Automated Synthetic Data Validation: Applying Noise Injection for Disclosure Avoidance

Saimun Habib[1], Bianica Pires[1],
Gary Benedetto[2], Rolando Rodriguez[2], Jordan Awan[3], Jordan Stanley[2],
Evan Totty[2], Giuseppe Germinario[2], Rich Stevenson[1]

[1]The MITRE Corporation, 7515 Colshire Drive, McLean, VA 22102
[2]U.S. Census Bureau, 4600 Silver Hill Road, Washington, DC 20233
[3]Purdue University, 150 N. University Street, West Lafayette, IN 47907

Abstract
The U.S. Census Bureau is exploring ways to modernize its disclosure avoidance methods. Of particular interest is increasing the use of synthetic microdata to alleviate privacy loss concerns. To improve performance of such datasets and ensure trust in synthetic outputs, researchers can have their synthetic model results validated against the real data and receive non-disclosive results. While the Census Bureau currently has a validation process in place for synthetic data, it is labor intensive. This paper discusses a proof of concept we are developing to validate the synthetic model output and return non-disclosive results based on real data. The approach includes a rule-based automated disclosure review solution and a noise injection methodology that uses synthetic estimates to decrease privacy loss when reporting estimates back to researchers. This is part of a larger effort to create an enterprise synthetic data validation system and will help inform how synthetic datasets impact the Census data user community. With this project, Census can assess performance of synthetic datasets, alleviate concerns regarding synthetic data, and report real results with reduced concerns of privacy loss. We demonstrate a pilot for operationalizing an automated disclosure avoidance system based on privacy rules that shifts the burden away from Census and towards researchers. Furthermore, we show that synthetic data can be used to generate noisy statistics that add more noise as the sensitivity of the statistic increases.

Key Words: Synthetic Data, Noise Injection, Privacy Loss, Disclosure Avoidance, Data Validation

1. Introduction

The U.S. Census Bureau finds itself in the position where it must consider the vulnerabilities of public use microdata given the changing landscape of publicly available data and the potential for linkages across disparate data sources. Differential privacy approaches for survey data are not at a satisfactory state for microdata with sample weighting schemes and potentially hundreds of covariates. To alleviate disclosure concerns on these data, Census intends to release experimental, fully synthetic microdata (Rubin, 1993) in the sense that every value of every variable has been drawn from an estimated probability distribution. As a result, it can be argued that any similarity between a synthetic record and an original record is simply due to random chance. Ideally, a good synthetic data generating function will result in synthetic data that mimics the analytic properties for populations that social scientists want to study. In practice, preserving the myriad of analyses possible with in-depth surveys has proven very challenging. In light of this, Census implemented a service that preserves privacy by only allowing external researchers to directly interact with the low-risk synthetic data to develop their analysis code, having Census staff execute this code on the original data, and releasing to the external researcher only the safely disclosable summaries desired. We call this process “validation.”

Validation alleviates researchers' justifiable concerns in the ability of fully synthetic data to preserve their analyses of interest and offers an opportunity for the data provider to crowd-source the evaluation of the synthetic data with real-life use cases, leading to higher quality synthetic data in the future, and a sense of partnership between the external research community and the data provider. However, this service poses a significant challenge to the data provider as well. While the synthetic data allows the development of analyses and code to be outsourced to the external researcher, the data provider still must run the code internally and review the output to ensure that it meets privacy standards. This part of the process is critical for maintaining the privacy benefits of the synthetic data but is also very labor intensive.

Two Census synthetic data products have already adopted a validation system: the Survey of Income and Program Participation (SIPP) Synthetic Beta (SSB) and the Synthetic Longitudinal Business Database (SynLBD) (Benedetto et al., 2018; Kinney et al., 2014). Both synthetic data products were made available to external researchers on a semi-restricted basis and were accessed through an external (to the Census Bureau) server that was similar to Census internal servers with similar software available for statistical analyses. These similarities in computing environments helped to ensure that the validation process could operate as smoothly as possible.

In this paper, we describe a proof-of-concept automated validation solution. While we sought to automate where feasible, we recognize that keeping the human in the loop is integral to ensuring the required standards for privacy. We pilot a rule-based disclosure review automation process and use noise from synthetic estimates to decrease privacy loss when reporting estimates back to researchers. We look to the multiple imputation background of the synthetic data literature to offer a noise injection strategy that might allow for less labor-intensive review while still offering validation results that can be more trusted than the synthetic data alone. This is part of a larger effort to create an enterprise synthetic data validation system and will help inform how synthetic datasets impact the Census data user community.

2. Synthetic Data Validation Overview

The automated validation solution will be leveraged as an environment to make synthetic replacements for public use microdata available to external researchers. The researcher will be able to access the synthetic data published by Census and perform independent analysis similar to what was done for the SSB and SynLBD. Once the researcher is approved for access to the synthetic data, a virtual workspace will be dynamically provisioned for the researcher to access the data. The researcher will then build models and perform analysis using the synthetic data. If the researcher would like to validate their research results on internal data, they will submit their analysis to Census staff for validation. The Census staff will perform a disclosure review, which is the last box shown in Figure 1 and the focus of this paper.

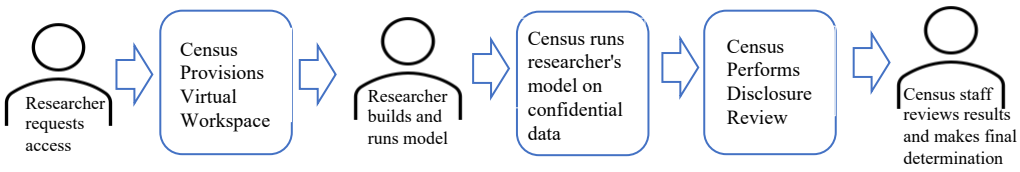

Figure 1: Synthetic data validation process

We focus the paper on two potential approaches currently in development for automating the validation process. However, ensuring that data privacy standards are met will involve integration of the approaches along with human intervention and review where appropriate. The two approaches are a rule-based automation approach for disclosure avoidance (discussed in Section 3) and a synthetic noise injection approach (discussed in Section 4).

3. A Rule-Based Automation Approach for Disclosure Avoidance

In this section, we discuss the proof of concept developed for automating the traditional set of disclosure avoidance (DA) methods that are currently applied manually by Census staff. These DA methods include volume of output, rounding, cell suppression, and population analysis for identifying geographic areas with small populations (GASPs).

Figure 2 provides a high-level depiction of the automated validation solution. After model development, the researcher will generate one or more output files, including the model output and supporting files. If the user is requesting validation, the researcher will upload the output files and provide the necessary metadata through a graphical user interface (GUI). Census staff will then access the researcher's output files, the metadata, and a standardized version of the researchers' model output. This set of information is run through a series of functions that generate "mini reports" with the results from each disclosure avoidance method. These results, in conjunction with the noise injected output (see Section 4), static code analysis, and potentially other privacy evaluation techniques, will be used for final disclosure determination.

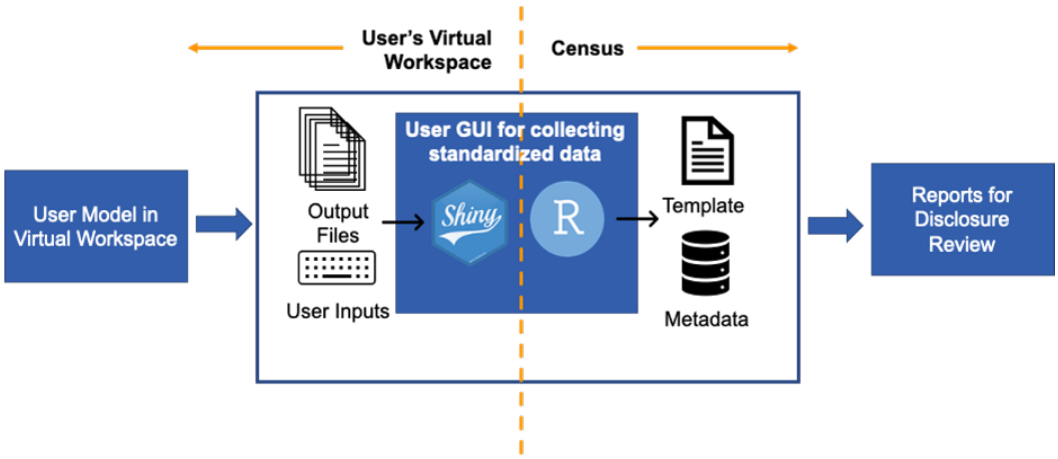

Figure 2: Rule-Based Disclosure Avoidance Automation Process

3.1 Standardized Template for Model Output

A major challenge for automation is processing the researchers' model output and extracting the necessary information for disclosure review. For the proof of concept, we identified two formats in which researchers usually publish their results – (1) a long format where variable names are the column headers and (2) a wide format where variable names are the row names. Assuming model output is in one of these two formats, we break out the researcher's model output into the following three tables:

- The main table: This table identifies the subpopulation for which a statistic is being created. For example, if the researcher is reporting statistics for women over 50 and women under 50 in the state of Virginia, it will create two separate keys, one for each subpopulation.
- The primary estimates table: This table contains the primary statistics associated with each subpopulation. It can include statistics such as mean income, the number of observations, and the most frequent education level.
  The estimate groupings table: This table links primary estimates to any associated estimates, such as measures of variance, degrees of freedom, and p-values.

These set of tables create a hierarchical structure where an analysis can have multiple subpopulations they are considering, each subpopulation has multiple statistics or estimates that the researcher is interested in, and each of those statistics can have other statistics that support them and

their interpretation. From this we can generate disclosure reports and flag areas which may need human oversight (this is discussed further in Section 3.3).

3.2 User Interface

Researcher's will use a GUI to upload their model output, upload any supporting files, and input the necessary metadata. Figure 3 shows a screenshot of the GUI currently developed in RShiny.

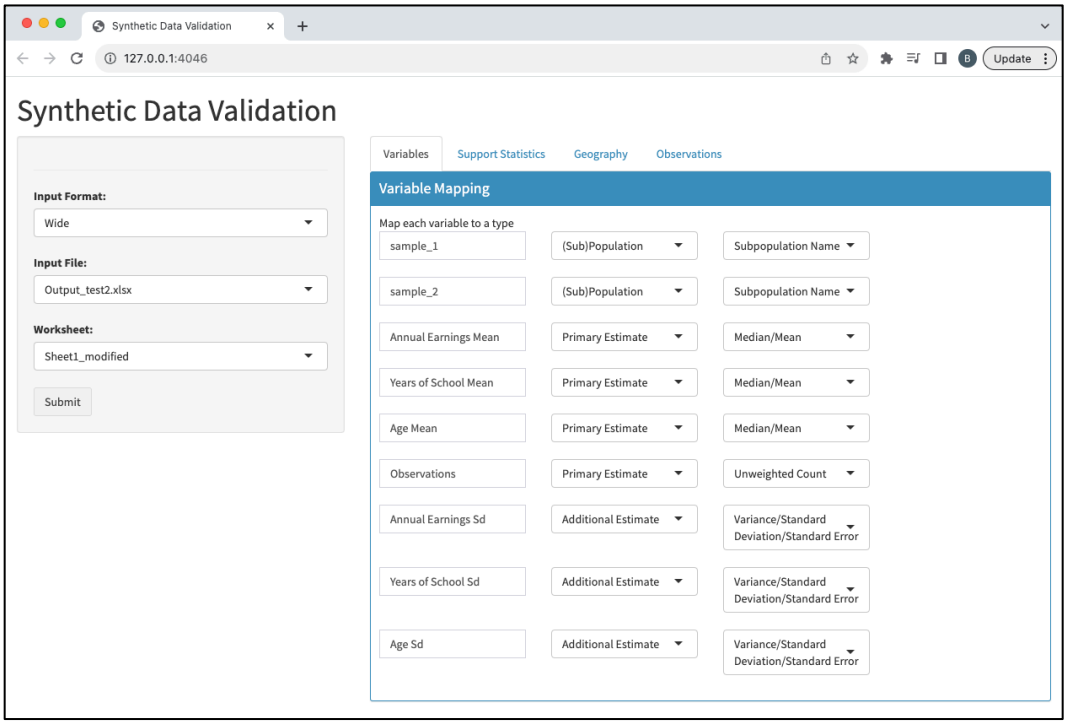

Figure 3: Graphical User Interface

The left panel is where the researcher will input the format of the data (wide or long), select the first file that contains the model data, and select the worksheet needing metadata. For each file and worksheet, the researcher will provide information across the four tabs in the right panel of the GUI. The four tabs are the *variables tab*, the *estimates tab*, the *geography tab*, and the *observations tab*. Each is discussed in detail below.

3.2.1 Categorizing variables in model output

The purpose of the *variables tab* is to determine the variable type in the researcher's model output and where to place it within the standardized template. Each variable is either a subpopulation, primary estimate, or additional estimate. Subpopulations are to be listed in the main table, primary estimates in the estimate table, and additional estimates in the estimate groupings table. Each variable is further categorized into a subtype. Subtypes are categories of statistics used for running the DA methods (e.g., median, standard deviation, proportion).

3.2.2 Identifying estimate groupings

The *estimates tab* is used for mapping primary estimates (as identified in the *variables tab*) with any associated additional estimates, such as measures of variance and degrees of freedom. For example, the standard deviation for age would be mapped to mean age. This is used for generating the estimates groupings table in the standardized template. It also ensures that estimates do not get inadvertently double counted, which is important for determining volume of output and potentially other privacy measures.

3.2.3 Defining spatial and temporal fidelity

The *geography tab* is used to collect metadata on the temporal and geographic aspects of each subpopulation in the data. This tab will appear only if further information is needed, as determined by the researcher's initial data pull. Researchers will use a standardized function (similar to *get_acs* in the tidycensus package in R) through a data access service to pull the initial data for their analysis. Static code analysis will parse out important features, such as the geography and year selected by the researcher. If the geography level requested by the researcher is at the state-level or above (nation), the geography tab will not appear as the data passes population analysis and no further analysis is required. If the data requested is at a geography level below state (e.g., county, tract, block group) however, population analysis will be performed, and the geography tab will appear.

For each subpopulation identified in the *variables tab*, the researcher will provide the associated year(s), geographic level (e.g., tract, block group), and entity type (e.g., person, household, firm), if applicable. The researcher will be further asked to provide the details on the geography included in each subpopulation. If a subpopulation includes tract level data for Fairfax County, Virginia, for instance, the researcher will need to select the exact tracts (or "All tracts") included in that subpopulation. The information will be used to determine if each subpopulation passes population analysis (discussed further in Section 3.3.3).

3.2.4 Determining observations counts

The *observations tab* is used to collect data on the number of unique entities in each subpopulation and in the analysis. This information is used to determine if the analysis passes volume of output and if cell suppression is required. Observation counts are often provided by researchers as a separate support file. In future work, support files will be loaded into the GUI for processing.

3.3 Automation of Disclosure Avoidance Methods

The rules associated with each DA method were coded in R to generate a series of "mini reports" with the results. The standardized version of the researcher's output, the metadata provided through the GUI, and specified thresholds (e.g., volume of output threshold, observations to cell ratio) provide the main inputs into the reports. Figure 4 shows a screenshot of an example report. Each tab in the workbook includes the results for the different DA methods.

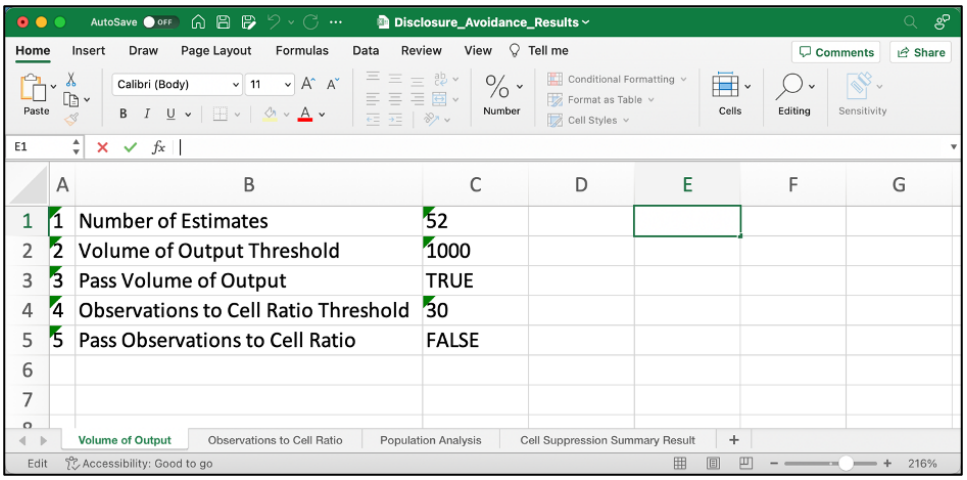

Figure 4: Disclosure Avoidance Report

Each DA report (worksheet in the report workbook) states whether the researchers' output passes the DA method, fails the DA method, or whether further investigation is needed given DA rules and threshold values. In addition, any information that went into that determination is provided, such as counts and thresholds. Because final disclosure determination can be nuanced, the intent behind these reports is to provide the reviewer the supporting data needed to make an informed decision, and not to dictate a final pass/fail determination. Each DA method is discussed below.

3.3.1 Volume of output

Volume of output is determined by counting the total number of estimates and computing the observation to cell ratio and ensuring that both are under a specified threshold. The number of estimates is determined by counting the number of primary estimates in the estimates table. Additional statistics identified that are not mapped to a primary statistic will also be counted as part of volume of output. The ratio of observations to cells is less straightforward as we may not know the number of unique entities across subpopulations that may or may not overlap. For this reason, we compute the ratio for each subpopulation individually. Then we compute the ratio across all subpopulations (in the GUI, we ask that the researcher provide the number of unique entities across all subpopulations). If any surpass the threshold, further investigation is required.

3.3.2 Rounding

Census maintains a Python script that highlights cells in the researcher's output data with potential rounding issues. We updated the Python script to work with the standardized template and call the code from R. In further work, we will generate a report with a summary of the cells needing rounding. We will also investigate ways to refine the rounding code to detect any rounding issues more accurately in the output.

3.3.3 Population analysis

Using the geographic data provided by the researcher through the initial data pull and the GUI, population analysis is performed on each subpopulation. Population analysis passes if the geographies used in the analysis are larger than the smallest congressional district for that year. If population analysis fails, noise injection or cell suppression is required.

3.3.4 Cell suppression

Statistics in the researcher's model output data must be derived from a minimum set of observations. Moreover, implicit samples must also meet this minimum threshold. Cells that do not meet this requirement should be suppressed and complementary suppression should also be investigated. For example, let's say the total count of people within some subpopulation is 6. The researcher includes the count of males, which we will say is 4 for this example. Even though the count of females is not provided, we can deduce that it would at most be 2, which is below a minimum threshold of 3. Because we can derive a statistic with less than 3 observations, the count of males needs to also be suppressed. Implicit samples should be provided by the researcher in the supporting files. However, Census staff may want to verify accuracy.

4. Disclosure Avoidance using Noise Injection

The use of a data synthesizer enables Census to create many synthetic replicates of the data. Between these synthetic replicates, there will be variability in the records and consequently, variability in results when a model is applied to the synthetic data. Census is interested in using this variability to create noisy statistics that can offer protection against disclosure. This is more experimental than the rule-based automated disclosure avoidance approach (see Section 3), but we expect that the protection offered using synthetic replicates will scale with the sensitivity of the statistic we are trying to privatize.

4.1 Framework and Challenges

Our setup for automated noise injection is depicted in Figure 5. In blue are objects the researcher may interact with in their workspace and in red are objects only seen by Census in their Internal Research Environment (IRE). All research teams will have the same copy of synthetic data available to them to create a model. To this same model, Census will pass the real data set and multiple copies of synthetic data as input. This will create a real estimate and an empirical distribution of synthetic estimates. Our task is to take the information from this synthetic distribution, use it to design a noise

distribution, and combine it with the real data to create a protected statistic where the amount of noise (protection) scales with the sensitivity of the statistic.

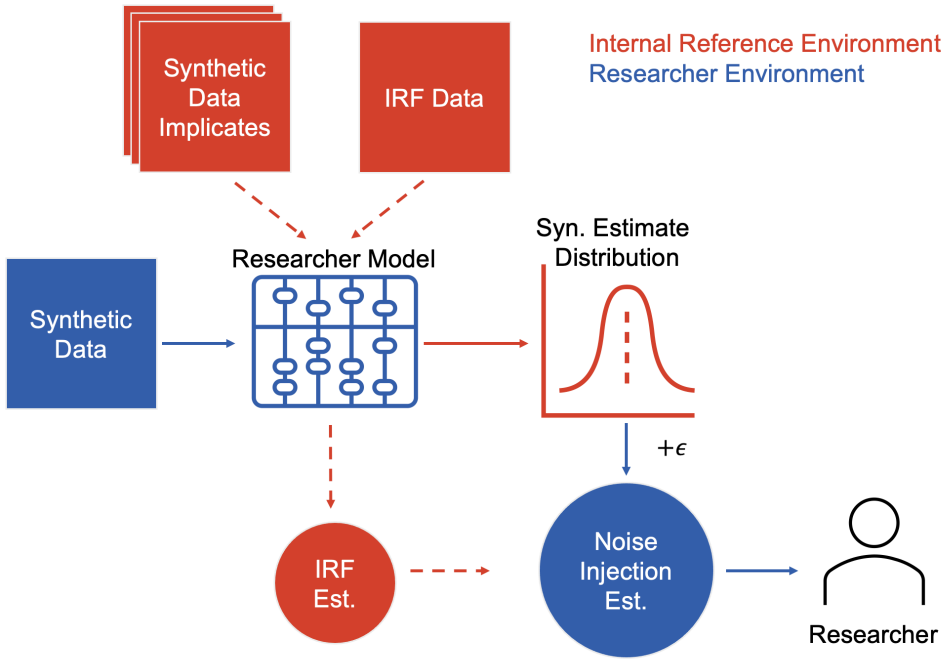

Figure 5: Automated Noise Injection Framework

4.2 Approximating Sensitivity

A major challenge for this tasking is estimating sensitivity. In the traditional differential privacy literature, sensitivity is often short for global sensitivity. Global sensitivity is the maximum amount a statistic differs when calculated between any two datasets differing by one individual's data. For example, a count statistic can only change by 1. Global sensitivity is often unbounded for many statistics that research teams may be interested in, such as regression coefficients where the maximum amount of change can be infinite. In addition, for regression-type queries, global sensitivity can become increasingly difficult to calculate even when it may be finite. Furthermore, the jump from univariate to multivariate regression makes the sensitivity calculation even more difficult. Due to this challenge, an alternative notion of sensitivity is *local* sensitivity, which may be more realistically estimated for the wide variety of uses of real life. Local sensitivity measures the maximum change of the statistic between the dataset at hand (i.e., the real Census data product) and any other dataset that differs by one individual's data; and is thus *local* to the dataset of interest. However, local sensitivity is still a theoretical value which cannot reasonably be derived for all statistics researchers may request, as researchers may request unforeseeable statistics.

To resolve this, we propose an empirical estimate of sensitivity that is fast and robust to a wide variety of statistics. To do so, we borrow heavily from Broderick et. al. (2023). Let $\vec{w}$ be a vector of *data weights* that indicates the number of times an observation is included in the dataset. Broderick et. al. (2023) suggest that a statistic can be viewed as a function, $\phi(\hat{\theta}(\vec{w}), \vec{w})$, of the data weights $\vec{w}$

and model parameters $\hat{\theta}(\vec{w})$ which itself may be a function of the data weights (as shown in Figure 6). We require $\phi(\hat{\theta}(\vec{w}), \vec{w})$ be continuously differentiable in both arguments.

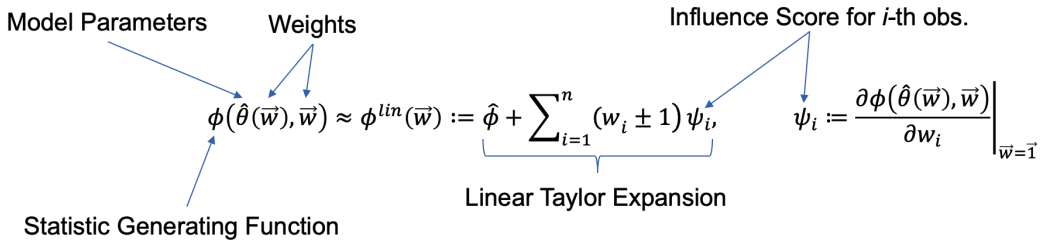

Figure 6: Sensitivity Estimation Equation

We can do a first order Taylor Series around the weights, $\phi^{lin}(\vec{w})$, to get a linear approximation of the statistic as a function of the weights. This linear approximation is equal to the observed statistic, $\hat{\phi}$, plus the change resulting when the weights are changed by 1. Note that subtracting 1 from the observation's weight, w_i, represents dropping it from the data, while adding 1 to w_i represents duplicating the observation. The change for each observation is scaled according to their influence, ψ_i, which is the partial derivative of the statistic generating function with respect to the observation's weight. Sorting these influence scores allows us to identify which observations cause the most change in the statistic when duplicated or dropped, while the linear approximation allows us to estimate the change without recalculating the statistic. This creates a very efficient approximation for local sensitivity of a statistic on a dataset. One important limitation of this approximation is that it only considers deleting or duplicating observations that are already in the dataset of interest, whereas the true local sensitivity would also consider adding observations that may not be present in the dataset.

4.3 Noise Generation Approaches

We looked at two approaches for generating noisy estimates based on estimated sensitivity. In Section 4.3.1, we discuss the first approach, which makes use of CenSyn, a classification and regression tree (CART) model created by Census to create synthetic data, and then applies noise as described in the framework above. In Section 4.3.2, we discuss the second approach, which makes use of a Dirichlet – Bayesian Bootstrap approach to create synthetic data and the protection is provided by the sampling procedure.

4.3.1 Approach 1

The first approach to noise injection relies on the implicate variance between synthetic estimates and the average bias of those estimates. Let there be M replicates of synthetic data, let q_0 be our estimate on the real data, and $q_1, \ldots, q_M$ the estimate on the respective synthetic replicate. Let $\bar{q} = \frac{1}{M}\sum_{i=1}^{M} q_i$ be the synthetic average. Then the average synthetic bias is $b = \bar{q} - q_0$ and the between implicate variance is $S^2 = \frac{\sum_{i=1}^{M}(q_i - \bar{q})^2}{M-1}$. Using these, we can construct a distribution to draw noise ϵ from a Gaussian distribution with mean 0 and a standard deviation of $S + \frac{|b|}{1+S}$, i.e.,

$$\epsilon \sim N\left(\mu = 0, \sigma = S + \frac{|b|}{1+S}\right)$$

This distribution satisfies several desirable properties:

1. As the bias approaches 0, the noise addition has the same scale as the synthetic data estimates.

2. As the between-implicate variance or the bias approach infinity, the scale of the noise also increases to infinity.
3. As the between-implicate variance approaches 0, the bias will provide the privacy protection.
4. As between implicate variance and bias simultaneously approach 0, we will identify an invariant of the synthetic data generation process under some model.

A negative of this approach is that it may be vulnerable to certain attacks, which choose artificial statistics designed to manipulate the bias and between-implicate variance in order to reveal sensitive information. For example, a motivated adversary can construct an analysis that can force the synthetic variability to be 0 and the bias to be fixed. One possible attack is as follows: if an adversary knows they were in the dataset, they may have code that looks for their specific response. The synthetic datasets would contain this response with very low probability and may all return False for this query. The real dataset would return True however, and additional code can be used to return a very biased result if, for example, a separate query of actual interest returns True. The results of this would then indirectly reveal the separate query. This attack has two stages:

1. Determine if the data being queried against is real or synthetic.
2. Ask a second query that will have heavy bias if it returned True/False on the real/synthetic datasets.

The first step in this attack is tricky to execute but is not entirely unfeasible. Static code analysis can help defend against code designed to make that distinction but it itself is not infallible.

4.3.2 Approach 2

Motivated by the vulnerability of the first approach to a potential attack, we also consider a different approach. This approach does not rely on CenSyn to create the synthetic data. Instead, it relies on using a Bayesian bootstrap to create synthetic replicates. The procedure to do so is as follows.

1. For a dataset with n records, take a draw from a Dirichlet distribution with constant vector α, i.e., $Dir(\overrightarrow{\alpha_{1\times n}})$. This draw constitutes a probability vector, v, of length n and we let v_i, the i-th entry in v, be the probability associated with drawing the i-th record in the dataset.
2. We sample n records from the dataset with the probabilities in v, M times to create M synthetic datasets.

With this approach, we do not expect to see every original observation in the synthetic dataset as some will appear more than once. However, we do expect the correlation structure between features to be well preserved. We proceed as expected and apply the same model to all the synthetic replicates. Under this approach, however, we do not inject noise into the real statistic and report that as the protected statistic. Instead, we report $\bar{q}$ from the synthesized datasets as our protected statistic, as the noise is produced by the Bayesian Bootstrap sampling strategy.

This approach may also be susceptible to potential attacks, such as a membership attack, since the Bayesian bootstrap can only create synthetic datasets which are combinations of records already present in the real dataset. Just as with the noise injection approach, human supervision will be necessary to identify and prevent such attacks.

4.4 Evaluation

To evaluate the two approaches, we use the 2021 Public Use Microdata Sample (PUMS) and 11 synthetic replicates of it for an experiment. In this experiment, we fit a model:

$$log(Wages) = \beta_0 + \beta_1 Age + \beta_2 Age^2 + \sum_{i=1}^{5} \beta_{i+2} EducationLevel_i$$

This model predicts the logarithm of a person's wages based off of their age, age squared, and education level. The sample for this model was filtered to males between the ages of 25 and 55 who are not unemployed for the last 5 years, self-employed, or working without pay. This model was chosen as it is a common toy model in the economics literature.

For each dataset (synthetic replicates and real) we

1. Draw a sample of the dataset(s)
    a. Sample sizes range from [10,150] in increments of 1, [160, 1000] in increments of 10, and [1200, 19400] in increments of 200.
2. Run the model on samples:
    a. Record the coefficient estimate produced on the real PUMS sample.
    b. Record the coefficient estimates produced on the synthetic samples.
    c. Calculate the sensitivity when one observation is added/dropped from the PUMS sample.
    d. Generate the noise distribution.

4.4 Results

Initial results with Approach 1 are promising. In Figure 7, we plot each sample and observe as we increase the sample size, the synthetic bias trends towards 0. Here the bias is simply how far the synthetic estimate is from the estimate with the real data. Similarly, we see as the sensitivity increases, the bias increases in variability. The suggests the noise that we generate from the synthetic replicates will be driven by both increasing synthetic variation and bias with respect to the sensitivity.

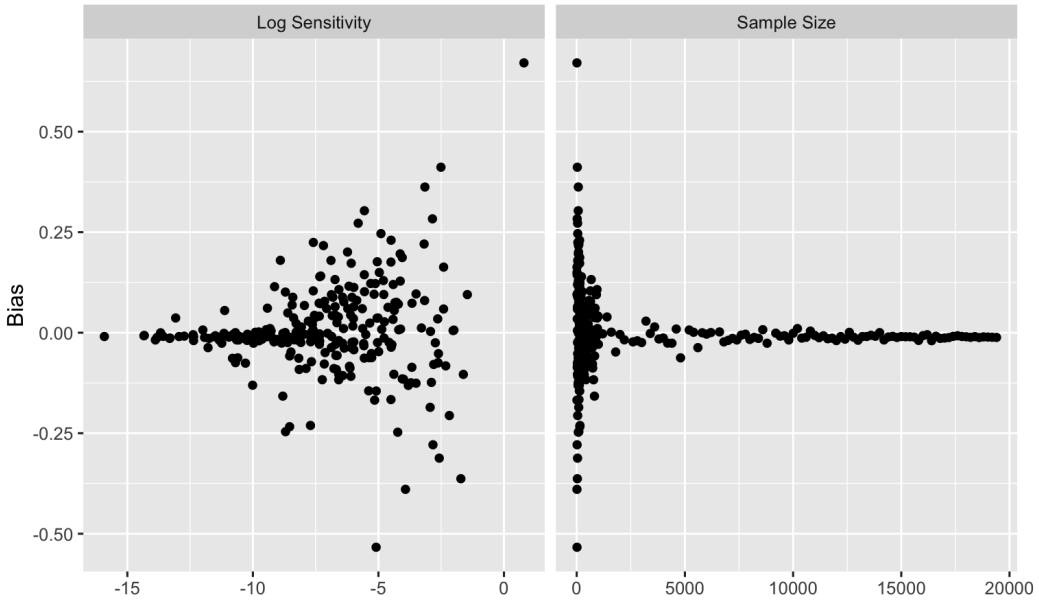

Figure 7: Sensitivity and Sample Size vs. Bias

In Figure 8, we consider the estimate for age on the y-axis and record the result when we run it on a sample from the real 2021 PUMS dataset. On the x-axis we have the estimated sensitivity for the statistic on that dataset. The ribbons around the points represent either one standard deviation of our constructed noise parameter, or the Root Mean Squared Error (RMSE) of the synthetic estimates, respectively. This plot shows that as sensitivity increases, the variability of these estimates increases as well, so the amount of noise that is added to the real statistic will also increase.

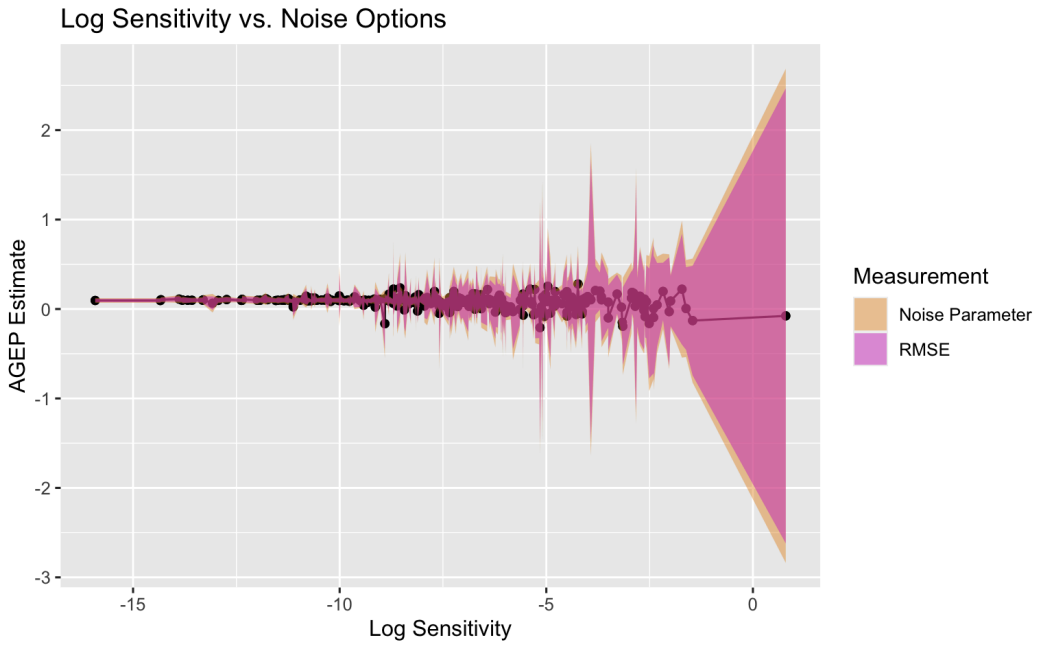

Figure 8: Sensitivity vs. Noise

The results for Approach 2 show similar promising results, depicted in Figure 9, where we observe a noisier and more biased estimate from the Bayesian Bootstrap results as the sensitivity of the statistic increases.

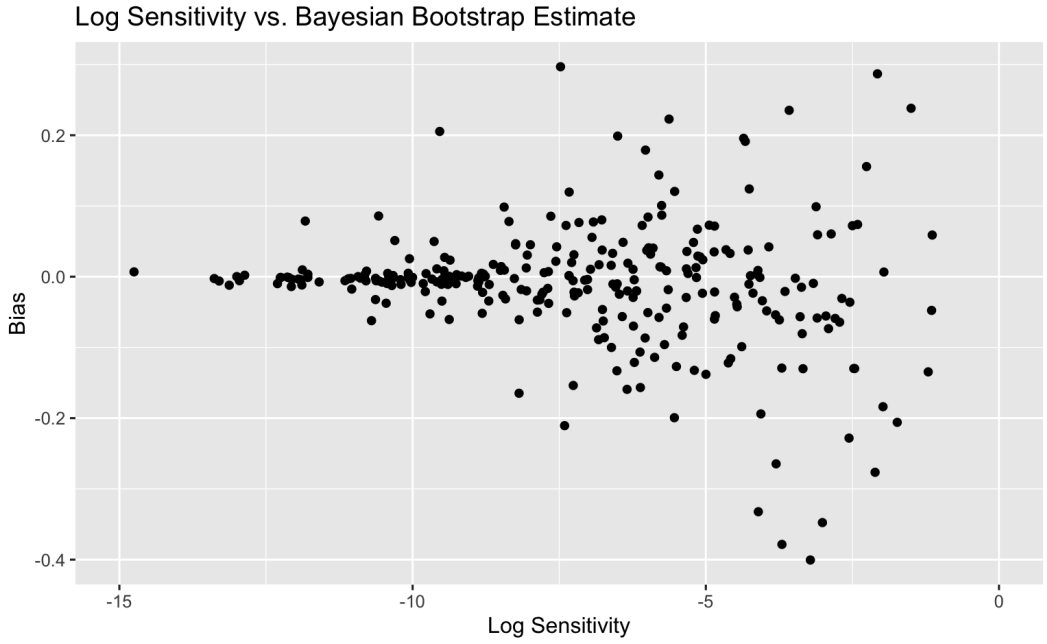

Figure 9: Log Sensitivity vs. Bayesian Bootstrap Estimates

5. Discussion

Overall, the two approaches for automating noisy estimates have shown a promising and desirable relationship with the sensitivity of a statistic. There's a natural question as to which approach is preferable. While in Approach 1, we observe that the noise added grows with how sensitive a statistic is, we saw that this approach is vulnerable to attacks. While we cannot rule out such attacks for Approach 2, it is less clear what a successful attack would look like. In either case, the automated disclosure avoidance described in Section 3 can help defend against such attacks. For example, it can help identify cells that need to be suppressed in model output tables. The aim of these different pilots is to fill in the gaps of one another. Ultimately, a suite of solutions is needed and no one solution can address everything at the moment.

For determining which of the two approaches for noise injection is better, further work needs to be completed to understand their privacy/utility trade off. Specifically, we will investigate the coverage probabilities of confidence intervals when using the noisy statistics and their Type 1 and 2 error rates. We will also investigate the usage of subsampling and aggregation style differentially private algorithms and whether they allow for valid and useful statistical inference. A strength of differential privacy solutions is that they limit the possibility of possible attacks; however, existing solutions currently suffer from limited utility. For the rule-based automated disclosure avoidance approach, we will investigate the usage of machine learning algorithms to infer privacy risk scores based off the metadata collected in the GUI and produced by the rule-based solution.

6. Conclusion

Census seeks to modernize disclosure avoidance as Census data products enjoy wider usage and the potential for reidentification attacks is increasing. To address this, Census is testing the usage of synthetic versions of its data products for researchers to work with. In turn, Census can act as an oracle and take the researcher models, run them on the original data and return disclosable and controlled results. This validation service is currently manual and labor intensive, but we've created a rule-based automation program that can scale to address the needs of the growing number of data users and shift the labor burden away from Census employees performing validation.

Nevertheless, this process will always require some human in the loop. Nuanced cases will require more careful consideration by a Census employee. We further investigate the usage of multiple replicates of synthetic data for creating noisy estimates of statistics. We find the noise introduced by our proposed methods increases as the sensitivity of the statistic increases and develop approaches for adding noise which are better behaved than using the root mean squared error of the synthetic estimates. Further work needs to be done in both avenues of automated validation and noise injection including merging the two into one pipeline. These tools are being developed to address the gaps in each other. Protecting privacy in Census data products will require a suite of solutions that address different concerns in the model creation, processing, and publishing steps. This is part of an ongoing effort to understand which tools are best for Census needs.

Acknowledgements

The authors would like to thank Megumi Ando for her guidance and expertise. The authors would also like to thank Kelly Christensen, Jeremy Lederman, David Lin, and Sayi Sathyavan for their support of the larger effort that influenced the direction of this research.

Any opinions and conclusions expressed herein are those of the authors and do not represent the views of the Census Bureau or other organizations. The Census Bureau has reviewed this data product to ensure appropriate access, use, and disclosure avoidance protection of the confidential source data used to produce this product. This research was performed under Census project P-7530064. No confidential data were used in this paper. Approved for Public Release; Distribution

References

Benedetto, G., Stanley, J., & Totty, E. (2018). The Creation and Use of the SIPP Synthetic Beta v7.0. *CES Technical Notes Series, 18-03*, Center for Economic Studies, U.S. Census Bureau.

Broderick, T., Giordano, R., Meager, R. (2023). An Automatic Finite-Sample Robustness Metric: When Can Dropping a Little Data Make a Big Difference?. V4.0. *arXiv*, 2011.14999.

Kinney, S. K., Reiter, J. P., and Miranda, J. (2014). SynLBD 2.0: Improving the Synthetic Longitudinal Business Database, *Statistical Journal of the International Association for Official Statistics, 30*, 129 - 135.

Rubin, D.B. (1993). Discussion: Statistical disclosure limitation. *Journal of Official Statistics, 9*, 462-468.