La Gestion des données dans le système SAS sous Windows v6.12

N.H. Fonton[1] , J.J. Claustriaux[2]

Résumé
La gestion des données dans le système SAS est une des difficultés majeures rencontrées par les utilisateurs de ce logiciel. Dans cette note, la mise à jour des données est présentée après un bref aperçu sur l'organisation et la création des données. Des procédures simples d'analyse statistique sont aussi présentées.

Summary: The data management with SAS system 6.12
The management of data is the major difficulties of the users in the SAS system. The note presents the procedures of data management. after a brief summary on the organization and creation of data. Some procedures for statistics analysis are also presented.

1. INTRODUCTION

Le logiciel SAS (*Statistical Analysis System*) est conçu en 1966 par SAS Institute en Caroline du Nord, aux Etats-Unis. Il est composé d'un ensemble de logiciels intégrés, ce qui lui vaut l'appellation de système dont le cœur est le logiciel ***Base SAS*** (SAS de base). Ce système est conçu sous format ou plate-forme micro-ordinateur,

[1] Maître-Assistant à la Faculté des Sciences Agronomiques, Université d'Abomey-Calavi, Bénin
[2] Professeur ordinaire à la Faculté Universitaire des Sciences Agronomiques de Gembloux, Belgique

station de travail, mini et maxi-ordinateur. Il se présente sous les systèmes d'exploitation Windows, UNIX, OS/2 et Open VMS Alpha pour la plupart des modules.

Depuis ses premières versions, ce logiciel a connu beaucoup d'améliorations, notamment la version Windows. Actuellement, il est disponible sur plate-forme PC avec nouvelle version, la 6.12.

Le logiciel SAS se caractérise par un système intégré dans la réalisation de quatre tâches fondamentales dont les données sont au centre (X, 1996a) à savoir :

- accès rapide aux données quel que soit le format dans lequel elles existent;
- gestion des données (mise à jour, combinaison, réaménagement, sélection);
- analyse des données conduisant à une aide appréciable de décision dont les outils sont assez variés depuis la statistique descriptive jusqu'à la statistique avancée ou spécialisée;
- bonne présentation de tableaux et des graphiques.

Pour réaliser ses différentes tâches, plus de 30 modules spécifiques ou logiciels sont disponibles. Ceux-ci sont en relation avec les différents domaines d'application de la statistique de part leur spécificité, d'une part, et à travers le développement de nouvelle théorie de traitement de l'information quantitative, d'autre part. Parmi ces modules, nous avons :

- SAS/ACCESS : Accès aux systèmes de gestion des bases de données ;
- SAS/STAT : Analyse statistique de base et avancée ;
- SAS/ETS : Traitement de données en séries chronologiques ;
- SAS/QC : Outils statistiques pour le contrôle de qualité ;
- SAS/OR : Recherche opérationnelle (gestion de projets et optimisation) ;
- SAS/GRAPH : Graphiques et cartes personnalisées à deux ou trois dimensions ;

- SAS/FSP : Visualisation et édition des données en mode écran, y compris le contrôle en temps réel ou différé ;
- SAS/AF : Conception d'applications conviviales (menus déroulants, icônes, boîtes de dialogues, etc.).

Quel que soit le module, ci-dessus énuméré, le module SAS/BASE est indispensable pour son fonctionnement.

Pour le management de l'information, la gestion des données constitue une phase très importante. Mais avant d'aborder celle-ci (paragraphe 3), nous présenterons l'organisation des données dans le langage SAS notamment dans ses principes généraux, adressage et nature des données (paragraphe 2). Viennent ensuite la mise à jour des données ou des tables SAS (paragraphe 4) et les concepts généraux des procédures d'analyse statistique avec SAS (paragraphe 5).

2. ORGANISATION DES DONNEES DANS LE SYSTEME SAS

2.1. Principes généraux du langage SAS

Le langage SAS nécessite des données lues et éventuellement transformées en tenant compte de l'objectif. Ainsi avant de procéder à l'analyse ou à l'édition d'un rapport, les données doivent être dans le format du système SAS, appelé ***Données SAS***. Comme le montre la figure 1, les données de départ sont soit des lignes de données soit dans le format d'un gestionnaire de base de données ou soit des données SAS.

Pour créer un fichier SAS, on dispose de trois possibilités. Il s'agit de l'étape DATA, de la procédure SQL et l'usage du logiciel SAS/ACCESS. Chacune de ses possibilités est utilisée en fonction du format dans lequel se trouve les données dont on dispose.

L'étape DATA est utilisée lorsque les données de départ sont des lignes de données (à saisir ou existant dans un fichier ASCII) ou des données SAS. La procédure SQL (*Structured Query Language*), quant à elle, permet de lire ou d'extraire les données du ou des fichiers SAS. Pour des fichiers provenant d'autres

logiciels, notamment les gestionnaires de base de données, le logiciel SAS/ACCESS permet d'exporter l'information et de créer des fichiers SAS ou des données SAS.

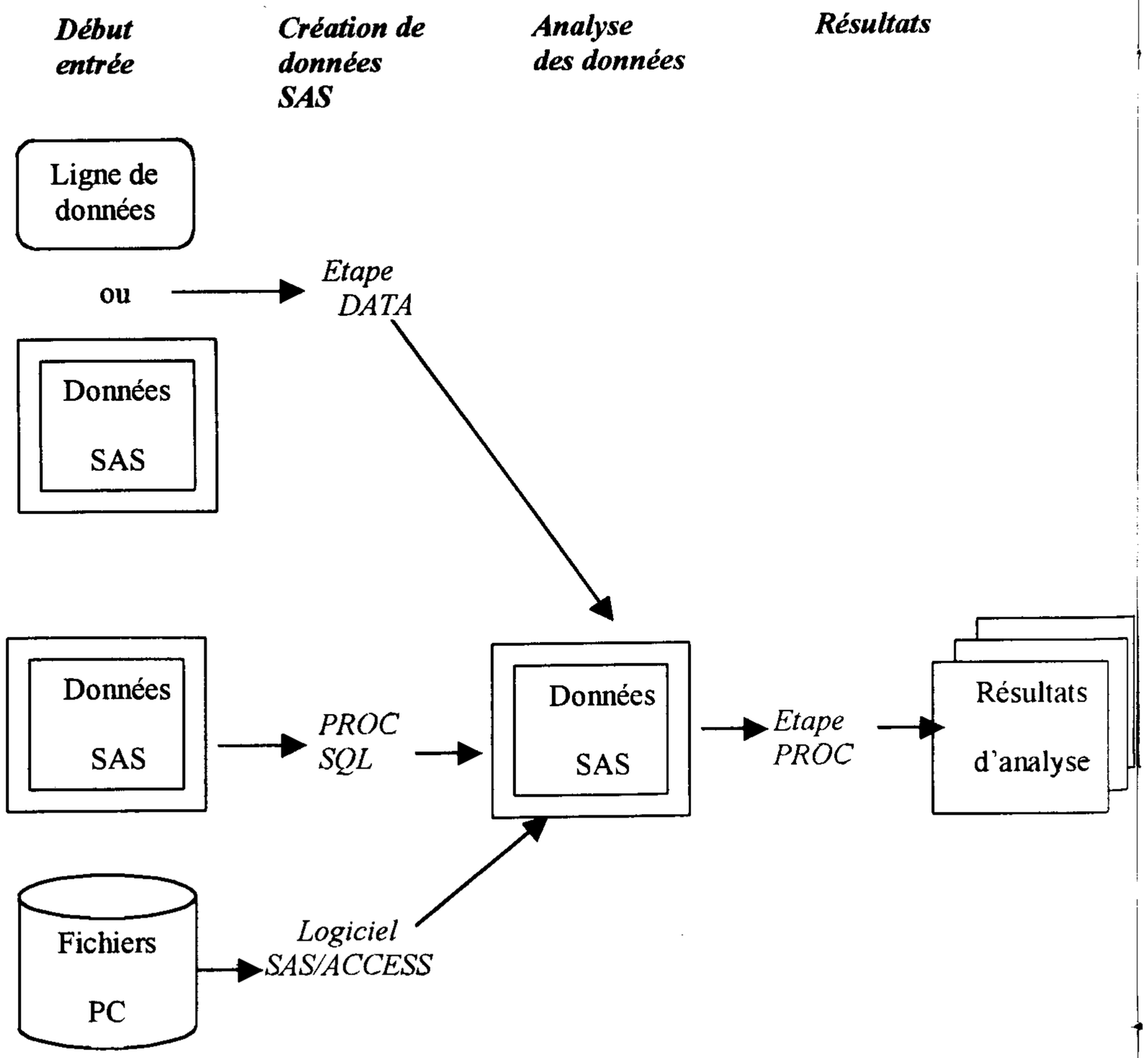

Figure 1. Différents processus de création de données SAS

Le format de données SAS donne lieu à une structure qui se caractérise par un descripteur d'informations et la valeur des

données. Le descripteur d'informations est transparent à l'utilisateur et consiste à décrire le contenu des données SAS pour le système. La valeur des données sont les données introduites ou calculées. Elles sont soit numérique, alphanumérique, logique ou date. L'information dans le système SAS se présente sous forme rectangulaire (X, 1996a) composée de colonnes et de lignes appelées respectivement variables et observations (figure 2). Même si les valeurs sont incomplètes, la forme rectangulaire est maintenue car le système SAS reconnaît les données manquantes représentées, par défaut, par un point.

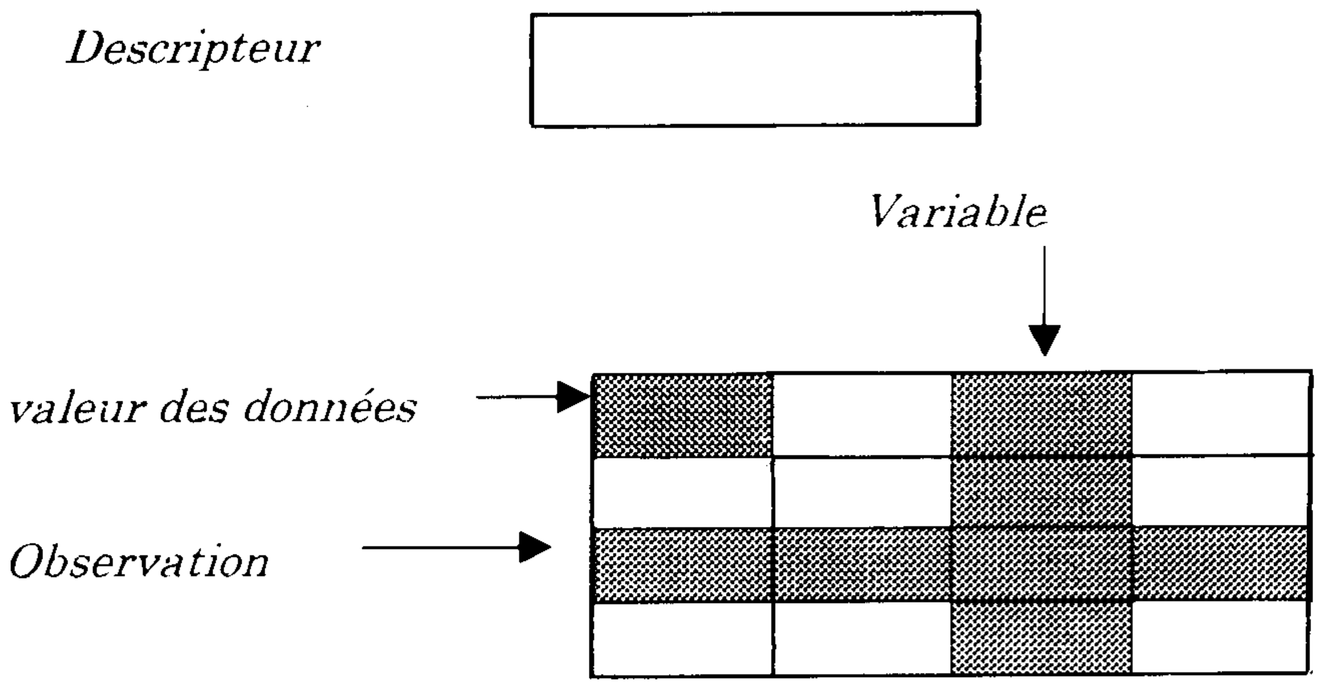

Figure 2. Représentation de l'information dans le système SAS

L'analyse des données, ainsi créées, est réalisée au cours des étapes PROC (figure 1) qui permettent l'utilisation des procédures en fonction des résultats attendus. Elle sera abordée dans ce document par des exemples d'analyse descriptive et inférencielle générale.

Pour la maîtrise de la gestion des données, l'adressage et la nature des données sont exposés dans les paragraphe 2.2 et 2.3.

2.2. Adressage des données SAS

Les données SAS sont stockées dans des fichiers portant chacun un nom. Dans le système SAS, le nom de fichier est régi par les règles essentielles suivantes :

- le nombre de caractères est inférieur ou égal à huit ;
- le nom de fichier ne doit pas inclure de symbole spécial tel que / ;
- le nom doit toujours commencer par un caractère alphabétique.

On distingue deux types de fichiers SAS: les fichiers permanents et les fichiers temporaires. La différence entre ces deux fichiers réside dans le répertoire de référence et la nécessité de le spécifier ou non .

Les fichiers temporaires sont situés dans le répertoire temporaire WORK de C:\SAS. Ce dernier est utilisé pour développer et tester de nouveaux programmes ou pour stocker des données intermédiaires. A la fin de la session, les fichiers de ce répertoire sont systématiquement effacés.

Quant aux fichiers permanents, ils sont stockés dans le sous-sous répertoire SASUSER de C:\SAS par défaut ou enregistrés dans tout autre répertoire ou unité de stockage spécifié par l'utilisateur.

Le nom de tout fichier permanent se compose de deux parties :

- le *libref* ou répertoire de référence ;
- le ***nom du fichier***.

Ainsi, pour le fichier permanent RENDT et pour le fichier ANIMAL du répertoire ETU98 de C:\ (disque dur), leurs désignations sont respectivement:

```
SASUSER.RENDT
C:\ETU98.ANIMAL
```


Par contre, le fichier temporaire HOUSES est désigné simplement par:

```
HOUSES
```

2.3. Types des données

Données numériques

Les données numériques sont désignées soit par la notation standard, soit par la notation scientifique ou soit par la notation hexadécimale.
La plupart des données sont désignées par la notation standard. Celle-ci est exprimée en donnée entière, précédée ou non du signe + ou – et peut inclure le signe décimal représenté par un point comme dans les exemples ci-après :

- 25 pour désigner un nombre entier ;
- 5.36 pour désigner un nombre avec décimal ;
- 01 pour désigner la valeur 1 ;
- -9 pour désigner un nombre entier négatif.

Lorsque la valeur numérique dépasse $10^{32} - 1$, on recourt à l'utilisation de la notation scientifique qui se caractérise par l'usage des puissances de 10 représentées par le symbole E. Le nombre placé avant le E est multiplié par la puissance de 10 du nombre qui vient après E. On peut donc écrire:

- 1.5 E23 qui correspond à $1{,}5 \times 10^{23}$;
- 0.02E15 qui vaut $0{,}02 \times 10^{15}$.

Données caractères

Les données caractères sont un ensemble de caractères alphabétiques. Toutefois, il faudra faire attention en débutant une donnée caractère par un blanc. De même, lorsque les données contiennent des points virgule, la commande CARDS4 est utilisée au lieu de CARDS et la fin des données est symbolisée par quatre points virgules comme présentée dans le paragraphe 3.2.

Données manquantes

Les données manquantes sont désignées soit par un blanc, soit par un point soit par un caractère spécial. Dans ce dernier cas, il faut inclure entre la commande DATA et la commande INPUT, la commande MISSING qui précise ces caractères spéciaux désignant les données manquantes.

Exemple 1. Spécification des caractères ***c*** et ***x*** désignant les données manquantes dans le fichier à créer REND.

```
DATA LSAS.REND;
MISSING c x;
INPUT Dep $ Village $ Spec $ Rendt;
CARDS;
Mono    Lalo    Manioc   635
Atlantique    Glo Manioc   c
Atlantique    Djigbe    x    c
Atlantique    Glo    x    475
.
.
.
;
RUN;
```

3. CREATION DE DONNEES SAS

3.1. Ouverture d'une session SAS sous Windows: Présentation générale

L'icône du bureau de l'ordinateur qui permet de lancer une session SAS est :

Le lancement de cette session peut être effectué aussi par le menu ***Démarrer*** – ***Programmes*** – ***The SAS System*** – ***The SAS System for Windows v6.12***.

Deux fenêtres apparaissent à l'écran selon la configuration. La première, la fenêtre LOG ou fenêtre de journal d'exécution, occupe par défaut, la première moitié supérieure de l'écran. La seconde, la fenêtre PROGRAM EDITOR ou fenêtre d'édition de programme, se trouve dans la deuxième moitié inférieure de l'écran comme le montre la figure 3.

La fenêtre LOG fournit le code source SAS et notifie :

- les messages d'erreurs qui ne bloquent pas l'exécution du programme dans la rubrique WARRING ;
- les messages d'erreurs qui bloquent l'exécution du programme dans la rubrique ERROR ;
- les informations sur les données et le temps d'exécution du programme dans la rubrique NOTE.

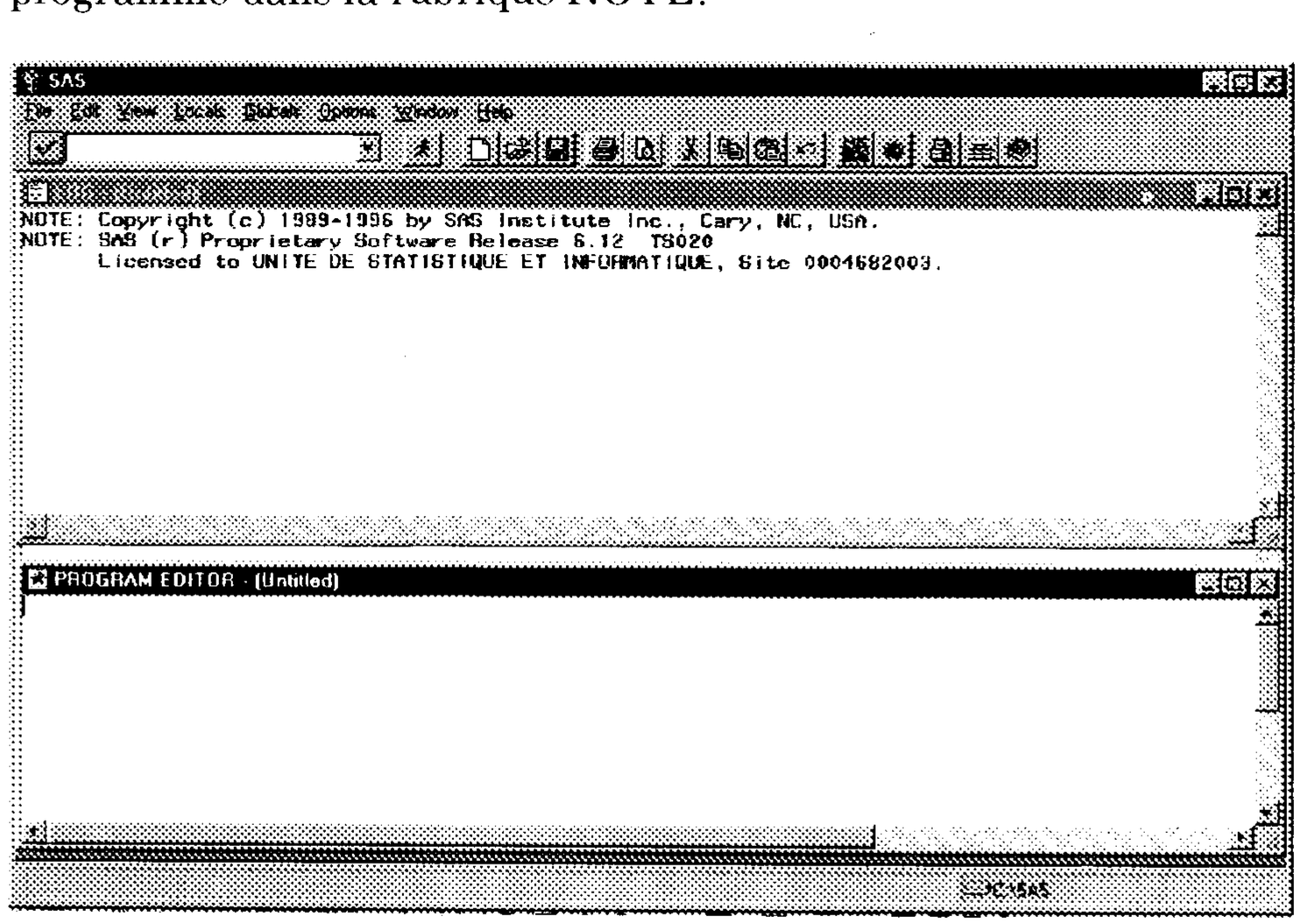

Figure 3. Ecran de lancement de SAS : Fenêtres LOG et PROGRAM EDITOR

Quant à la fenêtre PROGRAM EDITOR, elle permet l'édition des programmes SAS et dispose de fonctionnalités d'un éditeur de texte comme COPIER, COLLER, DEPLACER, INSERER, ENREGISTRER et OUVRIR. Elle possède aussi les fonctionnalités essentielles pour l'utilisateur à savoir :

· SUBMIT : permet l'exécution du programme édité ;

· RECALL TEXT : permet de rappeler le dernier programme exécuté ;

· SUBTOP *n* : permet l'exécution des procédures correspondantes aux *n* premières lignes du programme.

Une troisième fenêtre permet la restitution des résultats. Il s'agit de la fenêtre OUTPUT. Elle dispose aussi dans son menu OPTION des possibilités de mise en forme du texte des résultats, notamment la police, la taille et le style comme le montre la figure 4.

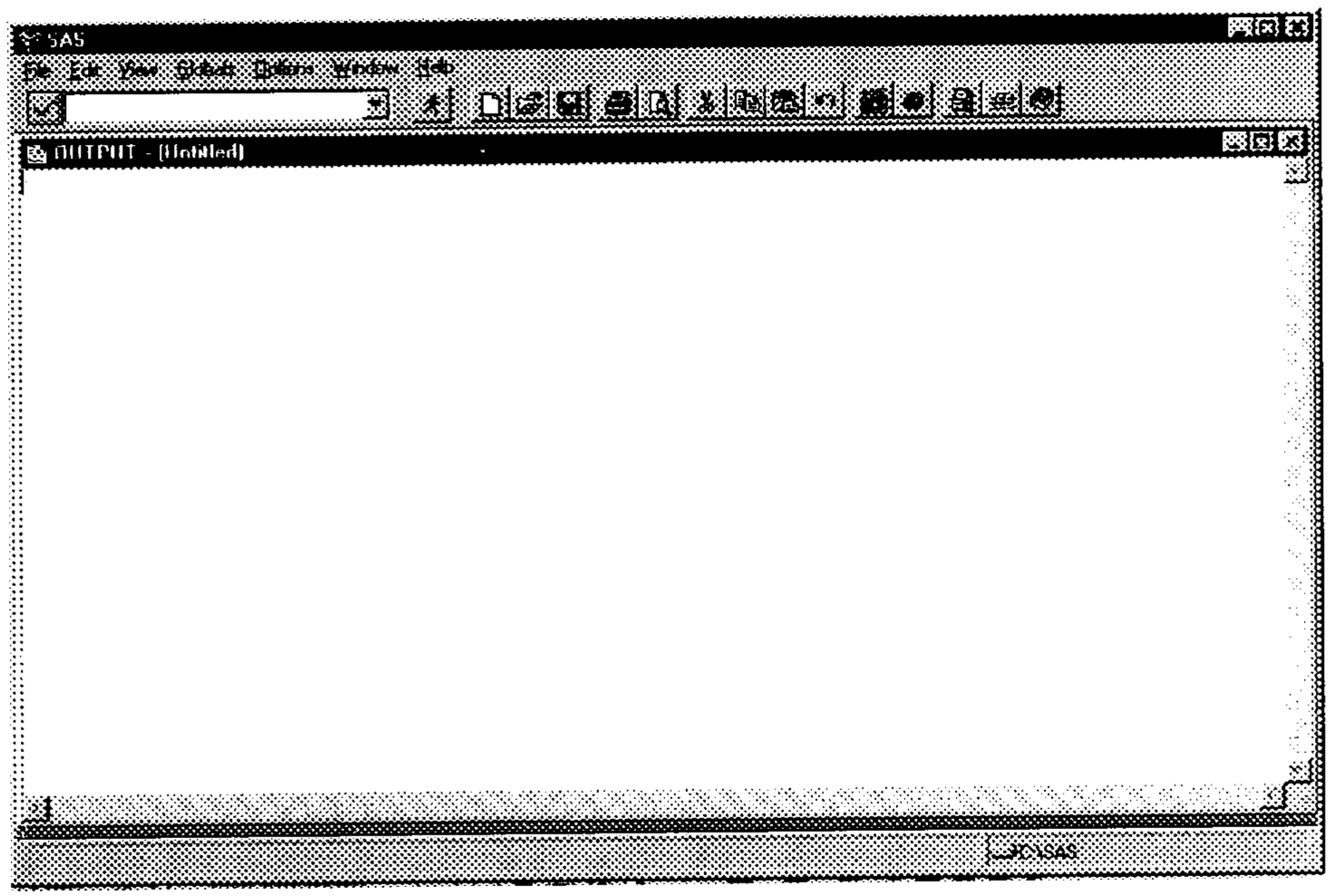

Figure 4. Ecran de lancement de SAS : Fenêtre OUTPUT

3.2. Création de fichier SAS à partir des Données saisies directement

Les différentes commandes de l'étape DATA pour créer un fichier SAS à partir des données saisies directement dans la procédure SAS sont les suivantes :

DATA :
Marque le début de l'étape DATA et permet de spécifier le nom du fichier SAS à créer.

INPUT :
Permet de spécifier le nom de chaque variable et sa localisation correspondant aux colonnes occupées (ce dernier n'est pas nécessaire). On peut aussi créer de nouvelles variables.

CARDS ou CARDS4 :
Marque le début des données.

Lignes de données

; ou ; ; ; ;
Marque la fin des lignes de données respectivement pour CARDS ou CARDS4.

RUN ;
Exécution de la procédure.

Dans l'exemple 2, nous donnons la procédure de création d'un fichier SAS avec des données directement saisies sans spécification des colonnes (a). Parfois, avec la saisie directe des données, les colonnes correspondantes à chaque variable sont précisées. Cette précison est donnée dans la commande INPUT avec les noms des variables précédées des rangs des colonnes qu'elles occupent (b).

Exemple 2 a : Création d'un fichier de données SAS, dont le nom est DENDRO dans le répertoire C:\FORM

```
LIBNAME LSAS 'C:\FORM' ;
DATA LSAS.DENDRO ;
INPUT Foret $ Age Hd Densite St ;
CARDS ;
     Djigbé    32  27.2  188  20.5
     Djigbé    37  28.9  133  19.0
     Agrimey   37  24.7  96   16.3
     Toffo     40  26.4  200  26.6
     Massi     7   13.7  425  14.4
     Agrimey   44  28.7  150  23.4
     ;
     RUN ;
```

Exemple 2 b : Création du fichier de données SAS DENDRO dans le répertoire C:\FORM avec précisiondes rangs des colonnes des variables

```
LIBNAME LSAS 'C:\FORM' ;
DATA LSAS.DENDRO
INPUT  Foret  $  1-8  Age  10-12  Hd  14-17
Densite 19-21 St 23-27;
CARDS ;
     Djigbé    32  27.2  188  20.5
     Djigbé    37  28.9  133  19.0
     Agrimey   37  24.7  96   16.3
     Toffo     40  26.4  200  26.6
     Massi     7   13.7  425  14.4
     Agrimey   44  28.7  150  23.4
     ;
     RUN ;
```

3.3. Création de fichier SAS à partir des données d'un fichier ASCII

Les commandes utilisées pour créer un fichier SAS à partir d'un fichier ASCII (extension . DAT) sont les suivantes :

DATA :
début de l'étape DATA et permet de spécifier le nom du fichier SAS à créer

INFILE :
identifie le fichier externe ASCII contenant les données en précisant son répertoire

INPUT :
permet de donner un nom à chaque variable et identifie sa position dans l'enregistrement des données. De plus, cette commande offre la possibilité de modifier les données, de créer de nouvelles variables, etc.

Exemple 3 : Création d'un fichier SAS du nom de DENDRO1 à partir du fichier ASCII DENDRO. DAT du répertoire FORM dont les variables sont *Foret*, *Age*, *Hd*, *Densite*, et *St* puis une nouvelle variable *St1* générée à partir de *St* et *Densité*.

```
LIBNAME LSAS 'C:\FORM' ;
DATA LSAS.DENDRO1 ;
INFILE 'C:\FORM\DENDRO.DAT';
INPUT Foret $ Age Hd Densite St ;
RUN ;
```

De la même manière, comme pour la création d'un fichier SAS à partir des données directement saisies, les numéros de colonnes peuvent être spécifiés dans la commande INPUT.

3.4. Création de données SAS à partir d'un format PC

Le logiciel SAS/ACCESS permet une interface entre le système SAS et les données d'autres formats provenant des logiciels de gestion de base de données, DbMS (Database Management System) tels que les formats DBF, DIF, WK1, Wk3, WK4 et XLS

Deux procédures sont utilisées à savoir la procédure d'entrée ACCESS et la procédure DBLOAD.
La procédure d'entrée, ACCESS, permet :

- de créer les fichiers de description des formats DbMS ou PC en système SAS ;
- de créer un fichier de données SAS.

La procédure DBLOAD permet de convertir les fichiers de données SAS ou autres types de données dans le format DbMS.

Une autre procédure est l'interface *view engine* avec laquelle les données des formats DbMS ou PC peuvent être lues et mises à jour.

- ***Création de fichier SAS à partir des formats DbMS***

La procédure permettant de créer un fichier SAS à partir d'un format DbMS ou PC est PROC ACCESS. La syntaxe se présente en trois groupes de commandes comme suit :

PROC ACCESS <option-descripteur-access>/option-descripteur-view ;

CREATE répertoire.nom-fichier.descripteur ;

PATH=répertoire et nom-fichier – extension ;

ASSIGN
DROP
RENAME
FORMAT
RENAME
LIST
SUBSET

RUN ;

Les spécifications en minuscule italique sont fournies par l'utilisateur. Elles se définissent comme suit :

PROC ACCESS descripteur-options : option du descripteur ACCESS qui permet de spécifier le format du fichier PC comme suit :

PROC ACCESS DBMS = DBF | DIF |XLS| Wkn

Ainsi pour un format DBF, l'écriture est la suivante :

PROC ACCESS DBMS = DBF

La commande **CREATE** permet de créer un descripteur ACCESS et/ou un VIEW descripteur. Cette commande est toujours précédée par PROC ACCESS. Elle nécessite la spécification du répertoire SAS où le descripteur sera stocké, le nom du fichier descripteur et le type de fichier SAS (ACCESS pour un descripteur et VIEW pour le descripteur de view).

La commande PATH permet de donner le nom du répertoire, le nom du fichier PC et son extension. On parle de la commande de description de la base de données. Cette commande précède toute commande d'édition. Pour un fichier Dbase REND qui se trouve dans le répertoire CULTURE du lecteur C, la commande s'écrit :

PATH = C:\CULTURE\REND.DBF

La Commande **ASSIGN** permet d'indiquer la génération automatique ou non des noms des variables SAS à partir de celle des formats PC. Elle inter-agit avec les commandes FORMAT, RENAME, RESET, UNIQUE. La syntaxe est la suivante :

ASSIGN=YES | NO | Y | N ;

Avec la spécification YES ou Y, les noms des variables sont générés en ne prenant que les 8 premiers caractères des noms des variables des formats PC. Avec YES, les noms des variables SAS peuvent être changés mais pas le descripteur View. Dès lors les autres commandes d'édition ne sont plus valables.

Les commandes FORMAT et RENAME sont utilisées pour modifier dans le format et le nom de la variable SAS à partir du format PC. On écrit :

FORMAT FMT ' *nom-de-colonne PC = SAS format* ; | *rang de colonne ;*

RENAME ' nom- de-colonne PC' = nom-variable –SAS ;

Quant à la commande DROP, elle permet d'éliminer des colonnes du descripteur ACCESS. Ces colonnes ne peuvent être observées dans le descripteur view.

La commande LIST permet d'obtenir des informations sur les colonnes. La syntaxe est la suivante :

LIST ALL / View ' nom-de-colonnes ' ;

Avec SUBSET, le critère de choix des lignes est donné. En effet, cette commande permet de spécifier le critère de choix lors de la création du descripteur d'accès et du descripteur de vue. La syntaxe est la suivante :

SUBSET critère-de-sélection ;

Exemple 4: Programme de création du fichier AK2 et AK3 respectivement avec ACCESS et VIEW à partir du fichier dbase AK.dbf

```
LIBNAME LSAS 'C:\FORM';
PROC ACCESS DBMS=DBF;
  CREATE LSAS.AK2.ACCESS;
  PATH='C:\FORM\AK.DBF';
  CREATE LSAS.AK3.VIEW;
SELECT ALL;
LIST ALL;
RUN;
```


- Création des formats DbMS à partir de fichier SAS

La syntaxe pour créer un fichier de base de données à partir d'un fichier SAS est similaire dans la succession des commandes à celle de la création de fichier SAS à partir de format DbMS. Elle se présente globalement comme suit :

PROC DBLOAD DBMS=format-PC DATA=nom-fichier-SAS;

· Commande d'identification du fichier source DbMS

PATH=répertoire- nom-fichier.extension ;

· Commandes d'édition

ASSIGN Y/N ;
DELETE <'> nom-variable SAS ;
RENAME 'nom-variable SAS' = 'nom-colonne' ;
LIST ALL / colums / fields / noms-variables spécifiés ;
LIMIT=number;
LOAD;
RUN ;

4. GESTION DES DONNEES

4.1. Création d'un sous-ensemble de données SAS

Lorsque l'analyse porte sur une partie des données SAS, il est indispensable de créer un fichier de données SAS regroupant ce sous-ensemble de données. Il existe deux méthodes pour sélectionner ces données:

· effacement des données ne répondant pas à la condition;
· sélection des données répondant à la condition.

Dans le premier cas on utilise la condition IF puis la commande DELETE précédée de THEN comme suit:

IF *condition* THEN DELETE;

Par exemple, lorsque les données à analyser ne concerne pas la forêt de Djigbé dans le fichier DENDRO du répertoire C:\FORM, la procédure à utiliser est la suivante:

```
LIBNAME LSAS 'C:\FORM';
DATA LSAS.DENDRO1;
SET LSAS.DENDRO;
IF foret = ''Djigbe'' THEN delete;
RUN;
```

Dans cette procédure, la commande SET permet de lire le fichier SAS sur lequel la sélection du sous-ensemble de données sera opérée.

Pour la deuxième méthode, il suffit d'utiliser la commande IF *condition* . Cette procédure spécifie la condition qui doit être vérifiée. Ainsi, en s'intéressant aux données sur la forêt d'Agrimey dans le fichier LSAS.DENDRO, la commande est:

IF foret = 'Agrimey' ;

Ces deux méthodes ne permettent que de créer un seul sous-ensemble de données à la fois. Le langage SAS permet de réaliser dans la même procédure la création de plusieurs sous-ensembles de données SAS avec la commande OUTPUT.

Ainsi pour créer le fichier de données portant sur la forêt d'Agrimey et celui des données des autres forêts, dont les noms respectifs sont AGRIMEY et FORET, la procédure est la suivante:

*DATA LSAS.*AGRIMEY *LSAS.*FORET*;*
*SET LSAS.*DENDRO*;*
*IF foret = 'Agrimey' THEN OUTPUT LSAS.*AGRIMEY*;*
*ELSE OUTPUT LSAS.*FORET;
RUN;

Le contenu des deux fichiers sont obtenus dans la fenêtre OUTPUT en ajoutant la commande PRINT à la suite de la précédente procédure comme suit :

```
PROC PRINT DATA=LSAS.AGRIMEY ;
TITLE " Données dendrométriques de Agrimey";
PROC PRINT DATA=LSAS.FORET ;
TITLE " Données dendrométriques des autres
forêts";
RUN ;
```

Le contenu de la fenêtre OUTPUT se présente comme suit :

Données dendrométriques de agrimey

OBS	FORET	AGE	HD	DENSITE	ST
1	Agrimey	37	24.7	96	16.3
2	Agrimey	44	28.7	150	23.4

Données dendrométriques des autres forêts

OBS	FORET	AGE	HD	DENSITE	ST
1	Djigbé	32	27.2	188	20.5
2	Djigbé	37	28.9	133	19.0
3	Toffo	40	26.4	200	26.6
4	Massi	7	13.7	425	14.4

4.2. Génération de nouvelles variables

A partir des variables d'un fichier-source, de nouvelles variables peuvent être générées. Lorsqu'il s'agit des variables numériques les nouvelles variables sont obtenues soit avec des opérations d'addition, de soustraction, de multiplication de division ou de puissance. De même de nouvelles variables sont crées à partir de conditions vérifiées sur les observations de variables. Des compléments d'informations sont fournis par X(1997a).

Exemple 5. Création de nouvelles variables à partir des variables HD, AGE du fichier source DENDRO. La variable VAR1 est le rapport entre HD et AGE. La variable IP est caractère et prend la valeur IP=1 lorsque la variable HD supérieure ou égale à 0,8 et la valeur IP=2 si non. La procédure SAS pour créer ces nouvelles variables se présente comme suit :

```
LIBNAME LSAS 'C:\FORM';
DATA LSAS.GENE;
SET LSAS.DENDRO;
VAR1=HD/AGE;
IF VAR1>=0.8 THEN IP='IP=1';
ELSE IP='IP=2';
RUN;
```

Pour imprimer le contenu du fichier GENE, il faut ajouter les lignes suivantes :

```
PROC PRINT DATA=LSAS.GENE ;
TITLE "Exemple de création de nouvelles
variables " ;
TITLE2 "à partir des données d'un fichier-
source.";
RUN ;
```

Le résultat est le suivant :

Exemple de création de nouvelles variables
à partir des variables du fichier-source DENDRO

OBS	FORET	AGE	HD	DENSITE	ST	VAR1	IP
1	Djigbé	32	27.2	188	20.5	0.85000	IP=1
2	Djigbé	37	28.9	133	19.0	0.78108	IP=2
3	Agrimey	37	24.7	96	16.3	0.66757	IP=2
4	Toffo	40	26.4	200	26.6	0.66000	IP=2
5	Massi	7	13.7	425	14.4	1.95714	IP=1
6	Agrimey	44	28.7	150	23.4	0.65227	IP=2

4.3. Combinaison des données SAS

Le système SAS permet de combiner plusieurs données SAS soit par concaténation, soit par combinaison triée par ordre, soit par fusion soit tout simplement une fusion avec mise à jour.

Concaténation des données SAS

Cette procédure permet de combiner deux ou plusieurs ensemble de données SAS, l'un après les autres en un seul ensemble de données. Le nombre d'observations du fichier résultant est la somme des observations des données de départ comme le montre la figure 6.

89		249		89
119		59		119
461	+	607	=	461
103		75		103
325				325
369				369
				249
				59
				607
				75

Figure 6. Concaténation de données

Pour réaliser cette action, deux procédures sont utilisées: la procédure DATA avec la commande SET et la procédure APPEND. Les syntaxes sont respectivement les suivants:

DATA nouveau-fichier-SAS;
SET liste des fichiers SAS;

PROC APPEND BASE=*nom-fichier-base* <DATA=*nom-fichier joint*><FORCE>;

Le fichier joint contient les observations à ajouter à la fin des observations du fichier de base. Il faut que le fichier de base et le fichier à ajouter aient la même structure, c'est-à-dire les mêmes variables. Dans le cas où le fichier de base contient une variable qui ne se trouve pas dans celui de DATA, l'option FORCE permet de procéder à la concaténation des deux fichiers et remplace les observations non contenues dans le fichier joint par des données manquantes. Ce qui n'est pas le cas lorsque le fichier de base ne contient pas une variable qui se trouve dans le fichier joint. Compte tenu du fait que cette variable ne se trouve pas dans le descripteur du fichier de base, elle ne sera pas prise en compte par la concaténation.
Si le fichier de base n'existe pas, cette procédure permet de le créer et d'y faire figurer le contenu du fichier joint.

Exemple 6: Création d'un nouveau fichier *AGDJ* contenant les données des fichiers Agrimey et Djigbe du répertoire C:\FORM. Le programme type est le suivant :

```
LIBNAME LSAS 'C:\FORM';
DATA LSAS.AGDJ;
SET LSAS.Agrimey LSAS.Djigbe;
RUN;
PROC PRINT DATA=LSAS.AGDJ;
TITLE ''Concaténation des fichiers Agrimey
et Djibe'';
TITLE2 ''pour créer le fichier agdj'';
Run;
```

Le même résultat peut être obtenu par l'ajout des observations du fichier Djigbe au fichier Agrimey du répertoire C:\FORM. Dans ce cas, la procédure utilisant APPEND est requise. Soit :

```
LIBNAME LSAS 'C:\FORM';
PROC APPEND BASE = LSAS.Agrimey DATA=
LSAS.Djigbe;
RUN;
PROC PRINT DATA=LSAS.Agrimey;
   TITLE "Fichier Agrimey après ajout des données
   du fichier Djigbé";
RUN;
```

Fusion des fichiers SAS

Pour la fusion des fichiers, le système SAS permet de réaliser la fusion de deux ou plusieurs fichiers avec des variables différentes ou non et des nombres d'observations identiques ou non. Le but essentiel de cette procédure est de constituer un nouveau fichier qui prend en compte les données d'autres fichiers. Prenons par exemple un fichier comportant la variable X dont les données sont X1, ..., X5 et un autre fichier avec les données Y1, ..., Y5. La figure exprime le résultat de fusion de fichiers.

Pour réaliser cette fusion, la procédure utilisée est MERGE dont la syntaxe est la suivante (X, 1997b) :

X1		Y1	=	X1 Y1
X2	+	Y2		X2 Y2
X3		Y3		X3 Y3
X4		Y4		X4 Y4
X5		Y5		X5 Y5

Figure 7. Structures des fichiers fusionnés et du fichier résultant.

MERGE *nom-fichier1 nom-ficher2 ;*
BY *var* ;

Dans la réalité, la nécessité de fusion des fichiers pour avoir une base de données cohérentes est courante. Par exemple un fichier dénommé PROTOCOL est relatif à la description des protocoles expérimentaux de cinq essais. Un autre, du nom de ANALYSE, contient les résultats d'analyse statistiques des essais 1 à 4 et d'un autre essai non reporté dans le fichier de description. Les données portées dans ce fichier concernent la variance résiduelle de deux méthodes à savoir M1 et M2 comme ils se présentent ci-après.

	PROTOCOL				ANALYSE		
OBS	ESSAI	DISP	VAR	OBS	ESSAI	M1	M2
1	1	BAC	Ht	1	1	6.68	7.30
2	2	BAC	Diam	2	2	1.07	1.47
3	3	CL	Ht	3	3	13.30	12.40
4	4	CL	Rdt	4	4	7.41	7.66
5	5	SP	Rdt	5	6	17.92	8.63

Pour réaliser la fusion des deux fichiers en un seul du nom de RESULT, le programme SAS se présente comme suit:

```
LIBNAME LSAS 'C:\FORM';
DATA LSAS.RESULT;
MERGE LSAS.PROTOCOL LSAS.ANALYSE;
      IF M1=' ' THEN M1=0;
     IF M2=' ' THEN M2=0;
```

```
        IF DISP=' ' THEN DISP='NONE';
        IF VAR=' ' THEN VAR='NONE';
RUN;
```

Le fichier RESULT se présente comme suit:

OBS	ESSAI	DISP	VAR	M1	M2
1	1	BAC	Ht	6.68	7.30
2	2	BAC	Diam	1.07	1.47
3	3	CL	Ht	13.30	12.40
4	4	CL	Rdt	7.41	7.66
5	5	SP	Rdt	0.00	0.00
6	6	none	none	17.92	8.63

Si l'on désire completer le fichier RESULT par le lieu où les essais se sont deroulés et le nom de l'institutition de recherche, pour rester le plus complet possible, SAS le permet. Comme il s'agit d'un seul lieu et d'une seule isntitutition, on parle alors d'ajout d'une observation à plusieurs observations. Si l'information concernant le lieu et l'institution se trouve dans le fichier IDENT soit seul ou avec d'autres enrégistrements, la syntaxe pour créer le fichier RESULT2 se présente respectivement comme suit:

```
DATA RESULT2;
IF _n_=1 THEN SET IDENT;
SET RESULT;
RUN;

DATA RESULT2;
IF _n_=1 THEN SET IDENT POINT=last;
NOBS=last;
SET RESULT;
RUN;
```

Considérons le fichier IDENT comme suit:

OBS	LIEU	INSTITUT
1	Toffo	INRAB
2	Calavi	FSA
3	Pahou	INRAB

Le contenu du fichier RESULT2 est le suivant:

OBS	LIEU	INSTITUT	ESSAI	DISP	VAR	M1	M2
1	Pahou	INRAB	1	BAC	Ht	6.68	7.30
2	Pahou	INRAB	2	BAC	Diam	1.07	1.47
3	Pahou	INRAB	3	CL	Ht	13.30	12.40
4	Pahou	INRAB	4	CL	Rdt	7.41	7.66
5	Pahou	INRAB	5	SP	Rdt	0.00	0.00
6	Pahou	INRAB	6	none	none	17.92	8.63

Des informations complémentaires sur différentes autres possibilités SAS pour fusionner les fichiers sont disponibles dans X(1995).

4.4. Gestion des données avec la procédure SQL

La procédure SQL est un langage d'interrogation structurée du logiciel SAS. Elle permet :

- de créer ou de supprimer une table dans une base de données ;
- d'ajouter ou de supprimer une ou plusieurs ligne(s) ou colonne(s) ;
- d'appliquer des fonctions sur des colonnes et de créer des tables virtuelles.

Syntaxe général de la procédure SQL

La syntaxe générale se présente comme suit (X, 1996b) :

PROC SQL ;
  CREATE TABLE *nom-de-table*
  SELECT *nom-de-colonnes*
  FROM *nom-de-table-source*
  WHERE *condition;*
RUN ;

La commande SELECT permet de sélectionner les colonnes spécifiées dans *nom-de-colonne*. Lorsque que l'on désire prendre l'ensemble des informations contenu dans le fichier source, *nom-de-colonne* est remplacé par *.
La sélection des lignes repose sur les valeurs prises par les lignes dans une colonne donnée. La commande utilisée à cet effet est WHERE *condition*.

Les opérateurs utilisés dans les conditions sont =, ^=, <, >, <=, >=, ou EQ, NE, LT, GT, LE, GE. Les opérateurs logiques sont aussi utilisés pour réaliser la condition de choix ou de suppression des lignes. Il s'agit de AND, OR et NOT.
La commande **FROM** identifie le fichier ou la table à partir duquel l'on recherche l'information dont les caractéristiques sont précisées par **SELECT** et **WHERE**.
Ces trois commandes peuvent être précédées ou non de la commande **CREATE TABLE**. Dans le premier cas, on parle de création de table contenant les colonnes spécifiées dans SELECT et les lignes vérifiant la condition spécifiée par **WHERE**. Dans le second cas la table créée est virtuelle.

Exemple 7. Création d'une table virtuelle à partir du fichier SAS CLIENT du répertoire C:\FORM contenant les colonnes NCLI, NOM et MONTANT.

La base de données du fichier source CLIENT se présente comme suit :

NCL	NOM	PRENOM	AGE	RESID	SOLDE
001	DOSSA	Georges	42	Calavi	90870
0045	ATCHI	Anselme	38	Cotonou	-4539
004	AMAKPE	Jonas	34	Ouidah	800
032	ADISSOH	Joël	25	Ouidah	700
067	JESS	Raphaël	29	Cotonou	6500
008	WINNOU	Anasthase	26	Cotonou	6500
098	HOUSSA	Mathieu	18	Godomey	-4000
009	MISSOH	Gaetan	30	Tankpe	890

Pour constituer une table virtuelle, la procédure est la suivante:

```
LIBNAME SQL 'C:\FORM ;
PROC SQL;
SELECT NCL NOM MONTANT;
FROM SQL.CLIENT ;
RUN;
```

Le résultat se présente comme suit:

NCL	NOM	SOLDE
001	DOSSA	90870
045	ATCHI	-4539
004	AMAKPE	800
032	ADISSOH	700
067	JESS	6500
008	WINNOU	6500
098	HOUSSA	-4000
009	MISSOH	890

Lorsque l'on s'intéresse à des clients dont les soldes sont négatifs, la procédure s'écrit :

```
LIBNAME SQL 'C:\ FORM ;
PROC SQL;
SELECT NCL NOM MONTANT;
FROM SQL.CLIENT
WHERE SOLDE<0;
RUN;
```

Le résultat est alors:

NCL	NOM	SOLDE
045	ATCHI	-4539
098	HOUSSA	-4000

Exemple 8 : Création d'une table ou d'un fichier SQL
Pour créer la table CLIENT2 composée des clients dont les soldes sont créditeurs, la procédure est la suivante :

```
LIBNAME SQL 'C:\FORM ;
PROC SQL;
CREATE TABLE SQL.CLIENT2 AS
SELECT NCL NOM MONTANT;
FROM SQL.CLIENT;
WHERE SOLDE>0;
RUN
```

Pour visualiser le contenu du fichier CLIENT2 ; il faut créer une table virtuelle en ajoutant avant la commande RUN ce qui suit :

```
SELECT * FROM SQL. CLIENT2 ;
```

et le résultat se présente comme suit:

NCL	NOM	SOLDE
fffffffffffffffffffffffffffff		
001	DOSSA	90870
004	AMAKPE	800
032	ADISSOH	700
067	JESS	6500
008	WINNOU	6500
009	MISSOH	890

5. Généralités sur les procédures d'analyse statistique de données: cas de la statistique descriptive

Pour utiliser les procédures statistiques, il est nécessaire de connaître les commandes correspondantes et les options possibles. Il faut aussi préciser le fichier de données SAS contenant les variables à analyser. Nous donnerons ici des exemples de statistique descriptive.

Pour obtenir la moyenne, l'écart-type le minimum et le maximum de même que le nombre d'observations des variables numériques d'un fichier SAS du nom DENDRO, la procédure MEANS peut être utilisée comme dans l'exemple 9.

Exemple 9 : Syntaxe sommaire de la procédure MEANS

```
LIBNAME LSAS'C:\FORM';
PROC MEANS DATA=LSAS.DENDRO;
VAR Age Hd densite St;
RUN;
```

Le résultat se présente comme suit:

Variable	N	Mean	Std Dev	Minimum	Maximum
AGE	6	32.8333333	13.2577022	7.0000000	44.0000000
HD	6	24.9333333	5.7175752	13.7000000	28.9000000
DENSITE	6	198.6666667	117.1010959	96.0000000	425.0000000
ST	6	20.0333333	4.5036282	14.4000000	26.6000000

D'autre part, les procédures UNIVARITE et TABULATE permettent aussi de réaliser une description de variables. La première avec l'option PLOT génère trois graphiques à savoir Steam-and-leaf plot, Box plot et Normal probability plot. La seconde génère un tableau de contingence relatif à deux modalités

ou variables et dont les cellules contiennent des paramètres de statistique descriptive comme ceux de MEANS, FREQ et SUMMARY.

Références

X (1995). Combining and modifying SAS data sets. Version 6 first Edition. Cary, SAS Institute Inc., 197 p.
X (1996a). Introducing the SAS System. Version 6 first Edition. Cary, SAS Institute Inc., 83 p.
X (1996b). SAS guide to the SQL Procedure: Usage and Reference. Version 6, First Edition. Cary, SAS Institute Inc., 210 p.
X (1997a). SAS language and procedures: Usage, Version 6 first Edition. Cary, SAS Institute Inc., 638 p.
X (1997b). SAS language and procedures: Usage 2, Cary, SAS Institute Inc., 649 p.

Remerciement

Le Centre de Biométrie et d'Informatique Générale remercie la Coopération Universitaire au Développement à travers le Conseil Inter-Universitaire Francophone de la Belgique pour avoir financé l'édition de ce numéro de Notes de Biométrie et d'Informatique.

La collection

Notes de Biométrie et d'Informatique réunit des publications, des documents didactiques et des notes de recherches dans les domaines de la biométrie et de l'informatique appliquée et surtout avec une orientation spécifiquement tropicale émanant des services de mathématiques appliquées et d'informatique des institutions d'enseignement supérieur et de recherche du Bénin.

Dépôt légal n°1756 du 19/06/2001 Bibliothèque Nationale,
Porto-Novo, Bénin
2001