du tableau contient les différentes catégories qui existent pour cette variable et la dernière colonne contient les proportions.



Id	statut prof	pourc
A0	empl publ	0,3753665689
A0	empl priv	0,3049853372
A0	ouvr priv	0,1143695015
A0	ouvr publ	0,0410557185
A0	indep	0,0674486804
A0	employ	0,0615835777
A0	autre	0,0351906158
A1	empl publ	0,2849740933
A1	empl priv	0,274611399
A1	ouvr priv	0,1036269430
A1	ouvr publ	0,0310880829
A1	indep	0,1243523316
A1	employ	0,0777202073
A1	autre	0,1036269430
B0	empl publ	0,3221957041
B0	empl priv	0,3221957041
B0	ouvr priv	0,1217183771
B0	ouvr publ	0,0477326969
B0	indep	0,062052506
B0	employ	0,0525059666
B0	autre	0,0715990453
B1	empl publ	0,3407350689
B1	empl priv	0,3208269525
B1	ouvr priv	0,1094946401

Enr : 1 sur 322

IMAGE 4.1 : Les tableaux en Access qui sont les entrées pour DB2SO.
Source : Access

[BOCK&00] présente plusieurs requêtes SQL possibles que l'on peut utiliser en DB2SO pour importer des tableaux des bases de données. Ces requêtes aident à construire les tableaux des assertions. Pour construire les objets symboliques, on a besoin de la requête SQL de type 3. Cette requête est utilisée pour obtenir de l'information de la base de données sur des groupes. Cela va mener à une variable multi-valuée qui décrit l'objet symbolique.

Une fois que l'on a obtenu la matrice de données symboliques en DB2SO, les données doivent encore être exportées. Cette étape va aboutir à la construction d'un fichier dans le format qui peut être utilisé par SODAS, c'est-à-dire le format ".sds".

4.2. Analyse des données

Dans les deux premiers paragraphes, nous allons faire des analyses sur le fichier de données qui est décrit ci-dessus, avec quelques méthodes de SODAS. Ce sont les méthodes DIV et PCM qui ont été utilisées pour faire ces analyses. Après, nous utiliserons le module SOEditor pour faire des analyses avec les étoiles. Et à la fin de cette partie, nous vérifierons si l'on peut trouver les mêmes résultats en utilisant un logiciel pour faire des analyses sur des données cartographiques.

4.2.1. Méthode DIV

Comme on a pu le voir dans la première partie, la méthode DIV fait une classification divisive sur un fichier SODAS. On obtient un arbre de classification qui montre les variables qui sont retenues pour faire toutes les divisions, et de l'information sur chacune des classes de la division finale.

Nous avons choisi de faire une classification divisive en trois classes. Si on avait continué à faire des divisions, les différentes classes devenaient trop petites et il devenait pratiquement impossible d'encore analyser les résultats. La manière de travailler retenue permet d'obtenir des classes qui ont presque toutes la même taille. Le tableau 4.2 ci-dessous montre les trois classes avec l'abréviation des régions qui font partie de ces classes. En annexe 2, la liste complète des régions avec leurs abréviations est donnée.

Partition en 3 classes		
Classe 1 (n=13)	Classe 2 (n=16)	Classe 3 (n=17)
J0 J1 J2 N0 N1	BN BZ CG DX EP	A0 A1 B0 B1 BL
N2 N3 N4 N6 N7	F0 GB LI M0 M1	CH DV F1 J3 J4
N8 N9 S0	MD NN S1 SM TX	LO N5 SL V1 V2
	V0	W0 W1
Inertie expliquée par le modèle : 36.002499 %		

TABLEAU 4.2 : Classification divisive

On voit dans ce tableau que la première classe contient toutes des régions qui sont situées dans le centre de Namur. Ce sont les régions des anciennes communes de Namur (N0 - N9), Jambes (J0 - J2) et Saint-Servais (S0). Il y a seulement la région Namur-Citadelle (N5) de l'ancienne commune de Namur qui n'est pas contenue dans cette classe. On pourra vérifier par après, avec une comparaison entre étoiles, si cette région diffère vraiment des autres régions du centre. L'arbre de classification, qui est généré par la méthode DIV, pourra montrer quelles variables sont utilisées pour faire la division entre les classes. Il est montré dans l'image 4.2 ci-dessous.

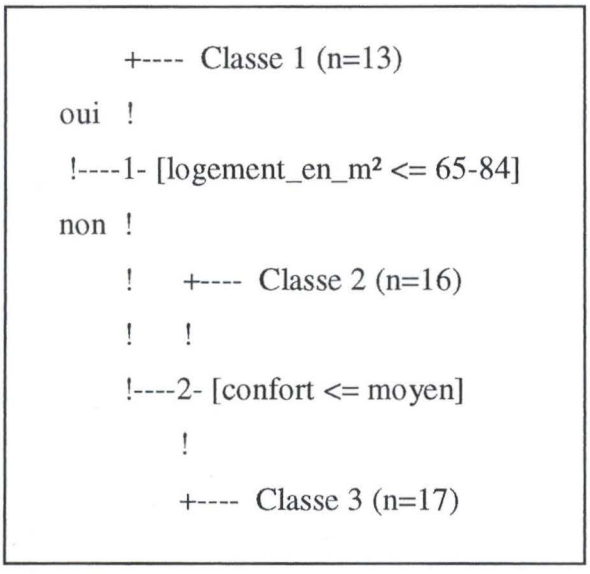


IMAGE 4.2 : Arbre de classification

Les numéros qui sont notés à chaque noeud représentent l'ordre des divisions. On voit que la première classe se distingue des autres classes par la variable qui mesure la superficie des logements en m². Il semble donc que les régions du centre de Namur ont des logements qui sont plus petits que les régions qui sont situées dans les banlieues. La deuxième et troisième classe se distinguent par la variable qui mesure le confort des logements. Les régions qui sont groupées dans la troisième classe ont un confort plus élevé que celles de la deuxième classe.

Il serait maintenant intéressant de regarder la localisation des régions des différentes classes sur une carte géographique. Cela permet de voir si cette division conduit à construire des groupes de régions qui se trouvent proches les unes des autres. Cette carte est illustrée par l'image 4.3 sur la page suivante.

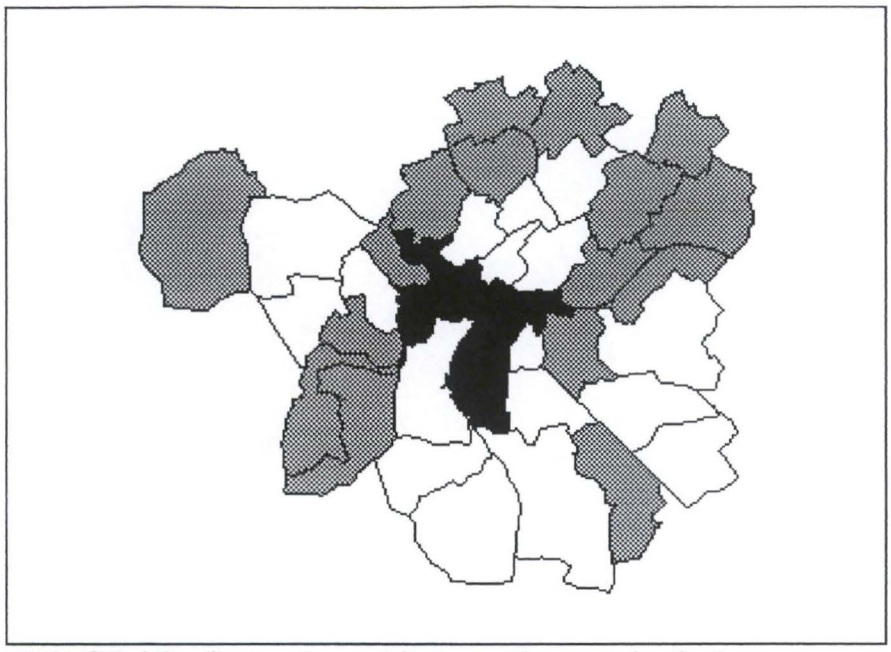


IMAGE 4.3 : Carte géographique représentant les 3 classes.
Noir : classe 1 -- gris : classe 2 -- blanc : classe 3.
Source : MapInfo

On peut voir sur ce graphique que les régions de la première classe se trouvent en effet toutes dans le centre de Namur. Pour les deux autres classes, la localisation est un peu moins claire, mais on peut cependant constater qu'il existe des groupes de régions pour ces deux classes. Il n'existe que deux régions qui ne sont pas limitrophes à d'autres régions de la même classe, à savoir la région Temploux (TX) à gauche et Naninne (NN) en bas, deux régions de la deuxième classe.

Le tableau 4.2 montre également que seulement 36 % de l'inertie totale est expliquée par cette analyse. Ceci veut dire qu'une grande partie de l'inertie qui est présente dans ce modèle reste inexpliquée. Une analyse en composantes principales pourra peut-être offrir de meilleurs résultats.

4.2.2. Méthode PCM

Comme on a pu voir dans la première partie, une analyse en composantes principales va réduire le nombre de variables. Nous allons essayer de trouver quelques mesures qui expliquent la plus grande partie des informations et qui discriminent le mieux possible les données du modèle. Et en plus, parce que l'on aura souvent obtenu un graphique à

deux dimensions, on pourra analyser la proximité entre les différents individus pour ces variables.

L'image 4.4 ci-dessous montre toutes les régions du modèle qui sont représentées dans un graphique à deux dimensions. Les deux axes sont formés par les deux facteurs qui expliquent la plus grande partie des informations. Le premier axe explique 40,78 % de l'inertie et le deuxième axe 21,03 %. Cette analyse en composantes principales explique donc une partie de l'inertie (61,81 %) qui est considérablement plus grande que l'analyse divisive réalisée précédemment (36 %).

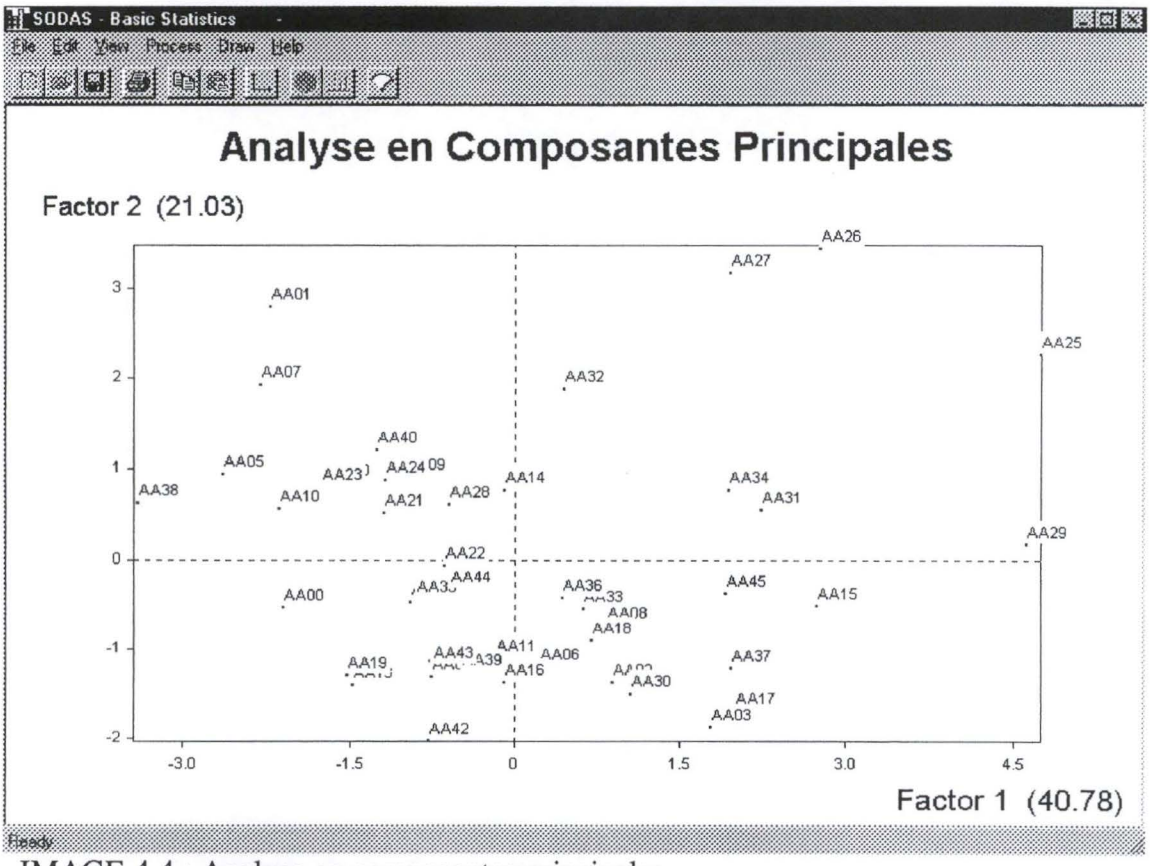


IMAGE 4.4 : Analyse en composantes principales.
Source : SODAS

Si l'on veut interpréter ce graphique, il est préférable de le faire à l'aide de la classification divisive que l'on a fait ci-avant. Les régions de la première classe se trouvent presque toutes à droite dans le graphique, ce qui veut dire qu'elles ont des valeurs positives pour le premier facteur. La division entre la deuxième et la troisième classe est moins claire. Peut-être que l'on pourrait combiner ces deux facteurs avec le troisième facteur de l'analyse en composantes principales pour mieux voir la distinction

entre ces deux classes. Le troisième facteur de l'analyse explique encore 14,21 % de l'inertie du modèle.

Comme on le voit sur l'image 4.5 ci-dessous, le troisième facteur aide en effet beaucoup à mieux positionner la deuxième et la troisième classe. A gauche, on voit une représentation des régions de la deuxième classe. Elles sont représentées dans un graphique avec le premier et le troisième facteur qui forment les axes. Elles sont presque toutes situées dans le quadrant à gauche en haut du graphique. Presque toutes les régions ont donc des valeurs négatives pour le premier et positives pour le troisième facteur.

A droite, seulement les régions qui font parties de la troisième classe sont représentées sur le graphique. Ces régions se situent surtout dans le quadrant à droite en bas du graphique, ce qui veut dire des valeurs négatives pour le deuxième et positives pour le troisième facteur.

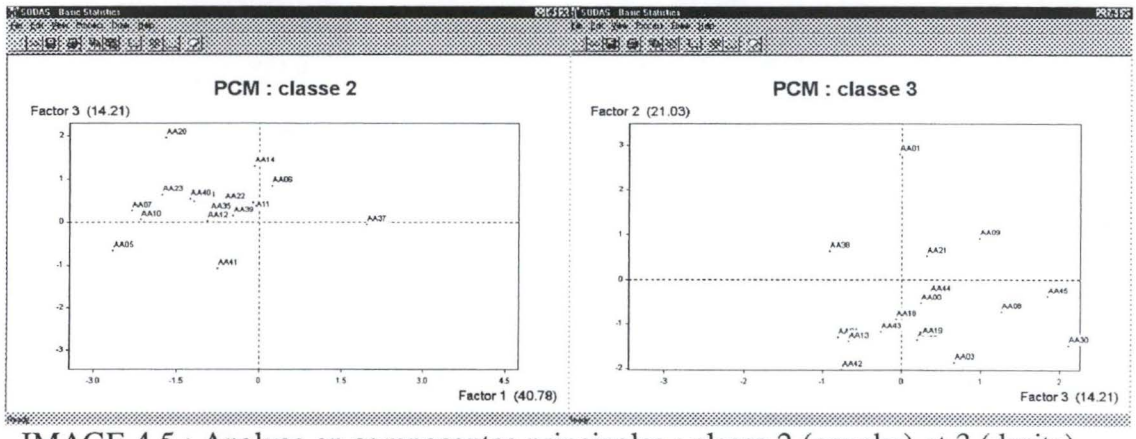


IMAGE 4.5 : Analyse en composantes principales : classe 2 (gauche) et 3 (droite).
Source : SODAS

Si on veut donner un sens à ces résultats, il vaut mieux regarder la matrice avec les corrélations entre les facteurs de l'analyse en composantes principales et les variables du modèle. Cette matrice est représentée dans le tableau 4.3 sur la page suivante qui reprend les trois premiers facteurs.

Variable du modèle	Facteur 1	Facteur 2	Facteur 3
Groupes d'âge	0,973	0,005	0,010
Confort des logements	0,848	0,215	-0,421
Année de construction	-0,196	0,697	0,455
Superficie des logements	0,671	0,628	-0,342
Personne habitant	0,658	-0,327	0,304
Population active - âge	0,726	-0,347	0,327
Statut professionnel	-0,032	0,726	0,234
Genre d'enseignement	0,383	0,051	0,617

TABLEAU 4.3 : Corrélations entre variables et facteurs.

On voit que le premier facteur correspond essentiellement à une grande proportion de personnes âgées, beaucoup de logements sans grand confort et d'une petite superficie, et une grande proportion de personnes âgées dans la population active. Ces paramètres sont très représentatifs pour les régions du centre de Namur. Et ce sont en effet les régions du centre, qui se trouve dans la première classe, qui ont des coefficients positifs pour ce premier facteur.

Le deuxième facteur est surtout caractérisé par une grande proportion de personnes qui sont actives comme indépendants, beaucoup de logements avec une petite superficie et de vieux bâtiments. Le troisième facteur correspond à une grande proportion de personnes qui suivent l'enseignement supérieur ainsi qu'à des logements avec beaucoup de confort et de vieux bâtiments.

Si l'on rassemble les analyses des facteurs, des classes et des variables, on peut aboutir à des conclusions sur la signification de la deuxième et de la troisième classe. La deuxième classe regroupe surtout les régions avec une grande proportion de logements avec beaucoup de confort et une grande superficie, mais avec une petite proportion de personnes âgées et beaucoup de personnes qui ont suivi l'enseignement supérieur. La troisième classe regroupe surtout les régions avec une grande proportion de logements d'une grande superficie et une petite proportion de personnes qui sont actives comme indépendants.

On doit encore mentionner que l'on n'a pris qu'une seule modalité par variable pour faire cette analyse en composantes principales, et donc pas toutes les modalités. SODAS ne permet en effet pas de travailler avec des proportions pour une analyse en composantes principales. Cette façon de procéder entraîne plus ou moins les mêmes résultats mais les montre plus clairement, ce qui en facilite l'interprétation.

Il serait également intéressant d'utiliser un autre logiciel qui travaille avec des données classiques pour faire cette analyse en composantes principales. Le logiciel qui est utilisé pour faire cette analyse classique est SPAD.

Pour faire cette analyse, on ne travaille donc plus avec des données symboliques, et les variables ne sont par conséquent plus en catégories. On perd donc l'information supplémentaire apportée par les données symboliques. Les mêmes catégories qu'auparavant ont été reprises, chaque catégorie étant considérée comme une variable séparée. On obtient ainsi 41 variables au total. L'analyse que nous faisons est une analyse en composantes principales et la représentation graphique est donnée ci-dessous dans l'image 4.6.

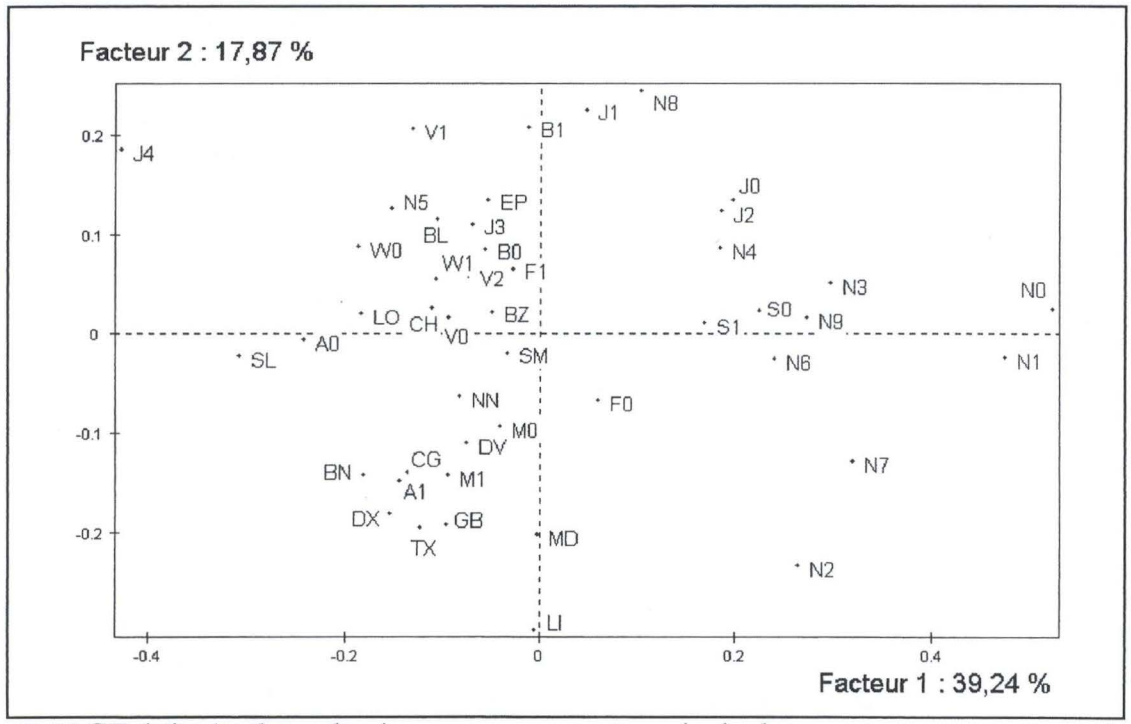


IMAGE 4.6 : Analyse classique en composantes principales.
Source : SPAD

L'image 4.6 montre la localisation des 46 régions par rapport aux deux facteurs, l'abréviation de chaque région étant donnée à côté du point. On peut constater que les régions de la première classe se trouvent toutes à droite sur le graphique. Les seules régions à droite sur le graphique qui n'appartiennent pas à la première classe sont la région S1 (St-Servais-ch de BXL) et F0 (Flawinne). La division entre la deuxième et la troisième classe est moins claire. On peut cependant voir que les régions de la deuxième classe ont plutôt des valeurs négatives pour les deux facteurs et elles se trouvent dès lors en grande partie dans le quadrant à gauche en bas du graphique.

Cette analyse devrait permettre de comparer les résultats d'une analyse classique avec une analyse symbolique faite auparavant. Mais sur base des analyses faites ici, il est difficile de dire si l'on obtient vraiment des meilleurs résultats avec une analyse symbolique. Comme l'analyse en composantes principales, que nous avons faite auparavant en SODAS, ne tenait compte que d'une catégorie par variable, une partie de l'information a été perdue. On peut cependant dire que les résultats d'une analyse classique ne sont pas entièrement différents, parce qu'en fin de compte l'on peut retrouver les différentes classes de l'analyse divisive.

Une analyse en composantes principales ne permet pas seulement de visualiser la proximité entre les différents individus, mais aussi la proximité entre les différentes variables. Dans notre analyse, les variables que nous avons utilisées sont les catégories de notre modèle. En reliant les différentes catégories d'une variable, on peut alors analyser le changement de cette variable à travers les catégories. C'est également le logiciel SPAD qui est utilisé pour exécuter cette analyse avec la méthode CORBI et qui fournit une analyse en correspondances binaires (analyse en composantes principales sur des proportions). La représentation graphique de cette analyse est montrée en image 4.7 sur la page suivante.

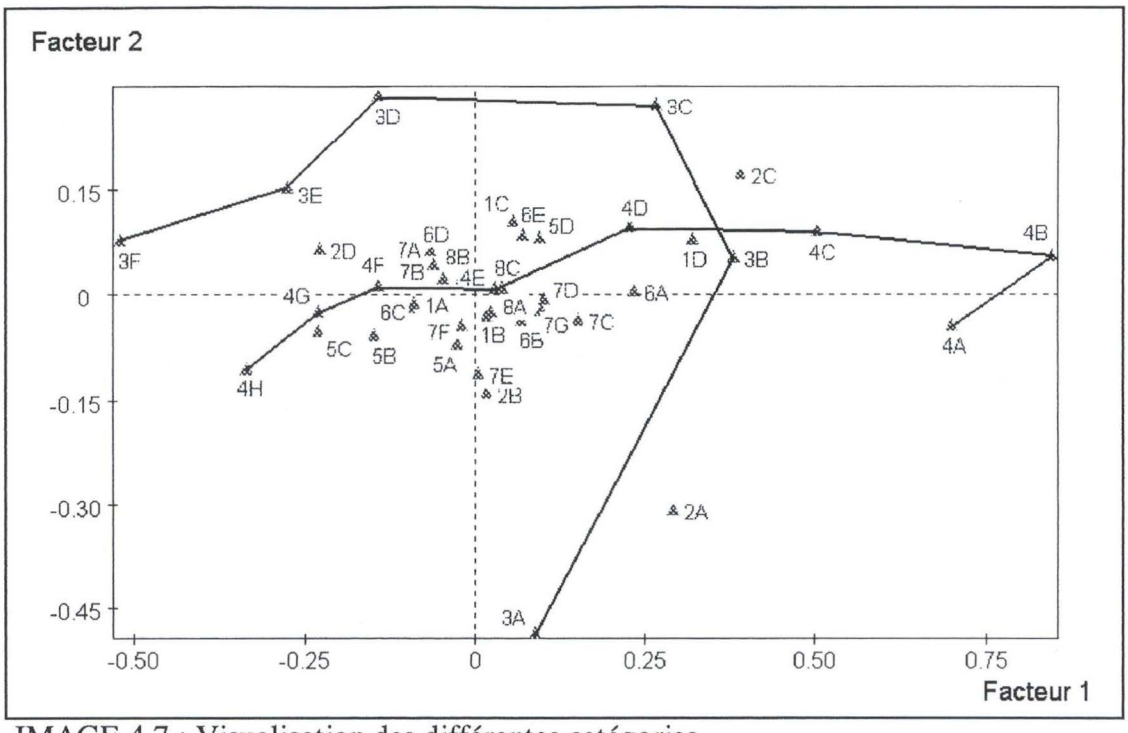


IMAGE 4.7 : Visualisation des différentes catégories.
Source : SPAD

Nous avons relié les catégories de deux variables, pour lesquelles le changement était le plus remarquable. Les catégories des autres variables changent peut-être plus par rapport aux autres facteurs de l'analyse en composantes principales.

Les différentes catégories de la variable "superficie du logement en m²" (4A - 4H) vont de droite à gauche sur le graphique. Le changement est donc en grande partie en relation avec le premier facteur de l'analyse. Comme la catégorie 4A représente les logements avec une petite superficie (<35 m²), on pourrait conclure que des valeurs élevées pour le premier facteur représentent une grande proportion de logements avec une petite superficie. On peut vérifier cette conclusion sur l'image 4.6 avec la localisation des différentes régions. Les régions qui ont des valeurs élevées pour le premier facteur et se trouvent à droite sur le graphique sont en effet surtout les régions de la première classe, donc du centre de Namur. Et en outre, la région qui se trouve toute à droite sur le graphique est la région N0 (Namur-Centre).

La deuxième variable dont nous avons relié les différentes catégories est la variable "année de construction des logements privés" (3A - 3F). On voit sur l'image 4.7 que les

différentes catégories de cette variable utilisent bien les deux facteurs de l'analyse en composantes principales. La catégorie 3A, qui représente les logements construits avant 1919, est liée à des valeurs négatives élevées pour le deuxième facteur. Les régions qui correspondent à cette localisation en image 4.6 sont surtout les régions de la deuxième classe. Les catégories 3D, 3E et 3F par contre (logements plus nouveaux), se situent dans le quadrant à gauche en haut du graphique, comme la plupart des régions de la troisième classe. Cette analyse montre donc encore une caractéristique en plus pour diviser ces deux classes. La deuxième classe contient des régions avec beaucoup de logements vieux, et les régions de la troisième classe se distinguent par une proportion plus élevée de nouveaux logements.

4.2.3. SOEditor

Dans l'analyse avec la méthode DIV, nous avions trouvé une division en trois classes. On avait pu remarquer que toutes les régions de l'ancienne commune de Namur étaient incluses dans la première classe, sauf la région Namur-Citadelle qui se trouvait même dans la troisième classe. Il serait alors intéressant de voir comment cette région diffère d'une autre région du centre. On va donc faire ici une comparaison entre cette région de Namur-Citadelle (N5) et la région de Namur-Centre (N0).

L'outil qui est adapté pour faire des analyses de ce genre est le module SOEditor de SODAS [NOIRHOMME&97]. Nous allons comparer ces deux régions avec l'étoile zoom pour voir où se trouvent les différences. L'image 4.8 sur la page suivante montre une comparaison des deux étoiles pour toutes les variables du modèle.

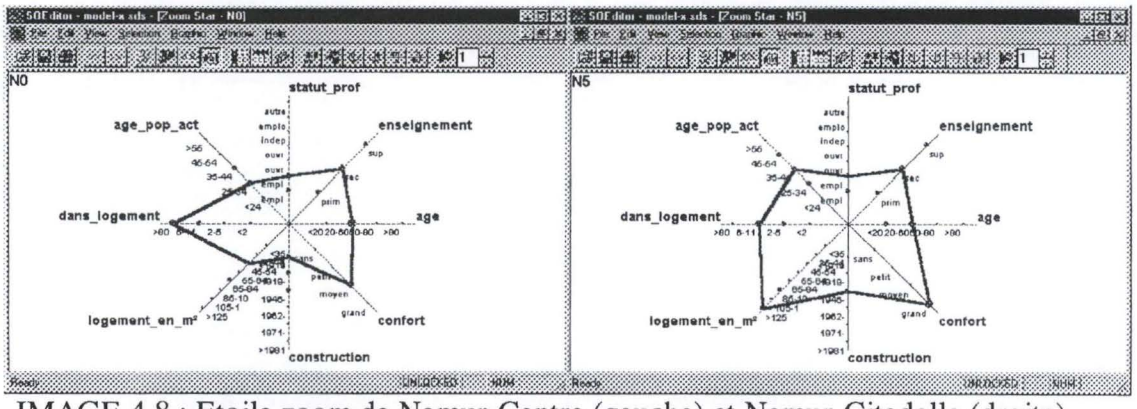


IMAGE 4.8 : Etoile zoom de Namur-Centre (gauche) et Namur-Citadelle (droite).
Source : SOEditor

Il apparaît immédiatement que les deux étoiles ont une forme qui est assez différente l'une de l'autre. Ce sont les catégories avec la proportion la plus élevée de chaque variable qui sont reliées sur les différents axes. Si on regarde cet aspect, on voit que la catégorie avec la proportion la plus élevée est différente pour cinq variables des huit que comporte le modèle.

Si l'on entre maintenant un peu plus dans le détail, on peut aller regarder les différentes distributions des variables. Une représentation graphique de cette forme aide à mieux voir les différences qu'il y a entre les deux régions. Une première variable dont on va aller regarder la distribution est la variable "superficie du logement en m²". L'image 4.9 ci-dessous montre les distributions pour cette variable pour les deux régions. A gauche, on voit la distribution pour la région Namur-Centre, et à droite celle de Namur-Citadelle.

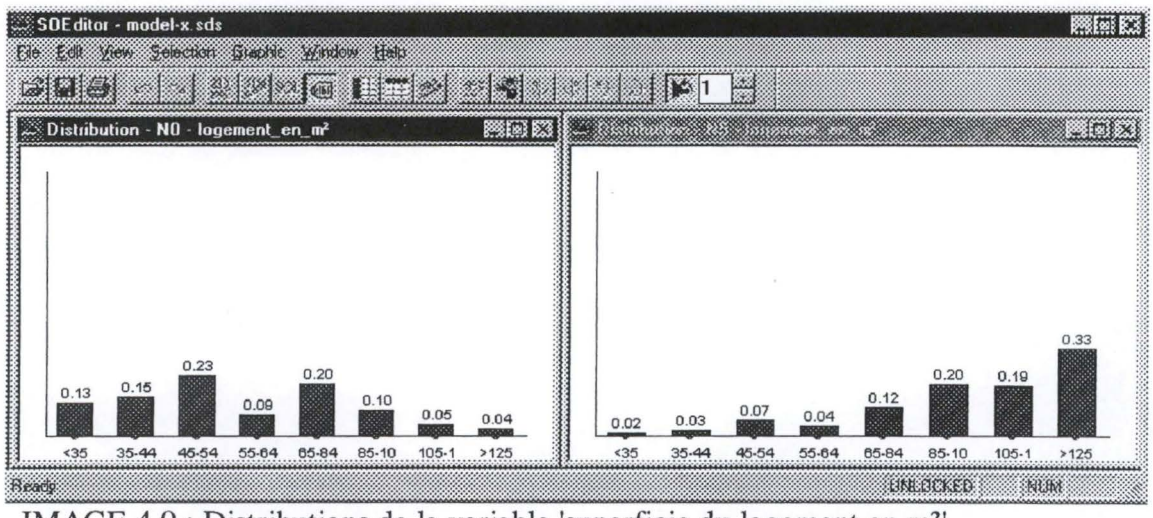


IMAGE 4.9 : Distributions de la variable 'superficie du logement en m²'.
Source : SOEditor

On voit sur les graphiques qu'il y a en effet une différence importante entre ces deux régions. Pour la région Namur-Centre, ce sont plutôt les logements avec des petites et moyennes superficies qui sont majoritaires. La catégorie avec la proportion la plus élevée est la catégorie 45-54 m². C'est entièrement différent pour la région Namur-Citadelle, où il y a une grande prédominance des logements avec une grande superficie. La catégorie avec la proportion la plus élevée pour la région Namur-Citadelle est la catégorie > 125 m². Dans la région Namur-Centre, seulement 19 % des logements ont

une superficie plus grande que 85 m², comparé aux 72 % des logements dans la région Namur-Citadelle. Pour cette variable "superficie des logements", il y a donc une grande différence.

On peut encore analyser la distribution d'une autre variable, comme par exemple la variable "confort des logements privés". Cette distribution est représentée par l'image 4.10 ci-dessous.

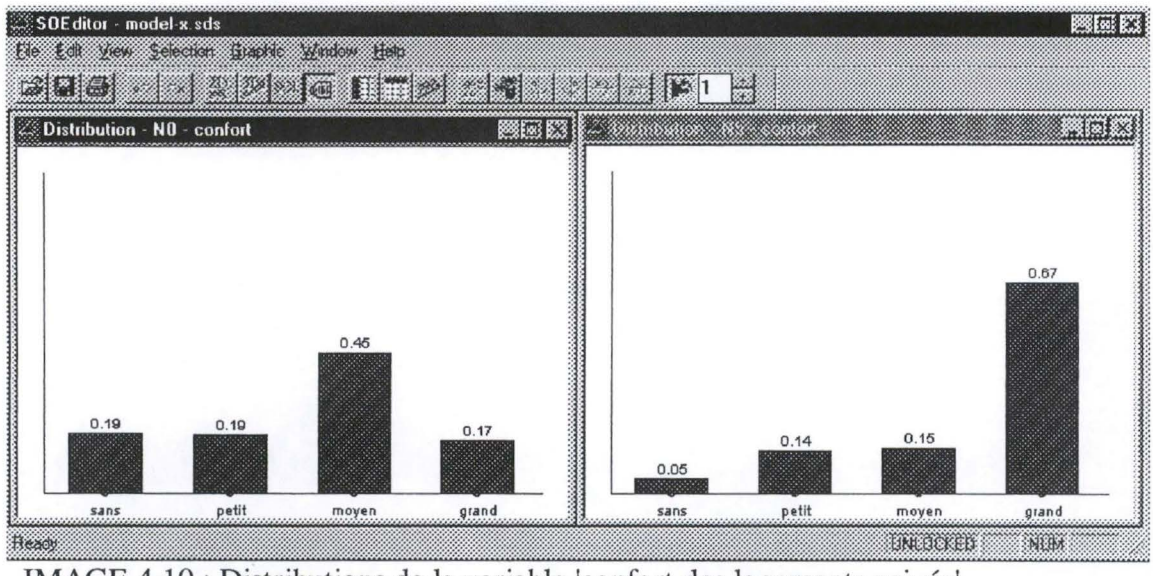


IMAGE 4.10 : Distributions de la variable 'confort des logements privés'.
Source : SOEditor

A nouveau, l'on voit qu'il y a des différences entre ces deux régions. La distribution de Namur-Centre (gauche) montre que la plupart des logements se trouvent dans la catégorie "moyen confort" (45 %). Par contre, si l'on regarde la distribution de Namur-Citadelle (droite), on peut voir une domination des logements qui sont dans la catégorie "grand confort" (67 %). De plus la proportion des logements dans la catégorie "sans confort" est devenue très petite.

On peut donc conclure que la division que l'on avait obtenue avec une classification divisive était normale. La région Namur-Citadelle est assez différente des autres régions du centre. Il est donc normal que cette région ne soit pas incluse dans la même classe que les autres.

Il serait également intéressant de pouvoir représenter les différentes classes qu'on a obtenu avec la classification divisive, en forme d'étoile. Si l'on pouvait représenter chacune des classes sous forme d'étoile, on pourrait facilement comparer les trois classes à l'aide de l'etoile zoom.

Mais une analyse de cette forme n'est cependant pas évidente à exécuter en SODAS. Il est vrai qu'un fichier de sortie du format ".cl" est créé par la méthode DIV avec la division des différentes régions dans les classes. Mais on doit quand même reconstruire un nouveau modèle de données pour pouvoir faire une analyse pareille. On doit passer à nouveau par DB2SO pour créer trois objets symboliques (un objet par classe) en utilisant une requête SQL de type 1. La représentation graphique des trois étoiles est montrée dans l'image 4.11 ci-dessous.

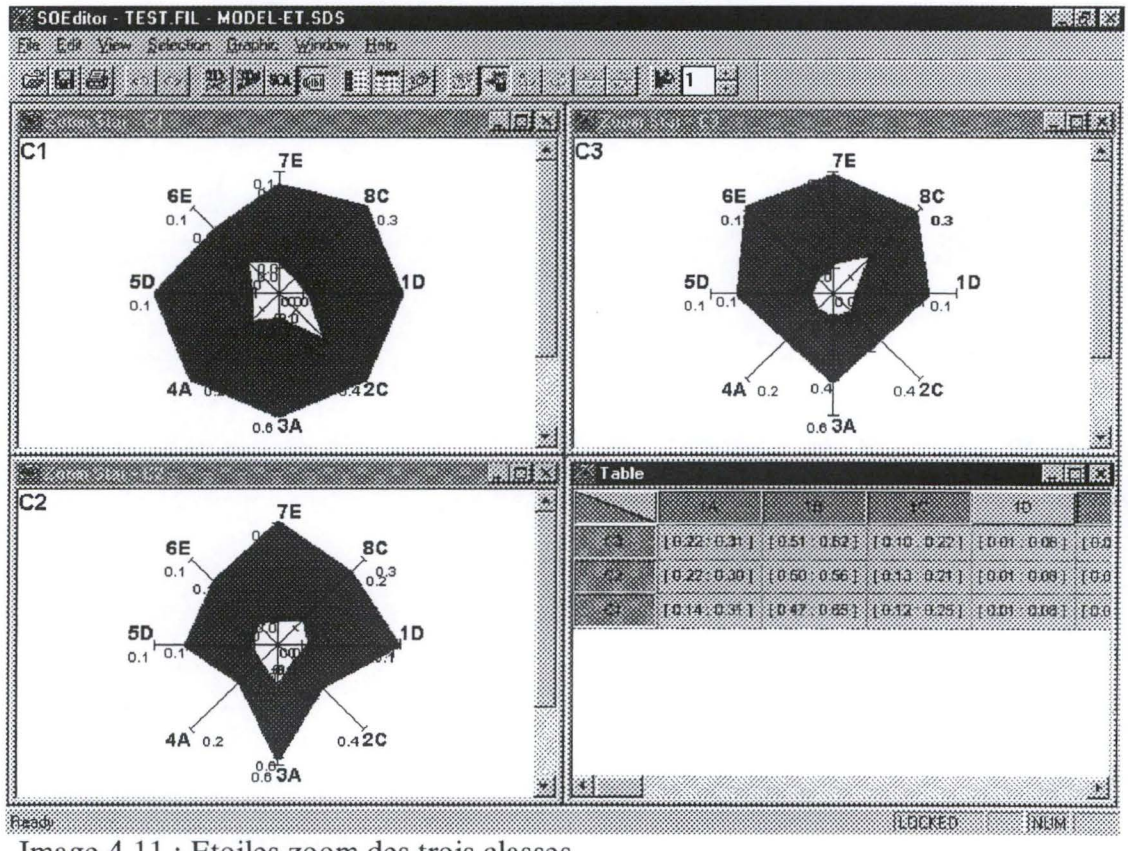


Image 4.11 : Etoiles zoom des trois classes.
Source : SOEditor

On constate sur l'image que l'étoile de la première classe (à gauche au-dessus) montre en général des intervalles de valeurs plus élevées que les deux autres classes. On retrouve donc à nouveau les mêmes conclusions qu'auparavant pour la première classe,

avec beaucoup de personnes âgées, des proportions élevées de logements avec petite surface et peu de confort, ... Mais il est surtout intéressant de trouver des différences entre les étoiles de la deuxième et la troisième classe. On voit qu'il existe en effet une différence pour la variable "année de construction des logements privés" (axe en bas dans l'image 4.11). La deuxième classe semble avoir une proportion plus élevée de logements vieux, ce que l'on avait également pu conclure de l'image 4.7 avec le changement des localisations des catégories. Une autre différence entre la deuxième et la troisième classe est trouvée en analysant la variable "population active par groupe d'âge" (axe à gauche en haut). La troisième classe a une proportion plus élevée de personnes âgées dans la population active que la deuxième classe.

Une analyse supplémentaire que l'on peut encore faire, est un biplot des trois classes qu'on a obtenues dans l'analyse précédente. On prend pour cela deux variables pour représenter les classes. Les carrés qu'on obtient sont les représentations des trois classes avec toutes les régions qui se trouvent dans cette classe. Ce biplot est montré dans l'image 4.12 ci-dessous.

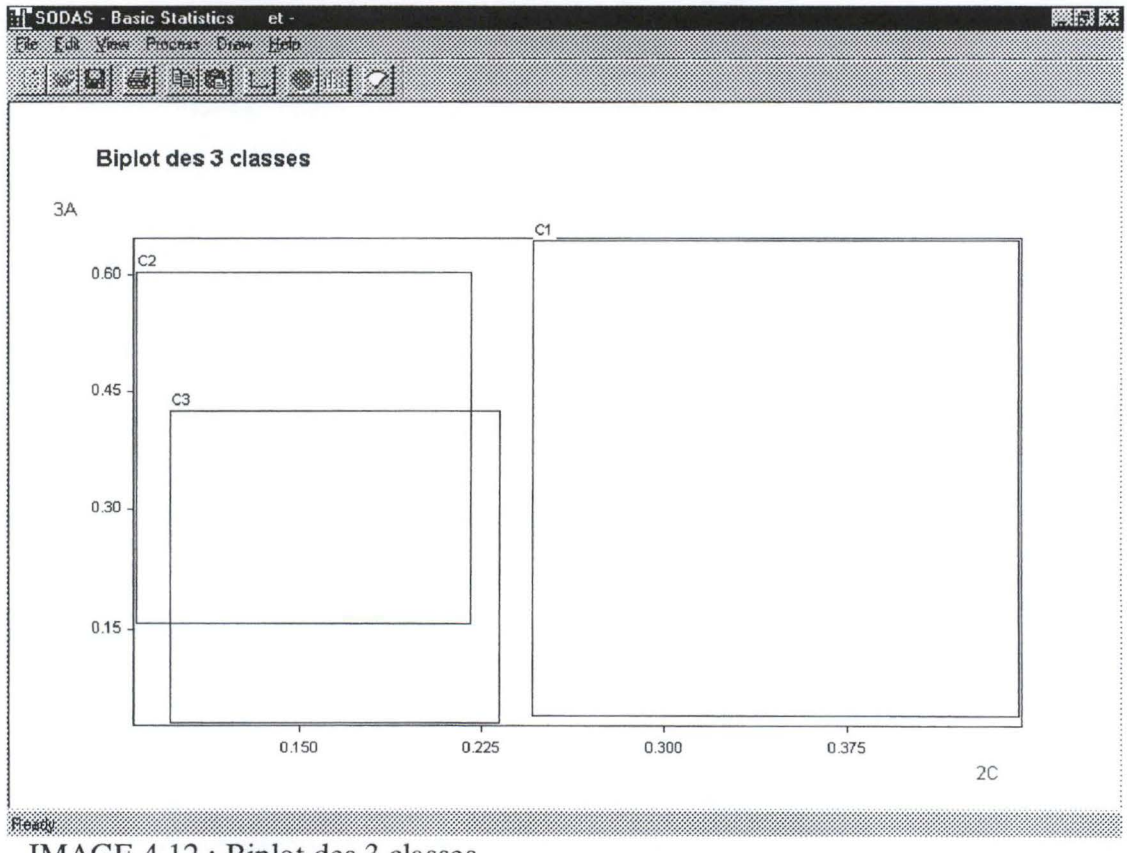


IMAGE 4.12 : Biplot des 3 classes.
Source : SODAS

Les variables qui sont représentées dans le biplot sont "confort des logements privés" et "année de construction des logements privés". On voit que la première classe se trouve assez éloignée des autres classes. Il n'y a même pas de chevauchement entre la première classe et les autres, ce qui est dû à la première variable. Cela peut également être observé dans l'image 4.11 sur l'axe à droite en bas. Les régions de la première classe ont donc des proportions plus élevées de logements avec moyen confort tandis que les autres ont plus de logements avec grand confort.

Il existe par contre un chevauchement important entre le carré de la deuxième et de la troisième classe. La deuxième classe se situe un peu plus en haut par rapport à la troisième classe. Cela représente donc une proportion plus élevée de logements vieux dans la deuxième classe, un résultat que l'on a déjà pu constater auparavant.

4.2.4. GIS

Il serait intéressant de voir si l'on trouve les mêmes résultats, avec des analyses réalisées au moyen d'un logiciel qui travaille sur des données cartographiques. Cela permettrait également de mieux positionner géographiquement les résultats que l'on a trouvé précédemment. Le logiciel utilisé pour faire ces analyses est MapInfo Professional version 5.5, qui est utilisé au département de géographie.

Nous avons choisi de commencer cette analyse avec la variable qui était utilisée dans la classification divisive. C'était la variable "superficie du logement" qui servait à faire une première distinction entre les classes, comme on le voit dans l'arbre de classification (image 4.2). L'image 4.13 sur la page suivante montre la représentation de l'analyse de cette variable.

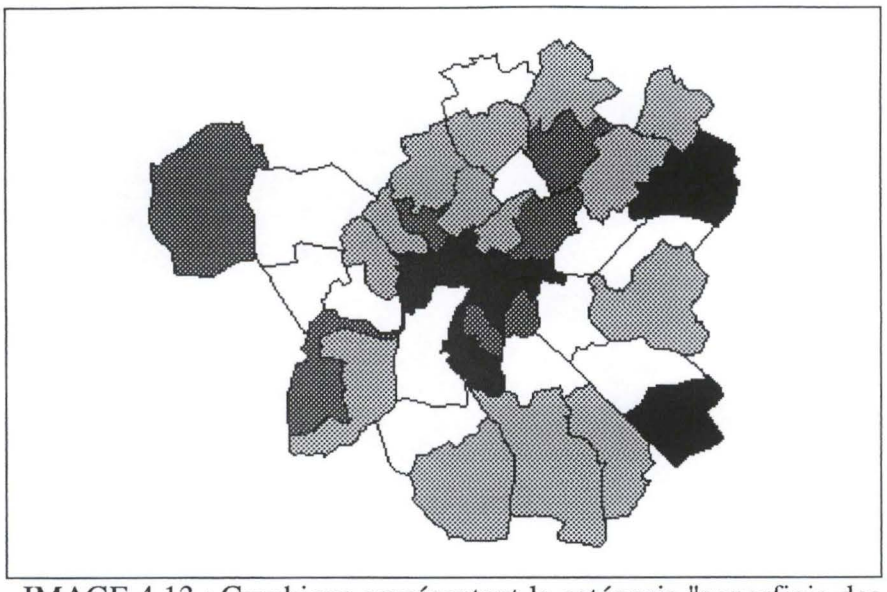


IMAGE 4.13 : Graphique représentant la catégorie "superficie des logements < 35 m²". Noir : beaucoup de logements avec petite superficie. Blanc : peu de logements avec petite superficie.
Source : MapInfo

On peut voir sur cette image que la plupart des régions qui ont beaucoup de logements avec une petite superficie se trouvent dans le centre de Namur. Ce résultat est le même que celui que l'on avait trouvé dans les analyses avec SODAS. On voit cependant aussi quelques régions qui sont coloriées en noir et qui se trouvent assez éloignées du centre. Il s'agit des régions MD (Marche-les-Dames) et A1 (Wierde) à droite sur la carte et de la région TX (Temploux) à gauche. Si on regarde les données de ces régions, on constate qu'elles ont en effet une proportion au-dessus de la moyenne pour les logements avec une superficie < 35 m². La raison pour laquelle ces régions n'étaient pas incluses dans la première classe de la classification divisive, pourrait être que dans cette analyse-ci on tient compte que de la catégorie < 35 m². Dans la classification divisive, on a en effet pris en considération toutes les catégories de la variable pour faire la division.

La deuxième variable qui était utilisée pour faire la classification divisive était la variable "confort des logements privés". Le résultat de l'analyse de cette variable avec MapInfo est représenté sur la page suivante (image 4.14).

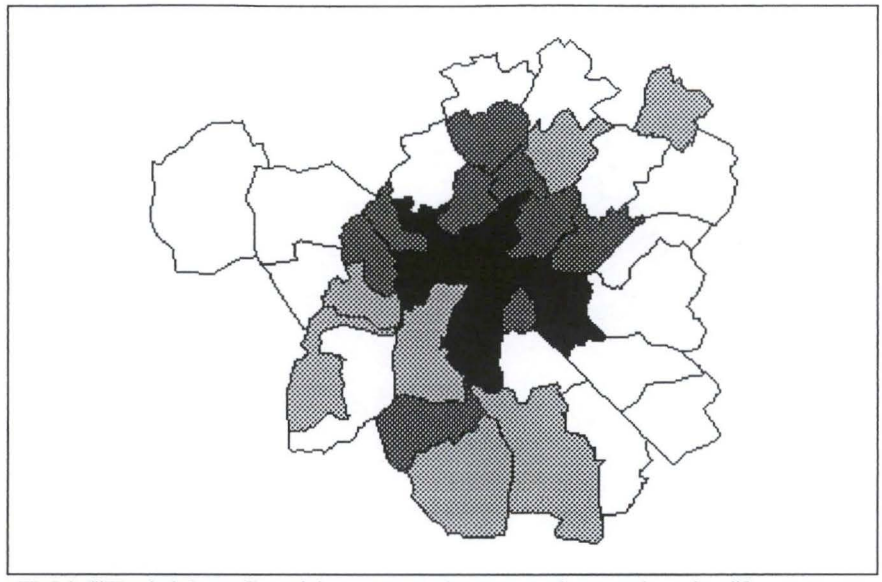


IMAGE 4.14 : Graphique représentant la catégorie "logements privés avec moyen confort". Noir : beaucoup de logements avec moyen confort. Blanc : peu de logements avec moyen confort.
Source : MapInfo

Cette deuxième carte montre encore mieux que la première, un groupement géographique des régions. Les régions qui sont situées dans le centre de Namur semblent avoir des proportions élevées de logements privés avec moyen confort. Les autres régions qui sont plus éloignées du centre ont moins de logements avec moyen confort.

De ce graphique on ne peut pas déduire si ces autres régions sont alors surtout des régions avec grand confort ou plutôt avec petit confort ou même sans confort. L'on est obligé d'avoir d'autres représentations graphiques, ou bien de consulter les données. On pourra alors constater que dans ce cas-ci, les autres régions ont surtout des proportions plus élevées pour la catégorie des logements avec grand confort. On peut d'ailleurs aussi voir ce résultat sur l'image 4.10 qui montre une prédominance des logements avec grand confort pour la région de Namur-Citadelle.

Les deux cartes précédentes montrent donc des résultats logiques et cohérents. Les logements qui sont plus éloignés du centre, sont en général des logements qui sont plus grands que les logements du centre, et ils ont aussi plus de confort que les logements du centre.

On pourra maintenant regarder quelques autres variables qui n'étaient pas incluses dans la classification divisive. Comme cela, on pourra voir s'ils présentent aussi un groupement géographique très clair comme les deux variables que l'on vient d'analyser. Nous commencerons par la variable qui mesure l'année de construction des logements privés. Le graphique avec les résultats de l'analyse avec MapInfo est représenté ci-dessous (image 4.15).

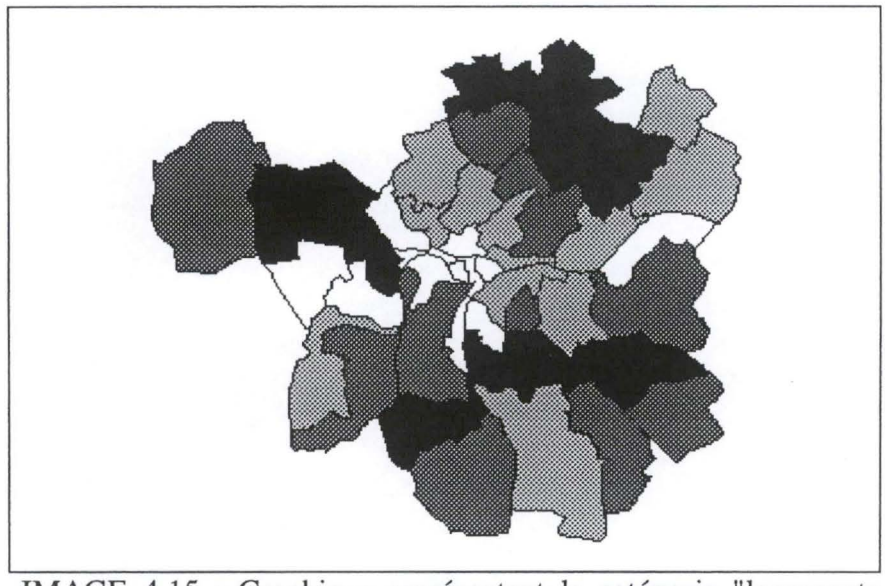


IMAGE 4.15 : Graphique représentant la catégorie "logements privés construits après 1981". Noir : beaucoup de logements privés construits après 1981. Blanc : peu de logements privés construits après 1981.
Source : MapInfo

Le graphique montre également un groupement des régions du centre de Namur, mais l'effet est beaucoup moins clair qu'avant. Dans le centre, il semble y avoir peu de logements privés construits après 1981, tandis que dans les banlieues, la proportion semble être beaucoup plus élevée. Mais l'on voit qu'il y a plusieurs régions qui ne satisfont pas à cette règle, comme par exemple la région LI (Lives-Brumagne) à droite sur l'image. Dans cette région il n'y a presque pas de logements privés construits après 1981. Et si on regarde le fichier des données, on constate que plus de 60 % des logements privés de cette région datent d'avant 1919.

Pour finir, l'on pourrait encore analyser une dernière variable. Nous avons choisi la variable qui représente les personnes des différents groupes d'âges habitant dans un logement. La catégorie de cette variable qui est représentée sur l'image ci-dessous (image 4.16) est le groupe d'âge de 6 à 11 ans.

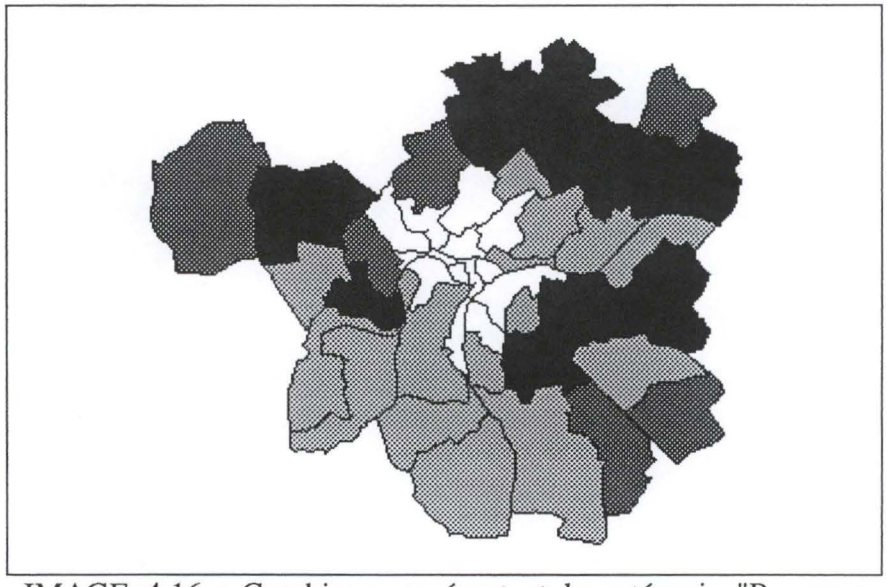


IMAGE 4.16 : Graphique représentant la catégorie "Personnes habitant dans les logements 6-11 ans". Noir : beaucoup de personnes 6-11 ans. Blanc : peu de personnes 6-11 ans.
Source : MapInfo

Les différentes régions de Namur montrent un groupement très élevé pour cette variable. Les régions du centre de Namur semblent avoir des proportions moins élevées de personnes du groupe d'âge de 6-11 ans, tandis que les régions qui sont plus éloignées du centre semblent avoir des proportions plus élevées. Et cette tendance semble être plus forte pour les différentes régions qui se situent au nord du centre que pour les régions du sud. Il serait intéressant de pouvoir analyser encore d'autres variables comme par exemple le nombre d'écoles secondaires, pour voir s'il y a une corrélation avec cette variable-ci.

Ces quelques analyses avec un logiciel pour traiter des données cartographiques, montrent que la classification divisive que l'on avait fait auparavant, était pertinente. Les différentes cartes que nous avons analysées dans cette dernière partie montrent toutes dans une certaine mesure un groupement des régions du centre de Namur. Ces

régions semblent donc avoir des caractéristiques particulières par rapport aux régions qui sont plus éloignées du centre.

Conclusion

Dans ce mémoire, nous avons effectué quelques analyses basées sur des données provenant des quartiers urbains de Namur. Pour ces analyses, des méthodes d'analyse spécialisées ont été mises en œuvre. Les logiciels spécifiques, qui ont été utilisés pendant les analyses, ont d'abord été présentés et leurs principales fonctionnalités sont été expliquées. Il s'agit principalement du logiciel SODAS et d'un de ses modules SOEditor, permettant des analyses avec l'étoile zoom. Le logiciel MapInfo, axé sur l'analyse avec de données cartographiques, a également été utilisé.

Les analyses ont montré que les différentes régions géographiques de Namur ont des caractéristiques assez différentes. Les analyses ont permis de distinguer des groupes de régions ayant des caractéristiques identiques ou assez semblables. La plupart des régions qui se situent dans le centre de Namur par exemple, présentent une très grande ressemblance. L'importance des techniques de visualisation pour découvrir des tendances intéressantes a été montrée. La complémentarité des deux types de logiciels, utilisant différentes types de représentations graphiques a également été montrée.

Une deuxième partie de ce mémoire consistait à développer un programme qui devait être capable de créer une liaison entre deux types de représentations graphiques, les représentations sous forme d'étoile et les représentations sous forme de cartes géographiques. Ce sont les deux types, qui sont également utilisés pour faire les analyses des données.

Bien que nous ayons rencontré quelques difficultés pendant l'implémentation du programme, nous avons quand même réussi à fournir un produit fini. Le programme peut être utilisé pour exécuter les deux types d'analyse, comme demandé dans les spécifications. Il serait néanmoins intéressant de continuer la recherche sur les cartes géographiques. Le type de fichiers ".dxf" offre en effet un moyen intéressant pour faire des transferts de données à partir des fichiers graphiques.

ANNEXES

ANNEXE 1 :

LISTE DES PARTENAIRES DE SODAS

- Thomson-CSF Division RCM : entreprise hi-tech française.
- Université Paris-IX Dauphine : département LISE-CEREMADE.
- FORTH – DIRT : département d'information pour la recherche et la technologie de la Crète.
- Facultés Universitaires Notre-Dame de la Paix Namur : Institut d'Informatique.
- INE : Instituto Nacional de estatistica Portugal.
- Central Statistical Office / Royal Holloway.
- CISIA : Centre international de statistique et d'informatique appliquées de France.
- CRP-CU STADE : Centre de Recherche Public du Luxembourg.
- DIB : Dipartimento di informatica en Italie.
- DMS : Université de Naples – Italie, Departement de Mathématiques et de Statistiques.
- Electricité de France – DER : Direction des Etudes et Recherches.
- EUSTAT – Euskal Estatistika Etakunden : Statistical Institute of Autonomous Community of Basco Country.
- INRIA : Institut National pour la Recherche en Informatique en France.
- LEAD : Laboratorio de Estatistica e Analise de Dados de l'université de Lisbonne en Portugal.
- RWTH : Institut de statistique à Aachen.
- SESRW : Service des Etudes et de la Statistique du ministère de la Région Wallonne.
- UCM : Universidad compiutense de Madrid

ANNEXE 2 :

LISTE DES 46 RÉGIONS DÉFINIES PAR PROF. POULAIN

Namur-Centre (N0)	St-Servais-ch de BXL (S1)	Temploux (TX)
Namur-Casernes (N1)	Beez-Foret (BZ)	Vedrin (V0)
Namur-Parc-Facultés (N2)	Belgrade (BL)	Vedrin-Arquet (V1)
Namur-St-Nicolas (N3)	Boninne (BN)	Vedrin-Comognes (V2)
Namur-La Plante (N4)	Bouge (B0)	Dave (DV)
Namur-Citadelle (N5)	Moulin à Vent (B1)	Erpent (EP)
Namur-Salzinnes (N6)	Champion (CH)	Lives-Brumagne (LI)
Namur-Bomel (N7)	Cognelée (CG)	Loyers (LO)
Namur-Balances (N8)	Daussoulx (DX)	Malonne-Fonds (M0)
Namur-Salz.-Sources (N9)	Flawinne (F0)	Malonne-Hauts (M1)
Jambes-Centre (J0)	Flawinne-Comognes (F1)	Naninne (NN)
Jambes-Amée (J1)	Gelbressée (GB)	Wépion (W0)
Jambes-Casernes (J2)	Marche-les-Dames (MD)	Wépion-Fooz (W1)
Jambes-Montagne (J3)	St-Marc (SM)	Andoy (A0)
Jambes-Géronsart (J4)	Suarlée-Rhisnes (SL)	Wierde (A1)
St-Servais (S0)		

LISTE DES TABLEAUX

LISTE DES IMAGES

BIBLIOGRAPHIE

[BISDORFF&98] Bisdorff, R., Noirhomme-Fraiture, M. & Rouard, M. (1998). *Utilisation de l'étoile zoom en exploration de données statistiques.* Proceedings of KESDA '98, Luxembourg, 27-28 April 1998, EUROSTAT. pp. 251-263.

[BOCK&00] Bock, H.H. & Diday E. (eds.) (2000). *Analysis of symbolic data. Exploratory methods for extracting statistical information from complex data.* Springer Verlag, Heidelberg. ISBN 3-540-66619.

[BODART98] Bodart F. (1998). *Cours introductif à la conception des interfaces homme-machine.* Institut d'Informatique – FUNDP. pp. 112.

[CHAVENT&98] Chavent, M., Hébrail, G., Lechevallier Y. & Stéphan, V. (1998). *SODAS – WP1 processing of ONS family expenditure survey.* INRIA / EDF-DER. pp. 83

[DIDAY93] Diday, E. (1993). *An introduction to symbolic data analysis.* Tutorial of the 4th conference of IFCS. Rapport INRIA n° 1936, Paris.

[HEBRAIL99] Hebrail, G. (1999). *DB2SO – Building symbolic objects from relational databases – Software user manual.* EDF-DER. pp. 24.

[JAMBU98] Jambu, M. (1998). *Introduction au data mining - Méthodes de base de l'analyse des données.* Collection technique et scientifique des télécommunications, Eyrolles. pp. 411.

[LEPRINCE&99] Leprince, V. & Morineau, A. (1999). *SODAS – Software usual manual version 2*. CISIA – CERESTA. pp. 131.

[MATHOT97] Mathot, V. (1997). *Une première approche de l'analyse des données symboliques.* FUNDP - Facultés des sciences - Département de mathématiques. pp. 172.

[NOIRHOMME&97] Noirhomme-Fraiture, M. & Rouard, M. (1997). *Zoom star : a solution to complex statistical object representation.* INTERACT '97, Sydney, July 14-18.

[NOIRHOMME&98] Noirhomme-Fraiture, M. & Rouard, M. (1998). *Visualisation de données multivariés : évaluation de la représentation en étoile zoom, in IHM 98*. Nantes. pp. 121-126.

[NOIRHOMME99] Noirhomme-Fraiture, M. (1999). *Introduction to data mining.* Notes du cours de data mining. FUNDP - Institut d'informatique. pp. 19.

Autodesk, Inc. *AutoCAD 2000 DXF reference.* Version u15.0.02. Site web : http://www.autodesk.com

CISIA (1998) – Centre international de statistique et d'informatique appliquées de France. *Logiciel SODAS.* Version 1.032.

CISIA-CERESTA (1999) – Centre international de statistique et d'informatique appliquées de France. *Logiciel SPAD.* Version 4.02.

ESRI - *Environmental Systems Research Institute, Inc.* Site web : http://www.esri.com/

European Commision Directorate General III. *SODAS – Symbolic Official Data Analysis System – Annex I.* Industrial RTD. pp. 37.

Institut d'Informatique – FUNDP (1998). *Module SOEditor : symbolic objects editor.* Version 2.1.