

NOTES DE STATISTIQUE ET D'INFORMATIQUE

2006/3

TRAITEMENT DES DONNÉES PAR
LE LOGICIEL SAS
INTRODUCTION AU MODULE DE BASE

I. CARLETTI, H. PRÉVOT

Faculté universitaire des
Sciences agronomiques

Centre de Recherches
agronomiques

GEMBLOUX

(Belgique)

TRAITEMENT DES DONNEES PAR LE LOGICIEL SAS INTRODUCTION AU MODULE DE BASE

I. CARLETTI¹ ET H. PRÉVOT²

RÉSUMÉ

Cette note constitue une introduction au module de base SAS. Après un bref aperçu de l'environnement de travail SAS et des principes du langage SAS, les étapes DATA et PROC d'un programme SAS sont exposées plus en détails.

SUMMARY

This paper is an introduction to Base SAS software. After an introduction to SAS workspace and SAS language basis, DATA and PROC Steps are more detailed.

1. INTRODUCTION

Le logiciel SAS (*Statistical Analysis System*), conçu en 1976 par SAS Institute (North Carolina, USA), est un logiciel permettant de générer, d'échanger et de traiter des données informatiques indépendamment de leur format ou de leur plate-forme de résidence.

Le logiciel SAS est en réalité un système intégré de modules dont le composant principal est le module de base (*Base SAS Software*) qui permet les opérations classiques d'accès aux données et d'analyse élémentaire de données. Pour des tâches plus élaborées, différents modules spécialisés peuvent être ajoutés à ce module de base.

¹ Assistante à la Faculté universitaire des Sciences agronomiques de Gembloux

² Informaticien à la Faculté universitaire des Sciences agronomiques de Gembloux


Le module de base SAS est présenté dans cette note en considérant que l'utilisateur dispose de la version 8 sous système d'exploitation Windows.

La première partie du document, intitulée « Premiers pas avec SAS », constitue une introduction rapide à l'environnement de travail SAS et aux principes de base du langage SAS (paragraphe 2).

Ensuite, les principes du langage sont détaillés, d'une part, pour les étapes DATA, relatives à la gestion des données (paragraphe 3), et, d'autre part, pour les étapes PROC, relatives à l'analyse des données (paragraphe 4).

Pour terminer, quelques références et quelques informations complémentaires sur les autres modules SAS sont renseignées au paragraphe 5.

2. PREMIERS PAS AVEC SAS

Lorsque l'on démarre SAS (), cinq fenêtres principales sont ouvertes : les fenêtres Explorer, Results, Editor, Log et Output (figure 1).

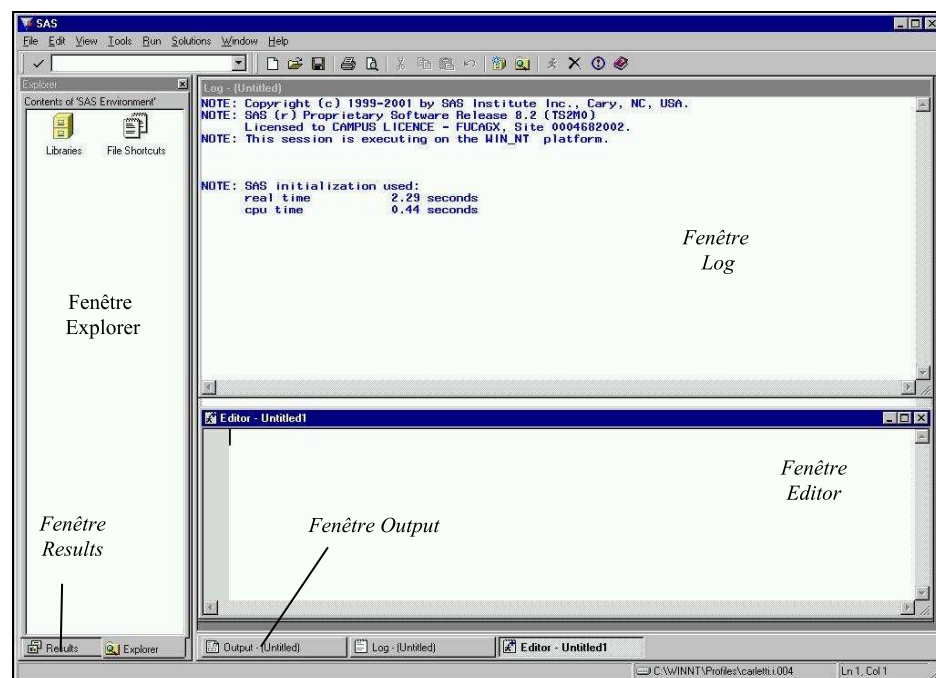


Figure 1. Environnement de travail de SAS v 8.2.

La fenêtre `Editor` permet l'écriture des programmes SAS ainsi que l'accès aux programmes SAS existants. Les mots-clés du langage sont mis en évidence grâce à un code de couleurs et un vérificateur de syntaxe fait apparaître les instructions incorrectes en rouge. Un exemple très simple de programme SAS est présenté à la figure 2. Il permet la lecture de données relatives à la teneur en matières organiques de parcelles cultivées (le type de culture est également précisé), l'impression de ces données et le calcul de la teneur en matières organiques moyenne pour l'ensemble des parcelles. Les fichiers contenant un programme SAS portent l'extension `.SAS`.

```

DATA sol ;
  INPUT culture $ 1-10 mo 12-14;
  DATALINES;
  escourgeon 330
  pois      320
  betterave 400
  pois      310
  escourgeon 345
  froment   270
  froment   355
  ;
RUN;

PROC PRINT DATA=sol ;
RUN;

PROC MEANS DATA=sol MEAN STD ;
  VAR mo ;
RUN;

```

Figure 2. La fenêtre Editor.

Dans le programme SAS présenté, on peut identifier deux grandes étapes.

La première étape consiste à convertir les données de l'utilisateur en une table SAS (étape `DATA`). En effet, l'entité de base traitée par le langage SAS est un tableau de données ou table SAS (*SAS data set*). Cette étape est un préalable nécessaire à tout traitement de données ultérieur.


Une table SAS est un tableau rectangulaire où les lignes et les colonnes sont appelées respectivement observations et variables. L'intersection d'une observation et d'une variable est dénommée valeur. Toute valeur peut être numérique, enregistrée par défaut sur huit octets, ou alphanumérique, formée de 1 à 32 767 caractères. Toute valeur manquante numérique est représentée par un point et toute valeur manquante alphanumérique par un blanc. Toute variable est identifiée par son nom, formé au

maximum de 32 caractères (lettres, chiffres et caractères de soulignement) dont le premier symbole est toujours une lettre ou le caractère de soulignement.

La deuxième étape de notre exemple reprend des procédures d'obtention de résultats (impression et calcul d'une moyenne). Ces procédures sont qualifiées d'étapes PROC.

Un programme SAS est donc un enchaînement d'étapes DATA et/ou PROC.

De plus, chaque instruction SAS se termine par un point virgule et SAS détecte la fin d'une étape quand il rencontre une instruction RUN (pour la plupart des étapes), une instruction QUIT (pour certaines procédures) ou le début d'une autre étape. Majuscules et minuscules peuvent être indifféremment utilisées pour écrire les instructions. Néanmoins, pour faciliter la compréhension du document, les instructions SAS sont présentées en majuscules et les noms choisis par l'utilisateur en minuscules.

Pour exécuter ce programme, il suffit de cliquer sur l'icône  de la barre d'outils ou de sélectionner Submit dans le menu Run.

Après exécution du programme, la fenêtre Log (figure 3) affiche les instructions du programme soumis et les messages du système ainsi que les éventuels messages d'erreurs. Les fichiers log portent l'extension .LOG.

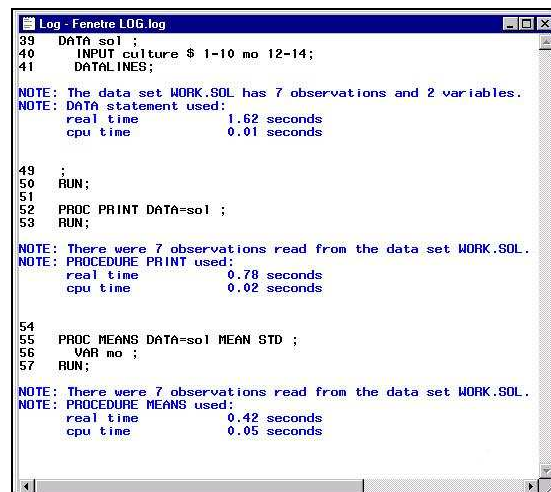


Figure 3. La fenêtre Log.

Si l'examen de la fenêtre log montre que le programme ne comporte aucune erreur, les résultats peuvent être visualisés dans la fenêtre Output (figure 4).

L'accès aux différents éléments des résultats est facilité par la fenêtre `results`. Dans notre exemple, on peut afficher les résultats des procédures `PROC PRINT` ou `PROC MEANS`. Les résultats peuvent être également sauvegardés et imprimés à partir du menu contextuel de chaque élément de résultat (figure 5). Les fichiers de résultats portent l'extension `.LST`.

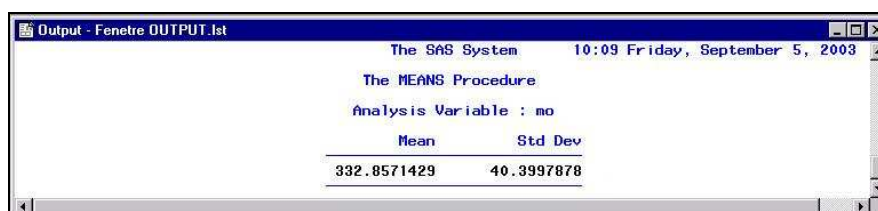


Figure 4. La fenêtre Output.

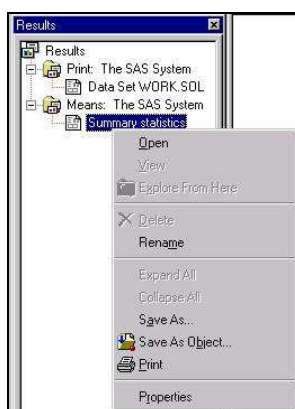


Figure 5. La fenêtre Results.

La fenêtre `Explorer` permet l'accès aux tables SAS. Par défaut, les tables SAS sont stockées dans la bibliothèque (*library*) ou répertoire `Work`. En double-cliquant sur la table SAS dénommée dans notre exemple « sol », on peut visualiser son contenu dans un tableur (figure 6). Les tables SAS portent l'extension `.SAS7BDAT` ou `.sd7` ou `.sd2` (selon les versions du logiciel).

The screenshot shows an "Explorer" window with "Contents of 'Work'" and a table with the following data:

	culture	mo
1	escourgeon	330
2	pois	320
3	betterave	400
4	pois	310
5	escourgeon	345
6	froment	270
7	froment	355

Figure 6. La fenêtre Explorer.

3. ETAPES DATA

3.1. Principes

L'étape DATA permet de créer des tables SAS ou des fichiers externes à partir de données encodées dans l'étape DATA elle-même, de tables SAS ou de fichiers externes. Elle permet également de modifier des tables SAS.

La syntaxe générale d'une étape DATA est la suivante³ :

```
DATA [[nom_logique.]nom_de_la_table_à_créer];  
    Instructions;  
RUN;
```

Par défaut (si aucun nom logique n'est associé au nom de la table à créer), les tables SAS sont temporaires. Elles sont stockées pendant la session de travail dans le répertoire *work* et sont détruites automatiquement à la clôture de la session de travail. Si l'utilisateur souhaite conserver une table SAS d'une session de travail à l'autre, il doit la rendre permanente. Pour ce faire, l'utilisateur doit d'abord choisir un répertoire accessible en écriture et lui attribuer un nom logique ou LIBREF (*library reference name*) à l'aide de l'instruction reprise ci-après :

```
LIBNAME nom_logique "chemin_d'accès_du_répertoire" ;
```

Le nom logique est choisi arbitrairement, il comprend au maximum huit caractères alphabétiques (exemple : LSAS). Les noms suivants sont réservés par le logiciel SAS et leur utilisation est donc proscrite : WORK, SASUSER, SASHELP et SASMSG.

Les données de l'exemple qui servira à illustrer les étapes DATA et les étapes PROC sont présentées en annexe.

³ dans le document, ce qui figure entre crochets est optionnel ou facultatif.

3.2. Lecture et écriture

Pour la lecture de données encodées dans l'étape DATA (programme 1), il faut commencer une étape DATA (instruction DATA) en indiquant le nom de la table SAS à créer. Dans ce cas-ci, il faut créer une table SAS permanente et donc introduire préalablement à l'étape DATA, une instruction LIBNAME.

Ensuite, pour décrire la manière de lire les données, il est nécessaire d'utiliser l'instruction INPUT suivie du nom des variables, accompagné du caractère \$ s'il s'agit de données alphanumériques. Lorsque la lecture est réalisée par colonne, il faut également indiquer le numéro des colonnes correspondant à l'espace pris par chacune des variables :

```
INPUT {nom_de_la_variable[$] numéro_de_la_première_
      colonne - numéro_de_la_dernière_colonne
      [.nombre_de_décimales]};4
```

A la suite de l'instruction INPUT, on indique le début des données par l'instruction DATALINES (ou CARDS, instruction plus ancienne) et la fin des données par l'instruction nulle (;).

Pour les valeurs relatives à des dates présentées selon le format YYMMDD ou DDMMYY, et en vue d'un traitement numérique éventuel, il importe de définir préalablement le format de lecture ou d'écriture sur base des instructions suivantes :

lecture :

```
INFORMAT nom(s)_de_la(ou_des)_variable(s) YYMMDD8.;
ou
INFORMAT nom(s)_de_la(ou_des)_variable(s) DDMMYY8.;
```

écriture :

```
FORMAT nom(s)_de_la(ou_des)_variable(s) YYMMDD8.;
ou
FORMAT nom(s)_de_la(ou_des)_variable(s) DDMMYY8.;
```

⁴ ce qui figure entre accolades est à répéter pour toutes les variables

Toute ligne débutant par * et se terminant par ; permet d'introduire des commentaires au sein d'un programme SAS. De même, toute suite de caractères comprise entre /* et */ est considéré comme un commentaire. De plus, dans ce programme, une procédure (paragraphe 4.4) est utilisée pour imprimer (PROC PRINT) le contenu de la table SAS, dont le résultat est ensuite proposé.

Programme 1

```

/* données introduites dans l'étape DATA et résultat de
la lecture dans une table SAS permanente sur disquette du
lecteur a : */
LIBNAME lsas "a:\" ;
DATA lsas.sol ;
  INFORMAT datecrea YYMMDD8. ;
  INPUT nparcult 1-2
        nomparc $ 4-14
        nomcult $ 16-19
        surface 21-25
        datecrea
        typesol $ 36-55
        N 57-58
        P 60-61
        K 63-64
        CaOH 66-69
        Mo 71-73
        pH 75-77 ;
  DATALINES ;
Données de l'exemple
;
PROC PRINT ;
  FORMAT datecrea DDMMYY8. ;
RUN ;

```

Pour la lecture (ou l'écriture) d'un fichier .DAT (programme 2), on utilise les instructions suivantes :

lecture :

```

INFILE "nom_du_fichier" [options];
INPUT {nom_de_la_variable[$] numéro_de_la_première_
      colonne - numéro_de_la_dernière_colonne
      [.nombre_de_décimales]};

```

écriture :

```
FILE "nom_du_fichier" [options];  
PUT {nom_de_la_variable [$] numéro_de_la  
    première_colonne- numéro_de_la_dernière_colonne  
    [.nombre_de_décimales]};
```

Pour la lecture (ou l'écriture) d'une table .SAS7bdat, on utilise les instructions suivantes :

lecture :

```
SET [[lsas.]nom_de_la_table]  
    [FIRSTOBS = numéro_de_la_première_observation_à_lire]  
    [OBS = numéro_de_la_dernière_observation_souhaitée];
```

écriture :

```
[OUTPUT [[lsas.]nom_de_la_table]];
```

Programme 2

```
/* constitution d'une table sas SOL à partir du fichier  
sol.dat */  
LIBNAME lsas "a:\" ;  
DATA lsas.sol ;  
    INFORMAT datecrea YMMDD8. ;  
    INFILE "a:\sol.dat" ;  
    INPUT nparcult 1-2  
          nomparc $ 4-14  
          nomcult $ 16-19  
          surface 21-25  
          datecrea  
          typesol $ 36-55  
          N 57-58  
          P 60-61  
          K 63-64  
          CaOH 66-69  
          Mo 71-73  
          pH 75-77 ;  
  
PROC PRINT ;  
    FORMAT datecrea DDMMYY8. ;  
RUN ;
```

Résultat 1 ou 2

obs	datecrea	nparcult	nomparc	nomcult	surface	typesol	N	P	K	CaOH	Mo	pH
1	01/09/92	1	BURLET	ESC	7.68	argileux dessus doux	14	19	7	2.20	330	7.5
2	01/09/92	2	CAMPAGNETTE	POIS	3.89	argile lourde	6	17	13	2.76	320	7.3
3	01/01/77	3	CARRIERE	BETT	6.81	limon argile	12	18	12	2.00	400	7.5
4	01/01/77	4	DEWAT	POIS	1.81	limon humide	12	19	6.60	310	6.5	
5	01/01/77	5	DV FERME		15.88	limon lourd argileux	13	22	12	2.00	400	7.5
6	01/01/91	6	JODOIGNE	ESC	6.00	limon argile sable	15	15	16	2.30	345	7.5
7	01/01/77	7	LONG-FOSSE	FROM	20.73	limon lourd argileux	15	15	6	2.20	270	7.2
8	01/09/92	8	LEMPEREUR		4.42	argile lourde	13	12	25	2.30	350	6.6
9	01/01/77	9	PRAIRIE	PRAI	2.08	limon argile	14	11	19	2.17	347	7.1
10	01/01/77	10	90 ARES	ESC	0.93	limon lourd argileux	16	17	6	2.24	320	7.5
11	01/01/77	11	THOREMBAIS	FROM	10.56	argile limon	12	17	6	2.24	355	7.5
12	02/09/95	12	LONG FOSS	ORGE	16.00	limon moyen	11	14	16	2.82	356	6.8
13	02/09/95	13	LONG FOSSE	ESC	4.73	limon argile sable	16	15	17	2.40	352	7.4
14	26/09/96	14	DV FERME	FROM	6.00	limon lourd argileux	12	16	15	2.38	372	7.2
15	26/09/96	15	DV FERME	ESC	6.00	limon lourd argileux	13	15	13	2.71	349	7.1
16	26/09/96	16	DV FERME	POIS	3.88	limon lourd argileux	12	15	17	2.73	369	7.2
17	18/08/96	17	LEMPEREUR	FROM	2.00	argile lourde	17	14	15	2.82	358	7.1
18	18/09/96	18	LEMPEREUR	BETT	2.42							
19	15/10/97	19	DV FERME	FROM	2.60	limon lourd argileux	15	12	14	2.58		7.3
20	15/10/97	20	DV FERME	BETT	13.28	limon lourd argileux	17	13	2.75	359	7.3	
21	01/10/97	21	ZELIA	POIS	1.25	limon moyen	12	25	12	2.30	350	7.0

Une autre façon de procéder pour la lecture et l'écriture de fichiers externes est l'utilisation des procédures PROC IMPORT et PROC EXPORT, qui permettent notamment la lecture et l'écriture de feuilles *Microsoft Excel* (programme 3), de tables *Microsoft Access* et de fichiers externes délimités.

Programme 3

```
/* lecture d'une feuille de travail "sol" et création  
d'une table SAS lsas.sol*/
```

```
LIBNAME lsas "a:\";
```

```
PROC IMPORT DATAFILE="a:\sol.xls" OUT=lsas.sol
```

```
DBMS=Excel97 ;
```

```
SHEET="sol";
```

```
RUN;
```

```
PROC PRINT DATA=lsas.sol;
```

```
RUN;
```

3.3. Opérations sur les variables et les observations

Les instructions énoncées ci-dessous concernent des opérations relatives aux variables et aux observations d'une table SAS.

- Elimination de variables

```
DROP nom(s)_de_la(ou des)_variable(s) à éliminer;
```

```
KEEP nom(s)_de_la(ou des)_variable(s) à conserver;
```

- Modification du nom

```
RENAME ancien_nom_de_la_variable = nouveau_nom_de_la_variable;
```

- Identification littérale

```
LABEL nom_de_la_variable = "expression alphabétique"
```

L'exemple ci-dessous (programme 4) crée une nouvelle table SAS (analyse) en ne retenant que les résultats de l'analyse de sol.

Programme 4

```
LIBNAME lsas "a:\" ;  
DATA lsas.analyse ;  
  SET lsas.sol ;  
  KEEP N P K CaOH Mo pH ;  
  LABEL N="teneur en azote";  
  LABEL P="teneur en P2O5";  
  LABEL K="teneur en K2O";  
  LABEL CaOH="teneur en CaOH";  
  LABEL Mo="matière organique" ;  
  
PROC PRINT DATA=lsas.analyse LABEL;  
  
RUN ;
```

Résultat 4

Obs	teneur en azote	teneur en P2O5	teneur en K2O	teneur en CaOH	matière organique	PH
1	14	19	7	2.20	330	7.5
2	6	17	13	2.76	320	7.3
3	12	18	12	2.00	400	7.5
4	.	12	19	6.60	310	6.5
5	13	22	12	2.00	400	7.5
6	15	15	16	2.30	345	7.5
7	15	15	6	2.20	270	7.2
8	13	12	25	2.30	350	6.6
9	14	11	19	2.17	347	7.1
10	16	17	6	2.24	320	7.5
11	12	17	6	2.24	355	7.5
12	11	14	16	2.82	356	6.8
13	16	15	17	2.40	352	7.4
14	12	16	15	2.38	372	7.2
15	13	15	13	2.71	349	7.1
16	12	15	17	2.73	369	7.2
17	17	14	15	2.82	358	7.1
18
19	15	12	14	2.58	.	7.3
20	.	17	13	2.75	359	7.3
21	12	25	12	2.30	350	7.0

▪ Expressions SAS

`Variable_résultat = EXPRESSION_SAS;`

Toute expression SAS peut contenir :

- des noms de variables ;
- des constantes numériques (exemples : 1.23, -10), caractères (exemple : "NUL"), dates, heures ;
- des symboles : (), **, *, /, +, - ;
- des opérateurs de comparaison : EQ (ou =), NE (ou ^=), GT (ou >), GE (ou >=), LT (ou <), LE (ou <=) ;
- des opérateurs logiques : AND (ou &), OR (ou |), NOT (ou ^) ;
- des fonctions :
 - ABS() : valeur absolue ;
 - EXP() : exponentielle ;
 - LOG() : logarithme en base e ;
 - LOG10() : logarithme en base 10 ;
 - SQRT() : racine carrée ;
 - etc.

- Répétition sans compteur

```
DO;  
  Instructions;  
END;
```

- Répétition avec compteur

```
DO variable_indice = valeur_de_départ TO valeur_de_fin  
  BY accroissement;  
  Instructions;  
END;
```

- Répétition logique

```
DO WHILE (expression_logique);  
  Instructions;  
END;
```

ou

```
DO UNTIL (expression_logique);  
  Instructions;  
END;
```

- Décision simple

```
IF expression_logique THEN instruction;
```

- Décision multiple (END étant imposé par la présence du DO)

```
IF expression_logique_1 THEN DO;  
  Bloc_d'instructions_1;  
END;  
[ELSE IF expression_logique_2 THEN DO;  
  Bloc_d'instructions_2;  
END;  
.  
.  
[ELSE IF expression_logique_p THEN DO;  
  Bloc_d'instructions_p;  
END;]]  
[ELSE DO;  
  Bloc_d'instructions_q;  
END;]
```

- Sélection d'observations

Une décision appliquée à une valeur accompagnée d'une des instructions suivantes permet :

- la suppression d'une observation dans la table SAS : DELETE ;
- la conservation d'une observation dans la table SAS : OUTPUT ;
- l'écriture d'une observation dans un fichier .LOG: LIST.

L'exemple ci-après (programme 5) élimine toutes les observations relatives aux parcelles "POIS".

Programme 5

```
LIBNAME lsas "a:\" ;  
DATA lsas.pois ;  
  SET lsas.sol ;  
  IF nomcult="POIS" THEN DELETE ;  
  KEEP nomcult pH ;  
  
PROC PRINT DATA=lsas.pois ;  
RUN ;
```

Résultat 5

Obs	nomcult	pH
1	ESC	7.5
2	BETT	7.5
3		7.5
4	ESC	7.5
5	FROM	7.2
6		6.6
7	PRAI	7.1
8	ESC	7.5
9	FROM	7.5
10	ORGE	6.8
11	ESC	7.4
12	FROM	7.2
13	ESC	7.1
14	FROM	7.1
15	BETT	.
16	FROM	7.3
17	BETT	7.3

L'instruction WHERE ou IF (*subsetting IF*) permet également de sélectionner un ensemble d'observations.

L'exemple ci-après (programme 6) génère une table SAS from.sas7bdat reprenant les parcelles en froment dont le pH est supérieur ou égal à 7.

Programme 6

```
LIBNAME lsas "a:\" ;  
DATA lsas.from ;  
    SET lsas.sol ;  
    IF nomcult="FROM" AND Ph>=7 ;  
    KEEP nparcult ph ;
```

```
PROC PRINT DATA=lsas.from ;  
RUN ;
```

```
LIBNAME lsas "a:\" ;  
DATA lsas.from ;  
    SET lsas.sol ;  
    WHERE nomcult="FROM" AND Ph>=7 ;  
    KEEP nparcult ph ;
```

```
PROC PRINT DATA=lsas.from ;  
RUN ;
```

Résultat 6

Obs	nparcult	pH
1	7	7.2
2	11	7.5
3	14	7.2
4	17	7.1
5	19	7.3

- création de plusieurs tables SAS

L'instruction OUTPUT, incluse dans une décision et suivie d'un nom de table SAS, permet de créer de nouvelles tables.

L'exemple suivant (programme 7) crée une table SAS rassemblant les parcelles de froment (*froment*) et une table SAS avec les parcelles d'escourgeon (*escourg*), chaque nouvelle table ayant la même structure que la table de départ.

Programme 7

```
LIBNAME lsas "a:\";
DATA lsas.froment lsas.escourg ;
  SET lsas.sol ;
  IF nomcult="FROM" THEN OUTPUT lsas.froment ;
  ELSE IF nomcult="ESC" THEN OUTPUT lsas.escourg ;
PROC PRINT DATA=lsas.froment ;
PROC PRINT DATA=lsas.escourg ;

RUN ;
```

Résultat 7

Obs	DATECREA	NPARCULT	NOMPARC	NOMCULT	SURFACE	TYPESOL	N	P	K	CAOH	MO	PH
1	6210	7	LONG-FOSSE	FROM	20.73	limon lourd argileux	15	15	6	2.20	270	7.2
2	6210	11	THOREMBATS	FROM	10.56	argile limon	12	17	6	2.24	355	7.5
3	13418	14	DV FERME	FROM	6.00	limon lourd argileux	12	16	15	2.38	372	7.2
4	13379	17	LEMPEREUR	FROM	2.00	argile lourd	17	14	15	2.82	358	7.1
5	13802	19	DV FERME	FROM	2.60	limon lourd argileux	15	12	14	2.58		7.3
Obs	DATECREA	NPARCULT	NOMPARC	NOMCULT	SURFACE	TYPESOL	N	P	K	CAOH	MO	PH
1	11932	1	BURLET	ESC	7.68	argileux dessus doux	14	19	7	2.20	330	7.5
2	11323	6	JODOIGNE	ESC	6.00	limon argile sable	15	15	16	2.30	345	7.5
3	6210	10	90 ARES	ESC	0.93	limon lourd argileux	16	17	6	2.24	320	7.5
4	13028	13	LONG FOSSE	ESC	4.73	limon argile sable	16	15	17	2.40	352	7.4
5	13418	15	DV FERME	ESC	6.00	limon lourd argileux	13	15	13	2.71	349	7.1

4. ETAPES PROC

4.1. principes

L'étape PROC permet d'effectuer des analyses sur une table SAS en produisant des sorties prédéfinies et/ou une table SAS résultante.

L'étape PROC peut faire suite à une étape DATA ou être exécutée indépendamment.

La syntaxe générale d'une procédure PROC est la suivante :

```
PROC nom_de_la_procédure  
    [DATA = [nom_logique.]nom_de_la_table]  
    [options];  
    [Sous-procédures;]
```

Toute procédure opère toujours sur une table SAS. Celle-ci est précisée dans les options de la procédure avec l'instruction DATA =. Cette option ne doit pas être confondue avec l'étape DATA présentée au paragraphe 3.

Les procédures PROC disposent de sous-procédures communes dont les principales sont décrites au paragraphe 4.2. Certaines options globales du système SAS affectent les sorties des procédures (paragraphe 4.3). Parmi les procédures de base, on distingue des procédures d'édition (paragraphe 4.4), des procédures utilitaires (paragraphe 4.5) et des procédures statistiques (paragraphe 4.6).

Chaque procédure présentée est définie avec une syntaxe simplifiée et est accompagnée d'un exemple.

4.2. Sous-procédures communes

- Sélection des variables à traiter :

```
VAR nom(s)_de_la(ou des)_variable(s)_sélectionnée(s) ;
```

- Définition des variables clefs :

```
BY nom(s)_de_la(ou des)_variable(s)_clefs ;
```

L'utilisation d'une sous-commande **BY** dans une procédure nécessite le tri préalable de la table par la procédure **PROC SORT**.

- Identification des variables de fréquences :

```
FREQ nom(s)_de_la(ou des)_variable(s) ;
```

- Identification des variables de pondération :

```
WEIGHT nom(s)_de_la(ou des)_variable(s) ;
```

- Identification des variables de classe :

```
CLASS nom(s)_de_la(ou des)_variable(s) ;
```

- Création d'une table SAS résultat :

```
OUTPUT OUT = nom_de_la_table ;
```

- Titre des variables :

```
LABEL nom_de_la_variable = "expression associée";
```

- Titre des résultats :

```
TITLEn "titre n";
```

n représente le numéro d'ordre du titre (de 1 à 10).

- Note de bas de page :

```
FOOTNOTEn "note n";
```

n représente le numéro de la note de bas de page (de 1 à 10).

4.3. Options globales

- Pagination ou non des résultats :

OPTIONS NUMBER;

ou

OPTIONS NONNUMBER;

- Remise du compteur de pagination à une valeur donnée :

OPTIONS PAGENO = *valeur* ;

- Centrage ou non des résultats :

OPTIONS CENTER;

ou

OPTIONS NOCENTER;

- Identification caractère de la valeur manquante :

OPTIONS MISSING = "i";

i est le caractère devant représenter le code de valeur manquante.

- Nombre de caractères par ligne :

OPTIONS LINESIZE = *nombre_de_caractères* ;

- Nombre de lignes par page :

OPTIONS PAGESIZE = *nombre_de_lignes* ;

- Présence ou non de la date sur une page :

OPTIONS DATE ;

ou

OPTIONS NODATE;

Il est également possible de changer le format des sorties des procédures en utilisant l'instruction ODS (*Output Delivery System*), qui permet entre autre la production de résultats en format HTML ou RTF.

4.4. Procédures d'édition

- Impression

```
PROC PRINT [DATA = [lsas.]nom_de_la_table];
  [WHERE nom_de_la_variable =
    expression_logique_de_sélection;]
  [BY ...;]
  [SUM nom(s)_de_la(ou des) variables
    sur_lesquelles_un_total_est_calculé;]
  [VAR ...;]
```

Programme 8

```
LIBNAME lsas "a:\" ;

PROC PRINT DATA=lsas.sol ;
  WHERE nomcult = "FROM" ;
  SUM surface ;
  VAR nomparc surface ;

RUN ;
```

Résultat 8

Obs	NOMPARC	SURFACE
7	LONG-FOSSE	20.73
11	THOREMBAIS	10.56
14	DV FERME	6.00
17	LEMPEREUR	2.00
19	DV FERME	2.60
		=====
		41.89

- Graphe

```
PROC PLOT [DATA = [lsas.]nom de la table];
  PLOT variable_y * variable_x [/options];
ou
  PLOT variable_y * variable_x = 'un caractère'
  [/options];
ou
  PLOT variable_y * variable_x = variable_alphanumérique
  _identifiant_les_couples_(xi,yi) [/options];
```

Les options possibles sont entre autre :

- pour spécifier l'échelle de l'axe des ordonnées :

```
/VAXIS = première_valeur TO dernière_valeur BY  
accroissement;
```

- pour spécifier l'échelle de l'axe des abscisses :

```
/HAXIS = première_valeur TO dernière_valeur BY  
accroissement;
```

- pour spécifier la position d'un axe horizontal :

```
/HREF = valeur_de_la_position;
```

- pour spécifier la position d'un axe vertical :

```
/VREF = valeur_de_la_position;
```

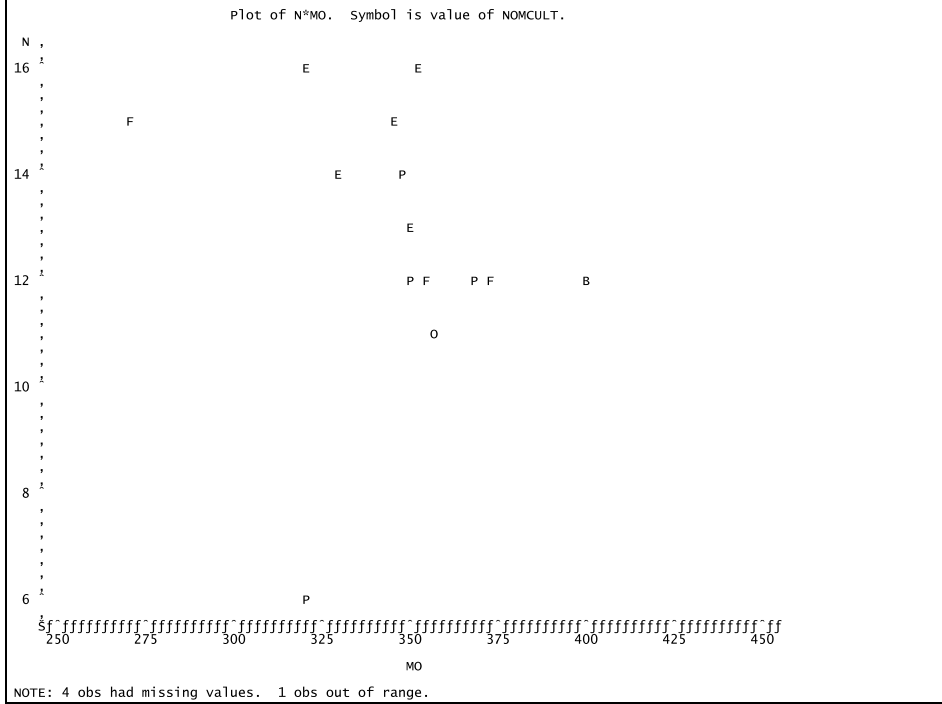
- pour superposer deux graphes :

```
/OVERLAY;
```

Programme 9

```
LIBNAME lsas "a:\" ;  
PROC PLOT DATA=lsas.sol ;  
    PLOT N * Mo = nomcult /  
    VAXIS = 6 TO 16 BY 2 HAXIS = 250 TO 450 BY 25;  
RUN ;
```

Résultat 9



- Diagramme

```
PROC CHART [DATA = [lsas.]nom_de_la_table];
  type_de_représentation nom_de_la_variable
  [/options];
```

Les différents types de représentations sont repris ci-dessous :

VBAR	histogramme en bâtons
PIE	diagramme en secteurs
BLOCK	diagramme en blocs
HBAR	histogramme horizontal
STAR	diagramme en étoile

Comme options, il est possible :

- de demander la prise en compte de données manquantes :

```
MISSING;
```

- de définir les points centraux des classes :

```
MIDPOINTS = première_valeur TO dernière_valeur BY
  accroissement;
```

ou

```
MIDPOINTS = "caractère_1" "caractère_2" ...;
```

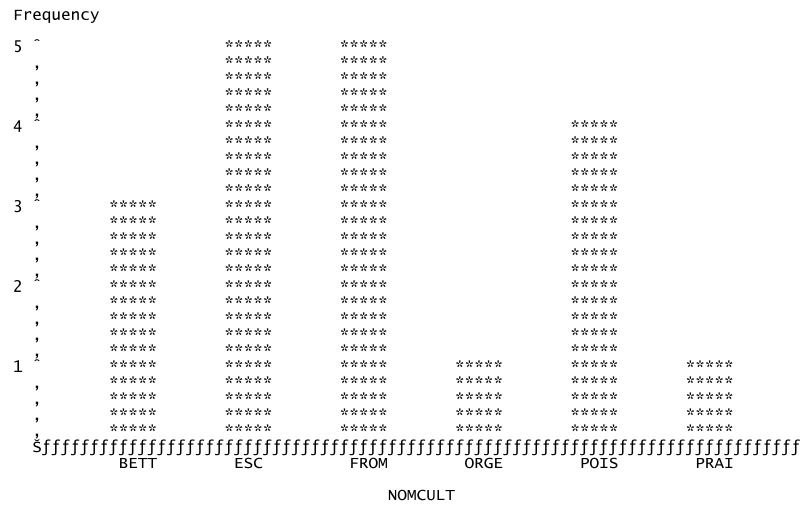
- d'imposer les valeurs représentées par l'axe des ordonnées :

fréquences absolues (par défaut)	TYPE = FREQ
fréquences relatives	TYPE = PERCENT
fréquences absolues cumulées	TYPE = CFREQ
fréquences relatives cumulées	TYPE = CPERCENT
sommes par classe	TYPE = SUM
moyennes par classe	TYPE = MEAN

Programme 10

```
LIBNAME lsas "a:\" ;  
PROC CHART DATA=lsas.sol ;  
    VBAR nomcult ;  
RUN ;
```

Résultat 10



En ce qui concerne la réalisation de graphiques, SAS propose un autre module, le logiciel SAS/GRAPH, qui permet d'obtenir des graphiques de qualité supérieure.

4.5. Procédures utilitaires

- Tri ascendant

```
PROC SORT [DATA = [lsas.]nom_de_la_table]
          [OUT = [lsas.]nom_de_la_table][NODUPRECS];
BY nom_variable(s);
```

L'option NODUPRECS conserve un seul exemplaire des observations identiques. Par observations identiques, il faut entendre toutes les observations qui possèdent les mêmes valeurs pour chacune des variables de la table.

L'option OUT permet la sauvegarde du tri dans une nouvelle table dont le nom diffère de celui défini dans DATA =. Par défaut, la table triée est enregistrée sous le même nom que celui repris après DATA =.

Programme 11

```
LIBNAME lsas "a:\\" ;
PROC SORT DATA = lsas.SOL
          OUT = lsas.tri ;
          BY nomcult ;
PROC PRINT DATA = lsas.tri ;
RUN ;
```

Résultat 11

obs	DATECREA	NPARCULT	NOMPARC	NOMCULT	SURFACE	TYPESOL	N	P	K	CAOH	MO	PH
1	6210	5	DV FERME		15.88	limon lourd argileux	13	22	12	2.00	400	7.5
2	11932	8	LEMPEREUR		4.42	argile lourd	13	12	25	2.30	350	6.6
3	6210	3	CARRIERE	BETT	6.81	limon argile	12	18	12	2.00	400	7.5
4	13410	18	LEMPEREUR	BETT	2.42							
5	13802	20	DV FERME	BETT	13.28	limon lourd argileux	17	13	2.75	359	7.3	
6	11932	1	BURLET	ESC	7.68	argileux dessus doux	14	19	7	2.20	330	7.5
7	11323	6	JODOIGNE	ESC	6.00	limon argile sable	15	15	16	2.30	345	7.5
8	6210	10	90 ARES	ESC	0.93	limon lourd argileux	16	17	6	2.24	320	7.5
9	13028	13	LONG FOSSE	ESC	4.73	limon argile sable	16	15	17	2.40	352	7.4
10	13418	15	DV FERME	ESC	6.00	limon lourd argileux	13	15	13	2.71	349	7.1
11	6210	7	LONG-FOSSE	FROM	20.73	limon lourd argileux	15	15	6	2.20	270	7.2
12	6210	11	THOREMBAIS	FROM	10.56	argile limon	12	17	6	2.24	355	7.5
13	13418	14	DV FERME	FROM	6.00	limon lourd argileux	12	16	15	2.38	372	7.2
14	13379	17	LEMPEREUR	FROM	2.00	argile lourd	17	14	15	2.82	358	7.1
15	13802	19	DV FERME	FROM	2.60	limon lourd argileux	15	12	14	2.58		7.3
16	13028	12	LONG FOSS	ORGE	16.00	limon moyen	11	14	16	2.82	356	6.8
17	11932	2	CAMPAGNETTE	POIS	3.89	argile lourd	6	17	13	2.76	320	7.3
18	6210	4	DEWAET	POIS	1.81	limon humide		12	19	6.60	310	6.5
19	13418	16	DV FERME	POIS	3.88	limon lourd argileux	12	15	17	2.73	369	7.2
20	13788	21	ZELIA	POIS	1.25	limon moyen	12	25	12	2.30	350	7.0
21	6210	9	PRAIRIE	PRAI	2.08	limon argile	14	11	19	2.17	347	7.1

- Ajout d'observations

```
PROC APPEND BASE = [lsas.]nom_de_la_table_SAS_initiale
            NEW = [lsas.]nom_de_la_table_SAS_à_ajouter;
```

Programme 12

```
LIBNAME lsas "a:\" ;
PROC APPEND BASE=lsas.froment NEW=lsas.escourg ;
PROC PRINT DATA = lsas.froment ;
RUN ;
```

Résultat 12

Obs	DATECREA	NPARCULT	NOMPARC	NOMCULT	SURFACE	TYPESOL	N	P	K	CAOH	MO	PH
1	6210	7	LONG-FOSSE	FROM	20.73	limon lourd argileux	15	15	6	2.20	270	7.2
2	6210	11	THOREMBAIS	FROM	10.56	argile limon	12	17	6	2.24	355	7.5
3	13418	14	DV FERME	FROM	6.00	limon lourd argileux	12	16	15	2.38	372	7.2
4	13379	17	LEMPEREUR	FROM	2.00	argile lourd	17	14	15	2.82	358	7.1
5	13802	19	DV FERME	FROM	2.60	limon lourd argileux	15	12	14	2.58		7.3
6	11932	1	BURLET	ESC	7.68	argileux dessus doux	14	19	7	2.20	330	7.5
7	11323	6	JODOIGNE	ESC	6.00	limon argile sable	15	15	16	2.30	345	7.5
8	6210	10	90 ARES	ESC	0.93	limon lourd argileux	16	17	6	2.24	320	7.5
9	13028	13	LONG FOSSE	ESC	4.73	limon argile sable	16	15	17	2.40	352	7.4
10	13418	15	DV FERME	ESC	6.00	limon lourd argileux	13	15	13	2.71	349	7.1

4.6. Procédures statistiques

- Paramètres descriptifs

```
PROC MEANS [DATA = [lsas.]nom de la table][PARAMÈTRES]
  MAXDEC = nombre_de_décimales_pour_les_paramètres
  (≤ 8) ;
BY ...;
FREQ ...;
OUTPUT OUT = [lsas.]nom_de_la_table ;
VAR ...;
WEIGHT ...;
```

Quelques paramètres spécifiques sont repris ci-dessous :

N	nombre d'observations
NMISS	nombre de valeurs manquantes
MEAN	moyenne
MIN	minimum
MAX	maximum
SUM	somme
STD	écart-type estimé
VAR	variance observée
CV	coefficient de variation

Programme 13

```
LIBNAME lsas "a:\";
PROC MEANS DATA=lsas.escourg N MIN MAX ;
      VAR N P K ;
RUN ;
```

Résultat 13

```

                                The MEANS Procedure
Variable      N      Minimum      Maximum
-----
N              5      13.0000000    16.0000000
P              5      15.0000000    19.0000000
K              5       6.0000000    17.0000000

```

- Tableaux (1 à 3 dimensions)

```
PROC TABULATE [DATA = [lsas.]nom_de_la_table];
CLASS noms_des_variables_de_la_hiérarchie_du_classement;
TABLE variable_de_classement_1
      [, variable_de_classement_2
      [, variable_de_classement_3]]
      [, FONCTION * (noms_des_variables)];
VAR ...;
BY ...;
```

Fonctions : N, MEAN, SUM, MIN, MAX, VAR, STD, CV, etc.

Programme 14

```
LIBNAME lsas "a:\" ;
PROC TABULATE DATA=lsas.sol ;
    CLASS nomcult ;
    VAR surface ;
    TABLE nomcult N*surface ;

RUN ;
```

Résultat 14

```
..ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff..fffffffffffffff+
|ffffffffffffff...ffffffffffff...ffffffffffff...ffffffffffff...ffffffffffff...ffffffffffff%
|          NOMCULT
|BETT      ESC      FROM      ORGE      POIS      PRAI      N
|          N          N          N          N          N          SURFACE
|          3.00,     5.00,     5.00,     1.00,     4.00,     1.00,     19.00,
|Sffffffffffff<ffffffffffff<ffffffffffff<ffffffffffff<ffffffffffff<ffffffffffff<ffffffffffffe
```

Programme 15

```
LIBNAME lsas "a:\" ;
PROC TABULATE DATA=lsas.sol ;
    CLASS nomcult ;
    VAR surface ;
    TABLE nomcult, SUM*(surface);

RUN ;
```

Résultat 15

```
.,fffffffffffffffffffffffff...fffffffffffffff+
,
, Sum
, ffffffffffffffffff%
, SURFACE
, ffffffffffffffffff%
, NOMCULT
, ffffffffffffffffff%
, BETT 22.51,
, ffffffffffffffffff%
, ESC 25.34,
, ffffffffffffffffff%
, FROM 41.89,
, ffffffffffffffffff%
, ORGE 16.00,
, ffffffffffffffffff%
, POTS 10.83,
, ffffffffffffffffff%
, PRAT 2.08,
, ffffffffffffffffff%
, ffffffffffffffffff%
, ffffffffffffffffff%
```

5. INFORMATIONS COMPLEMENTAIRES

Seul le module de Base SAS a été introduit dans cette note. Des informations complètes sur ce module sont disponibles dans la documentation SAS [X, 1999]. En complément, nous citons deux autres références introduisant les bases du langage SAS : REMOND-TIEDREZ *et al.* [1991] et CODY & PASS [1997].

Selon le domaine d'application, différents modules spécifiques peuvent être ajoutés au module de base.

Comme nous l'avons déjà précisé, il existe un module spécialisé pour la réalisation de graphiques de qualité supérieure : SAS/GRAPH. Une introduction au logiciel SAS/GRAPH est donnée par PRÉVOT [1995].

Pour la réalisation d'analyses statistiques plus élaborées (analyse de la variance, régression, analyse multidimensionnelle, ...), il est nécessaire d'utiliser le module SAS/STAT. Nous recommandons, pour l'analyse statistique avec SAS, le livre de CODY & SMITH [1997].

D'autres modules ont des tâches statistiques spécifiques tels que SAS/QC pour le contrôle statistique de la qualité, SAS/OR pour la recherche opérationnelle et SAS/ETS pour le traitement de séries chronologiques.

Nous pouvons également citer le module SAS/IML pour le langage de programmation matriciel et SAS/AF pour la conception d'applications conviviales.

Le système SAS possède également un logiciel plus convivial évitant à l'utilisateur la connaissance du langage SAS : *Enterprise Guide*. Cette application reprend les modules SAS/Base, SAS/STAT, SAS/QC et SAS/GRAPH.

6. BIBLIOGRAPHIE

CODY R., PASS R. [1997]. *SAS[®] programming by example*. Cary N.C., SAS Institute Inc., 337 p.

CODY R., SMITH J. [1997]. *Applied Statistics and the SAS[®] programming language, 4th Edition*. Upper Saddle River, Prentice Hall, 445 p.

PRÉVOT H. [1995]. Introduction au logiciel SAS/GRAPH. *Notes Stat. Inform.* (Gembloux) 95/2, 32 p.

REMOND-TIEDREZ E., LE MOIGNE L., SILBERGERG S. [1991]. *Le langage SAS*. Paris, Europstat, 204 p.

X.[1999]. *SAS OnlineDoc, version eight*. Cary N.C., SAS Institute Inc.

7. ANNEXE

L'exemple repris ci-dessous est extrait d'une base de données relative à la gestion de parcelles de plusieurs exploitations agricoles.

Les variables sont respectivement :

- le numéro d'ordre de l'observation ;
- le nom des parcelles, éventuellement des sous-parcelles ;
- l'abréviation du nom de la dernière culture, si celle-ci est connue ;
- la surface parcellaire exprimée en hectares ;
- la date de l'introduction de l'observation dans la table ;
- la texture pédologique globale de la parcelle;
- les résultats de la dernière analyse de sol pour les éléments N, P₂O₅, K₂O, CaOH, la teneur en matière organique et le pH.

1	BURLET	ESC	7.68	19920901	argileux dessus doux	14	19	7	2.2	330	7.5
2	CAMPAGNETTE	POIS	3.89	19920901	argile lourd	6	17	13	2.76	320	7.3
3	CARRIERE	BETT	6.81	19770101	limon argile	12	18	12	2	400	7.5
4	DEWAET	POIS	1.81	19770101	limon humide	.	12	19	6.6	310	6.5
5	DV FERME		15.88	19770101	limon lourd argileux	13	22	12	2	400	7.5
6	JODOIGNE	ESC	6	19910101	limon argile sable	15	15	16	2.3	345	7.5
7	LONG-FOSSE	FROM	20.73	19770101	limon lourd argileux	15	15	6	2.2	270	7.2
8	LEMPEREUR		4.42	19920901	argile lourd	13	12	25	2.3	350	6.6
9	PRAIRIE	PRAI	2.08	19770101	limon argile	14	11	19	2.17	347	7.1
10	90 ARES	ESC	.93	19770101	limon lourd argileux	16	17	6	2.24	320	7.5
11	THOREMBAIS	FROM	10.56	19770101	argile limon	12	17	6	2.24	355	7.5
12	LONG FOSS	ORGE	16	19950902	limon moyen	11	14	16	2.82	356	6.8
13	LONG FOSSE	ESC	4.73	19950902	limon argile sable	16	15	17	2.4	352	7.4
14	DV FERME	FROM	6	19960926	limon lourd argileux	12	16	15	2.38	372	7.2
15	DV FERME	ESC	6	19960926	limon lourd argileux	13	15	13	2.71	349	7.1
16	DV FERME	POIS	3.88	19960926	limon lourd argileux	12	15	17	2.73	369	7.2
17	LEMPEREUR	FROM	2	19960818	argile lourd	17	14	15	2.82	358	7.1
18	LEMPEREUR	BETT	2.42	19960918	
19	DV FERME	FROM	2.6	19971015	limon lourd argileux	15	12	14	2.58	.	7.3
20	DV FERME	BETT	13.28	19971015	limon lourd argileux	.	17	13	2.75	359	7.3
21	ZELIA	POIS	1.25	19971001	limon moyen	12	25	12	2.3	350	7.0

Une version électronique des données de l'exemple (fichiers *sol.dat* et *sol.xls*) est disponible à l'adresse <http://www.fsagx.ac.be/si/> rubrique *Publications*.

La collection

NOTES DE STATISTIQUE ET D'INFORMATIQUE

réunit divers travaux (documents didactiques, notes techniques, rapports de recherche, publications, etc.) émanant des services de statistique et d'informatique de la Faculté universitaire des Sciences agronomiques et du Centre de Recherches agronomiques de Gembloux (Belgique).

La liste des notes disponibles peut être obtenue sur simple demande à l'adresse ci-dessous :

*Faculté universitaire des Sciences agronomiques
Unité de Statistique et Informatique
Avenue de la Faculté d'Agronomie, 8
B-5030 GEMBLoux (Belgique)
E-mail : statinfo@fsagx.ac.be*

Plusieurs notes sont directement accessibles à l'adresse Web suivante, section Publications :

<http://www.fsagx.ac.be/si/>

En relation avec certaines notes, des programmes spécifiques sont également disponibles à la même adresse, section Macros.

Quelques titres récents sont cités ci-après :

- PALM R. [1999]. Indices d'aptitude des procédés de production. *Notes Stat. Inform.* (Gembloux) 99/5, 26 p.
- PALM R. [2000]. L'analyse de la variance multivariée et l'analyse canonique discriminante : principes et applications. *Notes Stat. Inform.* (Gembloux) 2000/1, 40 p
- PALM R., IEMMA A.F. [2002]. Conditions d'application et transformations de variables en régression linéaire. *Notes Stat. Inform.* (Gembloux) 2002/1, 34 p.
- BROSTAU X Y. [2002]. Introduction à l'environnement de programmation statistique R. *Notes Stat. Inform.* (Gembloux) 2002/2, 22 p.
- PALM R. [2003]. Le positionnement multidimensionnel : principes et application. *Notes Stat. Inform.* (Gembloux) 2003/1, 31 p.
- CLAUSTRIAUX J.J. [2006]. Un regard sur les activités de l'Unité de Statistique et Informatique de la Faculté universitaire des Sciences agronomiques de Gembloux (Belgique). *Notes Stat. Inform.* (Gembloux) 2006/1, 9 p.
- CARLETTI I., CLISSEN V., CLAUSTRIAUX J.J. [2006]. Introduction au logiciel Minitab sous WINDOWS. *Notes Stat. Inform.* (Gembloux) 2006/2, 23 p.