WP. 9
ENGLISH ONLY

UNITED NATIONS ECONOMIC COMMISSION FOR EUROPE (UNECE)

CONFERENCE OF EUROPEAN STATISTICIANS

EUROPEAN COMMISSION

STATISTICAL OFFICE OF THE EUROPEAN UNION (EUROSTAT)

Joint UNECE/Eurostat work session on statistical data confidentiality
(Tarragona, Spain, 26-28 October 2011)

Topic (ii): Software research and development

Key Variable Mapping System II

Prepared by Mark Elliot, Duncan Smith, Elaine Mackey and Kingsley Purdam,
University of Manchester, United Kingdom
.

Key Variable Mapping System II

Mark Elliot[*], Duncan Smith[*] Elaine Mackey[*] and Kingsley Purdam[*]
[*] Centre for Census and Survey Research, University

Abstract: The Key Variable Mapping System (KVMS) is an approach for identifying matching possibilities across datasets within a data environment. It is a formalised approach for identifying key variables. An overview of KVMS is provided in Elliot et al. (2010). The original KVMS was implemented in a spreadsheet. The user had to enter information about datasets and variable categorisations manually across several sheets. The KVMS incorporated the idea of a harmonization graph which related the various categorisations of a variable across different forms, although these had to be constructed manually and were not used within the KVMS.

The new KVMS is based on a more formal mathematical definition of a harmonization graph, and the harmonization graph is constructed automatically from graphical models created on data entry. Data entry is simplified and inconsistencies in categorisations are identified immediately by reference to the underlying graphical models that drive the system. Inferences regarding matching possibilities are implemented as algorithms on harmonization graphs. This produces significant performance increases.

1 Background

It is now generally accepted that a precursor to carrying out statistical disclosure risk analysis is the grounded generation of appropriate key variable sets. Work by Paass (1990), Elliot and Dale (1999) laid the ground work for an approach based on attack scenarios. This approach has been further extended by Mackey (2009) who describes the notion of the data environment. Elliot et al (2010) show how this notion might be formalised and describe a pilot system - the key variable mapping system. This is the first attempt to metricise the likelihood of a disclosure attempt and the work provides an indicator of how we might move beyond the data-centric approach (using just the properties of the to-be-released data to estimate risk).

This paper describes the latest version of the KVMS system, with an improved interface and a more principled approach to the underlying business logic based on aggregation graphs.

1.1 Data Environment Analysis

Data Environment Analysis (DEA) is a unique approach developed at the University of Manchester with funding from the Office of National Statistics (ONS). The goal of DEA to investigate, catalogue, categorise, and document available data in identification databases (those which could be used to link to target anonymised datasets in order to inform disclosure scenarios for data release).

Prior to this work there has been no (other) formal mechanism, within SDC, which allows the identification and classification of what additional, external, information might be utilizable by a would-be data intruder. This has meant that a key element of the scenario structure the *means of an attempt* (which centrally revolves around what key variables is potentially available to a would-be intruder), is based largely on informed guesswork. In the absence of such a formal method, we have hit a barrier preventing further development of our understanding of how a disclosure might occur. Without such understanding well grounded scenarios are impossible and we need well grounded scenarios to produce reasonably accurate disclosure risk measures. Otherwise we run the risk of making disclosure management decisions that are either too conservative or too liberal. These could lead to: (i) potentially risky data being released; (ii) valuable low-risk data not being released; (iii) data releases of limited utility because of the damage caused by data protection methods.

Another way to look at this is that there are two overarching themes captured in any scenario analysis: (i) *is the scenario likely* - which is assessed (using Elliot and Dale's classification scheme) by considering a plausible intruder's 'motivations' and 'opportunities for attack' and (ii) *is it possible, and if so how* - which is centrally assessed considering what additional information an intruder would require to successful identify and/or disclose new information about respondents from a data release. It is this second element that provides the rationale to DEA work.

We have identified that within a given DEA cycle there are two phases: (i) to investigate, catalogue, categorise and document what additional information may be available to an intruder and (ii) to develop grounded disclosure risk scenarios.

1.2 Data Environment Analysis Methods

Data environment analysis uses several interrelated methods. For each method the principle is to capture metadata which can inform disclosure risk decision making. Elliot et al (2010) describe the various methods by which such data can be collected. One of the core methods being 'form field analysis' which works on the assumption that if a form (paper or electronic) asks for a given piece of personal information, then that information will be stored on a database of individual records (which could form an identification file in an attempt to attack an anonymised data file). The second assumption is that the data will be stored at the level of detail that it is collected. These are not strong assumptions and therefore it is plausible to infer that each form provides veridical metadata for the correspondent database held by the organisation that collects the information.

So, in form field analysis the forms themselves are the raw data. Each form provides information about the content of a database and that information itself becomes a record within a *meta-database*. The rows in such a database are the forms and the columns are possible variables and their codings. So, each possible coding of a variable will be represented as a column in a meta-database. There are relationships between the different codings of a variable. For instance, one coding might represent the collapsing of sets of categories within another coding. In this case one would be strictly less detailed than the other. This gives rise to a transitive *refinement* relationship. These relationships between the codings of a variable can be represented in a graph.

The management of a meta-database is a complex process. To understand this consider that in response to each question on a new form there are three types of development that can happen:

i. *Assimilation to an existing structure* – here the question maps directly onto an existing coding and so the process is one of simply adding the form to that section of the database and indicate the code.

ii. Accommodation of an existing structure – here the question does not directly map on to an existing code meaning that at least one new coding will have to be created. The new code will then have to be placed into the graph for that key variable. This can be extremely complex and can require the creation of harmonization codes which link the new coding to one or more existing codings. The harmonization code for a set of codings *S* is the unique coding that is the most detailed coding of the variable that is not more detailed than any codings in *S*.

iii. *Creation of a new structure.* Occasionally a question is asked on a form which might result in the decision to create a new key variable. With each new form the meta-database develops and the overall structure, which in effect is a picture of the data environment, is enriched.

2 The New Key Variable Mapping System (KVMS II)

KVMS has been developed to enable the production of precise key variable specifications. Its overall function is to generate a key variable set which is the metadata intersection of two datasets (or indeed classes of datasets). KVMS compares codings of key variables across different datasets using a target dataset and either a single dataset or a summary of similar datasets.

KVMS, written in Python, sits on top of the meta-database. The system works over a set of stored variable codings, as outlined in section 2. These codings are continually developed as new metadata is added to the system. The codings are structured as a set of taxonomic graphs such as the one shown in Figure 1. The nodes in each graph are of two types:

1. observed data nodes – these correspond to coding systems actually used in collected metadata.

2. Harmonized data nodes which are codings produced by harmonizing two or more observed data nodes (all the observed data nodes that are descendants of the harmonized data node in the graph).

The graph contains a node for each harmonization for each subset of the set of observed codings. In practice many subsets harmonize to common codings, and often these are observed codings. The number of distinct codings can be surprisingly small. When a respondent is asked to write in a value for a variable it is often assumed that the coding is arbitrarily highly detailed. This gives rise to a special coding that harmonizes to all other codings and becomes a singular leaf node in the graph.

The ancestral graph for a coding (the graph induced by a coding and its ancestors) contains all the codings to which the coding harmonizes. A set *S* of codings can be used to generate a corresponding set of ancestral graphs. The graph induced by the intersection of the graphs' node sets is a graph with a single leaf node, This leaf node is the harmonization of all the nodes in *S*. The number of forms containing codings that harmonize to a given coding can be found easily by associating a count with each graph node (coding) and propagating the totals through the graph. The analysis described in Elliot et al., (2010) can be performed very efficiently.

This section describes an implementation of KVMS written in Python for Windows, Linux and Mac. The screenshots show the application running on Ubuntu (Linux).

2.1 The top level of KVMS

At the top level, the KVMS interface contains a notebook with 5 tabs:

- The Analysis tab contains a panel used for performing analyses.
- The Forms tab contains a grid for displaying form data. Initially there is no data to display and it only contains 3 empty columns for form metadata. The column labels are specified in *config.cfg* which can be manually edited to specify alternative column label.
- The Variables and Harmonizations tabs contain notebooks which have tabs for each variable. Initially these notebooks have zero tabs.
- The 5th tab contains a Python shell. This can be used to interact with the data environment directly.

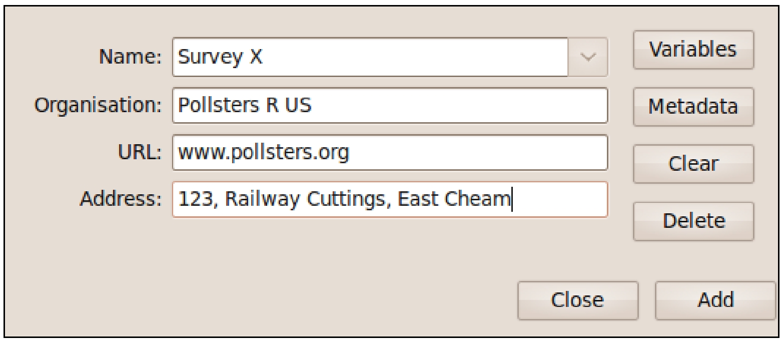
2.2 Data entry

Two types of data can be entered into KVMS: *Forms* and *Groups*. Here, for reasons of space, we are only concerned with forms. The general procedure is to create a form, add variables to the form, and then specify the categorisations for the variables (which will be automatically converted to codings and added to the relevant harmonization graph).

A form name must be entered into the Form Editor. Metadata are optional. The editor provides text controls for the metadata names such as URL (the desired names can be specified in the configuration file). Other name, value pairs can be entered via a grid that can be accessed via the *Metadata* button.

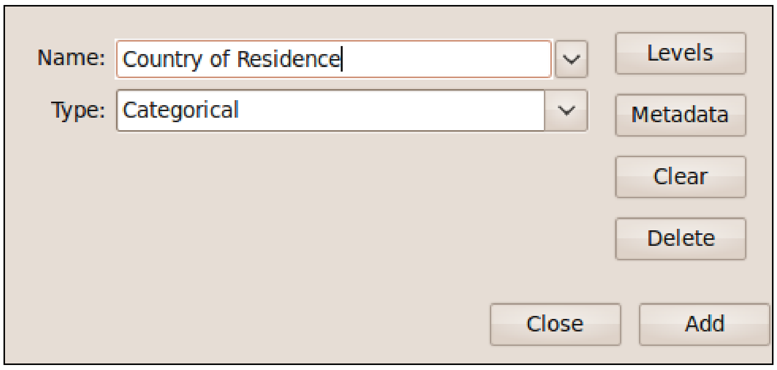
The editor initially has a blank working form, selecting a form via the combobox selects a copy of an existing form as working form, and clearing the editor creates a new blank working form. The editor does not close when a form is added. It must be explicitly closed once the user has finished with data entry.

To add or edit the variables associated with a form a Variable Editor is invoked via the *Variables* button.



Name: Survey X
Organisation: Pollsters R US
URL: www.pollsters.org
Address: 123, Railway Cuttings, East Cheam

Figure 2.1 The Form editor



Name: Country of Residence
Type: Categorical

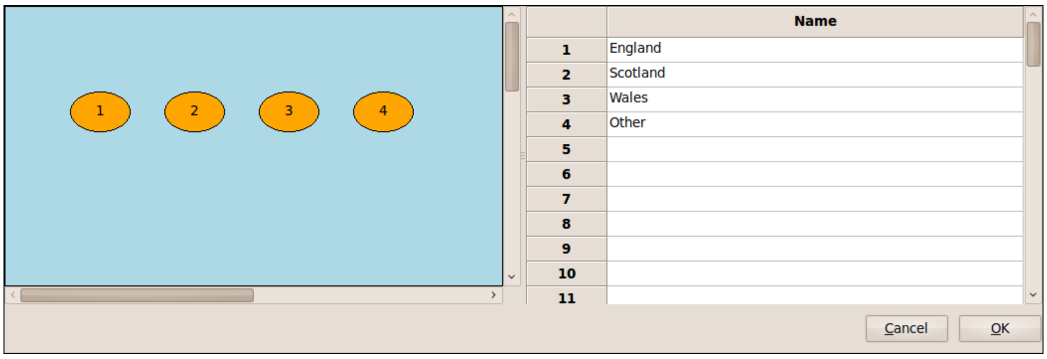
Figure 2.2 The Variable Editor

The Variable Editor is similar to the form editor. Initially it contains a blank working variable. But a copy of an existing variable can be made the working variable via selection from the combobox. The combobox separates variables already contained in the form from other variables contained in other forms so that the user can identify the variables already added to a form. The buttons operate similarly to those in the Form editor.

2.3 The aggregation Editor

The Aggregation Editor is a graphical tool building aggregation graphs. An aggregation graph is a directed acyclic graph with nodes for variable levels and the properties that a parent is equal to the union of its children and the children are mutually exclusive. Thus, Britain could be a parent of England, Scotland and Wales. Aggregation graphs contain information regarding the relationships between different categorisations. Each variable has a single aggregation graph. Although the methodology has not been outlined, it is the categorisations specified for a variable and the aggregation graph that contain the information used to construct the harmonization graph for the variable.

The Aggregation Editor consists of a canvas and a grid. The category names are entered into the grid, whilst an aggregation graph can be created / edited via the canvas. After creating / editing an aggregation graph the relevant variable levels are selected on the canvas.

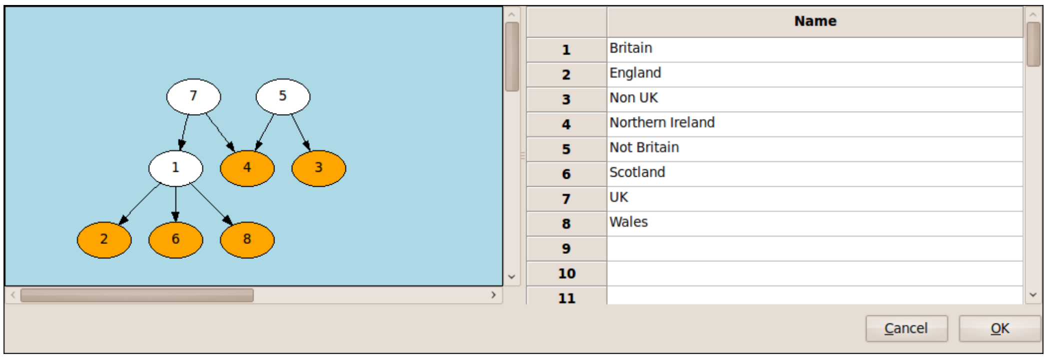


	Name
1	England
2	Scotland
3	Wales
4	Other
5	
6	
7	
8	
9	
10	
11	

Figure 2.3 The Aggregation Editor with selected categories

In Figure 2.3 the categorisation of Country of Residence is England, Scotland, Wales and Other. Clicking the OK button adds the categories to the working variable and closes the editor. Any changes to the aggregation graph are also saved. If none of the nodes is selected the variable contains no levels in its categorisation.

Starting from the above example, a second form was added by loading the existing form and variable in the editors, editing the aggregation graph, specifying an alternative categorisation, and changing the form name before adding to the data environment. Figure 2.4 shows the edited aggregation graph.

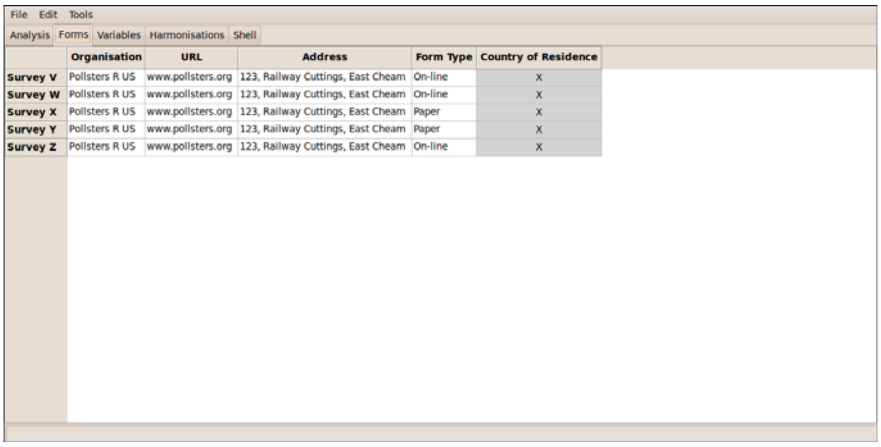


	Name
1	Britain
2	England
3	Non UK
4	Northern Ireland
5	Not Britain
6	Scotland
7	UK
8	Wales
9	
10	
11	

Figure 2.4 The edited aggregation graph

The 'Other' category was renamed to 'Not Britain' and 'Northern Ireland', 'UK', 'Non UK' and 'Britain' categories were added. The categorisation of this second form for Country of Residence is England, Scotland, Wales, Northern Ireland and Non UK.

The editor only allows valid changes to be made to an existing aggregation graph. Thus it is important that the first time an aggregation graph is specified the categories are mutually exclusive and exhaustive. If the Other category had not been specified originally, then creating the new categorisation would not have been possible. Note that a valid categorisation is a set of nodes such that all the leaf nodes (nodes with no children) are reachable (along directed paths) from the nodes in the categorisation, and each leaf node is reachable from exactly one node in the categorisation. Note that this is still true, in Figure 2.4, for the categorisation in the first form; England, Scotland, Wales and Other (renamed to Not Britain). Renaming of categories is performed globally, renaming the category in any instances of the variable contained in the data environment

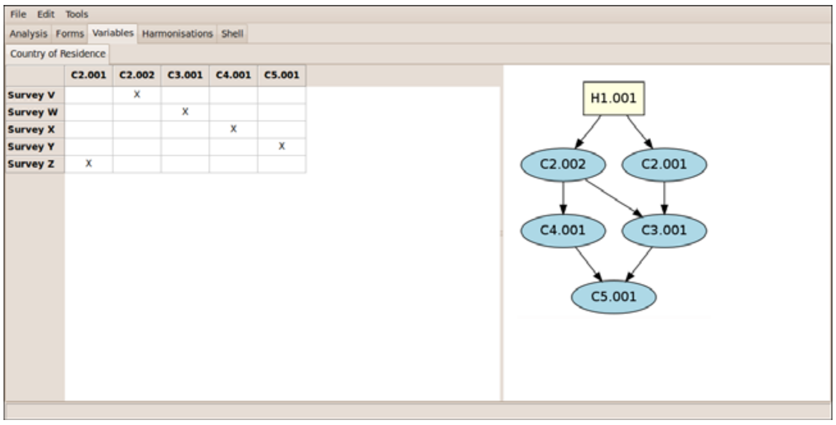


	Organisation	URL	Address	Form Type	Country of Residence
Survey V	Pollsters R US	www.pollsters.org	123, Railway Cuttings, East Cheam	On-line	X
Survey W	Pollsters R US	www.pollsters.org	123, Railway Cuttings, East Cheam	On-line	X
Survey X	Pollsters R US	www.pollsters.org	123, Railway Cuttings, East Cheam	Paper	X
Survey Y	Pollsters R US	www.pollsters.org	123, Railway Cuttings, East Cheam	Paper	X
Survey Z	Pollsters R US	www.pollsters.org	123, Railway Cuttings, East Cheam	On-line	X

Figure 2.5 The Forms grid with 5 forms added

The following figures show the user interface after adding several more forms containing various categorisations of Country of Residence.

When a form is added to the environment a row is added to the forms grid and new metadata and variable columns are added as needed. The reverse happens when forms are deleted. Figure 2.5 shows that form metadata on *Form Type* have been added. When a variable is added a new tab is added to the notebook under the Variables tab. This contains a grid and a canvas. The grid contains information about variable codes (corresponding to categorisations). The codes are of the form CX.Y, where X is the number of categories and Y is a positive integer (less than 1000) which distinguishes between codes with the same number of categories. The canvas contains a harmonization graph where an edge from a code *v* to a code *w* implies that *w* can be aggregated to form the categorisation *v* (and therefore any of *v*'s ancestors).

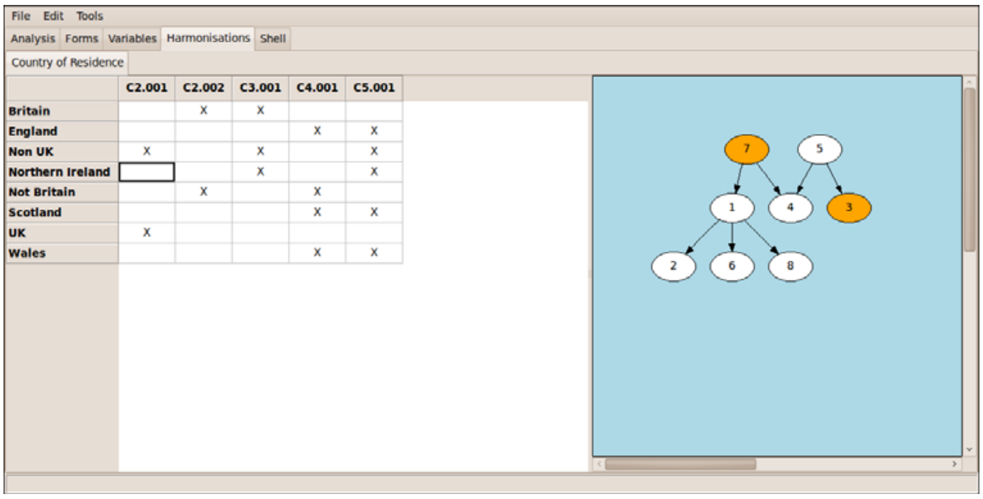


	C2.001	C2.002	C3.001	C4.001	C5.001
Survey V		X			
Survey W			X		
Survey X				X	
Survey Y					X
Survey Z	X				

Figure 2.6. The Variables notebook page for Country of Residence

When a variable is added to the data environment a new tab is also added to the notebook under the Harmonizations tab. This also contains a grid and a canvas. The grid provides details of the categorisations corresponding to the codes. The canvas contains the aggregation graph corresponding to the variable. If a column cell is selected the nodes corresponding to the categorisation are selected in the aggregation graph. Editing the graph on this canvas makes no difference to the underlying data environment. To do that the user would have to go back into the data entry system.

When a variable is deleted from the data environment (i.e. all forms) the corresponding tabs are removed from the notebooks under the Variables and Harmonizations tabs.



	C2.001	C2.002	C3.001	C4.001	C5.001
Britain		X	X		
England				X	X
Non UK	X		X		X
Northern Ireland			X		X
Not Britain		X		X	
Scotland				X	X
UK	X				
Wales				X	X

Figure 2.7. The Harmonizations notebook page for Country of Residence

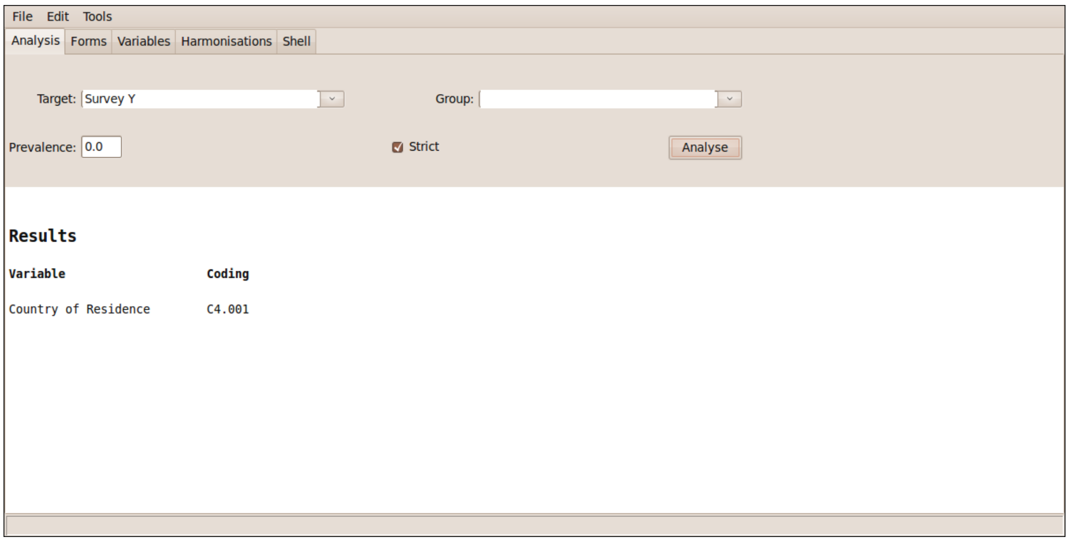
2.4 Analysis

The basic analytical principle is to find a set of harmonization codes between a target (the microdata dataset we are trying to protect) and a single form (or set of forms from the data environment). The set of harmonization codes are the key variables for the scenario involving the datasets represented by the form or group of forms.

The analysis tab contains a panel that allows the selection of a target form, a single (or group of) forms and a prevalence. If no group / form is selected then a group comprising all the forms is used. The results are displayed in a text control.

The Strict checkbox allows the consistency constraints on the analysis to be relaxed. Although it is difficult to enter inconsistent categorisations for a variable, it is not impossible. After all, the user has direct access to the data environment via the shell. In normal use the application will check for inconsistent categorisations and print a relevant message in the output, rather than to try to compute a result. If the checkbox is unchecked, then the consistency check is skipped and a result is computed. However, the user should not place too much trust in such outputs.

The codes can be looked up in the notebook under the Harmonizations tab if they are existing categorisations. If they are not, then they will be of the form HX.Y and can be looked up via the shell, as can any such codes shown in the harmonizations graphs in the notebook under the Variables tab.



Target: Survey Y
Group:
Prevalence: 0.0
Strict ✓

```
Results
Variable                 Coding
Country of Residence     C4.001
```

Figure 2.7. *Analysis output*

3 Concluding Remarks

The work reported here forms part of a portfolio of ongoing research which attempts to capture, describe and metricise the data environment. This process is important in enabling data stewardship organizations such as national statistical institutes to make principled decisions about data dissemination of research data products. As the work goes forward we are planning to run studies on informal public data, particularly that known via personal knowledge. A second key point of interest for data stewardship organisations is how information is changing over time; the importance of understanding data environment trends will be crucial to longer term planning of data dissemination.

REFERENCES

Elliot, M. and Dale, A. (1999): "Scenarios of Attack: A Data Intruder's Perspective on Statistical Disclosure Risk," *Netherlands Official Statistics*, 14, 6-10.

Elliot, M. (2010) "Privacy, Confidentiality and Disclosure: conflicts and resolutions", Paper presented to Angela Dale's retirement Colloquium; Manchester, June 2010. http://www.ccsr.ac.uk/events/adc/index.html

Elliot, M. J., Lomax, S., Mackey, E. and Purdam, P. (2010) 'Data Environment Analysis and the Key Variable Mapping System' with In J Domingo-Ferrer and E Magkos (eds) Privacy in Statistical Databases. Springer; Berlin.

Mackey, E. (2009) “A Framework for Understanding Statistical Disclosure Processes: A Case Study Using the UK’s Neighbourhood Statistics.” Phd Thesis submitted to the University of Manchester.

Paass, G. (1988): "Disclosure risk and disclosure avoidance for microdata", *Journal of Business and Economic Statistics* 6(4): 487-500.

Purdam, K., & Elliot, M. J. (2002): "An evaluation of the availability of public data sources which could be used for identification purposes - A Europe wide perspective", CASC project report. Manchester: University of Manchester.

Purdam, K., Mackey, E. and Elliot, M. (2004) The Regulation of the Personal, Policy Studies, Vol 25, No 4, Dec 2004 pp 267-282