CONFERENCE OF EUROPEAN STATISTICIANS

Joint UNECE/Eurostat Work Session on Statistical Data Confidentiality Session 2

29-31 October 2019, the Hague, the Netherlands 21 October 2019

Concepts for generalising tools implementing the cell key method to the case of continuous variables

Sarah Giessing, Reinhard Tent (Destatis, Germany)

Sarah.Giessing@destatis.de, Reinhard.Tent@destatis.de

Abstract and Paper

Funded by the EU project "Open Source tools for perturbative confidentiality methods" a cell key method useable for protecting frequency count tables by additive noise has been implemented in the package τ-Argus and as separate R package cellKey. An additional objective of the project was extending these implementations to the case of continuous variables. The paper addresses the main methodological issues of such an extension: In the case of magnitude variables, usually one applies multiplicative instead of additive noise. Typically, the noise is applied to the contributions of units with the highest assumed disclosure risk, but in some situations simpler options might be acceptable. Of course, there ought to be parameters controlling the magnitude of the multiplicative noise. The most important issue, however, is how to interface with noise design. The options we consider are Normal or Laplacian distributed noise, but also a generalisation of the noise design implemented in the R package ptable. The paper will propose an extended version of this package, able to provide perturbation tables with transition probabilities for non-integer valued data. In particular we suggest in which cases, and then, how to combine distributions provided in a perturbation table.

Concepts for generalising tools implementing the cell key method to the case of continuous variables

Sarah Giessing[*] Reinhard Tent[**],
[*] Statistisches Bundesamt, 65180 Wiesbaden, Germany, Sarah.Giessing@destatis.de
[**] Statistisches Bundesamt, 65180 Wiesbaden, Germany, Reinhard.Tent@destatis.de

Abstract: It was one objective of the EU project "Open Source tools for perturbative confidentiality methods" to extend implementations of the cell key method useable for protecting frequency count tables by additive noise implemented in the package τ-Argus and the separate R package cellKey to the case of continuous variables. The paper addresses the main methodological issues of the extension: In the case of magnitude variables, usually one applies multiplicative instead of additive noise. Typically, the noise is applied to the contributions of units with the highest assumed disclosure risk. Of course, there ought to be parameters controlling the magnitude of the multiplicative noise. The most important issue, however, is how to interface with noise design. We mainly consider a generalisation of the entropy maximization noise design implemented in the R package *p*table. The paper proposes an extended version of the *p*table package, able to provide perturbation tables with transition probabilities for non-integer valued data. In particular we explain how to apply convex combination of noise drawn from discrete noise distributions to generate noise for continuous data.

1. Introduction

The cell key method for statistical disclosure limitation by random noise is a well-known post-tabular perturbative disclosure control method for frequency count tables. Funded by the EU project "Open Source tools for perturbative confidentiality methods" a cell key method useable for protecting frequency count tables by additive noise has been implemented in the package τ-Argus and as separate R package cellKey (Meindl et al., 2018). Both packages rely on the R-package *p*table (Enderle and Giessing, 2019) to compute random distributions by maximizing entropy (Giessing, 2016; Marley and Leaver, 2011).

These basic implementations of the cell key method have now been extended to the case of continuous variables. Assuming some familiarity with the methodological concepts of the basic implementations, c.f. Giessing et al. (2018) (Appendix 1), in the present document we explain the most relevant methods and facilities of the enhanced implementation. For further details see Giessing et al. (2019).

The extended packages offer fairly generic implementations that would be useful for a wide range of applications. We have also designed the parametrization of extended packages in such a way that – along with some (future) guidelines – developing a suitable setting should be as intuitive as possible and manageable also for NSI's with little or no previous experience in the design of post-tabular random noise. Notably, the extensions require an extended version of the *p*table package as well, in particular also of the interface format of the package which provides cumulated transition probabilities.

We also considered suggestions of Ma et al (2016) for using different noise distributions for cells with even and odd number of contributions: If two cells A and B differ by only one respondent, the number of respondents will be even for either A or B, and odd for the other one. Using different noise distributions can therefore in such cases reduce disclosure risks of a specific differencing attack. See sec. A.1.3 for some more illustration of the issue and reference.

Section 2 introduces the concept of noise factors implementing a kind of noise that could be regarded on one hand as leading to *constant noise variance coefficients* (vs. the *constant noise variance* applied with a cell key method for protecting frequency tables). On the other hand, we allow some flexibility of those coefficients. How to use and adapt these basic principles in specific situations, like when non-negativity of the perturbed data is a requirement to be guaranteed, is the subject of section 3. Before finishing with a short summary, section 4 finally explains how the extended packages apply convex combination of noise drawn from discrete noise distributions to generate noise for continuous data building on a generalisation of.

2. Noise factors: Noise variance vs. noise variance "coefficient"

A fundamental concept of random noise for *frequency* count tables formulated in Fraser and Wooton (2005) and implemented in the R-package *p*table that supports the noise design of the implementation of CKM for frequency tables in τ-Argus and cellKey is that the noise variance should be constant for all table cells.

For tabulations of *continuous (magnitude)* variables, this principle will usually not be suitable: it will lead to either not enough protection for table cells with large contributions, or to too much protection (e.g. high information loss) for table cells with small contributions. Literature typically suggests multiplicative noise where – unlike in the case of the additive noise for frequency tables – not the noise variance, but a kind of noise *variance coefficient* should be – more or less – constant.

For example, the table builder of Thompson et al. (2013) implements noise that can be regarded as computing the noise by adding the sum of $topK$ random variables $\hat{X}_j$ $(j = 1,..,topK)$, where each component $\hat{X}_j$ has a fixed conditional distribution, i.e. conditional on the (weighted) top-*k*th contribution $w_j y_j (= x_j)$ to a table cell with ordered contributions $(|x_1| \geq |x_2| \geq ..)$. The coefficients $\frac{\sqrt{\mathrm{Var}(\hat{X}_j|x_j)}}{x_j}$ are constant. According to Thompson et al. (2013) this constant will be the user defined parameter m_j typically representing a percentage (of, e.g. x_j). In this document, we basically follow this concept. For ease of notation (and practicality), in the following we assume proportionality of the parameters, i.e. $m_j = \varepsilon_j m_1$ for $j = 2,..,topK$ with $0 \leq \varepsilon_j \leq 1$, and $\varepsilon_1 = 1$.

With these definitions we can express a realization of the jth noise component $\hat{X}_j$ as (2.1) $\hat{X}_j := x_j \varepsilon_j m_1 v_j$, with v_j a realization of a random variable V (c.f. sec. 4 for further discussion of how to design a suitable variable V), factor x_j the (weighted) top-*k*th contribution to the table cell to be protected by the noise, and factors ε_j and m_1 parameters to be fixed by the disseminator. Typically, x_j will be the largest factor of the noise (component).

As often top-*k*th observations are considered to have the highest disclosure risk, it is usually the most sensible approach to employ them as noise factors. However, we also included the following three alternative noise factor concepts that can be useful in special situations:

(1) Especially when the intention is to use the perturbed magnitude data to compute a mean value to data typically provided by banded distributions (like "Age"), the *difference between largest and smallest contribution* in a table cell as noise factor offers some nice properties.

(2) When data is not very skewed, the *cell mean* might be a simple, feasible option for a noise factor. It could be computed even with table builder packages not offering computation of largest contributors.

(3) In a very simple concept, the *cell value* itself might be considered as noise factor.

In either of these cases, we formally set $topK := 1$, and assume x_1 (in formula (2.1)) to represent the noise factor of the selected option (1) to (3).

Now, in order to introduce more flexibility, in sec. 2.1 we recall a concept of Giessing (2012) which would define coefficient m_1 as function of the respective noise factor[1] x_j, e.g. $m\left(x_j\right)$.

2.1 A flex-function for noise variance coefficients[2]

The idea of the flex-function is based on the consideration that for large observation x_j the product $x_j \varepsilon_j m_1 v_j$ which determines the absolute amount of the noise becomes quite large even for smaller "percentages" $\varepsilon_j m_1$. On the other hand, choosing a

[1] In case of the alternative concepts (1) to (3) above, of course we assume $topK = j = 1$.

[2] In this section we generally assume $x_j \geq 0$. For $x_j < 0$, let $m_1\left(x_j\right) = m_1\left(|x_j|\right)$.

small m_1 parameter to avoid too much perturbation leads to very low noise amounts (i.e. eventual underprotection) for smaller observations x_j.

Formula (2.2) defines a noise coefficient function $m(z)$ we refer to in the following as "*flex-function*". For $z \geq z_f, z_f$ referred to as "*flex-point*" , the flex-function $m(z)$ computes the m_1 parameter as a decreasing function of z. Apart from the flex-point, the parameters of the function are the desired maximum of the noise coefficient, σ_1 (reached at the flex-point), its lower limit σ_0 for large z and a parameter $q \geq 1$ affecting the shape of the function. For smaller observations below the flex-point we set $m(z) := \sigma_1$.

$$(2.2\,)\;\; m(z) := \begin{cases} \sigma_0 \left(1 + \frac{\sigma_1 z - \sigma_0 z_f}{\sigma_0 z_f} \left(\frac{2z_f}{z_f + z}\right)^q\right) & \text{for } z \geq z_f \\ \sigma_1 & \text{for } z < z_f \end{cases}.$$

See figure 1a below for illustration.

However, if observations may become very small, or even zero, the absolute noise amount determined by $x_j \varepsilon_j m(x_j) v_j$ $(= x_j \varepsilon_j \sigma_1 v_j)$ still may become eventually too small. As a way out we suggest to further modify the concept: For very small $x_1 \leq z_s$, with "*separation point"* z_s given by
(2.3) $z_s := \frac{\sigma_2}{\sigma_1\sqrt{E}}$ and $E := 1 + \sum_{j=2,..,topK} {\varepsilon_j}^2$, let

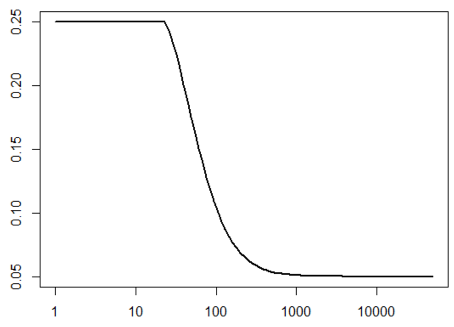
(2.4) $\hat{X}_1 := u$, with u a realization of a random variable U with noise variance ${\sigma_2}^2$ [3] .
To keep it simple, for $x_j \leq z_s$ when $j > 1$, let $\hat{X}_j := 0$.

Assuming a "standard" variance of 1 for the random component v_j ,with this definition of z_s we get for observation tuple $(x_j = z_s)_{j=1,\ldots,TopK}$the same noise variance of $\sum_{j=1,..,T} {z_s}^2 {\varepsilon_j}^2 {\sigma_1}^2 = {z_s}^2 {\sigma_1}^2 E = \left(\frac{\sigma_2}{\sigma_1\sqrt{E}}\right)^2 {\sigma_1}^2 E = {\sigma_2}^2$ as the noise variance ${\sigma_2}^2$ we get for tuples $(x_j < z_s)_{j=1,\ldots TopK}$, avoiding undesirable "jumps" in the the noise variance around $(x_j = z_s)_{j=1,\ldots TopK}$.

Figures 1a and 1b below show for the case $TopK = 1$ (hence: $E = 1$) and assuming variance of 1 for the random component v_1 an example of a noise coefficient flex-function $m(z)$ and the resulting noise standard deviation $z\,m(z)$ for selected

[3] A reasonable choice for parameter σ_2 could be the square root of the noise variance σ^2 selected by the disseminator when designing random noise for CKM to protect frequency tables. Note, we assume $z_s := 0$, if the concept should not foresee a separation.

arguments z. In the example, settings are $\sigma_0 = 0.05, \sigma_1 = 0.25, z_f = 23, q = 3$, and $\sigma_2{}^2 = 2$ (hence: separation point $z_s = \sqrt{2}/0.25 \cong 5.7$) .



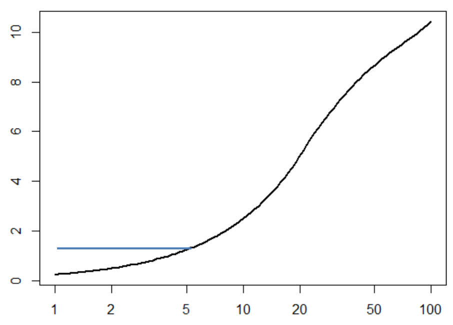


Fig. 1a: Example of a flex-function $m(z)$ for the noise coefficient

Fig. 1b: Example for noise std.dev. $z\,m(z)$. Blue line: constant std. dev. σ_2 (for $z < z_s$)

To summarize, when using a flex-function we compute the perturbed weighted sum of a continuous variable with original weighted sum x as

(2.5) $\hat{X} = x + \sum_{j=1,\dots,topK} \hat{X}_j$, with $\hat{X}_j$ given by (2.1), replacing in (2.1) $x_j m_1$ by $x_j\, m(x_1)$ for $x_j \geq z_s$, and $\hat{X}_1 := u$ for $x_1 < z_s$, and $\hat{X}_j := 0$ for $x_j < z_s, j > 1$. Notably, for $x_j \geq z_s$we can write (2.6) $\hat{X}_j := x_\delta \cdot V$, with $x_\delta = x_j \varepsilon_j\, m(x_1)$.

We also observe that choice of a large flex-point z_f (beyond the largest observation x_1 in the dataset) and identical $\sigma_1 = \sigma_0$ leads to constant $m(x_1) = m_1 = \sigma_1 = \sigma_0$. This allows us to generally assume use of a flex-function in the following section.

3. Adapting concepts to special cases and requirements

In section 3.1 we explain how to adapt the concepts of sec. 2 when non-negativity of the perturbed data is a requirement to be guaranteed. When on the other hand variables may take positive, as well as negative values, users of the packages have the option to allow, or not to allow changes of sign due to the perturbation. See sec. A.1.1 in the appendix for a short discussion of the options. Another special facility of the implementations is an option to increase the random noise by adding a fixed amount μ to the random factor when protecting sensitive cells, in particular in the case of data where cell sensitivity is assessed by concentration rules as suggested in Giessing (2012). Using such an extra protection option, the disseminator can enforce that the perturbed results are a fixed, “safe” distance away from the true cell value for sensitive cells[4]. Details of the implementation are explained in section A.1.2 in the appendix.

3.1 Dealing with non-negativity restrictions

In the case of positive variables, that is, if positivity should be ensured for the perturbed sum $\hat{X} = x + \sum_{j=1,\dots,topK} \hat{X}_j$ [5] from (2.4), we suggest to organize computation of the $TopK$ perturbation values $v_j, j = 1, \dots, TopK$ sequentially, starting with perturbation value v_1. The idea is to ensure first positivity of $\hat{X}^{(1)} := x + \hat{X}_1$, and then of

(3.1) $\hat{X}^{(j)} := \hat{X}^{(j-1)} + \hat{X}_j$ for $j = 1, \dots, TopK$.

Because of (2.1) and (2.2) , in the first step for $j = 1$, for not too small x_1 (i.e. above separation point z_s) we have:

(3.2) $\hat{X}^{(1)} = x +\cdot\, x_1\, m(x_1) \cdot v_1 \cdot =(\sum_{i=1}^{n} x_i\,) +\cdot\, x_1\, m(x_1) \cdot v_1$ which can be written as $\hat{X}^{(1)} = x + x_\delta \cdot v_1$ with $x_\delta := x_1\, m(x_1)$. With a sensible choice of the parameter function $m(z)$ (in particular: parameter σ_1), usually we will have $x_\delta \leq x$. Otherwise we set $x_\delta := x$.

Sec. 4.2 introduces a method to construct a suitable random variable V we assume to have variance 1 and how to “draw” from it a realization v_1, such that $x + x_\delta \cdot v_1$ is non-negative, if $x_\delta \leq x$. This method can thus be applied in the cases $x_1 \geq z_s$.

In the case $x_1 < z_s$ we obtain $\hat{X}^{(1)} = x + u$ (c.f. (2.4). How to deal with this “separation case” is explained in sec. 4.2.

In step $j, j > 1$ for x_j above separation point z_s because of (3.1) and (2.1) we have: $\hat{X}^{(j)} = \hat{X}^{(j-1)} + x_j\, \varepsilon_j\, m(x_j) v_j$. We could express this as $\hat{X}^{(j)} = \hat{X}^{(j-1)} + x_\delta \cdot v_j$ with $x_\delta := x_j\, \varepsilon_j m(x_j)$. But $\hat{X}^{(j-1)}$ (i.e. x after perturbation step $j-1$) could be very small. On the other hand, we generally assume step $j-1$ more important as step j (assuming higher disclosure risk for contribution x_{j-1}). So, if the perturbed value $\hat{X}^{(j-1)}$ is already very small, we consider it acceptable to require only minimal further perturbation. Therefore we set $x_\delta := \min(x_j\, \varepsilon_j\, m(x_j); \hat{X}^{(j-1)})$. This makes $x_\delta \leq \hat{X}^{(j-1)}$ and allows to apply the techniques of sec. 4.1.

In the case $x_j < z_s$, according to (2.4) we let $\hat{X}^{(j)} = 0$.

[4] For the p%-rule, (Giessing, 2012) argues a choice of $\mu := 2p$ for the extra noise amount.

[5] Notably, a similar issue may arise even with mixed positive/negative variables, i.e. when changes of sign due to perturbation should be avoided, see sec. A.1.1 in the appendix.

4. A generalization of the CKM method for continuous variables

In this section we present an idea for generalizing the maximum entropy based noise design method implemented in the package R package *p*table to the case of continuous variables. To this end a slight extension of the package is needed, but no fundamental changes have to be implemented.
In section 4.1 we briefly recall the structure of the output of the current version of the *p*table package and how an actual perturbation determined in the so called lookup step of the cell key method. Section 4.2 explains the changes that have to be made in order to adapt the procedure, originally designed for count tables, to the case of magnitude tables. In section 4.3 we present additional functionality to improve protection against disclosure risks in certain cases with small observation values. For better readability throughout the entire section we only consider positive magnitudes.

4.1 Recalling the implementation of the perturbation table lookup

As explained in section 2, the cell key method relies on some kind of random noise that is added to the actual data. In the simple case of count tables this can be expressed as $\hat{X} = x + v$, where x is the original cell value, v is some random noise and $\hat{X}$ is the perturbed value that will be published. Now v depends both on the original value x and the corresponding cell key z which is a random number between zero and one, c.f. (Giessing et al. 2018, appendix 1). Thus actually we can write $v(z, x)$. Now for the same combination of z and x the noise v will always be the same, ensuring consistency among logically identical tables. In order to determine $v(z, x)$ a perturbation table, supplied by the *p*table package, is needed. Such a perturbation table, also referred to as "lookup table", consists (amongst others) of columns named "*kum_p_u*", "*kum_p_o*", "i" and "*diff*". Here the last two consist of integer values only. Column "i" represents a reference value for the original value x, while "*kum_p_u*" and "*kum_p_o*" are a reference values for the cell key z. Now we define $v(z, x)$ as that value to be found by the so called "lookup step" in the column "diff", for the row where $i = x$ and $kum\_p\_u \leq z < kum\_p\_o$.
Note that the perturbation table is defined such that for every fixed value of "i" the corresponding rows of the perturbation table represent an inverse cumulative distribution function, so we actually apply the inverse transformation method. Using different distributions for smaller values of "i" is necessary to prevent changes of sign.

4.2 Main idea and structure of the perturbation table

In the following we describe the changes that have to be executed in order to extend the general implementation, according to the method described in section 2. In short these are:

- In favor of a finer gradation, perturbation values v computed by *p*table will not be limited to integers but to rational numbers instead. The distance between the possible outcomes is given by $\frac{1}{l}$, where l is a user defined number. This requires an extension of the format of the perturbation tables supplied by the $p$table package.

- It is not possible for any perturbation table to include the value of every possible original continuous cell value within a given range. Hence for any original value that is not part of the perturbation table we rely on an interpolation technique, explained later in this section.
- In order to use different noise distributions for even and odd numbers of counts, as mentioned in the introduction, we propose to actually store two different perturbation tables beneath each other in on file, adding an additional column to indicate which row belongs to which of those tables.

To motivate the approach explained in the following recall the case of frequency tables. Let n $(n \geq 1)$ some original cell value of a frequency table. Then the perturbed value $\widehat{N}$ is generated by $\widehat{N} = n + 1 \cdot V = n + n_\delta \cdot V$, for some integer valued random variable V. Note, in this case we have $n_\delta = 1 \leq n$.

Now, in case of continuous variables, disregarding the issue of extra protection for sensitive cells (sec. A.1.2 in the appendix) and assuming the case j= 1 [6] and not too small x_1 (i.e. above separation point z_s, see sec. 4.3 for the procedure for $x_1 < z_s$), we obtain a perturbed value for original value $x = \sum_{i=1}^{n} w_i y_i$ as

$$\hat{x} = x + x_\delta \cdot V,$$

with $x_\delta = x_1\, m(x_1)$, and assuming $x_\delta \leq x$, when we have to deal with non-negativity restrictions (c.f. 3.1)[7].

Let now the distribution of V be symmetrical and let D denote the maximum perturbation value of V. In case of continuous table values, of course V does not have to be integer valued. Hence we are free to choose

$$V \in \{ -D, \frac{1}{l} - D, \frac{2}{l} - D, \dots, D - \frac{1}{l}, D\} \quad .,$$

where l is a user defined number, like 4, 8, 10, 100, 1000.

One has to consider that for small original values a perturbation amounting to $x_\delta \cdot V$ might lead to a change of sign, if for example $V = -D$ and that usually this is not wanted by the user. Thus in case of small original values a skewed distribution of V is required, such that the maximum magnitude of V stays D, while the minimum magnitude is $\frac{-x}{x_\delta}$ or larger. This is equivalent to the case of frequency tables, where for

[6] For the case $j > 1$, see sec. 3.1 for how to determine x and x_δ

[7] C.f. sec. A.1.1 in the appendix, case (1), for how to proceed when there are no non-negativity restrictions.

$n_\delta = 1$ the *p*table package for each $n = \frac{n}{n_\delta} \in \mathbb{N}$ with $n \leq D$ automatically creates a skewed distribution such that $-n \leq n_\delta \cdot V$ and thus $V \geq -\frac{n}{n_\delta}$. Hence, when looking up the perturbation value, the original value $\frac{n}{n_\delta}$ must be taken into account.
Now clearly it is not possible to create and store a skewed distribution V for each $\frac{x}{x_\delta}$ with $x \leq D$ (and $x_\delta \leq x$, as explained in sec. 3.1) such that $V \geq -\frac{x}{x_\delta}$, so we will not even bother to achieve this. Fortunately, since we are not restricted to integer values, linear combinations of perturbation values again are appropriate perturbation values. Therefore in practice it suffices to generate one distribution for $\frac{x}{x_\delta} \geq D$ and one for the smallest value $\frac{x}{x_\delta}$ can reach, which is 1 (again assuming $x_\delta \leq x$). Now for each value of $\frac{x}{x_\delta}$ between 1 and D the perturbation value can be obtained by combining those two distributions into a new mixed noise distribution. Note that such a combination will in general not be integer valued and thus this approach is not suitable for count tables.

Illustrative example

For an illustrative example, let $D = 3$ be the maximum perturbation value and choose 0.5 as the step width.
Table 1 then shows the resulting perturbation table for a given preservation probability of 0.5 and a variance of 1. Note that preservation probabilities are a feature of the *p*table package which allows the user to define the probability for a cell value not to change at all. Let now $a \coloneqq \frac{x}{x_\delta}$ be a value between $a_0 \coloneqq 1$ and $a_1 \coloneqq D$, i.e. $a_0 < a < a_1$. Then for $\lambda = \frac{a-a_0}{a_1-a_0} \in (0,1)$ a can be written as $a = (1-\lambda) \cdot a_0 + \lambda \cdot a_1$. Now we define the perturbation value $\mathrm{V}(\mathrm{z}, a)$ of magnitude a and a given cell key z as the convex combination of the perturbation values $\mathrm{V}(\mathrm{z}, a_0)$ and $\mathrm{V}(\mathrm{z}, a_1)$, using that same parameter λ as described above, i.e.

$$\mathrm{V}(\mathrm{z}, a) = (1-\lambda) \cdot \mathrm{V}(\mathrm{z}, a_0) + \lambda \cdot \mathrm{V}(\mathrm{z}, a_1)$$

As an example we choose $a = \frac{x}{x_\delta} = 2.5$ and let $z = 0.18$ be the corresponding cell key. Since the largest value smaller than a that is to be found in the perturbation table is $a_0 = 1$ and the smallest value larger than a is $a_1 = 3$ we have $\lambda = \frac{2.5-1}{3-1} = 0.75$. Furthermore from **Table 1** we get $\mathrm{V}(\mathrm{z}, a_0) = -1$ and $\mathrm{V}(\mathrm{z}, a_1) = 0.5$. Now by interpolation we obtain $\mathrm{V}(\mathrm{z}, a) = 0.25 \cdot (-1) + 0.75 \cdot (-0.5) = -0.625$.

4.3 Constant noise variance for very small observations

If below some separation point $z_s > 0$ a constant noise variance shall be implemented that does not depend on x_1, and if there are non-negativity restrictions, we need a second perturbation table. From this second perturbation table which (according to sec. 2.1) should define noise distributions with noise variance ${\sigma_2}^2$ we obtain $\mathrm{U}(z, x)$, i.e. relating to the original cell value x, instead of $a = \frac{x}{x_\delta}$.

Otherwise, if in that case there are *no* non-negativity restrictions, i.e. to implement case (1) of sec. A.1.1 in the appendix, we can look up a perturbation $\tilde{u}$ in any perturbation table for continuous values (for example in a perturbation table relating to a "standard" noise variance of 1) with desired step width and maximal perturbation, in the block of rows of that perturbation table relating to the symmetric case (i.e. in the last block of lines), irrespective of the value of x. Finally we multiply with the desired standard deviation (from e.g., (2.4)), f.i. we set $u := \sigma_2 \cdot \tilde{u}$, if $\tilde{u}$ relates to a perturbation table defining noise with variance 1.

5. Summary and Final Remarks

Funded by the EU project "Open Source tools for perturbative confidentiality methods" a cell key method useable not only for protecting frequency count tables by additive noise, but also for the protection of continuous variables has been implemented in the package τ-Argus and as separate R package cellKey. Like the perturbation method for continuous data of Thompson et al. (2013) implemented in the ABS table builder, we basically build on a multiplicative approach. A flex-function has been integrated to allow some flexibility of the noise variance coefficients which will have to be chosen by a disseminator to determine the strength of the perturbations.

The main contribution of the work reported in this paper is a concept for convex combination of noise drawn from discrete noise distributions to generate noise for continuous data. The concept builds on a slight extension of the maximum entropy approach implemented so far to compute the transition matrices used as (discrete) distributions for the noise applied to frequency data.

The paper also discusses alternative ways for how to handle specific issues, like non-negativity restrictions for the perturbed data. Not a subject of the present paper, however, is advice and guidance how to choose parameters in a suitable way, i.e. balancing information loss and disclosure risk. This kind of work is envisioned as one of the tasks to be addressed by future projects.

Acknowledgements. The research leading to these results has partially received funding from the EU project "Open Source tools for perturbative confidentiality methods" (Specific grant agreement contract n° 2018.0108 under the Framework partnership agreement n° 11112.2014.005-2014.533)

References

Enderle, T and Giessing, S. (2018) *Implementation of a ‘p-table generator’ as separate R-package*, Deliverable D3.2 of Work Package 3 “Prototypical implementation of the cell key/seed method” within the Specific Grant Agreement “Open Source tools for perturbative confidentiality methods.

Fraser, B. and Wooton, J. (2006). *A proposed method for confidentialising tabular output to protect against differencing*. In: Monographs of Official Statistics. Work session on Statistical Data Confidentiality, Eurostat-Office for Official Publications of the European Communities, Luxembourg, 2006, pp. 299-302.

Giessing, S. (2012). *Flexible Rounding Based on Consistent Post-tabular Stochastic Noise*. In J. Domingo-Ferrer and I. Tinnirello (Eds.), Privacy in Statistical Databases, 22-34. New York: Springer- Verlag. LNCS 7556.

Giessing, S. (2016), ‘*Computational Issues in the Design of Transition Probabilities and Disclosure Risk Estimation for Additive Noise’.* In: Domingo-Ferrer, J. and Pejić-Bach, M. (Eds.), Privacy in Statistical Databases, pp. 237-251, Springer International Publishing, LNCS, vol. 9867.

Giessing, S., van de Laar, R., Enderle, Tent, R. (2018) *Methodological report on options of generalising the cell key method and eventual restrictions*, Deliverable D4.1 of Work Package 4 “More general implementation of the cell key/seed method” within the Specific Grant Agreement “Open Source tools for perturbative confidentiality methods*”*.

Giessing, S., Tent, R., Enderle, (2019) *Methodological concept for generalising the cell key method to the case of continuous variables*, Deliverable D4.2-part I of Work Package 4 “More general implementation of the cell key/seed method” within the Specific Grant Agreement “Open Source tools for perturbative confidentiality methods*”*.

Ma Y., Lin YX., Chipperfield J., Newman J., Leaver V. (2016) *A New Algorithm for Protecting Aggregate Business Microdata via a Remote System*. In: Domingo-Ferrer J., Pejić-Bach M. (eds) Privacy in Statistical Databases. PSD 2016. Lecture Notes in Computer Science, vol 9867. Springer, Cham

Marley, J. K., and V. L. Leaver. 2011. ‘*A Method for Confidentialising User-Defined Tables: Statistical Proper-Ties and a Risk-Utility Analysis*.’ Proceedings of 58th World Statistical Con-gress, 1072–81.

Meindl, B., Kowaric, A., De Wolf, P.P. (2018) *Prototype implementation of the cell key method, including test results and a description on the use for census data*, Deliverable D3.1 of Work Package 3 “Prototypical implementation of the cell key/seed method” within the Specific Grant Agreement “Open Source tools for perturbative confidentiality methods.

Thompson, G., Broadfoot, S., Elazar, D. (2013): “*Methodology for the Automatic Confidentialisation of Statistical Outputs from Remote Servers at the Australian Bureau of Statistics*”, paper presented at the Joint UNECE/Eurostat Work Session on Statistical Data Confidentiality (Ottawa, 28-30 Oktober 2013) available at

http://www.unece.org/fileadmin/DAM/stats/documents/ece/ces/ge.46/2013/Topic_1_ABS.pdf

Appendix A.1

A.1.1 Variables with positive and negative contributions

In this document, we basically consider three different ways of dealing with variables that may take positive, as well as negative values. Such variables might typically result from taking the difference of two initial positive variables, like the balance of people moving in and out of areas, or the difference between income observations for two subsequent years.

Let x an observation of a variable X taking positive as well as negative values. Let $sign(x) := 1$, if $x \geq 0$ and -1 otherwise.

We suggest implementing the following three variants:

(1) In the lookup step, look up the perturbation v_j in the block of rows of the perturbation table relating to the symmetric case (i.e. in the last block of lines), irrespective of the value of x or x_δ (defined by formulas (2.5) and (2.6)). In particular: also in the case $x = 0$. Compute the perturbed value according to (2.1) or (2.5), *resp.*.

(2) Do the lookup step like in case of variables taking only positive values, using the absolutes $|x|$ and $|x_\delta|$ when applying the techniques of sec. 4.1 to obtain perturbation v and finally multiply the perturbed value $\hat{X}$ by $sign(x)$.

(3) Like (2); for $|x| \geq |x_\delta|$. Like (1); otherwise, or, if changes of sign are considered very undesirable, using the technique of sec. 4.3.

A.1.2 Extra protection for sensitive cells

To enforce that perturbed results are a fixed, “safe” distance away from the true cell value for sensitive cells a user of the packages may request increased random noise when a sensitive cells should be protected, in particular in the case of data where cell sensitivity is assessed by concentration rules. In this case a fixed amount μ will be added to the random factor[8].

When integrating the extra protection into a noise factor concept based on top-*k*th observations with $topK > 1$ it makes sense to implement extra protection only into

[8] For the p%-rule, Giessing (2012) argues a choice of $\mu := 2p$ for the extra noise amount.

the first noise component $\hat{X}_1$ relating to the largest contribution to the cell to avoid differently directed extra protection components cancelling each other out, and not implementing it at all for very small observations below the separation point z_s.

In the denotation of section 2, this means for $x_1 > z_s$, we replace the noise component relating to the largest observation,$\cdot x_1 \cdot m(x) \cdot v_1$, by $\cdot x_1\, m(x_1) \cdot d_1 \cdot (\mu + |v_1|)$ where $d_1 := sign(v_1)$.

If there is a non-negativity restriction (c.f. sec. 3.1), in case of negative direction of the perturbation (i.e. $d_1 = -1$) we should avoid that the extra protection amount μ leads to negativity. With denotation of sec. 3.1, integration of the extra perturbation means $\hat{X}^{(1)} = x + \cdot x_1\, m(x_1) \cdot d_1 \cdot (\mu + |v_1|)$ $\cdot$(c.f. (3.2))and can be written as $\hat{X}^{(1)} = x + x_\delta \cdot v_1 + d_1\, x_\delta \mu$ with $x_\delta := x_1\, m(x_1)$, the term $d_1\, x_\delta \mu$ relating to the extra protection. To ensure non-negativity we first compute v_1 (with the techniques of sec. 4.1) such that $x + x_\delta \cdot v_1$ is non-negative, and then we set $\hat{X}^{(1)} := \max(x + x_\delta \cdot v_1 + d_1\, x_\delta \mu; 0)$.

A.1.3 Different noise distribution for even and odd number of contributions

Ma et al (2016) discuss a specific risk scenario: The scenario is based on a case where perturbed figures for some continuous variable are released for table cell A relating to a certain group of respondents, as well as for another group (table cell B) that differs from the first group by only one respondent C. In that case, a user of the perturbed data could estimate the value of the variable for respondent C by differencing between the perturbed figures for cell A and cell B. If the probability is high for both, for the perturbed value of cell A and for the perturbed value of cell B to change by only a small amount, similar for cell A and cell B, the estimate for respondent C will be close to the true value with still a relatively high probability. In order to reduce this risk, Ma et al (2016) suggest using different noise distributions for cells with even and odd number of contributions (note, if cells A and B differ by only one respondent, the number of respondents will be even for either A or B, and odd for the other one). They suggest using a distribution U_1 with low probabilities for small changes for, say, odd cells (i.e. cells with an odd number of contributions), and then U_2 with high probability of small changes for the even cells. Like for example U_1 a uniform distribution on the intervals $I_1 := [-1.5; -0.5] \cup [0.5; 1.5]$, and U_2 a uniform distribution on the intervals $I_2 := [-2; -1.5] \cup [-0.5; 0.5] \cup [1.5; 2]$.

Future versions of the *p*table package might support this idea, supplying perturbation tables with such pairs of “complementary” distributions, if requested this way by the user of the package. See section 3.2 of Giessing et al. (2019) for a suggested data structure of such a “mixed” perturbation table. Section 3.3 of Giessing et al. (2019) explains how to implement the lookup step accordingly in the CKM tools.

Appendix A.2

i	j	p	kum_p_u	kum_p_o	diff
0	0	1	0	1	0
1	0	0.18078	0	0.18078	-1
1	0.5	0.12789	0.18078	0.30867	-0.5
1	1	0.50000	0.30867	0.80867	0
1	1.5	0.06400	0.80867	0.87267	0.5
1	2	0.04528	0.87267	0.91795	1
1	2.5	0.03203	0.91795	0.94997	1.5
1	3	0.02266	0.94997	0.97263	2
1	3.5	0.01603	0.97263	0.98866	2.5
1	4	0.01134	0.98866	1	3
3	0	0.00850	0	0.00850	-3
3	0.5	0.01719	0.00850	0.02570	-2.5
3	1	0.03059	0.02570	0.05629	-2
3	1.5	0.04788	0.05629	0.10417	-1.5
3	2	0.06594	0.10417	0.17010	-1
3	2.5	0.07990	0.17010	0.25000	-0.5
3	3	0.50000	0.25000	0.75000	0
3	3.5	0.07990	0.75000	0.82990	0.5
3	4	0.06594	0.82990	0.89583	1
3	4.5	0.04788	0.89583	0.94371	1.5
3	5	0.03059	0.94371	0.97430	2
3	5.5	0.01719	0.97430	0.99150	2.5
3	6	0.00850	0.99150	1	3

Table 1: Example for a perturbation table in case of continuous values with step width 0.5 and maximal perturbation *D*=3.