VB and R codes using Households databases available in the NSI's : A prelude to statistical applied studies

Gane Samb Lo[*]

[1] LERSTAD, Université Gaston Berger, Saint-Louis, Sénégal.
[2] Affiliated to LSTA, Université Pierre et Marie Curie, Paris, France
[3] Associated to African University of Sciences and Technology, AUST, Abuja, Nigeria
1178 Evanston Drive, T3P 0J9, Calgary, Alberta, Canada

Abstract. We describe the main features of the households databases we can find in most of our National Statistics Institute. We provide algorithms aimed at extracting a diversity of variables on which different statistical procedures may be applied. Here, we particularly focus on the scaled income, as a beginning. Associated codes (MS Visual Basic ™ and **R** codes) have been successfully tested and delivered in the text and in a separate file.

Résumé. Nous décrivons les principales caractéristiques des bases de données sur les ménages que nous pouvons trouver dans la plupart de nos Instituts Nationaux de Statistiques (INS). Nous fournissons des algorithmes visant à extraire une diversité de variables quantitatives ou qualitatives sur lesquelles différentes procédures statistiques peuvent être appliquées. Ici, nous traitons entre autres de la variable *revenu* pondérée par l'échelle d'équivalence adulte, pour un début. Des codes correspondants (dans MS Visual Basic ™ et dans **R**) ont été testés avec succès et sont fournis dans le texte et dans un fichier séparé.

Presented by Dr Diakarya Barro, Université Ouaga I Joseph Ki-Zerbo
Corresponding Member of the Editorial Board

[*]Corresponding author : gane-samb.lo@ugb.edu.sn, gslo@aust.edu.ng

1. Introduction

In this paper, we want to share our experience in using some household databases one can find in most of African National Statistical institutes (*NSI*).

In most of the African countries, a number of surveys have been and continue to be conducted at a national wide level. The African Statistical Yearbook(2016), provided by the website of the African Center for Statistics *ACS*, uses data from 54 African countries. So, at the continental level, the availability of such households databases amounts to a considerable number of interesting and useful data collections. At the same time, it is very strange and disappointing that only a very few number of research works has been undertaken by African Scholars, at least, scholars in the Mathematics and Statistics Departments.

As mentioned in LO(2014), I witnessed, as editor of *Afrika Statistika* for almost a decade and as a member of a number of Master and PhD theses juries, that most of our colleagues and our students use data sets concerning western countries and picked out from the books to illustrate their theoretical works. I reviewed a very interesting *PhD* thesis, from Central Africa, on Generalized Poisson Laws in which the candidate used data from England, exactly the number rain days in London. In my report, I pointed out such statistics do exist in meteorology stations and I recommended to contact them and to get the data.

The explanations to such an unacceptable situation, which I think is more dramatic in French speaking countries, are contradictory. While in the *NSI*, we may hear that the African university scholars are hardly interested in applied works, and prefer to confine themselves in Theory, far from the real world. From the universities, it is sometimes said that the willingness to undertake applied works is real but the access to the data is not that simple.

Yet, there is a great number of studies using African data in the literature, mostly by international researchers. The works undertaken by the World Bank Living Standards Measurement Study (*LSMS*) are enough to show that.

We simply advocate that all parties to conjugate their efforts to make emerge major research teams and research fields on African data and at the same time, to use extensively local and real data in the every day teaching activities in our universities.

In each country, a National Statistical National System (*NSN*) is set to supervise national data production processes and its use. The universities are usually members of the *NSN* Board and *NSI*'s are the implementing bodies.

There must not be any problem to get the data when the request comes from the official heads of the Universities. This is the way used to get the data on which the current paper is based on.

This paper will be a report of data handling of households survey databases. We expect to be followed by colleagues in our statistics and and mathematics departments. In doing so,

the *AJAS* platform may become, very soon, a place where African data are explored and discussed at the light of most powerful and new theories and statistics trends.

For than a decade, we are working on statistical studies of Welfare issues using the highly elaborated theory of functional empirical processes. But, we used to apply the our findings on real data from Households databases from from Senegal, Mali and Mauritania (See for example : Lo(2008), Lo and al.(2009), Lo(2013), Mergane and LO(2013), Haidara and LO(2008), Haidara and Lo(2013)).

Here, we want to introduce simple methods on handling the databases before undertaking statistical procedures. we hope that this may help and encourage others to acquire the databases and get involved in the study of such treasures. In this paper, we are going to do the following :

(a) Describe common features of these databases.

(b) Show how to extract pertinent information from these databases.

(c) Share useful and replicable *R* and *MS Visual Basic* codes.

We hope that this paper will inspire readers to get similar databases and to use, update or expand the given codes to prepare data and undertake applied statistics studies from them. At the end, sharing those studies in the pages of African Journal of Applied Statistics or elsewhere, may be an efficient way to achieving the awakening we are calling for.

2. Households databases

In the 80's, in the 90's and in the first decades of the 21th century, a number of surveys have been conducted. Nowadays, surveys destined to collect longitudinal data are available. But this paper will be devoted to households surveys with the objectives of showing how researchers may be quickly introduced to be ready use such data.

2.1. Description of Households data

A - Organization of the files.

The households databases (*HDB* in short) gather information collected in a sample of households. In each household, all the member respond to the questions. At the arrival, a *HDB* usually contains three main files.

(1) A person file which contains the responses of individuals of all households. This file contains the disaggregated data.

(2) A household file which contains aggregated data at the household level provided experts of the *NSI*'s.

(3) A meta-file or dictionary which explains the features of each variable.

These files are provided in *MS Excel*™ or in *IBM SPSS*™ in most of the *NSI*'s. When **IBM SPSS**™ is used, the person file is given in an *SPSS*™ fold contained the data and the meta-file.

Figure 1 shows how the data are displayed.

Fig. 1. A view at the data file in a household database

Figure 2 provides information of the variables, which are listed in Column 2 called **Name**. For each variable, the type (categorical, logical, numerical, etc. is precised). For a factor or label variable, the range of the values are given in Column **Values**.

For MS Excel ™ files, a separate dictionary files is given in replacement of the *IBM SPSS* ™ meta-file.

It is not excluded that different software packages or other file formats (like *CSV*) may be used in some *NSI*'s. However, the files organization must be similar and automatic conversions are available.

Fig. 2. A view at the dictionary metafile in a household database

B - Organization of the data.

(a) First of all, the country is organized into geographical regions, districts or areas. The nature of these areas depends on each country. In Senegal, for example, they are the administrative **regions**. From one survey to another, the administration division system may change.

Here, the variable indicating this area in the person file is denoted by **Region**.

(b) Next, each region is divided into a number of districts. The variable factor associated to this district is named **District** or **Milieu**. This variable is re-initiated to one for each region.

(c) In each district, a number of households are included in the survey. The factor variable, in some **HDB**'s, is not re-initiated to zero. Instead, the numbering continues and the last value is the total number of households included in the survey. It is named **Household** or **Menage**

The computer programs we will give handle both situations where the numbering is re-initiated or not.

(d) It is not excluded that a further subdivision of the regions (using clusters for example) is operated. In any case, our programs will be updated in such a case.

(e) In each household, the member are numbered from one to the total size of the household. We do not give a name to this variable. It will we denoted by i in the packages.

The three strata variables *Region*, *District* [and cluster if present] and *Household* will be combined to aggregated the data at the household level.

C - Variables.

Each **HDB** includes a more or less large number of variables, which themselves depend of the objectives of the survey. For example, the variables in the **HDB ESAM 1** or **ESAM 2** of Senegal are quite different from those the *HDB ELIM 1* of Mali.

In future papers, we will be dealing with specific variables.

3. Income or Expense data

In this section, we learn how to create aggregated data at the level of the household from the person data file. Next, we will practice on real databases and provide successfully tested computer programs

A - Aggregation and equivalence scales.

Aggregating the data at the household level is a very important matter. In this paper, we will focus in quantitative analysis. We will have to deal with equivalence scales.

But before we proceed to it, we want to mention that aggregating categorical variables in the available data offers many opportunities to conduct interesting research based on random measures and on symbolic data analysis. For example, the age variable may not only be aggregated into a median or a mean. Instead, it would be interesting to considered it as a distribution, for example, on $D = [0, 100]$, and then to study random distributions. This kind of approach would open fruitful researches in our Statistics and Mathematics departments in relation of the development of very new trends of the theory.

Let us go back to the quantitative aggregation problems, especially concerning income or expense data.

We will not enter into the theoretical details on the adult-equivalence question. We will remain at a simple description level of the question. We want to be able to compare different groups on persons in terms of life quality, when the latter is measured by the income. If we decide to use the income variable as the measure of life quality, the problem is how to compare two groups of people, namely two households, in terms of the income of the members of each group despite of the difference of their structure, the difference of the contexts and perhaps, of the different environments.

To simplify, if we compare two households, each of them reduced to one adult person, certainly their incomes may help. Suppose now that we have two households such that :

(1) the first includes two male adults and a female adult,

(2) and the second is formed by a female single parent and one 10 year old girl. It is not fair to compare the well-being of these households by the total income of each of them.

One has to find a way to convert the individual incomes of the different members of one household to the income of one adult male person. This income is called the adult-equivalent income of the household. Finding the adult-equivalent incomes for all households using the same method would, actually, allow a fair comparison of the household.

A number of methods for providing such adult-equivalent incomes, in the form of scales, are available. We may cite some authors who gave interesting contributions in that topic : Pollack(1979), Bradbury(2003), Duclos and Mercadier-Pats(1996), Hourriez and Olivier(1997) and the references therein. Let h denote some household of size H. An adult-equivalence scale may be defined as an allocation of some weight $w(i,h)$, which is a number between zero and one, to each member of the household h and the income $I(i,h)$ of the $i - th$ member contribute the amounts $w(i,h)I(i,h)$ to the global adult-equivalent income of the households h, which is

$$X(h) = w(1,h)I(1,h) + \cdots + w(H,h)I(H,h).$$

The weight $w(i,h)$ is justified in each proposed scale in very elaborated researches (see already cited papers). The common scales mainly use the gender and the age. Usually, people older than 15 years are considered as adults. If not they are children. We present some common scales.

(a) **Oxford Scale**. This scale allocates the full weight to adult household chief and the weight 0.7 to other adults, and the weight 0.5 to non-adults.

(b) **FAO and OMS scale**. Here, the following allocation is used. Male adults : 1. Female adults : 0.8. Non-adults : 0.5.

(b) **Duclos and Mercadier-Pats (DMP)** scale. Denote by $N_a(h)$ the number of adults and $N_e(h)$ the number of non-adults in the household h and consider two real numbers c and s in $[0,1]$. For any choice of c and s, form

$$E(h) = (N_a(h) + cN_e(h))^s$$

The adult-equivalent income of the household h is the total income of the household divided by $E(h)$. We notice here that the DMP scale cannot be applied directly to the individual, to the contrary of the $Oxford$ and $Fao\text{-}Fam$ scales.

4. Practice

This version will focus on *MS VB*™ codes. We recommend to start by Open the *MS Excel*™ or the *IBM SPSS*™, and for each variable we want to use :

(1) Copy the associated column
(1) Open a text file (with Notepad, Bloc note, etc.)
(1) Paste the copied column
(1) Name the file and save it.

Using this method, we create the files : **region.text** (for the main area division), **milieu.txt**, **cluster.txt**, **household.txt**, **gender.txt**, **age.txt** holding the variables of the same names : region, milieu, cluster, household, gender, age.

We also need the variable **poswrchief** (position with respect to the chief of the household), which indicates the relation of the member with respect to the household chief. Usually, the label one is allocated to the household chief. We store this variable in the file **poswrchief.txt** In the sequel, we will comment the algorithms and computer packages provided in the Packages Section 6.

Mali's ELIM Database.

Step 1. Identification of the household.

In this paper, we explain the methods using the **Mali ELIM 1 database**. The programs we successfully used will be given here in Section 6. All the scripts are gathered in a file you can find in Lo(2016). However, this external resources file will be updated to include other applications.

In some *HDB*'s, the households are numbered for the one to the last one. In some others, the number is re-initialized in each district or cluster. In all cases, the following method works.

In the presence case, it is enough to use an identification in the form :
R+region+M+milieu+C+cluster+H+household.

We use the procedure *householdIdentifier()* provided in **(01) - Identification** in Section 6 to get the variable **identhousehold** restored in the file **identhousehold.txt**.

Since, this is our first use of a procedure or a function, we want to draw the attention of the readers on the following facts.

(a) A procedure or a function is written in a way that it allows the reader to render the program into any software he knows better. Lines beginning by **Rem** are commentaries, in which we explain what do the codes below. So, a reader will be able to adapt these

programs into a language he is acquainted with.

(b) In all the packages presented here, the files are supposed to be in a folder that is specified in Line 7 of the procedure in **(01) - Identification)**. The ready has to pay attention to this, to make sure his programs run correctly.

Step 2. Construction of adult-equivalence scales Functions.

Here, we should be aware of the different presentations of the age variables. Like in Mali, the age of an individual is the real age, for some months (a fraction of year) to maximum age in years. **In ELIM 1**, the code 99 is reserved to an unknown age. This is a clear indication that the survey did not meet an older person that 99 years in Mali during the survey.

In some databases, like in **ESAM 1 of Senegal**, the age is given by classes of five years. For example, the first class $]0, 5[$ may be denoted by 1, $[5, 10[$ by 2, $[10, 14[$ by 3, $[5, 10[$ by 5, etc.

As well, the labels of the gender, male next female, may be $(0, 1)$ or $(1, 2)$.

Rather than constructing directly the equivalence-adult from the *person data file*, we propose functions scale for the **FAO-FAM** and the **Oxford** scales to be used as subroutines. We notice that this is not possible for the DMP scale.

The functions given here, use a discriminatory variable **TypeAge**. If the age is given in years, we replace **TypeAge** by 1. We replace it by 2 if the age is given by classes of 5 year range. If the age is not provided in any of these forms, the functions are set to 0.99, which is an error code.

In the same manner, the variable **TypeGender** is used. If the gender is organized on the basis $(0 = Male, 1 = Female)$, we replace **TypeGender** by 1. We replace it by 2 if the gender is codifies as $(1 = Male, 2 = Female)$. If the gender is not provided in any of these forms, the functions are set to 0.99, which is an error code.

The adult-equivalence scales functions for the *Oxford* scales and the *Fao-Fam* are given Subsection **02 - Scales Oxford, Fao-Fam Functions** in the packages section 6.

We cannot have a *DPM scale* function at the level of individuals, since we have to compute the total number of children and the number of adults in a household. This scale is based on the household while the *Oxford* and the *Fao-Fam* scales are based on individuals.

We will create the file of *DPM* scaling at the level of the households.

Step 3. Construction of adult-equivalence scales files.

(A) To create the *Oxford* scaling file, we proceed from the persons files as follows :

(1) Open the identification files for households (**identhousehold.txt**, created at Step 1) [A], the age file [B], gender file [C]: for reading. Open the Oxford scaling file [D], and the household size file [E], for exporting. Initialize **NBRH=1**. Initialize the household size **HSIZE** = 1.

(2) Read the first line of the files [A], [B] and [C]. Compute the Oxford scale for this individual and initialize the scale value **OxfordScaleCurrent** for this household **HCurrent**.

(3) Continue reading the following lines. As long as the households is the same, increment the scale value **OxfordScaleCurrent** for this household **HCurrent** with the scale value computed at the corresponding line. And increment **HSIZE** by one.

(4) At the first line where the household is no longer **HCurrent** :

   (4.1) The current scale value **OxfordScaleCurrent** is the Oxford scale for that household **HCurrent**. Export the value to the file [D]. Export **HSIZE** to [E].

   (4.2) Give to **HCurrent** the value of household that is read at that line, and initialize its **OxfordScaleCurrent** at the scale value that is computed at that line. Initialize **HSIZE** to one.

   (4.3) Increment **NBRH**.

(5) Repeat steps (3) and (4).

(6) At the end the procedure, **HCurrent** is the last household, **NBRH** is the number of households. **OxfordScaleCurrent** is the scale value of the last household. Export it to [D] and **HSIZE** to [E].

(7) Close all the files.

At the end, we get the *Oxford scale file* for the database. (See the subroutine **createFilesScaleOxford()** in Sub-subsection **03 (a) Oxford Scale File** in in Section 6, for the package)

(B) To create the *Fao-Fam* scaling file, we proceed similarly. But we have to add the file on the status of the individual in the household (**powsrchief**, which is one for the household chief and greater that one for the others).

And we get the *fao-fam scale file*. (See the subroutine **createFilesFaoFam()** in Sub-subsection**03 (b) Oxford Fao-Fam File** in Section 6, for the package)

(C) to create the *DMC* scaling file, we use the following algorithm.

(1) Fix the values c and s (in the example : $s = 0.5$, $c = 0.7$).

(2) Open the identification files for households (**identhousehold.txt**, created at Step 1) [A], the age file [B] : for reading. Open the Oxford scaling file [C], for exporting. Initialize **NBRH=1**.

(3) Read the first line of the files [A] and [B]. Initialize number of children $NBRC = 0$ and the number of adults $NBRA = 0$. If the current individual is a children, increment $NBRC = 0$ by one, otherwise increment $NBRA = 0$ by one. Name the current household by **HCurrent**.

(4) Continue reading the following lines. As long as the households is the same, increment $NBRC = 0$ by one if the current individual is a children, otherwise increment $NBRA = 0$ by one.

(5). At the first line where the household is no longer **HCurrent** :

(5.1) The current values of $NBRC = 0$ and $NBRA = 0$ are respectively the number of Children and the number of adults. Export the value

$$(NBRA + c \times NBRA)^s$$

to [C].

(5.2) Give to **HCurrent** the value of household that is read at that line.

(5.3) Initialize number of children $NBRC = 0$ and the number of adults $NBRA = 0$. If the current individual is a children, increment $NBRC = 0$ by one, otherwise increment $NBRA = 0$ by one.

(6) Repeat steps (4) and (5).

(7) At the end the procedure, **HCurrent** is the last household, **NBRH** is the number of households. Export the value of Point (4.1) to [C].

(8) Close all the files.

At the end, we get the **DMP** scale file file for the database. (See the subroutine **createFilesDMP()** in Sub-subsection **03 (c) DMC Scale File** in Section 6, for the package)

Step 4. Construction of total income or expense files.

First, let us consider the case where the income of any member of the file is available (monthly or yearly). We will use the same algorithm that was used in Step 3. We proceed as follows.

(1) Open the identification files for households (**identhousehold.txt**, created at Step 1) [A], the income file [B]: for reading. Open the **totalincome** file [C], for exporting.

(2) Read the first line of the files [A] and [B]. Initialize the total income of the household to **TotalIncomeCurrentHousehold** to the value read in the file [B].

(3) Continue reading the following lines. As long as the households is the same, increment **TotalIncomeCurrentHousehold** by the value read from the file [B].

(4) At the first line where the household is no longer **HCurrent** :

   (4.1) The current value of **TotalIncomeCurrentHousehold** is the total income for that household **HCurrent**. Export the value in the file [C].

   (4.2) Give to **HCurrent** the value of household that is read at that line, and initialize **TotalIncomeCurrentHousehold** to the value read in the file [B].

(5) Repeat steps (3) and (4).

(6) At the end the procedure, **HCurrent** is the last household. Here, **TotalIncomeCurrentHousehold** is the total income the last household. Export it to [C].

(7) Close all the files.

In implementing this algorithm, we may have different variants. Let us highlight them with real examples.

In the *ELIM1 HDB* of Mali, the only variable giving information on the income amount is the amount earned in the month before the survey. And the variable is an categorical one with values A, B, ..., L corresponding to income ranges $0 − 29.000$, $29.001 − 100.000$, ..., $\geq 3.500.000$. In such a situation we have to transform this variable into a new numerical new. In the study, we created a new variable for which every individual is assigned the mean value of the range income into which he falls. For the last range, we assign the fixed income $3.000.000$ to corresponding individuals. The associated program is below at **(6) Example of categorical income variable : case of ELIM1, Mali**.

If the case of *ESAM 2* of Senegal, the income is not available, but the expense is present and broken into different categories such such as : health, education expenses. In this case, we should add to the algorithm all the expenses variable and use their sums at the place of the one value of the income.

Application to the case *ELIM1* of Mali. Based on the transformed income variable exported in the file **totalincome.txt**, the subroutine in Subsection **(4) Total income File** in Section 6.

Conclusion. Now, we have workable files from which we may do statistics works in the aggregated data. In the example of Mali's *ELIM1* database, we have :

(1) The total income data of the households.

(2) The adult-equivalence files (Oxford and Fao-Fam) for scaling the income. One example of a *DMP* adult-equivalence file is provided for particular cases of $c = 0.5$ and $s = 0.7$. But the package is there to do the same for values of c and s.

(3) The scale income *income/scale* be be used for statistical studies.

Step 4. One we have the scaled income for the households, we might be interested by applying of it many procedures using label data. This is interesting since most of the variables in *HDB*'s are categorical : the area, the level education, the profession, etc. To create any label variable for households based some variable, we proceed as previously.

Let us give two examples.

Algorithm for the Area label file.

(1) Open the identification files for households (**identhousehold.txt**, created at Step 1) [A], the gender file [B]: for reading. Open the **label area** file [C], for exporting.

(2) Read the first line of the files [A] and [B]. Take the read area value as area label of the current household. Export it to [C].

(3) Continue reading the following lines as long as the households is the same.

(4) At the first line where the household is no longer **HCurrent** :

  (4.1) Take the area value at that line as the area label of the current current household. Export it to [C].

  (4.2) Give to **HCurrent** the value of household that is read at that line.

(5) Repeat steps (3) and (4).

(6) Close all the files.

The corresponding subroutine is **createLabelVariableHouseHoldB()**. It is given in Sub-subsection **(5a) Creation of the area label file** in Section 6.

Algorithm for the household chief area label file.

This case is more elaborated since we have we have to combine with the individual status in the household (powsrchief, which is one for the household chief and greater that one for the others).

(1) Open the identification files for households (**identhousehold.txt**, created at Step 1) [A], the area (region) file [B], the **poswrchief** file [C] : for reading. Open the **Chief gender label** file [D], for exporting.

(2) Read the first line of the files [A] and [B]. Initialize **labelchiefgender** to XXX. If the value read from the **poswrchief** file is "1", take the read gender as the gender label of the current household and name it **labelchiefgender**.

(3) Continue reading the following lines. As long as the households is the same :
If the value read from the **poswrchief** file is "1", take the read gender as the gender label of the current household and name it **labelchief gender**. If not, do nothing.

(4) At the first line where the household is no longer **HCurrent** :

  (4.1) Export **labelchiefgender** to file [D].

  (4.2) Give to **HCurrent** the value of household that is read at that line.   (4.3) Initialize **labelchiefgender** to "XXX". If the value read from the **poswrchief** file is "1", take the read gender as the gender label of the current household and name it **labelchief gender**.

(5) Repeat steps (3) and (4).

(6) If **labelchiefgender** is not "XXX", then export it to file [D].

(7) Close all the files.

The corresponding subroutine is **createLabelVariableHouseHoldA()**. It is given in Sub-subsection **(5b) Creation of the Household Chief Gender label file** in Section 6.

The same packages given in **MS VB™**   are written in **R** codes. The only difference is that the files are not created. In the script file, the variables are created and are ready to be used in statistical procedures. The reader may find it as well in Lo(2016).

5. Conclusions

The papers showed how to proceed to handle households databases to extract workable quantitative random variables and label variables on which different statistical procedures may be applied. In the particular case where we are interested by the total income file, different adult-equivalence files and label variables such as the geographical label, the chief

gender label have been created. The packages in Visual Basic™ and in the free R software are provided to serve as models. The descriptions of the algorithms should allow to adapt them to any household database and to arbitrary databases. Their implementation in **Oracle Java**™ , **Python**, **PHP** etc., are easily set up. In the future, we will be able to deliver reports on some more or less complex studies using the available databases. It will be meant that the packages we will use are based on the current ones.

6. Packages

(01) - Identification.

```
Sub householdIdentifier()

Dim region As String, household As String, cluster As String,
Dim milieu As String, folder As String
Dim chain As String, newchain As String

Rem Use the right folder where the file are located
folder = "mali/elim1/"

Rem files in text formats, for getting the data
region = "region.txt"
household = "menage.txt"
cluster = "cluster.txt"
milieu = "milieu.txt"

Rem new file to get the identification
identhousehold = "identhousehold.txt"

Rem open the files : region, household, cluster, milieu, for reading
Open folder \& region For Input As \#1
Open folder \& milieu For Input As \#2
Open folder \& cluster For Input As \#3
Open folder \& household For Input As \#4


Rem open the files : identhousehold.txt, for export
Open folder \& identhousehold For Output As \#5

While Not EOF(1)
    newchain = ""
    Input \#1, chain:    newchain = newchain \& "D" \& chain
    Input \#2, chain:    newchain = newchain \& "M" \& chain
    Input \#3, chain:    newchain = newchain \& "C" \& chain
    Input \#4, chain:    newchain = newchain \& "H" \& chain


    Print \#5, newchain
Wend
Close \#1, \#2, \#3, \#4, \#5
MsgBox "identhousehold.text ready to be used in your folder!"
End Sub
```

02 - Scales Oxford, Fao-Fam Functions.

02 (a) Oxford Scale.

```
Function scaleOxford(ageType As Integer , age As Double , poswtchief As Integer) As Do

Rem ageType (1 for a real age in year), (2 if the age is given in classes of length
Rem The value poswtchief is one for the household chief . The alternate value is not
Rem If one the arguments is out of range , the scale is set to 0.99 , as an error code

If (ageType < 1 Or ageType > 2) Then
     scaleOxford = 0.99
Else
     Select Case ageType
          Case 1:
               If (age < 15) Then
                         scaleOxford = 0.5
               Else
                         If (poswtchief = 1) Then
                         scaleOxford = 1
                         Else
                         scaleOxford = 0.7
                         End If
               End If


          Case 2:

               If (age < 4) Then
                         scaleOxford = 0.5
               Else
                         If (poswtchief = 1) Then
                         scaleOxford = 1
                         Else
                         scaleOxford = 0.7
                         End If
               End If

End Select
End If

End Function
```

02 (b) Fao-FAM Scale.

```
Function scaleFaoFam(ageType As Integer, genderType As Integer, age As Double, gender
Rem ageType (1 for a real age in year), (2 if the age is given in classes of length
Rem genderType (1 for Male=0, female=1), (2 for Male=1, female=2)
Rem If one the argument is out of range, the scale is set to 0.99, as an error code

Dim testContinuation As Boolean

testContinuation = False

If ((ageType < 1) Or (ageType > 2) Or (genderType < 1) Or (genderType > 2)) Then
    scaleFaoFam = 0.99
Else
    Select Case ageType
        Case 1:
            If (age < 15) Then
                scaleFaoFam = 0.5
            Else
                testContinuation = True

            End If

        Case 2:
            If (age < 4) Then
                scaleFaoFam = 0.5
            Else
                testContinuation = True
            End If

End Select

End If

If (testContinuation) Then
            Select Case genderType
                Case 1:
                    If (gender = 0) Then
                        scaleFaoFam = 1
                    Else
                            If (gender = 1) Then
                                scaleFaoFam = 0.8
                            Else
                                scaleFaoFam = 0.99
                            End If
                    End If
```

```
            Case  2:

                  If  (gender = 1) Then
                      scaleFaoFam = 1
                  Else

                            If  (gender = 2) Then
                                scaleFaoFam = 0.8
                            Else
                                scaleFaoFam = 0.99
                            End  If

                  End  If
            End  Select
End  If


End  Function
```

03 - Creating Scales Files.

03 (a) Oxford Scale File.

```
Sub createFilesScaleOxford()

Dim identhouseholdfile As String, agefile As String, genderfile As String, poswrchie
Dim i As Double, j As Double, household As String, age As Double, gender As Integer,
Dim ageType As Integer, genderType As Integer, scaleValue As Double, nbrhouseholds As
Dim folder As String

Rem provide Type of Age / Type of Gender
Rem ageType (1 for a real age in year), (2 if the age is given in classes of length
Rem genderType (1 for Male=0, female=1), (2 for Male=1, female=2)
Rem If one the argument is out of range, the scale is set to 0.99, as an error code
ageType = "1"
genderType = "1"

Rem Test if Type of Age / Type of Gender are correct
If ((ageType < 1) Or (ageType > 2) Or (genderType < 1) Or (genderType > 2)) Then
    MsgBox "Type of Age and/or Type of Gender wrong Fix it Begin again"
    Exit Sub
End If


Rem Use the right folder where the file are located
folder = "C:/Data_gslo/gslo/bdm/bdm/packagevb6/mali/elim1/"


Rem files needed in text formats : identhousehold , age, gender, poswrchief, sizehous
identhouseholdfile = "identhousehold.txt"
agefile = "age.txt"
genderfile = "gender.txt"
poswrchieffile = "poswrchief.txt"

Rem new file to get the identification
scaleoxfordfile = "scaleoxford.txt"
sizehouseholdfile = "sizehousehold.txt"

Rem open the files : region, household, cluster, milieu, for reading
Open folder & identhouseholdfile For Input As #1
Open folder & agefile For Input As #2
Open folder & poswrchieffile For Input As #3
Open folder & genderfile For Input As #6
```

```
'open the files : identhousehold, for export
Open folder & scaleoxfordfile For Output As #4
Open folder & sizehouseholdfile For Output As #5

'rem read the first line and initialize scaleValue
    Input #1, household
    Input #2, agestr
    Input #3, poswrchiefstr
    Input #6, genderstr


    poswrchief = Int(Val(poswrchiefstr))
    age = Val(agestr)

    Rem Catch the current household
    householdCurrent = household

    Rem initialize scale value for the current household
    scaleValue = scaleOxford(ageType, age, poswrchief)


    Rem Initialize the size of the household at j=1
    sizehousehold = 1
    Rem Number of households
    nbrhouseholds = 1

    j = 1


      While Not EOF(1)
      j = j + 1
      Input #1, household
      Input #2, agestr
      Input #3, poswrchiefstr
      Input #6, genderstr
      poswrchief = Int(Val(poswrchiefstr))
      age = Val(agestr)


      If (household = householdCurrent) Then

            Rem As long as the household is the same, we contuinue to increment the s
            scaleValue = scaleValue + scaleOxford(ageType, age, poswrchief)
            sizehousehold = sizehousehold + 1
      Else
```

```
        Rem The first time we did quit the current household, we may save the current
        Rem of scaleValue as the scale of the the the household which is just complet
        Print #4, Trim(Str(scaleValue))
        Print #5, sizehousehold

        Rem Initialize the sclale and the size for the new household
        scaleValue = scaleOxford(ageType, age, poswrchief)
        Rem Initialize the size of the household at j=1
        sizehousehold = 1
        Rem inceremente number of househols
        nbrhouseholds = nbrhouseholds + 1
        householdCurrent = household
    End If

Wend
        Print #4, Trim(Str(scaleValue))
        Print #5, sizehousehold

Close #1, #2, #3, #4, #5, #6
MsgBox newchain & " number of households = " & nbrhouseholds & " Files scaleoxford.tx
End Sub
```

03 (b) Fao-Fam Scale File.


```
Sub createFilesScaleFaoFam()

Dim identhouseholdfile As String, agefile As String, genderfile As String, poswrchie
Dim i As Double, j As Double, household As String, age As Double, gender As Integer,
Dim ageType As Integer, genderType As Integer, scaleValue As Double, nbrhouseholds As
Dim folder As String

Rem provide Type of Age / Type of Gender
Rem ageType (1 for a real age in year), (2 if the age is given in classes of length
Rem genderType (1 for Male=0, female=1), (2 for Male=1, female=2)
Rem If one the argument is out of range, the scale is set to 0.99, as an error code
ageType = "1"
genderType = "2"

Rem Test if Type of Age / Type of Gender are correct
If ((ageType < 1) Or (ageType > 2) Or (genderType < 1) Or (genderType > 2)) Then
    MsgBox "Type of Age and/or Type of Gender wrong Fix it Begin again"
    Exit Sub
End If
```

```
Rem Use the right folder where the file are located
folder = "C:/Data_gslo/gslo/bdm/bdm/packagevb6/mali/elim1/"


Rem files needed in text formats : identhousehold , age, gender , poswrchief , sizehous
identhouseholdfile = "identhousehold.txt"
agefile = "age.txt"
genderfile = "gender.txt"
poswrchieffile = "poswrchief.txt"

Rem new file to get the identification
scalefaofamfile = "scalefaofam.txt"
sizehouseholdfile = "sizehousehold.txt"

Rem open the files : region , household , cluster , milieu , for reading
Open folder & identhouseholdfile For Input As #1
Open folder & agefile For Input As #2
Open folder & poswrchieffile For Input As #3
Open folder & genderfile For Input As #6


'open the files : identhousehold , for export
Open folder & scalefaofamfile For Output As #4
Open folder & sizehouseholdfile For Output As #5

'rem read the first line and initialize scaleValue
    Input #1, household
    Input #2, agestr
    Input #3, poswrchiefstr
    Input #6, genderstr


    poswrchief = Int(Val(poswrchiefstr))
    gender = Int(Val(genderstr))
    age = Val(agestr)

    Rem Catch the current household
    householdCurrent = household

    Rem initialize scale value for the current household
    scaleValue = scaleFaoFam(ageType, gender, age, poswrchief)

    Rem Initialize the size of the household at j=1
    sizehousehold = 1
    Rem Number of households
    nbrhouseholds = 1
```

```
      j = 1


       While Not EOF(1)
       j = j + 1
       Input #1, household
       Input #2, agestr
       Input #3, poswrchiefstr
       Input #6, genderstr

       Rem Transform strings into integers or double
       poswrchief = Int(Val(poswrchiefstr))
       age = Val(agestr)
       gender = Int(Val(genderstr))

       If (household = householdCurrent) Then

             Rem As long as the household is the same, we contuinue to increment the sc
             scaleValue = scaleValue + scaleFaoFam(ageType, gender, age, poswrchief)
             sizehousehold = sizehousehold + 1
      Else

          Rem The first time we did quit the current household, we may save the current
          Rem of scaleValue as the scale of the the the household which is just complet
          Print #4, Trim(Str(scaleValue))
          Print #5, sizehousehold

          Rem Initialize the sclale and the size for the new household
          scaleValue = scaleFaoFam(ageType, gender, age, poswrchief)
          Rem Initialize the size of the household at j=1
          sizehousehold = 1
          Rem inceremente number of househols
          nbrhouseholds = nbrhouseholds + 1
          householdCurrent = household
      End If

Wend

Print #4, Trim(Str(scaleValue))
Print #5, sizehousehold

Close #1, #2, #3, #4, #5, #6
MsgBox j & "/ Number of households = " & nbrhouseholds & " Files scalefaofam.txt and
End Sub
```

03 (c) DMP Scale File.

```
Sub createFilesScaleDMP()

Dim identhouseholdfile As String, agefile As String, genderfile As String, poswrchie
Dim i As Double, j As Double, household As String, age As Double, gender As Integer,
Dim ageType As Integer, genderType As Integer, scaleValue As Double, nbrhouseholds As
Dim folder As String, scaleChildren As Double, scaleAdult As Double, CDPM, SDPM As D

Rem provide Type of Age / Type of Gender
Rem ageType (1 for a real age in year), (2 if the age is given in classes of length o
Rem genderType (1 for Male=0, female=1), (2 for Male=1, female=2)
Rem If one the argument is out of range, the scale is set to 0.99, as an error code
ageType = "1"


Rem Give the values if c and s for the Duclos−Pats−Mercadier
CDPM = 0.5
SDPM = 0.7

Rem Test if Type of Age / Type of Gender are correct
If ((ageType < 1) Or (ageType > 2)) Then
    MsgBox "Type of Age wrong! Fix it Begin again"
    Exit Sub
End If


Rem Use the right folder where the file are located
folder = "C:/Data_gslo/gslo/bdm/bdm/packagevb6/mali/elim1/"


Rem files needed in text formats : identhousehold , age, gender, poswrchief, sizehous
identhouseholdfile = "identhousehold.txt"
agefile = "age.txt"

Rem new file to get the identification
scaleDPMfile = "scaleDPM−" & Trim(Str(CDPM)) & "−" & Trim(Str(SDPM)) & ".txt"
sizehouseholdfile = "sizehousehold.txt"

Rem open the files : region , household , cluster , milieu , for reading
Open folder & identhouseholdfile For Input As #1
Open folder & agefile For Input As #2

MsgBox "lo" & scaleDPMfile
'open the files : identhousehold , for export
```

```
Open folder & scaleDPMfile For Output As #4

'rem read the first line and initialize scaleValue
    Input #1, household
    Input #2, agestr


    age = Val(agestr)

    Rem Catch the current household
    householdCurrent = household

    Rem initialize scale value for the current household
    scaleChildren = 0: scaleAdult = 0
    If (ageType = 1) Then
                If (age < 15) Then scaleChildren = 1 Else scaleAdult = 1
    Else
                If (age < 4) Then scaleChildren = 1 Else scaleAdult = 1
    End If

    Rem Initialize the size of the household at j=1
    sizehousehold = 1
    Rem Number of households
    nbrhouseholds = 1

    j = 1


     While Not EOF(1)
     j = j + 1
     Input #1, household
     Input #2, agestr


     Rem Transform strings into integers or double
     poswrchief = Int(Val(poswrchiefstr))
     age = Val(agestr)
     gender = Int(Val(genderstr))

     If (household = householdCurrent) Then

            Rem As long as the household is the same, we contuinue to increment the sc

             If (ageType = 1) Then
                 If (age < 15) Then scaleChildren = scaleChildren + 1 Else scaleAdult
```

```
                Else
                        If (age < 4) Then scaleChildren = scaleChildren + 1 Else scaleAdult =
                End If

                sizehousehold = sizehousehold + 1
        Else

            Rem The first time we did quit the current household , we may save the current
            Rem of scaleValue as the scale of the the the household which is just complet
            Print #4, Trim(Str((scaleAdult + (CDPM * scaleChildren)) ^ (SDPM)))


            Rem Initialize the sclale and the size for the new household
            scaleChildren = 0: scaleAdult = 0
            If (ageType = 1) Then
                    If (age < 15) Then scaleChildren = 1 Else scaleAdult = 1
            Else
                    If (age < 4) Then scaleChildren = 1 Else scaleAdult = 1
            End If

            Rem Initialize the size of the household at j=1
            sizehousehold = 1
            Rem inceremente number of househols
            nbrhouseholds = nbrhouseholds + 1
            householdCurrent = household
        End If

Wend
Print #4, Trim(Str((scaleAdult + (CDPM * scaleChildren)) ^ (SDPM)))
Close #1, #2, #4
MsgBox "Number of households = " & nbrhouseholds & " Files scalefaofam.txt and sizeho
End Sub
```

(4) Total monthly income file

```
Sub createTotalIncomeMali()

Dim identhouseholdfile As String, household As String
Dim incomestr As String, income As Double, incomefile As String, incometotalfile As S

Rem Use the right folder where the file are located
folder = "C:/Data_gslo/gslo/bdm/bdm/packagevb6/mali/elim1/"


Rem files needed in text formats : identhousehold , monthlyincome
identhouseholdfile = "identhousehold.txt"
incomefile = "monthlyincome.txt"


'open the files : to receive Total Income/Household
incometotalfile = "totalincome.txt"

identhouseholdfile = "identhousehold.txt"
Open folder & identhouseholdfile For Input As #1
Open folder & incomefile For Input As #2
Open folder & incometotalfile For Output As #3

'rem read the first line and initialize scaleValue
    Input #1, household
    Input #2, incomestr

    income = Val(incomestr)

    Rem Catch the current household
    householdCurrent = household

    Rem initialize the value of the income
    incomeHousehold = income


    j = 1


    While Not EOF(1)
    j = j + 1
    Input #1, household
    Input #2, incomestr
    income = Val(incomestr)
```

```
    If (household = householdCurrent) Then

        Rem As long as the household is the same, we contuinue to increment the s

        incomeHousehold = incomeHousehold + income

    Else

        Rem The first time we did quit the current household, we may save the current
        Rem of scaleValue as the scale of the the the household which is just complet
        Print #3, incomeHousehold


        Rem Initialize the sclale and the size for the new household

            householdCurrent = household
            incomeHousehold = income
    End If

Wend

Close #1, #2, #3
MsgBox "Total Incom found and created in incometotal.txt, and ready to be used"
End Sub
```

(5) Example of creation of a label variable.

(5a) Creation of the area label file.

```
Sub createLabelVariableHouseHoldB()

Dim identhouseholdfile As String, household As String
Dim regionfile As String, labelregionfile As String, region As String


Rem Use the right folder where the file are located
folder = "C:/Data_gslo/gslo/bdm/bdm/packagevb6/mali/elim1/"


Rem files needed in text formats : identhousehold , gender, poswrchief
identhouseholdfile = "identhousehold.txt"
regionfile = "region.txt"


'open the files : to receive Total Income/Household
labelregionfile = "labelregion.txt"

Rem Open file for reading
identhouseholdfile = "identhousehold.txt"
Open folder & identhouseholdfile For Input As #1
Open folder & regionfile For Input As #2


Rem Open file for export
Open folder & labelregionfile For Output As #3


'rem read the first line and initialize scaleValue
    Input #1, household
    Input #2, region


    Rem Catch the current household
    householdCurrent = household

    Rem initialize the value of the income
    labelregion = region


    j = 1
```

```
   While  Not  EOF( 1 )
     j = j + 1
   Input  #1,  household
   Input  #2,  region

    If  Not  ( household = householdCurrent )  Then


        Rem The  first  time  we  did  quit  the  current  household ,  we  may  save  the  current
        Rem of  region  as  as  the  region  of  the  household
        Print  #3,  region

           Rem  initilize  the  household
           householdCurrent = household
           End  If

Wend

        Print  #3,  region

Close  #1,  #2,  #3
MsgBox  " Total  Incom  found  and  created  in  incometotal . txt ,  and  ready  to  be  used"
End  Sub
```

(5b) Creation of the Household Chief Gender label file.

```
Sub createLabelVariableHouseHoldA()

Dim identhouseholdfile As String, household As String
Dim genderfile As String, poswrchieffile As String, labelgenderfile As String, labelg
Dim gender As String, poswrchief As String

Rem Use the right folder where the file are located
folder = "C:/Data_gslo/gslo/bdm/bdm/packagevb6/mali/elim1/"


Rem files needed in text formats : identhousehold , gender, poswrchief
identhouseholdfile = "identhousehold.txt"
genderfile = "gender.txt"
poswrchieffile = "poswrchief.txt"


'open the files : to receive Total Income/Household
labelgenderfile = "labelgender.txt"

Rem Open file for reading
identhouseholdfile = "identhousehold.txt"
Open folder & identhouseholdfile For Input As #1
Open folder & genderfile For Input As #2
Open folder & poswrchieffile For Input As #3


Rem Open file for export

Open folder & labelgenderfile For Output As #4


'rem read the first line and initialize scaleValue
    Input #1, household
    Input #2, gender
    Input #3, poswrchief


    Rem Catch the current household
    householdCurrent = household

    Rem initialize the value of the income
    labelgender = gender
```

```
   j = 1

 Rem initialize labelgender to   "XXX"
 labelgender = "XXX"
 Rem test is this individual is household chief. If yes, take the gender as gender
 If Trim(poswrchief) = "1" Then
             labelgender = gender
 End If

While Not EOF(1)
   j = j + 1
 Input #1, household
 Input #2, gender
 Input #3, poswrchief


   If (household = householdCurrent) Then

       Rem test is this individual is household chief. If yes, take the gender as ge
           If Trim(Str(poswrchief)) = "1" Then
               labelgender = gender
           End If

    Else

       Rem The first time we did quit the current household, we may save the current
       Rem of scaleValue as the scale of the the the household which is just complet
       Print #4, labelgender


       Rem Initialize the sclale and the size for the new household

            householdCurrent = household

            Rem initialize labelgender to   "XXX"
            labelgender = "XXX"


               Rem test is this individual is household chief. If yes, take the
               If Trim(Str(poswrchief)) = "1" Then
                 labelgender = gender
                End If
    End If

Wend
```

```
        Print  #4, labelgender

Close #1, #2, #3, #4
MsgBox "gender label created and ready to be used"
End Sub
```

(6) Example of categorical income variable : case of ELIM1, Mali.

```
Sub transforIncomeMali()

Dim j As Double, newchain As String, valeurIncome As Double, incomefile As String, in
Rem Use the right folder where the file are located
folder = "C:/Data_gslo/gslo/bdm/bdm/packagevb6/mali/elim1/"


Rem files needed in text formats : identhousehold , age, gender, poswrchief, sizehous
incomeNTfile = "monthlyincomeNT.txt"
Rem open the files : Income not yet transformg
Open folder & incomeNTfile For Input As #1


Rem files to receive new income
incomefile = "monthlyincome.txt"
Rem open the files : Income not yet transformg
Open folder & incomefile For Output As #2


j = 0
While Not EOF(1)
        Input #1, newchain
        newchain = Trim(newchain)
    j = j + 1
    valeurIncome = 0
    Select Case newchain
            Case "A": valeurIncome = 29000 / 2
            Case "B": valeurIncome = (29000 + 50000) / 2
            Case "C": valeurIncome = (50000 + 100000) / 2
            Case "D": valeurIncome = (100000 + 150000) / 2
            Case "E": valeurIncome = (150000 + 200000) / 2
            Case "F": valeurIncome = (200000 + 30000) / 2
            Case "G": valeurIncome = (300000 + 500000) / 2
            Case "H": valeurIncome = (500000 + 750000) / 2
            Case "U": valeurIncome = (750000 + 1000000) / 2
            Case "J": valeurIncome = (1000000 + 1500000) / 2
            Case "K": valeurIncome = (1500000 + 2500000) / 2
            Case "L": valeurIncome = (2500000 + 3500000) / 2
    End Select
    Print #2, valeurIncome
Wend

Close #1
```

```
MsgBox  j
'MsgBox "Number  of  households  =  "  &  nbrhouseholds  &  "  Files  scalefaofam.txt  and  sizeh
End  Sub
```

References

African Centre for Statistics and African Economic Commission. 2016. African Statistical Yearbook. http://www.uneca.org/publications/african-statistical-yearbook-2016.

Bradbury Bruce (2003). The Welfare interpretation of consumer equivalence scales. *International Journal of Social Economics*, vol. 30, 7, pp.770-787.

Duclos J. Y. et Mercadier-Pats, M. (1996). *Households needs and poverty : with Application to Spain and UK*. Laval, MIMEO, Université de Laval, Canada.

Haidara, M. C and G. S. Lo. (2008). Sur la décomposabilité empirique des indicateurs de pauvreté. ArXiv:1701.02649

Haidara M.C and LO G.S.(2012) Statistical Estimation of Gap of Decomposability of the General Poverty Index. *International Journal of Statistics and Probability*, 1 (2). doi:10.5539/ijsp.v1n2p211

Hourriez J. M. and Olivier L.(1997). Niveau de vie et taille du ménage : estimation d'une échelle d'équivalence. *Economie et Statistique*, 308-309-310.

LO, G.S. (2014). Pourquoi JAFSA? Why AJAS. African Journal of Applied Statistics . Volume 1(1), pp 1-2 Doi : http://dx.doi.org/10.16929/ajas/2014.1.1.51

LO, G.S.(2016) Visual Basic and R packages for preparing households Databases for statistical studies. Available at : **http://www.jafristatap.net/archives/ajas-2016-01-06-HouseHoldsDatabases.txt**

Haidara, M. C and G. S. Lo.(2008). Sur la décomposabilité empirique des indicateurs de pauvreté.ArXiv:1701.02649

Lo, G. S. (2008). Estimation asymptotique des indices de pauvreté : modélisation continue et analyse spatio-temporelle de la pauvreté au Sénégal. Journal Africain de Communication Scientifique et de Technologies.
Available at $http://www.ufrsat.org/lerstad/pub/gslo\_isr2008.pdf$.

Lo, G. S., Sall, S.T. and Seck, C.T. (2009). Une théorie asymptotique des indicateurs de pauvreté . *C. R. Math. Rep. Acad. Sci. Canada*, 31 (2), 45-52.

Lo, G. S.(2013). On the General Poverty Index. (2013). *Far East Journal of Theoretical Statistics*. Volume 42. (1), 1-22

Mergane P. D. and Lo G.S. (2013). On the Functional Empirical Process and Its Application to the Mutual Influence of the Theil-Like Inequality Measure and the Growth. *Applied Mathematics*, 4 (7).
Doi : 10.4236/am.2013.47136

Pollack A. R. and Wales Terence J.(1979). Welfare comparison and Equivalence scales. *American Economics Association*, 69.

Lo, G.S. and Sall. S.T.(2010). Asymptotic representation theorems for poverty indices. *Afrika Statiska*, 5 (2010). 238–244. MR2920300.