2013 UK Stata Users Group meeting
Cass Business School, London

usecspro: Importing CSPro hierarchical datasets to Stata

Sergiy Radyakin
sradyakin@worldbank.org

Research Department (DECRG),
The World Bank

September 13, 2013

Introduction

"This is tricky" - William Gould, StataCorp, circa 1996

http://www.stata.com/support/faqs/data-management/reading-hierarchal-dataset-with-infile/

Census and Survey Processing System (CSPro)

- Developed and supported by the US Census Bureau, with funding from USAID
- CSPro is used for basic data processing: data entry, validation, corrections and recoding, tabulation, etc.
- Software is declared to be in public domain and is distributed freely from the website:

http://www.census.gov/population/international/software/cspro

- Actively used since about 2000 by hundreds of organizations to collect data as part of:
  - Censuses
  - Labour Force Surveys
  - Income and Expenditure Surveys
  - Demographic and Health Surveys

CSPro files

CSPro software uses multiple file types (more than 15) with different extensions and for different purposes. The minimal set of files that is needed to retrieve the data from this package is the following:

Dictionary file

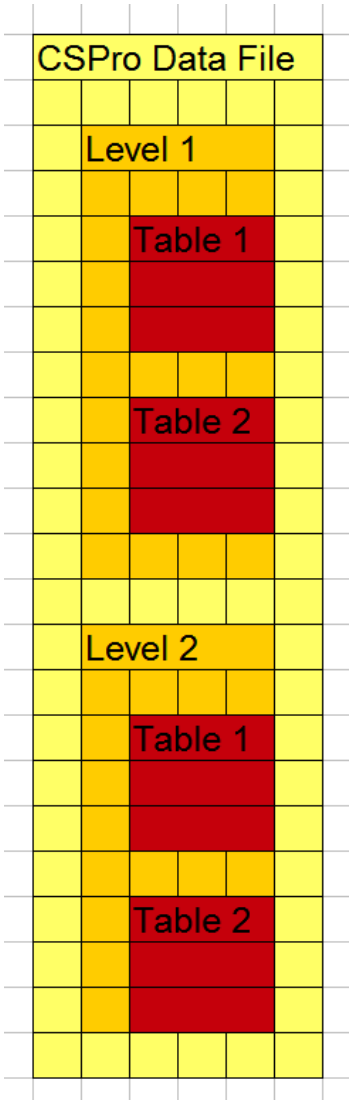
Dictionary files (*.dcf) define the structure of data. One dictionary may be applicable to multiple datasets.

Data file

Data files: (usually *.dat or without any extension) contain all data packed into a single file.

Both files are necessary to import the data.

CSPro data organization



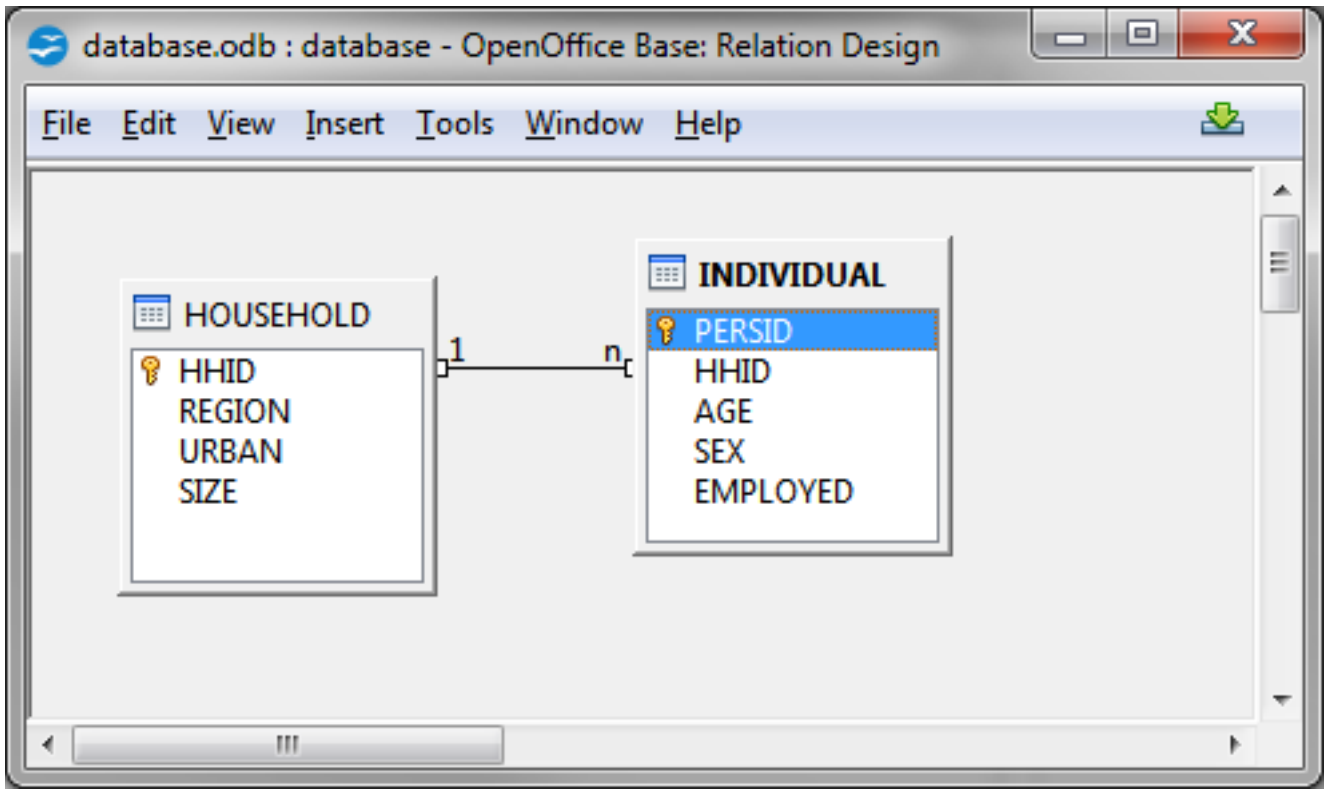
CSPro allows the database architect to describe the structure of the dataset in terms of levels, records, and data items.

- Level is a unit of data corresponding to the survey instrument (type of questionnaire). Most of surveys require only one level. Some surveys require several survey instruments, and hence corresponding databases consist of several levels.
- Record is a unit of data corresponding to a topic of the survey: such as a group of questions related to employment or health status. Here a table is a group of records of the same type.
- Data item is a unit of data corresponding to attributes (variables).

How does CSPro logically organize data, often times into a single level?

Relational database

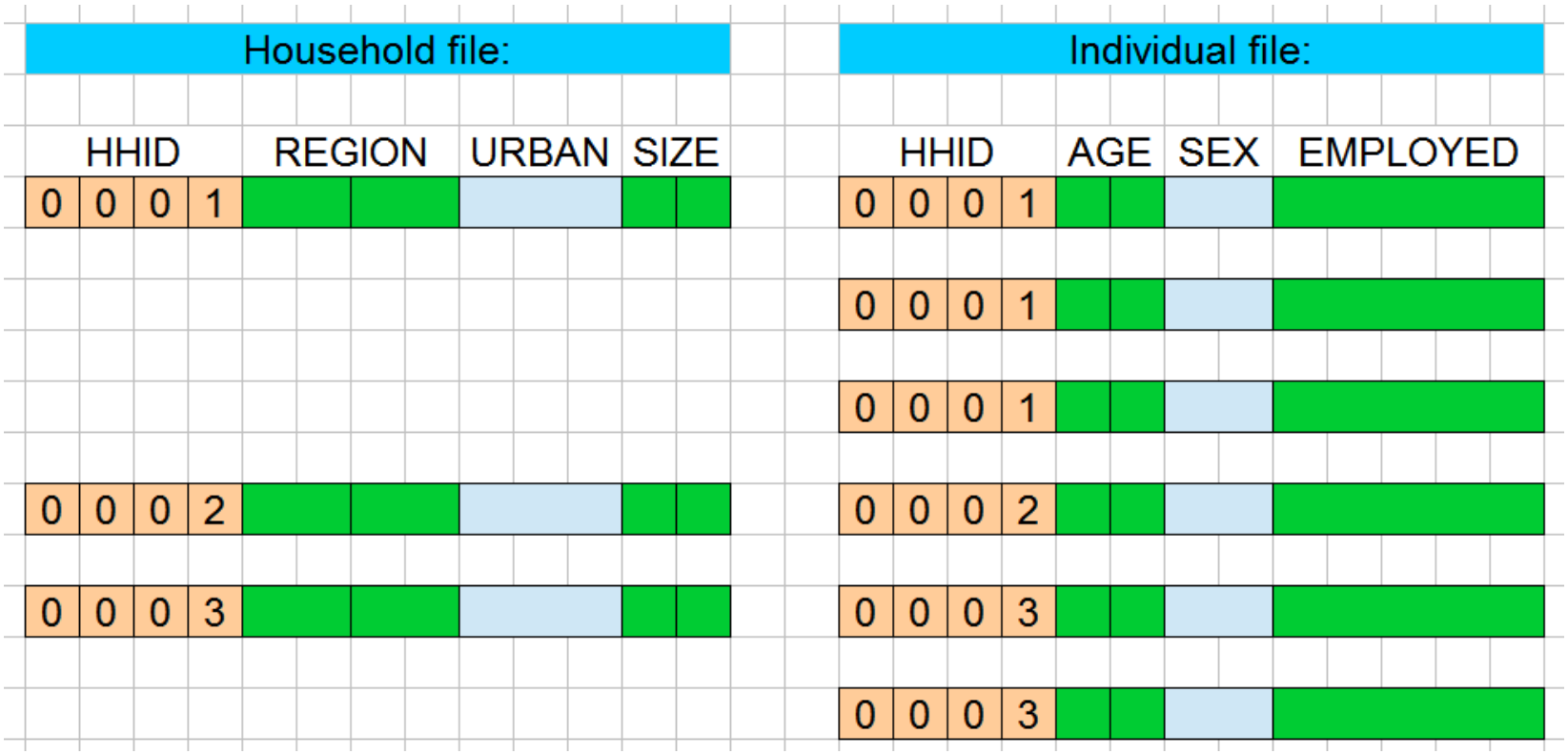
Consider a simple relational database (as used by e.g Microsoft Access, or OpenOffice Base):



Here we have two tables: of households and individuals, and a relationship indicating that more than one individual may reside in one household. The identifier matching the two tables is the field HHID (household number).

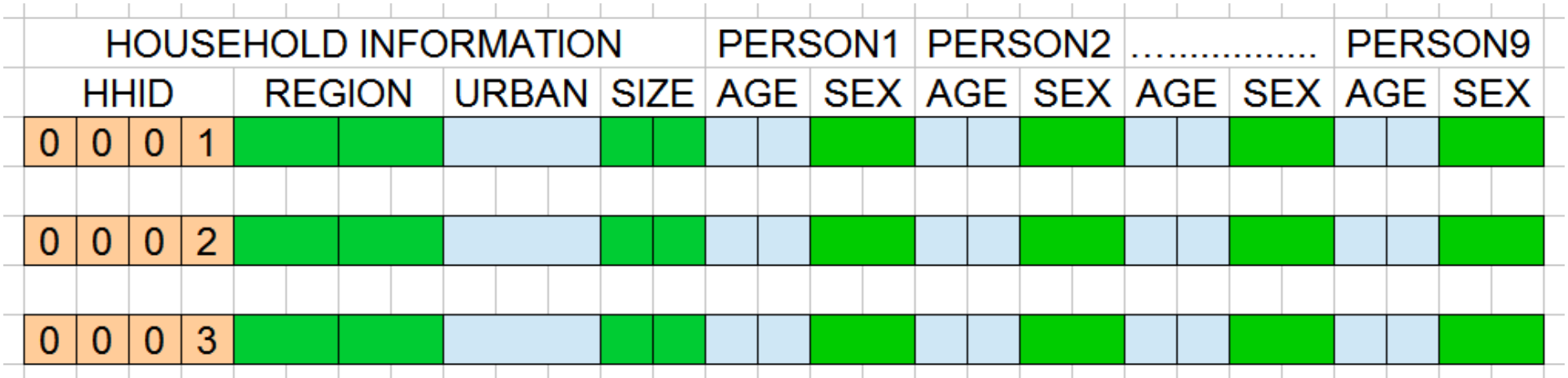
Relational database

It is natural to describe this as two-level dataset with first level being households and second level being individuals:



CSPro approach

In fact CSPro prefers the following single level layout of the same database:



This type of data layout is known in Stata's world as wide dataset.

Note that all attributes here are attributes of the household. Even though we tend to think of age as a characteristic of an individual, here it is in fact: age1 - age of the first member of the household, age2 - age of the second member of the household, etc.

The number of slots allocated for the person-specific information (in this example: 9) has to be preset at design time and should be sufficiently large to accommodate all realistic cases.

This could be quite wasteful in terms of the occupied space, since most households would have 'less-than-maximum' number of persons.

CSPro data file layout

Even when CSPro describes data as one level, it partitions it into multiple record types and interleaves records of all types in a single data file, marking each record type with a special 'record-type' identifier.

1	0	0	0	1	0	7	1	0	3
2	0	0	0	1	3	9	1	1	
2	0	0	0	1	2	5	0	0	
2	0	0	0	1	1	2	1	9	
1	0	0	0	2	1	2	0	0	1
2	0	0	0	2	4	9	1	1	
1	0	0	0	3	0	1	1	0	2
2	0	0	0	3	2	9	1	0	
2	0	0	0	3	2	7	0	1	

Colors legend
Record type
Household ID (4)
Region (2)
Urban
Household size (2)
Age (2)
Sex
Employed

CSPro records data as ASCII text (or utf-8 unicode text in CSPro 5.0), with each field of specified width (in characters); records do not have to be of the same width: here personal record is shorter than household record.

Implementation

- Stata itself does not import CSPro files directly, and neither Stat/Transfer (Circle Systems), nor DBMS/Copy (Conceptual Software, discontinued) import these files specifically.
- The only alternative remained to use CSPro itself to export data to Stata (it does support export to a plain text format plus a syntax file to generate variable and value labels.)
- Not all Stata users have CSPro installed. Only Windows users can potentially install it. There is no alternative for Mac and Linux users.
- -usecspro- is a package developed by *Sergiy Radyakin* for Stata 10 or newer that implements import of the hierarchical CSPro data files.
- -usecspro- provides a simple mouse-click solution to import the data, as well as provides a broad API for the advanced users.

Features

- **Data label** - supported, record name is used as the dataset label in the resulting Stata dataset.
- **Variable names** - fully supported (identical naming conventions), converted to lowercase.
- **Variable labels** - fully supported
- **Value labels** - partially supported:
  - only the first set of labels is used,
  - interval labels (no equivalent in Stata) are converted to discrete labels (when possible),
  - non-integer values are not labelled (not possible in Stata).
- **Missing values** - supported. The three special CSPro values reserved to denote missingness are converted to Stata's extended missing values:

  `DEFAULT`→.a, `NOTAPPL`→.b, `MISSING`→.c

- **Decimals** (including implied decimals) - supported. Even when decimal separator is not saved into the dataset, the decimals are imputed during conversion using the declared field properties in the dictionary.
- **Leading zeroes** - not supported.

Conversion strategy of -usecspro-

The strategy is (what happens behind the scenes):

1. parse the dictionary file to 'understand' the data organization
2. filter the dataset to have only records of the single type
3. write the Stata's dictionary file to read-in the data
4. write the Stata's do-file to do formatting adjustments, recode missing values, etc.
5. read the data from step #2 using the dictionary from step #3
6. execute the do-file from step #4

The user doesn't need to know nor understand the above. It is sufficient to use the following friendly syntax.

Basic syntax

Interactively

```
usecspro
```

In Stata do-files

```
usecspro using "data.dat", dictionary("dictionary.dcf")
level("LevelName") record("RecordName")
```

In Mata functions

```
cspro_convert("dictionary.dcf", "data.dat", "LevelName",
"RecordName")
```

Stata syntax

`cspro about`	Display information about the program
`usecspro using "file.dat", dictionary("file.dcf") level("levelname") record("recordname") [clear]`	Import CSPro data record 'recordname' of data level 'levelname' using specified dictionary (non-interactive)
`cspro dir ["path"]`	Show the list of CSPro files in the current (or specified) directory
`cspro use ["file.dcf"]`	Show the data levels of the specified dictionary file. If file not specified, open a dialog to pick a file.
`cspro import "file.dcf"`	Same as above
`cspro dump "file.dcf"`	Dump all records of all data levels as separate datasets into a subfolder. Datasets are nested into subfolders with name of the level.
`cspro join using "file.dat", dictionary("file.dcf") level("levelname") records("`R_1 R_2 ... R_N`")`	Join specified records of the same data level 'levelname' using CSPro identifier of this level.

Mata syntax

Basics

`cspro_about()`	Display information about the program
`cspro_convert("dictionary.dcf", "data.dat", "levelname", "recordname")`	Import CSPro data record 'recordname' of data level 'levelname' using specified dictionary (non-interactive version for programmers)

Interactivities

`cspro_import("folder")`	Writes to the output a clickable list of the CSPro dictionary files in the specified directory
`cspro_list_levels("dictionary.dcf")`	Writes to the output a clickable list of data levels contained in the dictionary file
`cspro_list_level_records( "dictionary.dcf","levelname")`	Writes to the output a clickable list of records for the specified data level in the dictionary file

Mata syntax (cont.)

Utilities

`cspro_give_levels("dictionary.dcf")`	Returns a list of data levels contained in the dictionary file
`cspro_give_level_records( "dictionary.dcf", "levelname")`	Returns a list of records for the specified data level in the dictionary file
`cspro_dump("dictionary.dcf", "data.dat", "folder")`	Dumps all records of all data levels as separate datasets into a subfolder
`cspro_dump_simple("dictionary.dcf")`	Dumps all records of all data levels into a subfolder, data file is assumed to be same as the dictionary file, with extension .dat, the output folder is in the dictionaryname˙DUMP in the folder where dictionary is located (will be created if does not exist)
`cspro_join("dictionary.dcf", "data.dat", "levelname", "`R_1 R_2 ... R_N`")`	Performs a join of two or more records of the "levelname" data level into a single dataset using CSPro identifier of this level
`cspro_guess_joinp("dictionary.dcf", "data.dat", "levelname", "`R_1 R_2 ... R_N`")`	Performs a join of two or more records of the same data level into a single dataset using an extended (guessed) ID
`*cspro_is_utf8_file("filename.ext")`	Checks if a given file is encoded in utf-8
`*cspro_convert_utf8_to_ansi( "utf8file.txt", "ansifile.txt")`	Converts a given file from utf-8 encoding to ANSI

* in Windows only

Interactive use example

```
. cspro dir c:\csprodemo\

CSPro Dictionary Files in c:\csprodemo\
```

File name	Version	Label
06-1-123-304	CSPro 4.1	**NPS-HhQ**
castetribe	CSPro 2.6	**Other answers**
doha	CSPro 3.3	**doha**
iabr52fl	CSPro 4.0	**IABR52FL**
iahr52fl	CSPro 4.0	**IAHR52FL**
sdes_wb	CSPro 4.1	**SDES_WB**

```
.
```

Interactive use example

```
. mata cspro_list_levels("c:\csprodemo\doha.dcf")
CSPro dictionary: c:\csprodemo\doha.dcf
Software version: CSPro 3.3
Dictionary's label: doha
File contains the following data levels:      DOHA_QUEST
.
```

Interactive use example

```
. mata cspro_list_level_records("c:\csprodemo\doha.dcf", "DOHA_QUEST")
```

```
Software version: CSPro 3.3
Dictionary's label: doha
Data level: DOHA_QUEST
Level's label: doha questionnaire
```

record name	max length	record label
HOUSE	1	12 House
*POP	50	15 POP

```
Total records in this level: 2 (including 1 optional records indicated with *)

.
```

Interactive use example

```
. mata cspro_convert("c:\csprodemo\doha.dcf", "c:\csprodemo\doha.dat", "DOHA_QUEST", "POP")
  -------------------------------------------------------------------------------
  oooo  oooo   .oooo.o  .ooooo.   .ooooo.   .oooo.o oo.ooooo.  oooo d8b  .ooooo.
  `888  `888  d88(  "8 d88' `88b d88' `"Y8 d88(  "8  888' `88b `888""8P d88' `88b
   888   888  `"Y88b.  888ooo888 888       `"Y88b.   888   888  888     888   888
   888   888  o.  )88b 888    .o 888   .o8 o.  )88b  888   888  888     888   888
   `V88V"V8P' 8""888P' `Y8bod8P' `Y8bod8P' 8""888P'  888bod8P' d888b    `Y8bod8P'
                                                     888
                                                    o888o                  v1.1

   ------ A simple, fast and reliable tool to import CSPro data to Stata ------

   Written by Sergiy Radyakin, Economist
              Research Department (DECRG) the World Bank
              sradyakin/at/worldbank.org
              May-Aug 2013

Processing dictionary file: c:\csprodemo\doha.dcf
Produced with software version: CSPro 3.3
Dictionary label: doha
Reading level: DOHA_QUEST
  Reading record: POP with typeid: 2

Preparing data
Processing data file: c:\csprodemo\doha.dat

Importing data
Applying labels
.
```

Interactive use example

```
. des

Contains data
  obs:          16,556                          doha
 vars:               7
 size:         132,448
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
doha_id         int    %9.0g                  ID
pn              byte   %9.0g                  Person number
relationship    byte   %8.0g                  Relationship
sex             byte   %8.0g       SEX_VS1    Sex
age             byte   %17.0g      AGE_VS1    Age
marital_status  byte   %8.0g                  Marital-status
children        byte   %9.0g                  Children
-------------------------------------------------------------------------------
Sorted by:
     Note:  dataset has changed since last saved

.
```

Interactive use example

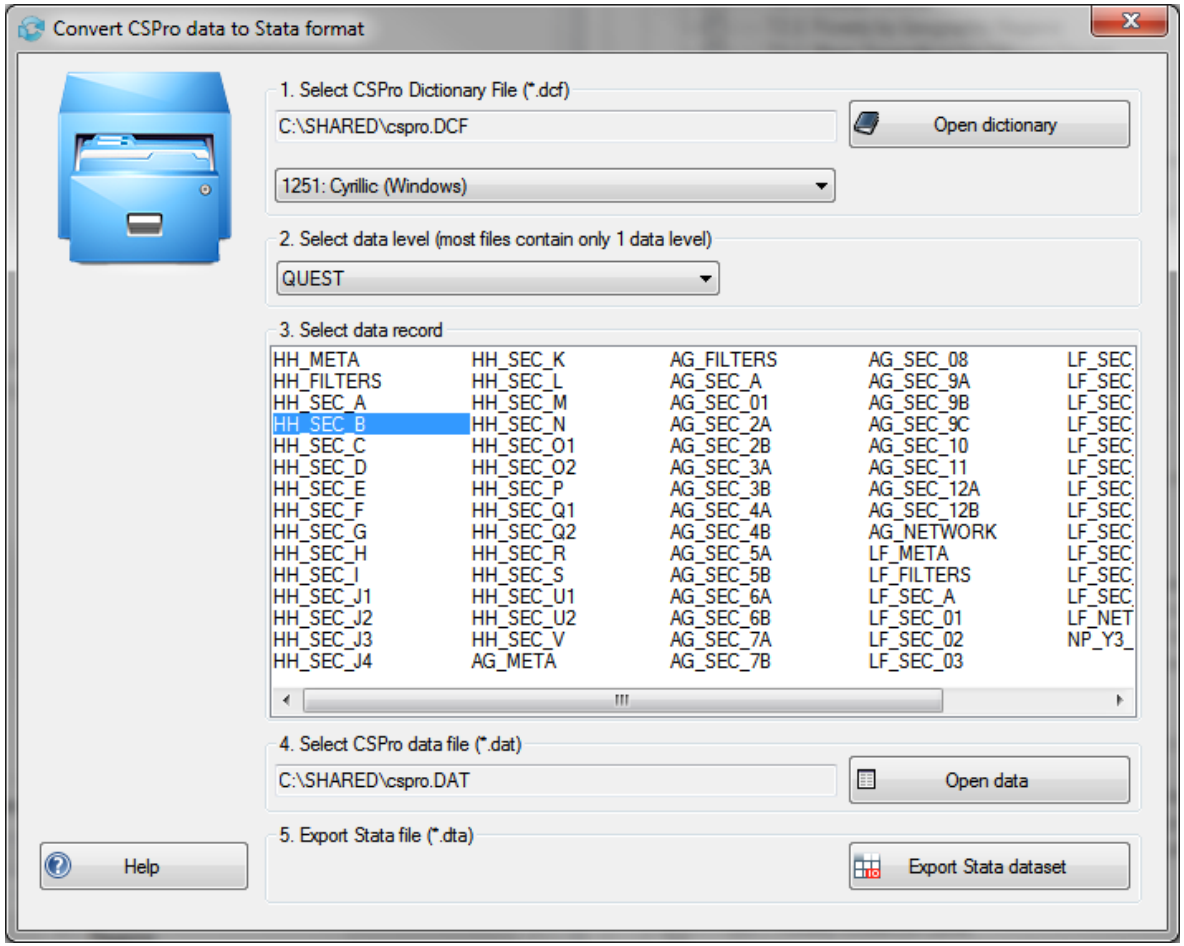
```
. l  in 1/5
```

	doha_id	pn	relati~p	sex	age	marita~s	children
1.	1	1	1	Female	25 - 29 years	5	0
2.	1	2	6	Male	25 - 29 years	5	.
3.	2	1	1	Male	35 - 39 years	1	.
4.	2	2	2	Female	35 - 39 years	1	1
5.	3	1	2	Male	35 - 39 years	1	.

```
.
```

Windows-only features (dialog)

Dialog interface:



Windows-only features (unicode)

- Unicode support - CSPro 5.0 is using utf-8 encoding for dictionary (variable and value labels) and data files (string values).
- Files are saved in unicode even if no non-ASCII characters are used. Hence files need to be converted to ASCII+ANSI since Stata does not support unicode.

- Windows users of -usecspro- can additionally specify a codepage to be used for non-ASCII characters.
- Only one codepage can be specified (e.g. *1251: Cyrillic (Windows)*) and it is applied to both dictionary file and data files. This choice is only available in the dialog if the dictionary is actually saved in utf-8 encoding.
- Codepage can be specified as an additional parameter (default=1252 Western European codepage) of both Stata and Mata commands:

```
usecspro using "file.dat", dictionary("file.dcf") ///
        level("L") record("R") codepage(1251)

mata cspro_convert("file.dcf", "file.dat", "L", "R", 1251)
```

Я Cyrillic smaller letter ya is now supported in variable and value labels.
Click to learn why it is so special?

Windows-only features (unicode)

To view the non-ASCII characters correctly in Stata, adjust the font settings of the output window to match the encoding that was selected during the conversion: