CASC PROJECT

Computational Aspects of Statistical Confidentiality

June 25, 2003

Report on the testing of τ-Argus Version 2.2

Work Package 6 of the CASC project

Deliverable 6-D5

Giovanni Maria Merola

gmmerola@istat.it
ISTAT, MPS/D
Via C. Balbo, 16
00184, Roma
Italy

Report on the testing of τ-Argus version 2.1.1

Introduction
τ-Argus is a computer program that applies state-of-the-art Statistical Disclosure Control (SDC) techniques to tabular data, developed within the European Project CASC (Computational Aspects of Statistical Confidentiality) with the final goal of creating a tool usable by NSI's as well as other agencies. This testing of τ-Argus is Work Package 6 of the CASC project and was carried out on the intermediate release 2.2

Testing is a crucial phase in the development of software; this testing has been designed for evaluating τ-Argus especially with respect to its integration in the data production process of statistical institutions. That is, having in mind the final user and his/her needs. Hence, the testing has been designed to check for bugs and limitations, collect suggestions for improvements, assess the clarity of the documentation, verify the portability (input/output formats, platform required etc.) and user friendliness of τ-Argus. In this way, scores and comments expressed by testers become a vital source of information to help developers not only to fix bugs and malfunctioning, but also to improve the program towards meeting the requirements of end users. For this reasons testers were selected among potential users of the software, privileging the quality of testing over the quantity of tests run.

Description of the software
τ-Argus 2.2 applies SDC to tabular data, that is, to tables carrying aggregates of individual records containing confidential information. It can run under different Microsoft Windows platforms (95/98 and NT/2000).

The program is designed in such a way that SDC can be applied in steps, comparing the effect of different methods and different parameters values in the same run. The first step consists of reading in the data to be protected. This version of Tau is capable of reading the individual records (the *microdata*) that will be used to build tables but also tables themselves. Microdata can be supplied in fixed ASCII format or in comma separated format (*csv*). Tables can be fed specifying the classes of the spawning variables for each cell, the cell values and the status of the cells, whether safe, unsafe or protected. Additionally, frequencies and first *m* contributions can be provided, as well. The description of the input data, the *metadata*, can be edited from within the program using the *specify metafile* option or supplied with an external file (in *.rda* format). The second step consists of specifying the tables to be protected, defining the spawning variables, the response variables and the risk criteria in the *specify tables* window; the risk criteria can be chosen among: minimum frequency threshold, dominance rule and p-q rule. The third step is the actual protection of the tables. In the present version of τ-Argus the following methods are available: variable recoding and secondary suppression. This last method can be carried out using different routines: two heuristic ones, GHMiter (R. D. Repsilber 1994. Preservation of Confidentiality in Aggregated Data, Second International Seminar on Statistical Confidentiality, Luxemburg) and Network Flow, and two exact ones, which differ for the way they deal with hierarchical classes: one uses the fully optimal approach and the other one applies the HiTas algorithm (Fischietti, M. and J.J. Salazar-González (1998). Models and Algorithms for Optimizing Cell Suppression in Tabular Data with Linear Constraints. Technical Paper, University of La Laguna, Tenerife.) The exact routines require commercial LP-solvers, Tau allows the choice of either Xpress or Cplex. The last step of the protection of tables is the production of the output, which consists of protected tables and a report, in HTML format.

Outline of the testing set up and questionnaire

Version 2.2 of τ-Argus was tested by 11 testers, three of which external to the CASC project and from "non-EU" countries (New Zealand, Israel and Slovenia) on a volunteer basis. Testers had knowledge of SDC for tabular data and access to the existing literature. A set of test data was provided in order to have standardized evaluations but testers were asked to try the program on other real data representative of their institution's practices.

The questionnaire was designed as a MS Excel worksheet and was sent electronically. The completed questionnaire with the results of the testing had to be returned a month later.
The questionnaire is structured in 7 sections, following an ideal table protection operation path using the program. The sections were: 1) preliminary issues (tester's identity and installation; 2) data importing; 3) data specification; 4) data modification; 5) output of protected files; 6) documentation; 7) general remarks. Some questions required an evaluation with four possible scores (I=insufficient, S=sufficient, G=good and E=excellent), others a yes/no answer and the rest just a comment. All questions admitted the possibility of adding comments and suggestions. Further details can be found in the questionnaire with the instructions in Appendix A. Additional documents provided by the testers are contained in Appendix C.

Testing report

An overview of the testing results for each section of the questionnaire will be given in the following section. Detailed summaries of scores and comments for each question can be found in the summary sheet in Appendix B (Tau2.2-Report.xls). In this document columns F to K give the frequencies of the scores given to each question, scores less frequent than 3 are highlighted in Yellow, frequencies of 3 or 4 are highlighted in Orange and higher frequencies are highlighted in Red. Column L gives a "satisfaction" indicator obtained either assigning zero to "No" and one to "Yes" or assigning the numerical values 0, 1/3, 2/3 and 1 to the scores I, S, G and E, respectively. In this way, its values range from 0 to 1, and the closer they are to 1, the closer the mean judgement is to Excellent or Yes and vice versa. Column M gives the number of missing answers, column N contains a summary of the comments given by the testers, which are shown in columns from Y to AI, columns from O to X show the scores expressed by each tester.

Overview of the testing results

1) Preliminary issues (tester's identity, NSI's practices and installation)

This section was designed to get an idea of the testers' skills, the equipment they were using for testing, the needs of their institutions and the problems encountered in installing the software.

Most testers were statistical researchers with good knowledge of SDC for tabular data. All tests were run on reasonably fast processors with good amount of RAM (minimum 256MB), most of them running Windows 2000 or Windows 98; one was running windows XP on which he could successfully run Tau-Argus.

All the NSI's participating to the testing do release tabular data (mostly for business data but also for social data) usually with less than 4 dimensions. Some testers mentioned the release of linked tables. Most used risk measure is frequency threshold, some apply the dominance and p-rule. Data are protected mainly by recoding and suppression. Some apply also rounding and perturbation.

The program could be installed successfully by all testers. Some required better instructions for the installation.

2) Data importing

This section was designed to gather impressions on issues arising before working with the data. We feel that data and metadata importing is a crucial issue for users and, therefore, a few question insisted on this topic. This version of τ-Argus allowed the importation of microdata separated by commas (or other symbols) (csv files) and of tables.

Importing data in fixed format gave little problems. Some problems with importing csv files arose because of different decimal and field separators used in different countries. A possible solution to this problem could be the program reading the international setting on Windows. Some testers noted program crashes when wrong metadata are fed to the program. Most testers found the possibility of entering data in tabular form useful. However a series of problems were brought up, some of which make working with tables difficult.

3) Data specification

Data specification is a crucial step for any statistical analysis; it should be easy and unambiguous in order to set all analysis on firm ground and make comparisons easy. This section deals with editing the metadata, defining the tables to be protected and specifying the parameters for the sensitivity rules to be applied. The whole process leads to computing the tables and identifying the sensitive cells.

The *specify metadata* window for editing the metadata was found clear and easy to use, however it was requested that some options be made clearer. Also the definition of hierarchical levels needs a better explanation. A facility for defining easily NACE subsections was required.

The *specify tables* option is considered clear and easy to use. However, some functionality problems were spotted and the inclusion of frequency tables was requested. Some additions to the safety rules comprised were requested. The definition of the parameters for the p-q rule are not clear to all users. The meaning of some options needs a better explanation. The documentation is clear but some topics need deeper explanation and the information seems to be too much scattered around the manual.

Option *Table Metadata* some testers found inconsistencies with the parameters entered and those applied.

4) Data modification [Protection Methods]

Protection of unsafe cells is the primary purpose for which τ-Argus was developed. It comprises two methods: variable recoding and secondary suppression. Variables for which recode levels have been specified can be recoded manually choosing the levels, via a graphic three. Recoding is designed in such a way that changes can be visualized and undone easily. The computation of secondary suppressions is a hard optimization problem. Secondary suppression patterns can be computed by four different routines, some of which can handle also linked tables. As mentioned above, two of these algorithms require commercial LP-solvers while the other two exploit freely available routines. Table protection is designed in such a way that users can view the results and easily undo what has been done.

The *select table* window is clear and easy to use but some users would like more information on the tables.

The *view table* window was well liked by testers. Better explanation of some options was required.

The *recode* facility was found satisfactory although some problems with the original definition of classes after applying the recoding were reported.

Some testers had the Cplex LP-solver but Xpress has been chosen by most institutions. Some testers did not have any solver for testing. The testers who used Cplex or Xpress will stick to their choice while the others who expressed a preference would choose Xpress. Suppression methods work in general but gave different types of problems and the comments were contradictory. For both LP-solvers difficulties installing the licences have been experienced by all testers. The documentation for the suppression algorithms and for the installation of both solvers needs improvement.

The facility for *linked tables* was rated *good* by most testers who followed the procedure correctly, however some improvements were required, such as the extension of the facility to tables imported as such.

The documentation for the data modification section was found *good* overall but some improvements were suggested.

Overall this section was rated *good*. Some additions were required. The liking of the data protection capability of the program, which is the heart of it, increased from that for the previous version.

5) Output of Protected Tables

The output is the final product of the whole SDC process. It consists of creating protected tables to be released and a report on the protection of the data. Since the protected tables and report could be handed to people not familiar with τ-Argus and SDC, all items should be easy to understand and to explain to others.

The *save table* option was rated generally "Good", however most testers would like to have the output file to be consistent with the input, hence re-readable by the program itself. The *report window* was found complete and clear.

6) Documentation

Documentation of a program can hardly ever satisfy all users. In the case of scientific programs, such as τ-Argus, a balance must be found between theoretical explanations and guidance to users. Furthermore the scores given may be influenced by the level of theoretical knowledge of the tester. In this section we sought an overall impression of the documentation, where specific critiques can be found in each section.

The on-line help was found clear and useful but in need of improvements, particularly, more examples were required. The index for the search facility should include more items.

The manual is clear and helpful, however it needs to be more comprehensive and the removal of some repetitions would make it easier to read. In previous comments it was noted that information is sometimes scattered around the two parts and further methodological explanations were required. Not all testers could carry out all the functions of the program because of lack of information

7) Global statements about the program

This section asks testers to express their overall opinion on the program, in terms of the organization of the SDC process and compatibility with institutions' practices.

The testing of τ-Argus showed that all testers had a good impression of the software with respect to the way it was concealed and designed. However, it also revealed some problems. The organization of the process of protecting tables was rated *Good* or better and all institutions would adopt the program as standard tool for SDC, although only after some modifications or additions be applied, as, for example, batch execution. The overall rating was "*Good*" for all testers but one, who considers it insufficient because too prone to crashing.