INFORMATION TO USERS

The most advanced technology has been used to photograph and reproduce this manuscript from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book. These are also available as one exposure on a standard 35mm slide or as a 17" x 23" black and white photographic print for an additional charge.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
The use of added error to avoid disclosure in microdata releases

Sullivan, Gary R., Ph.D.
Iowa State University, 1989
The use of added error to avoid disclosure in microdata releases

by

Gary R. Sullivan

A Dissertation Submitted to the Graduate Faculty in Partial Fulfillment of the Requirements for the Degree of DOCTOR OF PHILOSOPHY

Major: Statistics

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Iowa State University
Ames, Iowa

1989
**TABLE OF CONTENTS**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. INTRODUCTION AND LITERATURE REVIEW</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Literature Review</td>
<td>2</td>
</tr>
<tr>
<td>2. MODEL AND RESULTS FOR THE NORMAL DISTRIBUTION</td>
<td>27</td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>27</td>
</tr>
<tr>
<td>2.2 A General Approach</td>
<td>28</td>
</tr>
<tr>
<td>2.2.1 The confidentiality problem</td>
<td>28</td>
</tr>
<tr>
<td>2.2.2 Prediction of confidential variables</td>
<td>32</td>
</tr>
<tr>
<td>2.3 Preserving Confidentiality in Microdata Releases</td>
<td>44</td>
</tr>
<tr>
<td>2.3.1 The multivariate model</td>
<td>44</td>
</tr>
<tr>
<td>2.3.2 Choice of correlation error</td>
<td>46</td>
</tr>
<tr>
<td>3. DESCRIPTION OF THE MASKING ALGORITHM</td>
<td>59</td>
</tr>
<tr>
<td>3.1 Transformation of Variables</td>
<td>60</td>
</tr>
<tr>
<td>3.1.1 Construction of Bernoulli variables from classification variables</td>
<td>60</td>
</tr>
<tr>
<td>3.1.2 Transformation of quantitative variables to normality</td>
<td>61</td>
</tr>
<tr>
<td>3.1.3 Transformation of Bernoulli variables to normality</td>
<td>65</td>
</tr>
<tr>
<td>3.2 Masking the Normal Data Vectors</td>
<td>68</td>
</tr>
<tr>
<td>3.2.1 Construction of error vectors</td>
<td>69</td>
</tr>
<tr>
<td>3.2.2 The initial data mask and the distance criterion</td>
<td>70</td>
</tr>
<tr>
<td>3.3 Iterations to Attain Desired Correlations</td>
<td>72</td>
</tr>
<tr>
<td>3.3.1 Back transforming the data to original scale</td>
<td>72</td>
</tr>
<tr>
<td>3.3.2 The iterations for cross correlations</td>
<td>76</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>3.3.3 Iterations to adjust correlation structure of masked variables</td>
<td>78</td>
</tr>
<tr>
<td>3.4 Back Transforming Classification Variables</td>
<td>80</td>
</tr>
<tr>
<td>4. RESULTS OF MASKING TEST DATA</td>
<td>82</td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>82</td>
</tr>
<tr>
<td>4.1.1 Other data masking experiments involving adding error</td>
<td>82</td>
</tr>
<tr>
<td>4.1.2 Use of the masking algorithm</td>
<td>88</td>
</tr>
<tr>
<td>4.2 Mask of the First Data Set</td>
<td>89</td>
</tr>
<tr>
<td>4.2.1 Plots and two-way tables</td>
<td>91</td>
</tr>
<tr>
<td>4.2.2 Regression analyses of original and masked data sets</td>
<td>96</td>
</tr>
<tr>
<td>4.2.3 Examination of correlation structures and cross correlations</td>
<td>101</td>
</tr>
<tr>
<td>4.2.4 Comparison of univariate distribution functions</td>
<td>108</td>
</tr>
<tr>
<td>4.2.5 Assessment of protection against disclosure</td>
<td>108</td>
</tr>
<tr>
<td>4.3 Mask of the Second Data Set</td>
<td>117</td>
</tr>
<tr>
<td>4.3.1 Cross tabulations of the discrete and classification variables</td>
<td>120</td>
</tr>
<tr>
<td>4.3.2 Regression analysis with classificatory predictor</td>
<td>123</td>
</tr>
<tr>
<td>4.3.3 Examination of overall correlation structure and cross correlations</td>
<td>125</td>
</tr>
<tr>
<td>4.3.4 Comparisons within subgroups of the classification variable</td>
<td>129</td>
</tr>
<tr>
<td>4.3.5 Disparity of joint distribution functions</td>
<td>135</td>
</tr>
<tr>
<td>5. REFERENCES</td>
<td>141</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6. ACKNOWLEDGEMENTS</td>
<td>144</td>
</tr>
<tr>
<td>7. APPENDIX</td>
<td>145</td>
</tr>
<tr>
<td>7.1 An Alternative Prediction Approach</td>
<td>145</td>
</tr>
<tr>
<td>7.2 Preserving Confidentiality in the Alternative Approach</td>
<td>149</td>
</tr>
<tr>
<td>7.3 Differences in the Predictors</td>
<td>152</td>
</tr>
<tr>
<td>7.4 The Masking Program</td>
<td>154</td>
</tr>
</tbody>
</table>
1. INTRODUCTION AND LITERATURE REVIEW

1.1 Introduction

The expanding capacity of computers has produced an increasing demand for microdata. Government agencies like the Census Bureau and the National Center for Health Statistics receive data requests from economic, business and medical researchers. These government agencies, as well as other data providers, are faced with the problem of supplying statistically useful data that will not reveal the identity, or any confidential attribute, of a respondent.

For example, suppose an agency releases tables of annual income cross-classified by sex and occupation for residents of Boone, Iowa. Furthermore, suppose Joseph Average is the only male statistician living in Boone, Iowa. Any user of the table, who knows that Joseph Average is the only male statistician in Boone, Iowa, can obtain confidential information, such as annual income, about Joseph Average. Such an occurrence is called a case of attribute disclosure. As another example, suppose an agency releases a data set containing individual records. The records have been stripped of identifiers, such as names and addresses. Identity disclosure occurs if a user is able to positively link a respondent to the record of that respondent.

Data agencies must devise methods to prevent such disclosures in order to satisfy the pledges of confidentiality given to respondents. The methods employed to protect the anonymity of respondents are known as disclosure avoidance techniques.
1.2 Literature Review

The statistical community became concerned about maintaining respondent confidentiality in the late 1960s when files of linked records, data banks, and statistical file systems were initially requested by researchers. Steinberg and Pritzker (1967) suggested that files of linked records be created in a manner that would maintain confidentiality. They advised record linkers to "expunge all individual identifiers at the instant of creation." Bachi and Baron (1969) give a summary of the confidentiality problems faced in setting up data banks or linking records between data files.

Although these were two of the first articles to address the general problem from a statistical standpoint, the issue of data collection and dissemination has a long legal and historical background. Duncan and Lambert (1986) provide a good review of the federal statutes dealing with confidentiality. The earliest legislation allowing government use of private information was the Internal Revenue Act of 1864. The 1948 Trade Secrets Act protected statistical data of business establishments under the Criminal Code. More recent legislation regarding data dissemination and respondent confidentiality is the Freedom of Information Act (1966) which attempted to extend public access to government information with certain limitations. In 1970, a federal law (Title 20 USC 1232g) required that schools at all levels obtain the consent of students or parents before releasing student data for non-academic studies. In 1974, the Privacy Act
required the federal government to formulate confidentiality guidelines under which they would disseminate data. The key legislation concerning the collection and protection of data is Title 13 of the U.S. Code, under which the Census Bureau operates. Title 13 states that "collected data can be used for statistical purposes only and cannot be published in a manner whereby the data furnished by any particular establishment or individual can be identified." Abiding by the laws of confidentiality and still providing statistically useful data is not a simple task. As Mugge (1983) states, "there are so many different items of information about any subject individual or establishment in our typical surveys that the set of information could serve as a unique identifier for each subject."

Data collected by government agencies plays an important role in government policy making, as well as in the private sector. However, data collection is threatened if individuals are reluctant to respond because they feel their confidentiality rights are threatened. Duncan and Lambert (1986) list two cases where such reluctance was critical. The first is a proposed study of draft evaders that could not be conducted because the respondents did not feel convinced their anonymity could be assured. Duncan and Lambert also mention the decision of the Constitutional Court of West Germany to postpone the census after over one thousand lawsuits were filed against the census. The government agencies collecting data must assure potential respondents that their confidentiality rights are not being violated; hence, the need for research in the areas of disclosure avoidance.
Although research in the area of confidentiality is in its early stages, some standard techniques for various forms of released data are practiced by the releasing agencies. Mugge (1983) discusses confidentiality measures taken at the National Center for Health Statistics. Cox et al. (1985) provide a good discussion of Census Bureau data products and the techniques used to mask them before release. A summary of this discussion and other related articles is now given.

The Census Bureau releases data in three forms. These are sets of frequency count tables, sets of tables of aggregate magnitude data, and microdata files. Frequency count tables are two way tables of counts, which can be combined to form three or higher dimensional tables. For example, table counts for sex by income group may be released for an exhaustive set of \( k \) age groups. These \( k \) two-way tables form a single three-way table of sex by income group by age group. Hence, disclosure avoidance techniques are applied within and between sets of two-way tables, because disclosure of a respondent's attributes can occur if an intruder carefully examines relationships within and between tables. Four methods have been listed as possibilities to mask frequency count tables constructed from 1990 census data. These are cell suppression, random data perturbation, random rounding and controlled rounding. Descriptions and examples of these techniques are presented in the Cox et al. (1985) article. Also, see Fellegi (1975), Cox (1980), and Cox et al. (1986).

The Census Bureau also releases aggregate data for business establishments in the form of statistical tables. Sales of a certain
product or expenditures on a service are examples of the types of
aggregate data compiled and released by the Census Bureau. With
aggregate data, the confidentiality concern is related to the
contribution of a single business to the total for a data cell. For
example, suppose a table is constructed which contains the total
dishwasher sales of all businesses in Story County, Iowa. If only one
store sold dishwashers, a knowledgeable person would know this and a
disclosure would result. If three businesses contribute to a cell, a
coalition of two businesses could share their sales information to
expose the sales of the remaining company. To avoid such a problem, the
Census Bureau employs the \((n, k)\) cell concentration rule. The rule
states that if \(n\) or fewer respondents contribute more than \(k\) percent
to a cell value, the cell is deemed a disclosure cell. If such a
situation arises, the cell can be suppressed or collapsed with another
cell. As an additional protective measure, the Census Bureau treats the
values of \(n\) and \(k\) as confidential information. Cell suppression and
replacement of cell values by value ranges are two other disclosure
avoidance methods for aggregate data which are described in the Cox et
al. (1985) article.

The Census Bureau also releases data in the form of microdata
records. A microdata record contains detailed information about an
individual respondent. A sample of records, or microdata file, is a
valuable asset in economic modeling, statistical analysis and general
research. Unfortunately, public access to raw microdata records poses a
direct threat to confidentiality. Even after identifiers such as name
and address are stripped from the records, an indirect threat may still exist if the remaining information is abundant. Additional confidentiality measures must then be taken.

The Census Bureau follows a practice of suppressing geographic detail of a record drawn from a population of 100,000 persons or fewer. In addition, all released files are scrambled with respect to geographic sort order. Other disclosure control techniques for microdata records include collapsing, top coding, replacing tail observations, rounding, and adding random error. The technique of adding random error will be discussed at length later in this chapter.

Collapsing is the operation of representing continuous data in a categorical form. For example, a respondent’s income may be recorded to the nearest one hundred dollars. However, before a record is released, the actual income value may be replaced by a categorical value representing an income in a range of 5,000 dollars. Topcoding is a special form of collapsing, whereby large response values are grouped in a single category. In the income example, very large incomes pose a clear threat to confidentiality. Hence, all incomes exceeding $100,000 may be coded into an "Over $100,000" category. Replacing tail observations is another approach to the problem of small or large values. Under this method, the small (or large) values are not grouped into a category, but are replaced by the mean of the smallest (or largest) \( k \) percent of all values.

Rounding is a technique which can provide protection for non-categorical variables. Cox et al. (1985) suggest that a controlled
rounding method should be employed. Controlled rounding is a procedure that preserves the unbiasedness of estimates such as means and regression coefficients. Spruill (1982) presents a random rounding technique in which each value is rounded to either the smallest or largest closest multiple based on a probability, \( p \). The probability is usually equal to 0.5. Zero values may also be set to non-zero values with a small probability.

Spruill (1982) also discusses a data swapping procedure. Under this method, each of \( N \) records is divided into three sub-records having approximately one-third of the total variables of each record. Each sub-record based on the first one-third of the variables is matched with two others which have similar variable values. The remaining two-thirds of each record corresponding to these three subrecords is carried along to form a group of three full records. Three new records are formed by combining a subrecord from each of the three full records. For example, the first new record may consist of the first subrecord from record one, the third subrecord from record two and the second subrecord from record three. Two other new records are formed similarly from what is left.

Another form of disclosure control which has been researched is the technique of multiplying values of microdata records by random error. Spruill (1982) suggests multiplying data values by a non-normal error, \( \epsilon \), where \( \epsilon \) is a value between 0 and \( T \) (\( T = (\alpha + 1)^{-1}(\alpha + 2) \), \( \alpha > -1 \) ). Then each data value \( X \) is replaced by \( X\epsilon \) or 0, depending on the distribution of the variable (i.e., whether or not zero
is a common response). McGuckin and Nguyen (1988) suggest a deterministic transformation which disguises the data element $X$ as $X^*$ using

$$X^* = \phi X^\theta$$

where $\phi > 0$ and $\theta > 0$. Choosing $\theta = 1$ results in a multiplicative transformation similar to that discussed by Spruill. In fitting log-linear regression models, the deterministic transformation provides the same parameter estimates as the original data.

Microaggregation is another disclosure control method used to create microdata files for business establishment data. Microaggregation consists of creating synthetic records from suitable subsets of original microdata records. The synthetic records are then released as a microdata file. Different methods of microaggregation are created by the choice of the method of defining subsets of original records. Also, there are variations in the manner in which the synthetic records are formed. Govoni and Waite (1985) suggest ranking the $N$ establishments based on a single criterion variable. In their example, this variable is values of shipments. Groups of establishment records of size $m$ are created from the ranked list. A synthetic record is created from each group of $m$ by taking an average of the $m$ data vectors. The resulting file contains approximately $[N/m]$ synthetic records.

Wolf (1988) proposes several methods of creating synthetic records using general microaggregation ideas. Wolf's general method is as
follows: First, define a proximity measure for any two records based on a subset of the variables in each record. Select a target establishment record and define a cluster of the \( k \) records closest to the target record based on the proximity measure. The cluster could also be formed by choosing records within a given distance of the target establishment. Then, form a surrogate record by taking a weighted average of the records in the cluster. The \( k \) records are then removed from the data file, a new target establishment is selected, and the process is repeated until the file is exhausted.

Because establishment data usually include highly correlated subsets of variables, Wolf suggests avoiding redundancies through a principal components approach to the microaggregation technique. In addition, because business data are often longitudinal (i.e., measurements are recorded for the same establishments at specific time periods), the principal component analysis could be performed at a number of levels. For example, the principal components for each establishment can be constructed using the sample covariance matrix of all establishment data for the year. Because clusters of records can be formed using any one of many proximity measures, different synthetic records will result if different measures are used. For example, using average Euclidean distance over time will produce different aggregate clusters and different synthetic records than would be obtained using average Mahalanobis distance over time as the proximity measure.

In addition, Wolf also discusses tests which can be used to compare the original file to the microaggregate file. These include the
Kolmogorov-Smirnov test of two univariate distribution functions and a likelihood ratio test for the homogeneity of original and microaggregate correlation matrices. Finally, Wolf points out that microaggregation is a linear transformation of the data matrix, and provides a brief discussion of the matrix algebra theory.

With regard to disclosure controls for microdata records, McGuckin and Nguyen (1988) provide a good discussion of the advantages and disadvantages of some of the techniques presented here. In addition, the authors list several characteristics which they feel every public use data file should possess. First, the released data file should provide the same parameter estimates of a model as are obtained from analyzing the original data records. Also, it is important that a microdata file is capable of being linked with surrogate data records available from other sources. Furthermore, a microdata file should allow users the capability of working with relevant subsets of the file. Finally, a public use file should have the capability of being expanded easily to include data on new respondents or data for new time periods. These are valuable guidelines for a data provider to follow when preparing a file for public use.

Other alternatives to releasing masked microdata records or synthetic records have been suggested. One possibility discussed by McGuckin and Nguyen (1988) is the release of the variance covariance matrix and mean vector of the data. This would insure little risk of disclosure and would allow researchers to obtain correct model parameter
estimates. However, a release restricted to these sample statistics allows no opportunity to study subsets of the data.

Another alternative to microdata file releases is a public access database. Gates (1988) discusses the Luxembourg Income Study (LIS) database which could be accessed worldwide through a telecommunications network. Users of the database did not have direct access, but could submit program jobs to be run by the LIS staff. Users were provided with a technical description of the data set, a description of the variables, a codebook, a small sample file of the LIS database and information on software packages that could be used to analyze the data. With the database and other information, the potential user could design a study and perform necessary data analysis by accessing the database with submitted programs. The mode of research is little different than if the user had direct access to the data, except that analysis is less convenient because of the time it takes to get results.

The providers of the public access database must be wary of users designing output requests to compromise the database. Palley and Simonoff (1986) detail how such a database could be compromised through regression methodology. Palley and Simonoff first assume a regression model can be fit to a confidential variable based on several nonconfidential variables. An example is an individual's salary as a function of position in the company, age, education and years with the company. Even if users of the database are not permitted to submit regression runs due to confidentiality concerns, the database can still be compromised because the users can build a synthetic database by
submitting appropriate program runs. By requesting histograms of explanatory variables, the user can categorize continuous or non-classification variables. The user can then submit job runs for frequency counts of combinations of the categorized explanatory variables, along with means and standard deviations of the confidential variable for such combinations. Based on the counts of various combinations, and the mean and standard deviation of the confidential variable for these combinations, the user can create synthetic records comprised of the explanatory variables and the confidential variable. The result is a synthetic database from which a regression model can be fit.

Palley and Simonoff also state that this regression-based compromise approach was validated using actual census data. Synthetic records were created and a regression model was fit. The multiple R-squared values obtained from the synthetic database were quite close to those obtained from the regression on the original database.

Gates (1988) discusses additional concerns should the public access database method be adopted at the Census Bureau. These include concerns over access to data through the Census Bureau's telephone lines to its mainframe computer and fear that records from sample files may be linked to records in a public use file.

Basically, the confidentiality problem can be defined as an intruder attempting to pinpoint a respondent's identity or attributes. In a general approach to this problem, Duncan and Lambert (1986, 1989) attempt to quantify the intruder's beliefs about the target in the form
of an uncertainty function, $U(\cdot)$ . The uncertainty function is then used to formulate disclosure measures and disclosure control techniques.

The uncertainty function was first proposed by DeGroot (1962). DeGroot posed the problem of investigating the value of an unknown parameter, $\theta$ , through a sequence of experiments. In the context of the confidentiality problem, $\theta$ could be a target identity or characteristic value. After each experiment the knowledge or belief of the true value of $\theta$ can be expressed as a probability distribution, $\xi$ . Associated with each $\xi$ is an amount of uncertainty in the mind of the experimenter. The uncertainty at each step is quantified by evaluating $U(\cdot)$ , a non-negative concave function, at the current estimate of the probability distribution, $\xi$ . As experimentation continued, the natural progression would be for the uncertainty to decrease toward zero.

In their first paper, Duncan and Lambert (1986) research disclosure limiting procedures using general uncertainty function ideas. Assume an intruder has beliefs about a target value or individual in the form of a probability distribution. When a statistic, $S$ , is released, the intruder updates the probability distribution, called the predictive distribution by Duncan and Lambert. Three disclosure measures are proposed by Duncan and Lambert. These measures are used to formulate disclosure control techniques. The measures are called knowledge, knowledge gain and relative knowledge gain. In order to explain these terms, some notation must be defined. First, define $\xi_{pr}$ to be the intruder's predictive distribution of the target value prior to the
release of the statistic, $S$. Similarly, define $\xi_{po}$ to be the intruder's predictive distribution of the target value after the release of the statistic, $S$. Finally, the intruder's uncertainty about the target is summarized by applying $U(\cdot)$, a concave nonnegative uncertainty function, to the intruder's predictive distribution.

The knowledge measure of disclosure is $U(\xi_{po})$. The knowledge gain is the difference in the uncertainty functions evaluated at the prior and posterior distributions, namely, knowledge gain is $U(\xi_{pr}) - U(\xi_{po})$. The relative knowledge gain is a scale invariant form of the knowledge gain and is $[U(\xi_{pr})]^{-1}[U(\xi_{pr}) - U(\xi_{po})]$. Given these disclosure measures, disclosure limiting rules can be formulated for the release of a statistic, $S$. For example, only permit the release of a file if the knowledge gain from that file, $U(\xi_{pr}) - U(\xi_{po})$, is less than some positive real number, $\tau$. The value of $\tau$ depends on the level of protection desired, although the need of information to perform valid inference is an additional consideration in setting $\tau$.

In a second article, Duncan and Lambert (1989) take a more detailed approach to the confidentiality problem, using loss functions in conjunction with DeGroot's ideas on quantifying uncertainty. The research considers the case where the intruder's objective is identity disclosure and the case where the objective is attribute disclosure. Assume a data matrix of all records is denoted by an $N \times K$ matrix $X$. A microdata file, $Y (n \times k)$ is constructed by taking a subset of the rows and columns of $X$. (i.e., some variables and records are excluded from $X$ to form $Y$.) Suppose the intruder's target is $t_0$.,
which is an identifier $x_{10}$, if the intruder's objective is identity disclosure. Assume the target is the characteristic $x_{1j}$ if the objective is attribute disclosure. Also, given that $Y = y$, assume $p(s)$ is a predictive distribution describing the intruder's current beliefs about possible values $s$ of the target, and let $L(t, s)$ be the loss incurred by the intruder when the target is said to be $t$ but is $s$. Then, the intruder's uncertainty function about the target is

$$U(y) = \inf_{t} \int L(t, s)p(s)ds.$$ 

Given a possible loss function, the releasing agency can influence the expected loss for the intruder by controlling $Y$. In the case of identity disclosure, suppose

$$L(link, x_{10}) = \begin{cases} 
0, & \text{if link} = y_{10} \text{ and } y_{10} = x_{10} \\
l_1, & \text{if link} = \phi, \ x_{10} \in Y_0 \\
l_2, & \text{if link} = y_{10} \text{ and } y_{10} \neq x_{10} \ (1 \leq i \leq n) 
\end{cases}$$

where $Y_0$ contains $n$ records. If no record in the release is linked to the target, the link is null (link = $\phi$) and the intruder's expected loss is $l_1 \Sigma_{i=1}^{n} p(y_{1i})$. If a link is made, the intruder expects to incur a loss of $l_2[1-p(y_{1i})]$. Hence, the uncertainty about the target is

$$U(y) = \min(l_1 \Sigma_{i=1}^{n} p(y_{1i}), l_2[1 - \max_{1 \leq i \leq n} p(y_{1i})]).$$
The goal of the agency is to control $Y$ so

$$\ell_2[1 - \max p(y_{i0})] > \ell_1 \sum_{i=1}^{n} (y_{i0}) .$$

Duncan and Lambert consider the problem for cases of releasing the data to a naive intruder, an informed intruder and an intruder with some knowledge.

In the case of an intruder attempting attribute disclosure, the approach is similar. The difference is that the loss function becomes more complicated because a correct link requires a correct record identification and a correct attribute determination.

Spruill (1982) proposes a measure of confidentiality to evaluate the effectiveness of various disclosure avoidance techniques applied to microdata files. These are grouping (or microaggregation), random rounding, data swapping, multiplying by random error and adding random error. To insure confidentiality for a microdata release, it would be desirable if no record in the release could be matched successfully to a record in a public use file. There is no way to keep track of all public use files, and, hence, a check for matching is impossible. However, by attempting to match the masked records of the microdata release with the unmasked originals, a conservative check can be made.

In performing this check, Spruill proposes the following strategy. Compare each data record in the released file with each record from the original file. (Assume common records exist in these files.) The comparison is based on the distances between each masked record and
every original record. Each distance is computed as the sum of absolute deviations or the sum of squared deviations for all standardized data elements forming the records. Each masked record is then associated with the original record which corresponds to the smallest sum of deviations. A link is said to be made if a masked record is associated with its original counterpart. The proposed confidentiality criteria is the percentage of records for which a link is not made. This criteria can be redefined to be the percentage of records for which their unmasked counterpart is not among the three closest records.

With the exception of adding random error, all masking methods evaluated by Spruill have been discussed previously in this chapter. There are many ways of adding random error to mask data records, some of which will be discussed in this research. One strategy is to add to each data element a normal error, $u$, with mean zero and variance $k\sigma_{XX}$, where $k$ is a fixed number and $\sigma_{XX}$ is the variance of the underlying $X$ variable. Spruill's method is to replace any nonzero data element $x$ with $x + u$ or 0 and any zero data element with 0 or $x^* + u$, where $x^*$ is a value generated from the true underlying distribution of $X$.

Population files with 1,000 records and 36 variables were generated based on means and coefficients of variation of variables from an actual file. The confidentiality criteria was evaluated for records having anywhere from one to 32 variables. The results are based on ten 1% data releases for each of two generated populations. Spruill's results indicate that, for this size of sample, random rounding and data
swapping provide very little protection from disclosure. Grouping and multiplying by random error provide adequate confidentiality for up to nine common variables. Adding random error also provides good protection as long as the variance of the error is at least half the variance of the underlying X variable. Since confidentiality problems occur more frequently as the number of variables increases, Spruill suggests releasing data sets containing subsets of variables.

In another paper, Spruill (1983) examines the analytic usefulness of business microdata masked by each of the previous five disclosure avoidance techniques. Spruill investigates the effect of each masking method on the estimation of sample statistics and regression coefficients. Two data sets were used to evaluate masking effects. The first data set was test data generated from sample statistics of an IRS data file, consisting of 36 variables and 1,000 records. The second data set consisted of 1,000 records of actual tax return data (1979) having 27 economic variables. Spruill concluded that the random rounding technique provided the most analytically useful data among procedures that afforded adequate confidentiality. Data masked by adding random error provided reasonable estimates for the sample mean, but such data provided inaccurate standard deviations, correlations, and regression coefficients.

Paass (1985) investigates the case of an intruder attempting to determine the identity of a record in a microdata release by matching to a record in a public use file. Paass does not assume the records of the microdata release to be masked by adding error. However, Paass believes
the intruder's biggest problem in attempting to disclose identity is caused by measurement error in that the records in the microdata release will differ from those in the public use file. The problem defined by Paass is similar to matching records masked with added error to the unmasked originals.

Paass first discusses an identification method for known distributions. Assume \( U = \{u_1, \ldots, u_N\} \) is the set of all persons in the population. Let \( MD_u = \{y_1, \ldots, y_N\} \) define the set of all records in the microdata file, and let \( AK_u = \{x_1, \ldots, x_N\} \) define the set of records containing known data. For example, \( MD_u \) is the file from which the microdata release is generated, and \( AK_u \) is a public file having a corresponding data record for each record in the microdata file. The differences between the records are described completely by the error distribution, \( f(x|y_i) \), which is discussed below. Finally, assume the microdata release file is defined by \( MD = \{y_1, \ldots, y_n\} \), where \( n \leq N \). Assume the intruder's additional knowledge is one data record \( x \in AK_u \) and the intruder's objective is to find the microdata record corresponding to \( x \). Identifying the target, \( x \), can be posed as a problem of discriminant analysis.

Paass defines \( \Omega \) to consist of \( m+1 \) disjoint classes, \( \Omega_1, \Omega_2, \ldots, \Omega_{m+1} \). The classes \( \Omega_1, \Omega_2, \ldots, \Omega_m \) correspond to the data records \( y_1, y_2, \ldots, y_m \) in the microdata release file. The class \( \Omega_{m+1} \) is the residual class and contains all records not released. Bayesian methods can be used to estimate the posterior probabilities of \( x \) belonging to each class, \( \Omega_k \) (\( k=1, 2, \ldots, m \)). First, define
p(k) as the prior probability that the record x belongs to class \( \Omega_k \). Then

\[
p(k) = N^{-1} \quad \text{for } k = 1, 2, \ldots, m
\]

\[
= N^{-1}(N-m) \quad \text{for } k = m+1.
\]

The conditional distribution of x for a given class k is given by

\[
f(x|k) = f(x|y_k) \quad \text{for } k = 1, 2, \ldots, m
\]

\[
= (N-m)^{-1} \sum_{i=m+1}^{N} f(x|y_i) \quad \text{for } k = m+1.
\]

It follows that the distribution of x is

\[
f(x) = \sum_{k=1}^{m+1} p(k)f(x|y_k).
\]

Then, by Bayes formula, the posterior probability, p(k|x), that the observed x belongs to class k is

\[
p(k|x) = [f(x)]^{-1}p(k)f(x|k).
\]

A decision rule can be formed by choosing a cutoff \( p_0 \) and assigning x to class k if p(k|x) > \( p_0 \).
Paass also considers the problem of identification when neither \( f(x|k) \), \( k=1, 2, \ldots, m+1 \), nor \( f(x) \) is known. The approach is to estimate the distribution on the basis of what is known and then to insert the estimates into Bayes' formula. Assume the intruder has some knowledge of the observational error structures of \( MD_u \) and \( AK_u \). Through simulations, the intruder can generate a sample

\[
S(y_k) = \{x_{k,1}, \ldots, x_{k,n(k)}\}
\]

for each \( y_k \in MD \), and hence estimate parameters of the error distribution \( f(x|k) = f(x|y_k) \). Similarly, a set of records containing additional knowledge, \( S_{Ak} = \{x_1, \ldots, x_m\} \) can be generated according to \( f(x|y_k) \).

The identification of a target record, \( x \), is broken up into two steps. First, linear discriminant analysis is used to search for the target record \( \hat{k} \), providing the univariate components of \( f(x|k) \) are symmetric. After \( \hat{k} \) is determined, all remaining classes or records are combined to form an alternative class \( \Omega_{\hat{k}} \), where \( \Omega_{\hat{k}} \) is the union of all classes \( \Omega_k \), \( k \neq \hat{k} \). Under weak distributional assumptions, \( f(x|\hat{k}) \) and \( f(x|\bar{k}) \) can then be estimated for small regions \( \xi \in S \), where \( S \) is the sample space of the data records. Substituting \( f(x|\hat{k}) \) and \( f(\xi|\bar{k}) \) into the Bayes formula gives

\[
p(\hat{k}|x) = \frac{[f(\xi|\hat{k}) + (N-1)f(\xi|\bar{k})]^{-1}[f(\xi|\hat{k})]}{[f(\xi|\hat{k})]}
\]

Then, \( p(\hat{k}|x) \) can be used to estimate the probability of correct association for the presumed data record \( \hat{k} \).
To investigate the dependency of disclosure risk on factors such as type of common variables and structure of observational errors, Paass carried out disclosure experiments using two concrete Federal Republic of Germany data files - the 1978 income and consumption sample and the 1978 microcensus. Six realistic situations involving from seven to 68 variables were studied. To simulate a target search, an element of additional knowledge, \( u_k \), was created for each \( y_k \) in the microdata file using an error distribution model. Different error fractions were assumed for different types of variables. For example, year of birth was assumed to have a 1% error rate while income was assumed to have a 90% error rate.

The results indicated that the threat to respondent confidentiality is dependent upon the number of common variables and the frequency and distribution of the respective variable values. Paass also discovered that the intruder's knowledge about the observational error structure played a minor role in the success of the identification process. This led Paass to conclude that adding error alone is insufficient to guard against identity disclosure.

Paass further concludes that microdata files containing few variables can be released if standard disclosure measures are taken (e.g., removing identifiers and censoring outliers in the data). With regard to data files with many variables, Paass believes "the conflict between data demand of science and data protection requirements cannot be solved completely by disclosure avoidance techniques." Therefore,
Paass suggests special legal regulations which would bring sanctions against intruders who identify data records.

In response to Paass' findings, Kim (1986) proposed a masking scheme which combines the addition of error with a linear transformation that adds an additional layer of protection. As Paass noted, the more data points cluster in a given space, the more difficult identity disclosure becomes. Kim's objective is to compact the data points around the mean without disturbing the underlying correlation structure. A brief description of Kim's scheme is now given.

Define \( x_i \) as the variable to be masked and \( e_i \) as the random noise added to \( x_i \). Assume \( x_i \sim (\mu_i, \sigma_i^2) \) and \( e_i \sim (0, c\sigma_i^2) \), where \( c > 0 \) and \( x_i \) is independent of \( e_i \). Also, assume

\[
\text{Cov}(e_i, e_k) = c \text{Cov}(x_i, x_k)
\]

Hence, define

\[
y_{ij} = x_{ij} + e_{ij}, \quad i=1, 2, \ldots, p; \quad j=1, \ldots, n.
\]

Furthermore, transform \( y_{ij} \) by

\[
z_{ij} = ay_{ij} + b_i, \quad i=1, 2, \ldots, p; \quad j=1, \ldots, n,
\]

where \( a \) and \( b_i \) satisfy \( E(x_i) = E(Z_i) \) and \( V(x_i) = V(Z_i) \) for \( i=1, 2, \ldots, p \). That is, the first and second moments of the transformed variables are identical to the first and second moments of the original
variables. The constraint $E(x_i) = E(Z_i)$ implies

$$a\mu_i + b_i = \mu_i ,$$

$$b_i = (1-a)\mu_i .$$

Estimating $\mu_i$ by $\bar{x}_i$ gives

$$Z_{ij} = ay_{ij} + (1-a)\bar{x}_i .$$

Restricting $V(Z_i) = V(x_i)$ yields

$$a = [(n-1)(c+1)]^{-1/2} (n-1-c)^{1/2} .$$

For $n$ large, this reduces to $a = (c+1)^{-1/2} .$

For $\sigma_i^2$ known, $\mu_i$ estimated by $\bar{x}_i$, and large $n$, Kim shows that

$$E(Z_i) = E(x_i) ,$$

$$\text{Corr}(z_i, z_j) = \text{Corr}(x_i, x_j) ,$$

$$V(Z_i) = V(x_i) .$$

For $\sigma_i^2$ unknown, $\mu_i$ estimated by $\bar{x}_i$, and large $n$, Kim
demonstrates that
\[ E(Z_i) = E(x_i), \]
\[ V(Z_i) = V(x_i). \]

Other properties concerning correlations and variances are also given.

Kim also investigates the effects of this masking technique on regression analyses for cases where \( \sigma^2 \) is both known and unknown. For the case \( \sigma^2 \) known, if the first two moments of the original variables are preserved by the mask of the data, Kim shows that the estimated regression coefficients and intercept are, on the average, identical to those based on the original data. Also, standard errors of the regression coefficients based on this masking scheme are not different from those obtained from regression on the original data. The same is true for the standard error of the intercept. Likewise, the residual error variance of the regression on the masked data is the same as the residual error variance resulting from regression on the original data. Kim shows that the same claims about the standard error of the intercept and the residual error variance of the regression cannot be made for data masked by the standard approach of simply adding random noise \( (x_i^* = x_i + u_i) \).

For \( \sigma^2 \) unknown, Kim investigated the properties of masked data by computing correlations and running regressions on the original data, on data masked under Kim's scheme and on data masked under the standard
random noise approach. The original data were vectors of earnings data. Results from correlation computations showed that Kim's scheme preserved the correlation structure of the original data to a higher degree than did the standard random noise approach. Regressions involving masked variables displayed the same general results with regard to estimated coefficients, standard errors and residual variances.

In conclusion, Kim states that the amount of disclosure protection provided by the transformation scheme is unknown. Kim planned reidentification experiments similar to those made by Paass. Kim also felt that by lowering the value of "a" in his transformation, data records can be shifted toward the mean, thus reducing the possibility of identity disclosure.

The masking scheme proposed by Kim preserves valuable statistical properties of the original data, such as moments, correlation structure and regression properties. However, McGuckin and Nguyen (1988) point out two disadvantages of Kim's scheme. First, by constraining the covariance structure of the random errors, the researcher using the released data is restricted. Statistical properties of subsets of the data are not necessarily preserved. That is, subsets of the data may not have the same covariance structure as the entire data set. Furthermore, the addition of random noise distorts the original variables so that new variables, like first differences or growth rates, cannot be created and used in analysis.
2. MODEL AND RESULTS FOR THE NORMAL DISTRIBUTION

In this chapter, we consider the confidentiality problem. We assume that the agency releasing microdata has masked the data by adding error to the observation. We also assume that the original vector of observations is normally distributed. In Section 2.2, we study the problem from the standpoint of an intruder wishing to predict the confidential variables of a target record. In Section 2.3, we investigate the effect of the covariance matrix of the added error vector on the accuracy of the predictor derived in Section 2.2.

2.1 Introduction

Let \( S = (x_1, x_2, ..., x_N) \) represent the \( N \) data records belonging to a confidential sample. Each data vector \( x_j \) consists of \( k \) variables and is assumed to be a realization of a multivariate normal random vector. Assume \( x_1, x_2, ..., x_N \) are independent \( N_k(\mu, \Sigma) \) random vectors, where \( x_j = (x_{j1}, x_{j2}, ..., x_{jk}) \).

A microdata release is to be formed from the confidential sample and will contain \( m \) different records \( (m \leq N) \). Assume that the microdata release consists of the set of records \( (x_{n1}, x_{n2}, ..., x_{nm}) \). Before these \( m \) data vectors are released to the public, the \( k \) values from each of the records are masked with normally distributed measurement error. The released microdata file is denoted by

\[
\text{MD}_x = (x_{n1}, x_{n2}, ..., x_{nm})'
\]
where \( X'_{nj} = (X_{j1}, X_{j2}, \ldots, X_{jk}) \). Hence,

\[
X_{nj} = x_{nj} + u_{nj}, \quad j = 1, 2, \ldots, m
\]

(2.1.2)

where \( u_{nj} \) are independent \( N_k(0, \Sigma_{uu}) \) random vectors and \( u_{nj} \) is independent of \( x_{ni} \) for all \( i, j \). It follows that \( X_{n1}, X_{n2}, \ldots, X_{nm} \) are independent \( N_k(\mu, \Sigma_{xx} + \Sigma_{uu}) \) random vectors.

2.2. A General Approach

2.2.1. The confidentiality problem

A confidentiality problem arises when a record from an independent private data source, having an identification variable (e.g., name) is entirely known by an intruder. Suppose the record has \( l < k \) non-confidential variables in common with the records from the confidential sample. If the record is known to be contained in the confidential sample, the intruder can use statistical techniques in an attempt to match the target (i.e., the record from the private data source) to a record in the microdata release. Confidential information of the individual to whom the target record corresponds may then be exposed.

From the intruder’s perspective, the objective is to predict the values of the confidential variables of the target individual. If the target record does, in fact, correspond to a record in the microdata release, the information contained in the microdata record will provide the most information for the prediction of the confidential variables of the target record. The intruder’s immediate task is to determine the
probability that the j-th record in the microdata release corresponds to the target record. The intruder has available $x_0$ - true value of the target record and $(x_{n_1}, x_{n_2}, \ldots, x_{n_m})$ - set of released records.

Consider the following as an intruder's approach to this problem.

The variables in the target record are partitioned into two parts,

$$x_0 = \begin{pmatrix} x_{0,1} \\ x_{0,2} \end{pmatrix} \quad (2.2.1)$$

where $x_{0,1}(\ell \times 1)$ is known and $x_{0,2}$ is unknown. The records in the microdata release are partitioned in the same way,

$$x_{ni} = \begin{pmatrix} x_{n_i,1} \\ x_{n_i,2} \end{pmatrix}, \quad (2.2.2)$$

Also,

$$\begin{pmatrix} x_{n_i,1} \\ x_{n_i,2} \end{pmatrix} = \begin{pmatrix} x_{n_i,1} \\ x_{n_i,2} \end{pmatrix} + \begin{pmatrix} u_{n_i,1} \\ u_{n_i,2} \end{pmatrix} \quad (2.2.3)$$

where

$$\Sigma_{uu} = \begin{pmatrix} \Sigma_{uu11} & \Sigma_{uu12} \\ \Sigma_{uu21} & \Sigma_{uu22} \end{pmatrix} \quad (2.2.4)$$

and
\[ \Sigma_{xx} = \begin{pmatrix} \Sigma_{xx11} & \Sigma_{xx12} \\ \Sigma_{xx21} & \Sigma_{xx22} \end{pmatrix} \]  

(2.2.5)

are known positive definite matrices. For ease of notation, we assume \( \mu' = (\mu'_1, \mu'_2) = 0 \).

Assuming that the probability that \( \mathbf{x}_{n_j} \) corresponds to \( \mathbf{x}_0 \), prior to the release of the microdata, is \( m^{-1} \) for \( j = 1, 2, \ldots, m \), the conditional density of \( (\mathbf{x}_{n_1}, \mathbf{x}_{n_2}, \ldots, \mathbf{x}_{n_m}, \mathbf{x}_{0,1}) \) given that \( \mathbf{x}_{n_j} \) corresponds to the target is

\[
f[(\mathbf{x}_{n_1}, \mathbf{x}_{n_2}, \ldots, \mathbf{x}_{n_m}, \mathbf{x}_{0,1}) | \mathbf{x}_{n_j} = \mathbf{x}_0] \n = \left[ \frac{1}{2} \right] ^{mk+\ell} \left[ \frac{1}{2} \right] ^{-m/2} \left[ \frac{1}{2} \right] ^{-1/2} \n \times \exp\left(-\frac{1}{2} \left[ \sum_{t=1}^{m} \mathbf{x}_{n_t}' \Sigma_{xx}^{-1} \mathbf{x}_{n_t} + (\mathbf{x}_{0,1} - \mathbf{B} \mathbf{x}_{n_j})' \mathbf{A}^{-1} (\mathbf{x}_{0,1} - \mathbf{B} \mathbf{x}_{n_j}) \right] \right) \n \times \text{constant} \times \left[ \exp\left(-\frac{1}{2} \sum_{t=1}^{m} \mathbf{x}_{n_t}' \Sigma_{xx}^{-1} \mathbf{x}_{n_t} \right) \right] \gamma_{j0}, \tag{2.2.6} \]

where

\[
\mathbf{B} = \Sigma_{x_1x}' \Sigma_{xx}^{-1},
\]

\[
\Sigma_{x_1x} = E(\mathbf{x}_{n_j}, \mathbf{x}_{n_j}'),
\]
The quantity $\text{BX}^n_j$ is the conditional expectation of $x_{0,1} - x_{nj,1}$ given $x_n^j$, and $A$ is the variance of the conditional distribution of $x_{0,1}$ given $x_n^j$. Note that $x_0$ is treated as random in this derivation.

We now use $\gamma_{10}, \gamma_{20}, \ldots, \gamma_{m0}$, to define the conditional probabilities, $p_{10}, p_{20}, \ldots, p_{m0}$, that each record in the microdata release corresponds to the target record. The larger $\gamma_{j0}$ is relative to the $\gamma_{k0}$'s ($k \neq j$), the greater the conditional probability, conditional in $(x_{n1}, \ldots, x_{nm}, x_{0,1})$, that $x_n^j$ is the target record. Let

$$p_{j0} = \Pr(x_n^j = x_0 | (x_{n1}, x_{n2}, \ldots, x_{nm}, x_{0,1}))$$

$$= (\sum_{t=1}^{m} \gamma_{t0})^{-1} \gamma_{j0}, \text{ for } j = 1, 2, \ldots, m. \quad (2.2.9)$$

Thus, $p_{j0}$ is the conditional probability that the $j$-th record in the microdata release corresponds to the target record, given that the target record is contained in the released file and given
These probabilities are now used as weights in the construction of a predictor for the confidential variables of the target record.

2.2.2. Prediction of confidential variables

The ideal situation for the intruder attempting to predict the values of the confidential variables of the target record is to have $p_{k|0} = 1$ for some $k \in (1, 2, \ldots, m)$ and $p_{j|0} = 0$ for all other $j \in (1, 2, \ldots, m)$. In this case, record $x_{n_k}$ is the target record, where

$$x_{n_k} = x + u_{n_k}.$$

Partitioning $x_{n_k}$ into non-confidential and confidential sub-vectors gives

$$x_{n_k} = \begin{pmatrix} x_{n_k,1} \\ x_{n_k,2} \end{pmatrix} = \begin{pmatrix} x_{n_k,1} \\ x_{n_k,2} \end{pmatrix} + \begin{pmatrix} u_{n_k,1} \\ u_{n_k,2} \end{pmatrix}. \quad (2.2.10)$$

Since $x_{n_k}$ is the target, the problem of predicting $x_{0,2}$ is equivalent to predicting $x_{n_k,2}$. Assuming $x_0 = x_{n_k}$, the minimum mean square error predictor of $x_{n_k,2}$ is the mean of the conditional distribution, or the conditional expectation of $x_{n_k,2}$ given $(x_{n_k}, x_0, 1)$. Hence, the best predictor of $x_{n_k,2}$ when the $n_k$-th record is the target, is
\[
\hat{x}_{n_k,2} = E(x_{n_k,2}|(X_{n_k}, x_{0,1}) \text{ and } x_0 = x_{n_k})
\]

\[
= E((X_{n_k,2} - u_{n_k,2})|(X_{n_k}, x_{0,1}) \text{ and } x_0 = x_{n_k})
\]

\[
= X_{n_k,2} - E(u_{n_k,2}|(X_{n_k}, x_{0,1}) \text{ and } x_0 = x_{n_k})
\]

\[
= X_{n_k,2} - \hat{u}_{n_k,2}
\]  

(2.2.11)

where

\[
\hat{u}_{n_k,2} = \begin{pmatrix}
\Sigma_{uu12} \\
\Sigma_{uu22}
\end{pmatrix} \begin{pmatrix}
\Sigma_{xx11} + \Sigma_{uu11} & \Sigma_{xx12} + \Sigma_{uu12} & \Sigma_{xx11} \\
\Sigma_{xx21} + \Sigma_{uu21} & \Sigma_{xx22} + \Sigma_{uu22} & \Sigma_{xx21}
\end{pmatrix}^{-1} \begin{pmatrix}
x_{n_k,1} - \mu_1 \\
x_{n_k,2} - \mu_2 \\
x_{0,1} - \mu_1
\end{pmatrix}
\]

(2.2.12)

is the best predictor of \( u_{n_k,2} \). While it is well known that the conditional expectation is the best predictor, see Anderson (1984), we state the result for our problem.

**Theorem 2.2.1.** Assume the target record corresponds to data vector \( x_{n_k} \). Let the normal model hold, where

\[
X_{n_k} = \hat{x}_{n_k} + u_{n_k}
\]
\[
\begin{pmatrix}
X_{n_k,1} \\
X_{n_k,2}
\end{pmatrix} = \begin{pmatrix}
X_{n_k,1} \\
\mu_{k,2}
\end{pmatrix} + \begin{pmatrix}
\mu_{k,1} \\
\mu_{k,2}
\end{pmatrix}
\]

\begin{align*}
x_{n_k,1}, x_{n_k,2}, \ldots, x_{n_m} & \text{ are independent } N\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \Sigma_{xx11} & \Sigma_{xx12} \\ \Sigma_{xx21} & \Sigma_{xx22} \end{pmatrix}\right) \\
u_{n_k,1}, \nu_{n_k,2}, \ldots, \nu_{n_m} & \text{ are independent } N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \Sigma_{uu11} & \Sigma_{uu12} \\ \Sigma_{uu21} & \Sigma_{uu22} \end{pmatrix}\right)
\end{align*}

random vectors, and \(x\) is independent of \(\nu\) for all \(i, k \in \{1, 2, \ldots, m\}\). Then the minimum mean square error predictor of \(\nu_{n_k,2}\) based on \(X_{n_k}\) and \(x_{0,1}\) is given by (2.2.12).

**Proof.** Define \(Z_1 = \nu_{n_k,2}, Z_2 = [X_{n_k}, x_{0,1}], \hat{\nu}_{n_k,2} = E(Z_1|Z_2)\). Let \(f(Z_2)\) be any other predictor of \(Z_1\). To prove the result, we show \(\text{MSE}[f(Z_2)] - \text{MSE}[E(Z_1|Z_2)]\) is non-negative definite:

\[
\text{MSE}[f(Z_2)] = E([Z_1 - f(Z_2)][Z_1 - f(Z_2)']')
\]

\[
= \text{MSE}[E(Z_1|Z_2)] + E([E(Z_1|Z_2) - f(Z_2)][E(Z_1|Z_2) - f(Z_2)']')
\]

\[
+ E[W_1 W_2'] + E[W_2 W_1'],
\]

where \(W_1 = Z_1 - E(Z_1|Z_2)\) and \(W_2 = E(Z_1|Z_2) - f(Z_2)\).
\[
E(w_1, w_2) = E(Z_1[E(Z_1|Z_2)'] - Z_1[f(Z_2)'] - [E(Z_1|Z_2)][E(Z_1|Z_2)]' \\
+ [E(Z_1|Z_2)][f(Z_2)']
\]

\[
- E(E[Z_1[E(Z_1|Z_2)'] - Z_1[f(Z_2)'] - [E(Z_1|Z_2)][E(Z_1|Z_2)]' \\
+ [E(Z_1|Z_2)][f(Z_2)']|Z_2]
\]

\[
- E([E(Z_1|Z_2)][E(Z_1|Z_2)]' - [E(Z_1|Z_2)][f(Z_2)'] \\
- [E(Z_1|Z_2)][E(Z_1|Z_2)]' + [E(Z_1|Z_2)][f(Z_2)']')
\]

\[
= E(0) = 0.
\]

Hence,

\[
MSE[f(Z_2)] - MSE[E(Z_1|Z_2)] = E([E(Z_1|Z_2) - f(Z_2)][E(Z_1|Z_2) - f(Z_2)]')
\]

\[
\therefore MSE[f(Z_2)] - MSE[E(Z_1|Z_2)]
\]

is non-negative definite.

For each \(X_{nj}\) in the microdata release, the predictor of \(X_{0,2}\) assuming \(X_{nj}\) is the target record is
\[ \hat{x}_{n_j,2} = x_{n_j,2} - \mathbb{E}[\hat{u}_{n_j,2} | (X_{n_k}, x_{0,1})] \]

Hence,

\[
\text{MSE}(\hat{x}_{n_j,2} | x_{n_j} - x_0) = \mathbb{E}[(\hat{x}_{n_j,2} - x_{n_j,2})(\hat{x}_{n_j,2} - x_{n_j,2})'] | x_{n_j} - x_0
\]

\[
= \mathbb{E}[(\hat{u}_{n_j,2} - u_{n_j,2})(\hat{u}_{n_j,2} - u_{n_j,2})'] | x_{n_j} - x_0
\]

\[
= \text{MSE}(\hat{u}_{n_j,2} | x_{n_j} - x_0)
\]

Therefore, assuming \( x_{n_j} \) is the target record, \( \hat{x}_{n_j,2} \) is the minimum mean square error predictor of \( x_{0,2} \).

The situation with \( p_{k0} = 1 \) for some \( k \) will rarely occur in a real data situation. It will never occur if the agent masking the data is proficient. Although some instances may arise when one \( p_{j0} \) dominates the others, in most cases a small subset of the \( p_{j0} \)'s will dominate all others. For example, if \( p_{1,0} \neq p_{12,0} \neq p_{35,0} \neq 0.30 \) and \( p_{j0} < 0.001 \) for all other \( j \in \{1, 2, \ldots, m\} \), a good predictor of \( x_{0,2} \) would be dominated by \( X_{n_1}, X_{n_{12}}, \) and \( X_{n_{35}} \). If the \( p_{j0} \)'s are nearly equal, information from all records in the microdata release should enter into the prediction of \( x_{0,2} \). We now construct such a predictor.
The knowledge base for prediction of \( x_{0,2} \) is \( \Sigma_{uu}, \Sigma_{xx}, x_{0,1} \), and the records in the microdata release. The probability that \( x_{n_j} \) is the target, \( x_0 \), given \( (x_{n_1}, \ldots, x_{n_m}, x_{0,1}) \) is

\[
\Pr(x_{n_j} = x_0 | (x_{n_1}, \ldots, x_{n_m}, x_{0,1})) = p_{j0},
\]

\[
= \left( \sum_{t=1}^{m} \gamma_{t0} \right)^{-1} \gamma_{j0}, \text{ for } j = 1, 2, \ldots, m
\]

where \( \gamma_{j0} \) is defined in (2.2.8).

Under the assumption that the target record, \( x_0 \), corresponds to some record in the microdata release, we know that \( x_0 = x_{n_\ell} \) for some \( \ell \in \{1, 2, \ldots, m\} \). Hence, \( x_{0,2} \) has an unconditional normal distribution with mean \( 0 \) and variance \( \Sigma_{xx22} \). The conditional distribution of \( x_{0,2} \) given \( (x_{n_1}, \ldots, x_{n_m}, x_{0,1}) \) and \( x_{n_j} = x_0 \) is

\[
f[x_{0,2} | (x_{n_1}, \ldots, x_{n_m}, x_{0,1}) \text{ and } x_{n_j} = x_0]
\]

\[
= f[x_{0,2} | (x_{n_j}, x_{0,1}) \text{ and } x_{n_j} = x_0]
\]

\[
= \frac{1}{(2\pi)^{(k-\ell)/2}} |\Sigma_{\epsilon\epsilon}|^{-1/2} \exp\left(-\frac{1}{2} [x_{0,2} - G(x_{n_j}', x_{0,1}')]' \left[\Sigma_{\epsilon\epsilon}^{-1} [x_{0,2} - G(x_{n_j}', x_{0,1}')] \right] \right)
\]

\[
\times \Sigma_{\epsilon\epsilon}^{-1} [x_{0,2} - G(x_{n_j}', x_{0,1}')] \tag{2.2.14}
\]

where
\[
C = \begin{pmatrix}
\Sigma_{xx12} & \Sigma_{xx11} & \Sigma_{xx13} \\
\Sigma_{xx21} & \Sigma_{xx22} & \Sigma_{xx23} \\
\Sigma_{xx31} & \Sigma_{xx32} & \Sigma_{xx33}
\end{pmatrix}^{-1}
\]  
(2.2.15)

\[
\Sigma_{\epsilon\epsilon} = \Sigma_{xx22} - C(\Sigma_{xx12}, \Sigma_{xx22}, \Sigma_{xx12})'.
\]  
(2.2.16)

Hence, the conditional distribution of \(x_{0,2}\) given \((X_1, X_2, \ldots, X_n, x_{0,1})\) and given that \(x_0\) corresponds to some record in the microdata release, is

\[
f[x_{0,2} | (X_1, \ldots, X_n, x_{0,1}) \text{ and } x_0 = x_{n_j} \text{ for some } j \in \{1, 2, \ldots, m\}] = \sum_{j=1}^{m} \frac{1}{\Sigma_{\epsilon\epsilon}} \sum_{j=1}^{m} \sum_{j=1}^{m} \exp\left(-\frac{1}{2} \left[x_{0,2} - C(X_{n_j}', x_{0,1}')\right]^2\right) \\
\times \Sigma_{\epsilon\epsilon}^{-1}[x_{0,2} - C(X_{n_j}', x_{0,1}')]
\]  
(2.2.17)

where \(C\) and \(\Sigma_{\epsilon\epsilon}\) are defined in (2.2.15) and (2.2.16), respectively.

The best predictor of \(x_{0,2}\) is then the mean of the conditional distribution of \(x_{0,2}\), given \((X_1, \ldots, X_n, x_{0,1})\),
\[ \hat{x}_{0,2} = \sum_{j=1}^{m} \Pr(x_{n_j} = x_0 | (x_{n_1}, \ldots, x_{n_m}, x_{0,1})) \times E(x_{0,2} | (x_{n_j}, x_{0,1}) \text{ and } x_j = x_0) \]

\[ = \sum_{j=1}^{m} p_{j0} C(x'_{n_j}, x'_{0,1})' \]  

(2.2.18)

If we let \( J = (1,2,\ldots,m) \), the variance of the predictor error can be expressed as

\[ V(\hat{x}_{0,2} - x_{0,2}) | (x_{n_1}, \ldots, x_{n_m}, x_{0,1}) \text{ and } x_{n_j} = x_0 \text{ for some } j \in J \]

\[ = E_J(V(\hat{x}_{0,2} - x_{0,2}) | (x_{n_1}, \ldots, x_{n_m}, x_{0,1}) \text{ and } x_{n_j} = x_0) \]

\[ + V_J(E(\hat{x}_{0,2} - x_{0,2}) | (x_{n_1}, \ldots, x_{n_m}, x_{0,1}) \text{ and } x_{n_j} = x_0) \]

(2.2.19)

where \( E_J(*) \) denotes the expectation over all possible matches conditional on \((x_{n_1}, \ldots, x_{n_m}, x_{0,1})\), and \( V_J \) is the corresponding variance. We know that

\[ V(\hat{x}_{0,2} - x_{0,2}) | (x_{n_1}, \ldots, x_{n_m}, x_{0,1}) \text{ and } x_{n_j} = x_0 \] \[ = \Sigma_{\epsilon \epsilon} \]

and

\[ E(\hat{x}_{0,2} - x_{0,2}) | (x_{n_1}, \ldots, x_{n_m}, x_{0,1}) \text{ and } x_{n_j} = x_0 \] \[ = C(x'_{n_j}, x'_{0,1})' \]
where $\Sigma_{\epsilon \epsilon}$ and $C$ are defined in (2.2.16) and (2.2.15), respectively. Since $\Sigma_{\epsilon \epsilon}$ is a constant,

$$E_J(\Sigma_{\epsilon \epsilon}) = \sum_{j=1}^{m} p_j \Sigma_{\epsilon \epsilon}$$

$$= \Sigma_{\epsilon \epsilon} \quad (2.2.20)$$

Let

$$W_j = (X'_{n_j}, x'_0, 1)C' \quad (2.2.21)$$

for $j=1, 2, \ldots, m$. Thus,

$$V_J((X'_{n_j}, x'_0, 1)C') = V_J(W_j)$$

$$= \sum_{j=1}^{m} p_j (W_j - \bar{W})' (W_j - \bar{W})$$

$$= \sum_{j=1}^{m} p_j W_j - (\sum_{j=1}^{m} p_j W_j)' (\sum_{j=1}^{m} p_j W_j) \quad (2.2.22)$$

where $\bar{W} = \sum_{j=1}^{m} p_j W_j$. Therefore, from (2.2.20) and (2.2.22), we see that

$$V((X_{n_1}, \ldots, X_{n_m}, x'_0, 1) \text{ and } X_{n_j} = x'_0 \text{ for some } j \in J)$$

$$= \Sigma_{\epsilon \epsilon} + \sum_{j=1}^{m} p_j W_j - (\sum_{j=1}^{m} p_j W_j)' (\sum_{j=1}^{m} p_j W_j) \quad (2.2.23)$$
We now derive the mean square error of the best predictor of \( x_{0,2} \) using another approach. The error in the predictor (2.2.18) is

\[
\hat{x}_{0,2} - x_{0,2} = \sum_{j=1}^{m} p_{j0} \hat{x}_{n_j,2} - x_{0,2},
\]

where

\[
\hat{x}_{n_j,2} = G(X'_{n_j}, x'_0, 1)' \quad (2.2.24)
\]

is the predictor constructed under the assumption that \( X'_{n_j} \) is the masked value associated with the target record, \( x_0 \). If \( x_{n_j} \) is the record which matches \( x_0 \), then the prediction error is

\[
(\hat{x}_{0,2} - x_{0,2})_l = p_{l0}[G(X'_{n_l}, x'_0, 1)' - x_{0,2}] + \sum_{j=1}^{m} p_{j0}[G(X'_{n_j}, x'_0, 1)' - x_{0,2}], \quad (2.2.25)
\]

Under the condition that \( x_0 \) matches \( x_{n_l} \), let

\[
x_{n_l,2} = x_{0,2} = G(X'_{n_l}, x'_0, 1)' + \epsilon_{n_l},
\]

where \( \epsilon_{n_l} \) is independent of \( G(X'_{n_l}, x'_0, 1)' \). Then

\[
(\hat{x}_{0,2} - x_{0,2})_l = (p_{l0} - 1)G(X'_{n_l}, x'_0, 1)' - \epsilon_{n_l} + \sum_{j=1}^{m} p_{j0}G(X'_{n_j}, x'_0, 1)' + \sum_{j=1, j \neq l}^{m} p_{j0}G(X'_{n_j}, x'_0, 1)'
\]
Note that \( x_{0,2} - x_{n_j,2} \) is independent of \( x_{n_j} \) for \( j \neq \ell \). Also, \( C(x'_{n_j}, x'_{0,1})' \) is the best predictor of \( x_{0,2} \) conditional on the match of \( x_{n_j} \) to \( x_0 \). Then

\[
E((x_{0,2} - x_{0,2})'(x_{0,2} - x_{0,2})| (x_{n_1}, \ldots, x_{n_m}, x_{0,1}) \text{ and } x_{n_j} = x_0)
\]

\[
= -\sum_{\ell} W_{\ell}^T W_{\ell} - W_{\ell}^T \sum_{j=1}^m p_{j0} W_j - (\sum_{j=1}^m p_{j0} W_j)' W_{\ell} + (\sum_{j=1}^m p_{j0} W_j)' (\sum_{j=1}^m p_{j0} W_j),
\]

\[
(2.2.27)
\]

where \( W_j \) is defined in (2.2.21). If we weight this quantity by \( p_{j0} \), the probability that \( x_{n_j} \) matches \( x_0 \), and sum over all possible \( \ell \), we obtain

\[
E((\hat{x}_{0,2} - \hat{x}_{0,2})'(\hat{x}_{0,2} - \hat{x}_{0,2})| (x_{n_1}, \ldots, x_{n_m}, x_{0,1}))
\]

\[
= -\sum_{\ell} p_{j0} W_{\ell}^T W_{\ell} - (\sum_{j=1}^m p_{j0} W_j)' (\sum_{j=1}^m p_{j0} W_j),
\]

\[
(2.2.28)
\]

which agrees with (2.2.23). We state the best predictor and the associated variance as a result.

**Theorem 2.2.2.** Let the normal model hold, where

\[
x_{n_j} = x_{n_j} + u_{n_j}
\]
\[
\begin{pmatrix}
\mathbf{x}_{n_j,1} \\
\mathbf{x}_{n_j,2}
\end{pmatrix}
= \begin{pmatrix}
\mathbf{u}_{n_j,1} \\
\mathbf{u}_{n_j,2}
\end{pmatrix}
+ \begin{pmatrix}
\mathbf{x}_{n_j,1} \\
\mathbf{x}_{n_j,2}
\end{pmatrix},
\]

\[
\mathbf{x}_{n_j} \sim \text{NI} \left( \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \Sigma_{xx11} & \Sigma_{xx12} \\ \Sigma_{xx21} & \Sigma_{xx22} \end{pmatrix} \right),
\]

\[
\mathbf{u}_{n_j} \sim \text{NI} \left( \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \Sigma_{uu11} & \Sigma_{uu12} \\ \Sigma_{uu21} & \Sigma_{uu22} \end{pmatrix} \right),
\]

and \( \mathbf{x}_{n_j} \) is independent of \( \mathbf{u}_{n_k} \) for all \( j, k \in \{1, 2, \ldots, m\} \). Also, assume the target record, \( \mathbf{x}_0 \), corresponds to one of the data vectors in the microdata release, \( \{\mathbf{x}_{n_1}, \mathbf{x}_{n_2}, \ldots, \mathbf{x}_{n_m}\} \). Then, the minimum mean square error predictor of \( \mathbf{x}_{0,2} \) is given by

\[
\hat{\mathbf{x}}_{0,2} = \sum_{j=1}^{m} P_{j0} \mathbf{c} (\mathbf{x}_{n_j}^{'}, \mathbf{x}_{0,1}^{'}),
\]

where \( P_{j0} \) is defined in (2.2.9) and \( \mathbf{c} \) is defined in (2.2.15).

Furthermore, the mean square error of \( \hat{\mathbf{x}}_{0,2} \) conditional on \( \{\mathbf{x}_{n_1}, \ldots, \mathbf{x}_{n_m}, \mathbf{x}_{0,1}\} \), is

\[
E( (\hat{\mathbf{x}}_{0,2} - \mathbf{x}_{0,2}) (\mathbf{x}_{0,2} - \mathbf{x}_{0,2}) | \{\mathbf{x}_{n_1}, \ldots, \mathbf{x}_{n_m}, \mathbf{x}_{0,1}\} )
\]

\[
= \Sigma \epsilon \epsilon + \sum_{l=1}^{m} P_{l0} \mathbf{w}_l \mathbf{w}_l - \sum_{j=1}^{m} P_{j0} \mathbf{w}_j (\Sigma \mathbf{p}_j \mathbf{w}_j) (\Sigma \mathbf{p}_j \mathbf{w}_j),
\]
where $\Sigma_{\epsilon\epsilon}$ is the conditional variance of $x_{n_2,2}$ given $(X_{n_2,1}^\prime, x_{0,1}^\prime)$ and $x_{n_2} = x_0$, and $W_{\ell} = (X_{n_2}^\prime, x_{0,1}^\prime)\Sigma'$. 

2.3. Preserving Confidentiality in Microdata Releases

Given the model situation and prediction problem of the previous section, the best predictor, from an intruder's standpoint, of a target record's confidential variables is shown to be (2.2.18). Assuming the partial target record, obtained by the intruder from an independent source, is contained in the microdata release, the accuracy of the intruder's predictor is highly dependent on the match probability (2.2.9) assigned to the full target record in the microdata release. If a low match probability is associated with the target record in the microdata release, the intruder's best predictor will be less accurate. From a data provider's standpoint, a low match probability is a favorable situation for protecting an individual's confidentiality. Hence, the data provider should make every attempt to minimize the target record's match probability in the microdata release. An approach to this minimization problem in the multivariate case is now presented.

2.3.1. The multivariate model

We investigate the problem of selecting an error covariance matrix to use in masking normal data.

Consider the general case of $\ell$ non-confidential variables. As before, an intruder obtains a data record of non-confidential variables, $x_{0,1}^\prime(\ell \times 1)$, from a private source. The microdata release consists of
X_{n_1}', ..., X'_{n_m} \text{ with }

\begin{align*}
X'_{n_j} - (X'_{n_j,1}, X'_{n_j,2}) &= (x'_{n_j,1}, x'_{n_j,2}) + (u'_{n_j,1}, u'_{n_j,2}) \\
&= \text{(2.3.1)}
\end{align*}


for \( j=1, 2, ..., m \).

Certainly a large error variance will lower the probability of matching a record. At the same time, though, adding such error will excessively distort the data. The data provider must balance the objectives of providing a file that resembles the original data as closely as possible and also provides reasonable confidentiality protection for the respondents. It is always possible to transform the vectors so that

\[ X_{n_j} \sim N_k(0, \delta I_k). \]

We then assume, in this section, that the covariance matrix is \( \delta I_k \) and that a decision has been made to fix the ratio of error variance to total variance at \( (1 + \delta)^{-1} \) for all variables. Hence,

\[ u_{n_j} \sim N_k(0, \Sigma_{uu}) \]

where
\[
\Sigma_{uu} = \begin{pmatrix}
\Sigma_{uu11} & \Sigma_{uu12} \\
\Sigma_{uu21} & \Sigma_{uu22}
\end{pmatrix} = \begin{pmatrix}
1 & \rho_{12} & \rho_{13} & \cdots & \rho_{1k} \\
\rho_{12} & 1 & \rho_{23} & \cdots & \rho_{2k} \\
\rho_{13} & \rho_{23} & 1 & \cdots & \rho_{3k} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\rho_{1k} & \rho_{2k} & \rho_{3k} & \cdots & 1
\end{pmatrix}, \quad (2.3.2)
\]

\[|\rho_{ij}| < 1 \text{ for } i < j, \ i, j \in \{1, 2, \ldots, k\}, \text{ and } \Sigma_{uu} \text{ is positive definite. As before,}
\]

\[
\Sigma_{xx} = \Sigma_{xx} + \Sigma_{uu}
\]

(2.3.3)

where \( \Sigma_{uu} \) and \( \Sigma_{xx} \) are partitioned as in (2.2.4) and (2.2.5). We proceed to determine the optimal structure of \( \Sigma_{uu} \) from the standpoint of the data provider.

2.3.2. Choice of correlation error

Recall that the intruder computes the predictor defined in (2.2.18) as

\[
\hat{x}_{0,2} = \sum_{j=1}^{m} \gamma_{j0} \left( \hat{\Sigma} \gamma_{t0} \right)^{-1} \hat{x}_{n_j,2}
\]

where \( \gamma_{t0} \) and \( \hat{x}_{n_j,2} \) are defined in (2.2.8, 2.2.13).

Suppose \( x_{0,1} \) corresponds to the \( k \)-th record in the microdata release. A high probability for \( p_{kk} \) means a more accurate predictor
of the confidential variables. Hence, the data provider would like small \( p_{kk} \) for all \( k \) to cause uncertainty in the mind of the intruder. Because \( p_{kk} = \gamma_{kk}(\Sigma_{t=1}^{m} \gamma_{tk})^{-1} \), minimizing \( p_{kk} \) is closely related to minimizing the log odds ratio, \( \ln(p_{kk}^{-1}p_{jk}) \), for a randomly chosen element \( j \) and fixed \( k \). We consider the problem of choosing the correlations in \( \Sigma_{uu} \) to minimize the expected value of the log odds ratio, \( E(\ln(p_{kk}) - \ln(p_{jk})) \). The data dependence is removed by considering the expectation. We show that in order to minimize \( E(\ln(p_{kk}) - \ln(p_{jk})) \), the data provider should add vectors of error having a covariance matrix equal to a multiple of \( \Sigma_{xx} \), the covariance matrix of the unmasked data vectors. That is, in this special case where \( \Sigma_{xx} = \delta I_k \), we will show that the optimal \( \Sigma_{uu} \) is the identity matrix, \( I_k \). We first give some results that are used to prove the main theorem.

**Theorem 2.3.1.** Assume the microdata release consists of

\[ X_1, X_2, \ldots, X_m \]

where \( X_j \) satisfies (2.1.2) for \( j = 1, 2, \ldots, m \). Also, for all \( j \),

\[ x_{nj} \sim N_k(0, \Sigma_{xx}) \]

\[ u_{nj} \sim N_k(0, \Sigma_{uu}) \]

where \( \Sigma_{xx} = \delta I_k \) and \( \Sigma_{uu} \) is given by (2.3.2). Assume the target record, \( x_0 \), corresponds to the \( k \)-th record in the microdata release.
Then minimizing $E(\ln(p_{kk}) - \ln(p_{jk}))$ (j ≠ k), where $p_{jk}$ is defined in (2.2.7), is equivalent to minimizing $\text{tr}(A^{-1})$, where

$$A = \Sigma_{x X_1} - \Sigma_{x X_1} \Sigma_{x X_1}^{-1}$$ \hspace{1cm} (2.3.4)

and

$$\Sigma_{x X_1} = E(X_n x'_n, 1) .$$

Proof.

$$E(\ln(p_{kk}) - \ln(p_{jk}))$$

$$= E(\ln(\exp[-\frac{1}{2}(x_{n_k}'1 - B x_{n_k})'A^{-1}(x_{n_k}'1 - B x_{n_k})$$

$$+ \frac{1}{2}(x_{n_j}'1 - B x_{n_j})'A^{-1}(x_{n_j}'1 - B x_{n_j}))])$$

where

$$B = \Sigma_{x X_1} \Sigma_{x X_1}^{-1} .$$

Hence,
\[
E(\ln(p_{kk}) - \ln(p_{jk})) \quad \\
\quad = E(-\frac{1}{2} [ (x_{n_k}' - BX_{n_k})' A^{-1} (x_{n_k}' - BX_{n_k}) \\
- (x_{n_j}' - BX_{n_j})' A^{-1} (x_{n_j}' - BX_{n_j}) ]) \\
\quad = -\frac{1}{2} \text{tr}(E[(x_{n_k}' - BX_{n_k})(x_{n_k}' - BX_{n_k})'])A^{-1} \\
+ \frac{1}{2} \text{tr}(E[(x_{n_j}' - BX_{n_j})(x_{n_j}' - BX_{n_j})'])A^{-1} \\
\quad = -\frac{1}{2} \text{tr}(AA^{-1}) + \frac{1}{2} \text{tr}(E(x_{n_k}' x_{n_k}')A^{-1}) \\
\quad - \frac{1}{2} \text{tr}(E(x_{n_j} x_{n_j}')A^{-1}B) - \frac{1}{2} \text{tr}(E(x_{n_j}' x_{n_j}')B'A^{-1}) \\
\quad + \frac{1}{2} \text{tr}(E(x_{n_j} x_{n_j}')B'A^{-1}B) .
\]

Since \( x_{n_k} \) and \( x_{n_j} \) are independent, the cross product terms have expectation 0. We are left with

\[
E(\ln(p_{kk}) - \ln(p_{jk})) = -\ell/2 + \frac{1}{2} \text{tr}(\Sigma_{xx11}^{-1}) + \frac{1}{2} \text{tr}(\Sigma_{x1x'XXx1}^{-1}) \\
\quad = -\ell/2 + \text{tr}(\Sigma_{xx11}^{-1}) \\
\quad - \frac{1}{2} \text{tr}((\Sigma_{xx11} - \Sigma_{x1XXXx1})A^{-1})
\]
where $\delta$ and $\ell$ are fixed constants.

We have shown that minimizing $E(ln(p_{kk}) - ln(p_{jk}))$ is equivalent to minimizing $\text{tr}(A^{-1})$. Using additional matrix algebra to investigate $\text{tr}(A^{-1})$, we now simplify the problem. From (2.3.4),

$$A = \Sigma_{xx11} - \Sigma_{x_1xXXxx_1}^{-1} \Sigma_{xx11}$$

$$= \Sigma_{xx11}^{1/2} [I_{\ell} - \Sigma_{xx11}^{1/2} (\Sigma_{x_1xXXxx_1}^{-1} \Sigma_{xx11}^{1/2}) \Sigma_{xx11}^{1/2}]$$

where

$$\Sigma_{xx11} = \delta I_{\ell},$$

$$\Sigma_{x_1X} = (\delta I_{\ell}, 0).$$

Hence,

$$A = \delta[I_{\ell} - \delta^{-1/2} (\delta I_{\ell}, 0) \Sigma_{xx11}^{-1} (\delta I_{\ell}, 0)'] \delta^{-1/2}$$

$$= \delta[I_{\ell} - \delta (\Sigma_{xx11}^{1/2} - \Sigma_{xx12}^{1/2} \Sigma_{xx22} \Sigma_{xx21})^{-1} \Sigma_{xx11}^{1/2}]$$

(2.3.5)
and

$$
\delta A^{-1} = \delta (\delta^{-1} [I - E^{-1}]^{-1})
$$

$$
= [I - E^{-1}]^{-1}
$$

(2.3.6)

where

$$
E = \delta^{-1} (\Sigma_{XX1} - \Sigma_{XX2} \Sigma_{XX21}^{-1} \Sigma_{XX21}).
$$

(2.3.7)

To this point, we have shown that minimizing the expectation of the log-odds ratio of \( p_{jk} \) to \( p_{ik} \) with respect to the correlations \( (\rho_{ij}; i < j; i, j \in \{1, 2, \ldots, m\}) \) between elements of each error vector, is equivalent to minimizing \( \text{tr}([I - E^{-1}]^{-1}) \) where \( E \) is defined in (2.3.7). Let \( \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_r \) be the characteristic roots of \( (I - E^{-1}) \). Each \( \lambda_i \) satisfies the determinantal equation

$$
|\langle I - E^{-1} \rangle - \lambda I | = 0.
$$

(2.3.8)

We see that

$$
|\langle I - E^{-1} \rangle - \lambda I | = |(1 - \lambda) I - E^{-1}|
$$

$$
= |E^{-1/2} |(1 - \lambda)E - I | |E^{-1/2}|
$$

$$
= (1 - \lambda)^2 |E^{-1/2} |^2 |E - (1 - \lambda)^{-1} I|
$$

(2.3.9)
By (2.3.8) and (2.3.9), we have

\[ |E - (1 - \lambda)^{-1}I_2| = 0. \]

If we define the roots of \( E \) to be \( \alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_\ell \), then

\[ \alpha_i = (1 - \lambda_i)^{-1} \quad \text{and} \quad \lambda_i = \alpha_i^{-1}(\alpha_i - 1), \text{ for } i = 1, 2, \ldots, \ell. \]

Note also that, because \( E \) and \((I - E^{-1})\) are positive definite matrices,

\( \lambda_i \in (0, 1) \) and \( \alpha_i > 1 \) for all \( i \). So, minimizing

\[ \text{tr}(E^{-1}) = \sum_{i=1}^{\ell} \lambda_i^{-1} \]

is the same as minimizing \( \Sigma_{i=1}^{\ell} (\alpha_i - 1)^{-1} \alpha_i \), where \( \alpha_1, \alpha_2, \ldots, \alpha_\ell \) are the characteristic roots of \( E \). Furthermore,

\[ \sum_{i=1}^{\ell} (\alpha_i - 1)^{-1} \alpha_i = \sum_{i=1}^{\ell} (\alpha_i - 1)^{-1}(\alpha_i - 1 + 1) \]

\[ = \ell - \sum_{i=1}^{\ell} (\alpha_i - 1)^{-1}. \]

Hence, our problem of minimizing \( \Sigma_{i=1}^{\ell} (\alpha_i - 1)^{-1} \) with respect to the correlations,

\( (\rho_{ij}: i < j, i, j \in \{1, 2, \ldots, k\}) \), between the elements of each error vector. We proceed to show the minimum of \( \Sigma_{i=1}^{\ell} (\alpha_i - 1)^{-1} \) is attained when \( \rho_{ij} = 0 \) for all \( i < j \leq \ell \). We begin by demonstrating that the roots of \( E \) are individually maximized when \( \Sigma_{XX21} = 0 \).
Lemma 2.3.1. Assume $S$ and $T$ are real, symmetric $(n \times n)$ matrices. Define the characteristic roots of $S$ and $T$ to be $\nu_1(S) \geq \ldots \geq \nu_n(S)$ and $\nu_1(T) \geq \ldots \geq \nu_n(T)$, respectively. Then,

$$\nu_j(S) + \nu_n(T) \leq \nu_j(S + T) \text{ for } j = 1, 2, \ldots, n.$$ 

Proof. See Bhatia (1987, p.34).

Theorem 2.3.2. Let

$$\Sigma_{XX} = \begin{pmatrix}
\Sigma_{XX11} & \Sigma_{XX12} \\
\Sigma_{XX21} & \Sigma_{XX22}
\end{pmatrix}$$

be a real, symmetric $(k \times k)$ positive definite matrix, with $\Sigma_{XX11}$ having dimension $(\ell \times \ell)$. Let $\beta_1 \geq \beta_2 \geq \ldots \geq \beta_\ell$ and $\gamma_1 \geq \gamma_2 \geq \ldots \geq \gamma_\ell$ be the roots of $\Sigma_{XX11}$ and $(\Sigma_{XX11} - \Sigma_{XX12} \Sigma_{XX22}^{-1} \Sigma_{XX21})$, respectively. Then

$$\beta_i \geq \gamma_i \text{ for } i = 1, 2, \ldots, \ell.$$ 

Proof. In Lemma 2.3.1, let

$$T = \Sigma_{XX12} \Sigma_{XX22}^{-1} \Sigma_{XX21}$$

and

$$S = \Sigma_{XX11} - T.$$
Hence, $S + T = \Sigma_{XX11}$. Since $S$ and $S + T$ are positive definite matrices, their roots are positive. Also, the roots of $T$ are greater than or equal to zero because $T$ is non-negative definite. By Lemma 2.3.1, we have

$$\gamma_i + \epsilon \leq \beta_i, \quad i = 1, 2, \ldots, \ell,$$

where $\epsilon$ is the smallest root of $T$ and $\epsilon \geq 0$. Hence,

$$\gamma_i \leq \beta_i, \quad i = 1, 2, \ldots, \ell.$$

Therefore, by Theorem 2.3.2, the roots of

$$E = \delta^{-1}(\Sigma_{XX11} - \Sigma_{XX12} \Sigma_{XX22}^{-1} \Sigma_{XX21})$$

are individually maximized when $\Sigma_{XX21} = 0$ or when $E = \delta^{-1} \Sigma_{XX11}$.

We have now reduced the problem to minimizing $\Sigma_{i=1}^{\ell} (\alpha_i - 1)^{-1}$ where $\alpha_1 \geq \ldots \geq \alpha_\ell > 1$ are now the roots of

$$\delta^{-1} \Sigma_{XX11} = \delta^{-1} (\delta I_\ell + \Sigma_{u11})$$

$$= I_\ell + \delta^{-1} \Sigma_{u11}$$

with $\delta > 0$ and $\Sigma_{u11}$ having the structure of a correlation matrix. Since the roots of $\delta^{-1} \Sigma_{XX11}$ are equal to the roots of $\delta^{-1} \Sigma_{u11}$ each increased by one, our objective is to minimize
or, equivalently,
\[ \ell \sum_{i=1}^{\ell} \phi_i^{-1} = \text{tr}(\Sigma_{uull}^{-1}) \]

where \( \phi_1 \geq \phi_2 \geq \ldots \geq \phi_\ell \) are the roots of \( \Sigma_{uull} \). We show that the minimum is obtained when all correlations are set to zero.

**Theorem 2.3.3.** Assume \( \Sigma_{uull} \) is a symmetric non-singular matrix of the form given in (2.3.6). The minimum of \( \text{tr}(\Sigma_{uull}^{-1}) \) is attained when \( \Sigma_{uull} = I_\ell \).

**Proof.** There exist matrices \( Q \) and \( \Phi \) such that \( \Sigma_{uull} = QQ' \) where \( QQ' = I_\ell \) and \( \Phi = \text{diag}(\phi_1, \phi_2, \ldots, \phi_\ell) \) with \( \phi_i > 0 \) for \( i=1, 2, \ldots, \ell \). Also, \( \text{tr}(\Sigma_{uull}^{-1}) = \sum_{i=1}^{\ell} \phi_i = \ell \). Hence,

\[ \Sigma_{uull}^{-1} = (QQ')^{-1} = \Phi^{-1}Q' \]

and

\[ \text{tr}(\Sigma_{uull}^{-1}) = \sum_{i=1}^{\ell} \phi_i^{-1} \]

We use the Lagrangian multiplier procedure to minimize \( \sum_{i=1}^{\ell} \phi_i^{-1} \) subject to the restrictions that \( \sum_{i=1}^{\ell} \phi_i = \ell \) and \( \phi_i > 0 \) for all \( i \). The restriction \( \sum_{i=1}^{\ell} \phi_i = \ell \) is a weaker restriction than the original restriction that the diagonal elements of \( \Sigma_{uull} \) are equal to one.
Therefore, if the minimum under the weaker condition satisfies the stronger restriction, we will have bound the minimum under the stronger condition. Define \( \phi_i = z_i^2 \) (\( i, 2, \ldots, \ell \)) where \((z_1, z_2, \ldots, z_\ell) \in \mathbb{R}^\ell \). Then the Lagrangian equation is

\[
L = \sum_{i=1}^{\ell} z_i^{-2} - \nu \left[ \left( \sum_{i=1}^{\ell} z_i^2 \right) - \ell \right]
\]

where \( \nu \) is the Lagrange multiplier. Differentiation with respect to \( z_i \) (\( i=1, 2, \ldots, \ell \)) and \( \nu \) yields

\[
z_i^4 = -\nu^{-1} \quad (i=1, 2, \ldots, \ell),
\]

\[
\sum_{i=1}^{\ell} z_i^2 = \ell.
\]

Therefore,

\[
z_i^2 = (-\nu)^{-1/2}
\]

and

\[
\sum_{i=1}^{\ell} z_i^2 = \sum_{i=1}^{\ell} (-\nu)^{-1/2} = \ell.
\]

Hence,

\[
\nu = -1 \text{ and } z_i^2 = 1 \text{ (or } \phi_i = 1 \text{) for all } i.
\]

The point \((\phi_1, \phi_2, \ldots, \phi_\ell) = (1, 1, \ldots, 1)\) in positive \( \ell \)-space is a candidate for the point where \( f(\phi_1, \phi_2, \ldots, \phi_\ell) = \sum_{i=1}^{\ell} \phi_i^{-1} \) has a
minimum or maximum. However, since \( f(\phi_1, \ldots, \phi_n) \) is bounded below by zero and is unbounded above, the vector \((\phi_1, \ldots, \phi_n) = (1, \ldots, 1)\) is the point where \( f \) attains its minimum value.

The result that \( \text{tr}(\Sigma_{u11}^{-1}) \) is minimized when \( \Sigma_{u11} = I_2 \) follows since \( \Sigma_{u11}^{-1} = QI_2Q' = I_2 \).

We state the entire result.

**Theorem 2.3.4.** Assume the normal model,

\[
X_{nj} = x_{nj} + u_{nj},
\]

\[
x_{nj} = \begin{pmatrix} x_{nj,1} \\ x_{nj,2} \end{pmatrix} = \begin{pmatrix} \delta I_\ell \\ 0 \end{pmatrix} - NI \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} x_{nj,1} \\ x_{nj,2} \end{pmatrix} - NI \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \delta I_\ell \\ 0 \end{pmatrix},
\]

\[
u_{nj} = \begin{pmatrix} u_{nj,1} \\ u_{nj,2} \end{pmatrix} = \begin{pmatrix} \Sigma_{u11} \\ \Sigma_{u12} \end{pmatrix},
\]

where \( x_{nj} \) is independent of \( u_{nj} \) for all \( i, j \in \{1, 2, \ldots, m\} \), \( \delta > 0 \) and \( \Sigma_{uu} \) is a symmetric positive definite matrix having the form given in (2.3.2). Let

\[
P_{jk} = \left( \Sigma \gamma_{tk} \right)^{-1} \gamma_{jk},
\]
where $\gamma_{jk}$ is defined in (2.2.8). Then $E(\ln(p_{kk}) - \ln(p_{jk}))$ is minimized when $\Sigma_{u11} = I_2$ and $\Sigma_{u12} = 0$.

We have shown that when the ratio of error variance to total variance fixed at $(1 + \delta)^{-1}$, the correct match probability, $p_{kk}$, is minimized on the average when $\Sigma_{u11} = \delta I_2$. Therefore, when creating a microdata file, there is a sense in which a data provider affords respondents maximum protection against disclosure by adding error vectors which have a covariance matrix equal to a multiple of the covariance matrix of the original data vectors.
3. DESCRIPTION OF THE MASKING ALGORITHM

In this chapter, we present an algorithm for creating a masked data set from an original data set of vectors comprised of responses to continuous, discrete and classification variables. One objective of the masking algorithm is to create a data set that is statistically representative of the original set of data records in the following ways. First of all, each of the variables in the masked data set should have the same first and second moments as variables in the original data set. For a classification variable, the sample categorical proportions should be the same for the two data sets. Secondly, the correlation structures of the original and masked data sets should be nearly identical. Finally, the corresponding univariate distribution functions of the original and masked data sets should also be the same.

The other major objective of the masking algorithm is to sufficiently disguise the data records to afford the respondents protection against disclosure. The algorithm is designed to mask the data records in such a way that an intruder will have difficulty in matching a target record from an outside source to a record in the masked data file. Equivalently, the released data set is such that the variance of the intruder's prediction of an unknown attribute for a target record will be large.

To mask a data set, the algorithm first transforms the original data to normal variates. The normal variates are masked and then back transformed to the original scales of the input variables. We now outline the procedure.
3.1. Transformation of Variables

The algorithm which we describe masks continuous, discrete and classification variables. Responses to continuous and discrete variables are assumed to be quantitative, whereas a response to a classification variable takes on a nominal categorical value. In order to mask the input data set, all data values are first transformed to normal variates. The masking algorithm is designed to transform only quantitative variables (i.e., continuous or discrete) and Bernoulli variables. Hence, we must first transform each classification variable into a set of Bernoulli variables.

3.1.1. Construction of Bernoulli variables from classification variables

We let \( X_1, X_2, \ldots, X_n \) be the responses to a variable, \( X \), having categories \( \{C_1, C_2, \ldots, C_r\} \). For each response, \( X_t \), \( t=1, 2, \ldots, n \), we define the dummy variables \( Z_{t1}, Z_{t2}, \ldots, Z_{tr-1} \) as

\[
Z_{tj} = \begin{cases} 
1 & \text{if } X_t = C_j \\
0 & \text{otherwise} 
\end{cases}
\]  

(3.1.1)

In addition, we let \( \phi_j \) be the probability that an element is in category \( j \) given that it is not in the first \( j-1 \) categories. Hence, we define

\[
\phi_1 = \Pr(C_1),
\]
\[ \phi_j = P_x(C_j | \text{not } C_1, C_2, \ldots, C_{j-1}), \quad j=2, 3, \ldots, r-1. \quad (3.1.2) \]

We further define the pseudo Bernoulli variables, \( W_{t1}, W_{t2}, \ldots, W_{t,r-1} \), as

\[ W_{t1} = Z_{t1}, \]

\[ W_{tj} = Z_{tj} \quad \text{if } Z_{ti} = 0 \text{ for } i<j \quad (3.1.3) \]

\[ \begin{align*}
1 \text{ with probability } \phi_j \\
0 \text{ with probability } (1 - \phi_j)
\end{align*} \quad \text{otherwise} \]

The \( W_{tj} \) variables, \( j=1, 2, \ldots, r-1 \), are uncorrelated and go into the original data set to be masked as Bernoulli variables.

To mask the quantitative and Bernoulli variables, all observations are transformed to standard normal variates using the sample univariate distribution functions. The quantitative variables and the Bernoulli variables are transformed by two slightly different algorithms.

3.1.2. Transformation of quantitative variables to normality

We begin with the algorithm for quantitative variables, describing the procedure for a single variable, \( X \). First, the sample distribution function, \( \hat{F}_X \), is formed from the \( n \) observations on \( X \). The function \( \hat{F}_X \) will be a step function having \( m \) jumps, where \( m (\leq n) \) is the number of distinct values of the variable \( X \). Using
and a linear interpolation technique, a continuous distribution function, \( \hat{F}_X \), is constructed as follows. The values \( X(1), X(2), \ldots, X(m) \) are defined to be the \( m \) ordered, distinct values of the variable \( X \). Also, \( Y(1), Y(2), \ldots, Y(m) \) are defined by

\[
Y(j) = \hat{F}_X(X(j)) \quad \text{for} \quad j = 1, 2, \ldots, m.
\]

The constructed \( \hat{F}_X \) consists of \( m \) connected line segments where each segment is formed by joining the midpoints of the horizontal lines of adjacent "steps" in \( \hat{F}_X \). See Figure 3.1 for an illustration. The points which define the domain of each segment of \( \hat{F}_X \) are denoted by \( x(i) \), \( i = 0, 1, \ldots, m \). We define \( x(i) \) to be the point equidistant from the \( i \)-th and \((i+1)\)-th ordered distinct values of \( X \). Hence,

![Graphical example of linear interpolation technique](image)
\[ x(i) = \frac{1}{2} (X(1) + X(i+1)) , \ i=0, 1, 2, \ldots, m , \quad (3.1.4) \]

where

\[ X(0) = X(1) - (X(2) - X(1)) , \]

\[ X(m+1) = X(m) + (X(m) - X(m-1)) . \]

Table 3.1 contains the sets of values \((X(0), X(1), \ldots, X(m+1))\), \((Y(1), Y(2), \ldots, Y(m))\), and \((x(1), x(2), \ldots, x(m))\) corresponding to the illustration in Figure 3.1. Using point-slope algebra, we define \(\bar{F}_X(z)\) as

<table>
<thead>
<tr>
<th>i</th>
<th>(X(i))</th>
<th>(Y(i))</th>
<th>(x(i))</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.3</td>
<td>0</td>
<td>0.85</td>
</tr>
<tr>
<td>1</td>
<td>1.4</td>
<td>0.1</td>
<td>1.95</td>
</tr>
<tr>
<td>2</td>
<td>2.5</td>
<td>0.2</td>
<td>2.7</td>
</tr>
<tr>
<td>3</td>
<td>2.9</td>
<td>0.3</td>
<td>3.2</td>
</tr>
<tr>
<td>4</td>
<td>3.5</td>
<td>0.4</td>
<td>3.9</td>
</tr>
<tr>
<td>5</td>
<td>4.3</td>
<td>0.5</td>
<td>4.9</td>
</tr>
<tr>
<td>6</td>
<td>5.5</td>
<td>0.6</td>
<td>5.85</td>
</tr>
<tr>
<td>7</td>
<td>6.2</td>
<td>0.7</td>
<td>6.55</td>
</tr>
<tr>
<td>8</td>
<td>6.9</td>
<td>0.8</td>
<td>7.65</td>
</tr>
<tr>
<td>9</td>
<td>8.4</td>
<td>0.9</td>
<td>8.8</td>
</tr>
<tr>
<td>10</td>
<td>9.2</td>
<td>1</td>
<td>9.6</td>
</tr>
<tr>
<td>11</td>
<td>10.0</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
\[ F_X(z) = Y(0) + (X(i+1) - X(i))^{-1}(Y(i+1) - Y(i))z \]

for \( z \in (x(i), X(i+1)] \), \hspace{1cm} (3.1.5)

where \( Y(0) = 0 \) and \( Y(m) = 1 \).

The function \( F_X(z) \) is evaluated at \( X(1), X(2), \ldots, X(m) \) and the resultant values are denoted by \( p_1, p_2, \ldots, p_m \), where \( p_i \in (0, 1) \) for all \( i=1, 2, \ldots, m \). Standard normal values for each \( X \) are obtained by applying the inverse of the standard normal distribution function to \( p_i, i=1, 2, \ldots, m \). Hence, we define the standard normal values corresponding to \( X(1), X(2), \ldots, X(m) \) as

\[ Z(i) = \Phi^{-1}(p_i), \text{ for } i=1, 2, \ldots, m, \] \hspace{1cm} (3.1.6)

where \( \Phi(*) \) is the standard normal distribution function. We define \((Z_1, Z_2, \ldots, Z_n)\) to be the vector of standard normal values corresponding to the vector of original values, \((X_1, X_2, \ldots, X_n)\), where

\[ Z_i = Z(j) \text{ if } X_i = X(j), \] \hspace{1cm} (3.1.7)

for \( i=1, 2, \ldots, n \), and \( j=1, 2, \ldots, m \). Note that each \( X_i \) can equal only one \( X(j) \), so \( Z_i \) is uniquely defined for all \( i \).

If the original variables are standard normally distributed, the transformed variables are very close to the original variables. Also if
two variables are jointly normal, the correlation of the transformed variables is nearly equal to that of the original variables.

3.1.3. **Transformation of Bernoulli variables to normality**

In transforming Bernoulli variables to normal variables, one objective is for the resulting normal variables to have the same correlation structure as the original set of Bernoulli variables. In addition, we desire the cross correlation structure between the transformed Bernoulli variables and the transformed quantitative variables to be the same as the cross correlation structure between the original Bernoulli variables and original quantitative variables.

Consider the following construction of normal variables from Bernoulli variables. Let each original observation vector be given by \( X_t = (X_{ct}, X_{dt})' \) where \( X_{ct} \) and \( X_{dt} \) denote, respectively, the quantitative and Bernoulli subvectors of \( X_t \). Let \( \Sigma_{XX} \), the sample covariance matrix of the \( X_t \)'s, be analogously partitioned as

\[
\Sigma_{XX} = \begin{pmatrix}
\Sigma_{cc} & \Sigma_{cd} \\
\Sigma_{dc} & \Sigma_{dd}
\end{pmatrix} \quad (3.1.8)
\]

Let

\[
P_{XX} = \begin{pmatrix}
P_{cc} & P_{cd} \\
P_{dc} & P_{dd}
\end{pmatrix} \quad (3.1.9)
\]

be the sample correlation matrix of the \( X_t \)'s. Furthermore, let
be the estimated conditional covariance matrix of the Bernoulli variables, given the quantitative variables, and let

\[ B_{d,c} = m_{cc}^{-1} \]  

be the matrix of regression coefficients for the regression of the Bernoulli variables on the quantitative variables.

The transformed quantitative observation vectors are denoted by \( Z_1, Z_2, \ldots, Z_n \). If the original quantitative \( X_{ct} \) are approximately normal, then the \( Z_{ct} \) are approximately normal \((0, \Sigma_{cc})\). Let

\[ f_{d,ct} = B'_{d,c} \Sigma_{cc}^{-1} Z_{ct} + m_{dd,cc}^{1/2} e_{d,ct}, \]  

where \( \Sigma_{cc} = \text{diag}(m_{c,11}, \ldots, m_{c,s}) \), \( e_{d,ct} \) is a \( N(0, I) \) random vector, and \( s \) is the number of quantitative variables. Then, \( f_{d,ct} \) is approximately distributed as a normal random vector with mean 0 and variance matrix \( m_{dd} \). Also, the matrix of correlations between \( f_{d,ct} \) and \( Z_{ct} \) is approximately \( \Sigma_{cd} \). Hence, \( (Z_{ct}', f_{d,ct}) \) is approximately normal and has nearly the same correlation structure as the original data. However, \( f_{d,ct} \) does not depend on the true values of the Bernoulli variables. Only through the cross correlation with the
transformed continuous variables does \( f_{d,ct} \) contain information about
the true Bernoulli values.

To make the \( f_{d,ct} \)'s dependent on the \( X_{dt} \)'s, we further
transform the \( f_{d,ct} \)'s. First each \( f_{d,ct} \) is converted to a vector of
uniform random variables, \( \varepsilon_{d,ct} \), using the standard normal
distribution function. We define the elements of \( \varepsilon_{d,ct} \) as

\[
\varepsilon_{d,ctj} = \Phi(f_{d,ctj}), \quad j=1, 2, \ldots, p, \quad (3.1.13)
\]

where \( \Phi(\cdot) \) is the standard normal distribution function. We let \( p_{0j} \)
be the mean of the \( j \)-th Bernoulli variable. For each Bernoulli
variable, we divide the interval \((0, 1)\) into two parts. The first part
is the interval \((0, 1 - p_{0j})\) and the second part is the interval \([1 - \]
\( p_{0j}, 1)\). Let \( h_{d,ctj} \) be a uniform random vector with

\[
h_{d,ctj} = \begin{cases} 
\varepsilon_{d,ctj}(1 - p_{0j}) & \text{if } X_{dtj} = 0 \\
1 - p_{0j}(1 - \varepsilon_{d,ctj}) & \text{if } X_{dtj} = 1
\end{cases} \quad (3.1.14)
\]

for \( j=1, 2, \ldots, p \), where \( X_{dtj} \) is the original Bernoulli variable
response. The values \( h_{d,ct1}, h_{d,ct2}, \ldots, h_{d,ctn} \) are then assigned
the ranks \( R_1, R_2, \ldots, R_n \), where \( h_{d,ctj} \) is the \( R_j \)-th smallest
value. These ranks are standardized

\[
\tilde{R}_j = n^{-1}(R_j - 0.5), \quad (3.1.15)
\]
and the standard normal values associated with the responses to the t-th Bernoulli variable are defined as

\[ Z_{dtj} = \Phi^{-1}(\hat{R}_j), \quad j=1, 2, \ldots, n. \] (3.1.16)

Performing this transformation on all Bernoulli variables creates a set of transformed variables with correlation structure similar to that of the original variables.

The transformed data consist of \( Z_1, Z_2, \ldots, Z_n \) where \( Z_t = (Z_{ct}, Z_{dt}) \). Each \( Z_{tj} \) is normally distributed with mean zero. The covariance matrix of \( Z_t \) is \( \Sigma_{ZZ} \), where \( \Sigma_{ZZ} \) is "close" to the correlation matrix of the original observations. The data are not necessarily jointly normal.

3.2. Masking the Normal Data Vectors

The masked set of transformed data vectors are computed by adding a normally distributed error vector to each transformed vector of normal observations. Each error vector has mean zero and a covariance matrix approximately equal to a multiple of the covariance matrix of the \( Z_t \)'s. Hence, the masked data are

\[ \tilde{Z}_t = Z_t + \tilde{u}_t, \quad t=1, 2, \ldots, n \] (3.2.1)

where \( \tilde{u}_t \) is a vector of random variables with mean 0 and covariance matrix approximately equal to a multiple of \( \Sigma_{ZZ} \), where
Each $\mathbf{u}_t$ is a function of $\mathbf{u}_t$, a normal random vector having mean zero and an identity covariance matrix. The construction of the $\mathbf{u}_t$ from the $\mathbf{u}_t$ consists of several steps, which we discuss below.

Suppose we let $\mathbf{u}_t = (u_{t1}, \ldots, u_{tp}) \sim N(0, I_p)$ and generate $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$. The expected value of the off-diagonal elements in the matrix of sums of squares and cross products is zero. When using a normal random number generator, the observed covariances are not zero due to random variation. We now present an algorithm which reduces the random variation and insures nearly zero sample correlations.

3.2.1. Construction of error vectors

We wish to construct $n$ normally distributed vectors having mean zero and sample covariance matrix equal or "close" to the identity matrix. The first $[n/2]$ of the observations are generated with no restrictions. For $j=1, 2, \ldots, [n/2]$, let $\mathbf{u}_j$ be randomly generated normal vectors with mean 0 and covariance matrix $I$. The sum of the elements and the sums of squares and products matrix are computed for these observations. Thereafter, each observation vector is created as follows. A random normal vector is generated and denoted by $\mathbf{u}_t = (u_{t1}, \ldots, u_{tp})$. To initiate the procedure, set $j = 1$ and $t = [n/2] + 1$. The signs of the elements of $\mathbf{u}_t$ are assigned as follows. The sign of $u_{tj}$ is the negative of the sign of the $j$-th element in the vector of sums, specifically the negative of the sign.
of $\sum_{j=1}^{t-1} u_j^T u_j$. For $k \neq l$, the sign of $u_{tk}$ is equal to the sign of the $k$-th element in the $l$-th row of $\sum_{j=1}^{t-1} u_j^T u_j$, multiplied by the negative of the sign of $u_{tl}$. Thus, if $\sum_{j=1}^{t-1} u_j^T u_j < 0$ and $u_{tl}$ is positive, then $u_{tk}$ is positive. The value of $l$ is increased by one (mod $p$), $t$ is reset to $t+1$ and the next observation vector is generated. The procedure is repeated until a full set of error vectors has been generated.

3.2.2. The initial data mask and the distance criterion

Given the transformed observations $Z_t$ and a vector of errors $u_t$, we are ready to perform the initial mask of the transformed data. The error vectors, $u_1, \ldots, u_n$, are approximately independent normally distributed random vectors with mean zero and covariance matrix approximately equal to the identity. For each $u_t$, we define $\hat{u}_t$ as

$$\hat{u}_t = \alpha^{1/2} u_t T_{zz}$$

where $T_{zz} = P_{zz}$, $P_{zz}$ is the correlation matrix of the transformed data, and $\alpha > 0$. The value of $\alpha$ is specified by the user. Then the initially masked data set is

$$\tilde{Z}_t = Z_t + \hat{u}_t = Z_t + \alpha^{1/2} u_t T_{zz}, \quad t=1, 2, \ldots, n.$$  

To insure that the transformed data vectors are adequately masked, we then perform a distance check on the initially masked data. For each
$Z_t$, let the statistical distance between $Z_t$ and $\hat{Z}_j$ be denoted by $d_{tj}$, for $j=1, 2, \ldots, n$. Let $d_{tj}(1)$ be the smallest of the $n$ distances and let $d_{tj}(2)$ be the second smallest of the $n$ distances. If the distance between $Z_t$ and $\hat{Z}_t$, $d_{tt}$, is not one of the two smallest distances in the set $(d_{tj}: j=1, 2, \ldots, n)$, the vector $\hat{Z}_t$ is assumed to be adequately masked. If $d_{tt}$ is one of the two smallest of the $n$ distances, the vector $\hat{Z}_t$ is declared to be inadequately masked. Then, to remask the $t$-th observation, the magnitude of the error vector, $u_t$, is increased. One of two methods is used, depending on the current magnitude of the error vector, $u_t$. To determine the appropriate method, we first compare $d_{tt}$ to $C_0$, where

$$C_0 = p - (2p)^{1/2} + \epsilon,$$

$\epsilon > 0$, and $p$ is the dimension of the vector. The value $C_0$ is approximately one standard deviation less than the mean of a chi-square random variable with $p$ degrees of freedom. If $d_{tt}$ exceeds $C_0$, the associated error vector, $u_t$, is assumed to need a minor increase in magnitude. In such a case, a new error vector is created by multiplying $u_t$ by a random multiple such that the expected value of the distance between $Z_t$ and $\hat{Z}_t$ is slightly greater than the second largest distance between $Z_t$ and $\hat{Z}_j$, $j \neq t$.

If $d_{tt} < C_0$, the distance between $Z_t$ and $\hat{Z}_t$ is small because of the small magnitude of $u_t$. In this case, a new error vector, $u_t$, is generated. The elements of the new vector must meet certain absolute size restrictions. An acceptable randomly generated error vector, $u_t = (u_{t1}, \ldots, u_{tp})$, must satisfy $|u_{tj}| > \epsilon$ for all $j$, where $\epsilon$
is a pre-specified constant. Also $\sum_{j=5(i-1)}^{5i} u^2_j$ must exceed $\eta$, for $i=1, 2, \ldots, \lfloor p/5 \rfloor$, where $\eta$ is also a constant specified in the program. Furthermore, if $t > \lfloor n/2 \rfloor$, the signs of the elements of the newly generated $u_t$ are determined by the procedure described in Section 3.2.1. A new $\hat{u}_t$ is computed from the newly generated $u_t$ and the masked vector $\hat{z}_t$ is also recomputed. The distance check is again performed, and the process is repeated until the masked vector satisfies the distance requirement. The masked vectors are now denoted by $\hat{z}_t$. The procedure only makes one remasking pass through the data. It would be possible to modify the program to make additional passes.

3.3. Iterations to Attain Desired Correlations

After the pass through the data to modify the error for the distance criterion, the masked Z-data are back transformed into masked X-data. Then an iterative procedure is used to improve the agreement between the correlation structure of the transformed variables and the correlation structure of the original variables. The error terms are adjusted in an attempt to achieve nearly identical correlations between the masked and original data. We now provide some details of this iterative process.

3.3.1. Back transforming the data to original scale

Let $\hat{Z} = (\hat{z}_1, \hat{z}_2, \ldots, \hat{z}_p)$ be the matrix of masked, transformed data, where
\( \mathbf{\tilde{Z}}_j = (\mathbf{\tilde{z}}_{1j}, \mathbf{\tilde{z}}_{2j}, \ldots, \mathbf{\tilde{z}}_{nj})' \) \hspace{1cm} (3.3.1)

is the vector containing the masked observations for the j-th transformed variable. To begin the back transformation, consider each vector of n observations, \( \mathbf{\tilde{Z}}_j \), separately. First, we define \( \mathbf{\tilde{R}}_j \) to be the vector of ranks of the n observations of \( \mathbf{\tilde{Z}}_j \), with the rank "n" being assigned to the largest value. We create a normalized version of the elements of \( \mathbf{\tilde{u}}_j \) by multiplying each \( \mathbf{\tilde{u}}_{tj} \) by
\[
[(n-1)^{-1}\sum_{t=1}^{n}(u_{tj})^2]^{1/2}.
\]
The normalized vector form of \( \mathbf{\tilde{u}}_j \) is denoted by \( \mathbf{\tilde{u}}_j^+ \). Further, we let \( \mathbf{\tilde{D}}_j \) be the vector of normalized and adjusted ranks of \( \mathbf{\tilde{Z}}_j \). The t-th element of \( \mathbf{\tilde{D}}_j \) is defined to be
\[
\mathbf{\tilde{D}}_{tj} = n^{-1}[\mathbf{\tilde{R}}_{tj} + \psi(u_{tj}^+)], \text{ for } t=1, 2, \ldots, n, \hspace{1cm} (3.3.2)
\]

where \( \mathbf{\tilde{R}}_{tj} \) is the rank of the t-th observation of the j-th variable and \( \psi \) is a continuous function mapping \( u_{tj}^+ \) into \((-1, 0)\). The \( \psi(u_{tj}^+) \) values are perturbations to keep the \( \mathbf{\tilde{D}}_{tj} \) values from being the simple ranks multiplied by \( n^{-1} \). After these computations are performed for all p variables, we have
\[
\mathbf{\tilde{D}} = (\mathbf{\tilde{D}}_1, \mathbf{\tilde{D}}_2, \ldots, \mathbf{\tilde{D}}_p), \hspace{1cm} (3.3.3)
\]

where \( \mathbf{\tilde{D}}_j = (\mathbf{\tilde{D}}_{1j}, \mathbf{\tilde{D}}_{2j}, \ldots, \mathbf{\tilde{D}}_{nj})' \) and \( D_{tj} \epsilon (0, 1) \) for \( t=1, 2, \ldots, n \), and \( j=1, 2, \ldots, p \).
The next step in the procedure is the transformation of the $D_{tj}$ values to $X_{tj}$ values on the original scale. The final back transformation differs depending on whether a variable is quantitative or Bernoulli. We begin with the transformation for quantitative variables.

We consider the $j$-th quantitative variable, $X_j$, recalling the method of transforming the original data values of the $j$-th variable to normal values (see Section 3.1.2). The transformation of $D_{tj}$ to $X_{tj}$ is performed by applying the inverse of $F_{X_j}$ to $D_{tj}$. Thus, $X_{tj}$ is

$$X_{tj} = x(i-1) + (Y(i) - Y(i-1))^{-1}(D_{tj} - Y(i-1))(x(i) - x(i-1))$$

for $D_{tj} \in (Y(i-1), Y(i)]$ (3.3.4)

where $x(i)$ and $Y(i)$ are defined in Section 3.1.2.

To convert the transformed normal-Bernoulli variables back to original scale, the procedure is more straightforward. Let $P_{0k}$ be the mean of the $k$-th original Bernoulli variable. The masked Bernoulli value for $X_{tk}$ is

$$X_{tk} = \begin{cases} 
0 & \text{if } D_{tk} \in (0, 1 - P_{0k}) \\
1 & \text{if } D_{tk} \in [1 - P_{0k}, 1) 
\end{cases} \quad \text{for } t=1, 2, \ldots, n. \quad \text{(3.3.5)}$$

Once all variables have been transformed, masked, and back-transformed to original scale, the original and masked data matrices are
concatenated to form the \( n \times 2p \) matrix

\[
\bar{X} = (X, \hat{X}) = (X_1, \ldots, X_p, \hat{X}_1, \ldots, \hat{X}_p),
\]

(3.3.6)

where

\[
X_j = (x_{1j}, x_{2j}, \ldots, x_{nj})',
\]

\[
\hat{X}_j = (\hat{x}_{1j}, \hat{x}_{2j}, \ldots, \hat{x}_{nj})',
\]

for \( j = 1, 2, \ldots, p \). The sample correlation matrix of \( \bar{X} \) is computed and denoted by \( P_{\bar{X} \bar{X}} \). This matrix consists of the correlation matrix of the original data, \( P_{XX} \), the correlation matrix of the masked data, \( P_{\hat{X}\hat{X}} \), and the cross correlation matrix between the original and masked data, \( P_{XX}^* \). Hence, we have

\[
P_{\bar{X}\bar{X}} = \begin{pmatrix} P_{XX} & P_{\hat{X}\hat{X}} \\ P_{XX}^* & P_{XX}^* \end{pmatrix}.
\]

(3.3.7)

At this point, two groups of iterations are performed. The purpose of the first group of iterations is to bring all of the diagonal elements of \( P_{XX}^* \) close to a robust average of the initial cross correlations, \( \rho_{XX}^{*} \), \( j = 1, 2, \ldots, p \). The second group of iterations concentrates on adjusting the error vectors to make \( P_{XX}^{*} \) nearly identical to \( P_{XX} \). We begin by discussing the initial group of iterations.
3.3.2. The iterations for cross correlations

The target correlation for each \( \rho_{xx_{jj}} \) is

\[
\bar{\rho} = (p-2)^{-1} \left[ \sum_{j=1}^{P} \rho_{xx_{jj}} - \left( \max_{1 \leq j \leq p} \rho_{xx_{jj}} + \min_{1 \leq j \leq p} \rho_{xx_{jj}} \right) \right].
\] (3.3.8)

The target, \( \bar{\rho} \), is computed and permanently defined using the cross correlations between corresponding variables in the initial mask of the data. We discuss the general idea behind reducing or increasing a cross correlation.

The correlation between the \( j \)-th original and \( j \)-th masked variable is

\[
\rho_{xx_{jj}} = \left( m_{xx_{jj}} \right)^{-1/2} m_{xx_{jj}}^{**},
\] (3.3.9)

where

\[
m_{xx} = (n-1)^{-1} \sum_{t=1}^{n} (x_t - \bar{x})(x_t - \bar{x}),
\]

\[
m^{**}_{xx} = (n-1)^{-1} \sum_{t=1}^{n} (x_t^{**} - \bar{x}^{**})(x_t^{**} - \bar{x}^{**}),
\]

\[
m_{xx}^{**} = (n-1)^{-1} \sum_{t=1}^{n} (\bar{x}_t^{**} - \bar{x})(\bar{x}_t^{**} - \bar{x}).
\] (3.3.10)

We can write

\[
t_{ij} = \rho_{xx_{jj}} x_{ij} + \eta_{ij},
\] (3.3.11)
where $X_{tj}$ and $\eta_{tj}$ have variances $\sigma_{XX_{tj}}$ and $\sigma_{\eta_{tj}}$, respectively, and the covariance between $X_{tj}$ and $\eta_{tj}$ is zero. Hence,

$$\sigma_{\eta_{tj}} = \sigma_{XX_{tj}} (1 - \rho_{XX_{tj}}^2) . \quad (3.3.12)$$

If $\sigma_{\eta_{tj}}$ is decreased, an increase in $\rho_{XX_{tj}}^* \rho_{XX_{tj}}$ will result. Increasing the variance of the error has the opposite effect. Hence, depending on whether a cross-correlation is less than or greater than $\bar{\rho}$, we can decrease or increase the variance component of the corresponding element in the error. The amount by which we adjust each variance component is determined by a simple approximation technique. The adjustment is made to the error in the $\tilde{Z}_t$ and then the variables are transformed to the X-scale.

We define $B_{aa}$ to be a diagonal transformation matrix, with $B_{aajj}$, the $j$-th diagonal element of $B_{aa}$, defined by

$$B_{aajj} = [1 - \frac{1}{2} (\bar{\rho} + \rho_{XX_{tj}}^*)]^{-1} (1 - \bar{\rho}) , \quad (3.3.13)$$

for $j=1, 2, \ldots, p$. The new error vectors are $\tilde{u}_{tB_{aa}}$, for $t=1, 2, \ldots, n$, and the new masked transformed vectors are

$$\tilde{\tilde{Z}}_t = \tilde{Z}_t + \tilde{u}_{tB_{aa}} , \quad t=1, 2, \ldots, n . \quad (3.3.14)$$

Finally, the new $\tilde{\tilde{Z}}_t$ vectors are transformed back to the original scale. This iterative process is repeated a prespecified number of
times or until the observed cross correlations differ from the desired
cross correlations by less than a specified amount.

3.3.3. Iterations to adjust correlation structure of masked variables

After the iterative procedure to adjust the cross correlations is
completed, a second iterative process is performed in an attempt to make
the off-diagonal correlations in $\mathbf{P}_{XX}^{*}$ nearly identical to the
 correspondings correlations in $\mathbf{P}_{XX}$. The idea, again, is to modify the
error vectors to obtain the desired correlations. The error is modified
for one variable at a time. The variable $X_j$ for which the difference
$\sum_{k=1}^{p} (r_{XXjk} - r_{XXjk}^{**})^2$ is the largest is chosen for modification. We
denote the chosen variable by $X_1$ and define $\kappa$ to be the average
correlation between $X_1$ and the remaining $(p-1) X_j$ variables, $j=2,
...,$ $p$. Though we base the choice of the variable to be modified on
correlations computed for data on the original scale, it is the
transformed normal data, $\bar{Z}$, that is modified. The objective is to
create a linear combination of the variables $Z_1, \bar{Z}_1, \bar{Z}_2, \ldots, \bar{Z}_p$ that,
when back transformed to $\bar{X}_1$, give the correct correlation with $X_1$
and correct correlations with $\bar{X}_2, \bar{X}_3, \ldots, \bar{X}_p$.

The new transformed variable is

$$\bar{H}_1 = b_0 Z_1 + b_1 \bar{Z}_1 + b_2 \bar{Z}_2 + \ldots + b_p \bar{Z}_p$$  (3.3.15)

where $b_0 = 1 - b_1$. The system of equations defining the desired
properties is
Corr(\*{G}_1, X_1) = \kappa

\[ \text{Corr}(\*{G}_1, \*{X}_j) = \rho_{XXj1}, \ j=2, 3, \ldots, p, \] (3.3.16)

where \*{G}_1 is the back transformed variable corresponding to \*{X}_1.

Therefore, the system is

\[
\begin{pmatrix}
1 & \rho_{XX11} & \rho_{XX12} & \ldots & \rho_{XX1p} \\
\rho_{XX21} & \rho_{XX21} & \rho_{XX22} & \ldots & \rho_{XX2p} \\
\rho_{XX31} & \rho_{XX31} & \rho_{XX32} & \ldots & \rho_{XX3p} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\rho_{XXp1} & \rho_{XXp1} & \rho_{XXp2} & \ldots & 1
\end{pmatrix}
\begin{pmatrix}
b_1 \\
b_2 \\
b_3 \\
\vdots \\
b_p
\end{pmatrix}
= \begin{pmatrix}
1-b_1 \\
b_1 \\
b_2 \\
\vdots \\
b_p
\end{pmatrix}
\begin{pmatrix}
\kappa \\
\rho_{XX21} \\
\rho_{XX31} \\
\vdots \\
\rho_{XXp1}
\end{pmatrix}
\] (3.3.17)

The system can also be written

\[
\begin{pmatrix}
\rho_{XX11} - 1 & \rho_{XX12} & \ldots & \rho_{XX1p} \\
\rho_{XX21} - \rho_{XX21} & 1 & \ldots & \rho_{XX2p} \\
\rho_{XX31} - \rho_{XX31} & \rho_{XX32} & \ldots & \rho_{XX3p} \\
\vdots & \vdots & \ddots & \vdots \\
\rho_{XXp1} - \rho_{XXp1} & \rho_{XXp2} & \ldots & 1
\end{pmatrix}
\begin{pmatrix}
b_1 \\
b_2 \\
b_3 \\
\vdots \\
b_p
\end{pmatrix}
= \begin{pmatrix}
\kappa - 1 \\
\rho_{XX21} - \rho_{XX21} \\
\rho_{XX31} - \rho_{XX31} \\
\vdots \\
\rho_{XXp1} - \rho_{XXp1}
\end{pmatrix}
\] (3.3.18)
If we write the system as $Ab = c$, we solve by computing $b = A^+c$ where $A^+$ is the Moore-Penrose inverse of $A$. The vector $b$ is then used to create the linear combination

$$
\mathbf{\hat{a}}_1 = (Z_1, \hat{Z}_1, \hat{Z}_2, \ldots, \hat{Z}_p)(1-b_1, b_1, \ldots, b_p)' ,
$$

(3.3.19)

where the variances of $Z_1$ and of $\hat{Z}_j$, $j=1, 2, \ldots, p$, are unity. The $\mathbf{\hat{a}}_1$ is then standardized and back transformed into $\hat{a}_1$ or, equivalently, the new $\hat{x}_1$.

The iterative procedure to adjust the correlations is performed a pre-determined number of times or until a convergence criterion for the off-diagonal correlations is satisfied.

3.4. Back Transforming Classification Variables

At this point, the only remaining computation is to convert the sets of Bernoulli variables, created for the purpose of masking the classification variables, back to their categorical ranges. To mask the classification variable $X$ having $r$ categories, the Bernoulli variables $W_{t1}, \ldots, W_{t,r-1}$ are created for each response, $X_t$ (see Section 3.1.1). These responses to the Bernoulli variables are masked and denoted by $\hat{w}_{t1}, \hat{w}_{t2}, \ldots, \hat{w}_{t,r-1}$. To determine masked categorical values, $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n$, we first define $\hat{x}_{tj}$ as
\[ \hat{x}_{tj} = \begin{cases} \hat{x}_{tj} & \text{for } j=1 \\ [1 - \sum_{i=1}^{j-1} \hat{x}_{tj}] \hat{x}_{tj} & \text{for } j=2, 3, \ldots, r-1 \end{cases} \]  

(3.4.1)

Then \( \hat{x}_t \) is defined as

\[ \hat{x}_t = C_j \text{ if } \hat{x}_{tj} = 1, \ t=1, 2, \ldots, n, \]  

(3.4.2)

where \( C_j \) is the \( j \)-th category of the variable \( X \).

Upon obtaining the masked values for all categorical variables, the masking is complete.
4. RESULTS OF MASKING TEST DATA

In this chapter, we use the algorithm described in chapter 3 to mask two computer generated data sets. We begin by reviewing some of the previous work on the analytic usefulness and confidentiality aspects of masked data records. We then briefly discuss the implementation of our masking algorithm and describe the steps a user must follow in executing the programmed routine. We conclude by discussing the masking of the two data sets. For the first data set, we investigate the statistical usefulness of the masked data set by comparing statistical analyses performed on the original and masked data sets. We also examine the effectiveness of the mask with regard to protection against disclosure. For the second data set, we concentrate on evaluating the mask of discrete and classification variables, investigating statistical similarities and differences between the original and masked data sets within subgroups of the classification variable. Finally, we examine the joint distributions of the variables before and after the mask.

4.1. Introduction

4.1.1. Other data masking experiments involving adding error

Some work has been done investigating the effects of masking data records by adding random error. The articles discussed here are also summarized in the literature review of chapter one. In this section we provide more detail of the portion of each article which concentrated on the effectiveness of masking data by adding random error.
Spruill (1982) proposed measures of confidentiality based on a minimum distance matching criterion. In quantifying the amount of protection provided by a data mask, Spruill proposed attempting to match masked records to their true analogues in the original data file. This involved computing a distance between a target record in the microdata file and each record in the original file. The target is then matched to the record in the original file associated with the smallest distance. The proposed measure of confidentiality is the percentage of masked records with an incorrect link. A "correct link" is made if the masked record with the minimum distance to an original record is the masked record created from that original record. A "correct link" can alternatively be defined as occurring if the original record used to create the masked target record is associated with one of the three smallest distances to the masked record. The sum of absolute deviations between standardized variables and the sum of squared differences between standardized variables were proposed as distances. The standardized variables were such that all had the same variance. Spruill investigated the four combinations of distances and definitions of a link for the masking techniques of adding random error, multiplying by random error, grouping, random rounding, and data swapping.

In masking data by adding error, Spruill generated an independent normal error term for each variable of each data record. The errors were normally distributed with mean zero and variance equal to a multiple of the variance of the variable. The multiples considered were 0.1, 0.5, 1 and 2. Masked data records from an original file of 1000
records were used. Spruill concluded that adding random error in this manner only produced a high measure of confidentiality when the number of variables was small or the multiple of the variance was large.

Spruill (1983) also investigated the analytic usefulness of data sets masked under various techniques, including data swapping, multiplying by random error and adding random error. Two data sets were masked by various techniques and then analyzed. The first data set consisted of 36 variables and was constructed by applying normal deviates to means and coefficients of variation listed in an IRS publication. The second data set was an actual data set consisting of 27 variables of tax return data from 1979. Each data set contained 1000 records.

In masking by adding random error, Spruill used the same method as in the 1982 paper, but the added error term always had variance equal to one-half the variance of the variable. In evaluating data usefulness, Spruill computed, for the masked data records, the sample means, sample standard deviations and correlations between variables, as well as parameter estimates of regression models. The sample means of the masked variables were found to be equal to the means of the corresponding original variables. The direct estimates of regression and correlation parameters were judged inadequate for the technique of adding random error. Two avenues exist for improving the results of such analyses. First, alternative methods could have been used in masking the data. The analyses of the next section show that the correlation structure of the data can be preserved if correlated error
is added. Spruill added error which did not share the correlation structure of the original data. Second, because the variance of the added error is known, measurement error techniques (see Fuller (1987)) can be used to obtain consistent estimates of regression coefficients for data masked with error.

Paass (1985) conducted a matching experiment on data collected from the Federal Republic of Germany 1978 income and consumption sample (EVS) and the Federal Republic of Germany 1978 microcensus (MZ). The sampling fractions of the EVS sample and the MZ microcensus were 0.2% and 1% of the population, respectively. The EVS data set contained about 50,000 records on private households with about 370 household variables. The MZ data set consisted of 230,000 records on private households with about 50 variables. Match attempts were carried out for several scenarios, where the intruder had anywhere from seven to 68 variables to make use of in attempts to identify target records. Two different searches were conducted for each scenario. The directed search consisted of an intruder attempting to match the target to a microdata record when the intruder did not know whether the target record was included in the sample (i.e., the EVS or MZ file). The alternative search assumed the target was contained in the EVS or MZ file. The second procedure was used in the case where a 50% subsample of EVS records and a 20% subsample of MZ records had been released. It was unknown whether the target was included in the released subsamples.

Target records were generated by taking original records and adding error. The amount of error added to the variables to form the masked
records ranged from 1% for sex to 90% for some income variables. The typical error rates were 5-10% for categorical variables and 50-90% for quantitative variables.

Given a target record, the probability of correct association, \( p(k|x) \) in Paass' notation (see Section 1.2), was computed for each record in the released sample. The target was identified as the record in the release having the highest probability of correct association.

The directed search produced very unsuccessful results from the intruder's standpoint. For those scenarios having 45 and 68 variables, approximately 0.11% of the identification attempts were successful. For fewer variables the success rate was approximately zero.

The alternative search provided results that would be more encouraging to an intruder. Again, probabilities of correct association were computed using Paass' methods, and the target record was identified as the record in the subsample having the highest probability. If the target was known to be included in the released subsample, over 80% of the targets could be identified in the cases of 45 or 68 variables. Even when the target was not known to be in the subsample, over 55% of the targets could be correctly linked. Only for seven variables was the percentage of identifiable records equal to zero for both cases in the alternative search.

Paass concludes that microdata files containing few variables can be released. However, Paass believes that the information contained in records having many variables is so great that only excessive masking can secure confidentiality.
Kim (1986) developed a masking procedure which attempted to maintain the covariance structure of the data. He used three variations of his scheme to mask and analyze earnings data. The data were masked by adding error with variance equal to 25%, 50% and 100% of the variance of the unmasked variable. Also investigated was a standard masking technique, in which normal independent error terms with mean zero and variance equal to one-half the variance of the unmasked variables were added to each data value. The comparison of the masking methods involved the computation of means, correlations, regressions and analyses of variance.

Kim found the sample means of the masked variables to be different from the unmasked variables due to the sample variability of the mean of the added random noise. However, correlation coefficients between variables masked under Kim's scheme were the same as correlations between the unmasked variables. Correlations between variables masked by the standard technique of adding error were different from the true correlations. The same results were found to be true in the regression analyses. The estimated coefficients obtained from analyzing the data masked under Kim's scheme were nearly equal to the coefficients found from the true data. Much disparity existed between the coefficients estimated from the data masked by the standard technique and the true coefficients. In the analyses of variance based on the regressions, the mean square for error, the calculated F value, and the $R^2$ value exhibited the same pattern. Kim's masking scheme was clearly superior
to the standard technique with regard to maintaining statistical properties of the data.

4.1.2. Use of the masking algorithm

The masking algorithm described in Chapter 3 was implemented with the Interactive Matrix Language (IML) procedure of SAS. The documented code of the IML program is given in Chapter 6. In employing the program to mask a data set, the user must clearly define the input data set and also must specify a number of constants which determine how the data set will be masked.

To define the input data set, the user must first give the total number of input variables and the type of each variable. The three variable types are continuous, discrete and classificatory. Since parts of the masking procedure differ depending on the nature of the input variable, these specifications must be accurate. The distinction between discrete and classificatory variables deserves special mention. The difference lies in the meaning of the values of the variable. For a discrete variable, the integer response represents a quantity, while the response to a classification variable defines an individual category. Total family members is an example of a discrete variable, whereas race is a classification variable. For classificatory variables, the user must also specify the number of possible categories.

Once the variable types are specified, the user must indicate whether any variables have restricted response ranges. For example, a continuous variable may only take on non-negative values. The user must
define any such restrictions. After all variable values are masked, the routine will insure that these range restrictions are satisfied.

With regard to masking specifications, the user must first specify the magnitude of the error to be added in masking the transformed, standard normal values. With reference to Chapter 3, the relative variance of the error is represented by the value $\alpha$ in (3.2.4). If the value of $\alpha$ provided by the user is not large enough to satisfy the distance criterion discussed in Section 3.2.2, the magnitude of $\alpha$ is increased through the generation of larger errors.

In addition, the user must specify the number of iterations the program is to perform in attempting to attain the desired correlations between the original and masked data set and between the variables within the masked data set. If the correlations satisfy a closeness criterion before the specified number of iterations is exhausted, the program will omit the remaining iterations.

We now discuss the masking of the first computer generated data set.

4.2. Mask of the First Data Set

In using the IML program to mask a data set, we wanted to determine how well the algorithm performed in meeting the two primary masking objectives. These objectives are to create a new data set which is statistically representative of the original data and which provides respondents ample protection against disclosure. We chose to generate a data set made up of four continuous variables and four Bernoulli
variables. The four continuous variables consisted of three standard normal variables and a standard exponential variable. The generated Bernoulli variables had success probabilities of, approximately, 0.15, 0.30, 0.40 and 0.50. Because masking Bernoulli variables is an important special case of masking classification variables, we felt the mix of the standard continuous variables and the four Bernoulli variables would provide a good initial test for the program.

The generated data set consisted of 300 observations, each having a response to all eight variables. The data were generated so that high positive correlations existed between each possible pair of variables. The correlations of the masked data provide one indication of how representative the new data set is of the original data set.

To mask the data, the variance of the generated error term added to each transformed, standard normal value was set at 0.3. The maximum number of iterations permitted in attaining the desired cross correlations between masked and original variables and correlations between masked variables was set to fifteen in both cases.

After masking, in addition to comparing correlation structures, we constructed plots of the continuous variables and two-way tables of the Bernoulli variables, and we performed various regression analyses to determine whether the original and masked data sets were statistically similar. Furthermore, univariate distribution functions of original and masked variables were constructed to demonstrate their equivalence. To investigate the amount of protection the masking program affords, an
attempt was made to match each record in the masked data set to its corresponding record in the original data set.

Before discussing the analyses, we define the following notation. For the original data set, we denote the three normal variables by $X_1$, $X_2$, and $X_3$, and denote the exponential variable by $X_4$. The Bernoulli variables will be denoted by $X_5$, $X_6$, $X_7$, and $X_8$, having success probabilities of, approximately, 0.15, 0.3, 0.4 and 0.5, respectively. The masked variables will be denoted by $\hat{X}_j$, for $j = 1, 2, ..., 8$, where $\hat{X}_1$ is the masked analogue of $X_1$, and so on.

4.2.1. Plots and two-way tables

The initial analyses consisted of plots of the continuous variables and two-way tables of the Bernoulli variables. These plots and tables were constructed to illustrate the extent to which the original responses of each variable were altered in creating the new data set.

Plots were constructed for each continuous original variable against its masked counterpart. Because the plots of the normal variables have the same general shape, we only present the plot of $X_1$ vs. $\hat{X}_1$ in Figure 4.1(a). The points fall around the 45° line of the graph and the conditional variances of $X_1$ given a specific value of $\hat{X}_1$ tend to be similar. The sample correlation between $X_1$ and $\hat{X}_1$ is approximately 0.75, which indicates moderate, but not extreme, differences between each true and masked value.

Figure 4.1(b) contains the plot of the true and masked exponential values. As in the case of normal variables, the points tend to lie near
Figure 4.1. Original values vs. masked values for (a) the first normal variable, $X_1$, and (b) the exponential variable, $X_4$. 
the 45° line, although there is an abundance of points near the origin due to the skewness of the exponential distribution. The variance of $X_4$ conditional on $\hat{X}_4$ depends on $\hat{X}_4$. The sample correlation between $X_4$ and $\hat{X}_4$ is again approximately 0.75.

In Table 4.1, we give the means and standard deviations of the continuous original and masked variables. We observe that all means and all corresponding standard deviations are essentially equal.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Mean</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$</td>
<td>0.0038</td>
<td>0.9671</td>
</tr>
<tr>
<td>$X_2^*$</td>
<td>0.0040</td>
<td>0.9708</td>
</tr>
<tr>
<td>$X_3$</td>
<td>-0.0067</td>
<td>0.9920</td>
</tr>
<tr>
<td>$X_4^*$</td>
<td>-0.0063</td>
<td>0.9957</td>
</tr>
<tr>
<td>$X_6$</td>
<td>0.0146</td>
<td>0.9647</td>
</tr>
<tr>
<td>$X_7^*$</td>
<td>0.0155</td>
<td>0.9704</td>
</tr>
<tr>
<td>$X_8$</td>
<td>0.8693</td>
<td>0.8542</td>
</tr>
<tr>
<td>$X_9^*$</td>
<td>0.8716</td>
<td>0.8613</td>
</tr>
</tbody>
</table>

We now focus our attention on the two-way tables of the original and masked Bernoulli variables. In interpreting these tables, we concentrate on the tables of $X_6$ by $\hat{X}_6$ and $X_8$ by $\hat{X}_8$. The success probabilities in these two cases are approximately 0.3 and 0.5,
respectively. Again, we are looking for preliminary indications of the effectiveness of the program in masking Bernoulli variables.

Table 4.2 contains the cell totals and marginal totals for the table of $X_6$ by $\hat{X}_6$. In examining this table, we first note that the sample mean of $X_6$ is the same as the sample mean of $\hat{X}_6$. This is a special feature of the masking algorithm and will always result when a Bernoulli variable is masked. Also, we see that a total of 32 of the 300 observations were changed when the values of $X_6$ were masked. We shall refer back to this point after our discussion of the table of $X_8$ by $\hat{X}_8$. The sample correlation between $X_6$ and $\hat{X}_6$ is 0.74, which is similar to the sample correlations between the true and masked continuous variables.

<table>
<thead>
<tr>
<th></th>
<th>$\hat{X}_6$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$X_6$</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>199</td>
<td>16</td>
</tr>
<tr>
<td>1</td>
<td>16</td>
<td>69</td>
</tr>
<tr>
<td>Total</td>
<td>215</td>
<td>85</td>
</tr>
</tbody>
</table>

The two-way table of $X_8$ by $\hat{X}_8$ is illustrated in Table 4.3. We observe again that the sample means or success probabilities are equal. The mask of $X_8$ resulted in 21 values each changing from zero
Table 4.3. Crosstabulation of $X_8$ by $\hat{X}_8$

<table>
<thead>
<tr>
<th>$X_8$</th>
<th>0</th>
<th>1</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>138</td>
<td>21</td>
<td>159</td>
</tr>
<tr>
<td>1</td>
<td>21</td>
<td>120</td>
<td>141</td>
</tr>
<tr>
<td>Total</td>
<td>159</td>
<td>141</td>
<td>300</td>
</tr>
</tbody>
</table>

to one and an equal number changing from one to zero. This gives a sample correlation between $X_8$ and $\hat{X}_8$ of approximately 0.72.

We make one final observation in reference to the masking of any Bernoulli variable. Under the condition that the mean of $X_j$ is equal to the mean of $\hat{X}_j$, the sample correlation between $X_j$ and $\hat{X}_j$ is

$$\text{Corr}(X_j, \hat{X}_j) = \frac{\left( n_{11} + n_{12} \right) \left( n_{12} + n_{22} \right)}{\left( n_{11} + n_{12} + n_{21} + n_{22} \right) - 1} \left( n_{11} n_{22} - n_{12}^2 \right), \quad (4.2.1)$$

where $n_{ij}$ is the number of observations falling in the $(i,j)$-th cell of the table. Clearly, as $n_{12}$ increases, the sample correlation between $X_j$ and $\hat{X}_j$ decreases.

Now, let $p_j$ be the mean of the original and masked Bernoulli variable, $X_j$. As $n_{12}$ increases, we observe that the greater the absolute difference $|p_j - 0.5|$, the sharper the drop in the sample correlation between $X_j$ and $\hat{X}_j$. For example, consider the variables $X_8$ and $X_6$ with sample means $p_8 = 0.47$ and $p_6 = 0.283$, respectively. In the case of $X_8$, if only sixteen observations each
changed from zero for $X_8$ to one for $\tilde{X}_8$ and vice versa, the sample correlation between $X_8$ and $\tilde{X}_8$ would be 0.786. This value is clearly greater than the 0.74 correlation between $X_6$ and $\tilde{X}_6$ when, in fact, the number of observations masked from zero to one and one to zero was sixteen. Hence, in masking the Bernoulli variables $X_5$, $X_6$, $X_7$, and $X_8$, and anticipating the correlation between $X_j$ and $\tilde{X}_j$ to be around 0.75 for $j = 5, 6, 7, 8$, we would expect more values to change in the masking of $X_k$ than in the masking of $X_{k-1}$, for $k = 6, 7, 8$.

The results from these initial graphical and tabular analyses indicate that the algorithm is adequately, but not excessively, disguising the data. We now look more closely at the covariance structure of each data set through some regression analyses.

4.2.2. Regression analyses of original and masked data sets

Two linear models containing a subset of the eight variables were chosen and each was fit using ordinary least squares regression. Each regression was run separately on the original data set and on the masked data set. In addition to parameter estimates, we looked at standard errors of these estimates, $R^2$ values and diagnostic plots of residuals against predicted values. We now present results and interpretations for these models.

We first fit the model

$$X_1 = \beta_0 + \beta_2 X_2 + \beta_4 X_4 + \beta_6 X_6 + \text{error} \quad (4.2.2)$$
for both the original data and the masked data. Ordinary least squares gives the following estimates of the parameters, with standard errors in parentheses:

\[ X_1 = -0.237 + 0.671 X_2 + 0.257 X_4 + 0.077 X_6 \]
\[ (0.052) (0.041) (0.050) (0.077) \]  
\[ \text{(4.2.3)} \]

\[ \hat{X}_1 = -0.285 + 0.627 \hat{X}_2 + 0.296 \hat{X}_4 + 0.102 \hat{X}_6 \]
\[ (0.053) (0.042) (0.051) (0.075) \]  
\[ \text{(4.2.4)} \]

The \( R^2 \) values are 0.8194 for (4.2.3) and 0.8123 for (4.2.4).

In investigating these results, we first notice that the estimates for the corresponding parameters do not differ by very much. Only the two estimates of \( \beta_6 \) differ by more than 0.05. We also observe that the standard errors of the corresponding estimates are nearly identical, as are the \( R^2 \) values from the two fitted models.

An important consideration in any regression analysis is the plotting of residuals. We constructed plots of the residuals against the predicted values for the equations estimated from fitting the model in (4.2.2) on the original and masked data sets (see Figure 4.2). We observe that the two plots are very similar in that each contains a scattering of points with all residuals falling in the interval (-1.2, 1.2) in each plot. In comparing the parameter estimates, their standard errors, the \( R^2 \) values and the residual plots, we conclude that the regression of the masked data for the model (4.2.2) is very representative of the same regression on the original data set.
Figure 4.2. Residuals (E-HAT) vs. predicted values (X-HAT) for (a) the original data model, and (b) the masked data model
As a second model, we chose

\[ X_4 = \gamma_0 + \gamma_1 x_1 + \gamma_2 x_2 + \gamma_6 x_6 + \gamma_7 x_7 + \text{error} \quad (4.2.5) \]

and fit the model to both the original and masked data sets using ordinary least squares. We give the estimated equations with standard errors accompanying the parameter estimates:

\[ X_4 = 0.748 + 0.303 x_1 + 0.231 x_2 + 0.455 x_6 + 0.156 x_7 \quad (0.045) \quad (0.062) \quad (0.062) \quad (0.087) \quad (0.087) \quad (4.2.6) \]

\[ \hat{X}_4 = 0.791 + 0.332 \hat{x}_1 + 0.260 \hat{x}_2 + 0.416 \hat{x}_6 + 0.095 \hat{x}_7 \quad (0.042) \quad (0.060) \quad (0.058) \quad (0.085) \quad (0.081) \quad (4.2.7) \]

The \( R^2 \) values are 0.718 for the original data and 0.727 for the masked data. As in the first model, we observe that the estimates of the corresponding parameters are quite close. All fall within 0.06 of one another. Again the standard errors of the corresponding parameter estimates are nearly equal, as are the \( R^2 \) values. Finally, the plots of residuals against predicted values (Figure 4.3) constructed for each data set demonstrate exactly the same pattern. This identical residual pattern, along with the similarities between parameter estimates, standard errors, and \( R^2 \) values, further strengthens our claim that, with regard to regression analyses, the masked data set is representative of the original data set.
Figure 4.3. Residuals (E-HAT) vs. predicted values (X4-HAT) for (a) the original data model, and (b) the masked data model
4.2.3. **Examination of correlation structures and cross correlations**

We continue our statistical comparison of the original and masked data sets by investigating the correlations within each data set, as well as the correlations between corresponding original and masked variables. Let

\[
\mathbf{R}_{XX} = \begin{pmatrix} \mathbf{R}_{XX11} & \mathbf{R}_{XX12} \\ \mathbf{R}_{XX21} & \mathbf{R}_{XX22} \end{pmatrix}
\]

be the sample correlation matrix of the original data, where \( \mathbf{R}_{XX11} \) is the correlation submatrix of the continuous variables, \( \mathbf{R}_{XX22} \) is the correlation submatrix of the Bernoulli variables, and \( \mathbf{R}_{XX12} \) is the submatrix of correlations between the continuous and Bernoulli variables. We exhibit these subcorrelation matrices below and point out the high positive correlations, especially between the continuous variables:

\[
\mathbf{R}_{XX11} = \begin{pmatrix} 1 & 0.8927 & 0.8921 & 0.8005 \\ 0.8927 & 1 & 0.8923 & 0.7967 \\ 0.8921 & 0.8923 & 1 & 0.8059 \\ 0.8005 & 0.7967 & 0.8059 & 1 \end{pmatrix}
\]

\[
\mathbf{R}_{XX12} = \begin{pmatrix} 0.5897 & 0.6513 & 0.6967 & 0.6868 \\ 0.5891 & 0.6623 & 0.7133 & 0.7013 \\ 0.6057 & 0.6828 & 0.6881 & 0.7172 \\ 0.7168 & 0.7031 & 0.6864 & 0.6529 \end{pmatrix}
\]
We define the sample correlation matrix of the masked data as

\[
R_{XX22} = \begin{pmatrix}
1 & 0.6041 & 0.5624 & 0.4393 \\
0.6041 & 1 & 0.6979 & 0.6084 \\
0.5624 & 0.6979 & 1 & 0.6616 \\
0.4393 & 0.6084 & 0.6616 & 1
\end{pmatrix}.
\]

where the partitioning of \( R_{XX} \) is analogous to that of \( R_{XX} \). To assess the extent to which the correlation structure of the original data agrees with the correlation structure of the masked data, we look at the difference between the two correlation matrices. Let

\[
D_{XX} = \begin{pmatrix}
D_{XX11} & D_{XX12} \\
D_{XX21} & D_{XX22}
\end{pmatrix}
\]

be the difference between the original and masked correlation matrices. We investigate each submatrix of \( D_{XX} \) separately.

First, consider the difference between the correlation structures of the continuous variables of the two data sets. By definition,

\[
D_{XX11} = R_{XX11} - R_{XX11}^{**}
\]
where $D_{XX11}$ is a symmetric, $4 \times 4$ matrix. The $(i,j)$-th element of $D_{XX11}$ is

$$d_{XX11}(i,j) = r_{X_i X_j} - r_{\bar{X}_i \bar{X}_j}$$  \hspace{1cm} (4.2.11)

$$- d_{XX11}(j,i)$$

where $r_{X_i X_j}$ is the sample correlation between $X_i$ and $X_j$, and $r_{\bar{X}_i \bar{X}_j}$ is the sample correlation between $\bar{X}_i$ and $\bar{X}_j$. In our example,

$$D_{XX11} = \begin{pmatrix}
0 & 0.0083 & -0.0124 & -0.0113 \\
0.0083 & 0 & -0.0034 & -0.0066 \\
-0.0124 & -0.0034 & 0 & -0.0076 \\
-0.0113 & -0.0066 & -0.0076 & 0
\end{pmatrix}$$

indicating that all corresponding correlations for continuous variables in the original and masked data sets differ by less than 0.0125 in absolute value.

We now turn our attention to the correlations between Bernoulli variables, where, by definition

$$D_{XX22} = R_{XX22} - R_{\bar{X}\bar{X}22}.$$  

The $(i,j)$-th element of $D_{XX22}$ is
\[ d_{xx22}(i,j) = r_{X_k X_\ell} - r_{X_k X_\ell}^* \]

\[ = d_{xx}(j,1) \]

where \( k = i+4 \), \( \ell = j+4 \) and \( r_{X_k X_\ell} \) and \( r_{X_k X_\ell}^* \) are sample correlations between the subscripted variables. We find \( D_{xx22} \) to be

\[
D_{xx22} = \begin{pmatrix}
0 & 0.0407 & 0.0380 & 0.0184 \\
0.0407 & 0 & 0.0153 & 0 \\
0.0380 & 0.0153 & 0 & 0 \\
0.0184 & 0 & 0 & 0
\end{pmatrix}
\]

Hence, we conclude that \( r_{X_6 X_8} = r_{X_6 X_8}^* \) and \( r_{X_7 X_8} = r_{X_7 X_8}^* \), and we observe that all differences between corresponding Bernoulli variable correlations are less than 0.05 in absolute value.

To investigate differences between the corresponding sample correlations of continuous variables with Bernoulli variables, we compute

\[
D_{xx12} = R_{xx12} - R_{xx12}^*
\]

where \( D_{xx12} \) is a \( 4 \times 4 \) matrix, but \( D_{xx12}^* \) is not symmetric. The \((i,j)\)-th element of \( D_{xx12}^* \) is

\[
d_{xx12}(i,j) = r_{X_i X_\ell} - r_{X_i X_\ell}^*
\]

(4.2.12)
where $\ell = j + 4$, $r_{X_iX_\ell}$ is the sample correlation between $X_i$ and $X_\ell$, and $r_{X_iX_j}$ is the sample correlation between $X_i$ and $X_j$. In our example,

$$
D_{XX12} = \begin{bmatrix}
0.0627 & 0.0197 & 0.0523 & 0.0140 \\
0.0904 & 0.0319 & 0.0670 & 0.0380 \\
0.0815 & 0.0495 & 0.0473 & 0.0368 \\
0.0854 & 0.0231 & 0.0492 & 0.0393
\end{bmatrix}.
$$

We observe that all correlations between continuous and Bernoulli variables in the masked data set are somewhat lower than the same correlations in the original data set. In particular, correlations between $X^*_5$ and all other masked variables declined by the largest amount. We conclude that it is difficult to retain the high correlations between continuous and Bernoulli variables during the mask.

The difficulty with $X^*_5$ lies in the fact that only 15.67% of the original responses of $X_5$ are one, while the rest are zero. To adequately mask $X_5$, the algorithm changed a relatively high percentage of the ones to zeros. Consequently, the "new" zeros, which are associated with large values of other variables, decreased the correlation between $X^*_5$ and these other variables. The correlated error added to mask each data vector helped a moderate amount to retain the high correlations between $X^*_5$ and the other variables, but the all-or-nothing nature of the Bernoulli variable and the low success
probability of $X_5$ made the high positive correlations difficult to completely retain.

We now examine correlations between corresponding variables from the two data sets. Some of these cross correlations were briefly mentioned in our discussion of plots and two-way tables of original and masked values. Regarding cross correlations, the objective is to have each cross correlation approximately equal to a target correlation. This target correlation is a robust average of all the cross correlations at the point of the initial mask of the data. In this example, the target correlation was computed to be 0.75. A correlation of 0.75 produces values of the masked variables which are not too different from the values of the original variables from which they were created. This is consistent with our masking objective of not excessively distorting the data. At the same time, the target cross correlation of 0.75 creates enough distortion in the data to serve as considerable confidentiality protection, which is our second primary objective. The fact that this target correlation provides considerable protection in a sample of this size will be substantiated later.

The sample correlations between the corresponding original and masked continuous variables are
All are just below our target correlation of 0.75. For the Bernoulli variables, we have

\[
\begin{pmatrix}
    r_{X_1^*X_1} & r_{X_1^*X_2} & r_{X_1^*X_3} & r_{X_1^*X_4} \\
    r_{X_2^*X_1} & r_{X_2^*X_2} & r_{X_2^*X_3} & r_{X_2^*X_4} \\
    r_{X_3^*X_1} & r_{X_3^*X_2} & r_{X_3^*X_3} & r_{X_3^*X_4} \\
    r_{X_4^*X_1} & r_{X_4^*X_2} & r_{X_4^*X_3} & r_{X_4^*X_4}
\end{pmatrix} = \begin{pmatrix}
    0.7451 \\
    0.7448 \\
    0.7449 \\
    0.7442
\end{pmatrix}.
\]

Again, all fall just below the 0.75 target.

In the masking algorithm, an iterative procedure is performed which adjusts the error terms until the cross correlations between corresponding original and masked variables is within one one-hundredth of the target correlation (see Section 3.2.2). This is followed by a second iterative procedure, which further adjusts the data to equate correlations within the masked data set to corresponding correlations within the original data set. The computations in the second iterative procedure, in some instances, cause a decrease in one or more of the cross correlations. Though two of the cross correlations - \( r_{X_6^*X_6} \) and
have dropped, the decrease is only a slight one and is not cause for concern. We believe that all cross correlations are close enough to the target correlation of 0.75.

4.2.4. **Comparison of univariate distribution functions**

We conclude our statistical comparison of the original and masked data sets by demonstrating that the univariate distribution functions for corresponding original and masked variables are identical. For the continuous variables, we concentrate on the normal variable, $X_3$, and its masked analogue, along with the original and masked exponential variable. The sample distribution functions of $X_3$ and $\hat{X}_3$ are pictured in Figure 4.4. The two graphs are nearly identical. When overlayed on the same pair of axes, the two sample distribution functions appear as one. We found the same equivalence between the sample distribution functions of the original and masked exponential variables, $X_4$ and $\hat{X}_4$, respectively. The near equality of these two distribution functions is illustrated in Figure 4.5.

For Bernoulli variables, the sample distribution function is a step function. Since our masking algorithm guarantees that the mean of a masked Bernoulli variable is equal to the mean of the original Bernoulli variable, then the univariate distribution functions of the original and masked Bernoulli variables will be identical.

4.2.5. **Assessment of protection against disclosure**

Based on the analyses discussed above, the masked data set has retained important statistical properties of the original data set. As
Figure 4.4. Sample cumulative distribution function of $X_3$ for (a) the original data, and (b) the masked data.
Figure 4.5. Sample cumulative distribution function of $X_4$ for (a) the original data, and (b) the masked data.
far as protection against disclosure, the cross correlations given in section 4.2.4 indicate that the masked data values are close, but not identical to the original values. We now investigate the amount of confidentiality protection the mask has afforded the respondents of the original data set.

Disclosure occurs when an intruder obtains an identifiable target record from a private source and matches it to a record from a public use microdata file. The number of private data sources from which a target record may arise is not known. Hence, before releasing a masked file, a data agency cannot assure absolute protection from disclosure simply because the agency cannot account for every record in the private domain that corresponds to a record in their microdata file. Despite this, a data agency must determine whether a data file has been adequately masked against disclosure. A conservative method of assessing the amount of confidentiality protection is to attempt to match each record in the original data file to its corresponding record in the masked file using all variables. We call this a self-contained match attempt. If such a match attempt is unsuccessful, the data agency can claim that their mask of the data affords respondents a reasonable amount of protection. For this type of data releasing process to be successful, definitions of an "unsuccessful match attempt" and a "reasonable amount of protection" must be formulated by the data releasing agency or by the statistical community.

In order to assess the amount of protection afforded by our masking algorithm, we carried out such a self-contained match attempt. For each
record in the original data file, which we refer to as the target record, we first computed the probabilities that the target record corresponded to each record in the masked data file. These were computed as if the added error is independent of the original data. This is not a correct assumption because the algorithm modifies the error on the basis of adjacent observations. We refer to these probabilities as \((p_{1k}, p_{2k}, \ldots, p_{300,k})\), where the \(k\)-th original record is the target.

The values of \(p_{jk}\) for \(j,k \in K\), where \(K = \{1,2,\ldots,300\}\), were computed as follows. Let the original data set be defined by

\[ X = (X_1', X_2', \ldots, X_{300}') \]

where \(X_j = (X_{j1}, X_{j2}, \ldots, X_{j8})\), for \(j=1, 2, \ldots, 300\). Let the masked data set be defined by

\[ \hat{X} = (\hat{X}_1', \hat{X}_2', \ldots, \hat{X}_{300}') \]

where \(\hat{X}_j = (\hat{X}_{j1}, \hat{X}_{j2}, \ldots, \hat{X}_{j8})\) for \(j=1, 2, \ldots, 300\). Note that \(\hat{X}_{j\ell}\) is a function of \(X_{j\ell}\) for \(j=1, 2, \ldots, 300\) and \(\ell=1, 2, \ldots, 8\). The function involves transforming \(X_{j\ell}\) to normality, adding an error term, standardizing the sum and back transforming to the original scale of the \(j\)-th variable. We define

\[ M_{DD} = (299)^{-1}(D'D - 300 \bar{D}'\bar{D}) \]
where

\[ D = X - \bar{X}, \]

\[ D = (300)^{-1} \sum_{j=1}^{300} (X_j - \bar{X}_j). \]

To compute the \( p_{jk} \) values we first define

\[ \gamma_{jk} = \exp\left(-\frac{1}{2} (\bar{X}_j - X_k)^{\top} D^{-1} (\bar{X}_j - X_k)\right) \quad (4.2.14) \]

for \( j, k \in K \). Finally, we define

\[ p_{jk} = \left( \sum_{i=1}^{300} \gamma_{1k}^{-1} \gamma_{jk} \right)^{-1} \quad (4.2.15) \]

for \( j, k \in K \).

If the \( k \)-th record in the original file is the target record, we are particularly interested in the magnitude of \( p_{kk} \). While not probabilities for general \( X \), we treat these values as the posterior probability of matching the target record to its masked analogue. We also call these values match probabilities. We are interested in the largest probability among all values of \( p_{jk} \), for \( k \) fixed. We refer to this as \( P_{\text{max}}(k) \), and define

\[ P_{\text{max}}(k) = \max_{j \neq k} \{ p_{jk} \}. \quad (4.2.16) \]
We use these values in various ways to investigate and quantify the amount of protection provided by our masking algorithm.

First, we use the match probabilities to determine how many of the original records are closely associated to their corresponding masked records in a self-contained matching experiment. That is, for all three hundred original records, we determine the number of times the masked record created from the original had the highest match probability, one of the three highest match probabilities, and one of the five highest match probabilities. In mathematical terms, we first define

\[ P(1),k \geq P(2),k \geq \ldots \geq P(300),k \]  

as the 300 ordered match probabilities for \( k \) (\( k \) fixed). We then define \( U(\ell),300 \) as the proportion of original records, \( k \in K \), for which \( p_{kk} \) is one of the \( \ell \)-largest match probabilities \( \{p_{jk}: j \in K\} \), in a self contained match. Hence,

\[ U(\ell),300 = (300)^{-1} \sum_{k=1}^{300} I(p_{kk} \geq p(\ell),k) \]  

for \( \ell = 1, 3 \) and 5 where

\[ I(p_{kk} \geq p(\ell),k) = 0 \text{ if } p_{kk} < p(\ell),k \]  

\[ = 1 \text{ if } p_{kk} \geq p(\ell),k . \]
In this example, we found

\[ U_{(1),300} = 0.0967 \]
\[ U_{(3),300} = 0.2267 \]
\[ U_{(5),300} = 0.2800 \]

A quick interpretation of these results is that, in attempting to match an original record to its masked analogue using the highest probability, over 90% of the match attempts were unsuccessful and approximately 3/4 of the time the proper record was not among the five masked records with highest match probabilities.

We further illustrate the difficulty an intruder would have in matching a target record to the proper record in the microdata release. In the self-contained match, we constructed histograms of the \( P_{kk} \) values and \( P_{\text{max}}(k) \) values for all \( k \). Figure 4.6 illustrates that the highest incorrect match probabilities tend to exceed the correct match probabilities, misleading an intruder interested in predicting confidential attributes or properly linking a masked record to a target.

As a final indicator that the masking algorithm introduced in Chapter 3 provides protection against disclosure, we show that in the self contained match, even apparently obvious matches are typically incorrect. In looking at match probabilities, an intruder will feel positive about linking a released record to the target if the associated
Figure 4.6. Histograms of (a) the correct match probabilities, and (b) the highest incorrect match probabilities
probability is both large and dominant over the others. One criterion developed from this idea may be to make a link between a target record and a released microdata record if the associated match probability is both greater than 0.5 and at least twice as large as the second highest probability of matching another record to the target. In linking the k-th original record to a record in the masked data set, the criterion is satisfied if

\[ P(1),k > 0.5 \]

\[ P(1),k > 2P(2),k \]

where \( P(1),k \) and \( P(2),k \) are defined in (4.2.17).

In our example, the attempted self-contained match of each original record to a masked record results in a total of nineteen original records having match probabilities satisfying the conditions of (4.2.20). Of these nineteen situations where an intruder may feel that a positive link between target and released record can be made, only four of the nineteen cases would result in correct matches. Therefore, in cases where an intruder may feel that a correct match can be made, only about 20% of the record links would be correct.

4.3. Mask of the Second Data Set

In the mask of the first data set in section 4.2, we demonstrated that the algorithm did well in meeting the two primary masking
objectives. For a data set composed of normal, exponential and Bernoulli variables, the masking program created a data set which was statistically representative of the original data set and provided considerable protection against disclosure. In masking a second data set, we wanted to demonstrate some additional capabilities of the algorithm. These include the ability to sufficiently mask discrete and classification variables. Hence, we created a data set including a Poisson variable with parameter $\lambda = 1.8$ and a classification variable having a categorical range of $\{1, 2, 3, 4\}$. We also created a standard normal variable which was highly correlated with the Poisson variable. Additionally, we correlated large values of the Poisson and normal variables with the first category of the classification variable, and small values of the normal and Poisson variables with the fourth category of the classification variable. We could then determine whether the descriptive statistics of the normal and Poisson variables within the categorical groups of the classification variable are similar for the original and masked data sets.

In the mask of the first data set, we showed that the masking algorithm preserved the univariate distribution function of each variable. We cannot make the same claim about the joint distribution function of two or more of the variables. To demonstrate this, we included as a variable in the data set, the square of the standard normal variable. Naturally, if one knew that two variables were perfectly related, one would mask only one and create the other using
the functional relationship. Our purpose is to take an extreme case to demonstrate how the joint relationship can be modified by the masking.

In addition, we will study the correlation structure of the data sets within subgroups of the classification variable. The normal and Poisson variables are positively correlated with each other and nearly uncorrelated with the chi-square variable. However, within subgroups of the classification variable, the correlation structure is very different. We will demonstrate that in such cases the correlation structures within categories of the classification variable are not retained through the mask.

The second computer generated data set consisted of 300 observation vectors. Each vector consisted of a value for the normal, chi-square, Poisson and classification variables. In the masking algorithm, the variance of the error term added to each transformed standard normal value was set to 0.3. The maximum number of iterations permitted to attain the desired correlation structure in the masked data set and the cross correlations between original and masked variables was chosen to be ten in both cases. Although the data set has four variables, the algorithm had to mask six variables, because the classification variable was transformed into three Bernoulli variables.

In analyzing the new data set, we concentrated on the masked Poisson and classification variables. We began by constructing two-way tables of the Poisson and categorical variables against their masked counterparts. We also fit the same linear model to the original and masked data sets and compared the results. For both data sets, we
computed descriptive statistics of the normal, chi-square and Poisson variables within subgroups of the classification variable. We also computed the overall correlation structure of each data set and correlation structures of each data set within subgroups of the classification variable. To demonstrate that the masking algorithm does not preserve joint distribution functions, we plotted the chi-square variable against the normal variable in each data set and derived certain components of the sample joint distribution functions.

In discussing the results of the mask, we denote the variables $X_1$, $X_2$, $X_3$ and $X_4$ as corresponding to the standard normal, chi-square, Poisson and classification variables, respectively. Also, we refer to the corresponding masked analogues of these variables as $\hat{X}_1$, $\hat{X}_2$, $\hat{X}_3$ and $\hat{X}_4$.

4.3.1. Cross tabulations of the discrete and classification variables

We begin with the cross tabulation of $X_3$ by $\hat{X}_3$ in Table 4.4. We point out two important features of the procedure to mask discrete variables, which are illustrated in the table. First of all, we notice that the masked values are not too distant from their original values. Only for $X_3 = 0$ do we find a corresponding value of $\hat{X}_3$ such that $|X_3 - \hat{X}_3| > 2$. Secondly, let us define $(X_{3t}^*: t = 1,2,\ldots,300)$ and $(\hat{X}_{3t}^*: t = 1,2,\ldots,300)$ as the original and masked sets of responses for the Poisson variable. If we further define
Table 4.4. Frequencies for the original and masked Poisson variables

<table>
<thead>
<tr>
<th>$X_3$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>26</td>
<td>19</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>39</td>
<td>22</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>86</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>20</td>
<td>30</td>
<td>18</td>
<td>3</td>
<td>0</td>
<td>76</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>8</td>
<td>15</td>
<td>18</td>
<td>7</td>
<td>3</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>5</td>
<td>10</td>
<td>5</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>52</td>
<td>86</td>
<td>76</td>
<td>51</td>
<td>24</td>
<td>9</td>
<td>300</td>
</tr>
</tbody>
</table>

\[
\sum_{t=1}^{300} n_j = \sum_{t=1}^{300} I(X_{3t} - j) \tag{4.3.1}
\]

\[
\sum_{t=1}^{300} n^*_j = \sum_{t=1}^{300} I(\hat{X}_{3t} - j) \tag{4.3.2}
\]

where

\[
I(X_{3t} - j) = \begin{cases} 
0 & \text{if } X_{3t} \neq j \\
-1 & \text{if } X_{3t} = j 
\end{cases}
\]

then we see that $n_j = n^*_j$, for $j = 1, 2, \ldots, \max(X_{3t})$. That is, the masked Poisson variable has the same mean and variance as the original Poisson variable. This will be true when a discrete variable having a range of values which form a consecutive sequence of integers,
like \((0, 1, \ldots, 6)\), is masked. We also note that 175 of the 300 original Poisson values, or 58.3\%, were changed in the mask of the data set. Thus, a very different set of responses was created.

We turn our attention to Table 4.5 which contains cross tabulations for the original and masked classification variables, \(X_4\) and \(\hat{X}_4\). In masking a classification variable, we cannot guarantee that the frequencies for the categorical values of the original variable will be the same as those of the masked variable. However, we see from the

Table 4.5. Frequencies for the original and masked classification

<table>
<thead>
<tr>
<th>(X_4)</th>
<th>(\hat{X}_4)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>64</td>
<td>4</td>
<td>4</td>
<td>9</td>
<td>81</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>4</td>
<td>19</td>
<td>3</td>
<td>4</td>
<td>30</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>5</td>
<td>3</td>
<td>37</td>
<td>8</td>
<td>53</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>115</td>
<td>136</td>
</tr>
</tbody>
</table>

Total | 81 | 33 | 53 | 136| 300|


<table>
<thead>
<tr>
<th>n_{ij}</th>
<th># observations in cell ((i,j)),</th>
</tr>
</thead>
<tbody>
<tr>
<td>\n_{i.}</td>
<td>\sum_{j=1}^{4} n_{ij} ] ,</td>
</tr>
</tbody>
</table>

and

\[ n_{.j} = \sum_{i=1}^{4} n_{ij} , \]

then \( |p_{k.} - p_{.k}| \leq 0.01 \) for \( k = 1, 2, 3, 4 \) where

\[ p_{k.} = \left( \sum_{i=1}^{4} \sum_{j=1}^{4} n_{ij} \right)^{-1} n_{k.} , \]

\[ p_{.k} = \left( \sum_{i=1}^{4} \sum_{j=1}^{4} n_{ij} \right)^{-1} n_{.k} . \]

We also note that, of the 300 categorical values that were masked, 65 or 21.67% actually switched categories in the mask. Again, the algorithm is creating a new set of responses which differs substantially from the original set.

4.3.2. Regression analysis with classificatory predictor

To investigate the effects of masking discrete and classification variables on fitting a linear model, we used ordinary least squares to fit

\[ X_1 = \sum_{j=1}^{4} \alpha_j Z_j + \beta X_3 + \text{error} \quad (4.3.3) \]

where for \( j = 1, 2, 3, 4, \)
\[ Z_j = 0 \text{ if } X_4 \neq j \]
\[ = 1 \text{ if } X_4 = j. \]

We obtain the following estimates of the parameters for the original and masked data sets, with standard errors in parentheses:

\[
X_1 = -0.86 \, Z_1 - 0.79 \, Z_2 - 0.91 \, Z_3 - 1.14 \, Z_4 + 0.54 \, X_3 \quad (4.3.5)
\]
\[
\begin{align*}
\hat{X}_1 &= -0.73 \, Z_1 - 0.84 \, Z_2 - 0.81 \, Z_3 - 1.13 \, Z_4 - 0.51 \, X_3. \\
(0.12) &\quad (0.13) &\quad (0.12) &\quad (0.06) &\quad (0.03)
\end{align*}
\]

The respective R-squared values for the original and masked models are 0.6743 and 0.6496. We observe that, although the standard errors of the corresponding estimates are nearly identical, the estimates of \( \alpha_1 \), \( \alpha_2 \) and \( \alpha_3 \) more than slightly differ in the two models. Differences in the three pairs of parameter estimates result from the small numbers of observations in the three categories within each data set (see Table 4.4). The estimates for \( \alpha_2 \) are surprisingly close even though the numbers of observations falling in the second category are 30 and 33, respectively, for the original and masked data sets. We point out that, with a frequency count of 136 observations for the fourth category in each data set, the parameter estimates of \( \alpha_4 \) are within one one-hundredth of one another. For all parameter estimates, we obtain nearly equal standard errors due to the similarity of the sums of squares and cross products matrices and the closeness of the error mean squares,
which are 0.3357 and 0.3687 for the original and masked models. There exist no extremely visible differences between the residual plots in Figure 4.7 for the original and masked data sets. We conclude there is little difference between the two regressions.

4.3.3. **Examination of overall correlation structure and cross correlations**

We continue our analysis of the mask of the second data set by examining the correlations within the original and masked data set, as well as correlations between corresponding variables of the two data sets. In discussing correlations, we focus on the quantitative variables, $X_1$, $X_2$, and $X_3$. Let

$$R_{XX} = \begin{pmatrix} R_{XX} & R_{XX}^* \\ R_{XX}^* & R_{XX}^{**} \end{pmatrix},$$

where $R_{XX}$ is the sample correlation matrix of the original quantitative variables, $R_{XX}^{**}$ is the sample correlation matrix of the masked variables and $R_{XX}^*$ is the sample cross correlation matrix between the original and masked quantitative variables. We give the correlation matrices of the original and masked data sets, $R_{XX}$ and $R_{XX}^{**}$. The matrix of cross correlations is discussed later in this section.
Figure 4.7. Residuals (E-HAT) vs. predicted values (X1-HAT) for (a) the original data model, and (b) the masked data model.
To demonstrate the similarity between the correlation matrices of the original and masked data sets, we form the matrix of correlation differences,

\[ D_{XX} = R_{XX} - R_{XX}^\dagger, \]

where \( D_{XX} \) is a symmetric 3x3 matrix. The (i,j)-th element of \( D_{XX} \) is

\[ d_{XX}^{\dagger}(i,j) = r_{X_iX_j} - r_{X_i\hat{X}_j} \]

\[ = d_{XX}(j,i), \]

where \( r_{X_iX_j} \) is the sample correlation between \( X_i \) and \( X_j \), and \( r_{X_i\hat{X}_j} \) is the sample correlation between \( \hat{X}_i \) and \( \hat{X}_j \). In masking the second data set, we find
We see that the correlation structures of the original and masked data sets are nearly identical.

The sample correlations between corresponding variables in the original and masked data sets are the diagonal elements of $R_{XX}$. These are computed to be

\[
D_{XX} = \begin{bmatrix}
0 & -0.0069 & 0.0196 \\
-0.0069 & 0 & -0.0006 \\
0.0196 & -0.0006 & 0
\end{bmatrix}.
\]

The target correlation in this example was computed to be 0.74. The iterative procedure in the routine which has the purpose of equating the correlation matrices of the original and masked data sets has caused the correlation between the original and masked Poisson variable to fall slightly below the target correlation of 0.74, but a cross correlation of 0.7222 is still satisfactory.

With regard to the analyses and other statistical computations discussed above, we have found that the masked data set is statistically representative of the original data set. Only in the regression analysis, when the estimates of the categorical means were compared, did
we find noteworthy differences between the original and masked data sets. We further pursue these potential differences by looking at descriptive statistics and correlation structures of the quantitative variables within subgroups of the classification variable.

4.3.4. Comparisons within subgroups of the classification variable

We begin by considering the means and standard deviations of the normal, chi-square and Poisson variables within the four subgroups of the classification variable. The subgroup means and standard deviations for both the original and masked data sets are found in Table 4.6. The

<table>
<thead>
<tr>
<th>Variable</th>
<th>Category</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>$X_1$</td>
<td>0.764</td>
<td>0.261</td>
<td>0.369</td>
<td>-0.658</td>
</tr>
<tr>
<td></td>
<td>(0.810)</td>
<td>(0.628)</td>
<td>(0.725)</td>
<td>(0.848)</td>
</tr>
<tr>
<td>$\hat{X}_1$</td>
<td>0.728</td>
<td>0.321</td>
<td>0.398</td>
<td>-0.661</td>
</tr>
<tr>
<td></td>
<td>(0.833)</td>
<td>(0.776)</td>
<td>(0.718)</td>
<td>(0.846)</td>
</tr>
<tr>
<td>$X_2$</td>
<td>1.232</td>
<td>0.450</td>
<td>0.652</td>
<td>1.148</td>
</tr>
<tr>
<td></td>
<td>(1.489)</td>
<td>(0.694)</td>
<td>(0.794)</td>
<td>(1.483)</td>
</tr>
<tr>
<td>$\hat{X}_2$</td>
<td>1.163</td>
<td>0.646</td>
<td>0.659</td>
<td>1.168</td>
</tr>
<tr>
<td></td>
<td>(1.466)</td>
<td>(0.834)</td>
<td>(0.863)</td>
<td>(1.569)</td>
</tr>
<tr>
<td>$X_3$</td>
<td>2.975</td>
<td>1.933</td>
<td>2.358</td>
<td>0.882</td>
</tr>
<tr>
<td></td>
<td>(1.193)</td>
<td>(0.980)</td>
<td>(1.039)</td>
<td>(0.870)</td>
</tr>
<tr>
<td>$\hat{X}_3$</td>
<td>2.827</td>
<td>2.242</td>
<td>2.340</td>
<td>0.912</td>
</tr>
<tr>
<td></td>
<td>(1.212)</td>
<td>(1.061)</td>
<td>(1.287)</td>
<td>(0.839)</td>
</tr>
</tbody>
</table>
frequency counts for each category, given in Table 4.7, provide the population sizes from which the descriptive statistics were computed. To visually aid the comparison of subgroup means and standard deviations for the original and masked data sets, plots of the tabled values are included in Figure 4.8.

Table 4.7. Frequency counts for categories of the original and masked classification variables

<table>
<thead>
<tr>
<th>Category</th>
<th>Variable</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$X_4$</td>
<td>81</td>
<td>30</td>
<td>53</td>
<td>136</td>
</tr>
<tr>
<td></td>
<td>$\hat{X}_4$</td>
<td>81</td>
<td>33</td>
<td>50</td>
<td>136</td>
</tr>
</tbody>
</table>

From the tables and the graphs, we observe more disparity between corresponding variable means and standard deviations of the two data sets for subgroups having smaller frequency counts. For example, the subgroup means and standard deviations for the fourth category are very similar for the chi-square and Poisson variables, and nearly identical for the original and masked normal variables. The corresponding original and masked variable means and standard deviations for the second category differ much more than those of the fourth category. We also notice that the respective original and masked variable means for the third subgroup are nearly equal, though their respective standard deviations tend to differ. Finally, we observe that, over all
Figure 4.8. Original and masked sample (a) means and (b) standard deviations by classification subgroups of $X_4$. 
(categories, the means and standard deviations for the original and masked normal variables agree the most, while the means and standard deviations of the original and masked Poisson variable differ the most. All differences are small relative to the standard errors.

Let us now investigate the correlation structure of the quantitative variables. Before giving the four correlation matrices corresponding to original data values within the subgroups of the classification variable, we explain why these matrices will differ.

The chi-square variable, which is the square of the normal variable, is nearly uncorrelated with the normal and Poisson variables. Within subgroups of the classification variable, however, the chi-square variable is correlated with the normal and Poisson variables. Within the first category, the chi-square variable is positively correlated with the other two variables. This follows from the fact that records belonging to the first category have large positive Poisson and normal values and, hence, large chi-square values. Contrarily, records in the fourth category are associated with small values of the normal and Poisson variables. Hence, the chi-square variable is negatively correlated with the normal and Poisson variables in the category "4" subgroup of the classification variable. Though not interpreted easily, the correlation structures for the second and third categories also differ from the correlation structure of the entire data set. The four correlation matrices corresponding to data vectors belonging to the four categories of the classification variable are exhibited below, where $R_{XX}(j)$ denotes the correlation matrix of data
vectors for which \( X_4 = j \).

\[
R_{XX(1)} = \begin{pmatrix} 1.00 & 0.88 & 0.70 \\ 0.88 & 1.00 & 0.64 \\ 0.70 & 0.64 & 1.00 \end{pmatrix} \quad R_{XX(2)} = \begin{pmatrix} 1.00 & -0.33 & 0.57 \\ -0.33 & 1.00 & -0.18 \\ 0.57 & -0.18 & 1.00 \end{pmatrix}
\]

\[
R_{XX(3)} = \begin{pmatrix} 1.00 & 0.74 & 0.70 \\ 0.70 & 1.00 & 0.58 \\ 0.70 & 0.58 & 1.00 \end{pmatrix} \quad R_{XX(4)} = \begin{pmatrix} 1.00 & -0.78 & 0.73 \\ -0.78 & 1.00 & -0.45 \\ 0.73 & -0.45 & 1.00 \end{pmatrix}
\]

We proceed to compare the correlation structures of the original and masked data sets within their respective subgroups of the classification variable. We anticipate that the correlation matrix for the \( j \)-th subgroup of the original data set will differ from the correlation matrix for the \( j \)-th subgroup of the masked data set for the following reason. In masking a data set, the algorithm adds error vectors to the transformed data vectors. The error vectors have a covariance matrix which is a multiple of the covariance matrix of all the transformed data. That is, a transformed data vector which belongs to the first subgroup of the classification variable has an error vector added which has the same covariance as an error vector added to a transformed data vector belonging to the fourth subgroup. Hence, the correlation matrix for original data in the \( j \)-th subgroup will differ from the correlation matrix of the masked data belonging to the \( j \)-th subgroup.

Analogous to \( R_{XX(j)} \), we define \( R_{XX(j)}^{**} \) to be the correlation matrix of \( \hat{X}_1 \), \( \hat{X}_2 \) and \( \hat{X}_3 \) for masked observations with \( \hat{X}_4 = j \). Let
be the difference of the original and masked correlation matrices for the j-th subgroup, where $D_{\text{XX}(j)}$ is a symmetric $(3 \times 3)$ matrix. The $(i,k)$-th element of $D_{\text{XX}(j)}$ is

$$d_{\text{XX}ik}(j) = r_{X_1 X_k}(j) - r_{\hat{X}_1 \hat{X}_k}(j)$$

$$= d_{\hat{\text{XX}ki}(j)}$$

where $r_{X_1 X_k}(j)$ is the sample correlation between values of $X_1$ and $X_k$ within the j-th category of $X_4$. We give the matrices of differences, $D_{\hat{\text{XX}}(j)}$ for j=1, 2, 3 and 4 and note that the correlation structures within subgroups of the classification variable are quite different for the original and masked data sets.

$$D_{\hat{\text{XX}}(1)} = \begin{bmatrix} 0 & 0.57 & -0.14 \\ 0.57 & 0 & 0.32 \\ -0.14 & 0.32 & 0 \end{bmatrix}, \quad D_{\hat{\text{XX}}(2)} = \begin{bmatrix} 0 & -0.32 & -0.08 \\ -0.32 & 0 & -0.55 \\ -0.08 & -0.55 & 0 \end{bmatrix}$$

$$D_{\hat{\text{XX}}(3)} = \begin{bmatrix} 0 & 0.65 & -0.27 \\ 0.65 & 0 & 0.63 \\ -0.27 & 0.63 & 0 \end{bmatrix}, \quad D_{\hat{\text{XX}}(4)} = \begin{bmatrix} 0 & -0.45 & 0.18 \\ -0.45 & 0 & -0.41 \\ 0.18 & -0.41 & 0 \end{bmatrix}$$

In such cases, the sophisticated user can recover the correct covariance matrix using measurement error techniques. See Fuller (1987). We do not pursue that approach further.
In this discussion of subgroup correlation matrices, we have demonstrated that a user of a microdata release masked by our algorithm cannot be guaranteed that a non-random subset of the masked data will be statistically representative of the same non-random subset of the original data. Specifically, we focused on subsets defined by categories of a classification variable in the data set. We also saw that if the frequency count of the subgroup is relatively large, the subgroup mean and standard deviation of the masked data set tend to be similar to the original subgroup mean and standard deviation. However, we cannot expect the same agreement from the subgroup correlation matrices of the original and masked data sets. In general, when a data set is masked by our algorithm, statistical relationships between variables within a non-random subset are not preserved. On the other hand, our example was extreme in that we constructed it to have very different correlation structures in different subsets.

4.3.5. Disparity of joint distribution functions

To conclude the summary of masking the second data set, we demonstrate that the masking algorithm does not necessarily preserve the joint distribution function of two or more of the variables in the original data set. In the original data set, the values of $X_2$ are equal to the square of the standard normal values of $X_1$. This exact relationship between two variables (i.e., $X_2 = X_1^2$) will rarely occur in a data set. We created such a relationship for the sake of illustration. If an exact relationship like $X_j = f(X_1)$ does exist in
a data set which is to be masked, the proper procedure is to only mask
the variable $X_1$ and define $\hat{X}_j = f(\hat{X}_1)$ before releasing the masked
data set.

We illustrate the differences between the joint distribution of
$(X_1, X_2)$ and the joint distribution of $(\hat{X}_1, \hat{X}_2)$ by plotting $X_2$
against $X_1$ and $\hat{X}_2$ against $\hat{X}_1$. Figure 4.9 contains the plot of
$X_2$ versus $X_1$, which is a graph of $X_2 = X_1^2$ for a sample of 300
randomly generated standard normal values. Figure 4.10 contains the
plot of $\hat{X}_2$ against $\hat{X}_1$. Visually, the distributional relationship
between $X_1$ and $X_2$ has been destroyed by the mask.

Mathematically, the joint distribution of $X_1$ and $X_2$ is given by

$$F_{X_1, X_2}(x_1, x_2) = P(X_1 \leq x_1, X_2 \leq x_2)$$

$$= P(X_1 \leq x_1, X_1^2 \leq x_2)$$

$$= P(X_1 \leq x_1, -\sqrt{x_2} \leq X_1 \leq \sqrt{x_2}),$$

where $x_1 \in \mathbb{R}$ and $x_2 \geq 0$. Hence, the joint distribution of $X_1$
and $X_2$ is

$$F_{X_1, X_2}(x_1, x_2) = \begin{cases} 
0 & \text{if } x_1 \leq -\sqrt{x_2} \\
P(-\sqrt{x_2} \leq X_1 \leq \min(x_1, \sqrt{x_2})) & \text{o.w.}
\end{cases}$$
Figure 4.9. $X_2$ vs. $X_1$ for original data
Figure 4.10. $X_2$ vs. $X_1$ for masked data
It follows that the sample joint distribution functions of \((X_1, X_2)\) and \((\hat{X}_1, \hat{X}_2)\) are

\[
\hat{F}_{X_1, X_2}(x_1, x_2) = \begin{cases} 
0 & \text{if } x_1 < -\sqrt{x_2} \\
(300)^{-1} \sum_{t=1}^{300} I(\bar{X}_1 \leq x_1, \bar{X}_2 \leq x_2) & \text{o.w.}
\end{cases}
\]

\[
\hat{F}_{\hat{X}_1, \hat{X}_2}(x_1, x_2) = (300)^{-1} \sum_{t=1}^{300} I(\hat{X}_1 \leq x_1, \hat{X}_2 \leq x_2).
\]

We demonstrate that major differences exist between the two sample distribution functions.

The first major difference lies in the evaluation of \(\hat{F}_{X_1, X_2}(x_1, x_2)\) and \(\hat{F}_{\hat{X}_1, \hat{X}_2}(x_1, x_2)\) for \(-\infty < x_1 \leq 0\). Unless there is a value of \(X_1\) equal to \(x_1\), \(\hat{F}_{X_1, X_2}(x_1, x_2) = 0\), whereas \(\hat{F}_{\hat{X}_1, \hat{X}_2}(x_1, x_2)\) may be quite different from zero. For example, if \(x_1 = -1\), then \(\hat{F}_{X_1, X_2}(-1, 1) = 0\), but \(\hat{F}_{\hat{X}_1, \hat{X}_2}(-1, 1) = 0.0967\).

To illustrate the second major difference, consider any rectangle which lies inside the parabolic curve \(X_2 = X_1^2\). Let the points \((x_1, x_1^2)\), \((x_1, x_1^2 + c)\), \((-x_1, x_1^2 + c)\) and \((-x_1, x_1^2)\) define a rectangle inside the curve. The sample probability that a point \((X_1, X_2)\) falls inside the rectangle given any sample distribution function, \(\hat{F}\), is

\[
\hat{P}(x_1, c) = \hat{F}(x_1, x_1^2 + c) - \hat{F}(x_1, x_1^2) - \hat{F}(-x_1, x_1^2 + c) + \hat{F}(-x_1, x_1^2).
\]
which is equal to zero for $\hat{F} = \hat{F}_{X_1, X_2}$. However, the sample probability is not necessarily zero for $\hat{F}_{X_1, X_2}$. Suppose $x_1 = 1$ and $c = 3$, which defines the rectangle with vertices $(1, 1), (1, 4), (-1, 4)$ and $(-1, 1)$. We know that $\hat{P}_{X_1, X_2}(1, 3) = 0$, but $\hat{P}_{X_1, X_2}(1, 3) = 0.173$.

As a final major difference, consider lower right-hand quadrants containing ordered pairs $(X_1, X_2)$ such that $X_1 \geq x_1 > 0$ and $X_2 \leq x_2 = x_1^2$. The sample probability that a point falls in this quadrant given some $\hat{F}$ is

$$\hat{P}(x_1) = \hat{F}(\infty, x_1^2) - \hat{F}(x_1, x_1^2),$$

which is equal to zero for $\hat{F} = \hat{F}_{X_1, X_2}$, but may be non-zero for $\hat{F}_{X_1, X_2}$. For example, if $x_1 = 0.75$ then $\hat{P}_{X_1, X_2}(0.75) = 0$, but $\hat{P}_{X_1, X_2}(0.75) = 0.0867$.

Thus, we have demonstrated through plots of variables, expressions of sample distribution functions, and in specific instances that the masking algorithm may not preserve joint distribution functions.
5. REFERENCES


6. ACKNOWLEDGEMENTS

I give great thanks to Dr. Wayne A. Fuller for his guidance, encouragement, and enthusiasm throughout the research process and my graduate studies. I also wish to thank my committee members, Yasuo Amemiya, Dale Grosvenor, William J. Kennedy, Glen Meeden, and Debapriya Sengupta. I give special thanks to Yasuo Amemiya for his many helpful comments. I thank the Statistical Research Division of the Census Bureau for support of my research received through Joint Statistical Agreements. I am grateful to Chris Olson for her precise and repeated typings of the manuscript, and to Helen Nelson, the most thoughtful and dependable of secretaries. I must also mention my office mates, Linda Brands, Joel Halvorson, Carol Meeter and Todd Sanger, as well as many other friends who, in one way or another, always made graduate life a bit more enjoyable. Finally, I express my sincerest thanks to my parents, Mr. and Mrs. Patrick J. Sullivan, for their love and support throughout my graduate education.
7. APPENDIX

7.1. An Alternative Prediction Approach

Consider again the confidentiality problem discussed in Section 2.2.1. We assume an intruder has obtained a target record, 

\[ x_0 = \begin{pmatrix} x_{0,1} \\ x_{0,2} \end{pmatrix}^{k \times 1} \]

where \( x_{0,1} \) (\( l \times 1 \)) is known and \( x_{0,2} \) is unknown. We define \( (x_{n_1}, x_{n_2}, \ldots, x_{n_m}) \) to be the microdata release where

\[
\begin{align*}
X_j &= \begin{pmatrix} x_{j,1} \\ x_{j,2} \end{pmatrix} + \begin{pmatrix} u_{j,1} \\ u_{j,2} \end{pmatrix}, \\
X_j &\sim NI \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \Sigma_{xx11} & \Sigma_{xx12} \\ \Sigma_{xx21} & \Sigma_{xx22} \end{pmatrix}, \\
u_j &\sim NI \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \Sigma_{uu11} & \Sigma_{uu12} \\ \Sigma_{uu21} & \Sigma_{uu22} \end{pmatrix}, \\
\end{align*}
\]

and \( x_j \) is independent of \( u_i \) for all \( j, i \in \{1, 2, \ldots, m\} \).

Assuming that the target record corresponds to the \( j \)-th record in the microdata release, the values of the non-confidential variables of the target record may differ from the same variable values in \( x_j \).
j \in \{1, 2, \ldots, m\} by the error introduced to mask the released data record. That error is assumed to be described completely by what Paass (1985) calls the error distribution. Under our model, the error distribution is a multivariate normal distribution with mean 0 and covariance matrix \( \Sigma_{uu} \). The intruder has knowledge of the true values of the first \( \ell \) non-confidential variables, and of the corresponding marginal error distribution. Assuming \( x_0 \) corresponds to \( x_{nj} \), the marginal error distribution of \( u_{0,1} - u_{nj,1} \) is multivariate normal with mean 0 and covariance matrix \( \Sigma_{u0u1} \). The density of the masked vector given the known portion of the target record is

\[
 f(x_{0,1} | x_{0,1}) \\
= f(x_{n_j,1} | x_{0,1} \text{ and } x_{nj} = x_0) \\
= (2\pi)^{-\ell/2}|\Sigma_{u0u1}|^{1/2} \exp\left(-\frac{1}{2} (x_{0,1} - x_{n_j,1})' \Sigma_{u0u1} (x_{0,1} - x_{n_j,1})\right),
\]

for \( j = 1, 2, \ldots, m \). In this expression \( x_{0,1} \) is fixed and \( x_{n_j,1} \) is the random vector.

We assume that the records contained in the microdata release were selected from the confidential sample by simple random sampling. Hence, \( p(j) = N^{-1}m \) where \( p(j) \) is the probability that \( x_j \) was included in the microdata release. Let
for \( j = 1, 2, \ldots, m \). Since \( p(j) \) is equal for all \( j \), we ignore \( p(j) \) in this definition and we suppress other constant multipliers from the multivariate density.

Though not a probability, \( \tilde{\gamma}_{j0} \) is a measure of the likelihood that record \( x_{nj} \) corresponds to the target record. The larger \( \tilde{\gamma}_{j0} \) is relative to the values of \( \tilde{\gamma}_{k0} (k \neq j) \), the greater the estimated probability that \( x_{nj} \) is the target record.

The intruder may then use the values of \( \tilde{\gamma}_{10}, \tilde{\gamma}_{20}, \ldots, \tilde{\gamma}_{m0} \) to obtain probabilities, \( \tilde{p}_{10}, \tilde{p}_{20}, \ldots, \tilde{p}_{m0} \), that the target record corresponds to each record in the microdata release. As in Section 2.2, let

\[
\tilde{p}_{j0} = \left( \sum_{t=1}^{m} \tilde{\gamma}_{t0} \right)^{-1} \tilde{\gamma}_{j0}, \quad j = 1, 2, \ldots, m.
\]  

(7.1.4)

Thus, \( \tilde{p}_{j0} \) is the probability that the record \( x_{nj} \) in the microdata release corresponds to the target record, given that the target record is in the released file. The probability in (7.1.4) is constructed treating \( x_{0,1} \) as fixed. The probability in Section 2.2 was constructed under the assumption that \( x_{0,1} \) is random.

In using the match probabilities, \( \tilde{p}_{10}, \tilde{p}_{20}, \ldots, \tilde{p}_{m0} \), to form a predictor for the unknown component of the target record, we again follow the ideas used in Section 2.2. In the ideal prediction situation where \( \tilde{p}_{k0} = 1 \) for some \( k \in \{1, 2, \ldots, m\} \) and \( \tilde{p}_{j0} = 0 \) for all
j = k, record \( x_{n_k} \) is the target record. Since
\[ X_{n_k,2} - X_{n_k,1} + u_{n_k,2}, \]
we predict \( u_{n_k,2} \) from the knowledge base of
\( \Sigma_{xx}, \Sigma_{uu}, X_{n_k,1}, X_{n_k,2} \) and \( x_{0,1} \). The minimum mean square error
predictor of \( x_{0,2} \) in this case, given \( x_{n_k} = x_0 \), is

\[
\tilde{x}_{0,2} = x_{n_k,2} - X_{n_k,2} - \tilde{u}_{n_k,2}
\]

(7.1.5)

where \( \tilde{u}_{n_k,2} = E(u_{n_k,2}|(X_{n_k}, x_{0,1})) \). We see that

\[
\tilde{u}_{n_k,2} = E(u_{n_k,2}|(X_{n_k,1}, X_{n_k,2}, x_{0,1}))
\]

\[
\begin{pmatrix}
\Sigma_{uu} \\
\Sigma_{xx} \\
0
\end{pmatrix}
\begin{pmatrix}
\Sigma_{x11} + \Sigma_{uu} \Sigma_{x12} + \Sigma_{uu} \Sigma_{x11} \\
\Sigma_{x21} + \Sigma_{uu} \Sigma_{x22} + \Sigma_{uu} \Sigma_{x21} \\
\Sigma_{x11} \\
\Sigma_{x12} \\
\Sigma_{x21}
\end{pmatrix}^{-1}
\begin{pmatrix}
x_{n_k,1} - \mu_1 \\
x_{n_k,2} - \mu_2 \\
x_{0,1} - \mu_1
\end{pmatrix}
\]

(7.1.6)

which is equivalent to the predictor of \( u_{n_k,2} \) given in (2.2.12).

In the general data situation, where the \( P_j \)'s are nearly equal,
the predictor of \( x_{0,2} \) takes the same form as the predictor given in
(2.2.14), namely,

\[
\bar{x}_{0,2} = \sum_{k=1}^{m} p_k \bar{x}_{n_k,2}
\]

(7.1.7)
where \( \tilde{x}_{k,2} = X_{k,2} - \tilde{u}_{k,2} \) and \( \tilde{u}_{k,2} \) is defined in (7.1.6).

7.2. Preserving Confidentiality in the Alternative Approach

We now investigate the problem of selecting an error covariance matrix to use in masking normal data for the prediction approach given in 7.1. As in Section 2.3.1, we fix the ratio of error variance to total variance at \( (1 + \delta)^{-1} \) for all variables. We assume then that, under the model of (7.1.1),

\[
\begin{align*}
X_{n,1} & \sim N_p(\mu_1, \delta I_p), \\
U_{n,1} & \sim N_p(0, \Sigma_{uu11}),
\end{align*}
\]

where \( \delta > 0 \) and

\[
\begin{pmatrix}
1 & \rho_{12} & \rho_{13} & \cdots & \rho_{1\ell} \\
\rho_{12} & 1 & \rho_{23} & \cdots & \rho_{2\ell} \\
\rho_{13} & \rho_{23} & 1 & \cdots & \rho_{3\ell} \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
\rho_{1\ell} & \rho_{2\ell} & \rho_{3\ell} & \cdots & 1
\end{pmatrix}
\]  \hspace{1cm} (7.2.1)

with \( |\rho_{ij}| < 1 \) for all \( i < j \). The data provider chooses \( \rho_{ij} \) \((i < j)\) to minimize \( \tilde{t}_{kk} \). From (7.1.4),
\[
\tilde{p}_{kk} = (\sum_{t=1}^{m} \tilde{\gamma}_{tk})^{-1/2} \gamma_{kk}
\]

\[
\quad = \left[ \sum_{t=1}^{m} \gamma_{kk}^{-1} \gamma_{tk} \right]^{-1/2}
\]

\[
\quad = [1 + \sum_{t=1, t \neq k}^{m} \gamma_{kk}^{-1} \gamma_{tk}]^{-1/2}
\]

where \( \tilde{\gamma}_{tk} \) is defined in (7.1.3). Therefore, minimizing \( \tilde{p}_{kk} \) is equivalent to maximizing \( \Sigma_{t=1, t \neq k}^{m} \gamma_{kk}^{-1} \gamma_{tk} \). Further,

\[
\gamma_{kk}^{-1} \gamma_{tk} = \left[ \gamma_{tk} \gamma_{kk} \right]^{-1/2}
\]

\[
\quad = [p_{tk}^{-1} p_{kk}]^{-1/2},
\]

so maximizing \( \gamma_{kk}^{-1} \gamma_{tk} \) for each \( t \neq k \) is equivalent to minimizing \( p_{tk}^{-1} p_{kk} \) for each \( t \neq k \). Finally, if we take the natural log of this ratio and take the expectation to remove the data dependence, we see that minimizing \( \tilde{p}_{kk} \) is closely related to minimizing the expectation of the log-odds ratio of \( \tilde{p}_{kk} \) to \( \tilde{p}_{tk} \),

\[
E(\ln(\tilde{p}_{kk}) - \ln(\tilde{p}_{tk}))
\]

for randomly chosen \( t \neq k \).

We further simplify the problem by demonstrating that minimizing the expectation of the log-odds ratio of \( \tilde{p}_{kk} \) to \( \tilde{p}_{tk} \) is equivalent to minimizing \( \text{tr}(\Sigma_{uull}^{-1}) \).
Theorem 7.2.1. Assume the microdata release consists of $X_1, \ldots, X_m$ where each $X_j$ satisfies (7.1.1) for $j=1, 2, \ldots, m$. Also, for all $j$,

$$X_{j,1} \sim \mathcal{N}_2(\mu_j, \delta I_2),$$

$$u_{j,1} \sim \mathcal{N}_2(0, \Sigma_{uull}),$$

where $\delta > 0$ and $\Sigma_{uull}$ is defined in (7.2.1). Assume the target record, $x_0$, corresponds to the $k$-th record in the microdata release. Then minimizing $E(\ln(\tilde{p}_{kk}) - \ln(\tilde{p}_{jk}))$, where $\tilde{p}_{jk}$ is defined in (7.1.4), is equivalent to minimizing $\text{tr}(\Sigma_{uull}^{-1})$.

Proof. Assume without loss of generality that $\mu_1 = 0$.

$$E(\ln(\tilde{p}_{kk}) - \ln(\tilde{p}_{jk}))$$

$$= E(-\frac{1}{2} u_{k,1}' \Sigma_{uull}^{-1} u_{k,1} + \frac{1}{2} (X_{j,1} - x_{k,1})' \Sigma_{uull}^{-1}(X_{j,1} - x_{k,1}))$$

$$= -\frac{1}{2} \text{tr}(E[u_{k,1}' \Sigma_{uull}^{-1} u_{k,1}])$$

$$+ \frac{1}{2} \text{tr}(E[X_{j,1}' \Sigma_{uull}^{-1} X_{j,1}] + E[x_{k,1}' \Sigma_{uull}^{-1} x_{k,1}])$$

$$= -\frac{1}{2} \ell + \frac{1}{2} \text{tr}(E[X_{j,1}' X_{j,1}'] \Sigma_{uull}^{-1} + E[x_{k,1}' x_{k,1}'] \Sigma_{uull}^{-1})$$
From Theorem 2.3.3, we know \( \text{tr}(\Sigma_{uull}^{-1}) \) is minimized when \( \Sigma_{uull} = I_{\ell} \). Therefore, as was the case in the totally random approach of Chapter 2, the correct match probability, \( \tilde{p}_{kk} \), is minimized when error vectors having a covariance structure equal to a multiple of the covariance structure of the true data vectors are added to the true data vectors to form the records in the microdata release.

7.3. Differences in the Predictors

In predicting the confidential component of the target values, we have derived the predictors

\[
\hat{x}_{0,2} = \sum_{j=1}^{m} p_{j0} (\hat{x}_{n_j,2} - \hat{u}_{n_j,2})
\]

\[
\bar{x}_{0,2} = \sum_{j=1}^{m} \tilde{p}_{j0} (\bar{x}_{n_j,2} - \bar{u}_{n_j,2})
\]

where \( \hat{u}_{n_j,2}, \bar{u}_{n_j,2} \) are defined in (2.2.12, 7.1.6), and \( p_{j0}, \tilde{p}_{j0} \) are defined in (2.2.9, 7.1.4), respectively. Notice that the difference between the two predictors lies in the way the statistical distances are computed in \( \gamma_{j0} \) and \( \tilde{\gamma}_{j0} \), which are used to define the
match probabilities. Specifically, the difference between the
predictors lies in the way \( \mathbf{x}_{0,1} \) is treated in each of the approaches
to the problem. In the totally random approach of chapter two, \( \gamma_{jk} \) is
constructed from the conditional distribution of \( (\mathbf{x}_{n_1}, \ldots, \mathbf{x}_{n_m}, \mathbf{x}_{0,1}) \)
given \( \mathbf{x}_{n_j} = \mathbf{x}_0 \), namely \( f(\mathbf{x}_{n_1}, \ldots, \mathbf{x}_{n_m}, \mathbf{x}_{0,1} | \mathbf{x}_{n_j} = \mathbf{x}_0) \). In this
distribution, we treat \( \mathbf{x}_{0,1} \) as a random quantity. In the approach
presented in this appendix, \( \tilde{\gamma}_{jk} \) is constructed from the conditional
distribution of \( \mathbf{x}_{n_j,1} \), given \( \mathbf{x}_{0,1} \) and given that \( \mathbf{x}_{n_j} = \mathbf{x}_0 \). In
\( f(\mathbf{x}_{n_j,1} | \mathbf{x}_{0,1} \text{ and } \mathbf{x}_{n_j} = \mathbf{x}_0) \) defined in (7.1.2), \( \mathbf{x}_{0,1} \) is treated as a
fixed, known quantity.

With regard to the difference from a mathematical standpoint, in
the totally random approach,

\[
\gamma_{k0} = \exp\left(-\frac{1}{2}(\mathbf{x}_{0,1} - \mathbf{x}_{n_k})^T \Sigma_{xx}^{-1}(\mathbf{x}_{0,1} - \mathbf{x}_{n_k})\right)
\]

(7.3.3)

where

\[
B = \Sigma_{xx}^{-1} \Sigma_{xX} \Sigma_{xX}
\]

\[
A = \Sigma_{xx1} - \Sigma_{xX} \Sigma_{xx}^{-1} \Sigma_{xX}
\]

\[
\Sigma_{xx1} = E(\mathbf{x}'_1).
\]

In the approach presented above,
Simulations demonstrate that, in attempting to match a target from a sample back to the same masked sample, the match probabilities in the totally random approach tended to be higher for \( x_0 \) close to the center of the distribution. That is, when attempting to match \( x_k, \ k \in \{1, 2, \ldots, m\} \), to one of \( \bar{x}_1, \bar{x}_2, \ldots, \bar{x}_m \), \( p_{kk} \) tends to be higher than \( \bar{p}_{kk} \). We cannot make this claim for target records that are not close to the center of the distribution.

7.4. The Masking Program

We present the computer program based on the masking algorithm discussed in Chapter 3. The program is written in the Interactive Matrix Language (IML) of SAS (1985). The job control language to input and output data is appropriate for the NAS AS/9160 mainframe computer at Iowa State University. The source code consists of fourteen subroutines followed by a main driver routine. Though the entire program is documented, many references are made to the description of the masking algorithm in Chapter 3. An understanding of SAS, especially the IML procedure, would also be helpful in deciphering the source code.
A\nA Input the data: the INPUT statement must be adjusted depending on the number of input variables.

DATA SET1;
  INFILE IN1;
  INPUT X1 - X4;

* * * MASKING PROGRAM: This program will take a file of records, transform and mask each by adding an independent normal vector of measurement error to each vector observation in the transformed data set. Afterwards, the data are transformed back to their original scale. The objectives of the procedure are as follows:

(1) Mask the data in such a way that if all possible distances between true and masked data vectors are computed, no masked vector is closest to its true data vector.

(2) After the data are masked, the means and the covariance structure of the masked variables are nearly the same as the means and the covariance structure of the original variables.

(3) The univariate distribution function of each masked variable is nearly the same as the univariate distribution function of the corresponding original variable.

To attain these goals, we first compute the sample cumulative distribution function of each variable. We then compute a linearly interpolated continuous distribution. We take the points that the actual values map to on this function (Note: \( F(x) \) is distributed as \( U(0,1) \)) and transform them to standard normal values using the inverse of the standard normal distribution function. After the variables are transformed, error vectors with correlation structure nearly equal to the correlation structure of the transformed values are added to the transformed
data. The program is iterative in that the correlation structure of the error vector is adjusted after each mask in an attempt to attain a certain correlation structure for the masked variables. The first iteration is different from other iterations in that adjustments are made to insure the distance requirement. After the data vectors are masked, the values are transformed back to original scale using the linearly interpolated distribution function. Additional iterations are performed which should result in originally scaled masked variables with nearly the same correlation structure and univariate distribution functions as the original variables.

PROC IML;

SUBROUTINES: FINDCELL, PROPORTN, CONDTNAL, PSEUDO, ROUND, CDF, DISTINCT, LINTERP, ERRVEC, UNCORR, DISTANCE, FTILDA, PSPACE, ZTRANS.

(See each routine for its purpose and use.)

SUBROUTINE CODE

SUBROUTINE FINDCELL: Stores and/or determines the position of storage of a categorical value of a variable that is to be masked.

START FINDCELL(MAXCAT,CAT,J,ICELL,CLASS);

Given a categorical value, CAT, of the J-th classification variable, store the value in the next available cell in the J-th row of class and return the column number of storage in ICCELL.

If the value is already present, simply return its position in ICCELL.

DO K=1 TO MAXCAT;

IF CLASS(|J,K|)=-1 THEN DO;

Value not yet stored in CLASS - store and return ICCELL=k;

CLASS(|J,K|)=CAT;
ICELL=K;
GO TO CFOUND;
END;

ELSE DO;

* Value present in CLASS - return ICELL=K;

IF CLASS(J,K)=CAT THEN DO;
    ICELL=K;
    GO TO CFOUND;
END;

END;
END;

CFOUND:
FINISH;
*----------------------------- END FINDCELL -----------------------------;

*---------------------------------------------------------------

SUBROUTINE PROPORTN: Compute the sample proportions for
each of the categories of the classification variables
that will be masked.

*---------------------------------------------------------------

START PROPORTN(N,NC,TYPE,C,CLASS,PHAT);

* For each record and for each classification variable, determine
* where the categorical value is stored and increment the cor-
* responding count in PHAT. After all records are investigated,
* compute the sample proportions by dividing PHAT by N.;

DO I=1 TO N;
    JCAT=0;
    DO J=1 TO NC;

    NEXTVAR:

* Determine if the variable is a classification variable. If
* not, go on to next variable.;

    JCAT=JCAT+1;
    IF TYPE(J,2)=0 THEN GO TO NEXTVAR;

* Variable is of classification type - find location of category.;

    MAXCAT=TYPE(J,2);

    DO 1=1 TO N;

    JCAT=0;
    DO J=1 TO NC;

    NEXTVAR:

    END;

    END;

    END;

    CFOUND:
    FINISH;
*----------------------------- END FINDCELL -----------------------------;

*---------------------------------------------------------------
RUN FINDCELL(MAXCAT,C(I,J),J,ICELL,CLASS);
PHAT(J,ICELL)=PHAT(J,ICELL)+1;
END;
END;

PHAT=PHAT/N;
FINISH;

SUBROUTINE CONDTNAL: Assign 0-1 values to the pseudo variables created for the j-th variable of the i-th record.

START CONDTNAL(I,J,ICELL,JCOL,JCAT,XP,PHAT,TYPE);

* This routine is called if the category does NOT correspond to one of the last two values stored in J-th row of CLASS. That is, ICELL is equal to one of the first (NCX-1) categories, where NCX is one less than the total number of categories of the J-th classification variable. (See Section 3.1.1 of thesis.) Hence, we assign 0-1 values to the pseudo variables of categories (ICELL+1) to (NCX-1) based on conditional probabilities.;

* PHIDEN is the denominator value used to compute the conditional probability PHI. NCON is the number of pseudo variables which must be assigned values and UNFM is an array (NCON x 1) of uniform values which will determine whether each pseudo variable is assigned a 0 or 1 value.;

NCX=TYPE(JCAT,2)-1;
PHIDEN=1;
NCON=NCX-ICELL;
UNFM=J(NCON,1,0);

* Compute the denominator for PHI for the first pseudo variable.;

DO K=1 TO ICELL;
   PHIDEN=PHIDEN-PHAT(J,K);
END;

UNFM=RANUNI(J(NCON,1,0));

* For each pseudo variable, compute PHI where PHI = Prob{ cell L | Not cells 1,2,...,L-1}. NN defines the Uniform(0,1) cell to investigate: If the value
* in this cell of UNFM is less than PHI, assign 1 to the pseudo
* variable, XP(I,L+JCOL). Then subtract PHAT(J,L) to determine
* the denominator for PHI for pseudo variable L+1.;

DO L=ICELL+1 TO NCX;
    PHI=PHAT(|J,L|)/PHIDEN;
    NN=L-ICELL;
    IF UNFM(|NN,1|) < PHI THEN XP(|I,L+JCOL|)=1;
    PHIDEN=PHIDEN-PHAT(|J,L|);
END;
FINISH;

SUBROUTINE PSEUDO: Create pseudo variables for each catego-
rical value of each classification variable as described
in Section 3.1.1 of thesis.

START PSEUDO(C,XP,PHAT,N,NPSEUD,NC,CLASS,TYPE);

DO I=1 TO N;

* Given record I, create pseudo variables for one less than the
* the total number of categories of each classification variable.
* JCOL determines the starting column for the pseudo variables
* of class variable J in the XP (X-Pseudo) matrix. JCAT indexes
* the TYPE array so the total number of categories for
* classification variable J is known.;

    JCOL=0;
    JCAT=1;
    DO J=1 TO NC;

        CATNO:
            IF TYPE(|JCAT,2|)=0 THEN DO;
                JCAT=JCAT+1;
                GO TO CATNO;
            END;

        * Determine the location of the category in the J-th row of CLASS.
        * If ICELL corresponds to the last category, then all pseudo vars
        * are zero - go to the next classification variable.;

            MAXCAT=TYPE(|JCAT,2|);
            RUN FINDCELL(MAXCAT,C(|I,J|),J,ICELL,CLASS);
            IF ICELL=TYPE(|JCAT,2|) THEN GO TO NEXTCX;
Otherwise, compute JCELL, the column in XP whose value will be unity by definition. Then, every category up to (and including) the penultimate category has a 0-1 pseudo variable assigned it. (See SUBROUTINE CONDTNAL for details.);

\[
\text{JCELL}=\text{JCOL}+\text{ICELL};
\]
\[
\text{XP}(|\text{I},\text{JCELL}|)=1;
\]
\[
\text{IF } \text{ICELL} < \text{TYPE}(|\text{JCAT},2|)-1 \text{ THEN RUN CONDTNAL}(\text{I},\text{J},\text{ICELL}, \text{JCOL},\text{JCAT},\text{XP},\text{PHAT},\text{TYPE});
\]

When all pseudo variables for the J-th classification variable have been assigned, adjust JCOL and increment JCAT to prepare for the next classification variable.;

\[
\text{NEXTCX:} \\
\text{JCOL}=\text{JCOL}+\text{TYPE}(|\text{JCAT},2|)-1; \\
\text{JCAT}=\text{JCAT}+1;
\]

FINISH;

-----------------------------------------------------

**SUBROUTINE ROUND:** Rounds a real number to the nearest integer value.

**START ROUND(XREAL,INTGR);**

\[
\text{XUNIF}=\text{ABS}(\text{XREAL}-\text{INT}(\text{XREAL}));
\]
\[
\text{IADD}=0;
\]
\[
\text{IF } \text{XUNIF} >= 0.5 \text{ THEN } \text{IADD}=1;
\]
\[
\text{IF } \text{XREAL} > 0 \text{ THEN } \\
\text{INTGR}=\text{INT}(\text{XREAL})+\text{IADD}; \quad \text{ELSE} \\
\text{INTGR}=\text{INT}(\text{XREAL})-\text{IADD};
\]

FINISH;

-----------------------------------------------------

**SUBROUTINE CDF:** Given N scalar observations (responses to variable I), find the sample Cumulative Distribution Function of the data.
START CDF(II,N,XI,X,R,F);

* Rank the values of the II-th variable in ascending order.
* Assign probability 1/N to each value, then compute the
* sample CDF;

R(|II|)=RANK(X(|II|));
XI(|R(|II|)|)=X(|II|);
XLAST=XI(|1|);
KLAST=1;

* The values are ordered. Traverse through the values assigning
* distribution function values at each distinct point. (The
* value of the sample CDF at, say, x(i) is the number of values
* <= x(i) divided by N.);

DO K=2 TO N;
  IF XLAST < XI(|K|) THEN DO;
    DO L=KLAST TO K-1;
      F(|L,II|)=(K-I)/N;
    END;
    XLAST=XI(|K|);
    KLAST=K;
  END;
END;

* Assign unity to the sample CDF at the highest value(s);

DO K=KLAST TO N;
  F(|K,II|)=1.0;
END;
FINISH;
*-------------------------------- END CDF -----------------------------------;

*___________________________________________

SUBROUTINE DISTINCT: Eliminate duplicate values from the
* i-th column of X and its corresponding column in F.
*____________________________________________

START DISTINCT(N,II,M,XS,FS,XI,F);

* Create a subsample (of the original values) consisting of only
* distinct values. Store the values in XS and their correspon-
ding CDF values in FS.;

FS(|1,|)=0;
FS(|2,|)=F(|1,II|);
XS(|2,|)=XI(|1,|);
M=2;

DO L=2 TO N;

IF XI(|L,|) > XS(|M,|) THEN DO;
    M=M+1;
    XS(|M,|)=XI(|L,|);
    FS(|M,|)=F(|L,II|);  
END;
END;

* M counts the total number of distinct values plus
two: Define XS(1) as the smallest distinct value less
the difference of the two smallest distinct values.
Define XS(M) as the largest distinct value plus the
difference between the two largest distinct values.
Assign 0 as the value for the distribution function for XS(1).
XS(M) need not have a corresponding value for the distribution
function. (The reason for defining XS(1), XS(M) and FS(1)
becomes apparent in the formula for computing a continuous
distribution function in the LINTERP subroutine.);

XS(|1,|)=(XS(|2,|)-(XS(|3,|)-XS(|2,|)));  
M=M+1;
XS(|M,|)=(XS(|M-1,|)+(XS(|M-1,|)-XS(|M-2,|)));  

FINISH;

\*-------------------------------- END DISTINCT --------------------------------;

\*-------------------------------------- SUBROUTINE LINTERP: Use linear interpolation to estimate
a continuous cumulative distribution function. Upon evaluating the responses on this continuous CDF, the
the resultant values can be treated as Uniform(0,1)
random variables. (See Section 3.1.2 of thesis for
mathematical details on the estimation of the
continuous CDF.)

\*--------------------------------------

\* The sample CDF corresponding to the I-th variable is a step
function. A continuous function is formed from this step
* function if the midpoints of all the "steps" are connected.
* FX stores the values obtained when the N observations for the
* I-th variable are evaluated on this continuous function.;

START LINTERP(N,II,M,XS,FS,XI,FX);

L=1;

* Compute FHAT for each of the M distinct value responses to
* variable I. FHAT = F(x(k)) + (Slope) * (Delta x) where
* slope is (|F(x(k+1))-F(x(k))|)/{0.5*(|x(k+2)-x(k)|) and
* Delta-x is 0.5*(|x(k+1)-x(k)|)}. The equation reduces to
* FHAT (below) in matrix code.;

DO K=1 TO M-2;

* Notice for K=1 how XS(1) & FS(1) enter into the formula. Also,
* for K=M-2, XS(M) is used to compute FHAT but FS(M) is not
* necessary. (This relates back to the defining of XS and FS
* in DISTINCT.);

FHAT=FS(|K,|)+((FS(|K+1,|)-FS(|K,|))*(XS(|K+1,|)-XS(|K,|)) /
(XS(|K+2,|)-XS(|K,|)));

* Assign FHAT to every value identical to the k-th distinct
* value.;

LOOP:
FX( |L,|)=FHAT;
L=L+1;
IF L < N THEN DO;
   IF XI( |L+1,|)=XI( |L,|) THEN GOTO LOOP;
END;
IF L > N THEN GOTO ENDLINT;
END;

ENDLINT:
FINISH;

*---------------------------------------------------------------;

<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBROUTINE ERRVEC: Generates a P x 1 vector of standard</td>
</tr>
<tr>
<td>normal error whose components satisfy a vector length or</td>
</tr>
<tr>
<td>dot product restriction.</td>
</tr>
<tr>
<td>---------------------------------------------------------------</td>
</tr>
</tbody>
</table>

START ERRVEC(UT,LIM,ENDLIM,ILOW,IUPP,P,ITRY,ABSLIM,SSQLIM);
UT is a Pxl vector initially set to zero. This routine will assign randomly generated standard normal observations to UT five cells at a time. However, the five values (forming a 5 x 1 vector) must have a dot product of SSQSIM or more. Also, the absolute value of each individual error term must exceed ABSIM. Otherwise, a new set of five values is generated until the restrictions are satisfied.

LIM is the number of variables divided by five with the remainder truncated. If LIM is 0, the number of variables is < 5. This case is handled at the end of the routine.

IF LIM=0 THEN GOTO LESS5;

DO I=1 TO LIM;

* Generate sets of 5 normal observations. If they do not satisfy the length requirement, repeat the process.

REGEN:
UTH=RANNOR(J(5,1,0));
DO K=1 TO 5;

ABSCHK:
IF ABS(UTH(|K,|)) <= ABSIM THEN DO;
UTH(|K,|)=RANNOR(J(1,1,0));
GOTO ABSCHK;
END;

END;

D=UTH*UTH;
IF D <= SSQSIM THEN GOTO REGEN;

LIX & UIX are the lower and upper indices in UT to which the current set of five numbers correspond. Store the generated error terms in cells LIX thru UIX of UT.

UIX=5*I;
LIX=5*(I-1)+1;
UTH(|LIX:UIX,|)=UTH;
END;

* If P is not a multiple of five, the last P-ENNDLIM number of cells need only satisfy the individual absolute value not meet requirement (> ABSIM). So, generate the four or less standard normal observations and store in the last cells of UT.

LESS5:
IF ENNDLIM < P THEN DO;
NREM=P-5*LIM;
UTH=J(NREM,1,0);
UTH(|1:NREM,|)=RANNOR(J(NREM,1,0));
SUBROUTINE UNCORR: Insures nearly uncorrelated error vectors by adjusting the signs of the terms of the t-th generated error vector depending on the mean vector and covariance matrix of the previous (t-1) error vectors. (See Section 3.2.1 of thesis for details.)
SUBROUTINE DISTANCE: Determine whether or not the L-th
masked observation satisfies the distance requirement.

START DISTANCE(INDEC,L,N,SIGMA,Y,YSTAR,MINDX1,MINIX1,MINDX2,MINIX2,
DLL,P,ITRY);

* This routine is called for each observation in the sample, one
* after another. For observation L, the statistical distance be-
* tween the masked and original observation is computed and stored
* in DLL. To satisfy the masking requirement, when the distances
* between the L-th original observation and all other masked
* observations is computed, at least one distance must be exceeded
* by DLL.:

MINDX1=999998.0;
MINIX1=0;
MINDX2=999999.0;
MINIX2=0;
INDIC=0;
DLL=(Y( |L, |)-YSTAR( |L, |))*INV(SIGMA)*(Y( |L, |)-YSTAR( |L, |)));

* DKL is the statistical distance between the L-th original
* observation vector and the K-th masked observation vector.
* The second smallest distance is stored in MINDX (and the number
* record corresponding to it in MINIX). This information is
* needed if masking with random error is unsuccessful in
* maintaining distance between original records and their
* masked counterparts.:

DO K=1 TO N;
  IF K=L THEN GOTO SKIP;
  DKL=(Y( |L, |)-YSTAR( |K, |))*INV(SIGMA)*(Y( |L, |)-YSTAR( |K, |));
  IF DKL < DLL THEN INDIC=INDIC+1;
  IF DKL < MINDX THEN DO;
    MINDX2=MINDX1;
    MINIX2=MINIX1;
    MINDX1=DKL;
  END;

IF DKL < MINDX THEN DO;
  MINDX2=MINDX1;
  MINIX2=MINIX1;
  MINDX1=DKL;

END;
MINIX1=K;
END;
ELSE DO;
IF DKL < MINDX2 THEN DO;
  MINDX2=DKL;
  MINIX2=K;
  END;
END;
IF INDIC = 2 & ITRY < 9 THEN GOTO PASS;
SKIP:
END;
PASS:
FINISH;
*------------------------------------- END DISTANCE ------------------------------------*

SUBROUTINE FTILDA: Transform the U(0,1) values (i.e. the probabilities associated with the masked standardized values) for the I-th variable back to originally scaled X-values by using the inverse of the linearly interpolated continuous distribution function.

START FTILDA(I1,N,XS,FS,FI,XSTAR);

* FI contains the U(0,1) values that are to be mapped back to the x-values. FS(L,) contains the resultant function values obtained by evaluating the unique observed values for the I-th variable on the sample cumulative continuous distribution function. XS contains the (ordered) distinct values for the I-th variable. L indexes the distinct values in XS.;

L=1;
DO K=1 TO N;
ADDONE:

* If FS(L) <= FI(K) < FS(L+1), then, using simple "point-slope" arithmetic, given two points - (XS(L),FS(L)) & (XS(L+1),FS(L+1)) - determine the value of x - namely, XSTAR - that maps to FI(K) on the line formed by the above pair of points. XSTAR is then the originally scaled, masked value.;
IF $FI(|K,|) \geq FS(|L,|)$ & $FI(|K,|) < FS(|L+1,|)$ THEN

\[
XSTAR(|K,II|) = 0.5'Kxs(|L,|)+xs(|L+1,|))+
(0.5'KFI(|K,|)-FS(|L,|)^{*}
(XS(|L+2,|)-XS(|L,|))/(FS(|L+1,|)-FS(|L,|)));
\]

ELSE DO;

$$L=L+1;$$
GOTO ADDONE;

END;

END;

FINISH;

* END FTILDA -----------------------------------------------;

SUBROUTINE PSPACE: (Call only for TYPE = 1 & 2 variables.)
Take each masked, properly scaled value for variable I and verify that it falls in the restricted range. If not, adjust the value so it meets the specified restrictions.

START PSPACECN,11,NRI,TYPE,XSTAR,XBARII,RESTRICT);

* For each value, set XTEM - the value currently in question - to the proper variable type: (0,1) for Bernoulli (TYPE=3), integer for discrete (TYPE=2), or real for continuous (TYPE=1).
* Then, check the restriction (if one exists) for the variable of the value.;

* II is the variable whose masked values are being checked for restrictions, NRI is the number of the restriction currently being considered, and XBARII is the mean of the original variable if it happens to be a Bernoulli variable.;

DO K=1 TO N;

IND=0;
IF TYPE(|II,1|)=3 THEN DO;
XTEM=XSTAR(|K,II|);
IF XTEM < (1.0-XBARII) THEN XTEM=0;
ELSE XTEM=1.0;
END;
IF TYPE(|II,1|)=2 THEN RUN ROUND(XSTAR(|K,II|),XTEM);
IF TYPE(|II,1|)=1 THEN XTEM=XSTAR(|K,II|);

CHECK:
IND indicates whether the current value meets the restriction.
Different inequalities must be checked depending on the type
of restriction (RESTRICT(...) ). If the value is okay, set IND
to one. Then go to the adjustment part of the routine (INCR).

IF RESTRICT( |NRI,2| )=1 THEN DO;
    IF XTEM >= RESTRICT( |NRI,3| ) & XTEM <= RESTRICT( |NRI,4| )
        THEN IND=1;
    GO TO INCR;
END;
IF RESTRICT( |NRI,2| )=2 THEN DO;
    IF XTEM >= RESTRICT( |NRI,3| ) THEN IND=1;
    GOTO INCR;
END;
IF RESTRICT( |NRI,2| )=3 THEN DO;
    IF XTEM <= RESTRICT( |NRI,4| ) THEN IND=1;
    GOTO INCR;
END;

INCR:

* If IND is unity, set XSTAR and consider the next value.;

IF IND=1 THEN GOTO NEXTOB;

* The current value does not fall in the restricted range. Set
  XTEM to the closest endpoint of the restriction interval.;

IF RESTRICT( |NRI,2| )=1 THEN DO;
    IF XTEM < RESTRICT( |NRI,3| ) THEN
        XTEM=RESTRICT( |NRI,3| );
    ELSE IF XTEM > RESTRICT( |NRI,4| ) THEN
        XTEM=RESTRICT( |NRI,4| );
END;
IF RESTRICT( |NRI,2| )=2 THEN
    XTEM=RESTRICT( |NRI,3| );
IF RESTRICT( |NRI,2| )=3 THEN
    XTEM=RESTRICT( |NRI,4| );

* Finally, set XSTAR(K,I) - the released, masked k-th value for
  variable I - to XTEM.;

NEXTOB:
    XSTAR(|K,II|)=XTEM;
END;

NRI=NRI+1;

FINISH;
*--------------------------- END PSPACE ---------------------------;
SUBROUTINE ZTRANS: Converts the set of pseudo variables created to mask the II-th variable (or the INC-th classification variable) back to a categorical value in the variable's original range (for all N responses).

START ZTRANS(N,II,JCOL,INC,TYPE1,CLASS,XSTAR,CSTAR);

DO I=1 TO N;

* For the I-th record, take the values of the pseudo variables to determine the masked value of the response to II-th variable.
* JCOL+1 is the first pseudo variable in the set created for the INC-th classification variable.

DO K=JCOL+1 TO JCOL+TYPE1(II,2)-1;

* Determine the first pseudo variable in the set that has a value of one. Subtracting JCOL from the number of this variable gives the column in CLASS (the row is INC) whose cell contains the categorical value that will be the masked value for the original response value. (See Section 3.4 of thesis.);

IF XSTAR(I,K)=1 THEN DO;

CSTAR(I,INC)=CLASS(INC,K-JCOL);
GOTO NEXTREC;
END;
END;

* If none of the pseudo variables has the value unity, then the last categorical value of the classification variable (which has no corresponding pseudo variable) will be the masked value for the original response value.

CSTAR(I,INC)=CLASS(INC,TYPE1(II,2));
NEXTREC:
END;

FINISH;

************************************************* END ZTRANS *************************************************;

******************** MAIN ROUTINE ********************

START;

* Read variables into the matrix X;
USE SET1;
READ ALL VAR {X1 X2 X3 X4} INTO X;

* Set the current library so large matrices can be stored
during intermediate calculations;

RESET STORAGE=LMASK.MEM1;

/* KEY VARIABLES */

N --> Number of observations
P --> Number of variables per observation
ICORR --> The counter of masking iterations to equate
the cross correlations.
NCORR --> The counter of masking iterations to equate
the correlation matrices of the original and masked
data sets.
LIMICORR --> Maximum number of iterations permitted in
attaining the desired cross correlations.
LIMNCORR --> Maximum number of iterations permitted in
attempting to equate correlation matrices.
TYPE --> Specifies (1) the variable type -- continuous,
discrete, classification, and (2) the number of
categories for classification variables. (See below)
RESTRICT --> Specified by user, contains any restrictions
that must be met by continuous or discrete variables
that are being masked. (See below)
NR --> The total number of restrictions specified by the
user in RESTRICT.
NBER --> The number of Bernoulli variables being masked.
X --> Matrix in which observation vectors are initially
stored.
Z --> Matrix of transformed observation vectors (Nxp).
(See documented code below for description of
transformation.)
XALL --> Matrix of original data and masked data used in
calculation of correlation matrices before each
iteration to equate correlations. XALL is the
concatenation of the original data matrix an the
masked data matrix.
XSTAR --> Matrix of masked original observation vectors.
ZSTAR --> Matrix of masked transformed observation vectors.
F --> The j-th column contains the sample cumulative dis-
tribution function of the j-th variable in the data
set. So, F(i,j)=F(z) where z=X(i,j).
R --> The j-th column stores the ranking or ordering of
all values in the j-th column of X. So, if R(i,j)=1
then \( X(i,j) \) is the smallest value in column \( j \). \( R \) is used to order and re-order the original observations \( F \) can be computed. Hence, allowing \( Z, Z^{\text{STAR}}, \) and \( XS^{\text{ATR}}, \) to be determined.

----------

NOTE: The following variables are relevant only if classification variables (c-vars) are present in the input data.

- \( XO \) -- Stores data column vectors corresponding to continuous or discrete variables.
- \( C \) -- Stores data column vectors corresponding to classification variables.
- \( XP \) -- Matrix of column vectors representing pseudo variables created to mask classification variables.
- \( POSIT \) -- Array which stores the original ordering of the columns (or variables) as opposed to their ordering the new \( X \) matrix. \( POSIT \) is used to then re-order the masked variables before they are output.
- \( NC \) -- Total number of classification variables.
- \( NPSEUD \) -- Total number of pseudo variables created to mask the classification variables.
- \( PO \) -- Stores the original number of variables to be masked, while \( P \) is reset to the number of discrete and continuous variables plus the number of pseudo variables.
- \( CLASS \) -- Two dimensional array, row \( j \) stores the categorical values of the \( j \)-th classification variable as the categories are encountered in traversing through the records. (e.g. If values 3, 1, and 2 are values of 1st 3 records for variable \( j \), then \( CLASS(j,1)=3, CLASS(j,2)=1, \) and \( CLASS(j,3)=2. \))
- \( PHAT \) -- Two dimensional array (same as \( CLASS \)), row \( j \) stores the sample proportions of the categories of the \( j \)-th classification variable. The cells in \( PHAT \) correspond directly to the cells in \( CLASS \), which contain the categorical values.
- \( TYPEl \) -- Same structure and purpose as \( TYPE \) applied to the new \( X \) matrix (i.e. the one with pseudo variables in place of class-variables.)

----------

NOTE: Other KEY VARIABLES are documented within the subroutines.

----------

I. Defining the Masking Specifications

----------
N=NROW(X);
P=NCOL(X);
ICORR=0;
NCORR=0;
LIMICORR=10;
LIMNCORR=10;

* Define TYPE for each variable (1,...,P) as follows:
* TYPE(j,1) = 1 --> Mask as a continuous variable
* = 2 --> Mask as a discrete variable (e.g. Number
* of children in a family can be 0,1,2,...
* The numbers represent quantities.)
* = 3 --> Mask as a classification variable (The
* numbers represent categories.)
* TYPE(j,2) = 0 --> Set to zero if TYPE(j,1) is 1 or 2
* = K --> K>1 is the number of categories which the
* classification variable can take on.;

TYPE= {1 0 , 1 0 , 2 0 , 3 4};

* Enter restrictions on any variables coded as "1" or "2" in
* column one of the TYPE array.
* NR --> The total number of restrictions
* RESTRICT --> NR x 4 matrix with columns defined as follows:
* 1: Variable number to which the restriction applies
* 2: Restriction Code:
* 1 --> "Finite" interval [a,b]
* 2 --> Right "infinite" interval [a, infinity)
* 3 --> Left "infinite" interval (-infinity, b]
* 3: Left endpoint of the interval (see above)
* 4: Right endpoint of the interval (see above)

* NOTE: For TYPE 1 variables, only one interval restriction is
* permitted. For TYPE 2 variables, one and only interval
* is required. If the interval is infinite, enter "99999"
* for infinity. All intervals are assumed to be closed.
* To specify the open interval (a,infinity), enter
* "a + 1.0E-8" as the left endpoint.

* Example: Variable whose values are nonnegative (see below).;

RESTRICT= { 3 2 0 99999};
NR=1;
NBER=0;
NC=1;

* Set ALPHA, the fraction of error to be added to the transformed
* normal values in the mask. (See Section 3.2.2 of thesis.)

ALPHA=0.3;
* NOTE: All pseudo variables created to mask classification
* variables are 0-1 variables. Hence, no restrictions
* are entered for type "3" variables.;

* Program Initializations -- DO NOT ALTER!!;

II. Transforming Categorical Variables to
Sets of 0-1 Pseudo Variables

In the case of no class-variables, set XALL to X and
bypass code defining POSIT, TYPE1, CLASS and PHAT.;

IF NC = 0 THEN GOTO NOMULT;

* Initialize POSIT & TYPE1, and set NPSEUD. ICOLC
* and ICOLX serve as column indices for defining C
* and XO. JTYPEC indexes the current row of TYPE1
* and JTYPEX indexes the current row of XO.;

XO=J(N,P-NC,0);
C=J(N,NC,0);
POSIT=J(P,1,0);
NPSEUD=TYPE(|+,2|)-NC;
TYPE1=J(P-NC+NPSEUD,2,0);
RSTRICT1=J(NR+NPSEUD,4,0);
IF NR > 0 THEN RSTRICT1(|1:NR,|)=RESTRICT;
ICOLC=1;
ICOLX=1;
JTYPEX=1;
JTYPEC=NC+1;

* For each variable, store the column responses in C or
* XO, and define the current row of TYPE1 and POSIT.;

DO I=1 TO P;

* For class-variables, take the column of responses and store in
* C and store the position of that variable in the original X
* matrix. Also, set the JTYPE row of TYPE1 as corresponding to
* a class variable with two categories (0 and 1).;

IF TYPE(|I,1|)=3 THEN DO;
C(|I,ICOLC|)=X(|I,|I|);
POSIT( |ICOLC+P-NC,|)=I;
ICOLC=ICOLC+1;
DO J=1 TO TYPE(|I,2|)-1;
TYPE1(|JTYPEC,1|)=3;
TYPE1(|JTYPEC,2|)=2;
RSTRICT1(NR+1,1)=JTYPEC;
RSTRICT1(NR+1,2:4)={ 1 0 1 };
JTYPEC=JTYPEC+1;
NR=NR+1;
END;
END;

* For discrete or continuous variables, store the column of responses in XO, the position of the variable in POSIT and set TYPE1 for the current column.;
ELSE DO;
   XO(,,ICOLX)=X(,,I);
   POSIT(I,,ICOLX)=I;
   ICOLX=ICOLX+1;
   TYPE1(I,,JTYPEX,1)=TYPE(I,I);
   JTYPEX=JTYPEX+1;
END;
END;

* Initialize PHAT and CLASS as 2-dim arrays, NC x MAXCAT (MAXCAT is the max number of categories of any class-var.). The cells of CLASS are initialized at "-1" since "0" may be a category for some variable.;

MAXCAT=MAX(TYPE(I,2));
PHAT=J(NC,M,AXCAT,0);
CLASS=J(NC,M,AXCAT,-1);

* Compute sample proportions;
RUN PROPORTN(N,NC,TYPE,C,CLASS,PHAT);

* Initialize XP, the array of pseudo values, and call PSEUDO to define the pseudo variables of the class-variables.;
XP=J(N,NPSEUD,0);
RUN PSEUDO(C,XP,PHAT,N,NPSEUD,NC,CLASS,TYPE);
FREE C;

* Since later code always recognizes TYPE as storing types of variables, switch the locations of storage. Also, for the same reason, redefine PO & P as well as RESTRICT & RSTRICT1.;
TEMP=TYPE;
FREE TYPE;
TYPE=TYPE1;
FREE TYPE1;
TYPE1=TEMP;
FREE TEMP;
TEMP=RESTRICT;
FREE RESTRICT;
RESTRICT=RESTRICT1;
FREE RESTRICT1 TEMP;

PO=P;
P=PO-NC+NPSEUD;
NBER=NBER+NPSEUD;

* The values which are to be masked can now be stored in X.;
X=XO||XP;
FREE XO XP;
XOLD=X;
STORE XOLD;

* If there are no classification variables, the program
* skips to the "NOMULT" label and computations continue.;

NOMULT:

*---------------------------------------------------------------------*
* III. Transforming Data to Standard Normality *
*---------------------------------------------------------------------*

* Initialize F, R and Z.;
STORE X;

*---------------------------------------------------------------------*
* III. A. Transformation of Continuous Variables *
*---------------------------------------------------------------------*

F=J(N,P,0);
R=J(N,P,0);
STORE F R;
Z=J(N,P,0);
STORE Z;
XI=J(N,1,0);
XS=J(N+2,1,0);
FS=J(N+2,1,0);
FX=J(N,1,0);

* Transform the data values to standard normal values - first,
* the quantitative variables.;
DO II=1 TO P;
LOAD X;

* If the current variable is of the Bernoulli or pseudo type,
* it will be transformed after the quantitative variables.;

IF TYPE(|II,1|)\neq 3 THEN GOTO NEXTCDF;

* For each variable (II), call CDF to compute the sample distri-
* bution function and store in F. Then, call the subroutines
* DISTINCT and LINTERP. These routines will essentially evaluate
* the values of variable II on the linearly interpolated continu-
* ous distribution function. The resulting values (which fall
* between 0 and 1, NOT inclusive) are stored in FX. They are
* finally transformed to normal values by using the inverse of
* the standard normal distribution function (PROBIT).;

LOAD R F;
RUN CDF(II,N,XI,X,R,F);
STORE X R;
RUN DISTINCT(N,II,M,XS,FS,XI,F);
STORE F;
RUN LINTERP(N,II,M,XS,FS,XI,FX);
LOAD R Z;
Z(\{I,II\})=PROBIT(FX(\{R(\{I,II\}),\}));
STORE R Z;

NEXTCDF:

END;

IF NBER=0 THEN GOTO GENERR;

LOAD X Z;

*------------------------------------------------------------------------*
* III. B. Transformation of 0-1 Variables                                 *
*------------------------------------------------------------------------*

* Transform all 0-1 variables into standard normal values.
* This transformation will retain the correlation structure
* of the original variables.;

* First, compute the sample covariance matrix of the original
* data. Then, partition this matrix into submatrices based on
* whether the corresponding variables are quantitative or 0-1.;

XBAR=X(\{+,\}/N;
MXX=(Xc\times X-N\times XBARc+XBAR)/(N-1);
IVX12 = INV(DIAG(SQRT(VECDIAG(MXX))));
RHOX = IVX12 * MXX * IVX12;
PRINT XBAR, MXX, RHOX;
MXX11 = MXX(1:P-NBER, 1:P-NBER);
MXX12 = MXX(1:P-NBER, P-NBER+1:P);
MXX21 = MXX(P-NBER+1:P, 1:P-NBER);
MXX22 = MXX(P-NBER+1:P, P-NBER+1:P);

* Regress the 0-1 variables on the quantitative variables (BDC)
* and compute the conditional variance of the 0-1 variables given
* the quantitative variables (MDDCC);
BDC = MXX21 * INV(MXX11);
MDDCC = MXX22 - MXX21 * INV(MXX11) * MXX12;

* Generate N independent (NBER x 1) vectors of standard normal
* values with identity covariance matrix;
EDC1 = RANNOR(J(N, NBER, 0));
EDCBAR = EDC1(1:N)/N;
MEE = (EDC1 * EDC1 - N * EDCBAR * EDCBAR) / (N - 1);
TEE = ROOT(MEE);
EDC = (EDC1 - (J(N, 1, 1)) * EDCBAR) * INV(TEE);

* Compute FDC1 which has an approximate normal distribution with
* mean zero and variance MXX22 (same as original covariance
* between the 0-1 variables). Standardize FDC and determine the
* associated standard normal probabilities (store in GDC);
MDDCC12 = ROOT(MDDCC);
LCC12 = DIAG(SQRT(VECDIAG(MXX11)));
FDC1 = Z(1:P-NBER) * LCC12 * BDC + EDC * MDDCC12;
FDBAR = FDC1(1:N)/N;
MFF = (FDC1 * FDC1 - N * FDBAR * FDBAR) / (N - 1);
FDC = FDC1 * INV(DIAG(SQRT(VECDIAG(MFF)));
GDC = PROBNORM(FDC);

* Now, transform the 0-1 values to continuous values on the
* interval (0,1) in the following way (Note that the values
* of GDC are distributed uniform (0,1));
* (1) If the original value is a zero, map the correspon-
* ding U(0,1) value in GDC into the interval (0,1-PO),
* where PO is the mean of the Bernoulli variable.
* (2) If the original value is a one, map the corresponding
* GDC value into the interval (1-PO,1).
* These values are stored in the matrix XX;
XX = J(N, NBER, 0);
DO L = 1 TO NBER;
PO=XBAR(|,L+P-NBER|);
DO I=1 TO N;
   IF X(|I,L+P-NBER|)=0 THEN XX(|I,L|)=GDC(|I,L|)*(1.0-PO);
   ELSE XX(|I,L|)=1.0-PO*(1-GDC(|I,L|));
END;
END;

* Rank the new values stored in XX, then standardize the ranks
* so they fall between zero and one. Finally, treating these
* standardized ranks as probabilities, use the PROBIT function
* to determine the standard normal value associated with each
* rank. These are the standard normal (Z) values for the
* Bernoulli variables and are stored with the other transformed
* values in Z.;

RBER=J(N,NBER,0);
DO I=1 TO NBER;
   RBER(|I|)=RANK(XX(|I|));
END;
UBER=(RBER-J(N,NBER,0.5))*(1/N);
ZBER=PROBIT(UBER);
Z(|,P-NBER+1:P|)=ZBER;
FREE GDC XX ZBER;

IV. Masking the Transformed Data
* Compute the mean vector (ZBAR) and the sample covariance matrix
* (MZZ) of Z, the matrix of transformed vector observations.;
FREE FX;
ZBAR=Z(|+,|)/N;
MZZ=(Zc*Z - N*ZBARc*ZBAR)/(N-1);
IVZ12=INV(DIAG(SQRT(VECDIAG(MZZ))));
RHOZ=IVZ12*MZZAIVZ12;
PRINT ZBAR , MZZ , RHOZ;
FREE MZZ;

IV. A. Generating the Error Vectors
* Generate N error vectors to be added to the matrix of Z-values.
* The error vectors will be transformed to have a correlation
* structure closely resembling the correlation structure of the
* Z's. In order to give the errors the desired correlation, we
* initially need the error vectors to have a near identity co-
variance matrix. If we simply randomly generate standard normal error vectors, we get some correlation between the elements of the vector. To prevent this, we generate half the error vectors randomly and then use an algorithm when generating the other half that nearly guarantees no correlation between elements of each error vector.

Initialize the error matrix \( U \) and observation counter \( IOBS \).

\[ U = J(N, P, 0); \]
\[ IOBS = 0; \]

When generated, each error vector must meet two restrictions:

1. Each element of the vector must be absolutely greater than a specified bound, which may change depending on the number of attempts made at generating an error vector.
2. The sum of squares of each segment of five elements in the error vector must also exceed a specified bound.

The following variables must be created so these conditions can be checked:

- \( LIM \) — The number of segments of length five in each vector
- \( ENDLIM \) — The last vector element belonging to a segment
- \( ILow, IUPP \) — Define the last group of residual elements of each vector not belonging to a segment of five
- \( NHALF \) — The cutoff point at which generated error vectors go through the algorithm to guarantee uncorrelated elements;

\[ LIM = \text{INT}(P/5); \]
\[ ENDLIM = LIM * 5; \]
\[ ILow = ENDLIM + 1; \]
\[ IUPP = ENDLIM + \text{MOD}(P, 5); \]
\[ NHALF = \text{INT}((N-1)/2) + 1; \]

Generate \( NHALF-1 \) error vectors by calling ERRVEC. There is no requirement on the absolute magnitude of the errors, but the sum of squares of each group of five elements must exceed one.

RANERR:

\[ UT = J(P, 1, 0); \]
\[ IOBS = IOBS + 1; \]
\[ \text{RUN ERRVEC}(UT, LIM, ENDLIM, ILow, IUPP, P, 0, 0, 1.0); \]
\[ U(\lfloor IOBS, \rfloor) = UT; \]
\[ \text{IF} IOBS < NHALF \text{ THEN GOTO RANERR; } \]

The remaining error vectors will be adjusted in the UNCORR routine so the sample covariance matrix of the error vectors is nearly equal to the identity.
* Initialize LO at one. LO is a value which runs between 1 and P. For each new error vector the LO-th element of the vector is adjusted to give a zero mean for all vectors and zero correlations between elements of each error vector.;

LO=1;

* Compute U1SUM and MUU11, the cumulative sum and matrix of sums of squares and cross products of all error vectors thus far generated. These are passed to the UNCORR routine for sign and magnitude adjustment along with UT, the generated error vector for the IOBS-th Z vector. After UT is adjusted, it is stored in U, LO is incremented, and the procedure continues.;

NEXTVEC:
   MUU11=Uc*U/(N-1);
   U1SUM=U(|+,|)/N;
   UT=J(P,1,0);
   IOBS=IOBS+1;
   RUN ERRVEC(UT,LIM,ENDLIM,ILLOW,IUPP,P,0,0,1.0);
   RUN UNCORR(UT,MUU11,U1SUM,LO,P);
   U(IOBS,|)=UTc;
   LO=LO+1;
   IF LO > P THEN LO=1;
   IF IOBS < N THEN GOTO NEXTVEC;

UBAR=U(|+,|)/N;
MUU=(Uc*U-N*UBARC^UBAR)/(N-1);

*---------------------------------------------------------------------
* V. Masking the Data Sufficiently
*---------------------------------------------------------------------

* We compute the Cholesky decomposition of the correlation of the Z's. Using this, we add an error vector to each Z that is approximately normally distributed with mean zero and variance matrix equal to the correlation matrix of the Z's.;

TZZ=ROOT(RHOZ);
ZSTAR=Z+SQRT(ALPHA)*U*TZZ;

* On this (and only this) initial mask, a distance check is made to insure that each masked vector is NOT one of the two statistically closest vectors to its original counterpart.;

* CLIM is approximately one standard deviation less than the mean of a Chi-square random variable with P degrees of freedom. It is used as a cutoff point in determining if a new error vector should be generated for a Z-value or if the current error vector need only be adjusted.;
CLIM=P-SQRT(2*P)+0.6;
LO=0;

DO L=1 TO N;

* For each masked observation:
  (1) If the observation has a corresponding error vector
      which was adjusted in UNCORR, compute the cumulative
      sum and matrix of cross products for those vectors
      generated before it.
  (2) Compute the distances between the masked observation
      vector and all original observation vectors by calling
      DISTANCE. The variable INDIC will indicate whether
      the original vector corresponding to the masked vector
      is one of the two closest vectors to it.;
  (3) If the above is in fact true, one of two avenues is
      taken to correct the problem:
      a. If the distance between original and masked is
      less than CLIM, generate a new error vector (call
      ERRVEC) with elements of higher magnitude.
      b. If the distance between original and masked is
      greater than CLIM, the magnitude of the original
      error vector is increased to insure that the
      distance between the masked and original is not
      one of the smallest two distances;

ITRY=1;
IF L > NHALF THEN DO;
   LO=LO+1;
   IF LO > P THEN LO=1;
   U1=U(|1:L-1,|);
   MUU11=U1*U1/(N-1);
   U1SUM=U1(|+,|);
END;
RETRY:

* If ITRY > 1, a new error vector has been generated and the
distance requirement must be checked.;

  MINDX1=0;
  MINIX1=0;
  MINDX=0;
  MINIX=0;
  DXL=0;

RUN DISTANCE(INDIC,L,N,ALPHA*RHOZ,Z,ZSTAR,MINDX1,MINIX1,MINDX,
MINIX,DXL,P,ITRY);

* The distance requirement has not been satisfied - either
generate a new error vector or adjust the magnitude of
the current error vector.;

IF INDIC < 1 THEN DO;
  ITRY=ITRY+1;
IF DXL < CLIM THEN DO;
* Depending on the attempt (ITRY), increase the bounds for (1)
* absolute magnitude of each error vector element and (2) sum
* of squares of each group of five elements.;

IF ITRY = 2 THEN RUN ERRVEC(UT,LIM,ENDLIM,ILOW,IUPP,
P,ITRY,0.0,1.0);
IF ITRY = 3 | ITRY = 4 THEN RUN ERRVEC(UT,LIM,ENDLIM,
ILOW,IUPP,P,ITRY,0.178,2.67);
IF ITRY = 5 | ITRY = 6 THEN RUN ERRVEC(UT,LIM,ENDLIM,
ILOW,IUPP,P,ITRY,0.356,3.18);

* If the current observation is in the second half of the data,
call UNCORR to adjust signs and magnitudes to insure
uncorrelated error vector elements.;

IF L > NHALF THEN RUN UNC0RR(UT,MUU11,U1SUM,L0,P);
  U(|L,|)=UTC;
  ZSTAR(|L,|)=Z(|L,|)+SQR(UGAMMA)*U(|L,|)*TZZ;
IF ITRY=7 THEN GOTO NEXTL;
GOTO RETRY;
END;

* Adjust the error vector so that the distance between masked and
original is not the smallest distance. Generate a U(ULOW,UUPP)
number (SRAN) and multiply the error vector by

AMULT=sqrt(SRAN/DXL).

This makes DXL equal to (AMULT**2)*DXL, which is greater than
MINDX1, the smallest distance. Hence, the distance requirement
is now satisfied.;

ULOW=(2*MINDX1+MINDX)/3;
UUPP=1.232*MINDX;
SRAN=(RANUNI(J(1,1,0)))*(UUPP-ULOW)+ULOW;
AMULT=SQR(SRAN/DXL);
UL=U(|L,|);
U(|L,|)=AMULT*UL;
END;

NEXTL:
END;

* Make the error vectors have mean zero. In addition, compute
* TAU, which estimates the multiple of RHOZ assigned to be the
* variance matrix of each error vector.;

UBAR=U(|+,|)/N;
UNEW=U-(J(N,1,1))*UBAR;
MUU=(UNEWc*UNEW)/(N-1);
TAU=(ALPHA/P)*TRACE(MUU);
PRINT TAU;

FREE U MUU;

* Define U to have the desired covariance structure, recompute
* ZSTAR and set UNEW to U.;

U=SQRT(ALPHA)*UNEW*TZZ;
ZSTAR=(Z+U);
UNEW=U;
FREE U;

* VI. Iterating to Obtain the Desired Correlations *

ADDZ:

* This marks the start of the iteration loop. We have a matrix
* storing the masked transformed data. Begin the process of
* transforming the data back to the original scale .;

* VI. A. Transforming the Data Back to Original Scale *

* Initialize XSTAR;

XSTAR=J(N,P,0);
STORE XSTAR;

* Compute and output the sample covariance of the error vectors.
* The variances of the elements of the UNEW vectors will be
* used below in a standardization.;

UBAR=UNEW(|+,|)/N;
MUU=(UNEWc*UNEW-N*UBARC*UBAR)/(N-1);

* Begin the retransformation procedure;;
* For each variable, determine the rank of each masked observation and store the ranks in RZSTAR.

RZSTAR=J(N,P,0);
DO K=1 TO P;
   RZSTAR(:,K)=RANK(ZSTAR(:,K));
END;

* Initialize DSTAR to store standardized ranks. Standardize each element of each error vector and standardize the ranks using the DSTAR equation below.

DSTAR=J(N,P,0);
UTILDA=UNEW*INV(DIAG(SQRT(VECDIAG(MUU))));
DSTAR=(1/N)*((RZSTAR-J(N,P,0.97)) + 0.94*PROBNORM(UTILDA));

* For each variable, transform its data values back to the original scale in an unrestricted form (i.e. the values may not fall in the parameter space.). Initialize M in preparation for the call to DISTINCT.

M=0;
DO 11=1 TO P;
   * Omit the 0-1 variables. They will be transformed back later as a group.
   IF TYPE(11,1)=3 THEN GOTO SKIPBE;
   * For variable 11, set FI to the 11-th column of DSTAR. Proceed to sort the values of FI in increasing order.
   FI=DSTAR(:,11);
   FX=FI;
   R2=RANK(FI);
   FI(:,R2)=FX;
   LOAD X R;
   XI=J(N,1,0);
   * Store the original x-values of variable in XI then arrange them in increasing order.
   XII=X(:,11);
   XI(R(:,11),:)=X(:,11);
   STORE R X;
   LOAD F;
XS=J(N+2,1,0);
FS=J(N+2,1,0);

* Find the distinct values of XI and store them in XS.
* Store the corresponding values of FI in FS;

RUN DISTINCT(N,II,M,XS,FS,XI,F);
STORE F;
LOAD XSTAR;

* Form the linear interpolated continuous distribution function
* ( LICDF) for the x values of variable II. Using the values
* stored in FI (all on the unit interval), determine the x values
* which would be mapped by the LICDF to the values in FI. These x
* values constitute the unrestricted values for XSTAR;

RUN FTILDA(II,N,XS,FS,FI,XSTAR);
XI=XSTAR(1,II);
XSTAR(1,II)=XI(R2,1);
STORE XSTAR;

SKIPBE:
END;

* NOTE: An alternative to using a function of the ranks for FI
* is to apply the standard normal distribution function
* to the values in ZSTAR. This gives a set of probabili-
* ties which can be used as the values in FI. The advan-
* tage of using the rankings is that we get values for
* XSTAR that have nearly the same variance as the original
* X values. This is not true if we use probabilities
* obtained from the values in ZSTAR.

LOAD XSTAR;

* The unrestricted values of XSTAR for all 0-1 variables are
* simply the values in DSTAR for these variables. These values
* on the unit interval will be mapped to 0 or 1 in PSPACE.

IF NBER ^= 0 THEN XSTAR(P-NBER+1:P)=DSTAR(P-NBER+1:P);

* Insure that all masked values meet their restrictions.

NRI=1;
DO II=1 TO P;
* If the II-th variable has a restriction associated with it,
* set XBARII to the mean of the original variable values (used
* only in the case of Bernoulli variables) then call PSPACE
* to insure the values of XSTAR are in their parameter spaces.;

IF NRI > NR THEN GOTO FINRES;
XBARII=0;
IF TYPE(jll,1I)=3 THEN XBARII=XBAR(|,II|);
IF RESTRICT(|NRI,1I|=II THEN RUN PSPACE(N,II,NRI,TYPE,
XSTAR,XBARII,RESTRICT);

END;

FINRES:

* Standardize ZSTAR by multiplying by the inverse of the diagonal
* matrix of standard deviations of the masked variables.

ZBAR=ZSTAR(|+,|)/N;
MZZ=(ZSTARc*ZSTAR-N*ZBARc*ZBAR)/(N-1);
ZSTAR=ZSTAR*INV(DIAG(SQRT(VECDIAG(MZZ))));

* Combine the two data matrices and compute the sample covariance
* and correlation matrices.;

ZALL=Z|ZSTAR;
STORE Z ZSTAR;
ZBAR=ZALL(|+,|)/N;
MZZ ALL=(ZALLc*ZALL-N*ZBAR c *ZBAR)/(N-1);
VZALL=INV(DIAG(SQRT(VECDIAG(MZZALL))));
RHOZALL=VZALL*MZZALLAVZALL;

MZZ11=MZZALL(1:P,1:P|);
MZZ12=MZZALL(1:P,P+1:2*P|);
MZZ21=MZZALL(P+1:2*P,1:P|);
MZZ22=MZZALL(1:P+1:2*P,1:P+1:2*P|);

RHO11=RHOZALL(1:P,1:P|);
RHO12=RHOZALL(1:P,P+1:2*P|);
RHO21=RHOZALL(1:P+1:2*P,1:P|);
RHO22=RHOZALL(1:P+1:2*P,P+1:2*P|);

PRINT RHO11 , RHO12 , RHO22;
FREE ZBAR MZZALL MZZ12 MZZ21 RHO11 RHO12 RHO21 RHO22;

* Load the matrix of original values. This is the matrix X if
* there are no classification variables and the matrix XOLD if
* classification variables are present in the original data.;

IF NC=0 THEN DO;
LOAD X;

XOLD=X;
END;

IF NC~0 THEN LOAD XOLD;

* Concatenate the matrices of original and masked values. Compute
* and output the sample covariance and correlation matrices to
* give the user an idea of how the matrix of masked correlations
* is converging to the matrix of original correlations:

XALL=XOLD|XSTAR;
XBAR=XALL(+,)/N;
MXX=(XALL*AXALL-N*XBAR*XBAR)/(N-1);
IVX12=INV(DIAG(SQRT(VECDIAG(MXX))));
RHOX=IVX12*MXX*IVX12;

MXX11=MXX(1:P,1:P);
MXX12=MXX(1:P,P+1:2*P);
MXX21=MXX(P+1:2*P,1:P);
MXX22=MXX(P+1:2*P,P+1:2*P);

RHO11=RHOX(1:P,1:P);
RHO12=RHOX(1:P,P+1:2*P);
RHO21=RHOX(P+1:2*P,1:P);
RHO22=RHOX(P+1:2*P,P+1:2*P);

PRINT ICORR, XBAR, RHO11, RHO12, RHO22;

IF NC=0 THEN STORE X;
IF NC~0 THEN STORE XOLD;

*-----------------------------------------------*
* VI. B. Equating the Cross Correlations to a target *
*-----------------------------------------------*

* If ICORR, the iteration counter for setting cross correlations,
* is equal to LIMCORR go to the second iterative procedure:

IF ICORR=LIMICORR THEN GOTO FIXCORR;

* The purpose of iterating is to obtain the desired correlations
* within the masked set of variables and between masked and origi-
* nal variables. Here, we are concerned with correlations
* between the variables X(i) and XSTAR(i), for i=1,2,...P. After
* the first mask, we compute the average of these correlations,
* excluding the highest and lowest cross correlations, and store
* this robust mean in PBAR. We then adjust the elements of the
* error vectors at each iteration depending on whether the
* correlation between the corresponding original and masked
* variables is higher or lower than PBAR:
MINCOR=1.0;
MAXCOR=0.0;

IF ICORR=0 THEN DO;
    SUMCOR=0;

    * Compute the sum of the correlations.;

    DO K=1 TO P;
        IF RHO12(|K,K|) < MINCOR THEN MINCOR=RHO12(|K,K|);
        IF RHO12(|K,K|) > MAXCOR THEN MAXCOR=RHO12(|K,K|);
        SUMCOR=SUMCOR+RHO12(|K,K|);
    END;

    * Subtract the largest and smallest correlations and compute the
    * mean of what remains. Output the target cross correlation.;

    PBAR=(SUMCOR-(MINCOR+MAXCOR))/(P-2);

    PRINT PBAR;
END;

ELSE DO;

    DO K=1 TO P;
        IF ABS(RHO12(|K,K|)-PBAR) > 0.01 THEN GOTO SETBAA;
    END;

    ICORR=LIMICORR;
    GOTO FIXCORR;
END;

SETBAA:

    * BAA12 is the vector of multipliers corresponding to each element
    * of the error vectors. BAA12 is re-initialized to the unit
    * vector at every iteration. Then, if a cross correlation is
    * close to the target, the error elements corresponding to that
    * correlation will not be changed.;

    BAA12=J(P,1,1);

    * The diagonal elements of RHO12 are the correlations between
    * each original and masked variable. Linear interpolation is
used to set the multiplier of the \( K \)-th element of the error vector, which will adjust the magnitude of the error to give the desired correlation, \( \frac{\text{PBAR}}{} \).

\[
\text{DO } K=1 \text{ TO } P;
\]

If the cross correlation is within 0.01 of the target, the corresponding error terms are not adjusted. Otherwise, the error terms are increased if the target is less than the cross correlation, and the error terms are decreased if the opposite is true.

\[
P\text{DIFF} = \text{PBAR} - \text{RHO12}(|K,K|);
\]

\[
\text{IF } |P\text{DIFF}| \leq 0.01 \text{ THEN GOTO NEXTBA;}
\]

\[
\text{PSTAR} = 0.5^* \text{PBAR} + \text{RHO12}(|K,K|);
\]

\[
\text{BAA12}(|K,I|) = \frac{1.0 - \text{PSTAR}}{1.0 - \text{RHO12}(|K,K|)};
\]

\[
\text{NEXTBA:}
\]

Multiply each error vector by the diagonal matrix of multipliers. These new error vectors will be added to the transformed \( Z \)-data to form new masked, transformed data.

\[
U = \text{UNEW} \times \text{DIAG(BAA12)};
\]

\[
\text{UNEW} = U;
\]

FREE \( U \);

Recompute \( Z\text{STAR} \) with the new set of error vectors and increment the iteration counter. Go to label "ADDZ" where the retransformation of the \( Z\text{STAR} \) values is made continuing the iteration procedure.

\[
\text{LOAD } Z \text{ ZSTAR};
\]

\[
Z\text{STAR} = (Z + \text{UNEW});
\]

\[
\text{ICORR} = \text{ICORR} + 1;
\]

GOTO ADDZ;

---

VI. C. Equating Original & Masked Correlation Matrices

---

If \( \text{NCORR} \) has exceeded \( \text{LIMNCORR} \), the maximum number of iterations has been made. Proceed to make final transformations and output.
* the masked data.;

NCORR=NCORR+1;
IF NCORR > LIMNCORR THEN GOTO SETXOUT;

* For NCORR-th iteration, set SQCORR to zero. The sum of squared
differences between corresponding correlations in the original
and masked correlation matrices, is computed and stored in
SQCORR. This value is then output to give the user an idea
of how well the masked correlation matrix is converging to the
original correlation matrix.;

SQCORR=0;
DO I=1 TO P-1;
   DO L=I+1 TO P;
      SQCORR=SQC0RR+(RH011(I,L)-RH022(I,L))**2;
   END;
END;

* Determine the variable whose original correlations with other
variables are most different with those same masked
correlations. (See Section 3.3.3 for details of this selection
criterion.) That is, compute the sum of squared differences
between corresponding rows in the original and masked
correlation matrices. The row (or variable) with the highest
sum of squares is the variable to be modified.;

   SET IVARX to zero. Initialize DXCORR, the maximum of the sum
of squares, at zero. Also, set KAPPA, the average of the cross
correlations, to zero.;

IVARX=0;
DXCORR=0;
KAPPA=0;

DO I=1 TO P;
   KAPPA=KAPPA+(RH012(I,I)/P);
END;

DO I=1 TO P;
   DXRHO=0;
   DO L=1 TO P;
      DXRHO=DXRHO+(RH011(I,L)-RH022(I,L))**2;
   END;
   IF DXRHO > DXCORR THEN DO;
      DXCORR=DXRHO;
      IVARX=I;
   END;
END;
Output the iteration number, the squared differences between all correlations, the average cross correlation and the variable to be modified.

PRINT NCORR SQCORR KAPPA IVARX;

To obtain the desired masked correlations between the masked variable of IVARX and the other masked variables, we work with the transformed masked Z-values of variable IVARX. We reset these transformed masked values to a linear combination of the transformed masked values of the other variables and the original transformed values of the IVARX variable. The linear combination is determined by fitting a model based on the correlations between the masked and original IVARX, and between the other masked variables. See Section 3.3.3 for the exact model. It is too complicated to reproduce here.

XRHO represents the design matrix and YRHO the vector of responses in the model XRHO^B^YRHO. B is the vector which is predicted and defines the linear combination discussed above.

XRHO=J(P,P,0);
YRHO=J(P,1,0);

Set the XRHO matrix and the YHO vector based on model (3.3.19) of Section 3.3.3 of the thesis.

XRHO(1,1)=RHO12(|IVARX,IVARX|)-1;
YRHO(1,1)=KAPPA-1;

The j-th element of YRHO is the correlation between IVARX and the j-th variable not equal to IVARX minus the correlation between IVARX and the j-th masked variable not equal to IVARX. The i-th element of row one of XRHO is the correlation between variable IVARX and the i-th masked variable not equal to IVARX. The j-th element of column one of XRHO is the correlation between the j-th masked variable not equal to IVARX and the masked IVARX variable minus the correlation between the j-th masked variable not equal to IVARX and the original variable IVAR. The (j,i)-th element is unity if i equals j. Otherwise, the (j,i)-th element is equal to the correlation between the j-th masked variable not equal to IVARX and the i-th masked variable not equal to IVARX. In the settings described above, 2 <= i,j <= P.

VDUM=0;

DO I=1 TO P-1;
   IF IVARX <= I THEN VDUM=1;
   XRHO(|1,1+1|)=RHO12(|IVARX,I+VDUM|);
XRHO(|I+1,1|)=\rho_{22}(|IVARX,I+VDUM|)-\rho_{12}(|IVARX,I+VDUM|);
YRHO(|I+1,|)=\rho_{011}(|IVARX,I+VDUM|)-\rho_{012}(|IVARX,I+VDUM|);

END;

VDUM1=0;

DO I=1 TO P-1;
   IF IVARX <= I THEN VDUM1=0;
   VDUM2=0;
   DO L=1 TO P-1;
      IF IVARX <= L THEN VDUM2=1;
      IF I=L THEN XRHO(|I+1,L+1|)=1;
      IF I<>L THEN XRHO(|I+1,L+1|)=\rho_{22}(|I+VDUM1,L+VDUM2|);
   END;
END;

* Solve the system by multiplying YRHO by the generalized inverse
* of XRHO.;
BTILDA=GINV(XRHO)*YRHO;
BHAT=J(P,1,0);

* Reset the values of BTILDA in BHAT so a new masked, transformed
* vector of values for the IVARX-th variable can be computed by
* multiplying BHAT by the (P-1) transformed, masked vectors
* not corresponding to IVARX and the unmasked, transformed
* vector of responses corresponding to IVARX.;
IDUM=0;
IBHAT=1;
BHAT(|IVARX,|)=BTILDA(|1,|);

DO I=1 TO P-1;
   IF IVARX <= I THEN IDUM=1;
   BHAT(|I+IDUM,|)=BTILDA(|I+1,|);
END;

PRINT XRHO YRHO BHAT;

LOAD Z ZSTAR;

* Compute a new vector of masked, transformed responses for the
* IVARX-th variable and continue the iterative procedure.;
ZTILDA=Z(|,IVARX|)*(1-BHAT(|IVARX,|))+ZSTAR*BHAT;
ZSTAR(|,IVARX|)=ZTILDA;
GOTO ADDZ;

* The iterative process is complete. Transform any sets of
* Bernoulli variables created to mask classification variables
* back to masked classification variables.;

SETXOUT:

* If classification variables do not exist, XOUT is already set to
* XSTAR. Go to the section where the masked data is output.;

IF NC = 0 THEN DO;
  XOUT=XSTAR;
  GOTO OUTP;
END;

* Classification variables are present. Initialize CSTAR, which
* will store the masked values of the classification variables.
* JCOL is the column in XSTAR before the pseudo variables for
* the next classification variable begin. INC indexes the
* columns of CSTAR so the actual categorical values can be
* stored column after column.;

CSTAR=J(N,NC,0);
JCOL=PO-NC;
INC=1;

DO II=1 TO PO;

* If variable II is not a classification variable, go on to the
* next variable. Otherwise, call ZTRANS. This will, for each
* record, take the values of the pseudo variables for that record
* and determine the appropriate masked categorical value to be
* reported. See ZTRANS for details of this back transformation.;

  IF TYPE1(II,1) = 3 THEN DO;
    RUN ZTRANS(N,II,JCOL,INC,TYPE1,CLASS,XSTAR,CSTAR);
    INC=INC+1;
    JCOL=JCOL+TYPE1(II,2)-1;
  END;

END;

FREE XS FS FX FI F R Z;

* Use the POSIT array to determine the original ordering of the
* the variables. Store the columns of variable responses in this
* order in XOUT.;

XOUT=J(N,PO,0);

DO I=1 TO PO;

  IF I <= PO-NC THEN XOUT(I,POSIT(I,|I|))=XSTAR(I,|I|);
  IF I > PO-NC THEN XOUT(I,POSIT(I,|I|))=CSTAR(I,I-PO+NC);

END;
* Output the masked data vectors to the SAS data set MDATA. Then,
* read the data set MDATA into SAS variables so the masked values
* can be output to the WYLBUR data set designated in the JCL.;

OUTP:
FREE XALL;
SHOW STORAGE;
VARNAMES={X1 X2 X3 X4};
CREATE MDATA FROM XOUT(|COLNAME=VARNAMES|);
APPEND FROM XOUT;
CLOSE MDATA;

FINISH;

************************************************************************ END MAIN ROUTINE ************************************************************************;

RUN;

* SAS Data step: Set SET1 to MDATA (the masked variables)
* and output to the file named in OUT1 of the JCL. Delete
* the IML library used to temporary store matrices.;

DATA SET1; SET MDATA;
FILE OUT1;
PUT (X1 X2 X3 X4)(2*9.4 1*5.0 1*4.0);

//LMASK DD DSN=MATRIX.IML.LIB,UNIT=Disk,DISP=(OLD,DELETE),
// VOL=SER=UCC004
//