WP.46
ENGLISH ONLY

UNITED NATIONS
ECONOMIC COMMISSION FOR EUROPE

CONFERENCE OF EUROPEAN STATISTICIANS

Work Session on Statistical Data Editing
(Ljubljana, Slovenia, 9-11 May 2011)

Topic (vii): New and emerging methods

TEIDE2: new improvements and new results

Invited Paper

Prepared by Mª Salomé Hernández García & Juan José Salazar González,
University of La Laguna, Tenerife, Spain

I. Introduction

1. TEIDE2 is a software for automatic data editing of surveys. It is being implemented at University of La Laguna (Tenerife, Spain). It works on quantitative and qualitative microdata, and it allows the consistent edits to be mathematical and/or logical rules. Details on a previous version can be found in [2].

2. The software is self-contained and can be executed in all type of computers (mainframe, laptop, notebook; with Microsoft Windows or with Linux or with Mac-OS). It was fully implemented in ANSI C++ programming language, without using any commercial tool. It is free and open-source software, thus any user can check what and how each function was implemented and even modified it to better integrate it inside any data system. TEIDE2 read and writes microdata in Microsoft Access, Microsoft Excel, Oracle and XML. Other formats may be included on request. In addition to the source code, a user can also get a single stand-alone executable file to run TEIDE2. In other words, executing TEIDE2 does not need any installation procedure to run on a new computer.

3. TEIDE2 has been extensively used by several statistical agencies in Spain (including the regional agencies in Canary Islands, in Andalucía and in Baleares Islands) on several surveys (including social, economic, technological, tourisms, etc.). The largest of these surveys contained about 20,000 records, about 1,000 variables and about 500 edits. Once all the input (including microdata, variable definitions, consistent edits and internal parameters) have been inserted in TEIDE2, the full automatic process (including editing and imputation) has been done in less than one hour on a standard laptop computer. The imputation is based on a combination of the widely used hot-dock donor procedure and multi-regression analysis.

II. TEIDE2

A. Techniques for Editing and Imputation of Statistical Data, version 2

4. TEIDE2 is a software which is implemented in ANSI C++ and makes use of Qt (a multi-platform library for developing graphical user interfaces). It allows reading data in different format, like Microsoft Office Access, Microsoft Office Excel, Oracle and XML. As C++, Qt, Oracle and XML are available in all major multiplatforms we can presume that this product can be used on Linux, Windows, Mac, etc.

5. TEIDE2 is a tool for debugging, editing and imputing statistical surveys. Its main functions are to detect errors that may have the surveys (edition) and to correct these errors (imputation). When TEIDE2

load a microdata, no matter the format, three windows are displayed. One window refers to the variables, another window to the microdata, and the third windows shows the consistent rules.

B. Edition

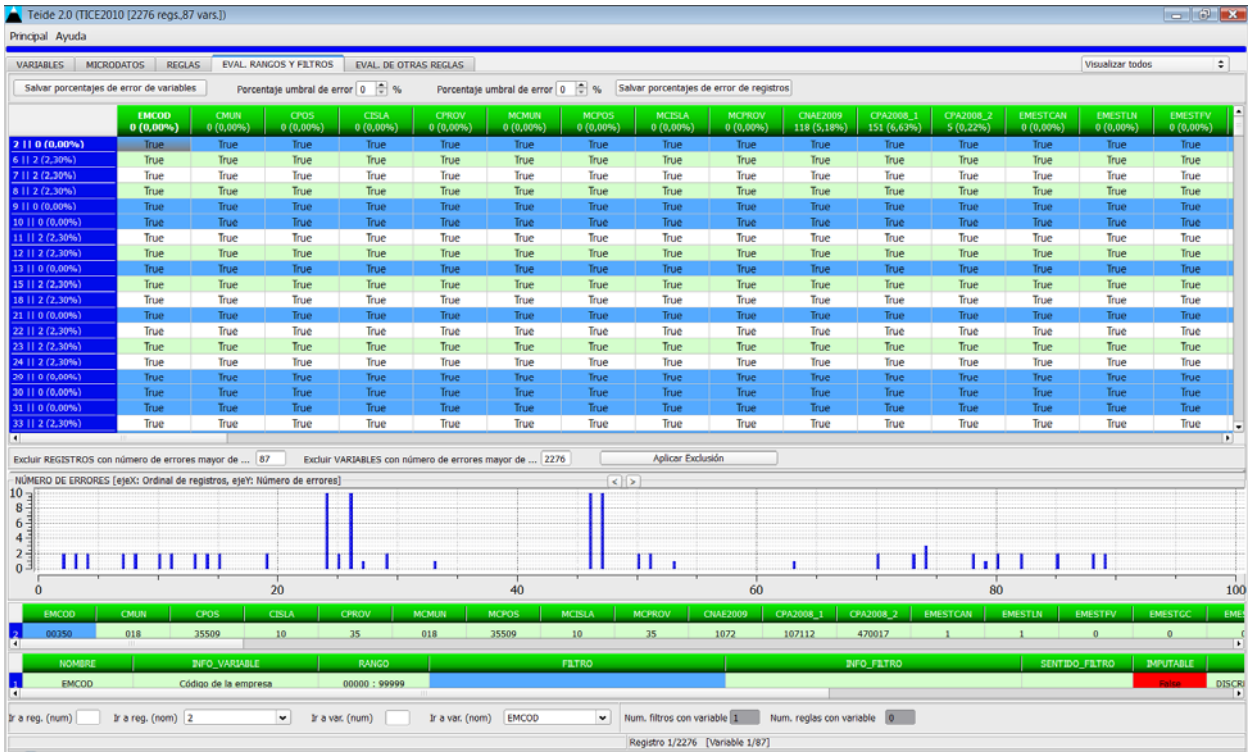
6. Editing in TEIDE2 consists in detecting which records are incorrect. A record is considered wrong when:

(a) The value of some variables is not in the range of values that have to take that variable.

(b) A variable does not comply a logical condition (call "filter") fixed by the user. A "filter" is a logical condition that a variable must satisfy in order to assume a value with meaning; otherwise, the variable should assume the value "not applicable" (i.e., the NO_PROCEDE value).

(c) When the record does not satisfy at least one of the edits.

7. This checking is done on two new windows, where line represents a record. In one window a column represents a variable, and gives the value True or False depending if conditions (a) and (b) are satisfied for this variable. In the other window a column represents an edit, and the value True or False refers to condition (c).



Edition: Evaluation of ranges, filters and rules.

C. Imputation

8. The imputation consists of correcting the incorrect records. TEIDE2 does it by means of the methodology of the donor record (Fellegi and Holt, 1976).

9. Donor records are those records that the editing phase classified as correct. A distance function is defined to measure the similarity between a correct and a non-correct record. Using this function, the set of donor records are sorted for each record that need to be corrected. Then, starting from the closest correct records to each incorrect record and considering different combinations of variables, values from donor records are giving to variables so as to get a corrected record with the least number of variables changed.

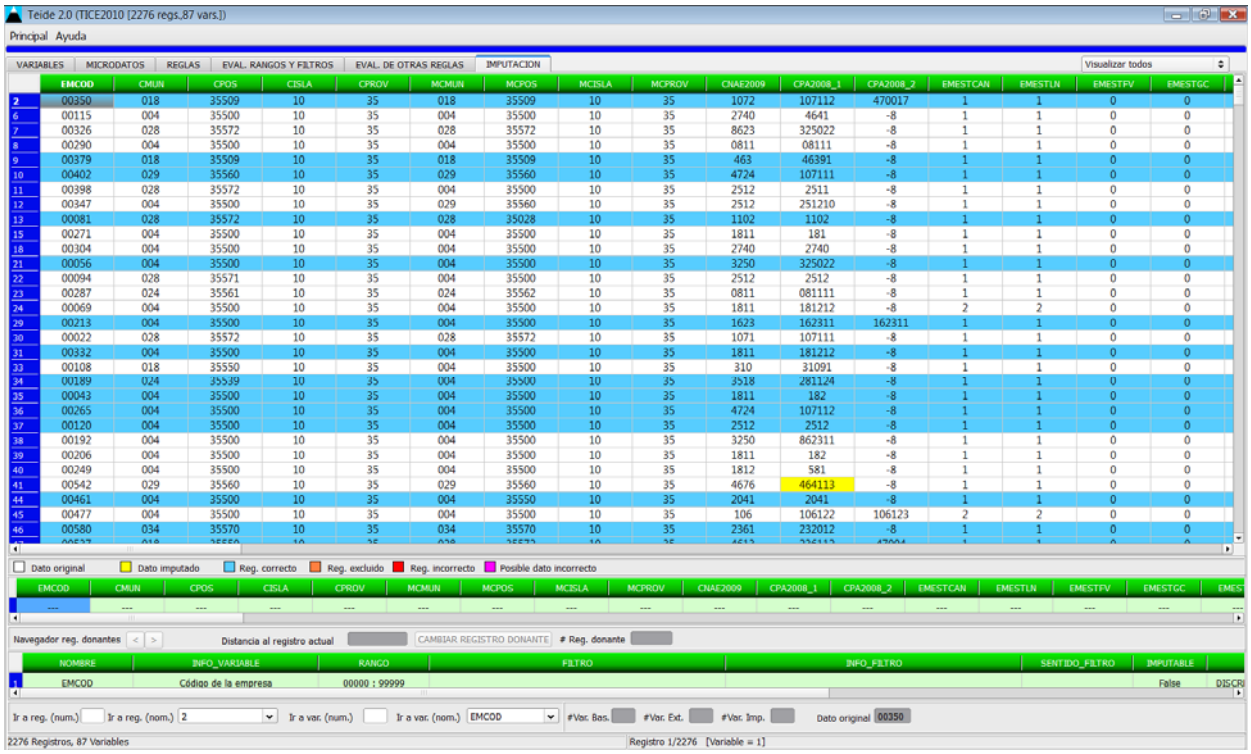
10. A crucial element for the quality of the imputation is the definition of the distance function. In TEIDE2 this function is the sum of a "similarity" between the two values for each variable. This

similarity is defined for each variable according to its type. While for numerical variables it is the difference, for logical variables it is a 0-or-1 value depending if the two values are identical or not. Each similarity is multiply by the importance (or weight) of each variables, as defined by the use in the input data. Moreover, the similarity of a variable is also affected by another factor depending whether it is a variable is a wrong edit for this record, or it is a variable in a correct edit but sharing this edit with another variable in a wrong edit, or otherwise. Briefly, we try to emphasize the similarities of variables which are likely to content correct values in the incorrect record.

11. Although TEIDE2 suggest default values for several parameters affecting the definition of the distance function, the user has also the possibility of changing these parameters and measure how the imputation results is affected by different parameter settings.

12. Another crucial element for the performance of the approach is the selection of the subset of variables which are donated from the correct record to the incorrect record. In the default implementation of TEIDE2, an enumerative approach tries all subsets with less than a given number (say less than 5 variables). These subsets are sorted according to the importance of the variables, so the smaller subsets with the minimum variable weights are first tried to correct a record. If this enumerative procedure does not succeed in solving an incorrect record, then a heuristic approach is applied to potentially correct the record. If the incorrect record is corrected then it is classified as “warning”, so the user will have the opportunity to check it at the end of the process. If the correcting is not possible, then the record is classified as “incorrect” and again the user will see it at the end of the process.

13. At the end of the process a report is created with all the modifications done my TEIDE2 on a copy of the original data. If no warning and no incorrect records remain then the user will find statistics to compare the original and the modified data. The ideal conclusion should be to find that TEIDE2 has changed an average of one record for each incorrect record. In our experiences this ideal results is hard to achieve, but typically TEIDE2 gets very close to it since the average is very close to one record.



Imputation.

III. Tests of compilation and execution

A. Surveys treated

14. To illustrate the performance of TEIDE2 on a real-world survey, we are considering here a survey collected by the Regional Statistical Agency of Canary Islands (ISTAC). The survey concerns the implantation of Information Technology and Communication in companies of Canary Islands ("Tecnologías de la Información y la Comunicación en empresas", *TIC companies*) and houses of Canary Islands ("Tenologías de la Información y la Comunicación en hogares", TIC-HC), both in the period 2009-2010. The survey TIC companies aims to analyze the implantation and use of information technologies and communications and electronic commerce in the business sector in the Canary Islands. The survey ICT-HC allows knowing the deployment of technologies of information and communication in homes, the equipments in these technologies, and the use that people make of the computer, Internet and e-commerce. TIC-HC has been divided into two: *TIC homes* and *TIC people*. Therefore we have three surveys which are inter-connected.

15. The size of each survey is described by the following table:

	# variables	**# edits**	**# records**
TIC companies	151	77	2276
TIC people	94	20	7088
TIC homes	100	4	3058

These surveys are not the largest one where TEIDE2 has been tested. Indeed, TEIDE was tested also on surveys with close to 30,000 records. However, we are selecting these surveys in this paper because they are the most recent real-world data for which we are allowed to show computational results.

B. Features of compilation and implementation

16. Using TEIDE2 on the three surveys, we carried out computational experiments in two computers, each one with two different operating systems. They are following:

Intel Core 2 Duo 3,34GHz. Memory (RAM) 4 GB Operating Systems - Ubuntu 10.04 to 64 bits - Windows Vista Business to 64 bits
Intel Core 2 Duo 1,67GHz. Memory (RAM) 3 GB Operating Systems - Ubuntu 10.04 to 32 bits - Windows Vista Home to 32 bits

17. For compiling and linking the source code of TEIDE2 we have used the following tools:
   (a) On Windows computers, we have produced executables of TEIDE2 with two standard C++ compilers: MinGW and Microsoft Visual Studio 2008. For producing the executable of TEIDE2 only one compiler is necessary. Therefore, we have produced different executables to be compared. The MinGW compiler is 32 bits, and cannot create a 64 bits executable. However, with Visual Studio we have obtained both 32 bits and 64 bits versions.
   (b) On Linux computers we have been used the standard GNU g++ compiler to produce 32 bits and 64 bits executables for TEIDE2.

18. Therefore, we have been able to run TEIDE2 on five different scenarios:
- Windows 32 using Visual Studio
- Windows 32 using MinGW
- Windows 64 using Visual Studio

- Linux 32
- Linux 64

19. We did not try on Mac computers because we do not have them when the experiments with our real-world surveys have been conducted.

C. Windows time

20.

(a) Time in data load. It makes reading the database and displays the tables of variables, microdata and rules (edits).

TIC companies		
MinGW 32 bits	**Visual Studio 32 bits**	**Visual Studio 64 bits**
2 seconds	2 seconds	2 seconds

TIC people		
MinGW 32 bits	**Visual Studio 32 bits**	**Visual Studio 64 bits**
5 seconds	5 seconds	4 seconds

TIC homes		
MinGW 32 bits	**Visual Studio 32 bits**	**Visual Studio 64 bits**
1 second	1 second	1 second

(b) Time of process in this section ranges and filters. Consists of the error checking of ranges and filters.

TIC companies		
MinGW 32 bits	**Visual Studio 32 bits**	**Visual Studio 64 bits**
1 second	1 second	1 second

TIC people		
MinGW 32 bits	**Visual Studio 32 bits**	**Visual Studio 64 bits**
5 seconds	4 seconds	4 seconds

TIC homes		
MinGW 32 bits	**Visual Studio 32 bits**	**Visual Studio 64 bits**
1 second	1 second	1 second

(c) Time of process in the test section. Consists in checking for errors that have occurred in the rules given.

TIC companies		
MinGW 32 bits	**Visual Studio 32 bits**	**Visual Studio 64 bits**
1 second	1 second	1 second

TIC people		
MinGW 32 bits	**Visual Studio 32 bits**	**Visual Studio 64 bits**
3 seconds	3 seconds	2 seconds

TIC homes		
MinGW 32 bits	**Visual Studio 32 bits**	**Visual Studio 64 bits**
1 second	1 second	1 second

(d) Time of process in the imputation section. It makes the correction of erroneous records in the editing phase.

TIC companies		
MinGW 32 bits	**Visual Studio 32 bits**	**Visual Studio 64 bits**
4 seconds	4 seconds	2 seconds

TIC people		
MinGW 32 bits	**Visual Studio 32 bits**	**Visual Studio 64 bits**
18 seconds	18 seconds	11 seconds

TIC homes		
MinGW 32 bits	**Visual Studio 32 bits**	**Visual Studio 64 bits**
0 seconds	0 seconds	0 seconds

D. Linux time

21.

(a) Time in data load.

TIC companies	TIC people	TIC homes
G++ 32 bits	**G++ 32 bits**	**G++ 32 bits**
2 seconds	5 seconds	1 second

TIC companies	TIC people	TIC homes
G++ 64 bits	**G++ 64 bits**	**G++ 64 bits**
2 seconds	4 seconds	1 second

(b) Time of process in this section ranges and filters.

TIC companies	TIC people	TIC homes
G++ 32 bits	**G++ 32 bits**	**G++ 32 bits**
8 seconds	25 seconds	10 seconds

TIC companies	TIC people	TIC homes
G++ 64 bits	**G++ 64 bits**	**G++ 64 bits**
8 seconds	24 seconds	10 seconds

(c) Time of process in the test section.

TIC companies	TIC people	TIC homes
G++ 32 bits	**G++ 32 bits**	**G++ 32 bits**
8 seconds	28 seconds	10 seconds

TIC companies	TIC people	TIC homes
G++ 64 bits	**G++ 64 bits**	**G++ 64 bits**
8 seconds	23 seconds	10 seconds

(d) Time of process in the imputation section.

TIC companies	TIC people	TIC homes
G++ 32 bits	**G++ 32 bits**	**G++ 32 bits**

7 seconds	31 seconds	0 seconds

TIC companies	**TIC people**	**TIC homes**
G++ 64 bits	**G++ 64 bits**	**G++ 64 bits**
7 seconds	28 seconds	0 seconds

E. Comments

22. In a Windows computer, Visual Studio 2008 to 64 bits is the best compiler.

23. Nor surprising, in a Linux computer a 64 bits executable is better than a 32 bits executable.

24. TEIDE2 is implemented to read data in several format, including Microsoft Office Access (*.mdb), Microsoft Office Excel (*.xls), Oracle and XML. However, we cannot guarantee all formats on all operating system. For example, we found problems to read Microsoft Excel data on Linux computers. The following table gives details on the format that each executable can read:

Version	**Access**	**Excel**	**Oracle**	**XML**
MinGW 32 bits	YES	YES	YES	YES
Visual Studio 32 bits	YES	YES	YES	YES
Visual Studio 64 bits	YES	YES	YES	YES
Linux 32 bits	NO	NO	YES	YES
Linux 64 bits	NO	NO	YES	YES

For the experiments we have used the XML format to be user that all executable can work.

25. The input database contains:

	# variables	**# working variables**	**# edits**	**# records**
TIC companies	151	87	77	2276
TIC people	94	61	20	7088
TIC homes	100	24	4	3058

The column “working variables” is the number of variables which can be treated, i.e. if the variable is not ignorable or the variable has data. Indeed, the user can always decide which variables are allowed to be changed by TEIDE2 and in many situations in practice they are not necessary the whole set of variables. However, it is important to notice that correcting a wrong record is more difficult when the set of working variables is small.

26. During the imputation phase TEIDE2 produces the following results on each survey:

	Donor records (o correct)	**Records to correct**	**Correct records**	**Incorrect records**	**Records warning**
TIC companies	1375	901	901	0	0
TIC people	3057	4031	4031	0	0
TIC homes	3058	0	0	0	0

This table shows that the surveys “TIC companies” and “TIC people” contained more than 50% of records with wrong values. TEIDE2 was able to correct all the wrong records without producing even warning records. The survey “TIC homes” did not contain wrong records from the beginning.

27. A reader may have the impression that TEIDE2 was not useful in debugging the survey TIC homes because there were no incorrect records. However, TEIDE2 was also very useful because many inconsistencies were detected while records were arriving to the statistical agency. The main reason of these inconsistencies was due to misunderstanding of the initial queries by the data collectors, and it was

TEIDE2 the tool used to detect these inconsistencies in early time. Experiences of many practitioners using TEIDE2 on real-world surveys show that TEIDE2 is not useful only in the final phase of debugging the microdata, but also in the initial phase of checking that the record are being collected correctly.

28. Considering the two surveys with records to be corrected, the quality of the imputation is described by the final report produced by TEIDE2 in the last step, and which is summarized in the following tables:

TIC companies	
Average imputed variables per record (total)	4.95
Average imputed variables per record (no reg. warning)	4.95
Average of errors in range per record	2.88
Average of variables involved in edits (rules) incorrect for record	3.89
Average of variables involved in total mistakes for record	5.48
Average of variables involved in comp. connected with mistake for record	3.25
Average distance to donor records	0.00

TIC people	
Average imputed variables per record (total)	5.29
Average imputed variables per record (no reg. warning)	5.29
Average of errors in range per record	5.00
Average of variables involved in edits (rules) incorrect for record	5.20
Average of variables involved in total mistakes for record	5.20
Average of variables involved in comp. connected with mistake for record	40.03
Average distance to donor records	0.00

29. From these tables we observe that the average number of variables with a wrong value (due to out-of-range or to the filter) is 2.88 for “TIC companies” and 5.00 for “TIC people”. After the imputation phase, the average number of modifications in a wrong record was 4.95 for “TIC companies” and 5.29 for “TIC people”. Considering each survey at a time, the proximity of these two values supports the good quality of the imputation done. In addition it is also relevant to observe that a connected component in “TIC companies” contains an average of 3.25, so the variables are not so inter-connected. Instead, the connected component in “TIC people” contains an average of 40.03 variables, thus the imputation is more complicated to be done.

IV. Ongoing work

A. Selective imputation

30. Recently TEIDE2 has been improved with a new module to perfume “selective imputation”. Although the tool is already and available inside TEIDE2, we still need to use it on real-world survey and measure the satisfaction of users.

31. This module has been designed and implemented during a visit of María Salomé Hernández García to the Spanish National Statistical Agency (INE, Madrid) and was done in collaboration with Pedro Revilla and Ignacio Arbués.

32. Selective debugging is to detect records to be corrected without the use of edits. The search for a good selective editing strategy is stated as an optimization problem in which the objective is to minimize the expected workload with the constraint that the expected error of the aggregates (computed with the edited data) is below a certain constant. The deterministic problem in this case can be expressed as

$$\max_{r} \; 1^T r - \sum_k \lambda_k (\Delta^k r - \epsilon_k^2)$$
$$s.t. \quad r_i \in [0,1]$$

By applying the Karush–Kuhn–Tucker conditions, we get a solution given by

$$r_i = \begin{cases} 1 & if\ \lambda^T \Delta_i < 1 \\ 0 & if\ \lambda^T \Delta_i > 1, \end{cases}$$

where if $r_i = 1$ then not debug is necessary, and otherwise debug is necessary. This problem will return the records to be imputed in a second step.

33. We recommend the article [1] for details on this methodology.

34. With the incorporation of this methodology TEIDE2 allows the user to choose the method for data debugging. In other words, now it is possible to decide if the records should be edited either by using consistency rules or by using selective debugging. In both cases the non-correct records will go to the imputation phase of TEIDE2.

B. Parallelization

35. TEIDE2 has also been improved with new techniques to use all processors available in the computer that is running the application. Today technology produces computers with several cores (like Core Duo or Core Quad), so it is quite advisable to do parallel computation instead of the traditional sequential programming on old computers.

36. TEIDE2 detects the number of cores available in a computer and throws as many threads as the number of cores. This allows correcting several wrong records in different cores, in a parallel and independent way. After the initial phase, devoted to loading and checking the microdata, the set of correct records is fixed, and therefore the distance function evaluations and imputation phase for each non-correct record can be done in parallel. Because the imputation of a wrong record does not affect the imputation of a different donor record, the parallel implementation has a tremendous impact on the time consumption, almost being linearly proportional to the number of cores in the computer.

V. Conclusion

37. TEIDE2 is an open-source code that shows a user what and how methods have been implemented. This has the additional advantage of allowing users to modify the code and easily integrate the code in their own database system. Moreover, any new methodology can easily be implemented and experimented inside TEIDE2 and therefore provide the international community with new modules to perform data editing and imputation of high quality.

38. TEIDE2 can be executed in any type of computer, no matter the operating system, without effort. It applies automatic data editing and imputation through a graphical and user-friendly approach. All the process is illustrated through colors and reports, and allows different parameter settings to experimented users. By default TEIDE2 has default parameters to allow an all-in-one-step run.

39. TEIDE2 has been used and improved in the last 5 years by different statistical agencies on real-world surveys, so the current version is quite robust and stable. However each survey and each statistical office may wish to add in TEIDE2 new features, thus it is an alive and open project for new testers.

VI. Acknowledgments

40. This article has been funded by “Ministerio de Ciencia e Innovación” through the research project MTM2009-14039-C06-01. We also thanks INE (Spain) for accepting a visit of María Salome Hernández García during three months in 2010 and ISTAC (Canary Islands) for providing us with the real-world microdata used in the computational section of this article.

VII. References

[1] I. Arbués, M. González, P. Revilla (2010) “A class of stochastic optimization problems with application to selective data editing', Submited to “Optimization”.

[2] S. Delgado-Quintero, J.J. Salazar-González (2008) “A new approach for data editing and imputation”, Mathematical Methods of Operations Research 68:407-428.