The Microdata Analysis System at the U.S. Census Bureau[1]

Michael Freiman[2], Jason Lucero[3], Lisa Singh[4]
Jiashen You[5], Michael DePersio[6], Laura Zayatz[7]
[2]U.S. Census Bureau, 4600 Silver Hill Road, Washington, DC 20233,
michael.freiman@census.gov
[3]Freddie Mac, 8200 Jones Branch Drive, McLean, VA 22102
[4]Georgetown University Department of Computer Science, 329A St. Mary's Hall,
Washington, DC 20057
[5]University of California-Los Angeles Department of Statistics, 8125 Math Sciences
Bldg., Box 951554, Los Angeles, CA 90095
[6]University of Delaware Department of Mathematical Sciences, 501 Ewing Hall,
Newark, DE 19716
[7]U.S. Census Bureau, 4600 Silver Hill Road, Washington, DC 20233

Abstract

The U.S. Census Bureau has the responsibility to release high quality data products while maintaining the confidentiality promised to all respondents under Title 13 of the U.S. Code. This paper describes a Microdata Analysis System (MAS) that is currently under development, which will allow users to receive certain statistical analyses of Census Bureau data—such as cross-tabulations, regressions (with diagnostic plots), histograms and scatterplots—without ever having access to the data themselves. Such analyses must satisfy several statistical confidentiality constraints; those that fail these constraints will not be output to the user. In addition, all analyses are performed after application of the Drop *q* Rule, a data subsampling routine, and regressions involving categorical predictors sometimes require modification. We describe the system's capabilities, as well as the confidentiality protections and the major types of attacks they prevent, then conclude with a description of other approaches to creating a system of this sort, and some directions for future research.

Key Words: data confidentiality, remote access servers, universe subsampling, synthetic data, regression, disclosure

1. Introduction

The U.S. Census Bureau collects its survey and census data under Title 13 of the U.S. Code, which prohibits the Census Bureau from releasing any data "... whereby the data furnished by any particular establishment or individual under this title can be identified." In addition to Title 13, the Confidential Information Protection and Statistical Efficiency Act of 2002 (CIPSEA) requires the protection of information collected or acquired for exclusively statistical purposes under a pledge of confidentiality. However, to fulfill its

[1] This report is released to inform interested parties of ongoing research and to encourage discussion of work in progress. The views expressed are those of the authors and not necessarily those of the U.S. Census Bureau.

mission, the agency also must release data for the purpose of statistical analysis. In common with most national statistical institutes, our goal is to release as much high quality data as possible without violating the pledge of confidentiality.

This paper discusses a Microdata Analysis System (MAS) that is under development at the U.S. Census Bureau. Much of the framework for the system was described in Steel and Reznek [2005] and Steel [2006]. The system is designed to allow data users to perform various statistical analyses (regressions, cross-tabulations, correlation coefficients, etc.) on confidential survey and census microdata without seeing or downloading the underlying microdata.

In Section 2, we give some background on the MAS and the motivation for its development. In Section 3, we discuss the current state of the prototype system, including its capabilities and the rules that protect confidentiality. In Section 4, we examine some other approaches to the problem of creating a remote access system such as the MAS. In Section 5, we conclude with remarks on future research and the further development of the system.

2. Background on the MAS

The problem of data confidentiality—at the Census Bureau and other statistical agencies around the world—has motivated the creation of *remote access systems* that allow the user to request a statistical analysis and receive the result without having direct access to the underlying microdata. Common to almost all remote access systems is that the ability to receive desired results is not absolute: in some instances, the result might be based on perturbed data, and most proposals for remote access systems include the rejection of some queries to preserve confidentiality. The idea of a remote access system goes back at least to Keller-McNulty and Unger [1998], although the concept of allowing customized queries was proposed much earlier; see the description of the Geographically Referenced Data Storage and Retrieval System in Fellegi [1969]. Fellegi *et al.* [1972] anticipate the need to screen the query results to ensure that confidentiality is adequately protected.

Adam and Worthmann [1989] describe several restrictions that can be imposed on systems that release counts of numbers of people with particular characteristics. These include keeping a log of each user’s queries and checking each new query against the log to verify nondisclosure. However, they acknowledge that the last of these is sufficiently time consuming and storage intensive as to be unfeasible. They also consider the possibility of partitioning the data into indivisible units of two or more observations each and allowing only queries that operate on unions of the units, rather than on arbitrary sets of observations.

The Microdata Analysis System will allow the U.S. Census Bureau to provide a controlled, cost-effective setting in which data users have access to more detailed and accurate information than is currently available in our public use microdata files. The data accessible through the MAS can identify smaller geographic areas and show more detail in certain variables where our public use files would be coarsened. Our goal for the MAS is to allow access to as much high quality data as possible, while lessening the need for data to be released in less secure or more expensive manners, such as those described in Weinberg *et al*. [2007]. Rowland and Zayatz [2001] describe a predecessor of the MAS.

Unlike the proposal in Schouten and Cigrang [2003], our plan is to make the MAS available to anyone who wishes to use it. The MAS will allow access to data from demographic surveys and decennial censuses, with the goal of eventually including economic survey and census data, as well as linked datasets. We will initially make available regression analyses and cross-tabulations, with other analyses to be added in the future. Currently, we intend to keep a record of all of the queries entered into the system, but not the identities of the users making the queries. Although the record will not directly affect the output that the system provides, it will allow us to see how the system is being used so that we can improve the user experience and enhance disclosure avoidance techniques if necessary.

Our current plan—as described in Chaudhry [2007]—is to offer the MAS within the Census Bureau's free DataFERRETT service, with the intention that the system will be used by people needing fairly standard statistical analyses. The MAS has a graphical interface, programmed in Java, that allows users to select variables of interest from a list. In the case of regression, variables can be dragged into equations and, with a few clicks, users may create variable interactions and transformations of selected variables.

3. Overview of the MAS Confidentiality Rules

The Census Bureau contracted with Synectics to develop an alpha prototype of the MAS using the SAS language, with Dr. Jerome Reiter of Duke University to help in developing confidentiality rules for the system and with Dr. Stephen Roehrig of Carnegie Mellon University to help in testing these rules. Some rules were developed and modified as a result of the testing. We are using the publicly available data from the Current Population Survey March 2008 Demographic Supplement to test the system.

The MAS software is programmed with several confidentiality rules and procedures that uphold disclosure avoidance standards. Some of these are limitations on the allowable universes on which to perform the analysis. Regression analyses are further subjected to restrictions on the use of predictor and response variables. We plan to explore whether additional rules are necessary for correlation coefficients.

3.1 Confidentiality Protection for Universe Formation

MAS users are allowed to run their statistical analyses on a universe, or sub-population, of interest. Users are presented with a set of variables and category levels from which they can define a universe using condition statements on the variables, including unions and intersections as desired. For example, the user may select a universe consisting of the sub-population of all females. A more complicated universe could consist of all people who are male or unemployed, or of all people whose income in dollars falls into the union of [9180,20155] and [31662,43468], although admittedly the last of these may be of dubious utility. One of the confidentiality rules requires that all variables used to define universes must be categorical.

Since a user may want to define a universe based on variables that are not inherently categorical (i.e., those that are continuous), raw numerical variables are presented to the user as categorical recodes based on output of a separate binning routine. This cutpoint program, outlined in Lucero *et al.* [2009], creates bins of numerical values and ensures a pre-specified minimum number of observations between any two cutpoint values.

To define a universe using a numerical variable, a user is required to choose from a predetermined list of ranges the range that best meets her goal. For example, if a user wished to run an analysis on people with income of $46,000, the user would select the range that contains this value, which might be ($45000,$53000], and this would define the universe as the sub-population of all individuals whose income is between $45,000 and $53,000 (inclusive on the upper end but exclusive on the lower end). Note that a user cannot define the universe based on the income range ($39000,$46000] unless $39,000 and $46,000 are among the pre-determined cutpoints, but instead must choose cutoff values consistent with the cutpoints that are given. This is a crucial restriction on what a user can do, since allowing arbitrary universe formation on continuous data could lead to a differencing attack disclosure. Such a disclosure would occur, for example, if a user requested a table for the universe of individuals with income of at least $11,313 and the corresponding table for the universe of individuals with income of at least $11,314, and then manually compared the two tables. If only one person in the dataset had an income of $11,313, then this person's other attributes could easily be deduced, as described in Section 3.1.1.

All universes must pass certain preliminary checks to ensure that they are large enough and are not likely to lead to disclosures in combination with other universes. There are two main criteria that any universe must pass, and we describe these only briefly here.

The *Rule of 75* requires that any allowable universe must have at least 75 observations. Furthermore, if a universe is defined as the union of smaller universes, then each of these smaller universes must have at least 75 observations, as must all non-empty intersections of two or more of the smaller universes. The rule has some more nuances, addressed in Lucero [2009], which we will not discuss here. It is also important to note that we may modify this rule before the system becomes operational, with some other number in place of 75.

The *No Marginal 1s or 2s Rule* states that if a universe is defined based on m variables, the m-way contingency table based on those variables must not have any m-1 dimensional marginal totals that are equal to 1 or 2. For example, we might have available a binary variable indicating whether a person has moved in the most recent year, a variable indicating a subjective assessment of the person's health (excellent, very good, good, fair, poor) and a binary variable indicating eligibility for Medicaid. Then a universe could be constructed consisting of those who have not moved in the last year and are eligible for Medicaid and in poor health. This universe would not be allowable if any of the two-way marginals with respect to these three variables is 1 or 2. It is important to note that a marginal total can cause a universe to be disallowed even if none of the people who are counted in that total are in the universe. So, for example, if there are 1 or 2 people who have not moved and are in excellent health, the universe would not be allowed, even though non-movers in excellent health are not in the desired universe. It is enough that the marginal is defined by two of the three variables in the universe definition.

We use the notation U(n) to denote a universe with n observations, and although this is in principle ambiguous, in most cases, it should be clear from the context which n observations lie in the universe.

3.1.1 Confidentiality by Random Record Removal

A *differencing attack disclosure* occurs when a data intruder attempts to reconstruct a confidential microdata record by subtracting the statistical analysis results obtained through two queries on similar universes. Suppose a data intruder first creates two universes on the MAS, U(n) and $\mathrm{U}(n-1)$, where both contain the same n observations with the exception of one observation missing from the second universe, i.e., $\left|\mathrm{U}(n) \setminus \mathrm{U}(n-1)\right| = 1$. The difference $\mathrm{U}(n) \setminus \mathrm{U}(n-1) = \mathrm{U}(1)$ is a manipulated universe that contains the single target observation. While our rules on allowable universes provide some protection of the confidential data in the MAS, they do not completely prevent differencing attack disclosures. For example, suppose a data intruder has prior knowledge of demographics in a small geographic area, and in particular is aware of individuals, households or establishments with unique characteristics within that area. It may be the case that the intruder knows that there is only one non-citizen among the *n* residents of the area. Then the intruder may create $\mathrm{U}(n)$ and $\mathrm{U}(n-1)$, where $\mathrm{U}(n)$ is the full universe of people in the area and $\mathrm{U}(n-1)$ is the universe of citizens who live in the area. Suppose the data intruder then requests two separate cross-tabulations for the same underlying table variables; we call these two tables T_n and $\mathrm{T}_{n\text{-}1}$, as shown in Figure 1. Since $\mathrm{U}(n)$ and $\mathrm{U}(n-1)$ differ by a unique observation, $\mathrm{T}_{n\text{-}1}$ will be the same as T_n, less one unique cell count. The tables in Figure 1 show a differencing attack based on a tabulation of age (a binary classification of whether the person is at least 45 years old) versus income (a binary classification of whether income is at least $50,000).

All People

T_n	<$50,000	≥$50,000
Age<45	323	170
Age≥45	45	58

-

Citizens Only

$\mathrm{T}_{n\text{-}1}$	<$50,000	≥$50,000
Age<45	323	169
Age≥45	45	58

=

Non-Citizens Only

T_1	<$50,000	≥$50,000
Age<45	0	1
Age≥45	0	0

Figure 1: An example of performing a differencing attack by matrix subtraction.

We may perform the matrix subtraction T_n-$\mathrm{T}_{n\text{-}1}$=T_1, where T_1 is a two-way table of gender by employment status built upon the one unique observation contained in $\mathrm{U}(n) \setminus \mathrm{U}(n-1) = \mathrm{U}(1)$. As shown in Figure 1, T_1 contains a cell count of 1 in the cell of people under age 45 and with income of at least $50,000, which tells the data intruder that the one unique observation contained in U(1) has these two characteristics. By performing differencing attacks similar to the one just described, a data intruder can successfully rebuild the confidential microdata record for the one unique observation contained in U(1).

A differencing attack may also be a concern if there are two observations within an area that have a certain characteristic, particularly if the intruder is himself one of these two. Suppose, for example, that the universe contains only two non-citizens, one of whom is the intruder. The intruder could then construct the full universe $\mathrm{U}(n)$ and the portion of the universe consisting solely of citizens $\mathrm{U}(n-2)$. Since the intruder knows his own personal characteristics, he may manually remove himself from $\mathrm{U}(n)$ to get $\mathrm{U}(n-1)$ and then perform a differencing attack as above by comparing $\mathrm{U}(n-1)$ and $\mathrm{U}(n-2)$ to obtain information on the other non-citizen in the area.

To help protect against differencing attacks, the MAS implements a universe subsampling routine called the *Drop q Rule*. Traditionally, subsampling has usually been used to estimate parameters when a population is too large to analyze in an efficient manner and a (usually small) subset can give approximately the same results as the full population. Our aims are very different here: the Drop *q* Rule is intended to remove just enough observations from the dataset to thwart a differencing attack. A differencing attack performed while the Drop *q* Rule is in place will not lead to a meaningful outcome, when the attack is of one of the types described above.

The Drop *q* Rule works as follows. A user-defined universe that passes all of the previous rules has *q* records removed at random. To do this, the MAS will first draw a random integer value of *q* such that $2 \leq q \leq k$ and such that when the universe is modified by omitting *q* records, the number of remaining records is a multiple of 3. Here *k* is some predetermined number, which may depend on the size of the universe. The exact method for determining the maximum possible number *k* of observations to remove is still under consideration. Then, given *q*, the MAS will subsample the universe U(*n*) by removing *q* records at random from U(*n*) to yield a new subsampled universe $\mathrm{U}(n-q)$.

Within the MAS, all statistical analyses are performed on the subsampled universe $\mathrm{U}(n-q)$ and not on the original universe U(*n*). Each unique universe U(*n*) that is defined on the MAS will be subsampled independently according to the Drop *q* Rule. To prevent an "averaging of results" attack, the MAS will produce only one subsampled universe $\mathrm{U}(n-q)$ for each unique universe U(*n*), with this unique subsample persisting for the lifetime of the system. That is, all users who select a specific universe U(*n*) will have all analyses performed on exactly the same subsampled universe $\mathrm{U}(n-q)$. The MAS accomplishes consistent subsampling of universes by using the same random seed to perform the subsampling every time a given universe comes up. To receive the full disclosure protection offered by the Drop *q* Rule, it is necessary that the seed, while constant for a given universe, differs across universes, and this can be implemented by having the seed be a function of the set of units in the universe.

The differencing attacks of most concern require, among other things, that two universes are available that differ in size by 1 or 2. However, under the Drop *q* Rule described above, all subsampled universes have sizes that are multiples of 3, and no pair of multiples of 3 (including pairs where both numbers are the same) can have a difference of 1 or 2. Hence the Drop *q* Rule eliminates the possibility of this sort of disclosure, or even of an apparent disclosure where taking the difference of the resulting tables gives an answer that is plausible (because it has nonnegative numbers in all cells) but is not correct.

The Drop *q* Rule is a generalization of the previously used *Drop 1 Rule* and *Drop 2 Rule*, where a small and fixed number of observations were removed before analysis. These rules led to tables that were susceptible to differencing attacks. One notable vulnerability could be exploited by starting, as usual, with two universes U(*n*) and $\mathrm{U}(n-1)$, identical with the exception of one unit, with the intention of performing a differencing attack. For example, an intruder might know that a certain geographical region contains exactly one Korean War veteran. The intruder could then consider the universe of all people in that region, as compared to the universe of all non-Korean War veterans in the region. However, instead of requesting a tabulation of these two universes, the intruder may augment each universe by adding to it the full population of a non-overlapping geographical region of size $N>>n$, such as a large state that does not contain the original region. Then a three-way tabulation could be done of veteran status versus state versus the variable that the intruder wishes to disclose for the augmented universes U(*n*+*N*) and U(*n*-1+*N*). In the case of the Drop 2 Rule, it is overwhelmingly likely that all four of the dropped observations—two for each universe—will be in the large region of size *N*, thus leaving the portions of the provided tables representing the original region of interest unmodified. The MAS currently prevents a "padding" attack of this sort by restricting the types of geographies on which an analysis can be performed, and we are looking into how to further strengthen the system against this type of attack.

3.2 Confidentiality Protection for Regression Models

The MAS implements a series of confidentiality rules for regression models, in addition to the universe restrictions already mentioned. For example, users may only select up to 20 independent variables for any single regression equation. Users are allowed to transform numerical variables (either predictor or response), but they must select their transformations from a pre-approved list. This prevents the user from performing transformations that deliberately overemphasize individual observations such as outliers. Currently, the allowable transformations are square, square root and natural logarithm; others may be added in the future.

Any fully interacted regression model that contains only dummy variables as predictors poses a significant potential disclosure risk, as described in Reznek [2003] and Reznek and Riggs [2004]. Therefore, users are allowed to include only two-way and three-way interaction terms within any specified regression model, and no fully interacted models are allowed. Furthermore, a two-way interaction is allowed only if both of the interacted variables appear by themselves in the model, and a three-way interaction is allowed only if all three variables appear uninteracted in the model and each of the three associated two-way interactions also appears. However, interactions do not count against the 20-variable limit (so that, for example, if a model includes two predictor variables and their interaction, this is considered two variables, not three, for the purpose of the limit). Categorical predictor variables are included in the model through the use of dummy variables for all categories except one reference category. The MAS uses the most common category as the reference category. In addition, each predictor dummy variable must represent a category containing a certain minimum number of observations; if this minimum is not met, the dummy variable is omitted from the model. In effect, this means that sparse categories are absorbed into the reference category. We denote the minimum allowable number of observations in a category by the parameter *m*, which is initially set to $m = 3$, but can be modified as described below.

Prior to passing any regression output back to the user, the MAS also checks that R^2 is not too close to 1. If R^2 is too close to 1, then the MAS will suppress the output of the regression analysis, as releasing the results of the regression would allow estimation of the response variable with a high degree of accuracy if the values of the predictor variables for any unit were known. This is somewhat different from the usual regression context, as a more familiar situation is one in which a high R^2 is desirable, whereas here it is seen as problematic. It may also be the case that the regression does not have an unreasonably high R^2, but that there exists a certain type of subset of observations where all observations have their response values perfectly predicted by the regression (up to rounding errors in the software). Regressions with this feature will also not be provided to the user; one may think of this as a check on the local goodness of fit to complement the R^2 check on the global goodness of fit. Furthermore, output will not be given if there exists a dummy variable that assumes a value of 1 fewer than three times in the dataset. If the dummy variable represents a category of a single categorical variable, this is redundant with the use of the parameter *m* at the end of the preceding paragraph, but if the dummy variable is based on two or three interacted variables, then this is not necessarily the case.

When categorical variables are used as predictors, the rules above can be very restrictive, especially on relatively small datasets or when categorical variables are interacted, making it potentially unlikely that the system will give the desired output. Since the goal of the MAS is to provide output whenever possible, we make a slight modification to the regression in this case, in the hope that we can provide less detailed output, rather than no output at all. This is done by increasing the lower bound *m* on the number of observations that a category must contain to have its own dummy variable and avoid being absorbed into the reference category. By absorbing more categories into the reference category, we hope to alleviate the conditions that prevented the regression from being output. The MAS continues to increase *m* until either a regression is found that can safely be output—in which case that regression is fit—or *m* is large enough that one of the categorical predictors is reduced to having just a single level, with all other levels being absorbed into the reference level, leading the system to refuse output.

A shortcoming of our current approach is that it will sometimes combine categories in undesirable ways, particularly in the case of a predictor variable with ordinal structure. The method we have described above does not consider any ordinal structure that may be present, but we hope to improve this aspect of the system in the future.

If all of the requirements for a regression are satisfied, either before or after adjusting the parameter *m*, then the MAS will pass output to the user. The output includes regression coefficients; their standard errors, t-statistics and P-values; the F-statistic for the regression and its P-value; the R^2 for the regression; and an ANOVA table. All of these are rounded, to thwart any attack based on exact values of regression coefficients from large numbers of regressions.

Although most of the discussion above has focused on rules for ordinary least squares regression, the MAS also has the capability of performing logistic (either binary or multinomial) regression when the response variable is categorical or Poisson regression when the response variable is a count. In these cases, the rules for ordinary least squares regression are adapted to the new context. Limits on interactions and the approach to categorical predictors are the same. To measure whether the fit is “too good” and a regression needs to be withheld (or have *m* increased), we use pseudo-R^2 measures as a measure of global goodness of fit; if these are too high, the regression will not be given or will have *m* increased. We have also developed local goodness of fit checks for these other types of regression. As before, the rounded estimated coefficients, along with their standard errors, test statistics and P-values, are provided, as is the Analysis of Deviance table in the Poisson or binary multinomial case.

When logistic or Poisson regression is performed, the results output are similar to those for ordinary least squares regression: rounded versions of the regression coefficients, their standard errors, test statistics and P-values and, except in the multinomial logistic case, the Analysis of Deviance table.

Sparks *et al.* [2008] propose some other confidentiality rules for regression, such as using robust regression to lessen the influence of outliers, although at the moment, we still plan to use ordinary least squares regression when the response variable is numerical.

3.2.1 Synthetic Residual Plots

To determine whether an ordinary least squares regression adequately describes the data, diagnostics such as residual plots are necessary, and these are provided in almost all cases. Actual residual values pose a potential disclosure risk, since a data intruder can obtain the values of the dependent variable by adding the residuals to the fitted values obtained from the regression model. Therefore, the MAS does not pass the actual residual values back to the user. To help data users assess the fit of their ordinary least squares regression models, diagnostic plots are based on synthetic predictor (or fitted) values and synthetic residuals. These plots are designed to mimic the patterns seen in the scatterplots of the real residuals versus the real fitted values, or of the real residuals versus the values of the individual variables.

The first step in creating synthetic residual plots is to create the synthetic dataset in such a way that the synthetic data mimic the actual data. For a plot of residuals versus a quantitative predictor variable, we first create a synthetic version of the predictor, then create the synthetic residuals. We summarize the methodology here; it was devised by Reiter [2003], who provides considerably more detail than we do, especially on determining the synthetic residuals. The distribution of the predictor variable **x** is simulated using a kernel density estimator, and then sampling is used to generate $\mathbf{x}^{s}_{p}$ from the approximate distribution. When Reiter's method is used, there is no one-to-one correspondence between real observations and synthetic observations, so there need not be any particular relationship between the size of the actual dataset and the size of the synthetic sample. The lack of such a correspondence helps to protect outliers, as an outlier in the original data may not appear in the synthetic plot or may appear more than once. In the case of categorical predictor variables, we let the synthetic sample size equal the actual sample size, while in the case of numerical predictor variables, we let the synthetic sample size be the minimum of 5,000 and the actual sample size. This is because when making the synthetic and actual sample sizes equal in the numerical case, we found that the system was slow when dealing with large datasets, and that the vast majority of the time that the analysis took was spent on creating the synthetic residual plots for numerical variables. A shortcoming of the method for creating synthetic continuous predictors is that the kernel density estimator is not able to identify a probability mass at a single point, but rather will assume that the probability density function should be high in the neighborhood of that point. This should not invalidate the method, but it will affect the distribution along the x-axis for a predictor variable such as income, for which many people have a true value of 0, and for which round numbers are frequently reported.

For categorical variables $\mathbf{x}_{p}$, the values of $\mathbf{x}^{s}_{p}$ are generated by bootstrap sampling the real data. If some categories are sparsely populated and homogeneous in their residuals, there is the potential for using the synthetic residual values at the sparse category to disclose real residuals, but otherwise this part of the algorithm poses negligible disclosure risk. One possible approach to this problem is to suppress residuals for categories that are sufficiently sparse.

It should be noted that both of these methods for creating the synthetic data work with one variable at a time, i.e., $\mathbf{x}^{s}_{p}$ are drawn marginally, not jointly, and thus no valid analysis can be performed based on the joint distribution of the synthetic variables. This is not currently a major concern, as it is not our intention to release synthetic data through the MAS. However, this does impose a limitation on the range of diagnostics that we can make available in the future based on synthetic variables generated using this method.

The next step is to generate the standardized synthetic residuals $\mathbf{t}^{s}_{p}$ so that the relationship between $\mathbf{t}^{s}_{p}$ and $\mathbf{x}^{s}_{p}$ at any point x^{s}_{kp} in $\mathbf{x}^{s}_{p}$ is consistent with the relationship between **t** and $\mathbf{x}_{p}$ around point x^{s}_{kp}. To accomplish this, we must make a different set of synthetic residuals for each predictor variable. Note that x^{s}_{kp}, if numerical, will not necessarily be a value observed in continuous real data, because of the use of the kernel density approach..

For each variable, the goal is to give the user something akin to a plot of the standardized residuals of the full (possibly multiple) regression model versus the value of $\mathbf{x}_{p}$. For a variable *p* and an index *k*, define

$$t^{s}_{kp} = b_{kp} + v_{kp} + n_{kp}.$$

The first term b_{kp} gives the expected value of the standardized residual for any given value of *p*; this is determined by fitting a generalized additive model (GAM) to the relationship between the real values of the predictor variable and the real values of the residuals. The second term v_{kp} accounts for the deviation of residuals from the GAM curve, and is found by examining the deviations of the real points whose *x* values are near

x^s_{kp} from the GAM curve. The third term n_{kp} is a homoscedastic noise term to further protect the true residuals.

When all steps are complete, the system creates a scatterplot of the synthetic residuals versus each numerical synthetic predictor variable, as well as a scatterplot of the synthetic residuals against the fitted value, with a kernel smoother used to show the general shape of the latter curve. To protect outliers, the scatterplot requires all synthetic standardized residuals to be in the interval [-4,4], with values that would otherwise be outside this range truncated appropriately.

Since categorical predictors do not lend themselves to scatterplots, the residual plots for categorical variables are replaced by side-by-side boxplots, with one boxplot for each value of the predictor. We have described above how the synthetic values of the predictors are found; the synthetic value of a residual in this case is chosen by selecting at random an observation from the original dataset with the desired value of the categorical predictor, using this as an initial value of the synthetic residual, and then adding homoscedastic random noise as before.

For logistic regression, whether binary or multinomial, diagnostic plots are also given, following the method in Reiter and Kohnen [2005].

3.2.2 Testing Residuals for Normality

Another useful diagnostic for a linear regression is a test of the normality of the residuals. This may be done in the MAS by using a normal Q-Q plot, or by choosing from a number of available test statistics, such as the Anderson-Darling statistic. The MAS produces both the Q-Q plot and the numerical output. Since a direct Q-Q plot of the actual residuals poses a potential disclosure risk, the plot uses synthetic residuals rather than real residuals. However, because the synthetic residuals include normal noise, we would expect them to look more normal than the real residuals. Allowing the test statistics to use real residuals seems reasonable, as it is difficult to see how an individual observation could be revealed by looking at these. Hence we make the following recommendation: use the test statistics to determine normality, but use the Q-Q plot to assess the nature of the deviation from normality, if any.

4. Additional Features

Among the additional features being developed are histograms and scatterplots of numerical data. Each of these poses a disclosure risk if unperturbed.

4.1 Histograms

Histograms do not seem to be a major disclosure risk, except when outliers are present. The Drop *q* Rule already gives some protection against disclosure; we modify the method used to create the histogram so that there is further protection. The main concern with a histogram is that it may be used to find outlier values of the variable being plotted.

We begin by removing from the distribution any extreme outliers. When this has been done, we use a kernel density function to find a smoothed estimate of the distribution of the variable. We then draw a sample from the smoothed distribution equal in size to the original dataset. The smoothing can be thought of as a horizontal perturbation in the histogram, since it may move some of the probability mass caused by one observation from that observation’s bin to nearby bins. Drawing the sample from the distribution may be considered a vertical perturbation, as the number of observations in each bin (the height of the bin) need not equal the number of observations expected to be in the bin based on the smoothed density. Note that because of the smoothing, the bounds of the estimated density will fall beyond the bounds of the observed data, so it is possible for the synthetic histogram to extend further than the data itself. However, if there is one observation that is somewhat more extreme than the others, but perhaps not so extreme as to be excluded as an outlier, it is possible that when drawing from the smoothed distribution, none of the sample will come from the area around that observation, so the histogram may also extend less far in that direction than the real data. The discreteness of the bins of the histogram also acts as a sort of de facto perturbation of the data.

To further protect unusual values, we require that any bin in the histogram must have a minimum of three observations. Bins with fewer than three observations have more observations added to augment them to three. Bins with three observations (after the augmentation) are colored red when the histogram is plotted, whereas the other bins are colored gray.

We are still testing and modifying the method of creating histograms to ensure that it does not create a disclosure risk.

4.2 Scatterplots

We are considering a variety of approaches to the problem of making a disclosure-proof scatterplot of two numerical variables.

One approach is to use the same method as for synthetic residuals. A possible downside to this is that this method treats the two variables in an asymmetric fashion, so that a synthetic plot of *y* versus *x* need not look like a synthetic plot of *x* versus *y*. In the case of a residual plot, this asymmetry between the variables is natural, but in a more general scatterplot, we may want both variables treated similarly.

Another approach is to use a method that starts with the true scatterplot—or, if this includes too many points, a subset of the points of the scatterplot—and then moves each point a random distance in a random direction. Points that are close to other points will be moved only a little, whereas an outlier, even a modest one, will be moved more. This is somewhat similar in spirit to the approach of You [2010].

Sparks *et al.* [2008] use a method of side-by-side boxplots to replace both residual plots and ordinary scatterplots. When this method is used, the x variable is split into bins and a histogram of the y variables for each x bin is made, then the histograms are plotted side by side. If certain precautions are taken, such as Winsorizing the data to protect outliers, disclosure risk can be minimized. Sparks *et al.* argue that in many cases, side-by-side boxplots not only have less disclosure risk than scatterplots, but also have more utility to the user.

5. Other Approaches

Since the idea of a remote access system has been in existence for several years, a number of approaches have been proposed that differ from ours to varying degrees, and we survey some of them here.

Schouten and Cigrang [2003] present a variant of the idea of a remote access system, which allows outstanding versatility, but is also difficult to create and expensive and laborious to maintain. Their proposed system allows users to submit queries by email, written in any of several statistical programming languages. If a query is approved, the user receives the results by email. Before the analysis is performed, an automated system determines the legitimacy of the request, with particularly difficult cases handled manually. As with the MAS, certain types of output are allowed and certain types are not, but since the code is user-generated, rather than generated by the system behind the scenes, it is challenging to identify all unallowable queries. This is especially true because, as the authors emphasize, the validity of a query may depend on information already released as a result of previous successful queries. The authors write, “Computers are simply not fast enough and the construction of a system that fully evaluates the risk of disclosure may be too costly and complex and therefore not feasible.” Thus, in a system like this, it may be necessary to perform some disclosure avoidance analysis on a query after the result of the query has already been returned. This is not ideal, as a query that is a disclosure threat might not be identified until its output has already been provided. However, such a method could be effective if the users are from large institutions and have signed a contract describing their research and pledging to uphold confidentiality. In this case, the fear of a user or institution’s jeopardizing its future access to the data may serve as a sufficient deterrent to deliberate submission of an invalid query. In this type of system, a username and password would be necessary so that individual users' actions could be properly tracked.

A system of the general variety that Schouten and Cigrang [2003] propose has been implemented by the Luxembourg Income Study (LIS), a research institute collecting data on income, wealth and various other measurements, founded in 1983 (see *LIS Micro-data Access* [2009a]). The LIS data are an aggregation of household surveys taken by various contributing countries. LIS’s remote access system—called LISSY—allows registered users to submit their own code via email or an online form, which may be written in SAS, SPSS or STATA. Output, when deemed allowable, is returned by email and is viewable on the form. The system does not allow certain commands that could be used to obtain a disclosure relating to an individual or household. Also prohibited are “sequences of commands and/or variables that would end up breaching the rules on data confidentiality;” these, as well as requests that give overly long output, are flagged for manual analysis or are denied outright. Further specifics are given in *LIS – Micro-data Access – Job Syntax* [2009b]. Schouten and Cigrang [2003] also note that the LIS contains an archive of jobs submitted, which can be further evaluated to make sure the data are being used properly.

Sparks *et al.* [2008] propose a system—Privacy-Preserving Analytics®—that performs a number of methods for disclosure avoidance, including keeping track of the regression models a user requests and ensuring that only a limited (although large) number are run for each possible response variable. They also ensure that a user does not make too many closely related requests.

Gomatam *et al.* [2005] make a distinction between *static servers* and *dynamic servers*. A static server has a pre-determined set of queries to which it will provide an answer. A dynamic server receives a query and makes a decision on whether to provide an answer. A dynamic server—such as the one described in Schouten and Cigrang [2003]—would keep a running record of all previously answered queries, and whenever a new query was submitted, it would be compared against the list to determine whether providing an answer would lead to a disclosure risk when the new answer was combined with previously provided answers. A dynamic server has the highly undesirable property that the order in which queries are submitted by the collective group of users plays a large role in determining which queries are answered, and that eventually the server reaches a point where no new queries can be answered. Since queries are answered or rejected as they are received, the set of queries that are ultimately answered is not the result of a careful assessment of which analyses would provide the most utility to legitimate researchers while keeping disclosure risk at an acceptable level. Gomatam *et al*. [2005] write that “[w]hether dynamic servers are possible remains an open question.” The MAS is at its heart a static server, since it operates under a set of rules that do not depend on previous queries. However, it operates in a dynamic fashion, since the rules are checked for each new query that is submitted, rather than comparing it to a pre-computed list, as creating such a list would be prohibitive. In a way, the MAS does not fit into the framework of Gomatam *et al*. [2005], as it sometimes will provide regression output that is less detailed than the user might have liked instead of refusing output altogether.

Another approach to protecting privacy from a query-accepting statistical database is to suppress from any tables any cells that are deemed a disclosure risk, either directly or indirectly. Adam and Worthmann [1989] discuss this possibility and note that in certain systems, cell suppression is not a feasible solution to the disclosure problem.

5. Future Work

The MAS will continue to be developed within DataFERRETT. We will soon be testing the software itself and the confidentiality rules within the MAS beta prototype to ensure that they properly uphold disclosure avoidance standards. In addition, we plan to draft a set of confidentiality rules for cross-tabulations, and to add different types of statistical analyses within the system, such as descriptive statistics and significance tests. We also plan to modify the system to deal better with missing values. In addition, we will explore other intruder tactics and determine what rules must be put into place to prevent their success.

References

N. Adam and J. Worthmann. Security-control methods for statistical databases: a comparative study. *ACM Computing Surveys (CSUR)*, 21(4):515-556, 1989. ISSN 0360-0300.

M. Chaudhry. Overview of the Microdata Analysis System. Statistical Research Division internal report, U.S. Census Bureau, 2007.

I. Fellegi. On the question of statistical confidentiality. *Journal of the American Statistical Association*, 67(337):7-18, 1972. ISSN 0162-1459.

I. Fellegi, S. Goldberg and S. Abraham. *Some Aspects of the Impact of the Computer on Official Statistics*, Dominion Bureau of Statistics, 1969.

S. Gomatam, A. Karr, J. Reiter and A. Sanil. Data dissemination and disclosure limitation in a world without microdata: a risk-utility framework for remote access analysis servers. *Statistical Science*, 20(2):163-177, 2005. ISSN 0883-4237.

S. Keller-McNulty and E. Unger. A database prototype system for remote access to information based on confidential data. *Journal of Official Statistics*, 14:347-360, 1998.

LIS Micro-data Access. Luxembourg Income Study, http://www.lisproject.org/data-access/lissy.htm, 2009a. Accessed May 10, 2011.

LIS Micro-data Access – Job Syntax. Luxembourg Income Study, http://www.lisproject.org/data-access/lissy.syntax.htm, 2009b. Accessed May 10, 2011.

J. Lucero. Confidentiality rules for universe formation and geographies for the Microdata Analysis System. Statistical Research Division Confidential Research Report CCRR-2009/01, U.S. Census Bureau, 2009.

J. Lucero. Confidentiality rule specifications for performing regression analysis on the Microdata Analysis System. Statistical Research Division Confidential Research Report, U.S. Census Bureau, 2010.

J. Lucero, L. Zayatz and L. Singh. The current state of the Microdata Analysis System at the Census Bureau. In *Proceedings of the American Statistical Association, Government Statistics Section*, 2009.

J. Reiter. Model diagnostics for remote access regression servers. *Statistics and Computing*, 13(4):371-380, 2003. ISSN 0960-3174.

J. Reiter and C. Kohnen. Categorical data regression diagnostics for remote access servers. *Journal of Statistical Computation and Simulation*, 75(11):889-903, 2005. ISSN 0094-9655.

A. Reznek. Disclosure risks in cross-section regression models. In *Proceedings of the Section on Government Statistics, JSM*, 2003.

A. Reznek and T. Riggs. Disclosure risks in regression models: some further results. In *Proceedings of the Section on Government Statistics, JSM*, 2004.

S. Rowland and L. Zayatz. Automating access with confidentiality protection: The American FactFinder. In *Proceedings of the Section on Government Statistics*, 2001.

B. Schouten and M. Cigrang. Remote access systems for statistical analysis of microdata. *Statistics and Computing*, 13(4): 381-389, 2003. ISSN 0960-3174.

R. Sparks, C. Carter, J. Donnelly, C. O'Keefe, J. Duncan, T. Keighley and D. McAullay. Remote access methods for exploratory data analysis and statistical modelling.: Privacy-Preserving Analytics®. *Computer Methods and Programs in Biomedicine*, 91(3):208-222, 2008. ISSN 0169-2607.

P. Steel. Design and development of the Census Bureau's Microdata Analysis System: Work in progress on a constrained regression server. Presentation at Federal Committee on Statistical Methodology Policy Seminar, December 2006.

P. Steel and A. Reznek. Issues in designing a confidentiality preserving model server. *Monographs of Official* Statistics, 9:29, 2005.

D. Weinberg, J. Abowd, P. Steel, L. Zayatz and S. Rowland. Access methods for United States microdata. Paper for Institute for Employment Research Workshop on Data Access to Micro-Data, Nuremberg, Germany, August 2007.

J. You. Data-driven quality-preserving methods for synthesizing microdata on a remote-access regression server. Unpublished manuscript, 2010.