Introduction au logiciel SAS

Paul-Antoine Chevalier*

Septembre 2009

Ce document est un support de cours pour les enseignements de méthodes quantitatives et d'introduction à l'économétrie du département de sciences sociales de l'ENS de Cachan.

Il existe de nombreux autres logiciels de statistiques et d'économétrie (SPSS, Stata, R, etc). SPSS est sans doute le logiciel le plus facile d'accès. Stata est très utilisé par les économistes [1]. R a l'avantage d'être un logiciel libre et polyvalent [2]. SAS a la particularité de travailler directement sur les bases de données sans les charger en mémoire. Cela permet de manipuler d'importantes bases de données (comme le recensement) qui sont difficilement manipulables par les logiciels concurrents.

Les remarques, conseils et corrections des lecteurs sont les bienvenus.

1 LES PREMIERS PAS

1.1 L'environnement SAS

L'interface de SAS comprend essentiellement cinq fenêtres.

La fenêtre EDITOR/EDITEUR (F5) contient les instructions SAS à exécuter. Elle dispose d'une fonction de coloration syntaxique permettant de reconnaître facilement les commandes SAS. Les programmes SAS peuvent être sauvegardés au format *.sas*.

La fenêtre LOG/JOURNAL (F6) contient toutes les informations relatives à l'exécution d'un programme (erreurs, avertissements, temps de calcul, etc). Il est nécessaire de consulter cette fenêtre avant de consulter les résultats. Les anomalies sont notées en vert et les erreurs en rouge. La log peut se sauvegarder avec l'extension *.log*.

La fenêtre OUTPUT/RESULTAT (F7) contient les résultats des instructions exécutées par SAS. Les résultats peuvent être sauvegardés avec l'extension *.lst*

La fenêtre RESULTS permet d'accéder facilement aux différentes pages de l'OUTPUT.

La fenêtre EXPLORER (CTRL + D) permet de naviguer à travers les librairies et d'accéder aux fichiers de données pour les visualiser dans le tableur SAS.

L'aide (F1)

*email : chevalier(a)ensae.fr, site web : http://pachevalier.free.fr

1. Pour une introduction au logiciel Stata, on peut se référer au manuel d'Antoine Bozio http://www.jourdan.ens.fr/~bozio/documents/stata.pdf.
2. http://cran.r-project.org/

1.2 Travailler dans SAS

Dans SAS, on travaille à l'aide de programmes contenant l'ensemble des instructions que l'on souhaite donner au logiciel. Un programme n'est rien d'autre qu'un fichier texte destiné à être lu et interprété par le logiciel.

A la différence d'autres logiciels qui lisent les programmes ligne par ligne, SAS lit le programme par étape. Une étape est composée d'un ensemble d'instructions. Chaque instruction se termine par un point-virgule. Chaque étape se termine par l'instruction `run ;` . Pour chaque instruction, on peut en général définir des options. On distingue les étapes de création et manipulation de bases de données (étapes DATA), des étapes de procédures (étapes PROC) permettant notamment une manipulation temporaire des données, un traitement statistique, la création de graphiques.

Pour exécuter un programme, il faut sélectionner la partie du programme que l'on souhaite exécuter et cliquer sur le bouton `RUN` (`soumettre` en version française) ou appuyer sur la touche F3. Par défaut, le programme entier est exécuté. Après avoir exécuté un programme, il faut vérifier dans la `log` que le programme n'a pas généré d'erreurs ou d'avertissement. Ensuite, on peut regarder l'`output` et/ou les bases de données créées par le programme.

Les programmes ont l'avantage de pouvoir être facilement réutilisés plus tard pour répliquer une analyse. Pour les réutiliser, il est préférable d'écrire des programmes facilement lisibles en ajoutant des commentaires et des titres.

Les commentaires peuvent être insérer de deux manières :
– Entre une astérisque et un point-virgule. : `* Commentaires ;`
– Encadrés par les commandes slash-astérisque et astérisque-slash : `/* Commentaires */`
A la différence des commentaires, les titres sont affichés dans l'OUTPUT. Lorsqu'on on exécute un grand nombre de commandes en même temps, ils permettent de mieux se repérer dans l'OUTPUT. Comme dans un traitement de texte, on peut définir plusieurs niveaux de titres. Ceux-ci restent valables jusqu'à ce qu'un nouveau titre du même niveau ou d'un niveau supérieur ait été déclarés. Les titres se placent n'importe où dans le document (par exemple, on peut les placer au sein d'une étape PROC).

```
TITLE1 "...";
TITLE2 "...";
...
TITLE6 "...";
```

Enfin notons que par défaut SAS ne fait pas de différence entre les majuscules et les minuscules.

1.2.1 Les options standards

Afin d'améliorer la présentation de l'OUTPUT et de la LOG, on spécifie en général dans les premières lignes d'un programme les options suivantes.

– linesize : détermine la largeur des lignes dans l'OUTPUT
– pagesize : détermine la taille de la page.
– pageno=1 : la numérotation des pages recommence à 1 à chaque fois qu'on efface l'OUTPUT.
– noovp : pour ne pas imprimer trois fois les erreurs
– errors=1 : si un type d'erreur se répète pour plusieurs observations, il n'est signalé que pour la 1ère observation

Ce qui donne :

```
OPTIONS linesize=78 noovp pageno=1 errors=1;
```

Il est également courant de nettoyer l'OUTPUT et la LOG de manière à rendre ces fenêtres plus lisibles. On place alors en tête de programme l'instruction suivante :

```
DM "CLEAR LOG ; CLEAR OUTPUT ; " ;
```

1.3 Ressources

Le site de support de SAS http://support.sas.com/. Il contient notamment une galerie très pratique de graphiques avec le code permettant de les générer :
http://support.sas.com/sassamples/graphgallery/
ainsi qu'une liste et une description des procédures disponibles dans SAS 9.1.3 :
http://support.sas.com/onlinedoc/913/getDoc/fr/allprods.hlp/a003135046.htm

Le site de UCLA contient un grand nombre de tutoriaux et de programmes pour les logiciels R, Stata, SPSS et SAS :
www.ats.ucla.edu/stat/sas.
C'est un site très bien fait et très pédagogique.

Le wikilivre SAS : un manuel libre et collaboratif sur SAS : http://en.wikibooks.org/wiki/SAS

Le polycopié d'Axelle Chauvet : Le manuel SAS de l'ENSAE http://www.ensae.fr/ParisTech/OMI1C5/Langage_SAS_Ensae.pdf

J Gardner's Introduction to SAS (10 pages en anglais) http://dataninja.wordpress.com/2005/12/23/introduction-to-sas/

Google Code Search : Le moteur de recherche Google Code Search permet de rechercher du code SAS en spécifiant dans la requête `lang:sas` ou en utilisant la fonction de recherche avancée.

2 MANIPULER LES DONNEES

2.1 Comment lire les données

2.1.1 Les librairies

Pour indiquer au logiciel dans quel répertoire se trouvent les données, on définit des *librairies* grâce à l'instruction LIBNAME. Une librairie ne fait rien d'autre que de donner un nom à un répertoire.

```
LIBNAME nom_librairie "chemin_de_la_librairie" ;
```

Par exemple, on peut créer la librairie *mqss* et la librairie *modelisation*

```
LIBNAME mqss "C:\Documents and Settings\Bureau\MQSS\" ; LIBNAME
modelisation "C:\Documents and Settings\Bureau\Modelisation\" ;
```

Cette instruction se place généralement en tête de programme.

2.2 Décrire les données

2.2.1 Le tableur

Il est conseillé de regarder les données brutes à l'aide du tableur. Pour cela, il faut naviguer dans la fenêtre EXPLORER (CTRL + D), sélectionner la librairie où sont sauvegardées les données et cliquer sur le fichier correspondant.

2.2.2 La PROC PRINT

On peut aussi afficher les données brutes dans l'OUTPUT grâce à la PROC PRINT.

```
PROC PRINT data=lib.table1;
RUN;
```

Attention lorsque les données sont volumineuses, il n'est pas avantageux de les afficher dans l'OUTPUT. On peut décider d'afficher certaines observations et / ou certaines variables.

```
PROC PRINT data=lib.table1 (FIRSTOBS = 30 OBS = 40)(KEEP = var1 var2 var3) NOOBS;
RUN;
```

L'option FIRSTOBS indique la première observation qui sera affichée dans l'OUTPUT. L'option OBS indique le numéro de la dernière observation à afficher. L'option KEEP sélectionne les variables à afficher (utile lorsque le nombre de variable est grand). L'option NOOBS permet de supprimer le numéro d'observation.

Les options FIRSTOBS, OBS et KEEP ne sont pas spécifiques à la PROC PRINT et peuvent être utilisées chaque fois que SAS lit une base de données.

2.2.3 La PROC CONTENTS

La PROC CONTENTS permet de décrire de manière synthétique une base de données. Elle indique la date de création, le nombre d'observations, le nombre de variable, la liste des variables avec pour chacune des variables leur type (caractère ou numérique) leur format et leur label.

```
PROC CONTENTS data=lib.table <SHORT>;
RUN ;
```

L'option SHORT permet de n'éditer que la liste des variables.

2.3 Créer de nouvelle tables : l'étape DATA

L'étape DATA est l'outil générique permettant de créer une nouvelle base de données. La syntaxe est la suivante :

```
DATA lib.table <Options>;
<Instructions>;
RUN ;
```

L'instruction `DATA lib.table ;` crée une table appelée `table` dans la librairie `lib` définie au préalable. Si la librairie n'est pas spécifiée, la table est créée dans la librairie par défaut WORK. Le contenu de cette librairie est effacé à la fin de chaque session SAS.

La table ainsi créée est vide. Selon les cas, on lui affecte un contenu en entrant soi même les données (partie 2.3.10), en reprenant les données d'une base existante (instruction
set) ou en générant aléatoirement des données.

2.3.1 L'instruction SET

L'instruction SET permet d'affecter à une nouvelle base de données les valeurs d'une ou de plusieurs bases de données existantes. La syntaxe est la suivante :

```
DATA lib.table2 <Options>;
SET lib.table1 <Options> ;
RUN ;
```

Les options KEEP et DROP permettent de n'utiliser que certaines variables de la table.

```
SET lib.table1 (KEEP = var1 var2 var3);
SET lib.table1 (DROP = var4 var5);
```

On peut aussi sélectionner un sous-ensemble d'observations à l'aide des options FIRSTOBS ET OBS ou de l'instruction WHERE.

```
SET lib.table1 (FIRSTOBS = 30 OBS = 40) ;
SET lib.table1 (WHERE = (var1=1) ) ;
```

2.3.2 Les principales instructions

Les instructions au sein d'une étape DATA permettent essentiellement de sélectionner des variables ou des observations et de créer de nouvelles variables.

Les instructions KEEP et DROP permettent de sélectionner des variables :

```
KEEP var1 var2 var3 ;
DROP var1 var2 var3 ;
```

Pour sélectionner des observations, on peut les instructions WHERE ou IF ... DELETE.

```
WHERE var1 < 1000 ;
IF var1 < 1000 THEN  DELETE ;
```

On peut bien évidemment spécifier plusieurs conditions :

```
WHERE var1 > . and var2 ^= 0 and car1 ^ = " " and car2 = "nsp" ;
```

2.3.3 Créer de nouvelles variables

Pour créer de nouvelles variables, on peut déclarer leur type à l'aide de l'instruction LENGTH.

```
LENGTH var1 var2 4.;
```

Cette étape n'est pas nécessaire car il existe des types définis par défaut mais elle peut permettre de gagner de la place lorsque l'on manipule d'importantes bases de données.

Ensuite, il suffit de donner le nom de la variable et de dire à quoi cette variable est égale. Par exemple, on peut définir la variable `var2` comme égale à la variable `var1` :

```
var2 = var1 ;
```

On peut aussi utiliser l'ensemble des fonctions possibles (racine carrée sqrt(), log ln(), ...). Pour cela, on peut se référer à la liste des opérateurs et des fonctions en langage SAS dans les sections 5.6 et 5.7. Voici quelques exemples :

```
var2 = log (var1) ;
var3 = var1 + var2 ;
var4 = sqrt(var1)
;
```

2.3.4 Renommer les données

```
RENAME var1 = sexe ;
RENAME var2 = dipl ;
```

2.3.5 Labels et formats

Grâce aux formats, on peut spécifier à SAS les propriétés d'affichage des valeurs d'une variable. Si la variable est monétaire, le format DOLLAR permet d'afficher le symbole devant les sommes, alors que le format EURO affichera notre petit euro-symbole. Pour les variables discrètes (exemple : réponses lors d'un questionnaire), on préférera visualiser du texte (réponses : : Oui / Non) plutôt que des valeurs numériques (0 / 1). La plupart du temps, on utilisera un format juste après la variable pour qu'il prenne effet. Certaines procédures peuvent être appliquées sur plusieurs variables. Pour différencier le format d'un nom de variable, les noms de format comprennent toujours un point (.). La syntaxe générique est donc :

```
VAR var1 format1. var2 format2. ;
```

Pour les variables numériques, le format `w.d` permet d'indiquer le nombre de caractères à imprimer (w) et le nombre de décimales après la virgule (d).

On pourra par exemple taper :

```
VAR salary 8.2 ;
```

Pour les variable numériques, le format `$w.` permet d'indiquer le nombre de caractères à imprimer.

Pour rendre les données plus lisibles, il est conseillés d'utiliser des labels pour décrire les variables et des formats pour décrire les modalités des variables qualitatives.

Pour définir un label, il suffit d'ajouter dans une étape DATA, l'instruction suivante :

```
LABEL
          var1    = "Salaire annuel en francs"
       var2    = "Diplome"
     var3    = "CSP"
               ;
```

Pour décrire les modalités d'une variable qualitative codée de manière numérique, il faut d'abord définir le *format*. Par exemple, on peut définir le format :

```
PROC FORMAT ; VALUE $sex "1" = "Homme"      "2" = "Femme" ; RUN ;
```

Dans une deuxième étape, on peut appliquer ce format à une ou plusieurs variables avec l'instruction :

```
FORMAT var1 $sex. ;
```

Cette instruction peut être insérer au sein d'une étape DATA ou au sein d'une PROC. On remarque que les noms des formats sont précédés par un signe $ et se termine par un point lorsqu'ils sont appliqués.

2.3.6 Les valeurs manquantes

Les variables numériques manquantes sont généralement codées .A, .B, ...Z. Ce sont en réalité les plus petites valeurs qui existent.

2.3.7 Trier les données : LA PROC SORT

```
PROC SORT data=lib.table <OUT = lib.table2> ; BY <DESCENDING> var1
; RUN;
```

Par défaut, les valeurs sont triés par ordre croissant. En spécifiant l'option DESCENDING, on peut demander un tri par ordre décroissant. La table triées peut être sauvegardée à l'aide l'option OUT.

2.3.8 Fusionner des bases de données

Si l'on a deux bases de données contenant les mêmes variables mais des observations différentes, il est très facile de les mettre bout à bout pour créer une nouvelle base de données.

```
DATA lib.table3;
    SET lib.table1 lib.table2;
RUN;
```

Si l'on a deux base de données contenant les mêmes observations mais des variables différentes, on peut les fusionner à l'aide d'un identifiant et de l'instruction MERGE.

```
DATA lib.table3 ;
MERGE lib.table1 lib.table2 ;
BY ident ;
RUN ;
```

ATTENTION : Avant d'utiliser la commande BY, il faut trier les données à l'aide d'une PROC SORT.

2.3.9 Séparer des bases de données

On peut aussi utiliser une étape DATA pour créer plusieurs bases de données à l'aide de l'instruction OUTPUT.

```
DATA  lib.table2 lib.table3 ;
SET lib.table1;
IF sexe = 1 THEN OUTPUT table2 ;
IF sexe = 2 THEN OUTPUT table3 ;
RUN ;
```

2.3.10 Importer des données

Depuis la version 8, les données SAS sont sauvegardées avec l'extension *.sas7bdat*. Dans les versions antérieures, les données sont sauvegardées avec l'extension *.sd2*. Lorsque les données ne sont pas sauvegardées dans l'un de ces formats, il faut les convertir au format SAS.

Importer un fichier EXCEL Les fichiers EXCEL sont enregistrés avec l'extension `.xls`. On peut les importer dans SAS à l'aide d'une PROC IMPORT.

```
PROC IMPORT DATAFILE="C:\...\Table.xls" OUT=lib.table REPLACE ; <
GETNAMES = yes;  > < SHEET = sheet_name; > RUN;
```

L'instruction DATAFILE indique à SAS l'emplacement des données. L'instruction OUT indique le nom de la table SAS créée par la PROC IMPORT. L'option REPLACE permet de remplacer la base de données SAS si elle existe. L'instruction GETNAMES spécifie que la première ligne contient le nom des variables. L'instruction SHEET spécifie l'onglet que l'on veut importer.

Importer des données au format CSV Le format CSV (`comma separated value`) est très courant. Il a l'avantage de pouvoir être lu par la plupart des logiciels. Il s'agit d'un simple fichier texte dont les colonnes sont séparées par des virgules ou des point-virgules. Il peut être enregistré avec différentes extensions (.csv,.dat,.txt, etc). On peut les importer dans SAS à l'aide d'une étape DATA ou d'une PROC IMPORT.

```
DATA lib.table; INFILE "c:\...\file.dat"    DELIMITER=';' ; INPUT
var1 $ var2 $ var3  var4; RUN;
```

L'instruction INFILE précise l'adresse du fichier dans lequel se trouve les données. L'instruction INPUT précise le nom de chacune des variables. Les variables caractères sont suivies d'un signe $.

La PROC IMPORT est plus facile d'usage.

```
PROC IMPORT datafile="E:\...\table.csv" OUT=table replace; DELIMITER
= ";" ; RUN ;
```

L'instruction DELIMITER permet de préciser la nature du délimiteur séparant les colonnes. En général c'est une virgule ou un point virgule.

Les logiciels de conversion des données Dans certains cas, il est plus facile d'utiliser des logiciels spécifiques à la conversion de données comme Stat Transfer ou DBMS Copy. Lorsque ces logiciels ne sont pas disponibles, il est toujours possibles d'utiliser un format du type CSV (Comma Separated Values) qui peut être lu par n'importe quel logiciel. Enfin notons qu'il est aussi possible de passer par le logiciel R qui peut lire les données en provenance de n'importe quel logiciel (SAS (`read.xport`), Stata (`read.dta`), SPSS (`read.spss`)) et exporter ces données vers le logiciel souhaité (`write.foreign`) grâce à la librairie `foreign`.

2.3.11 Entrer directement les données à la main

Lorsque les données sont petites, on peut les taper à la main dans le programme grâce aux instructions INPUT et CARDS.

```
DATA lib.table ; INPUT nom $ prenom $ salaire age csp $ prime ;
            CARDS ;
            A pierre 2000 30 ouvrier 100
            B marcel 5000 30 ouvrier 100
            C jean 8000 49 employe 200
            D jacques 10000 25 cadre 300
            ;
RUN;
```

L'instruction INPUT précise le nom et le type des variables. L'instruction CARDS permet de taper directement les données dans le fichier texte.

2.3.12 Exporter les données vers d'autres logiciels

Pour exporter des données, on peut utiliser la PROC EXPORT. On précise le nom de la base de données que l'on souhaite exporter avec l'instruction DATA et le nom du nouveau fichier grâce à l'instruction OUTFILE.

```
PROC EXPORT data=lib.table  OUTFIILE="C:\...\Table.xls" REPLACE ;
RUN;
```

3 STATISTIQUES DESCRIPTIVES

3.1 Statistiques univariées

3.1.1 Décrire une variable continue

Pour décrire la distribution d'une variable continue, on peut utiliser une PROC MEANS ou une PROC UNIVARIATE.

```
PROC MEANS data=lib.table <N MEAN MIN MAX STD> ; VAR  var1 var2
var3 ; <OUTPUT OUT = table_results MEAN = var1 var2 var3;> RUN;
```

L'instruction VAR spécifie les variables auxquelles s'applique la procédure. Par défaut, SAS applique la procédure à l'ensemble des variables numériques.

L'instruction OUTPUT permet d'exporter les résultats vers une table SAS. On spécifie le nom de cette table grâce à l'instruction OUT et les grandeurs que l'on veut exporter ensuite en précisant le nom des variables.

On peut obtenir plus de détails grâce à la PROC UNIVARIATE

```
PROC UNIVARIATE data=lib.table1  ; VAR var1 ; RUN ;
```

La PROC UNIVARIATE permet également de représenter graphiquement des estimations de la densité (histogrammes et kernel).

```
PROC UNIVARIATE data = lib.table ; HISTOGRAM var1 / KERNEL ; RUN ;
```

Enfin, pour réaliser un histogramme, on peut aussi utiliser la PROC GCHART.

```
PROC GCHART data = lib.table ; VBAR var1 ; RUN ; QUIT ;
```

3.1.2 Décrire une variable discrète

Pour décrire une variable catégorique, on peut regarder des tables de fréquences. La PROC FREQ permet d'afficher les fréquences, les pourcentages et les pourcentages cumulés pour chacune des modalités de la variables. Les pourcentages cumulés n'ont de sens que pour les variables catégoriques ordonnées.

```
PROC FREQ data=lib.table <ORDER = > ; TABLES var1 var2 var3
<OUT=>; RUN;
```

L'option ORDER permet de choisir l'ordre d'apparition des modalités dans le tableau. On peut notamment spécifier ORDER = FREQ pour que les modalités apparaissent par ordre de fréquence décroissant ou l'option ORDER = DATA pour que les modalités apparaissent selon l'ordre d'apparition des modalités dans la base de données.

L'option OUT = permet de définir les résultats de la PROC FREQ dans une nouvelle base de données.

Représentation graphiques La PROC GCHART permet de réaliser des représentations graphiques pour les variables catégoriques. Pour représenter la distribution d'une variable catégorique, on peut notamment utiliser des diagrammes en bâtons ou des camemberts grâce aux instructions HBAR, VBAR et PIE. Les diagrammes en bâtons (HBAR et VBAR) sont en général préférables aux camemberts car ils permettent une meilleure comparaison des quantités représentées.

Par exemple :

```
PROC GCHART data=lib.table ; VBAR var1 ; RUN ;
```

3.2 Statistiques bivariées

3.2.1 Deux variables continues

La PROC CORR permet de calculer la matrice de corrélation entre un ensemble de variables.

```
PROC CORR data=lib.table <SPEARMAN>; VAR var1 var2 var3 ...; RUN;
```

Par défaut, la PROC CORR calcule la corrélation de Bravais-Pearson. L'option SPEARMAN permet de calculer la corrélation de rang de Spearman.

Représentation graphique On peut représenter la relation entre deux variables continues à l'aide d'un nuage de points (scatterplot).

```
SYMBOL INTERPOL = scatter  COLOR = blue VALUE = square ;

PROC GPLOT data=lib.table1 ; PLOT y*x ; RUN ;
```

La première variable est représentée sur l'axe des ordonnées et la seconde sur l'axe des abscisses. L'instruction SYMBOL permet de définir l'aspect du graphique. Pour représenter le nuage de point, on spécifie l'option INTERPOL = scatter. On peut également spécifier la couleur (blue, red, black, etc) et la valeur des symboles représentés sur le graphiques (square, circle, etc).

3.2.2 Une variable continue et une variable catégorique

On peut réaliser un test de Student d'égalité des moyennes à l'aide de la PROC TTEST.

```
PROC TTEST data=table ; CLASS var1 ; VAR var2 ; RUN ;
```

L'instruction CLASS prend en argument une variable catégorique et l'instruction VAR une variable continue.

3.2.3 Deux variables discrètes

Pour décrire la relation entre deux variables catégoriques on utilise un *tableau de contingence* (ou tableau croisé). Ce tableau est obtenu à l'aide d'une PROC FREQ avec l'instruction TABLES

```
PROC FREQ data=lib.table ; TABLES var1*var2 / CHISQ EXPECTED
DEVIATION CELLCHI2 ; RUN;
```

L'option CHISQ permet de réaliser le test d'indépendance du χ^2. L'option EXPECTED permet d'afficher les effectifs théoriques de chaque cellule sous l'hypothèse nulle d'indépendance entre les deux variables. L'option CELLCHI2 permet d'afficher la contribution de chaque cellule à la statistique du χ^2. DEVIATION calcule l'écart entre la valeur observée et la valeur théorique.

4 PROCEDURES ECONOMETRIQUES

4.1 Le modèle linéaire

Le modèle linéaire peut-être estimé à l'aide d'une PROC REG.

```
PROC REG data=lib.table  OUTEST=table_parametres SIMPLE COVOUT;
MODEL y  = x1 x2 x3  / NOINT; OUTPUT  OUT = table_residus R =
resid ; OUTPUT  OUT = table_prediction P = predictions ; TEST x1 =
x2 = 0 ; RUN; QUIT;
```

La constante est inclue par défaut. L'option SIMPLE permet l'impression de statistiques descriptives du modèle. OUTEST permet de créer une table contenant les paramètres estimés. COVOUT permet d'afficher la matrice de variance-covariance L'instruction MODEL permet de spécifier le modèle à estimer. L'option NOINT de l'instruction MODEL permet de supprimer la constante (INTERCEPT). TEST Permet de tester la nullité jointe de plusieurs coefficients (test de Fisher).

L'instruction OUTPUT OUT permet de créer une nouvelle table ici `table_residus` ou `table_prediction` contenant les valeurs prédites de la régression (mot clef `P`) ou les résidus (mot clef `R`)

Remarque : la PROC REG est une procédure interactive. Par conséquent, il est nécessaire de terminer la procédure par l'instruction QUIT ; .

4.2 Le modèle dichotomique

Le modèle logistique peut être estimé à l'aide d'une PROC LOGISTIC.

```
PROC LOGISTIC data=lib.table DESCENDING ;CLASS x3 ;MODEL y = x1 x2
x3 ;RUN ;QUIT ;
```

L'option DESCENDING permet de tester la probabilité que y = 1 plutôt que 0. L'option CLASS permet d'inclure dans la régression des variables polytomiques sans avoir à les recoder en indicatrices au préalable.

4.3 Variables Instrumentales

Pour mettre en œuvre, la méthode des doubles moindres carrés, on utilise une PROC SYSLIN (Système Linéaire) avec l'option 2SLS.

```
PROC SYSLIN data=lib.table 2SLS ;ENDOGENOUS x ;INSTRUMENTS z;MODEL
y = x; RUN ; QUIT ;
```

L'option 2SLS (2 stage least square) indique à SAS de calculer l'estimateur des doubles moindres carrés. L'instruction ENDOGENOUS spécifie l'ensemble des variables explicatives suspectées d'endogénéité. L'instruction INSTRUMENTS spécifie l'ensemble des instruments à utiliser. Enfin l'instruction MODEL permet de spécifier l'équation que l'on veut estimer.

5 POUR ALLER PLUS LOIN

5.1 L'analyse factorielle

5.1.1 ACP : Analyse en composantes principales

```
PROC PRINCOMP data = lib.table  N=5  OUT = coord ;VAR var1 var2 var3 ;
RUN ;
```

5.1.2 AFC : Analyse factorielle des correspondances

```
PROC CORRESP data = lib.table DIMENS = 3 OUTC = coord MCA ;TABLES var1 var2 ;
RUN ;
```

5.2 Exporter les résultats : ODS

L'option ODS (OUTPUT DELIVERY SYSTEM) permet de contrôler et d'exporter les résultats. Pour la plupart des procédures statistiques, on peut obtenir la liste des bases de données exportables par ODS grâce à l'instruction `ODS TRACE`

```
ODS TRACE ON / LISTING ; Proc Univariate data = ... ; ... ;
Run ;
ODS
TRACE CLOSE ;
```

On peut exporter les résultats vers Excel :

```
ods tagsets.excelxp file='W:\cachan_model\myreport.xls'; title2
"Description des données" ;
proc contents data=work.australia ;
run ; ods _all_ close ;
```

On peut aussi exporter les résultats vers un document PDF :

```
ods pdf file = 'W:\cachan_model\report.pdf' ; proc gplot data =
australia ; plot lifesat * gdp2002 ; run ; ods pdf close ;
```

Pour en savoir plus, on peut se référer au document d'Yves Aragon : http://w3.univ-tlse1.fr/GREMAQ/Statistique/Yvesweb/docs/SAS/ods_mde.pdf

5.3 Exporter un graphique

Pour exporter un graphique, il faut spécifier le format d'export à l'aide de l'instructions GOPTIONS :

```
GOPTIONS RESET = ALL ; GOPTIONS DEVICE = JPEG GSFNAME = newgraph
GSFMODE = REPLACE ;
```

Ensuite, on attribue au nom de fichier un emplacement physique grâce à l'instruction FILENAME et on exécute une procédures générant un graphique.

```
FILENAME newgraph = "E:\Cachan_M\graphe.jpg"  ; PROC GPLOT DATA
=table ; PLOT y * x ; RUN ;
```

On peut aussi exporter aux formats PS (Post Script), GIF, PDF, PDFC (PDF en couleurs) ou EMF (Windows MetaFile).

Remarque : Dans l'éditeur de graphique, on peut aussi utiliser le menu contextuel (clic droit) pour exporter le graphique au format souhaité.

5.4 Les macros

On distingue deux types de macros : celles qui contiennent simplement une chaîne de caractère (string) et celles qui contiennent un code SAS à compiler.

```
%let city = new orleans ;
title "data for &city" ;
```

```
%macro plot;
proc plot;
plot income*age;
run;
%mend plot;
```

Ensuite on peut invoquer la macro en tapant simplement :

```
%plot ;
```

On peut aussi passer des arguments dans la macro

```
%macro plot(yvar= ,xvar= );
proc plot;
plot &yvar*&xvar;
run;
%mend plot;
```

Dans ce cas, on peut invoquer la macro de la manière suivante :

```
%plot(yvar=income,xvar=age)
```

Pour en savoir plus sur les macros :

5.5 Les erreurs fréquentes

Fermer le tableur SAS ne permet pas de modifier une table qui soit ouverte dans le tableur. Il faut toujours vérifier qu'une table est fermée avant de lancée le programme qui la crée à nouveau ou la modifie.

Point-virgules Il est fréquent d'oublier des point-virgules.

RUN Il est également fréquent d'oublier le RUN ;

Conserver les données d'origine SAS a la particularité de travailler directement sur les bases de données. Par conséquent, il faut faire attention à ne pas écraser les bases de données originales car il ne sera pas possible de les récupérer.

Eviter les répétitions D'une manière générale, il faut écrire du code de manière à éviter de se répéter. Chaque répétition d'un morceau de code est la source d'une erreur potentielle. Il faut essayer dans la mesure du possible de systématiser les opérations répétitives à l'aide de boucles.

Sauvegarder le programme Il faut penser à régulièrement sauvegarder le fichier *.sas*

Le programme s'efface à l'exécution Il arrive que le programme s'efface au moment où il est exécuté. Pour désactiver cette fonctionnalité, il faut aller dans le menu `TOOLS>OPTIONS>ENHANCED EDITOR` puis aller dans le sous menu, `GENERAL` et vérifier que la case `Effacer le texte à l'exécution` a bien été désactivée.

L'instruction BY L'instruction BY doit impérativement être précédée d'une étape de tri des données (PROC SORT).

5.6 Les opérateurs en langage SAS

Opérateur	Equivalence	Traduction
OPERATEURS ARITHMETIQUES		
+, -		addition, soustraction
/, *		division, multiplication
**		puissance
OPERATEURS DE COMPARAISON		
=	EQ	égal à
^=	NE	non égal à
>	GT	plus grand que
<	LT	plus petit que
>=	GE	plus grand ou égal à
<=	LE	plus petit ou égal à
<>		maximum
><		minimum
OPERATEURS LOGIQUES OU BOOLEENS		
&	AND	ET logique
| ou !	OR	OU logique
^	NOT	NON logique
AUTRES		
IN		appartenance à la liste des modalités qui suivent
|| ou !!		concaténation
MIN		retourne la plus petite valeur
MAX		retourne la plus grande valeur

5.7 Les fonctions

5.7.1 Les fonctions numériques

Fonction	Rôle
SUM	Somme des valeurs non manquantes.
MIN	Minimum des valeurs non manquantes.
MAX	Maximum des valeurs non manquantes.
N	Nombre de valeurs non manquantes.
NMISS	Nombre de valeurs manquantes.
MEAN	Moyenne arithmétique (=SUM/N).
VAR	Variance
STD	Ecart-type.

5.7.2 Les fonctions caractères

Fonction	Role
! !	Concatène (écrit à la suite) deux textes.
LENGTH(X)	Renvoie la position dans X du dernier caractère non blanc ; si X est vide, LENGTH(X) vaut 1.
SUBSTR(X,N,L)	Extrait un mot de longueur L à partir de la position N dans X.
INDEX(X,CHAINE)	Renvoie la position dans X de la 1ère apparition de la chaîne ou 0 si la chaîne n'est pas trouvée.
COMPRESS(X)	Compacte X en supprimant tous les blancs à l'intérieur de la chaîne de caractères
COMPBL(X)	Compacte X comme COMPRESS, mais en laissant un espace entre les mots.
LEFT(X)	Supprime les blancs à gauche dans X.
RIGHT(X)	Supprime les blancs à droite de X.
LOWCASE(X)	Convertit les lettres majuscules en minuscules.
UPCASE(X)	Convertit les lettres minuscules en majuscules.

Table des matières