

Introduction au logiciel SAS
SAS : Statistical Analysis System
Version 9 Windows

Notes de cours S. Dufour-Kippelen

1.	Présentation du logiciel	2
1.1.	Les composantes du logiciel.....	2
1.2.	L'interface	3
1.3.	Les noms des variables et des bases de données	3
1.4.	Les bibliothèques de tableaux de données : SAS Library	4
1.5.	Structure d'un programme SAS	5
2.	Gérer une base de données	6
2.1.	Importer une base de données dans SAS	6
2.2.	Lecture et enregistrement d'une base de données SAS	6
2.2.a.	Proc CONTENTS	6
2.2.b.	Proc PRINT	7
2.2.c.	Etape DATA/SET	8
2.3.	Sélectionner / définir variables et observations utiles	8
2.3.a.	Supprimer des variables (DROP, KEEP)	8
2.3.b.	Changement de nom des variables et libellés (RENAME, LABEL)	9
2.3.c.	Supprimer des observations (IF, IF...THEN DELETE)	9
2.4.	Gestion de plusieurs bases	10
2.4.a.	Concaténation (SET)	10
2.4.b.	Fusion (MERGE)	10
3.	Eléments de programmation.....	12
3.1.	Opérateurs et fonctions	12
3.2.	Les instructions conditionnelles IF...THEN...ELSE.....	13
4.	Statistiques descriptives	14
4.1.	Procédure FREQ	14
4.2.	Procédure MEANS	17
4.3.	Procédure UNIVARIATE	18
4.4.	Procédure CORR	22
4.5.	Procédure GPLOT	23
4.6.	Procédure GCHART.....	25
4.7.	Utilisation des sorties, sauvegarde.....	26
5.	Régression linéaire	27
6.	Bibliographie	29
7.	Annexes	30
7.1.	Base Vietnam.....	30
7.2.	Base Revenu-Consommation EU 1929-1970.....	30

Avertissements :

Dans ce document, la construction d'une base de données brute n'est pas abordée. On travaille sur une base déjà existante.

Les procédures sont présentées dans leur syntaxe la plus courante. Pour de plus amples détails (syntaxe détaillée, exemples, aide à la lecture des résultats), il faut se reporter à la documentation fournie dans l'aide du logiciel.

Version septembre 2015

1. Présentation du logiciel

1.1. Les composantes du logiciel

Le logiciel SAS est un logiciel généraliste de traitement et d'analyse de données, ce qui englobe de nombreuses fonctions.

On peut considérer d'une part le traitement de données brutes :

- Saisie de données ;
- Importations de données venant d'autres logiciels ;
- Tris de données selon un ou plusieurs critères ;
- Etc.

D'autre part, SAS permet l'analyse des données :

- Statistiques descriptives ;
- Graphiques des séries brutes ;
- Traitements économétriques ;
- Recherche opérationnelle ;
- Scoring ;
- Etc.

SAS permet d'étudier les bases de la conception d'un programme car le langage de programmation SAS est similaire aux autres langages ; permet de centraliser les tâches de plusieurs logiciels spécifiques (de gestion de bases de données, de statistique, d'économétrie).

SAS fonctionne à partir de deux éléments principaux : les tableaux de données¹ (contiennent les informations que l'on doit traiter) et les procédures (définissent, via la programmation, le détail des traitements à effectuer).

SAS est un logiciel modulaire *i.e.* formé d'autant de procédures qu'il y a de types de traitements ou d'analyses possibles. Les procédures sont regroupées selon leur objet dans des modules séparés.

- Le module de base : **SAS/base** contient toutes les procédures non spécifiques : lecture/création/manipulation des tableaux de données ;
- Le module graphique : **SAS/graph** permet la création de graphiques en deux ou trois dimensions ;
- Le module de statistiques : **SAS/Stat** ajoute au module de base des procédures d'analyses statistiques plus spécifiques (analyse de la variance, régressions logistiques, modèles de durée, méthodes de scoring, analyses multivariées, classifications automatiques, tests de proportions, de variances,...) ;
- Le module de traitement des séries temporelles : **SAS/Ets** permet d'appliquer diverses techniques de prévisions sur séries temporelles, de résoudre des systèmes d'équations linéaires et non linéaires... ;
- Le module de recherche opérationnelle : **SAS/Or** permet l'utilisation de techniques de gestion de projets et de modéliser des réseaux (chemins critiques, modèles sous contraintes...).

SAS ne s'achète pas : il se loue à l'année.

Un fichier contient les mots de passe et le numéro de licence de l'année : SETINIT ou SID.

¹ On désignera indifféremment les données à traiter par les termes *bases de données* et *tableaux de données*. Il s'agit de tableaux à deux entrées : le plus souvent, en lignes les individus, en colonne les variables.

1.2. L'interface

○ Le Menu

Fichier : gestion des fichiers, importation, exportation de fichiers d'autres formats

Affichage : choix de la fenêtre ou raccourcis claviers (F9 fournit la liste des raccourcis)

Outils/solutions : appel de procédures qui pourront être programmées

Aide : Le détail des modules et de leurs procédures est fourni dans l'*aide et documentation de SAS* accessible par la barre d'outils.

○ SAS utilise trois fenêtres :

La fenêtre **Editeur** (ou *program editor* dans la version en anglais) où l'on écrit les programmes et d'où on lance leur exécution (bouton Soumettre dans la barre de menu (Bonhomme qui court ; « RUN »)).

Les fichiers créés dans cette fenêtre sont des programmes SAS et portent l'extension « **.sas** ».

La fenêtre **Journal** (ou *log*) où le logiciel « écrit » tous les détails de l'exécution du programme que vous avez lancé par l'éditeur, et notamment tous les messages d'erreurs.

Les fichiers créés dans cette fenêtre portent l'extension « **.log** ».

La fenêtre **Sortie** (ou *output*) qui contient les résultats du programme lancé.

Les fichiers créés dans cette fenêtre portent l'extension « **.lst** ».



SAS n'efface le contenu des fenêtres que lorsque l'on ferme la session SAS.

Il faut régulièrement effacer soi-même le contenu des fenêtres Journal et Sortie, sous peine de ne pas s'y retrouver dans les commentaires ou les sorties.

1.3. Les noms des variables et des bases de données

Les noms peuvent contenir 32 caractères : lettres, chiffres (mais ne doivent pas commencer par un chiffre). Sas ne différencie pas les minuscules des majuscules.

Les fichiers contenant les bases de données SAS portent l'extension « **.sas7bdat** ».

1.4. Les bibliothèques de tableaux de données : SAS Library

Les bases de données SAS sont stockées dans des bibliothèques (SAS Library) qui correspondent physiquement à des répertoires sous Windows.

Par défaut, SAS travaille sur la bibliothèque temporaire WORK : à la fin de la session SAS les fichiers seront perdus (la session s'achève lorsque l'on quitte SAS).

Au début du programme SAS on précise donc le nom de la (ou des) bibliothèque(s) sur lesquelles on travaille. Ces bibliothèques sont concrètement les répertoires du disque dur sur lesquels on va lire et enregistrer les tableaux de données.

La commande `LIBNAME` définit une bibliothèque. Elle se construit de la façon suivante :

LIBNAME nom_logique_de_la_bibliothèque **'chemin sur le disque depuis la racine'** ;

Exemple :

La base Vietnam98.sas7bdat est stockée dans le répertoire : C:\CoursStataSas

On veut définir le répertoire `C:\CoursStataSas` comme lieu de stockage et d'enregistrement (*i.e.* comme bibliothèque) des bases de données dans le cadre d'un programme SAS : cette bibliothèque aura pour nom logique LIB (mais pourrait être appelée TOTO, VIETNAM, ...).

On écrit :

libname lib 'C:\CoursStataSas' ;

↑ ↑ ↗ ↑

Instruction Nom logique Emplacement sur le disque Fin de la commande

Lorsque l'on fait appel à un tableau de données (« .sas7bdat ») qui est dans la bibliothèque LIB ou que l'on veut l'y enregistrer, le nom logique de ce tableau dans le programme SAS est :

nom_logique_de_la_bibliothèque.racine_du_nom_du_tableau

Exemple :

La base **Vietnam98**.sas7bdat stockée est dans la bibliothèque **LIB**, c'est-à-dire dans **C:\CoursStataSas**.

Elle sera désignée dans le programme SAS par `lib.vietnam98`.

1.5. Structure d'un programme SAS

Un programme SAS est composé d'une (ou plusieurs) étape(s) DATA et d'étapes PROC.

Une étape est une suite d'instructions SAS.



Chaque instruction commence par un mot clé et termine par un point virgule.

Les espaces servent de séparateurs : **il doit y avoir un espace entre chaque mot.**

Un programme doit finir par « **RUN ;** » pour pouvoir être exécuté.

L'étape DATA crée un tableau de données ou modifie un tableau existant. C'est dans cette étape que l'on crée, importe, modifie, extrait ou fusionne les données.

Cette étape est descriptive : c'est l'utilisateur qui dicte au programme la façon de créer la base de données et les variables qui la composent.

Cette étape commence toujours par l'instruction DATA suivie du *nom logique* d'un tableau.

Les étapes PROC (procédures) permettent l'analyse des tableaux de données créés dans l'étape DATA. C'est avec ces procédures que l'on effectuera les traitements statistiques, économétriques...

Chaque procédure de SAS est un sous-programme : elle applique une méthode de calcul prédéfinie. L'utilisateur ne programme pas la façon de traiter les données mais indique au logiciel la méthode d'analyse à appliquer.

Chaque procédure peut être complétée par des *instructions* (statements) et des *options*.

Une option importante dans chaque procédure est l'option **data=nom_fichier** qui indique le nom de la base sur laquelle doivent être réalisés les calculs de la procédure.

Par défaut, ce sera *la dernière base de données ouverte* qui sera utilisée.

Des commentaires peuvent être insérés dans le programme. Le texte doit alors être précédé par **/*** et suivi de ***/** : ainsi **/*ce texte ne sera pas lu par le logiciel*/** et apparaîtra en vert dans la fenêtre Editeur.

On peut également encadrer les commentaires d'un astérisque et d'un point-virgule :

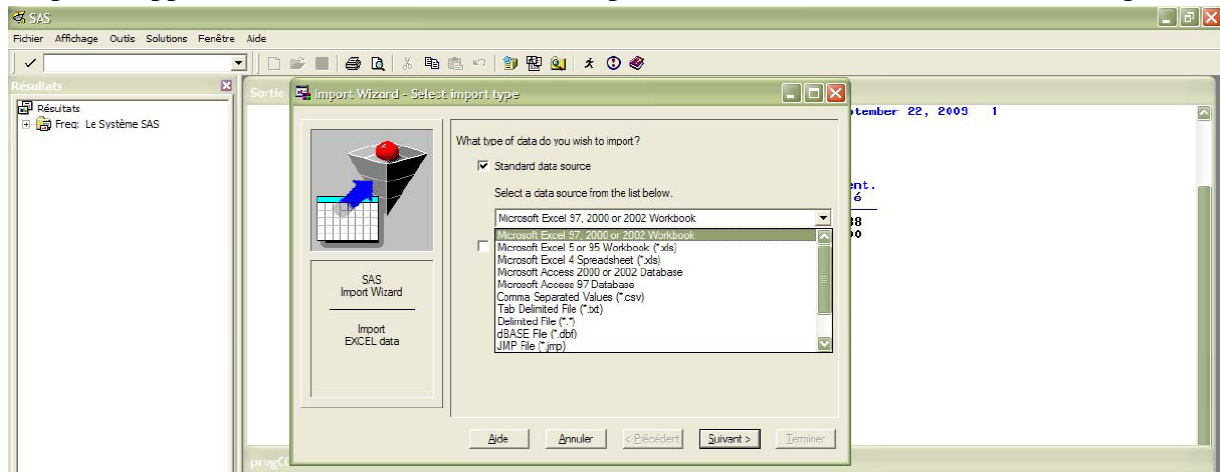
*** commentaire ;**

2. Gérer une base de données

2.1. Importer une base de données dans SAS

On peut directement importer une base (Excel, Stata, Access, DBase...) via SAS Import Wizard :

Etape 1 – appel de l'interface Menu Fichier/importer et choix du format du fichier à importer.



Etape 2- Workbook : adresse du répertoire où se trouve le fichier à importer

Etape 3- nom du fichier à importer

Etape 4 - choix de la bibliothèque où sera enregistrée la base SAS et le nom cette nouvelle base (member)

2.2. Lecture et enregistrement d'une base de données SAS

La base Vietnam98.sas7bdat est stockée dans le répertoire : C:\CoursStataSas

Fenêtre EDITEUR

```
libname lib 'C:\CoursStataSas';
```

2.2.a. Proc CONTENTS

Proc CONTENTS permet d'afficher le nom des variables contenues dans la base, le nombre d'observations, etc :

Fenêtre EDITEUR

```
proc contents data=lib.vietnam98;
title 'Présentation base VIETNAM';
run;
```

Fenêtre JOURNAL

```

7  libname lib 'C:\CoursStataSas';
NOTE: Libref LIB attribué comme suit :
      Moteur :          V9
      Nom physique : C:\CoursStataSas
8  proc contents data=lib.vietnam98; title 'Présentation base VIETNAM';
9  run;
    
```

NOTE: La procédure CONTENTS a utilisé (Durée totale du processus) :

temps réel	0.01 secondes
temps processeur	0.01 secondes

Fenêtre SORTIE

Présentation base VIETNAM

La procédure CONTENTS

Nom de la table SAS	LIB.VIETNAM98	Observations	5968
Type d'entrée	DATA	Variables	14
Moteur	V9	Index	0

Etc.

Liste alphabétique des variables et attributs

#	Variable	Type	Long.	Libellé
11	AGEHH	Num	8	Age of HH head
1	COUNTRY	Alph	3	Country name
10	EDUHH	Num	8	Education of hh head
4	HHSIZE	Num	8	Household size

etc

2.2.b. Proc PRINT

Proc PRINT affiche le contenu de la base de données :

PROC PRINT *options*; VAR *liste_variables* ; WHERE *instruction*;

Parmi les options : *data=nom de la base* : choix de la base de données à afficher. Par défaut, la dernière base ouverte est utilisée; *data=nom de la base (obs=n)*: affichage des n premières observations de la base signalée par l'option data.

L'instruction (*statement*) VAR permet de sélectionner les variables à afficher.

L'instruction WHERE permet de sélectionner les observations à afficher : *par défaut*, l'intégralité des observations sont affichées.

Exemple:

Fenêtre EDITEUR

```

proc print data=lib.vietnam98;
var idh agehh sexhh;
where sexhh=1;
run;

/*affiche les variables identifiant et la
tranche d'âge pour les ménages de la base dont
le chef est un homme - soit 4230 ménages*/
    
```

Fenêtre Sortie

Obs	IDH	AGEHH	SEXHH
3	105	3	1
8	111	3	1
9	112	3	1
10	113	3	1
11	114	4	1
13	121	3	1
15	123	3	1
16	201	3	1

Etc.

Remarque : les données se lisent en ligne.

Une ligne correspond à une observation (ici un ménage), une colonne à une variable.



Si une variable n'est pas renseignée pour un individu (**valeur manquante**, *missing value*) elle est codée par SAS :

- par un point « . » si c'est une variable numérique ;
- par un espace si c'est une variable alphanumérique.

2.2.c. Etape DATA/SET

Dans la pratique, on travaille rarement sur une base de données « source » ou complète (celle qui est issue d'une enquête, d'une collecte d'informations). Il faut souvent extraire certaines séries, en définir d'autres, corriger des données manquantes, etc.

Dans l'étape DATA on procèdera à différentes actions permettant de mettre en forme les données sur lesquelles seront effectuées les procédures.

Pour préserver les données « sources », on donne un nom différent à la base de données initiale et à la base sur laquelle on va effectuer les traitements. On le fait à l'aide de l'instruction SET.

<pre>libname lib 'C:\CoursStataSas'; data lib.tpVN; set lib.vietnam98; run;</pre> <p>/*Ce programme enregistre la base VIETNAM98.sas7bdat sous le nom tpVN.sas7bdat dans la bibliothèque LIB. C'est sur le fichier défini dans DATA (tpVN ici) que seront effectués les traitements.*/</p>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> Nom logique de la copie de Vietnam.sas7bdat </div> <div style="border: 1px solid black; padding: 5px;"> Nom logique de la base Vietnam.sas7bdat </div>
--	--

2.3. Sélectionner / définir variables et observations utiles

2.3.a. Supprimer des variables (DROP, KEEP)

Des variables peuvent être supprimées en utilisant les instructions DROP ou KEEP.

DROP *liste_variables* ; exclut les variables citées

KEEP *liste_variables* ; conserve les variables citées

Exemple :

<pre>libname lib 'C:\CoursStataSas'; data lib.tpVN; set lib.vietnam98; drop agehh hsize; run;</pre>	<p>Dans le fichier TPVN il y aura 12 variables (14-2) car agehh et sizehh sont retirées</p>
<pre>libname lib 'C:\CoursStataSas'; data lib.tpVN; set lib.vietnam98; keep agehh hsize; run;</pre>	<p>Le fichier TPVN sera constitué des 2 variables agehh et sizehh</p>

2.3.b. Changement de nom des variables et libellés (RENAME, LABEL)

Pour changer le nom d'une variable : `RENAME ancien_nom = nouveau_nom ;`
 Ou `RENAME = (ancien_nom = nouveau_nom) ;`

Pour donner un libellé pour expliciter le nom des variables :

`LABEL nom_variable='libellé' ;`

Exemple : pour définir les libellés des variables de la base, on a procédé de la façon suivante :

```
data lib.tpVN;
set lib.vietnam98;
label agehh='Age of HH head'
      country='country name'
      eduhh='Education of hh head'
      ...
      year='survey year';
run;
```

	Country name	Survey year	Household ID number	Household size	Household population weighted	Sampling weights	Urban/Rura dummy	Region Area	Gender of hh head	Education of hh head	Age of HH head	Economic class of hh head	househol income	household consumption
1	VNM	1998	101	6	10062	1677	1	2	2	1	4	2	379.12	2032.5390625
2	VNM	1998	103	6	10062	1677	1	2	2	.	3	2	1013.83	2253.083252
3	VNM	1998	107	6	10062	1677	1	2	2	.	4	2	1354.83	2295.6887207
4	VNM	1998	108	8	13416	1677	1	2	2	1	4	3	1393.58	2843.1064453
5	VNM	1998	109	7	11739	1677	1	2	2	.	4	3	918.363	2985.020752
6	VNM	1998	110	9	15093	1677	1	2	2	1	4	2	844.003	2203.6196289

2.3.c. Supprimer des observations (IF, IF...THEN DELETE)

Pour conserver uniquement les observations désignées par le filtre IF : `IF condition;`

Pour supprimer des observations de la base de données on utilise :

`IF condition THEN DELETE;`

Exemple : on veut construire une base dans laquelle ne figurent que les ménages dont les chefs de famille sont des femmes :

```
data lib.tpVN_F;
set lib.vietnam98;
if sexhh = 1 then delete;
run;
OU
data lib.tpVN_F;
set lib.vietnam98;
if sexhh=2;
run;
```

La base TPVN_F ne contient que les ménages dont le chef de ménage est une femme.

/*filtre qui ne garde que les ménages pour lesquels SEXHH=2 */

2.4. Gestion de plusieurs bases

2.4.a. Concaténation (SET)

La concaténation consiste à « empiler » plusieurs bases ayant les mêmes variables. « On ajoute seulement des observations ».

Instruction SET *liste_bases* ;

Exemple :

Soit TPVN_F : 14 variables, 1738 (lignes) ménages dont le chef de ménage est une femme

Soit TPVN_H : 14 variables, 4230 (lignes) ménages dont le chef de ménage est un homme

Si l'on réunit les 2 fichiers TPVN_H et TPVN_F, on reconstitue la base TPVN de 5968 ménages dans la base CONCATHF.

```
data lib.concatHF;
set lib.tpv_n_h lib.tpv_n_f;
run;
```



Les bases sont empilées dans l'ordre d'apparition dans la liste de l'instruction SET.



Si une variable n'est pas dans l'une des bases, des données manquantes apparaîtront dans la base « cumul »

Exemple:

```
data concat; set tab1 tab2; run;
```

tab1

ident	age
a	15
b	20
c	30

tab2

ident	age2008
e	35
f	40

Concat=

ident	age	age2008
a	15	.
b	20	.
c	30	.
e	.	35
f	.	40

2.4.b. Fusion (MERGE)

La fusion de plusieurs tableaux *ayant des individus en commun* consiste à ajouter des variables ou des individus.

Par exemple, souvent les données d'enquête sont composées de plusieurs fichiers correspondant à différentes parties du questionnaire d'enquête : un fichier pour les données démographiques, un autre pour les données de revenus...

Chaque individu interrogé aura un identifiant (numéro Insee du ménage pour des données individuelles, année de la série pour des données temporelles...) qui sera le même dans chacune des bases constituant l'enquête. Ainsi, dans chaque fichier, les informations concernant le même individu pourront être retrouvées.

La procédure MERGE permet de fusionner des bases ayant un identifiant commun.

Dans le programme qui suit, pour chaque valeur de l'identifiant (IDENT par exemple, pour chaque individu, *i.e.* une ligne de tableau) SAS réunit dans une même ligne d'une seule base les variables des 2 fichiers base1 base2.

```
Proc sort data=lib.base1 ; by IDENT ;
Proc sort data=lib.base2 ; by IDENT ;
```

Les bases doivent être triées selon les modalités de la variable de fusion IDENT

```
Data lib.fusion;
Merge lib.base1 (in=a) lib.base2(in=b);
by ident;
```

in=a: crée une indicatrice qui vaut 1 si la modalité de la variable IDENT figure dans base1.

```
if a and b;
run;
```

on retient les données des individus appartenant aux deux bases simultanément : **cylindrage**

Ainsi, par exemple :

Base1	In(a)	a	Base 2	In(b)	b	Base Fusion	If a and b	
IDENT	Statut		IDENT	Revenu		IDENT	Statut	Revenu
22	Marié	1	21	1000	1	22	Marié	1500
23	Célibataire	1	22	1500	1	24	Célibataire	2000
24	célibataire	1	24	2000	1			

Exemple 2 :

TPVN_REVCONSO : 3 variables (revenus et consommation et IDH (numéro d'identification du ménage)) et 5968 lignes : **Créé par :**

```
data lib.tpvN_revconso;set lib.vietnam98;keep idh totcons totinc; run;
```

TPVN_sansREVCONSO : 12 variables et 5968 lignes : **Créé par :**

```
data lib.tpvN_SANSrevconso; set lib.vietnam98; drop totcons totinc;
```

Dans un même fichier fusionrevconso on regroupe toutes les variables dont on dispose pour les ménages.


```
proc sort data=lib.tpvN_SANSrevconso; by idh;
proc sort data=lib.tpvN_revconso; by idh;


data lib.fusionrevconso;
merge lib.tpvN_SANSrevconso(in=a) lib.tpvN_revconso(in=b);
by idh;
if a and b;
run;
```

NOTE: 5968 observations copiées de la table LIB.TPVN_SANSREVCONSO.

NOTE: 5968 observations copiées de la table LIB.TPVN_REVCONSO.

NOTE: La table LIB.FUSIONREVCONSO a 5968 observations et 14 variables.

 En dépit du soin apporté à la programmation, il est recommandé de toujours vérifier dans la base de données si les résultats obtenus sont ceux escomptés. Il est rare que l'on parvienne du premier coup à un résultat propre !

 Si deux variables portent le même nom dans les deux bases, seule celle de la dernière base citée dans l'instruction MERGE sera reportée.

3. Éléments de programmation

Ci-après quelques éléments de programmation simple pour modifier des variables existantes ou créer de nouvelles variables ; pour sélectionner des observations selon des valeurs prises par certaines variables.

3.1. Opérateurs et fonctions

Opérateurs arithmétiques

	<i>signification</i>	<i>Exemple</i>
**	puissance	3**2 donne 9
*	multiplication	3*2 donne 6
/	division	3/2 donne 1.5
+	addition	3+2 donne 5
-	soustraction	3-2 donne 1

Quelques fonctions mathématiques :

<i>Fonction</i>	<i>signification</i>
ABS(x)	Valeur absolue de x
INT(x)	Partie entière de x
SQRT(x)	Racine carrée de x
EXP(x)	Exponentielle de x
LOG(x)	Logarithme népérien de x
Lag(x)	X en t-1
Dif(x)	$X_t - X_{t-1}$

Le résultat des opérateurs de comparaison et des opérateurs logiques est égal à 1 si la comparaison ou la proposition est vraie, à 0 sinon.

<i>Opérateurs de comparaison</i>	<i>Signification</i>	<i>Exemple</i>
= ou EQ	Egal à	3=2 donne 0
~= ou NE	Différent de	3 ne 2 donne 1
> ou GT	Supérieur à	3 > 2 donne 1
< ou LT	Inférieur à	3 < 2 donne 0
>= ou GE	Supérieur ou égal à	3 >= 2 donne 1
<= ou LE	Inférieur ou égal à	3 <= 2 donne 0

Les opérateurs peuvent être combinés via des opérateurs logiques.

Les opérateurs logiques permettent de filtrer des données et de construire des instructions conditionnelles, c'est-à-dire des instructions qui ne sont effectuées que sur les observations qui vérifient la condition donnée.

<i>Opérateurs logiques</i>	<i>signification</i>
& ou AND	et
 ou OR	ou
^ ou NOT	négation

La condition complexe **x=1 or x=2 or x=3** peut être écrite **x in(1, 2, 3)**.

3.2. Les instructions conditionnelles IF...THEN...ELSE

IF...THEN...ELSE permet d'effectuer des instructions sous certaines conditions.

La syntaxe est la suivante :

```
IF condition THEN instruction_à_effectuer_si_condition_vraie ;
                ELSE instruction_à_effectuer_si_condition_fausse ;
```

La commande ELSE est optionnelle : si on ne la met pas, aucune instruction spécifique ne sera effectuée si la condition est fausse. Attention, cela peut aussi générer des valeurs manquantes.

Pour utiliser plusieurs instructions après THEN ou ELSE, il faut utiliser la structure DO...END.

```
IF condition THEN DO ; instruction1 ; instruction2 ; END ;

                ELSE DO ; instruction3 ; Instruction4 ; END ;
```

Exemples

1) Part de la consommation dans le revenu du ménage :

```
partconso=totcons/totinc;
```

2) création d'une variable de revenus en tranches à partir d'une variable quantitative : REVT en 3 tranches.

```
If totinc <= 460 then REVT=1 ;
If 460<totinc <=820 then REVT=2 ;
If totinc>820 then REVT=3;
If totinc = '.' then REVT=4 ; /*modalité « données manquantes »*/
```

NB : Dans le cadre des données VIETNAM98 la modalité 4 n'est pas nécessaire car il n'y a pas de données non renseignées pour le revenu.

Ne pas oublier de tenir compte des valeurs manquantes lorsque l'on construit une nouvelle variable. Le plus souvent on définit une modalité « donnée manquante » (quand on travaille sur données individuelles).

3) Indicatrice signalant les ménages dirigés par un homme :

On dispose de sexHH= 1 si le ménage est dirigé par un homme et 2 s'il est dirigé par une femme.

On crée la variable dummy : chefHom=1 si le ménage est dirigé par un homme ; 0 sinon.

```
if sexHH=1 then chefHom=1; else chefHom=0;
OU
chefHom=(sexhh=1);

proc print; var sexhh chefhom;run;
```

Obs	SEXHH	chef Hom
1	2	0
2	2	0
3	1	1
4	2	0
...		
7	2	0
8	1	1

4. Statistiques descriptives

FREQ	Fréquences, tableaux de contingence, tableaux à deux entrées
MEANS	Statistiques descriptives de base (moyenne, écart-type, min, max...)
UNIVARIATE	Statistiques descriptives de base plus développée que MEANS
CORR	corrélations
CHART	Graphiques (histogrammes, diagrammes circulaires...)
PLOT, GPLOT	Graphiques à deux ou trois dimensions

4.1. Procédure FREQ

Permet d'effectuer des tris à plat ou croisés, et fournit les pourcentages en lignes, en colonnes, et permet de faire des tests de Khi-deux. Convient pour les variables qualitatives et les variables quantitatives à peu de modalités (le plus souvent en classes ou tranches).

PROC FREQ < options > ;
BY variables ;
TABLES variables ou var1*var2 < / options > ;
WEIGHT variable < / option > ;

L'instruction BY permet de traiter les données par sous-groupes définis par les modalités de la variable citée après BY. Si cette instruction est utilisée, il faut trier la base de données selon la variable citée dans BY avant d'appeler la procédure FREQ (proc sort).

L'instruction TABLES est suivie de la liste des variables pour lesquelles on veut un tri à plat (TABLES var1 ;) ou un tableau croisé (TABLES var1*var2).

Parmi les options de TABLES :

- 1- TABLES / CHISQ permet le calcul de la statistique du test d'indépendance du Khi-deux.

H0 : indépendance entre X et Y

Au seuil de 5%, on rejette H0 si proba critique < 0,05.

- 2- TABLES / EXPECTED fournit les effectifs attendus (théoriques) sous l'hypothèse d'indépendance
- 3- TABLES / DEVIATION fournit la différence entre effectifs observés et théoriques
- 4- TABLES / OUT permet de créer un tableau pour sauvegarder les résultats:

Exemple :

```
libname lib 'C:\CoursStataSas';
data lib.tpVN;
set lib.vietnam98;
proc freq; tables agehh / out=lib.tutu ;run; tableau avec effectifs et %
proc freq; tables agehh / out=lib.tutu2 OUTCUM;run; tableau avec effectifs, %, effectifs cumulés et % cumulés
```

Exemple : tri simple selon sous-groupes

```

libname lib 'C:\CoursStataSas';
data lib.tpVN;
set lib.vietnam98;
proc sort; by sexhh;
proc freq; by sexhh; tables agehh; run;
    
```

Gender of hh head=1 -Age of HH head					Gender of hh head=2 - Age of HH head				
AGEHH	Fréquence	Pourcentage	Fréquence cumulée	Pctage. cumulé	AGEHH	Fréquence	Pourcentage	Fréquence cumulée	Pctage. cumulé
1	79	1.87	79	1.87	1	27	1.55	27	1.55
2	1583	37.42	1662	39.29	2	456	26.24	483	27.79
3	1750	41.37	3412	80.66	3	787	45.28	1270	73.07
4	818	19.34	4230	100.00	4	468	26.93	1738	100.00

Exemple: tris croisés

```

proc freq; tables sexhh*agehh / chisq;
    
```

Table de SEXHH par AGEHH

SEXHH(var X)		AGEHH(var Y)				Total	Distribution marginale de X
Fréquence	Pourcentage	1	2	3	4		
Pctage en ligne							
Pctage en col.							
1		79	1583	1750	818	4230	Distribution marginale de X
		1.32	26.52	29.32	13.71	70.88	
		1.87	37.42	41.37	19.34		
		74.53	77.64	68.98	63.61		
2		27	456	787	468	1738	Distribution marginale de X
		0.45	7.64	13.19	7.84	29.12	
		1.55	26.24	45.28	26.93		
		25.47	22.36	31.02	36.39		
Total		106	2039	2537	1286	5968	Distribution marginale de Y
		1.78	34.17	42.51	21.55	100.00	

Distribution conditionnelle de Y sachant X

Distribution conditionnelle de X sachant Y

Exemple

```
proc freq ; tables sexhh*agehh / chisq;
```

Dans cet exemple, on cherche à établir s'il y a une liaison entre l'âge et le sexe du chef de ménage.

Statistiques pour la table de SEXHH par AGEHH			
Statistique	DDL	Valeur	Prob
Khi-2	3	83.1606	<.0001
Test du rapport de vraisemblance	3	84.1486	<.0001
Khi-2 de Mantel-Haenszel	1	77.0258	<.0001
Coefficient Phi		0.1180	
Coefficient de contingence		0.1172	
V de Cramer		0.1180	

Taille de l'échantillon = 5968

Commentaire : la distance du Khi-deux est élevée (83) on rejette H0 (au seuil 1%) (pba critique du test: inf à 0,1%) : il y a dépendance entre l'âge et le genre du chef de ménage.

A la lecture du tableau croisé: les femmes chefs de ménage sont sur représentées dans les classes d'âge élevées.

4.2. Procédure MEANS

La procédure MEANS calcule la moyenne arithmétique et l'écart-type d'une variable quantitative :

```
PROC MEANS options ;
VAR liste_var ;           /*instruction précisant le nom des variables pour lesquelles on
                             veut la moyenne*/
BY var ;                   /*instruction permettant de calculer les statistiques sur des
                             sous-groupes définis par les modalités de la variable citée*/
                             NB : l'instruction BY nécessite que la base de données soit triée selon la
                             variable de sélection : on utilisera la procédure SORT avant la procédure
                             Means.
WHERE condition logique ; /*instruction permettant de sélectionner les observations sur
                             lesquelles porteront les calculs*/
                             Seules les observations pour lesquelles la condition logique est vraie seront
                             prises en compte :
                             Si where sex=1 les calculs porteront sur les observations correspondant aux
                             hommes (sex=1).

WEIGHT varpoids ;
```

Parmi les options possibles :

Data=*Nom base* nom de la base de données utilisée
nmiss nombre de valeurs manquantes
std écart-type
min, max minimum et maximum

PROC MEANS < options > ;

BY variables ;
VAR variables ;
WEIGHT variable ;
CLASS variable ;

Exemples

```
proc means maxdec=2; var totinc; where totinc>0; run;
```

Variable d'analyse : TOTINC household income

N	Moyenne	Ecart-type	Minimum	Maximum
5929	1317.23	1911.78	0.63	44441.51

```
proc means ; var totinc; weight WGT;
```

La procédure MEANS

Analysis Variable : TOTINC household income				
Nb	Moyenne	Écart-type	Minimum	Maximum
5968	1186.48	103866.09	-81010.13	44441.51

Attention, on remarque que dans la sortie SAS il n'est pas fait mention de la pondération.

```
proc means ; var totinc; class sexhh; where totinc>0;
OU
Proc sort; by sexhh;
proc means ; var totinc; by sexhh; where totinc>0;
```

→ 2 moyennes: moyenne des revenus pour les ménages dirigés par une femme/par un homme

Variable d'analyse : TOTINC household income							
Gender of hh head	N Obs	N	Moyenne	Ecart-type	Minimum	Maximum	
1	4201	4201	1351.75	1882.84	0.63	35134.11	
2	1728	1728	1233.32	1978.40	2.50	44441.51	

4.3. Procédure UNIVARIATE

Calcule les indicateurs de statistique descriptive usuels d'une variable quantitative : mode, moyenne, écart-type, quantiles, coefficient d'asymétrie, d'aplatissement.... En outre, sont fournis des tests associés à la moyenne et à la médiane, des tests de normalité.

PROC UNIVARIATE < options > ;

BY variables ;

VAR variables ;

HISTOGRAM variables ;

WEIGHT variable ;

L'instruction BY permet de traiter les données par sous-groupes définis par les modalités de la variable citée après BY. Si cette instruction est utilisée, il faut trier la base de données selon la variable citée dans BY avant d'appeler la procédure UNIVARIATE.

L'instruction HISTOGRAM permet de représenter les données par un histogramme.

L'instruction de pondération WEIGHT doit être utilisée lorsque les données sont agrégées ou que les observations d'une enquête sont pondérées afin de rendre l'échantillon représentatif. Ainsi, chaque observation de la base (i.e. chaque ligne) sera comptée autant de fois que le signale la variable de pondération.

Pour comparer des distributions de variables quantitatives: option PLOT (boîtes à moustaches, Box plot)

Exemple 1 : distribution d'une variable

```
proc univariate; var totinc; /*weight wgt*/;
where totinc > 0; /*suppression des valeurs négatives*/
run;
```

Procédure UNIVARIATE Variable : TOTINC (household income)

Moments			
N	5929	Somme des poids	5929
Moyenne	1317.23277	Somme des observations	7809873.06
Ecart-type	1911.77646	Variance	3654889.24
Skewness	7.82588209	Kurtosis	108.247939
Somme des carrés non corrigée	3.19536E10	Somme des carrés corrigée	2.16662E10
Coeff Variation	145.135812	Std Error Mean	24.8282658

Mesures statistiques de base			
Tendance centrale		Variabilité	
Moyenne	1317.233	Ecart-type	1912
Médiane	819.347	Variance	3654889
Mode	600.000	Intervalle	44441
		Ecart interquartile	1095

Tests de tendance centrale : Mu0=0				
Test	Statistique		p-Value	
t de Student	t	53.05376	Pr > t 	<.0001
Signe	M	2964.5	Pr >= M 	<.0001
Rang signé	S	8789743	Pr >= S 	<.0001

Quantiles (Définition 5)	
Quantile	Valeur estimée
100Max 100%	44441.5117
99%	8271.8535
95%	3862.1177
90%	2731.0132
75% Q3	1528.9166
50% Médiane	819.3470
25% Q1	433.6075
10%	237.4420
5%	159.3272
1%	54.1667

Quantiles (Définition 5)	
Quantile	Valeur estimée
0% Min	0.6250

Observations extrêmes			
Plus bas		Plus haut	
Valeur	Obs.	Valeur	Obs.
0.62500	5023	25674.6	1355
1.12500	5203	31013.3	294
2.50000	5226	31751.4	1356
2.50000	5174	35134.1	603
3.57407	2870	44441.5	1399

Exemple 1bis : distribution d'une variable avec PONDERATION

```
proc univariate; var totinc; weight WGT; where totinc >0;
```

	5929	Somme des poids	15945532
Moyenne	1209.98179	Somme des observations	1.92938E10
Ecart-type	88738.183	Variance	7874465115

Mesures statistiques pondérées de base

Tendance centrale		Variabilité	
Moyenne	1209.982	Ecart-type	88738
Médiane	775.522	Variance	7874465115
Mode	600.000	Intervalle	44441
		Ecart interquartile	965.57153

Exemple 2: Box plots - comparaisons des distributions de la taille des ménages selon le sexe du chef de ménage

```
proc sort; by sexhh;
```

```
proc univariate plot; var hhsiz; by sexhh;run;
```

Procédure UNIVARIATE - Variable : HHSIZE (Household size) - Gender of hh head=1

Moments			
N	4230	Somme des poids	4230
Moyenne	4.94893617	Somme des observations	20934
Ecart-type	1.85365311	Variance	3.43602984

Mesures statistiques de base

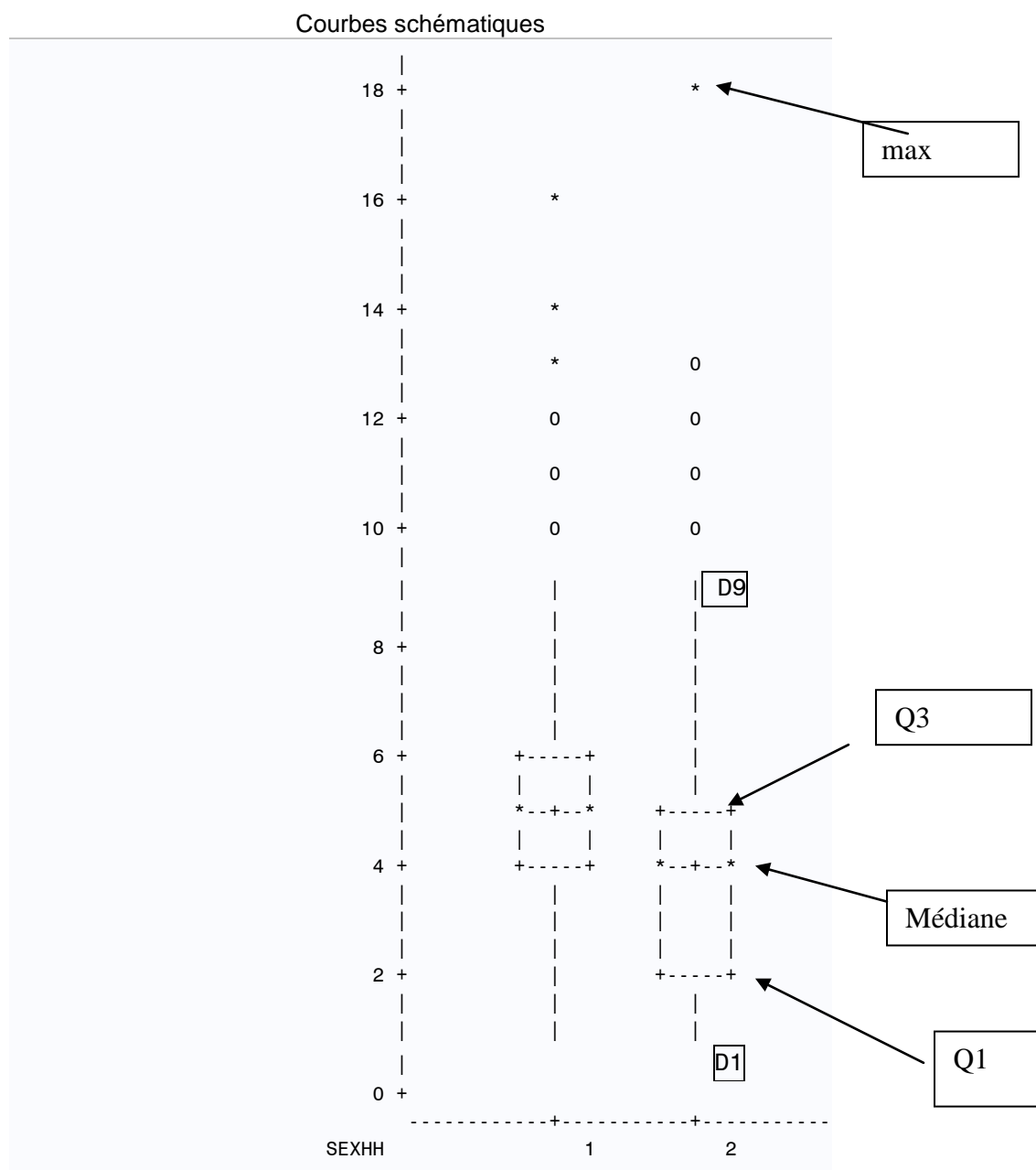
Tendance centrale		Variabilité	
Moyenne	4.948936	Ecart-type	1.85365
Médiane	5.000000	Variance	3.43603
Mode	5.000000	Intervalle	15.00000
		Ecart interquartile	2.00000

Procédure UNIVARIATE - Variable : HHSIZE (Household size) - Gender of hh head=2

Moments			
N	1738	Somme des poids	1738
Moyenne	3.83601841	Somme des observations	6667
Ecart-type	1.9321039	Variance	3.73302547

Tendance centrale		Variabilité	
Moyenne	3.836018	Ecart-type	1.93210
Médiane	4.000000	Variance	3.73303

Variable : HHSIZE (Household size)



Exercice: Retrouver ces graphiques à l'aide de la procédure Box Plot.

```
proc boxplot; plot hhsizesexhh; run;
```

4.4. Procédure CORR

Calcule des coefficients de corrélation.

PROC CORR < options > ;

BY variables ;

PARTIAL variables ; /* pour définir des corrélations partielles*/

VAR variables ; /*liste des variables dont on veut calculer le coeff de corrélation*/

WEIGHT variable ;

WITH variables ; /*liste des variables dont on veut calculer le coeff de corrélation avec les variables citées en VAR*/

Pour calculer le coeff de corrélation entre X et Y on peut écrire :

proc corr ; **var** X Y; OU **proc corr** ; **var** X ; **with** Y;

Exemple

proc sort; **by** sexhh;

proc corr ; **var** totinc ; **with** totcons; **weight** WGT; **by** sexhh;

```
----- Gender of hh head=1 ----- La procédure CORR
1 Avec Variables : TOTCONS
1 Variables : TOTINC
Pondération Variable : WGT
Statistiques simples
Variable Nb Moyenne Écart-type Somme Minimum Maximum
TOTCONS 4230 1090 44773 1.2481E10 100.98138 20753
TOTINC 4230 1209 107736 1.38412E10 -81010 35134
Statistiques simples
Variable Libellé
TOTCONS household consumption
TOTINC household income
Coefficients de corrélation de Pearson, N = 4230
Prob > |r| under H0: Rho=0
TOTINC
TOTCONS 0.40532
household consumption <.0001

---- Gender of hh head=2 ---- La procédure CORR
1 Avec Variables : TOTCONS
1 Variables : TOTINC
Pondération Variable : WGT
Statistiques simples
Variable Nb Moyenne Écart-type Somme Minimum Maximum
TOTCONS 1738 1078 55577 4956201469 63.24846 15312
TOTINC 1738 1132 93752 5200839608 -1601 44442
Coefficients de corrélation de Pearson, N = 1738
Prob > |r| under H0: Rho=0
TOTINC
TOTCONS 0.64282
household consumption <.0001
```

Le coefficient de corrélation entre le revenu et la consommation est de 0,4 lorsque le chef de ménage est un homme et 0,64 lorsque c'est une femme.

Pour calculer plusieurs coefficients : **proc corr** ; **var** X1 X2 ... Xm ; **with** Y1 Y2 ...Yn;
donne les coefficients suivants: r(Xi, Yj)

Par exemple : **proc corr** ; **var** totinc totcons; **with** hhsize; **weight** WGT;
donne r(totinc, hhsize) et r(totcons, hhsize)

4.5. Procédure GPLOT

Permet de faire des graphiques représentant des variables quantitatives : nuages de points, courbes, avec des présentations qui peuvent être choisies à l'aide d'instructions.

PROC GPLOT < options > ;
PLOT y*x < / options > ;

On peut choisir :

*La méthode d'interpolation entre deux points :
 interpol=none : pas de liaison entre les points (nuage de points)
 interpol=join : les points sont reliés par une droite, etc.

*la couleur : color=blue, black..

*la forme des points : value=dot, plus, circle...

Ces paramétrages sont à effectuer par l'instruction SYMBOLi (i=1 pour la première courbe, 2 pour la deuxième ...) avant d'appeler la procédure GPLOT.

Options de l'instruction PLOT :

HAXIS : échelle des X

VAXIS : échelle des Y

Overlay : superposition de courbes

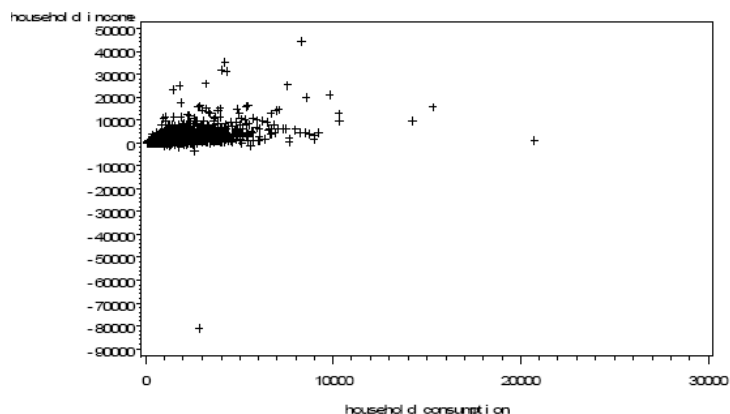
Legend

Exemple: nuage de points

```
data lib.tpVN;
set lib.vietnam98;
proc gplot;
plot totinc*totcons;
run;
```

le graphique sera plus lisible si l'on supprime les ménages dont les revenus sont négatifs :

```
proc gplot;
plot totinc*totcons;
where totinc >0; run;
```



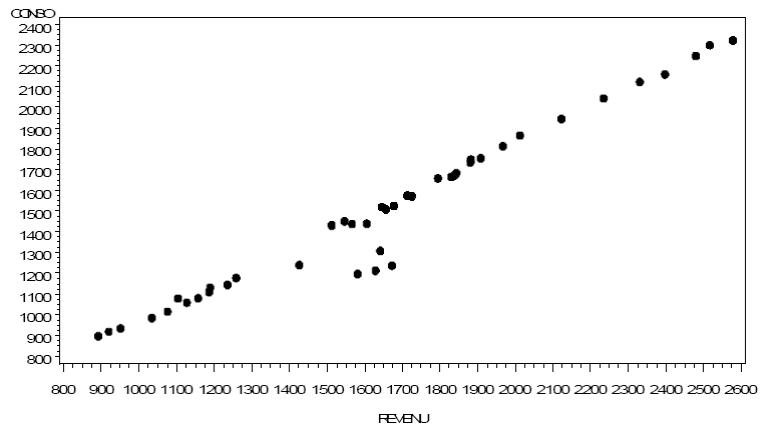
Exemple sur données temporelles consommations et revenus aux EU entre 1929 et 1970.

`/*3 variables : conso, revenu, annee*/`

Courbe consommation-revenu
(les points sont des années).

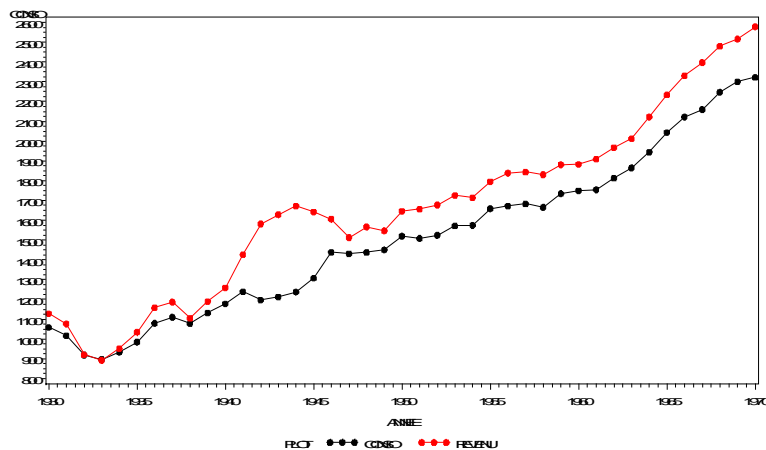
```
data lib.consol ;
set lib.revconso;
symbol1 v=dot;
proc gplot;
plot conso*revenu;
```

on observe un décrochage de la
conso par rapport au revenu
Durant 4 années (1942 à 1945)



Superposition de deux courbes représentées sur le même axe

```
symbol1 i=join v=dot;
symbol2 i=join v=dot;
proc gplot;
plot conso*annee revenu*annee / haxis= 1930 to 1970 by 5 overlay legend;
run;
```



Il existe la procédure Proc PLOT qui fournit des graphiques dans la fenêtre de résultats qui sont en format « texte ».

4.6. Procédure GCHART

La procédure GCHART produit les graphiques associés aux variables qualitatives ou quantitatives en classes: tuyaux d'orgues, diagrammes en bâtons, histogrammes, diagrammes en secteurs...

PROC GCHART;

BLOCK chart-variable(s) </ option(s)>;

HBAR | **HBAR3D** | **VBAR** | **VBAR3D** chart-variable(s) </ option(s)>;

PIE | **PIE3D** | **DONUT** chart-variable(s) </ option(s)>;

STAR chart-variable(s) </ option(s)>;

BLOCK: diagrammes en bloc (effet 3D)

HBAR: diagrammes en bâtons horizontaux

VBAR: diagrammes en bâtons verticaux (pas de stat)

PIE : camemberts

STAR : diagrammes en étoile

Les options sont :

DISCRETE si la variable à représenter est discrète

TYPE= précise ce que représente le graphique : **FREQ** (fréquences), **CFREQ** (fréquences cumulées), **PCT** (fréquences en % du total), etc.

GROUP=*variable* permet d'effectuer des représentations graphiques séparées selon les modalités de la variable indiquée (sur le même graphique)

SUBGROUP=*variable* permet de donner la répartition d'un bâton selon les modalités de la variable indiquée (tuyaux d'orgue)

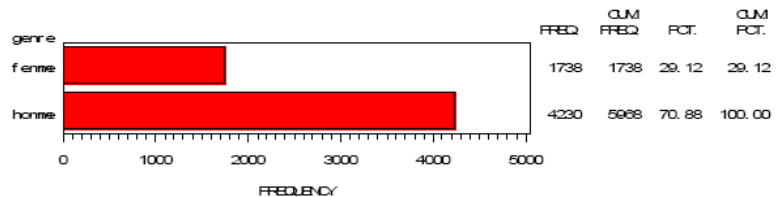
AXIS=*min max* : valeur max et min à prendre à compte

LEVEL: fixe le nombre de classes (de largeur égale) à représenter pour une variable quantitative

RANGE : affiche les intervalles des classes au lieu des centres de classes (mid-points)

```
data lib.tpVN;
set lib.vietnam98;
if sexhh=1 then genre='homme';
if sexhh=2 then genre='femme';

proc gchart; hbar genre;
run;
```



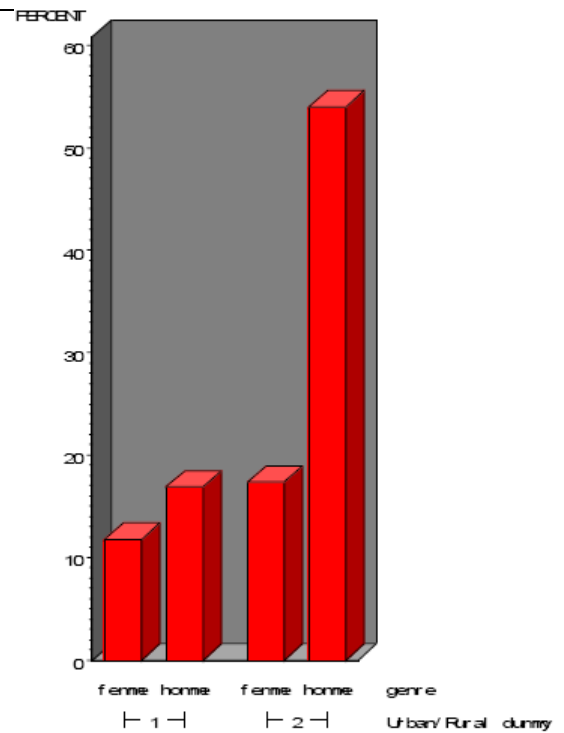
pour supprimer les statistiques (`hbar genre / nostat`)

Diagramme en bâtons 3 D:

```
proc gchart;
Vbar3D genre / type= pct group=urban;
run;
```

OU

```
proc gchart;
Vbar3D genre / type= pct subgroup=urban;
run;
```

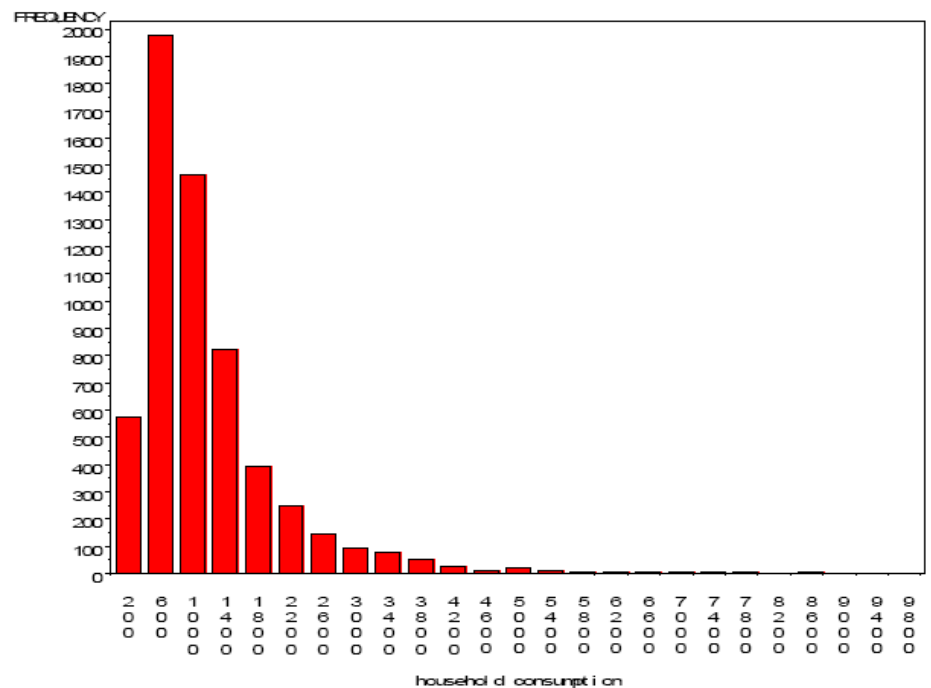


Histogramme de classes d'amplitudes égales Pour la variable de consommation

```
proc gchart;
vbar totcons ;
where totcons < 10000;
run;
```

Pour afficher les classes:

```
vbar totcons / range;
```



4.7. Utilisation des sorties, sauvegarde

Graphiques : copier/coller, passer via JPEG ou PDF et mettre un titre dans le traitement de texte.

Des instructions OUT/OUTPUT permettent de conserver des statistiques automatiquement et de les réutiliser.

5. Régression linéaire

La procédure REG permet d'estimer un modèle linéaire par rapport aux paramètres a_k par la méthode des moindres carrés ordinaires (MCO, OLS Ordinary Least Squares):

$$Y_t = a_0 + a_1 X_{1t} \dots + a_{Kt} X_{Kt} + u_t \text{ avec } t=1 \dots T$$

$$Y_i = a_0 + a_1 X_{1i} \dots + a_{Ki} X_{Ki} + u_i \text{ avec } i=1 \dots N$$

La variable à expliquer (*dependent, response variable*) est quantitative ; les variables explicatives (*regressor*) X_k peuvent être quantitatives et qualitatives.

Toutefois, la procédure exige des variables définies numériquement dans SAS. Ainsi, les variables qualitatives doivent être préalablement recodées à l'aide de variables binaires.

Par exemple : on veut estimer l'impact de la CSP du chef de ménage sur le montant des dépenses de consommation.

Soit X_k la variable signalant la CSP d'un individu :

$$X_{ki} = \begin{cases} 1 & \text{si } i \text{ est cadre supérieur} \\ 2 & \text{si } i \text{ est cadre intermédiaire} \\ 3 & \text{si } i \text{ est employé} \\ 4 & \text{si } i \text{ est ouvrier} \end{cases}$$

Elle est codée numériquement mais ses valeurs n'induisent pas de classement entre les individus. Il ne faudrait pas utiliser cette variable telle qu'elle : SAS la considérerait comme numérique en supposant que la valeur de X_k est quatre fois plus importante pour un individu cadre par rapport à un ouvrier, sans tenir compte de la signification des modalités.

En outre, il n'y aurait aucune modalité de référence permettant d'interpréter le résultat.

Ainsi, X_{ki} doit être recodée en 4 indicatrices (étape DATA) :

Cad sup_i = 1 si i est cadre supérieur ; Cad sup_i = 0 sinon

Cad int_i = 1 si i est cadre intermédiaire ; Cad int_i = 0 sinon

Employ ei = 1 si i est employé ; employ ei = 0 sinon

ouvrier i = 1 si i est ouvrier ; ouvrier i = 0 sinon

Dans la programmation, on introduira 3 de ces indicatrices, la quatrième étant la modalité retenue comme référence (voir cours d'économétrie linéaire sur les indicatrices).

PROC REG < options > ;
 MODEL *dependents = regressors* < / options > ;
 BY *variables* ;
 WEIGHT *variable* ;
 OUTPUT OUT=*fichier* predicted=*nom* residual=*nom* ;

Options disponibles dans l'instruction MODEL :

NOINT supprime la constante du modèle (elle est introduite par défaut)

SELECTION=*name* : permet de choisir la méthode de sélection des variables explicatives : forward (F), backward (B), Stepwise, None (on utilise toutes les explicatives citées dans l'instruction MODEL). Par défaut selection=NONE.

DW calcule la statistique de Durbin et Watson

P affiche les prévisions pour la variable expliquée

R : affiche les résidus de l'estimation

L'instruction OUTPUT permet de conserver les valeurs prédites, les résidus de l'estimation, ... dans le fichier défini en OUT.

Les sorties :

1) analyse de la variance :

décomposition de la somme des carrés. « model » se réfère à la somme des carrés expliqués ;

« error » à la somme des carrés des résidus

F Value : statistique de Fischer du test de significativité globale ($H_0 : a_1 = \dots = a_K = 0$)

Prob>F la proba critique associée : H_0 refusée si $p < \text{seuil du test } \alpha$

Root MSE : erreur quadratique moyenne

Dep Mean : moyenne arithmétique de la variable expliquée Y

CV : coefficient de variation de Y (en %)

R-square : coefficient de détermination R^2

Adjusted R-sq : R^2 ajusté

2) Estimation des paramètres

Exemple :

```
libname lib 'C:\CoursStataSas';

data lib.tpVN;
set lib.vietnam98;

/****regression Y=totcons X=totinc**/
proc reg;
model totcons=totinc;
output p=consohat r=residus;

proc print; var consohat residus;
run;
```

The REG Procedure
 Model: MODEL1
 Dependent Variable: TOTCONS household consumption

Number of Observations Read 5968
 Number of Observations Used 5968

Analyse de variance

Source	DF	Somme des carrés	Carré moyen	Valeur F	Pr > F
Model	1	1386923758	1386923758	1694.88	<.0001
Error	5966	4881982535	818301		
Corrected Total	5967	6268906294			

Root MSE	904.59980	R-Square	0.2212
Dependent Mean	1172.95312	Adj R-Sq	0.2211
Coeff Var	77.12156		

Résultats estimés des paramètres						
Variable	Libellé	DF	Résultat estimé des paramètres	Erreur std	Valeur du test t	Pr > t
Intercept	Intercept	1	888.23817	13.59936	65.31	<.0001
TOTINC	household income	1	0.22032	0.00535	41.17	<.0001

6. Bibliographie

Documentation SAS (une collection papier complète est disponible en bibliothèque de recherche, la version électronique est disponible dans le logiciel).

Duguet E. (2004) Introduction à SAS, Economica

Der G., Everitt B. (2008) A handbook of statistical analyses using SAS, 3ème édition, Chapman & Hall/CRC

7. Annexes

7.1. Base Vietnam

WGT : échantillon de 5968 ménages représentatifs de 16 millions de ménages

Idh	Numéro d'identification du ménage	<p>Régions du Vietnam</p> 
Wgt	Poids du ménage dans l'échantillon	
Hhsize	Taille du ménage (rang : de 1 à 18)	
Popw	Pondération (tient compte du poids du ménage dans l'échantillon et de la taille du ménage)	
Urban	Code zone urbaine/rurale (1- urbaine / 2- rurale)	
Region	Code région	
	1 Nord-Est (Northern Uplands)	
	2 Delta du fleuve Rouge (Red River Delta)	
	3 Centre-Nord (North Central)	
	4 Région littorale du Centre (Central Coast)	
	5 Hauts-plateaux du Centre (Central Highlands)	
	6 Région du Nam Bo oriental (Southeast)	
	7 Delta du Mékong (Mekong River Delta)	
Sexhh	Sexe du chef de famille (1-homme / 2-femme)	
Ocuhh	Occupation du chef de famille	
	1. Travailleur indépendant agricole	
	2. Travailleur indépendant non agricole	
	3. Travailleur salarié	
Agehh	Age du chef de famille (années)	
	1. Jusqu'à 25	
	2.]25, 40]	
	3.]40, 60]	
	4. plus de 60	
Eduhh	Niveau d'éducation du chef de famille	
	1. Aucun ('no education')	
	2. Primaire ('primary')	
	3. Secondaire ('secondary')	
	4. Secondaire technique et professionnel ('vocational')	
	5. Universitaire ('higher')	
Totcons	Consommation total du ménage (en milliers de dongs)	
Totinc	Revenu total du ménage (en milliers de dongs)	

7.2. Base Revenu-Consommation EU 1929-1970

Données américaines sur Revenu disponible et consommation par habitant (en \$ 1958) de 1929 à 1970 (source : *Econometrics*, Maddala).