

clear all

clear matrix

set more off

*set maxvar 10000

*set mem 500m

cap log close

Nepal MICS 2014

***Working Folder Path ***

global workingfolder_in "T:\7. Research\MPI 2015\0. Raw data\Nepal MICS 2014\Data"

global workingfolder_out "T:\7. Research\MPI 2015\0. Raw data\Nepal MICS 2014"

global workingfolder_in "C:\Nepal MICS 2014"

global workingfolder_out "C:\Nepal MICS 2014"

*Log file: please search on google the official ISO country code for the country you are analysing and use this three digits to identify the documents and datasets this dofile creates

*Please look at the front page of the report

log using "\$workingfolder_out\npl_mics14_dataprep.log", replace //Change country name and year correspondingly//

* Step 1: Data preparation

***** Selecting main variables from WM and CH recode and merging with HL recode

*** Step 1.1 CH – Children’s Level (Children under 5) *****

use "\$workingfolder_in\ch.dta", clear //Change to the corresponding CH-file//

*** Step 1.1a Key variable for merging

*I convert the variable names to lower caps to save time

rename _all, lower

*** Generate individual unique key variable required for data merging

*** hh1= cluster number; hh2 = household number; ln = child's line number in household

gen double ind_id = hh1*100000 + hh2*100 + ln //Please check length of variables for a correct generation of the id variable//

format ind_id %20.0g

label var ind_id "Individual ID"

duplicates report ind_id

duplicates tag ind_id, gen(duplicates)

tab ln if duplicates!=0 //A number of children are not listed in the household//

*** For Children not listed in the household we create a false household line to check at merging stage

bys ind_id: gen line = (_n)

replace ind_id = hh1*100000 + hh2*190 + line if duplicate!=0 //We assume consecutive hh line starting at 90//

duplicates report ind_id //We should not have any duplicate at this stage//

gen child_CH = 1 //Identification variable for observations in CH recode//

*** Step 1.1b Children Nutrition

//NOTE: Not all MICS surveys have information about children nutrition. If this is the case, please skip this section until step 1.1c "Replacement of missing for nutrition"//

*** For this part of the do-file we use the WHO Anthro and macros to calculate the z-scores of the children's

*** nutritional variables (source: <http://www.who.int/childgrowth/software/en/>).

*** Please follow the instructions:

*** 1) Unzip the file "igrowup_stata.zip"

*** 2) Save the .ado files in "C:\ado\plus\a\"

*** 3) Save the dta files of the "igrowup_stata" folder into the directory that you want to use

*** as "Directory of reference tables", we recommend the following: "C:\igrowup_stata"

*** Following, we indicate to the Stata compiler where the igrowup_restricted.ado file is stored

*** using the following command:

```
adopath + "C:\ado\plus\a\"
```

*** We will now proceed to create three nutritional variables: weight-for-age (to measure Underweight),

*** weight-for-height (or height-for-weight, the same, to measure Wasting) and height-for-age (to measure Stunting)

*** We specify the first three parameters we need in order to use the ado file: replib, datalib & datalab

*** We use 'replib' to specify the package directory where the five STATA datasets (.dta files) containing

*** the WHO Child growth Standards are stored ("Directory of reference tables" indicated above)

*** Note that we use strX to specify the length of the path in string. If the path is long, such as

*** "C:\Documents and Settings\user\My Documents\Mauritania 2007" and you specify str20, it won't run.

*** In that case you will need to specify str55 or more.

*replib: to specify the package directory where the five STATA datasets containing the WHO Child growth Standards are stored

```
gen str100 replib = "C:\igrowup_stata" //Modify directory according specification above//
```

```
lab var replib "Directory of reference tables"
```

*datalib: to specify the working directory where the input STATA data set containing the anthropometric measurement is stored

```
gen str100 datalib = "$workingfolder_out" //Modify directory accordingly //
```

```
lab var datalib "Directory for datafiles"
```

*datalab: to specify the name that will prefix the output files that will be produced from using this ado file (datalab_z_r_rc and datalab_prev_rc)

```
gen str30 datalab = "children_nutri_npl" //
```

```
lab var datalab "Working file"
```

*** We now indicate the variables the WHO ado needs to calculate the z-scores: sex, age, weight, height

* Variable "Sex"

```
tab hl4, miss //Check the variable for "sex" has "1" for male, "2" for female and all missing values are "."//
```

```
gen gender = hl4
```

```
desc gender
```

```
tab gender
```

```
label define gender 1"male" 2"female", replace
```

```
label values gender gender
```

* Variable "Age"

* The age variable can be expressed in months or days

```
ta cage,m //Check all missing values are "."//
```

```
ta caged,m //Check all missing values are "."//
```

```
codebook cage caged //Check unit of measure – in this case age is measured in months//
```

```
clonevar age_months = cage
```

```
clonevar age_days = caged
```

```
replace age_months = . if cage==99 //No missing data here//
```

```
gen str6 ageunit = "days" //We define the unit of measure, in this case age is measured in months, otherwise please change to ageunit = "days"//
```

```
lab var ageunit "Days"
```

* Variable "body weight" – it must be in kilograms

```
ta an3, miss //Check all missing values are "."//
```

```
codebook an3 //Check unit of measure – in this case weight is measured in kilograms//
```

```
clonevar weight = an3
```

```
desc weight
```

```
replace weight = . if an3>=99 //All missing values or out of range are replaced as "."//
```

```
tab an2 an3 if an3>=99 | an3==., miss //an2: result of the measurement//
```

```
summ weight
```

* Variable "height" – it must be in centimetres

```
ta an4, miss //Check all missing values are "."//
```

```
codebook an4 //Check unit of measure – in this case age is measured in centimetres//
```

```
clonevar height = an4
```

```
desc height
```

```
replace height = . if an4>=999 //All missing values or out of range are replaced as "."//
```

```
tab an2 an4 if an4>=999 | an4==., miss
```

```
summ height
```

//For children aged below 24 months (<731 days), it makes a difference whether the child was measured lying down (recumbent), in which case his/her length was measured, than whether he was measured standing, in which his/her measure is height. The ado can adjust for this difference. When the kid was measured standing, the ado converts the height to recumbent length by adding 0.7cm, and for children aged 24 months or above, who are measured in a recumbent position, the ado converts the length to standing height by subtracting 0.7 cm. The exported variable is _clenhei, which is the converted length/height according to age//

//In MICS all children under 24 months are supposed to have been measured in recumbent position, and all children above 24 months are supposed to have been measured in standing position. However, the survey provides a variable that controls for this: an4a, in case a different practice was used in some cases.//

```
//NOTE: if an4a is not available, please create as [gen str1 measure=""]//
```

```
codebook an4a
```

```
gen measure = "l" if an4a==1 //Child measured lying down//
```

```

replace measure = "h" if an4a==2 //Child measured standing up//
replace measure = " " if an4a==. // No information
replace measure = " " if an4a==9 //Replace with " " if unknown or missing//
tab measure,m

```

```

* Variable "Oedema" //Only if the variable is available to control for oedema//
//NOTE: if not available, please create as [gen str1 oedema = "n"]//
gen    str1 oedema = "n"
*replace oedema = "y" if an5==1
*replace oedema = " " if an5==3 | an5==7 | an5==9 | an5==.
desc    oedema

```

```

* Variable "Sampling weight"
//NOTE: if not available, please create as [gen sw = 1]//
clonevar sw = chweight
desc sw
summ sw

```

**** We now run the command to calculate the z-scores with the adofile**

```

igrowup_restricted reflib datalib datalab gender age_days ageunit weight height measure oedema
sw

```

**** We now turn to using the dta file that was created and that contains the calculated z-scores**

use "\$workingfolder_out\children_nutri_npl_z_rc.dta", clear //Change name accordingly//

```

gen    z_scorewa = _zwei
replace z_scorewa = . if _fwei==1
lab var z_scorewa "z-score weight-for-age WHO"

```

*scatter z_scorewa waz2 if waz2<90 // we obtain the same results/100 as the MICS

** Now we create the under weight-for-age variable with WHO

```
gen    underwa2 = (z_scorewa < -2.0) //Takes value 1 if the child is under 2 stdev below the median
and 0 otherwise//
```

```
replace underwa2 = . if z_scorewa==.
```

```
lab var underwa2 "Child is undernourished (weight-for-age) 2sd - WHO"
```

* Ultrapoverty indicator

```
gen    underwa3 = (z_scorewa < -3.0) //Takes value 1 if the child is under 3 stdev below the median
and 0 otherwise//
```

```
replace underwa3 = . if z_scorewa==.
```

```
lab var underwa3 "Child is undernourished (weight-for-age) 3sd - WHO"
```

```
gen stunting = (_zlen < -2.0)
```

```
replace stunting = . if _zlen == . | _flen==1
```

```
lab var stunting "Child is undernourished (lenght/height-for-age) 2sd - WHO"
```

```
gen stunting3 = (_zlen < -3.0)
```

```
replace stunting3 = . if _zlen == . | _flen==1
```

```
lab var stunting3 "Child is undernourished (lenght/height-for-age) 3sd - WHO"
```

```
gen wasting = (_zwfl < - 2.0)
```

```
replace wasting = . if _zwfl == . | _fwfl == 1
```

```
lab var wasting "Child is undernourished (weight-for-lenght/height) 2sd - WHO"
```

```
gen wasting3 = (_zwfl < - 3.0)
```

```
replace wasting3 = . if _zwfl == . | _fwfl == 1
```

```
lab var wasting3 "Child is undernourished (weight-for-lenght/height) 3sd - WHO"
```

*We do not have information on ethnicity or religion for children

*** Save a temp file for merging with WM, IR and MR

```
keep ind_id hl4 cage hh1 hh2 ln chweight child_CH z_score* under* stun* wast* _*
```

```
order ind_id hl4 cage hh1 hh2 ln chweight child_CH z_score* under*
```

```
sort ind_id
```

```
save "$workingfolder_out\npl14_CH.dta", replace
```

```
*****  
*****
```

```
*** Step 1.2 WM – Women Level (all eligible females between 15-49 years in the household) ***
```

```
*****  
*****
```

```
use "$workingfolder_in\wm.dta", clear
```

```
*I convert names to lower caps
```

```
rename _all, lower
```

```
*** Generate individual unique key variable required for data merging
```

```
*** hh1 = cluster number; hh2 = household number; ln = respondent's line number
```

```
gen double ind_id = hh1*100000 + hh2*100 + ln //Please check length of variables for a correct  
generation of the id variable//
```

```
format ind_id %20.0g
```

```
label var ind_id "Individual ID"
```

```
duplicates report ind_id
```

```
gen women_WM = 1 //Identification variable for observations in WM recode//
```

```
ta ma5 cm1,m //No subsample of ever-married women//
```

```
* We retain the following variables: wb2: age; wmweight: sample weight; cm1: ever given birth;  
cm8: ever had child who later died;
```

```
* cm9a: sons who have died; cm9b: daughters who have died; cm10: total children ever born; ma1:  
currently married; ma5: ever married
```

```
*no info of ethnicity or religion
```

```
codebook wb2 wm7 cm1 cm8 cm9a cm9b cm10 ma1 ma5 ind_id women_WM wmweight
```

```
keep wb2 wm7 cm1 cm8 cm9a cm9b cm10 ma1 ma5 ind_id women_WM wmweight ta1-ta17
```

```
order wb2 wm7 cm1 cm8 cm9a cm9b cm10 ma1 ma5 ind_id women_WM wmweight
```

```
sort ind_id
```


*** Save temp file for future merging

save "\$workingfolder_out\npl14_WM.dta", replace

*** Step 1.3 HH – Household Characteristics

use "\$workingfolder_in\hh.dta", clear //Change to the corresponding HH-file//

*I convert names to lower caps

rename _all, lower

*** Generate a household unique key variable required to prepare the MPI indicators at the household level ***

*** We construct the household id using hh1: cluster number and hh2: household number

gen double hh_id = hh1*1000 + hh2 //Please check length of variables for a correct generation of the id variable//

format hh_id %20.0g

lab var hh_id "Household ID"

*** Save temp file for future merging

save "\$workingfolder_out\npl14_HH.dta", replace

Step 1.4 No Male Recode , but Birth recode

/*use "\$workingfolder_in\mn.dta", clear //Change to the corresponding MN-file//

*I convert names to lower caps

rename _all, lower

gen double hh_id = hh1*1000 + hh2 //Please check length of variables for a correct generation of the id variable//

format hh_id %20.0g

```
lab var hh_id "Household ID"
```

```
*** Generate individual unique key variable required for data merging
```

```
*** hh1 = cluster number; hh2 = household number; ln = respondent's line number
```

```
gen double ind_id = hh1*100000 + hh2*100 + ln //Please check length of variables for a correct  
generation of the id variable//
```

```
*** Save temp file for future merging
```

```
save "$workingfolder_out\npl14_MN.dta", replace
```

```
*/
```

```
use "$workingfolder_in\bh.dta", clear
```

```
rename _all, lower
```

```
*** Generate individual unique key variable required for data merging
```

```
*** hh1 = cluster number; hh2 = household number; ln = respondent's line number
```

```
gen double ind_id = hh1*100000 + hh2*100 + ln //Please check length of variables for a correct  
generation of the id variable//
```

```
format ind_id %20.0g
```

```
label var ind_id "Individual ID"
```

```
gen double ind_c = hh1*10000000 + hh2*10000 + ln*100 + bhln //Please check length of variables  
for a correct generation of the id variable//
```

```
duplicates report ind_c
```

```
gen women_BH = 1 //Identification variable for observations in WM recode//
```

```
* We retain the following variables: bh5: still alive; bh9u: unit of age at dead; bh9n: age at dead;  
bh9c: check; bh9f: flag.
```

```
bys ind_id: egen tbirths_ = max(bhln)
```

```
ta bh5 bh9u,m //only not alive offspring has age at death
```

```

keep wmweight hh1 hh2 ln tbirths_ bhln bh5 bh9* ind_id women_BH

*check that age at dead inputed is valid for almost all children not alive

bys bh9f : ta bh9c bh5 ,m //all but 1 of 15 imputations are under 5 years old, most of them lack of
unit This dofile relies on the imputation.

bys bh9f : ta bh9c bh9u if bh9f>=7 & bh9f<=8 ,m //all but 1 of 15 imputations are under 5 years old,
most of them lack of unit This dofile relies on the imputation.


gen dufive_ = cond(bh9c<60,1,0)

replace dufive_ = . if bh9c == .

ta bh9f dufive, m //2072 deaths of under five of which 14 has imputed age, 10 of them we have no
further info.

ren bh5 bh5_
ren bh9* bh9*_

reshape wide bh5 bh9* dufive_ i(ind_id) j(bhln)

egen tdufive = rowtotal(dufive_*)

ta tdufive, m //1110+(289*2)+(3*74)+(4*20)+(5*7)+47 = 2072 deaths under five which coincide
with the original dataset


sort ind_id

*** Save temp file for future merging

save "$workingfolder_out\npl14_BH.dta", replace


*****

*** Step 1.5 HL – Household Member Level *****

*****

use "$workingfolder_in\hl.dta", clear

*I convert names to lower caps

rename _all, lower

```

*Please check the official name of the country you are analysing in the List of Member States of the United Nations

*<http://www.un.org/en/members/index.shtml>

gen country = "Nepal"

*Please create a string of the year of fieldwork of the survey according to the report

*p. 2 of Nepal's report says from February till June 2014

gen year_published = "2014"

*Please create a variable that records the numeric year of survey. If it occurred in two years, please select the most recent of them

gen year = 2014

gen survey = "MICS"

*** Generate a household unique key variable required to prepare the MPI indicators at the household level ***

*** We construct the household id using hh1= cluster number and hh2 = household number

gen double hh_id = hh1*1000 + hh2 //Please check length of variables for a correct generation of the id variable//

format hh_id %20.0g

label var hh_id "Household ID"

*** Generate individual unique key variable required for data merging

*** hh1 = cluster number; hh2 = household number; hl1 = respondent's line number

gen double ind_id = hh1*100000 + hh2*100 + hl1 //Please check length of variables for a correct generation of the id variable//

format ind_id %20.0g

label var ind_id "Individual ID"

*** Step 1.6 DATA MERGING (HL, WM, CH) & Control Variables *****

*** 1.6a DATA MERGING *****

*** Merging WM Recode

merge 1:1 ind_id using "\$workingfolder_out\npl14_WM.dta"

gen temp = (hl7>0)

tab women_WM temp, miss col

tab wm7 if temp==1 & women_WM==. //Total of eligible women not interviewed//

ta hl6 hl4 if _m==1,m //only men and non-eligible women are not merged.

drop temp _merge

erase "\$workingfolder_out\npl14_WM.dta"

/*No Merging MN Recode (Male) for Nepal

merge 1:1 ind_id using "\$workingfolder_out\mwi13-14_mn.dta"

drop _merge

erase "\$workingfolder_out\mwi13-14_mn.dta"

*/

*** Merging HH Recode

merge m:1 hh_id using "\$workingfolder_out\npl14_HH.dta"

ta hh9 if _m==2,m

* 570 HHs not merged from Using, all of them due to problems in the interview

drop if _merge==2

drop _merge

erase "\$workingfolder_out\npl14_HH.dta"

*** Merging CH Recode

merge 1:1 ind_id using "\$workingfolder_out\npl14_CH.dta"

count if ln==0 //The children without household line are unique to the CH Recode//

```
ta hl6 if _m==1,m //only 5 years old and older are not merged.

replace hh_id = hh1*1000 + hh2 if ln==0 //Creates hd_id for children without household line, but
this is not a problem //

drop _merge
```

```
erase "$workingfolder_out\npl14_CH.dta"
```

```
sort ind_id
```

```
*Merging BH Recode for Nepal
```

```
merge 1:1 ind_id using "$workingfolder_out\npl14_BH.dta"
```

```
ta cm1 if _m==1,m //only those who have not given birth are not merged.
```

```
drop _merge
```

```
erase "$workingfolder_out\npl14_BH.dta"
```

```
*** MPI is based on de jure population only.
```

```
*****
*****
```

```
** 1.6b Create control variables if household has 'Eligible' members for WM and CH Levels
```

```
*****
```

```
*****
*****
```

```
//At this step we check if there is no female member eligible for the interview, if there are no
children to measure anthropometrics, ///
```

```
//if there are no women to measure BMI (Body Mass Index). This is important because for those
households we will have no info on some indicators of health.//
```

```
// We will consider these households as non-deprived in those particular indicators. For further
details see Alkire and Santos (2010)//
```

```
***Eligible Women for IR Recode***
```

```
*Please make sure that hl7 has no missing values. 'fem_eligible' below should acknowledge
```

*that (hl7>0 & hl7<.) are fem_eligible

gen fem_eligible = (hl7>0)

bys hh_id: egen hh_n_fem_eligible = sum(fem_eligible) //Number of eligible women for interview in the hh//

gen no_fem_eligible = (hh_n_fem_eligible==0) //Takes value 1 if the household had no eligible females for an interview//

label var no_fem_eligible "Household has no eligible women"

Eligible Men for MR Recode

/*No male recode in Nepal, Please active the following lines and change them accordingly//

*Please make sure that hl7a has no missing values. 'male_eligible' below should acknowledge

*that (hl7a>0 & hl7a<.) are male_eligible

gen male_eligible = (hl7a>0)

bys hh_id: egen hh_n_male_eligible = sum(male_eligible) //Number of eligible men for interview in the hh//

gen no_male_eligible = (hh_n_male_eligible==0) //Takes value 1 if the household had no eligible males for an interview//

label var no_male_eligible "Household has no eligible men"

*/

Eligible Children anthropometrics

gen child_eligible = (hl7b>0 | child_CH==1) //line number for mother/caretake for children under five//

bys hh_id: egen hh_n_children_eligible = sum(child_eligible) //Number of eligible children for anthropometrics//

gen no_child_eligible = (hh_n_children_eligible==0) //Takes value 1 if there were no eligible children for anthropometrics//

lab var no_child_eligible "Household has no children eligible"

No eligible Children AND no eligible Women - it won't be used for nutrition as only children have information on nutrition

gen no_child_fem_eligible = (no_child_eligible==1 & no_fem_eligible==1)

lab var no_child_fem_eligible "Household has no children or women eligible"

No eligible Women AND no eligible Men - Used for the applicable population of Child mortality

*gen no_eligibles = (no_fem_eligible==1 & no_male_eligible==1) //Takes value 1 if the household had no eligible members for an interview//

gen no_eligibles = (no_fem_eligible==1) //Takes value 1 if the household had no eligible members for an interview//

lab var no_eligibles "Household has no eligible women"

*drop hh_n_fem_eligible child_eligible hh_n_children_eligible

bys hh_id: gen id = _n

count if id==1

count

*Please copy in the Country Survey Info file the results from below

* people

count if no_eligible==1

* households

count if no_eligible==1 & id==1

* people

count if no_fem_eligible==1

* households

count if no_fem_eligible==1 & id==1

/* people

count if no_male_eligible==1

* households

count if no_male_eligible ==1 & id==1

*/

* people

count if no_child_eligible==1

* households

count if no_child_eligible==1 & id==1

* people

count if no_child_fem_eligible==1

* households

count if no_child_fem_eligible==1 & id==1

***** Step 2: Data preparation

***** Standardization of the 10 indicators and identification of deprived individuals

*** Rename basic variables

clonevar weight = hhweight //Sample weight//

codebook hh6

label list HH6

clonevar urban = hh6 //Type of place of residency: urban/rural//

replace urban = 0 if urban == 2 //Redefine the coding and labels to 1/0//

label define lab_urban 1 "urban" 0 "rural"

label values urban lab_urban

clonevar relationship = hl3 //Relationship to the head of household//

clonevar sex = hl4 //Sex household member//

replace sex = 0 if sex==2 //Redefine the coding and labels to 1/0//

replace sex = . if sex==9 //No missing values

label define sex 1 "male" 0 "female"

```
label val sex sex
```

```
clonevar age = hl6 //Age household member//
```

```
replace age = . if age==98 | age==99 // no missing values
```

```
//hhs should only include de jure members in the household. Most MICS do not allow identifying usual residents, so we proceed as follows. Please check whether your country has this information, and adjust the following lines accordingly//
```

```
sort hh_id
```

```
by hh_id: gen temp = _n
```

```
by hh_id: egen hhs = max(temp)
```

```
label var hhs "Household size"
```

```
drop temp
```

```
/*p.3Nepal MICS was designed to provide estimates for a large number of indicators
```

```
on the situation of children and women at the national level, for urban and rural areas, and for the following 15 sub-regions. The urban and rural areas within each sub-region were identified as the main sampling strata and the
```

```
sample was selected in two stages. Within each stratum, a specified number of census enumeration areas were selected systematically with probability proportional to size...The sample included 520 clusters*/
```

```
decode hh7, gen(temp)
```

```
encode temp, gen(region)
```

```
drop temp
```

```
*****  
*****
```

```
*** Step 2.1 Years of Schooling
```

```
*****
```

```
*****  
*****
```

```
//The entire household is considered deprived if no household member has completed five years of schooling//
```

```
*** Renaming and recoding Education Variables
```

//IMPORTANT: The coding of these variables in each country may differ so you need to convert it to the coding listed below each variable when defining the label corresponding to that var. Here we follow the same procedure as with housing characteristics//

*** MICS does not provide the number of years of education so we need to construct that variable from the edulevel and eduhighyear variables created in what follows ***

*tab ed4b ed4a, miss //ed4a does not exist in Nepal

codebook ed4b, ta(30)

tab age ed6b if ed5==1, miss

*Creating educational level variable

//I create this by looking at the number of years assigned to the level of education of head of household

ta ed4b helevel if relationship==1,m

gen edulevel = . //Highest educational level attended//

replace edulevel = . if ed4b==. | ed4b==98 | ed4b==99 //These are all missing values, (nsp: ne se pronounce pa and manquant: missing) check that that is also the case in your data//

replace edulevel = 0 if ed3==2 //Never attended school//

replace edulevel = 1 if ed4b>=0 & ed4b<=5 // Replacing higher education when the person has university or post university degree

replace edulevel = 2 if ed4b>=6 & ed4b<=10

replace edulevel = 3 if ed4b>=11 & ed4b<=14

replace edulevel = 0 if ed4b==94 //Preschool is treated as primary for the head of household, but in other countries we treat it as no education. Hence, it is treated as no education in here.

label define lab_edulevel 0 "None" 1 "Primary" 2 "Secondary" 3 "Higher", replace

label values edulevel lab_edulevel

label var edulevel "Highest educational level attended"

ta ed4b edulevel ,m nol

ta ed3 edulevel if ed4b==. ,m //some are missing values, some are never attended school

gen eduhighyear = ed4b //Highest grade of education completed//

```
replace eduhighyear = . if ed4b==. | ed4b==98 | ed4b==99 //These are all missing values, check that that is also the case in your data//
```

```
replace eduhighyear = 0 if ed3==2 //Never attended school//
```

```
replace eduhighyear = 0 if ed4b==94 //Preschool is treated as primary for the head of household, but in other countries we treat it as no education. Hence, it is treated as no education in here.
```

```
lab var eduhighyear "Highest year of education completed"
```

```
tab eduhighyear edulevel, m
```

**** Cleaning inconsistencies**

```
replace eduhighyear = 0 if age<10 //Based on UNDP agreed bottom age threshold
```

```
replace eduhighyear = . if edulevel==1 & eduhighyear>5 // According to the report (page 166) Primary school is until 8th grade
```

```
replace eduhighyear = . if edulevel==2 & eduhighyear>10
```

```
replace eduhighyear = . if edulevel==3 & eduhighyear>14 // secondary education covers 8grades of education
```

```
replace eduhighyear = 0 if edulevel==0
```

**** Now we create the years of schooling**

```
gen edueyears = eduhighyear
```

*I won't need to adjust this as the variable comes as continuous years of education originally, higher education may assume continues years but this does not affect poverty.

**** Checking for further inconsistencies**

```
//There are some cases in which the years of schooling are greater than the age of the individual, which is clearly a mistake in the data. There might also be individuals that show too much schooling given their age (e.g. a 7 year-old with 5 years of schooling). Please check whether this is the case in your country and correct when necessary//
```

```
replace edueyears = . if age<=edueyears & age>0
```

```
replace edueyears = 0 if age<10 //agreed with UNDP. 39 cases of children with 5 years of education are not considered as they are younger than 10 years old (7,8&9).
```

```
lab var edueyears "Total number of years of education accomplished"
```

```
ta edueyear eduhighyear,m
```

```
gen    years_edu5 = (edueyears>=5)
replace years_edu5 = . if edueyears==.
```

//Following a control variable is created on whether there is information on years of education for at least 2/3 of the household members – this is the no_missing_edu variable//

```
gen    temp = (edueyears~=.)
bys    hh_id: egen no_missing_edu = sum(temp)
by hh_id: gen temp2 = sum(age>10 & age!=.)
by hh_id: egen hhs2 = max(temp2)
label var hhs2 "Household size aged 10 or older"
replace no_missing_edu = no_missing_edu/hhs2
replace no_missing_edu = (no_missing_edu>=2/3)
drop   temp temp2
```

```
bys    hh_id: egen hh_years_edu5 = max(years_edu5)
replace hh_years_edu5 = . if hh_years_edu5==0 & no_missing_edu==0
replace hh_years_edu5 = . if hh_id==. //we exclude non-usual residents//
```

//The final variable is missing if the household has less than 2/3 of members with info on years and for the ones for which it has info, it is less than 5 years//

```
lab var hh_years_edu5 "Household has at least one member with 5 years of edu"
```

*** The indicator: it takes value 1 if at least someone in the hh has reported 5 years of edu or more, and 0 if for those hh for which at least 2/3 of the members reported years, no one has 5 years or more. The indicator has a missing value when there is missing info on years of edu for 2/3 or more of the hh members with no one reporting 5 years of education, or when all household members have missing information on years of education.

*** Ultrapoverty indicator: nobody completed at least 1 year of schooling

```
gen    years_edu1 = (edueyears>=1)
replace years_edu1 = . if edueyears==.
```

```

bys hh_id: egen hh_years_edu1 = max(years_edu1)
replace hh_years_edu1 = . if hh_years_edu1==0 & no_missing_edu==0
replace hh_years_edu1 = . if hh_id==. //we exclude non-usual residents//
lab var hh_years_edu1 "Household has at least one member with 1 year of edu"

```

```

*****
*****

```

*** Step 2.2 Child School Attendance

```

*****

```

```

*****
*****

```

//The entire household is considered deprived if any school-aged child is not attending school up to class 8. Data Source for age children start school: United Nations Educational, Scientific and Cultural Organization, Institute for Statistics database, Table 1. Education systems [UIS, <http://stats.uis.unesco.org/unesco/TableViewer/tableView.aspx?ReportId=163>]//

//Note that MICS may have more than one variable for school attendance. Use the corresponding line below, according to the variable you have. Please check that it corresponds 1=attending, 0=not attending//

*Ed5 questions about formal school

```

gen attendance = .
replace attendance = 0 if ed5==2 //Not currently attending//
replace attendance = 1 if ed5==1 //Currently attending//
replace attendance = . if age<5 | age>24 | ed5==. | ed5==9
replace attendance = 0 if ed3==2 //Never attended//

```

//IMPORTANT: Please change the age range according to the compulsory schooling age of the particular country. You will need to check this in <http://stats.uis.unesco.org/unesco/TableViewer/tableView.aspx?ReportId=163>. Look at the starting age and add 8 (regardless of the finalizing compulsory age in the country. So for example, if the compulsory age is 5-11 years, your age range will be 5-13 (=5+8) years.//

///p.12 table HH5 reports attendance from 5. UIS report 5 as the starting school age for primary, the dataset report attendance from 5. I decided to take 5 as starting school age.

```

gen child_schoolage = (age>=5 & age<=13) // According to the information in the report, it coincides with UNESCO UIS

```

```

bys hh_id: egen hh_children_schoolage = sum(child_schoolage)

```

```
replace hh_children_schoolage = (hh_children_schoolage>0) //Control variable: It takes value 1 if
the household has children in school age//
```

```
gen child_not_atten = (attendance==0) if child_schoolage==1
replace child_not_atten = . if attendance==. & child_schoolage==1
bys hh_id: egen any_child_not_atten = max(child_not_atten)
gen hh_all_child_atten = (any_child_not_atten==0)
replace hh_all_child_atten = . if any_child_not_atten==.
replace hh_all_child_atten = 1 if hh_children_schoolage==0
tab hh_all_child_atten
```

//The indicator takes value 1 if ALL children in school age are attending school and 0 if there is at least one child not attending. Households with no children receive a value of 1 as non-deprived. The indicator has a missing value only when there are all missing values on children attendance in households that have children in school age. Please create a missing variable only if ed4 is all missing//

```
*gen hh_all_child_atten=.
```

```
label var hh_children_schoolage "Household has children in school age"
```

```
label var hh_all_child_atten "Household has all school age children in school"
```

* Ultrapoverty indicator

```
gen child_schoolage_6 = (age>=5 & age<=11)
bys hh_id: egen hh_children_schoolage_6 = sum(child_schoolage_6)
replace hh_children_schoolage_6 = (hh_children_schoolage_6>0)
lab var hh_children_schoolage_6 "Household has children in school age (6 years of school)"
```

```
gen child_atten_6 = (attendance==1) if child_schoolage_6==1
replace child_atten_6 = . if attendance==. & child_schoolage_6==1
bys hh_id: egen any_child_atten_6 = max(child_atten_6)
gen hh_all_child_atten_6 = (any_child_atten_6==1)
replace hh_all_child_atten_6 = . if any_child_atten_6==.
replace hh_all_child_atten_6 = 1 if hh_children_schoolage_6==0
```

```
lab var hh_all_child_atten_6 "Household has at least one school age children (6 years of school) in school"
```

```
*****
```

*** Step 2.3 Nutrition

```
*****
```

```
//The entire household is considered deprived if any adult or child for whom there is nutritional information is malnourished in the household. Adults are considered undernourished if their BMI is below 18.5 m/kg2. Children are considered malnourished if their z-score of weight-for-age is below minus two standard deviations from the median of the reference population. Alternative estimations were performed using stunting and wasting. For further details please see Alkire and Santos (2010).//
```

*** Low BMI of mother or daughter

```
//The BMI of the mother/daughter is not captured in Nepal MICS 2014
```

```
gen hh_no_low_bmi = .
```

```
gen f_low_bmi = .
```

```
gen m_low_bmi = .
```

*** Household Child Undernutrition Dummy ***

```
//Households with no eligible children to be measured will receive a value of 1//
```

```
bys hh_id: egen temp = max(underwa2)
```

```
gen hh_no_underwa2 = (temp==0) //Takes value 1 if no child in the hh is underweighted, 0 if at least one is//
```

```
replace hh_no_underwa2 = . if temp==.
```

```
drop temp
```

```
lab var hh_no_underwa2 "Household has no child under weight-for-age – 2 stdev"
```

Now we replace the children nutrition indicators with 1 for the households that had no children eligible

```
replace hh_no_underwa2 = 1 if no_child_eligible==1
```

```
//NOTE that hh_no_underwh2 takes value 1 if: (a) no any eligible children in the hh is undernourished or (b) there are no eligible children in the hh. The variable takes values 0 for those
```


households that have at least one measured child undernourished. The variable has missing values only when there is missing info in nutrition for ALL eligible children in the household//

* Ultrapoverty indicator

```
bys hh_id: egen temp = max(underwa3)
```

```
gen hh_no_underwa3 = (temp==0) //Takes value 1 if no child in the hh is underweighted, 0 if at least one is//
```

```
replace hh_no_underwa3 = . if temp==.
```

```
replace hh_no_underwa3 = 1 if no_child_eligible==1
```

```
lab var hh_no_underwa3 "Household has no child under weight-for-age - 3 stdev"
```

```
drop temp
```

****Finally we create the nutrition indicator to be used in the MPI: hh_nutrition

//The indicator takes value 1 if there is no undernourished adult or children. It also takes value 1 for the households that had no eligible women AND no eligible children. The indicator takes value 0 if any adult or child for whom there is nutritional information is undernourished in the household. The indicator takes value missing "." only if all eligible women and eligible children have missing information in their respective nutrition variable. If the nutritional variable is missing altogether in the dataset, this indicator will not be included in the MPI and the mortality indicator will receive the full health weight//

```
gen hh_nutrition = hh_no_underwa2
```

//If your nutritional variable has observations only in the children nutrition variable (as happens in all MICS as far as we know), label as//

```
label var hh_nutrition "Household has no child undernourished (weight-for-age)"
```

* Replacement for household without eligible children

//The indicator takes value 1 for the households that had no eligible children.//

```
replace hh_nutrition = 1 if no_child_eligible==1
```

//If there is no nutritional information, then we create an empty variable//

```
*gen hh_nutrition=.
```

* Ultrapoverty indicator

```
gen      hh_nutrition_3_17 = hh_no_underwa3
```

```
replace hh_nutrition_3_17 = 1 if no_child_eligible==1
```

```
lab var hh_nutrition_3_17 "Household has no child undernourished (weight-for-age)"
```

*** Step 2.4 Child Mortality *****

//The entire household is considered deprived if any child has died in the family. Mortality at any age was considered since this is the information available in MICS datasets. For further details, see: Alkire and Santos (2010)//

```
egen child_mortality = rowtotal (cm9a cm9b), missing //cm9a: number of sons who have died,  
cm9b: number of daughters who have died//
```

```
replace child_mortality = 0 if cm1==2 | cm8==2  //For consistency control; cm1: ever had children;  
cm8: ever had child who later died//
```

```
bys ma1: ta ma5 cm1,m
```

*The question was asked to all women, so the 2 following lines are not needed

*replace child_mortality= 0 if ma1==2 // Never married

*replace child_mortality = 0 if ma5==2 // Replacing as non-deprived those women who are not married or have not lived with a men // Even though, the questionnaire ask child mortality questions to all women, 3,459 observations had missing values in cm1 aspect that is related to answer never married and no currently married (ma1 and ma5)

```
tab child_mortality,m
```

```
label var child_mortality "Occurrence of child mortality in the household"
```

```
bys hh_id: egen temp = max(child_mortality)
```

```
gen hh_no_child_mortality = (temp==0)
```

```
replace hh_no_child_mortality = . if temp==.
```

```
label var hh_no_child_mortality "Household had no child mortality"
```

```
replace hh_no_child_mortality = 1 if no_eligibles==1
```

//The final indicator takes value 1 if the household was free of child mortality and 0 if at least one children died – according to the Women Level recode. It is missing if there was missing info on both sons and daughters (cm9_a and cm9_b). Households with no children/no women being interviewed receive a value of 1//

* Ultrapoverty indicator: at least 2 children or more died in the hh

```
egen temp_f = rowtotal(cm9a cm9b), missing
```

```
replace temp_f = 0 if cm1 == 2 | cm8==2
```

```
bys hh_id: egen child_mortality_f = sum(temp_f), missing
```

```
/*egen temp_m = rowtotal(mcm9a mcm9b), missing
```

```
replace temp_m = 0 if mcm1 == 2 | mcm8==2
```

```
bys hh_id: egen child_mortality_m = sum(temp_m), missing
```

```
*/
```

```
egen child_mortality_2 = rowmax(child_mortality_f)
```

```
gen hh_no_child_mortality_2 = (child_mortality_2 < 2)
```

```
replace hh_no_child_mortality_2 = . if child_mortality_2 ==.
```

```
replace hh_no_child_mortality_2 = 1 if no_eligibles == 1
```

```
lab var hh_no_child_mortality_2 "Household has less than two children died"
```

```
tab temp child_mortality_2, m
```

```
tab hh_no_child_mortality*, m
```

*Under-five mortality

```
bys hh_id: egen child_mortality_uf = sum(tdufive), missing
```

```
gen hh_no_child_mortality_uf = (child_mortality_uf == 0) if child_mortality_uf != .
```

```
gen hh_no_child_mortality_uf_2 = (child_mortality_uf < 2) if child_mortality_uf != .
```

```
replace hh_no_child_mortality_uf = 1 if no_fem_eligible == 1
```

```
replace hh_no_child_mortality_uf_2 = 1 if no_fem_eligible == 1
```

```
lab var hh_no_child_mortality_uf "Household had no children under five dead"
```

```
lab var hh_no_child_mortality_uf_2 "Household has less than two children under five dead"
```

```
tab tdufive child_mortality_2, m // both measures disagree in 2 or more cases of child mortality  
that come from different eligible women.
```

```
tab tdufive child_mortality, m // both measures disagree in 2 or more cases of child mortality that  
come from different eligible women.
```

```
tab tdufive child_mortality_uf, m // both measures disagree in 2 or more cases of child mortality  
that come from different eligible women.
```

```
bys hh_n_fem_eligible: ta tdufive if cm1==1, m
```

```
drop temp_* child_mortality_f child_mortality_2 temp
```

```
*****
```

```
*** Step 2.5 Electricity
```

```
*****
```

```
//Members of the household are considered deprived if the household has no electricity //
```

```
tab hc8a, m
```

```
clonevar electricity = hc8a
```

```
replace electricity = 0 if electricity==2 // Now yes electricity =1, no electricity =0//
```

```
replace electricity = . if electricity==9 //Please check that missing values remain missing //
```

```
label define lab_yes_no 0 "no" 1 "yes" //Deprived if no electricity //
```

```
label var electricity "Electricity"
```

```
label values electricity lab_yes_no
```

```
tab electricity, m
```

```
* Standard MPI indicator = Ultrapoverty indicator
```

```
gen electricity_ultra = electricity
```

```
*****
```

```
*** Step 2.6 Improved Sanitation
```

//Members of the household are considered deprived if the household's sanitation facility is not improved, according to MDG guidelines, or if it is improved but shared with other household. //

//Following the definition of the MDG indicators: "A household is considered to have access to improved sanitation if it uses: Flush or pour flush to piped sewer //

//system, septic tank or pit, latrine; Pit latrine with slab; Composting toilet; Ventilated improved pit latrine. The excreta disposal system is considered improved if it is private or shared by a reasonable number of households." Source: "The Challenge of Slums: Global Report on Human Settlements 2003 (Revised version, April 2010), Chapter 1, p. 16.

<http://www.unhabitat.org/pmss/listItemDetails.aspx?publicationID=1156>"

clonevar toilet = ws8 //Save the original variable//

codebook toilet, tab(20) //Check coding//

clonevar shared_toilet = ws9

replace shared_toilet = 0 if shared_toilet==2 //0=no 1=yes//

replace shared_toilet = . if shared_toilet==9

gen toilet_mdg = (toilet<=13 | (toilet >=15 & toilet<=22) | toilet==31) & shared_toilet!=1 //
Households have access to improved sanitation if: flush to piped sewer system, flush to septic tank.
flush to pit latrine; flush to unknown place/not sure/DK where; to ventialed improved pit latrine; pit
latrine with slab; composting toilet (report p.93 WS5).

replace toilet_mdg = 0 if (toilet==14 | toilet==23 | toilet==41 | toilet==95 | toilet==96) &
shared_toilet==1 //the report states a hanging toilet/latrine that is not in the dataset.

replace toilet_mdg = . if toilet==. | toilet==99

lab var toilet_mdg "Household has improved sanitation with MDG Standards"

* Ultrapoverty indicator

gen toilet_ultra = .

replace toilet_ultra = 0 if toilet==95 | toilet==96 //95 open defecation, 96 Other//

replace toilet_ultra = 1 if toilet!=95 & toilet!=96 & toilet!=. & toilet!=99

bys shared_toilet : ta toilet toilet_mdg [iw=weight],m nofre cell

*** Step 2.7 Safe Drinking Water

//Members of the household are considered deprived if the household does not have access to safe drinking water according to MDG guidelines, or safe drinking water is more than a 30-minute walk from home roundtrip.

// "A household has improved drinking water supply if it uses water from sources that include: piped water into dwelling, plot or yard; public tap/ stand pipe; tube well/borehole; protected dug well; protected spring; rain water collection. (...) Households using bottled water are only considered to be using improved water when they use water from an improved source for cooking and personal hygiene." Source: "The Challenge of Slums: Global Report on Human Settlements 2003 (Revised version, April 2010), Chapter 1, p. 16, 21.

<http://www.unhabitat.org/pmss/listItemDetails.aspx?publicationID=1156>

// "Access to safe water refers to the percentage of the population with reasonable access to an adequate supply of safe water in their dwelling or within a convenient distance of their dwelling. The Global Water Supply and Sanitation Assessment 2000 Report defines reasonable access as "the availability of 20 litres per capita per day at a distance no longer than 1,000 metres". "Indicators for Monitoring the Millennium Development Goals", p. 64-65. [As distance is not available, we convert the 1000 metres distance into 30 minutes. This is the DHS standard too.]//

clonevar water = ws1 //Save the original variable//

replace water = . if water==99

clonevar timetowater = ws4 //Save the original variable//

clonevar ndwater = ws2 //Non-drinking water//

***The criteria followed the national report page 84

gen water_mdg = 1 if water==11 | water==12 | water==13 | water==14 | water==21 |
water==31 | water==41 | water==51 | water==91 //Non deprived if water is "piped into dwelling",
"piped to yard/plot", "piped to neighborhood", "public tap/standpipe", "protected covered well", "well
or borehole to PMH" "protected spring", "rainwater" or "bottled water"//

replace water_mdg = 0 if water==32 | water==42 | water==61 | water==81 | water==96 //Deprived
if it is "Modern well non-covered", "traditional well protected", "traditional well unprotected"
"unprotected spring", "tanker-truck", "cart with small tank/drum" "surface water" or "other"//

replace water_mdg = 0 if water_mdg==1 & timetowater >= 30 & timetowater~= . &
timetowater~=998 & timetowater~=999 //Deprived if water is at more than 30 minutes' walk
(roundtrip). Please check the value assigned to 'in premises' and if this is different from 996 or 995,
add the condition: & timetowater~=XXX accordingly//

replace water_mdg = . if water==.

replace water_mdg = 0 if water==91 & ndwater==61 //Deprived if bottled water as the source of drinking water and unimproved non-drinking water. 83 cases use non-improved sources of water for other purposes.

lab var water_mdg "Household has drinking water with MDG standards (considering distance)"

* Ultrapoverty indicator

gen water_mdg_45 = .

replace water_mdg_45 = 1 if water==11 | water==12 | water==13 | water==14 | water==21 |
water==31 | water==41 | water==51 | water==91

replace water_mdg_45 = 0 if water==32 | water==42 | water==61 | water==81 | water==96

replace water_mdg_45 = 0 if water_mdg_45==1 & timetowater>45 & timetowater~= . &
timetowater~=998 & timetowater~=999

replace water_mdg_45 = . if water==.

replace water_mdg_45 = 0 if water==91 & ndwater==61

lab var water_mdg_45 "Household has drinking water with MDG standards (45 minutes distance)"

*** Step 2.8 Flooring

//Members of the household are considered deprived if the household has a dirt, sand or dung
floor//

clonevar floor = hc3 //Save the original variable//

gen floor_imp = 1

replace floor_imp = 0 if (floor<=12 | floor==96) //Deprived if "earth/sand", "dung" or "other"//

replace floor_imp = . if floor==. | floor==99 //Please check that missing values remain missing//

lab var floor_imp "Household has floor that it is not earth/sand/dung"

* Standard MPI indicator = Ultrapoverty indicator

gen floor_ultra = floor_imp

*** Step 2.9 Cooking Fuel

//Members of the household are considered deprived if the household cooks with solid fuels: wood, charcoal, crop residues or dung. "Indicators for Monitoring the Millennium Development Goals", p.63//

clonevar cookingfuel = hc6 //Save the original variable//

gen cooking_mdg = 1

replace cooking_mdg = 0 if (cookingfuel>=6 & cookingfuel!=95 & cookingfuel!=96)

replace cooking_mdg = . if cookingfuel==. | cookingfuel==99

lab var cooking_mdg "Househod has cooking fuel according to MDG standards"

//Non deprived if: 1"electricity", 2"LPG" 3"gas" 4"Biogas" 5"Kerosene" 95 "no food cooked in household" or 96"other" //not solid fuels according to p. 74, table CH1

//Deprived if: 6"coal/lignite" 7"charcoal", 8"wood", 9"straw/shrubs/grass" 10"animal dung" 11"agricultural crop" // The report does not mention if no food cooked in the household is considered or not as deprived. See pag 78 report

* Ultrapoverty indicator: similar to Cooking Fuel MDG but coal/lignite and charcoal are now not deprived

gen cooking_ultra = cooking_mdg

replace cooking_ultra = 1 if cookingfuel<=7

lab var cooking_ultra "Househod has cooking fuel according to MDG standards (charcoal and coal are not deprived"

*** Step 2.10 Assets ownership

//Members of the household are considered deprived if the household does not own more than one of: radio, TV, telephone, bike, motorbike or refrigerator and does not own a car or truck//

// Check that for all assets in living standards: "no"==0 and yes=="1"//

clonevar television = hc8c

*rename sh31bw_television //Note that in most MICS surveys only have the variable 'television', and not the black and white (except probably for Eastern Europe countries. If this is the case skip the line referring to black and white TV//

clonevar radio = hc8b

clonevar telephone = hc8d

clonevar mobiletelephone = hc9b

clonevar refrigerator = hc8e

clonevar car = hc9f //Car or truck//

clonevar bicycle = hc9c

clonevar motorbike = hc9d

//9 and 99 are replaced as missing values. Please check that 9, 99 and 8, 98 are missing or non-applicable//

foreach var in television radio telephone mobiletelephone refrigerator car bicycle motorbike {

replace `var' = 0 if `var'==2 //0=no; 1=yes//

replace `var' = . if `var'==9

}

//Skip the following lines if black and white tv or mobile phone were missing //

replace telephone = 1 if telephone==0 & mobiletelephone==1

replace telephone = 1 if telephone==. & mobiletelephone==1

*replace television=1 if television==0 & bw_television==1

*replace television=1 if television==. & bw_television==1

*** Combined Assets Indicator

egen n_small_assets = rowtotal(television radio telephone refrigerator bicycle motorbike), missing

lab var n_small_assets "Household Number of Small Assets Owned"

gen hh_assets = (car==1 | n_small_assets>1)

replace hh_assets = . if car==. & n_small_assets==.

```
lab var hh_assets "Household Asset Ownership: HH has car or more than 1 of small assets"
```

```
* Ultrapoverty indicator: only "No Assets" is deprived
```

```
gen    hh_assets_ultra = (car==1 | n_small_assets>0)
```

```
replace hh_assets_ultra = . if car==. & n_small_assets==.
```

```
lab var hh_assets_ultra "Household Asset Ownership: HH has car or 1 small assets"
```

```
** Quality check
```

```
count if child_CH==1 & no_child_eligible==0
```

```
mat childcount = r(N)
```

```
*egen tot_hh = sum(id==1)
```

```
egen tot_hh = sum(hl1==1)
```

```
gen i=1
```

```
egen tot_pop = sum(i)
```

```
drop i
```

```
gen distance = cond(timetowater>=30 & timetowater < 995,1,0)
```

```
gen bottledwater = cond(water==91,1,0)
```

```
foreach var in urban hh_nutrition tot_hh tot_pop hhs bottledwater distance{
```

```
sum `var' [w=weight] if no_child_fem_eligible==0
```

```
mat `var'=r(mean)
```

```
}
```

```
mat total=(urban, hh_nutrition, tot_hh, tot_pop, hhs, bottledwater, distance)
```

```
foreach var in radio telephone mobiletelephone television refrigerator bicycle motorbike car  
shared_toilet {
```

```
sum `var' [w=weight] if no_child_fem_eligible==0 & hl1==1
```

```
mat `var' = r(mean)
```

```
}
```

```
mat
assets=(radio,telephone,mobiletelephone,television,refrigerator,bicycle,motorbike,car,shared_toilet
)
```

```
foreach var in electricity toilet_mdg water_mdg water_floor_imp cooking_mdg hh_years_edu5
hh_all_child_atten hh_no_underwa2 hh_no_child_mortality {
proportion `var' [pw = weight] if no_child_fem_eligible==0 & hl1==1
mat `var'2=e(b)
mat `var'=`var'2[1,1]
mat drop `var'2
}
```

```
mat
qcheck=(electricity,toilet_mdg,water_mdg,floor_imp,cooking_mdg,hh_years_edu5,hh_all_child_att
en,childcount,hh_no_underwa2,hh_no_child_mortality,total,assets)

mat colnames qcheck = elect toil wat floor_imp cooking_mdg hh_years_edu5 hh_all_child_atten
childcount hh_no_underwa2 hh_no_child_mortality urban hh_nutrition tot_hh tot_pop hhs
bottledwater distance radio telephone mobiletelephone television refrigerator bicycle motorbike car
shared_toilet

svmat qcheck

outsheet qcheck* using "$workingfolder_out\quality_checks_npl_mics14.xls", replace

mat drop _all

drop qcheck*
```

*** Sampling design variables

```
clonevar strata = stratum
```

```
*psu already defined
```

```
codebook strata psu
```

*We keep info on the material of walls, roof and wealth index wscore windex5 wscoreu windex5u
wscorer windex5r

```
clonevar walls = hc5
```

```
clonevar roof = hc4
```

```
clonevar religion=hc1a
```

clonevar ethnicity=hc1c

clonevar language=hc1b

*Check of missing values: if larger than 5%, please check genuinely missing information rather than non-applicable information

mdesc hh_years_edu5 hh_all_child_atten hh_nutrition hh_no_child_mortality electricity toilet_mdg water_mdg floor_imp cooking_mdg hh_assets television radio telephone refrigerator bicycle motorbike car

*** Keep main variables require for MPI calculation

keep hh1 hh2 hl1 relationship sex age hl7b hl7 urban region weight country year survey hh_id ind_id wb2 wm7 cm1 cm8 cm9a cm9b cm10 ma1 ma5 women_WM ed3 ed4b ed5 ///

wmweight water ndwater ws3 timetowater toilet shared_toilet floor cookingfuel electricity radio television telephone refrigerator mobiletelephone bicycle ///

motorbike car hhsex psu strata ln chweight child_CH z_scorewa underwa2 underwa3 fem_eligible no_fem_eligible no_child_eligible no_child_fem_eligible ///

no_eligibles id hhs edulevel eduhighyear edueyears years_edu5 no_missing_edu hh_years_edu5 years_edu1 hh_years_edu1 attendance child_schoolage hh_children_schoolage ///

child_not_atten any_child_not_atten hh_all_child_atten child_schoolage_6 hh_children_schoolage_6 child_atten_6 any_child_atten_6 hh_all_child_atten_6 ///

hh_no_low_bmi f_low_bmi m_low_bmi hh_no_underwa2 hh_no_underwa3 hh_nutrition hh_nutrition_3_17 child_mortality hh_no_child_mortality ///

hh_no_child_mortality_2 electricity_ultra toilet_mdg toilet_ultra water_mdg water_mdg_45 floor_imp floor_ultra ///

cooking_mdg cooking_ultra n_small_assets hh_assets hh_assets_ultra tot_hh tot_pop distance bottledwater hhs2 ///

stunting* wasting* *_* tdufive *uf* walls roof wscore windex5 wscoreu windex5u wscorer windex5r ta1-ta17 religion ethnicity language hc1a hc1b hc1c cl2a-cl12 ///

*** Order file

order hh1 hh2 hl1 relationship sex age hl7b hl7 urban region weight country year survey hh_id ind_id wb2 wm7 cm1 cm8 cm9a cm9b cm10 ma1 ma5 women_WM ed3 ed4b ed5 ///

wmweight water ndwater ws3 timetowater toilet shared_toilet floor cookingfuel electricity radio television telephone refrigerator mobiletelephone bicycle ///

```
motorbike car hhsex psu strata ln chweight child_CH z_scorewa underwa2 underwa3 fem_eligible
no_fem_eligible no_child_eligible no_child_fem_eligible ///
```

```
no_eligibles id hhs edulevel eduhighyear edueyears years_edu5 no_missing_edu hh_years_edu5
years_edu1 hh_years_edu1 attendance child_schoolage hh_children_schoolage ///
```

```
child_not_atten any_child_not_atten hh_all_child_atten child_schoolage_6
hh_children_schoolage_6 child_atten_6 any_child_atten_6 hh_all_child_atten_6 ///
```

```
hh_no_low_bmi f_low_bmi m_low_bmi hh_no_underwa2 hh_no_underwa3 hh_nutrition
hh_nutrition_3_17 child_mortality hh_no_child_mortality ///
```

```
hh_no_child_mortality_2 electricity_ultra toilet_mdg toilet_ultra water_mdg water_mdg_45
floor_imp floor_ultra ///
```

```
cooking_mdg cooking_ultra n_small_assets hh_assets hh_assets_ultra tot_hh tot_pop distance
bottledwater ///
```

```
sort ind_id
```

```
save "$workingfolder_out\npl_mics14_pov.dta", replace //Save a copy of the prepared dataset//
```

```
*****
*****
```

```
*** Step3 MPI Calculation
```

```
*** Weights, Poverty cut-off (k), Aggregation (MPI, H, A, Vulnerable to Poverty, Severe Poverty ),
raw
```

```
*** headcount, censored headcount and decomposition by indicator
```

```
*****
*****
```

```
*** This section follows: Alkire, Sabina and James Foster (2006) Counting and multidimensional
```

```
*** poverty measurement, Journal of Public Economics, vol.95, issue 7-8, pages 476-487.
```

```
*****
*****
```

```
use "$workingfolder_out\npl_mics14_pov.dta"
```

```
//Final check to see the total number of missing values you have for each variable. Variables should
not have at this stage high proportion of missing. The command might need to be installed: write
findit mdesc in the command window, and install it//
```

mdesc hh_years_edu5 hh_all_child_atten hh_nutrition hh_no_child_mortality electricity toilet_mdg
water_mdg floor_imp cooking_mdg hh_assets television radio telephone refrigerator bicycle
motorbike car

*** Define Sample Weight and total Population

gen sample_weight = weight

*** List with the 10 indicators included in the MPI

local varlist_pov hh_years_edu5 hh_all_child_atten hh_no_child_mortality hh_nutrition electricity
toilet_mdg water_mdg floor_imp cooking_mdg hh_assets

local varlist_pov_ultra hh_years_edu1 hh_all_child_atten_6 hh_no_child_mortality_2
hh_nutrition_3_17 electricity_ultra toilet_ultra water_mdg_45 cooking_ultra floor_ultra
hh_assets_ultra

*** List of sample without missing values the MPI

gen sample = (hh_years_edu5~= . & hh_all_child_atten~= . & hh_no_child_mortality~= . &
hh_nutrition~= . & electricity~= . & toilet_mdg~= . & water_mdg~= . & floor_imp~= . & cooking_mdg~= .
& hh_assets~= .) //If your country has information on usual residents, you should only include in the
sample those individuals who are de jure members of the household//

*** Percentage sample after/before dropping missing values

sum sample [iw = sample_weight]

gen per_sample_weighted = r(mean)

sum sample

gen per_sample = r(mean)

*** Define deprivation matrix 'g0'

*** which takes values 1 if the individual is deprived in the particular

*** indicator according to deprivation cutoff z as defined during step 2

```
foreach var of varlist `varlist_pov' {  
    gen    g0_`var' = 1 if `var'==0  
    replace g0_`var' = 0 if `var'==1  
}
```

```
foreach var of varlist `varlist_pov_ultra' {  
    gen    g0_u_`var' = 1 if `var'==0  
    replace g0_u_`var' = 0 if `var'==1  
}
```

*** Raw Headcounts

```
foreach var of varlist `varlist_pov' {  
    sum    g0_`var' if sample==1 [iw = sample_weight]  
    gen    raw_H_`var' = r(mean)*100  
    lab var raw_H_`var' "Raw Headcount: Percentage of people who are deprived in ..."  
}
```

```
foreach var of varlist `varlist_pov_ultra' {  
    sum    g0_u_`var' if sample==1 [iw = sample_weight]  
    gen    raw_H_u_`var' = r(mean)*100  
    lab var raw_H_u_`var' "Raw Headcount (Ultra): Percentage of people who are deprived in  
..."  
}
```

*** Define vector 'w' of dimensional and indicator weights

//If survey lacks one or more indicators, weights need to be adjusted so they add up to the total number of indicators.

// DIMENSION EDUCATION

```
foreach var in hh_years_edu5 hh_all_child_atten {  
    gen w_`var' = 1/6  
}
```

// DIMENSION HEALTH

```
foreach var in hh_no_child_mortality hh_nutrition {  
    gen w_`var' = 1/6  
}
```

// DIMENSION LIVING STANDARD

```
foreach var in electricity toilet_mdg water_mdg floor_imp cooking_mdg hh_assets {  
    gen w_`var' = 1/18  
}
```

// DIMENSION EDUCATION ultra

```
foreach var in hh_years_edu1 hh_all_child_atten_6 {  
    gen w_u_`var' = 1/6  
}
```

// DIMENSION HEALTH ultra

```
foreach var in hh_no_child_mortality_2 hh_nutrition_3_17 {  
    gen w_u_`var' = 1/6  
}
```



```
// DIMENSION LIVING STANDARD ultra
```

```
foreach var in electricity_ultra toilet_ultra water_mdg_45 cooking_ultra floor_ultra hh_assets_ultra  
{
```

```
    gen w_u_`var' = 1/18
```

```
}
```

```
*****
```

```
*** Generate the weighted deprivation matrix 'w' * 'g0'
```

```
*****
```

```
foreach var of varlist `varlist_pov' {
```

```
    gen w_g0_`var' = w_`var' * g0_`var'
```

```
    replace w_g0_`var' = . if sample!=1 //The estimation is based only on observations that have  
non-missing values for all variables in varlist_pov and on usual residents only//
```

```
}
```

```
foreach var of varlist `varlist_pov_ultra' {
```

```
    gen w_u_g0_`var' = w_u_`var' * g0_u_`var'
```

```
    replace w_u_g0_`var' = . if sample!=1 //Estimation based only on obs that have non-missing  
values for all variables in varlist_pov and on usual residents only//
```

```
}
```

```
*****
```

```
*** Generate the vector of individual weighted deprivation count 'c'
```

```
*****
```

```
egen c_vector = rowtotal(w_g0_*)
```

```
replace c_vector = . if sample!=1
```

```
drop w_g0_*
```

```
egen    c_vector_u = rowtotal(w_u_g0_*)
```

```
replace c_vector_u = . if sample!=1
```

```
drop    w_u_g0_*
```

```
sort ind_id
```

```
save "$workingfolder_out\npl_mics14_pov.dta", replace //Save a copy of the prepared dataset//
```

```
*****
```

```
*** Identification step according to poverty cutoff  $k \geq 1/3$ 
```

```
*****
```

```
gen     multidimensionally_poor = (c_vector>=0.3333)
```

```
replace multidimensionally_poor = . if sample!=1 //Takes value 1 if individual is mult. poor//
```

```
gen     multidimensionally_ultra_poor = (c_vector_u>=0.3333)
```

```
replace multidimensionally_ultra_poor = . if sample!=1 //Takes value 1 if individual is mult. poor//
```

```
*****
```

```
*** Generate the censored vector of individual weighted deprivation count 'c(k)'
```

```
*****
```

```
gen     c_censored_vector = c_vector
```

```
replace c_censored_vector = 0 if multidimensionally_poor==0 //Provide a score of zero if a person is not poor//
```

```
gen     c_censored_vector_u = c_vector_u
```

```
replace c_censored_vector_u = 0 if multidimensionally_ultra_poor==0 //Provide a score of zero if a person is not poor//
```

*** Define censored deprivation matrix 'g0(k)'

foreach var of varlist `varlist_pov' {

 gen g0_k_`var' = g0_`var'

 replace g0_k_`var' = 0 if multidimensionally_poor==0

 replace g0_k_`var' = . if sample!=1

}

foreach var of varlist `varlist_pov_ultra' {

 gen g0_u_k_`var' = g0_u_`var'

 replace g0_u_k_`var' = 0 if multidimensionally_ultra_poor==0

 replace g0_u_k_`var' = . if sample!=1

}

*** Generates Multidimensional Poverty Index (MPI), Headcount (H) and Intensity of Poverty (A)

keep if sample==1

*** Multidimensional Poverty Index (MPI)

sum c_censored_vector [iw = sample_weight]

gen MPI = r(mean)

lab var MPI "Multidimensional Poverty Index (MPI = H*A): Range 0 to 1"

*** Multidimensional Poverty Index Ultra

sum c_censored_vector_u [iw = sample_weight]

```
gen    MPI_u = r(mean)
```

```
lab var MPI_u "Multidimensional Poverty Index Ultra (MPI_u = H*A): Range 0 to 1"
```

```
*** Headcount (H)
```

```
sum    multidimensionally_poor [iw = sample_weight]
```

```
gen    H = r(mean)*100
```

```
lab var H "Headcount ratio: % Population in multidimensional poverty (H)"
```

```
sum    multidimensionally_ultra_poor [iw = sample_weight]
```

```
gen    H_u = r(mean)*100
```

```
lab var H_u "Headcount ratio: % Population in multidimensional poverty ultra (H_u)"
```

```
*** Intensity of Poverty (A)
```

```
sum    c_censured_vector [iw = sample_weight] if multidimensionally_poor==1
```

```
gen    A = r(mean)*100
```

```
lab var A "Intensity of deprivation among the poor (A): Average % of weighted deprivations"
```

```
sum    c_censured_vector_u [iw = sample_weight] if multidimensionally_ultra_poor==1
```

```
gen    A_u = r(mean)*100
```

```
lab var A_u "Intensity of deprivation among the poor - ultra (A_u): Average % of weighted deprivations"
```

```
*** Population vulnerable to poverty (who experience 20-32.9% intensity of deprivations)
```

```
gen    temp = 0
```

```
replace temp = 1 if c_vector>=0.2 & c_vector<0.3299
```

```
replace temp = . if sample!=1
```

```
sum    temp [iw = sample_weight]
```

```
gen    vulnerable = r(mean)*100
```

```
lab var vulnerable "% Population vulnerable to poverty (who experience 20-32.9% intensity of deprivations)"
```

```
drop temp
```

```
gen temp = 0
```

```
replace temp = 1 if c_vector_u>=0.2 & c_vector_u<0.3299
```

```
sum temp [iw = sample_weight]
```

```
gen vulnerable_u = r(mean)*100
```

```
lab var vulnerable_u "% Population vulnerable to poverty (Ultra) (who experience 20-32.9% intensity of deprivations)"
```

```
drop temp
```

```
*** Population in severe poverty (with intensity 50% or higher)
```

```
gen temp = 0
```

```
replace temp = 1 if c_vector>0.49
```

```
replace temp = . if sample!=1
```

```
sum temp [iw = sample_weight]
```

```
gen severe = r(mean)*100
```

```
lab var severe "% Population in severe poverty (with intensity 50% or higher)"
```

```
drop temp
```

```
gen temp = 0
```

```
replace temp = 1 if c_vector_u>0.49
```

```
sum temp [iw = sample_weight]
```

```
gen severe_u = r(mean)*100
```

```
lab var severe_u "% Population in severe poverty (Ultra) (with intensity 50% or higher)"
```

```
drop temp
```

```
*** Censored Headcount
```

```

foreach var of varlist `varlist_pov' {
    sum    g0_k_`var' [iw = sample_weight]
    gen    cen_H_`var' = r(mean)*100
    lab var cen_H_`var' "Censored Headcount: Percentage of people who are poor and deprived
in ..."
}

```

```

foreach var of varlist `varlist_pov_ultra' {
    sum    g0_u_k_`var' [iw = sample_weight]
    gen    cen_H_u_`var' = r(mean)*100
    lab var cen_H_u_`var' "Censored Headcount (Ultra): Percentage of people who are poor and
deprived in ..."
}

```

*** Dimensional Contribution

```

foreach var of varlist `varlist_pov' {
    gen    cont_`var' = (w_`var' * cen_H_`var')/MPI
    lab var cont_`var' "% Contribution in MPI of indicator..."
}

```

```

foreach var of varlist `varlist_pov_ultra' {
    gen    cont_u_`var' = (w_u_`var' * cen_H_u_`var')/MPI_u
    lab var cont_u_`var' "% Contribution in MPI (Ultra) of indicator..."
}

```

*** Prepare results to export: *_povest.dta files

```

keep country year survey per_sample_weighted per_sample MPI H A vulnerable severe raw_*
cen_* cont_*

```

```

order country year survey per_sample_weighted per_sample MPI H A vulnerable severe raw_*
cen_* cont_*

```

```
gen temp = (_n)
```

```
keep if temp==1
```

```
drop temp
```

```
codebook, compact
```

```
save "$workingfolder_out\npl_mics14_povest.dta", replace //Save a copy of the prepared dataset//
```

```
log close
```

```
clear all
```

```
exit
```