PN-ABF-201

66334

Rapport ASPAP/DAI N° 100

Manuel de Référence SPSS-PC Version 3.0

Projet des Enquêtes Agricoles et Analyse
des Politiques Economiques du Secteur Rural

(ASPAP: Agricultural Surveys and Policy Analysis Project)

Contract USAID n° 696-0126

Dr. Randall D. Schnepf**
Consultant de Courte Durée DAI***

Kigali, Rwanda
décembre, 1989

** Research Associate, Division of Nutritional Sciences, Cornell University, Ithaca, New York, 14853, USA.

*** Development Alternatives, Incorporated (DAI), 624 Ninth Street, N.W. Washington, D.C. 20001, principal contractant institutionnel pour l'ASPAP, USAID Contrat N° 696-0126-C-00-7777-00.

SPSS/PC+ V3.0 -- Références et Commandes d'Utilisation

I. Introduction au Programme SPSS

A.	Information Générale	3
B.	Fichiers SPSS	4
C.	La Façon de Traiter les Données absentes en SPSS	4
D.	Operateurs et Fonctions du Programme	6
E.	Les Règles d'Opération	8

II. SPSS Traitement de Texte: REVIEW

A.	Utilisation des clés 'F'	9
B.	Utilisation des clés de mouvement	10

III. Les Commandes pour la Manipulation des Fichiers

A.	GET FILE	11
B.	SAVE OUTFILE	11
C.	N	11
D.	SAMPLE	12
E.	LIST	12
F.	MODIFY VARS	13
G.	JOIN MATCH	14
H.	JOIN ADD	16
I.	SORT	18
J.	SYSFILE INFO	19
K.	FINISH	19

IV. Les Commandes pour la Manipulation et Creation des Variables

A.	MISSING VALUE	20
B.	COMPUTE	20
C.	COUNT	22
D.	RECODE	22
E.	IF	23
F.	SELECT IF	24
G.	PROCESS IF	25
H.	VARIABLE LABEL	26
I.	VALUE LABEL	26
J.	FORMAT	27
K.	DISPLAY	27
L.	AGGREGATE	28

V. Les Commandes pour Calculer les Statistiques

A.	DESCRIPTIVES	30
B.	FREQUENCIES	31
C.	MEANS	34
D.	T-TEST	35
E.	CROSSTABS	36
F.	ANOVA	37
G.	ONEWAY	39
H.	NPAR	41
I.	REGRESSION	42

A. Information Générale

Le programme SPSS comprend plusieurs aspects du traitement des données.

D'abord, il y a le programme DATA ENTRY pour faire la saisie des données (y compris la préparation de la forme de saisie, la définition des variables, la capacité d'importer les fichiers ASCII afin de les transformer en fichier interne de SPSS (*.SYS) et d'exporter un fichier en forme ASCII à partir d'un fichier interne de SPSS (*.SYS).

Deuxièmement, le SPSS comprend les commandes pour la manipulation des fichiers. Ces commandes comprennent l'habilité de combiner, diviser, et modifier un fichier actif, d'appeler un fichier à la mémoire temporaire (ce qui devient le fichier actif) sauver le fichier actif en fichier permanent, lister les cas du fichier actif, et appeler sur l'écran des informations concernant un fichier et ses variables.

Troisièmement, le SPSS inclut les commandes pour la manipulation et la création des variables à partir du fichier actif. Ces commandes comprennent l'habilité de limiter les cas du fichier actif qui sont utilisés par un tel programme, de créer des nouvelles variables ou de modifier des anciennes variables, de changer les valeurs des cas, de créer ou changer les étiquettes des variables ou des valeurs, d'ajuster le format des variables, de rassembler les cas existant dans une forme agrégée par catégorie discrète d'une variable de référence et d'obtenir des informations sur les variables d'un fichier.

Quatrièmement, le SPSS comprend les commandes pour entreprendre une analyse statistique des variables du fichier actif. Ces statistiques incluent les statistiques descriptives, les fréquences, les moyennes par catégories discrètes, un t-test pour comparer deux moyennes, une ANOVA et ses statistiques associées pour comparer les moyennes par catégories discrètes, les statistiques non-paramétriques, et la régression multiple. De plus, chaque commande connaît plusieurs options des statistiques supplémentaires.

Finalement, Le SPSS comprend des autres programmes sur les statistiques avancées (ADVANCED STATISTICS), la présentation des résultats en tableaux de haute qualité (TABLES), et en graphiques (GRAPHS).

Ce texte s'occupe seulement des quatre premiers points. De plus, parmi ces quatre points ils existent des commandes qui ne seront pas traitées. Plus spécifiquement, les commandes suivantes qui concernent la manipulation des fichiers sont ignorées:

```
BEGIN DATA      END DATA
EXPORT          IMPORT
TRANSLATE.
```

Toutes ces commandes sont rarement utilisées après l'arrivée du programme DATA ENTRY. Aussi, la commande PLOT pour créer des graphiques est ignorée vue l'abondance des logiciels spécialisés en graphiques.

B. Fichiers SPSS:

Dans le SPSS il existe trois principales sortes de fichiers:

*.SYS = fichier des données ou de la saisie des données
*.LOG = fichier de programme
*.LIS = fichier des résultats

Quant à un fichier des données qui est déjà transformé en fichier interne de SPSS, il a deux états: *passif* et *actif*.

Un fichier est *passif* s'il existe en forme permanente sur le disque dur ou sur une disquette.

Un fichier est *actif* s'il est appelé dans la mémoire active où il peut être traité par les commandes d'un programme de SPSS. Le fichier *actif* est représenté aux commandes par l'asterisque (*).

C. La Façon de Traiter les Données Manquantes en SPSS.

Les cas d'un fichier sont traités comme manquants par les programmes de SPSS suivant deux façons.

(1) **SYSMIS** ou "." défini internellement par le programme SPSS pour tous les cas dont les données n'existent pas (c'est-à-dire les cas absents). Ces cas sont nommés MISSING aux résultats imprimés, et sont ignorés des pourcentages "valid", des pourcentages cumulatifs, et des autres statistiques et calculs du SPSS par défaut.

(2) **MISSING** défini extérieurement par l'utilisateur du programme en utilisant la commande **MISSING VALUE**. C'est-à-dire que l'utilisateur peut déclarer que tous les cas d'une variable spécifique qui sont égaux à une valeur indiquée sont traités comme des cas absents et ignorés par le programme. Ces cas sont groupés ensemble avec les cas SYSMIS et nommés MISSING aux résultats imprimés, et sont ignorés des pourcentages "valid", des pourcentages cumulatifs, et des autres statistiques et calculs du SPSS par défaut.

La différence donc, entre les cas SYSMIS et MISSING est qu'aucune observation n'existe pour les cas SYSMIS, tandis que les cas MISSING ont des observations et sont définis MISSING par l'utilisateur en utilisant la commande **MISSING VALUE** ci-dessous.

La commande pour définir des valeurs MISSING par l'utilisateur.

MISSING VALUE *liste des variables (valeur)*

Toutes les variables nommées dans la liste seront traitées comme absentes (MISSING) pour la valeur indiquée entre les parenthèses.

L'utilisateur de SPSS rencontrera deux méthodes aussi de traiter les cas SYSMIS/MISSING dans les calculs des statistiques qui demandent la présence de plusieurs variables (telles que la régression multiple, le tableau-croisé, etc.): **listwise** et **pairwise**.

listwise

La méthode **listwise** (le défaut) considère les cas de toutes les variables pour chaque cas. S'il y a même une variable parmi la liste des variables nommées pour faire la statistique en question qui a une valeur SYSMIS ou MISSING pour le cas particulier, le cas est rejeté pour toutes les variables. Donc, même si le calcul en question n'utilise que deux variables (le calcul d'une corrélation comme partie d'une régression multiple, par exemple), le cas est traité comme MISSING s'il y a des autres variables qui font partie de la sous-commande en question sans une valeur valide pour le cas particulier en dehors des deux variables nécessaires pour le calcul de la corrélation.

pairwise

Le programme considère seulement les cas des variables nécessaires pour chaque calcul. Alors, la situation peut se présenter où toutes les variables nommées pour une statistique quelconque n'ont pas des cas valides sauf les deux variables nécessaires pour le calcul. **Pairwise** va traiter ce cas comme valide et ignore le fait que les autres variables n'ont pas des cas valides.

Il existe aussi les fonctions qui réfèrent directement aux cas SYSMIS ou MISSING.

Fonctions des Valeurs-Absentes:

- SYSMISC ()** assigne la valeur 1 (c'est à dire vraie dans un sens logique) si la variable est SYSMIS pour le cas en question; sinon elle assigne la valeur 0 (c'est à dire fausse dans un sens logique).
- MISSINGC ()** assigne la valeur 1 (c'est-à-dire vraie dans un sens logique) si la variable est MISSING ou SYSMIS pour le cas en question; sinon elle assigne la valeur 0 (c'est à dire fausse dans un sens logique).
- VALUEC ()** utilise la valeur du cas en question, même si l'utilisateur l'a déjà assigné MISSING.
-

D. Opérateurs et Fonctions du Programme

Opérateurs arithmétiques:

+	Addition
-	Soustraction
*	Multiplication
/	Division
**	Exponentiel

Opérateurs qui gouvernent la relation entre deux variables:

EQ ou =	est égale à
NE ou <>	n'est pas égale à
LT ou <	est inférieur à
LE ou <=	est égale ou inférieur à
GT ou >	est supérieur à
GE ou >=	est égale ou supérieur à

Opérateurs Logiques

AND ou &	ET: le programme inclut seulement les cas qui satisfont les deux conditions à la fois. <u>par exemple:</u> condition 1 AND condition 2
OR ou	OU: le programme inclut les cas qui satisfont l'une ou l'autre (ou toutes les deux) des deux conditions. <u>par exemple:</u> condition 1 OR condition 2
NOT ou ~	NON/PAS: le programme exclut tous les cas qui satisfont la condition. <u>par exemple:</u> NOT(condition 1)

Fonctions Numériques:

arg = Argument, c'est à dire le nom d'une variable ou une expression mathématique. La transformation s'applique à tous les cas y impliqués.

ABS(<i>arg</i>)	Valeur absolue. Devient SYSMIS si l'argument est absent.
RND(<i>arg</i>)	Arrondir la valeur absolue, puis rattacher le signe mathématique ($\pm$). Devient SYSMIS si l'argument est absent.
TRUNC(<i>arg</i>)	Donner la partie entière d'un nombre. Devient SYSMIS si l'argument est absent.
MOD10(<i>arg</i>)	La reste après avoir divisé par 10. Devient 0 si l'argument est égal à 0 et SYSMIS si l'argument est absent.

SQRT(arg)	Élever au carré. Devient SYSMIS si l'argument est négatif ou absent.
EXP(arg)	"e" est élevé à l'argument (qui sert de l'exposant) ou "e" = 2.7148... c'est à dire le logarithme naturel. Devient SYSMIS si l'argument est absent ou si l'argument produit un résultat trop grand pour être représenté.
LG10(arg)	Logarithme de base 10. Devient SYSMIS si l'argument est négatif, absent, ou égale à 0.
LN(arg)	Logarithme naturel, c'est à dire de base "e". Devient SYSMIS si l'argument est absent.
SIN(arg)	Sinus de l'argument. La fonction assume que l'argument est défini en radians. Devient SYSMIS si l'argument est absent.
COS(arg)	Cosinus de l'argument. La fonction assume que l'argument est défini en radian. Devient SYSMIS si l'argument est absent.
ARTAN(arg)	Arc-tangente de l'argument. Devient SYSMIS si l'argument est absent. Le résultat est présenté en radians.

Fonctions qui gouvernent les chiffres au hasard:

UNIFORM(arg)

Un chiffre généré d'une façon "supposée au hasard" par une distribution uniforme; c'est à dire que le résultat est un chiffre tiré au hasard d'une distribution uniforme qui varie entre 0 et la valeur de l'argument. L'argument doit être soit une constante numérique, soit le nom d'une variable.

NORMAL(arg)

Un chiffre généré d'une façon "supposée au hasard" par une distribution normale, c'est à dire que le résultat est un chiffre tiré au hasard d'une distribution normale avec une moyenne de 0 et un écart-type égale à la valeur de l'argument. L'argument doit être soit une constante numérique, soit le nom d'une variable. La fonction devient SYSMIS si l'argument est absent ou égal à 0.

Fonction de la Date:

YRMODAC(l'année, le mois, le jour)

Convertir l'année, le mois, et le jour en chiffre qui représente le nombre des jours depuis le 15 Octobre, 1582.

Fonction qui s'applique aux cas successifs:

LAG(arg)

La valeur du cas précédent. L'argument doit être une variable numérique. LAG(X) est la valeur du cas précédent la variable X. Si les cas sont tirés d'un fichier, LAG est toujours la valeur du cas tiré précédemment. Pour le premier cas qui n'a pas de précédent LAG devient SYSMIS.

E. Les Règles d'Opération:

1. Les caractères foncés représentent les commandes d'un programme de SPSS.
2. Le terme DEFAULT (défaut) veut dire les sous-commandes qui seront utilisées par le programme si aucune autre indication est donnée.
3. Le programme SPSS reconnaît les trois premiers caractères de chaque commande. Par conséquent,
FREQUENCIES = FRE = FREXXXX.
4. Toutes les commandes d'un programme doivent terminer avec un point.
5. Le nom d'une variable ne peut pas dépasser huit caractères.
6. Le nom d'une variable doit nécessairement commencer avec soit un caractère alphabétique, soit un chiffre numérique.

Acceptable:	A123xx_s	122csxx	
Pas acceptable:	_123xxA	\$ccx12	%ss12

(3) toucher F8 pour travailler avec le bloc défini.
(4) Finalement, toucher F7 une dernière fois pour éliminer les indications sur le bloc.

LINES = Définir un bloc par lignes de texte
RECTANGLE = Définir un bloc par rectangle
COMMAND = Indiquer des commandes comme région

F8 MANIPULER UNE RÉGION DÉFINIE:

COPY = Copier la région définie
MOVE = Déplacer la région définie
DELETE = Effacer la région définie

F9 WRITE WHOLE FILE = Faire la sauvegarde de fichier actuel (SPSS va demander le nom)
DELETE = Effacer le fichier (SPSS va demander le nom)

F10 RUN FROM CURSOR = Exécuter le programme entier qui se trouve sur l'écran
RUN MARKED AREA = Faire marcher la région définie
EXIT TO PROMPT = Quitter SPSS pour retrouver le DOS

B. Utilisation des clés de mouvement:

Home Aller au début de l'écran visible
End Aller à la fin de l'écran visible
Ctrl-Home Aller au début du fichier ou texte
Ctrl-End Aller à la fin du fichier ou texte
Ctrl-PgUp Placer la ligne du curseur à la première ligne de l'écran.
Ctrl-PgDn Placer la ligne du curseur à la dernière ligne de l'écran.
Ctrl-(←) Aller au début de la ligne où le curseur se trouve
Ctrl-(→) Aller à la fin de la ligne où le curseur se trouve

III. Les Commandes pour la Manipulation des Fichiers:

- A. GET FILE
- B. SAVE OUTFILE
- C. N
- D. SAMPLE
- E. LIST
- F. MODIFY VARS
- G. JOIN MATCH
- H. JOIN ADD
- I. SORT
- J. SYSFILE INFO
- K. FINISH

GET FILE = 'nom-du-fichier.SYS'
/DROP = liste des variables.

SPÉCIFICATION MINIMALE:

GET FILE = 'nom-du-fichier.SYS'.
.....

La commande GET FILE appelle à la mémoire active un fichier SPSS déjà existant.

/DROP On a l'option de dropper les variables nommées du fichier appelé, tel que le fichier actif ne comprendra que les variables autre que les variables droppées.

SAVE OUTFILE = 'nom-du-fichier.SYS'
/DROP = liste des variables.

SPÉCIFICATION MINIMALE:

SAVE OUTFILE = 'nom-du-fichier.SYS'.
.....

Pour sauvegarder en permanence en fichier SPSS, le fichier qui existe actuellement dans la mémoire active.

/DROP On a l'option de dropper les variables nommées du fichier actif tel que le fichier sauvé ne comprendra que les variables autre que les variables droppées.

N le-nombre-d'enregistrement-désiré.

Pour limiter le nombre de cas à utiliser dans le programme. Cette commande est utile pour expérimenter le programme.

SAMPLE *fraction de l'échantillon* ou *n* FROM *N*.

La commande **SAMPLE** tire au hasard un sous-échantillon des cas du fichier actif qui sera temporairement actif pendant les calculs de la procédure suivante et après laquelle le fichier actif entier sera encore dominant.

fraction de l'échantillon La fraction de l'échantillon qu'on veut utiliser comme sous-échantillon.

n FROM *N* Le nombre des cas (*n*) à tirer du nombre de cas totaux (*N*).

LIST VARIABLES = *liste des variables*
/CASES = FROM *premiere-cas* TO *derniere-cas* BY *le-pas*
/FORMAT = *genre de format*.

SPÉCIFICATION MINIMALE:

LIST.

.....
Pour lister à l'écran les cases spécifiées. Si on ne spécifie pas le CASES, la commande LIST affiche tous les cas du fichier.

/CASES Indiquer les cas à lister sur l'écran.
FROM Lister à partir du numéro indiqué après FROM.
TO Lister jusqu'au numéro indiqué après TO.
BY Lister tous les *n* cas où *n* est le numéro indiqué après BY.

/FORMAT Indiquer le format de la présentation de la liste des cas.

unnumbered N'inclure pas le numéro d'ordre (c.a.d., de la séquence) de chaque cas listé. [Le défaut]
numbered Inclure le numéro de l'ordre (c.a.d., de la séquence) de chaque cas listé.
wrap Si le cas dépasse une seule ligne, continue sur la prochaine ligne. [Le défaut]
single Chaque cas est limité à une seule ligne.
weight Lister la pondération du cas au fichier actif.

```

JOIN MATCH /TABLES = 'nom-du-fichier.SYS'
           /KEEP = liste des variables
           /DROP = liste des variables
           /RENAME = (anciens noms des variables =
                       nouveaux noms des variables)
           /FILE = 'nom-du-fichier.SYS'
           /FILE = 'nom-du-fichier.SYS'
           .
           .
           /BY liste des variables classificatrices
           /MAP.

```

SPÉCIFICATION MINIMALE:

```

JOIN MATCH /TABLES = 'nom-du-fichier.SYS'
           /FILE = 'nom-du-fichier.SYS'
           /BY nom d'une variable commune pour trier les cas.
.....

```

- /TABLE** Le nom du fichier de base auquel les autres variables seront ajoutées.
- /RENAME** Assigne des nouveaux noms aux variables spécifiées du fichier actif.

NOTER: on utilise la commande **RENAME** pour éviter le double emploi du même nom par deux variables différentes qui se trouvent dans deux fichiers différents nommés par le **JOIN MATCH** commande. Pour que la **JOIN MATCH** commande réussisse, il faut que toutes les variables qui seront combinées dans le nouveau fichier actif aient les noms différents.

- /DROP** Élimine les variables indiquées du fichier actif, ce qui est souvent important si le programme est long et les variables sont nombreuses.
- /KEEP** Retient seulement les variables indiquées dans le fichier actif.

NOTE: on doit choisir entre **KEEP** et **DROP**; c'est à dire que les deux commandes ne peuvent pas se présenter à la fois.

- /FILE** Le nom du fichier duquel le programme prendra des variables pour ajouter aux cas similaires du fichier de base.
 - /BY** Liste les variables classificatrices entre le fichier de base nommé à la sous-commande **/TABLES**, et les autres fichiers nommés aux sous-commandes **/FILES**.
 - /MAP** Montre un tableau des variables et leur ordre dans le fichier actif. Cette commande permet de voir si les changements désirés seront produit par les sous-commandes spécifiées à la commande **MODIFY VARS**.
-

Par Exemple:

Supposons qu'il y a deux fichiers: FICH_X et FICH_Y qui possèdent les variables X1 ... XI et Y1 ... YJ, respectivement. Et les variables communes: PREFID (pour identifier la prefecture), MEN (pour identifier le ménage), et ORD (pour identifier l'individu). FICH_X contient N cas, pendant que FICH_Y contient M cas.

Cas	PREFID	MEN	ORD	X1	...	XI		Cas	PREFID	MEN	ORD	Y1	...	YI
1	:	:	:	:		:		1	:	:	:	:		:
...								...						
N								M						

Ensuite, on forme la commande:

```
JOIN MATCH TABLE = 'FICH_X.SYS'  
/FILE = 'FICH_Y.SYS'  
/BY PREFID MEN ORD.
```

Le résultat est le nouveau fichier actif de:

Cas	PREFID	MEN	ORD	X1	...	XI	Y1	...	YI
1	:	:	:	:		:	:		:
...									
N									

Les autres options dont on peut s'en servir sont: **KEEP**, **DROP**, **RENAME** et **MAP**. Les considérations additionnelles sont les suivantes:

(1) Si le cas indiqué par les variables d'identification (PREFID, MEN, et ORD) n'existe pas dans FICH_Y les variables Y1 ... YJ du dit cas auront la valeur de SYSMIS au nouveau fichier.

(2) Si un cas existe au fichier spécifié par le **FILE** qui n'existe pas au fichier spécifié par le **TABLE** il sera ignoré par le nouveau fichier créé par la commande **JOIN MATCH**.

```

JOIN ADD /FILE = 'nom-du-fichier.SYS'
        /KEEP = liste des variables
        /DROP = liste des variables
        /RENAME = (anciens noms des variables =
                    nouveaux noms des variables)
        /FILE = 'nom-du-fichier.SYS'
        /FILE = 'nom-du-fichier.SYS'
        .
        .
        /BY liste des variables classificatrices
        /MAP.

```

SPÉCIFICATION MINIMALE:

```

JOIN ADD /FILE = 'nom-du-fichier.SYS'
        /FILE = 'nom-du-fichier.SYS'.

```

.....

/FILE Le nom du fichier duquel le programme prendra des cas pour ajouter aux cas des variables similaires du fichier de base.

/RENAME Assigne des nouveaux noms aux variables spécifiées du fichier précédent.

NOTE: On utilise la commande **RENAME** pour éviter deux noms différents pour la même variable qui se trouve à deux fichiers différents. C'est à dire qu'une variable qui se trouve aux deux fichiers différents doit avoir un seul nom dans le fichier créé par la commande **JOIN ADD**, sinon la commande ne réussira pas.

/DROP Élimine les variables indiquées du fichier actif, ce qui est souvent important si le programme est long et les variables sont nombreuses.

/KEEP Retient seulement les variables indiquées dans le fichier actif.

NOTE: on doit choisir entre **KEEP** et **DROP**; c'est à dire que les deux commande ne peuvent pas se présenter à la fois.

/BY Suivant les variables classificatrices nommées après la sous-commande **BY**, les cas de tous les fichiers seront rangés en ordre montant.

/MAP Montre un tableau des variables et leur ordre dans le fichier actif. Cette commande permet de voir si les changements désirés seront produit par les sous-commandes spécifiées dans la commande **MODIFY VARS**.

.....

Par Exemple:

Supposons qu'il y a deux fichiers: FICH_X1 et FICH_X2 qui possèdent les variables communes: PREFID (pour identifier le prefecture), MEN (pour identifier le ménage), ORD (pour identifier l'individu) et X1 ... XI. FICH_X1 contient N cas, pendant que FICH_X2 contient M cas.

Cas	PREFID	MEN	ORD	X1	...	XI
1	:	:	:	:	:	:
...						
N						

FICH_X1

Cas	PREFID	MEN	ORD	X1	...	XI
1	:	:	:	:	:	:
...						
M						

FICH_X2

Ensuite, on forme la commande:

```
JOIN ADD /FILE = 'FICH_X1.SYS'  
        /FILE = 'FICH_X2.SYS'.
```

Le résultat est le nouveau fichier actif de:

Cas	PREFID	MEN	ORD	X1	...	XI
1	:	:	:	:	:	:
...						
N+M			*			

=

FICH_X1
FICH_X2

Les autres options dont on peut s'en servir sont: **BY KEEP**, **DROP**, **RENAME** et **MAP**. Les considérations additionnelles sont les suivants:

(1) Si on ajoute la commande **BY** suivie par une liste des variables classificatrices, tous les cas des fichiers nommés (/FILE=) seront rangés selon les valeurs des dites variables classificatrices.

(2) Si le cas indiqué par les variables d'identification (PREFID, MEN, et ORD) existe dans tous les deux fichiers (FICH_X1 & FICH_X2) le cas du premier fichier aura priorité, même s'il est **SYSMIS** au premier fichier.

POINTS COMMUNS ENTRE JOIN MATCH & JOIN ADD:

1- On peut substituer * à la commande **JOIN MATCH** pour représenter le fichier actif, ce qui est utile si on a créé des nouvelles variables.

2- Tous les fichiers utilisés à la commande **JOIN MATCH** doivent être déjà rangés (en utilisant la commande **SORT**) par les variables indiquées à la commande **BY**.

Dans la même façon, avant d'utiliser la commande **BY** tous les fichier utilisé à la commande **JOIN ADD** doivent être déjà rangés (en utilisant la commande **SORT**) par les variables indiquées à la commande **BY**.

SORT BY *liste des variables.*

Pour classer toutes les variables du fichier suivant l'ordre des valeurs des variables nommées dans la liste. Par "default" les cas sont rangés par l'ordre ascendant; ce qui veut dire 0,1,2,3,... pour les chiffres et A,B,C,... pour les caractères.

La classification commence par la première variable nommée ensuite pour chaque valeur de la première variable la deuxième variable est rangée en ordre et ainsi de suite. Chaque variable successive est classée en ordre pour chaque valeur de sa variable précédente.

.....
Par Exemple: **SORT BY** **PREF** **SECT** **MENAGE** **ORDRE.**

Cette commande classera les observations par l'ordre des individus dans chaque ménage, puis par l'ordre des ménages dans chaque secteur, et finalement par l'ordre des secteurs dans chaque préfecture qui sont elles-mêmes classées en ordre. Le tableau suivant détaille cette procédure:

PREF1	SECT1	MENAGE1	ORD1
			ORD2
			.
			.
		MENAGE2	ORD1
			ORD2
		.	.
		.	.
		.	.
	SECT2	MENAGE1	ORD1
			ORD2
			.
			.
		MENAGE2	ORD1
			ORD2
	.	.	.
	.	.	.
	.	.	.
PREF2	SECT1	MENAGE1	ORD1
			ORD2

Pour changer l'ordre de classement, il suffit de placer le caractère D entre parenthèses derrière la variable (ou les variables) dans la liste des variables.

Par Exemple: **SORT BY** **PREF** **SECT** **MENAGE** **ORD (D)**

Cette commande produira le même tableau qu'en haut mais l'ordre des individus dans le ménage sera inversé.

SYSFILE INFO 'nom-de-fichier.SYS'.

Pour presenter des informations contenues dans le fichier appelle, c'est-à-dire, les variables, la date de création, le nombre des cas, etc.

FINISH.

Cette commande est utilisé pour terminer une session de travail avec le programme SPSS.

IV. Les Commandes pour la Manipulation et Creation des Variables

- A. MISSING VALUE
 - B. COMPUTE
 - C. COUNT
 - D. IF
 - E. SELECT IF
 - F. PROCESS IF
 - G. RECODE
 - H. VARIABLE LABEL
 - I. VALUE LABEL
 - J. FORMAT
 - K. DISPLAY
 - L. AGGREGATE
-

MISSING VALUE *liste des variables (valeur)*

Toutes les variables nommées dans la liste seront traitées comme absentes (MISSING) pour la valeur indiquée entre les parenthèses.

COMPUTE variable-cible = expression.

La commande **COMPUTE** crée une nouvelle variable ou modifie une variable existante pour chaque cas des variables utilisés dans l'expression mathématique. L'expression mathématique peut se servir des opérateurs¹ et fonctions détaillés ci-dessous.

Opérateurs arithmétiques:

- + Addition
- Soustraction
- * Multiplication
- / Division
- ** Exponentiel

Opérateurs qui gouvernent la relation entre deux variables:

- EQ ou = est égale à
- NE ou <> n'est pas égale à
- LT ou < est inférieur à
- LE ou <= est égal ou inférieur à
- GT ou > est supérieur à
- GE ou >= est égal ou supérieur à

Opérateurs Logiques

- AND ou & ET: le programme inclut seulement les cas qui satisfont toutes les deux conditions à la fois.
 - OR ou | OU: le programme inclut les cas
-

¹ Voir la section I.D. pour plus de détail sur les opérateurs et fonctions.

qui satisfont l'une ou l'autre (ou toutes les deux) des deux conditions.

NOT ou **~** **NON/PAS**: le programme exclut tous les cas qui satisfaisaient la condition.

Fonctions qui gouvernent les chiffres au hasard:

UNIFORM(arg) Produire un chiffre au hasard tiré d'une distribution uniforme.

NORMAL(arg) Produire un chiffre au hasard tiré d'une distribution normale.

Fonctions des Valeurs-Absentes:

SYSMIS(.) assigne la valeur 1 si la variable est SYSMIS pour le cas en question; sinon elle assigne la valeur 0

MISSING(.) assigne la valeur 1 si la variable est MISSING ou SYSMIS pour le cas en question; sinon elle assigne la valeur 0

VALUE(.) utilise la valeur du cas en question, même si l'utilisateur l'a déjà assigné MISSING.

Fonctions Numériques:

ABS(arg) Valeur absolue.

RND(arg) Arrondir la valeur absolue, puis rattacher le signe mathématique ($\pm$).

TRUNC(arg) Donne la partie entière d'un nombre.

MOD10(arg) La reste après avoir divisé par 10.

SQRT(arg) Elever au carré.

EXP(arg) "e" est élevé à l'argument (qui sert de l'exponentielle de l'argument)

LG10(arg) Logarithme de base 10.

LN(arg) Logarithme naturel.

SIN(arg) Sinus de l'argument.

COS(arg) Cosinus de l'argument.

ARTAN(arg) Arc-tangente de l'argument.

Fonction de la Date:

YRMODA(l'année, le mois, le jour)

Fonction qui s'applique aux cas successifs:

LAG(arg) La valeur du cas précédent.

.....

Par exemple: **COMPUTE DNAN = DNI/10000.**

COUNT *nom de la variable cible =*
liste des variables (liste des valeurs)

La commande COUNT crée une nouvelle variable, appelée la variable cible, qui compte la fréquence d'une valeur spécifique (ou une liste des variables spécifiques) pour chaque cas des variables indiquées.

.....

Par Exemple:

COUNT NVAR = VAR1 VAR2 VAR3 (1).

Le programme comptera la fréquence de la valeur 1 pour chaque cas de ces trois variables (VAR1, VAR2, et VAR3) et la mettra dans une nouvelle variable appelé NVAR. Donc, les valeurs possibles pour un cas de la nouvelle variable NVAR sont 0, 1, 2, et 3.

RECODE *liste des variables (liste des valeur = valeur)*
/liste des variables (liste des valeur = valeur).

Cette commande permet de changer les valeurs différentes en une nouvelle valeur commune. Mais il faut noter que toutes les commandes qui suivent RECODE sont conditionnées par ces changements et que ces changements sont permanents si on fait la sauvegarde du fichier après, c'est-à-dire, les anciennes valeurs sont perdues.

.....

Par exemple:

RECODE AGEMERE AGEPERE (15,16,17,18,19=1) (20,21,22,23,24=2).

IF (expression logique) variable-cible = expression affectée.

La commande IF est utilisée pour créer des nouvelles variables. Si l'expression logique est vérifiée la variable-cible égale l'expression qui se trouve à droite de l'égalité. L'expression logique peut se servir des opérateurs² et fonctions détaillés ci-dessous.

Opérateurs qui gouvernent la relation entre deux variables:

EQ ou =	est égale à
NE ou <>	n'est pas égale à
LT ou <	est inférieur à
LE ou <=	est égal ou inférieur à
GT ou >	est supérieur à
GE ou >=	est égal ou supérieur à

Opérateurs Logiques

AND ou &	ET: le programme inclut seulement les cas qui satisfont toutes les deux conditions à la fois.
OR ou	OU: le programme inclut les cas qui satisfont l'une ou l'autre (ou toutes les deux) des deux conditions.
NOT ou ~	NON/PAS: le programme exclut tous les cas qui satisfaisaient la condition.

Fonctions des Valeurs-Absentes:

SYSMIS(·)	assigne la valeur 1 si la variable est SYSMIS pour le cas en question; sinon elle assigne la valeur 0
MISSING(·)	assigne la valeur 1 si la variable est MISSING ou SYSMIS pour le cas en question; sinon elle assigne la valeur 0
VALUE(·)	utilise la valeur du cas en question, même si l'utilisateur l'a déjà assigné MISSING.

.....
Par exemple: COMPUTE ENFANT = 1.
 IF (AGE GT 16) ENFANT = 0.

² Voir la section I.D. pour plus de détail sur les opérateurs et fonctions.

SELECT IF (*expression logique*).

Pour tirer un sous-échantillon du fichier actif. Toutes les commandes qui suivent le **SELECT IF** vont utiliser ce sous-échantillon. L'expression logique peut se servir des opérateurs³ et fonctions détaillés ci-dessous.

Opérateurs qui gouvernent la relation entre deux variables:

EQ ou =	est égale à
NE ou <>	n'est pas égale à
LT ou <	est inférieur à
LE ou <=	est égale ou inférieur à
GT ou >	est supérieur à
GE ou >=	est égale ou supérieur à

Opérateurs Logiques

AND ou &	ET: le programme inclut seulement les cas qui satisfont toutes les deux conditions à la fois.
OR ou !	OU: le programme inclut les cas qui satisfont l'une ou l'autre (ou toutes les deux) des deux conditions.
NOT ou ~	NON/PAS: le programme exclut tous les cas qui satisfont la condition.

Fonctions des Valeurs-Absentes:

SYSMISC (.)	assigne la valeur 1 si la variable est SYSMIS pour le cas en question; sinon elle assigne la valeur 0
MISSINGC (.)	assigne la valeur 1 si la variable est MISSING ou SYSMIS pour le cas en question; sinon elle assigne la valeur 0
VALUEC (.)	utilise la valeur du cas en question, même si l'utilisateur l'a déjà assigné MISSING .

.....
Par exemple: **SELECT IF (AGEMERE LE 30).**

³ Voir la section I.D. pour plus de détail sur les opérateurs et fonctions.

PROCESS IF (expression logique).

Pour tirer un sous-échantillon du fichier actif. Seulement la commande qui suit le PROCESS IF va utiliser ce sous-échantillon. Dès que la commande qui suit PROCESS IF est terminée le programme reprend l'ensemble du fichier. L'expression logique peut se servir des opérateurs⁴ et fonctions détaillées ci-dessous.

Opérateurs qui gouvernent la relation entre deux variables:

EQ ou =	est égale à
NE ou <>	n'est pas égale à
LT ou <	est inférieur à
LE ou <=	est égale ou inférieur à
GT ou >	est supérieur à
GE ou >=	est égale ou supérieur à

Opérateurs Logiques

AND ou &	ET: le programme inclut seulement les cas qui satisfont toutes les deux conditions à la fois.
OR ou !	OU: le programme inclut les cas qui satisfont l'une ou l'autre (ou toutes les deux) des deux conditions.
NOT ou ~	NON/PAS: le programme exclut tous les cas qui satisfaisaient la condition.

Fonctions des Valeurs-Absentes:

SYSMISC ()	assigne la valeur 1 si la variable est SYSMIS pour le cas en question; sinon elle assigne la valeur 0
MISSING ()	assigne la valeur 1 si la variable est MISSING ou SYSMIS pour le cas en question; sinon elle assigne la valeur 0
VALUE ()	utilise la valeur du cas en question, même si l'utilisateur l'a déjà assigné MISSING.

.....

Par exemple: PROCESS IF (AGEMERE LE 30).
 FREQ VARS = AGEMERE.

⁴ Voir la section I.D. pour plus de détail sur les opérateurs et fonctions.

VARIABLE LABELS *liste des variables* 'etiquette'
/ *liste des variables* 'etiquette'.

Le programme assignera l'etiquette indiquée à la variable nommée (ou à toutes les variables nommées) quand les résultats sont produit en fichier '*.LIS'. L'objectif de l'etiquette est d'améliorer l'identification des variables avec leurs résultats. Les noms habituellement utilisés par le programme pour représenter les variables sont brefs et parfois énigmatiques.

.....

Par Exemple:

VARIABLE LABELS AGEMERE 'Age de la Mere'
/AGEPERE 'Age du Pere'
/AGEENF 'Age de Enfant'.

Note: Entre les apostrophes il ne faut pas utiliser les autres apostrophes sinon le programme traduira la deuxième apostrophes comme la fin de l'etiquette. Par exemple, si on écrit:

AGEENF 'Age de l'Enfant'

le programme va utiliser seulement 'Age de l' pour l'etiquette d'AGEENF. Dans ces conditions, il vaut mieux mettre les guillemets:

AGEENF "Age de l'Enfant"

VALUE LABELS *liste des variables* valeur1 'etiquette'
valeur2 'etiquette' valeur3 'etiquette' etc.

Le programme assignera une étiquette à chaque modalité de la variable indiquée,

.....

Par Exemple:

VARIABLE LABELS PREFECTURE 1 'Kigali' 2 'Butare' 3 'Ruhengeri' ...

FORMATS *liste des variables (spécification de format)*
/liste des variables (spécification de format).

Spécification du Format (Numérique):

Fw.d w = spécifie combien d'espaces (c.à.d., caractères ou chiffres) y compris le point; d = combien de places à la gauche du point.

COMMAw.d Les résultats inclura les virgules (pour indiquer milles).

DOLLARw.d Les résultats inclura la marque indicative du dollar (\$).

.....

Par Exemple:

FORMAT VAR1 (F4.2) VAR2 (DOLLAR9.2) VAR3 (COMMA7.3).

Cette commande produira:

- pour la première variable un format de 4 colonnes, deux qui sont à gauche du point;
- pour la deuxième variable un format de 9 colonnes y compris les virgules et le point avec deux colonnes à gauche du point;
- pour la troisième variable un format de 7 colonnes y compris les virgules et le point avec trois colonnes à gauche du point.

DISPLAY *liste des variables.*

Pour présenter des informations concernant les variables indiquées du fichier actif.

AGGREGAT **OUTFILE** = *nom du fichier*
/PRESORTED
/MISSING = COLUMNWISE
/BREAK = *liste des variables*
/nom de la variable d'aggregation 'etiquette'
= *FONCTION(liste des variables, arguments).*

La commande **AGGREGATE** un nouveau fichier (ou actif, ou sauver au système) à partir du fichier actif. Les cas pour ce nouveau fichier sont créés en rassemblant le nombre des cas du fichier actif qui tombent dans les catégories distinguées par les variables classificatrices nommées à la sous-commande **BREAK**. Le rassemblement est fait suivant les calculs de la fonction indiquée. Les fonctions disponibles pour cette commande sont ci-dessous.

OUTFILE Indique la destination du nouveau fichier créé par la commande **AGGREGATE**.

/PRESORTED Indique que les cas du fichier actif sont déjà triés par les variables classificatrices nommées à la sous-commande **BREAK**.

/MISSING = COLUMNWISE

Déclare le cas **MISSING** dans le nouveau fichier si le cas associé de l'une des variables est **MISSING**. Par défaut, un cas **MISSING** est simplement ignoré, et l'aggregation se fait en utilisant les autres cas valides.

/BREAK Liste les variables classificatrices. Chaque combinaison unique des valeurs définit une catégorie de rassemblement.

LISTE DES FONCTIONS:

SUM(variables) Additionne les cas pour chaque variable listée.

MEAN(variables) Calculer les moyennes des cas pour chaque variable listée.

SD(variables) Calculer les écarts types des cas pour chaque variable listée.

MAX(variables) La valeur maximale des cas pour chaque variable listée.

MIN(variables) La valeur minimale des cas pour chaque variable listée.

PGT(variables, valeur) %age des cas supérieurs à la valeur.

PLT(variables, valeur) %age des cas inférieurs à la valeur.

PINC(variables, valeur1, valeur2) %age des cas entre les deux valeurs.

POUT(variables, valeur1, valeur2) %age des cas qui ne sont pas entre

	les deux valeurs.
FGT(variables, valeur)	La fraction des cas supérieurs à la valeur.
FLT(variables, valeur)	La fraction des cas inférieurs à la valeur.
FIN(variables, valeur1, valeur2)	La fraction des cas entre les deux valeurs.
FOUT(variables, valeur1, valeur2)	La fraction des cas qui ne sont pas entre les deux valeurs.
N(variables)	Le nombre pondéré des cas par "BREAK" groupe.
NUC(variables)	Le nombre non-pondéré des cas par "BREAK" groupe.
NMISS(variables)	Le nombre pondéré des cas "missing" par "BREAK" groupe.
NUMISS(variables)	Le nombre non-pondéré des cas "missing" par "BREAK" groupe.
FIRST(variables)	La première valeur qui n'est pas MISSING ou SYSMIS par "BREAK" groupe.
LAST(variables)	La dernière valeur qui n'est pas MISSING ou SYSMIS par "BREAK" groupe.

V. Les Commandes pour Calculer les Statistiques

- A. DESCRIPTIVES
- B. FREQUENCIES
- C. MEANS
- D. CROSSTABS
- E. T-TEST
- F. ANOVA
- G. ONEWAY
- H. NPAR
- I. REGRESSION

DESCRIPTIVES VARIABLES = liste des variables (all)
/OPTIONS = liste des options
/STATISTICS = liste des statistiques.

SPÉCIFICATION MINIMALE:

DESCRIPTIVES liste des variables.

.....
OPTIONS:

- 1- Inclure les cas MISSING.
- 2- Supprimer les étiquettes des variables.
- 3- Sauver les z-scores au fichier actif.
- 5 Exclure les cas SYSMIS/MISSING listwise.
- 6- Présentation des résultats en serie, c.à.d. les statistiques au-dessous de chaque variable associée.
- 7- Présentation étroite sur la page (79 colonnes).
- 8- Supprimer les noms des variables et utiliser seulement les étiquettes.

STATISTICS:

1	Moyenne de Variable	9	Maximum - Minimum = Amplitude
2	Ecart-Type de Moyenne	10	Minimum
5	Ecart-Type de Variable	11	Maximum
6	Variance	12	Somme
7	Kurtosis	13	Moyenne, Ecart-Type, Min, Max
8	Skewness		

La procédure **DESCRIPTIVES** calcule les statistiques univariates pour toutes les variables numériques spécifiées.

Si on spécifie **all** suivant la sous-commande **VARIABLES**, le programme calculera les statistiques univariates pour toutes les variables numériques

FREQUENCIES VARIABLES = *liste des variables (all)*
 /FORMAT = *liste des choix*
 /MISSING = *include*
 /BARCHART = *specifications*
 /HISTOGRAM = *specifications*
 /HBAR = *specifications*
 /NTILE = *n*
 /PERCENTILES = *liste des valeurs*
 /STATISTICS = *liste des statistiques.*

SPECIFICATION MINIMALE:

FREQUENCIES VARIABLES = *liste des variables (all).*

La procédure FREQUENCIES produit une table des fréquences de chaque valeur, son pourcentage de toutes les observations, le pourcentage cumulatif des observations qui sont égales ou inférieurs à la valeur indiquée, ainsi que d'autres options. HISTOGRAM ajoute un histogramme après la table des fréquences; FREQ(*n*) ordonne les fréquences des cases à tous les "*n*" cases; NORMAL place la distribution normale sur l'histogramme; NTILE=*n* ajoute en bas les *n* cases qui divisent la distribution en "*n*-tiles"; STATS ajoute en bas les quatre caractéristiques: moyenne, écart-type, min, et max.

Les détails des sous-commandes suivent:

/FORMAT	Le défaut produira les tableaux successifs ligne par ligne en ajoutant toutes les étiquettes. Il mettra autant que possible sur une page selon la longueur du défaut. Pour changer le défaut, il faut indiquer des choix parmi les suivants.
condense	Un format condensé. Produire les résultats en trois colonnes avec les chiffres arrondis en nombres entiers.
onepage	Utilise le format condensé seulement pour voir si les résultats nécessiteront plus d'une page.
newpage	Chaque tableau commence sur une nouvelle page.
nolabels	Supprime les étiquettes des valeurs.
double	Rapporte les résultats toutes les deux lignes au lieu de chaque ligne par défaut.
afreq	SORT (range) les catégories par ordre des fréquences qui montent.
dfreq	SORT (range) les catégories par ordre des fréquences qui descendent.
dvalue	SORT (range) les catégories par ordre des valeurs qui descendent.
limit(<i>n</i>)	Les tableaux sont limités à <i>n</i> catégories de

fréquence. Cette commande évite de faire les tableaux des variables continues.

notable Ne produit pas un tableau.

/BARCHART Produit un diagramme à lignes horizontales pour chaque variable nommée.

min(n) La limite basse au-dessous de laquelle les valeurs sont ignorées.

max(n) La limite haute au-dessus de laquelle les valeurs sont ignorées.

freq(n) L'axe horizontal divisé en fréquences où n est le maximum.

/HISTOGRAM Produit une histogramme pour chaque variable nommée.

min(n) La limite basse au-dessous de laquelle les valeurs sont ignorées.

max(n) La limite haute au-dessus de laquelle les valeurs sont ignorées.

percent(n) L'axe horizontal est divisé en pourcentages où n est la maximume.

freq(n) L'axe horizontal est divisé en fréquences où n est l'unité optionnelle.

increment(n) La largeur de l'intervalle est spécifié par n .

normal Le programme imposera la courbe d'une distribution normale au dessus de l'histogramme.

/HBAR Produit un plot pour toutes les variables (numériques et non-numériques). Pour les variables numériques par défaut le programme produira un plot à lignes horizontales s'il y a suffisamment de place sur une page, sinon il produira une histogramme. **HBAR** produira un plot à lignes horizontales pour les variables non-numériques. Toutes les spécifications de la sous-commande **HISTOGRAM** et **BARCHART** s'appliquent pour la commande **HBAR**.

/NTILES Calcule les pourcentages qui divisent la distribution des cas en les nombres des catégories spécifiées (n).

/PERCENTILES Produit la valeur du cas pour laquelle le pourcentage spécifié des cas tombe. On peut spécifier plusieurs pourcentage.

/STATISTICS;

mean Les moyennes des variables.

semean L'erreur standard de la moyenne.

median La médiane.

mode	Le mode.
stddev	L'écart type.
variance	La variance.
skewness	"skewness", c.a.d., le biais vers une direction
seskew	L'erreur standard de la statistique de "skewness".
kurtosis	"kurtosis": le biais au troisièm degré.
sekurt	L'erreur standard de la statistique de "kurtosis".
range	L'étendu des cas pour chaque variable.
minimum	La valeur minimale pour chaque variable.
maximum	La valeur maximale pour chaque variable.
sum	Le effectif.
default	mean, stddev, minimum, maximum.
all	Montrer toutes les caractéristiques.
none	Ne produit pas des caractéristiques statistiques.

/MISSING = include Inclure les cas MISSING.

MEANS VARIABLES = liste 1 des variables
BY liste 2 des variables DISCRETES
/OPTIONS = liste des options
/STATISTICS = numero de statistique désirée.

SPÉCIFICATION MINIMALE:

MEANS VARIABLES = variable dépendante BY variable indépendante.
.....

OPTIONS:

- 1- Inclure les cas MISSING.
- 2- Exclure les cas qui ont des cas MISSING chez la variable dépendante.
- 3- Supprimer toutes les étiquettes.
- 5- Supprimer les comptes de cellules.
- 6- Montrer les sommes des groupes discretes.
- 7- Supprimer les écarts types des groupes discretes.
- 8- Supprimer les étiquettes des valeurs.
- 9- Supprimer les noms des variables indépendantes.
- 10- Supprimer les valeurs des variables indépendantes.
- 11- Supprimer les moyennes des groupes discretes.
- 12- Montrer les variances des groupes discretes.

STATISTICS:

- 1 One-way analysis of variance (Analyse de la Variance à une dimension)
- 2 Test of linearity (test de linéarité)

La commande **MEANS** fournit une liste des moyennes des variables dans liste 1 par catégorie discrète des variables dans liste 2. On a la possibilité de choisir parmi des options. De plus l'addition de la commande **statistics** permet d'inclure une analyse des variances (ANOVA) des moyennes des variables dans liste 1 entre les catégories des variables à liste 2. Cette option (option 1) est intéressante parce qu'elle comprend un F-test qui mesure l'importance des moyennes de la variable dépendante (liste 1) telle que définie par les catégories des variables indépendantes (liste 2) pour expliquer la variation de la variable dépendante autour de sa moyenne globale.

CROSSTABS TABLES = liste-des-variables BY liste-des-variables
/OPTIONS = liste des options
/STATISTICS = numero de la statistique.

SPÉCIFICATION MINIMALE:

CROSSTABS TABLES = variable BY variable.
.....

OPTIONS:

- 1- Inclue les cas MISSING.
- 2- Supprime toutes les étiquettes.
- 3- Montre la pourcentage de cellule par ligne.
- 4- Montre la pourcentage de cellule par colonne.
- 5- Montre les pourcentages totals d'un tableau 2-way (c.a.d., qui compare seulement deux variables discrètes.
- 6- Supprime les étiquettes des valeurs.
- 8- Range les lignes horizontales par valeurs descendantes.
- 12- Supprime les tableaux (produit simplement les statistiques commandés).
- 13- Supprime les comptes des cellules.
- 14- Montre les fréquences prédites.
- 15- Montre les résiduels de la statistique chi-carré.
- 16- Montre les résiduels standardisés de la statistique chi-carré.
- 17- Montre les résiduels standardisés ajustés de la statistique chi-carré.
- 18- Montre toutes les informations possibles des cellules.
- 19- Supprime les valeurs du tableau.

STATISTICS:

- 1 Chi-carré
- * Il y a plusieurs autres statistiques mais ce n'est que le chi-carré qui merite d'être mentionner si l'utilisateur n'est pas un statisiticien par formation.

Le commande **CROSSTABS** produira un tableau d'une distribution jointe des variables nommées. Chaque variable sera catégorisée par les valeurs de l'autre variable telle que une cellule est créé pour chaque croisement distinct entre les deux variables. Par conséquent il faut que les variables soient discrètes chacune ne disposant de plus de cinq ou six catégories (bien entendu que le nombre des catégories praticable dépendra sur la quantité et la distribution des données.

(Pour les Cas Indépendents: Independent Samples)

T-TEST GROUPS = variable discrète (spécification de gamme)
/VARIABLES = liste des variables dépendantes
/OPTION = les numéros des options.

SPECIFICATION MINIMALE:

T-TEST GROUPS = nom de la variable discrète
/VARIABLES = nom de la variable dépendante.

(Pour les Cas Accouplés: Paired Samples)

T-TEST PAIRS = liste des variables WITH liste des variables
/OPTION = les numéros des options.

SPECIFICATION MINIMALE:

T-TEST PAIRS = les noms de deux variables.

OPTIONS:

- 1- Inclure les cas MISSING.
- 2- Exclure les cas SYSMIS/MISSING suivant la méthode listwise.
- 3- Supprime les étiquettes des variables.
- 5- Accouplement spécial pour WITH; la première variable avant WITH est comparée avec la première variable après WITH; la deuxième variable avant WITH est comparée avec la deuxième variable après WITH, etc.

La procédure T-TEST compare les moyennes estimées à partir de l'échantillon avec un t-test (Student's t) et produit la probabilité de signification (à deux cotés de la distribution) pour la différence entre les deux moyennes.

Pour les Cas Indépendents (Independent Samples)

La GROUPS commande est limitée à une seule variable qui comprend deux catégories discrètes. Si elle comprend plus de deux catégories il faut ajouter la spécification de gamme entre les parenthèses suivant la variable discrète. Quant à la spécification de gamme entre les parenthèses on a deux possibilités: (1) un seul chiffre qui divise la variable en deux; les cas égaux ou supérieurs, et les cas inférieurs; (2) deux chiffres dont le premier représente une catégorie et le deuxième représente l'autre catégorie (toutes les autres valeurs étant supprimées).

Pour les Cas Accouplés (Paired Samples)

Si on n'inclut pas le mot clé WITH, toutes les variables nommées seront comparées l'une avec l'autre. Si on inclut le mot WITH, toutes les variables avant WITH seront comparées avec toutes les variables après WITH.

ANOVA VARIABLES = liste des variables dependentes
BY liste des FACTORS(MIN,MAX)
WITH liste des COVARIATES
/OPTIONS = liste des options
/STATISTICS = numero de la statistique.

SPÉCIFICATION MINIMALE:

ANOVA VARIABLES = variable dependente BY FACTOR(MIN,MAX).

.....

OPTIONS:

- 1 Inclure les valeurs MISSING
- 2 Supprime les étiquettes
- 3 Supprime toutes les interactions
- 4 Supprime les interactions de 3 dimensions
- 5 Supprime les interactions de 4 dimensions
- 6 Supprime les interactions de 5 dimensions
- 7 Quant aux "sommes des carrés": les COVARIATES sont décomposées simultanément avec les FACTORS
- 8 Quant aux "sommes des carrés": les COVARIATES sont décomposées après les FACTORS
- 9 Utilise la méthode de régression multiple: tous les effets (FACTORS, COVARIATES, et INTERACTIONS) sont considérés simultanément en tenant compte l'un de l'autre.
- 10 Utilise une méthode hiérarchique: les FACTORS et les COVARIATES sont considérés suivant un ordre hiérarchique selon leur ordre dans leur liste particulière; chaque effet considérant seulement les effets précédents

STATISTICS:

- 1- Analyse des classifications multiples. Le résultat dépend de l'option choisi.
 - 2- Montre les coefficients d'une régression non-standardisée qui est calculée au moment que les COVARIATES entre le calcul de la décomposition de la variance en sommes carrés.
 - 3- Montre les moyennes et fréquences de la DEP VAR (variable dépendante) par cellule des FACTORS croisés.
- ALL- Montre les résultats produits par les statistiques 2 et 3 ci-dessus.
-

La procédure ANOVA analyse la variance de la variable dépendante appelée DEP VAR par rapport aux variables catégoriques (discrètes) appelées FACTORS (ou effets principaux) et par rapport aux variables continues appelées COVARIATES. Une analyse sera produit pour chaque DEP VAR listée.

Les valeurs de la DEP VAR sont séparées par les catégories

discrètes des FACTORS. L'ANOVA utilise un F-test pour tester l'hypothèse que les moyennes de la DEP VAR sont égales pour toutes les catégories discrètes des FACTORS. Le programme prendra en considération les effets des COVARIATES ainsi que des interactions entre toutes les permutations mathématiques des FACTORS jusqu'à la cinquième (5 dimensions) permutation (c'est à dire, toutes les combinaisons des cinq facteurs à la fois).

DEFAULT: Le programme inclut toutes les interactions jusqu'à la cinquième permutation des FACTORS. Les "sommes des carrés" sont décomposées suivant une approche classique de la méthode expérimentale; c'est à dire que l'on considère les COVARIATES, les FACTORS, et finalement les ordres descendantes des interactions.

ONEWAY VARIABLES = liste des variables dependentes
BY FACTOR(MIN,MAX)
/POLYNOMIAL = n
/CONTRAST = liste des coefficients
/RANGES = nom du test
/OPTIONS = liste des options
/STATISTICS = numero de la statistique.

SPÉCIFICATION MINIMALE:

ONEWAY VARIABLES = variable dependente BY FACTOR(MIN,MAX).
.....

OPTIONS:

- 1- Inclure les cas MISSING.
- 2- Exclure les cas "listwise" (le DEFAULT est "pairwise").
- 3- Supprime les étiquettes des variables.
- 4- Garde une matrice des fréquences, moyennes, et écarts types par catégorie du FACTOR
- 6- Utilise les étiquettes des valeurs pour les catégories du FACTOR.
- 7- Lire une matrice des fréquences, moyennes, et écarts types.
- 8- Lire une matrice des fréquences, moyennes, variances combinées et le nombre des "degré de liberté" (degrees of freedom).

STATISTICS:

- 1- Calcule les statistiques descriptives des cas par catégorie.
- 2- Calcule les statistiques "Fixed -Effects" et "Random -Effects".
- 3- Teste l'homogénéité des variances par catégories.

TEST DU RANGE:

- LSDC(p) Différence significative la plus petite.
DUNCAN(p) Test des rangées multiples.
SNK Test de Student-Newman-Keuls.
BTUKEY Procédure alternative de Tukey.
TUKEY Différence significative "honnête".
MODLSDC(p) Version modifiée de la différence significative la plus petite.
SCHEFFEC(p) Test de Scheffe.

La procédure ONEWAY est limitée à une seule variable indépendante (discrète) appelée le FACTOR pour catégoriser la variable dépendante appelée la DEP VAR. Les catégories du FACTOR déterminent les groupements de la DEP VAR. Les résultats (c.à.d. F-test, Prob. de signification, sommes des carrés, etc.) seront reproduits pour chaque DEP VAR listée.

Le test de ONEWAY compare et teste toutes les moyennes de la DEP VAR par catégories du FACTOR simultanément. Cette méthode ignore

la possibilité d'une différence significative entre deux moyennes de deux catégories spécifiques si la différence de toutes les moyennes n'est pas simultanément importante.

Le commande **RANGE** permet de choisir parmi plusieurs tests qui mesurent la relation de toutes les catégories l'une par rapport à l'autre, ainsi que l'ensemble des moyennes.

Le commande **CONTRAST** permet les comparaisons entre les groupement divers des moyennes de la DEP VAR. Un t-test est utilisé pour tester les hypothèses des contrastes.

Par Exemple:

Supposons que la DEP VAR (Y) est divisée en cinq catégories par le FACTOR (X). On voudrait tester l'hypothèse que la somme des moyennes des premières trois catégories (Y_1, Y_2, Y_3) est égale à la somme des deux dernières catégories (Y_4, Y_5).

$$\text{Hypothèse: } \bar{Y}_1 + \bar{Y}_2 + \bar{Y}_3 = \bar{Y}_4 + \bar{Y}_5$$

Le commande nécessaire pour tester cette hypothèse est le suivant:

$$\text{/CONTRAST} = 1 \quad 1 \quad 1 \quad -1 \quad -1$$

Le programme permettra jusqu'à 10 contrastes.

NPAR TESTS *nom du test = specifications*
/OPTIONS = liste des options
/STATISTICS = numéro de la statistique.

La procédure **NP**AR comprend plusieurs tests non-paramétriques. Un test non-paramétrique est basé sur des suppositions minimales quant à la distribution des données. Au contraire, les autres tests tels que le t-test et le F-test sont basés sur la supposition que les données sont tirées d'une distribution "normale".

Souvent la supposition d'une distribution normale n'est pas justifiée par les données, ou peut-être une autre supposition nécessaire pour valider un test quelconque ne tient pas. Par exemple, le t-test demande que les deux distributions soient normales et que les variances des deux groupements soient homogènes. Les résultats du programme **SPSS** produisent aussi un F-test qui teste l'homogénéité des deux variances. Si le F-test rejette l'hypothèse d'homogénéité le résultat du t-test n'est plus valable. Alors, on a la possibilité d'utiliser le Mann-Whitney (**M-W**) U test (également appelé le Wilcoxon W-test) pour tester la différence entre deux groupements différents (comme classifié par une variable discrète). Le Mann-Whitney test n'exige ni l'homogénéité des variances, ni la normalité de la distribution des variables.

/OPTIONS

- 1- Inclure les cas **MISSING**.
- 2- Exclure les cas **MISSING** suivant la méthode **listwise**.
- 3- Pour deux échantillons rapportés faire l'accouplement des cas.
- 4- Tire des sous-échantillons au hasard.

/STATISTICS

- 1- Moyenne, maximum, minimum, écart-type, et le nombre des cas.
- 2- Quartile et le nombre des cas par quartile.

Noms des tests:

BINOMIAL
CHISQUARE
SIGN
WILCOXON

Reprenez - vous au livre pour des autres tests, et pour le détail sur les tests nommés ci-dessus.

REGRESSION VARS = liste des variables

- /DESCRIPTIVES = liste des statistiques descriptives
- /SELECT = nom-de-variable relation valeur
- /MISSING = choix
- /WIDTH = largeur de la page des résultats
- /REGWGT = nom de variables pour pondérer les calculs

- /READ = lire une matrice des statistiques
- /WRITE = garder une matrice des statistiques

- /STATISTICS = liste des statistiques d'analyse
- /CRITERIA = critère de tolérance des statistiques
- /ORIGIN (la régression traversera l'origine)
- /NOORIGIN (la régression calculera son propre point de traverser l'axe vertical)
- /DEPENDENT = liste des variables dépendantes (DEP VAR)

- /METHOD = la méthode d'entrer les variables à la régression
liste des variables à évaluer suivant la méthode nommée

- /RESIDUALS = la genre d'analyse des résidus de la régression précédentes
- /CASEWISE (un graphique cas par cas des résidus pour tous les résidus qui ont un z-score supérieur à 3)
- /SCATTERPLOT = (nom de variable, nom de variable temporaire) taille du graphique
- /PARTIALPLOT = liste des variables taille du graphique
- /SAVE = variable-temporaire(nouveau nom).

SPÉCIFICATION MINIMALE:

REGRESSION VARIABLES = liste des variables indépendantes & dépendantes
/DEPENDENT liste des variables dépendantes
/METHOD.

.....

La procédure **REGRESSION** calcule les régressions multiples et les statistiques et graphiques (plots) associés aux régressions multiples. Le programme permet le choix de la méthode à faire entrer les variables indépendantes (IND VAR) à la régression. De plus, il y a plusieurs statistiques pour analyser les résiduels et les observations de forte influence.

Une description des sous-commandes suit en commençant par les sous-commandes de la spécification minimale.

/VARIABLES Il y a trois possibilités:

all Inclure toutes les variables du fichier actif.

collect Inclure toutes les variables nommées aux sous-commandes **DEPENDENT** et **METHOD**

liste des variables Nomme toutes les variables pour la régression.

/DEPENDENT Spécifie une liste de toutes les variables dépendente pour laquelle on voudrait produire une régression.

/METHOD Il y a six possibilités:

ENTER Toutes les variables nommées sont entrées simultanément.

BACKWARD À chaque étape la variable avec la Probabilité de Significance la plus élevée pour le F-test est éliminée si la valeur dépasse **POUT**. Les statistiques sont produites pour chaque variable traitées.

FORWARD À chaque étape la variable avec la Probabilité de Significance la plus petite pour le F-test est ajoutée si la valeur est inférieure à **PIN**. Les statistiques sont produites pour chaque variable traitées.

STEPWISE Pour les **IND VAR** déjà dans la régression, la variable avec la Probabilité de Significance la plus élevée pour le F-test est éliminé à chaque étape, pendant que pour les variables qui ne sont pas dans la régression la variable avec la Probabilité de Significance la plus petite pour le F-test est ajoutée si la valeur est inférieure à **PIN**. Les statistiques sont produites pour chaque variable traitées.

REMOVE On spécifie une liste des variables, ensuite, toutes les variables dans la liste sont éliminées de la régression dans un seul coup. Les statistiques sont produites pour le groupement des variables éliminées.

TEST On spécifie des groupements des variables, ensuite le programme teste la présence et absence de chaque groupement sur le changement du R-carré à son niveau de probabilité de signficance. Les statistiques sont produites pour chaque groupement.

.....

Les sous-commandes de contrôle globale:

/DESCRIPTION les statistiques descriptives de toutes les variables de la régression, y compris les **IND VAR** (variables indépendantes ou explicatives) et la **DEP VAR** (variable dépendante ou à expliquer) . Il y a douze possibilités:

none Élimine toutes les statistiques descriptives.
defaults **mean** (moyenne), **stddev** (écart-type), et **corr**
 (corrélation).
mean Les moyennes.
stddev Les écart types.
variance La variance.
corr La matrice des corrélations entre les variables.
sig La probabilité des coefficients de corrélations (d'un
 coté).
badcorr Produit la matrice des corrélations seulement s'il y
 a des corrélations non-calculables.
cov La matrice des variance-covariances entre les
 variables.
xprod Les déviations-croisées des moyennes des variables.
n Le nombre des cas utilisés pour calculer chaque
 coefficient de corrélation.
all Produire toutes les statistiques descriptives.

/SELECT Utilise pour produire une matrice des corrélations et
 les statistiques de la régression qui sont basées sur
 un sous-échantillon des cas (c.à.d. les cas qui
 satisfaient la relation logique spécifiée).

/MISSING Il y a quatre méthode de traiter les cas SYSMIS ou
 MISSING. Le défaut est LISTWISE.

listwise Élimine les cas s'ils ont des variables qui n'ont
 pas des observations.

pairwise Chaque coefficient de corrélation est calculé en
 considérant seulement les cas des variables y
 impliquées. Si l'une des variables a des cas
 absent, lesdits cas sont éliminés.

meansubstitution Remplace les cas manquants avec la moyenne de la
 variable.

include Inclut tous les cas spécifiés MISSING (c.à.d.
 déclarés MISSING par l'utilisateur)

/WIDTH La largeur des résultats produit par la procédure.

/REGWGT La variable utilisé pour pondérer tous les calculs de
 la procédure.

Les sous-commandes qui concernent l'écriture et lecture d'une
 matrice:

/READ Spécifie les matrices/vecteurs à lire par le
 programme. Les possibilités qui sont incluses

sont les suivants:

defaults mean (moyenne), stddev (écart-type), corr (corrélation), et n (le nombre des observations).
means Un vecteur des moyennes qui précède la matrice.
stddev Un vecteur des écart types qui précède la matrice.
variance Un vecteur des variances qui précède la matrice.
corr Une matrice des coefficients de corrélation.
cov Une matrice des coefficients de variance-covariance.
n Un vecteur qui contient le nombre des cas utilisé pour calculer chaque coefficient de corrélation. Il suit la matrice des coefficients de corrélations. Applicable si on a spécifié **pairwise** ou **meansubstitution** pour le traitement des cas absents.

/WRITE

Spécifie les statistiques qui seront sauver dans une matrice temporaire nommé SPSS.PRC. Les statistiques disponibles pour ce sous-commande comprend:

defaults mean (moyenne), stddev (écart-type), corr (corrélation), et n (le nombre des observations).
means Un vecteur des moyennes.
stddev Un vecteur des écart types.
variance Un vecteur des variances.
corr Une matrice des coefficients de corrélation.
cov Une matrice des coefficients de variance-covariance.
n Un vecteur qui contient le nombre des cas utilisé pour calculer chaque coefficient de corrélation. Applicable si on a spécifié **pairwise** ou **meansubstitution** pour le traitement des cas absents.
none Aucune matrice est sauvée (le défaut).

Les sous-commandes qui contrôlent la forme et l'analyse de la régression

/STATISTICS

defaults: R, anova, coeff, outs.
all Toutes les statistiques à l'exception de F, line, end.

Statistiques de l'équation de régression

R R-carré (R^2), R-carré ajusté ($\bar{R}^2$), écart-type standard de l'estimation.
anova Tableau qui présente les statistiques de l'analyse de la variance (régression, sommes des carrés des résidus, F-test, Prob. de Sign. du F-test, etc.)
cha Changement au R-carré changement des F-test et sa

Prob. de Sign., et le F-test et sa Prob. de Sign. après chaque étape.

bcov Matrice des variances-covariances pour les coefficients de la régression non-standardisé.

xtx Matrice de transformation.

cond Les bornes du numéro conditionnel.

coeff Les coefficients de la régression.

outs Les statistiques pour les variables qui ne sont pas encore entrées dans la régression.

zpp Les corrélations des ordre zéro, partie, et partial.

ci Interval de confiance de 95% pour les coefficients non-standardisés de la régression.

ses Erreur standard approximatif des coefficients provenant de la régression standardisé.

tol La tolérance.

F La F-valeur pour les coefficients de la régression et sa probabilité de signification.

line Quant à la méthode **stepwise** d'entrer les variables à la régression, **line** produit une seule ligne d'information sommaire pour chaque étape.

end Produit la même ligne d'information que **line** pour chaque étape pour la méthode **stepwise**, et après chaque groupe des variables pour les autres méthodes.

history Produit un rapport sommaire des étapes.

/CRITERIA Contrôle les critères de tolérance pour les calculs statistiques.

defaults **PINC(0.05)**, **POUT(0.10)**, et **TOLERANCE(0.01)**.

PINC(prob) La probabilité de F pour faire entrer une variable.

FINC(val) La valeur de F pour faire entrer une variable.

POUT(prob) La probabilité de F pour éliminer une variable.

FOUT(val) La valeur de F pour éliminer une variable.

tolerance() La valeur de tolérance.

maxsteps() Le nombre maximum des étapes.

/ORIGIN La régression traversera l'origine par force.

/NOORIGIN La régression calculera son propre point qui traverse l'axe vertical.

/DEPENDENT Voir la description ci-dessus.

La sous-commande qui contrôle la méthode d'entrer les variables à la régression:

/METHOD Voir la description ci-dessus.

On a la possibilité de faire une analyse profonde sur les résidus qui sortent de l'analyse de régression en utilisant la commande **RESIDUALS**. Le programme crée des variables temporaires basées sur les résidus. Ces variables temporaires sont analysées et les statistiques présentées suivant le choix des commandes optionnelles. On peut disposer de 12 variables temporaires:

pred	Les valeurs prédites non-standardisées.
resid	Les résidus non-standardisés.
dresid	Les résidus éliminés.
adjpred	Les valeurs prédites ajustées.
zpred	Les valeurs prédites standardisées.
zresid	Les résidus standardisés.
sresid	Les résidus ajustés à la façon de Student.
sdresid	Les résidus éliminés qui sont ajustés à la façon de Student.
sepred	Les erreurs standards des valeurs prédites.
mahal	Les distances de Mahal.
cook	Les distances de Cook.
lever	Les valeurs d'influence.

Les sous-commandes qui contrôlent l'analyse des résidus de la régression:

/RESIDUALS	Contrôle la présentation et la nomenclature des informations sommaires sur les résidus y compris: la statistique de Durbin-Watson; les histogrammes; un graphique sur la probabilité que les résidus sortent d'une distribution normale; les graphiques et statistiques des "outliers" (c.à.d., les cas éloignés de la moyenne qui influencent fortement le calcul des statistiques), etc.
defaults	size(large) , durbin , normprob(zresid) , histogram(zresid) , outliers(zresid) .
size()	La dimension de la figure est entre les parenthèses: small (petit) ou large (grand).
histogram()	Un histogramme des variables standardisées temporaires est faites pour les variables temporaires nommées entre les parenthèses.
normprob()	Un figure basée sur les probabilités normales des variables temporaire aux valeurs standardisées.
outliers()	Les dix "outliers" (les cas les plus éloignés par rapport aux autres cas), basés sur les valeurs absolues les plus grandes pour leurs variables temporaires nommées entre parenthèses.
durbin	La statistique de Durbin-Watson.
id()	L'identification du cas est rapportée sur les plots des "outliers".
pooled	Si le commande SELECT a été activée, le commande

pooled permet d'utiliser tous les cas, même les éliminer sous la commande **SELECT**.

- /CASEWISE** Produit un plot "cas par cas" des résidus basé sur les sous-commandes choisis parmi les suivants.
- defaults** **outliers(3), plot(zresid), dependent, pred, resid.**
 - outliers()** Inclut seulement les cas pour lesquels la valeur standardisé de la variables est en excès de la valeur entre parenthèses.
 - all** Inclut tous les cas dans le graphique.
 - plot()** Inclut les valeurs standardisés des variables temporaires nommées entre parenthèses.
 - liste des vars** Montre les valeurs des variables pour chaque cas à coté de son point dans le graphique
- /SCATTERPLOT** Produit un graphique des variables accouplées et contrôler les dimensions du graphique.
- (,)** Plote les deux variables spécifiées entre parenthèses est séparées par une virgule.
 - size()** Spécifie la dimension du graphique entre parenthèses: **small** (petit) et **large** (grand).
- /PARTIALPLOT** Produit un graphique de la variable dépendente et l'une des variables indépendentes après avoir régressé ces deux variables sur la reste des variables indépendantes.
- liste des vars** Liste les variables à produire en graphique.
 - size()** Spécifie la dimension du graphique entre parenthèses: **small** (petit) et **large** (grand).
- /SAVE** Sauve au fichier actif des variables temporaires créés au cour de l'analyse de régression.