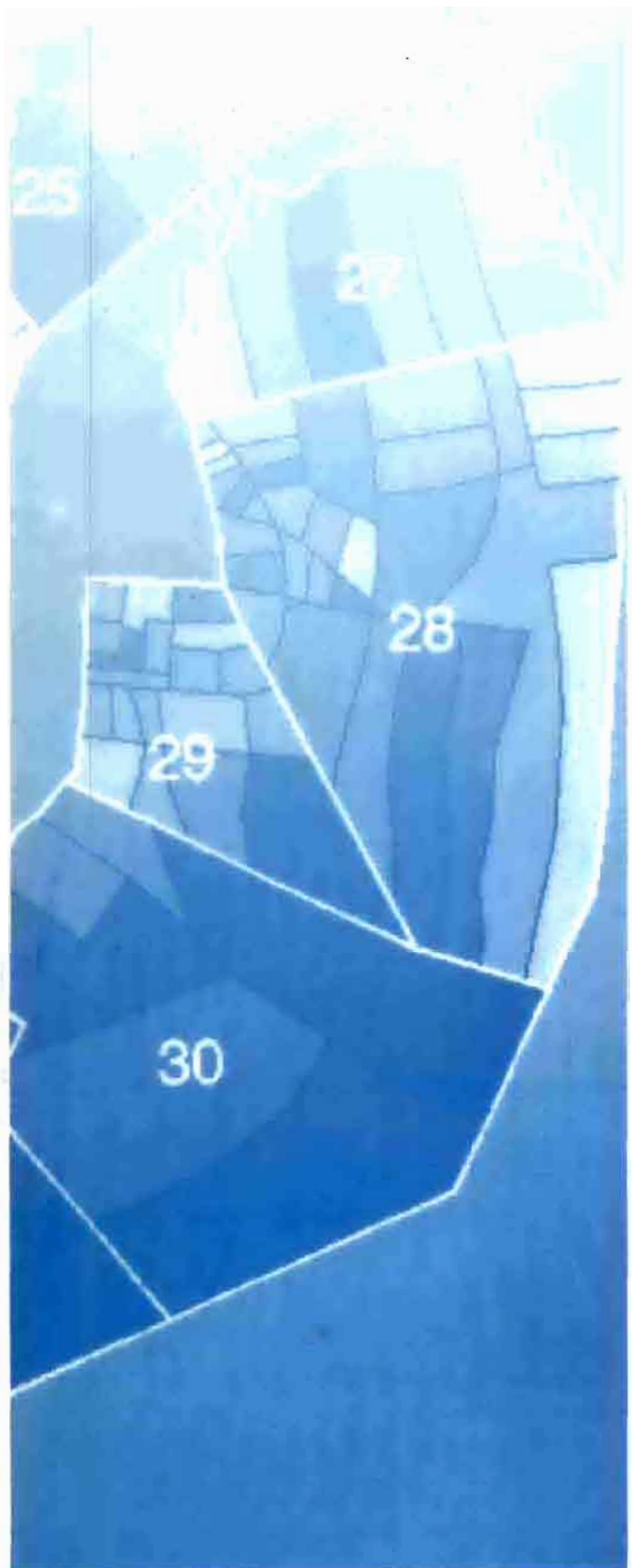


BRUSE

Dominique
COURET

Françoise
PELLETIER



**DE
L'ENQUÊTE
À LA
CARTE
AVEC *SAS*[®]**

CRISTOM
Éditions

DE L'ENQUÊTE À LA CARTE
AVEC *SAS*®

Dominique COURET
Françoise PELLETIER

DE L'ENQUÊTE À LA CARTE AVEC *SAS*®




SEP. 1994

La publication de cet ouvrage a reçu l'agrément de *SAS Institute*

ORSTOM Éditions

INSTITUT FRANÇAIS DE RECHERCHE SCIENTIFIQUE POUR LE DÉVELOPPEMENT EN COOPÉRATION

Collection  ACTIQUES

PARIS 1994

F 39360 ex 2

La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'article 41, d'une part, que les «copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective» et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, «toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause, est illicite» (alinéa 1^{er} de l'article 40).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles 425 et suivants du Code pénal.

AVANT-PROPOS

À l'origine de la conception de ce document est un constat : avec SAS n'est pas bon ouvrier celui qui a toujours ses outils! SAS est certes une merveilleuse boîte à outils mais bien lourde et le large éventail des possibilités offertes pour un même traitement rend vite le néophyte perplexe! Et comme tout logiciel international de bon aloi, SAS s'accompagne d'une superbe documentation... en anglais « technique »!!! qui n'est pas facile d'accès aux non-initiés. Aussi, autant que le temps passé par les uns à découvrir telle possibilité ne soit pas réinvesti à l'infini par les autres!

C'est pourquoi il nous a semblé utile de transmettre, sous forme d'exemples, les résultats de nos premiers pas avec SAS. Les procédures présentées ici n'ont donc pas la prétention d'être les plus rapides, ni les plus logiques, ni les plus « académiques », ni les seules... Les exemples peuvent quelque fois apparaître peu significatifs du point de vue de l'interprétation, les données qui ont été utilisées n'étaient pas, en effet, toujours adaptées. L'effort a porté avant tout sur la mise en valeur des possibilités graphiques de SAS notamment pour la réalisation de cartographies thématiques et de graphiques élaborés. De part la spécificité de l'Orstom, un grand nombre d'équipes de recherche sont amenées soit à réaliser leurs propres collectes de données, soit à exploiter les ensembles d'informations produits localement par les organismes officiels. Si les outils statistiques informatisés font déjà partie des étapes usuelles de « dépouillement d'enquête », il existe encore une grande méconnaissance des outils graphiques, en particulier de ceux qui font intervenir la localisation géographique de l'information. Dans ce cadre, nous avons donc travaillé à mettre en œuvre l'ensemble des traitements nécessaires qui permettent de passer des résultats d'enquête à la carte; l'impasse sur les nombreuses possibilités de traitement statistique offertes par ailleurs par SAS est donc intentionnelle.

Les sorties graphiques noir et blanc servant d'illustrations ont été réalisées avec une imprimante Laserwriter Plus, et les images couleurs sont des recopies d'écran sur imprimante couleur Canon FP-510.

Ce document complète les deux autres fascicules existants sur le traitement des données avec le logiciel SAS et largement utilisés à l'Orstom :

- « Pratique de l'analyse statistique sur PC/PS, mini et gros systèmes », M. Cosinschi et Ph. Waniez, Collection Reclus Modes d'Emplois, Ed. GIP RECLUS/MGM, Montpellier 1989.
- « Manuel de référence ADDADSAS », P. O. Flavigny, M. O. Lebeaux, M. Piron, document interne LIA, Bondy 1989.

L'ENVIRONNEMENT DE TRAVAIL AVEC SAS

Généralités sur une session SAS

Ouvrir une session SAS, c'est accéder à la librairie SAS. Chaque appel à SAS active la lecture d'un fichier **config.sas**. SAS recherche ce fichier dans le système de fichiers. La recherche commence dans le répertoire courant, se poursuit dans le répertoire associé au **login** et finalement dans le répertoire de la librairie SAS si ce fichier n'existait pas dans les niveaux inférieurs. **config.sas** contient les options d'environnement, options automatiquement exécutées à l'ouverture d'une session. Outre les chemins d'accès aux différents modules exécutables de SAS, ce fichier précise le mode de la session, interactive ou non, et le chemin du répertoire de la librairie temporaire de travail propre à SAS. SAS crée dans ce répertoire un sous-répertoire dont le nom est de la forme **SAS#workxnnnn** où se placeront les tableaux de travail créés par l'utilisateur et ceux générés par SAS. Ce sous-répertoire et les fichiers qu'il contient sont détruits automatiquement à la fin de la session, sauf s'il y a fin anormale de la session auquel cas il faut penser à les effacer soi-même.

Dans les brochures de documentation de SAS, il est fortement conseillé à chaque utilisateur de contrôler son environnement de session en paramétrant un fichier **config.sas** qui se trouverait dans son répertoire courant.

Une session interactive et la gestion des fenêtres

Par défaut dans notre environnement, les sessions SAS sont des sessions interactives. Taper **sas** fait apparaître sur l'écran trois fenêtres de travail avec, de bas en haut :

- la fenêtre **pgm**, éditeur où sera introduit le programme à exécuter. L'éditeur dispose de fonctions permettant d'écrire ou de modifier un programme (cf. SAS Language Guide 6.03, pages 332 et suivantes),
- la fenêtre **log** déroule les étapes du programme et affiche les messages d'erreurs : erreurs de syntaxe ou numéros des observations posant des problèmes,
- la fenêtre **output** affiche les résultats des programmes.

La première ligne d'une fenêtre est la ligne de commande; c'est là que l'utilisateur introduit les différentes commandes : ordres d'exécution, sauvegarde de résultats, etc. Nous appellerons fenêtre active, la fenêtre où est positionné le curseur. À l'ouverture de la session, la fenêtre active est la fenêtre **pgm**.

Pour exécuter un programme, il faut qu'il soit chargé dans l'**éditeur**. Si le programme existe dans un fichier externe, la commande **include nomprogramme** introduit le programme dans l'éditeur. Taper la commande **submit** ou **sub** (ou appuyer sur la touche fonction équivalente du clavier) provoque l'exécution du programme. La lettre **R** est inscrite sur le cadre inférieur droit de l'**éditeur** durant l'exécution du programme. Le déroulement des différentes étapes du programme apparaissent commentées dans la fenêtre **log**, ainsi que les erreurs éventuelles rencontrées pendant l'exécution. Les résultats non graphiques s'affichent dans la fenêtre **output**, et les résultats graphiques, s'il y en a, apparaissent à l'écran ou dans un fichier selon le choix de l'utilisateur. On notera que SAS n'interrompt pas un programme lorsqu'il y a erreur, et exécutera les étapes ultérieures pour lesquelles les tableaux existent. Cela peut en particulier avoir pour conséquence d'obtenir des résultats qui correspondent à un programme précédent; nous reviendrons sur ce sujet lorsque nous parlerons des tableaux SAS dans la première partie de ce fascicule. Taper **file nomfichier** sur la ligne commande d'une des trois fenêtres permet de sauvegarder le contenu de la fenêtre dans le fichier **nomfichier**. La commande **endsas** met fin à la session interactive.

SAS dispose d'une liste de mots-clés pour se déplacer ou travailler dans les fenêtres (cf. SAS Language Guide 6.03, chapitre 10). Nous avons vu **submit**, **file**, **include**. Taper **log** ou **pgm** ou **out** sur une ligne commande déplace le curseur respectivement dans les fenêtres **log**, **pgm** ou dans la fenêtre **output**; **clear** nettoie la fenêtre active, **recall** recharge dans l'éditeur le précédent programme, **top** réaffiche les premières lignes, etc. Ces fonctions peuvent être activées à l'aide de touches du clavier alphanumérique. Pour personnaliser les touches du clavier, taper **keys** sur une ligne commande. Une nouvelle fenêtre s'ouvre, disposant aussi d'une ligne commande. À certaines touches du clavier (colonne de gauche dans la fenêtre) sont associées par défaut des mots-clés; on peut redéfinir ces touches et en définir de nouvelles (cf. SAS Language Guide 6.03, pages 366 et suivantes). Cette modification de l'environnement de travail sera valable pour toutes les sessions si l'on tape **save** sur la ligne commande; sinon elle cessera à la fin de la session. Et **end** referme cette fenêtre.

Une session non interactive SAS

Une session non interactive ne dure que le temps nécessaire à l'exécution d'un programme. Comme nous l'avons dit dans les généralités d'une session SAS, l'appel à SAS active la lecture du fichier **config.sas** contenant les options d'environnement. Introduire l'option **-nodms** dans le fichier **config.sas**, rendra la session de SAS non interactive. La commande

sas nomprog

lancera alors l'exécution du programme écrit en langage SAS contenu dans le fichier **nomprog**. Il est possible aussi de soumettre une session non interactive sans modifier le fichier **config.sas** initial. Il faut alors faire suivre l'appel à SAS de l'option **-nodms** et lui indiquer le nom du programme à exécuter, précédé du mot-clé **-sysin**, soit :

sas -nodms -sysin nomprog

Les résultats que nous aurions obtenus en session interactive dans les fenêtres **output** et **log** sont sauvegardés automatiquement dans des fichiers dont le nom par défaut est celui du programme suivi respectivement de l'extension **.lst** et **.log**. Des options permettent de définir soi-même les noms de ces fichiers.

Pour un complément d'information sur les options de configuration, cf. SAS Language Guide 6.03, pp. 463-469.

Les instructions globales

Les paramètres de l'environnement de travail, initialisés à l'ouverture de la session, sont modifiables. Par exemple, on peut vouloir changer le nombre de lignes d'erreurs s'affichant dans la fenêtre **log**, ou centrer les résultats dans la fenêtre **output** et donc éventuellement dans un fichier. Ce sont des instructions (exemples : **options**, **title**) qui modifieront ces paramètres. Nous les appellerons globales, parce qu'effectives durant toute la session si elles ne sont pas redéfinies.

Options

Cette instruction peut être introduite à n'importe quel moment durant une session ou un programme. Elle restera valide jusqu'à ce qu'une nouvelle instruction **options** vienne la modifier.

exemple : **options nodate nocenter;**

Cette instruction supprime l'affichage de la date et le centrage des résultats dans la fenêtre **output**.

syntaxe : **options option1 ... optionn;**

Les options possibles les plus courantes :

- **pagesize=n** ou **ps=n** : nombre de lignes dans une page; 21 lignes par défaut,
- **linesize=n** ou **ls=n** : nombre de caractères par ligne; 78 caractères par défaut. *n* est compris entre 64 et 256,
- **number** / **nonumber** : numéro / pas de numéro de page sur les résultats,
- **center** / **nocenter** : affichage des résultats centré dans la page / affichage calé à gauche,
- **date** / **nodate** : affichage / pas d'affichage de la date,
- **firstobs=n** : *n* est le numéro d'ordre de la première observation à traiter dans un tableau ou un fichier externe,
- **obs=n** : nombre d'observations à traiter.

Les options ci-dessus, options de l'instruction **options** affectent l'environnement de la session (modification de l'affichage ou paramètres précisant les informations à traiter).

Title

Cette instruction globale permet d'afficher sur les différentes sorties du programme SAS, des chaînes de caractères sur une ou plusieurs lignes.

syntaxe : **titlen 'chaîne';**

L'impression se fera sur la *n*ème ligne (*n* variant de 1 à 10). Dans une même session, il y a mémorisation des titres d'une soumission à l'autre donc réaffichage des titres précédents s'ils n'ont pas été réinitialisés ou supprimés par l'instruction :

syntaxe : **title;**

Pour supprimer tous les titres de la *n*ème à la dernière ligne, on utilisera l'instruction :

syntaxe : **titlen;**

Syntaxe générale d'une étape SAS

Un programme SAS est un ensemble d'instructions SAS regroupées en étapes. Chacune de ces étapes correspond à un traitement statistique, graphique ou de gestion de données. La syntaxe générale est donc la suivante :

```
NOM D'ETAPE    optionA ... optionN;
                INSTRUCTION1 / option1;
                INSTRUCTION2 / option2 option3;
```

- **l'instruction de début d'étape : le nom d'étape**

syntaxe : **proc nom d'étape**;

Il existe une exception, l'étape **data** pour laquelle on écrit directement **data**, non précédé de **proc**.

syntaxe : **data nom**;

L'étape **data** est réservée à la gestion de données : introduction, création, calcul; **proc** correspond aux étapes de traitements statistiques ou graphiques, de description ou de visualisation des données.

- **les instructions supplémentaires**

syntaxe : **nom de l'instruction**;

Au sein de chaque étape, une ou plusieurs instructions peuvent se succéder. Le point virgule sépare le niveau Etape et le niveau Instruction ou indique la fin d'une instruction.

- **de nombreuses options**

Ces options peuvent être introduites soit au début de l'étape, soit au niveau instruction supplémentaire suivant leur action. Selon le niveau, la syntaxe est alors différente : le séparateur utilisé au niveau Etape est le caractère blanc, celui du niveau Instruction peut être un blanc ou un blanc suivi d'un /. Dans les deux cas, l'option est placée avant le point virgule du niveau. Si plusieurs options se succèdent, elles sont séparées par un blanc.

syntaxe : **nom de l'option** ou **nom de l'option=**

exemple : **noprint** ou **out=**

N.B. : la manière de déclarer les tableaux sources et les tableaux de travail est différente dans une **étape data** et dans une étape **proc**

```
data tableau de sortie; set tableau d'entrée;
```

set est une instruction; elle est séparée de l'étape **data** par un point virgule.

```
proc nomprocédure data=tableau d'entrée out=tableau de sortie;
```

Au sein de l'étape, **data** et **out** sont des options; elles sont séparées par un blanc et placées avant le point virgule. Le tableau dont le nom suit l'option **out** contient les résultats de l'étape.

Les fonctions SAS arithmétiques, trigonométriques, mathématiques, statistiques, logiques, etc.

Elles sont utilisées soit pour réaliser des calculs, soit pour faire des statistiques ou des opérations sur les observations de une ou plusieurs variables d'un tableau. Elles sont introduites au niveau Etape au sein de calculs, sélection, etc., ou au niveau Instruction, sous forme d'options.

syntaxe : **nom de la fonction (var)**

ou

nom de la fonction (var1, var2, ..., varn)

Exemples d'utilisation dans une **étape data** :

var=int (var1) : la valeur de var est la partie entière de var1

var=sum (var1,var2) : la valeur de var est la somme de var1 et de var2

var=min (**sum**(var1,var2),var3) : la valeur de var correspond au minimum de la somme de var1 et var2 et de var3.

Exemple d'utilisation de fonction statistique dans une **étape proc** :

```
proc summary data=base.tdc;
                class secteur; var var1-var5;
                output out=unitgeo sum= ;
```

L'étape détermine, pour toutes les variables, les effectifs par secteurs. Le tableau unitgeo contient les résultats.

Pour la liste des fonctions SAS, cf. SAS Language Guide 6.03, chapitre 4.

Remarques sur les programmes des chapitres suivants

Les étapes **proc print** et **proc contents**, fréquentes dans les programmes ici présentés, sont introduites pour permettre de visualiser le contenu d'un tableau ou les valeurs d'observations au fur et à mesure des traitements.

Les résultats non graphiques des programmes sont généralement présentés sur la page de droite tels qu'ils sortent sur l'imprimante, c'est-à-dire avec un en-tête où apparaissent le nom du fichier, la date de sortie sur l'imprimante, et la numérotation des pages du listing.

Dans les programmes SAS, les lignes commentaires sont encadrées par les signes */** et **/*. Dans la documentation présente, elles sont écrites en italique.

I
ETAPE DATA

I.A LES TABLEAUX SAS : INTRODUCTION ET LECTURE DES DONNÉES

Une démarche de cartographie thématique se base généralement sur deux ensembles de données :

- les variables descriptives (qualitatives et quantitatives) des unités géographiques choisies, ce qui se traduit informatiquement par un fichier descriptif,
- une cartographie de ces unités, ce qui se traduit informatiquement par un fichier contenant les coordonnées des unités dans un repère quelconque que nous appellerons fichier géographique.

Une variable commune aux deux fichiers permettra de relier les deux types de données. Ce montage constitue ce que nous appelons dans ce fascicule une base et ce qui s'appelle en terminologie SAS une librairie.

Deux étapes sont donc fondamentales : l'introduction du fichier descriptif d'une part, l'introduction du fichier géographique d'autre part. Dans cette partie nous traitons de la première, c'est-à-dire l'introduction et la manipulation d'un fichier de données descriptives.

Pour introduire des données descriptives, il faut :

- indiquer où trouver le fichier qui les contient : instruction **libname**
- décrire l'enregistrement : instruction **input**
- éventuellement recoder les valeurs des variables introduites : instruction **format**
- préciser le contenu des variables par un libellé : instruction **label**. Ce label, chaîne de 40 caractères maximum, apparaît sur certains résultats des programmes SAS.

Deux types de tableaux existent dans SAS :

- le tableau permanent dont le nom se compose de deux parties : le nom de la librairie (celui du **libname**), relié au nom du tableau par un point.

exemple : *base.etat*

Ce tableau apparaît dans le répertoire défini par le **libname**, avec une extension **.ssd** *

exemple : *etat.ssd*

- le tableau provisoire : **nom du tableau**

exemple : *prov*

Ce tableau est précédé automatiquement du préfixe **work** dans la fenêtre **output** lors de l'affichage de résultats ou dans la fenêtre **log** lors de l'affichage du déroulement des opérations, mais n'apparaît pas dans le répertoire défini par le **libname**.

exemple : *work.prov*

Il est possible de créer un tableau SAS en lui donnant un nom déjà existant. Le tableau original sera détruit lors de l'écriture du nouveau tableau sous le même nom*, sauf si une erreur intervient à la fin de l'étape. Cette remarque est valable pour tous les tableaux, permanents ou temporaires de la librairie et explique pourquoi certains résultats peuvent être incohérents. En effet, si une étape ne s'est pas exécutée correctement, le tableau de sortie des résultats soit sera vide s'il n'existait pas précédemment, soit contiendra les résultats non modifiés d'une étape antérieure. Et on ne peut qu'insister sur la nécessité de contrôler le bon déroulement du programme dans la fenêtre **log** avant même d'examiner les résultats de celui-ci.

* Les concepts sont différents avec la version 6.07 de SAS.

I.A

1.A.1 Lecture des données

I.A.1.1 Lecture d'un fichier externe

```
libname    base          '/usr/usr3/graph/docu';
           base : nom logique de la base SAS
           docu : nom physique du répertoire où se trouvent les tableaux de la base

filename   entre        '/usr/usr3/graph/docu/ouaga2.dat';
           entre : nom logique SAS du fichier externe à introduire
           /usr/usr3/graph/docu/ouaga2.dat : répertoire/nom du fichier externe. Le chemin peut être différent de
           celui du libname

options    ps=60         nodate         nonumber;
           60 lignes par page pas de date pas de numérotation de page

data base.ouaga2; infile entre;
           ouaga2 : nom du tableau permanent SAS créé à partir des informations contenues dans le fichier externe entre
input  ident $ 1-5 axx 6-11 axy 12-17;
           description du format de lecture des données
           nomvar $ : la variable nomvar est de type caractère

proc contents data=base.ouaga2;
           descriptif du tableau SAS créé : nombre d'observations, liste des variables du tableau ouaga2 répertorié dans la
           librairie base
run;
           lance l'exécution du programme
```

I.A.1.2 Les différents modes de lecture possibles d'un fichier externe

```
options    ps=60         nodate         nonumber;

1re solution : lire toutes les variables en utilisant le mode liste

data      prov;
input    ident $ v1 v2;
cards;
aaaa    10 0.79
bbbb    25 1.01
cc      4 1.25
;
           chaque ligne comprise entre cards; et ; correspond à un enregistrement. Le blanc est le séparateur des données
proc contents data=prov;
           impression de la liste des variables du tableau provisoire prov et de leur type
proc print data=prov;
           impression du contenu du tableau prov
```

• • •

I.A.1.1

CONTENTS PROCEDURE

Data Set Name: BASE.OUAGA2 Type:
 Observations: 188 Record Len: 25
 Variables: 3
 Label:

----Alphabetic List of Variables and Attributes----

#	Variable	Type	Len	Pos	Label
2	AXX	Num	8	9	
3	AXY	Num	8	17	
1	IDENT	Char	5	4	

I.A.1.2

CONTENTS PROCEDURE

Data Set Name: WORK.PROV Type:
 Observations: 3 Record Len: 28
 Variables:
 Label:

----Alphabetic List of Variables and Attributes----

#	Variable	Type	Len	Pos	Label
1	IDENT	Char	8	4	
2	V1	Num	8	12	
3	V2	Num	8	20	
	OBS	IDENT	V1	V2	
	1	aaaa	10	0.79	
	2	bbbb	25	1.01	
	3	cc	4	1.25	

• • •

I.A

• • •

2^e solution : lire toutes les variables en utilisant le mode colonne

```
data prov1;
  input  ident $ 1-5          v1 6-9 v2 10-14;
      ident : variable alphanumérique      v1,v2 : variables numériques
cards;
aaaa  10 0.79
bbbb  25 1.01
cc    4 1.25
;
proc contents      data=prov1;  proc  print  data=prov1;
```

3^e solution : ne lire que certaines variables du fichier d'entrée en utilisant le mode colonne

```
data prov2;      input  ident $ 1-5    v2 10-14;
cards;
aaaa  10 0.79
bbbb  25 1.01
cc    4 1.25
;
proc contents      data=prov2;  proc  print  data=prov2;
```

4^e solution : lire toutes les variables en combinant les modes, mode format pour les variables numériques, mode liste pour la variable alphanumérique

```
data prov3;      input  ident $ v1 4. v2 5.2;
cards;
aaaa  10 0.79
bbbb  25 1.01
cc  25 1.01
;
la variable alphanumérique, lue en mode liste doit être suivie d'un séparateur (caractère blanc par défaut)
proc contents      data=prov3;  proc  print  data=prov3;
```

5^e solution : lire toutes les variables en combinant les modes, mode format pour les variables numériques, mode colonne pour la variable alphanumérique

```
data prov4;      input  ident $ 1-5 v1 4. v2 5.2;
cards;
aaaa  10 0.79
bbbb  25 1.01
cc    25 1.01
;
proc contents      data=prov4;  proc  print  data=prov4;
run;
```


I.A

I.A.2 Lecture d'un fichier externe dont les données sont sur plusieurs enregistrements

I.A.2.1 Premier exemple

```

libname base '/usr/usr3/graph90/docu';
      base : nom logique de la base SAS
options ls=80 nodate nonumber nocenter;
filename entre '/usr/usr3/graph90/docu/ouag1';
      entre : nom logique ouag1 : nom physique du fichier d'entrée

data prov; infile entre;
  input ident $4 (v1-v63) (4.0)
        #2 (v64-v127) (4.0)
        #3 (v128-v132) (4.0);
      2e enregistrement pour satisfaire la liste d'entrée
      3e enregistrement pour satisfaire la liste d'entrée
description de l'enregistrement externe : 132 variables numériques de longueur 4 sur 3 enregistrements physiques
drop ident v1-v2 v4-v38 v43 v45-v67 v71-v74
      v76-v127 v130-v132 v69;
suppression de variables de la liste d'entrée
label v3='secteur' v39='pers./menage' v40='sexe'
      v41='age' v42='st.matrimonial' v44='ethnie'
      v68='date arrivee parc' v70='date arrivee ouaga'
      v75='activite' v128='st. occupation' v129='origine';
définition d'un nom pour chaque variable
rename v3=sect v39=pers v40=sexe v41=age
      v42=smat v44=ethn v68=arpr v70=arou
      v75=act v129=orig v128=stoc v133=migr;
changement du nom des variables : ancien nom = nouveau nom

proc contents data=prov;
proc print data=prov (obs=10);
impression des 10 premières valeurs du tableau temporaire prov
run;

```

CONTENTS PROCEDURE

Data Set Name: WORK.PROV Type:
 Observations: 1020 Record Len: 92
 Variables: 11
 Label:

-----Alphabetic List of Variables and Attributes-----

#	Variable	Type	Len	Pos	Label
9	ACT	Num	8	68	activite
4	AGE	Num	8	28	age
8	AROU	Num	8	60	date arrivee ouaga
7	ARPR	Num	8	52	date arrivee parc
6	ETHN	Num	8	44	ethnie
11	ORIG	Num	8	84	origine
2	PERS	Num	8	12	pers./menage
1	SECT	Num	8	4	secteur
3	SEXE	Num	8	20	sexe
5	SMAT	Num	8	36	st.matrimonial
10	STOC	Num	8	76	st. occupation

OBS	SECT	PERS	SEXE	AGE	SMAT	ETHN	ARPR	AROU	ACT	STOC	ORIG
1	16	2	2	65	4	3	85	85	12	3	3
2	16	1	1	30	5	2	82	75	3	3	2
3	16	3	1	29	1	3	79	62	5	1	3
4	16	8	1	41	1	2	67	67	6	1	3
5	16	3	1	70	2	3	57	56	1	1	3
6	16	3	1	28	1	3	58	58	1	3	1
7	16	4	1	31	1	3	76	55	3	1	1
8	16	2	1	28	1	3	76	58	3	3	1
9	16	7	1	53	1	2	83	76	3	1	1
10	16	7	1	37	1	3	75	49	3	1	1

I.A

I.A.2.2 Exemple avec positionnement en colonne

```

libname base '/usr/usr3/graph90/docu';
options ps=60 nonumber nodate nocenter;
filename entre '/usr/usr3/graph90/docu/ouag1';
      entre : nom logique      ouag1 : nom physique du fichier d'entrée

data prov; infile entre;
  input @157 (v1-v4) (4*4.) @177 v5 4.
        #2 @17 (v6-v8) (3*4.) @45 v9 4.
        2e enregistrement pour satisfaire la liste d'entrée
        #3 (v10-v11) (2*4.);
        3e enregistrement pour satisfaire la liste d'entrée
  label v1='pers./menage' v2='sexe'
        v3='age' v4='st.matrimonial'
        v5='ethnie' v6='date arrivee parc'
        v7='migr.intra-urb.' v8='date arrivee ouaga'
        v9='activite' v10='st. occupation'
        v11='origine';
  drop v7;

proc contents data=prov;
run;

```

- **input** : par défaut, la longueur maximale d'un enregistrement lu par SAS est de 256 caractères. L'option **lrecl=longueur** de l'option **infile** permet de redéfinir la longueur de l'enregistrement à lire. L'option **linesize=n** de l'option **infile** limite la lecture aux **n** premières colonnes de l'enregistrement.
- **#n** : numéro d'ordre séquentiel de l'enregistrement à lire
 exemple : input ident \$4 (v1-v63) (4.0)
 #2 (v64-v127) (4.0)
 #3 (v128-v132) (4.0);
 c'est-à-dire lire séquentiellement trois enregistrements physiques pour satisfaire la liste d'entrée. Le premier contient la variable **ident** et les variables **v1** à **v63**; le second, les variables **v64** à **v127** et enfin le troisième enregistrement, de la 128^e variable à la 132^e.
- **@** : déplace, dans l'enregistrement, le pointeur sur la colonne dont le numéro d'ordre correspond à la valeur qui suit **@**.
 exemple : @157 (v1-v4) (4*4.) @177 v5 4.
 c'est-à-dire à partir de la colonne 157, lire quatre variables numériques consécutives, de longueur 4 puis, à partir de la colonne 177, lire la variable **v5** sur 4 positions.
- **label** : libellé (la longueur maximale de la chaîne est de 40 caractères) qui explicite le contenu de la variable référencée. Le label, lorsqu'il existe, remplace le nom de la variable sur certaines sorties d'étapes.
- **length** définit la longueur maximale que peut prendre une variable dans le tableau. La longueur d'une variable numérique peut varier entre 3 et 8 (valeur par défaut), celle d'une variable caractère entre 1 et 200 (12 caractères par défaut). Cette instruction sera utilisée lorsque la longueur d'une variable est supérieure à la longueur que SAS affecte par défaut en fonction du type (alphanumérique ou numérique). La déclaration des longueurs réelles maximales des données, si elles sont inférieures aux options par défaut, permet d'optimiser la taille des tableaux.
 exemple : length var1 \$ 6 var2 4;
 c'est-à-dire la première variable est de type alphanumérique et de longueur 6; la seconde, de type numérique, a une longueur n'excédant pas 4.

CONTENTS PROCEDURE

Data Set Name: WORK.PROV Type:
 Observations: 1020 Record Len: 84
 Variables: 10
 Label:

----- Alphabetic List of Variables and Attributes -----

#	Variable	Type	Len	Pos	Label
1	V1	Num	8	4	pers./menage
2	V2	Num	8	12	sexe
3	V3	Num	8	20	age
4	V4	Num	8	28	st.matrimonial
5	V5	Num	8	36	ethnie
6	V6	Num	8	44	date arrivee parc
7	V8	Num	8	52	date arrivee ouaga
8	V9	Num	8	60	activite
9	V10	Num	8	68	st. occupation
10	V11	Num	8	76	origine

OBS	V1	V2	V3	V4	V5	V6	V8	V9	V10	V11
1	2	2	65	4	3	85	85	12	3	3
2	1	1	30	5	2	82	75	3	3	2
3	3	1	29	1	3	79	62	5	1	3
4	8	1	41	1	2	67	67	6	1	3
5	3	1	70	2	3	57	56	1	1	3
6	3	1	28	1	3	58	58	1	3	1
7	4	1	31	1	3	76	55	3	1	1
8	2	1	28	1	3	76	58	3	3	1
9	7	1	53	1	2	83	76	3	1	1
10	7	1	37	1	3	75	49	3	1	1

I.B SÉLECTION

On peut à l'introduction, décider de ne retenir que certaines variables de la liste d'entrée (I.B.1) ou que les enregistrements satisfaisants à une ou plusieurs conditions (I.B.2).

I.B.1 Sélection de variables

```
libname base '/usr/usr3/graph90/docu';
options linesize=80 ps=60 nodate nonumber;

proc contents data=base.popul position;
descriptif du tableau permanent initial popul : une première liste des variables par ordre alphabétique et une seconde
liste (option position) ordonnée en fonction de la position des variables dans le tableau

avec drop
data prov1;
  set base.popul; drop v2--v11 v24 v15-v20;
création d'un tableau provisoire prov1 à partir du tableau permanent popul et contenant une partie des variables du
tableau initial
proc contents data=prov1;

avec keep
data prov2;
  set base.popul; keep v2--v11 v24 v15-v20;
création d'un tableau provisoire prov2 à partir du tableau permanent popul et contenant une partie des variables du
tableau initial
proc contents data=prov2;
run;
```

- Dans cet exemple, les deux résultats d'**étape data** sont complémentaires : prov1, résultat de l'étape avec **drop**, contient les variables qui sont exclues de prov2, étape avec **keep**.
- Différence entre - et -- :
 - v2--v11 : toutes les variables comprises entre v2 et v11 selon l'organisation interne du tableau c'est-à-dire voisines,
 - v15-v20 : toutes les variables comprises entre v15 et v20 selon l'ordre alphabétique de la liste des variables du tableau.

CONTENTS PROCEDURE

Data Set Name: BASE.POPUL Type:
 Observations: 1131 Record Len: 416
 Variables: 37
 Label:

-----Alphabetic List of Variables and Attributes-----

#	Variable	Type	Len	Pos	Label
9	IDENT	Char	12	68	Número bloc
10	V2	Char	12	80	Quartier
11	V3	Char	12	92	Número rue
12	V4	Char	12	104	Par. inondee
13	V5	Char	12	116	Puits
1	V6	Num	8	4	surf.maison
2	V7	Num	8	12	pers./nuit
3	V8	Num	8	20	anciennete
14	V9	Char	12	128	statut occ.
15	V10	Char	12	140	prof. chef
16	V11	Char	12	152	Electricite
17	V12	Char	12	164	eau
18	V13	Char	12	176	coupure eau
19	V14	Char	12	188	longue coup.
20	V15	Char	12	200	toilettes
21	V16	Char	12	212	marche
4	V17	Num	8	28	age mere
5	V18	Num	8	36	age enfant
22	V19	Char	12	224	sexe enfant
23	V20	Char	12	236	fiche pesee
24	V21	Char	12	248	SMI
6	V22	Num	8	44	poids 0 jour
7	V23	Num	8	52	poids actuel
8	V24	Num	8	60	age pesee
25	V25	Char	12	260	BCG
26	V26	Char	12	272	TETRACOQ
27	V27	Char	12	284	ROUVAX
28	V28	Char	12	296	ROUGEOLE
29	V29	Char	12	308	H. rougeole
30	V30	Char	12	320	TOUX
31	V31	Char	12	332	Soins toux
32	V32	Char	12	344	FIEVRE
33	V33	Char	12	356	Soins fievre
34	V34	Char	12	368	Diarrhees
35	V35	Char	12	380	Soins diarr.
36	V36	Char	12	392	\$ sante
37	V37	Char	12	404	\$ associatif

CONTENTS PROCEDURE

-----Variables Ordered by Position-----

#	Variable	Type	Len	Pos	Label
1	V6	Num	8	4	surf.maison
2	V7	Num	8	12	pers./nuit
3	V8	Num	8	20	anciennete
4	V17	Num	8	28	age mere
5	V18	Num	8	36	age enfant
6	V22	Num	8	44	poids 0 jour
7	V23	Num	8	52	poids actuel
8	V24	Num	8	60	age pesee
9	IDENT	Char	12	68	Número bloc
10	V2	Char	12	80	Quartier
11	V3	Char	12	92	Número rue
12	V4	Char	12	104	Par. inondee
13	V5	Char	12	116	Puits
14	V9	Char	12	128	statut occ.
15	V10	Char	12	140	prof. chef
16	V11	Char	12	152	Electricite
17	V12	Char	12	164	eau
18	V13	Char	12	176	coupure eau
19	V14	Char	12	188	longue coup.
20	V15	Char	12	200	toilettes
21	V16	Char	12	212	marche
22	V19	Char	12	224	sexe enfant
23	V20	Char	12	236	fiche pesee
24	V21	Char	12	248	SMI
25	V25	Char	12	260	BCG
26	V26	Char	12	272	TETRACOQ
27	V27	Char	12	284	ROUVAX
28	V28	Char	12	296	ROUGEOLE
29	V29	Char	12	308	H. rougeole
30	V30	Char	12	320	TOUX
31	V31	Char	12	332	Soins toux
32	V32	Char	12	344	FIEVRE
33	V33	Char	12	356	Soins fievre
34	V34	Char	12	368	Diarrhees
35	V35	Char	12	380	Soins diarr.
36	V36	Char	12	392	\$ sante
37	V37	Char	12	404	\$ associatif

CONTENTS PROCEDURE

Data Set Name: WORK.PROV1 Type:
 Observations: 1131 Record Len: 260
 Variables: 23
 Label:

-----Alphabetic List of Variables and Attributes-----

#	Variable	Type	Len	Pos	Label
6	IDENT	Char	12	44	Número bloc
1	V6	Num	8	4	surf.maison
2	V7	Num	8	12	pers./nuit
3	V8	Num	8	20	anciennete
7	V12	Char	12	56	eau
8	V13	Char	12	68	coupure eau
9	V14	Char	12	80	longue coup.
10	V21	Char	12	92	SMI
4	V22	Num	8	28	poids 0 jour
5	V23	Num	8	36	poids actuel
11	V25	Char	12	104	BCG
12	V26	Char	12	116	TETRACOQ
13	V27	Char	12	128	ROUVAX
14	V28	Char	12	140	ROUGEOLE
15	V29	Char	12	152	H. rougeole
16	V30	Char	12	164	TOUX
17	V31	Char	12	176	Soins toux
18	V32	Char	12	188	FIEVRE
19	V33	Char	12	200	Soins fievre
20	V34	Char	12	212	Diarrhees
21	V35	Char	12	224	Soins diarr.
22	V36	Char	12	236	\$ sante
23	V37	Char	12	248	\$ associatif

CONTENTS PROCEDURE

Data Set Name: WORK.PROV2 Type:
 Observations: 1131 Record Len: 160
 Variables: 14
 Label:

-----Alphabetic List of Variables and Attributes-----

#	Variable	Type	Len	Pos	Label
4	V2	Char	12	28	Quartier
5	V3	Char	12	40	Número rue
6	V4	Char	12	52	Par. inondee
7	V5	Char	12	64	Puits
8	V9	Char	12	76	statut occ.
9	V10	Char	12	88	prof. chef
10	V11	Char	12	100	Electricite
11	V15	Char	12	112	toilettes
12	V16	Char	12	124	marche
1	V17	Num	8	4	age mere
2	V18	Num	8	12	age enfant
13	V19	Char	12	136	sexe enfant
14	V20	Char	12	148	fiche pesee
3	V24	Num	8	20	age pesee

I.B

I.B.2 Sélection d'observations

```

libname base '/usr/usr3/graph90/docu';
options linesize=80 nodate nonumber;
title 'SELECTION D'OBSERVATIONS';

proc contents data=base.ouaga2; run; title;
descriptif du tableau initial réinitialisation du titre

avec if
data prov1; set base.ouaga2;
  if ident='29 14' or ident='29 15';
le tableau prov1, sous-ensemble de ouaga2, contient les observations pour lesquelles la variable ident remplit la condition exprimée
proc contents data=prov1; proc print data=prov1;

avec where
data prov2; set base.ouaga2;
  where ident='29 14' or ident='29 15';
proc contents data=prov2; proc print data=prov2;

avec where
data prov3;
  set base.ouaga2 (where=(ident='29 14' or ident='29 15'));
proc contents data=prov3; proc print data=prov3;

avec where
data prov4;
  set base.ouaga2 (where=(ident='29 14' or axx=-94));
proc contents data=prov4; proc print data=prov4;

avec where
data prov5;
  set base.ouaga2 (where=(ident ne '29 14' and axx ne -94));
proc contents data=prov5;

run;

```

- **if condition(s)** : le tableau final contient les observations du tableau initial pour lesquelles le test est vrai.
- **if condition(s) then delete** : les observations du tableau initial satisfaisant la condition ne sont pas conservées dans le tableau final.
- Les tableaux de sortie des quatre premières étapes contiennent les mêmes observations : prov1, prov2, prov3, prov4, sont identiques. Le tableau prov5 est le tableau complémentaire de prov1, prov2, prov3 ou prov4 : il contient les observations du tableau initial exclues des étapes précédentes. Ces exemples présentent diverses syntaxes possibles pour aboutir à un même résultat.
On notera que, si les résultats de traitement de tableau de données avec **if** et **where** sont identiques, il existe toutefois une différence dans la gestion interne des tableaux par SAS. Pour un complément d'informations, cf. SAS Language Guide 6.03, pp. 262-265.

SELECTION D'OBSERVATIONS

CONTENTS PROCEDURE

Data Set Name: BASE.OUAGA2 Type:
 Observations: 188 Record Len: 25
 Variables: 3
 Label:

-----Alphabetic List of Variables and Attributes-----

#	Variable	Type	Len	Pos	Label
2	AXX	Num	8	9	
3	AXY	Num	8	17	
1	IDENT	Char	5	4	

CONTENTS PROCEDURE

Data Set Name: WORK.PROV1 Type:
 Observations: 2 Record Len: 25
 Variables: 3
 Label:

-----Alphabetic List of Variables and Attributes-----

#	Variable	Type	Len	Pos	Label
2	AXX	Num	8	9	
3	AXY	Num	8	17	
1	IDENT	Char	5	4	

OBS	IDENT	AXX	AXY
1	29 14	502	298
2	29 15	-94	-43

CONTENTS PROCEDURE

Data Set Name: WORK.PROV2 Type:
 Observations: 2 Record Len: 25
 Variables: 3
 Label:

-----Alphabetic List of Variables and Attributes-----

#	Variable	Type	Len	Pos	Label
2	AXX	Num	8	9	
3	AXY	Num	8	17	
1	IDENT	Char	5	4	

OBS	IDENT	AXX	AXY
1	29 14	502	298
2	29 15	-94	-43

CONTENTS PROCEDURE

Data Set Name: WORK.PROV3 Type:
 Observations: 2 Record Len: 25
 Variables: 3
 Label:

-----Alphabetic List of Variables and Attributes-----

#	Variable	Type	Len	Pos	Label
2	AXX	Num	8	9	
3	AXY	Num	8	17	
1	IDENT	Char	5	4	

OBS	IDENT	AXX	AXY
1	29 14	502	298
2	29 15	-94	-43

CONTENTS PROCEDURE

Data Set Name: WORK.PROV4 Type:
 Observations: 2 Record Len: 25
 Variables: 3
 Label:

-----Alphabetic List of Variables and Attributes-----

#	Variable	Type	Len	Pos	Label
2	AXX	Num	8	9	
3	AXY	Num	8	17	
1	IDENT	Char	5	4	

OBS	IDENT	AXX	AXY
1	29 14	502	298
2	29 15	-94	-43

CONTENTS PROCEDURE

Data Set Name: WORK.PROV5 Type:
 Observations: 186 Record Len: 25
 Variables: 3
 Label:

-----Alphabetic List of Variables and Attributes-----

#	Variable	Type	Len	Pos	Label
2	AXX	Num	8	9	
3	AXY	Num	8	17	
1	IDENT	Char	5	4	

I.C RECODAGE

Dès l'introduction du tableau dans la base, on peut modifier la valeur d'une ou plusieurs variables, par recodage conditionnel, par classification ou par calcul.

I.C.1 Recodage simple conditionnel

```
libname base '/usr/usr3/graph90/docu';
options nodate nonumber ps=60 ls=80;
proc freq data=base.popul; tables v2 v7 v8 v5;
calcul des effectifs par modalité : fréquences absolue et en pourcentage, fréquences cumulée et cumulée en pourcentage
data popul; set base.popul;
  if v2 gt '50' then v2='50';
  if v5 eq '1' then v5='avec puits';
  une condition
  if v7 eq 2 and v8 gt 10 then v8=99;
  conditions multiples
  keep v2 v7 v5 v8;
  sélection de variables
création d'un travail provisoire et recodage de variables
proc print data=base.popul (obs=10); var v2 v7 v8 v5;
impression des 10 premières observations des variables de la liste, variables du tableau permanent popul
proc print data=popul (obs=10);
impression des 10 premières observations de toutes les variables du tableau de travail popul
run;
```

- Opérateurs de comparaison :

eq	ou	=
gt	ou	>
ge	ou	>=
ne		différent de
lt	ou	<
le	ou	<=
- Autres syntaxes des tests :

if ... and ... then;	else ... ;
if ... or ... then;	
if ... eq ... then;	
where ... eq ...; etc.	

Quartier					anciennete				
V2	Frequency	Percent	Cumulative Frequency	Cumulative Percent	V8	Frequency	Percent	Cumulative Frequency	Cumulative Percent
31	96	8.5	96	8.5	0	1	0.1	1	0.1
32	58	5.1	154	13.6	1	140	12.4	141	12.5
34	45	4.0	199	17.6	2	115	10.2	256	22.6
41	84	7.4	283	25.0	3	108	9.5	364	32.2
45	72	6.4	355	31.4	4	65	5.7	429	37.9
46	64	5.7	419	37.0	5	54	4.8	483	42.7
47	84	7.4	503	44.5	6	33	2.9	516	45.6
51	16	1.4	519	45.9	7	32	2.8	548	48.5
52	36	3.2	555	49.1	8	28	2.5	576	50.9
53	38	3.4	593	52.4	9	9	0.8	585	51.7
55	60	5.3	653	57.7	10	23	2.0	608	53.8
57	61	5.4	714	63.1	11	11	1.0	619	54.7
58	225	19.9	939	83.0	12	15	1.3	634	56.1
68	115	10.2	1054	93.2	13	19	1.7	653	57.7
69	76	6.7	1130	99.9	14	8	0.7	661	58.4
absent	1	0.1	1131	100.0	15	15	1.3	676	59.8
					16	22	1.9	698	61.7
					17	19	1.7	717	63.4
					18	17	1.5	734	64.9
					19	11	1.0	745	65.9
					20	34	3.0	779	68.9
					21	20	1.8	799	70.6
					22	16	1.4	815	72.1
					23	9	0.8	824	72.9
					24	7	0.6	831	73.5
					25	22	1.9	853	75.4
					26	4	0.4	857	75.8
					27	11	1.0	868	76.7
					28	9	0.8	877	77.5
					29	6	0.5	883	78.1
					30	12	1.1	895	79.1
					31	2	0.2	897	79.3
					32	5	0.4	902	79.8
					33	1	0.1	903	79.8
					35	14	1.2	917	81.1
					36	2	0.2	919	81.3
					37	1	0.1	920	81.3
					38	1	0.1	921	81.4
					39	2	0.2	923	81.6
					40	1	0.1	924	81.7
					50	1	0.1	925	81.8
					81	1	0.1	926	81.9
					82	1	0.1	927	82.0
					99	204	18.0	1131	100.0

pers./nuit

V7	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	1	0.1	1	0.1
2	59	5.2	60	5.3
3	189	16.7	249	22.0
4	173	15.3	422	37.3
5	157	13.9	579	51.2
6	141	12.5	720	63.7
7	105	9.3	825	72.9
8	90	8.0	915	80.9
9	67	5.9	982	86.8
10	53	4.7	1035	91.5
11	32	2.8	1067	94.3
12	31	2.7	1098	97.1
13	9	0.8	1107	97.9
14	9	0.8	1116	98.7
15	8	0.7	1124	99.4
16	1	0.1	1125	99.5
18	1	0.1	1126	99.6
19	2	0.2	1128	99.7
20	2	0.2	1130	99.9
23	1	0.1	1131	100.0

Puits

V5	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	1001	88.5	1001	88.5
1	129	11.4	1130	99.9
absent	1	0.1	1131	100.0

OBS	V2	V7	V8	V5
1	absent	0	0	absent
2	34	7	17	0
3	31	5	17	0
4	31	3	22	1
5	31	13	29	1
6	31	6	4	0
7	31	4	99	0
8	31	4	24	0
9	31	3	99	0
10	31	4	2	0

OBS	V7	V8	V2	V5
1	0	0	50	absent
2	7	17	34	0
3	5	17	31	0
4	3	22	31	avec puits
5	13	29	31	avec puits
6	6	4	31	0
7	4	99	31	0
8	4	24	31	0
9	3	99	31	0
10	4	2	31	0

I.C

I.C.2 Recodage par classification

I.C.2.1 Variable numérique

```
libname base '/usr/usr3/graph90/docu';
options nodate nonumber ps=80 ls=60;
title 'CLASSIFICATION DE VARIABLE NUMERIQUE';
```

```
proc print data=base.ouaga2 (obs=10);
impression des 10 premières observations du tableau ouaga2
```

```
proc format; value fxy
  low - -500 = 'moins de -500'
  -500 - 0 = 'de -500 a 0'
  0 - 500 = 'de 0 a 500'
  500 - high = 'plus de 500';
```

définition de classes pour une variable numérique dans un format appelé fxy

```
data prov; set base.ouaga2; format axx fxy;
```

création d'un tableau provisoire prov à partir du tableau permanent ouaga2 et classification de la variable axx selon les critères définis dans le format fxy

```
proc print data=prov (obs=10);
impression des 10 premières observations de toutes les variables du tableau de travail prov
run;
```

I.C.2.2 Variable alphanumérique

```
libname base '/usr/usr3/graph90/docu';
options ls=80 ps=60 nodate nonumber nocenter;
title 'CLASSIFICATION DE VARIABLE ALPHANUMERIQUE';
```

```
proc print data=base.ouaga2g (firstobs=850 obs=870);
impression de 20 observations de toutes les variables de ouaga2g, à partir de l'observation 850
```

```
proc format; value $fid
  '15 :-'19999:' = '1'
  '20 :-'29999:' = '1'
  '30 :-'30999:' = '1'
  other = '0';
```

définition de classes pour une variable alphanumérique dans un format appelé fid. Si l'on avait écrit '15 :-'high='1', la classe 1 comprendrait toutes les valeurs supérieures à 15 ainsi que les valeurs 2 et 3, valeurs que nous voulons faire appartenir à la classe 0

```
data ouagag; set base.ouaga2g; format ident $fid;
```

création du tableau provisoire ouagag et classification de la variable ident selon les critères définis dans le format fid

```
proc print data=ouagag (firstobs=850 obs=870);
impression de 20 observations du tableau temporaire ouagag, à partir de l'observation 850
run;
```

• • •

• • •

- **value nom** : définition, dans un format appelé **nom**, d'un recodage applicable à une variable numérique. L'utilisation de ce format permettra dans une étape ultérieure, d'associer à chaque valeur d'une variable numérique, la chaîne de caractères correspondant à la classe à laquelle elle appartient.

syntaxe : **value** valeurs='chaîne'

- **value \$nom** : définition, dans un format appelé **nom**, d'un recodage applicable à une variable alphanumérique. L'utilisation de ce format substituera la chaîne de caractères correspondant à la valeur initiale par la nouvelle chaîne.

syntaxe : **value** 'chaînes'='chaîne'

- **low, high** et **other** sont des mots-clés SAS. **Low** et **high** correspondent respectivement à la plus petite et à la plus grande valeur pouvant exister dans le tableau sur lequel s'appliquera la classification (attention à « l'ordre » lorsqu'il y a des chiffres dans le nom des variables alphanumériques). **Other** permet de rassembler dans une même classe, un ensemble de valeurs sans avoir à les énumérer toutes (cf. I.C.2.2).

I.C.2.1

CLASSIFICATION DE VARIABLE NUMERIQUE

OBS	IDENT	AXX	AXY
1	15 1	-386	4
2	15 2	-176	-240
3	15 3	93	-235
4	15 4	125	-425
5	15 5	-130	-247
6	16 1	-383	-172
7	16 2	374	-154
8	16 3	-59	-471
9	16 4	-330	-128
10	16 5	-287	-177

CLASSIFICATION DE VARIABLE NUMERIQUE

OBS	IDENT	AXX	AXY
1	15 1	de -500 a 0	4
2	15 2	de -500 a 0	-240
3	15 3	de 0 a 500	-235
4	15 4	de 0 a 500	-425
5	15 5	de -500 a 0	-247
6	16 1	de -500 a 0	-172
7	16 2	de 0 a 500	-154
8	16 3	de -500 a 0	-471
9	16 4	de -500 a 0	-128
10	16 5	de -500 a 0	-177

I.C.2.2

CLASSIFICATION DE VARIABLE ALPHANUMERIQUE

OBS	IDENT	X	Y	SEGMENT
850	19 14	9044	3362	1
851	19 14	9034	3360	1
852	19 14	9028	3351	1
853	19 14	9025	3341	1
854	19 14	9018	3341	1
855	19 14	9006	3349	1
856	19 14	9003	3340	1
857	2 0	9446	2968	1
858	2 0	9629	2969	1
859	2 0	9625	3109	1
860	2 0	9618	3216	1
861	2 0	9569	3206	1
862	2 0	9453	3121	1
863	2 0	9466	3067	1
864	2 0	9455	3018	1
865	20 1	8869	3469	1
866	20 1	8876	3485	1
867	20 1	8882	3501	1
868	20 1	8900	3521	1
869	20 1	8918	3537	1
870	20 1	8934	3553	1

CLASSIFICATION DE VARIABLE ALPHANUMERIQUE

OBS	IDENT	X	Y	SEGMENT
850	1	9044	3362	1
851	1	9034	3360	1
852	1	9028	3351	1
853	1	9025	3341	1
854	1	9018	3341	1
855	1	9006	3349	1
856	1	9003	3340	1
857	0	9446	2968	1
858	0	9629	2969	1
859	0	9625	3109	1
860	0	9618	3216	1
861	0	9569	3206	1
862	0	9453	3121	1
863	0	9466	3067	1
864	0	9455	3018	1
865	1	8869	3469	1
866	1	8876	3485	1
867	1	8882	3501	1
868	1	8900	3521	1
869	1	8918	3537	1
870	1	8934	3553	1

I.C

I.C.2.3 Création de nouvelles variables après recodage

```
libname base '/usr/usr3/graph90/docu';
options ls=80 ps=60 nodate nonumber nocenter;
title 'CLASSIFICATION ET CREATION DE VARIABLE';
```

```
proc format; value fpers
  low-1 = '1'
  2-5 = '2'
  6-9 = '3'
  10-high = '4';
value fage
  15-25 = '1'
  26-35 = '2'
  36-45 = '3'
  46-55 = '4'
  56-high = '5';
```

définition de deux formats de classification applicables à une ou plusieurs variables numériques

```
data prov; set base.enquete;
length persc agec 4;
déclaration de la longueur des deux variables numériques
persc=put(pers,fpers.);
agec=put(age,fage.);
création de deux nouvelles variables numériques. La valeur de la variable persc, pour chaque observation, est la classe à laquelle appartient la valeur correspondante de la variable pers dans le tableau enquete
keep sect pers persc age agec;
sélection de variables
proc print data=prov (obs=10);
run;
```

CLASSIFICATION ET CREATION DE VARIABLE

OBS	SECT	PERS	AGE	PERSC	AGEC
1	16	2	65	2	5
2	16	1	30	1	2
3	16	3	29	2	2
4	16	8	41	3	3
5	16	3	70	2	5
6	16	3	28	2	2
7	16	4	31	2	2
8	16	2	28	2	2
9	16	7	53	3	4
10	16	7	37	3	3

I.C

I.C.3 Recodage par calcul

```

libname base '/usr/usr3/graph90/docu';
options ls=80 ps=60 nodate nonumber nocenter;
title 'RECODAGE PAR CALCUL';

proc freq data=base.popul; tables v4 v7;
data popul; set base.popul;
  if v7 lt 10 then v7=100+v7;
  première condition premier calcul
  else
  if v7 ge 20 then v7=300+v7;
  seconde condition second calcul
  recodage conditionnel de modalités de la variable v7
  v4=substr(v4,1,2);
  extraction d'une sous-chaîne
  keep v4 v7;
proc freq data=popul; tables v4 v7;
impression, par modalité, des fréquences des variables v4 et v7
run;

```

- Autres calculs possibles sur variables numériques :

```

var1 = var1/var2;
var1 = var1*var2;
var1 = var1*100/var2;
var1 = sum(var1,var2,var3), etc.

```

- Méthodes de recodages possibles pour les variables alphanumériques :

- Extraction de sous-chaînes : **substr**

syntaxe : **substr (chaîne, position, n)**

la fonction **substr** réalise l'extraction à partir de la variable **chaîne**, d'une sous-chaîne commençant au caractère spécifié par **position**, et ayant une longueur de **n** caractères.

exemple : `nvellev=substr(var1,2,3);`

- Concaténation de chaînes : **!!**

syntaxe générale : **chaîne1!!chaîne2**

!! ajoute le contenu de la variable **chaîne2** à la suite du dernier caractère de **chaîne1**.

Exemple de concaténation de trois chaînes :

```
nvellev=substr(var1,2,3)!!'00'!!substr(var2,2,4);
```

- Pour recoder on peut utiliser :

les opérateurs arithmétiques : +, -, /, *, ** (élévation à la puissance),

les opérateurs logiques : **and**, **or**, **not**,

les opérateurs de comparaison : **gt**, **ge**, **ne**, **lt**, **le**, **eq**,

l'opérateur de concaténation : **!!**,

les fonctions : **min**, **max**, **mod**, **sqrst**, **sign**, **mean**, **n**, **nmiss**, **range**...

RECODAGE PAR CALCUL

Par. inondee

V4	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	965	85.3	965	85.3
1	165	14.6	1130	99.9
absent	1	0.1	1131	100.0

pers./nuit

V7	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	1	0.1	1	0.1
2	59	5.2	60	5.3
3	189	16.7	249	22.0
4	173	15.3	422	37.3
5	157	13.9	579	51.2
6	141	12.5	720	63.7
7	105	9.3	825	72.9
8	90	8.0	915	80.9
9	67	5.9	982	86.8
10	53	4.7	1035	91.5
11	32	2.8	1067	94.3
12	31	2.7	1098	97.1
13	9	0.8	1107	97.9
14	9	0.8	1116	98.7
15	8	0.7	1124	99.4
16	1	0.1	1125	99.5
18	1	0.1	1126	99.6
19	2	0.2	1128	99.7
20	2	0.2	1130	99.9
23	1	0.1	1131	100.0

RECODAGE PAR CALCUL

Par. inondee

V4	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	965	85.3	965	85.3
1	165	14.6	1130	99.9
ab	1	0.1	1131	100.0

pers./nuit

V7	Frequency	Percent	Cumulative Frequency	Cumulative Percent
10	53	4.7	53	4.7
11	32	2.8	85	7.5
12	31	2.7	116	10.3
13	9	0.8	125	11.1
14	9	0.8	134	11.8
15	8	0.7	142	12.6
16	1	0.1	143	12.6
18	1	0.1	144	12.7
19	2	0.2	146	12.9
100	1	0.1	147	13.0
102	59	5.2	206	18.2
103	189	16.7	395	34.9
104	173	15.3	568	50.2
105	157	13.9	725	64.1
106	141	12.5	866	76.6
107	105	9.3	971	85.9
108	90	8.0	1061	93.8
109	67	5.9	1128	99.7
320	2	0.2	1130	99.9
323	1	0.1	1131	100.0

I.D CRÉATION DE NOUVELLES VARIABLES

I.D.1 Initialisation d'une nouvelle variable

Cette initialisation peut se faire par affectation d'une valeur

```

syntaxe :  var=valeur
           ou
           var=chaînedecaractère
  
```

ou par initialisation de la variable.

```

libname    base    '/usr/usr3/graph90/docu';
options    ls=80   ps=60       nodate       nonumber   nocenter;
title      'INITIALISATION DE VARIABLES';

data prov;  set    base.popul;
            retain compteur 0;
            initialise la variable compteur à zéro
            if    v19='1'      then  compteur=compteur+2;
            incrémente la variable compteur de la valeur 2 à chaque nouvelle observation rencontrée pour laquelle v19=1
            keep  ident compteur v19;

proc print  data=prov    (obs=15);
run;
  
```

- Dans l'exemple ici traité, la variable **compteur** est initialisée au début de l'étape à 0. Elle est ensuite modifiée sur condition au cours de l'étape. Si l'on avait simplement

```

data  prov;  set  base.popul;
                compteur=0;
                if condition then résultat ;
  
```

la valeur de la variable **compteur** resterait égale à 0 pour toutes les observations du tableau prov.

- Autre exemple avec utilisation de l'instruction **retain** :

```

data  prov;  set  base.popul;
                retain  compteur 0;
                compteur=compteur+1;
  
```

Cette étape permettrait de créer une variable **compteur** prenant les valeurs 1, 2, 3, 4, 5, 6, 7, et, ... dans l'ordre où se présentent les observations au sein du tableau d'entrée. Dans une **étape data**, SAS crée automatiquement une variable qui s'incrémente ainsi à chaque nouvelle observation traitée, c'est la variable **_N_** que l'on trouve labellée **OBS** sur certaines sorties. Cette variable **_N_** peut être utilisée dans une **étape data** pour accéder à une observation donnée en fonction de sa position dans le tableau.

INITIALISATION DE VARIABLE

OBS	IDENT	V19	COMPTEUR
1	absent	absent	0
2	106	2	0
3	121	1	2
4	123	1	4
5	124	1	6
6	125	1	8
7	126	1	10
8	127	1	12
9	128	1	14
10	129	2	14
11	130	2	14
12	131	1	16
13	133	1	18
14	134	1	20
15	135	1	22

I.D

I.D.2 Création d'une nouvelle variable à partir de conditions sur les variables existantes

```

libname base '/usr/usr3/graph90/docu';
options ls=80 ps=60 nodate nonumber nocenter;
title 'CREATION SUR CONDITION';

proc freq data=base.popul; tables v7;
affichage du nombre d'observations par valeurs de la variable v7
data popul; set base.popul;
  if v7 lt 10 then v70=100+v7;
  première condition premier calcul
  if v7 ge 10 and v7 lt 20 then v70=200+v7;
  seconde condition second calcul
  if v7 ge 20 then v70=300+v7;
  troisième condition troisième calcul
  keep v7 v70;
  label v70='variable calculée';
création de la variable v70 calculée à partir des conditions exprimées sur la variable v7

proc freq data=popul; tables v70;
affichage du nombre d'observations par valeurs de la variable v70
run;

```

I.D.3 Création d'une nouvelle variable par calcul sur les variables existantes

syntaxe : $nvar = var1 \text{ opérateur } var2$

$nvar$ est le résultat d'une opération arithmétique (+, -, *, /, **), celui d'une opération logique ou un indice calculé à partir des fonctions SAS (**sum**, **max**, **min**, etc.). Cf. SAS Language Guide 6.03, page 42.

CREATION SUR CONDITION

pers./nuit

V7	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	1	0.1	1	0.1
2	59	5.2	60	5.3
3	189	16.7	249	22.0
4	173	15.3	422	37.3
5	157	13.9	579	51.2
6	141	12.5	720	63.7
7	105	9.3	825	72.9
8	90	8.0	915	80.9
9	67	5.9	982	86.8
10	53	4.7	1035	91.5
11	32	2.8	1067	94.3
12	31	2.7	1098	97.1
13	9	0.8	1107	97.9
14	9	0.8	1116	98.7
15	8	0.7	1124	99.4
16	1	0.1	1125	99.5
18	1	0.1	1126	99.6
19	2	0.2	1128	99.7
20	2	0.2	1130	99.9
23	1	0.1	1131	100.0

CREATION SUR CONDITION

variable calculee

V70	Frequency	Percent	Cumulative Frequency	Cumulative Percent
100	1	0.1	1	0.1
102	59	5.2	60	5.3
103	189	16.7	249	22.0
104	173	15.3	422	37.3
105	157	13.9	579	51.2
106	141	12.5	720	63.7
107	105	9.3	825	72.9
108	90	8.0	915	80.9
109	67	5.9	982	86.8
210	53	4.7	1035	91.5
211	32	2.8	1067	94.3
212	31	2.7	1098	97.1
213	9	0.8	1107	97.9
214	9	0.8	1116	98.7
215	8	0.7	1124	99.4
216	1	0.1	1125	99.5
218	1	0.1	1126	99.6
219	2	0.2	1128	99.7
320	2	0.2	1130	99.9
323	1	0.1	1131	100.0

I.E GESTION DE TABLEAUX

fusion=« mise à côté », concaténation=« mise à la suite »

I.E.1 Fusion de tableaux SAS

```
libname base '/usr/usr3/graph90/docu';
options ls=80 ps=60 nodate nonumber nocenter;
title 'FUSION DE TABLEAUX';

proc contents data=base.tdc position;
descriptif du tableau initial

data prov1; set base.tdc; keep ident age1-age5;
data prov2; set base.tdc; keep ident prs1-prs4;
data prov3; set base.tdc; keep ident sect sex1 sex2;
création, pour la démonstration, de trois tableaux différents à partir d'un tableau de la base. Les trois tableaux ont une variable commune ident.

proc sort data=prov1; by ident;
tri du premier tableau sur la variable de jointure retenue
proc sort data=prov2; by ident;
tri du second tableau sur la variable de jointure retenue
proc sort data=prov3; by ident;
tri du troisième tableau sur la variable de jointure retenue

data result; merge prov2 prov3 prov1; by ident;
jointure des trois fichiers sur la variable commune ident
Dans cet exemple, les trois tableaux étant des sous-ensembles du tableau initial, le tableau result, résultat de la fusion, contient l'ensemble de l'information initiale.

proc contents data=result position;
run;
```

● Schéma illustratif

Tableaux initiaux

prov1				prov2				prov3			
ident	age1	...	age5	ident	prs1	...	prs4	ident	sct	sex1	sex2
1	val1	...	val2	1	x	...	y	1	a	1	2
2	val3	...	val4	2	z	...	w	2	b	2	1
etc.				etc.				etc.			

Tableau final : result

ident	age1	...	age5	prs1	...	prs4	sct	sex1	sex2
1	val1	...	val2	x	...	y	a	1	2
2	val3	...	val4	z	...	w	b	2	1
etc.									

I.E

I.E.2 Concaténation de tableaux

```

libname base '/usr/usr3/graph90/docu';
options ls=80 ps=60 nodate nonumber nocenter;
title 'CONCATENATION DE TABLEAUX';

proc contents data=base.tdc position;

data prov1; set base.tdc; keep ident age1-age5;
data prov2; set base.tdc; keep ident prs1-prs4;
data prov3; set base.tdc; keep ident sect sex1 sex2;
création, pour la démonstration, de trois tableaux différents à partir d'un tableau de la base

data result; set prov2 prov3 prov1;
concaténation des trois fichiers, l'un derrière l'autre. La variable ident commune aux trois tableaux n'apparaît qu'une fois dans la liste des variables du tableau final.

proc contents data=result;
proc print data=result (firstobs=2570);
run;

```

● Schéma illustratif

Tableaux initiaux

prov1				prov2				prov3			
ident	age1	...	age5	ident	prs1	...	prs4	ident	sct	sex1	sex2
1	val1	...	val2	1	x	...	y	1	a	1	2
2	val3	...	val4	2	z	...	w	2	b	2	1
etc.				etc.				etc.			

Tableau final : result

ident	age1	...	age5	prs1	...	prs4	sct	sex1	sex2
1	val1	...	val2	•	...	•	•	•	•
2	val3	...	val4	•	...	•	•	•	•
etc.									
1	•	...	•	x	...	y	•	•	•
2	•	...	•	z	...	w	•	•	•
etc.									
1	•	...	•	•	...	•	a	1	2
2	•	...	•	•	...	•	b	2	1
etc.									

Les points • représentent la valeur manquante (**missing value**). Cette valeur est automatiquement affectée par SAS aux observations des variables qui n'ont pas de valeur, ou qui n'existent pas, dans le tableau initial.

- À propos de la fusion et de la concaténation de tableaux : **merge** effectue une fusion sur la variable commune (identifiant) des tableaux, c'est-à-dire associe à chaque identifiant les valeurs de variables des tableaux initiaux. Le nombre d'observations reste inchangé. La concaténation crée un tableau dont la liste des variables est constituée de l'ensemble des variables des tableaux initiaux. Le nombre d'observations du tableau créé correspond à la somme des nombres d'observations des tableaux initiaux.

II

TRAITEMENTS STATISTIQUES

II.A DESCRIPTION STATISTIQUE DE VARIABLES QUALITATIVES

II.A.1 Effectifs et fréquences simples

Forme générale :

```
proc freq data=base.tableau; tables variables;
effectifs et fréquences des variables de la liste qui suit l'instruction tables

proc freq data=base.tableau;
  tables variables / missprint;
effectifs et fréquences intégrant les valeurs manquantes

proc chart data=base.tableau; vbar variable;
diagramme des fréquences absolues barres verticales; horizontales si hbar

proc chart data=base.tableau; vbar variable / type=percent;
diagramme des fréquences en pourcentage
run;
```

Application :

```
libname base '/usr/usr3/graph90/docu';
options ls=80 ps=60 nodate nonumber nocenter;
title 'FREQUENCES SIMPLES';

proc freq data=base.popul; tables v4 v5;

visualisation
proc chart data=base.popul;
  vbar v4 / type=cfreq;
diagramme en bâtons verticaux des fréquences cumulées
proc chart data=base.popul;
  hbar v4 v5 / type=percent;
diagramme en bâtons horizontaux des pourcentages
run;
```

FREQUENCES SIMPLES

Par. inondee

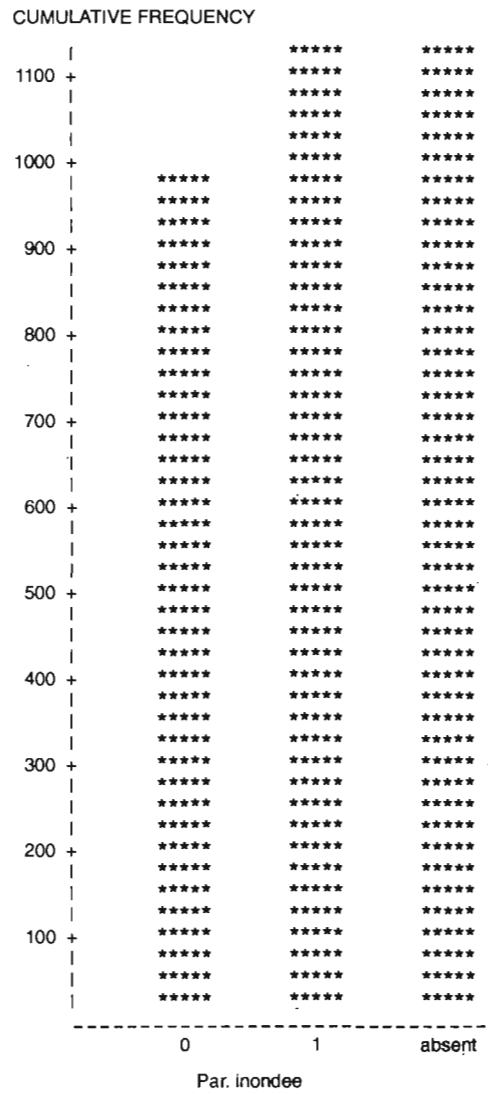
V4	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	965	85.3	965	85.3
1	165	14.6	1130	99.9
absent	1	0.1	1131	100.0

Puits

V5	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	1001	88.5	1001	88.5
1	129	11.4	1130	99.9
absent	1	0.1	1131	100.0

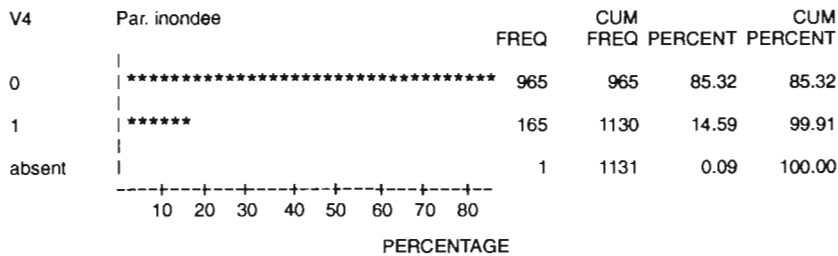
FREQUENCES SIMPLES

CUMULATIVE FREQUENCY OF V4



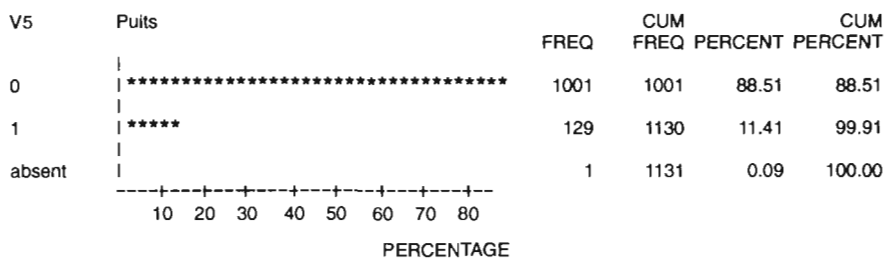
FREQUENCES SIMPLES

PERCENTAGE OF V4



FREQUENCES SIMPLES

PERCENTAGE OF V5



II.A

II.A.2 Effectifs et fréquences en fonction d'une partition

On peut désirer aussi connaître les effectifs ou les fréquences d'une variable donnée en fonction des catégories, classes, modalités ou partition d'une autre variable. Dans l'exemple traité, on s'intéresse au nombre de parcelles inondées et non inondées en fonction de la présence et non-présence d'un puits dans la parcelle.

Forme générale :

```
proc sort  data=base.tableau  out=trav;    by  variable;
tri        tableau d'entrée      tableau de sortie  variable de partitionnement

proc freq  data=trav;
tables var1;          by  var2;
                        variable de partitionnement
tables var1 /        missprint;
                        avec intégration des valeurs manquantes dans les calculs
                        by  var2;

proc chart data=trav;
vbar var1 /          type=cfreq;  by  var2;
un diagramme vertical des fréquences cumulées de var1 par modalité de var2

proc chart data=trav;
hbar var1 /          type=percent; by  var2;
un diagramme horizontal des fréquences en pourcentage de var1 par modalité de var2

run;
```

Application :

```
libname base '/usr/usr3/graph90/docu';
options ls=80 ps=60 nodate nonumber nocenter;
title 'FREQUENCE EN FONCTION D'UNE PARTITION';

proc sort data=base.popul out=popul; by v5;
tri, par défaut selon un ordre croissant, du tableau permanent popul sur la variable de partitionnement v5 et résultat
dans le tableau temporaire popul

proc freq data=popul; tables v4; by v5;

proc chart; hbar v4 / type=percent; by v5;
par défaut si aucun tableau d'entrée n'est indiqué dans l'étape, implicitement le tableau traité sera celui de l'étape
précédente

run;
```

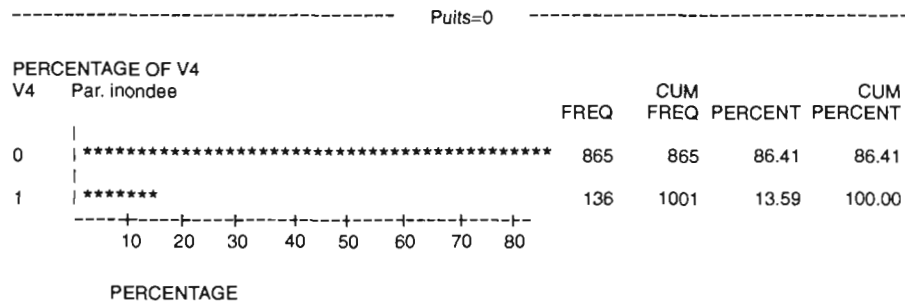
FREQUENCE EN FONCTION D'UNE PARTITION

----- Puits=0 -----

Par. inondee

V4	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	865	86.4	865	86.4
1	136	13.6	1001	100.0

FREQUENCE EN FONCTION D'UNE PARTITION



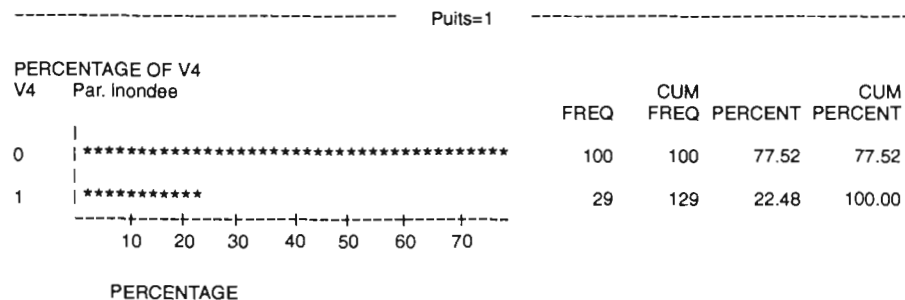
FREQUENCE EN FONCTION D'UNE PARTITION

----- Puits=1 -----

Par. inondee

V4	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	100	77.5	100	77.5
1	29	22.5	129	100.0

FREQUENCE EN FONCTION D'UNE PARTITION



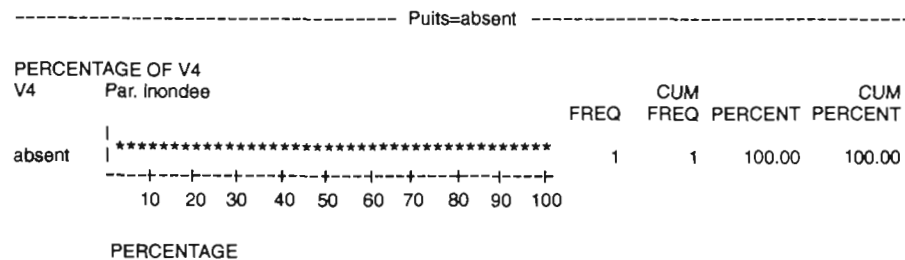
FREQUENCE EN FONCTION D'UNE PARTITION

----- Puits=absent -----

Par. inondee

V4	Frequency	Percent	Cumulative Frequency	Cumulative Percent
absent	1	100.0	1	100.0

FREQUENCE EN FONCTION D'UNE PARTITION



II.A

II.A.3 Tris croisés sur variables qualitatives

Il s'agit de la même démarche que celle qui permet de calculer les effectifs et les fréquences en fonction d'une partition. La différence tient à la présentation qui est ici faite des résultats. Elle est réalisée par défaut sous la forme d'un tableau croisé dont les lignes correspondent aux modalités de la première variable et les colonnes à celles de la seconde. La valeur de chaque cellule du tableau correspond au nombre d'observations communes aux deux modalités croisées.

Forme générale :

```
proc freq data=base.tableau;
  tables var1*var2;
  la liste des variables varie de 2 à n
  tables var1*var2 / chisq;
  avec test d'indépendance du KHI2
  tables var1*var2 / missprint chisq;
  idem en incluant les valeurs manquantes dans les calculs
  tables var1*var2 / missprint norow nofreq;
  pourcentages en colonne sans les effectifs
  tables var1*var2 / missprint nocol nofreq;
  pourcentages en ligne sans les effectifs
  tables var1*var2 / missprint nocol norow;
  effectifs de var1 et var2
  tables var1*var2 / missprint list;
  effectifs et fréquences des combinaisons des modalités
  tables var1*var2*var3 / missprint list;
  effectifs et fréquences des combinaisons des modalités
  tables (var1 var2)*(var3 var4)/ missprint;
  équivalent à var1*var3 var2*var3 var1*var4 var2*var4
  tables var1*var2; weight var3;
  effectifs cumulés de var3 (codée en effectifs) en fonction des modalités croisées de var1*var2
run;
```

Application :

```
libname base '/usr/usr3/graph90/docu';
options ls=80 ps=60 nodate nonumber nocenter;
proc freq data=base.popul;
  title 'TRI CROISE SIMPLE';
  tables v11*v12; run;
tableau de fréquences croisées v11*v12 (eau*électricité); résultats en nombre de parcelles concernées
```

...

TRI CROISE SIMPLE

TABLE OF V11 BY V12

V11(Electricite)		V12(eau)							
Frequency	Percent	1	2	3	4	5	6	absent	Total
Row Pct	Col Pct								
0	0	214	221	329	129	29	0	922	
	0.00	18.92	19.54	29.09	11.41	2.56	0.00	81.52	
	0.00	23.21	23.97	35.68	13.99	3.15	0.00		
	0.00	57.53	91.32	95.64	98.47	93.55	0.00		
1	10	158	21	15	2	2	0	208	
	0.88	13.97	1.86	1.33	0.18	0.18	0.00	18.39	
	4.81	75.96	10.10	7.21	0.96	0.96	0.00		
	100.00	42.47	8.68	4.36	1.53	6.45	0.00		
absent	0	0	0	0	0	0	1	1	
	0.00	0.00	0.00	0.00	0.00	0.00	0.09	0.09	
	0.00	0.00	0.00	0.00	0.00	0.00	100.00		
	0.00	0.00	0.00	0.00	0.00	0.00	100.00		
Total	10	372	242	344	131	31	1	1131	
	0.88	32.89	21.40	30.42	11.58	2.74	0.09	100.00	

TRI CROISE AVEC TEST DU KHI2

TABLE OF V11 BY V12

V11(Electricite)		V12(eau)							
Frequency	Percent	1	2	3	4	5	6	absent	Total
Row Pct	Col Pct								
0	0	214	221	329	129	29	0	922	
	0.00	18.92	19.54	29.09	11.41	2.56	0.00	81.52	
	0.00	23.21	23.97	35.68	13.99	3.15	0.00		
	0.00	57.53	91.32	95.64	98.47	93.55	0.00		
1	10	158	21	15	2	2	0	208	
	0.88	13.97	1.86	1.33	0.18	0.18	0.00	18.39	
	4.81	75.96	10.10	7.21	0.96	0.96	0.00		
	100.00	42.47	8.68	4.36	1.53	6.45	0.00		
absent	0	0	0	0	0	0	1	1	
	0.00	0.00	0.00	0.00	0.00	0.00	0.09	0.09	
	0.00	0.00	0.00	0.00	0.00	0.00	100.00		
	0.00	0.00	0.00	0.00	0.00	0.00	100.00		
Total	10	372	242	344	131	31	1	1131	
	0.88	32.89	21.40	30.42	11.58	2.74	0.09	100.00	

STATISTICS FOR TABLE OF V11 BY V12

Statistic	DF	Value	Prob
Chi-Square	12	1407.275	0.000
Likelihood Ratio Chi-Square	12	286.364	0.000
Mantel-Haenszel Chi-Square	1	180.252	0.000
Phi Coefficient		1.115	
Contingency Coefficient		0.745	
Cramer's V		0.789	

Sample Size = 1131

WARNING: 48% of the cells have expected counts less than 5. Chi-Square may not be a valid test.

...

II.A

```

...
proc freq data=base.popul;
  title 'TRI CROISE AVEC TEST DU KHI2';
  tables v11*v12 / chisq; run;

proc freq data=base.popul;
  title 'TRI CROISE AVEC RESULTATS EN LISTE';
  tables v11*v12 / list; run;

proc freq data=base.popul;
  title 'TRI CROISE AVEC IMPRESSION DES POURCENTAGES PAR COLONNE SANS
  LES EFFECTIFS';
  tables v11*v12 / norow nofreq; run;

proc freq data=base.popul;
  title 'TRI CROISE AVEC IMPRESSION DES POURCENTAGES PAR LIGNES SANS
  LES EFFECTIFS';
  tables v11*v12 / nofreq nocol; run;

proc freq data=base.popul;
  title 'TRI CROISE AVEC IMPRESSION DES EFFECTIFS';
  tables v11*v12 / nocol norow; run;

proc freq data=base.popul;
  title 'TRI CROISE PONDERE';
  tables v11*v12; weight v7; run;

```

*tableau de fréquences croisées v11*v12 (eau*électricité) pondérées par les valeurs de la variable v7 (nombre de personnes par parcelle); résultats en nombre de personnes concernées*

- Dans cet exemple v7, nombre de personnes dans la parcelle, est introduit pour pondérer le poids de chaque observation (parcelle) dans le calcul de fréquence.

... TRI CROISE AVEC RESULTAT EN LISTE

	V11	V12	Frequency	Percent	Cumulative Frequency	Cumulative Percent
	0	2	214	18.9	214	18.9
	0	3	221	19.5	435	38.5
	0	4	329	29.1	764	67.6
	0	5	129	11.4	893	79.0
	0	6	29	2.6	922	81.5
	1	1	10	0.9	932	82.4
	1	2	158	14.0	1090	96.4
	1	3	21	1.9	1111	98.2
	1	4	15	1.3	1126	99.6
	1	5	2	0.2	1128	99.7
	1	6	2	0.2	1130	99.9
	absent	absent	1	0.1	1131	100.0

...

• • • TRI CROISE AVEC IMPRESSION DES POURCENTAGES PAR COLONNE SANS LES EFFECTIFS

TABLE OF V11 BY V12

V11(Electricite)		V12(eau)						Total
Percent Col Pct		1	2	3	4	5	6	
0	0.00	18.92	19.54	29.09	11.41	2.56	0.00	81.52
	0.00	57.53	91.32	95.64	98.47	93.55	0.00	
1	0.88	13.97	1.86	1.33	0.18	0.18	0.00	18.39
	100.00	42.47	8.68	4.36	1.53	6.45	0.00	
absent	0.00	0.00	0.00	0.00	0.00	0.00	0.09	0.09
	0.00	0.00	0.00	0.00	0.00	0.00	100.00	
Total	10	372	242	344	131	31	1	1131
	0.88	32.89	21.40	30.42	11.58	2.74	0.09	

TRI CROISE AVEC IMPRESSION DES POURCENTAGES PAR LIGNE SANS LES EFFECTIFS

TABLE OF V11 BY V12

V11(Electricite)		V12(eau)						Total
Percent Row Pct		1	2	3	4	5	6	
0	0.00	18.92	19.54	29.09	11.41	2.56	0.00	81.52
	0.00	23.21	23.97	35.68	13.99	3.15	0.00	
1	0.88	13.97	1.86	1.33	0.18	0.18	0.00	18.39
	4.81	75.96	10.10	7.21	0.96	0.96	0.00	
absent	0.00	0.00	0.00	0.00	0.00	0.00	0.09	0.09
	0.00	0.00	0.00	0.00	0.00	0.00	100.00	
Total	10	372	242	344	131	31	1	1131
	0.88	32.89	21.40	30.42	11.58	2.74	0.09	

TRI CROISE AVEC IMPRESSION DES EFFECTIFS

TABLE OF V11 BY V12

V11(Electricite)		V12(eau)						Total
Frequency Percent		1	2	3	4	5	6	
0	0	214	221	329	129	29	0	922
	0.00	18.92	19.54	29.09	11.41	2.56	0.00	
1	10	158	21	15	2	2	0	208
	0.88	13.97	1.86	1.33	0.18	0.18	0.00	
absent	0	0	0	0	0	0	1	1
	0.00	0.00	0.00	0.00	0.00	0.00	0.09	
Total	10	372	242	344	131	31	1	1131
	0.88	32.89	21.40	30.42	11.58	2.74	0.09	

TRI CROISE PONDERE

TABLE OF V11 BY V12

V11(Electricite)		V12(eau)						Total
Frequency Percent Row Pct Col Pct		1	2	3	4	5	6	
0	0	1213	1253	1997	810	211	5484	
	0.00	17.79	18.38	29.29	11.88	3.09	80.43	
	0.00	22.12	22.85	36.42	14.77	3.85		
	0.00	54.37	90.47	96.01	98.66	87.55		
1	60	1018	132	83	11	30	1334	
	0.88	14.93	1.94	1.22	0.16	0.44	19.57	
	4.50	76.31	9.90	6.22	0.82	2.25		
	100.00	45.63	9.53	3.99	1.34	12.45		
Total	60	2231	1385	2080	821	241	6818	
	0.88	32.72	20.31	30.51	12.04	3.53	100.00	

II.B DESCRIPTION STATISTIQUE DE VARIABLES QUANTITATIVES

II.B.1 Indices simples : proc means

Forme générale :

```
proc means data=tableau de départ
  maxdec=2          mean std min max;
  précision de deux décimales indices statistiques demandés
  var  var1 var2 var3;
  liste des variables à traiter
  output out= resultat;
  tableau SAS contenant les résultats, c'est-à-dire la moyenne, l'écart-type, le minimum, et le maximum des
  variables de la liste qui suit var
run;
```

Principaux indices que l'on peut demander dans une proc means (cf. SAS Procedures Guide 6.03, p. 235) :

n	:	nombre d'observations (sans compter les valeurs manquantes),
nmiss	:	nombre de valeurs manquantes,
mean	:	moyenne,
std	:	écart-type (standard deviation),
min	:	minimum,
max	:	maximum,
range	:	étendue,
sum	:	somme (à utiliser avec by),
var	:	variance,
uss	:	somme des carrés non corrigée,
css	:	somme des carrés corrigée,
cv	:	coefficient de variation,
stderr	:	erreur standard de la moyenne.

Pour le détail des calcul de ces différents indices, cf. SAS Procedures Guide 6.03, p. 234

Principales options possibles de proc means (cf. SAS Procedures Guide 6.03, p. 234) :

noprint	:	sans impression,
fw	:	longueur du champ pour impression; par défaut, fw=12,
missing	:	prendre en compte les valeurs manquantes,
descending	:	tri descendant des données; par défaut, ordre ascendant.

• • •

• • •

Principales instructions possibles à l'intérieur d'une `proc means` (cf. SAS Procedures Guide 6.03, p. 235) :

- var variables** : limite le traitement aux variables de la liste. En l'absence de **var**, traitement par défaut de toutes les variables numériques du tableau d'entrée.
- by variables** : un résultat par modalité de la ou des variables qui suivent **by**; le tableau doit être préalablement trié.
- class variables** : même fonction que **by**, mais sur un tableau non trié.
- freq variable** : pour chaque observation, la valeur de la variable est considérée comme un effectif; les valeurs doivent donc toutes être des valeurs entières.
- id variables** : permet de conserver certaines variables supplémentaires dans le tableau de sortie en plus des indices calculés.
- weight variable** : les valeurs de cette variable qui doit être numérique, permettent de pondérer les valeurs des observations en fonction de l'écart-type.
- output out=nom** : nom du tableau de résultats. Optionnellement peuvent être définis le type d'opération que l'on veut réaliser, les variables sur lesquelles on veut réaliser ces opérations, et les noms des nouvelles variables conservant les résultats.
Exemple : `proc means data=tableau de départ;`
`output out=nom de tableau des résultats`
`sum=liste de variables`
`std=liste de variables`
`mean=liste de variables;`

Remarque : `sum=liste de variables` utilisé précédé d'un **by** permet de sommer les valeurs des observations correspondant à une même modalité de la variable indiquée dans le **by** (cf. « Pratique de l'Analyse statistique SAS », M. Cosinschi et Ph. Waniez - Collection RECLUS Modes d'emploi n° 15, p. 58).

Application :

```
libname base '/usr/usr3/graph90/docu';
options ls=80 ps=60 nodate nonumber nocenter;
title1 'MEANS : PRINCIPAUX PARAMETRES DE LA DISTRIBUTION';
title2 'Statistiques sur les variables AGE MERE et AGE ENFANT';

proc means data=base.popul; var v17 v18;
run;
```

MEANS : PRINCIPAUX PARAMETRES DE LA DISTRIBUTION
Statistiques sur les variables AGE MERE et AGE ENFANT

N Obs	Variable	Label	N	Minimum	Maximum	Mean
1131	V17	age mere	1131	0	47.0000000	25.6790451
	V18	age enfant	1131	0	24.0000000	14.1494253

N Obs	Variable	Label	Std Dev
1131	V17	age mere	6.2806412
	V18	age enfant	6.9406936

II.B

II.B.2 Étude statistique complète avec diagramme et graphique : proc univariate

```
libname base '/usr/usr3/graph90/docu';
options ls=80 ps=60 nodate nonumber nocenter;
title1 'ETUDE STATISTIQUE COMPLETE';
title2 'Univariate : statistiques sur la variable AGE';
proc univariate data=base.enquete
  plot
    option de diagramme en bâton et de diagramme en boîte (Box Plot proposé par le statisticien Tuckey)
  normal;
    option du graphique permettant de visualiser la distribution normale et de juger de la distorsion
  var age; output out=resultats;
    liste des variables à traiter tableau des résultats de l'étape
proc print data=resultats; run;
```

Graphique complémentaire de la distribution : histogramme horizontal de la fréquence en pourcentage

```
title2 'Graphique complémentaire';
proc chart data=base.enquete;
  hbar age / type=percent
  midpoints=15 to 95 by 10; run;
définition des valeurs centrales et de l'étendue des classes. Les valeurs 15 et 95 correspondent aux valeurs extrêmes de la variable traitée.
```

- Autres options possibles de l'étape **proc univariate** :

noprint : pas d'impression des résultats,
freq : effectifs et fréquences des variables,
pctldef=value : choix de la méthode de calcul des percentiles.

- Autres instructions possibles de l'étape **proc univariate** :

var variables : traitement des variables numériques de la liste,
freq variable : variable dont l'effectif intervient dans l'étude,
weight variable : variable de pondération à partir de l'écart-type
output out=nom options : nom du tableau de résultats. Les traitements statistiques et les noms des variables qui, dans ce tableau, contiendront les résultats sont définis par des options.

Options possibles de l'instruction **output out** :

n : nombre d'observations,
sum : somme,
mean : moyenne,
var : variance...

- Explication du diagramme en boîte **Box Plot** :

lignes limitées par des +, + ——— + : premier et troisième quartiles
lignes limitées par des *, * ——— * : médiane
le signe + sur colonne centrale : moyenne
le signe 0 sur colonne centrale : valeurs < aux premiers quartiles
ou valeurs < 3 écarts-types
ou valeurs > au troisième quartile
ou valeurs > 3 écarts-types.

- Pour plus d'informations, cf. SAS Procedures Guide 6.03, pp. 405-411.

ETUDE STATISTIQUE COMPLETE
Univariate : statistiques sur la variable AGE

UNIVARIATE PROCEDURE

Variable=AGE age

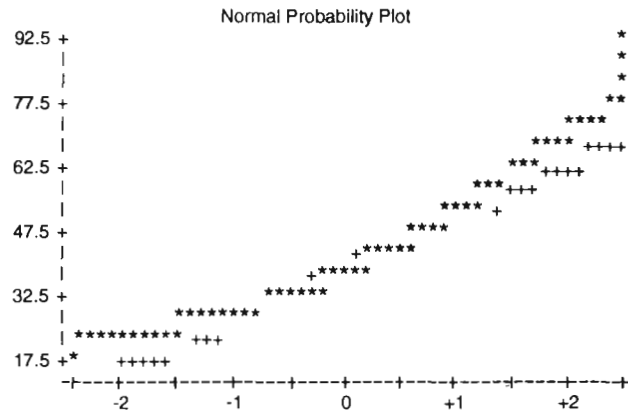
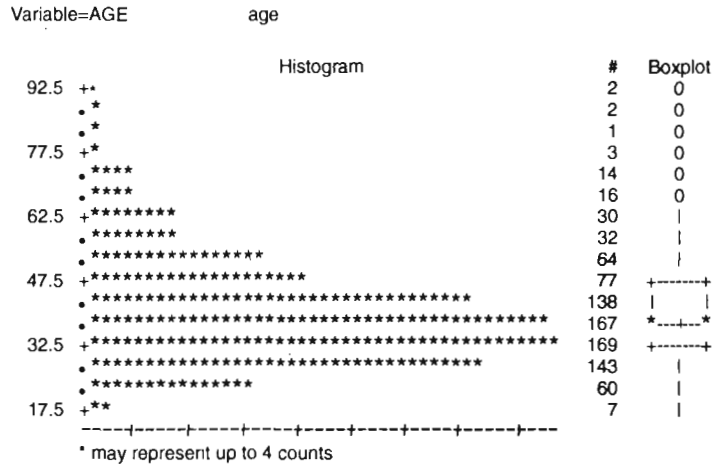
Moments			
N	925	Sum Wgts	925
Mean	38.92865	Sum	36009
Std Dev	11.95162	Variance	142.8412
Skewness	0.987733	Kurtosis	1.103958
USS	1533767	CSS	131985.3
CV	30.70135	Std Mean	0.392967
T:Mean=0	99.06344	Prob> T	0.0
Sgn Rank	214137.5	Prob> S	0.0001
Num ^= 0	925		
W:Normal	0.93088	Prob<W	0.0

Quantiles(Def=5)			
100% Max	90	99%	73
75% Q3	45	95%	62
50% Med	37	90%	56
25% Q1	30	10%	26
0% Min	16	5%	23
		1%	20
Range	74		
Q3-Q1	15		
Mode	36		

Extremes			
Lowest	Obs	Highest	Obs
16(915)	80(125)
18(640)	85(481)
18(314)	85(676)
18(142)	90(403)
19(659)	90(599)

ETUDE STATISTIQUE COMPLETE
Univariate : statistiques sur la variable AGE

UNIVARIATE PROCEDURE



ETUDE STATISTIQUE COMPLETE
Graphique complementaire

PERCENTAGE OF AGE

AGE MIDPOINT	age	CUM FREQ	FREQ	PERCENT	CUM PERCENT
15	*	11	11	1.19	1.19
25	*****	236	247	25.51	26.70
35	*****	336	583	36.32	63.03
45	*****	196	779	21.19	84.22
55	*****	95	874	10.27	94.49
65	****	34	908	3.68	98.16
75	*	13	921	1.41	99.57
85		4	925	0.43	100.00
95		0	925	0.00	100.00

PERCENTAGE

II.C CLASSIFICATION À PARTIR DE CALCULS STATISTIQUES

II.C.1 Centrage/réduction (écart-type)

La procédure **standard** permet un recodage direct des valeurs des observations en fonction d'une nouvelle échelle de mesure ayant pour origine la moyenne et pour unité, l'écart-type.

```

libname base '/usr/usr3/graph90/docu';
options ls=80 ps=30 nocenter nodate nonumber;
title1 'STANDARD : CENTRAGE ET REDUCTION DE LA DISTRIBUTION';

proc sort data=base.popul out=prov; by v17;
tri préalable obligatoire sur la variable à traiter
proc standard data=prov out=prov mean=0 std=1;
var v17; run;

recodage de la variable v17 : centrage et réduction en fonction de l'écart-type

title1 'Extraits des tableaux de données';
proc print data=base.popul (obs=5); var ident v17;
proc print data=prov (obs=5); var ident v17; run;

title1 'CINQ CLASSES CENTREES SUR LA MOYENNE';
data resultat; set prov;
length v17r $ 20;
if v17<-1.5 then v17r='<-1.5std';
else if v17>=-1.5 and v17<-0.5 then v17r='>-1.5std & <-0.5std';
else if v17>=-0.5 and v17<+0.5 then v17r='>-0.5std & <0.5std';
else if v17>=0.5 and v17<+1.5 then v17r='>0.5std & <1.5std';
else if v17>=+1.5 then v17r='>1.5std';
keep ident v17 v17r;

création d'un tableau comprenant l'identifiant, la variable recodée v17 et la nouvelle variable classée v17r. Les cinq classes de la variable v17r sont centrées autour de la moyenne.
proc print data=resultat (obs=5); run;

title1 'HUIT CLASSES AUTOUR DE LA MOYENNE';
data resultat; set prov;
if v17<-3 then v17r='<-3std';
else if v17>=-3 and v17<-2 then v17r='<-2std';
else if v17>=-2 and v17<-1 then v17r='>=-2std & <-1std';
else if v17>=-1 and v17<0 then v17r='>=-1std & <0std';
else if v17>=0 and v17<+1 then v17r='>=0std & <1std';
else if v17>=+1 and v17<+2 then v17r='>=+1std & <2std';
else if v17>=+2 and v17<+3 then v17r='>=+2std & >2std';
else if v17>=+3 then v17r='>=3std';
keep ident v17 v17r;

création d'un tableau comprenant l'identifiant, la variable recodée v17 et la nouvelle variable classée v17r. Les huit classes de la variable v17r se situent de part et d'autre de la moyenne.
proc print data=resultat (obs=5); run;

```


STANDARD : CENTRAGE ET REDUCTION DE LA DISTRIBUTION

NAME	MEAN	STD	N	LABEL
V17	25.679045093	6.2806412339	1131	age mere

Extraits des tableaux de donnees

OBS	IDENT	V17
1	absent	0
2	106	23
3	121	21
4	123	29
5	124	20

Extraits des tableaux de donnees

OBS	IDENT	V17
1	absent	-4.08860
2	3057	-1.70031
3	342	-1.54109
4	381	-1.54109
5	394	-1.54109

CINQ CLASSES CENTREES SUR LA MOYENNE

OBS	V17	IDENT	V17R
1	-4.08860	absent	<-1.5std
2	-1.70031	3057	<-1.5std
3	-1.54109	342	<-1.5std
4	-1.54109	381	<-1.5std
5	-1.54109	394	<-1.5std

HUIT CLASSES CENTREES AUTOUR DE LA MOYENNE

OBS	V17	IDENT	V17R
1	-4.08860	absent	<-3std
2	-1.70031	3057	>-2std & <-1std
3	-1.54109	342	>-2std & <-1std
4	-1.54109	381	>-2std & <-1std
5	-1.54109	394	>-2std & <-1std

II.C

II.C.2 Classes d'égal effectif (quantiles, médiane)

```

libname base '/usr/usr3/graph90/docu';
options ls=80 ps=60 nodate nonumber nocenter;
title 'CLASSES D'EGAL EFFECTIF';

data prov; set base.popul; keep ident v17;
sélection des variables
proc sort data=prov; by v17;
tri préalable obligatoire sur la variable à traiter

proc print data=prov (firstobs=265 obs=275);

proc rank data=prov out=prov groups=4;
recodage de la variable v17 avec les valeurs 0, 1, 2 ou 3, valeurs définies par l'option groups

title1 'QUARTILES'
data resultat; set prov;
length v17r $ 2;
if v17=0 then v17r='1Q';
else if v17=1 then v17r='2Q';
else if v17=2 then v17r='3Q';
else if v17=3 then v17r='4Q';
keep ident v17 v17r;
création d'une variable classifiée v17r à partir de la variable recodée v17

proc print data=resultat (firstobs=265 obs=275); run;

```

- De façon générale, **rank** classe les valeurs de la plus petite à la plus grande et affecte 1 à la valeur la plus faible, 2 à la suivante, etc. Dans cet exemple, la procédure **rank** recode automatiquement la valeur de chaque observation en fonction du quantile auquel elle appartient. Cependant, elle a pour défaut de « décaler » les codes : le code du premier quantile est 0 et non pas 1.
- Instructions possibles à l'intérieur de **proc rank** :
 - ranks variables** : les résultats de la classification effectuée par la procédure **rank** sont affectés aux variables de la liste de l'option **ranks** et ne se substituent donc plus aux valeurs initiales des variables traitées.
 - by variables** : permet d'obtenir une analyse séparée pour chaque modalité des variables qui suivent le **by**.

CLASSES D'EGAL EFFECTIF

OBS	V17	IDENT
265	20	1079
266	20	1080
267	20	1095
268	20	1103
269	20	1106
270	21	121
271	21	158
272	21	217
273	21	293
274	21	295
275	21	303

QUARTILES

OBS	V17	IDENT	V17R
265	0	1079	1Q
266	0	1080	1Q
267	0	1095	1Q
268	0	1103	1Q
269	0	1106	1Q
270	1	121	2Q
271	1	158	2Q
272	1	217	2Q
273	1	293	2Q
274	1	295	2Q
275	1	303	2Q

II.C

II.C.3 Classes d'égale étendue (max, min, mode)

```

libname base '/usr/usr3/graph90/docu';
options ls=80 ps=30 nonumber nocenter nodate;

title1 'TABLEAU DES INDICES STATISTIQUES';
proc sort data=base.popul out=prov; by v17;
tri préalable obligatoire sur la variable à traiter

proc means data=prov noprint; var v17;
pas d'impression des résultats de la proc means
    output out=indices max=vmax min=vmin;
tableau des résultats les variables vmin et vmax correspondent respectivement
aux valeurs maximale et minimale de la variable v17
calcul des indices statistiques min et max de la variable v17.
proc print data=indices;
data indices; set indices;
    call symput ('bornsup',vmax);
    call symput ('borninf',vmin); run;
création de deux macrovariables bornsup et borninf auxquelles sont respectivement affectées les valeurs de vmax et
vmin; bornsup est la valeur maximum de la variable v17, borninf sa valeur minimale.

title1 'CLASSES D'EGALE ETENDUE';
data cod; set prov;
    interval=(&bornsup-&borninf)/4;
la valeur de la variable interval est égale au quart de l'amplitude de la variable v17
    born1=&borninf+interval;
    born2=&borninf+(2*interval);
    born3=&borninf+(3*interval);
calcul des valeurs des bornes de quatre classes
    if v17<born1 then v17r='1C';
    else if v17>=born1 and v17<born2 then v17r='2C';
    else if v17>=born2 and v17<born3 then v17r='3C';
    else if v17>=born3 then v17r='4C';
    keep ident v17r v17 born1--born3;
création d'une variable classifiée

proc print data=cod (obs=10); run;

```

- Dans le tableau de résultats statistiques créé par l'étape **proc means**, apparaissent deux variables qui sont générées directement par SAS **_TYPE_** et **_FREQ_** :
 - _TYPE_** comptabilise le nombre de partitions de la population d'observations. Chaque classe est affectée d'un code. S'il n'y a pas de classification préliminaire, alors **_TYPE_** contient une seule observation qui correspond à la classe théorique totale.
 - _FREQ_** comptabilise les effectifs d'observations par classe ou unité de partition. Lorsqu'aucune classification n'est appliquée, **_FREQ_** contient une seule observation dont la valeur correspond à l'effectif total des observations.
- Pour plus de détails sur ces variables générées par SAS, cf. SAS Procedures Guide 6.03, pp. 321-324

• • •

• • •

- **call symput**, instruction permettant de créer, dans une étape **data**, une macrovariable ou d'affecter une nouvelle valeur à une macrovariable existante. La syntaxe générale est de la forme :

call symput (macrovar,valeur)

où **macrovar** et **valeur** sont des noms de variables d'une **étape data**, une chaîne de caractères (entre quotes) ou une expression.

Dans l'exemple

```
call symput ('bornsup',vmax)
call symput ('borninf',vmin)
```

call symput affecte aux macrovariables bornsup et borninf respectivement les valeurs des variables vmax, vmin, résultats de l'étape précédente. Ces macrovariables bornsup et borninf sont les variables paramétrables de l'instruction **call symput**. Le langage **macro SAS** permet d'optimiser les programmes SAS. Ce langage dispose de macrovariables, de macrofonctions et de macroinstructions. Programmer avec le langage macro permet de paramétrer une ou plusieurs étapes du traitement en fonction de résultats d'étapes antérieures.

- L'étape **proc means** calcule les indices synthétiques (**min, max, mean, std...**) et les conserve dans le tableau indices. La première **étape data** sert à affecter à des variables (vmin et vmax) les valeurs de ces indices synthétiques de façon à pouvoir les réutiliser dans une **étape data** ultérieure. Ainsi dans l'exemple traité, la seconde **étape data** utilise ces indices synthétiques pour calculer l'étendue qui sert à la définition des classes.

TABLEAUX DES INDICES STATISTIQUES

Analysis Variable : V17 age mere

N Obs	N	Minimum	Maximum	Mean	Std Dev
1131	1131	0	47.0000000	25.6790451	6.2806412

TABLEAUX DES INDICES STATISTIQUES

OBS	_TYPE_	_FREQ_	VMAX	VMIN
1	0	1131	47	0

CLASSES D'EGALE ETENDUE

OBS	V17	IDENT	BORN1	BORN2	BORN3	V17R
1	0	absent	11.75	23.5	35.25	1C
2	15	3057	11.75	23.5	35.25	2C
3	16	342	11.75	23.5	35.25	2C
4	16	381	11.75	23.5	35.25	2C
5	16	394	11.75	23.5	35.25	2C
6	16	563	11.75	23.5	35.25	2C
7	16	590	11.75	23.5	35.25	2C
8	16	454	11.75	23.5	35.25	2C
9	16	466	11.75	23.5	35.25	2C
10	16	783	11.75	23.5	35.25	2C

III
CHANGEMENT D'UNIVERS D'ANALYSE :
AGRÉGATION

III.A CONSTRUCTION D'UN TABLEAU DISJONCTIF COMPLET

Au cours de l'exploitation des données d'une enquête, il est fréquent que l'on ait besoin de changer de population d'étude. Par exemple après un travail statistique sur les individus d'une population donnée, l'utilisateur peut vouloir considérer l'ensemble des variables en fonction des groupes d'âges. Dans ce cas, ceux-ci doivent devenir les nouveaux « individus » (observations) du tableau. Ce procédé est particulièrement fréquent en géographie humaine où on est généralement amené à d'abord étudier une population dans son ensemble, puis à étudier celle-ci en fonction d'une partition ou d'un découpage géographique (carte des régions, des départements...). Une telle démarche peut être réalisée à l'aide de SAS qui offre le choix entre différents types d'agrégation : l'agrégation simple qui permet de disposer des effectifs d'individus au niveau de la nouvelle unité; l'agrégation avec calcul qui permet d'obtenir des indices synthétiques au niveau de la nouvelle unité (fréquences, moyenne par unité...). Cette dernière possibilité est particulièrement intéressante lorsque les nouvelles unités sont de nature géographique puisqu'on débouche ainsi sur les notions de moyenne locale, écart-type local, fréquence locale, etc.

Le processus d'agrégation est simple lorsqu'il s'agit de variables quantitatives, codées en effectifs; il est plus complexe lorsque les variables originelles sont qualitatives, codées en catégories ou modalités. Dans ce cas, il est alors nécessaire « d'éclater » chaque variable qualitative en autant de nouvelles variables qu'elle a de modalités (construction d'un tableau disjonctif complet), avant de pouvoir calculer les effectifs correspondants à chaque nouvelle unité (unité d'agrégation). Le schéma illustratif suivant synthétise les différentes étapes de traitement réalisé par les programmes III.A et III.B.

Pour plus de détails sur la construction de ces tableaux, consulter "Introduction à l'analyse des données", C. Mullon, document interne Orstom - LIA, 1990.

Schéma illustratif

Tableau de départ : individus/variables

ident	Var1	Var2	Var3	N° secteur
i1	code1	code2	code1	341
i2	code2	code1	code1	341
i3	code1	code2	code1	341
i4	code2	code1	code2	342
i5	code1	code3	code3	342
etc.				

Tableau intermédiaire : individus/modalités de variables (disjonctif complet)

ident	code1Var1	code2Var1	code1Var2	code2Var2	...	N° Secteur
i1	1	0	0	1		341
i2	0	1	1	0		341
i3	1	0	0	1		341
i4	0	1	1	0		342
i5	1	0	0	0		342
etc.						

Tableau créé : unité spatiale d'agrégation / modalités des variables du niveau individus

N° Secteur	code1Var1	code2Var1	code1Var2	code2Var2	...
341	2	1	1	2	
342	1	1	1	0	
etc.					


```

libname base '/usr/usr3/graph90/docu';
options ls=80 ps=60 nodate nonumber nocenter;
title1 'TABLEAU DISJONCTIF COMPLET';
title2 'TABLEAU INITIAL';
proc print data=base.enquete (obs=10);
var ident sect pers age sexe; run;

proc format;
value fpers low-1 = '1'
            2-5 = '2'
            6-9 = '3'
            10-high = '4';
value fage 15-25 = '1'
          26-35 = '2'
          36-45 = '3'
          46-55 = '4'
          56-high = '5';

data prov; set base.enquete;
length persc agec 4;
longueur des nouvelles variables persc et agec
persc=put(pers,fpers.);
agec=put(age,fage.);
affecte comme valeurs à chaque observation des variables persc et agec, le résultat du recodage en classe, selon les formats fpers et fage, des variables pers, age
if sexe=-9 or act=6 then delete;
supprime les enregistrements (individus) correspondant à l'absence de réponse (codes -9 pour sexe et 6 pour act)

title; title1 'TABLEAU APRES CLASSIFICATION';
proc print data=prov (obs=10);
var ident sect persc agec sexe; run;

```

construction du tableau disjonctif complet

```

data base.tdc; set prov;
array a(persc) prs1-prs4;
array b(sexe) sex1-sex2;
array c(agec) age1-age5;
syntaxe : array nomarray (variable) liste des modalités de la variable
prs1-prs4 : permet de donner un nom de variable aux différentes modalités de persc; prs1 correspond à la
modalité 1 de persc, prs2 à la modalité 2, prs3 à la modalité 3 et prs4 à la 4e modalité.
array t a b c;
t : tableau construit à partir des tableaux a, b, c
array z prs1--age5;
z : nom du tableau
prs1--age5 : liste des variables constituant le tableau z

```

• • •

III.A

• • •

```

do over z;    z=0;  end;
do over t;    t=1;  end;
création du tableau disjonctif complet
keep  ident sect prs1--age5;
sélection de variables
length  _numeric_ 3;
retient une longueur de trois positions pour toutes les variables numériques

title1      'TABLEAU DISJONCTIF COMPLET';
proc print   data=base.tdc (obs=10);
proc contents data=base.tdc;
run;

```

- Principe de construction d'un tableau disjonctif complet : à partir d'une variable codée de 1 à n , on crée n nouvelles variables prenant les valeurs 0 ou 1 (variables logiques).

Ceci se fait avec SAS dans une **étape data** et de la façon suivante :

- les variables logiques issues d'une même variable d'origine sont générées dans un tableau (**array**) indicé par cette variable origine.

exemple : **array** b(sexe) sex1-sex2;

- la boucle **do over z** initialise toutes les variables du tableau z à 0. La boucle suivante, **do over t**, affectera pour chaque observation, la valeur 1 à la variable logique correspondant à la modalité existant pour cette observation dans la variable origine.
- un passage dans la boucle **do over t** correspond à l'éclatement d'une variable d'origine. Dans cette boucle, on assigne à t la valeur 1. Dans l'exemple, au premier passage sur la boucle, t « pointe » sur le premier élément du tableau b qui est l'**array** correspondant à la variable d'origine sexe; ce tableau b est lui-même indicé par la variable à recoder sexe. Si pour l'observation courante, sexe vaut 2, c'est alors $b(2)$, c'est-à-dire la nouvelle variable logique sex2, qui prend la valeur 1.

III.B AGRÉGATION DU TABLEAU DISJONCTIF COMPLET PAR UNITÉS GÉOGRAPHIQUES

```

libname base '/usr/usr3/graph90/docu';
options ls=80 ps=60 nodate nonumber nocenter;
title 'TABLEAU DISJONCTIF COMPLET';
proc print data=base.tdc (obs=10); run;

proc summary data=base.tdc nway;
  class sect;
  statistiques par sous-groupes de la variable sect
  var prs1--age5;
  liste des variables numériques à traiter
  output out=unitgeo
  nom du tableau de sortie des résultats (secteurs géographiques/variables)
  sum= mean(prs1)=mprs1 std(age4)=stdage4;
  liste des indices statistiques que l'on veut faire apparaître dans le tableau de sortie
  sum= calcule les effectifs par secteurs pour toutes les variables. Les variables du nouveau tableau porteront le même nom que les variables initiales.
  mean(prs1)=mprs1 : la nouvelle variable mprs1 a pour valeur la moyenne par secteur géographique de la variable prs1.
  std(age4)=stdage4 : la nouvelle variable stdage4 a pour valeur l'écart-type de la variable age4 par secteur (écart-type local).

title 'TABLEAU AGREGÉ PAR UNITÉS GÉOGRAPHIQUES';
proc print data=unitgeo; run;
data base.unitgeo; set unitgeo (drop=_type_ _freq_);
permet de ne pas retenir dans le tableau résultat les variables _type_ et _freq_ que SAS crée automatiquement dans l'étape summary
proc contents data=base.unitgeo; run;

```

- Comme dans le cadre d'une **proc means**, SAS génère au cours d'une **proc summary** deux variables **_TYPE_** et **_FREQ_** qui stockent respectivement pour chaque partition, un code rang et un effectif total des observations du tableau originel appartenant à la nouvelle unité ou « individu » du tableau créé.
- Le même résultat peut être obtenu par l'étape **proc means** suivante :

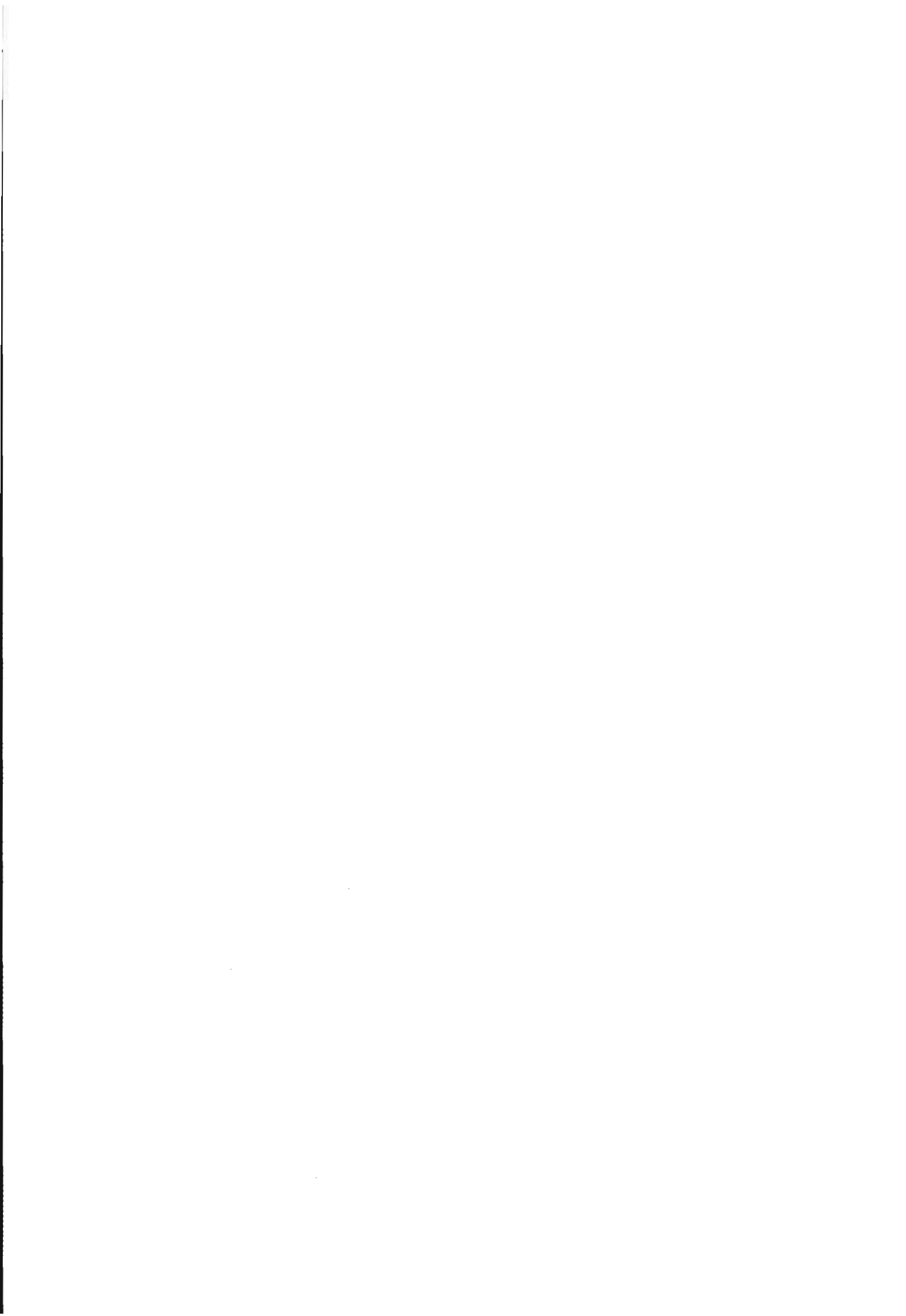
```

proc sort data=base.tdc out=prov; by sect;
proc means data=prov noprint; by sect; var prs1--age5;
  output out=prov1 sum=;
proc print data=prov1; run;

```

Réaliser l'agrégation à l'aide de la **proc means** de l'exemple ci-dessus plutôt qu'avec la **proc summary** du programme revient rigoureusement au même. Seule la valeur de la variable **_TYPE_** générée par SAS est différente selon la procédure utilisée.

IV
SAS/GRAPH



GÉNÉRALITÉS SUR SAS/GRAPH

QUE PEUT-ON FAIRE AVEC SAS/GRAPH ?

- **Connaître l'environnement de la session :**

- goptions** : étape de visualisation des paramètres définis pendant la session (table des couleurs et trames ou table des symboles, liste des légendes ou titres),
- gtestit** : étape de test des représentations ou de sortie des options graphiques de l'installation.

- **Créer son environnement de travail :**

- gdevice** : étape de visualisation et de modification des paramètres du périphérique graphique,
- goptions** : instruction permettant de personnaliser sa session SAS : choix du périphérique, couleur de fond de l'écran, redéfinition de la table des couleurs, de la couleur d'affichage des textes et de leur hauteur, de l'unité courante, et de la position du graphique dans la page.

- **Représenter les variables de la base :**

- gmap** : étape de création de carte en deux ou trois dimensions,
- gplot** : étape de représentation graphique en deux dimensions de deux variables (courbes),
- gchart** : étape de représentation graphique de variables (diagrammes),
- gcontour** : étape de représentation en deux dimensions de trois variables,
- g3d** : étape de représentation en trois dimensions de trois variables.

- **Créer une nouvelle information à partir de tableaux SAS :**

- gproject** : étape de changement de projection,
- greduce** : étape de préparation à la généralisation,
- gremove** : étape de création d'un nouveau fond de carte,
- g3grid** : étape d'interpolation et de lissage de données.

- **Créer une nouvelle information graphique :**

- tableau annotate** : création d'un tableau graphique dans une **étape data**,
- gfont** : création de caractères.

- **Afficher des tableaux :**

- ganno** et **gslide** : visualisation de **tableaux annotate**,
- gprint** : habillage de sortie d'étape non graphique,
- greply** : gestion d'une ou plusieurs sorties graphiques.

L'ENVIRONNEMENT GRAPHIQUE AVEC SAS/GRAPH

Paramétrage du périphérique graphique

La sortie graphique d'une étape SAS/GRAPH est indépendante de tout périphérique. Un pilote (driver) de périphérique transforme cette sortie dans le langage graphique du périphérique choisi, en tenant compte des caractéristiques de celui-ci. Les différents périphériques accessibles par SAS et leurs caractéristiques sont répertoriés dans une librairie interne à SAS.

Il y a deux façons de paramétrer un périphérique :

- **gdevice**, étape permettant de modifier des données dans le catalogue. La modification affectera alors toutes les sessions SAS.

- **goptions**, instruction qui permet optionnellement, dans un programme ou pour une session SAS, de choisir le périphérique graphique et de définir certaines caractéristiques de la sortie. Par exemple, supposons que, dans notre environnement nous voulions avoir une visualisation sur l'écran et une sortie sur l'imprimante laser. Une première instruction

goptions device=sun;

précéderait l'étape graphique et le résultat s'afficherait sur la station Sun. Une instruction

goptions device=aplplus nomfichier;

avant une seconde étape graphique orienterait la sortie correspondante vers un fichier destiné à l'imprimante laser. Si, au cours de cette même session nous trouvions, dans le même programme (après une instruction **run**) ou dans un autre programme, l'instruction **goptions** avec un autre nom de **device**, ce dernier périphérique deviendrait le périphérique de sortie.

Outre le choix du périphérique, **goptions** permet de redéfinir les tables internes initialisées à l'ouverture de la session SAS (couleurs, trames, type de ligne, etc.). Parmi les options les plus courantes de cette instruction, nous citerons **ctext**, couleur des chaînes de caractères, **ftext** qui définit la fonte de caractères par défaut, **htext** pour la hauteur par défaut des textes, **border** pour avoir un cadre autour de la sortie, **cback** pour définir la couleur de fond de la sortie, **colors** pour réinitialiser la palette de couleurs, **gunit** pour choisir l'unité courante, **rotate** pour positionner verticalement ou horizontalement la sortie et **device** vu ci-dessus.

Les instructions globales

Ce sont des instructions SAS, effectives dès leur apparition dans un programme et dont l'effet cesse à la réinitialisation suivante c'est-à-dire lors du prochain appel à cette instruction ou à la fin de la session. C'est le cas de **goptions** (cf. paragraphe ci-dessus). D'autres permettent de construire son propre environnement graphique.

Les instructions **title**, **note** et **footnote**, habillage de la sortie.

Elles permettent d'écrire, comme leur nom l'indique, des titres, notes ou notes de bas de page. Des options seront associées à ces instructions si l'on veut positionner des textes, choisir leur taille ou leur couleur, c'est-à-dire modifier les choix par défaut du système. La hauteur des textes peut être définie dans plusieurs unités : **in** pour inches, **cm** pour cm, **pct** pour pourcentage, l'unité par défaut étant le pixel **cells**. Pour la syntaxe de ces instructions, se reporter au début du fascicule SAS/GRAPH.

Les instructions **pattern**, **symbol**, **legend** et **axis**, présentation des résultats d'une étape graphique.

Ce sont des instructions qui modifieront les options par défaut du système lors de l'affichage des résultats de l'étape et qui servent à personnaliser la sortie.

- **pattern** : définition des trames ou pourcentage de couleur dans certaines étapes (**gchart**, **gcontour**, **gmap**, **gplot**).

syntaxe : **patternn options;**

- **n** : numéro d'ordre de la représentation qui sera associée à la nième classe existante d'une étape **gmap** ou **gchart**,
- **options** : permet de définir la couleur et la trame à affecter à la classe.
Parmi les options possibles :
c=nom pour la couleur; si cette option est absente, la couleur par défaut sera celle de l'environnement général,
v=nom où **nom** correspond à une trame, à l'absence de couleurs ou à 100% de la couleur (aplat); si cette option est absente, SAS affecte la représentation par défaut.

- **symbol** : choix du symbole pour une étape **gplot**.

syntaxe : **symboln options;**

Parmi les options possibles : couleur, hauteur, type de symbole, etc., mais aussi choix de la méthode d'interpolation entre deux points successifs.

- **legend** : mise en forme de la légende dans les étapes **gmap**, **gchart**, **gplot**.

syntaxe : **legendn options;**

Parmi les options possibles : horizontale ou verticale, avec ou sans cadre; mettre un titre, modifier l'intitulé des classes, choisir une fonte et une hauteur de caractères, etc.

- **axis** : tracé d'axes dans une étape **gchart**

syntaxe : **axis options**;

Parmi les options possibles : définir l'origine, la longueur, la couleur, les subdivisions, le texte à afficher, etc.

Quand et comment SAS utilise ces instructions?

Les paramètres de ces instructions, dès leur lecture, réinitialisent les tables internes correspondantes. Pour **pattern** et **symbol**, il y a affectation de la définition de la nième trame ou du nième symbole à la pième classe à représenter. En effet, si une classe est absente, la représentation de la classe suivante est la première disponible dans la table des **pattern**. Ce sont des options d'étape, **gmap** ou **gplot** par exemple, qui activent **legend** et **axis**.

exemple : **legend=legend1**

dans une étape **gmap** présentera la légende de la carte selon les critères définis dans **legend1**.

Ces instructions se placent n'importe où dans un programme et restent effectives jusqu'à ce qu'un nouvel appel à l'une d'elles modifie le contexte; c'est pourquoi on parle d'instructions globales. Si elles ne sont pas modifiées, elles constituent les options par défaut lors de la soumission d'un nouveau programme dans la même session SAS.

Les tableaux annotate

Une sortie graphique SAS est constituée du résultat d'une étape faisant appel au graphique et d'une ou plusieurs informations complémentaires. Chaque élément de la sortie de l'étape peut être défini par l'intermédiaire d'options. L'option **annotate** permet de superposer à la sortie graphique divers graphiques ou textes.

syntaxe : **annotate=nomtableau**

annotate est une option des étapes SAS/GRAPH suivantes : **ganno**, **gslide**, **gprint**, **gcontour**, **gplot**, **gchart**, **gmap**. Tous les noms de variables d'un **tableau annotate** sont prédéfinis par SAS. Ce tableau contient les informations élémentaires nécessaires à une représentation graphique : textes à écrire, tracé de lignes ou de symboles, légende de cartes,... et est communément appelé **tableau annotate**. Il est constitué d'un ensemble de points de coordonnées (variables SAS **x** et **y**) sur lequel s'effectue une action (variable SAS **function**). Un **tableau annotate** contient au minimum ces trois variables,

x et **y** : coordonnées du point où s'effectue l'action,
function : type d'action à effectuer.

Les actions possibles : déplacement dans la zone d'affichage avec ou sans tracé de ligne, positionnement d'un symbole ou d'un texte, etc.

Chaque action peut s'exécuter dans un environnement particulier grâce aux variables associées à chaque valeur possible de la variable **function** : **line** pour le type de la ligne, **size** pour l'épaisseur, **label** pour indiquer que l'action traite du texte, **position** pour indiquer où sera placé le texte par rapport au point, **color** pour la couleur, **style** pour le type de trame, etc. En fait, il faut considérer les options comme des éléments qui permettent de construire un ensemble correspondant à son besoin. Ces variables ont un caractère momentané dont l'effet cesse à la fin de l'étape.

Les variables **xsys** et **ysys** d'un **tableau annotate** déterminent le système de référence de l'espace où s'affiche le graphique. Le positionnement final d'un point de coordonnées (x,y) est directement lié aux valeurs de **xsys** et **ysys**.

L'affichage d'un point fait appel à trois notions :

- l'espace graphique de visualisation,
- les unités du repère graphique,
- la relation entre les points : une coordonnée est absolue si elle se rapporte à une origine dans un repère, relative si elle se rapporte à la coordonnée précédente.

L'espace de visualisation graphique peut être :

- l'écran graphique de visualisation,
- la fenêtre graphique : l'espace précédent sur lequel sera réservé l'espace nécessaire à l'affichage des titres et bas de pages,
- l'espace utilisateur limité par les valeurs extrêmes des données à traiter.

Trois unités possibles dans ces repères :

- pixels si écran ou fenêtre graphique,
- pourcentage par rapport aux axes si écran ou fenêtre graphique,
- celle des données de l'utilisateur si fenêtre graphique des données.

Pour plus de détails sur les systèmes de référence lors de l'affichage d'un point de coordonnées (x,y), cf. SAS/GRAPH User's Guide 6.03 pp. 120-123.

On notera que l'on peut avoir plusieurs systèmes de référence dans un même programme et qu'il y a indépendance de système sur les axes : le système de l'axe des x peut être différent de celui de l'axe des y, chaque coordonnée étant positionnée dans le repère correspondant à son axe.

Dans l'environnement par défaut d'une étape **gmap** ou **gplot** par exemple, SAS gère l'occupation de l'écran en fonction des éléments à visualiser. Il réserve l'espace nécessaire à l'affichage de titres, légendes ou notes de bas de page; l'espace restant est celui disponible pour la carte ou le graphique. Il calcule alors le facteur d'échelle à appliquer aux coordonnées des éléments à visualiser. Il ne sera utile de s'intéresser au système de repérage que si l'on utilise l'option **annotate** dans ces étapes.

La couleur avec SAS

Les périphériques noir et blanc et les périphériques couleur permettent de représenter les classes par des trames et aplats de couleur selon les définitions des options des instructions **pattern**. Par un artifice décrit ci-dessous, il est possible d'obtenir des dégradés de gris (pourcentages de noir) sur un périphérique noir et blanc.

La définition de la couleur pour un périphérique couleur tient compte des possibilités d'affichage de celui-ci : nombre de couleurs, palette de couleurs définissable ou non par l'utilisateur.

A une couleur correspond un code. Ce code fait référence au mode RGB (couleur obtenue à partir des trois composantes rouge, vert, bleu) ou au mode HLS (combinaison de la teinte, de la luminosité et de la saturation de la couleur), à une gamme de gris ou encore à une liste de noms de couleurs prédéterminés par SAS.

Le code d'une couleur en mode RGB est de la forme **CXrrggb** où **rrggb** sont respectivement les composantes de la couleur rouge, du vert et du bleu. Ces valeurs sont des valeurs hexadécimales comprises entre 0 et 255. Par exemple, cx0000ff correspond au bleu et cxff00ff correspond à un violet.

En mode HLS, le code de la couleur est de la forme **Hhhllss**. Les valeurs de **ll** et **ss** varient, en hexadécimal de 00 à ff soit de 0 à 255. En théorie, **hhh** peut varier de 000 à fff. En réalité, ce paramètre correspond à une valeur angulaire comprise entre 0 et 360°, c'est-à-dire entre 0 et 168 en hexadécimal.

En gamme de gris, nous aurons **grayll** où **ll** valeur hexadécimale de la luminosité varie de 00 à ff. Ce sont des couleurs choisies dans cette gamme qui sortiront en pourcentage de noir sur une imprimante laser.

Pour plus de détails, cf. SAS/GRAPH User's Guide 6.03 p. 30.

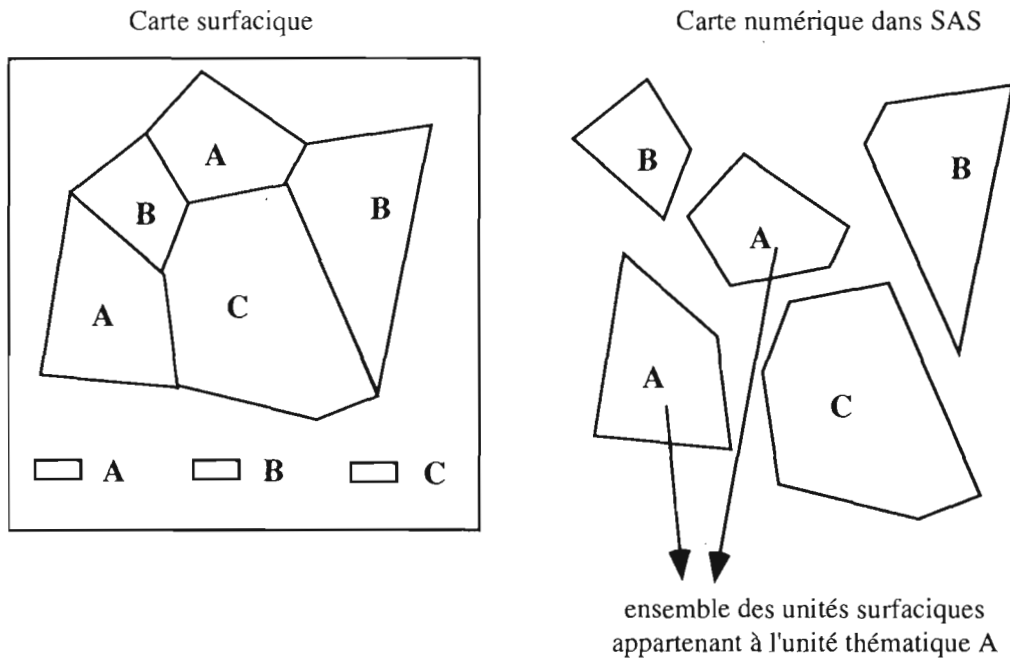
La variété des nuances de la palette de couleurs prédéfinies par SAS dépend des possibilités du terminal graphique. La liste des codes possibles est ordonnée selon les classes de couleur (cf. SAS/GRAPH User's Guide, pp. 33-38); et l'on trouve par exemple 'bl' qui correspond à du noir, 'rbk' à du 'reddish black' ou 'brbl' à du 'brownish black'...

Terminologie utilisée dans ce fascicule

- **tableau géographique** : on appellera tableau géographique un tableau contenant les coordonnées d'informations relatives à un fond de carte,
- **tableau graphique** : les coordonnées traitées se rapportent à des informations issues de tableaux géographiques ou à des représentations graphiques dans un repère quelconque.

Structure des données géographiques dans SAS

La carte numérique représente un ensemble d'unités thématiques (dans notre exemple A, B, C). Une ou plusieurs unités surfaciques, c'est-à-dire un ou plusieurs polygones (dans la terminologie SAS, un ou plusieurs segments) correspondent à une unité thématique.



L'information géographique dans SAS se présente donc de la façon suivante :

unité thématique 1, localisation 1^{er} polygone
 unité thématique 1, localisation 2^e polygone
 unité thématique 1, localisation 3^e polygone
 unité thématique 2, localisation 1^{er} polygone
 unité thématique 2, localisation 2^e polygone
 etc.

où localisation est l'ensemble des points de coordonnées x,y qui décrivent le contour du polygone. La variable **segment**, dont le nom est prédéfini par SAS, indique à quel polygone d'une unité thématique appartient un point de coordonnées (x,y) .

Le tableau de coordonnées géographiques devient alors (nous affectons à l'unité thématique le nom de variable « identifiant ») :

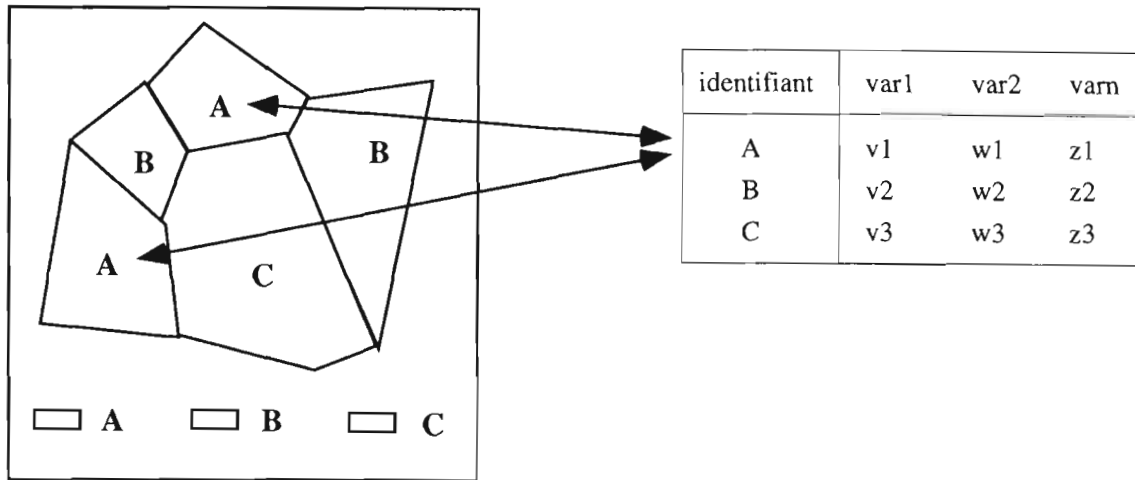
identifiant	x	y	segment
A	x1	y1	1
A	x2	y2	1
...
A	xm	ym	1
A	x1	y1	2
A
...
A	xn	yn	2
B	x1	y1	1
...
B	xp	yp	1
etc.			

x et **y** sont comme **segment**, des noms de variables prédéfinis par SAS.

Deux remarques sur cette structure :

- il n'y a pas de topologie : chaque polygone est indépendant,
- un point, lorsqu'il appartient à deux unités est présent deux fois dans le tableau. Donc plus la saisie de l'information graphique est fine, plus les tableaux seront volumineux.

Vouloir créer une base SAS et faire, en outre, de la cartographie suppose que l'on dispose de deux types d'information : des variables quantitatives ou qualitatives, déjà informatisées ou non, qui sont des résultats d'enquête ou d'analyse et d'une carte. La légende de la carte contiendra au moins l'information relative à une des variables descriptives



La variable qui va permettre à SAS d'associer l'information thématique et géographique est celle que nous appelons identifiant. Pour créer notre base, nous disposons donc

- d'informations géographiques :

identifiant	x	y	segment

- et d'informations descriptives:

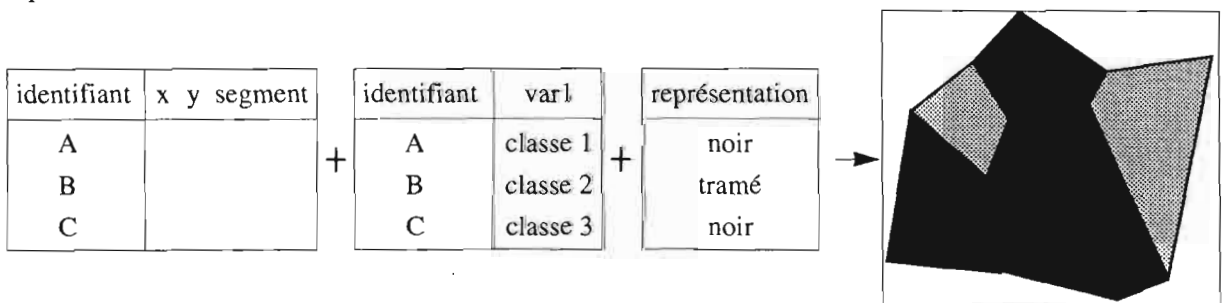
identifiant	var1	var2	varn

Nous avons vu dans les paragraphes précédents comment introduire l'information descriptive. Comment introduire maintenant l'information géographique?

Deux possibilités : avoir un seul tableau qui contiendrait les variables descriptives et géographiques soit

identifiant	var1	var2	...	varn	x	y	segment

ou deux tableaux qui seraient alors le reflet de la réalité, l'un représentant la carte, l'autre son contenu. Cette dernière solution est celle qui est le plus généralement retenue. La variable de l'option **id** des étapes de cartographie est la variable commune aux deux tableaux (celle que nous appelons identifiant) : elle permet à SAS de déterminer les polygones associés à une classe de la variable à cartographier et de leur affecter la représentation de la classe.



Comment obtenir ce résultat avec SAS?

gmap, étape de cartographie d'une variable, travaille optionnellement sur deux tableaux de données. L'option **data** définit le tableau contenant la variable à cartographier et l'option **map**, celui qui contient les variables de localisation (**x**, **y** et **segment**); les deux tableaux ont une variable commune (celle de l'option **id** de **gmap**).

Ce qui dans le programme SAS donne :

```

pattern1 v=s    c=black;
pattern2 v=r1   c=black;
... étape de classification de la variable vari à cartographier
proc gmap      data=tableau descriptif  map=tableau géographique;
  id identifiant;
  choro vari   /      discrete;

```

Nous verrons dans les exemples qui suivent que les tableaux des options **data** et **map** peuvent être les mêmes. Avoir deux tableaux n'est ni une contrainte ni une obligation : SAS, logiciel de traitement statistique et aussi gestionnaire de données localisées. Et en tant que tel, il peut sans difficulté associer les valeurs thématiques et les unités surfaciques sans qu'il soit nécessaire pour chaque point de répéter l'ensemble des informations des variables associées à l'unité thématique à laquelle il appartient. C'est-à-dire sans qu'il y ait redondance de données dans la base.

Nous rappelons que SAS ne possède pas de modules de cartographie des informations géographiques de type linéaire (représentation de flux) ou ponctuel (cercles proportionnels ou autres symboles ponctuels).

Les données géographiques des exemples de ce chapitre

Les programmes qui suivent et qui traitent de données relatives à Ouagadougou constituent la chaîne de traitement qui a abouti aux cartes incluses dans la thèse 'Structure de l'information à plusieurs niveaux et analyse de données', thèse soutenue par Marie PIRON le 7 janvier 1991 à l'université de Paris 6. En effet, un chapitre oriente l'étude du système d'échelles vers l'analyse de l'organisation spatiale en considérant deux niveaux d'information imbriqués : Zone de dénombrement et Secteur.

Du fait de la nature géographique des deux niveaux, on peut cartographier les structures dégagées par chaque analyse du changement d'échelle sur les unités spatiales de ces niveaux. Pour cela, le choix des couleurs des zones à cartographier dépend de la position de ces zones sur le plan factoriel principal des variables. Ce plan factoriel constituera la légende de la carte.

L'objectif est d'une part de déduire visuellement la position des unités spatiales sur le plan factoriel et d'autre part de localiser les phénomènes dégagés par ce même plan. Par conséquent, il s'agit de concevoir la légende de manière à traduire les oppositions que dégage chaque axe et de respecter l'indépendance des deux axes.

Le premier facteur oppose le vert (partie négative) et le magenta (partie positive); l'opposition entre le noir (partie négative) et le blanc (partie positive) caractérise le deuxième facteur. Une unité spatiale affectée d'un vert clair est représentative des types de ménages identifiés sur le quadrant en haut et à gauche du plan factoriel des variables. La précision de cette identification est affinée par un dégradé au sein des verts clairs.

La couleur du fond de carte est celle des unités spatiales proches de l'origine du plan factoriel, c'est-à-dire de la couleur 'neutre'. Cela offre l'avantage de mieux faire ressortir les valeurs extrêmes des axes.

Dans le cadre de ce travail, nous disposons de deux découpages géographiques : les secteurs et les zones de dénombrement, sous-ensemble des secteurs. La numérisation a été effectuée en mode vecteur avec le logiciel MYGALE (logiciel développé par l'Orstom). Les limites communes aux deux découpages ont été saisies une seule fois. Cette possibilité du logiciel, outre un gain de temps, assure la précision de la superposition de l'information d'un découpage à l'autre. Les cartes finales correspondent à l'ensemble de la zone d'enquête et non à l'ensemble des secteurs numérisés. Cela explique pourquoi certains résultats dans la fenêtre **output** présenteront des **valeurs manquantes** pour les variables, ou pourquoi, pour les cartes finales, nous ne retenons que les secteurs dont l'identifiant correspond aux secteurs étudiés.

IV.A INTRODUCTION DE COORDONNÉES GÉOGRAPHIQUES

IV.A.1 Unités spatiales ponctuelles

```

libname base '/usr/usr3/graph90/docu';
options nocenter nodate nonumber;
title 'tableau d"unités ponctuelles';
filename entre '/usr/usr3/graph90/docu/ouaga1b.cen';
      entre : nom logique du fichier externe ouaga1b.cen : nom physique

data base.ouaga1ce;
      ouaga1ce : tableau permanent SAS de coordonnées géographiques
infile entre;
input x 1-6 y 7-12
      x et y : coordonnées du point
      cle $ 13-16;
      identifiant qui permettra de faire le lien avec les données descriptives

proc contents data=base.ouaga1ce position;
proc print data=base.ouaga1ce (obs=10);
run;

```

- Extrait du fichier d'entrée :

9645	33153
9678	34821
9365	34661
9237	33151
9243	31319
9335	29438
9518	28377
9684	28946
9551	30762
9699	30881

- Ce programme sert à introduire une information géographique de type ponctuel (exemple : localisation géographique de villes, points bathymétriques ou de sondage, etc.).

tableau d'unites ponctuelles

CONTENTS PROCEDURE

Data Set Name: BASE.OUAGA1CE Type:
 Observations: 33 Record Len: 24
 Variables: 3
 Label:

----- Alphabetic List of Variables and Attributes -----

#	Variable	Type	Len	Pos	Label
3	CLE	Char	4	20	
1	X	Num	8	4	
2	Y	Num	8	12	

tableau d'unites ponctuelles

CONTENTS PROCEDURE

----- Variables Ordered by Position -----

#	Variable	Type	Len	Pos	Label
1	X	Num	8	4	
2	Y	Num	8	12	
3	CLE	Char	4	20	

tableau d'unites ponctuelles

OBS	X	Y	CLE
1	9645	3315	3
2	9678	3482	12
3	9365	3466	11
4	9237	3315	10
5	9243	3131	9
6	9335	2943	8
7	9518	2837	7
8	9684	2894	6
9	9551	3076	2
10	9699	3088	1

IV.A

IV.A.2 Unités spatiales surfaciques (polygones)

```

libname base '/usr/usr3/graph90/docu';
options nocenter nodate nonumber;
title 'Tableau d"unités surfaciques';
filename entre '/usr/usr3/graph90/docu/ouaga2.sas';
      entre : nom logique du fichier externe ouaga2.sas : nom physique

data base.ouaga2g;
      ouaga2g : tableau permanent SAS des coordonnées géographiques
infile entre;
input ident $ 1-5 x 21-26 y 27-32
      identifiant x et y : coordonnées d'un point
      segment 33-35;
      numéro d'ordre du polygone. Il s'agit d'une variable, dont le nom est prédéfini par SAS, nécessaire
      lorsque plusieurs unités spatiales ont un même identifiant (cf. structure des données géographiques
      dans les généralités au début de ce paragraphe).

proc contents data=base.ouaga2g;
proc print data=base.ouaga2g (obs=10);
run;

```

- Extrait du fichier d'entrée :

```

1 0 9618 3216 1
1 0 9625 3109 1
1 0 9629 2969 1
1 0 9764 2979 1
1 0 9772 3052 1
1 0 9760 3135 1
1 0 9732 3238 1
10 0 9084 3250 1
10 0 9487 3267 1
10 0 9379 3339 1

```

- Ce programme sert à introduire une information géographique de type surfacique (exemple : départements, régions, îlots urbains non jointifs, etc.).

Tableau d'unités surfaciques

CONTENTS PROCEDURE

Data Set Name: BASE.OUAGA2G Type:
 Observations: 2497 Record Len: 33
 Variables: 4
 Label:

-----Alphabetic List of Variables and Attributes-----

#	Variable	Type	Len	Pos	Label
1	IDENT	Char	5	4	
4	SEGMENT	Num	8	25	
2	X	Num	8	9	
3	Y	Num	8	17	

Tableau d'unités surfaciques

OBS	IDENT	X	Y	SEGMENT
1	1 0	9618	3216	1
2	1 0	9625	3109	1
3	1 0	9629	2969	1
4	1 0	9764	2979	1
5	1 0	9772	3052	1
6	1 0	9760	3135	1
7	1 0	9732	3238	1
8	10 0	9084	3250	1
9	10 0	9487	3267	1
10	10 0	9379	3339	1

IV.A

IV.A.3 Unités spatiales linéaires (arcs)

```

options      nocenter      nodate      nonumber;
libname      base          '/usr/usr3/graph90/docu';
filename     entre        '/usr/usr3/graph90/docu/ouaga1b.lim';
title1      'Tableau provisoire des limites zonales';

data inter;
  tableau provisoire des arcs dans la base
  infile     entre;
  input      cle $ 1-4
            identifiant
            etat 21      x 22-27      y 28-33;
            etat : variable indiquant si le point de coordonnées (x,y) est le premier point ou non d'un arc

proc print   data=inter    (obs=10);
run;
run : provoque l'exécution des étapes qui précèdent avec leur environnement propre (titre, options, etc.)

title1      'Tableau permanent des limites zonales';
modification de l'environnement
data        base.limsect;
            limsect : tableau permanent des arcs dans la base
length     fonction $ 8;
            variable alphanumérique d'un tableau annoté dont le nom est prédéfini par SAS
            et qui indique le type d'action à exécuter

set        inter;
if         etat=1 then    fonction='move';
            déplacement sans tracé du point précédent au point courant
            else         fonction='draw';
            déplacement avec tracé du point précédent au point courant

keep      cle fonction x y;

proc contents data=base.limsect;
proc print   data=base.limsect    (obs=10);
run;

```

- Déplacement avec ou sans tracé : les variables **xlast** et **ylast** sont des variables internes de SAS qui mémorisent le couple de coordonnées (x,y) du point précédemment traité. Ce qui permet de dire que la valeur **'draw'** de la variable **fonction** correspond à un déplacement plume baissée (avec tracé) du point précédent au point courant.
- Ce programme sert à introduire une information géographique de type linéaire (exemple : limites de départements, tracés de routes, réseau hydrographique, etc.). Cette information peut être considérée comme un élément de « fond de carte » : en effet, SAS ne dispose pas de module de traitement des informations de type réseaux. Cela explique aussi la structure du tableau que nous décrivons ici, structure conçue pour une utilisation uniquement graphique (dans un **tableau annoté**).

• • •

• • •

● Extrait du fichier d'entrée

```

1 1 9618 3216
1 2 9625 3109
1 2 9629 2969
1 2 9764 2979
1 2 9772 3052
1 2 9760 3135
1 2 9732 3238
1 2 9618 3216
10 1 9084 3250
10 2 9487 3267

```

La valeur 1 de la variable etat signifie que le point de coordonnées (x,y) est le premier de l'arc.

Tableau provisoire des limites zonales

OBS	CLE	ETAT	X	Y
1	1	1	9618	3216
2	1	2	9625	3109
3	1	2	9629	2969
4	1	2	9764	2979
5	1	2	9772	3052
6	1	2	9760	3135
7	1	2	9732	3238
8	1	2	9618	3216
9	10	1	9084	3250
10	10	2	9487	3267

Tableau permanent des limites zonales

CONTENTS PROCEDURE

```

Data Set Name:  BASE.LIMSECT
Observations:  810
Variables:      4
Label:
Type:
Record Len: 32

```

-----Alphabetic List of Variables and Attributes-----

#	Variable	Type	Len	Pos	Label
2	CLE	Char	4	12	
1	FUNCTION	Char	8	4	
3	X	Num	8	16	
4	Y	Num	8	24	

Tableau permanent des limites zonales

OBS	FUNCTION	CLE	X	Y
1	move	1	9618	3216
2	draw	1	9625	3109
3	draw	1	9629	2969
4	draw	1	9764	2979
5	draw	1	9772	3052
6	draw	1	9760	3135
7	draw	1	9732	3238
8	draw	1	9618	3216
9	move	10	9084	3250
10	draw	10	9487	3267

IV.B REPRÉSENTATION GRAPHIQUE DES DONNÉES

Les représentations graphiques des données d'une base seront différentes selon le type de tableaux sur lequel on travaille : les tableaux descriptifs font appel à des procédures qui visualiseront l'information sous forme de courbes ou autres représentations habituelles de ce type de données; les tableaux géographiques, associés ou non à un tableau graphique, permettent de créer des cartes.

La procédure **gmap** cartographie les variables dans un espace à deux dimensions (2D) ou à trois dimensions (3D). Elle a besoin, optionnellement, de deux tableaux : un tableau **data** dont l'une des variables est la variable à cartographier, et un tableau **map** qui contient l'information géographique. Ces deux tableaux doivent avoir au moins une variable commune qui permettra à SAS d'affecter à chaque unité géographique la valeur thématique correspondante de la variable traitée.

La gestion du graphique appliquée à des tableaux descriptifs permet de composer des graphes (diagrammes et courbes) plus sophistiqués que les procédures non graphiques équivalentes (type **chart** ou **plot**) : gestion des éléments de représentation liés au graphique (axes, légendes, symboles, etc), calculs d'extrapolation (droites et courbes de régression).

IV.B.1 Cartographie en deux dimensions : proc gmap option choro

IV.B.1.1 Représentation choroplèthe ou par plages

initialisations

```
libname base '/usr/usr3/graph90/docu';
options nocenter nodate nonumber; goptions reset=all;
goptions device=sun border cback=white ctext=black;
affichage sur l'écran avec cadre et fond d'écran blanc; écriture des textes en noir
title1 f=swiss c=black 'OUAGADOUGOU';
titre général de la sortie graphique avec choix de la fonte de caractères et texte en noir
footnote '';
pattern1 c=black v=solid; pattern2 c=green v=solid;
pattern3 c=yellow v=solid; pattern4 c=red v=solid;
définition des paramètres de la représentation des classes (c : couleur, v : trame ou pourcentage de la couleur)
```

```
proc format; value faxx
  low - -500 = 'moins de -500'
  -500 - 0 = 'de -500 a 0'
  0 - 500 = 'de 0 a 500'
  500 - high = 'plus de 500';
format pour classification de variable en quatre classes
```

```
data prov; set base.ouaga2;
  keep ident axx axy;
sélection des variables descriptives utiles à la cartographie
```

1^{re} carte : cartographie 2D de la 1^{re} variable

```
title2 f=swiss c=black 'Secteurs selon axx';
sous-titre spécifique à la carte : affichage, en noir, sur la deuxième ligne, d'un titre 'Secteurs selon la variable axx', dans la police de caractère swiss
```

• • •



IV.B

```

...
proc gmap data=prov      map=base.ouaga2g;
étape de création de la carte
  prov : tableau des variables à cartographier
  ouaga2g : tableau des coordonnées géographiques
  id      ident;
  variable commune aux deux tableaux permettant d'associer les valeurs de la variable et les unités géographiques correspondantes
  choro axx /      discrete
  variable discrète à cartographier
  couleur des contours des unités spatiales de la variable
  couline=white;

  format axx faxx.;
  recodage de la variable axx suivant la classification définie ci-dessus
run;

2e carte : cartographie 2D de la 2e variable
title2      f=swiss      c=black      'Secteurs selon axy';
sous-titre spécifique à la carte
proc gmap data=prov      map=base.ouaga2g;
  id      ident;
  choro axy /      discrete      couline=white;
  format axy faxy.;
run;

```

- Définir des **pattern** permet de choisir la représentation des classes de la variable cartographiée (couleurs ou trames). Au moment de l'affichage, chaque classe est affectée d'une représentation correspondant à son rang dans la table des représentations possibles : la première classe est affectée de la représentation du pattern1, la seconde de celle du pattern2, etc. sachant que si la classe n n'existe pas, le n ième pattern sera affecté à la classe suivante. Dans l'exemple traité, s'il se trouvait qu'une des classes ne soit pas représentée dans la zone d'étude (par exemple 'de 0 à 500'), la classe suivante ('plus de 500') prendrait la représentation du 3^e pattern au lieu de celle du 4^e, car il s'agirait alors du premier disponible dans le tableau des représentations possibles.

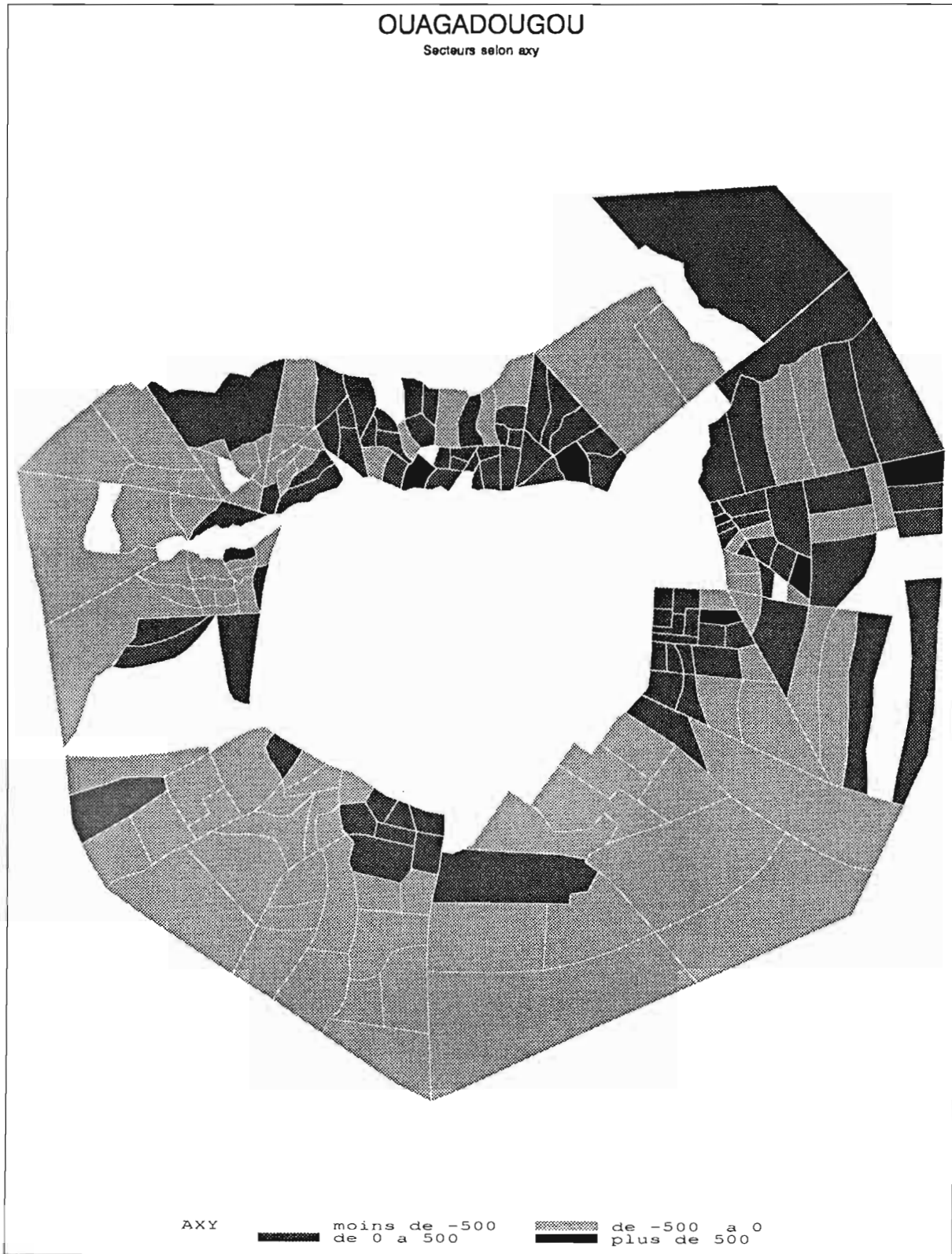
Choix des représentations	Classes à représenter
pattern1 c=black v=solid;	'moins de -500'
pattern2 c=green v=solid;	'de -500 à 0'
pattern3 c=yellow v=solid;	'de 0 à 500'
pattern4 c=red v=solid;	'plus de 500'

Affectation théorique choisie

1 ^{re} classe	'moins de -500'	1 ^{er} pattern
2 ^e classe	'de -500 à 0'	2 ^e pattern
3 ^e classe	'de 0 à 500'	3 ^e pattern
4 ^e classe	'plus de 500'	4 ^e pattern

Affectation réelle si, par exemple, la classe 'de 0 à 500' était absente

1 ^{re} classe	'moins de -500'	1 ^{er} pattern
2 ^e classe	'de -500 à 0'	2 ^e pattern
3 ^e classe	'de 0 à 500'	classe non représentée
4 ^e classe	'plus de 500'	3 ^e pattern



IV.B

IV.B.1.2 Représentation des limites des unités surfaciques

initialisations

```
libname base '/usr/usr3/graph90/docu';
goptions reset=all;
goptions device=sun border cback=white
          ctext=black htext=1;
```

affichage sur l'écran avec cadre et fond d'écran blanc. Écriture des textes en noir, leur taille est égale à une unité courante.

```
title; title1 f=swiss 'VUE D ENSEMBLE DE LA ZONE D ETUDE';
```

réinitialisation des titres; titre général de la carte

```
data prov; set base.ouaga2g;
              h=1.;
              création d'une variable discrète pour l'ensemble du tableau
```

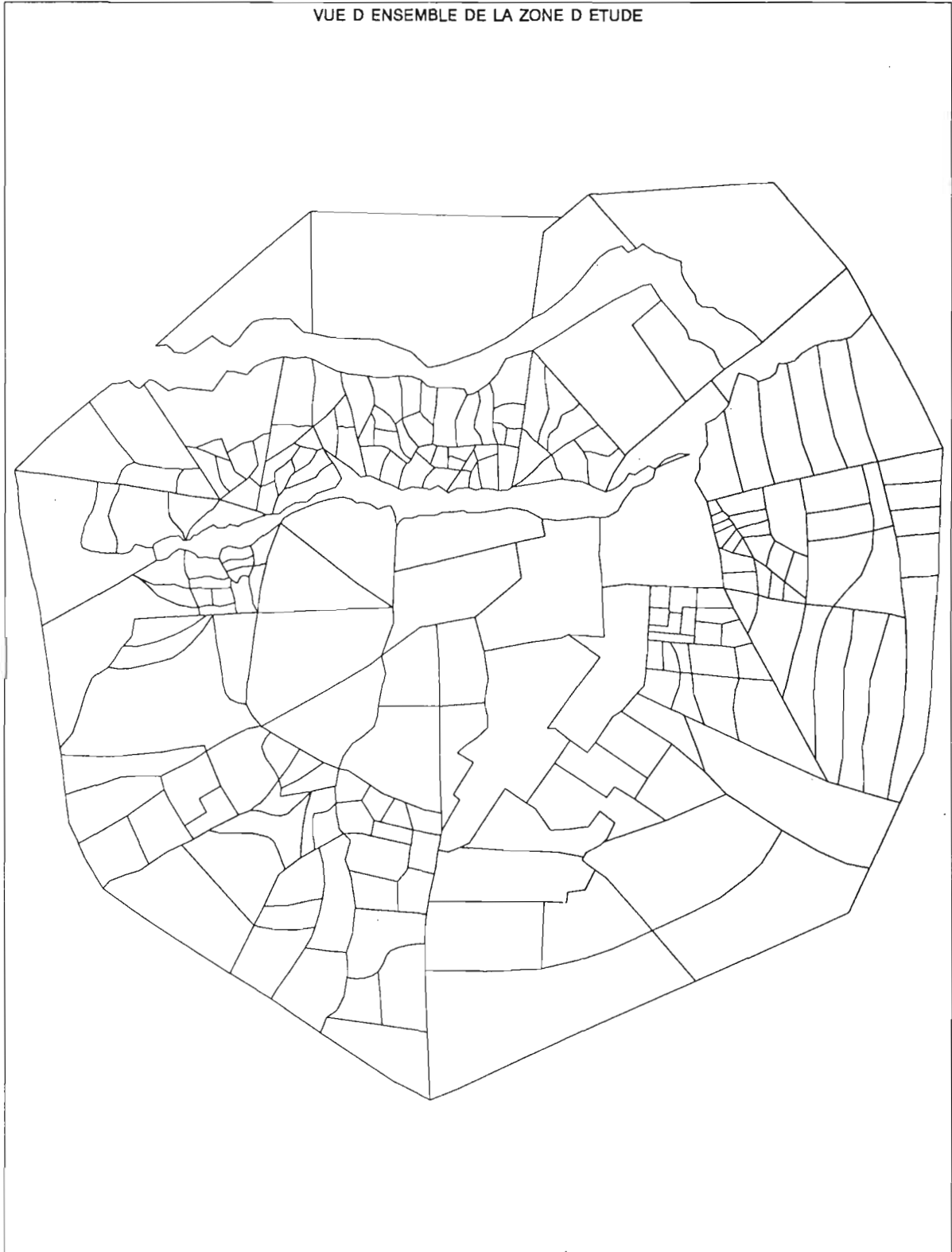
```
keep ident h;
sélection des variables utiles à la cartographie
```

cartographie 2D de la variable h

```
proc gmap data=prov
          tableau des données descriptives
          map=base.ouaga2g;
          tableau des coordonnées géographiques
          id ident;
          identifiant commun aux deux tableaux qui permet de relier données descriptives et données géographiques
          choro h / nolegend;
          variable à cartographier carte sans légende
          pattern1 c=black v=e;
          représentation de la classe : couleur noire, pas de trame ou couleur à l'intérieur de l'unité surfacique
run;
```

- Cette carte représente l'ensemble des unités spatiales du découpage géographique « Zones de dénombrement » de la base Ouagadougou.
- La valeur 1 affectée à la variable h permet de réunir l'ensemble des unités surfaciques à cartographier dans une seule classe. Les unités surfaciques de la classe seront représentées par le tracé des contours, comme cela est défini dans le **pattern** (pas de représentation à l'intérieur du polygone).

VUE D ENSEMBLE DE LA ZONE D ETUDE



IV.B

IV.B.1.3 Cartographie d'une variable descriptive selon les modalités d'une autre variable

initialisations de l'environnement graphique

```
goptions device=sun border cback=white colors=(black);
définition du périphérique, de l'environnement graphique et réinitialisation de la palette de couleurs
title1 f=swiss c=black 'OUAGADOUGOU';
title3 '';
```

titres. Utiliser des titres dont la chaîne est vide permet d'aérer la présentation générale.

```
pattern1 c=orange v=solid; pattern2 c=brown v=solid;
pattern3 c=green v=s; pattern4 c=yellow v=s;
```

c=orange affecte la couleur 'orange' à la première classe rencontrée, v=solid ou v=s correspond à 100% de la couleur

définition des paramètres de la représentation des classes

```
legend1 value=(f=swiss h=0.75)
value=(options) sert à définir l'environnement graphique spécifique de la légende; f définit la police et h la hauteur des caractères.
down=4
positionne verticalement quatre pavés de légende (position horizontale par défaut ou across=n pour réinitialiser)
label=none frame
pas de titre de légende cadre autour de la légende
c=black;
couleur à utiliser dans la légende
```

définition des paramètres de la légende à associer au graphique

initialisations de l'environnement non graphique

```
libname base '/usr/usr3/graph90/docu';
options nonumber nocenter nodate;
options symbolgen;
```

affichage, dans la fenêtre output, des informations relatives aux macrovariables

```
proc format; value faxy
low - -500 = 'moins de -500'
-500 - 0 = 'de -500 a 0'
0 - 500 = 'de 0 a 500'
500 - high = 'plus de 500';
```

• • •

Fichier log:

```

"Hello World!"
NOTE: AUTOEXEC processing completed.
 1 /* ----- */
 2 /* CARTOGRAPHIE D'UNE VARIABLE DESCRIPTIVE */
 3 /* SELON LES MODALITES D'UNE AUTRE VARIABLE */
 4 /* ----- */
 5
 6 goptions device=sun border cback = white colors=(black);
 7 options symbolgen ;
 8 title1 f=swiss c=black 'OUAGADOUGOU';
 9 title3 '';
10 pattern1 c=grayf8 v=solid;
11 pattern2 c=gray90 v=solid;
12 pattern3 c=gray40 v=solid;
13 pattern4 c=gray00 v=solid;
14 legend1 value=(f=swiss h=0.75 c=black)
15 down=4
16 label=none frame;
17
18 libname base 'usr/usr3/graph90/docu';
19
20 proc format ; value faxy
21 low - -500='moins de -500'
22 -500 - 0='de -500 a 0'
23 0 - 500='de 0 a 500'
24 500 - high='plus de 500';
NOTE: Format FAXY has been output.
25
26 %macro carto(val1,val2,classe);
27 data prov;
28 set base.ouaga2;
29 if axx>&val1 and axx<&val2;
30 format axy faxy;
31 /* CARTOGRAPHIE 2D DE LA 2eme VARIABLE SELON MODALITES DE LA 1ere*/
32 title2 f=swiss c=black &classe;
33 proc gmap data=prov map=base.ouaga2g all;
34 id ident;
35 choro axy/ discrete
36 coutline=black
37 cempty=black
38 ctext=black
39 legend=legend1;
40 %mend carto;
41
42 goptions gaccess='sasgastd>graph1' device=aplplus;
43 %carto(-1500,-500,'AXES YY selon XX < -500');
NOTE: The PROCEDURE FORMAT used 1.00 seconds.
SYMBOLGEN: Macro variable VAL1 resolves to -1500
SYMBOLGEN: Macro variable VAL2 resolves to -500
SYMBOLGEN: Macro variable CLASSE resolves to 'AXES YY selon XX < -500'
NOTE: The data set WORK.PROV has 7 observations and 3 variables.
NOTE: The DATA statement used 2.00 seconds.
44
45 goptions gaccess='sasgastd>graph2' device=aplplus;
46 %carto(-500,0,'AXES YY selon -500 < XX < 0');
WARNING: The legend will leave the space for the specified DOWN=4 but there are not enough entries in the legend to use all of the rows.
NOTE: The PROCEDURE GMAP used 22.00 seconds.
SYMBOLGEN: Macro variable VAL1 resolves to -500
SYMBOLGEN: Macro variable VAL2 resolves to 0
SYMBOLGEN: Macro variable CLASSE resolves to 'AXES YY selon -500 < XX < 0'
NOTE: The data set WORK.PROV has 66 observations and 3 variables.
NOTE: The DATA statement used 5.00 seconds.
47
48 goptions gaccess='sasgastd>graph3' device=aplplus;
49 %carto(0,500,'AXES YY selon 0 < XX < 500');
WARNING: The legend will leave the space for the specified DOWN=4 but there are not enough entries in the legend to use all of the rows.
NOTE: The PROCEDURE GMAP used 20.00 seconds.
SYMBOLGEN: Macro variable VAL1 resolves to 0
SYMBOLGEN: Macro variable VAL2 resolves to 500
SYMBOLGEN: Macro variable CLASSE resolves to 'AXES YY selon 0 < XX < 500'
NOTE: The data set WORK.PROV has 98 observations and 3 variables.
NOTE: The DATA statement used 6.00 seconds.
50
51 goptions gaccess='sasgastd>graph4' device=aplplus;
52 %carto(500,2500,'AXES YY selon XX > 500');
NOTE: The PROCEDURE GMAP used 20.00 seconds.
SYMBOLGEN: Macro variable VAL1 resolves to 500
SYMBOLGEN: Macro variable VAL2 resolves to 2500
SYMBOLGEN: Macro variable CLASSE resolves to 'AXES YY selon XX > 500'
NOTE: The data set WORK.PROV has 17 observations and 3 variables.
NOTE: The DATA statement used 3.00 seconds.
53
54 run;
WARNING: The legend will leave the space for the specified DOWN=4 but there are not enough entries in the legend to use all of the rows.
NOTE: The PROCEDURE GMAP used 21.00 seconds.
55
NOTE: SAS Institute Inc., SAS Circle, PO Box 8000, Cary, NC 27512-8000

```

IV.B

• • •

macro-procédure définissant une suite de traitements

```
%macro carto (val1,val2,classe);
```

%macro : macroprocédure appelée carto

(val1,val2,classe) : liste de paramètres à transmettre à chaque appel de la macroprocédure

```
data prov; set base.ouaga2;
```

```
if axx>&val1 and axx<&val2;
```

sélectionne toutes les observations dont la valeur de la variable axx est comprise entre les valeurs de val1 et de val2

```
format axy faxy.;
```

classification de la variable axy pour toutes les observations sélectionnées

cartographie 2D de la 2^e variable selon les modalités de la 1^{re}

```
title2 f=swiss c=black &classe;
```

&classe : variable paramétrée correspondant au sous-titre que l'on veut associer à la carte au moment de l'exécution de la macroprocédure. Ce titre sera écrit en noir et dans la police swiss.

```
proc gmap data=prov map=base.ouaga2g all;
```

prov : tableau des données descriptives

ouaga2g : tableau des données géographiques

all : toutes les unités surfaciques seront tracées

```
id ident;
```

```
choro axy /discrete ctext=black
```

```
coutline=black
```

couleur des limites des unités surfaciques de la variable

```
cempty=black
```

couleur des limites des autres unités surfaciques

```
legend=legend1;
```

présenter la légende selon les options indiquées dans legend1

```
%mend carto;
```

%mend indique la fin de la macroprocédure appelée carto

cartographie

```
%carto (-1500,-500,'AXES YY selon XX < -500');
```

L'appel à la macroprocédure «carto» %carto, peut être comparé à l'appel d'un sous-programme : il permet d'effectuer le même traitement plusieurs fois sans réécrire toutes les instructions SAS relatives à celui-ci.

Résultat : une carte de la variable axy en fonction de la première modalité de axx, c'est-à-dire représentation en 4 classes de la variable axy pour toutes les unités dont la valeur de la variable axx est comprise entre -1500 et -500. Elle aura pour titre 'AXES YY selon XX < -500'.

```
%carto (-500,0,'AXES YY selon -500 < XX < 0');
```

cartographie des classes de axy en fonction de la seconde modalité de axx

```
%carto (0,500,'AXES YY selon 0 < XX < 500');
```

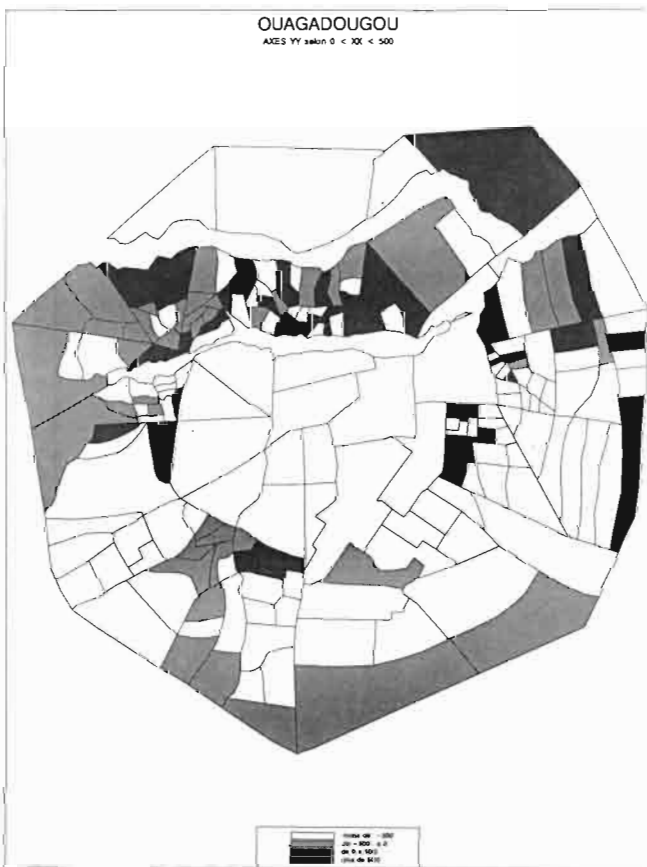
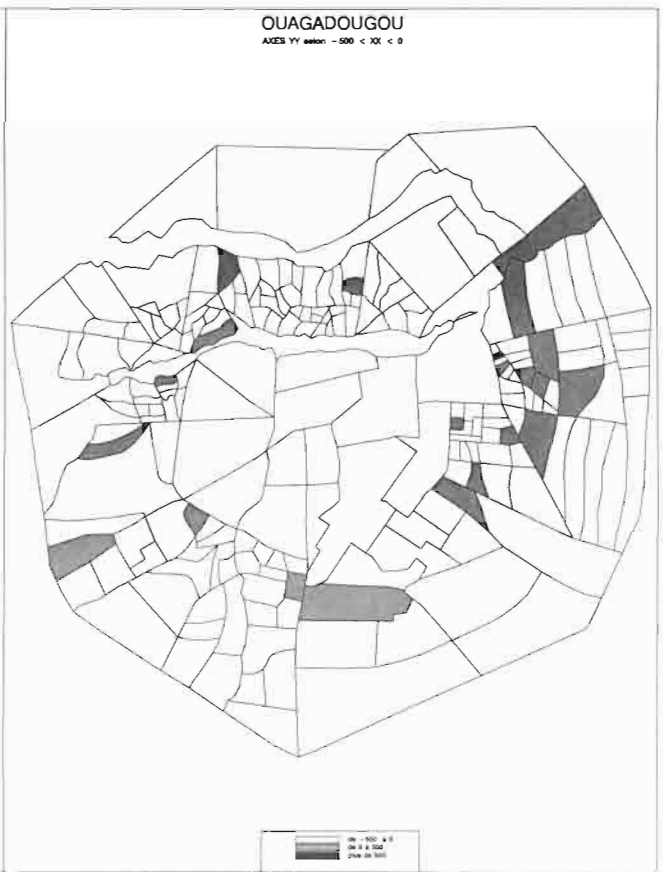
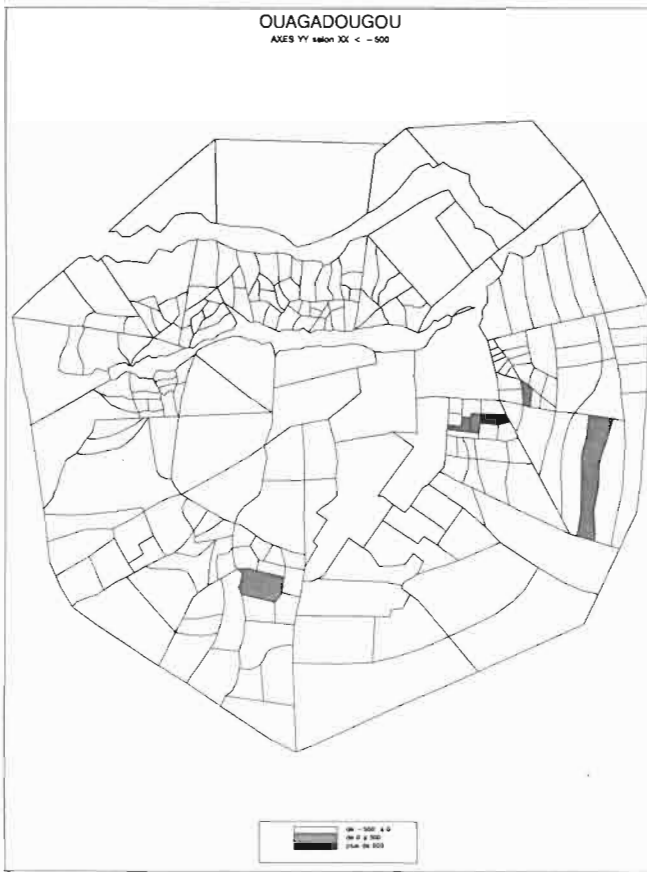
cartographie de axy en fonction de la troisième modalité de axx

```
%carto (500,2500,'AXES YY selon XX > 500');
```

cartographie de axy en fonction de la quatrième modalité de axx

```
run;
```

- Voir sur la sortie de la fenêtre **log**, les messages relatifs à l'exécution de %carto.
- Chaque carte correspond à une modalité de la variable de partitionnement axx.



IV.B

IV.B.1.4 Cartographie d'une variable descriptive selon les modalités d'une autre variable : option by de choro

initialisation de l'environnement

```
libname base '/usr/usr3/graph90/docu';
goptions reset=all;
réinitialise l'environnement graphique
goptions device=sun border cback=white ctext=gray90;
options nocenter nodate nonumber;
title1 f=swiss c=black 'OUAGADOUGOU';
footnote '';
pattern1 c=grayf8 v=solid; pattern2 c=gray90 v=solid;
pattern3 c=gray40 v=solid; pattern4 c=gray00 v=solid;

proc format; value faxx
    low - -500 = 'moins de -500'
    -500 - 0 = 'de -500 a 0'
    0 - 500 = 'de 0 a 500'
    500 - high = 'plus de 500';
```

création d'un tableau unique de données

```
data provv; set base.ouaga2; format axx faxx.;
classification de la variable axx
proc print data=provv (obs=10);
proc sort data=provv out=prov; by ident;
proc sort data=base.ouaga2g out=provg; by ident;
data entier; merge prov provg; by ident;
fusion des tableaux descriptif et géographique
proc sort data=entier out=prov1; by axx;
tri sur la variable de partitionnement, variable de l'option by de la proc gmap
proc print data=prov1 (obs=10);
```

cartographie

```
proc gmap data=prov1 map=prov1;
    données descriptives et géographiques dans un même tableau
    title2 f=swiss c=black 'Secteurs selon axx';
    sous-titre de la carte
    id ident; by axx;
    variable de sélection
    choro axy / discrete missing outline=gray40;
    traiter les valeurs manquantes de axy
    format axy faxx.;
run;
```

- Le résultat est peu différent de celui du programme précédent : 5 cartes représentant les différentes modalités de la variable à cartographier axy en fonction des modalités de axx, variable de partitionnement. Les valeurs des deux variables sont réparties en 5 classes : 4 classes définies par le format de classification et la classe des valeurs manquantes.

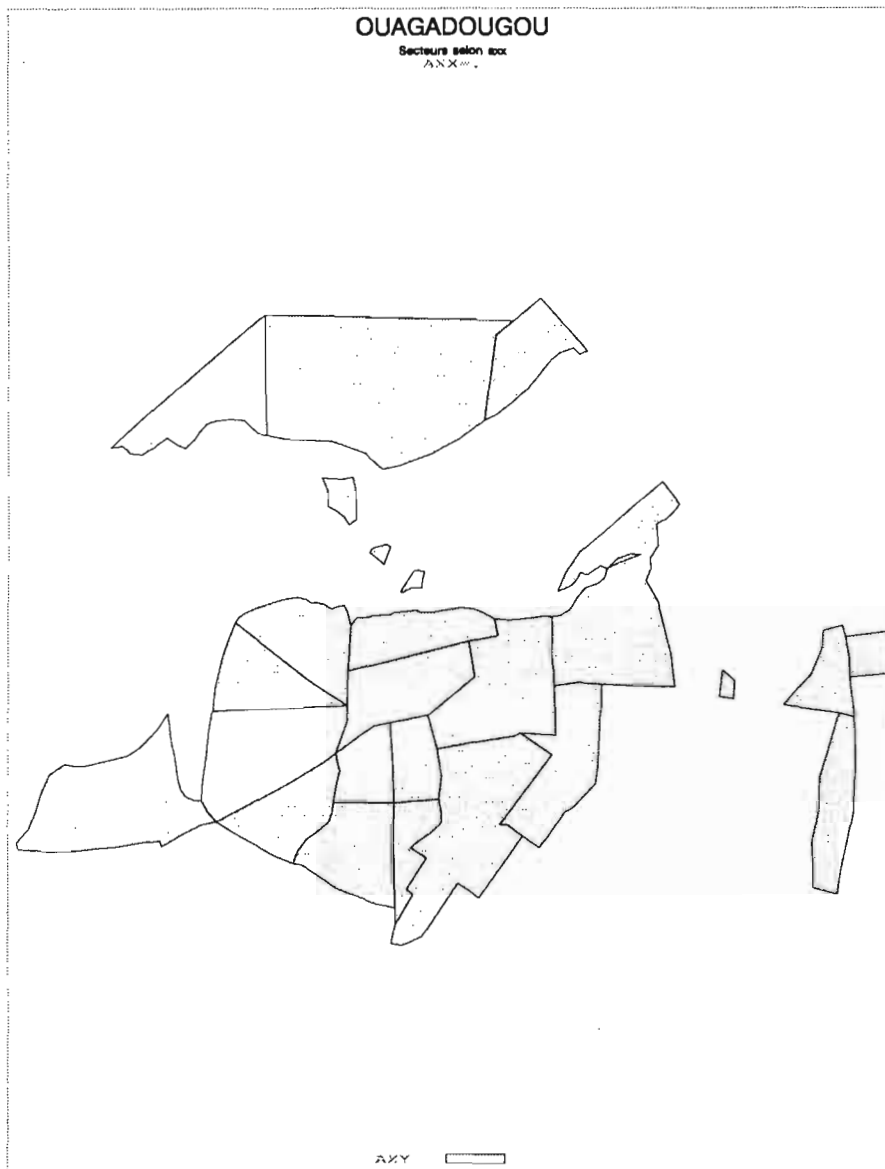
• • •

OUAGADOUGOU

OBS	IDENT	AXX	AXY
1	15 1	de -500 a 0	4
2	15 2	de -500 a 0	-240
3	15 3	de 0 a 500	-235
4	15 4	de 0 a 500	-425
5	15 5	de -500 a 0	-247
6	16 1	de -500 a 0	-172
7	16 2	de 0 a 500	-154
8	16 3	de -500 a 0	-471
9	16 4	de -500 a 0	-128
10	16 5	de -500 a 0	-177

OUAGADOUGOU

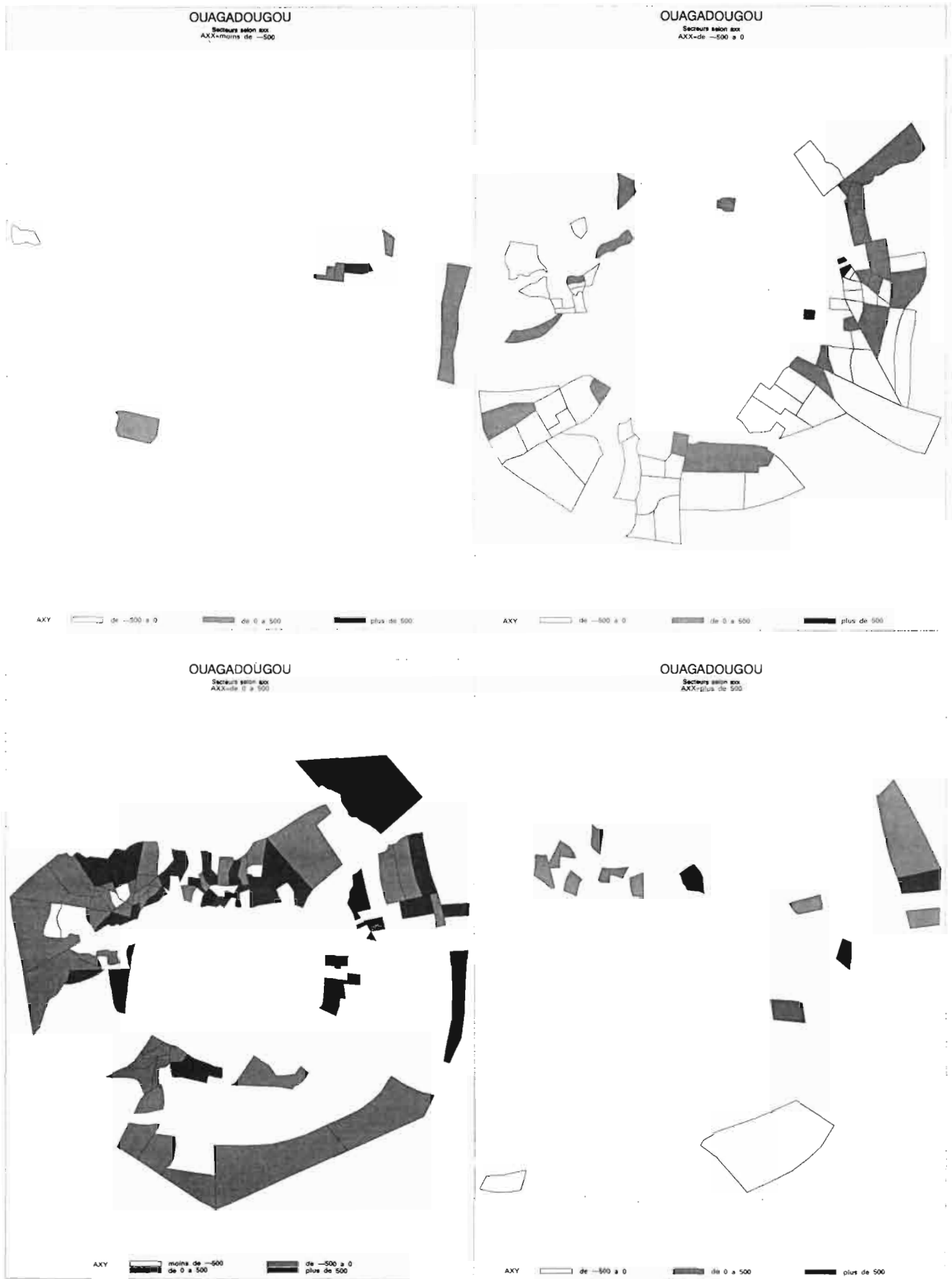
OBS	IDENT	AXX	AXY	X	Y	SEGMENT
1	1 0	.	.	9618	3216	1
2	1 0	.	.	9625	3109	1
3	1 0	.	.	9629	2969	1
4	1 0	.	.	9764	2979	1
5	1 0	.	.	9772	3052	1
6	1 0	.	.	9760	3135	1
7	1 0	.	.	9732	3238	1
8	10 0	.	.	9084	3250	1
9	10 0	.	.	9487	3267	1
10	10 0	.	.	9379	3339	1



• • •

- Le titre 'OUAGADOUGOU' et le sous-titre 'AXES YY selon XX' seront écrits en noir (option de title1 et de title2); SAS ajoute 'AXX=nomclasse' dans la couleur définie pour l'écriture des textes dans l'option **goptions**.
- Si nous comparons la première carte du programme précédent et la seconde de celui-ci, l'information est la même mais sa localisation dans l'espace est différente. En effet, **gmap** détermine les valeurs minimales et maximales des coordonnées à afficher pour chaque axe, puis détermine le coefficient de transformation à affecter aux coordonnées pour que le graphique à visualiser occupe tout l'espace. Les cartes du programme précédent sont toutes à la même « échelle » : l'ensemble des unités surfaciques sont représentées avec ou sans trame. Dans ce programme, « l'échelle » varie en fonction de la surface à cartographier, c'est-à-dire de la localisation des unités surfaciques représentées.
- Les options **xsize** et **ysize** de **gmap** permettent de préciser la dimension physique de la carte finale; si ces valeurs excèdent les valeurs par défaut du périphérique choisi, elles ne sont pas prises en compte.
- Les illustrations présentées ici sont des réductions des sorties graphiques du programme. Ces sorties auraient pu être obtenues directement avec l'étape **gmap** suivante :

```
proc      gmap  data=prov1  map=prov1;
          title2  f=swiss    c=black    'Secteurs selon axx';
          id      ident;      by        axx;
          choro   axy        /        discrete missing  outline=gray40
                                xsize=10 cm  ysize=10 cm;
          format  axy faxx.;
```



IV.B

IV.B.2 Cartographie en trois dimensions : proc gmap options block, prism ou surface

Les options **block** et **prism** représentent la variable par blocs ou par prismes (polyèdres). Le bloc ou le prisme est placé sensiblement au centre de l'unité spatiale et sa hauteur est fonction de la valeur de la variable pour cette unité.

L'option **surface** donne un résultat semblable à ce que l'on a coutume d'appeler « bloc 3D ». La hauteur du point central de chaque unité est fonction de la valeur de la variable pour cette unité.

IV.B.2.1 Représentation en trois dimensions option block

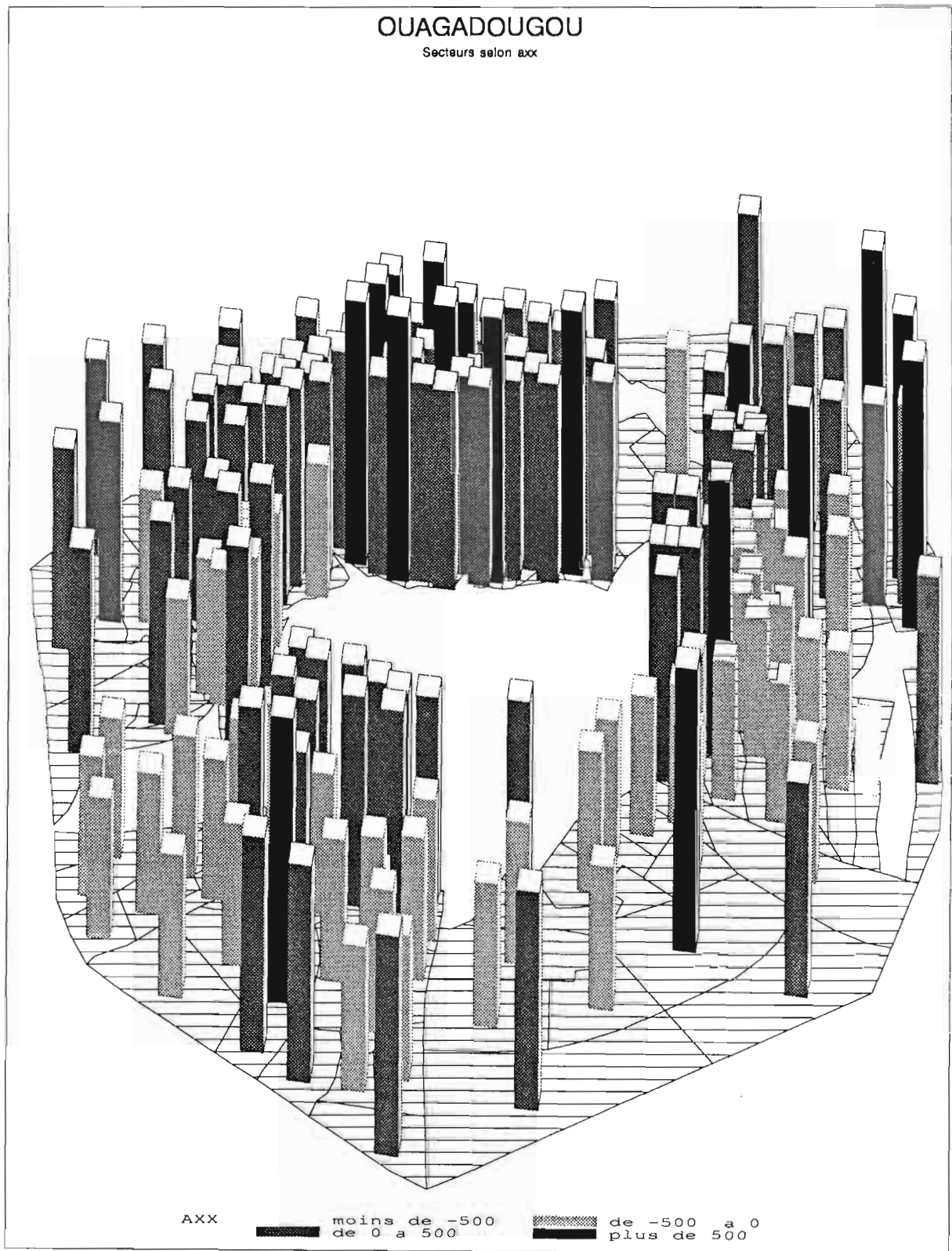
initialisations

```
libname      base          '/usr/usr3/graph90/docu';
goptions    device=sun    border      cback=white      ctext=black;
title1      f=swiss      c=black      'OUAGADOUGOU';
footnote    '';
pattern1    c=grayf8     v=solid;    pattern2    c=gray90     v=solid;
pattern3    c=gray40     v=solid;    pattern4    c=gray00     v=solid;

proc format;      value  faxx
                 low  - -500 = 'moins de -500'
                 -500 - 0   = 'de -500 a 0'
                 0 - 500   = 'de 0 a 500'
                 500 - high = 'plus de 500';
```

cartographie 3D de la variable

```
proc gmap data=base.ouaga2      map=base.ouaga2g;
         title2 f=swiss      c=black      'Secteurs selon axx';
         id      ident;
         block  axx /      discrete    coutline=black;
         format axx faxx.;
run;
```



IV.B

IV.B.2.2 Représentation en trois dimensions option prism

initialisations

```

libname base '/usr/usr3/graph90/docu';
goptions reset=all;
goptions device=sun border cback=white ctext=black;
title1 f=swiss c=black 'OUAGADOUGOU';
footnote '';
pattern1 c=grayf8 v=solid; pattern2 c=gray90 v=solid;
pattern3 c=gray40 v=solid; pattern4 c=gray00 v=solid;

proc format; value faxx
    low - -500 = 'moins de -500'
    -500 - 0 = 'de -500 a 0'
    0 - 500 = 'de 0 a 500'
    500 - high = 'plus de 500';

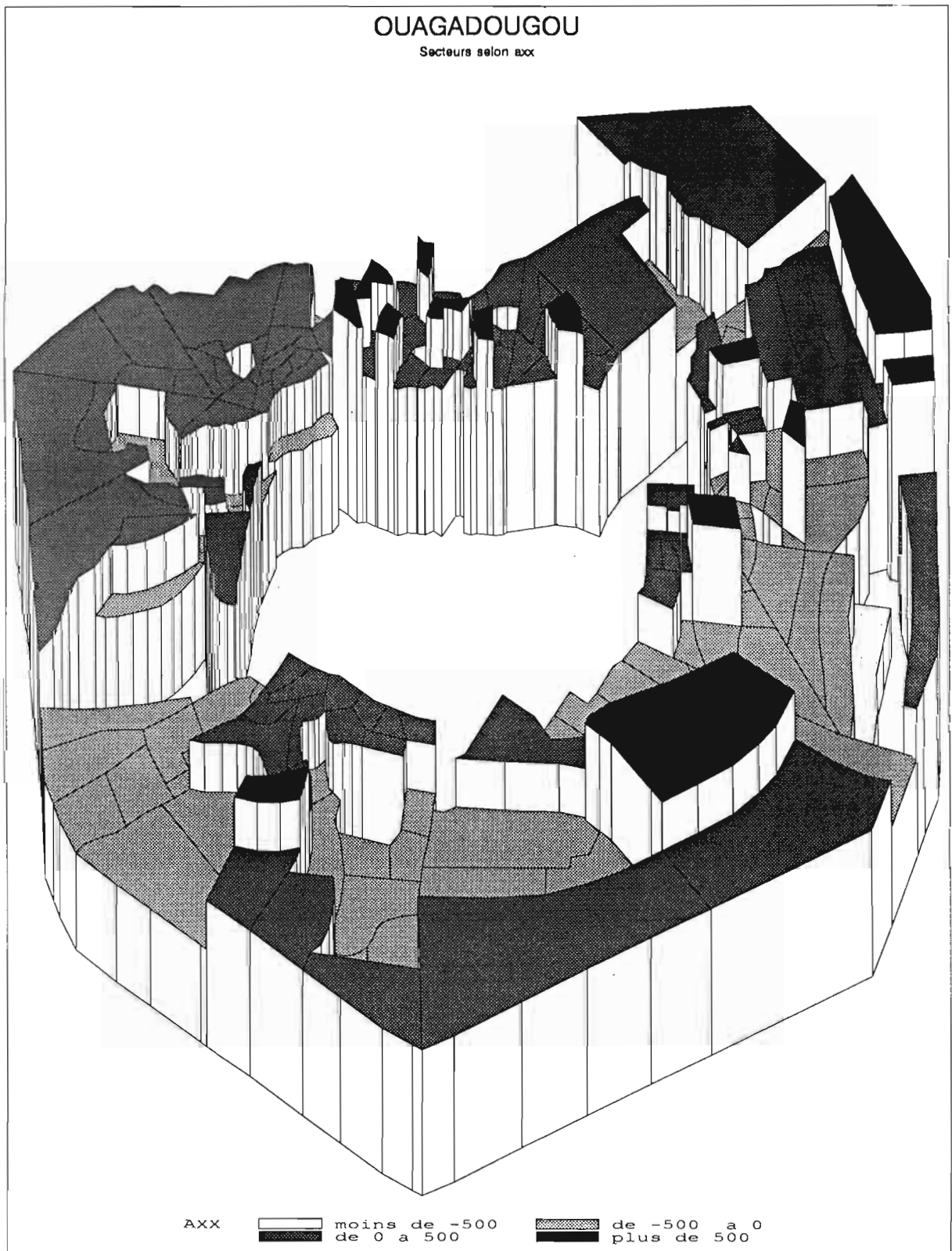
```

cartographie 3D de la variable

```

proc gmap data=base.ouaga2 map=base.ouaga2g;
    title2 f=swiss c=black 'Secteurs selon axx';
    id ident;
    prism axx / discrete coutline=black;
    format axx faxx.;
run;

```



IV.B

IV.B.2.3 Représentation en trois dimensions option surface

initialisations

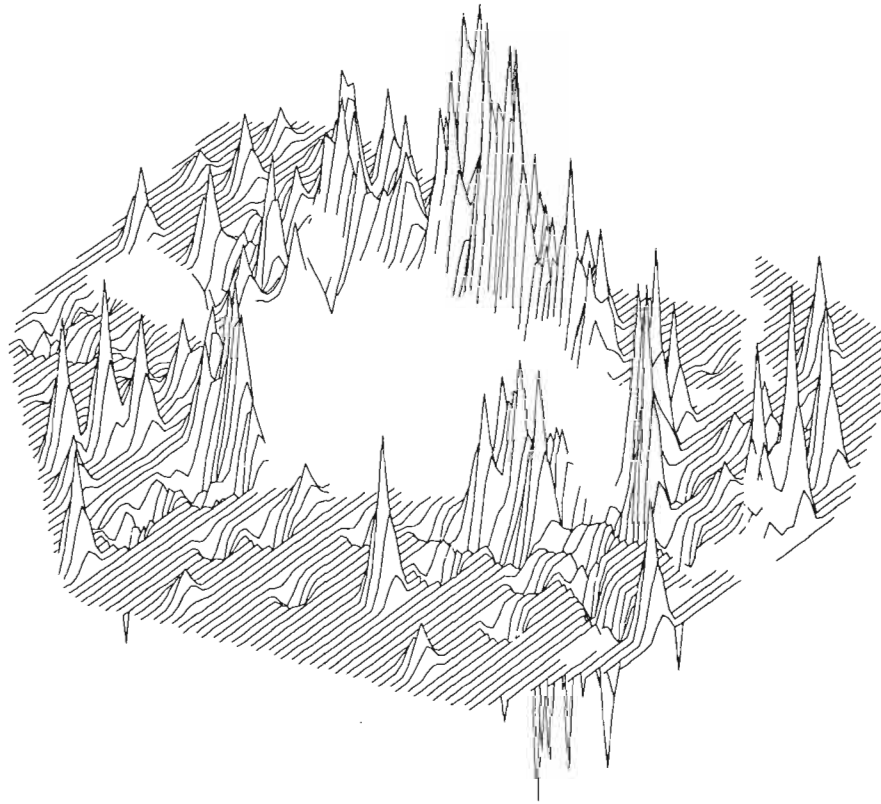
```
libname    base          '/usr/usr3/graph90/docu';
goptions   reset=all;
title1     f=swiss      c=black    'OUAGADOUGOU';
footnote   '';
goptions   gaccess='sasgastd>graph'  device=aplplus;
           nom du fichier du résultat graphique  périphérique de sortie
```

cartographie 3D de la variable

```
proc gmap  data=base.ouaga2          map=base.ouaga2g;
           title2 f=swiss          c=black    'Secteurs selon axx';
           id      ident;
           surface axx /          cbody=black    n=100
           couleur de la surface          nombre de lignes
           rotate=45
           angle de rotation de la carte autour de l'axe des z,
           tilt=50;
           autour de l'axe des x
run;
```


OUAGADOUGOU

Secteurs selon axe



IV.B

IV.B.3 Représentations graphiques de distributions simples : proc gchart

IV.B.3.1 Diagrammes en bâton : proc gchart options vbar ou hbar

initialisations

```
libname base '/usr/usr3/graph90/docu';
options nocenter nodate nonumber;
title 'HISTOGRAMME';
```

préparation d'un tableau de démonstration

```
data prov; set base.popul;
  if v4 ne 'absent'; var17=v17;
  keep ident var17 v17 v4;

proc means data=prov mean max min; var v17;
calcul des indices pour la réalisation des classes

proc sort data=prov; by var17;
proc rank data=prov out=prov groups=4; var var17;
tri du tableau, calcul des quartiles de la variable var17, et recodage de celle-ci en fonction des quartiles (codes 0, 1, 2, 3)
proc print data=prov (obs=10);

data resultat; set prov;
  length v17r $ 2;
  if var17=0 then v17r='1e';
  else
  if var17=1 then v17r='2e';
  else
  if var17=2 then v17r='3e';
  else
  if var17=3 then v17r='4e';
  label v17r='Quartiles age mere';
création de la variable v17r par recodage de la variable var17 (une classe=un quartile de v17)
proc print data=résultat (obs=10);
```

initialisations de l'environnement graphique

```
goptions device=sun border cback=white
  ctext=black ftext=swiss;
pattern1 v=11 c=black; pattern2 v=s c=gray00;
11 : code prédéfini de trame (cf. SAS/GRAPH User's Guide 6.03, pages 57 à 60)
title f=swiss h=1.25 'HISTOGRAMME';
legend1 value=(h=1.)
  label=(h=1. 'quantiles');
legend2 value=(h=1.)
  label=(h=0.75 'effectifs');
```

• • •

HISTOGRAMME

Analysis Variable : V17 age mere

N Obs	Minimum	Maximum	Mean
1130	15.0000000	47.0000000	25.7017699

HISTOGRAMME

OBS	V17	IDENT	V4	VAR17
1	15	3057	0	0
2	16	342	0	0
3	16	381	1	0
4	16	394	0	0
5	16	563	1	0
6	16	590	0	0
7	16	454	0	0
8	16	466	0	0
9	16	783	0	0
10	16	2074	0	0

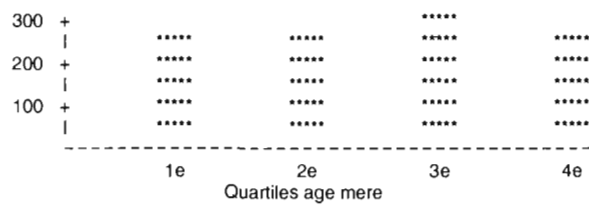
HISTOGRAMME

OBS	V17	IDENT	V4	VAR17	V17R
1	15	3057	0	0	1e
2	16	342	0	0	1e
3	16	381	1	0	1e
4	16	394	0	0	1e
5	16	563	1	0	1e
6	16	590	0	0	1e
7	16	454	0	0	1e
8	16	466	0	0	1e
9	16	783	0	0	1e
10	16	2074	0	0	1e

HISTOGRAMME

FREQUENCY OF V17R

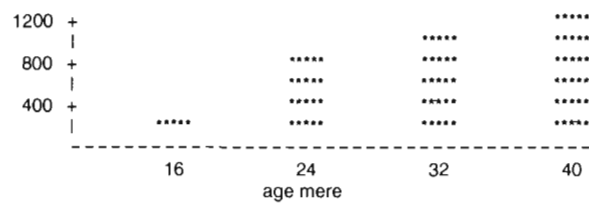
FREQUENCY



HISTOGRAMME

CUMULATIVE FREQUENCY OF V17

CUMULATIVE FREQUENCY



IV.B

• • •

```
legend3 value=(h=1 t=1 'non inondee' t=2 'inondee')
        les paramètres t=n 'chaîne' permettent de remplacer le libellé des classes de la légende par la chaîne
        de caractères désirée
        down=2
        label=(h=1. 'Parcelle');
        hauteur et titre de la légende
```

visualisation des résultats**dans la fenêtre output**

```
proc chart data=resultat;
  vbar v17r / discrete type=freq;
  diagramme des fréquences d'une variable discrète
  vbar v17 / levels=4 type=cfreq;
  levels=n discrétise de façon arbitraire la variable en n classes
  Autres possibilités : type=pct /sum /mean, ou cpct
  diagramme des fréquences cumulées d'une variable continue
```

sur le périphérique défini dans l'instruction goptions soit sur l'écran

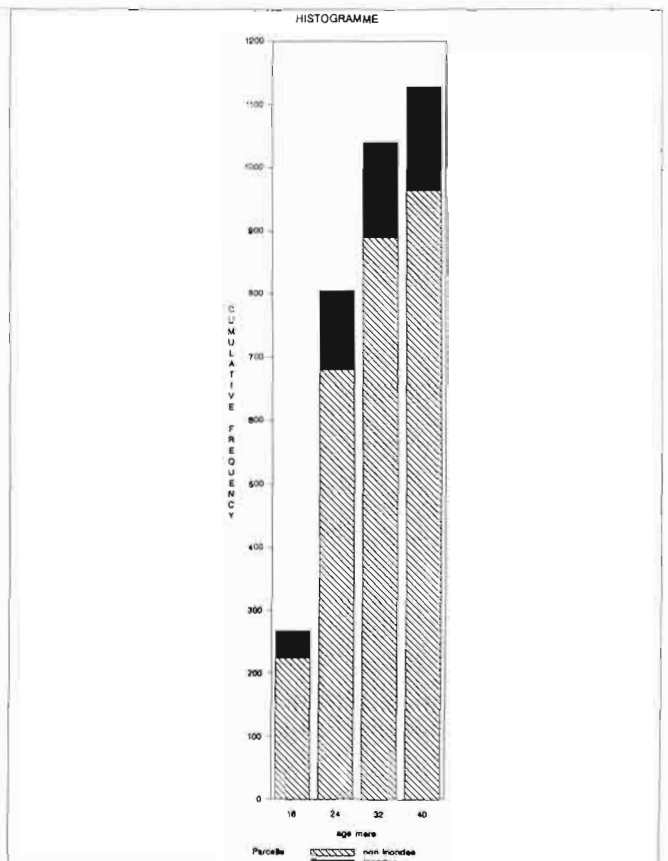
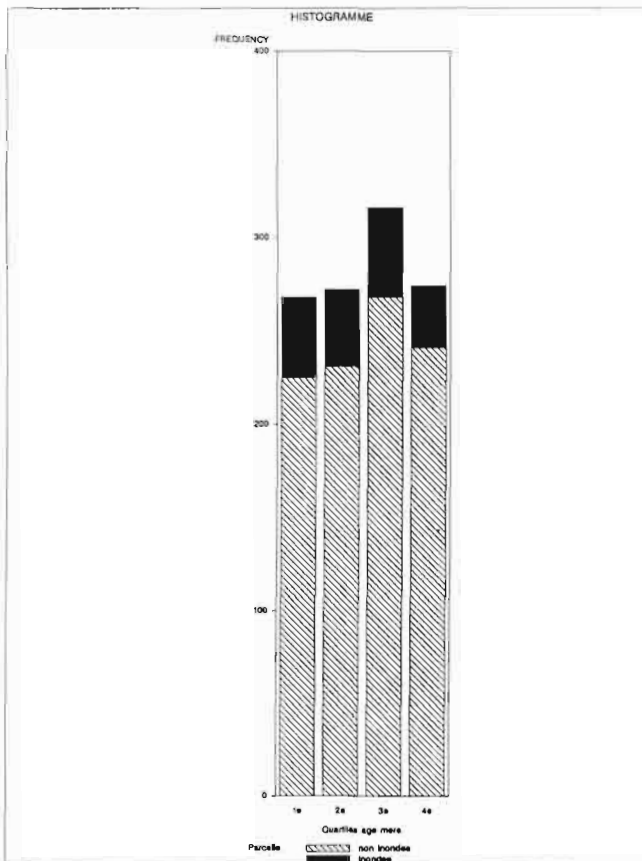
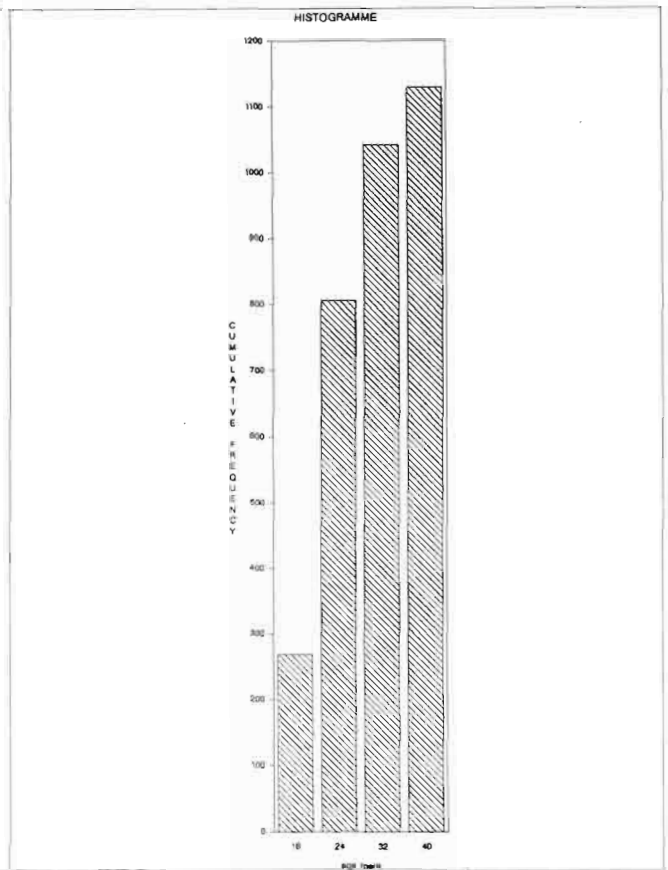
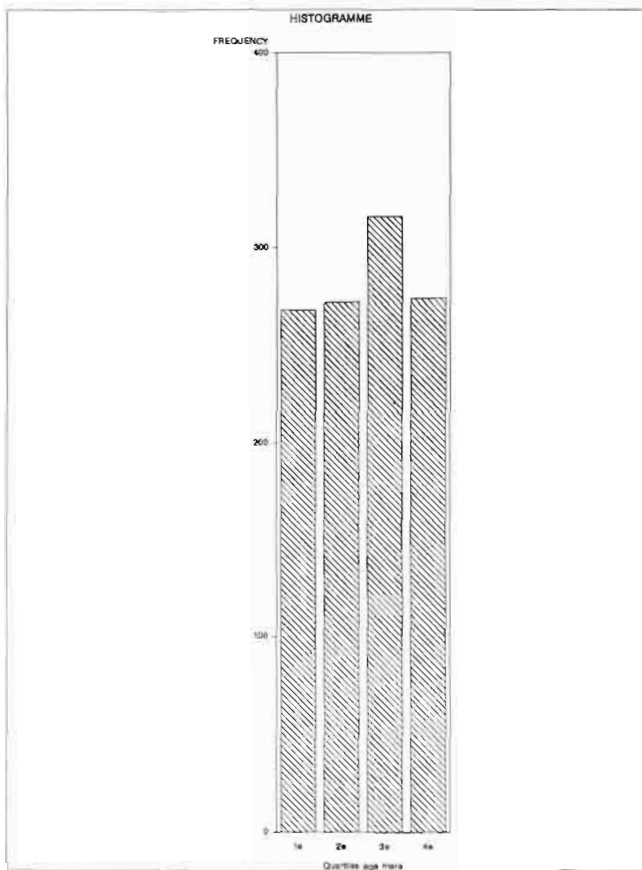
```
proc gchart data=resultat;
  vbar v17r / discrete type=freq
  ctext=black cfr=grayf8
  couleur des textes couleur de fond du diagramme
  coutline=black caxis=black
  couleur du tracé extérieur des bâtons, des axes
  legend=legend1;
  présente la légende selon les options définies dans legend1
  vbar v17 / levels=4 type=cfreq
  4 classes d'égale étendue
  coutline=black caxis=black
  ctext=black cfr=grayf8
  legend=legend2;
  présente la légende selon les options définies dans legend2
```

Représentation des variables v17r et v17 par un diagramme en bâtons

```
proc gchart data=resultat;
  vbar v17r / discrete subgroup=v4 type=freq
  coutline=black caxis=black
  ctext=black cfr=grayf8
  legend=legend3;
  présente la légende selon les options définies dans legend3
  Chaque bâton du diagramme représente une classe de la variable traitée v17r. Les subdivisions de chaque bâton
  représentent les modalités de la variable de l'option subgroup (dans l'exemple v4)
```

```
vbar v17 / levels=4 subgroup=v4 type=cfreq
  coutline=black caxis=black
  ctext=black cfr=grayf8 legend=legend3;
  Représentation des variables v17r et v17 par un diagramme en bâtons subdivisé en fonction des modalités de v4
```

```
run;
```



IV.B

IV.B.3.2 Diagrammes en « block » : proc gchart option block

```

libname base '/usr/usr3/graph90/docu';
options nocenter nodate ps=200 nonumber;
data prov; set base.popul;
  if v4 ne 'absent'; var17=v17;
  keep ident var17 v17 v4;
proc means data=prov mean max min; var v17;
proc sort data=prov; by var17;
proc rank data=prov out=prov groups=4; var v17;
proc print data=prov (obs=10);

```

```

data resultat; set prov; length v17r $ 2;
  if var17=0 then v17r='1Q';
  else
  if var17=1 then v17r='2Q';
  else
  if var17=2 then v17r='3Q';
  else
  if var17=3 then v17r='4Q';
  label v17r='Quartiles age mere';

```

création de la variable v17r par recodage de la variable var17 (une classe=un quartile de v17)

```
proc print data=resultat (obs=10);
```

initialisations de l'environnement graphique

```

goptions device=sun border cback=white
  ctext=black ftext=swiss;
pattern1 v=r5 c=black; pattern2 v=s c=black;
  r5 : code prédéfini de trame (cf. SAS/GRAPH User's Guide 6.03, pages 57 à 60)
title f=swiss h=1.25 'AGE DE LA MERE';
legend1 value=(h=1.)
  label=(h=1. 'quantiles');
legend3 value=(h=1. t=1 'non inondée' t=2 'inondée')
  down=2
  label=(h=1. 'Parcelle');

```

visualisation des résultats**dans la fenêtre output**

```

proc chart data=resultat;
  block v17r / discrete type=freq;
  un bloc par classe effectifs sur l'axe vertical
  block v17 / levels=4 subgroup=v4 type=freq;
  4 classes variable de subdivision des blocs

```

un bloc par classe de v17; les subdivisions de chaque bloc représentent les modalités de la variable de l'option subgroup

• • •

Analysis Variable : V17 age mere

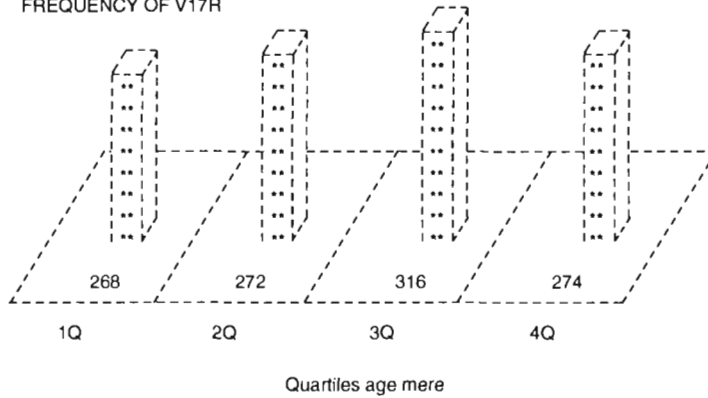
N Obs	Minimum	Maximum	Mean
1130	15.0000000	47.0000000	25.7017699

OBS	V17	IDENT	V4	VAR17
1	15	3057	0	0
2	16	342	0	0
3	16	381	1	0
4	16	394	0	0
5	16	563	1	0
6	16	590	0	0
7	16	454	0	0
8	16	466	0	0
9	16	783	0	0
10	16	2074	0	0

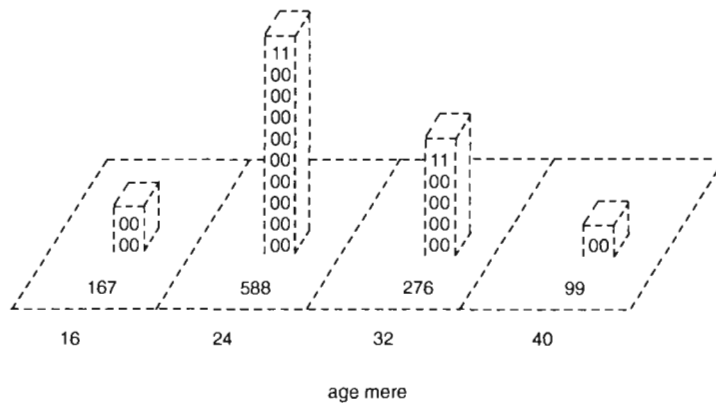
AGE DE LA MERE

OBS	V17	IDENT	V4	VAR17	V17R
1	15	3057	0	0	1Q
2	16	342	0	0	1Q
3	16	381	1	0	1Q
4	16	394	0	0	1Q
5	16	563	1	0	1Q
6	16	590	0	0	1Q
7	16	454	0	0	1Q
8	16	466	0	0	1Q
9	16	783	0	0	1Q
10	16	2074	0	0	1Q

AGE DE LA MERE
FREQUENCY OF V17R



AGE DE LA MERE
FREQUENCY OF V17



SYMBOL V4	0	1
SYMBOL V4	0	1

IV.B

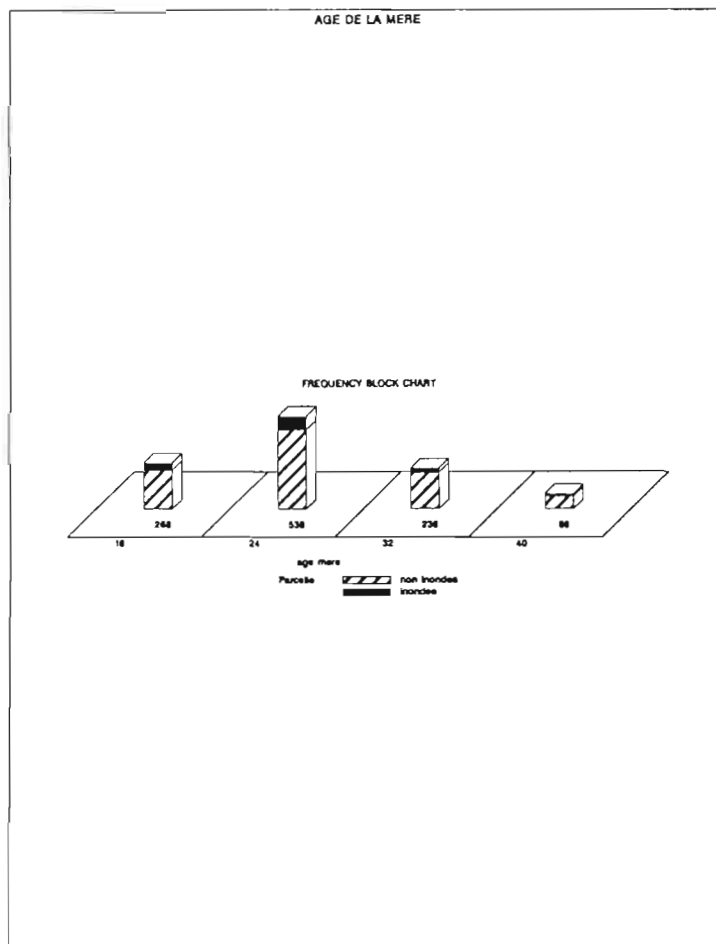
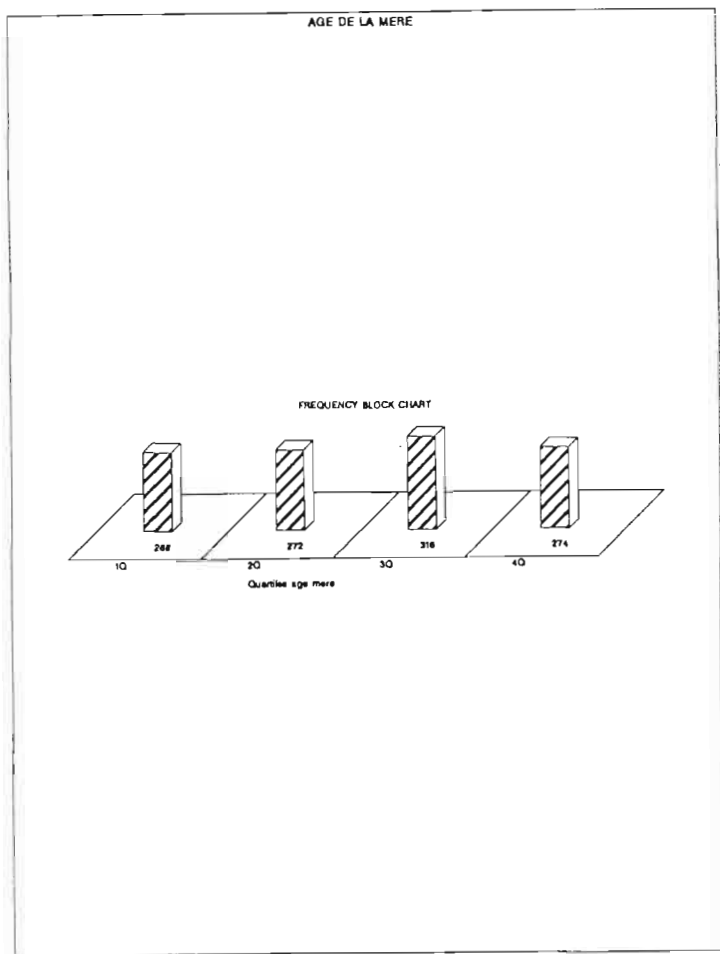
• • •

sur le périphérique défini dans l'instruction goptions soit sur l'écran

```
proc gchart data=resultat;
  block v17r / discrete type=freq
               coutline=black caxis=black
               couleur des contours des blocs, de l'axe
               ctext=black legend=legend1;
               couleur des textes légende selon les options de legend1
  block v17 / levels=4 subgroup=v4 type=freq
              coutline=black caxis=black
              ctext=black legend=legend3;
```

un bloc par classe; les subdivisions de chaque bloc représentent les modalités de la variable de l'option subgroup
Représentation des variables v17r et v17 par blocs

```
run;
```

IV.B

IV.B.3.3 Diagrammes en cercles : proc gchart option pie

```
libname base '/usr/usr3/graph90/docu';
options nocenter nodate nonumber;
title 'POIDS PAR AGE';
```

préparation d'un tableau de démonstration

```
data prov; set base.popul;
  if v4 ne 'absent'; var17=v17;
  keep ident var17 v17 v4;

proc means data=prov mean max min; var v17;
proc sort data=prov; by var17;
proc rank data=prov out=prov groups=4; var v17;

data resultat; set prov;
  length v17r $ 2;
  if var17=0 then v17r='1Q';
  else
  if var17=1 then v17r='2Q';
  else
  if var17=2 then v17r='3Q';
  else
  if var17=3 then v17r='4Q';
  label v17r='Quartiles age mere';
```

création de la variable v17r par recodage de la variable var17 (une classe=un quartile de v17)

initialisations de l'environnement graphique

```
goptions device=sun border cback=white
  ctext=black ftext=swiss;
goptions colors=(grayf8 gray90 gray40 gray00 );
réinitialisation de la table des couleurs
title f=swiss h=1.25 'AGE DE LA MERE';
```

visualisation des résultats

```
proc gchart data=resultat;
  pie v17r / discrete type=freq
  valeurs absolues sur l'axe vertical
  group=v4
  un cercle par modalité de v4. Les subdivisions de chaque cercle représentent
  les modalités de la variable v17r.
  explode='1Q' '2Q'
  les portions de cercle représentant les valeurs 1Q et 2Q seront excentrées
  percent=inside
  écriture, à l'intérieur du cercle, du pourcentage représenté par chaque classe
  slice=outside
  affichage, à l'extérieur du cercle, du nom de chaque modalité
```

• • •

Analysis Variable : V17 age mere

N Obs	Minimum	Maximum	Mean
1130	15.0000000	47.0000000	25.7017699

IV.B

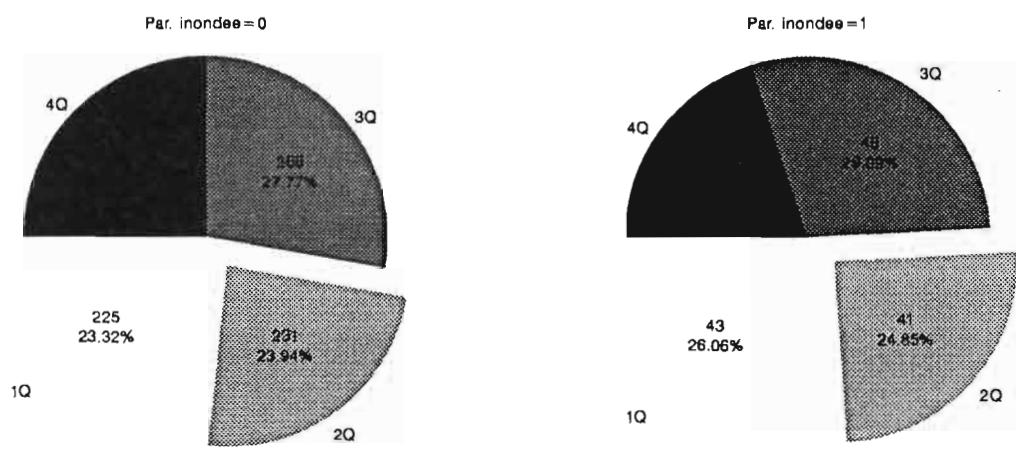
• • •

```
value=inside
affichage, à l'intérieur du cercle, de la valeur de la modalité représentée
angle=180
angle définissant l'origine sur le cercle de la première classe
fill=solid
choix de la représentation
fill=X : chaque classe sera représentée par une trame croisée définie par pattern
cfill=black : exécuter le tracé du contour de chaque quartier en noir
ctext=black
couleurs des textes
across=2;
nombre de cercles à représenter sur l'axe horizontal par page graphique
down=n : nombre de cercles à représenter sur l'axe vertical
```

run;

- Résultat : deux cercles parce que la variable v4 possède deux modalités. Chaque cercle représente pour une modalité de la variable v4, la répartition des quatre modalités de la variable v17r.
- Dans ce programme, il n'y a pas de déclaration de **pattern** pour définir la représentation des classes. Par défaut, SAS affecte alors aux classes les valeurs contenues dans la table interne des couleurs. Cette table est ici réinitialisée par l'option **colors** de l'instruction **goptions**.

AGE DE LA MERE
FREQUENCY of V17R



IV.B

IV.B.4 Représentation graphique des distributions simples : proc gplot

Une distribution simple : à une valeur sur l'axe horizontal correspond une seule observation.

IV.B.4.1 Distributions simples à deux variables

```
libname    base          '/usr/usr3/graph90/docu';
options   nocenter      nodate          nonumber;
title1    'POIDS PAR AGE';
```

création d'un tableau de démonstration : un seul individu par x

```
proc sort  data=base.popul    out=prov1;  by    v18;
data prov2; set    prov1;
    if    v18 ne .;          if    v22 ne .;
    if    v23 ne .;          if    v18 ne 0;
    keep  ident v18 v23 v22;
```

suppression de certaines observations et sélection de variables

```
proc summary  data=prov2  nway; class  v18;  var    v23 v22;
                                output out=prov2  mean(v23 v22)= ;
```

calcul de la moyenne par âge (v18) des variables poids actuel (v23) et poids à 0 jour (v22)

```
proc print    data=prov2;
proc contents  data=prov2;  run;
```

initialisations du périphérique de sortie

```
goptions  reset=all;
réinitialisation de l'environnement graphique
goptions  device=sun  border          cback=white          ctext=black
          ftext=swiss  colors=(green);
```

• • •

POIDS PAR AGE

OBS	V18	_TYPE_	_FREQ_	V23	V22
1	4	1	90	5988.17	3030.42
2	5	1	59	6764.24	3172.54
3	6	1	60	6858.00	3078.83
4	7	1	49	7475.10	3081.02
5	8	1	43	7933.26	3256.05
6	9	1	57	7372.72	3045.11
7	10	1	40	8074.88	3149.25
8	11	1	39	7990.26	3082.18
9	12	1	99	8391.21	3158.05
10	13	1	44	8597.27	3071.36
11	14	1	36	8353.61	3316.39
12	15	1	47	8286.38	3112.32
13	16	1	27	7962.78	3140.74
14	17	1	37	8288.65	3036.76
15	18	1	79	8521.49	2925.13
16	19	1	24	8781.67	3145.42
17	20	1	20	8087.00	3101.50

POIDS PAR AGE

OBS	V18	_TYPE_	_FREQ_	V23	V22
18	21	1	9	8935.56	2801.33
19	22	1	13	8918.46	2992.31
20	23	1	16	8582.50	2882.50
21	24	1	242	8584.98	3051.06

POIDS PAR AGE

CONTENTS PROCEDURE

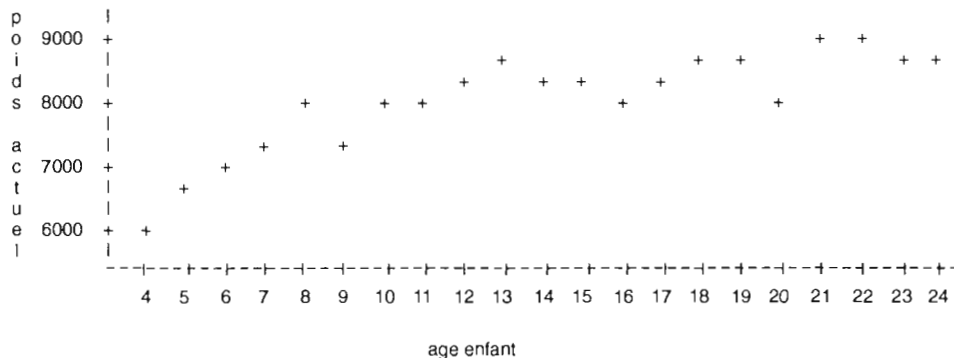
Data Set Name: WORK.PROV2 Type:
 Observations: 21 Record Len: 44
 Variables: 5
 Label:

-----Alphabetic List of Variables and Attributes-----

#	Variable	Type	Len	Pos	Label
1	V18	Num	8	4	age enfant
5	V22	Num	8	36	poids 0 jour
4	V23	Num	8	28	poids actuel
3	_FREQ_	Num	8	20	
2	_TYPE_	Num	8	12	

POIDS ACTUEL PAR AGE
 POINTS ET COURBE DE CONNECTION

Plot of V23*V18. Symbol used is '+'.
 (Note: The plot actually shows V23 vs V18)



IV.B

• • •

1^{er} graphique : tracé des points avec courbe de connection**visualisation dans la fenêtre output**

```

title1      'POIDS ACTUEL PAR AGE';
proc plot   data=prov2; plot   v23*v18='+';

```

représentation des valeurs par le symbole '+'

visualisation sur l'écran

```

title2      'POINTS ET COURBE DE CONNECTION';
axis1       value=(c=red) order=0 to 28 by 2   minor=(n=1)
            value=(c=color h=n f=font 'text') : couleur, hauteur, fonte du texte à associer à l'axe
            order=x to y by z : bornes et « pas » de la graduation de l'axe
            Autres possibilités : order=1 10 100 1000
            order='mars' 'avril' 'mai' 'juin'
            major=(n=n h=n c=color) : nombre, hauteur et couleur des graduations principales
            minor=(n=n h=n c=color) : nombre, hauteur et couleur des graduations intermédiaires
            label=('AGE DES ENFANTS EN MOIS');
            label=(h=n c=color 'texte') : hauteur, couleur du texte à associer à l'axe

axis2       value=(c=blue)      order=0 to 18000 by 2000   minor=(n=1)
            label=('POIDS MOYENS EN GRAMMES')   c=black;

proc gplot   data=prov2;
  symbol1    c=black          v=dot
            couleur          symbole : toutes les lettres de 'A' à 'W', les chiffres de 1 à 9 ou des
                               symboles spéciaux (plus, star, square, triangle, point, dot, circle)

            l=1
            type de ligne : continue si 1, pointillée si 2, tiretés si 3, 4, 5 ou 6, alternance de tiretés
                               et de points si 8, 9, 10 ou 11, etc.
            i=j;
            interpolation pour la jonction des points : simple si j, en marche d'escalier si step; le point
            est à gauche de la marche si step1, au centre si stepc, à droite si stepr, au centre et les
            marches sont reliées par des traits verticaux si stepcj.

  paramètres du symbole à associer à la représentation
  plot       v23*v18=1 /
            numéro du symbole à associer à la représentation

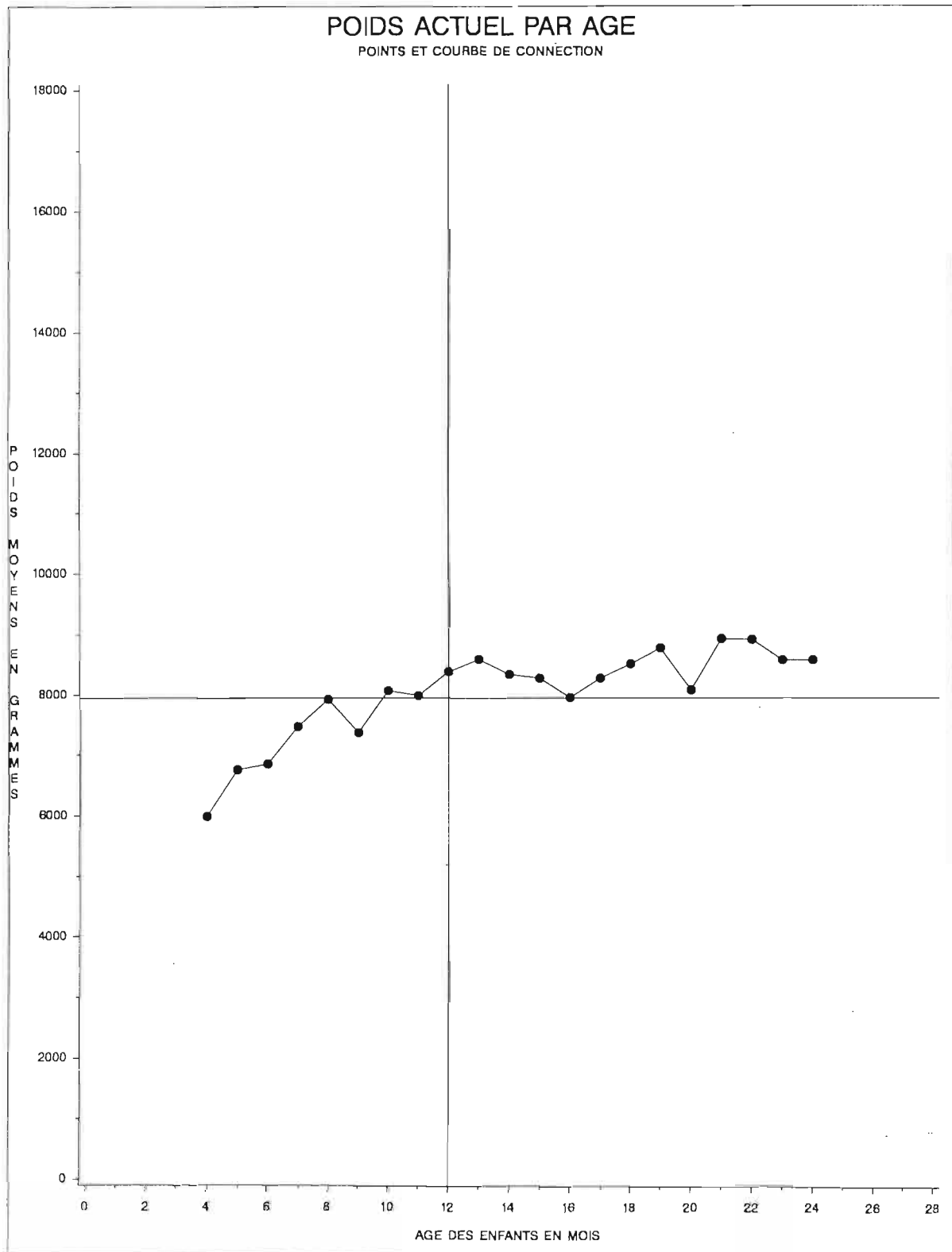
            haxis=axis1   vaxis=axis2
            présenter l'axe horizontal selon les options définies dans axis1 et l'axe vertical selon
            les options de axis2

            href=12
            tracé d'une ligne horizontale pour y=12 (âge moyen)
            vref=7950
            tracé d'une ligne verticale pour x=7950 (poids moyen)
            caxis=black;
            couleur des axes

```

run;

• • •



IV.B

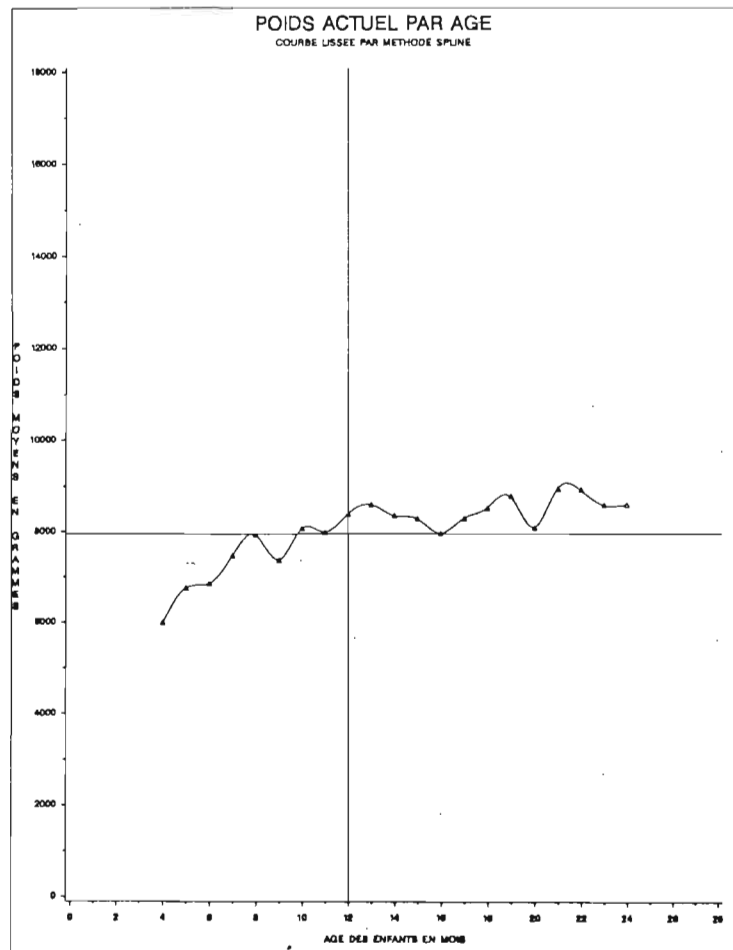
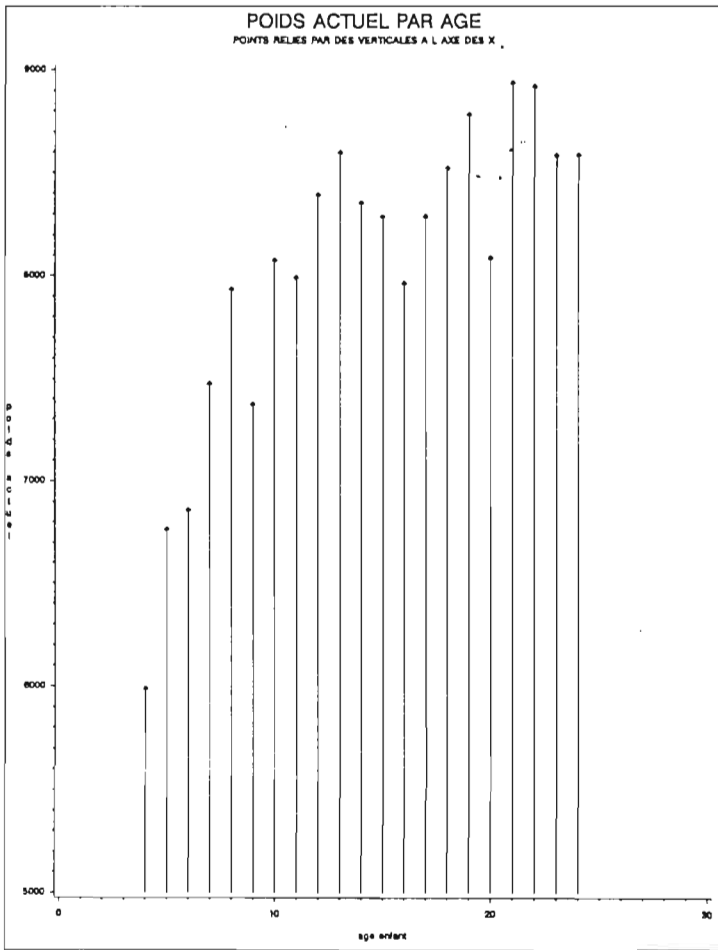
• • •

2^e graphique : les points sont reliés à l'axe des x

```
proc gplot      data=prov2;
  title2       'POINTS RELIES PAR DES VERTICALES A L'AXE DES X';
  symbol1      i=needle
               chaque point est relié à l'axe horizontal par une droite
               v=diamond  c=black    l=1;
               symbole      couleur      type de ligne
  plot         v23*v18=1;
run;
```

3^e graphique : les points sont reliés par une courbe lissée

```
proc gplot      data=prov2;
  title2       'COURBE LISSEE PAR METHODE SPLINE';
  symbol1      i=spline
               les points sont reliés par une courbe lissée selon la méthode cubique
               Autres possibilités : i=splinep, méthode spline paramétrique
               i=11 ou 13 ou 15, méthode d'interpolation Lagrange
               i=11p ou 13p ou 15pc, méthode Lagrange paramétrique
               v=triangle  c=black    l=1;
  plot         v23*v18=1 / haxis=axis1  vaxis=axis2
                          vref=7950   href=12    caxis=black;
run;
```



IV.B

IV.B.4.2 Distributions simples à trois variables

```
libname base '/usr/usr3/graph90/docu';
options nocenter nodate ps=30;
title1 'DISTRIBUTIONS SIMPLES A TROIS VARIABLES';
```

1^{er} graphique : courbe (la troisième variable est une variable discrète)

création d'un tableau de démonstration

```
data prov1; set base.popul;
              if v19 ne 'absent'; keep v18 v19 v23;
proc sort data=prov1 out=prov3; by v18;
proc summary data=prov3; class v19; by v18;
              output out=prov3 mean(v23)= ;
```

calcul de la moyenne de la variable poids actuel (v23) par sexe (v19) et par âge (v18)

```
data prov3; set prov3; if _TYPE_ ne 0; run;
```

initialisations de l'environnement graphique

```
goptions reset=all;
goptions device=sun border cback=white ctext=black ftext=swiss;
title2 'POIDS MOYENS ACTUELS PAR AGE';
title3 'GRAPHIQUE PAR SEXE AVEC COURBE DE CONNECTION';
axis1 value=(c=red) order=0 to 28 by 2 minor=(n=1)
      label=('AGE DES ENFANTS EN MOIS');
axis2 value=(c=blue) order=4000 to 12000 by 2000
      minor=(n=1) c=black
      label=('POIDS MOYENS EN GRAMMES');
symbol1 c=black h=.4 cm v=star i=spline;
symbol2 c=black v=dot i=spline;
```

visualisation

dans la fenêtre output

```
proc plot data=prov3; plot v23*v18=v19;
```

sur l'écran

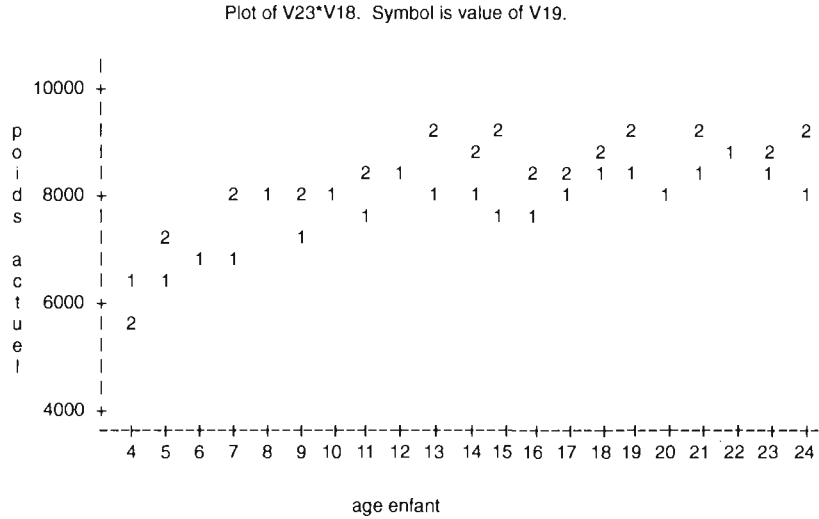
```
proc gplot data=prov3;
  plot v23*v18=v19 / haxis=axis1 vaxis=axis2
                    href=12 vref=7950
                    caxis=black;
```

un symbole et une courbe par modalité de la variable v19. Les symboles sont positionnés à l'intersection des valeurs des variables v23 et v18.

```
run;
```

...

Distributions simples a trois variables
 POIDS MOYENS ACTUELS PAR AGE
 GRAPHIQUE PAR SEXE AVEC COURBE DE CONNECTION



NOTE: 6 obs hidden.

Distributions simples a trois variables
 POIDS MOYENS ACTUELS PAR AGE
 REPRESENTATION PAR CERCLES PROPORTIONNELS DES EFFECTIFS ENQUETES

OBS	V18	_FREQ_	V23	V22
1	4	90	5988.17	3030.42
2	5	59	6764.24	3172.54
3	6	60	6858.00	3078.83
4	7	49	7475.10	3081.02
5	8	43	7933.26	3256.05
6	9	57	7372.72	3045.11
7	10	40	8074.88	3149.25
8	11	39	7990.26	3082.18
9	12	99	8391.21	3158.05
10	13	44	8597.27	3071.36
11	14	36	8353.61	3316.39
12	15	47	8286.38	3112.32
13	16	27	7962.78	3140.74
14	17	37	8288.65	3036.76
15	18	79	8521.49	2925.13
16	19	24	8781.67	3145.42
17	20	20	8087.00	3101.50
18	21	9	8935.56	2801.33
19	22	13	8918.46	2992.31
20	23	16	8582.50	2882.50
21	24	242	8584.98	3051.06

IV.B

• • •

**2^e graphique : cercles proportionnels (la troisième variable est une variable continue
création d'un tableau de démonstration**

```
proc sort      data=base.popul      out=prov1;  by    v18;
data prov2;   set    prov1;         keep   ident v18 v23 v22;
proc summary  data=prov2;          class  v18;  var   v23 v22;
                                output out=prov2  mean(v23 v22)= ;
```

calcul de la moyenne par âge des variables v23 (poids actuel) et v22 (poids à 0 jour)

```
data prov2; set    prov2;
              if    v18 ne .;      if    v22 ne .;
              if    v23 ne .;      if    v18 ne 0;
              drop  _TYPE_ ;
proc print   data=prov2;
```

initialisations de l'environnement graphique

```
title2      h=1 cm      'POIDS MOYENS ACTUELS PAR AGE';
title3      h=0.5 cm   'REPRESENTATION PAR CERCLES PROPORTIONNELS
DES EFFECTIFS ENQUETES';
axis1       value=(c=black)  order=0 to 28 by 2  minor=(n=1)
              label=('AGE DES ENFANTS EN MOIS');
axis2       value=(c=black)  order=0 to 18000 by 2000
              minor=(n=1)    c=black;
              label=('POIDS MOYENS EN GRAMMES');
```

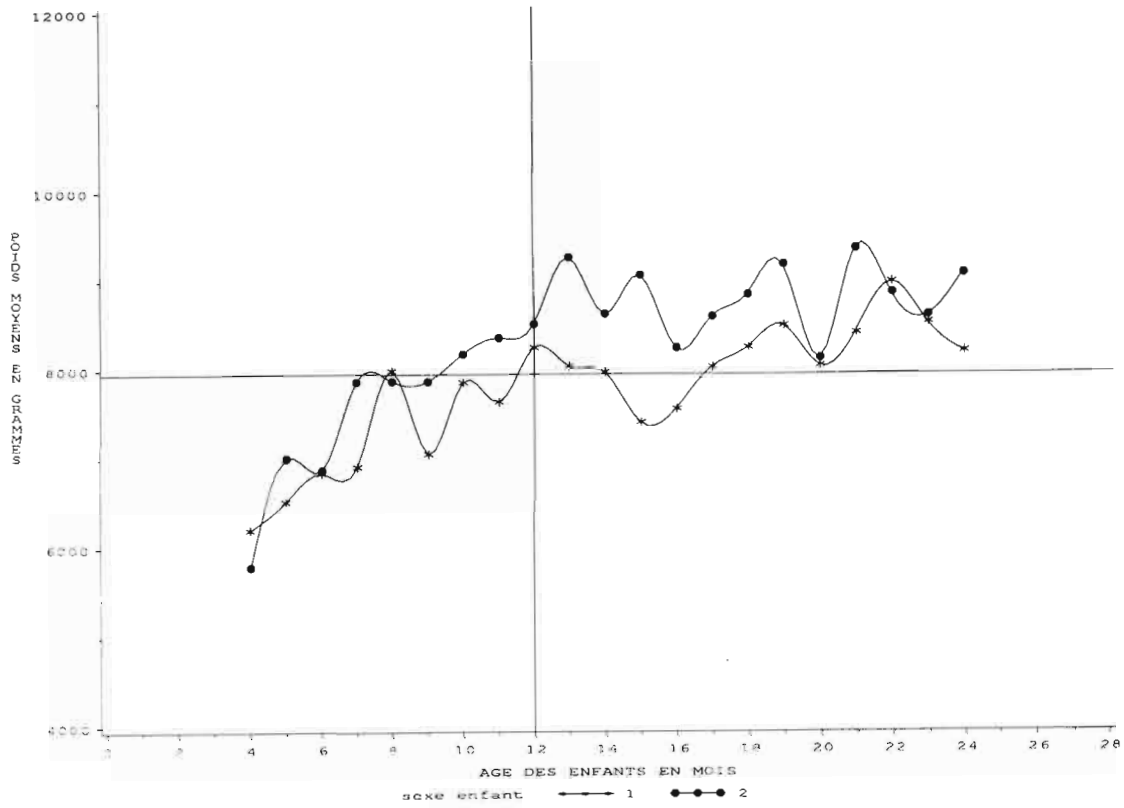
visualisation

```
proc gplot  data=prov2;
              bubble v23*v18=_FREQ_
                  /   bcolor=black      bsize=50
                    couleur du contour  taille maximum
                    bscale=radius
                    le rayon est proportionnel à la valeur de la variable; l'aire est l'option par défaut.
                    haxis=axis1  vaxis=axis2  caxis=black
                    href=12      vref=7950;
```

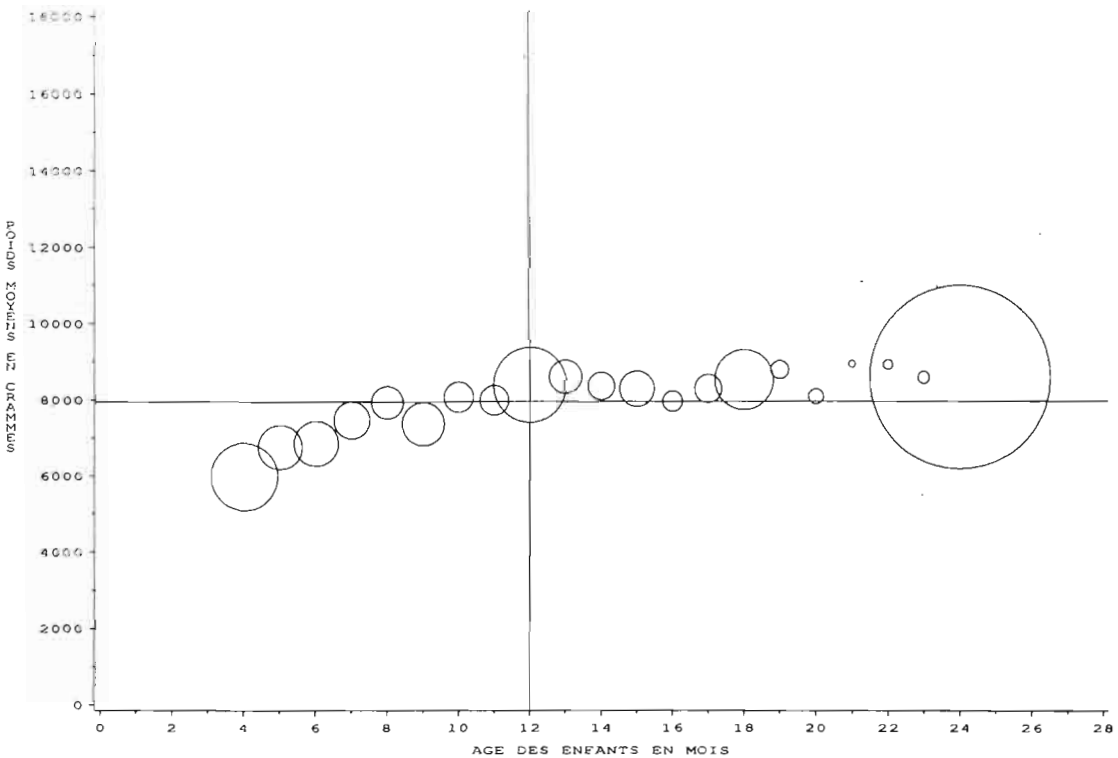
Les cercles sont positionnés à l'intersection des valeurs des variables v23 (poids moyen à la naissance) et v18 (âge actuel). La taille du cercle est proportionnelle au nombre d'individus de cet âge.

```
run;
```

Distributions simples a trois variables
 POIDS MOYENS ACTUELS PAR AGE
 GRAPHIQUE PAR SEXE AVEC COURBE DE CONNECTION



Distributions simples a trois variables
 POIDS MOYENS ACTUELS PAR AGE
 REPRESENTATION PAR CERCLES PROPORTIONNELS DES EFFECTIFS ENQUETES



IV.B

IV.B.4.3 Représentation de distributions simples par des aires

```
libname base '/usr/usr3/graph90/docu';
options nocenter nodate nonumber ps=30;
title1 'RAPPORT POIDS A LA NAISSANCE ET AGE ACTUEL';
goptions reset=all;
goptions device=sun border cback=white ctext=black ftext=swiss;
```

création d'un tableau de démonstration

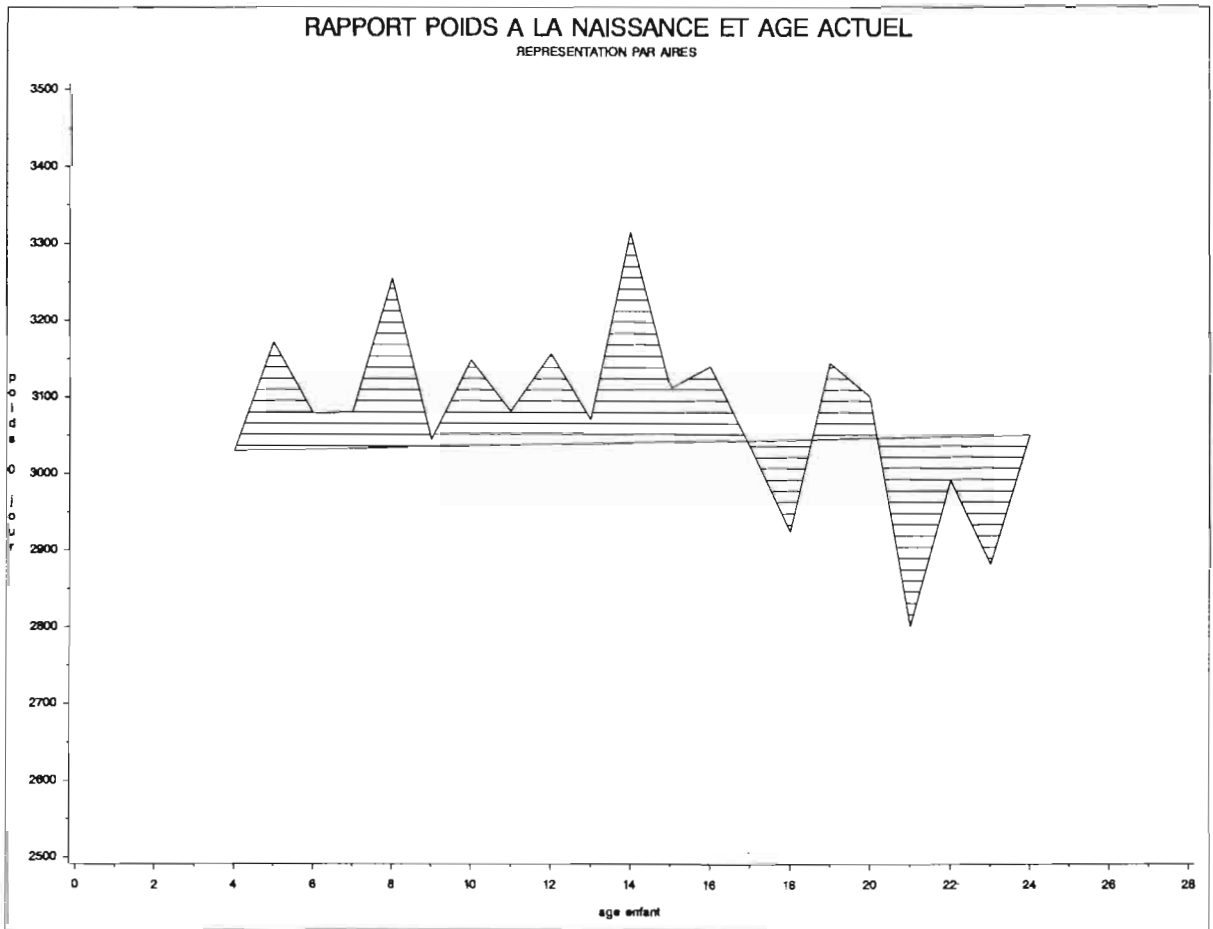
```
proc sort data=base.popul out=prov1; by v18;
data prov2; set prov1;
if v18 ne .; if v22 ne .; if v18 ne 0;
keep ident v18 v22;
proc summary data=prov2; class v18 nway; var v22;
output out=prov2 mean(v22)= ;
```

calcul de la moyenne par âge (v18) de la variable poids à 0 jour (v22)

visualisation

```
proc gplot data=prov2;
title2 'REPRESENTATION PAR AIRES';
symbol1 i=mr2
choix de l'aire
l=1 c=black;
type et couleur de la ligne
initialisations de l'environnement du graphique
plot v22*v18=1 /
haxis=0 to 28 by 2 vaxis=2500 to 3500 by 100
bornes et graduations sur les axes
hminor=1 vminor=1 caxis=black;
« pas » de la graduation intermédiaire des axes

run;
```

IV.B

IV.B.4.4 Représentation de deux distributions simples sur un même graphique

```
libname base '/usr/usr3/graph90/docu';
options nocenter nodate nonumber ps=30;
goptions reset=all;
goptions device=sun border cback=white ctext=black ftext=swiss;
title1 'RAPPORT POIDS A LA NAISSANCE ET AGE ACTUEL';
```

création d'un tableau de démonstration

```
proc sort data=base.popul out=prov1; by v18;
data prov2; set prov1;
if v18 ne .; if v22 ne .;
if v23 ne .; if v18 ne 0;
keep ident v18 v23 v22;
proc summary data=prov2; class v18 nway; var v23 v22;
output out=prov2 mean(v23 v22)= ;
```

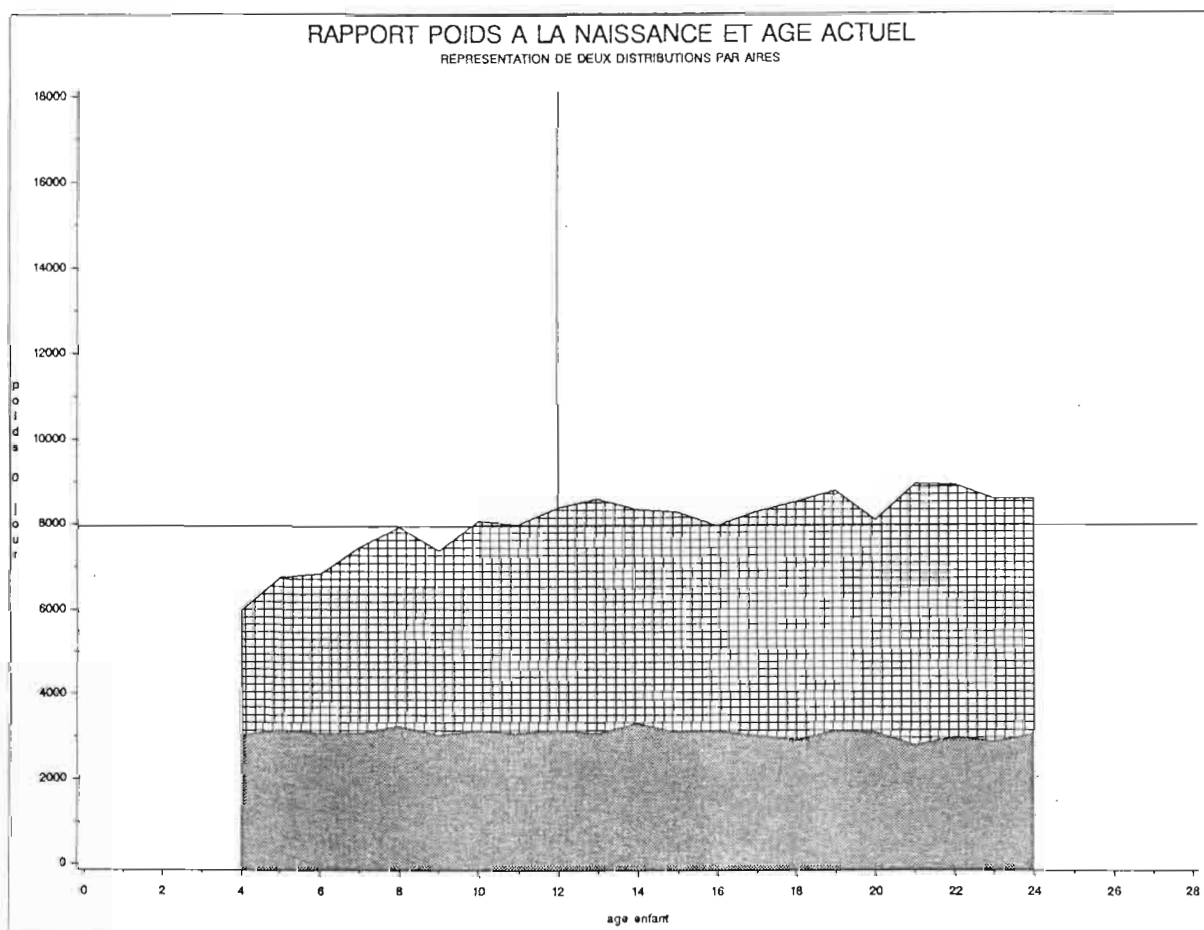
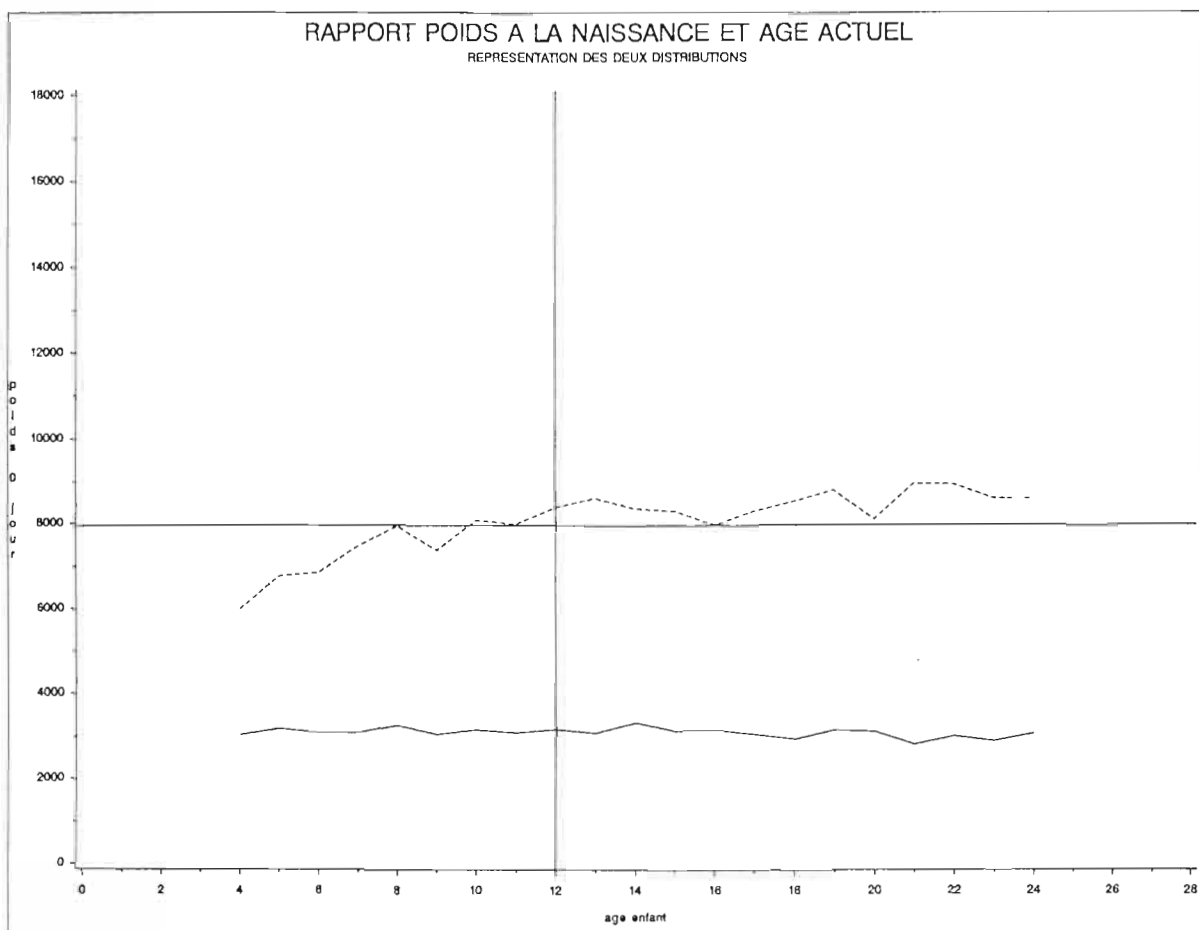
1^{er} graphique : courbe

```
proc gplot data=prov2;
title2 'REPRESENTATION DE DEUX DISTRIBUTIONS';
symbol1 v=none c=black l=1 i=j;
symbol2 v=none c=black l=2 i=j;
plot (v22 v23)*v18 /
overlay
tracé de plusieurs distributions sur le même graphique
haxis=0 to 28 by 2 vaxis=0 to 18000 by 2000
hminor=1 vminor=1 caxis=black
href=12 vref=7950; run;
```

deux courbes sur un même graphique : l'une représente la variable v22 (poids moyen à 0 jour) et l'autre la variable v23 (poids moyen actuel); les deux variables sont traitées en fonction de l'âge (v18).

2^e graphique : aire

```
proc gplot data=prov2;
title2 'REPRESENTATION DE DEUX DISTRIBUTIONS PAR AIRES';
symbol1 v=none c=black l=1 i=j;
symbol2 v=none c=black l=2 i=j;
pattern1 c=gray90 v=s; pattern2 c=black v=m3x;
plot (v22 v23)*v18 /
overlay areas=2
nombre d'aires. La première aura pour représentation celle du premier pattern et la deuxième, celle du second.
haxis=0 to 28 by 2 vaxis=0 to 18000 by 2000
hminor=1 vminor=1 caxis=black
href=12 vref=7950; run;
```



IV.B

Nuage de points : à une valeur sur l'axe horizontal correspond une ou plusieurs observations.

IV.B.4.5 Représentation graphique d'un nuage de points

```
libname base '/usr/usr3/graph90/docu';
options nocenter nodate nonumber;
proc means data=base.popul mean max min; var v18 v23;
calcul d'indices de base
```

initialisations de l'environnement graphique

```
goptions reset=all;
goptions device=sun border cback=white
          ctext=black ftext=swiss colors=(green);
title1 'POIDS PAR AGE'; title2 'NUAGE DE POINTS';
```

visualisation

dans la fenêtre output

```
proc plot data=base.popul;
  plot v23*v18='+';
  symboles possibles : toutes les lettres de 'A' à 'W', les chiffres de 1 à 9 et les symboles spéciaux
```

sur l'écran

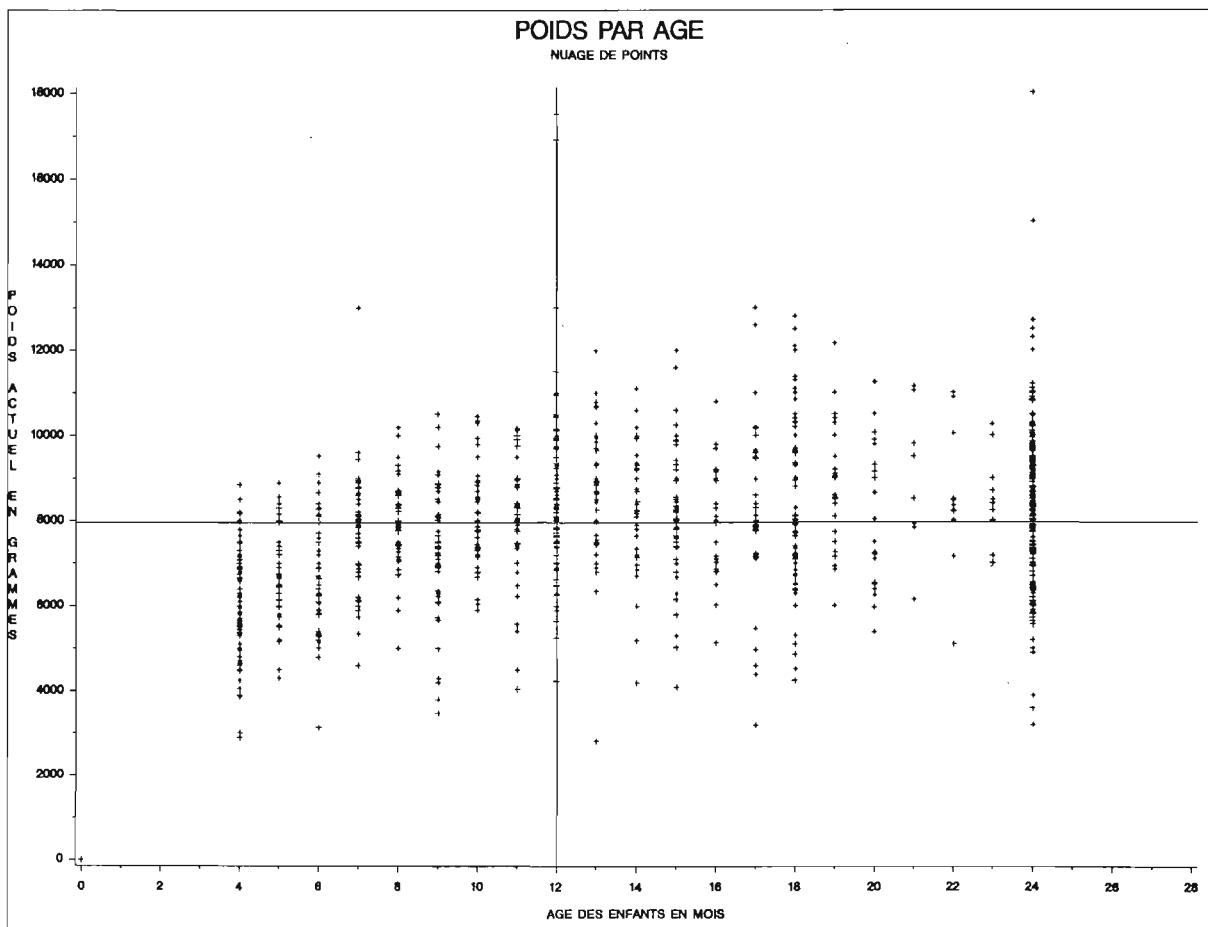
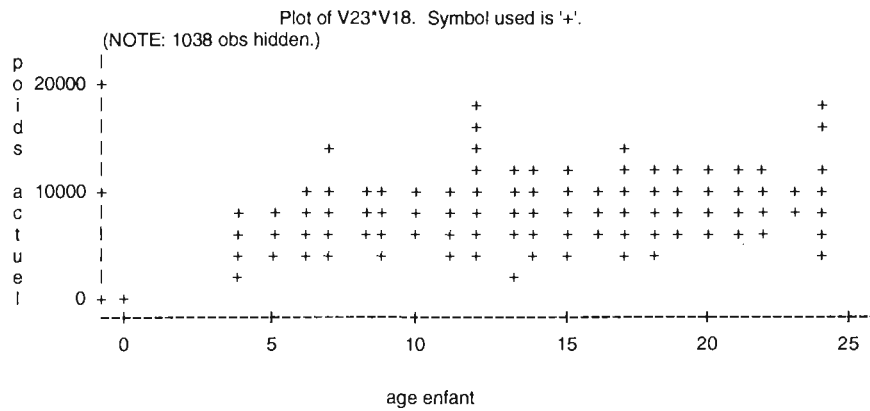
```
proc gplot data=base.popul;
  axis1 value=(c=red) order=0 to 28 by 2
        minor=(n=1) label=('AGE DES ENFANTS EN MOIS');
  axis2 value=(c=blue) order=0 to 18000 by 2000
        minor=(n=1) label=('POIDS ACTUEL EN GRAMMES');
  symbol1 c=red v=plus;
  initialisations de l'environnement du graphique
  plot v23*v18=1 /
        haxis=axis1 vaxis=axis2 caxis=black
        href=12
        tracé d'une ligne horizontale pour y=12 (âge moyen)
        vref=7950
        tracé d'une ligne verticale pour x=7950 (poids moyen)
```

```
run;
```

POIDS PAR AGE
NUAGE DE POINTS

N Obs	Variable	Label	Minimum	Maximum	Mean
1131	V18	age enfant	0	24.0000000	14.1494253
	V23	poids actuel	0	16000.00	7948.36

POIDS PAR AGE
NUAGE DE POINTS



IV.B

IV.B.4.6 Calcul et représentation graphique de courbes associées

```

libname      base          '/usr/usr3/graph90/docu';
options      nocenter      nodate          nonumber;
goptions     reset=all;
goptions     device=sun     border         cback=white
            ctext=black    ftext=swiss;
title        'POIDS ACTUEL PAR AGE';
axis1        value=(c=black)  order=0 to 28 by 2      minor=(n=1)
            label=('AGE DES ENFANTS EN MOIS');
axis2        value=(c=black)  order=0 to 18000 by 2000  minor=(n=1)
            label=('POIDS MOYENS EN GRAMMES')  c=black;

```

1^{er} graphique : courbe de la moyenne et valeurs à deux écarts-type

```

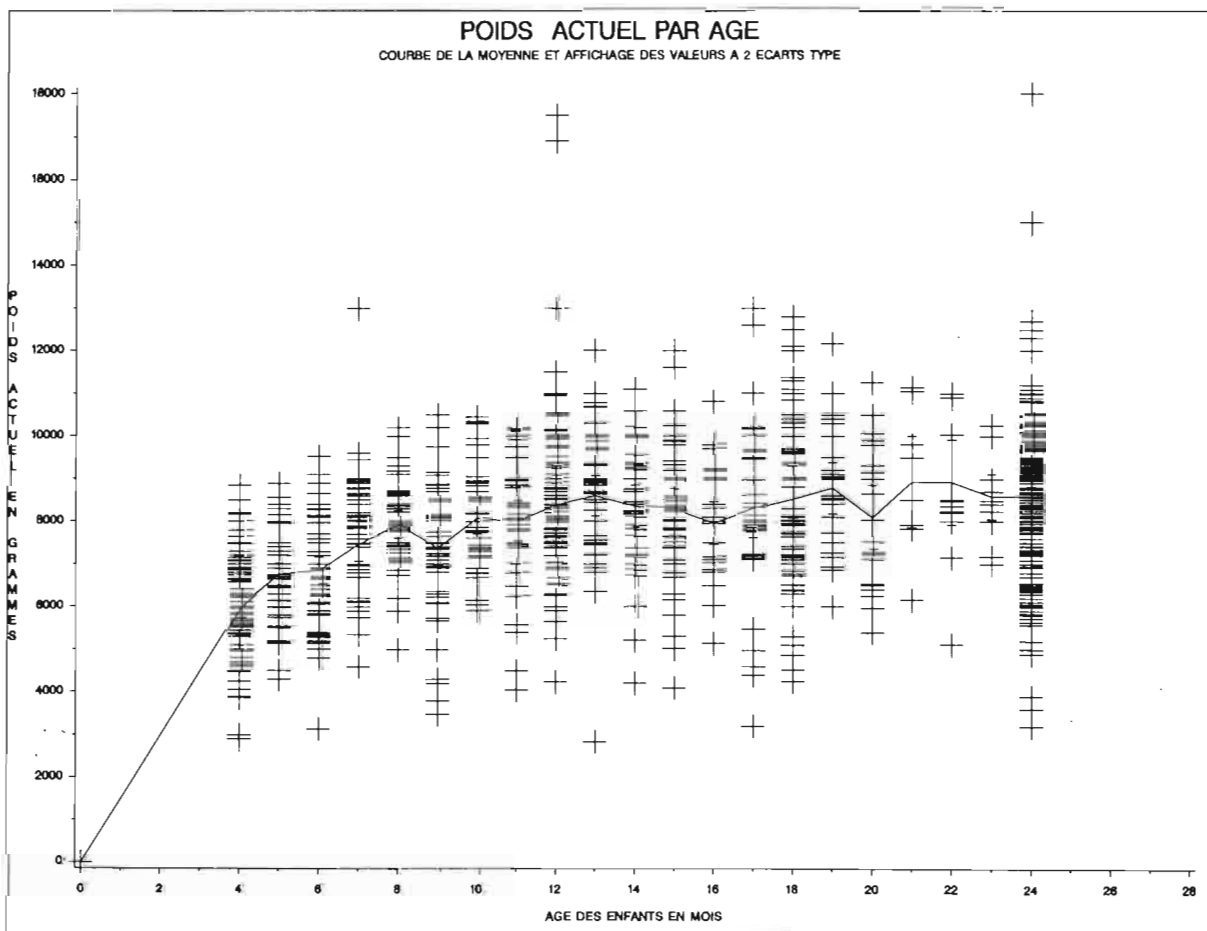
proc gplot  data=base.popul;
title2      'COURBE DE LA MOYENNE ET AFFICHAGE DES VALEURS A 2 ECARTS
TYPE';
symbol1     c=black        h=1 cm         v=plus          l=1
            signe plus, en noir, d'une hauteur égale à 1 cm      choix de la ligne
            i=std2mjt;
            i=stdk : représentation de la valeur y moyenne et des bornes de l'intervalle (+k,-k) écarts-
            type pour chaque valeur de x. Par défaut k=2.
            j : tracé d'une ligne de jonction entre les valeurs de la moyenne
            m : erreur standard à la moyenne
            t : tracé d'un sommet à chaque extrémité de l'intervalle d'erreur
            Autres possibilités p : 'sample variances'
            b : tracé d'une droite entre les extrémités de l'intervalle et la moyenne
initialisations de l'environnement du graphique
plot        v23*v18      /
            haxis=axis1  vaxis=axis2      caxis=black;
Représentation du poids actuel (v23) en fonction de l'âge (v18)
run;

```

• • •

POIDS ACTUEL PAR AGE
 COURBE DE LA MOYENNE ET AFFICHAGE DES VALEURS A 2 ECARTS TYPE

N Obs	Variable	Label	Minimum	Maximum	Mean
1131	V18	age enfant	0	24.0000000	14.1494253
	V23	poids actuel	0	18000.00	7948.36
	V22	poids 0 jour	0	4750.00	3079.74



IV.B

• • •

2^e graphique : courbe de la moyenne, minimum et maximum

```
proc gplot data=base.popul;
  title2 'COURBE DE LA MOYENNE, MINIMUM ET MAXIMUM DES VALEURS';
  symbol1 v=point c=black l=1
          point en noir type de ligne
          i=hilopt;
          tracé pour chaque valeur de x d'une ligne joignant la moyenne du min et du max des y
          j : tracé d'une ligne de jonction entre les moyennes
          t : tracé d'un sommet à chaque extrémité de l'intervalle d'erreur
          b : tracé d'une droite entre les extrémités de l'intervalle et la moyenne
          Si i=hiloc (high,low,close) : tiret de part et d'autre de la moyenne
  symbol2 v=star c=red;
          étoile rouge
  plot v23*v18=1 v23*v18=2 /
       nuage de points courbe des moyennes
       overlay
       plusieurs tracés sur le même graphique
       haxis=axis1 vaxis=axis2 caxis=black;
```

Représentation du poids actuel (v23) en fonction de l'âge (v18)

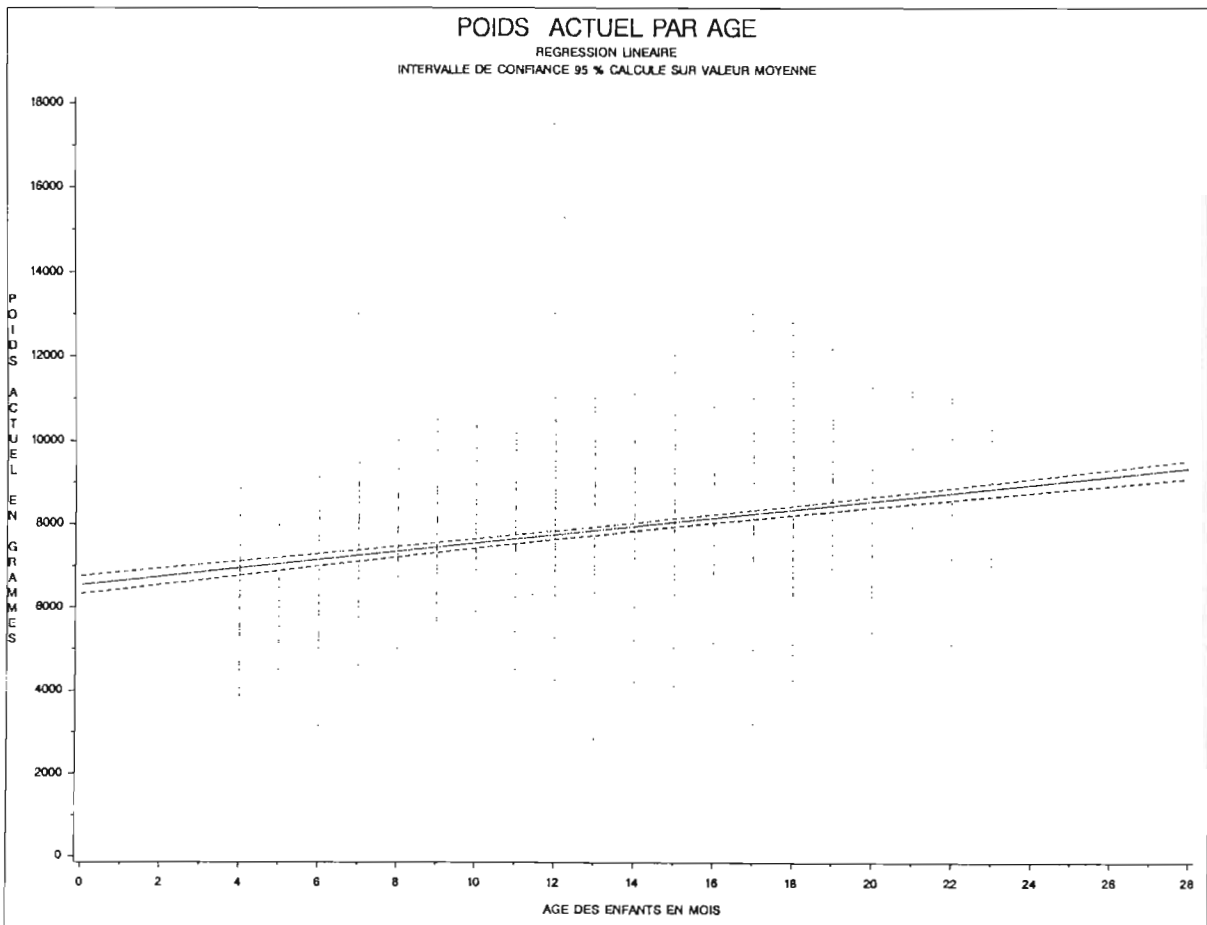
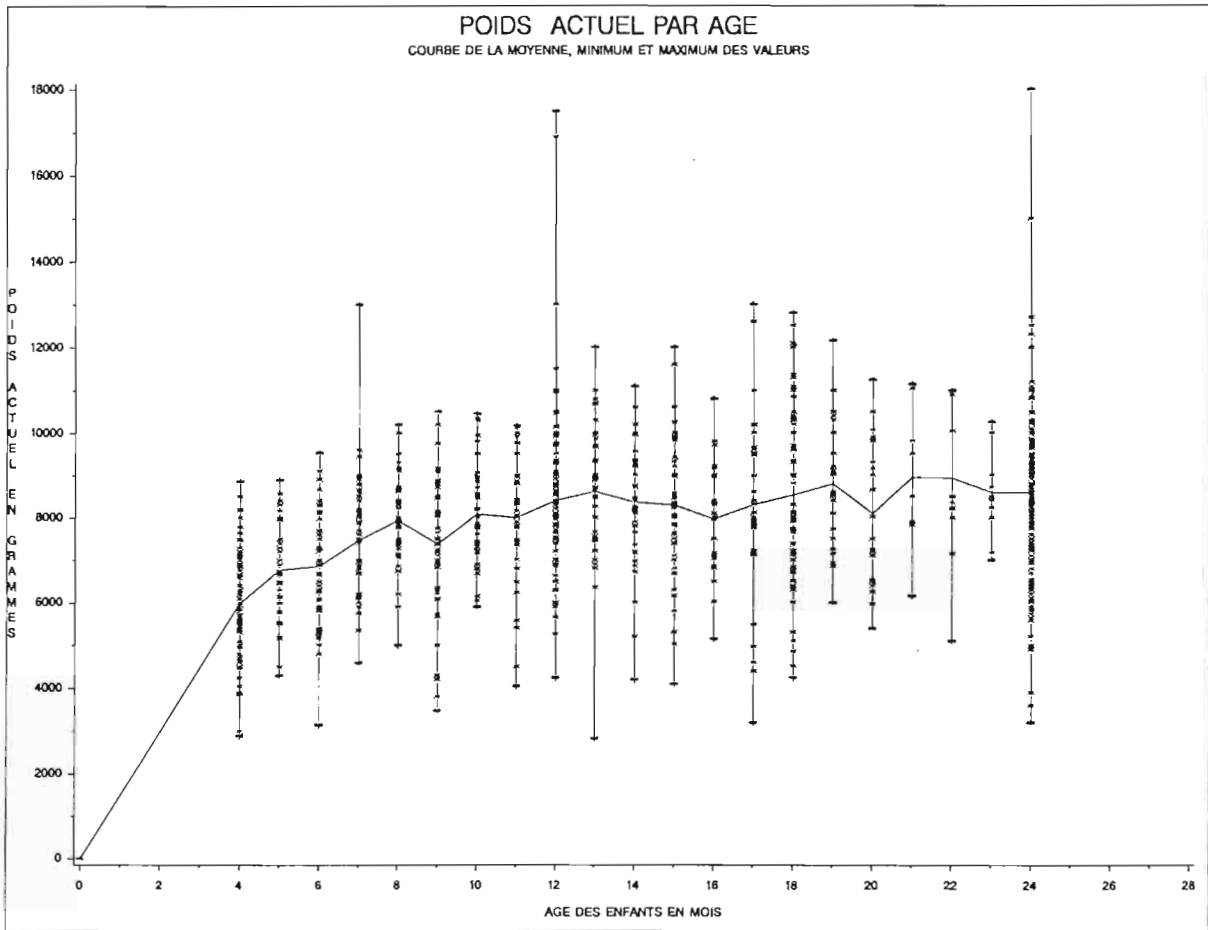
```
run;
```

3^e graphique : droite de régression linéaire

```
proc gplot data=base.popul;
  title2 'REGRESSION LINEAIRE ';
  title3 'INTERVALLE DE CONFIANCE 95 % CALCULE SUR VALEUR MOYENNE';
  symbol1 c=red h=3 cm v=point l=1
          rouge hauteur point type de ligne
          i=rlclm;
          droite de régression linéaire (rl) et l'intervalle de confiance à 95% à partir des valeurs
          moyennes des prédicats (clm)
          Autres possibilités : i=rlcli90 : intervalle de confiance à 90% à partir des valeurs individuelles
                               des prédicats
                               i=rlcli95 : intervalle de confiance à 95%
                               i=rlcli99 : intervalle de confiance à 99%
  initialisations de l'environnement du graphique
  plot v23*v18=1 /
       haxis=axis1 vaxis=axis2 caxis=black;
```

```
run;
```

• • •



IV.B

• • •

4^e graphique : régression quadratique

```
proc gplot data=base.popul;
  title2 'REGRESSION QUADRATIQUE';
  title3 'INTERVALLE DE CONFIANCE 90 % CALCULE SUR VALEUR INDIVIDUELLE';
  symbol1 v=point c=blue h=3 cm l=1
  i=rqcli90;
  courbe de régression quadratique et intervalle de confiance à 90 % à partir des valeurs
  individuelles des prédicats
  Autres possibilités : i=rqcli95 : intervalle de confiance à 95 %
  i=rqcli99 : intervalle de confiance à 99 %
  i=rqclm : intervalle de confiance à 95 % à partir des valeurs moyennes
  des prédicats (clm)

  plot v23*v18=1 /
      haxis=axis1 vaxis=axis2 caxis=black;
```

run;

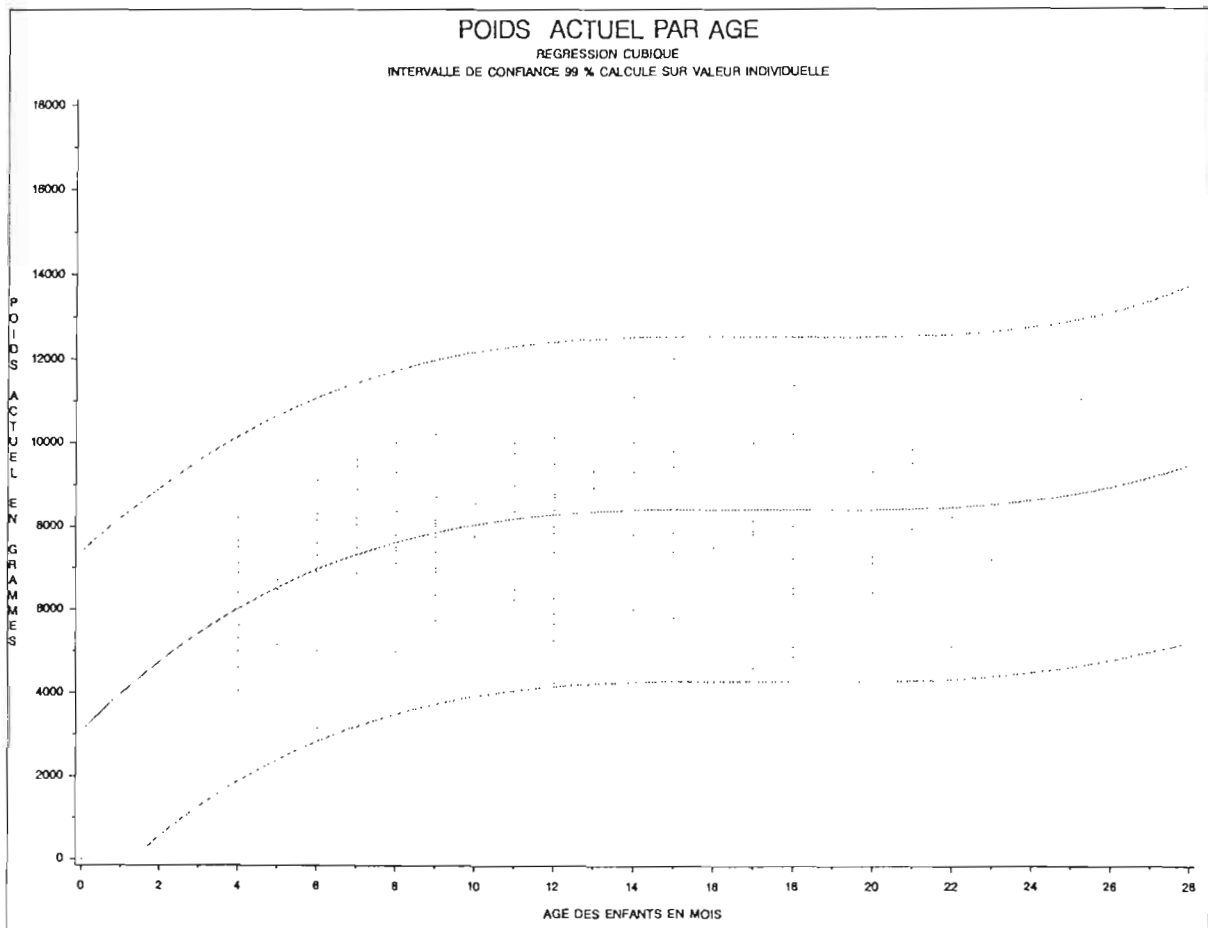
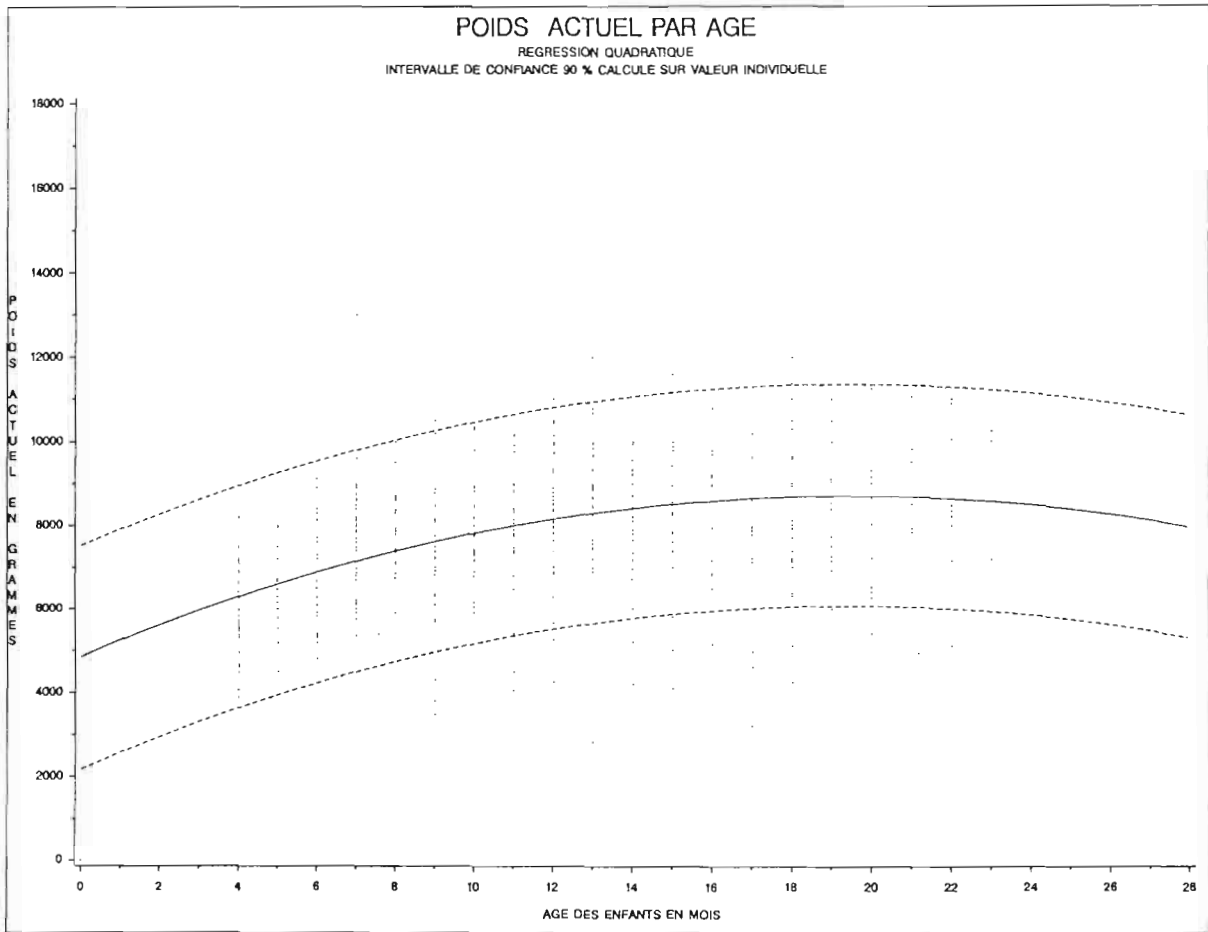
5^e graphique : régression cubique

```
proc gplot data=base.popul;
  title2 'REGRESSION CUBIQUE';
  title3 'INTERVALLE DE CONFIANCE 99 % CALCULE SUR VALEUR INDIVIDUELLE';
  symbol1 c=gold h=3 cm v=point l=1
  i=rccli99;
  courbe de régression cubique et intervalle de confiance à 99 % à partir des valeurs
  individuelles des prédicats
  Autres possibilités : i=rccli90 : intervalle de confiance à 90 %
  i=rccli95 : intervalle de confiance à 95 %
  i=rcclm : intervalle de confiance à 95 % à partir des valeurs moyennes
  des prédicats (clm)

  plot v23*v18=1 /
      haxis=axis1 vaxis=axis2 caxis=black;
```

run;

• • •



IV.B

• • •

6^e graphique : tracé simultané des courbes de régression linéaire, cubique et quadratique

```

proc gplot data=base.popul;
  title2 'REGRESSIONS LINEAIRE, CUBIQUE ET QUADRATIQUE';
  symbol1 c=red v=none l=1 i=rl;
  régression linéaire
  symbol2 c=blue v=none l=2 i=rq;
  régression quadratique
  symbol3 c=gold v=none l=3 i=rc;
  régression cubique
  symbol4 c=black v=none h=0.5 cm v=star;
  nuage de points

  plot v23*v18=1
  droite de régression linéaire
  v23*v18=2
  courbe de régression quadratique
  v23*v18=3
  courbe de régression cubique
  v23*v18=4 /
  nuage de points
  overlay caxis=black haxis=axis1 vaxis=axis2;
run;

```

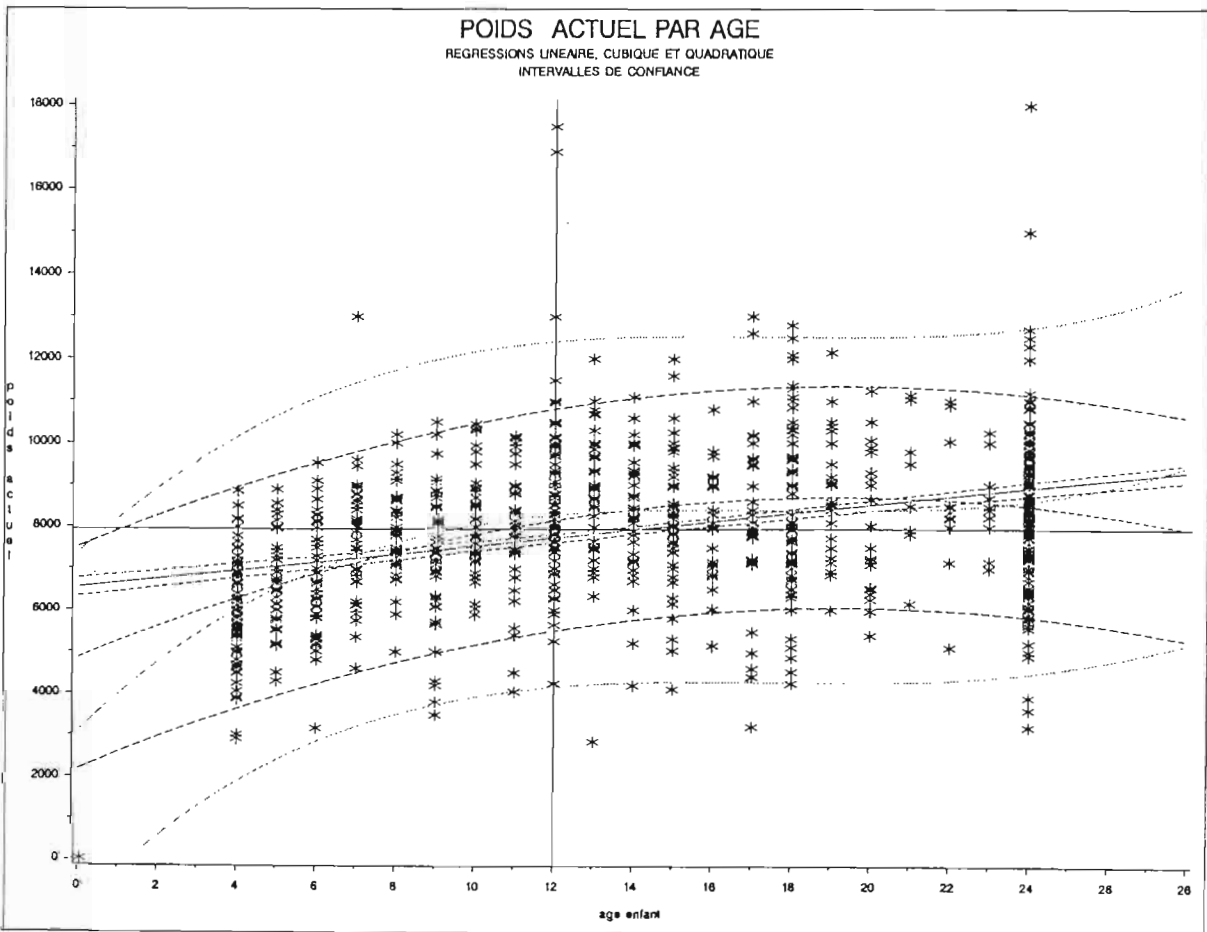
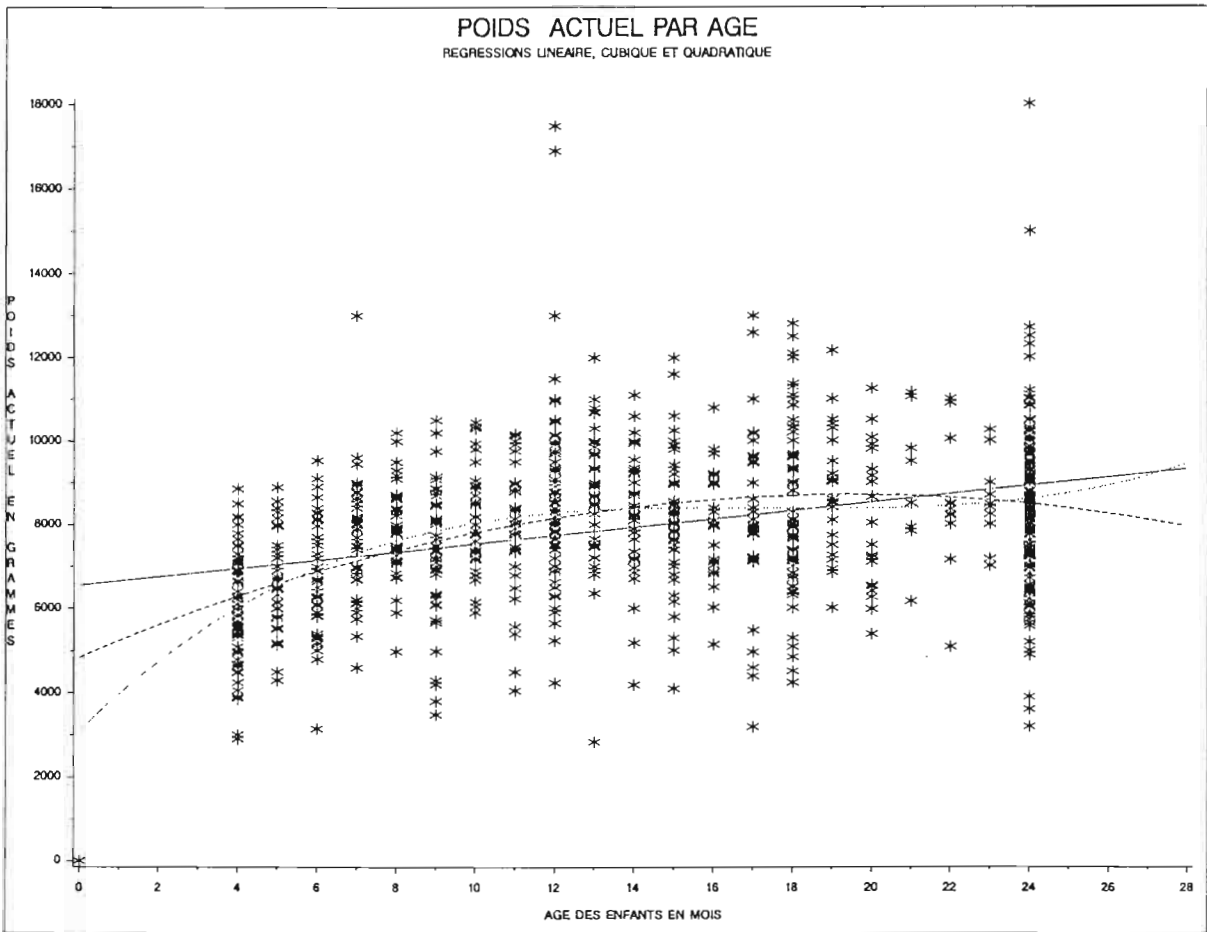
7^e graphique : tracé simultané des courbes de régression linéaire, cubique, quadratique et intervalles de confiance

```

proc gplot data=base.popul;
  title2 'REGRESSIONS LINEAIRE, CUBIQUE ET QUADRATIQUE';
  title3 'INTERVALLE DE CONFIANCE';
  symbol1 c=red v=none l=1 i=rlclm;
  régression linéaire
  symbol2 c=blue v=none l=2 i=rqli90;
  régression quadratique
  symbol3 c=gold v=none l=3 i=rccli99;
  régression cubique
  symbol4 c=black h=0.5 cm v=star;
  nuage de points

  plot v23*v18=1
  droite de régression linéaire
  v23*v18=2
  courbe de régression quadratique
  v23*v18=3
  courbe de régression cubique
  v23*v18=4 /
  nuage de points
  overlay
  haxis=0 to 28 by 2 vaxis=0 to 18000 by 2000
  hminor=1 vminor=1
  href=12 vref=7950 caxis=black;
run;

```



IV.C CRÉATION DE NOUVEAUX TABLEAUX GRAPHIQUES OU GÉOGRAPHIQUES

Soit à partir d'un tableau géographique existant

- créer le fond de carte relatif à une variable, ce qui équivaut à créer un sur-découpage,
- simplifier le contour des unités spatiales en ne retenant qu'une partie des points du tableau géographique initial,
- créer un tableau graphique, utilisable dans certaines procédures avec l'option **annotate**.

Soit créer un tableau graphique à visualiser indépendamment (**gslide**) ou en combinaison avec une carte (option **annotate** de **gmap**).

IV.C.1 Création d'un nouveau fond de carte à partir du fond de carte de la base

initialisations

```
libname base 'usr/usr3/graph90/docu';
options nocenter nodate nonumber;

proc format; value faxx
  low - -500 = '1'
  -500 - 0 = '2'
  0 - 500 = '3'
  500 - high = '4';
```

création d'un tableau intermédiaire regroupant les données géographiques et la variable discrétisée dont on veut créer le fond de carte

```
proc sort data=base.ouaga2 out=prov; by ident;
tri avant fusion
da3

by ident;
keep ident axx x y segment; format axx faxx.;
discrétisation de la variable
```

fusion des tableaux géographiques et descriptifs avant l'étape gremove

```
proc print data=zones (obs=10);
proc sort data=zones; by axx;
tri sur la variable traitée
```

création d'un nouveau fond de carte

```
proc gremove data=zones
tableau intermédiaire du fond de carte initial
out=nouveau;
tableau créé contenant les coordonnées des nouvelles unités spatiales

id ident;
identifiant des unités spatiales du tableau intermédiaire

by axx;
identifiant des unités spatiales du nouveau tableau

proc print data=nouveau (obs=10);
```

...

OUAGADOUGOU
FOND DE CARTE DE LA VARIABLE AXX

OBS	IDENT	AXX	X	Y	SEGMENT	
1	1	0	.	9618	3216	1
2	1	0	.	9625	3109	1
3	1	0	.	9629	2969	1
4	1	0	.	9764	2979	1
5	1	0	.	9772	3052	1
6	1	0	.	9760	3135	1
7	1	0	.	9732	3238	1
8	10	0	.	9084	3250	1
9	10	0	.	9487	3267	1
10	10	0	.	9379	3339	1

OUAGADOUGOU
FOND DE CARTE DE LA VARIABLE AXX

OBS	X	Y	SEGMENT	AXX
1	8533	2886	1	.
2	8553	2942	1	.
3	8570	3003	1	.
4	8593	3057	1	.
5	8639	3087	1	.
6	8669	3081	1	.
7	8704	3083	1	.
8	8749	3093	1	.
9	8790	3103	1	.
10	8829	3114	1	.

IV.C

• • •

initialisations de l'environnement graphique

```

goptions    device=sun    border cback=white;
pattern1    c=orange      v=s;    pattern2    c=brown      v=solid;
pattern3    c=green       v=s;    pattern4    c=yellow     v=s;
tittle1     f=swiss c=black 'OUAGADOUGOU';
tittle2     'FOND DE CARTE DE LA VARIABLE AXX';
footnote    '';

```

visualisation de la variable pour contrôle

```

proc gmap  data=nouveau      map=nouveau;
           données descriptives et géographiques sont dans le même tableau
           id      axx;
           identifiant des unités spatiales du nouveau découpage
           choro  axx /      discrete
                   coutline=black;
                   couleur des contours des unités spatiales
run;

```

- Les zones apparaissant sur la carte ci-jointe représentent les unités surfaciques par classe de la variable axx. Chaque nouvelle unité correspond à une ou plusieurs zones du fond de carte initial. L'option **missing** n'ayant pas été utilisée, les unités surfaciques correspondant à la **valeur manquante** de axx ne sont pas visualisées.
- La procédure **gremove** permet de créer un nouveau fond de carte, à partir des tableaux géographiques existants, qui correspond aux unités spatiales d'une des variables descriptives. Le tableau d'entrée de la procédure **gremove** doit être préalablement trié sur cette variable.



IV.C

IV.C.2 Généralisation d'un fond de carte

Greduce est une étape qui permettra ensuite dans une **étape data**, de réduire le nombre de points décrivant les polygones et donc de simplifier les contours des unités spatiales.

initialisations

```
goptions border cback=white ctext=black;
libname base '/usr/usr3/graph90/docu';
options nonumber nocenter nodate;
tittle1 f=swiss c=black 'OUAGADOUGOU';
title2 'GENERALISATION DES LIMITES DE ZONES';
pattern1 c=black v=e;
```

réduction des points du contour

```
proc greduce data=base.ouaga2g out=prov; id ident;
La procédure crée dans le tableau de sortie, une variable density. La valeur de cette variable représente pour chaque observation du tableau, donc pour chaque point du polygone, le niveau de précision auquel il correspond par rapport au tableau initial ouga2g.
```

```
proc contents data=prov;
proc print data=prov (obs=10);
proc freq data=prov; tables density;
```

pour connaître la répartition des points par niveau de densité

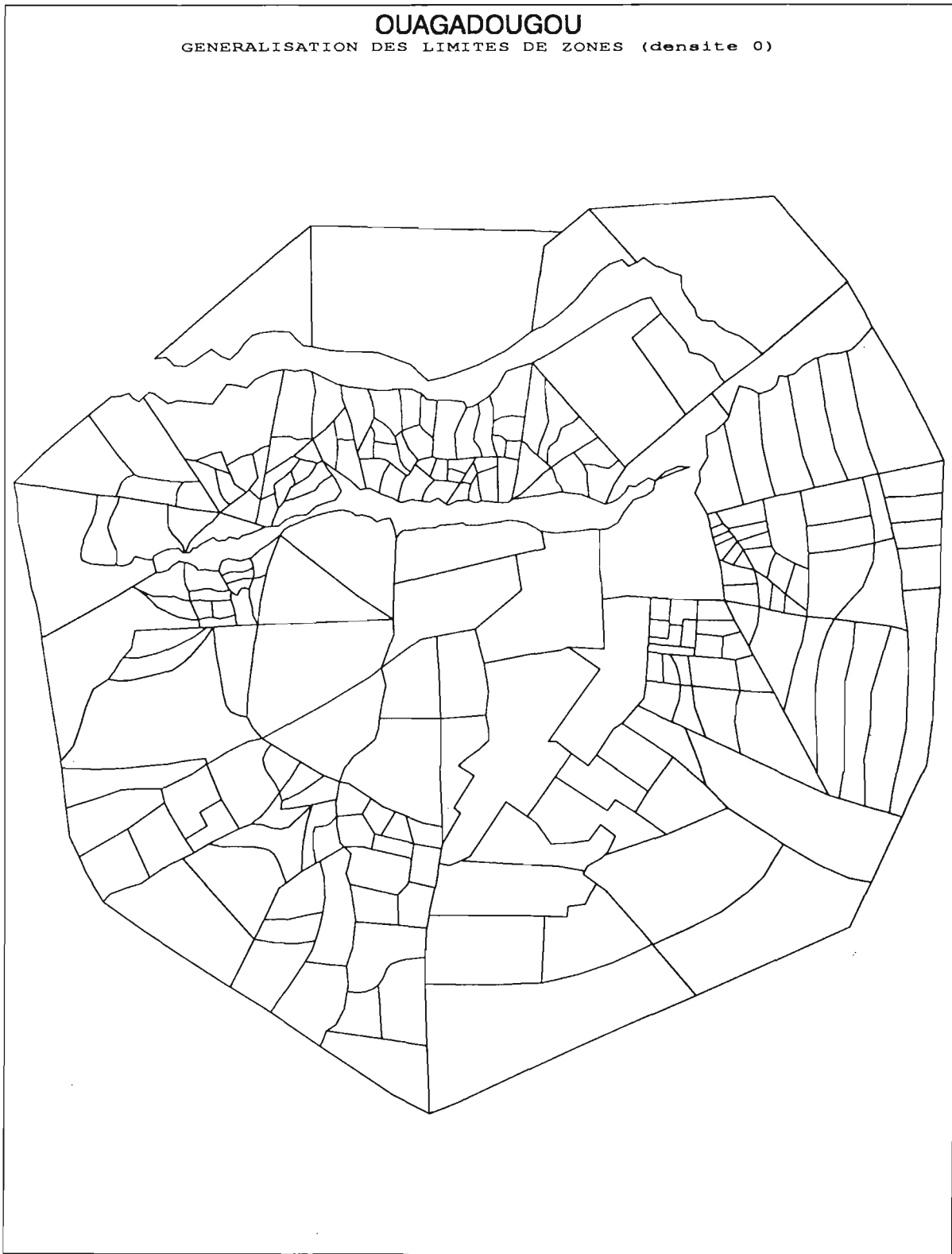
```
data prov1; set prov; h=1; if density=0;
sélection des points correspondant au niveau de densité le plus faible et création d'une variable pour la cartographie
run;
```

visualisation

```
goptions device=aplplus gaccess='sasgastd>graph2';
définition du périphérique de sortie (l'imprimante laser) et nom du fichier des résultats
title2 'GENERALISATION DES LIMITES DE ZONES (densité=0)';
proc gmap data=prov1 map=base.ouaga2g;
id ident;
choro h / discrete nolegend;
run;
```

- **greduce** crée une variable **density** codée de 0 à 6. Chaque point d'un polygone est affecté d'une valeur qui correspond à son niveau de précision par rapport au document initial. Ce calcul tient compte à la fois de la distance entre les points et du nombre de points dans le fond de carte.
- Les sommets communs à plusieurs segments seront codés 0, niveau qui correspond au niveau de généralisation maximale possible sans perte de polygones. Les points codés en 1 apportent une information supplémentaire au niveau 0 et donc une plus grande précision des contours. Le niveau suivant sera encore plus proche du document initial et donc la généralisation des limites de zones sera plus faible, etc., jusqu'au niveau 6 où tous les points du tableau initial sont conservés. La fréquence de la variable **density** aide à définir le niveau de densité à choisir en fonction du résultat attendu.

• • •



• • •

- Dans la fenêtre **log** apparaît :

GREDUCE PROCEDURE
REDUCTION RESULTS

— N VALUE SUMMARY —		
DENSITY	TARGET	MAXIMUM
3	19	19
4	21	21
5	33	33
— E VALUE SUMMARY —		
DENSITY	INITIAL	FINAL
3	0.0	92.103448
4	0.0	23.025862
5	0.0	5.756466

Par défaut SAS calcule, pour chaque niveau de densité, un nombre maximum de points par polygone et la distance euclidienne minimale entre deux points consécutifs.

Ce nombre de points et cette distance par niveau peuvent être définis par les options de la procédure **greduce** **Nn** et **En**.

Exemples : **N1=50** : nombre de points maximum à retenir au niveau de densité 1, pour un polygone,
E1= 25.75 : distance minimum entre deux points pour le niveau de densité 1.

OUAGADOUGOU
GENERALISATION DES LIMITES DE ZONES

CONTENTS PROCEDURE

Data Set Name: WORK.PROV Type:
Observations: 2497 Record Len: 37
Variables: 5
Label:

-----Alphabetic List of Variables and Attributes-----

#	Variable	Type	Len	Pos	Label
5	DENSITY	Num	4	33	
1	IDENT	Char	5	4	
4	SEGMENT	Num	8	25	
2	X	Num	8	9	
3	Y	Num	8	17	

OUAGADOUGOU
GENERALISATION DES LIMITES DE ZONES

OBS	IDENT	X	Y	SEGMENT	DENSITY	
1	1	0	9618	3216	1	0
2	1	0	9625	3109	1	6
3	1	0	9629	2969	1	0
4	1	0	9764	2979	1	0
5	1	0	9772	3052	1	5
6	1	0	9760	3135	1	0
7	1	0	9732	3238	1	0
8	10	0	9084	3250	1	0
9	10	0	9487	3267	1	0
10	10	0	9379	3339	1	5

OUAGADOUGOU
GENERALISATION DES LIMITES DE ZONES

DENSITY	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	1112	44.5	1112	44.5
2	9	0.4	1121	44.9
4	104	4.2	1225	49.1
5	361	14.5	1586	63.5
6	911	36.5	2497	100.0

IV.C

IV.C.3 Création d'un tableau graphique à partir d'un tableau géographique existant

initialisations

```
libname base '/usr/usr3/graph90/docu';
goptions device=sun border cback=white ftext=swiss;
affichage sur l'écran avec un cadre et la couleur de fond blanc
options nocenter nodate nonumber;
title; title1 'TABLEAU ANNOTATE';
```

```
proc print data=base.ouagag1 (obs=10);
proc means data=base.ouaga1g max min;
permet d'obtenir les valeurs extrêmes des coordonnées géographiques
data temp; set base.ouaga1g;
x=(x-8250)/30; y=(y-1750)/30;
```

À cause du choix ultérieur du système de référence de l'affichage (*xsys* et *ysys*), il est nécessaire que les coordonnées soient incluses dans l'intervalle 0-100; d'où cette **étape data** intermédiaire à concevoir en fonction des valeurs minimales et maximales des coordonnées *x* et *y* affichées par **means**.

```
proc means data=temp max min;
permet de vérifier l'encadrement des coordonnées x et y
```

création du tableau annotat

```
data prov; set temp;
length function color $8;
définition de la longueur des variables fonction et color du tableau
line=1 color='black' size=2 style='l1';
choix, couleur, épaisseur du tracé de la ligne choix de la trame
xsys='3'; ysys='3';
choix du système de référence pour l'affichage : 0 à 100 % de la page graphique (cf. SAS/GRAPH User's Guide 6.03, page 123)
```

```
cle=lag(ident); seg=lag(segment);
lag : fonction SAS dont l'argument peut être numérique ou alphanumérique et qui conserve la valeur de l'argument issu du dernier appel à la fonction. Donc cle contient la valeur de la variable ident de l'observation précédente et seg celle de la variable segment de l'observation précédente.
```

```
function='poly';
l'action consiste à tracer un polygone dont le point de coordonnées (x,y) est le premier point
```

```
if cle eq ident then
  if seg eq segment then function='polycont';
tests qui permettent de comparer deux valeurs successives de segment et d'ident : la valeur de fonction égale 'poly' chaque fois que l'on change d'identifiant ou de segment, c'est-à-dire à chaque nouveau polygone.
```

Le polygone sera représenté dans la couleur de l'option **color**, avec la trame de l'option **style**. Le type de ligne de son contour est défini par l'option **line** et l'épaisseur du trait par l'option **size**.

```
keep function x y xsys ysys line color style size;
proc print data=prov (obs=10);
```

visualisation pour contrôle

```
proc gslide annotate=prov;
tableau graphique contenant l'ensemble des informations à afficher
run;
```

• • •



• • •

- Exemple d'utilisation de ce programme : afficher un découpage géographique ou un graphique (échelle, références, etc.) sur un résultat de procédure **gmap** avec l'option **annotate** sous réserve d'utiliser comme valeur de l'option **style** la valeur **'e'** (pas de couleur dans le polygone) ou les valeurs **'move'** et **'drawn'** comme valeurs de la variable optionnelle **function**.
- La taille d'une sortie graphique est différente selon le périphérique défini dans l'instruction **goptions**. Dans un environnement comme le nôtre, station Sun et imprimante LaserWriter Plus, on peut constater que les sorties sur imprimante Laser déforment l'image : le « pas » des unités sur l'axe des y est différent de celui de l'axe des x.

TABLEAU ANNOTATE

OBS	IDENT	X	Y	SEGMENT
1	1	9618	3216	1
2	1	9625	3109	1
3	1	9629	2969	1
4	1	9764	2979	1
5	1	9772	3052	1
6	1	9760	3135	1
7	1	9732	3238	1
8	10	9084	3250	1
9	10	9487	3267	1
10	10	9379	3339	1

TABLEAU ANNOTATE

N Obs	Variable	Minimum	Maximum
772	X	8355.00	11126.00
	Y	1781.00	4542.00
	SEGMENT	1.0000000	2.0000000

TABLEAU ANNOTATE

N Obs	Variable	Minimum	Maximum
772	X	3.5000000	95.8666667
	Y	1.0333333	93.0666667
	SEGMENT	1.0000000	2.0000000

TABLEAU ANNOTATE

OBS	X	Y	FUNCTION	COLOR	LINE	SIZE	STYLE	XSYS	YSYS
1	45.6000	48.8667	poly	black	1	2	11	3	3
2	45.8333	45.3000	polycont	black	1	2	11	3	3
3	45.9667	40.6333	polycont	black	1	2	11	3	3
4	50.4667	40.9667	polycont	black	1	2	11	3	3
5	50.7333	43.4000	polycont	black	1	2	11	3	3
6	50.3333	46.1667	polycont	black	1	2	11	3	3
7	49.4000	49.6000	polycont	black	1	2	11	3	3
8	27.8000	50.0000	poly	black	1	2	11	3	3
9	41.2333	50.5667	polycont	black	1	2	11	3	3
10	37.6333	52.9667	polycont	black	1	2	11	3	3

IV.C

IV.C.4 Création d'un tableau graphique sans données initiales : palette de gris

```

goptions device=aplplus gaccess='sasgastd>graph';
définition du périphérique de sortie (l'imprimante laser) et du fichier des résultats
options symbolgen;
title;

%macro carre (couleur,xb,yb,xh,yh);
description d'un quadrilatère
color=&couleur;
définition de la couleur lors de l'exécution de la macroprocédure
function='move'; x=&xb; y=&yb; output;
positionnement sans tracé sur le point de coordonnées (xb,yb), point bas du quadrilatère
function='bar'; x=&xh; y=&yh; output;
déplacement avec tracé selon les axes x et y du point précédent au point courant de coordonnées (xh,yh), point haut du quadrilatère
%mend carre;

dessin de carrés de couleurs
data squares;
length function style color $ 8;
déclaration des variables de noms prédéfinis par SAS pour les tableaux annotatés
xsys='2'; ysys='2';
définition du système de référence de l'affichage
style='solid';
choix de la représentation; ici aplat de couleur

%carre ('gray00', 4, 4, 28, 16);
1er appel à la macroprocédure « carre » : dessin d'un rectangle de couleur gris (gray00). (4,4) et (28,16) sont respectivement les coordonnées des points bas gauche et haut droit du rectangle.
%carre ('gray08', 28, 4, 52, 16);
%carre ('gray10', 52, 4, 76, 16);
%carre ('gray18', 76, 4, 100, 16);
fin de la première ligne

%carre ('gray20', 4, 16, 28, 28);
%carre ('gray28', 28, 16, 52, 28);
%carre ('gray30', 52, 16, 76, 28);
%carre ('gray38', 76, 16, 100, 28);

%carre ('gray40', 4, 28, 28, 44);
%carre ('gray48', 28, 28, 52, 44);
%carre ('gray50', 52, 28, 76, 44);
%carre ('gray58', 76, 28, 100, 44);

%carre ('gray60', 4, 40, 28, 52);
%carre ('gray68', 28, 40, 52, 52);
%carre ('gray70', 52, 40, 76, 52);
%carre ('gray78', 76, 40, 100, 52);

```

...

IV.C

• • •

```

%carre ('gray80', 4, 52, 28, 64);
%carre ('gray88', 28, 52, 52, 64);
%carre ('gray90', 52, 52, 76, 64);
%carre ('gray98', 76, 52, 100, 64);

%carre ('gray80', 4, 52, 28, 64);
%carre ('gray88', 28, 52, 52, 64);
%carre ('gray90', 52, 52, 76, 64);
%carre ('gray98', 76, 52, 100, 64);

%carre ('graya0', 4, 64, 28, 76);
%carre ('graya8', 28, 64, 52, 76);
%carre ('grayb0', 52, 64, 76, 76);
%carre ('grayb8', 76, 64, 100, 76);

%carre ('grayc0', 4, 76, 28, 88);
%carre ('grayc8', 28, 76, 52, 88);
%carre ('grayd0', 52, 76, 76, 88);
%carre ('grayd8', 76, 76, 100, 88);

%carre ('graye0', 4, 88, 28, 100);
%carre ('graye8', 28, 88, 52, 100);
%carre ('grayf0', 52, 88, 76, 100);
%carre ('grayf8', 76, 88, 100, 100);

```

affichage des composantes des couleurs

```

data nomc;
length function text $8;
xsys='2'; ysys='2';
système de référence de l'affichage
function='label'; style='swiss'; size=1.1;
écriture de texte dans la fonte 'swiss' et d'une hauteur égale à 1.1 unité courante
color='white';
couleur des textes
x=7; y=6; text='gray00'; output;
écriture de l'observation courante (instruction output) dans le tableau courant c'est-à-dire écriture de la chaîne de caractères gray00, composante de la couleur du rectangle, sur le point de coordonnées (x,y)
x=31; y=6; text='gray08'; output;
x=55; y=6; text='gray10'; output;
x=79; y=6; text='gray18'; output;

x=7; y=18; text='gray20'; output;
x=31; y=18; text='gray28'; output;
x=55; y=18; text='gray30'; output;
x=79; y=18; text='gray38'; output;

x=7; y=30; text='gray40'; output;
x=31; y=30; text='gray48'; output;
x=55; y=30; text='gray50'; output;
x=79; y=30; text='gray58'; output;

```

• • •

IV.C

• • •

```

x=7;          y=42;          text='gray60';      output;
x=31;         y=42;          text='gray68';      output;
x=55;         y=42;          text='gray70';      output;
x=79;         y=42;          text='gray78';      output;

x=7;          y=54;          text='gray80';      output;
x=31;         y=54;          text='gray88';      output;
x=55;         y=54;          text='gray90';      output;
x=79;         y=54;          text='gray98';      output;

x=7;          y=66;          text='graya0';      output;
x=31;         y=66;          text='graya8';      output;
x=55;         y=66;          text='grayb0';      output;
x=79;         y=66;          text='grayb8';      output;

x=7;          y=78;          text='grayc0';      output;
x=31;         y=78;          text='grayc8';      output;
x=55;         y=78;          text='grayd0';      output;
x=79;         y=78;          text='grayd8';      output;

color='black';
réinitialisation de la couleur des textes
x=7;          y=90;          text='graye0';      output;
x=31;         y=90;          text='graye8';      output;
x=55;         y=90;          text='grayf0';      output;
x=79;         y=90;          text='grayf8';      output;

```

```

data      anno;      set      squares nomc;
concaténation des deux tableaux

```

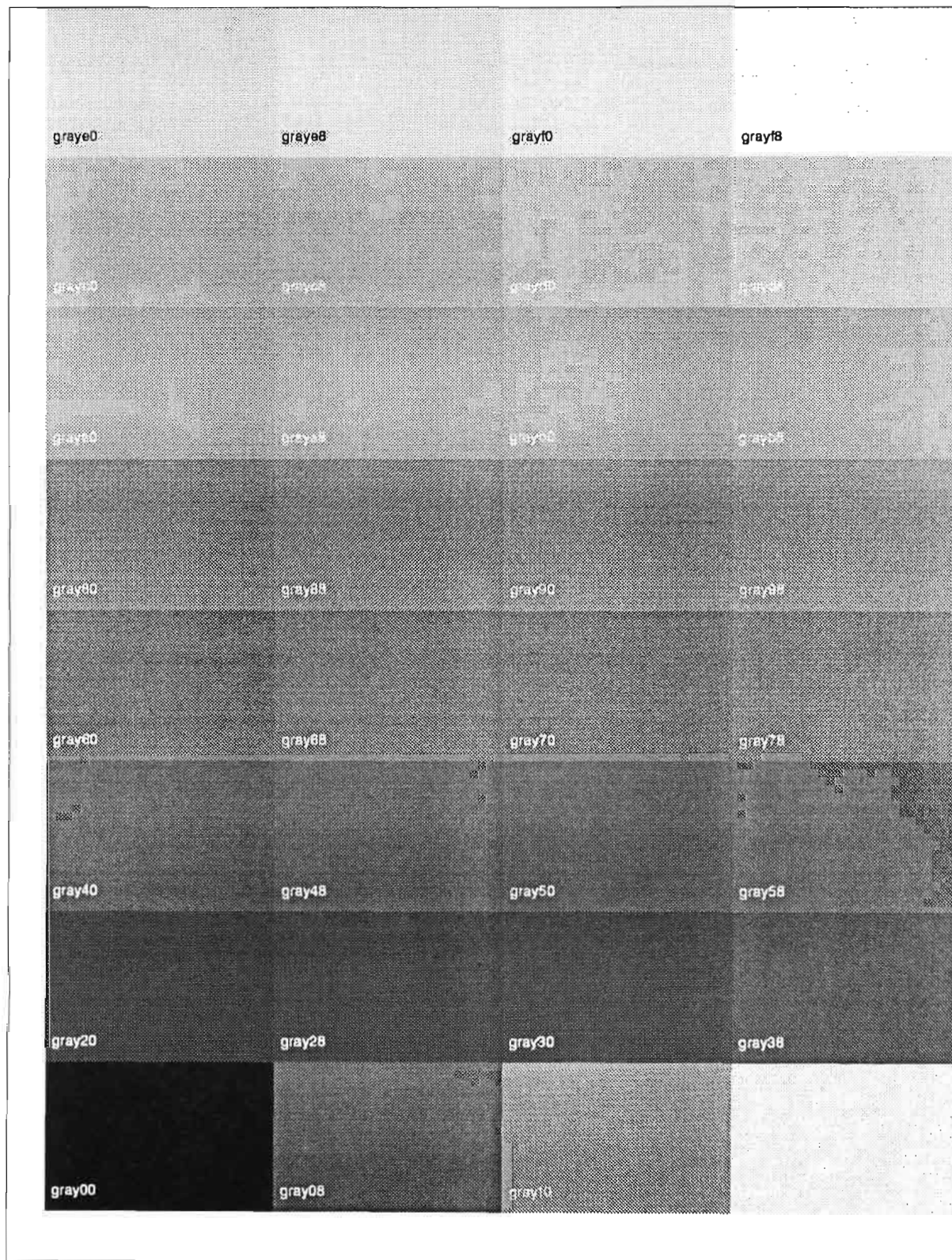
visualisation

```

goptions  border      cback=white;
proc      gslide      annotate=anno;
étape de visualisation de tableaux annotate
run;

```

- Remarque : le temps d'exécution de ce programme est relativement long, en particulier à cause du temps nécessaire à l'affichage des étapes relatives aux appels de la macroprocédure. Pour obtenir rapidement une palette de couleurs, on peut :
 - créer de façon permanente un **tableau annotate** contenant l'ensemble des informations du programme ci-dessus; il suffirait alors de le visualiser dans un fichier ou sur l'écran,
 - faire exécuter le programme en mode non interactif ce qui réduit le temps d'exécution.



IV.D EXEMPLE : REPRÉSENTATION SPATIALE D'UN PLAN FACTORIEL

IV.D.1 Cartographie du plan factoriel

Visualisation simultanée d'une carte créée par une étape **proc gmap** et d'un plan graphique contenu dans un **tableau annotate** (limites et codes des secteurs de l'étude)

initialisations de l'environnement

```

goptions      device=sun      border      cback=gray  ctext=black;
libname       base           '/usr/usr3/graph90/docu';
title1        f=swiss        c=black     'OUAGADOUGOU';
title2        f=swiss        c=black     'NIVEAU ZONE';
title3        f=swiss        c=black     'INTER-CLASSES';
pattern1      c=vig          v=s;
pattern2      c=vilg         v=s;
pattern3      c=lig          v=s;
pattern4      c=vliyg        v=s;
pattern5      c=dayg         v=s;
pattern6      c=moyg         v=s;
pattern7      c=molg         v=s;
pattern8      c=vderp        v=s;
pattern9      c=dagra        v=s;
pattern10     c=gray         v=s;
pattern11     c=grlg         v=s;
pattern12     c=palg         v=s;
pattern13     c=dap          v=s;
pattern14     c=grrp         v=s;
pattern15     c=parp         v=s;
pattern16     c=pkgr         v=s;
pattern17     c=strp         v=s;
pattern18     c=magenta      v=s;
pattern19     c=vlip         v=s;
pattern20     c=vpap         v=s;

```

création d'un tableau annotate relatif au découpage des secteurs

limites des secteurs

```

data provzon;      set      base.limsect;
length            function color $ 8;
line=1;          size=2;      color='white';
choix,           épaisseur,   couleur de la ligne
xsys='2';        ysys='2';    when='a';
système de référence           moment où s'effectuera l'affichage
if      cle ne 'hors';          if      cle ne '1';      if      cle ne '2';
if      cle ne '3';            if      cle ne '4';      if      cle ne '5';
if      cle ne '6';            if      cle ne '7';      if      cle ne '8';
if      cle ne '9';            if      cle ne '10';     if      cle ne '11';
if      cle ne '12';           if      cle ne '13';     if      cle ne '14';

```

sélection des identifiants correspondant aux limites des secteurs enquêtés

• • •

IV.D

...

noms des secteurs

```

data nomcle;      set    base.ouaga1ce;
  length          fonction $ 5 text $ 4;
  size=2;        color='white';
  fonction='label'; style='swiss'; position='5';
écriture de texte dans la fonte de caractères 'swiss', centré sur le point de coordonnées (x,y)
  xsys='2';      ysys='2';      when='a';
when='a' : variable de séquençement qui définit le moment où s'effectue l'affichage de ces données par rapport
à l'affichage de l'ensemble des autres données ('a' : après ou 'b' avant une action c'est-à-dire en début
d'affichage).
  if    cle ne 'hors';          if    cle ne '1';          if    cle ne '2';
  if    cle ne '3';            if    cle ne '4';          if    cle ne '5';
  if    cle ne '6';            if    cle ne '7';          if    cle ne '8';
  if    cle ne '9';            if    cle ne '10';         if    cle ne '11';
  if    cle ne '12';           if    cle ne '13';         if    cle ne '14';
sélection sur la variable cle des identifiants des secteurs enquêtés
  text=cle;

```

```

data anno; set    provzon nomcle;
concaténation du tableau des contours et du tableau des noms

```

création et classification de la variable à cartographier

```

data prov; set    base.ouaga2;
  if axx<-450 and axy<-450                                then axay=11;
  else
  if axx<-450 and axy>-450 and axy<-150                  then axay=12;
  else
  if axx<-450 and axy>-150 and axy<150                    then axay=13;
  else
  if axx<-450 and axy>150 and axy<450                     then axay=14;
  else
  if axx<-450 and axy>450                                  then axay=15;
  else
  if axx>-450 and axx<-150 and axy<-450                  then axay=21;
  else
  if axx>-450 and axx<-150 and axy>-450 and axy<-150    then axay=22;
  else
  if axx>-450 and axx<-150 and axy>-150 and axy<150    then axay=23;
  else
  if axx>-450 and axx<-150 and axy>150 and axy<450      then axay=24;
  else
  if axx>-450 and axx<-150 and axy>450                  then axay=25;
  else
  if axx>-150 and axx<150 and axy<-450                   then axay=31;
  else
  if axx>-150 and axx<150 and axy>-450 and axy<-150    then axay=32;
  else

```

...

IV.D

•••

```

if axx>-150 and axx<150 and axy>-150 and axy<150      then axay=33;
else
if axx>-150 and axx<150 and axy>150 and axy<450      then axay=34;
else
if axx>-150 and axx<150 and axy>450                  then axay=35;
else
if axx>150 and axx<450 and axy<-450                  then axay=41;
else
if axx>150 and axx<450 and axy>-450 and axy<-150     then axay=42;
else
if axx>150 and axx<450 and axy>-150 and axy<150     then axay=43;
else
if axx>150 and axx<450 and axy>150 and axy<450      then axay=44;
else
if axx>150 and axx<450 and axy>450                  then axay=45;
else
if axx>450 and axy<-450                               then axay=51;
else
if axx>450 and axy>-450 and axy<-150                 then axay=52;
else
if axx>450 and axy>-150 and axy<150                 then axay=53;
else
if axx>450 and axy>150 and axy<450                  then axay=54;
else
if axx>450 and axy>450                               then axay=55;

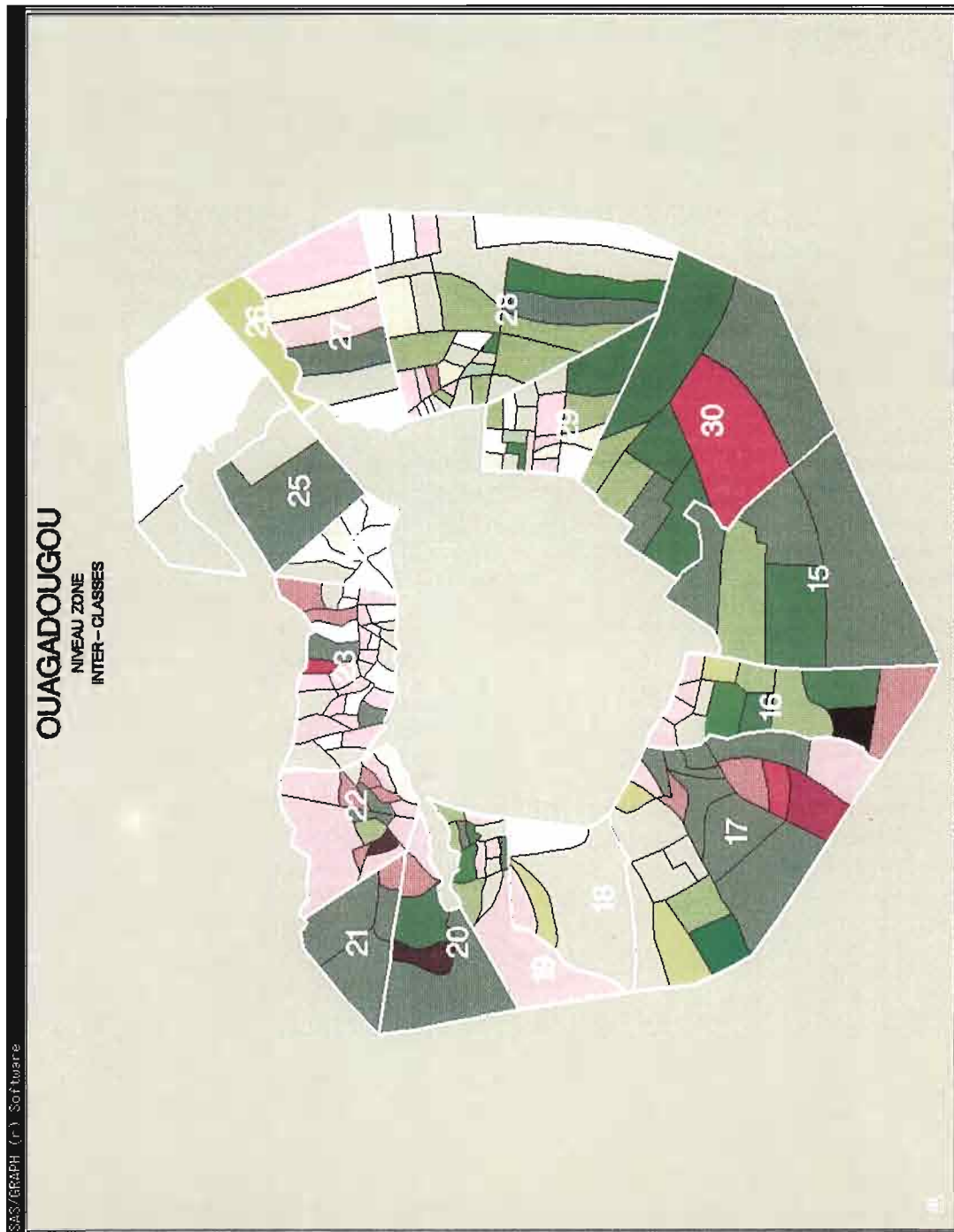
```

cartographie de la variable avec habillage

```

proc gmap data=prov      map=base.ouaga2g;
  id      ident;
  choro  axay /          discrete      annotate=anno
                                     tableau contenant l'habillage
                                     nolegend   outline=black;
                                     sans légende      couleur du contour des unités surfaciques
run;

```

IV.D

IV.D.2 Plan factoriel : légende associée à la carte

Il s'agit de la légende à associer à la carte précédente : elle visualise le plan factoriel défini par le système d'axes *axx* et *axy* et donne, dans ce plan, la position des variables étudiées.

initialisations

```
options symbolgen; goptions reset=all;
goptions device=sun border cback=white;
```

définition des pavés du plan factoriel, légende de la carte

```
%macro carre (couleur,xb,yb,xh,yh);
  color=&couleur;
  fonction='move'; x=&xb; y=&yb; output;
  positionnement sans tracé sur le point de coordonnées (xb,yb), point bas gauche du rectangle
  fonction='bar'; x=&xh; y=&yh; output;
  positionnement avec tracé d'un rectangle du point précédent au point courant de coordonnées (xh,yh), point haut droit du rectangle
%mend carre;
```

création du plan factoriel, légende de la carte

```
data squares;
  length function style color $ 8;
  xsys='3'; ysys='3';
  système de référence pour l'affichage qui suppose des valeurs de coordonnées incluses dans l'intervalle 0-100
  style='solid';
  choix de la représentation du rectangle
  %carre ('deyg', 0, 0, 16, 16);
  %carre ('deg', 16, 0, 32, 16);
  %carre ('vderp', 32, 0, 48, 16);
  %carre ('dap', 48, 0, 64, 16);
  %carre ('virp', 64, 0, 80, 16);

  %carre ('vig', 0, 16, 16, 32);
  %carre ('dayg', 16, 16, 32, 32);
  %carre ('dagray', 32, 16, 48, 32);
  %carre ('grrp', 48, 16, 64, 32);
  %carre ('strp', 64, 16, 80, 32);

  %carre ('vilg', 0, 32, 16, 48);
  %carre ('moyg', 16, 32, 32, 48);
  %carre ('gray', 32, 32, 48, 48);
  %carre ('parp', 48, 32, 64, 48);
  %carre ('magenta', 64, 32, 80, 48);

  %carre ('lig', 0, 48, 16, 64);
  %carre ('molg', 16, 48, 32, 64);
  %carre ('grlg', 32, 48, 48, 64);
  %carre ('pkgr', 48, 48, 64, 64);
  %carre ('vlip', 64, 48, 80, 64);
```

• • •

IV.D

• • •

```

%carre      ('vliyg',    0,    64,    16,    80);
%carre      ('lilg',     16,    64,    32,    80);
%carre      ('palg',     32,    64,    48,    80);
%carre      ('ltgray',   48,    64,    64,    80);
%carre      ('vpap',     64,    64,    80,    80);

```

positionnement du titre

```

data titre;
  length      function $8 text $100;
  xsys='3';   ysys='3';
  function='label'; style='swiss'; color='black'; size=1.5;
  écriture de texte, fonte de caractères 'swiss', en noir et d'une hauteur égale à 1.5 unité courante
  position='C'; x=2; y=82;
  texte positionné en haut et à droite du point de coordonnées (2,82)
  text='legende : plan factoriel principal de l'analyse inter-classes associee au niveau
  Zone';      output;
  chaîne de caractères à écrire

```

positionnement des noms des classes sur les axes

```

data nomc;
  length      function text $25;
  xsys='3';   ysys='3';
  function='label'; style='swiss'; color='black'; size=1.1;
  x=14;       y=12;       text='non lotie';      output;
  x=34;       y=28;       text='MODE D"ATTRIBUTION'; output;
  x=44;       y=24;       text='MODERNE ET RECENTE'; output;
  x=64;       y=20;       text='recommemt lotie'; output;
  x=72;       y=44;       text='agriculteur';    output;
  x=56;       y=60;       text='originaire';     output;
  x=56;       y=58;       text='de Ouagadougou'; output;
  x=66;       y=69;       text='MODE';           output;
  x=66;       y=62;       text='D"ATTRIBUTION'; output;
  x=66;       y=55;       text='TRADITIONNELLE'; output;
  x=66;       y=49;       text='ET';             output;
  x=66;       y=42;       text='ANCIENNE';      output;
  x=5;        y=75;       text='ETUDIANT';      output;
  x=42;       y=18;       text='menage parti';   output;
  x=51;       y=34;       text='proprietaire';   output;
  x=10;       y=70;       text='locataire';     output;
  x=55;       y=65;       text='en cours de lotissement'; output;
  x=25;       y=68;       text='groupe';        output;
  x=25;       y=66;       text='polyresidentiel'; output;
  x=13;       y=43;       text='non mossi';     output;
  x=15;       y=30;       text='arrivee recente'; output;
  x=75;       y=66;       text='arrivee ancienne'; output;
  x=12;       y=54;       text='lotie';         output;
  x=13;       y=50;       text='celibataire';   output;

```

• • •

IV.D

• • •

tracé des axes factoriels

```
data axex;
  length          function color $ 8;
  xsys='3';      ysys='3';
  function='move';  x=0;          y=40;          output;
  positionnement sans tracé au début de l'axe
  function='draw';  x=80;         y=40;
  positionnement avec tracé du point précédent au point courant
  color='black';    line=1;       size=2 ;      output;
  tracé en noir avec un trait continu d'une épaisseur correspondant à l'option size
```

```
data axey;
  xsys='3';      ysys='3';
  function='move';  x=40;          y=0;
  function='draw';  x=40;          y=80;
  color='black';    line=1;       size=2;      output;
```

positionnement du nom des axes

```
data noma;
  length          function text $ 8;
  xsys='3';      ysys='3';
  function='label';  style='swiss';  size=1.25;
  écriture de texte, fonte de caractères 'swiss' et d'une hauteur égale à 1.25 unité courante
  color='black';    position='6';
  couleur des textes point au centre et à gauche du texte
  x=76;            y=42;  text='Axe 1';  output;
  coordonnées du point chaîne à écrire
  x=41;            y=78;  text='Axe 2';  output;
```

concaténation des tableaux

```
data anno;      set  squares titre nomc axex axey noma;
```

affichage de la légende

```
proc gslide      annotate=anno;
run;
```


INDEX

INDEX	TYPE	NIVEAU	SYNTAXE	PAGES	COMMENTAIRES
a	mot-clé	when	when='a'	159	le tableau annotate sera le dernier à s'afficher
across	option	legend	across= <i>n</i>	92	<i>n</i> pavés de légende sur l'axe horizontal
across	option	pie	across= <i>n</i>	116	<i>n</i> graphiques sur l'axe horizontal par page
all	mot-clé	reset	reset=all	86	annule toutes les options locales d'environnement
all	option	gmap	all	94	toutes les unités surfaciques seront visualisées
and	mot-clé	data	<i>test condition1</i> and <i>condition2</i>	24	vrai si condition1 et condition2
angle	option	pie	angle= <i>valeur</i>	116	définition de l'origine de la première classe sur le cercle
annotate ou anno	option	gslide, choro	annotate= <i>tableau</i>	75, 150, 160, 164	introduit le nom du tableau contenant les informations graphiques à visualiser
areas	option	plot	areas= <i>n</i>	130	représentation surfacique des résultats
array	instruction	data	array <i>nom(variable)</i> <i>options</i>	65	tableau dont les colonnes correspondent aux modalités de la variable et les lignes aux observations du tableau d'entrée
axis	instruction		axis <i>n options</i>	75	paramètres de présentation des axes
b	mot-clé	when	when='b'	159	le tableau annotate s'affichera en premier
base	généralités			14	nom du répertoire correspondant à la librairie
bcolor	option	bubble	bcolor= <i>couleur</i>	126	couleur des contours de la représentation
block	instruction	gmap, chart	block <i>variable</i>	100, 110	cartographie des valeurs de la variable sur l'axe des z
border	option	goptions	border	74, 86	cadre autour de la sortie graphique

INDEX	TYPE	NIVEAU	SYNTAXE	PAGES	COMMENTAIRES
box plot	généralités			54	nom d'un diagramme de présentation de résultats
bscale	option	bubble	<i>bscale=mot-clé</i>	126	le mot-clé détermine qui, du rayon ou de l'aire du cercle, est proportionnel à la valeur
bsize	option	bubble	<i>bsize=valeur</i>	126	la valeur correspond à la taille maximale d'un cercle
bubble	instruction	gplot	<i>bubble var1*var2=var3</i>	126	représentation par des cercles proportionnels des modalités de la troisième variable aux intersections des deux premières variables
by	instruction	chart, sort, means, rank, gmap	<i>by variables</i>	53, 54, 56, 58, 96	un résultat par modalité pour les variables de la liste
c	option	title, pattern, legend <i>n</i> , value=, major=, minor=, label=	<i>c=couleur</i>	86, 92, 120, 124	abréviation de l'option color
call	instruction	data	<i>call sous-programme</i>	60	appel à un sous-programme
cards	instruction	data	cards	14	les lignes du programme qui suivent sont des données
carte numérique	généralités			77	carte sur support informatique
caxis	option	vbar, plot	<i>caxis=couleur</i>	108, 112	définit la couleur des axes du graphique
cback	option	goptions	<i>cback=couleur</i>	74	couleur de fond de la sortie
cbody	option	surface	<i>cbody=couleur</i>	104	couleur du « bloc 3d »
cells	mot-clé	htext	cells	74	l'unité courante est le pixel
cempty	option	choro	<i>cempty=couleur</i>	94	définit la couleur des contours des unités qui ne représentent pas la variable traitée
center	option	options	center	8	centrer les résultats
cfill	option	pie	<i>cfill=couleur</i>	116	couleur des contours

INDEX	TYPE	NIVEAU	SYNTAXE	PAGES	COMMENTAIRES
cfr	option	vbar	<i>cfr=couleur</i>	108	définit la couleur de fond du diagramme
cfreq	option	type=	<i>type=cfreq</i>	44	choix de la fonction fréquence cumulée
chart	étape		<i>proc chart</i>	44, 54, 108	représente les valeurs d'une variable sous forme de diagrammes
chisq	option	tables	<i>chisq</i>	48	effectue le test du Khi2
choro	instruction	gmap	<i>choro variables</i>	96	une représentation par plages pour chaque variable
class	instruction	means	<i>class variables</i>	53	un résultat par modalité pour les variables de la liste
clear	mot-clé		<i>clear</i>	7	nettoyer la fenêtre
cm	mot-clé	htext	<i>cm</i>	74, 134	l'unité courante est le centimètre
color	option	title	<i>color=couleur</i>	75, 86, 150	couleur du titre
color	variable	tableau annotate	<i>color=couleur</i>	75, 150	couleur associée à la fonction graphique
colors	option	goptions	<i>colors=liste</i>	74, 92, 116	réinitialisation de la palette de couleurs
config.sas	généralités			7	fichier définissant l'environnement de la session SAS
contents	étape		<i>proc contents</i>	10, 14	imprime les informations relatives à un tableau de données et la liste alphabétique des variables
courbe de la moyenne et valeurs à 2 écarts-type	généralités			134	titre d'un graphique
courbe de la moyenne, minimum et maximum	généralités			136	titre d'un graphique
coutline	option	choro	<i>coutline=couleur</i>	88	définit la couleur des contours des unités représentant la variable traitée

INDEX	TYPE	NIVEAU	SYNTAXE	PAGES	COMMENTAIRES
outline	option	vbar	outline= <i>couleur</i>	108	définit la couleur des contours du diagramme en bâton
cpct	option	type=	type=cpct	108	appliquer la fonction pourcentage cumulé à la variable
ctext	option	goptions, choro, vbar, pie	ctext= <i>couleur</i>	74, 86, 94, 108, 116	couleur des textes
cv	fonction SAS		cv (<i>variables</i>)	52	calcule le coefficient de variation des variables
cx	préfixe	généralités	cxrrggb	76	couleur en mode RGB
data	étape		data <i>tableau</i>	9	création et gestion de tableaux de données
data	option	proc	data= <i>nom</i>	9, 14, 79, 88, 142, 146	précède le nom du tableau d'entrée des données
date	option	options	date	8	afficher la date
delete	instruction	data	si <i>condition</i> then delete	24	annule le traitement sur l'observation courante
density	variable automatique	greduce		146	« poids » d'une observation dans le tableau de coordonnées
descending	option	means	descending	52	ordonne les valeurs dans le tableau de sortie selon un ordre décroissant
device	option	goptions	device= <i>nom</i>	74, 86	définition du périphérique de sortie
discrete	option	choro, block	discrete	88, 100	la variable traitée est de type discret
do over	instruction	data	do over <i>nom</i>	66	exécution des instructions de la boucle do sur un tableau
down	option	legend <i>n</i>	down= <i>n</i>	92, 108	<i>n</i> pavés de légende sur l'axe vertical
down	option	pie	down= <i>n</i>	116	<i>n</i> graphiques sur l'axe vertical par page
draw	mot-clé	function	'draw'	84	déplacement avec tracé

INDEX	TYPE	NIVEAU	SYNTAXE	PAGES	COMMENTAIRES
droite de régression linéaire	généralités			136	titre d'un graphique
drop	instruction	data	<i>drop liste</i>	18, 22	supprimer les variables de la liste dans le tableau de sortie
e	mot-clé	v ou value	<i>v=e</i>	90	abréviation de empty
éditeur	généralités			7	fenêtre où l'on introduit le programme
empty	mot-clé	v ou value	<i>v=empty</i>	90	pas de remplissage des unités graphiques
end	commande		<i>end</i>	7	fermer la fenêtre
endsas	commande		<i>endsas</i>	7	met fin à une session interactive
eq	opérateur de comparaison		<i>var1 eq var2</i>	26	test d'égalité
étape data	généralités			9, 73	étape de gestion des données
explode	option	pie	<i>explode=liste de modalités</i>	114	précise la représentation graphique de ces modalités
f	option	titlen, value	<i>f=police</i>	86, 92	abréviation de l'option ftext
fichier descriptif	généralités			13	fichier de données qualitatives et quantitatives
fichier géographique	généralités			13	fichier de coordonnées géographiques
file	commande		<i>file nom</i>	7	sauvegarder le contenu de la fenêtre dans le fichier <i>nom</i>
filename	instruction		<i>filename nom fichier</i>	14	associe le nom d'un tableau SAS et le fichier externe correspondant
fill	option	pie	<i>fill=mot-clé</i>	116	le mot-clé détermine le type de remplissage
firstobs	option	data= <i>nom</i>	<i>(firstobs=n)</i>	28	numéro de la première observation à traiter
firstobs	option	options	<i>firstobs=n</i>	8	numéro de la première observation à traiter

INDEX	TYPE	NIVEAU	SYNTAXE	PAGES	COMMENTAIRES
footnoten	instruction		footnoten 'chaîne'	74	équivalent à title mais se plaçant en bas de la sortie
format	instruction	data, gmap	format <i>nomvar nomformat.</i>	28, 88	applique la classification à la variable traitée
format	étape		proc format	28	définition des conversions à appliquer aux valeurs lors de l'écriture
frame	option	legend <i>n</i>	frame	92	encadrement de la légende
freq	instruction	means, univariate	freq <i>variable</i>	53, 54	pour chaque observation, la valeur de la variable est considérée comme un effectif
freq	étape	means	proc freq	26, 32, 48	calcul des effectifs et fréquences
freq	option	univariate	freq	53, 54	ajoute aux résultats les effectifs et les fréquences
ftext	option	goptions, titlen	ftext= <i>police</i>	74, 106	choix de la police de caractères
function	variable	tableau annotate	function= <i>mot-clé</i>	75, 84, 150	type d'action à exécuter sur le point de coordonnée (x,y)
fw	option	means	fw= <i>n</i>	52	largeur du champ d'impression des résultats
gaccess	option	goptions	gaccess= <i>nom</i>	154	précède le nom du fichier externe des résultats graphiques
ganno	étape		proc ganno	73	visualisation de tableaux annotate
gchart	étape		proc gchart	73, 106	représentation graphique de variables
gcontour	étape		proc gcontour	73	représentation en deux dimensions de trois variables
gdevice	étape		proc gdevice	73, 74	visualisation et modification des paramètres du périphérique graphique
ge	opérateur de comparaison		<i>var1 ge var2</i>	26	test de supériorité
gfont	étape		proc gfont	73	création de caractères

INDEX	TYPE	NIVEAU	SYNTAXE	PAGES	COMMENTAIRES
gmap	étape		proc gmap	73, 79, 86, 144 , 160	cartographie des résultats
goptions	étape		proc goptions	73	visualisation des paramètres de la session
goptions	instruction		goptions <i>options</i>	73, 74, 114	personnalisation de l'environnement
gplot	étape		proc gplot	73, 120	représentation en deux dimensions de deux variables
gprint	étape		proc gprint	73	habillage de sortie d'étape non graphique
gproject	étape		proc gproject	73	changement de projection
gray	préfixe	généralités	grayll	76	couleur en gamme de gris
greduce	étape		proc greduce	73, 146	préparation à la généralisation
gremove	étape		proc gremove	73, 142	création d'un nouveau fond de carte
greplay	étape		proc greplay	73	gestion d'une ou plusieurs sorties graphiques
group	option	pie	group= <i>variable</i>	114	un graphique par modalité de cette variable
groups	option	rank	groups= <i>n</i>	58, 106	définit le nombre de classes
gslide	étape		proc gslide	73, 150, 156, 164	visualisation de tableaux annotés
gt	opérateur de comparaison		<i>var1</i> gt <i>var2</i>	26	test de supériorité stricte
gtestit	étape		proc gtestit	73	visualisation des options graphiques
gunit	option	goptions	gunit= <i>mot-clé</i>	74	choix de l'unité courante
g3d	étape		proc g3d	73	représentation en trois dimensions de trois variables
g3grid	étape		proc g3grid	73	interpolation et lissage de données
h	option	title, value, major=, minor=, label=	h= <i>valeur</i>	92, 120, 124	abréviation de l'option htext
h	préfixe	généralités	hhhhllss	76	couleur en mode HLS

INDEX	TYPE	NIVEAU	SYNTAXE	PAGES	COMMENTAIRES
haxis	option	plot	<code>haxis=axisn</code>	122	l'axe horizontal sera représenté selon les options de <i>axisn</i>
hbar	instruction	chart	<code>hbar variables options</code>	44, 54	représente les valeurs des variables sous forme d'un diagramme horizontal
high	mot-clé	value	<code>high</code>	29	fait référence à la plus grande valeur d'un tableau
href	option	plot	<code>href=valeurs</code>	122	pour chaque valeur, tracer une ligne verticale
htext	option	goptions	<code>htext=valeur</code>	74, 90	choix de la hauteur de caractères
i	option	symbol	<code>i=mot-clé</code>	122, 128, 130, 134	le mot-clé détermine la méthode d'interpolation employée pour relier les points
id	instruction	means	<code>id variables</code>	53	ajoute au tableau de sortie les variables de la liste
id	instruction	gmap, gremove, greduce	<code>id variable</code>	79, 88, 142, 146	précède le nom de la variable qui permet de lier géographique et descriptif
if	instruction	data	<code>if conditions</code>	24	si vrai, alors traiter l'observation
if-then	instruction	data	<code>if conditions then instruction</code>	24	si vrai, alors exécuter l'instruction
if-then;else	instruction	data	<code>if condition then instruction1; else instruction2</code>	24	si vrai, alors exécuter l'instruction 1; sinon, l'instruction 2 sera exécutée
in	mot-clé	htext	<code>in</code>	74	l'unité courante est le pouce
include	commande		<code>include nom</code>	7	introduire le fichier externe <i>nom</i> dans l'éditeur
infile	instruction	data	<code>infile nom</code>	14	identifie le fichier externe lu par l'instruction input
information descriptive	généralités			78	information thématique
information géographique	généralités			78	information de localisation

INDEX	TYPE	NIVEAU	SYNTAXE	PAGES	COMMENTAIRES
input	instruction	data	input <i>nom</i>	13, 14, 20	description de l'enregistrement à traiter
int	fonction SAS		int(<i>argument</i>)	9	retourne la valeur entière de l'argument
j	mot-clé	i=	i=j	120	les points sont reliés par une droite
keep	instruction	data	keep <i>liste</i>	22	conserver les variables de la liste dans le tableau de sortie
keys	commande		keys	7	ouvre la fenêtre de définition des touches de fonction
l	option	symbol	l= <i>n</i>	120	le type de ligne dépend de la valeur <i>n</i>
label	instruction	data	label <i>variable</i> ='nom'	13, 18, 20	donner un nom à la variable
label	mot-clé	function	function='label'	163	l'action consiste à écrire du texte
label	option	legend <i>n</i>	label= <i>options</i>	92, 106	définit les paramètres du titre de la légende
label	option	data	label='chaîne'	75	introduit la chaîne de caractères à écrire
lag	fonction SAS		lag(<i>argument</i>)	150	mémorise la valeur précédente de l'argument
le	opérateur de comparaison		<i>var1</i> le <i>var2</i>	26	test d'infériorité
legend	option	choro, vbar, block	legend=legend <i>n</i>	74, 94, 108, 112	présentation de la légende selon les options définies dans legend<i>n</i>
legend<i>n</i>	instruction		legend <i>n options</i>	74	paramètres de présentation de la légende
length	instruction	data	length <i>nom longueur</i>	20 , 30	définit la longueur des variables
levels	option	vbar	levels= <i>n</i>	108	discrétise en <i>n</i> classes la variable à traiter
libname	instruction		libname <i>nom</i>	13, 14	permet d'associer la librairie SAS et le répertoire correspondant
librairie	généralités			7, 14	répertoire où se trouvent les fichiers permanents
librairie temporaire	généralités			7	répertoire où se trouvent les fichiers de travail générés par SAS

INDEX	TYPE	NIVEAU	SYNTAXE	PAGES	COMMENTAIRES
line	variable	tableau annotaté	<code>line=valeur</code>	75, 150	type de ligne
linesize	option	infile	<code>linesize=n</code>	20	limite la longueur de lecture de l'enregistrement
linesize	option	options	<code>linesize=n</code>	8	définit le nombre de caractères par ligne
list	option	tables	<code>list</code>	48	impression des résultats sous forme de liste
ln	mot-clé	i=	<code>i=ln</code>	122	méthode d'interpolation entre les points
lnp	mot-clé	i=	<code>i=lnp</code>	122	méthode d'interpolation paramétrique entre les points
log	commande			7	rend active la fenêtre visualisant les étapes du programme et les messages d'erreurs
low	mot-clé	value	<code>low</code>	29	fait référence à la plus petite valeur
lrecl	option	infile	<code>lrecl=longueur</code>	20	définit la longueur de l'enregistrement à lire
ls	option	options	<code>ls=n</code>	8	définit le nombre de caractères par ligne
lt	opérateur de comparaison		<code>var1 lt var2</code>	26	test d'infériorité stricte
major	option	axisn	<code>major=(options)</code>	120	définition des caractéristiques des graduations principales de l'axe
map	option	gmap	<code>map=tableau</code>	79, 88	désigne le tableau des coordonnées géographiques
max	fonction SAS		<code>max(liste)</code>	32, 52	retourne la plus grande valeur de la liste
maxdec	option	means	<code>maxdec=n</code>	52	définit la précision des résultats lors de l'impression
mean	fonction SAS		<code>mean(liste)</code>	32, 52	retourne la moyenne arithmétique de la liste
mean	option	output	<code>mean(liste)=noms</code>	54, 130	permet de définir les noms des variables contenant les résultats de la fonction statistique pour les variables de la liste
mean	option	type=	<code>type=mean</code>	108	appliquer la fonction « moyenne » à la variable

INDEX	TYPE	NIVEAU	SYNTAXE	PAGES	COMMENTAIRES
means	étape		proc means	52	statistiques descriptives de variables numériques
mean=	option	standard	mean= <i>valeur</i>	56	définit la valeur comme étant la moyenne des résultats
merge	instruction	data	merge <i>tableaux</i>	38, 96	fusionne les tableaux
midpoints	option	vbar, hbar	midpoints= <i>valeurs</i>	54	définit les valeurs centrales des classes
min	fonction SAS		min(<i>liste</i>)	9, 52	retourne la plus petite valeur de la liste
minor	option	axis <i>n</i>	minor=(<i>options</i>)	120	définition des caractéristiques des graduations secondaires de l'axe
missing	option	means, choro	missing	52, 96	traiter les « valeurs manquantes »
missing value	généralités			40	« valeur manquante »
missprint	option	tables	missprint	44, 48	imprime l'effectif des « valeurs manquantes »
mod	fonction SAS		mod(<i>valeur1,valeur2</i>)	32	reste de la division des deux valeurs
mode colonne	généralités	data		16	mode de lecture de données
mode format	généralités	data		16	mode de lecture de données
mode HLS	généralités			76	définit les composantes des couleurs
mode RGB	généralités			76	définit les composantes des couleurs
move	mot-clé	fonction	'move'	84	déplacement sans tracé
n	fonction SAS		n(<i>liste</i>)	32, 52	retourne le nombre de valeurs non manquantes d'une liste de valeurs numériques
n	option	major=, minor=	n= <i>valeur</i>	120	nombre de graduations sur un axe
n	option	surface	n= <i>valeur</i>	104	nombre de lignes du « bloc 3d »
ne	opérateur de comparaison		<i>var1</i> ne <i>var2</i>	26	test de différence
nmiss	fonction SAS		nmiss(<i>liste</i>)	32, 52	retourne le nombre de « valeurs manquantes » d'une liste de valeurs numériques

INDEX	TYPE	NIVEAU	SYNTAXE	PAGES	COMMENTAIRES
nocenter	option	options	nocenter	8	caler les résultats à gauche
nocol	option	tables	nocol	48	pas de calculs sur les colonnes du tableau
nodate	option	options	nodate	8, 14	ne pas afficher la date
nodms	option	config.sas	- nodms	8	rendre la session SAS non interactive
nofreq	option	tables	nofreq	48	pas d'impression des effectifs du tableau
nolegend	option	choro	nolegend	90	supprime l'affichage de la légende
none	mot-clé	label	label=none	92	pas de texte ou titre
nonumber	option	options	nonumber	8, 14	sans numérotation des pages
noprint	option	options, means, univariate	noprint	9, 52, 54	pas d'impression des résultats
normal	option	univariate	normal	54	visualise graphiquement la distribution normale
norow	option	tables	norow	48	pas de calculs sur les lignes du tableau
note	instruction		<i>note options 'chaîne'</i>	74	permet, dans une étape graphique, d'afficher des lignes de texte
number	option	options	number	8	avec numérotation des pages
nway	option	summary	nway	128	sélectionne les observations à traiter
obs	option	data= <i>nom</i>	(obs= <i>n</i>)	18, 28	numéro de la dernière observation à traiter
obs	option	options	obs= <i>n</i>	8	numéro de la dernière observation à traiter
options	instruction		options <i>options</i>	8, 14	modifier les paramètres de la session
or	opérateur de comparaison	data	<i>test condition1 or condition2</i>	24	vrai si condition1 ou condition2
order	option	axis <i>n</i>	order=(<i>valeurs</i>)	120	valeurs des graduations de l'axe
other	mot-clé	value	other	28	spécifie un ensemble de valeurs

INDEX	TYPE	NIVEAU	SYNTAXE	PAGES	COMMENTAIRES
out	option	data, sort, rank, gremove, greduce, means, univariate, standard, rank, gremove, greduce	out= <i>nom</i>	9, 46, 53, 54, 56 58, 142, 146	précède le nom du tableau de sortie
output	commande		output	7	rend active la fenêtre où s'affichent les résultats
output	instruction	data	output	155	provoque l'écriture de ce qui précède dans le tableau de données
output	instruction	means, univariate	output <i>options</i>	53, 54	les résultats seront conservés dans un tableau
overlay	option	plot	overlay	140	tracé de plusieurs distributions sur un même graphique
pagesize	option	options	pagesize= <i>n</i>	8	définit le nombre de lignes dans une page
pattern	instruction		pattern <i>n options</i>	75, 86 , 116	définition de représentations graphiques surfaciques
pct	mot-clé	htext	pct	74	l'unité courante s'exprime en pourcentage de la surface d'affichage
pctldef	option	univariate	pctldef= <i>valeur</i>	54	calcul des percentiles selon la méthode définie par la valeur
percent	option	type=	type=percent	44	appliquer la fonction « pourcentage »
percent	option	pie	percent= <i>mot-clé</i>	114	option d'écriture des pourcentages que représentent les subdivisions du cercle
pgm	commande		pgm	7	rend active la fenêtre éditeur où sera introduit le programme SAS à exécuter
pie	instruction	gchart	pie <i>variables</i>	114	représentation en « camembert » des valeurs des variables
plot	étape		proc plot	120, 132	représente graphiquement l'intersection des valeurs de deux variables

INDEX	TYPE	NIVEAU	SYNTAXE	PAGES	COMMENTAIRES
plot	option	gplot	plot (<i>var1*var2</i>)* <i>var3</i>	130	représente graphiquement les intersections de chacune des deux premières variables avec la troisième
plot	option	gplot	plot <i>variable1*variable2</i> = <i>variable3</i>	124	représente les intersections des valeurs des deux premières variables par les modalités de la troisième
plot	option	plot, gplot	plot <i>variable1*variable2e</i> = <i>symbol</i>	120, 132, 136	représente l'intersection des valeurs des deux variables par un symbole
plot	option	univariate	plot	54	visualise graphiquement les résultats
poly	mot-clé	function	function='poly'	150	début d'un polygone
polycont	mot-clé	function	function='polycont'	150	point courant d'un polygone
polygone	généralités			76	ligne polygonale fermée
position	option	contents	position	22	impression de la liste des variables en fonction de l'ordre interne dans le tableau
position	variable	tableau annotate	position= <i>valeur</i>	75, 159	définit la position du texte par rapport au point de coordonnées (x,y)
print	étape		proc print	10, 14, 16	imprime les observations d'un tableau de données
prism	instruction	gmap	prism <i>variables</i>	102	une carte par variable; les modalités sont représentées sur l'axe des z
proc	étape		proc <i>nom</i>	9	précède le nom d'une étape de traitement des données
ps	option	options	ps= <i>n</i>	8, 14	définit le nombre de lignes dans une page
put	instruction	data	put (<i>variable nomformat.</i>)	30, 65	écriture de la variable suivant la classification
range	fonction SAS		range(<i>liste</i>)	22	ordonne les valeurs numériques de la liste
range	fonction SAS		range(<i>liste</i>)	52	calcul l'étendue

INDEX	TYPE	NIVEAU	SYNTAXE	PAGES	COMMENTAIRES
rank	étape		proc rank	58	recode les observations suivant un ordre croissant
ranks	instruction	rank	ranks <i>noms</i>	58	permet de conserver les valeurs initiales des variables traitées
recall	commande		recall	7	recharge dans l'éditeur le programme précédent
régression cubique	généralités			138	titre d'un graphique
régression quadratique	généralités			138	titre d'un graphique
rename	instruction	data	rename <i>var=nom</i>	18	renomme les variables
répertoire courant	généralités			7	nom du répertoire où l'on travaille
reset	option	goptions	reset= <i>mot-clé</i>	86	annule les modifications d'environnement
retain	instruction	data	retain <i>variable valeur</i>	34	initialise la valeur de la variable
rotate	option	goptions	rotate= <i>mot-clé</i>	74	le mot-clé détermine l'orientation de la sortie dans la page
rotate	option	surface	rotate= <i>valeur</i>	104	angle de rotation du « bloc 3d » autour de l'axe des z
run	instruction		run	14	provoque l'exécution des étapes précédentes
s	mot-clé	v ou value	v=s	92	abréviation de solid
sas	commande		sas	7	ouvre une session interactive
sas	commande		sas <i>nomprog</i>	8	lance l'exécution du programme en mode non interactif
SAS#workxnnnn	généralités			7	nom du répertoire de travail de SAS
save	commande		save	7	sauvegarde le contenu de la fenêtre
segment	variable		segment	77, 79, 82	compteur des polygones d'une entité géographique

INDEX	TYPE	NIVEAU	SYNTAXE	PAGES	COMMENTAIRES
set	instruction	data	<i>set nom</i>	9, 22	définit les tableaux SAS à lire en entrée
sign	fonction SAS		<i>sign(valeur)</i>	34	retourne une valeur permettant de connaître le signe de la valeur
size	variable	tableau annoté	<i>size=valeur</i>	75, 150, 163	épaisseur de l'élément à traiter
slice	option	pie	<i>slice=mot-clé</i>	114	option d'écriture des valeurs moyennes des classes représentées dans le cercle
solid	mot-clé	value, fill	<i>v=solid</i> ou <i>fill=solid</i>	86, 114	remplissage des unités graphiques
sort	étape		proc sort	38, 46	tri des observations d'un tableau
spline	mot-clé	i=	<i>i=spline</i>	122	lissage entre les points par méthode cubique
sqrt	fonction SAS		<i>sqrt(valeur)</i>	32	retourne la racine carrée de la valeur
standard	étape		proc standard	56	centrage et réduction des observations
std	fonction SAS		<i>std(valeurs)</i>	52	calcule l'écart à la moyenne
std	option	out=	<i>std=noms</i>	56	permet de définir les noms des variables contenant le résultat de la fonction statistique
stderr	fonction SAS		<i>stderr (valeurs)</i>	52	calcule l'écart-type à la moyenne
std=	option	standard	<i>std=valeur</i>	56	définit la valeur comme étant l'écart-type
step	mot-clé	i=	<i>i=step</i>	120	représentation en « marche d'escalier »
stepc	mot-clé	i=	<i>i=stepc</i>	120	identique à step avec le point au centre de la marche
stepj	mot-clé	i=	<i>i=stepj</i>	120	représentation en « marche d'escalier » avec liaison verticale entre les marches
stepl	mot-clé	i=	<i>i=stepl</i>	120	identique à step avec le point à gauche de la marche
stepr	mot-clé	i=	<i>i=stepr</i>	120	identique à step avec le point à droite de la marche

INDEX	TYPE	NIVEAU	SYNTAXE	PAGES	COMMENTAIRES
style	variable	tableau annotat	<i>style=trame</i> ou <i>fonte</i>	75, 150	choix de la trame ou de la fonte de caractères
sub	commande		sub	7	provoque l'exécution d'une série d'instructions à partir de la fenêtre pgm
subgroup	option	vbar, block	subgroup= <i>variable</i>	108, 110	les modalités de cette variable sont visualisées sur le graphique
submit	commande		submit	7	provoque l'exécution d'une série d'instructions à partir de la fenêtre pgm
substr	fonction SAS		substr (<i>variable</i> , <i>position,n</i>)	32	extraction d'une sous-chaîne sur les valeurs de la variable
sum	fonction SAS		sum(<i>arguments</i>)	9	calcule la somme des arguments de la liste
sum	option	out=	sum= <i>noms</i>	53, 68	permet de définir les noms des variables contenant le résultat de la fonction statistique
sum	option	type=	type=sum	110	appliquer la fonction « somme » à la variable
summary	étape		proc summary	68	statistiques descriptives de variables numériques
surface	instruction	gmap	surface <i>variables</i>	104	une sortie graphique par variable sous forme de « bloc 3d »
symbol	instruction		symbol <i>n options</i>	74, 120, 128	définition de représentations graphiques ponctuelles
symbolgen	option	options	options symbolgen	92, 154, 162	permet d'afficher la valeur affectée à une macrovariable
symput	sous-programme SAS	data	call symput(<i>arguments</i>)	61	affecte une valeur à une macrovariable ou crée une macrovariable
sysin	option	sas	sas -sysin <i>nomprog</i>	8	lance l'exécution du programme en mode non interactif
t	option	value	t= <i>n 'chaîne'</i>	108	remplace dans la légende le libellé de la classe <i>n</i> par la chaîne de caractères

INDEX	TYPE	NIVEAU	SYNTAXE	PAGES	COMMENTAIRES
tableau annoté	généralités			73	tableau d'informations graphiques
tableau géographique	généralités			76	tableau de coordonnées géographiques
tableau graphique	généralités			76	tableau de coordonnées graphiques
tableau permanent	généralités			13	tableau SAS existant dans le répertoire associé au libname
tableau provisoire	généralités			13	tableau SAS n'existant pas dans le répertoire associé au libname
tables	option	freq	tables <i>liste</i>	26	traitement sur les variables de la liste
text	variable	tableau annoté	text='chaîne'	163	chaîne de caractère à écrire
tilt	option	surface	tilt= <i>valeur</i>	104	rotation du « bloc 3d » autour de l'axe des x
titlen	instruction		titlen <i>options</i> ;	8, 24, 74	titre qui se placera en haut de page sur la <i>nième</i> ligne
top	commande		top	7	réaffiche dans l'éditeur le début du programme et positionne le curseur
type	option	vbar, hbar, block	type= <i>mot-clé</i>	44, 108	définit la fonction à appliquer aux valeurs avant leur représentation
univariate	étape		proc univariate	54	statistiques descriptives de variables numériques
uss	fonction SAS		uss (<i>variables</i>)	52	calcule la somme non corrigée des carrés
value	instruction	format	value <i>nom valeurs</i> ='chaîne'	29	redéfinir le libellé des modalités
value	option	axis <i>n</i>	value=(<i>options</i>)	120	définition des paramètres des graduations de l'axe
v ou value	option	pattern	v= <i>valeur</i>	86	choix de la représentation graphique
v ou value	option	symbol	v= <i>mot-clé</i>	120	choix du symbole

INDEX	TYPE	NIVEAU	SYNTAXE	PAGES	COMMENTAIRES
value	option	legendn	value (<i>options</i>)	92	définition des paramètres des valeurs dans la légende
value	option	pie	value= <i>mot-clé</i>	116	option d'écriture des valeurs que représentent les subdivisions du cercle
var	fonction SAS		var (<i>variables</i>)	52	calcule la variance des variables
var	instruction	print, means, univariate, rank	var <i>variables</i>	26, 52, 54, 106	définit les variables à prendre en compte dans l'étape
vaxis	option	plot	vaxis=axisn	130	l'axe vertical sera représenté selon les options de axisn
vbar	instruction	chart	vbar <i>variables/options</i>	44, 106	représente les valeurs des variables sous forme d'un diagramme vertical
vref	option	plot	vref= <i>valeurs</i>	130	pour chaque valeur, tracer une ligne horizontale
weight	option	freq, means, univariate	weight <i>variable</i>	50, 53, 54	fait intervenir les effectifs de la variable dans le résultat du tableau croisé
when	variable	data	when= <i>mot-clé</i>	159	détermine le moment où s'affichera le contenu du tableau
where	option	data	where <i>conditions</i>	24	si vrai, alors traiter l'observation
work	généralités		work. <i>nom</i>	13	préfixe des tableaux provisoires SAS
x	variable		x	75, 77, 80	abscisse d'un point
xlast	variable automatique	tableau annotate	xlast	84	mémorise l'abscisse du point précédent
xsize	option	choro	xsize= <i>valeur</i>	98	longueur de la sortie graphique
xsy	variable	tableau annotate	xsy= <i>valeur</i>	75, 154, 158, 162	définit le système de référence sur l'axe des x
y	variable		y	75, 77, 80	ordonnée d'un point
ylast	variable automatique	tableau annotate	ylast	84	mémorise l'ordonnée du point précédent
ysize	option	choro	ysize= <i>valeur</i>	98	largeur de la sortie graphique

INDEX	TYPE	NIVEAU	SYNTAXE	PAGES	COMMENTAIRES
ysys	variable	tableau annotat	<i>ysys=valeur</i>	75, 154, 158, 162	définit le système de référence sur l'axe des y
FREQ	variable automatique	means, summary	<i>_freq_</i>	60, 68	effectif des classes de la partition
NUMERIC	variable automatique	data	<i>_numeric_</i>	66	concerne toutes les variables numériques
N	variable automatique	data	<i>_N_ ou _n_</i>	34	numéro d'ordre d'observations dans un tableau
TYPE	variable automatique	means, summary, if	<i>_type_</i>	60, 68, 124	nombre de partitions des variables
!!	opérateur de concaténation		<i>chaîne1!!chaîne2</i>	32	concatène les deux chaînes
#	caractère spécial	data	<i>#n</i>	20	définit l'enregistrement à lire
\$	caractère spécial	data	<i>nomvar \$</i>	14	définit une variable alphanumérique
%macro	instruction	macroprogramme	<i>%macro nom (arguments)</i>	94, 154, 162	début d'un sous-programme
%mend	instruction	macroprogramme	<i>%mend nom</i>	94, 154, 162	fin d'un sous-programme
(espace)	séparateur			9	sépare les options au niveau étape ou instruction
*	opérateur arithmétique		<i>val1*val2</i>	32	multiplication
*	séparateur	plot	<i>variable1*variable2</i>	120	tableau croisé
**	opérateur arithmétique		<i>val1**val2</i>	32	puissance
+	opérateur arithmétique		<i>val1+val2</i>	32	somme
-	opérateur arithmétique		<i>val1-val2</i>	32	différence
-	caractère spécial	<i>liste de variables</i>	<i>var1-varn</i>	22	définit un ensemble de variables
--	caractères spéciaux	<i>liste de variables</i>	<i>var1--varn</i>	22	définit un ensemble de variables
.	caractère spécial		<i>.</i>	40	correspond à « valeur manquante »
.log	généralités		<i>nomprog.log</i>	8	suffixe du fichier décrivant les étapes en session non interactive

INDEX	TYPE	NIVEAU	SYNTAXE	PAGES	COMMENTAIRES
.lst	généralités		<i>nomprog.lst</i>	8	suffixe du fichier des résultats en session non interactive
.ssd	généralités		<i>nom.ssd</i>	13	suffixe de tableau permanent de la librairie
/	séparateur			9	sépare les options au niveau instruction
/	opérateur arithmétique		<i>val1/val2</i>	32	division
/* */	séparateurs			10	encadre un commentaire
;	séparateur		;	9	indique la fin d'une étape ou d'une instruction
<	opérateur de comparaison		<i>var1<var2</i>	26	test d'infériorité stricte
<=	opérateur de comparaison		<i>var1<=var2</i>	26	test d'infériorité
=	opérateur de comparaison		<i>var1=var2</i>	26	test d'égalité
>	opérateur de comparaison		<i>var1>var2</i>	26	test de supériorité stricte
>=	opérateur de comparaison		<i>var1>=var2</i>	26	test de supériorité
@	caractère spécial	data	@ <i>n</i>	20	pointeur de lecture dans l'enregistrement

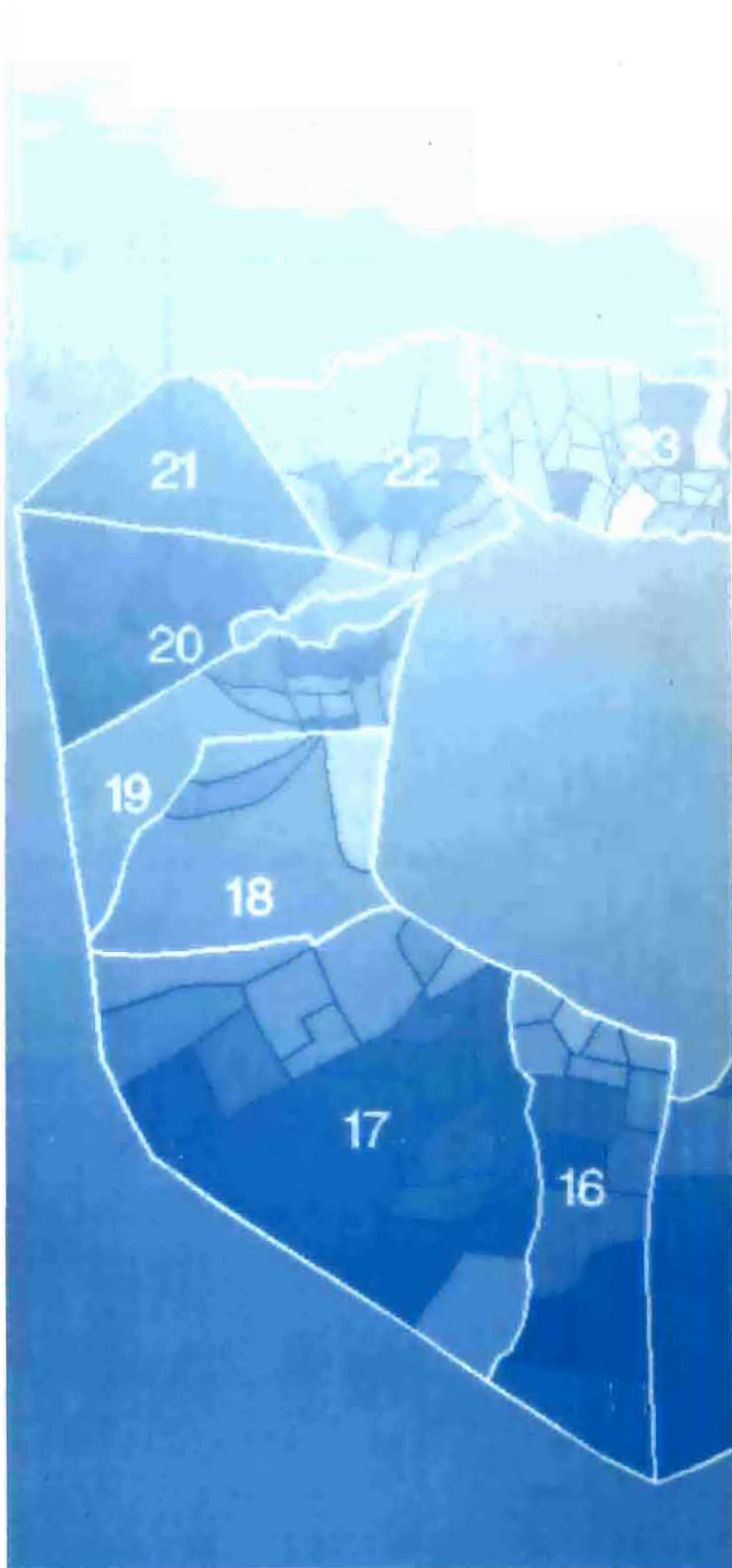
TABLE DES MATIÈRES

	Pages
L'ENVIRONNEMENT DE TRAVAIL AVEC SAS	7
I. ETAPE DATA	11
I.A LES TABLEAUX SAS : INTRODUCTION ET LECTURE DES DONÉES	13
I.A.1 Lecture des données	14
I.A.1.1 Lecture d'un fichier externe	14
I.A.1.2 Les différents modes de lecture possibles d'un fichier externe	14
I.A.2 Lecture d'un fichier externe dont les données sont sur plusieurs enregistrements	18
I.A.2.1 Premier exemple	18
I.A.2.2 Exemple avec positionnement en colonne	20
I.B SÉLECTION	22
I.B.1 Sélection de variables	22
I.B.2 Sélection d'observations	24
I.C RECODAGE	26
I.C.1 Recodage simple conditionnel	26
I.C.2 Recodage par classification	28
I.C.2.1 Variable numérique	28
I.C.2.2 Variable alphanumérique	28
I.C.2.3 Création de nouvelles variables après recodage	30
I.C.3 Recodage par calcul	30
I.D CRÉATION DE NOUVELLES VARIABLES	34
I.D.1 Initialisation d'une nouvelle variable	34
I.D.2 Création d'une nouvelle variable à partir de conditions sur les variables existantes	36
I.D.3 Création d'une nouvelle variable par calcul sur les variables existantes	36
I.E GESTION DE TABLEAUX	38
I.E.1 Fusion de tableaux SAS	38
I.E.2 Concaténation de tableaux	40

	Pages
II TRAITEMENTS STATISTIQUES	43
II.A DESCRIPTION STATISTIQUE DE VARIABLES QUALITATIVES	44
II.A.1 Effectifs et fréquences simples	44
II.A.2 Effectifs et fréquences en fonction d'une partition	46
II.A.3 Tris croisés sur variables qualitatives	48
II.B DESCRIPTION STATISTIQUE DE VARIABLES QUANTITATIVES	52
II.B.1 Indices simples : proc means	52
II.B.2 Etude statistique complète avec diagramme et graphique : proc univariate	54
II.C CLASSIFICATION A PARTIR DE CALCULS STATISTIQUES	56
II.C.1 Centrage/réduction (écart-type)	56
II.C.2 Classes d'égal effectif (quantiles, médiane)	58
II.C.3 Classes d'égale étendue (max, min, mode)	60
III CHANGEMENT D'UNIVERS D'ANALYSE : AGRÉGATION	63
III.A CONSTRUCTION D'UN TABLEAU DISJONCTIF COMPLET	64
III.B AGRÉGATION DU TABLEAU DISJONCTIF COMPLET PAR UNITÉS GÉOGRAPHIQUES	68
IV SAS/GRAPH	71
GÉNÉRALITÉS SUR SAS/GRAPH	73
IV.A INTRODUCTION DE COORDONNÉES GÉOGRAPHIQUES	80
IV.A.1 Unités spatiales ponctuelles	80
IV.A.2 Unités spatiales surfaciques (polygones)	82
IV.A.3 Unités spatiales linéaires (arcs)	84

	Pages
IV.B REPRÉSENTATION GRAPHIQUE DES DONNÉES	86
IV.B.1 Cartographie en deux dimensions : proc gmap option choro	86
IV.B.1.1 Représentation choroplèthe ou par plages	86
IV.B.1.2 Représentation des limites des unités surfaciques	90
IV.B.1.3 Cartographie d'une variable descriptive selon les modalités d'une autre variable	92
IV.B.1.4 Cartographie d'une variable descriptive selon les modalités d'une autre variable : option by de choro	96
IV.B.2 Cartographie en trois dimensions : proc gmap options block, prism ou surface	100
IV.B.2.1 Représentation en trois dimensions option block	100
IV.B.2.2 Représentation en trois dimensions option prism	102
IV.B.2.3 Représentation en trois dimensions option surface	104
IV.B.3 Représentations graphiques de distributions simples : proc gchart	106
IV.B.3.1 Diagrammes en bâton : proc gchart options vbar ou hbar	106
IV.B.3.2 Diagrammes en « block » : proc gchart option block	110
IV.B.3.3 Diagrammes en cercles : proc gchart option pie	114
IV.B.4 Représentation graphique des distributions simples : proc gplot	118
IV.B.4.1 Distributions simples à deux variables	118
IV.B.4.2 Distributions simples à trois variables	124
IV.B.4.3 Représentation de distributions simples par des aires	128
IV.B.4.4 Représentation de deux distributions simples sur un même graphique	130
IV.B.4.5 Représentation graphique d'un nuage de points	132
IV.B.4.6 Calcul et représentation graphique de courbes associées	134
IV.C CRÉATION DE NOUVEAUX TABLEAUX GRAPHIQUES OU GÉOGRAPHIQUES	142
IV.C.1 Création d'un nouveau fond de carte à partir du fond de carte de la base	142
IV.C.2 Généralisation d'un fond de carte	146
IV.C.3 Création d'un tableau graphique à partir d'un tableau géographique existant	150
IV.C.4 Création d'un tableau graphique sans données initiales : palette de gris	154
IV.D EXEMPLE : REPRÉSENTATION SPATIALE D'UN PLAN FACTORIEL	158
IV.D.1 Cartographie du plan factoriel	158
IV.D.2 Plan factoriel : légende associée à la carte	162
INDEX	167

ORSTOM Éditeur
Dépôt légal : mars 1994
Microédition Impression
ORSTOM BONDY



*Couverture :
Spatialisation de l'analyse
inter-classe sur les zones
de dénombrement.
Ouagadougou,
(Burkina Faso).*

ORSTOM Éditions
213, rue La Fayette
F-75480 Paris Cedex 10

Diffusion
72, route d'Aulnay
F-93143 Bondy Cedex
ISSN : 1142-2580
ISBN : 2-7099-1179-5

