WP. 11
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

Negative cell values, singletons and linked tables in Tau-Argus

Prepared by Anco Hundepool and Peter-Paul de Wolf, Statistics Netherlands

Negative cell values, singletons and linked tables in τ-ARGUS

Anco Hundepool* (ahnl@cbs.nl) and Peter-Paul de Wolf* (pwof@cbs.nl)

* Statistics Netherlands, Department of Methodology and Quality, P.O. Box 24500, 2490 HA Den Haag, The Netherlands.

Abstract. τ-ARGUS is a software program that deals with statistical disclosure control of tabular data. This software is being used by many National Statistical Institutes as well as by Eurostat. With this software one can apply several different SDC methods, with the main attention to secondary cell suppression techniques. Due to increasing insight in SDC methods, the software is continuously updated. Recently, some major improvements have been implemented mainly concerning the modular approach: it is now possible to deal with tables that have some negative cell values, a new solution for cell suppression to cope with the presence of singletons have been implemented and an automated way to deal with a set of linked tables is now available. In this paper we will describe those improvements and show the effects on some example tables.

1 Introduction

To be able to apply statistical disclosure techniques to tabular data, software is needed. The τ-ARGUS software is a widely used program: many National Statistical Institutes make use of it as well as Eurostat. τ-ARGUS has been developed as part of several European projects.

Recently, some major improvements have been implemented. These improvements do not concern the detection of (primary) unsafe cells using linear sensitivity rules, but are related to finding a suppression pattern for protecting a table with unsafe cells. These improvements mainly concern the modular approach. In the current paper, we will give a description of the major changes and show some of the resulting effects using example tables. The complete collection of improvements is part of τ-ARGUS from version 3.5 onwards (see Hundepool et al., 2011).

In section 2 we will give some details about the implemented improvements. Some of the improvements can have a significant effect on the results. In section 3 we will give some examples of such effects. Finally, in section 4 some conclusions will be drawn.

The views expressed in this paper are those of the authors and do not necessarily reflect the policies of Statistics Netherlands

2 Improvements

In this section we will describe the recent major improvements on the modular approach in τ-ARGUS. Since version 3.5 it is possible with the modular approach to deal with tables that have some negative cell values and to deal with sets of linked tables in an automated way. Additionally, the way the modular approach deals with singleton cells is improved.

Throughout this paper will use the following terminology concerning tables. An n-dimensional table is a crossing of n spanning variables $X_1, \ldots, X_n$. Each spanning variable X_j has K_j categories, denoted by $X_j(1), \ldots, X_j(K_j)$. The total of variable X_j over its categories is denoted by $X_j(0)$. A cell of an n-dimensional table is uniquely determined by the categories of the n spanning variables to which the contributors to that cell belong: $(X_1(i_1), \ldots, X_n(i_n))$. The values $i_1, \ldots, i_n$ are called the coordinates of the cell. Each coordinate i_j can thus attain the values $0, 1, \ldots, K_j$.

An interior cell is a cell for which none of the coordinates equals zero, i.e., $\prod_{k=1}^{n} i_k \neq 0$ and a marginal cell is a cell for which at least one coordinate equals zero, i.e., $\prod_{k=1}^{n} i_k = 0$.

A row R of a table is the subset of all cells of the table, with all but one coordinate fixed. I.e., in a three dimensional table of size $K_1 \times K_2 \times K_3$ a row could be e.g., the set of cells with coordinates $\{(i, 1, 0) : i = 0, \ldots, K_1\}$ or the set of cells with coordinates $\{(1, i, 3) : i = 0, \ldots, K_2\}$. Note that with this definition a column in a two dimensional table is also called a row. In this paper we will use the terms row and column interchangeably whenever we deal with a simple two dimensional table.

2.1 Negative cell values

Most models concerning statistical disclosure control of magnitude tables assume the data to be non-negative. Indeed, most linear sensitivity measures used to determine the unsafe cells only make sense when all underlying data are non-negative. See e.g., Hundepool et al. (2009) for a description of commonly used linear sensitivity rules.

However, in practice some contributions may be slightly negative due to e.g. correction techniques applied to the underlying microdata. In many magnitude tables this fact will not be visible to the end-user: aggregating the possibly negative *contributions* to a cell often results in a non-negative *cell* value. In some cases however, the cell value itself may still be negative.

When protecting a magnitude table, the first thing to do is to define which cells are (primary) unsafe. Whenever negative contributions are present, most linear sensitivity measures can not be applied unmodified. Only the minimum frequency rule can still be applied. The way τ-ARGUS has been improved does not involve this step of the process. It is assumed that the (primary) unsafe cells have been appointed and that 'only' the secondary suppressions need to be found. The improvement of τ-ARGUS concerns this step of finding a suppression pattern.

The original procedures to find an optimal suppression pattern, were not able to deal with negative cell values. Indeed, the original procedures just *assumed* that all cells were non-negative, without testing the validity of this assumption. In case of some negative cell

values, sometimes the original procedures were unable to find any suppression pattern (resulting in an error), sometimes a suppression pattern was constructed that was infeasible. Obviously, this had an effect on the modular approach as well, since that approach makes use of the optimal suppression routines for each individual subtable.

To circumvent this problem in the modular approach, τ-ARGUS now 'lifts' a subtable whenever at least one cell value is negative. That way, a feasible problem will be constructed with only non-negative cell values. That problem will be solved in the traditional way, after which the suppression pattern is imposed on the original subtable that had some negative cell values.

Lifting a subtable is described as follows:

- Determine the smallest (most negative) cell value of the subtable, C_L say.
- Add $-C_L + 1$ to the value of each interior cell of the subtable.
- Add $-C_L + 1$ to the apriori lower and upper bound of each interior cell of the subtable.
- Adjust the value and the apriori bounds of the marginal cells accordingly, taking the size of the table into account.

The protection levels are not adjusted, since it is assumed that the absolute values of the protection levels are needed and not the relative values.

2.2 Protection in case of singletons

Cells within a table sometimes consist of exactly one contributor. Such a cell is called a singleton. Linear sensitivity rules will usually label this cell as (primary) unsafe. When cell suppression is used to protect a table with unsafe cells, these singletons need to be taken care of in a special way.

Within a suppression pattern, contributors in singletons may be able to recalculate other suppressed cells. Obviously, a contributor could always insert its own contribution and thereby recalculate its own suppressed cell. This could in turn lead to the possibility of recalculating other suppressed cells in the same suppression pattern. Whenever such a recalculated cell is (primary) unsafe, this means disclosure.

Within the current models used to determine suppression patterns, it is not possible to take all possible situations into account when singletons are part of a suppression pattern. However, an important group of instances of disclosure by singletons, is when a singleton is part of a row R with exactly one additional (also primary) suppression.

To prevent this kind of disclosure, it would be sufficient to force an additional (third) suppression in the same row R. In prior versions of τ-ARGUS this was accomplished by increasing the protection levels of one of the (primary) unsafe cells in row R. In short, the protection level of one of the primary suppressed cells was raised in such a way that the other primary suppression would not be able to give sufficient protection. The largest

primary unsafe cell in row R got the *cell value* of the other unsafe cell in row R, plus a small value, as protection level. Indeed, this forces a third suppression in row R.

However, since the *cell value* of one of the suppressed cells was involved, this meant that the increased protection level of this cell could become quite large, which would have an effect on the suppression pattern in one of the other dimensions. In certain situations this led to oversuppression.

To circumvent this problem, the newly implemented approach adds a virtual cell to the table. That virtual cell is assigned a value equal to the sum of the two primary suppressed cells in row R, and is given the status '(primary) unsafe'. That virtual cell then only has to be protected against exact disclosure, i.e., it suffices to impose a small protection interval.

Table 1 shows an example table, displaying the singleton problem. In Table 1(a), the values of the cells are given, with in bold italic the (primary) unsafe cells. Table 1(b) shows the names of the cells, where c_{ij} stands for the cell with coordinates (i, j). Now

	Total	X1	X2	X3	X4
Total	227	76	33	93	25
A	146	52	***15***	62	***17***
B	81	24	18	31	8

(a): Cell values

	Total	X1	X2	X3	X4
Total	c_{00}	c_{01}	c_{02}	c_{03}	c_{04}
A	c_{10}	c_{11}	$\boldsymbol{c_{12}}$	c_{13}	$\boldsymbol{c_{14}}$
B	c_{20}	c_{21}	c_{22}	c_{23}	c_{24}

(b): Cell names

Table 1: Example table to explain Singleton Problem. Bold italic means (primary) unsafe.

assume that cell c_{12} = (A, X2) is a singleton and cell c_{14} = (A, X4) is unsafe according to a p%-rule with $p = 10$. Hence, cell c_{14} is the only other (primary) unsafe cell in that row. To protect cell c_{14} against disclosure by the contributor of singleton c_{12}, a 'virtual cell' c_v is defined with value 32. Moreover, that virtual cell is given a small protection interval, (32, 33) say. The relations that define the table structure, including the virtual cell, are given in Figure 1.

$$
\begin{aligned}
c_{00} &= c_{01} + c_{02} + c_{03} + c_{04} \\
c_{10} &= c_{11} + c_{12} + c_{13} + c_{14} \\
c_{20} &= c_{21} + c_{22} + c_{23} + c_{24} \\
c_{00} &= c_{10} + c_{20} \\
c_{01} &= c_{11} + c_{21} \\
c_{02} &= c_{12} + c_{22} \\
c_{03} &= c_{13} + c_{23} \\
c_{04} &= c_{14} + c_{24} \\
c_{v} &= c_{12} + c_{14}
\end{aligned}
$$

Figure 1: Relations defining table structure of Table 1

Within τ-ARGUS, this procedure is implemented in both the optimal approach as well as in the modular approach. For the modular approach, this procedure is applied to each subtable separately, whenever a subtable is dealt with within the modular approach.

This special attention to singletons is only given when the other suppressed cell in the same row is a 'true' primary suppression. This is natural, since it has to be done prior to the search for secondary suppressions. In the modular approach, a hierarchical table is divided into many, non-hierarchical, subtables. Secondary suppressions in one table sometimes temporarily become primary suppressions in other tables during the process. I.e., those suppression are not 'true' primary suppressions. It is therefore also natural not to construct virtual cells in case a singleton is in the same row with exactly one other primary suppression that was originally a secondary suppression. This is indeed the way it is implemented in the modular approach.

2.3 Set of linked tables

The third improvement of τ-ARGUS concerns the possibility to deal with sets of linked tables using the modular approach. This is an implementation of the method discussed in De Wolf and Giessing (2009) and De Wolf and Hundepool (2010).

Traditionally, τ-ARGUS can be used to protect a set linked tables by searching for suppression patterns for the tables successively. Any suppression pattern found for a particular table then has to be imposed on all other tables, prior to their protection. This procedure is iterative in its nature: whenever a suppression pattern for table T_i involves cells within any suppression pattern of a previously protected table T_j, table T_j has to be protected anew. This has to be repeated until no more changes in suppression patterns emerge. Obviously, this is rather time consuming and may depend on the order in which the tables are protected.

In the new method, this is all done automatically. One only has to indicate the way the tables are linked and then the complete set of tables will be protected in one run of τ-ARGUS. This new method cannot deal with sets of arbitrarily linked tables, but it can deal with an important class of linked tables, often seen at NSI's. A set of linked tables that this method can deal with, should be such that it will be possible to define a covering table. A covering table is an n-dimensional table of which the tables in the set of linked tables are all proper subtables. Basically, the modular approach is then applied to the covering table, but all subtables that are not part of the set of linked tables, will not be considered during the search for suppression patterns.

3 Examples

3.1 Singletons

To show the effect of the new implementation of cell suppression in the presence of singletons, we will apply both the old and the new approach to the example Table 1. Table 2(a) shows the suppression pattern when the old approach is applied, whereas Table 2(b) shows

the suppression pattern when the new approach is applied. We have used the cellvalue as the cost variable that measures the information loss. In both cases we see that the single-

	Total	X1	X2	X3	X4
Total	227	76	×	93	×
A	146	×	×	62	×
B	81	×	18	31	×

(a): Old approach

	Total	X1	X2	X3	X4
Total	227	76	33	93	25
A	146	×	×	62	×
B	81	×	×	31	×

(b): New approach

Table 2: Protection in the presence of singletons

ton (A, X2), which is in a row with exactly one additional (primary) unsafe cell (A, X4), leads to an additional suppression in that row. So indeed, the singleton is not able to break the protection pattern to disclose the other (primary) unsafe cell.

Clearly, the old approach leads to overprotection: more and larger cells are suppressed than necessary. This is easily explained: in the old approach the largest unsafe cell in row A gets the value of the smallest unsafe cell (plus epsilon) as its protection level, i.e., cell (A, X4) gets the protection interval $[1, 33]$. If cell (B, X4) would be the only additional suppression in column X4, this would lead to an upper bound of the feasibility interval of cell (A, X4) of 25, which is too small.

Note that in the old approach, marginal cells would have to be suppressed. In this simple example this has only the immediate effect on the table itself. However, when something like this happens when the modular approach is applied to a hierarchical table, this would yield a much larger effect. Indeed, it would lead to backtracking and hence most likely to additional suppressions all over the table.

In the new approach, the protection required for virtual cell c_v leads to an additional suppression in row A as well, as required. However this does not have any effect on the protection interval required for cell (A, X4). I.e., that protection interval is derived from the linear sensitivity rule that was used and hence equals $[16, 18]$. So in this case suppressing only cell (B, X4) in column X4 does give enough protection. Moreover, no additional backtracking is needed, if this had been a subtable of a hierarchical table in the modular approach.

3.2 Linked tables

The tables of the example we will discuss in this subsection are given in the Appendix. The example will show the effect of using the linked approach as opposed to not using the linked approach, i.e., applying cell suppression to each table on its own. We will consider a set of three tables, {T1, T2, T3}, where Table T1 is the link between the three tables. Table T1 is a one dimensional table with the most detailed spanning variable that is also present in the other two tables but with less detail. The other tables each have a different additional spanning variable.

In the examples we show both the solutions if the table are protected individually and when the linked tables approach is used. The problem of the linked tables arises when the protection of a table implies that cell will be suppressed that is present in more than one table. Often this happens when a marginal cell needs to be suppressed.

In Table T2 the cell (C, X5)* is (primary) unsafe with a protection interval of slightly less than 1% of the cell value. It is this cell that clearly shows the effect of the linked approach. Because of the required protection interval the other interior cells (C, X1)...(C, X4) are not enough to protect (C, X5) and hence the marginal cell (C, Total) has to be suppressed. And this has consequences for the other tables.

Using that linked approach, the protection of this cell causes four additional suppressions in Table T1 and five additional suppressions in Table T3. The new version of τ-ARGUS does this automatically in one run.

4 Conclusions

The improvements discussed in this paper all have a positive effect on the suppression patterns. The implementation of a method that deals with tables containing negative cell values makes it now possible to deal with those kind of tables. It should however be noted, that this implementation deals with finding *secondary* suppressions. In order to deal with negative cell values, the used sensitivity measures should be adjusted as well. This is however not part of the improvements discussed in this paper.

The way τ-ARGUS now deals with tables containing singletons generally yields less information loss. The example of subsection 3.1 clearly shows this effect in a very simple table. In more complex hierarchical tables this effect can be even much bigger.

Finally, it is now possible to deal with linked tables in τ-ARGUS using the modular approach. It has been common knowledge that linked tables should be treated consistently when applying a suppression method. For quite some time has has been time consuming: it had to be done by hand in an iterative way. The new method, as implemented, now deals with this situation in an automated way. This improves the efficiency of finding suppression patterns in sets of linked tables. The example shows that more cells need to be suppressed when the set of tables is protected simultaneously (as opposed to one at a time, independent of the other tables). This might seem to be a negative effect, but these additional secondary suppressions are really needed to get a suppression pattern that really protects all (primary) unsafe cells.

References

Hundepool, A., Wetering, A. van de, Ramaswamy, R., Wolf, P.P. de, Giessing, A., Fischetti, M., Salazar, J.J., Castro, J. and Lowthian, Ph. (2011). τ-ARGUS version 3.5 user manual, available at http://neon.vb.cbs.nl/casc/tau.htm

*This cell is highlighted in the appendix in Table T2.a.

Hundepool, A., Domingo-Ferrer, J., Franconi, L., Giessing, S., Lenz, R., Longhurst, J., Schulte Nordholt, E., Seri, G. and Wolf, P.P. de (2009), ESSNET handbook on Statistical Disclosure Control, ESSNET-SDC project, available at http://neon.vb.cbs.nl/casc/handbook.htm

Wolf, P.P. de and S. Giessing (2009). Adjusting the τ-ARGUS modular approach to deal with linked tables. Data & Knowledge Engineering, volume 68, pp. 1160–1174.

Wolf, P.P. de and A. Hundepool (2010). Three ways to deal with a set of linked SBS tables using τ-ARGUS. In: Domingo-Ferrer and Magkos (Eds.) *Privacy in Statistical Databases, Proceedings PSD 2010, Corfu*, LNCS 6344, Springer-Verlag Berlin Heidelberg, pp. 66–73.

Appendix

In this appendix, the three tables used in example 3.2 are given, T1, T2 and T3. Of each table, three versions are presented: the original table, the table protected using the linked approach and the table protected on its own, i.e., without taking the other tables in the set of linked tables into account. In the original tables, the (primary) unsafe cells are marked red. In the protected tables, the red crosses are the primary suppressions and the blue crosses are the secondary suppressions.

	Total
Total	310494024
A	98594438
A1	53658761
A11	5810499
A12	47848262
A2	5085959
A3	3473565
A4	36319246
A41	12803597
A42	22803597
A43	527537
A5	56907
B	485280
B1	360036
B2	125244
C	211414306
C1	30058974
C2	177248232
C21	48646824
C22	1471
C23	128599937
C3	4107100

T1.a: Original table

	Total
Total	310494024
A	×
A1	53658761
A11	5810499
A12	47848262
A2	5085959
A3	×
A4	36319246
A41	12803597
A42	22803597
A43	527537
A5	56907
B	×
B1	360036
B2	×
C	×
C1	30058974
C2	177248232
C21	×
C22	1471
C23	×
C3	×

T1.b: Linked approach

	Total
Total	310494024
A	98594438
A1	53658761
A11	5810499
A12	47848262
A2	5085959
A3	3473565
A4	36319246
A41	12803597
A42	22803597
A43	527537
A5	56907
B	485280
B1	×
B2	×
C	211414306
C1	30058974
C2	177248232
C21	×
C22	1471
C23	×
C3	4107100

T1.c: On its own

Table T1

	Total	X1	X2	X3	X4	X5
Total	310494024	93255	965706	4884192	22569500	281981371
A	98594468	82813	869263	4565717	20978979	72097666
A1	53658761	54805	529378	3356967	14906348	34811263
A2	5085959	7046	60888	283250	1329901	3404874
A3	3473565	383	–	–	94009	3379173
A4	36319246	20362	277319	925500	4593709	30502356
A5	56907	217	1678	–	55012	–
B	485280	461	1646	12865	125683	344625
B1	360036	461	1646	12865	125683	219381
B2	125244	–	–	–	–	125144
C	211414306	9981	94797	305610	1464838	209539080
C1	30058974	4810	35682	24081	285419	29708982
C2	177248232	770	2675	18355	281137	176945295
C3	4107100	4401	56440	263174	898282	2884803

T2.a: Original table

	Total	X1	X2	X3	X4	X5
Total	310494024	93255	965706	4884192	22569500	281981371
A	×	82813	869263	4565717	20978979	×
A1	53658761	54805	529378	3356967	14906348	34811263
A2	5085959	×	×	283250	1329901	3404874
A3	×	×	–	–	×	×
A4	36319246	20362	277319	925500	4593709	30502356
A5	56907	×	×	–	×	–
B	×	×	×	12865	125683	×
B1	360036	×	×	12865	125683	×
B2	×	–	–	–	–	×
C	×	×	×	305610	1464838	×
C1	30058974	4810	35682	24081	285419	29708982
C2	177248232	×	×	18355	281137	176945295
C3	×	4401	×	263174	898282	×

T2.b: Linked approach

	Total	X1	X2	X3	X4	X5
Total	310494024	93255	965706	4884192	22569500	281981371
A	×	82813	869263	4565717	20978979	×
A1	53658761	54805	529378	3356967	14906348	34811263
A2	5085959	×	×	283250	1329901	3404874
A3	×	×	–	–	×	×
A4	36319246	20362	277319	925500	4593709	30502356
A5	56907	×	×	–	×	–
B	×	×	×	12865	125683	×
B1	360036	×	×	12865	125683	×
B2	×	–	–	–	–	×
C	×	×	×	305610	1464838	×
C1	30058974	4810	35682	24081	285419	29708982
C2	177248232	×	×	18355	281137	176945295
C3	×	4401	×	263174	898282	×

T2.c: On its own

Table T2

	Total	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8
Total	310494024	19132304	8958165	138494788	2568938	31457760	13392799	80467221	16022049
A	98594438	682390	8883025	14706515	2440214	31218563	12800052	16625947	11237732
A1	53658761	617623	4428895	10784528	1694671	12295059	6606372	10812944	6418669
A2	5085959	42600	405728	1328125	62283	1329366	122450	1165469	629938
A3	3473565	–	–	–	–	2510372	755090	33958	174145
A4	36319246	22167	4048402	2593645	683260	15045929	5316140	4594723	4014980
A5	56907	–	–	217	–	37837	–	18853	–
B	485280	–	39557	364794	11372	7222	–	628	61707
B1	360036	–	39557	239550	11372	7222	–	628	61707
B2	125244	–	–	125244	–	–	–	–	–
C	211414306	18449914	35583	123423479	117352	231975	592747	63840646	4722610
C1	30058974	89472	–	13036240	70165	164242	6483	13185706	3506666
C2	177248232	18360442	–	110387239	28085	701	7769	47316576	1147420
C3	4107100	–	35583	–	19102	67032	578495	3338364	68524

T3.a: Original table

Table T3

	Total	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8
Total	310494024	×	8958165	138494788	×	31457760	13392799	80467221	16022049
A	×	682390	8883025	×	×	31218563	×	×	11237732
A1	53658761	617623	4428895	10784528	1694671	12295059	6606372	10812944	6418669
A2	5085959	×	405728	×	×	×	122450	×	×
A3	×	–	–	–	–	×	×	×	×
A4	36319246	×	4048402	2593645	×	15045929	5316140	4594723	4014980
A5	56907	–	–	×	–	×	–	×	–
B	×	–	×	364794	×	7222	–	×	×
B1	360036	–	×	×	×	7222	–	×	×
B2	×	–	–	×	–	–	–	–	–
C	×	×	×	×	117352	231975	×	×	×
C1	30058974	×	–	×	×	164242	6483	×	×
C2	177248232	×	–	×	×	×	×	×	×
C3	×	–	×	–	×	×	×	×	68524

T3.b: Linked approach

	Total	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8
Total	310494024	×	8958165	138494788	×	31457760	13392799	80467221	16022049
A	98594438	682390	8883025	×	×	31218563	×	×	11237732
A1	53658761	617623	4428895	10784528	1694671	12295059	6606372	10812944	6418669
A2	5085959	×	405728	×	×	×	122450	×	×
A3	3473565	–	–	–	–	×	×	×	×
A4	36319246	×	4048402	2593645	×	15045929	5316140	4594723	4014980
A5	56907	–	–	×	–	×	–	×	–
B	485280	–	×	364794	×	7222	–	×	×
B1	×	–	×	×	×	7222	–	×	×
B2	×	–	–	×	–	–	–	–	–
C	211414306	×	×	×	117352	231975	×	×	×
C1	30058974	×	–	×	×	164242	6483	×	×
C2	177248232	×	–	×	×	×	×	×	×
C3	4107100	–	×	–	×	×	×	3338364	68524

T3.c: On its own

Table T3