Subject: Re: Next IPUMS-DHS Release
Posted by Yawo on Tue, 19 May 2020 16:09:37 GMT
View Forum Message <> Reply to Message

Liz:

Good afternoon. I hope you are doing well in the midst of this pandemic.

I am following up with your suggestions re creating an integrated regionids. I did but got stumped towards the last section where I am supposed to attach labels to the regionids. Unfortunately, the labels were not created.

```
* This look attaches the proper region labels to regionid.
sort subnational
lab def subnational_gen 1 "temp"
levelsof(idregion), local(levels)

foreach l of local levels {
 gen temp = ""
 replace temp = region_label_gen if idregion == `l'
 levelsof(temp), local(templabel)
 lab def reg_label_gen `l' `templabel', modify
 drop temp
}
label values idregion reg_label_gen
label variable idregion "Subnational regions"
```

STATA did not give any error, and neither was the value label in the list of value labels (under Data - Data Utilities - Label Utilities)].

I can attach the dataset if you require.

Thanks in advance for your assistance.

CY

boyle014 wrote on Mon, 24 June 2019 12:04The men's data are now available through IPUMS DHS.

Regions are easy if you're comparing one country over time. The IPUMS DHS integrated geography variables, described by Dr. King, will work well for you.

If you want unique labels for regions across countries, that's trickier. The process with the IPUMS data is similar to the process with the original DHS files. There's no way to make it simpler because the IPUMS folks cannot know which surveys or survey years researchers will want to use.

Here's some Stata code that will work with your IPUMS DHS data file to apply region labels to multiple surveys. The notes provide the information on how to tailor the code to your specific data.

I believe you are looking over time and across countries, so you'll probably want to use the integrated geography variables. Other researchers, who are comparing across only the most recent surveys, will want to use the single-survey geography variables.

```
* Create variable ct that is a string of the country value (e.g., Nigeria, Senegal)
decode(country), gen(ct)
levelsof ct, local(ctstring)

* Good time for a reminder: Do not save changed data in Stata. Always keep your data in
* its original form.
* (If you accidentally change your data, you can retrieve the original
* file under MY DATA on the IPUMS DHS home page. Find the file; click Revise; then
* click Submit Data Extract.)

* Create a temporary variable that combines all the country-specific region codes.
* For this to work, you must only have one geo_ variable for each country. In this
* example, I drop the less specific geography (see paragraph above).

drop geo_ke2014 geo_rw2014
egen region_temp = rowmax(geo_*)

* Use the temporary variable you created above, plus the country variable, to
* create a unique number for each region in your pooled dataset.
gen subnational = country*100 + region_temp

* The next command creates string variables for each geographic code.
* Find the geo_ variables in your variable list. Substitute your first and last
* geo variables for geoalt_ke2014 and geo_ug2016, respectively:
foreach var of varlist geoalt_ke2014-geo_ug2016 {
 decode `var', gen(`var'str)
}
* The following code create idregion, a single variable with values for every sample.
egen region_label_t_gen = concat(geo*str)
gen region_label_gen = ct + " " + substr(region_label_t_gen,1,30)
egen idregion = group(region_label_gen)

* This look attaches the proper region labels to regionid.
sort subnational
```

```
lab def subnational_gen 1 "temp"
levelsof(idregion), local(levels)

foreach l of local levels {
 gen temp = ""
 replace temp = region_label_gen if idregion == `l'
 levelsof(temp), local(templabel)
 lab def reg_label_gen `l' `templabel', modify
 drop temp
}
label values idregion reg_label_gen
label variable idregion "Subnational regions"

* Get rid of the temporary variables
drop subnational geo*str region_temp region_label_t_gen region_label_gen
```