



HAL
open science

La macro SASSIC ou comment permettre aux utilisateurs de SAS de travailler rapidement avec SICLA

T. Cembrzynski

► **To cite this version:**

T. Cembrzynski. La macro SASSIC ou comment permettre aux utilisateurs de SAS de travailler rapidement avec SICLA. RT-0089, INRIA. 1987, pp.22. inria-00070077

HAL Id: inria-00070077

<https://inria.hal.science/inria-00070077>

Submitted on 19 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INRIA

UNITÉ DE RECHERCHE
INRIA-ROCCOUCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France
Tel. (1) 39.63.55.11

Rapports Techniques

N° 89

LA MACRO SASSIC OU COMMENT PERMETTRE AUX UTILISATEURS DE SAS DE TRAVAILLER RAPIDEMENT AVEC SICLA

Thierry CEMBRZYNSKI

NOVEMBRE 1987

**La Macro SASSIC ou comment permettre aux utilisateurs de
SAS de travailler rapidement avec SICLA**

**The SAS macro SASSIC or how it is possible for SAS users to
work quickly with the SICLA system.**

Thierry CEMBRZYNSKI
INRIA
Domaine de Voluceau
Rocquencourt
FRANCE

Résumé: Nous présentons dans ces quelques pages l'origine de la macro SAS de passage entre une structure de données SAS (un tableau SAS) et les éléments nécessaires à la construction d'une structure de données SICLA (un tableau de données et un fichier descriptif des variables). Celle-ci a été développée pour permettre la mise en œuvre rapide des traitements statistiques de très volumineuses données, sur deux ordinateurs très différents IBM 3090 sous MVS/XA et un BULL DSP8 MULTICS. Nous montrerons également un exemple d'utilisation de cette procédure.

Abstract: We submit in this paper the origin of the SAS macro used for transforming a SAS dataset in the components (a data set formatted and a dictionary file of the variables). This program was developed for starting quickly data analysis over two different computers such as an IBM 3090 (MVS/XA) and a BULL DPS8 (MULTICS). We will present a demonstration of this macro.

Le logiciel SAS est un produit très puissant et bien connu des statisticiens. Celui-ci permet à partir de données disponibles sur des fichiers, de les gérer, et d'effectuer différents calculs et divers traitements statistiques.

Les traitements statistiques peuvent aussi bien relever de la statistique descriptive simple (comptage, moyenne, écart-type, corrélation, histogrammes) que de traitements statistiques plus avancés (régression, analyse de la variance, discrimination, classification automatique).

Il permet également de trier les données, de les concaténer, de les interclasser, et également de les représenter graphiquement avec son sous-produit graphique SAS GRAPH.

Malgré tout son potentiel (qui est très grand, notons le bien) SAS souffre quand même de quelques manques liés à son origine américaine très orientée modélisation (Modèle Linéaire, Séries Chronologiques). L'école française d'analyse des données de Benzécri y est totalement absente (pas d'Analyse Factorielle des Correspondances, ni d'Analyse des Correspondances Multiples) que ce soit pour les versions 5 et 6, et il en est de même pour la classification sur données qualitatives. Ainsi le dépouillement d'enquêtes "à la française" devient alors avec SAS, très délicat.

Sans vouloir faire un procès, il n'en demeure pas moins que dans d'autres domaines SAS soit également peu satisfaisant; remarquons les différences de qualité entre les représentations graphiques de haute qualité des procédures de construction d'histogrammes (proc CHART), avec celle des arbres de classifications hiérarchiques (proc TREE).

Pour l'utilisateur qui souhaite alors effectuer des traitements non prévus par SAS, ou dont les sorties ne lui conviennent pas (ce qui était notre cas en classification hiérarchique), trois possibilités s'offrent à lui.

1/ Intégrer dans SAS un programme FORTRAN de son cru par exemple les méthodes d'analyses factorielles des correspondances, ou encore un tracé de dendrogramme. Cette opération est cependant très délicate, et n'est pratiquement possible que par les administrateurs qui assurent sur l'ordinateur la gestion système SAS; cela sous-entend en effet d'avoir à sa disposition des routines d'accès à SAS dont l'utilisateur "classique" (programmeur d'applications) ne dispose pas.

Ce travail assez long présente cependant de nombreux avantages, l'utilisateur peut alors:

- bénéficier de tout l'environnement SAS (base de données) concernant la manipulation et la transformation des données;
- supprimer les différents types de fichiers de travail, et stocker les résultats de ses analyses sous la forme de tableaux SAS;
- appeler ses propres méthodes dans un langage proche de celui du statisticien, avec souvent un certain nombre d'options par défaut qui suppriment les traditionnelles cartes paramètres du FORTRAN, sources d'erreurs fréquentes.

2/ Développer avec le langage de MACRO la procédure statistique souhaitée. C'est certainement pour l'utilisateur courant la meilleure solution, cependant trois problèmes se posent alors:

- il est indispensable d'avoir une bonne connaissance du langage de MACRO développé par SAS; lequel n'est pas évident. A cela s'ajoute un long travail de programmation qui devra être nécessairement suivi par des tests de validité de la procédure;
- il faut que l'application soit "programmable" sous forme de MACRO; en effet orienté statistique, le langage SAS diffère profondément des autres langages évolués de programmation comme FORTRAN ou PASCAL, en particulier s'il est assez délicat de traiter des vecteurs lignes, il devient en dehors de la procédure MATRIX impossible de manipuler des matrices;
- il faut que ce travail de programmation soit "rentable"; développer par exemple une macro d'Analyse des Correspondances pour un seul traitement n'est pas valable, voir n'a pas de sens.

3/ Développer ou utiliser un programme SAS existant qui permettra à partir d'une structure de données SAS (SAS data set), d'organiser ces mêmes données pour pouvoir les traiter rapidement avec d'autres logiciels d'analyse des données; cela revient alors à construire ou à utiliser une procédure d'interfaçage. C'est cette troisième solution que nous proposons à travers la procédure SASSIC de passage de SAS à SICLA.

Le logiciel SICLA (Système Interactif de Classification Automatique, RALAMBONDRAINNY thèse d'état 1986) développé à l'INRIA est un autre logiciel d'analyse des données; certes moins puissant que le logiciel SAS, mais qui toutefois possède quelques caractéristiques très intéressantes.

SICLA est très orienté sur la Classification Automatique, on y trouve en

particulier toutes les variantes de l'algorithme des Nuées Dynamiques que se soit sur variables quantitatives ou qualitatives, éventuellement à nombre de classes non obligatoirement imposé par l'utilisateur (CEMBRZYNSKI 1987). L'utilisateur dispose aussi d'un programme de construction de partitions centrales (CELEUX 1984) permettant de construire à partir des formes fortes issues d'un grand nombre d'expériences des Nuées Dynamiques, une partition "moyenne". SICLA dispose également de procédures hiérarchiques et divisives (segmentation) comme le programme DNP qui réalise une discrimination non paramétrique (CELEUX et LECHEVALLIER 1980). Les techniques linéaires de discrimination, de régression et d'analyse de la variance y sont également présentes.

SICLA est couplé au logiciel SPAD (Système Portable pour l'Analyse des Données) qui est quant à lui, très orienté vers le dépouillement d'enquêtes. Enfin SICLA dispose d'outils pour la manipulation des données via ses propres procédures ou avec un SGBD relationnel (PEPIN), et d'autres écrits avec les normes GKS, pour la représentation graphique des données.

L'origine de la procédure SASSIC est à rechercher dans le fait que nous fûmes dans le cadre de nos travaux, amenés à utiliser ces deux logiciels sur deux ordinateurs très différents: SAS sur l'IBM 3090 sous MVS/XA de l'EDF de Clamart, et SICLA sur le BULL DPS8 HB68 sous multics de l'INRIA. Il nous est alors apparu intéressant (pour gagner du temps) de disposer d'une procédure permettant de structurer un tableau SAS sous la forme des fichiers d'entrée de SICLA, fichiers que nous placions sur une bande pour les analyser ultérieurement sur le site de l'INRIA.

L'auteur de cette procédure est plus un statisticien qu'un véritable informaticien spécialiste des interfaces, c'est pourquoi l'idée d'une interface complète entre les deux logiciels SAS et SICLA ne l'a même pas effleuré. Il lui a semblé en revanche plus intéressant de développer une macro écrite en langage SAS, permettant d'organiser un tableau SAS (avec son descriptif) en un tableau de données formatées, et un fichier dictionnaire contenant les informations sur les variables (type de variables quantitatif, qualitatif, nombre de modalités...).

C'est cette procédure SASSIC que nous fournissons ici, qui permet de "passer" d'un tableau SAS aux éléments préliminaires à une structure de données SICLA. L'opération de retour ne pouvant pas être réalisée directement par l'auteur de SASSIC qui ne connaît pas le mode d'organisation

d'une structure de données SICLA. Cependant SICLA offre la possibilité de générer un tableau de données sous la forme d'un fichier séquentiel formaté à partir d'une structure de données (commande GENFIC).

Cette macro SASSIC est destinée à des utilisateurs "responsables" qui ne doivent pas chercher à la "planter".

Le descriptif des variables du tableau SAS doit être facultativement construit avant l'appel de la macro, si on souhaite disposer de libellés clairs et explicites. Il s'agit d'une PROC FORMAT respectant la syntaxe suivante:

Une valeur imposée "\$FMTVAR" pour associer au nom de chaque variable quantitative un libellé.

Pour chaque variable qualitative une valeur "NOM DE LA VARIABLE" qui définit pour chaque modalité, le libellé de celle-ci.

Ainsi nous avons:

```
PROC FORMAT;
{pour chaque variable quantitative du tableau SAS}.
value $FMTVAR
    'nom de la variable j ' = ' libellé de la variable j'
;
{pour chaque variable qualitative du tableau SAS}
VALUE    nom_de_la_variable_qualitative
    1     = 'libellé de la première modalité'
    ...
    i     = 'libellé de la i-ième modalité'
    ...
    n     = 'libellé de la n-ième modalité'
;
RUN;
```

NB: Les modalités des variables qualitatives doivent être numérotées de 1 à N. Il ne doit pas y avoir de modalité 0 pour utiliser le logiciel SICLA.

Si vous n'avez pas le courage de décrire les variables, les noms de variables seront utilisés pour générer les libellés SICLA des différentes variables.

Les résultats sont obtenus dans un fichier partitionné (sur le système IBM MVS/XA qui correspond à une librairie) associé à l'étiquette &DDNAME , dans lequel on trouve le fichier dictionnaire (membre DICTIO), et le fichier de données (DATAS). Ce dernier est formaté avec le format FORTRAN suivant (A4,1X,A40,(/6E12.5)).

Le fichier dictionnaire contient des abrégés en 4 caractères pour les différentes variables, ceux-ci sont construits de la façon suivante:

1/ pour les variables quantitatives: Q1 à Qn

2/ pour les variables qualitatives: 1Nxx à nNxx

où xx : désignent les numéros des différentes modalités.

Une limitation apparaît alors: au plus 99 variables qualitatives d'au plus 9 modalités, ou 9 variables qualitatives d'au plus 99 modalités.

CONCLUSIONS

Nous souhaitons que cette procédure puisse être utilisée librement par tous, l'exemple que nous donnons étant en lui même parfaitement clair nous ne serons pas inutilement bavard: on trouve un programme test, le fichier LOG correspondant, et les sorties (dictionnaire et fichier des données) qui se font dans un fichier partitionné affecté à l'étiquette DDNAME=SICLA1.

Le lecteur trouvera ensuite une session MULTICS d'utilisation SICLA d'enregistrement des données générées par la macro SASSIC.

L'auteur laisse la libre utilisation de cette macro, mais demande une citation d'utilisation si vous vous en servez. Il est possible d'en demander le source à messieurs CEMBRZYNSKI ou LECHEVALLIER du projet CLOREC de l'INRIA de Rocquencourt.

BIBLIOGRAPHIE

G. CELEUX (1984)

Approximation rapide et interprétation d'une partition centrale pour les algorithmes de partitionnement
INRIA Rapport de recherche n°302

G.CELEUX et Y. LECHEVALLIER (1980)

Méthodes de discrimination non paramétrique asymptotiquement efficaces au sens de BAYES
INRIA Rapport de recherche n°52

T. CEMBRZYNSKI (1987)

PONDER et MNDOPT une méthodologie en classification automatique. Son application à l'étude des effets de renforcements du réseau de transport à très haute tension d'EDF.
INRIA Rapport de recherche n°726

H. RALAMBONDRAINNY (1986)

Le système SICLA
Thèse d'état deuxième partie

SPAD Système Portable pour l'Analyse de Données
CESIA 82 rue de Sèvres 75007 PARIS

OPTIONS PS=60 LS=72 NOSOURCE;

*****;

*;

%MACRO SASSIC(DATA=_LAST_,VAR=_NUMERIC_,VNOM=,IID=,DICOVAR=,
DDNAME=SICLA);

* MACRO SAS DE PASSAGE DE SAS A SICLA

*

*

* DATA=TABLEAU SAS A TRANSFORMER POUR SICLA

* VAR =VARIABLES QUANTITATIVES

* VNOM=VARIABLES NOMINALES (QUALITATIVES)

* IID =UNE VARIABLE CONTENANT LE NOM DES INDIVIDUS (IDENTIF)

* DICOVAR=DICTIONNAIRE DES VARIABLES POUR OBTENIR DES

* LIBELLES EN CLAIR : PROC FORMAT A CONSTRUIRE AVANT

* L APPEL A LA MACRO SASSIC .IL DOIT CONTENIR IMPERATI-

* VEMENT LA VALUE : \$FMTVAR : NOM DES VARIABLES CONTINUES

* (OUI/NON)

* DDNAME=DDNAME DU FICHIER PARTITIONNE POUR LES DONNEES

* LE DICTIONNAIRE

*

* CR : T CEMBRZYNSKI

*

* TOUTE UTILISATION DE CE PROGRAMME A DES FINS LUCRATIVES

* EST RIGOREUSEMENT INTERDITE

*;

*****;

* RECOPIAGE DES DONNEES ;

*****;

*;

DATA _TAB;

SET &DATA;

LENGTH DEFAULT=4;

_IID=PUT(_N_,4.);

RUN;

*;

%LET DICO=0;

*****;

* VARIABLES QUANTITATIVES ;

*****;

*;

*****;

* DETERMINATION DU NOMBRE DE VARIABLES ACTIVES;

*****;

%IF %LENGTH(&VAR) NE 0 %THEN %DO;

*;

DATA _QUALI;

SET _TAB;

KEEP &VAR;

RUN;

*;

PROC TRANSPOSE DATA=_TAB(KEEP=&VAR OBS=1) OUT=_VARQ(KEEP=_NAME_);

RUN;

*;

DATA _NULL_;

SET _VARQ END=FIN;

IF FIN THEN DO;

VN=PUT(_N_,8.);

CALL SYMPUT('NVA',VN);

END;

RUN;

*;

%LET NVA=%EVAL(&NVA);

*;

```
*-----*  
* CREATION DU DICTIONNAIRE POUR LES VARIABLES QUANTITATIVES *  
*-----*  
%DO I=1 %TO &NVA;  
*;  
DATA _DICO;  
SET _VARQ(OBS=&I) END=FIN;  
LENGTH NV $ 40;  
LENGTH TY $ 2;  
LENGTH ID $ 4;  
LENGTH MD $ 2;  
LENGTH FI $ 2;  
IF FIN THEN DO;  
%IF %LENGTH(&DICOVAR) NE 0 %THEN %DO;  
NV=PUT(_NAME_,$FMTVAR.);  
%END;  
%ELSE %DO;  
NV=PUT(_NAME_,40.);  
%END;  
TY='00';  
ID=PUT(COMPRESS('Q' || PUT(&I,3.)),4.);  
FI=' 1';  
MD=' 1';  
OUTPUT;  
END;  
KEEP TY ID NV FI MD;  
RUN;  
*;  
*-----*  
* FUSION DU DICTIONNAIRE PARTIEL *  
*-----*  
%IF &I=1 %THEN %DO;  
DATA DICO_;  
SET _DICO;  
RUN;  
*;  
%LET DICO=1;  
%END;  
%ELSE %DO;  
DATA DICO_;  
SET DICO_ _DICO;  
RUN;  
%END;  
%END;  
*;  
*-----*  
* IMPRESSION DU DICTIONNAIRE QUANTITATIF *  
*-----*  
PROC PRINT DATA=DICO_;  
TITLE1 *-----*;  
TITLE2 * DICTIONNAIRE DES VARIABLES QUANTITATIVES *;  
TITLE3 *-----*;  
RUN;  
TITLE ;  
*;  
*-----*  
* VARIABLES NOMINALES *  
*-----*  
*;  
%IF %LENGTH(&VNOM) NE 0 %THEN %DO;  
*;  
*-----*  
* CONSTRUCTION DES DONNEES :QUANTITATIVES,QUALITATIVES *  
*-----*  
%IF %LENGTH(&VAR) NE 0 %THEN %DO;
```

```
DATA _QUALI;
  MERGE _QUALI _TAB(KEEP=&VNOM);
RUN;
%END;
%ELSE %DO;
  DATA _QUALI;
    SET _TAB;
    KEEP &VNOM;
  RUN;
%END;
*;
PROC TRANSPOSE DATA=_TAB(KEEP=&VNOM OBS=1) OUT=_VNOM(KEEP=_NAME_);
RUN;
*;
DATA _NULL_;
  SET _VNOM END=FIN;
  IF FIN THEN DO;
    VN=PUT(_N_,8.);
    CALL SYMPUT('VNO',VN);
  END;
RUN;
*;
%LET NOMI=%EVAL(&VNO);
*;
%DO I=1 %TO &NOMI;
  *;
  DATA _NULL_;
    SET _VNOM(OBS=&I) END=FIN;
    LENGTH NV $ 10;
    IF FIN THEN DO;
      NV=TRIM(_NAME_)||'.';
      NC=INDEX(NV,'. ');
      CALL SYMPUT('NOMVAR',_NAME_);
      CALL SYMPUT('FMTNM',SUBSTR(NV,1,NC));
    END;
  RUN;
  %LET NOM&I=&NOMVAR;
  %LET FMT&I=&FMTNM;
  *;
%END;
*;
*-----*;
* VARIABLES NOMINALES : NOMBRE DE MODALITES *-----*;
PROC FREQ DATA=_TAB;
  %DO I=1 %TO &NOMI;
    TABLES &&NOM&I / OUT=MOD&I(KEEP=COUNT &&NOM&I) NOPRINT;
  %END;
RUN;
*;
%DO I=1 %TO &NOMI;
  *;
  %LET NOMVAR=&&NOM&I;
  %LET FMTNM=&&FMT&I;
  *;
  DATA _NULL_;
    SET MOD&I END=FIN;
    IF FIN THEN DO;
      NV=PUT(&NOMVAR,8.);
      CALL SYMPUT('NBMOD',NV);
    END;
  RUN;
  *;
  %LET NBMOD=%EVAL(&NBMOD);
  *;
*-----*;
```

* VARIABLES NOMINALES : CREATION DU CHAMP DE L VARIABLE ;

----- ;

```
DATA _DICO;
  LENGTH NV $ 40;
  LENGTH TY $ 2;
  LENGTH ID $ 4;
  LENGTH MD $ 2;
  LENGTH FI $ 2;
  NOM=SYMGET('NOMVAR');
  %IF %LENGTH(&DICOVAR) NE 0 %THEN %DO;
    NV=PUT(NOM,$FMTVAR.);
  %END;
  %ELSE %DO;
    NV=PUT(NOM,40.);
  %END;
  TY='00';
  ID=PUT(COMPRESS(PUT(&I,3.)||'N'),4.);
  FI=' 2';
  %IF &NBMOD > 9 %THEN %DO;
    MD=PUT(SYMGET('NBMOD'),2.);
  %END;
  %ELSE %DO;
    MD=' '||PUT(SYMGET('NBMOD'),1.);
  %END;
  OUTPUT;
  KEEP TY ID NV FI MD;
RUN;
*;
```

----- ;

* VARIABLES NOMINALES : CREATION DU CHAMP DES MODALITES (9 MAXI) ;

* SI IL Y A PLUS DE 9 VARIABLES QUALITATIVES ;

* VARIABLES NOMINALES : CREATION DU CHAMP DES MODALITES (99 MAXI) ;

* SI IL Y A MOINS DE 9 VARIABLES QUALITATIVES ;

----- ;

```
%IF &NOMI > 9 %THEN %DO;
  %IF &NBMOD > 9 %THEN %DO;
    %PUT ERREUR AVEC &NOMVAR QUI A &NBMOD MODALITES;
  %END;
%END;
%ELSE %DO;
  %IF &NBMOD > 99 %THEN %DO;
    %PUT ERREUR AVEC &NOMVAR QUI A &NBMOD MODALITES;
  %END;
%END;
*;
```

```
DATA _DIC;
  LENGTH NV $ 40;
  LENGTH TY $ 2;
  LENGTH ID $ 4;
  LENGTH MD $ 2;
  LENGTH FI $ 2;
  RETAIN NOM;
  %DO J=1 %TO &NBMOD;
    NOM=SYMGET('NOMVAR');
    MMD=&J;
    %IF %LENGTH(&DICOVAR) NE 0 %THEN %DO;
      NV=PUT(MMD,&FMTNM);
    %END;
    %ELSE %DO;
      NV=PUT(COMPRESS(PUT(&J,2.)||'.'||NOM),40.);
    %END;
    TY=' ' ;
    %IF &NOMI > 9 %THEN %DO;
      ID=PUT(COMPRESS(PUT(&I,2.)||'N'||PUT(&J,1.)),4.);
    %END;
    %ELSE %DO;
```

```

        ID=PUT(COMPRESS(PUT(&I,1.)||'N'||PUT(&J,2.)),4.);
    %END;
    FI='  ';
    MD='  ';
    %IF &NBMOD > 9 %THEN %DO;
        TY=PUT(&J,2.);
    %END;
    %ELSE %DO;
        TY='  '||PUT(&J,1.);
    %END;
    OUTPUT;
    %END;
    KEEP TY ID NV FI MD;
    RUN;
    *;
    DATA _DICO;
        SET _DICO _DIC;
    RUN;
    *;
* -----;
* FUSION DU DICTIONNAIRE PARTIEL
* -----;
    %IF &DICO=0 %THEN %DO;
        DATA DICO_;
            SET _DICO;
        RUN;
    *;
        %LET DICO=1;
    %END;
    %ELSE %DO;
        DATA DICO_;
            SET DICO_ _DICO;
        RUN;
    %END;
    %END;
%END;
* -----;
* IMPRESSION DU DICTIONNAIRE
* -----;
PROC PRINT DATA=DICO_;
TITLE1 ' *-----*';
TITLE2 ' * DICTIONNAIRE DES VARIABLES QUALITATIVES ET QUANTITATIVES *';
TITLE3 ' *-----*';
RUN;
TITLE ;
*;
* -----;
* VIDANGE DU DICTIONNAIRE
* -----;
DATA _NULL_;
    FILE &DDNAME(DICTIO);
    SET DICO_;
        PUT @1 TY 2. +1 ID 4. +1 NV 40. +1 FI 2. +1 MD 2.;
RUN;
*;
* -----;
* VIDANGE DU FICHIER DE DONNEES
* FORMAT DE SORTIE FORTRAN (A4,1X,A40,100(/,6E12.5))
* -----;
%IF %LENGTH(&IID) NE 0 %THEN %DO;
    DATA _NULL_;
        FILE &DDNAME(DATAS);
        MERGE _TAB(KEEP=&IID _IID) _QUALI;
        %IF %LENGTH(&VAR) NE 0 AND %LENGTH(&VNOM) NE 0 %THEN %DO;
            PUT @1 _IID 4. +1 &IID 40.
                (&VAR &VNOM) (/ @1 6 * E12.5);
        %END;
    %END;

```

```
%END;
%IF %LENGTH(&VAR) NE 0 AND %LENGTH(&VNOM) EQ 0 %THEN %DO;
  PUT @1 _IID 4. +1 &IID 40.
  (&VAR) (/ @1 6 * E12.5);
%END;
%IF %LENGTH(&VAR) EQ 0 AND %LENGTH(&VNOM) NE 0 %THEN %DO;
  PUT @1 _IID 4. +1 &IID 40.
  (&VAR) (/ @1 6 * E12.5);
%END;
RUN;
%END;
%ELSE %DO;
  DATA _NULL_;
  FILE &DDNAME(DATAS);
  MERGE _TAB(KEEP=_IID) _QUALI;
  %IF %LENGTH(&VAR) NE 0 AND %LENGTH(&VNOM) NE 0 %THEN %DO;
    PUT @1 _IID 4.
    (&VAR &VNOM) (/ @1 6 * E12.5);
  %END;
  %IF %LENGTH(&VAR) NE 0 AND %LENGTH(&VNOM) EQ 0 %THEN %DO;
    PUT @1 _IID 4.
    (&VAR) (/ @1 6 * E12.5);
  %END;
  %IF %LENGTH(&VAR) EQ 0 AND %LENGTH(&VNOM) NE 0 %THEN %DO;
    PUT @1 _IID 4.
    (&VAR) (/ @1 6 * E12.5);
  %END;
RUN;
%END;
* -----;
* DESTRUCTION DES TABLEAUX SAS CREES ;
* -----;
PROC DELETE DATA=_TAB _QUALI _DICO DICO_;
RUN;
*;
%MEND SASSIC;
*;
DATA ALIMENT;
INPUT _ALIM_ $ 1-8 UFL PDIN PDIE PHOSP CALCI TYP_CONC JSUP;
CARDS;
ORGE_2RG 1.00 70 88 3.4 0.8 1 1
BLE 1.05 75 86 3.2 0.6 1 1
MAIS 1.10 69 100 3.0 0.3 1 1
AVOINE 0.87 75 86 3.0 0.8 1 1
SEIGLE 1.02 59 78 3.8 0.6 1 1
ORGE_6RG 1.00 65 80 3.2 0.7 1 1
FEVEROLE 0.98 162 117 3.7 1.1 2 0
SOJA 1.18 261 199 5.0 2.3 2 0
T_SOJA44 1.03 306 230 6.2 3.3 3 1
T_SOJA50 1.06 340 251 6.6 2.9 3 1
T_COLZA 0.84 241 178 11.9 7.8 3 1
T_LIN 0.36 207 116 8.4 4.2 3 1
T_COTON 0.80 295 243 9.0 3.4 3 1
;
PROC FORMAT;
VALUE $FMTVAR 'UFL' = 'ENERGIE (UNITE FOURRAGERE LAIT/KG MS)'
'PDIN' = 'PROTEINE DIGESTIBLE DE L AZOTE G/KG MS'
'PDIE' = 'PROTEINE DIGESTIBLE DE L ENERGIE G/KG MS'
'PHOSP' = 'TENEUR EN PHOSPHORE G / KG MS'
'CALCI' = 'TENEUR EN CALCIUM G /KG MS'
;
VALUE TYP_CONC 1 = 'CONCENTRE EN GRAINES '
2 = 'CONCENTRE EN GRAINES AZOTEES'
3 = 'TOURTEAU'
;
RUN;
```

* _____;
* AVEC DICTIONNAIRE _____;
* _____;

%SASSIC(DATA=ALIMENT,VAR=UFL—CALCI,VNOM=TYP_CONC,IID=_ALIM_,
DICOVAR=OUI,DDNAME=SICLA1);RUN;

1 SAS(R) LOG OS SAS 5.16 MVS/XA TSO USER W6CHHTS STEP AL
NOTE: COPYRIGHT (C) 1984,1986 SAS INSTITUTE INC., CARY, N.C. 27511, U.S.A.
NOTE: SAS RELEASE 5.16 AT SAS INSTITUTE S.A.-TRIAL INSTALLATION (60005001).
NOTE: CPUID VERSION = 40 SERIAL = 140709 MODEL = 3090 .
CPUID VERSION = 40 SERIAL = 240709 MODEL = 3090 .
CPUID VERSION = 40 SERIAL = 340709 MODEL = 3090 .

NEWS

VERSION SAS 5.16 MAI 1987

- * Support Technique : D.E.R. CLAMART SERVICE I.M.A DEPT T.I.E.M.
- * C. Derquenne 47 65 59 07
- * B. Gerardin 47 65 50 70

1 OPTIONS PS=60 LS=72 NOSOURCE;
00010001

NOTE: DATA SET WORK.ALIMENT HAS 13 OBSERVATIONS AND 8 VARIABLES. 690 OBS /TRK

NOTE: FORMAT \$FMTVAR HAS BEEN OUTPUT.
NOTE: FORMAT TYP_CONC HAS BEEN OUTPUT.

NOTE: DATA SET WORK._TAB HAS 13 OBSERVATIONS AND 9 VARIABLES. 652 OBS/TRK

NOTE: DATA SET WORK._QUALI HAS 13 OBSERVATIONS AND 5 VARIABLES. 1066 OBS /TRK

NOTE: THE DATA SET WORK._VARQ HAS 5 OBSERVATIONS AND 1 VARIABLES. 3912 OBS/TRK.

NOTE: DATA SET WORK._DICO HAS 1 OBSERVATIONS AND 5 VARIABLES. 868 OBS/TRK

NOTE: DATA SET WORK.DICO_ HAS 1 OBSERVATIONS AND 5 VARIABLES. 868 OBS/TRK

NOTE: DATA SET WORK._DICO HAS 1 OBSERVATIONS AND 5 VARIABLES. 868 OBS/TRK

NOTE: DATA SET WORK.DICO_ HAS 2 OBSERVATIONS AND 5 VARIABLES. 868 OBS/TRK

NOTE: DATA SET WORK._DICO HAS 1 OBSERVATIONS AND 5 VARIABLES. 868 OBS/TRK

NOTE: DATA SET WORK.DICO_ HAS 3 OBSERVATIONS AND 5 VARIABLES. 868 OBS/TRK

NOTE: DATA SET WORK._DICO HAS 1 OBSERVATIONS AND 5 VARIABLES. 868 OBS/TRK

NOTE: DATA SET WORK.DICO_ HAS 4 OBSERVATIONS AND 5 VARIABLES. 868 OBS/TRK

NOTE: DATA SET WORK._DICO HAS 1 OBSERVATIONS AND 5 VARIABLES. 868 OBS/TRK

1 SAS(R) LOG OS SAS 5.16 MVS/XA TSO USER W6CHHTS

0

NOTE: DATA SET WORK.DICO_ HAS 5 OBSERVATIONS AND 5 VARIABLES. 868 OBS/TRK

NOTE: DATA SET WORK._QUALI HAS 13 OBSERVATIONS AND 6 VARIABLES. 902 OBS/

TRK

NOTE: THE DATA SET WORK._VNOM HAS 1 OBSERVATIONS AND 1 VARIABLES. 3912
OBS/TRK.

NOTE: THE DATA SET WORK.MOD1 HAS 3 OBSERVATIONS AND 2 VARIABLES. 2346
OBS/TRK.

NOTE: DATA SET WORK._DICO HAS 1 OBSERVATIONS AND 5 VARIABLES. 868 OBS/TRK

NOTE: DATA SET WORK._DIC HAS 3 OBSERVATIONS AND 5 VARIABLES. 868 OBS/TRK

NOTE: DATA SET WORK._DICO HAS 4 OBSERVATIONS AND 5 VARIABLES. 868 OBS/TRK

NOTE: DATA SET WORK.DICO_ HAS 9 OBSERVATIONS AND 5 VARIABLES. 868 OBS/TRK

NOTE: FILE SICLA1(DICTIO) IS:
DSNAME=W6CHHTS.S.F1(DICTIO),
UNIT=DISK,VOL=SER=EXTS0G,DISP=OLD,
DCB=(BLKSIZE=3200,LRECL=80,RECFM=FB)
NOTE: 9 LINES WERE WRITTEN TO FILE SICLA1(DICTIO).

NOTE: FILE SICLA1(DATAS) IS:
DSNAME=W6CHHTS.S.F1(DATAS),
UNIT=DISK,VOL=SER=EXTS0G,DISP=OLD,
DCB=(BLKSIZE=3200,LRECL=80,RECFM=FB)
NOTE: 26 LINES WERE WRITTEN TO FILE SICLA1(DATAS).

NOTE: SAS INSTITUTE, SAS CIRCLE, BOX 8000, CARY, N.C. 27511-8000

1 ORGE_2RG
1.00000E+00 7.00000E+01 8.80000E+01 3.40000E+00 8.00000E-01 1.00000E+00
2 BLE
1.05000E+00 7.50000E+01 8.60000E+01 3.20000E+00 6.00000E-01 1.00000E+00
3 MAIS
1.10000E+00 6.90000E+01 1.00000E+02 3.00000E+00 3.00000E-01 1.00000E+00
4 AVOINE
8.70000E-01 7.50000E+01 8.60000E+01 3.00000E+00 8.00000E-01 1.00000E+00
5 SEIGLE
1.02000E+00 5.90000E+01 7.80000E+01 3.80000E+00 6.00000E-01 1.00000E+00
6 ORGE_6RG
1.00000E+00 6.50000E+01 8.00000E+01 3.20000E+00 7.00000E-01 1.00000E+00
7 FEVEROLE
9.80000E-01 1.62000E+02 1.17000E+02 3.70000E+00 1.10000E+00 2.00000E+00
8 SOJA
1.18000E+00 2.61000E+02 1.99000E+02 5.00000E+00 2.30000E+00 2.00000E+00
9 T_SOJA44
1.03000E+00 3.06000E+02 2.30000E+02 6.20000E+00 3.30000E+00 3.00000E+00
10 T_SOJA50
1.06000E+00 3.40000E+02 2.51000E+02 6.60000E+00 2.90000E+00 3.00000E+00
11 T_COLZA
8.40000E-01 2.41000E+02 1.78000E+02 1.19000E+01 7.80000E+00 3.00000E+00
12 T_LIN
3.60000E-01 2.07000E+02 1.16000E+02 8.40000E+00 4.20000E+00 3.00000E+00
13 T_COTON
8.00000E-01 2.95000E+02 2.43000E+02 9.00000E+00 3.40000E+00 3.00000E+00

00	Q1	ENERGIE (UNITE FOURRAGERE LAIT/KG MS)	1	1
00	Q2	PROTEINE DIGESTIBLE DE L AZOTE G/KG MS	1	1
00	Q3	PROTEINE DIGESTIBLE DE L ENERGIE G/KG MS	1	1
00	Q4	TENEUR EN PHOSPHORE G / KG MS	1	1
00	Q5	TENEUR EN CALCIUM G /KG MS	1	1
00	1N	TYP_CONC	2	3
1	1N1	CONCENTRE EN GRAINES		
2	1N2	CONCENTRE EN GRAINES AZOTEES		
3	1N3	TOURTEAU		

```
*****
* SICLA              Cembrzynski          Date : 13/11/87 *
* S.D. :              Archive :          *
* Listing : lis10_20  Mode interactif    *
* version 2.2 : (faire help sicla)       *
* _____ *
*****
* SICLA              Commande SICLA      Version Multics 1.0 *
*****
```

entrez le nom de la structure de donnees choisie

aliment

———> fichier : aliment.sdo

ce sont des nouvelles donnees a enregistrer ?

oui

```
*****
* donnees : aliment.sdo  individus : 0 * variables : 0 *
* _____ *
*                               quantitatives : 0 qualitatives : 0 *
*****
```

que desirez-vous faire ?

- 1 : saisie du descriptif en conversationnel
(a deconseiller si nombre de variables important)
- 2 : saisie du descriptif a partir d'une autre s.d.
- 3 : saisie du descriptif a partir d'un fichier dictionnaire
(a conseiller si nombre de variables important)

3

entrez un titre pour les donnees :

saisie des donnees transformees par SASSIC

entrez le nom du fichier dictionnaire

dico

entrez un majorant du nombre total de variables traitees :

10

entrez un majorant du nombre total de modalites traitees :

10

desirez-vous une impression du dictionnaire pour controle ?

oui

- 1. variable : Q1 : ENERGIE (UNITE FOURRAGERE LAIT/KG MS) type : 1
modalites = 1
- 2. variable : Q2 : PROTEINE DIGESTIBLE DE L AZOTE G/KG MS type : 1
modalites = 1
- 3. variable : Q3 : PROTEINE DIGESTIBLE DE L ENERGIE G/KG MS type : 1
modalites = 1
- 4. variable : Q4 : TENEUR EN PHOSPHORE G / KG MS type : 1

modalites = 1

5. variable : Q5 :TENEUR EN CALCIUM G /KG MS type : 1
modalites = 1

6. variable : 1N :TYP_CONC type : 2
modalites = 3

- 1.modalite :1N1 CONCENTRE EN GRAINES
- 2.modalite :1N2 CONCENTRE EN GRAINES AZOTEES
- 3.modalite :1N3 TOURTEAU

nombre total de variables : 6

variables quantitatives : 5
variables qualitatives : 1
nombre total de modalites : 3

fin de lecture du dictionnaire

les variables quantitatives comportent-elles des valeurs manquantes ?

non

```

*****
* nu * id * typs * mod * libelles des variables *
*****
* 1 * Q1      1      1 *ENERGIE (UNITE FOURRAGERE LAIT/KG MS) *
* 2 * Q2      1      1 *PROTEINE DIGESTIBLE DE L AZOTE G/KG MS *
* 3 * Q3      1      1 *PROTEINE DIGESTIBLE DE L ENERGIE G/KG MS *
* 4 * Q4      1      1 *TENEUR EN PHOSPHORE G / KG MS *
* 5 * Q5      1      1 *TENEUR EN CALCIUM G /KG MS *
* 6 * 1N      2      3 *TYP_CONC *
*****

```

fin de l enregistrement du descriptif des variables

que desirez-vous faire ?

- 1. modifier le descriptif des variables
- 2. passer a la phase d'enregistrement des donnees relatives aux individus

2

nombre maximum d'individus ?

40

nombre de caracteres de l'identificateur des individus ?

24

nombre de caracteres de l'identificateur abrege des individus

4

Type de format de lecture des donnees :

- 1 : lecture des donnees a l'aide d'un format FORTRAN
- 2 : lecture des donnees en format variable (faire ? pour plus de precisions)

1

entrez le nom du fichier sequentiel contenant les donnees :

datas

Impression des individus pour controle

(format d'impression g7.2)

image du premier enregistrement de votre fichier

0 1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890

1 ORGE_2RG

les ordres formats acceptables sont :

(n)*a1 pour l identificateur de l individu

f,g,e pour la lecture des valeurs reelles ,i est interdit

entrez le format fortran de lecture (2 lignes maxi)

(ex (4a1,10f3.0))

(24a1,/,6e12.5)

individu : 1 ORGE_2RG
1.0 70. 88. 3.4 .80 1.0

individu : 2 BLE
1.0 75. 86. 3.2 .60 1.0

—> ok ?

o

—>fin de lecture des donnees : 13 objets

Table with 6 columns: nu, id, typs, mod, libelles des variables. Rows include Q1 (ENERGIE), Q2 (PROTEINE DIGESTIBLE DE L AZOTE), Q3 (PROTEINE DIGESTIBLE DE L ENERGIE), Q4 (TENEUR EN PHOSPHORE), Q5 (TENEUR EN CALCIUM), and 1N (TYP_CONC).

entrer une commande : (aide : ? fin : /)

DESQAN

* SICLA Commande DESQAN Version 04/03/85 *
* donnees : aliment.sdo individus : 13 * variables : 6 *
* quantitatives : 5 qualitatives : 1 *

description des variables quantitatives

* variables * minimum * maximum * moyenne * ecart-type *

```
* Q1 * .36000 * 1.1800 * .94538 * .19681 *
* Q2 * 59.000 * 340.00 * 171.15 * 103.54 *
* Q3 * 78.000 * 251.00 * 142.46 * 64.784 *
* Q4 * 3.0000 * 11.900 * 5.4154 * 2.7329 *
* Q5 * .30000 * 7.8000 * 2.2154 * 2.0497 *
```

==== Fin de DESQAN ====

entrer une commande : (aide : ? fin : /)

DESQAL

```
*****
* SICLA Commande DESQAL Version 04/03/85 *
*****
* donnees : aliment.sdo individus : 13 * variables : 6 *
*
* quantitatives : 5 qualitatives : 1 *
*****
```

description des variables qualitatives

1. variable :1N TYP_CONC

- 1N1 : CONCENTRE EN GRAINES
- 1N2 : CONCENTRE EN GRAINES AZOTEES
- 1N3 : TOURTEAU

```
*****
* modl * nbre * % *
*****
* 1N1 * 6 * 46 *
* 1N2 * 2 * 15 *
* 1N3 * 5 * 38 *
*****
```

==== Fin de DESQAL ====

entrer une commande : (aide : ? fin : /)