Extracting and reshaping World Fertility Survey data in Stata

Germán Rodríguez, Princeton University
Trevor Croft, The DHS Program, ICF International

15 March 2017

Abstract

BACKGROUND
The Demographic and Health Surveys (DHS) program has made available online a large number of public-use files from its predecessor, the World Fertility Survey (WFS) program, see wfs.dhsprogram.com.

CONTRIBUTION
To encourage and facilitate the use of this data, we provide a Stata command that can be used to extract and reshape the data, using local copies or working directly with the DHS data archive pages.

1. Introduction

The World Fertility Survey (WFS) program, the forerunner to the Demographic and Health Surveys (DHS) program, was a groundbreaking demographic data collection effort that conducted over 40 surveys between 1972 and 1984. To ensure that WFS's valuable datasets continue to be available to the research community, the DHS program has published a large number of public-use data files through its data archive. Please visit wfs.dhsprogram.com for a list of the surveys available.

This paper introduces a Stata command that facilitates access to WFS data, with a user-friendly syntax that allows one to view the documentation, extract a set of variables from a survey, reshape the union or birth histories from wide to long format or vice versa, and make local copies of the data and dictionary files, although users also have the option of working directly from the DHS data archive.

A pioneering feature of the WFS program was its use of machine-readable data dictionaries to document the data files. This allows our command to read the

dictionaries as well as the data and to produce self-documented fully labeled Stata datasets.

By automating WFS data extraction, our command is a substantial time saver, but this does not exempt users from familiarizing themselves with the studies, recognizing that there are variations in survey design, questionnaire contents, and recoding of variables, that should inform the analysis of the data.

2. The `wfs` command

To install the command from net-aware Stata, type

```
net from http://wfs.dhsprogram.com
```

and click on the `wfs` link, or simply type

```
net install wfs, from(http://wfs.dhsprogram.com)
```

This will install the command and its help file.

The syntax of the command has five variants that address increasing functionality, from viewing to data extraction to reshaping the histories as well as making local copies of files. For ease of reference we provide below a syntax diagram and a brief description of each variant, followed by a more detailed explanation including examples in sections 3 to 7. In describing the syntax we follow Stata conventions, using a `roman` font for keywords, *`italics`* for user-specified elements, square brackets for options, and `underline` for minimal abbreviations.

1. `wfs`
   will open the home page of the WFS section of the DHS data archive using your default web browser. From there you can navigate to the country pages, as discussed in section 3.

2. `wfs using` *`filename`* `[, dhs]`
   will open the data dictionary for a given dataset using Stata's viewer, examples follow in section 4. The file name or dataset name should not include an extension. The `dhs` option reads the data dictionary from the DHS archive; otherwise it assumes you have a local copy.

3. `wfs` *`varlist`* `using` *`filename`* `[, dhs clear]`
   will extract a set of variables from the specified dataset. The varlist follows Stata conventions, as explained in more detail in section 5 below. The `dhs` option is the same as syntax 2. The `clear` option overwrites the dataset in memory, if any.

4. `wfs reshape long|wide [, births|unions nodrop]`
   will reshape the WFS data in memory from wide to long format or vice versa using the birth or union histories, as further explained in section 6 below.

5. `wfs copy` *`filename`* `[, directory(`*`folder`*`) replace]`
   will make a local copy of the data and dictionary for a given dataset, saving it in the specified folder or the current working directory. The `replace` option will overwrite existing files. Examples follow in section 7 below.

3. Viewing surveys

When you visit the WFS section of the DHS data archive, you will see links to 43 surveys from 42 countries. (The Dominican Republic had WFS surveys in 1975 and 1980.) If you click on the link for a country and year, you will see a description of the survey. For example, the page for Colombia, 1976, is at wfs.dhsprogram.com/index.cfm?ccode=co.

The country pages always include a table listing all public-use files. These files are available in two different formats: WFS, a plain text ASCII format, and ISSA, a binary format that requires the ISSA (Integrated System for Survey Analysis) program, an old DOS application that is no longer available. Our command works with files in WFS format.

You will find individual standard recode data and dictionary files, such as `cosr02.dat` and `cosr02.dct` for Colombia. You will also find household data and dictionaries, such as `vehh02.dat` and `vehh02.dct` for Venezuela, as well as household member data and dictionaries, such as `bdhm01.dat` and `bdhm01.dct` for Bangladesh. Many countries also have supplemental standard files with additional variables, such as `myss02.dat` and `myss02.dct` for Malaysia, and some have community data files, such a `egcd04.dat` and `egcd04.dct` for Egypt. The files are named following the scheme `ccftvn`, where `cc` is a country code, `ft` is a file type, and `vn` is a version number, with extensions `.dct` for the dictionary and `.dat` for the data. These are the files we can work with.

4. Viewing dictionaries

To view a data dictionary, you use the commmand

`wfs using` *`filename`* `[, dhs]`

where the filename omits the dictionary extension. For example, `wfs using cosr02, dhs` will show the machine-readable dictionary for the Colombia Standard Recode as stored in the WFS section of the DHS data archive.

Variables are typically grouped in sections. The individual standard recode files, for example, include sampling information and key dates, a union history, nuptiality variables, a birth history, fertility variables, breastfeeding, exposure status, fertility preferences, knowledge and use of contraception, background characteristics of the woman and the husband, and interview data.

The woman-level variables have names starting with `V` and a three-digit number, with the hundreds representing the content group. For example, all background variables have numbers in the 700s, and `V702` is 'Type of place of residence'.

The union histories include data for up to eight unions per woman and have names starting with `M`, followed by a two-digit union number and a third digit for the variable itself. For example, `M012` is the date of the first union and `M022` the date of the second union.

Birth histories include data for up to 24 births per woman using a similar naming scheme starting with `B`, so `B012` is the date of the first birth and `B242` the date of the 24th birth.

Some surveys also describe the union and birth histories using a compact table style with its own names. For example, `BDAT` refers to the dates of birth in `B012` to `B242`, in addition to the detailed list. Our command, however, uses the individual variable names.

Specific surveys may include additional standard recode variables, which have names starting with `X`, as well as country-specific variables, with names that start with `S`. For example Colombia includes `X701`, 'Always lived in this locality', and `S107`, 'Marital status <5 groups>'.

All variables have labels, and most also have value labels. For example, `V702`, 'Type of place of residence', is coded 1 for 'Urban', 2 for 'Rural' and 99 for 'Not stated'.

The dictionaries also specify the code used for 'not applicable', usually a string of 8s, and special codes, which are all codes in excess of a specified value, often a string of 9s used for 'not stated'. For example, `V702` defines `99` as a special code and the value label tells us that it means 'Not stated'. Variable `V301` is breastfeeding in the open birth interval and defines all codes above `96` as special; the value labels tell us `96` is 'Still breastfeeding', `98` is 'Did not breastfeed' and 99 is 'Not stated'.

For more detailed documentation on these data files please refer to the WFS *Data Processing Guidelines*(WFS Central Staff 1980).

5. Extracting data

Once you have selected the variables you are interested in, the next step is to extract them, which you do using the syntax

`wfs` *`varlist`* `using` *`filename`* `[, dhs]`

The `varlist` is simply a list of the variables you want to extract, Here we follow Stata conventions and refer to all variables using lowercase names, even though they have uppercase names in the dictionary. If you wanted age and type of place of residence from the Colombian Standard Recode file, for example, you would type the command `wfs v010 v702 using cosr02, dhs`.

The variable list may also use

- a hyphen to refer to consecutive variables in the dictionary. For example, you may refer to `v701-v705` to get the five residence and education variables.
- wildcards `*` and `?` with the same meaning as in Stata, so `v7*` will extract all variables in section 7 and `m??2` will extract the date of union for all unions.
- `births` to refer to all variables in the birth history, `unions` to refer to all variables in the union history, and `all` to extract all variables in the dictionary.

These three conventions may be freely combined in a list; for example, `v010 births m??2 v701-v705` is a valid list of variables. Note, however, that wildcards may not be used to specify the beginning or end of a range, just as in Stata. And of course `all` only makes sense by itself.

We extract variable and value labels for all selected variables. We also recode 'not applicable' values using Stata's `.n` missing value. We do not recode special codes, because we think it is better for users to make their own decisions on how to handle these values.

To extract age at interview and age at first union in the Colombian survey and then tabulate age at interview, all we need is

```
. wfs v011 v110 using cosr02, dhs clear
extracting variables v011 v110
(5378 observations read)
. tab v011
```

Age <5 yr grps>	Freq.	Percent	Cum.
15-19	1,423	26.46	26.46
20-24	1,051	19.54	46.00
25-29	842	15.66	61.66
30-34	599	11.14	72.80
35-39	579	10.77	83.56
40-44	476	8.85	92.41
45-49	408	7.59	100.00
Total	5,378	100.00	

If we tabulate age at first marriage, Stata will exclude single women because they have been coded 'not applicable':

```
. tab v110
```

Age at first union <7 grps>	Freq.	Percent	Cum.

< 15	348	10.54	10.54
15-17	965	29.22	39.76
18-19	707	21.41	61.18
20-21	506	15.32	76.50
22-24	419	12.69	89.19
25-29	256	7.75	96.94
30 +	101	3.06	100.00
Total	3,302	100.00	

You could include single women in the table by adding the `missing` option (abbreviated `m`) of the Stata `tab` command, which then becomes `tab v110, missing`.

6. Reshaping the histories

The union histories are stored in wide format, with one record per woman. For some purposes the analyst may need the data in long format, with one record per union. This can be done with the `wfs reshape long, unions` command, provided of course one has extracted the union histories. Here's an example:

```
. wfs v010 unions using cosr02, dhs clear
extracting variables v010 m011-m084
(5378 observations read)

. wfs reshape long, unions
(note: j = 01 02 03 04 05 06 07 08)
Data                                wide   ->   long
-----------------------------------------------------------------------------
Number of obs.                      5378   ->   43024
Number of variables                   34   ->       7
j variable (8 values)                      ->   union
xij variables:
                     m011 m021 ... m081    ->   m1
                     m012 m022 ... m082    ->   m2
                     m013 m023 ... m083    ->   m3
                     m014 m024 ... m084    ->   m4
-----------------------------------------------------------------------------

3868 unions
```

Our command uses Stata's own `reshape` command to do the bulk of the work. The Colombia dataset, which included 5,378 women, now has data for 3,868 unions. We can tabulate the type of union

```
. tab m1
    Type of
union <1-7> |      Freq.     Percent        Cum.
------------+-----------------------------------
   Marriage |      2,401       62.07       62.07
 Common law |      1,467       37.93      100.00
------------+-----------------------------------
      Total |      3,868      100.00
```

As you can see, the wide variables `m011` to `m081`, which have type of union for unions 1 to 8, are now simply `m1`, and the variable and value labels have adapted to the reshape, so `m1` is 'Type of union <1-7>'. We also have a new variable, `union`, to record the union number.

We could use `wfs reshape wide, unions` to go back to a wide format with one record per woman, but of course we would then have only ever married women in the dataset, unless you specified `nodrop`, as explained below.

Everything said so far applies to the birth histories as well. If we extract all births, we can reshape the data to obtain one record per birth:

```
. wfs v010 births using cosr02, dhs clear
extracting variables v010 b011-b245
(5378 observations read)

. wfs reshape long, births
(note: j = 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23
>  24)
Data                                wide   ->   long
-----------------------------------------------------------------------------
Number of obs.                      5378   ->  129072
Number of variables                  122   ->       8
j variable (24 values)                     ->   birth
xij variables:
                     b011 b021 ... b241    ->   b1
                     b012 b022 ... b242    ->   b2
                     b013 b023 ... b243    ->   b3
                     b014 b024 ... b244    ->   b4
                     b015 b025 ... b245    ->   b5
-----------------------------------------------------------------------------

14432 births
```

The standard output from Stata's reshape shows that each birth variable had up to 24 instances, so we went from 5,378 women to 129,072 observations (5,378 $\times$ 24), before dropping the padding to end with 14,432 births.

The variables `b015` to `b245` containing the date of death (or not applicable) for each child have now become `b5`. We can tabulate age at death using

```
. tab b5
  Age at death <8
 groups>   <1-19> |      Freq.     Percent        Cum.
------------------+-----------------------------------
        < 1 Month |        560       30.32       30.32
    1 - 2 Months |        151        8.18       38.49
    3 - 5 Months |        183        9.91       48.40
    6 -11 Months |        255       13.81       62.21
          1 Year |        286       15.48       77.69
     2 - 4 Years |        254       13.75       91.45
     5 - 9 Years |         89        4.82       96.26
       10+ Years |         52        2.82       99.08
     Not stated |         17        0.92      100.00
------------------+-----------------------------------
           Total |      1,847      100.00
```

to exclude surviving children, or `tab b5, missing` to include them. In both cases children who died but whose age at death is missing are included in the 'Not stated' category. Recall that we code 'not applicable' as missing `.n` but leave 'not stated' for you to handle.

The birth history may be reshaped back to wide format, but then the dataset would include only mothers, unless `nodrop` was specified. At that point one can freely go from wide to long and back to wide format as often as desired.

As noted above, when we reshape into long format, by default we keep only entries in the union or birth history that contain data, dropping empty entries which are coded 'not applicable' on all fields. The option `nodrop` can be used with `reshape long` to keep all of the entries; just type `wfs reshape long, births nodrop`. If you then reshape back to wide format, you will have all the original entries, including empty ones added for padding.

7. Local copies

A nice feature of the `wfs` command is that it can work directly from the DHS data archive. If you are planning to extract several subsets of variables, however, it probably pays to download the data and dictionary files to your local computer, which you can do using the syntax

`wfs copy using` *filename* `[, directory(`*folder*`) replace]`

By default, the data and dictionary files are copied to the current working directory, but the `directory(`*folder*`)` option may be used to provide an alternative destination, provided the target directory already exists. The `replace` option may be used as usual to overwrite existing data and dictionary files. For example, to copy the data and dictionary for the Colombian Standard Recode file to the current working directory, you would type the command `wfs copy using cosr02, replace`. You would then be able to extract and reshape the data using these local copies instead of the `dhs` option.

8. How it works

The `wfs` command works by reading the dictionary file or downloading it if the `dhs` option was specified, so it can obtain information about all the variables you wish to extract. It then writes a Stata script using `infix dictionary` and runs it to read the data, which is downloaded by Stata if the `dhs` option is in effect and otherwise read from the local file system. The next step is to label the variables, create value labels, and handle any 'not applicable' codes via recodes.

The `wfs reshape` command uses Stata's `reshape` to do most of the work, generating the appropriate command, but then modifies variable and value labels to adapt to the change in the unit of analysis. The command works only if the complete union or birth histories have been downloaded.

9. Acknowledgments

This work has been supported by NIH grant P2CH0047879 to Princeton University and by the DHS Program funded by USAID and implemented by ICF International.

References

WFS Central Staff (1980) Data processing guidelines. *World Fertility Survey basic documentation* 11(1-2). Voorburg: International Statistical Institute.