

MINISTERE DE L'AGRICULTURE DE
L'HYDRAULIQUE ET DES
RESSOURCES HALIEUTIQUES

SECRETARIAT GENERAL

DIRECTION GENERALE DES
PREVISIONS ET DES STATISTIQUES
AGRICOLES

DIRECTION DES STATISTIQUES
AGRICOLES (DSA)

BURKINA FASO
Unité – Progrès – Justice

Tél: 50 – 32 – 45 – 79/80
Fax: 50 – 30 – 54 – 86
E-mail : agristat@ fasonet.bf



RECENSEMENT GENERAL DE L'AGRICULTURE RGA 2006-2010

Protocole d'apurement du volet agricole du RGPH

Document technique N° 04

Mars 2007

Avec l'appui de :



SHER Ingénieurs-Conseils s.a.

Consultants : Pascal Delorme
Tiral Sidi

Sommaire

Introduction	3
1. Première étape : exhaustivité.....	3
1.1 Identifiant : valeurs manquantes.....	3
1.2 Identifiant : codes erronés	4
2. Deuxième étape : Correction des codes et valeurs erronés des autres variables.....	6
2.1 Variables nom, sexe, age, organisation paysanne.....	6
2.2 Variables concernant les activités	7
2.3 Autres variables.....	8
3. Troisième étape : Variables d'intérêt	8
4. Quatrième étape : Constitution du fichier des exploitations agricoles	9
NB : Ces différentes tâches peuvent être automatisées par un code SPSS global. 6. Annexes	12
6. Annexes	13
Annexe 1 : Schéma d'apurement.....	13
Annexe 2 : code SPSS d'apurement.....	14

Introduction

Les contrôles lors de la saisie ne permettent pas de détecter et de corriger toutes les erreurs susceptibles d'entacher les données collectées et saisies. On instaure donc un dernier niveau de contrôle qui prend en compte :

- Les types d'incohérences qui ne sont pas détectées par le programme de saisie.
- Les cas d'enregistrement manquants (non-réponses)
- Le traitement des données manquantes (non-détermination)

Le principal objectif en ce qui concerne le volet agricole du RGPH est la création d'un fichier des exploitations agricoles.

Quatre étapes principales seront passées en revue.

1. Première étape : exhaustivité

Avant d'examiner les variables du fichier, il faut :

- vérifier l'exhaustivité des villages ;
- inclure les hameaux de culture et les quartiers dans leurs villages de rattachement ;
- traiter les doublons (correction ou suppression).

L'apurement du fichier doit d'abord commencer par l'identifiant avant les autres variables d'intérêt.

1.1 *Identifiant : valeurs manquantes*

Déceler les valeurs manquantes de tous les champs composant l'identifiant. Il s'agit de :

- Région
- Province
- Commune/arrondissement
- Village/secteur
- ZC/ZD
- Concession dans la ZD
- Ménage dans la concession
- Nom du CM
- Prénoms du CM

Pour cela, il suffit de faire un tri ascendant du fichier sur chacune de ces variables. Les valeurs manquantes occuperont ainsi le haut du fichier. Ces valeurs manquantes se présentent sous deux formes.

Exemple variable CONCESS

localite	zc_zd	concess
rural		
rural		99
rural		58
rural		46
rural		37

Exemple variable ZC/ZD

Il suffira de noter ou d'imprimer cette partie et de vérifier sur le questionnaire si cette valeur est réellement manquante. Deux cas de figures peuvent se présenter : soit que la valeur est réellement manquante, soit qu'il s'agit d'une erreur de saisie. Le tableau suivant permet de noter ces informations.

N° de l'enregistrement après tri	Est-ce que la valeur est manquante dans le questionnaire ?	
	Si oui, mettre une croix ici	Si non, inscrire la vraie valeur ici

Ce tableau qui constitue un rapport sur la situation réelle des différentes variables qui composent l'identifiant, sera utilisé pour corriger le fichier de chaque province.

1.2 Identifiant : codes erronés

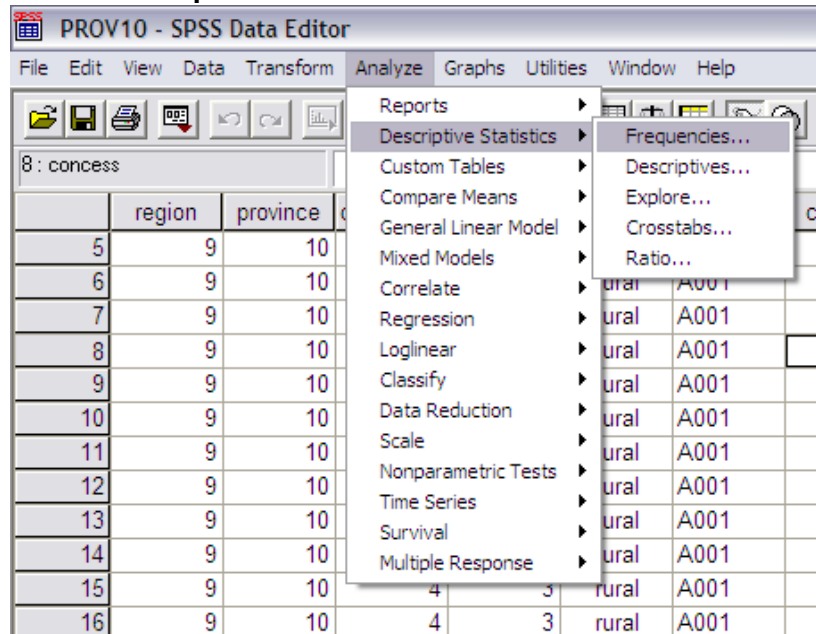
Après cette étape de vérification-corrrection sur le questionnaire, il faut utiliser la commande FREQUENCIES pour déceler éventuellement les codes erronés des différentes variables. Deux méthodes :

- méthode interactive ;
- exécution d'un code SPSS.

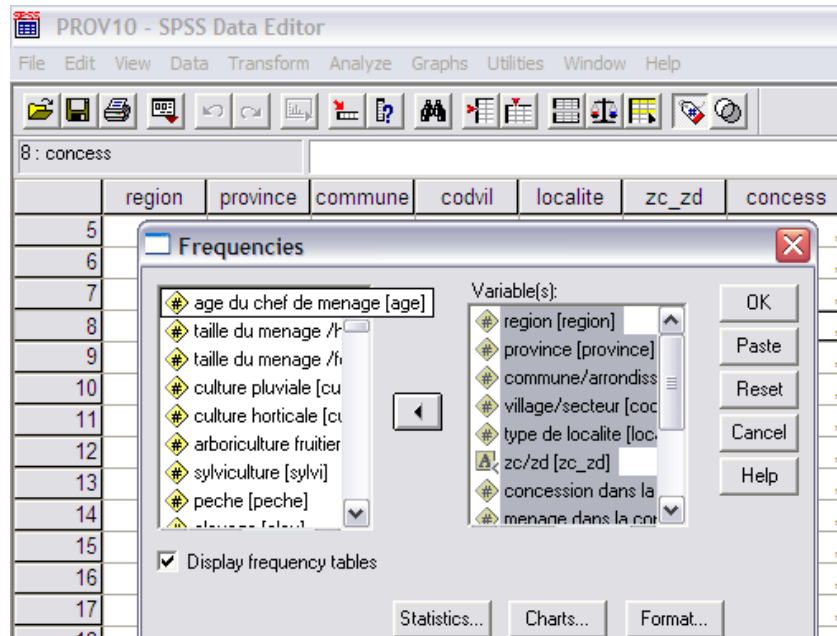
La plus simple est la méthode interactive qui permet d'ailleurs de générer en même temps le code SPSS correspondant.

Méthode interactive : écrans successifs

Choix de l'option FREQUENCIES



Choix des variables



Résultats

Output1 - SPSS Viewer

File Edit View Insert Format Analyze Graphs Utilities Window Help

→ **Frequencies**

Statistics

		region	province	commune/arrondissement	village/secteur	type de localite	zc/zd	concession dans la zd	menage dans la concession	nom du chef de village	prenoms du chef de menage
N	Valid	142998	142999	142999	142998	142998	142999	142122	142068	142999	142999
	Missing	1	0	0	1	1	0	877	931	0	0

Frequency Table

region

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	1	,0	,0	,0
	9	142997	100,0	100,0	100,0
	Total	142998	100,0	100,0	
Missing	System	1	,0		
	Total	142999	100,0		

province

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	10	142999	100,0	100,0	100,0

2. Deuxième étape : Correction des codes et valeurs erronés des autres variables

Le nom du fichier obtenu après l'étape 1 sera de la forme **pxx.sav** (**p** pour province, **xx** le code de la province **.sav** extension SPSS d'un fichier de données). Le renommer **pnpxx.sav** (**p** et **n** : prénom et nom du responsable de l'apurement du fichier en question)

Exemple : **oyp01.sav** (**oy** : Ouambi Yameogo, **p01** : province 01(Bam))

NB : Travailler uniquement sur le fichier **pnpxx.sav** et ne pas toucher à **pxx.sav**.

2.1 Variables nom, sexe, age, organisation paysanne

Variables Nom et prénoms :

- Trier le fichier **pnpxx.sav** sur le nom (respectivement prénom) du CM
- effacer toutes les valeurs composées de chiffres seulement ou de caractères illisibles

NB : Effacer la valeur de la variable et non l'enregistrement qui la contient.

Variable Sexe :

- Trier le fichier **pnpxx.sav** sur le sexe
- Identifier les codes erronés qui peuvent être 0, 9, etc.)

- Faire un rapprochement avec le prénom du CM et mettre le code approprié s'il n'y a aucune ambiguïté (1 pour homme et 2 pour femme). En cas de doute et quand le prénom est manquant un retour au questionnaire est nécessaire.

Variable mois : variable à ne pas traiter, car intérêt très limité

Variable année de naissance :

- Recoder toutes les cases complètement vides par **SYMIS**
- Recoder toutes les années inférieures à 1900 et celles supérieures à 2006 par des **SYMIS**

Variable age : Recoder 99 en **SYMIS**.

Variable calculée agecal : A partir des variables *anneenai* (année de naissance) et *age* , il faut créer une nouvelle variable *agecal* à l'aide du code suivant :

```
IF (sysmis(anneenai)=0 and anneenai >=1900 and anneenai<=2006 and sysmis(age)=1)
agecal = 2006-anneenai .
EXECUTE .
IF (sysmis(anneenai)=0 and sysmis(age)=0) agecal = age .
EXECUTE .
IF (sysmis(anneenai)=1) agecal = age .
EXECUTE .
```

Variables *nbrehomm* (taille du ménage/homme) et *nbrefemm* (taille du ménage/femme) : Les valeurs 0 seront recodées en **SYMIS**.

Variable organp (organisation paysanne) :

- Trier le fichier **pnpxx.sav** sur cette variable
- Identifier les codes erronés ie tous ceux différents de 1, 2 et **SYMIS** ;
- transformer ces codes erronés en **SYMIS** s'ils ne proviennent pas d'une erreur de saisie (à vérifier)

La partie B6 qui récapitule la partie C ne sera pas non plus traitée

2.2 Variables concernant les activités

Exemple : arboriculture fruitière

Chercher les ménages supposés ne pas pratiquer l'arboriculture fruitière (code 2) et qui ont des arbres fruitiers ou ceux qui n'ont pas d'arbres fruitiers et qui déclarent pratiquer l'arboriculture fruitière (code 1). Pour cela il suffira de croiser la variable pratique avec la variable nombre d'arbres fruitiers.

NB : Ce type de test d'incohérence sera à faire pour toutes les pratiques (cultures pluviales, irriguées, arboriculture fruitière, élevage, pêche).

Il faut noter que cette incohérence peut provenir de la codification des agents recenseurs et qu'il est probable donc que si l'agent de saisie voit sur la fiche que par exemple le ménage pratique la culture du mil alors qu'il a répondu non à la question : « *pratiquez-vous des cultures pluviale ?* », de saisir l'information «oui» pour le mil et de corriger et de transformer le « non » de la question « *pratiquez-vous des cultures pluviale ?* », en « oui ».

Les informations sur les types des cultures, d'arbres, etc. seront à privilégier. A cet effet, la procédure automatique simple suivante peut corriger ces incohérences. Avant de l'appliquer, il serait important de vérifier ne serait-ce que sur un petit échantillon des fiches concernées.

Procédure à traduire en code SPSS

*Si pratique = non et une au moins des cultures=oui, alors,
Faire pratique = oui ;
Fin.*

*Si pratique = oui et aucune culture=oui, alors,
Faire pratique = non ;
Fin.*

Cette procédure qui concerne les cultures pluviales, sera adaptée pour chaque type d'activité.

2.3 Autres variables

Le même schéma que celui du § 2.2 sera adopté pour les variables :

- *Variable equipa* (possession d'équipement agricole) :
- *Variable possed* (possède animaux de trait) :

On obtient un fichier ***pnpxcor.sav*** qui sera soumis à la troisième et quatrième étape.

3. Troisième étape : Variables d'intérêt

Les variables concernées sont : le nombre d'arbres fruitiers par espèce et le nombre d'animaux. Ces variables seront soumises à deux types de vérification :

- la recherche des valeurs manquantes
- la recherche des valeurs extrêmes ou atypiques

Valeurs manquantes

Il suffira de conduire la procédure proposée dans le cas des variables de l'identifiant. L'idée étant toujours de vérifier que la valeur manquante est réellement manquante ou bien que c'est une erreur de saisie auquel cas la correction sera faite.

Valeurs atypiques

La méthode interactive de l'option ***FREQUENCIES*** (voir ci-dessus) permet de déceler rapidement les valeurs douteuses. Ce caractère douteux ne dit pas systématiquement qu'elles sont fausses. Il faudra vérifier par un retour au questionnaire.

Des tests complexes permettent de déceler des observations dites hors rang. Le cas classique est de rechercher les observations hors de la fourchette de la « moyenne \pm 2 écarts types » pour le contrôle par rapport à une variable quantitative. Dans ce cas il est évident qu'on ne s'intéresse qu'aux variables fondamentales dites variables d'intérêt.

Les observations hors rang sont étudiées de prêt pour essayer de comprendre leur raison d'être : il s'agit soit d'erreurs (d'observation ou de saisie) auquel cas elles méritent les corrections appropriées, soit de l'expression d'une réalité particulière demandant un traitement spécifique.

Pour les variables manquantes l'imputation se fera de la façon suivante :

- par la médiane des plus proches voisins en ce qui concerne les variables dichotomiques ; si la valeur trouvée n'est pas un nombre entier, la décision proposée consiste à prendre l'arrondi de la valeur trouvée ;
- par la moyenne ou la médiane en ce qui concerne les variables quantitatives ou la moyenne (nombre d'arbres fruitiers, d'animaux, âge, nombre de matériels, etc.).

NB : il est important que cette imputation se fasse dans une logique ascendante. En d'autres termes, il faut faire l'imputation des variables des sous-questions avant de remonter aux variables des questions correspondantes.

Exemple : Pratique des cultures pluviales

Dans ce cas, corriger d'abord les valeurs manquantes des différents types de cultures avant de corriger conséquemment la pratique de la culture.

4. Quatrième étape : Constitution du fichier des exploitations agricoles

La variable qui sera utilisée est *explag* (Exploitation agricole d'appartenance)

Cette variable est très importante ; le fichier des exploitations agricoles sera constitué à partir de cette variable par niveau géographique souhaité.

Le fichier ***pnpxcor.sav*** sera soumis aux sous-étapes suivantes :

1. Créer une variable *nomexpl* (nom du CE) et une variable *prenexpl* (prénom du CE) à l'aide du code suivant :

```
IF (expl=menage) nomexpl = nomc .  
EXECUTE.  
IF (expl=menage) prenexpl = prenc .  
EXECUTE.
```

2. Trier le fichier par la variable *explag* (appartenance à une exploitation agricole)
3. Noter le nombre de valeurs manquantes après en avoir vérifié quelques-unes. Elles correspondent au nombre de ménages dont on ne sait pas s'ils font partie ou pas d'une exploitation agricole.

NB : Il faut toujours trier le fichier sur les variables d'agrégation avant d'utiliser la fonction **AGGREGATE**.

4. Agréger le fichier ***npexpl-pxx.sav*** à l'aide de la fonction **AGGREGATE** sur :

- Région
- Province
- Commune/arrondissement
- ZC/ZD
- village
- Concession dans la ZD
- Exploitation agricole d'appartenance

Renommer la variable par défaut *N_BREAK* par *nbmen* (nombre de ménages)

Pour toutes les autres, l'agrégation se fera en fonction leur type quantitatif ou qualitatif.

Variables quantitatives : il suffira de faire la somme des valeurs de la variable pour tous les ménages de l'exploitation considérée. Il s'agit de :

- nbrehomm (taille du menage /hommes)
- nbrefemm (taille du menage /femmes)
- nombre d'arbres (manguiers, bananiers, goyaviers, etc.)
- nombre d'animaux (bovins, ovins, caprins, etc.)
- nombre d'animaux de trait
- nombre matériels fonctionnels (charrues, tracteurs, etc.)

Variables qualitatives : il faut d'abord, s'assurer qu'il n'y a pas de codes erronés (en principe la deuxième étape a permis de corriger les codes erronés).

Un premier groupe de ces variables est dichotomiques (oui/non) codés (1/2). Pour ce groupe de variables, le « *oui* » est prépondérant (absorbant) c'est-à-dire que s'il y a **un seul** ménage qui répond « *oui* » sur l'ensemble des ménages d'une exploitation agricole, alors le « *oui* » (ou 1) sera attribuée à l'exploitation concernant ladite variable. Et pour que le « *non* » soit attribué à une exploitation, il faut que **tous** les ménages répondent « *non* ».

Exemple : considérons une exploitation agricole composée de 3 ménages et prenons le cas de l'arboriculture fruitière.

Plusieurs cas de figures :

- les 3 ménages pratiquent l'activité → l'exploitation pratique
- 2 ménages la pratiquent et 1 ne la pratique pas → l'exploitation pratique
- 1 seul ménage pratique et 2 ne pratiquent pas → l'exploitation pratique
- Les 3 ménages ne pratiquent pas → l'exploitation ne pratique pas

La traduction mathématique (logique) est :

n le nombre de ménages d'une exploitation

RM_i la réponse du ménage *i* (oui/non=1/2)

RE la réponse de l'exploitation (oui/non=1/2)

On aura donc :

$$RE = \min(RM_1, RM_2, \dots, RM_i, \dots, RM_n)$$

Le deuxième groupe de variable est polytomique c'est-à-dire des variables qui ont plus de deux modalités. La seule variable qui appartient à ce groupe est le **type d'élevage** (1=traditionnel, 2=transhumant, 3=intensif)

Pour ce type de variable, la dichotomisation est une solution. Il s'agit de créer une variable dichotomique par modalité de la façon suivante :

- La variable *tradi* (pour traditionnel) : (oui (codé 1) si le ménage pratique ce type d'élevage, non (codé 0) si le ménage ne pratique pas ce type) ;
- La variable *transhu* (pour transhumance) : (oui (codé 1) si le ménage pratique ce type d'élevage, non (codé 0) si le ménage ne pratique pas ce type) ;
- La variable *intens* (pour intensif) : (oui (codé 1) si le ménage pratique ce type d'élevage, non (codé 0) si le ménage ne pratique pas ce type).

On obtient le fichier ***npexpl-pxx.sav*** des exploitations agricoles.

Le nombre d'exploitations agricoles et de ménages pour un niveau géographique donnée se fera en utilisant la fonction AGREGATE jusqu'au niveau souhaité.

1. **Fichier concession** : Agréger le fichier ***npexpl-pxx.sav*** à l'aide de la fonction AGGREGATE sur :

- Région
- Province
- Commune/arrondissement
- ZC/ZD
- village
- Concession dans la ZD

Faire la somme de *nbmen* et renommer N_BREAK par *nbexpl*.

On obtient un fichier ***npexp-conc-pxx.sav*** qui donne le nombre de ménages et le nombre d'exploitations par concession

2. **Fichier village** : Agréger le fichier ***npexp-conc-pxx.sav*** à l'aide de la fonction AGGREGATE sur :

- Région
- Province
- Commune/arrondissement
- Village

Faire la somme de *nbmen* et *nbexpl*

On obtient un fichier ***npexpl-vill-pxx.sav*** qui donne le nombre de ménages et le nombre d'exploitations par village.

NB : Vérifier la taille des villages en terme de ménages agricoles, identifier celles qui semblent trop petites ou trop grandes et noter les villages concernés.

3. **Fichier ZD** : Agréger le fichier ***npexp-conc-pxx.sav*** à l'aide de la fonction AGGREGATE sur :

- Région
- Province
- Commune/arrondissement
- ZC/ZD

Faire la somme de *nbmen* et *nbexpl*

On obtient un fichier ***npexpl-zd-pxx.sav*** qui donne le nombre de ménages et le nombre d'exploitations par zd

4. **Fichier commune** : Agréger le fichier ***npexpl-vill-pxx.sav*** à l'aide de la fonction AGGREGATE sur :

- Région
- Province
- Commune/arrondissement

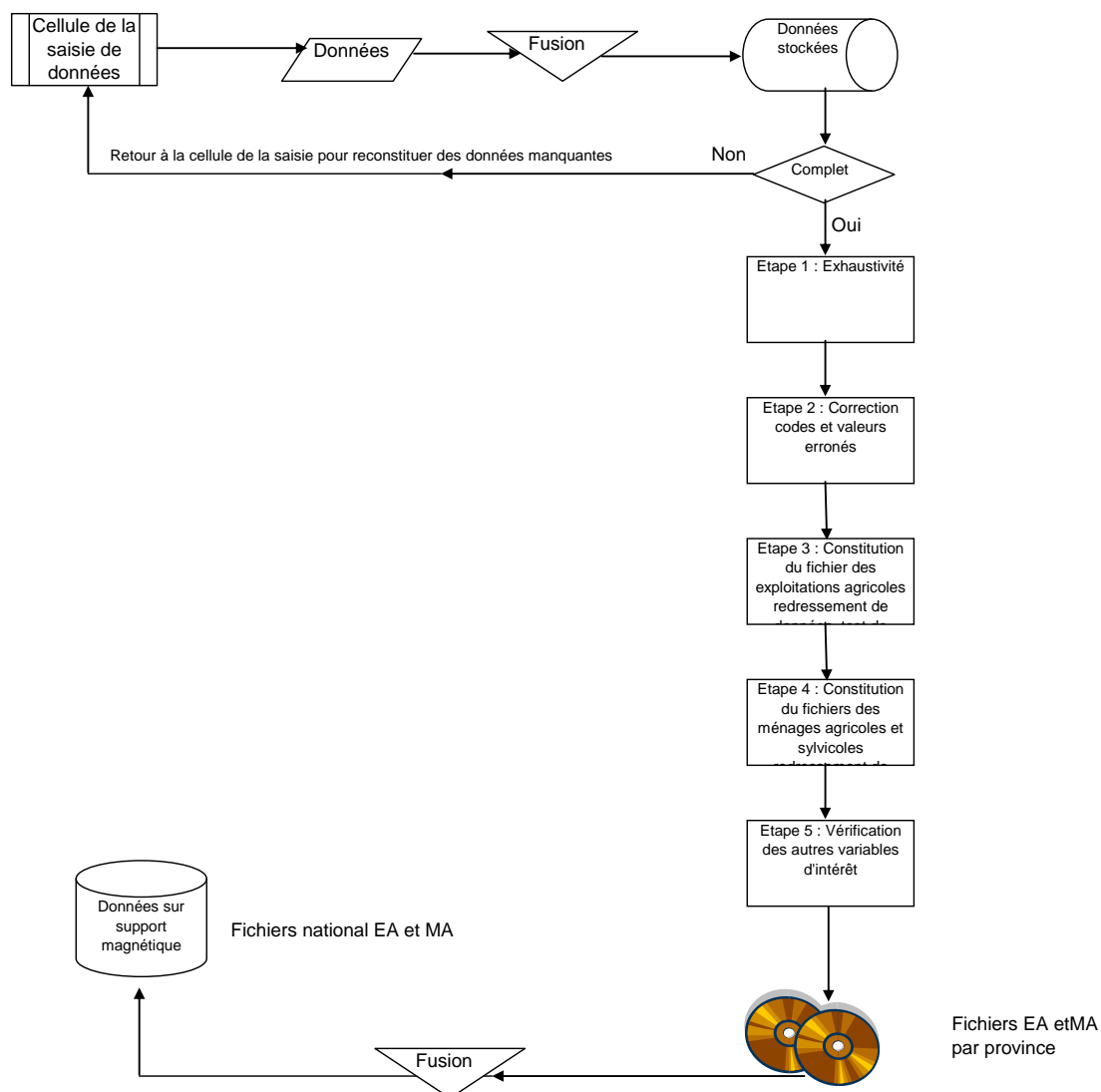
Faire la somme de *nbmen* et *nbexpl*

On obtient un fichier ***npexpl-comm-pxx.sav*** qui donne le nombre de ménages et le nombre d'exploitations par commune

NB : Ces différentes tâches peuvent être automatisées par un code SPSS global.

6. Annexes

Annexe 1 : Schéma d'apurement



Etape	Contenu	Responsables	Appui
Etape 1	Saisie et exhaustivité	KABORE Bernadette	Tiral Sidi (Consultant)
Etape 2	Correction codes et valeurs erronés	OUEDRAOGO Abdoulaye	
Etape 3	Constitution du fichier des exploitations agricoles	LOADA Martin	
Etape 4	Constitution du fichier des ménages agricoles	TAOUYANDE Maurice	
Etape 5	Variables d'intérêt	YAMEOGO Ouambi	

Annexe 2 : code SPSS d'apurement

*PROGRAMME DE TRAITEMENT DU MODULE AGRICOLE DU RGPH.

*COMMENCEMENT DE LA PROCEDURE AUTOMATIQUE.

Set workspace 100000.

Save outfile='C:\TAONDYANDE\Module agricole du RGPH\mtP17.sav'.

*DEUXIEME ETAPE : CORRECTION DES CODES ET VALEURS ERRONEES DES AUTRES VARIABLES.

*Variable Age du chef de ménage.

RECODE ann (Lowest thru 1899=SYSMIS) (2007 thru Highest=SYSMIS).

Recode age (99=sysmis).

IF (sysmis(ann)=0 & sysmis(age)=0) agecal=2006-ann.

IF (sysmis(ann)=0 & sysmis(age)=1) agecal=2006-ann.

IF (sysmis(ann)=1) agecal=age.

Compute agec=agecal.

DO IF (sysmis(age)=0 & sysmis(agecal)=0).

IF (agecal > age) agec=agecal.

IF (agecal <= age) agec=age.

END IF.

VARIABLE LABELS agec 'Age corrigé du chef de ménage'.

**TRAITEMENT DES MODALITES FANTOMES DES VARIABLES DICHOTOMIQUES ET DES VARIABLES TYPE D'ELEVAGE.

*Faire un tri à plat de toutes les autres variables dichotomiques pour mesurer l'ampleur des modalités fantômes et des valeurs manquantes.

FREQ cultp culth arbof sylvi peche elev equipa pratcul mil1 mil2 mais1 mais2 soja1 soja2 riz1 riz2 fonio1

fonio2 coton1 coton2 arach1 arach2 sesam1 sesam2 nieb1 nieb2 voand1 voand2

igna1 igna2 pat1 pat2 manio1 manio2 sorgbl1 sorgbl2 sorgr1 sorgr2 souch1

souch2 fabira1 fabira2 hortp fleur1 fleur2 oign1 oign2 auber1 auber2 gomb1

gomb2 piment1 piment2 fraise1 fraise2 tabac1 tabac2 melon1 melon2 pasteq1

pasteq2 autre1 autre2 pratarb pratsyl amandk grain tamarin boisv pepin pratp

poiss grenouil pratel possed possma boety mouty antyp chevt chaty porct poulest

pintat canardt dindont pigeont autrety.

EXECUTE.

RECODE cultp culth arbof sylvi peche elev equipa pratcul mil1 mil2 mais1 mais2 soja1 soja2 riz1 riz2 fonio1

fonio2 coton1 coton2 arach1 arach2 sesam1 sesam2 nieb1 nieb2 voand1 voand2

igna1 igna2 pat1 pat2 manio1 manio2 sorgbl1 sorgbl2 sorgr1 sorgr2 souch1

souch2 fabira1 fabira2 hortp fleur1 fleur2 oign1 oign2 auber1 auber2 gomb1

gomb2 piment1 piment2 fraise1 fraise2 tabac1 tabac2 melon1 melon2 pasteq1

pasteq2 autre1 autre2 pratarb pratsyl amandk grain tamarin boisv pepin prtp
 poiss grenouil pratel possed possma organp
 (0=sysmis) (3 thru highest=sysmis).
 RECODE boety mouty antyp chevt chaty porct poulest
 pintat canardt dindont pigeont autrety
 (0=sysmis) (4 thru highest=sysmis).

*Variables concernant les activités (correction des incohérences au niveau de chaque activité).

*Cultures pluviales.

DO IF (Min(mil1 to fabira2)=1).
 RECODE prtpcul (2=1) (sysmis=1).
 END IF.
 DO IF (Min(mil1 to fabira2)=2 | (sysmis(mil1)=1 & sysmis(mil2)=1 & sysmis(mais1)=1 &
 sysmis(mais2)=1
 & sysmis(soja1)=1 & sysmis(soja2)=1 & sysmis(riz1)=1 & sysmis(riz2)=1 & sysmis(
 fonio1)=1 &
 sysmis(fonio2)=1 & sysmis(coton1)=1 & sysmis(coton2)=1 & sysmis(arach1)=1 &
 sysmis(arach2)=1
 & sysmis(sesam1)=1 & sysmis(sesam2)=1 & sysmis(nieb1)=1 & sysmis(nieb2)=1 &
 sysmis(voand1)=1
 & sysmis(voand2)=1 & sysmis(igna1)=1 & sysmis(igna2)=1 & sysmis(pat1)=1 &
 sysmis(pat2)=1
 & sysmis(manio1)=1 & sysmis(manio2)=1 & sysmis(sorgbl1)=1 & sysmis(sorgbl2)=1
 & sysmis(sorgr1)=1
 & sysmis(sorgr2)=1 & sysmis(souch1)=1 & sysmis(souch2)=1 & sysmis(fabira1)=1 &
 sysmis(fabira2)=1)).
 RECODE prtpcul (1=2) (sysmis=2).
 RECODE mil1 to fabira2 (sysmis=2).
 END IF.

*Cultures horticoles.

DO IF (Min(fleur1 to autre2)=1).
 RECODE hortp (2=1) (sysmis=1).
 END IF.
 DO IF (Min(fleur1 to autre2)=2 | (sysmis(fleur1)=1 & sysmis(fleur2)=1 & sysmis(oign1)
)=1 & sysmis(oign2)=1
 & sysmis(auber1)=1 & sysmis(auber2)=1 & sysmis(gomb1)=1 & sysmis(gomb2)=1 &
 sysmis(piment1)=1 &
 sysmis(piment2)=1 & sysmis(fraise1)=1 & sysmis(fraise2)=1 & sysmis(tabac1)=1 &
 sysmis(tabac2)=1
 & sysmis(melon1)=1 & sysmis(melon2)=1 & sysmis(pasteq1)=1 & sysmis(pasteq2)
)=1 & sysmis(autre1)=1
 & sysmis(autre2)=1)).
 RECODE hortp (1=2) (sysmis=2).
 RECODE fleur1 to autre2 (sysmis=2).
 END IF.

*Arboriculture fruitière.

RECODE mangue to agrume (9999=sysmis).
 DO IF (SUM(mangue to agrume)>0).
 RECODE pratarb (2=1) (sysmis=1).
 END IF.

```
DO IF (SUM(mangue to agrume) =0 | (sysmis( mangue)=1 &
sysmis( banane )=1 & sysmis( papay )=1 & sysmis( goyav )=1 & sysmis( anaca )=1
& sysmis( datte )=1 & sysmis( anan )=1 & sysmis( agrume )=1)).
    RECODE pratarb (1=2) (sysmis=2).
    RECODE mangue to agrume (sysmis=0).
END IF.
```

*Sylviculture.

```
DO IF (Min(amandk to pepin)=1).
    RECODE pratsyl (2=1) (sysmis=1).
END IF.
DO IF (Min(amandk to pepin)=2 | (sysmis( amandk )=1 & sysmis( grain )=1
& sysmis( tamarin )=1 & sysmis( boisv )=1 & sysmis( pepin)=1)).
    RECODE pratsyl (1=2) (sysmis=2).
    RECODE amandk to pepin (sysmis=2).
END IF.
```

*Pêche.

```
DO IF (Min(poiss to grenouil)=1).
    RECODE pratp (2=1) (sysmis=1).
END IF.
DO IF (Min(poiss to grenouil)=2 | (sysmis( poiss )=1 & sysmis( grenouil )=1)).
    RECODE pratp (1=2) (sysmis=2).
    RECODE poiss to grenouil (sysmis=2).
END IF.
```

*Animaux de trait.

```
RECODE boeu to cham (99=sysmis).
DO IF (SUM(boeu to cham)>0).
    RECODE possed (2=1) (sysmis=1).
END IF.
DO IF (SUM(boeu to cham)=0 | (sysmis( boeu )=1 & sysmis( ane )=1 & sysmis( cheva )=1 &
sysmis( cham )=1)).
    RECODE possed (1=2) (sysmis=2).
    RECODE boeu to cham (sysmis=0).
END IF.
```

*Matériel agricole.

```
RECODE charru to ruches (9999=sysmis).
DO IF (SUM( charru to ruches)>0).
    RECODE possma (2=1) (sysmis=1).
END IF.
DO IF (SUM( charru to ruches)=0 | (sysmis( charru )=1 & sysmis( tracteu )=1 & sysmis(
motop )=1 &
sysmis( charet )=1 & sysmis( ruches)=1)).
    RECODE possma (1=2) (sysmis=2).
    RECODE charru to ruches (sysmis=0).
END IF.
```

*Organisation paysanne.

```
DO IF (min(cultp,culth,arbof,sylvi,peche,elev)=2).
    Recode organp (1=2) (sysmis=2).
END IF.
```

FREQ pratcul hortp pratarb pratsyl pratp possed possma.

*Les variables ci-dessus ne doivent plus comporter des valeurs manquantes
Résolvez le problème avant de continuer au cas elles comportent toujours des valeurs manquantes.

EXECUTE.

Save outfile= 'C:\TAONDYANDE\Module agricole du RGPH\mtP17cor.sav'.

*A ce stade, nous avons un fichier corrigé de toutes les incohérences sauf la partie sur l'élevage qui doit être traitée après l'imputation des valeurs manquantes sur les variables animaux de trait.

Save outfile= 'C:\TAONDYANDE\Module agricole du RGPH\mtP17imp.sav'.

*IMPUTATION DES VALEURS MANQUANTES.

`SORT CASES BY reg (A) prov (A) com (A) local (A)
codvil (A) zc_zd (A) conce (A) men (A) .`

*Sexe du chef de ménage.

`RMV`

`/sexec=MEDIAN(sexe 11).`

`Compute sexec=RND(sexec).`

*Age du chef de ménage.

`RMV`

`/agec=SMEAN(agec).`

*Constitution du fichier des chefs d'exploitation.

*Création des variables nom, prénom, sexe et âge du chef d'exploitation.

`STRING nomexpl (A20).`

`STRING prenexpl (A20).`

`IF (explag=men) nomexpl=nomc.`

`IF (explag=men) prenexpl=prenc.`

`IF (explag=men) sexepl=sexec.`

`IF (explag=men) ageexpl= agec.`

`VARIABLE LABELS nomexpl 'Nom du chef d"exploitation'`

`prenexpl 'Prénom du chef d"exploitation'`

`sexepl 'Sexe du chef d"exploitation'`

`ageexpl 'Age du chef d"exploitation'.`

`Save outfile='C:\TAONDYANDE\Module agricole du RGPH\mtP17imp.sav'.`

`SELECT IF(explag=men).`

`Save outfile='C:\TAONDYANDE\Module agricole du RGPH\mtP17-CE.sav'`

`/Keep= reg prov com local codvil zc_zd conce explag nomc prenc sexec agec.`

`Get file ='C:\TAONDYANDE\Module agricole du RGPH\mtP17imp.sav'.`

*Taille du ménage.

`RMV`

`/nbrehc=SMEAN(nbreh)`

`/nbrefc=SMEAN(nbref).`

*Cultures pluviales.

`RMV`

`/mil1c=MEDIAN(mil1 11) /mil2c=MEDIAN(mil2 11) /mais1c=MEDIAN(mais1 11)`

/mais2c=MEDIAN(mais2 11) /soja1c=MEDIAN(soja1 11) /soja2c=MEDIAN(soja2 11)
 /riz1c=MEDIAN(riz1 11) /riz2c=MEDIAN(riz2 11) /fonio1c=MEDIAN(fonio1 11)
 /fonio2c=MEDIAN(fonio2 11) /coton1c=MEDIAN(coton1 11) /coton2c=MEDIAN(coton2
 11) /arach1c=MEDIAN(arach1 11) /arach2c=MEDIAN(arach2 11)
 /sesam1c=MEDIAN(sesam1 11) /sesam2c=MEDIAN(sesam2 11) /nieb1c=MEDIAN(nieb1
 11) /nieb2c=MEDIAN(nieb2 11) /voand1c=MEDIAN(voand1 11)
 /voand2c=MEDIAN(voand2 11) /igna1c=MEDIAN(igna1 11) /igna2c=MEDIAN(igna2 11)
 /pat1c=MEDIAN(pat1 11) /pat2c=MEDIAN(pat2 11) /manio1c=MEDIAN(manio1 11)
 /manio2c=MEDIAN(manio2 11) /sorgbl1c=MEDIAN(sorgbl1 11)
 /sorgbl2c=MEDIAN(sorgbl2 11) /sorgr1c=MEDIAN(sorgr1 11) /sorgr2c=MEDIAN(sorgr2
 11) /souch1c=MEDIAN(souch1 11) /souch2c=MEDIAN(souch2 11)
 /fabira1c=MEDIAN(fabira1 11) /fabira2c=MEDIAN(fabira2 11).

*Horticulture.

RMV

/fleur1c=MEDIAN(fleur1 11) /fleur2c=MEDIAN(fleur2
 11) /oign1c=MEDIAN(oign1 11) /oign2c=MEDIAN(oign2 11) /auber1c=MEDIAN(auber1
 11) /auber2c=MEDIAN(auber2 11) /gomb1c=MEDIAN(gomb1 11)
 /gomb2c=MEDIAN(gomb2
 11) /piment1c=MEDIAN(piment1 11) /piment2c=MEDIAN(piment2 11)
 /fraise1c=MEDIAN(fraise1 11) /fraise2c=MEDIAN(fraise2 11)
 /tabac1c=MEDIAN(tabac1 11) /tabac2c=MEDIAN(tabac2 11) /melon1c=MEDIAN(melon1
 11) /melon2c=MEDIAN(melon2 11) /pasteq1c=MEDIAN(pasteq1 11)
 /pasteq2c=MEDIAN(pasteq2 11) /autre1c=MEDIAN(autre1 11)
 /autre2c=MEDIAN(autre2 11).

*Arboriculture.

RMV

/manguec=MEDIAN(mangue 11)
 /bananec=MEDIAN(banane 11)
 /papayc=MEDIAN(papay 11)
 /goyavc=MEDIAN(goyav 11)
 /anacac=MEDIAN(anaca 11)
 /dattec=MEDIAN(datte 11)
 /ananc=MEDIAN(anan 11)
 /agrumec=MEDIAN(agrumes 11).

*Sylviculture.

RMV

/amandkc=MEDIAN(amandk 11)
 /grainc=MEDIAN(grain 11)
 /tamarinc=MEDIAN(tamarin 11)
 /boisvc=MEDIAN(boisv 11)
 /pepinc=MEDIAN(pepin 11).

*Pêche.

RMV

/poissc=MEDIAN(poiss 11)
 /grenouic=MEDIAN(grenouil 11).

*Animaux de trait.

RMV

/boeuc=MEDIAN(boeu 11)

```
/anec=MEDIAN(ane 11)
/chevac=MEDIAN(cheva 11)
/chamc=MEDIAN(cham 11).
```

*Elevage.

```
RECODE boeunb moutn annbr chevnb chanb porcn poulesn
pintadn canardn dindonsn pigeonb autresn (999=sysmis).
Recode porct poulest pintat canardt dindont pigeonb (2=sysmis).
```

*Contrôle de cohérence avec la partie sur les animaux de trait.

```
DO IF (sysmis(boeunb)=1 & boeuc>0).
    Compute boeunb=boeuc.
END IF.
DO IF (boeunb>=0 & boeuc>boeunb).
    Compute boeunb=boeuc.
END IF.
```

```
DO IF (sysmis( annbr )=1 & anec >0).
    Compute annbr=anec.
END IF.
DO IF (annbr>=0 & anec>annbr).
    compute annbr=anec.
END IF.
DO IF (sysmis( chevnb )=1 & chevac >0).
    Compute chevnb=chevac.
END IF.
DO IF (chevnb>=0 & chevac>chevnb).
    compute chevnb=chevac.
END IF.
DO IF (sysmis(chanb)=1 & chamc>0).
    Compute chanb=chamc.
END IF.
DO IF (chanb>=0 & chamc>chanb).
    compute chanb=chamc.
END IF.
```

*Correction des incohérences sur le type d'elevage des animaux.

```
DO IF (boeunb=0).
    RECODE boety (lowest thru highest=sysmis).
END IF.
DO IF (moutn=0) .
    RECODE mouty (lowest thru highest=sysmis).
END IF.
DO IF (annbr =0) .
    RECODE antyp (lowest thru highest=sysmis).
END IF.
DO IF (chevnb =0) .
    RECODE chevt (lowest thru highest=sysmis).
END IF.
DO IF (chanb =0) .
    RECODE chaty (lowest thru highest=sysmis).
END IF.
DO IF (porcn=0) .
    RECODE porct (lowest thru highest=sysmis).
```

```

END IF.
DO IF (poulesn =0) .
    RECODE poulest (lowest thru highest=sysmis).
END IF.
DO IF (pintadn =0) .
    RECODE pintat (lowest thru highest=sysmis).
END IF.
DO IF (canardn =0) .
    RECODE canardt (lowest thru highest=sysmis).
END IF.
DO IF (dindonsn =0) .
    RECODE dindont (lowest thru highest=sysmis).
END IF.
DO IF (pigeonb =0) .
    RECODE pigeont (lowest thru highest=sysmis).
END IF.
DO IF (autresn =0) .
    RECODE autrey (lowest thru highest=sysmis).
END IF.

```

*Imputation des valeurs manquantes sur les variables effectif des animaux.
 *L'imputation est faite sous la condition d'existence du type d'élevage.
 *C'est la moyenne au niveau provincial qui est utilisée.

```

AGGREGATE
/OUTFILE='C:\TAONDYANDE\Module agricole du RGPH\mtp17-Moycheptel.sav'
/BREAK=reg prov
/mboeunb = MEAN(boeunb)
/mmoutn = MEAN(moutn)
/mannbr = MEAN(annbr)
/mchevnb = MEAN(chevnb)
/mchanb = MEAN(chanb)
/mporcn = MEAN(porcn)
/mpoulesn = MEAN(poulesn)
/mpintadn = MEAN(pintadn)
/mcanardn = MEAN(canardn)
/mdindons = MEAN(dindonsn)
/mpigeonb = MEAN(pigeonb)
/mautresn = MEAN(autresn).

```

```

MATCH FILES /FILE=*
/TABLE='C:\TAONDYANDE\Module agricole du RGPH\mtp17-Moycheptel.sav'
/BY reg prov.
Compute boeunbc=boeunb.
Compute moutnc=moutn.
Compute annbrc=annbr.
Compute chevnbc=chevnb.
Compute chanbc=chanb.
Compute porcnc=porcn.
Compute poulesnc=poulesn.
Compute pintadnc=pintadn.
Compute canardnc=canardn.
Compute dindonsc=dindonsn.
Compute pigeonbc=pigeonb.
Compute autresnc=autresn.

```

```

IF (sysmis( boety )=0 & sysmis( boeunb)=1) boeunbc=mboeunb.
IF (sysmis( mouty )=0 & sysmis( moutn )=1) moutnc=mmoutn.
IF (sysmis( antyp )=0 & sysmis(annbr)=1) annbrc=mannbr.
IF (sysmis( chevt )=0 & sysmis(chevnb)=1) chevnbc=mchevnb.
IF (sysmis( chaty )=0 & sysmis( chanb )=1) chanbc=mchanb .
IF (sysmis( porct )=0 & sysmis( porcn )=1) porcnc=mporcnc.
IF (sysmis(poulest)=0 & sysmis( poulesn)=1) poulesnc=mpoulesn.
IF (sysmis(pintat)=0 & sysmis( pintadn)=1) pintadnc=mpintadn.
IF (sysmis( canardt)=0 & sysmis( canardn )=1) canardnc=mcanardn.
IF (sysmis(dindont)=0 & sysmis(dindonsn)=1) dindonsc=mdindons.
IF (sysmis( pigeon )=0 & sysmis( pigeonb )=1) pigeonbc=mpigeonb.
IF (sysmis( autrety )=0 & sysmis( autresn)=1) autresnc=mautresn.

```

*Correction des incohérences sur les variables pratique de l'élevage et effectif des animaux.

```

DO IF (SUM(boeunbc, moutnc, annbrc, chevnbc, chanbc, porcnc, poulesnc, pintadnc,
canardnc, dindonsc, pigeonbc, autresnc)>0 ).

```

```

    RECODE pratel (2=1) (sysmis=1).

```

```

END IF.

```

```

DO IF (SUM(boeunbc, moutnc, annbrc, chevnbc, chanbc, porcnc, poulesnc, pintadnc,
canardnc, dindonsc, pigeonbc, autresnc)=0

```

```

| (sysmis( boeunbc )=1 & sysmis(moutnc)=1 & sysmis( annbrc)=1 & sysmis(chevnbc)=1
& sysmis( chanbc )=1 & sysmis(porcnc)=1 & sysmis(poulesnc)=1 & sysmis(pintadnc)=1 &
sysmis(canardnc)=1 &
sysmis(dindonsc)=1 & sysmis(pigeonbc)=1 & sysmis(autresnc)=1)).

```

```

    RECODE pratel (1=2) (sysmis=2).

```

```

    RECODE boeunbc, moutnc, annbrc, chevnbc, chanbc, porcnc, poulesnc, pintadnc,
canardnc, dindonsc, pigeonbc, autresnc (sysmis=0).

```

```

END IF.

```

```

FREQ pratel.

```

*Une valeur manquante sur cette variable doit vous alerter.

```

EXECUTE.

```

*Dichotomisation des variables sur le type d'élevage.

```

Recode boety (1=1) (lowest thru highest=sysmis) into boeutr.

```

```

Recode boety (2=1) (lowest thru highest=sysmis) into boeuth.

```

```

Recode boety (3=1) (lowest thru highest=sysmis) into boeuin.

```

```

Recode mouty (1=1) (lowest thru highest=sysmis) into moutr.

```

```

Recode mouty (2=1) (lowest thru highest=sysmis) into mouth.

```

```

Recode mouty (3=1) (lowest thru highest=sysmis) into moutin.

```

```

Recode antyp (1=1) (lowest thru highest=sysmis) into anntr.

```

```

Recode antyp (2=1) (lowest thru highest=sysmis) into annth.

```

```

Recode antyp (3=1) (lowest thru highest=sysmis) into annin.

```

```

Recode chevt (1=1) (lowest thru highest=sysmis) into chevtr.

```

```

Recode chevt (2=1) (lowest thru highest=sysmis) into chevth.

```

```

Recode chevt (3=1) (lowest thru highest=sysmis) into chevin.

```

```

Recode chaty (1=1) (lowest thru highest=sysmis) into chatr.

```

```

Recode chaty (2=1) (lowest thru highest=sysmis) into chath.

```

```

Recode chaty (3=1) (lowest thru highest=sysmis) into chatin.

```

```

Recode porct (1=1) (lowest thru highest=sysmis) into porctr.

```

```

Recode porct (3=1) (lowest thru highest=sysmis) into porcin.

```

Recode poulest (1=1) (lowest thru highest=sysmis) into pouletr.

Recode poulest (3=1) (lowest thru highest=sysmis) into poulin.

Recode pintat (1=1) (lowest thru highest=sysmis) into pintatr.

Recode pintat (3=1) (lowest thru highest=sysmis) into pintatin.

Recode canardt (1=1) (lowest thru highest=sysmis) into canardtr.

Recode canardt (3=1) (lowest thru highest=sysmis) into canardin.

Recode dindont (1=1) (lowest thru highest=sysmis) into dindontr.

Recode dindont (3=1) (lowest thru highest=sysmis) into dindonin.

Recode pigeont (1=1) (lowest thru highest=sysmis) into pigeontr.

Recode pigeont (3=1) (lowest thru highest=sysmis) into pigeonin.

Recode autrety (1=1) (lowest thru highest=sysmis) into autretr.

Recode autrety (2=1) (lowest thru highest=sysmis) into autreth.

Recode autrety (3=1) (lowest thru highest=sysmis) into autrein.

*Imputation des valeurs manquantes sur le type d'élevage.

*L'imputation se fait sous condition de la présence d'au moins un animal.

*La technique d'imputation utilisée est le mode au niveau village.

AGGREGATE

/OUTFILE='C:\TAONDYANDE\Module agricole du RGPH\mtP17-Modelevag.sav'

/BREAK=reg prov com local codvil

/nboeutr = NU(boeutr) /nboeuth = NU(boeuth) /nboeuin = NU(boeuin) /nmoutr =

NU(moutr) /nmouth = NU(mouth) /nmoutin = NU(moutin) /nanntr = NU(anntr)

/nannth = NU(annth) /nannin = NU(annin) /nchevtr = NU(chevtr) /nchevth =

NU(chevth) /nchevin = NU(chevin) /nchatr = NU(chatr) /nchath = NU(chath)

/nchatin = NU(chatin) /nporctr = NU(porctr) /nporcin = NU(porcin) /npouletr = NU(pouletr)

/npoulin = NU(poulin) /npintatr = NU(pintatr) /npintati =

NU(pintatin) /ncanadtr = NU(canardtr) /ncanadin =

NU(canardin) /ndindotr = NU(dindontr) /ndindoin =

NU(dindonin) /npigeotr = NU(pigeontr) /npigeoin =

NU(pigeonin) /nautretr = NU(autretr) /nautreth = NU(autreth) /nautrein =

NU(autrein).

MATCH FILES /FILE=*

/TABLE='C:\TAONDYANDE\Module agricole du RGPH\mtP17-Modelevag.sav'

/BY reg prov com local codvil.

DO IF (boeunbc>0 & sysmis(boety)=1).

IF (Max(nboeutr,nboeuth,nboeuin)=nboeutr) boeutr=1.

IF (Max(nboeutr,nboeuth,nboeuin)=nboeuth) boeuth=1.

IF (Max(nboeutr,nboeuth,nboeuin)=nboeuin) boeuin=1.

END IF.

DO IF (moutnc>0 & sysmis(mouty)=1).

IF (Max(nmoutr, nmouth, nmoutin)=nmoutr) moutr=1.

IF (Max(nmoutr, nmouth, nmoutin)=nmouth) mouth=1.

IF (Max(nmoutr, nmouth, nmoutin)=nmoutin) moutin=1.

END IF.

DO IF (annbrc>0 & sysmis(antyp)=1).

```

        IF (Max( nannttr, nannth ,nannin )=nannttr)  annttr=1.
        IF (Max(nannttr, nannth ,nannin)=nannttr)  annth=1.
        IF (Max(nannttr, nannth ,nannin)=nannttr)  annin=1.
    END IF.
DO IF ( chevnbc>0 & sysmis( chevt)=1).
    IF (Max(nchevtr, nchevth, nchevin )=nchevtr)  chevtr=1.
    IF (Max(nchevtr, nchevth, nchevin)=nchevth)  chevth=1.
    IF (Max(nchevtr, nchevth, nchevin)=nchevin)  chevin=1.
END IF.
DO IF ( chanbc>0 & sysmis( chaty )=1).
    IF (Max(nchatr, nchath, nchatin )=nchatr)  chatr=1.
    IF (Max(nchatr, nchath, nchatin)=nchath)  chath=1.
    IF (Max(nchatr, nchath, nchatin)=nchatin)  chatin=1.
END IF.
DO IF ( porcnc>0 & sysmis(porct)=1).
    IF (Max(nporctr, nporcin )=nporctr)  porctr=1.
    IF (Max(nporctr,nporcin)=nporcin)  porcine=1.
END IF.
DO IF ( poulesnc>0 & sysmis( poulest)=1).
    IF (Max(npouletr, npoulin)=npouletr)  pouletr=1.
    IF (Max(npouletr, npoulin)=npoulin)  poulin=1.
END IF.
DO IF ( pintadnc>0 & sysmis( pintat)=1).
    IF (Max(npintatr, npintati)=npintatr)  pintatr=1.
    IF (Max(npintatr, npintati)=npintati)  pintatin=1.
END IF.
DO IF ( canardnc>0 & sysmis( canardt )=1).
    IF (Max(ncanadr, ncanadin )=ncanadr)  canardtr=1.
    IF (Max(ncanadr, ncanadin)=ncanadin)  canardin=1.
END IF.
DO IF ( dindonsc>0 & sysmis(dindont)=1).
    IF (Max(ndindotr, ndindoin)=ndindotr)  dindontr =1.
    IF (Max(ndindotr, ndindoin)=ndindoin)  dindonin=1.
END IF.

DO IF ( pigeonbc>0 & sysmis(pigeont)=1).
    IF (Max(npigeotr,npigeoin)=npigeotr)  pigeontr =1.
    IF (Max(npigeotr, npigeoin)=npigeoin)  pigeonin=1.
END IF.
DO IF ( autresnc>0 & sysmis(autrety)=1).
    IF (Max(nautretr, nautreth, nautrein)=nautretr)  autretr =1.
    IF (Max(nautretr, nautreth, nautrein )=nautreth)  autreth =1.
    IF (Max(nautretr, nautreth, nautrein )=nautrein)  autrein=1.
END IF.

DO IF (boeunbc>0).
    RECODE boeutr boeuth boeuin (sysmis=2).
END IF.
DO IF (moutnc>0).
    RECODE moutr mouth moutin (sysmis=2).
END IF.
DO IF (annbrc>0).
    RECODE anntr annth annin (sysmis=2).
END IF.
DO IF (chevnbc>0).

```

```

        RECODE chevtr chevth chevin (sysmis=2).
    END IF.
    DO IF (chanbc>0).
        RECODE chatr chath chatin (sysmis=2).
    END IF.
    DO IF (porcnc>0).
        RECODE porctr porcin (sysmis=2).
    END IF.
    DO IF (poulesnc>0).
        RECODE pouletr poulin (sysmis=2).
    END IF.
    DO IF (pintadnc>0).
        RECODE pintatr pintatin (sysmis=2).
    END IF.
    DO IF (canardnc>0).
        RECODE canardtr canardin (sysmis=2).
    END IF.
    DO IF (dindonsc>0).
        RECODE dindontr dindonin (sysmis=2).
    END IF.
    DO IF (pigeonbc>0).
        RECODE pigeontr pigeonin (sysmis=2).
    END IF.
    DO IF (autresnc>0).
        RECODE autretr autreth autrein (sysmis=2).
    END IF.

```

*Matériels agricoles.

```

RMV
/charruc=MEDIAN(charru 11)
/tracteuc=MEDIAN(tracteu 11)
/motopc=MEDIAN(motop 11)
/charetc=MEDIAN(charet 11)
/ruchesc=MEDIAN(ruches 11).

```

*Organisation paysanne.

```

RMV
/organpc=MEDIAN(organp 11).
compute organpc=RND(organpc).

```

*Mise à jour des variables recapitulatives sur les activités agricoles.

```

Compute cultpc=pratcul.
Compute culthc=hortp.
Compute arbofc=pratarb.
Compute sylvic=pratsyl.
Compute pechec=pratp.
Compute elevc=pratel.
Compute equipac=MIN(possma,possed).

```

*Evaluation de la méthode d'imputation.

```
EXECUTE.
```

*TROISIEME ETAPE: CONSTITUTION DES EXPLOITATIONS AGRICOLES.

*CREATION DU FICHIER DES EXPLOITATIONS AGRICOLES.

Sort cases by reg prov com local codvil zc_zd conce explag.

Save outfile='C:\TAONDYANDE\Module agricole du RGPH\mtP17imp.sav'.

AGGREGATE

/OUTFILE='C:\TAONDYANDE\Module agricole du RGPH\mtP17-expl.sav'

/BREAK=reg prov com local codvil zc_zd conce explag

/nmen=N

/nbreh = SUM(nbrehc)

/nbref = SUM(nbrehc)

/cultp = MIN(cultpc)

/culth = MIN(culthc)

/arbof = MIN(arbofc)

/sylvi = MIN(sylvic)

/peche = MIN(pechec)

/elev = MIN(elevc)

/equipa = MIN(equipac)

/pratcul = MIN(pratcul)

/mil1 =MIN(mil1c)

/mil2 = MIN(mil2c)

/mais1 = MIN(mais1c)

/mais2 = MIN(mais2c)

/soja1 = MIN(soja1c)

/soja2 = MIN(soja2c)

/riz1 = MIN(riz1c)

/riz2 = MIN(riz2c)

/fonio1 = MIN(fonio1c)

/fonio2 = MIN(fonio2c)

/coton1 =MIN(coton1c)

/coton2 = MIN(coton2c)

/arach1 = MIN(arach1c)

/arach2 = MIN(arach2c)

/sesam1 = MIN(sesam1c)

/sesam2 = MIN(sesam2c)

/nieb1 = MIN(nieb1c)

/nieb2 = MIN(nieb2c)

/voand1 = MIN(voand1c)

/voand2 = MIN(voand2c)

/igna1 = MIN(igna1c)

/igna2 = MIN(igna2c)

/pat1 = MIN(pat1c)

/pat2 = MIN(pat2c)

/manio1 = MIN(manio1c)

/manio2 = MIN(manio2c)

/sorgbl1 = MIN(sorgbl1c)

/sorgbl2 = MIN(sorgbl2c)

/sorgr1 = MIN(sorgr1c)

/sorgr2 =MIN(sorgr2c)

/souch1 = MIN(souch1c)

/souch2 = MIN(souch2c)

/fabira1 = MIN(fabira1c)

/fabira2 = MIN(fabira2c)

/hortp = MIN(hortp)

/fleur1 = MIN(fleur1c)
 /fleur2 = MIN(fleur2c)
 /oign1c = MIN(oign1c)
 /oign2c = MIN(oign2c)
 /auber1 = MIN(auber1c)
 /auber2 = MIN(auber2c)
 /gomb1 = MIN(gomb1c)
 /gomb2 = MIN(gomb2c)
 /piment1 = MIN(piment1c)
 /piment2 = MIN(piment2c)
 /fraise1 = MIN(fraise1c)
 /fraise2 = MIN(fraise2c)
 /tabac1 = MIN(tabac1c)
 /tabac2 = MIN(tabac2c)
 /melon1 = MIN(melon1c)
 /melon2 = MIN(melon2c)
 /pasteq1 = MIN(pasteq1c)
 /pasteq2 = MIN(pasteq2c)
 /autre1 = MIN(autre1c)
 /autre2 = MIN(autre2c)
 /pratarb = MIN(pratarb)
 /mangue = SUM(manguec)
 /banane = SUM(bananec)
 /papay = SUM(papayc)
 /goyav = SUM(goyavc)
 /anaca = SUM(anacac)
 /datte = SUM(dattec)
 /anan = SUM(ananc)
 /agrume = SUM(agrumec)
 /pratsy = MIN(pratsyl)
 /amandk = MIN(amandkc)
 /grain = MIN(grainc)
 /tamarin = MIN(tamarinc)
 /boisv = MIN(boisvc)
 /pepin = MIN(pepinc)
 /pratp = MIN(pratp)
 /poiss = MIN(poissc)
 /grenouil = MIN(grenouic)
 /pratel = MIN(pratel)
 /boeunb = SUM(boeunbc)
 /boeutr = MIN(boeutr)
 /boeuth = MIN(boeuth)
 /boeuin = MIN(boeuin)
 /moutn = SUM(moutnc)
 /moutr = MIN(moutr)
 /mouth = MIN(mouth)
 /moutin = MIN(moutin)
 /annbr = SUM(annbrc)
 /anntr = MIN(anntr)
 /annth = MIN(annth)
 /annin = MIN(annin)
 /chevnb = SUM(chevnbc)
 /chevtr = MIN(chevtr)
 /chevth = MIN(chevth)
 /chevin = MIN(chevin)

```

/chanb = SUM(chanbc)
/chatr = MIN(chatr)
/chath = MIN(chath)
/chatin = MIN(chatin)
/porcn = SUM(porcnc)
/porctr = MIN(porctr)
/porcin = MIN(porcin)
/poulesn = SUM(poulesnc)
/pouletr = MIN(pouletr)
/poulin = MIN(poulin)
/pintadn = SUM(pintadnc)
/pintatr = MIN(pintatr)
/pintatin = MIN(pintatin)
/canardn = SUM(canardnc)
/canardtr = MIN(canardtr)
/canardin = MIN(canardin)
/dindons = SUM(dindonsc)
/dindontr = MIN(dindontr)
/dindonin = MIN(dindonin)
/pigeonb = SUM(pigeonbc)
/pigeontr = MIN(pigeontr)
/pigeonin = MIN(pigeonin)
/autresn = SUM(autresnc)
/autretr = MIN(autretr)
/autreth = MIN(autreth)
/autrein = MIN(autrein)
/possed = MIN(possed)
/boeu = SUM(boeuc)
/ane = SUM(anec)
/cheva = SUM(chevac)
/cham = SUM(chamc)
/possma = MIN(possma)
/charru = SUM(charruc)
/tracteu = SUM(tracteuc)
/motop = SUM(motopc)
/charet = SUM(charetc)
/ruches = SUM(ruchesc)
/organp = MIN(organpc).

```

*IMPORTATION DES CARACTERISTIQUES DEMOGRAPHIQUES DES CHEFS D'EXPLOITATION.

Get file='C:\TAONDYANDE\Module agricole du RGPH\mtP17-CE.sav'.

Sort cases by reg prov com local codvil zc_zd conce explag.

MATCH FILES /FILE=*

/FILE='C:\TAONDYANDE\Module agricole du RGPH\mtP17-expl.sav'

/BY reg prov com local codvil zc_zd conce explag.

Save outfile='C:\TAONDYANDE\Module agricole du RGPH\mtP17-expl.sav'.

EXECUTE.