
Subject: Next IPUMS-DHS Release

Posted by [Yawo](#) on Tue, 30 Apr 2019 19:22:23 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello,

I am interested in giving your website a try. But some of the data are not available. For example, Chad is not

Also, when do you plan to release the male DHS datasets?

Thanks, Yy

Subject: Re: Next IPUMS-DHS Release

Posted by [mking](#) on Tue, 30 Apr 2019 20:18:15 GMT

[View Forum Message](#) <> [Reply to Message](#)

We will release DHS data for men through IPUMS DHS by the end of May.

IPUMS DHS is created by a small research team with funding from the National Institutes of Health, and we are adding more samples and countries three times a year, as quickly as we can. I expect that all currently available public African standard DHS samples will be released through IPUMS DHS by early 2020. We are working on Chad data now. We are currently funded to harmonize standard DHS samples from Africa, the Middle East, and South Asia, and we will soon apply for additional funding to cover more regions of the world.

If the country or countries you need are not currently available, check back again.

Miriam King
IPUMS-DHS Project Manager

Subject: Re: Next IPUMS-DHS Release

Posted by [Yawo](#) on Wed, 01 May 2019 18:31:52 GMT

[View Forum Message](#) <> [Reply to Message](#)

Thanks very much, Miriam: I understand your constraints and I am looking forward to your next release.

But I have a question about the appending process.

When I append multiple rounds/surveys, all my value labels are messed up. For example, the "region" variable seemed to include only the value labels from the last data appended. So, if country A has regions 1 2 3, and country B has regions 3 4 5, I would expect the appended data to include all 6 regions. But in my case, only regions 3 4 5 are populated.

Do you have any hints, strategies to synchronize the value labels, given your experience?

Than, Yawo

Subject: Re: Next IPUMS-DHS Release

Posted by [kingx025](#) on Wed, 01 May 2019 19:52:53 GMT

[View Forum Message](#) <> [Reply to Message](#)

I suggest that you use IPUMS-DHS to avoid the frustration you experience when working with the original DHS files, with labels varying across samples. We recode the data from the original DHS files so that the same meaning is given the same code and label across samples, and you don't have to deal with the issue you describe.

For example, when you are working with multiple samples for a country, choose the "Integrated geography" variables from the drop down menu in IPUMS-DHS:

https://www.idhsdata.org/idhs-action/variables/group?id=geog_integ

This integrated region variable includes as many samples as possible and has consistent codes identifying areas with the same geographic footprint across all samples included for a country. There may be less detail than in the single sample region variables but there will be comparability in codes, labels, and meanings for the GEO_ integrated variables.

Our data harmonization is designed to insure that the same variable meaning has the same codes and labels across samples; that is why we integrate the data. We also release single-sample region variables for all samples, because they sometimes include more detail than is available in the integrated geographic variable, but most other variables are integrated across multiple sample years (and, apart from a few variables dealing with geography and ethnicity, across countries).

Miriam King
IPUMS-DHS Project Manager

Subject: Re: Next IPUMS-DHS Release

Posted by [boyle014](#) on Mon, 24 Jun 2019 16:04:04 GMT

[View Forum Message](#) <> [Reply to Message](#)

The 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
```

Subject: Re: Next IPUMS-DHS Release

Posted by [Yawo](#) on Thu, 27 Jun 2019 05:09:52 GMT

[View Forum Message](#) <> [Reply to Message](#)

Thanks. I will try it out. I have since changed my research question to focus on most recent surveys and extracted the single-survey geography variables. Do I just substitute these for the integrated version in the STATA code below?

Secondly, I asked the question about "Region" because I was going to use it to create unique stratum codes (egen stratum=group(survey v024 v0025)) for pooled data across countries/years using the original DHS data, consistent with advice given by Dr. Pullum and others on this board.

But now that I am using the IPUMS data (which has its own stratum/IPU customized variables), do I still need to go through this process if I don't need the "region" variables? Wouldn't I just svyset my data using the following:

```
svyset [pw=perweight], psu(idhspsu) strata(idhsstrata)., without harmonizing region codes/values?
```

thanks - Yawo

Subject: Re: Next IPUMS-DHS Release

Posted by [boyle014](#) on Thu, 27 Jun 2019 17:26:16 GMT

[View Forum Message](#) <> [Reply to Message](#)

Quote:Do I just substitute these for the integrated version in the STATA code below?
That's correct.

As you note, you don't need to construct unique strata variables; they are already in IPUMS DHS (idhsstrata).

Your weighting command is correct. If you get an error message about single strata, add "singleunit(center)" to the end of it:

```
svyset [pw=perweight], psu(idhspsu) strata(idhsstrata) singleunit(center)
```

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
```
