Training in Household Surveys

An overview of data analysis commands using STATA (11.1)

By

James Muwonge

Uganda Bureau of statistics

1 Data Management in household surveys

Introduction

- Data management is an integral part of the entire household survey planning process. Integration of data management across the board is key
  - Field data entry should be pre-planned
    - Data manager should participate in questionnaire design
  - CAPI also demands prior planning for data management resources
  - Planning for data entry (resources needed, expected future demand and new technological adoptions needed

  - Data storage mechanisms
    - Data bases

1 Data Management in household surveys cont'd

- Double data entry
- Closely checking for
  - range checks,
  - checks against reference data,
  - skip checks,
  - consistency checks (of occupations, of expenditures, income etc) and
  - typographic checks in data

1 Data Management in household surveys cont'd

- Data entry platforms
- CsPro, etc
  - Data entry screen features,
  - Data entry workloads
- Data storage
- Flat files or ??
  - What is the practice in Suriname?
  - Is the data structure user friendly ?

Data analysis

- Data analysis is an integral part of the household survey process
- It is sometimes undertaken as data is being processed
  - Before data analysis commences: understand the data you are going to deal with
  - Its structure, whether its categorical, or quantitative
  - Get and read about the basic descriptions of the survey (objectives, coverage, sample design, and limitations)
- The statistical tests you want to perform
- The validity of the data and its reliability
- BUT before analysis, A tabulation plan is needed

1 Data analysis

Explore the data

- View the data and understand its structure
- Statistical packages have provision to view the data (stata for instance has a data viewer (next page)

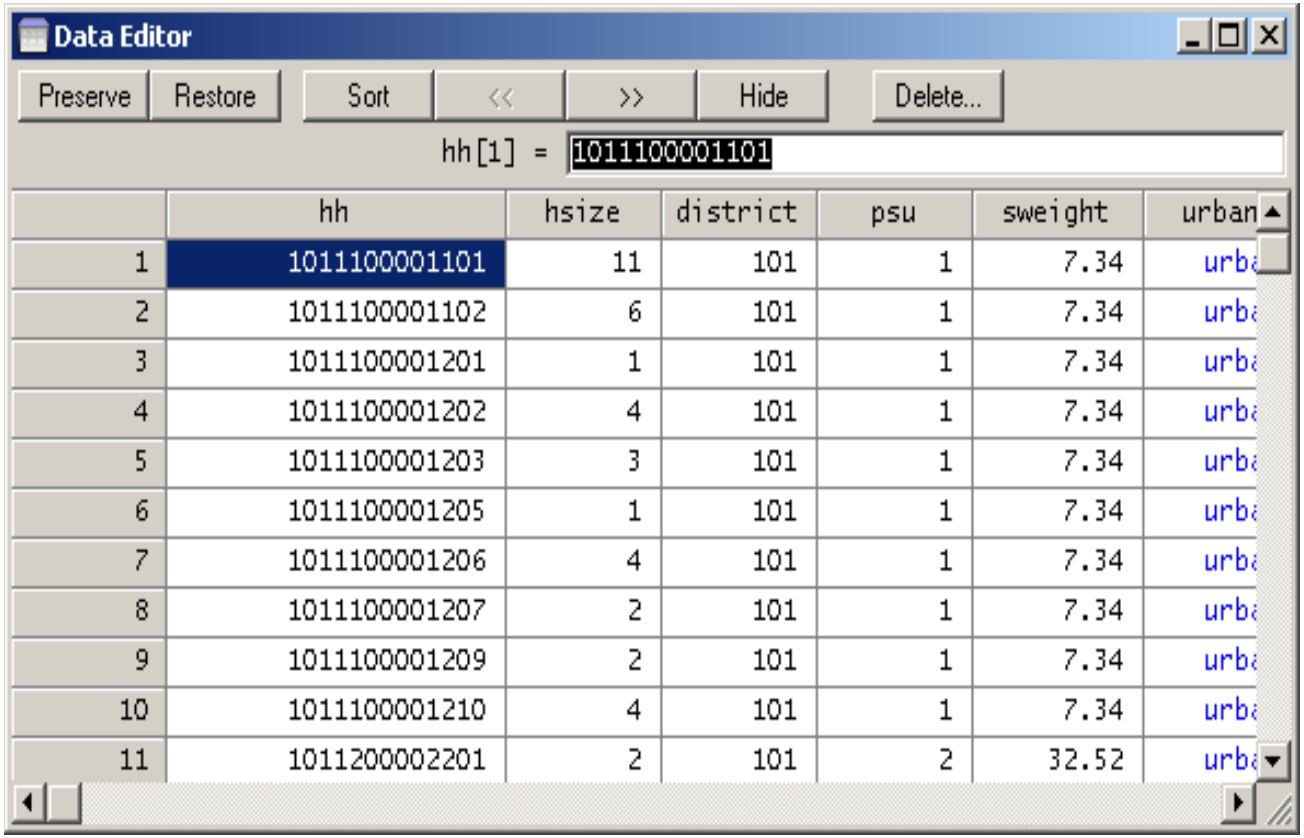
The Data Viewer/Editor

The format of the Data Viewer is similar to that of the Data Editor. However, this tool can be used only to visualize the dataset without changing it.

To display the Data Viewer:

- Click on the icon of the **Data Viewer**;
- Or type the command browse.

Editing and visualising the Stata datafiles
2.6.1 The Data Editor



	hh	hsize	district	psu	sweight	urban
1	1011100001101	11	101	1	7.34	urba
2	1011100001102	6	101	1	7.34	urba
3	1011100001201	1	101	1	7.34	urba
4	1011100001202	4	101	1	7.34	urba
5	1011100001203	3	101	1	7.34	urba
6	1011100001205	1	101	1	7.34	urba
7	1011100001206	4	101	1	7.34	urba
8	1011100001207	2	101	1	7.34	urba
9	1011100001209	2	101	1	7.34	urba
10	1011100001210	4	101	1	7.34	urba
11	1011200002201	2	101	2	32.52	urba

Good practices

- If you like to explore the dataset without changing it, it is more appropriate to use the Data Viewer to avoid the no desired changes.

Easy ways

- By default, for the labelled variable values (Ex. 1: rural area / 2: urban area). By default, label values of thes variables will be displayed in the Data Editor or the Data Viewer. If you wish to edit the values, you must add the option nolabel (e.g. edit area, nolabel).
- Whereas the alphanumeric contents (string) are displayed in red

Explore data cont’d

- Runs some basic descriptive statistics (mean, median, skewness etc) to further understand the data.

Descriptive analysis and exploration of data

Stata makes it possible to inspect the variables or to calculate their simple descriptive statistics.

5.1 Inspecting and comparing variables

The command **inspect** provides a fast synopsis of a numeric variable. It gives the number of negative, zeros, and positive values; the number of integer and real values; the number of single values; and the number of missing values; and it produces a small histogram. Its goal is not analytical, but it makes it possible to be familiarized quickly with the unknown data. The syntax of this command is:

Inspect [varlist] [if]

Example

```
trip:                                        Number of Observations
-------                                ------------------------------
                                        Total   Integers   Nonintegers
|                   #        Negative       -          -             -
|                   #        Zero           -          -             -
|                   #        Positive    3153       3153             -
|                   #                   -----      -----         -----
|  #   #   #   #   #        Total       3153       3153             -
|  #   #   #   #   #        Missing      273
+----------------------                 -----
1                     6                  3426
  (6 unique values)
```

Producing the simple descriptive statistics: the commands summarize and tabstat

The command **summarize e** compute the descriptive statistics for the numerical variables2. The syntax of this command is:

[**by** varlist :] ***summarize*** varlist [if] [in] [weight] [, options]

Insofar as no option is specified, Stata produced for each variable of the varlist the number ofobservations (Obs), the average (Mean), the standard deviation (Std. Dev.), the minimal value (Min.) and the maximum value (Max.).

The option detail generates more detailed statistics, such as the kurtosis and the skewness (a measurement of the asymmetry of the distribution), etc.

Example :

The command line

bysort sex **: summarize** welfare

produces the descriptive statistics of the variable income for each of the modalities of the variable education.

The command **tabstat** makes it possible to produce almost the same results like of those of **summarize**, but allows more flexibility in the choice of the descriptive statistics

Example:

The command line

tabstat welfare hsize, stats(mean, median, variance, sd, skewness)

produces the mean, the median, the variance, the standard deviation, and the skewness of the variables income and size

Frequency and cross tabulations statistics: the command tabulate

The command **tabulate** allows to produce the one-way table of frequency counts. The syntax of this command is:

[**by** varlist :] **tabulate** varname [if] [in] [weight] [, options]

Examples:

The command line

Tabulate hsex if region == 5, nolabel

gives the frequencies of the variable sex (number of males and that of females) in the strata that take the code 5.

tabulate hsex, generate(x)

gives the frequencies of the variable sex and generates dummy variables for each of the modalities of the variable hsex.

In addition, the command **tabulate** allows to create crossing table based on two categorical variables.

[**by** varlist :] **tabulate** var1 var2 [if] [in] [weight] [, options]

The option chi2 allows performing the Pearson test of independence (Null Hypothesis: independence of the crossing lines and columns).

Remarks

1. The command **tabulate** is more appropriate with the categorical variables.

2. If we wish to produce the frequency counts for more than one categorical variable, we can use the command **tab1**:

tab1 varlist [if] [in] [weight] [, options]

Example:
tab1 region hsex

3. **I we wish to produce the crossing table frequencies for more than one combination of two variables, we can use the command tab2**:

Tab2 varlist [if] [in] [weight] [, options]

Example:
Tab2 region hsex

Obtaining more elaborated descriptive statistics on a given variable:

the command table

The command **table** represents a combination of the commands **summarize** and **tabulate**. It provides a descriptive statistical table.

Examples:

table region

It provides a table of frequencies to the variable region.

table region, contents (mean income median income)

It provides the mean and median of the variable income by region.

table region sex, c(mean welfare median hsize)

It provides mean of the variable welfare and the median of the variable hsize for each of the modalities of the variable region and by education level.

Analyzing the correlation between variables : the command correlate

The command **correlate** allows estimating the correlation or covariance matrix for a list of variables. The syntax of this command is:

[**by** varlist :] **correlate** varlist [if] [in] [weight] [, options]

The usually options of this command are:

Options

covariance	display covariances.
means	display means, standard deviations, minimums, and maximums of variables in addition to the matrix.

Examples

correlate welfare hsize in 1/100, means

Estimate the correlation matrix of the variables income, education and size when the observations are the 100 firsts observations.

correlate welfare hsize, c

Estimate the variance-covariance matrix *of the variables income, education and size*.

Remark

The command **pwcorr** displays all the pairwise correlation coefficients between the variables in varlist or between all the variables in the dataset if varlist is not specified.

Tests on the mean, the variance of variables: the commands ttest and prtest

The command **ttest** allows performing the statistical tests for the estimate mean or to test the equality of the estimated means of two variables. To perform the tests with the statistic mean, the syntax is:

ttest varname == # [if] [in] [, level(#)]

However, if we compare between means of two variables, the syntax is:

ttest varname1 == varname2 [if] [in] [, options]

The command **ttest** allows also testing the difference in mean between two population groups.

ttest varname [if] [in], by(groupvar) [options]

by(groupvar) specifies the groupe variable.
groups.

The syntax of the command **prtest** is similar to that of the command **ttest,** but this command allows performing tests with the statistic proportion. The syntax of this command is:

prtest varname == p [if] [in] [, level(#)]

The variable varname is supposed to be a dummy variable. Moreover, when it is a question of testing if two variables have the same proportion, the syntax is:

prtest varname1 == varname2 [if] [in] [, options]

Lastly, when it is a question of testing the difference in proportion of a variable between two groups of the population, syntax is:

prtest varname [if] [in], by(groupvar) [options]

Tests on the mean, the variance of variables: the commands ttest and prtest

Examples

ttest size == 5 if region==3

allows to test if the average size of household equals to 5 in region 3

ttest income1990 == income2000

allows to test if the difference in average incomes equals to zero between the years of 1990 and 2000.

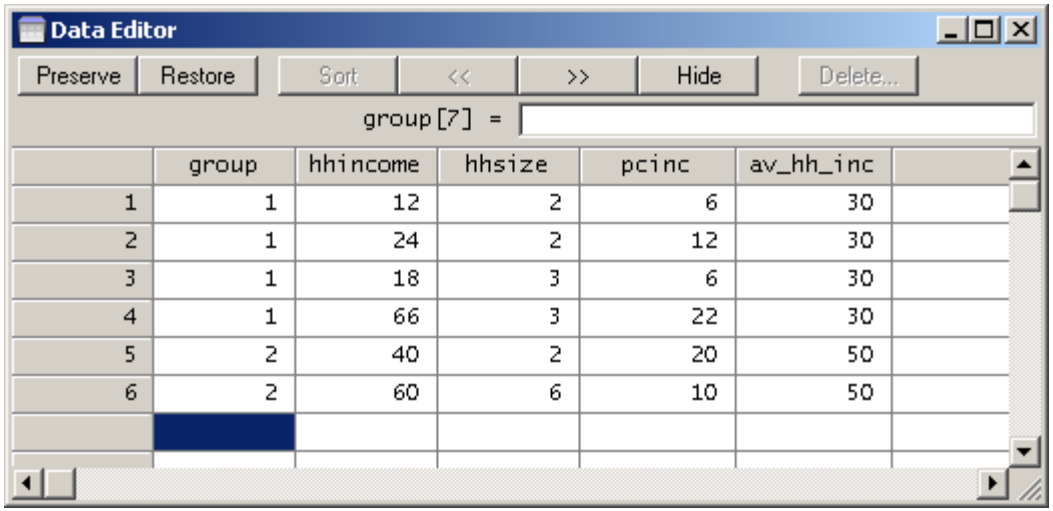
ttest welfare, by(hsex) unequal

allows to test if the difference in average welfare equals to zero between male and female heads.

6 Generating new variables

There is two main commands to generate new variables. The command **generate** allows to generate variables that require simple arithmetic computations (observation to observation).

The command **egen**
(*extended generate*) is more appropriate when computations are based on the whole or a part of
observations (observations to observation).



	group	hhincome	hhsize	pcinc	av_hh_inc
1	1	12	2	6	30
2	1	24	2	12	30
3	1	18	3	6	30
4	1	66	3	22	30
5	2	40	2	20	50
6	2	60	6	10	50

6.1 The command generate

The command **generate** allows to generate new variables. Values of these variables are given by **=**
exp.

generate [type] newvar[:lblname] =exp [if] [in]

If the type of variable is not indicated, the type of the new variable is determined automatically by the type of result returned by expression **= exp**. A variable with type float or double is generated if the result is numeric, and a string variable is generated if the result is a text.

Examples

use "E:\UBOS\Training\UDHS.dta", clear
generate hage^2 = hage*hage
generate poor = wefare <2140 & iwelfare !=.
gen year = 2007 /* *generates a constant* variable year that equals to 2007 */
gen x1 = "poor" in 1/10 /* *generates a string variable string that equals to poor in the 10 first observations* */
gen x2 = (x1 == "poor") /* x2 = 1 *if* x = "poor" *and* 0 *otherwise* */
gen x3 = (income <= 500) /* x3 = 1 *if* income <= 500 *and* 0 *otherwise* */

7 Combining the datafiles

Stata can open only one database at the same time. To clean the Stata memory, the command clear should be used. It is an essential operation before charging another datafile.

To use several datafiles, the simplest method consists in opening the first datafile, to use it, then to close it and open thereafter the second datafile, etc. However, when one needs at the same time variables or observations stored in different datafiles, it is necessary to combine these datafiles and to create a new one. For this end, three principal methods can be used. Each one of them meets a specific need.

7.1 Appending datafiles -vertical concatenation- (append)

The command append can be used to add new observations to the current datafile. We have to open the first datafile.

use *E:\UBOS\Training\UDHS*, clear

After that, we use the command **append**:

append using *E:\UBOS\Training\unhs06 ,*[*, nolabel*]

This makes it possible to complete the observations contained in the first datafile with those contained in the second datafile.

7.2 Merging datafiles -horizontal concatenation- (merge)

Usually, for some computations and analysis, we need variables that are stored in different datafiles, but are of the same sample. This is usually the case with household surveys and where the dataset is saved in different datafiles according to the main parts of the questionnaire, for instance, household characteristics, household expenditures, etc.

The command merge allows adding new variables to the current datafile.

The command **merge** requires looking for certain rules:

- There is a master datafile and a secondary datafile (using datafile).

- By default, if a variable is present in the two datafiles, then values of the master datafile will remain unchanged after the merging process.

- If some variables of the secondary datafile have the same names of variables in master datafile, but contents of variables are different, one must change the names of these variables in one of the two datafiles before making fusion (for instance by using the command **rename)**.

The use of the command **merge** involves the creation of a new variable named _merge which summarizes the result of merge. The possible values of _merge are:

- _merge = 1 when the data of the observation comes exclusively from the master datafile;
- _merge = 2 when the data of the observation comes exclusively from the secondary datafile;
- _merge = 3 when the data of the observation comes from the two datafiles

```
use "E:\UDHS\Training\data\GSEC3.dta", clear
merge 1:1  hh  pid  using  "E:\UDHS\Training\data\GSEC2clnv.dta" ,
```

```
. tabulate _m
```

_merge	Freq.	Percent	Cum.
master only (1)	444	2.81	2.81
using only (2)	498	3.16	5.97
matched (3)	14,835	94.03	100.00
Total	15,777	100.00	

Remarks

1. One should not sort the data before merging with the option: *observation by observation*.

2. With the merge observation by observation, _merge = 3 means that the two datafiles have the same number of observations.

3. It is strongly recommended to make the fusion by using key variables, which represents also the unique identifier of observations.

Weighting of data

- Weight the data during analysis because it was based on a sample
- Always compare your results with what is published
- Document any modifications you may have made to the data

inform the GBS of these changes if they will improve future data quality

Regression Analysis

- Know the data and the relationships you want to study
- Always start with descriptives

LOCATION	Nobserved	Mean	Minimum	Maximum	Median
ASSA	53	1.025	0.4000	2.200	0.900
BILISA	111	1.843	0.4000	5.600	1.800

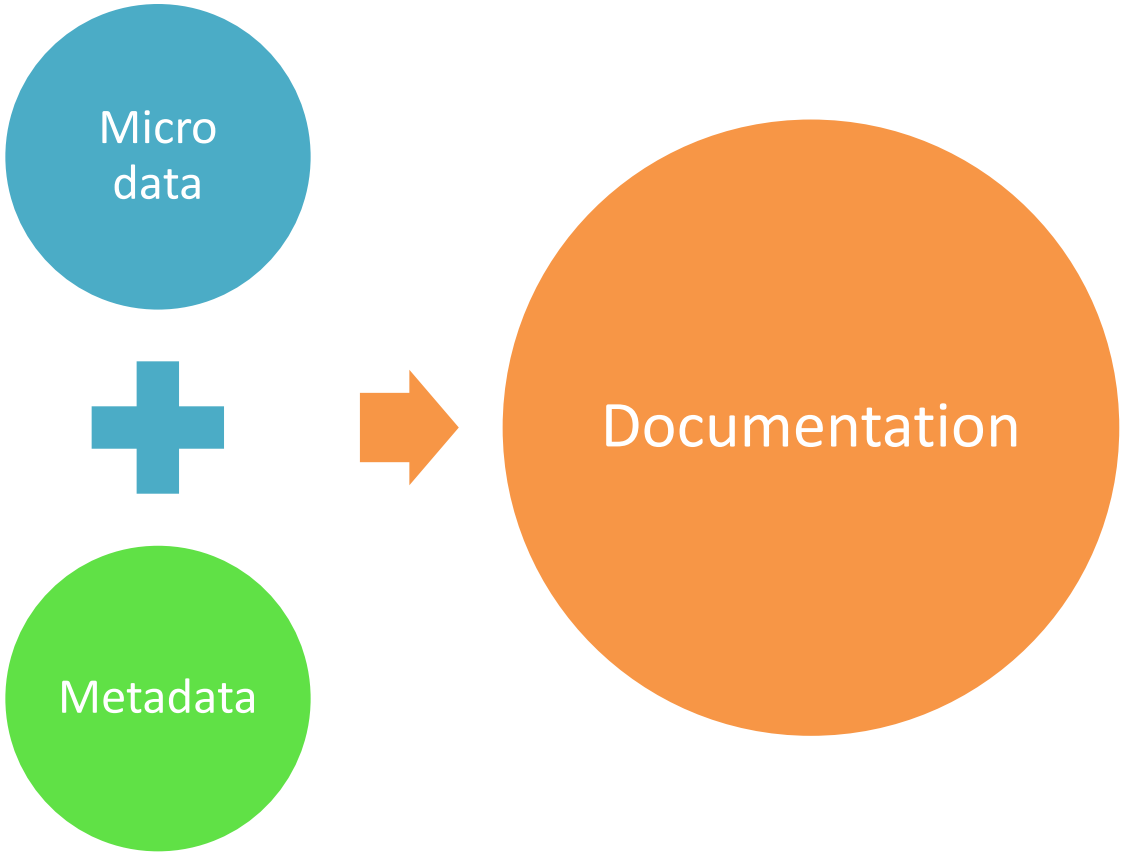
- Start with an OLS

Regression Analysis cont'd

- Delve into specifics depending on the objectives of your study
- How many regressors to include in the model?
  - Depends
- Need to be coversant with some regression theory and econometric principles to fully appreciate regression analysis methods
- You may consider an inhouse course on this in future

Documentation and Archiving

Documentation



Microdata

- Microdata description
  - Data archives
  - Variables and codes dictionary
    - Tag (name)
    - Description
    - type
    - length
    - Decimals

Metadata

- Conceptual framework of investigation
- Reference terms
- Offer, if it is the case
- Contracts
- Work plan
- Pilot report
- Questionnaires
- Manuals
- Ethic's committee presentation

- Sampling plan
- Field advances report
- Final field report
- Data dictionary
- Analytical reports
- Others

Data access points

- Public access points
  - Resource centre/libraries
  - Protocols for accessing it online
    - Document procedure
    - Expected time lag within which feed back is expected

END

THANK YOU