Statistical Disclosure Control for Tabular Data in R

Kazuhiro MINAMI (kminami@ism.ac.jp)
The Institute of Statistical Mathematics / National Statistics Center, Japan

Yutaka ABE (yabe3@nstac.go.jp)
National Statistics Center, Japan

ABSTRACT

To perform statistical disclosure control (SDC) on tabular data is a challenging task because we need to ensure that every suppressed cell of a table has a sufficient width of a confidentiality interval under the presence of linear relations among cell variables. However, we find that the existing SDC tool (i.e., τ-ARGUS) does not effectively support an output checking process of the on-site use program in Japan. We therefore develop a new SDC tool in R, which produces safe tabular data with auxiliary information that is necessary for an output checker to verify its safety.

In this paper, we describe the major features of our SDC tool and discuss possible extensions in the future. Our SDC tool performs primary suppressions on a frequency table and a magnitude table with the minimum frequency rule and an occupancy rule (e.g., (n,k)-rule), respectively. We implement the optimal secondary suppression mechanism based on the technique of Benders decomposition.

Keywords: *Statistical Disclosure Control, Output checking, Cell suppressions, Linear programming*

JEL Classification*: Z00*

1. INTRODUCTION

In Japan, we start an on-site use program this year so that a researchers visiting secure on-site facility can perform exploratory analysis accessing microdata of various public surveys. Since microdata contains sensitive information, we perform output checking to verify the safety of analysis results before they are taken out from the facility. We find that to check the safety of tabular data challenging because we need to ensure that every suppressed cell of a table possesses a sufficient width of a confidentiality interval under the presence of linear relations among cell variables. Also, we need to reduce information loss of the original table by minimizing the number of suppressed cells. Such involved tasks requiring an optimization algorithm in linear programming definitely needs a software tool that handles them automatically.

τ-ARGUS [1], which is developed by Statistics Netherlands, is a software tool for conducting statistical disclosure control (SDC) on tabular

data. τ-ARGUS supports both primary and secondary suppression of tabular data while providing a researcher with a rich set of options for occupancy rules for determining sensitive cells and for algorithms of secondary cell suppression. τ-ARGUS is mainly designed to serve the needs of a researcher who are familiar with SDC techniques so that she can interactively obtain the most useful result by examining various options in τ-ARGUS.

However, we find that τ-ARGUS is not suitable for a task of output checking for our on-site use program where we separate the role of output checking from a researcher; that is, a researcher is responsible for supplying enough evidences to convince an output checker of the safety of a submitted table. In addition, there is a usability issue in τ-ARGUS when a researcher produces a table with another analytic tool such as SAS or R. To import a table to τ-ARGUS, we need to prepare a meta file to specify the format of the table. Also, we need to supply the values of the largest and the second largest units in each cell to check the safety of a magnitude table with an occupancy rule (e.g., *p%* rule).

We thus develop a new SDC tool for tabular data in R, which supports both primary and secondary cell suppressions for two-dimensional frequency and magnitude tables. Since our tool is implemented as a set of R functions, a researcher who prepares tabular data in R can seamlessly work on SDC tasks in the same environment. Our public function for cell suppression takes an original table as an input and outputs a suppressed safe table with auxiliary information for an output checker to verify its safety later. We also provide a tailor-made function for producing a magnitude table. That function also outputs two tables of the largest and the second largest unit in each cell of the magnitude table, since we need those two additional tables to check sensitive cells of the table with an occupancy rule.

We implement a secondary suppression algorithm that minimizes the number of suppressed cells. To solve linear programming problems efficiently, we adopt the approach of Castro [2] based on Benders decomposition [3].

The rest of the paper is organized as follows. Section 2 describes the on-site use service in Japan and Section 3 introduces issues of using τ-Argus for our on-site use service. Section 4 describes the major functions of our SDC tool in R. Section 5 discusses our future work and Section 6 concludes.

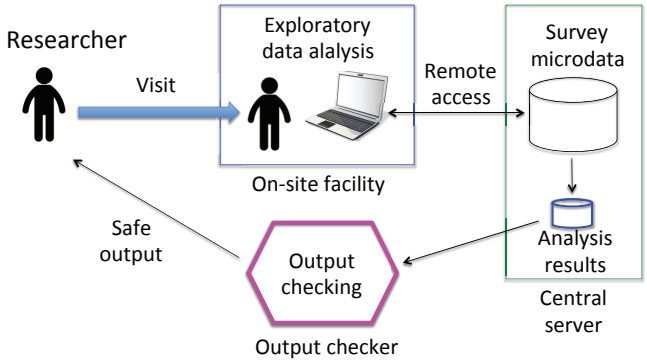
2. ON-SITE USE SERVICE IN JAPAN

In Japan, we launch a trial phase of the on-site use service in 2017, as shown in Figure 1. A researcher visiting an onsite facility can remotely access micro data on the server via a thin client terminal to conduct exploratory analysis. The researcher stores analysis results on the server and applies for output checking to take the analysis results out of the facility.

We base our output checking rules on Eurostat's guideline [4]. As for tabular data, we determine sensitive cells of a frequency table and a magnitude table with a minimum frequency rule and an occupancy rule respectively. Also, we explicitly set a threshold value for the width of a confidentiality interval for each suppressed cell of the table.

On-site use service in Japan

Figure 1



3. ISSUES IN T-ARGUS

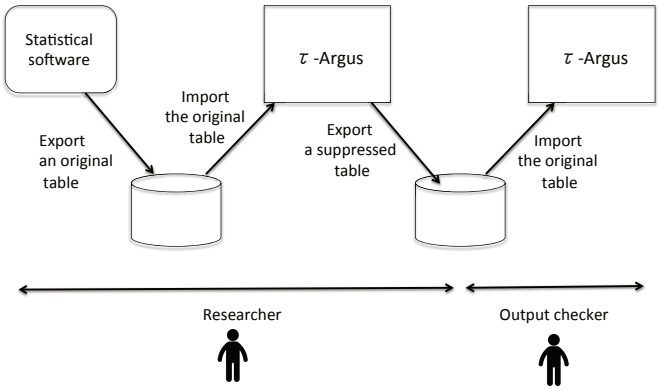
τ-Argus is a software for performing disclosure control on tabular data. τ-Argus is mainly designed for researchers with strong background on statistical disclosure control. Since τ-Argus supports several variants of occupancy rules, such as the threshold rule, *(n,k)*-rule, *p*% rule, for magnitude tables, a researcher can compare results of sensitive cells to be primarily suppressed by applying different rules. Also, τ-Argus allows a researcher to choose a solver for linear programming among multiple choices to conduct secondary suppressions.

However, there are several issues with τ-Argus. The first issue is that it is tedious to import a table, which is produced by a different analytic tool. Although τ-Argus has the capability to construct a magnitude table from imported micro data, we expect that the majority of researchers use popular statistical software such as R, SAS, and STATA to produce tabular data. If we use τ-Argus to protect tabular data, a researcher needs to import a table to τ-Argus, as shown in Figure 2, by preparing a meta file to specify the format of the table. Also, the researcher needs to prepare a pair of (the largest unit value, the 2nd largest unit value) for each cell of the magnitude table. Otherwise,

τ-Argus cannot perform primary suppression on the table. After the researcher performs secondary suppression on the table, he must export the suppressed table into a file again so that an output checker can examine it. An output checker needs to import that table again to check the safety of the submitted table.

Task flow with τ-Argus

Figure 2



The second issue is a user interface of τ-Argus, which gives many options concerning occupancy rules of tabular cells and optimization methods. Since we conduct output checking, we use the same occupancy rule with the fixed parameters, which should not be modified by a researcher. Thus, it is not desirable to show the user interface of τ-Argus to researchers. The third issue is the functional limitation of τ-Argus. τ-Argus does not support rules for preventing group disclosures, which we plan to adopt to our on-site service.

4. SDC TOOL IN R

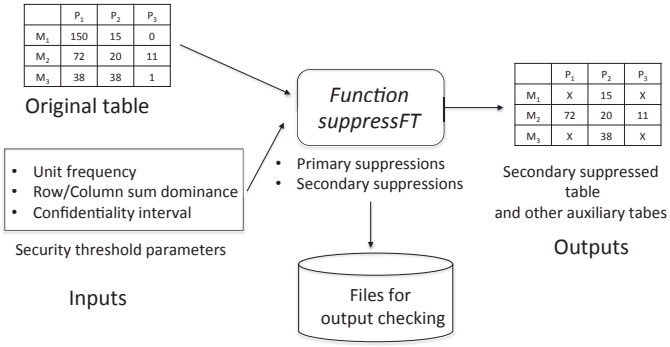
We develop a prototype tool in R to address the issues of τ-Argus we describe in Section 3. We provide several functions for primary and secondary suppressions in R, and, thus, as long as a researcher produces tabular data in R, we do not have any issue of converting data format of tabular data. Our tool, which is a set of R functions, does not provide any GUI interface as τ-Argus does, but it exports all necessary information to files so that an output checker later verifies the safety of submitted tables.

4.1 Checking frequency tables

We first introduce the function *suppressFT*, which performs primary and secondary suppressions on a frequency table. Figure 3 shows the functionality of the function *suppressFT*, which takes as inputs an original table and four security threshold parameters for unit frequency, row sum dominance ratio, column sum dominance ratio, and confidentiality interval, and outputs a secondary suppressed table and the confidentiality intervals of the suppressed cell variables. The function also exports the same information to the file as shown in Figure 4 so that an output checker can refer them to check the safety of the suppressed table.

Function *suppressFT*

Figure 3



Exported files by the function *suppressFT*

Figure 4

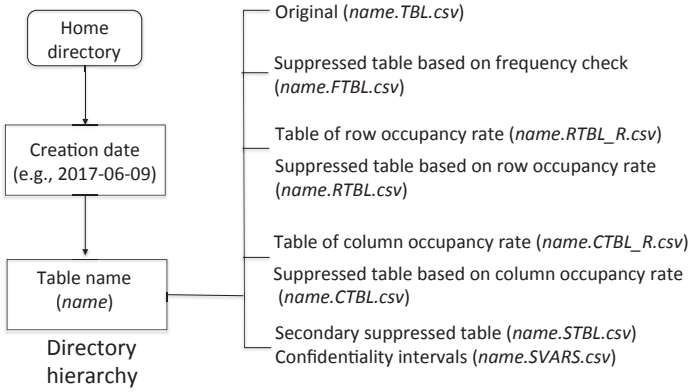
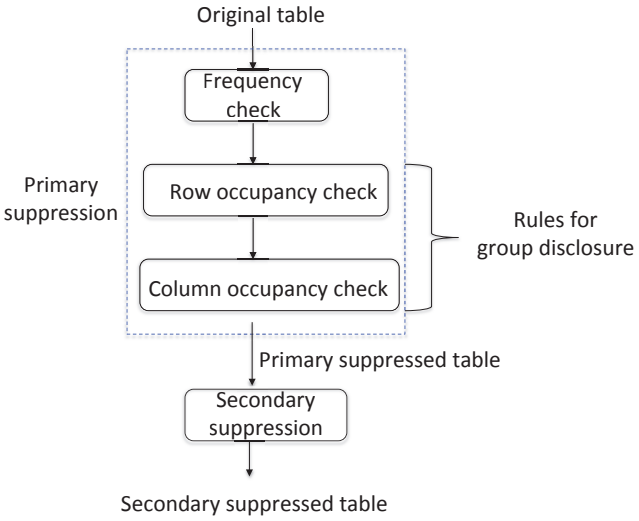


Figure 5 shows the flow of suppression process, which consists of four steps. At a high level, the process is divided into two stages: the primary suppression process and secondary suppression process. The primary suppression process is further broken down into three steps: frequency checking, row sum dominance checking, and column sum dominance checking. An original table is incrementally suppressed at each step of the primary suppression process, and then the primary suppressed table is further suppressed such that the confidentiality interval of each suppressed cell is greater than the specified threshold interval.

Incremental suppressions on a frequency table

Figure 5



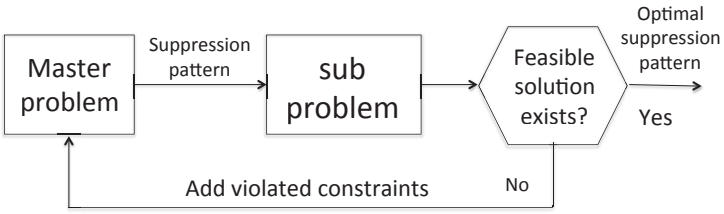
The three steps in the dotted rectangle consist of the primary suppression stage. After the primary suppressions are performed, the suppressed table is further suppressed at the secondary suppression stage

We implement an optimal secondary suppression algorithm based on Benders decomposition proposed by Castro [2]. When there are n cells in a table, there are 2^n suppression patterns to be considered. Therefore, to find the optimal cell suppression is a NP-hard problem, which means that there is no algorithm, which runs in polynomial time at the worst case. However, linear programming algorithm based on Benders decomposition runs efficiently for most of the cases and outputs a suppression pattern that is guaranteed to be optimal.

Figure 6 describes the overview of our optimization algorithm, which adopts Benders decomposition. We divide the original program of finding an optimal suppression pattern into two problems: a master problem and a subprolem. The master problem asks us to find the optimal suppression pattern and the subproblem asks us to check if a given suppression pattern satisfies all constraints of cell variables. The main idea is that although there are huge number of candidate suppression patterns, we can check whether a given suppression pattern satisfies constraints concerning the confidentiality interval of each suppressed cell efficiently.

Overview of the optimization algorithm for cell suppression

Figure 6



The master problem of the algorithm in Figure 6 begins with no constraints, and thus the suppression pattern where the bits for primary suppressed cells are '1's is passed to the subproblem. The subproblem checks whether the given input pattern satisfies all constraints. If there is a feasible solution satisfying all the constraints, that suppression pattern is optimal. Otherwise, the violated constraints are added to the master problem and we iterate the same process again until we get the optimal solution.

Since Castro's paper [2] only gives a high-level description of the algorithm, we have to fill in many details to implement the algorithm. We find that equation (13) of the paper, which shows the dual representation, contains an error in converting the relation $Ax = 0$ in equation (12) improperly.

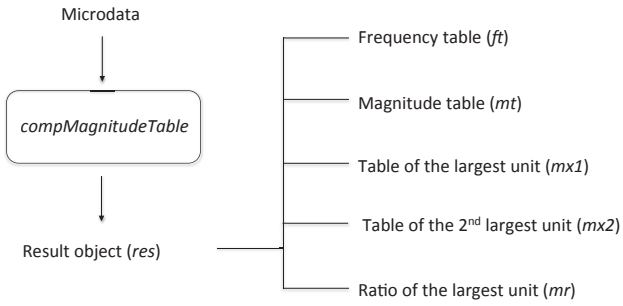
4.2 Checking magnitude tables

To suppress magnitude tables requires more involved procedures. Since to check an occupancy rule on a magnitude table requires the largest and the second largest unit values of each cell of the magnitude table, we implement the function *compMagnitudeTable* in Figure 7, which takes microdata as an input and outputs a magnitude table and tables and the largest unit and the second largest units.

Figure 8 shows the procedure for suppressing a magnitude table. A magnitude table is primary suppressed based on dominance rule (e.g., *(2, k)*-dominance rule, *p*% rule, etc.). Separately, the corresponding frequency table must be suppressed as well, and the cell positions of suppressed cells in the frequency table must be incorporated into the result of primary suppressions for the magnitude table. We finally perform secondary cell suppressions on this merged magnitude table and obtain the final result.

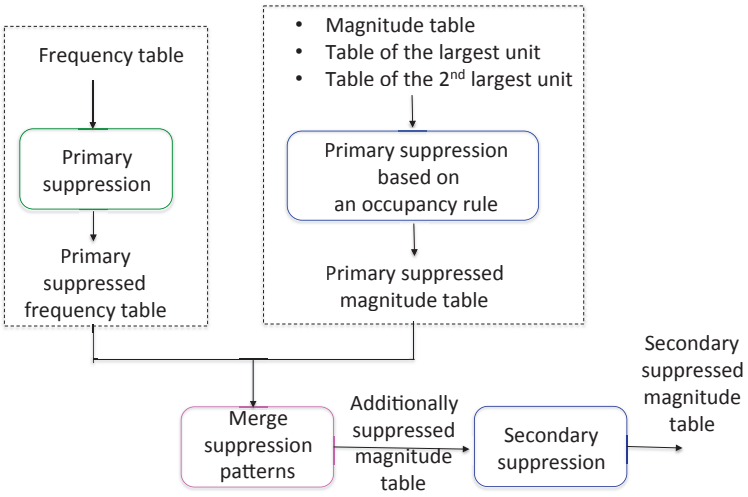
Function *compMagnitudeTable*

Figure 7



Procedure for suppressing a magnitude table

Figure 8



5. FUTURE WORK

We are currently extending our tool to support higher dimensions of tables because researchers often publish tables whose rows and/or columns are hierarchically structured with multiple variables. Such tables are essentially equivalent to those of multiple dimensions. Once we finish this task, we plan to develop a mechanism for preventing differential attacks [5,6] where a malicious person steals sensitive information by taking the delta of multiple tables. Although it is impossible to keep track of all similar tables available publically, we think that to check multiple tables from the same researcher is possible. Our basic approach is that, taking multiple tables as inputs, we construct an integrated table by introducing more detailed categorical variables that correspond to the overlaps of the variables from all input tables. We also plan to port the same functionality to other platforms for data analysis, such as SAS.

5. CONCLUSION

We present major functions of our SDC tool in R. Our tool is designed to automate the process of output checking for the on-site use service in Japan. We identify several issues in τ-Argus and address those issues in our tool. Since we implement our tool as a set of R functions, a researcher conducting data analysis in R can easily produce safe outputs by calling our SDC functions.

We find that a process of checking a magnitude table is much more complicated than that of a frequency table, and thus provide a tailor-made function for creating a magnitude table with auxiliary tables that prove the safety of the magnitude table. We also need to incorporate suppression patterns of the corresponding frequency table into that of the magnitude table.

As future work, we plan to check the usability of our tool with researchers using our on-site service while extending our tool to support higher dimension of tables.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Numbers JP15K00195 and JP16H02013.

References

1. τ-Argus homepage. http://neon.vb.cbs.nl/casc/tau.htm.
2. **Jordi Castro**, 2012, Recent advances in optimization techniques for statistical tabular data protection. European Journal of Operational Research, 216(2):257-269.
3. **Benders, J. F.**, 1962, Partitioning Procedures for Solving Mixed-Variables Programming Problems, Numerische Mathematik, Vol. 4.
4. **A. Hundepool, J. Domingo-Ferrer, L. Franconi, S. Giessing, R. Lenz, J. Naylor, E. S. Nordholt, G. Seri, and P.-P. D. Wolf**. Handbook on statistical disclosure control.

5. **Peter-Paul De Wolf and Anco Hundepool**. 2010, Three ways to deal with a set of linked sbs tables using τ-ARGUS. In Proceedings of the 2010 In- ternational Conference on Privacy in Statistical Databases, PSD'10, pages 66–73, Berlin, Heidelberg.Springer-Verlag.
6. **Alessandra Capobianchi and Luisa Franconi**., 2009, Cell suppression in linked tables from structural busi- ness statistics using tau argus 3.3.0: a conceptual framework. In Proceedings of the New Techniques and Technologies for Statistics(NTTS) 2009 Con- ference, .