BUREAU OF THE CENSUS
STATISTICAL RESEARCH DIVISION REPORT SERIES
SRD Research Report Number: Census/SRD/RR-90/04

Constrained Noise for Masking Microdata Records

by

Laura Voshell
Statistical Research Division
Bureau of the Census
Washington, D.C. 20233

This series contains research reports, written by or in cooperation with staff members of the Statistical Research Division, whose content may be of interest to the general statistical research community. The views reflected in these reports are not necessarily those of the Census Bureau nor do they necessarily represent Census Bureau statistical policy or practice. Inquiries may be addressed to the author(s) or the SRD Report Series Coordinator, Statistical Research Division, Bureau of the Census, Washington, D.C. 20233.

Recommended by: Brian Greenberg

Report completed: January 20, 1990

Report issued: January 20, 1990

Constrained Noise for Masking Microdata Records

ABSTRACT

The objective of this report is to present two algorithms which transform data generated by a random number generator into data satisfying certain constraints on means and variance-covariance structure. One algorithm uses a linear transformation and translation to force data generated from a multivariate normal distribution to have a specific mean vector and variance-covariance matrix. The other algorithm uses a series of additions and subtractions to ensure that data generated from a uniform distribution has a certain mean and variance. Data sets such as these may be beneficial when used for introducing noise in order to mask microdata as a disclosure avoidance technique.

I. INTRODUCTION

The objective of this report is to present two algorithms which transform data generated by a random number generator into data satisfying certain constraints on means and variance-covariance structure. One algorithm uses a linear transformation and translation to force data generated from a multivariate normal distribution to have a specific mean vector and variance-covariance matrix. The other algorithm uses a series of additions and subtractions to ensure that data generated from a uniform distribution has a certain mean and variance. Data sets such as these may be beneficial when used for introducing noise in order to mask microdata as a disclosure avoidance technique.

Random number generators are designed to generate a finite set of random numbers from a given infinite population with a specified distribution. In the process of masking microdata files, random number generators are used to obtain data sets of random numbers from a multivariate normal distribution with some specified mean vector $\mathbf{m}$ and some specified variance-covariance matrix $\mathbf{V}$. These random numbers are added to the microdata or to some transformation of the microdata to produce noise in the data and reduce risk of disclosure through reidentification. One must realize that the mean vectors and variance-covariance matrices of data sets created by random number generators are not exactly the specified $\mathbf{m}$ and $\mathbf{V}$. In fact, the values in this vector and this matrix can be off to a degree such that the mean vector and variance-covariance matrix of the masked data are considerably different from those which the masking technique was designed to yield.

Masking data through the addition of noise for disclosure avoidance has been discussed in numerous papers including (Spruill 1982), (Cox, et al. 1985), (Kim 1986), and (McGuckin and Nguyen 1988). The basic idea is that by adding noise to individual records, an intruder will not be able to link a respondent to a microdata record using an exact match based on information in the intruder's data base. In (Kim 1986), the author proposed a method of masking microdata which involves adding random noise generated from a multivariate normal distribution with mean vector **m**=**0** and variance-covariance matrix **V**=c**W** to the data where c is some constant and **W** is the variance-covariance matrix of the original data. The resulting data set is then transformed in such a manner that the mean vector and variance-covariance matrix of the noise-added data are preserved. Unfortunately, the mean vector of the noise-added data is typically not the mean vector of the original data as desired, and the variance-covariance matrix of the noise-added data is not (1+c)**W** as also desired. This is due to the fact that the mean vector of the noise is not exactly **0**, and the variance-covariance matrix of the noise is not exactly c**W**.

In this paper, we describe an algorithm which transforms data created by a random number generator designed to generate random numbers from a multivariate normal distribution with mean vector **m** and variance-covariance matrix **V** into a finite data set with mean vector exactly **m** and variance-covariance matrix exactly **V**. That is, the finite sample randomly drawn from the infinite population is constrained to have the specified mean vector and variance-covariance matrix. Accordingly, when masking a data file using these values, we will be adding constrained noise. The algorithm is described in Section II. Also described in Section II is an algorithm which slightly adjusts data created by a random number generator designed to generate random numbers from a uniform distribution on a given interval in order to obtain a data set with the exact mean and arbitrarily close variance of this distribution. Benefits in using such data sets for the purpose of adding constrained noise to microdata are discussed in Section III.

II. THE ALGORITHMS

A. MULTIVARIATE NORMAL DISTRIBUTION

1. ALGORITHM

Let us assume we use a random number generator to obtain a data set of N observations in P variables from a multivariate normal distribution with mean vector **m** and variance-covariance matrix **V**. Call the matrix of this data set $\mathbf{E}_0$ (NxP). The transformation process of the data set involves three steps. We begin by calculating the mean vector of the data set represented by matrix $\mathbf{E}_0$. This mean vector is subtracted from each row of $\mathbf{E}_0$ yielding a data set with mean vector **0**. Call this revised data set $\mathbf{E}_1$.

Let

$$\mathbf{V}_1 = \mathbf{E}_1' * \mathbf{E}_1 \text{ (PxP)}$$

be the variance-covariance matrix of this new data set -- where $\mathbf{E}_1'$ is the transpose of the matrix $\mathbf{E}_1$. We make the assumption that the columns of $\mathbf{E}_1$ are linearly independent so that $\mathbf{V}_1$ is nonsingular and positive definite. We now make use of Cholesky Decomposition. This is a special form of triangular factorization of matrices which enables us to write any positive definite, symmetric matrix **A** (NxN) as $\mathbf{A} = \mathbf{B}' * \mathbf{B}$ where **B** is nonsingular (NxN). See (Kennedy and Gentle 1980) for a further description of this process. We use Cholesky Decomposition to obtain a nonsingular matrix $\mathbf{C}_1$ (PxP) such that

$$\mathbf{C}_1' * \mathbf{C}_1 = \mathbf{V}_1.$$

Cholesky Decomposition is also used to obtain a matrix **C** (PxP) such that

$$\mathbf{C}' * \mathbf{C} = \mathbf{V}$$

where **V** is the desired variance-covariance matrix which we assume to be positive definite, hence nonsingular. Thus **C** is nonsingular. We wish to obtain a nonsingular matrix **T** (PxP) which could be multiplied with $\mathbf{E}_1$ to yield an (NxP) matrix

$$\mathbf{E}_2 = \mathbf{E}_1 * \mathbf{T},$$

with the desired variance-covariance structure. That is, we seek **T** such that

$$\mathbf{T}' * \mathbf{E}_1' * \mathbf{E}_1 * \mathbf{T} \ (= \mathbf{E}_2' * \mathbf{E}_2 \ ) = \mathbf{V}.$$

Note that

$$\mathbf{T}' * \mathbf{E}_1' * \mathbf{E}_1 * \mathbf{T} \quad = \quad \mathbf{T}' * \mathbf{V}_1 * \mathbf{T} \quad = \quad \mathbf{T}' * \mathbf{C}_1' * \mathbf{C}_1 * \mathbf{T}$$

so our requirement is satisfied when

$$\mathbf{T} = \mathbf{C}_1^{-1} * \mathbf{C}$$

as can be seen through direct substitution above. Thus we multiply matrices to obtain the matrix

$$\mathbf{E}_2 = \mathbf{E}_1 * \mathbf{T}.$$

Our new data set represented by the matrix $\mathbf{E}_2$ has variance-covariance matrix **V**.

In order to obtain our desired mean vector **m**, we then calculate the mean vector of $\mathbf{E}_2$, subtract it from each row of $\mathbf{E}_2$, and add the vector **m** to each row of $\mathbf{E}_2$. The resulting data set, call it $\mathbf{E}_3$, has mean vector **m** and variance-covariance matrix **V**.

2. EXAMPLE

A short and simple Fortran program has been written to implement this algorithm. It has successfully run on several data sets, one of which is given in Table 1 as an example. The data set resulting from the algorithm has exactly the properties we would like it to have. Code is available from the author.

Table 1

Example of a Noise Data Set Before and After Transformation

N = 100
P = 4

Desired Mean Vector = $(\ 0.00 \quad 0.00 \quad 0.00 \quad 0.00)$

Desired Variance-Covariance Matrix =

$$\begin{matrix} 5.0 & -1.0 & 3.0 & 0.0 \\ -1.0 & 6.0 & -2.0 & -5.0 \\ 3.0 & -2.0 & 4.0 & 1.0 \\ 0.0 & -5.0 & 1.0 & 5.0 \end{matrix}$$

Mean Vector of Data Set Obtained by Random Number Generator = $(-0.15 \quad -0.18 \quad 0.06 \quad 0.22)$

Variance-Covariance Matrix of Data Set Obtained by Random Number Generator =

$$\begin{matrix} 5.53 & -1.79 & 3.37 & 0.43 \\ -1.79 & 6.63 & -2.00 & -5.24 \\ 3.37 & -2.00 & 3.97 & 0.80 \\ 0.43 & -5.24 & 0.80 & 5.06 \end{matrix}$$

Mean Vector of Data Set
Resulting from this Algorithm =

$$(-0.75 \times 10^{-8} \quad -0.36 \times 10^{-6} \quad 0.22 \times 10^{-6} \quad -0.28 \times 10^{-6})$$

Variance-Covariance Matrix of Data
Set Resulting from this Algorithm =

$$\begin{matrix} 4.99999332 & -1.00000381 & 2.99999523 & -0.13 \times 10^{-5} \\ -1.00000381 & 5.99998283 & -1.99999523 & -4.99999619 \\ 2.99999523 & -1.99999523 & 3.99998951 & 1.00000000 \\ -0.13 \times 10^{-5} & -4.99999619 & 1.00000000 & 4.99999428 \end{matrix}$$

B. UNIFORM DISTRIBUTION

1. BACKGROUND

There are also random number generators designed to generate a finite set of random numbers from the uniform distribution on the interval (a,b). A typical set of numbers generated in this fashion will not have mean exactly (a+b)/2 nor variance exactly $(b-a)^2/12$. If desired, a set of random numbers from the uniform distribution on the interval (a,b) generated by a random number generator can be adjusted and constrained to have a mean of exactly (a+b)/2 and a variance arbitrarily close to $(b-a)^2/12$. For all practical purposes, we can treat the new data set as if it were randomly drawn from a uniform distribution.

An algorithm has been devised which takes a set of numbers randomly generated from the uniform distribution on the interval (-1,1) and slightly adjusts several of the numbers in the set so that the mean and variance of the data set are exactly 0 and arbitrarily close to 1/3 (the mean and variance of a uniform distribution on the interval (-1,1)). Note here that the user can start with a data set generated from the interval (-1,1) or with a data set generated from any other interval (c,d). In the latter case, a simple transformation may be used to change the data from the interval (c,d) to data on the interval (-1,1). For every number x_0 in the data set on the interval (c,d), let the corresponding number x_1 in the data set on the interval (-1,1) be

$$x_1 = (2/(d-c)) * x_0 + ((c+d)/(c-d))$$

After the data on the interval (-1,1) has been altered to obtain a mean of exactly 0 and a variance arbitrarily close to 1/3, a simple transformation can be used to change the data on the interval (-1,1) to data on the desired interval (a,b). For every number x_1 on the interval (-1,1), let the corresponding number x_2 on the interval (a,b) be

$$x_2 = ((b-a)/2) * x_1 + ((a+b)/2)$$

The mean of the data on the interval (a,b) will be exactly (a+b)/2 and the variance will be arbitrarily close to $(b-a)^2/12$.

2. ALGORITHM

Let us assume a random number generator is used to generate N random numbers from the uniform distribution on the interval (-1,1). The adjustment of the data set to obtain the desired mean and variance is done by adding or subtracting two small positive numbers, call them k_1 and k_2, to a subset of the values in the original data set.

The first part of the algorithm adjusts the mean of the data set. The sum, S, of the random numbers is calculated. If S=0, we have our desired mean. Otherwise choose a large integer L and let $k_1 = |S|/L$. Typically, we have been using L=10,000. If S is less than 0, then one of the numbers from the data set which is less than or equal to $1-k_1$ is randomly chosen and k_1 is added to it. This is done L times. The sum of the resulting numbers, and hence the mean, will thus be 0. On the other hand, if S is greater than 0, then one of the numbers from the data set which is greater than or equal to k_1-1 is randomly chosen and k_1 is subtracted from it. This is done L times to produce a data set with a mean of 0.

The second part of the algorithm adjusts the variance of the data set. The desired variance is 1/3. Note:

$$f(x) = 0.5 \qquad -1 <= x <= 1$$

$$E(x) = \int_{-1}^{1} 0.5 * x \, dx = 1/4 - 1/4 = 0$$

and

$$Var(x) = E(x^2) - (E(x))^2 = \int_{-1}^{1} 0.5 * x^2 \, dx - 0 = 1/6 + 1/6 = 1/3$$

Given that the mean is 0, the variance of the sample data set will equal 1/3 if the sum of the random numbers squared is equal to N/3. That is,

$$(1/N) * \sum_{i=1}^{N} (x_i - \bar{x})^2 = 1/3$$

so

$$\sum_{i=1}^{N} {x_i}^2 = N/3$$

Following the adjustment of the mean, the sum of the random numbers squared, SS, is calculated. If SS=N/3, we are done. Otherwise, consider the case that this sum is less than N/3. This means that the numbers are too clustered around the mean of 0 and need to be spread out. Choose a large integer L. Typically, we have been using L=10,000. Let

$$k_2 = ((N/3)-SS)/(2*L).$$

k_2 will be a very small number. The addition of k_2 to one of the random numbers greater than 0 and less than $1-k_2$ in the data set would cause an increase in the sum of the random numbers squared. The subtraction of k_2 from one of the random numbers between k_2-1 and 0 would also cause an increase in the sum of the random numbers squared. One integrates to find the average increase in the square of a number between 0 and $1-k_2$ when k_2 is added to that number. Note that this average increase will equal the average increase in the square of a number between k_2-1 and 0 when k_2 is subtracted from that number. Call this average increase I_a.

$$I_a = \int_{0}^{1-k_2} (x + k_2)^2 / (1-k_2) \, dx - \int_{0}^{1-k_2} x^2 / (1-k_2) \, dx = k_2$$

It is interesting to note that $I_a = k_2$. An increase in the sum of the random numbers squared is brought about by the appropriate addition or subtraction of k_2 to several randomly chosen numbers in the data set. The process requires that k_2 is added to a random number the same number of times that k_2 is subtracted from a random number in order to maintain the mean of 0. Because the numbers in the data set are approximately uniformly distributed, L additions and L subtractions of k_2 to appropriate, randomly chosen numbers in the data set will yield an expected increase in SS of (N/3)-SS. To start the process, one of the numbers between 0 and $1-k_2$ is randomly chosen, and k_2 is added to it. This is done L times. Then one of the numbers between k_2-1 and 0 is chosen and k_2 is subtracted from it. This is also done L times. Thus the numbers are spread out, the variance is adjusted, and the mean remains 0.

A similar process is carried out if the sum of the random numbers squared is initially too large. If this is the case, the random numbers are too spread out and need to be pulled in a little toward 0. Choose a large integer L. Again, we typically have been using L=10,000. Let

$$k_2 = (SS-(N/3))/(2*L)$$

k_2 will be a very small number. The subtraction of k_2 from one of the numbers between k_2 and 1 would cause a decrease in the sum of the random numbers squared. The addition of k_2 to one of the numbers between -1 and $-k_2$ would also cause a decrease in the sum of the random numbers squared. Integration can be used to find the average decrease in the square of a number between k_2 and 1 when k_2 is subtracted from that number. Note that this average decrease will equal the average decrease in the square of a number between -1 and $-k_2$ when k_2 is added to that number. Call this average decrease D_a.

$$D_a = \int_{k_2}^{1} x^2 \ / \ (1-k_2) \ dx \ - \int_{k_2}^{1} (x-k_2)^2 \ / \ (1-k_2) \ dx = k_2$$

It is also interesting to note that $D_a = I_a = k_2$. A decrease in the sum of the random numbers squared is brought about by the appropriate addition or subtraction of k_2 to several randomly chosen numbers in the data set. The process again requires that k_2 is added to a number the same number of times that k_2 is subtracted from a number in order to maintain the mean of 0. L additions and L subtractions of k_2 to appropriate, randomly chosen numbers in the data set will yield an expected decrease in the SS of SS-(N/3). To start this process, one of the numbers between k_2 and 1 is randomly chosen and k_2 is subtracted from it. This is done L times. Then one of the numbers between -1 and $-k_2$ is randomly chosen, and k_2 is added to it. This is also done L times. Thus the numbers are pulled in, the variance is adjusted, and the mean remains 0.

Note that the part of the algorithm which adjusts the variance of the data set is based on the expected value of the change in SS brought about by the addition or subtraction of k_2 to appropriate, randomly chosen numbers in the data set. This does not guarantee a final variance of exactly 1/3. However, because we have chosen such a large L and therefore obtained such a small k_2, the variability of the change in SS is negligible, and the resulting variance will be closer to 1/3 than the original variance. If not satisfied with the variance of the data set after it has been run through the algorithm once, the user may run the data set through the algorithm a few more times until a satisfactory variance is obtained or the difference between the variance and 1/3 is less than computer tolerance.

3. EXAMPLES

A simple Fortran program has been written to carry out this algorithm. It has successfully run on several data sets, yielding means and variances of 0 and 1/3 without changing the basic level of uniformity of the data. See Tables 2 and 3 for two examples. Code is available from the author.

Table 2

Example of a Constrained Data Set from the Uniform Distribution

```
Original Data Set, Sample Size = 1000

Mean     = 0.00999873

Variance = 0.322009

                 HISTOGRAM                   #          BOXPLOT
 0.95+****************************          55             ]
     .*****************                     34             ]
     .****************************          55             ]
     .**********************                43             ]
     .***********************               46             ]
     .**********************************    68          +-----+
     .************************              47          ]     ]
     .********************************      63          ]     ]
     .***************************           54          ]     ]
     .**********************                44          *--+--*
     .*******************************       62          ]     ]
     .*********************                 42          ]     ]
     .****************************          56          ]     ]
     .********************                  40          ]     ]
     .***************************           53          +-----+
     .************************              47             ]
     .************************              48             ]
     .************************              48             ]
     .*************************             49             ]
-0.95+***********************               46             ]
      ----+----+----+----+----+----+----
      * MAY REPRESENT UP TO 2 COUNTS
```

Table 2, continued

Constrained Data Set, Passed through Algorithm Once, Sample Size = 1000

Mean = $3.436 * 10^{-7}$

Variance = 0.333163

```
                 HISTOGRAM                 #                  BOXPLOT
 0.95+****************************         56                     ]
     .******************                   35                     ]
     .***************************          53                     ]
     .*************************            49                     ]
     .********************                 40                     ]
     .**********************************   68                  +-----+
     .*************************            49                  ]     ]
     .*******************************      61                  ]     ]
     .***************************          54                  ]     ]
     .**********************               43                  *--+--*
     .**************************           51                  ]     ]
     .*********************                42                  ]     ]
     .*********************************    65                  ]     ]
     .*****************                    34                  ]     ]
     .*************************            50                  ]     ]
     .**********************               44                  +-----+
     .****************************         55                     ]
     .************************             47                     ]
     .************************             47                     ]
-0.95+*****************************        57                     ]
      ----+----+----+----+----+----+----
      * MAY REPRESENT UP TO 2 COUNTS
```

Table 2, continued

Constrained Data Set, Passed through Algorithm Twice, Sample Size = 1000

Mean = 5.208 * 10^{-8}

Variance = 0.333332

```
                   HISTOGRAM                  #                  BOXPLOT
   0.95+****************************         56                      ]
       .******************                   35                      ]
       .***************************          53                      ]
       .*************************            49                      ]
       .********************                 40                      ]
       .**********************************   68                  +-----+
       .*************************            49                  ]     ]
       .*******************************      61                  ]     ]
       .***************************          54                  ]     ]
       .**********************               43                  *--+--*
       .**************************           51                  ]     ]
       .*********************                42                  ]     ]
       .*********************************    65                  ]     ]
       .*****************                    34                  ]     ]
       .*************************            50                  ]     ]
       .**********************               44                  +-----+
       .****************************         55                      ]
       .************************             47                      ]
       .************************             47                      ]
  -0.95+*****************************        57                      ]
        ----+----+----+----+----+----+----
        * MAY REPRESENT UP TO 2 COUNTS
```

Table 3

Example of a Constrained Data Set from the Uniform Distribution

```
Original Data Set, Sample Size = 10000

Mean     = 0.00236354

Variance = 0.329832

                     HISTOGRAM                           #       BOXPLOT
 0.95+******************************************        502         ]
     .*******************************************       509         ]
     .***************************************           464         ]
     .******************************************        502         ]
     .*********************************************     529      +-----+
     .********************************************      519      ]     ]
     .****************************************          475      ]     ]
     .******************************************        500      ]     ]
     .******************************************        502      ]     ]
     .*****************************************         481      ]  +  ]
     .****************************************          476      *-----*
     .*********************************************     540      ]     ]
     .*********************************************     533      ]     ]
     .******************************************        494      ]     ]
     .*****************************************         488      +-----+
     .*******************************************       509         ]
     .**********************************************    544         ]
     .****************************************          478         ]
     .*****************************************         483         ]
-0.95+****************************************          472         ]
      ----+----+----+----+----+----+----+----+----+-
      * MAY REPRESENT UP TO 12 COUNTS
```

Table 3, continued

Constrained Data Set, Passed through Algorithm Once, Sample Size = 10000

Mean = 3.370×10^{-6}

Variance = 0.333439

```
                        HISTOGRAM                          #      BOXPLOT
  0.95+*******************************************        506        ]
      .*******************************************        509        ]
      .****************************************           470        ]
      .******************************************         502        ]
      .********************************************       526     +-----+
      .*******************************************        515     ]     ]
      .****************************************           479     ]     ]
      .******************************************         497     ]     ]
      .*******************************************        510     ]     ]
      .***************************************            460     ]  +  ]
      .***************************************            459     *-----*
      .*********************************************      531     ]     ]
      .**********************************************     545     ]     ]
      .*****************************************          491     ]     ]
      .*****************************************          487     ]     ]
      .*******************************************        506     +-----+
      .**********************************************     541        ]
      .*****************************************          484        ]
      .*****************************************          485        ]
 -0.95+******************************************         497        ]
       ----+----+----+----+----+----+----+----+----+-
       * MAY REPRESENT UP TO 12 COUNTS
```

Table 3, continued

Constrained Data Set, Passed through Algorithm Twice, Sample Size = 10000

Mean = 3.615 * 10^{-7}

Variance = 0.333333

```
                      HISTOGRAM                          #     BOXPLOT
   0.95+*******************************************     506       ]
       .*******************************************     509       ]
       .****************************************        470       ]
       .******************************************      502       ]
       .********************************************    526    +-----+
       .*******************************************     515    ]     ]
       .****************************************        479    ]     ]
       .******************************************      497    ]     ]
       .*******************************************     510    ]     ]
       .***************************************         460    ]  +  ]
       .***************************************         459    *-----*
       .*********************************************   531    ]     ]
       .**********************************************  545    ]     ]
       .*****************************************       491    ]     ]
       .*****************************************       487    ]     ]
       .*******************************************     506    +-----+
       .**********************************************  541       ]
       .*****************************************       484       ]
       .*****************************************       485       ]
  -0.95+******************************************      497       ]
        ----+----+----+----+----+----+----+----+----+-
        * MAY REPRESENT UP TO 12 COUNTS
```

III. RELATIONSHIP TO MASKING FOR DISCLOSURE AVOIDANCE

A. MULTIVARIATE NORMAL

1. CONSTRAINED NOISE

As stated before, microdata can be masked for disclosure avoidance purposes by adding noise to the data, thus inhibiting intruders from linking respondents to their individual records. Noise added to the data can be created by a random number generator used to generate random numbers from a multivariate normal distribution with a specified mean vector and variance-covariance matrix. Unfortunately, the noise data resulting from such a random number generator will not have the exact mean vector and variance-covariance matrix desired by those masking the data. Thus, due to variability when generating random numbers, a set of numbers created by a random number generator, when added to a data set, can yield a data set with a mean vector and a variance-covariance matrix considerably different from those for which the masking technique was designed. If this same set of random numbers was constrained to have a specified mean vector and a specified variance-covariance matrix, more control could be maintained over the noise-added data.

2. SAMPLE VARIANCES OF MASKED DATA

a. FULL DATA SET

There are also benefits in using a noise data set with constraints on the mean vector and variance-covariance matrix in terms of the sample variances of the means of the masked data. Consider one variable, call it y, of a masked data set consisting of n records. An observation y_i, $i=1,\ldots,n$ consists of the sum of the original microdata value, call it x_i, and the noise value, call it ϵ_i:

$$y_i = x_i + \epsilon_i.$$

Now consider the mean of the y_i, $\bar{y}$, and note that

$$\bar{y} = \bar{x} + \bar{\epsilon}$$

and

$$\mathrm{var}(\bar{y}) = \mathrm{var}(\bar{x}) + \mathrm{var}(\bar{\epsilon}).$$

If the variable y has been masked using noise generated by a random number generator,

$$\mathrm{var}(\bar{\epsilon}) = \mathrm{var}(\epsilon)/n \neq 0.$$

We could generate several noise data sets using a random number generator, and because the ϵ_i's are being taken from an infinite population, the $\bar{\epsilon}$'s will be different, so $\mathrm{var}(\bar{\epsilon}) \neq 0$. If, however, the noise used for masking has been constrained to have a given mean, then all possible noise data sets would have the same $\bar{\epsilon}$. Thus in the case of <u>constrained noise</u>, the variance of a sample mean is smaller because

$$\mathrm{var}(\bar{\epsilon}) = 0,$$

and

$$\mathrm{var}(\bar{y}) = \mathrm{var}(\bar{x}).$$

Also note that if <u>constrained noise</u> is used and $\bar{\epsilon} = 0$, then $\bar{y} = \bar{x}$.

b. SUBSETS

Users of microdata are often interested in specific subsets of the data.

See (McGuckin and Nguyen 1988). The benefits of using constrained noise in terms of sample variances of means of the full data set as described above extend to sample variances of the means of subsets of the masked data. Consider the same variable y of the masked data set with n records where

$$y_i = x_i + \epsilon_i,$$

the sum of the original data value and the noise value. A user might be particularly interested in a specific subset of the y_i. Call this subset ys_j $j=1,\ldots,n_1$ such that

$$ys_j = xs_j + \epsilon s_j$$

where the xs_j are the original data values of the subset and the ϵs_j are the noise values which were added to those original values. Then

$$\overline{ys} = \overline{xs} + \overline{\epsilon s}$$

and

$$\mathrm{var}(\overline{ys}) = \mathrm{var}(\overline{xs}) + \mathrm{var}(\overline{\epsilon s}).$$

If the variable y has been masked using noise generated by a random number generator, then

$$\mathrm{var}(\overline{\epsilon s}) = \mathrm{var}(\epsilon)/n_1$$

because the n_1 ϵs_j's have been generated from an infinite population. If, however, the noise used for masking this variable has been constrained to have a specific mean and variance, then

$$\mathrm{var}(\overline{\epsilon s}) = (1-n_1/n)*\mathrm{var}(\epsilon)/n_1.$$

Here, a finite correction factor can be used because the ϵj's are a subset of size n_1 of a finite noise data set of size n. Thus the use of constrained noise leads to a smaller variance of a sample mean of a subset of the masked data.

B. UNIFORM

Although noise that is added to data sets for the purpose of masking is usually thought of as from a multivariate normal distribution, one may want to add noise from other distributions. This was one reason for devising the algorithm which slightly adjusts the set of random numbers from the uniform distribution in order to obtain the exact mean and variance of that distribution. We hoped that such a constrained set of uniform random numbers would be useful when obtaining values from other distributions with given parameters. The reason for this is that generated random numbers from many distributions are often transformations of generated random numbers from the uniform distribution on the interval (0,1). It was hoped that a data set of random numbers from some distribution derived in such a manner would have parameter values closer to those desired if the set of uniform random numbers from which they came had a mean of exactly 1/2 and a variance of exactly 1/12 (the mean and variance of a uniform distribution on the interval (0,1)). Testing has shown, however, that this is not the case. In fact, under non-linear transformations, the constraints on the original data set do not ensure the corresponding constraints on the parameters of the resulting data set.

IV. CONCLUSION

In this report, we have presented two algorithms which transform data generated by a random number generator into data satisfying certain constraints. The motivation of these algorithms and the benefits of their use have been discussed in terms of masking microdata for disclosure avoidance purposes. Examples of program performance have also been provided. Code is available from the author.

V. ACKNOWLEDGEMENTS

I thank Jay Kim for pointing out this problem to me and encouraging me to work on it. I also thank Jim Fagan, Brian Greenberg, and Jay Kim for a number of helpful comments given during the pursuit of this work.

VI. REFERENCES

Cox, L. H., McDonald, S., and Nelson, D. (1986), "Confidentiality Issues at the United States Bureau of the Census", Journal of Official Statistics, Vol. 2, No. 2, 135-160.

Kennedy, W. J. and Gentle, J. E. (1980), Statistical Computing, Marcel Dekker, Inc., New York, 294-301.

Kim, J. (1986), "A Method for Limiting Disclosure in Microdata Based on Random Noise and Transformation", Proceedings of the Section on Survey Research, American Statistical Association, 370-374.

McGuckin, R. H. and Nguyen, S. V. (1988), "Use of 'Surrogate Files' to Conduct Economic Studies with Longitudinal Microdata", Proceedings of the Third Annual Research Conference; Bureau of the Census, Washington, D.C., 193-209.

Spruill, N. L. (1982), "Measures of Confidentiality", Proceedings of the Section on Survey Research Methods, American Statistical Association, Washington, D.C., 260-265.

Wolf, M. K. (1988), "Microaggregation and Disclosure Avoidance for Economic Establishment Data", Proceedings of the Section on Business and Economic Statistics, American Statistical Association, Washington, D.C., 355-360.