Synthetic Data Disclosure Control for American Community Survey Group Quarters

Rolando Rodríguez
Statistical Research Division, U.S. Census Bureau, 4600 Silver Hill Rd., Washington, DC, 20233

1. Introduction

Abstract

Last year, we reported on an effort to construct a disclosure control method using partially-synthetic data for American Community Survey (ACS) group quarters. This effort was in anticipation of the first planned release of ACS group quarters public-use microdata samples. As ACS data for 2006 becomes available, we have been able to test our method for the first time on a full-size group quarters sample. We give results of our test, along with discussions of new modeling methods, issues faced in the synthesis workflow, and possibilities for future research.

KEY WORDS: disclosure, synthetic data, ACS

This report is released to inform interested parties of ongoing research and to encourage discussion of work in progress. Any views expressed on statistical, methodological, technical, or operational issues are those of the author and not necessarily those of the U.S. Census Bureau.

Statistical agencies must live with compromises: between survey size and cost, between computational time and program accuracy, between model complexity and interpretability, and so on. When preparing survey data for public consumption, agencies must compromise between the statistical utility of the data and the confidentiality of survey respondents. This "risk-utility tradeoff" is a key issue in data release and at the heart of statistical disclosure-control practices.

U.S. law prohibits the Census Bureau from publicly releasing data or data summaries that can compromise the confidentiality of survey respondents. For household surveys, a respondent's confidentiality is compromised when data users can successfully assign publicly-released information back to the respondent. The information can range from demographic variables (age, sex, race) to survey participation itself. Protecting confidentiality in data slated for public release is a key role of disclosure control.

Disclosure control is particularly important for microdata releases. Microdata consists of samples of survey responses, as opposed to aggregated measurements given in tables. A properly constructed microdata sample allows data users to perform statistical analyses without the aid of the releasing agency. Since microdata contains actual survey responses, care must be taken to limit the level of detail available. Although allowing such directly identifying information such as names and addresses to remain in the released sample would be an obvious and egregious violation of confidentiality, other seemingly innocuous information could be combined to identify a respondent.

The focus of our research is protecting the confidentiality of survey respondents in the American Community Survey (ACS), specifically those respondents residing in group quarters. Groups quarters (GQ) are facilities that house many often non-related residents. Examples are college dormitories, nursing homes, and prisons. In 2007, the public will see the first release of ACS GQ microdata, in the form of ACS public-use microdata samples (PUMS). Our task has been to investigate new methods for providing disclosure control for ACS group quarters, beyond the techniques currently

used to protect respondents in the ACS household sample.

Our method for providing disclosure control involves using statistical models to modify respondent records that we feel pose a disclosure risk. The models generate new variable values, called synthetic data, that attempt to mask individuals while maintaining the overall statistical properties of the PUMS. Since we last reported on our project, we have utilized new methods for synthetic data generation and have obtained the complete ACS GQ sample for testing.

2. Disclosure Procedures

Numerous methods of disclosure control exist, including: topcoding, which censors values for a variable that exceed a prescribed upper bound; swapping, which matches records on a set of attributes and then swaps variable values between them; suppression, which simply blanks those values that are deemed risky; and synthetic data, which replaces risky data with new values that are generated from models.

We have used a synthetic-data approach throughout our research; that is, we fit statistical models to the data and then draw from those models to obtain synthetic values for at-risk variables and records of our choosing. The goal of such a procedure is to produce values that are modified enough, per record, to remove disclosure risk, but that have enough distributional similarity to the original data to maintain its statistical properties. More succinctly, we want to change individual attributes but preserve population (and sub-population) attributes. To understand this goal, it is helpful to think of two opposing pathological models. A fully saturated model would refit the data perfectly, but would thus afford no protection from risk. A completely random model (where values are drawn uniformly across the variable support independently for each variable) would offer excellent risk protection, but would mar the statistical properties of the data, especially the multivariate properties. We therefore seek a median approach, governed by the risk we perceive in the data and the statistical integrity demanded by our data users.

Our initial attempts at synthetic-data production used simple models: a Bayesian linear regression for continuous variables, and a Bayesian multinomial model for categorical variables. Although simple to understand and execute, these models did have a number of noticeable and significant drawbacks. The linear regression model tended to give significant numbers of synthetic values outside of the support of the original data, which forced us either to censor those values to a bound or redraw until they lay within the support, increasing computational burden. The multinomial model had the drawback of being limited as to the number of variables we could model simultaneously, since the number of cells to be estimated by the model increases at least exponentially with the number of included variables. This added difficulty to the maintenance of multivariate relationships within the data.

Our latest modeling methodology makes use of several statistical devices, namely: additive regression, bootstrapping, and predictive mean matching. The R function aregImpute, in library Hmisc, is the computational backbone for our methodology. (see Harrell, 2007). The method can be seen as a hybrid between purely synthetic data methods (such as fully-Bayesian regression) and donor-based systems (such as hot-deck imputation and swapping). It was originally intended to impute values for missing data, but with

minor adaptations it is well suited for synthesizing data for disclosure control.

We first discuss the method as it is intended for missing data. The input to the method is a list of variables, including variables containing missing data and variables to be used as predictors. The algorithm proceeds through each variable having missing values, using a combination of techniques to complete the variable. Once completed, the imputed values are used in any future calculations involving the variable.

For a particular variable requiring imputation, the first stage in the algorithm involves taking a full-size sample with replacement from those observations in the data that do not have a missing value for the variable. This bootstrapping step introduces variability into the model fit, since the algorithm used to estimate the model is otherwise deterministic. As such, this step is useful for allowing this method to be used in the context of multiple imputation, (see Reiter, 2003a).

The next stage involves fitting an additive regression model to the sample. This model is of the form:

$g(y) = \sum_{i=1}^{k} f_i(x_i)$, where:

- $g(y)$ is a transformation of the response variable y
- $f_i(x_i)$ is a transformation of the predictor variable x_i

We can view this model as stating that there are transformations of the response and predictors such that the mean transformed response is a linear function of the transformed predictors. The aregImpute function uses the alternating conditional expectation (ACE) algorithm to estimate these transformations.

ACE operates by finding transformations that maximize the variance explained by the linear regression of the transformed response on the transformed predictors. The fraction of variance not explained by the transformed regression is:

$$e^2 = \frac{E\left\{\left[g(Y) - \sum_{i=1}^{p} f_i(x_i)\right]^2\right\}}{E\left[g^2(Y)\right]}$$

Where $g(Y)$ is the transformation of the response and $f_i(x_i)$ is the transformation of predictor i.

The functions g and f_i that minimize this value are considered to produce the optimal model fit. Breiman and Friedman (1985) show that the ACE algorithm converges to these optimizing functions. The algorithm iteratively minimizes e^2 by minimizing the numerator expectation with respect to a single function (either g or an f_i) and then using the minimizing function in the next iteration.

We describe the algorithm in words then give pseudocode. Throughout, it is assumed that the actual distributions of the response and the predictors are known, and that conditional expectations can be derived. The algorithm also assumes that all initial distributions have zero mean, and that the response has unit variance.

The algorithm initializes g to the standardization of Y and initializes each f_i to the zero function. After initialization, the algorithm proceeds through two loops. The inner loop minimizes the value of e^2 for each x_i, and stores the minimizing function, $f_i(x_i)$, as the current transformation of x_i. This loop is iterated until the value of e^2 fails to decrease (that is, the minimal value for an iterate exceeds the minimal value from the previous iterate). The outer loop minimizes

the value of e^2 for Y, using the values of $f_i(x_i)$ determined by the inner loop; again the loop is iterated until the value of e^2 fails to decrease. Maximizing values for each loop are given by Breiman and Friedman (1985).

The algorithm pseudocode is given in Figure 1. The final values of the functions are used as the optimal transformations. More information on the algorithm, including convergence, can be found in Breiman and Friedman (1985).

The algorithm assumes that the distributions of the response and all predictors are known, and that conditional expectations can be calculated. This is never true in practice, and so we must replace the population quantities with their sample estimates (see Breiman and Friedman (1985)). In particular, estimating the conditional expectations requires careful consideration. When the conditioning variable is categorical, Breiman and Friedman give the following straightforward estimate:

$$\hat{E}[X|Z=z] = \frac{\sum_{z_k=z} x_k}{\sum_{z_k=z} 1}$$

where Z is the categorical conditioning variable. This amounts to taking the mean of the values of X that have an associated value of z on the conditioning variable.

When the conditioning variable is ordinal or continuous, more complicated estimates are needed. A simple method for accomplishing this would be to use a linear regression to estimate the conditional expectation. This is unrealistic, however, since the reason for fitting an additive model is that we do not believe the assumptions of a linear regression hold. Breiman and Friedman instead base their estimation on scatterplot smoothers. Specifically, they chose for implementation the so-called super-smoother of Friedman and Stuetzle (1982). Other options are possible, such as lowess fitting and kernel smoothing. Scatterplot smoothers attempt to estimate the conditional expectation via numerical means by fitting a curve through the plot of the conditioning variable versus the dependent variable. The value of the curve at a specific value of the conditioning variable is the estimate of the conditional expectation. The specific details of smoothing algorithms are beyond the scope of this paper; please see Breiman and Friedman (1985) and Friedman and Stuetzle (1982).

Once the ACE algorithm converges, the resulting additive model is used to obtain predicted transformed values of $g(Y)$ for every observation, including those with an initially missing value of Y. To obtain the final synthetic value, aregImpute then performs predictive mean matching, given the values predicted by the additive model. For a given observation, say k, with a missing value of Y_k, the matching is accomplished as follows:

1. Use the ACE-derived model to obtain $g(Y_1^*), \ldots, g(Y_N^*)$, the predicted transformed values of the response for all observations
2. Input these predicted values into a weighting function to obtain weights for each observation.
3. Select an observation at random, drawing from a multinomial distribution using the weights to determine the probabilities.

Once the matching record is found, its original Y-value is used as the synthetic value for observation k.

The weighting function used in aregImpute is:

$$W_k(g(y_i)) = \begin{cases} \left(1 - \left(\dfrac{|g(y_i) - g(y_k)|}{\frac{h}{n} \cdot \sum_j |g(y_j) - g(y_k)|}\right)^3\right)^3 \\ \text{, when } |g(y_i) - g(y_k)| < \dfrac{h}{n} \cdot \sum_j |g(y_j) - g(y_k)| \\ 0 \\ \text{, otherwise} \end{cases}$$

where:
. $g(y_k)$ is the predicted transformed value for the observation to be synthesized
. $g(y_i)$ is the predicted transformed value for observation i
. h is a smoothing parameter
. n is the number of records

This weighting function assigns positive weight to those observations whose predicted transformed values lie within the average absolute distance to the target record's value. The weights are then used to select a record to act as a donor for the synthetic value of the current missing record. This is accomplished as follows for a given missing record k:

1. Take a random draw, u_k, from a uniform distribution on the interval (0 , 1)
2. Calculate the sum of the weights for all possible donor records:

$$s_k = \sum_{i=1}^{n} W_k(g(y_i))$$

3. Standardize the weights by the sum:

$$Z_k(g(y_i)) = \frac{W_k(g(y_i))}{s_k}$$

4. Add up these standardize weights until the sum exceeds u_k. The record at which this occurs is chosen as the donor.
5. Use the donor's untransformed value of Y as the synthetic value of the missing record.

Since we are not using aregImpute for missing data, but rather complete data that requires disclosure control, we must modify the workflow for inputting the data into the function. Our strategy is to stack a copy of the data on top of itself. One copy of the data is complete, while the other copy has had the variable to be synthesized blanked (set to missing). This stacked data is then input into aregImpute, which completes the missing values in the blanked copy. We use these completed values as the synthetic data for those records flagged as requiring disclosure control.

3. Defining Risk

Our attempts at using statistical models to mask data at risk of disclosure would be in vain if we did not consistently and adequately identify those records where confidentiality might be compromised. Defining what is "at risk" is not trivial. Numerous aspects of the data release: sample size, number of variables, level of detail, universe sampling scheme, etc., can have drastic effects on confidentiality as well as data quality. In addition, the information held by prospective data users, both gregarious and malicious, plays a significant role. A major role of the public-data steward is therefore to produce and apply a method of risk identification that takes all these factors into account.

An important first step is determining how a data user might use his data to unmask the identities of individuals in public-use microdata. For our purposes, we assume that the data user has compiled a data set that contains identifying information (names, addresses, social security numbers, etc.), along with variables that mirror variables found in the public-release data. To match records, the data user searches for records in the two data sets that agree on a

set of variables common between the data sets. An identity disclosure occurs when the data user correctly ascribes a record in the public-use data with a record from his data, thus yielding new information about a specific respondent. By "correctly" we mean that the data user obtains new and correct information about a respondent by extracting variables from the matched record.

From this basic definition we see two ways in which our disclosure control procedures can ameliorate the risk of disclosure: by preventing record matching and by modifying true values in the case a match is made. We therefore focus our procedures on those aspects of the data that users will use for matching and those that are sought by users for augmenting their own records. This means we must identify those variables that could potentially be used for matching and those variables that are commonly considered sensitive to respondents but are not necessarily used for matching.

Given a set of matching variables, we assume that the user identifies records as follows: he first generates the multi-way tabulation of possible combinations of the variable values based on the public-use data. For each record in his data, he obtains the associated matching variable combination and identifies the appropriate cell in the table. All records in the public-use data that fall within that cell are matched to the user's record. For further discussion, we refer to those records in the matching cell as "matching records" and the record chosen from the user's data as the "reference record". The number of records in the matching cell is vital to our concept of risk. We first consider possibilities for matching in the absence of disclosure control.

If the matching cell contains no records, then the user can assume several possibilities: the respondent associated with the reference record was not in the survey sample; the respondent was in the sample but not subsequently included in the public-use data; the matching variables are recorded with error in either or both the data sets; or the coding scheme for the matching variables differs between the two data sets. In any case, the user has no matches and must therefore consider re-selecting or recoding his set of matching variables.

The case of one matching record in a cell is more intriguing. We call such a record a "sample unique." Ostensibly, sample uniques give the user exactly what he wants: a single matching record from which he may extract new information; however, there are underlying difficulties. The fact that a record is unique in a cell in the sample does not imply that the record is unique in a cell in the population. If the matched record is not a population unique then there is no guarantee that the information in the record, other than the matching variables, corresponds with true information for the reference record. Thus to effectively draw new information based upon a sample unique, the user must have a certain degree of confidence that either the record is also unique in the population, or that the other associated records in the population share the sample record's characteristics on those variables chosen for extraction. Skinner et al. (1990) give a thorough overview of disclosure risk in microdata.

Disclosure control procedures complicate the matching process by producing uncertainty as to the validity of the matching records. In the presence of disclosure control, a data user having matches must consider several possibilities for errors in matching:

1. The matching is incorrect because the matching variables used have been modified by the disclosure control procedure.
2. The matching is correct, but the extra information contained in the matching record(s) has been modified by the disclosure control procedure
3. Both 1 and 2 have occurred.

We can see how disclosure control procedures can add multiple layers of protection to a publicly-released data set. If we modify those variables that users commonly use for matching records, we will produce uncertainty in their matches, thus throwing into question the validity of any extra data they might obtain from matched records. But even if data users use unmodified variables for matching, if we have used our disclosure control method to alter other variables in the data, then the user still cannot guarantee validity on a per-record basis.

4. Workflow

The production and release of public microdata involves numerous stages, and agencies must decide at what point the application of disclosure control methods should occur. The interaction of disclosure control with other data quality procedures has consequences for overall data quality and for the complexity of the workflow and the individual components thereof.

Editing is a common element of the ACS workflow that can have intimate interactions with disclosure control. Editing is the process of deterministically correcting errors in variable values. Often the errors involve nonsensical, contradictory, or ineligible values for an individual record which conflict with the record as a whole.

One of the main goals of editing is to make variables in the data set conform to so-called universe definitions. These definitions define which values of a variable are allowable for various subsets of the data. For example, the universe definition for variable X might required that all records with a value on another variable Y, say Y = 1, should have a missing value of X. Thus if a record contains Y = 1 and X is not missing, the editing procedures would change either the value of Y or X, or both, to satisfy the universe definition.

We considered three options for including both editing and disclosure control into the workflow:

1. Apply disclosure control first, allowing the editing procedure to correct conflicts between universe definitions and synthetic values
2. Apply disclosure control first, ensuring that synthetic values adhere to the universe definitions in the modeling
3. Apply editing first and ensure that the disclosure control procedure maintains universe definitions

The first option is the easiest in terms of applying the disclosure control procedure. Since the editing procedure will correct out-of-universe values, any model we used to synthesize data for disclosure control would not need to inherently restrict the range of drawn values. If this framework is used, caution must be taken to insure that synthesized values be clearly marked as such, otherwise one risks confounding original out-of-universe values and those generated by the disclosure control model. This could prevent internal analysis of patterns of universe-related problems.

The second option complicates the disclosure procedure by requiring the models to produce synthetic values congenial with the universe definitions. Despite this added complication, we feel this is a better option than the first, since it forces us to produce models that yield synthetic values that more closely match the reality of the data (assuming the universe definitions are sensible).

These options have a common problem that makes the third worth investigating. When performing disclosure control before editing, we run the risk of fitting models to possibly erroneous data. We never expect the data as a whole to be of low quality; however, certain variables that we chose to model might be afflicted by profligate universe errors. This can have a significant impact on the validity of our model, as we are fitting to data that possibly does not match reality. Although the editing procedure will correct values from the disclosure model, we could not say that the model would have produced vastly different synthetic values had we performed synthesis after editing.

Thus we chose the third option, which is to perform disclosure control after editing. The difficulty in this case, as in the second, is that the disclosure model cannot be allowed to generate out of universe synthetic values. The synthesis models, however, are based on edited data, which can help reduce the number of out-of-universe values as well as give the models added validity.

The decision to perform disclosure control after editing played a large part in our decision to use aregImpute as our modeling procedure. The predictive mean matching step ensures that the final synthesized value will not lie outside of the support of the data used for fitting. This means that as long we fit the models to data that already satisfy the universe definitions, we will not produce out-of-universe values. To guarantee this happens, we perform our modeling within subpopulations determined by the universe definitions. As long as each subpopulation is associated with a unique universe, conflicts will be avoided.

Our work focuses on generating synthetic data for release in the ACS PUMS; however, sampling for PUMS is performed after synthetic data generation. This is advantageous for two reasons. First, it allows us to fit our models on the data as a whole, affording us larger sample sizes and all the benefits thereof. Second, it allows the data to be sampled multiple times without the need for reapplying disclosure control. This is advantageous if ACS GQ records are to be included in any future microdata releases other than the 2007 PUMS.

We would like to note two other important aspects of the workflow. First, the disclosure procedure we have discussed has currently only been planned for use in the 2007 ACS PUMS, and only for group quarters respondents. There has not been an official decision as to the use of this procedure for multi-year estimates and for household data. Second, the modeling procedure was run before the application of survey weights, which implies that these weights were not available for use in modeling.

5. Results

As this time, the list of variables considered for synthesis is confidential, which limits the scope of results that may be presented. We first analyzed an unweighted synthesized continuous variable. For this variable, there were no significant differences in the mean,

quartiles, or the extreme observations between the synthetic and original data. This last observation is particularly important. Topcoding is a common method of data protection for public-use files, which provides protection by setting a maximum-allowable value for a given sensitive variable. The use of topcoding prevents analysis of the extreme observations of the variables to which it is applied. Model-based methods for disclosure can generate new data in these extreme regions, allowing for statistical analysis while affording protection.

In terms of the unweighted synthesized categorical variables, we found a kappa statistic of 0.9720 for the agreement of the multi-way tabulation of these variables between the original data and the synthetic data. This indicates that the original and synthetic data agree on the distribution of these variables.

As another measure of statistical validity, we performed an unweighted regression of a synthesized continuous variable on a combination of synthetic and non-synthetic variables. Results for this are given in Table 1 (see below). We see no appreciable differences between the coefficient estimates and their standard errors, as well as no change in their significance.

Other analyses performed internally show equally positive results in terms of statistical integrity. Results from re-identification experiments (not shown here), indicate that the synthetic data provide the necessary disclosure protection required for public-use data. We feel confident in presenting this method as a viable option for providing disclosure control to ACS group quarters.

6. Future Research

Research continues into the use of synthetic data for the ACS. Foremost is the possibility of using this method to provide protection to individuals in households (who are currently protected in ACS PUMS via data swapping). Household data complicates the modeling process by requiring the maintenance of demographic relationships among family members, which is not an issue in group quarters.

Another major focus of research is the use of multiple imputation within the synthetic data framework for ACS. Multiple imputation involves releasing several copies of the synthetic data; this allows data users to estimate the variability added by the synthesis model, affording accurate standard errors (see Rubin, 1987 and Reiter, 2003).

Other areas of research include improvements to the process of flagging records as needing disclosure protection. Currently a record is flagged on the complete cross-tabulation of a list of key identifying variables. This gives us no information on which particular variables are putting a particular record at risk. We are investigating the option of flagging records based upon minimal combinations of the key variables, which would allow us to avoid synthesizing variables that do not put the record at risk.

Finally, since the aregImpute function was originally intended to impute missing data, we would like to investigate the use of this method to perform item non-response imputation for the ACS. Having the same procedure for both imputation and disclosure could possibly simply both our internal workflow and user analyses.

7. Acknowledgements

The author would like to thank: Sam Hawala, Yves Thibaudeau, Laura Zayatz, Mark Asiala, John Stiller, and other colleagues from the Census Bureau, and Jerry Reiter from Duke University for their contributions to the development of this method.

8. References

Breiman, L., Friedman, J.H. (1985). Estimating Optimal Transformations for Multiple Regression and Correlation. *Journal of the American Statistical Association.*

Fox, J. (2002). Nonparametric Regression. *Appendix to An R and S-PLUS Companion to Applied Regression.*

Friedman, J.H., Stuetzle, W. (1982). Smoothing of Scatterplots. *Technical Report Orion 003.* Stanford University.

Gomatam, S., Karr, A., Liu C., Sanil, A. (2003). Data Swapping: A Risk–Utility Framework and Web Service Implementation. *National Institute of Statistical Sciences Technical Report.*

Frank E Harrell Jr and with contributions from many other users. (2007). Hmisc: Harrell Miscellaneous. *R package version 3.4-2*.

Little, R. (1988). Missing-Data Adjustments in Large Surveys. *Journal of Business and Economic Statistics.*

Skinner, C.J., Marsh, C., Openshaw, S., Wymer, C. (1990) Disclosure Avoidance for Census Microdata in Great Britain. *Proceedings of the 1990 Annual Research Conference*. Census Bureau.

Reiter, J. P. (2003a). Inferences for partially synthetic, public use microdata sets. *Survey Methodology*.

Rubin, D.B. (1987). *Multiple Imputation for Nonresponse in Surveys*. Hoboken: John Wiley & Sons.

Tibshirani, R. (1988). Estimating Transformations for Regression Via Additivity and Variance Stabilization. *Journal of the American Statistical Association.*

Figure 1: Pseudocode for the ACE Algorithm

Assume $E[g^2(Y)] = 1$ (this can be guaranteed by first transforming Y by subtracting its mean) and that all functions have expectation of zero:

Initialize:

$g(Y) = Y \Big/ E\left[Y^2\right]^{\frac{1}{2}}$

(standardize the response)

$f_i(X_i) = 0$, $i = 1, \ldots, p$

(initialize f to zero function)

Outer Loop (iterate until e^2 fails to decrease):

Inner Loop (iterate until e^2 fails to decrease):

For k = 1 to p:

$$f_{k,1}(X_k) = E\left[g(Y) - \sum_{i \neq k} f_i(X_i) \middle| X_k \right]$$

(compute expectation of the difference of the transformed response and the transformed predictors)

$f_k(X_k) = f_{k,1}(X_k)$

(set the current f to this expectation)

End;

End;

$$g_1(Y) = \frac{E\left[\sum_{i=1}^{p} f_i(X_i) \middle| Y \right]}{E\left\{ E\left[\sum_{i=1}^{p} f_i(X_i) \middle| Y \right]^2 \right\}^{\frac{1}{2}}}$$

(compute expectation of a linear regression of the sum of the transformed predictors on the transformed response. Standardize this expectation (which is a function of the response))

$g(Y) = g_1(Y)$

(set the current g to this expectation)

End;

Table 1: Linear Regression of Continuous Synthesized Variable

	Estimate		Estimated Standard Error		Critical Value	
Effect	Original	Synthetic	Original	Synthetic	Original	Synthetic
Intercept	*	*	0.2088	0.2089	<.0001	<.0001
Variable 1	-3.3940	-3.4102	0.0714	0.0714	<.0001	<.0001
Variable 2	0.7986	0.8115	0.1571	0.1571	<.0001	<.0001
	1.4593	1.4305	0.3083	0.3085	<.0001	<.0001
	-0.7410	-0.7220	0.3866	0.3867	0.0552	0.0619
	5.2827	5.3050	0.2362	0.2363	<.0001	<.0001
Variable 3	3.4346	3.4362	0.0638	0.0638	<.0001	<.0001
	0.7353	0.7461	0.2745	0.2746	0.0074	0.0066
Variable 4	-10.6433	-10.6375	0.1080	0.1081	<.0001	<.0001
	-30.6871	-30.6650	0.2100	0.2102	<.0001	<.0001
	29.5641	29.5483	0.1085	0.1086	<.0001	<.0001
	-11.3107	-11.3478	0.5563	0.5566	<.0001	<.0001
	-28.6310	-28.6374	0.1096	0.1097	<.0001	<.0001
	-21.1350	-21.1322	0.1705	0.1706	<.0001	<.0001