Journal of Privacy and Confidentiality
Vol. 9 (2) 2019

Submitted 20 Mar 2019
Published Oct 2019

RELEASING EARNINGS DISTRIBUTIONS USING DIFFERENTIAL PRIVACY: DISCLOSURE AVOIDANCE SYSTEM FOR POST-SECONDARY EMPLOYMENT OUTCOMES (PSEO)

ANDREW FOOTE, ASHWIN MACHANAVAJJHALA, AND KEVIN MCKINNEY

Center for Economic Studies, U.S. Census Bureau
e-mail address: andrew.foote@census.gov

Department of Computer Science, Duke University
e-mail address: ashwin@cs.duke.edu

Center for Economic Studies, U.S. Census Bureau
e-mail address: kevin.l.mckinney@census.gov

Abstract. The U.S. Census Bureau recently released data on earnings percentiles of graduates from post-secondary institutions. This paper describes and evaluates the disclosure avoidance system developed for these statistics. We propose a differentially private algorithm for releasing these data based on standard differentially private building blocks, by constructing a histogram of earnings and the application of the geometric noise mechanism to recover a differentially-private CDF of earnings. We demonstrate that our algorithm can release earnings distributions with low error, and our algorithm out-performs prior work based on the concept of smooth sensitivity from Nissim, Raskhodnikova and Smith (2007).

Received by the editors March 20, 2019.
Key words and phrases: differential privacy; education data.

JPrivacy Confidentiality
www.journalprivacyconfidentiality.org
DOI:10.29012/jpc.722

1. Introduction

The Post-Secondary Employment Outcomes (PSEO) data is produced by the U.S. Census Bureau, and is a data product that publishes earnings and employment outcomes of graduates from post-secondary institutions.[1] Originally released in March 2018, the first release of PSEO published earnings percentiles by institution, degree level, degree field, graduation cohort, and year post-graduation[2] by linking transcript data from colleges and universities to the Longitudinal Employer-Household Dynamics (LEHD) data (Abowd et al., 2009). This data product is the first to publish earnings using national earnings data, and is used to inform administrators, policy-makers, parents and students about differences in earnings outcomes by institution and field of study.

While this data significantly advances our knowledge of post-secondary outcomes, the risk with such data is that an individual's outcomes could be identified. The U.S. Census Bureau is bound by Title 13 Section 5,[3] which does not allow disclosure of individual characteristics or jobs. Violation of this requirement carries significant penalties for employees and contractors, and can result in fines up to $250,000 and imprisonment up to five years.

One aspect of the data that increases the risk of disclosure is that states also release data on earnings using the same microdata for the Unemployment Insurance (UI) and transcript linkage, but are constrained to UI earnings in the same state.[4] This feature of outside parties having the data frame (graduates from an institution) and a partial record of earnings (all in-state earnings) increases the likelihood of disclosing an individual's earnings.[5] For these reasons, differential privacy (DP) (Dwork, 2006) is an ideal choice for a privacy definition, as it gives us the strongest protection of an individual's earnings.

Our paper makes contributions to the application of differential privacy algorithms. First, we describe our DP algorithm, which allows the release of an arbitrary number of percentiles of the distribution. Our algorithm first estimates differentially private cumulative distribution function (CDFs) for subsets of the student population, by (i) estimating the histogram over earnings using the Geometric noise mechanism, (ii) inferring a CDF function from these noisy counts, and (iii) reading off percentiles from the above constructed CDF. The algorithm uses geometric noise on a histogram, and takes advantage of composition properties of differential privacy. Nissim, Raskhodnikova and Smith (2007)'s smooth sensitivity algorithm is another algorithm for protecting percentiles, and we show that our algorithm is more accurate for most privacy loss values.[6] Additionally, while our algorithm outputs percentiles that satisfy common sense constraints (e.g. 50th percentile less than 75th percentile), repeated invocation of the smooth sensitivity based approach may not satisfy these constraints. Our histogram approach also solves this problem, and allows for the release of additional, higher-level cells. Given that the focus of our paper is to compare the statistical properties of different methods of protecting data, it is in the same vein as Wasserman and Zhou (2010), while focusing on a specific application.

[1] The data from PSEO are available here: https://lehd.ces.census.gov/data/pseo_beta.html

[2] At the time of this writing, there are a total of 28593 possible crossings of these data elements.

[3] https://www.law.cornell.edu/uscode/text/13/chapter-5

[4] For example, Colorado and Texas both publish earnings outcomes using in-state earnings data.

[5] Suppose all individuals except one stay in a state for employment. If we release the national earnings and the state releases similar in-state numbers, the earnings of the missing individual would be disclosed.

[6] Another differentially-private algorithm for protecting percentiles is the exponential mechanism, which is Algorithm 2 in Smith (2011).

The remainder of the paper proceeds as follows. Section 2 provides definitions of differential privacy that we refer to for rest of the paper, while Section 3 describes the datasets we use and the need for differential privacy. Section 4 describes the algorithm we use to protect the PSEO data, and provides a proof that our algorithm satisfies ϵ-differential privacy. Section 5 then evaluates the algorithm compared to other algorithms, and Section 6 concludes.

2. Preliminaries

This section provides definitions of differential privacy and dataset definitions for the remainder of the paper. For more complete treatments of differential privacy, consult Dwork (2006), Dwork et al. (2006) and Dwork and Roth (2014).

Database Definition Let D be a database of records with k variables $(A_1, ..., A_k)$. The domain of each variable A_i is denoted $dom(A_i)$. D has n observations.

Our focus for the remainder is count queries over tables, where a count query is defined below:

Definition 2.1 Marginal Query**.** *The count query $q_\phi(D)$ is the number of observations from D that satisfy ϕ, which is an arbitrary boolean predicate on the attributes $(A_1, ..., A_k)$. In plain terms, it is the count of observations that have certain values of the variables. For example, the number of students graduating with an English degree in 2008, or the number of students earning less than $25,000.*

Differential Privacy An algorithm is differentially private if its output is not significantly affected by the presence or absence of a single record from D. Consider two databases, D and D', which differ by the presence of a single record. These databases are called *neighbors*.

Definition 2.2 Differential Privacy**.** *Let M be an algorithm to output data, and tables D and D' be neighboring databases (i.e. $|(D \setminus D') \cup (D' \setminus D)| = 1$). Then M satisfies ϵ-differential privacy if for all D and D' and for all $S \subset range(M)$,*

$$\frac{Pr[M(D) \in S]}{Pr[M(D') \in S]} \leq e^\epsilon$$

To satisfy differential privacy, for a given query $q_v(D)$, we have to add noise to that result which is related to the sensitivity of the query.

Definition 2.3 Sensitivity**.** *Let L denote the set of all possible tables, and q be a query function on tables. The sensitivity of the query is denoted Δ_q and is defined as:*

$$\Delta_q = max_{D,D' neighbors \in L} ||q(D) - q(D')||_1$$

For count queries, the sensitivity of the query is 1.[7]

Definition 2.4 Local Sensitivity**.** *Let D be a table, and D' be a table that differs by one element, and q be a query function on tables. The local sensitivity of the query is denoted LS_q and is defined as:*

$$LS_q = ||q(D) - q(D')||_1$$

The following theorems are adapted from Dwork and Roth (2014).

[7]For completeness, a proof is in the appendix.

Theorem 2.1 (Sequential Composition)**.** *Let M and B be ϵ_1- and ϵ_2-differentially private algorithms. Releasing the outputs of $M(D)$ and $B(D)$ on the same database D results in $(\epsilon_1 + \epsilon_2)$-differential privacy.*

Theorem 2.2 (Parallel Composition)**.** *Let a database D be partitioned into k disjoint subsets, D_i, and k queries $B_i(D_i)$, each of which are ϵ-differentially private. Then the results of these queries, $B(D)$, is also ϵ-differentially private.*

2.1. **Algorithms.**

Definition 2.5 Geometric Mechanism**.** *Let $q(D)$ be a query on a database D. Let $\eta \sim Geo(X, p) - Geo(Y, p)$ denote a random variable draw from the distribution generated from the difference of two random variables (X,Y) which are distributed according to the geometric distribution, where $p = 1 - e^{-\epsilon}$. The algorithm which returns $\tilde{q}(D) = q(D) + \eta^d$ satisfies ϵ-differential privacy, where η^d is a vector of d independently drawn Geometric random variables.*

This definition draws on the definition from Ghosh, Roughgarden and Sundararajan (2012).

3. Datasets and Issues

The input database for the PSEO, D, has attributes A which we separate as follows: stratifying attributes A_c, which define the cells over which we calculate earnings characteristics; and earnings, A_e.

We denote $\times_{i \in c} dom(A_i)$ the cross product of all domains, which represents the space of all possible records in D. Each combination $i \in \times_{i \in c} dom(A_i)$ will be referred to as a *cell*. We perform queries on each cell separately, taking advantage of the parallel composition theorem from Section 2. For the purpose of this paper, we consider only a static set of queries on the data, and leave the question of dynamic queries to future work.

To produce the PSEO, we combine two datasets. First, the earnings information comes from the LEHD, which has quarterly earnings records from 50 states and the District of Columbia. We supplement these earnings data with earnings records from the Office of Personnel Management; these data cover a large share of the federal workforce, but exclude certain occupations and departments (such as Department of Defense).

Graduate records are from education partners, and as of September 2019, we include data from the University of Texas System; the Colorado Department of Higher Education; University of Michigan-Ann Arbor. These data include institution, field of study, degree date, degree level, and background characteristics. We match these data to produce cell-level estimates of earnings, where the cell is defined by a combination of degree level, degree field, institution, graduation cohort, and year after graduation.[8]

We total earnings from all jobs for an individual, and restrict our sample to individuals earning more than the equivalent of full-time work at the prevailing federal minimum wage.

[8]Graduation cohorts are three or five year groups of graduates, depending on the degree level.

3.1. **Utility.** Researchers and analysts working with these data are interested in the earnings outcomes of graduates by the stratifying attributes described above. A number of states have produced similar data, but have been unable to measure earnings outcomes for graduates that move out of state.[9] Additionally, College Scorecard produced similar earnings outcomes by institution for enrolees, not graduates.[10]

There are four different outcomes that we release for every cell. First, we measure the 25th, 50th and 75th percentiles of earnings. Second, we release the cell count. For the purpose of this paper, we are focusing on protecting the earnings percentiles.

We outline two use cases for these data.

Students and Parents Students and parents want to be informed about potential outcomes of graduating with a degree in a certain field, or from a specific institution. In this case, the users of the data care about how closely the released reported earnings values correspond with the true earnings values. Additionally, they may care about errors less when true value is larger, implying a similar sized error has less utility cost when the true earnings are $100,000 than if the true earnings are $40,000.[11]

State Boards of Education As of 2018, ten states included some measure of labor market outcomes for students into their performance-based funding formulas for public post-secondary institutions (Li, 2018).[12] Many of these formulas focus on job placement and entry-level earnings, but in a non-linear way. For example, the Florida College System uses entry-level wages and compares them with entry-level wages in the colleges area. In the Florida College System, colleges receive credit for earnings of graduates up to 100% of earnings up to entry-level wages, but no additional credit for having graduate earnings above that cutoff.[13] This formula implies that errors are more costly for lower true earnings values, compared with higher earnings values (where there may be no additional benefit to the institution).

Accuracy Measure In both of the above use cases, accuracy is more valuable at lower earnings levels than higher earnings levels. For that reason, to compare our differentially-private algorithms, we use a relative accuracy measure (rather than an additive measure of error), which we describe in Section 5.[14]

3.2. **Privacy Requirements.** There are a number of privacy requirements for the PSEO data, which are covered by Title 13 of the U.S. Code. Under Title 13, the Census Bureau cannot "make any publication whereby the data furnished by any particular establishment or individual under this title can be identified." This statute has two implications for our work.

[9]Colorado's earnings data on graduates is here: https://highered.colorado.gov/Data/Workforce/EdPays.html. Florida's earnings data is here: https://www.floridacollegesystem.com/resources/data/fcs_graduate_outcomes_dashboard.aspx

[10]Documentation on College Scorecard is available at https://collegescorecard.ed.gov/data/documentation/

[11]Individuals may care less about an equally sized error if the true value is lower because of diminishing marginal utility, a concept in economics that assumes that the utility function in concave. (Varian, 2010)

[12]https://www.thirdway.org/report/lessons-learned-a-case-study-of-performance-funding-in-higher-education

[13]For more details on the actual formula, see https://www.floridacollegesystem.com/sites/www/Uploads/Publications/Funding%20Formula/Wages_1718Model.pdf

[14]One interesting alternative error measure is one that penalizes positive errors more than negative errors, due to loss aversion.

First, we cannot disclose the earnings of an individual. Additionally, we cannot disclose the existence of a job (a linkage between an employee and an employer) held by an individual.

Similar privacy requirements are also affirmed in the recent re-introduction of the College Transparency Act, which explicity states in the legislation, "In carrying out the public reporting and disclosure requirements of this Act, the Commissioner shall use appropriate statistical disclosure limitation techniques necessary to ensure that the data released to the public cannot include personally identifiable information or be used to identify specific individuals." These privacy requirements present the constraints under which we can release data from PSEO.

The new information the PSEO includes over previous datasets is national earnings data, which may change the dataset D in one of two ways. First, it may change the value of A_e for an individual. Second, it may include an individual row in the dataset that was not previously there (if an individual had no in-state earnings). Given the privacy requirements and the new information provided by PSEO, the object that we need to keep private is the addition or removal of a single row from the dataset D, creating D', which includes employment and earnings information about an individual, and makes differential privacy a very appropriate privacy protection method in this setting.

4. Methodology

In this section, we discuss the differentially private algorithms we evaluate in the next section. In Section 4.1 we describe the histogram approach, while in Section 4.2 we describe the smooth sensitivity approach from Nissim, Raskhodnikova and Smith (2007). For each of these algorithms, we describe the input and output data, and then describe the algorithm in detail. The next section compares the relative accuracy for each algorithm.[15]

4.1. **Histogram Algorithm.**

The histogram algorithm for extracting percentiles can be seen as an extension of Dwork et al. (2006), who describes the perturbation of a histogram; we put structure on the histogram, and extract additional moments from the protected counts.

Inputs. For each cell, the input of the algorithm is a list of earnings values, $e_1, e_2, ..., e_N$, which are earnings for all individuals in a given cell.

Outputs. There are two outputs of the algorithm. The first is a list of protected counts for each histogram bin within a cell, $(\tilde{q}_1^c, \tilde{q}_2^c, ..., \tilde{q}_M^c)$ (there are M bins in the histogram). Using these counts, the second output obtained is a list of percentiles, which are read from the empirical cumulative density function (CDF).

Constructing the Histogram. To construct the histogram, consider a set of bin definitions, such that earnings value e_i is in bin j if $b_j \leq e_i < b_{j+1}$, where the values b_j are public information and the same across all cells in the dataset.

Using these bin definitions, consider a function q_j^C, which returns the count of earnings values that fall in bin j. The list of values $q_1^c, q_2^c, ..., q_M^c$ summarize the histogram.

[15]The code which describes these algorithms is available at Foote, Machanavajjhala and McKinney (2019).

Protecting Bin Counts. From the definition of the histogram above, we protect the queries q_j^c, which returns the count of the observations in a given bin j, with a privacy loss of ϵ. Additionally, these queries imply the corresponding empirical CDF:

$$F(j) = \frac{\sum_{i=1}^{j} q_i^c}{\sum_{i=1}^{M} q_i^c} \tag{4.1}$$

The sensitivity of each of these queries is 1, and therefore we can protect each of these queries with privacy loss ϵ by adding geometric noise as described above in Section 2. Therefore, our protected counts are:

$$\tilde{q}_j^c = q_j^c + \zeta$$

Where ζ is distributed according to the geometric noise distribution.

Calculating Protected Percentiles. We use these fuzzed values to create a fuzzed CDF,

$$\tilde{F}(j) = \frac{\sum_{i=1}^{j} \tilde{q}_i^c}{\sum_{i=1}^{M} \tilde{q}_i^c}$$

If we assume that earnings are distributed uniformly within a bin, we can use $\tilde{F}(j)$ to extract protected percentiles. Note that $\tilde{F}(j)$ will not necessarily be non-decreasing, because there may be cases when $\tilde{q}_j < 0$.[16]

To calculate a percentile Y, we find the first bin J such that

$$\frac{\sum_{i=1}^{J-1} \tilde{q}_i^c}{\sum_{i=1}^{M} \tilde{q}_i^c} < Y/100 \leq \frac{\sum_{i=1}^{J} \tilde{q}_i^c}{\sum_{i=1}^{M} \tilde{q}_i^c} \tag{4.2}$$

Then, the Yth percentile is $b_J + (b_{J+1} - b_J) \times \frac{(Y/100 \times \sum^{J} \tilde{q}_i^c) - \sum^{J-1} \tilde{q}_i^c}{\tilde{q}_J^c}$.[17]

We use this technique to calculate the 25th, 50th, and 75th percentile values.

4.1.1. *Choosing Bin Definitions.* The key question with the above technique is how to define the bins. (that is, the b_is from above) There are two interrelated decisions; first, how many bins to have (that is, what is M); second, what the width of the bins are.

In the next section, we evaluate the accuracy rates of different choices. We compare two main ways to decide what the bin widths are. First, the log normal approach, which uses percentiles of a log normal distribution as the bin widths. This approach has the advantage of making it equally likely that an observation is in any of the bins, since earnings are typically distributed log-normally.[18]

[16]While conceptually we are able to post-process the data to guarantee $\tilde{q}_j \geq 0$, and in some instances accuracy will improve, we find these potential gains in accuracy of the outputs are minimal.

[17]In words, if a bin J includes the Yth percentile, and the Yth percentile is W of the way through the interval defined by bin J, then the Yth percentile is the lower-bound value of bin J, b_J, plus $W \times width$. Choosing the first such bin J that satisfies the constraint ensures that the percentiles are properly ordered.

[18]In our application for the PSEO, we define the bins as follows. The bottom cutoff is \$10,000, which is very close to the minimum value in the data by construction. For the next 19 b_is, we choose every 5th percentile of the log normal distribution with mean 11.003 and standard deviation 0.753. The mean and standard deviation were calculated using the 5-year ACS Public-Use Microsample. We calculated the mean and standard deviation of wage and salary income for employed individuals with a BA or above. Additionally, for b_M, we use the 97.5th percentile value of the distribution, which is about \$260,000. Finally, for any

The second approach, which we call the "even bins" approach, evenly spaces the bins between the minimum and maximum values. This approach has the advantage of being very transparent and easy to use, particularly if there is not an obvious parametric approximation for the distribution of the outcome being protected.

For both approaches, the bottom cutoff is \$10,000. The sample frame for the PSEO is full-time equivalent at the federal minimum wage, which is close to \$10,000.

Proposition 4.1. *The histogram algorithm satisfies ϵ-differential privacy*
***Proof**:*

The algorithm can be broken down into three steps.

1. Choosing bin definitions: This is done without looking at the private data, and hence it does not incur any privacy loss.

2. Measuring bin counts noisily: Each bin count, $\tilde{q}_i^c$, is released under ϵ-differential privacy, using the geometric mechanism. Since the bins are disjoint sets, releasing all the bin counts satisfies ϵ-differential privacy.

3. Computing percentile values: From the composition property of differential privacy, the following function is also ϵ-differentially private:

$$\sum_{i=1}^{J} \tilde{q}_i^c$$

Since Yth percentile $= b_J + (b_{J+1} - b_J) \times \frac{(Y/100 \times \sum^J \tilde{q}_i^c) - \sum^{J-1} \tilde{q}_i^c}{\tilde{q}_J^c}$ is a function of ϵ-differentially private values, the Yth percentile is also ϵ-differentially private, as is the histogram algorithm. □

In addition to the above proposition, the clear corollary is that the histogram list of counts $(\tilde{q}_1, \tilde{q}_2, ..., \tilde{q}_M)$ is ϵ-differentially private (Proposition 1 in Hay et al. (2010)); this result allows the list of values $(\tilde{q}_1, \tilde{q}_2, ..., \tilde{q}_M)$ to also be considered releasable.

4.2. **Competing Algorithms .** This subsection describes the smooth sensitivity algorithm for protecting percentiles, originally from Nissim, Raskhodnikova and Smith (2007).

Inputs. For each cell (defined as in Section 2), the input of the algorithm is a list of sorted earnings values, $A_{ie} = (e_1, e_2, ..., e_N)$, which are earnings for all individuals in a given cell i.

Outputs. The outputs of the algorithm are the 25th, 50th and 75th percentile values, which we refer to as $\tilde{P}_{25}, \tilde{P}_{50}$ and $\tilde{p}_{75}$.

---

earnings greater than that value, we count it in the final bin, M. In the case where a percentile is in the largest bin, we define b_{M+1} to be the 99.9th percentile of earnings from the log normal distribution, which is 614597. Together, we have 21 bins. For reference, these histogram values are in the appendix. Additionally, in this particular application, a log-normal histogram made the most sense; however, the method is more general. The goal of the histogram should be such that a randomly chosen observation has an equally likely probability of landing in any bin, thereby decreasing the number of bins with no observations in them. The specific distribution chosen depends on the expected distribution of the underlying data.

Calculating Percentiles. Let $p_x(D)$ denote the query that returns the Xth percentile of an input dataset D.

Set ϵ_x such that $\sum_{x \in X} \epsilon_x = \epsilon$.[19]

For each X in 25,50,75, compute the true percentile $y = p_x(D)$. Then return the protected percentile, $\tilde{y} = y + S_{px}(D) \cdot Gamma(1/\epsilon_x)$, where the $Gamma(.)$ distribution is defined as $1/|1+x|^4$.

We next describe how to compute the smooth sensitivity, $S_{px}(D)$ for the percentile query.

Definition 4.1 Smooth Sensitivity for Percentiles**.** *Define the smooth upper bound, $S_{q_X}(d)$ to $LS_{q_{med}}(d)$, such that adding noise proportional to S_{q_X} satisfies differential privacy requirement.*[20]

These smooth upper bounds must satisfy the following requirements:

$$\begin{aligned} \forall d, S_q(d) &\geq LS_q(d) \\ \forall d, d' \textit{ differing by one entry} S_q(d) &\leq exp(\beta) S_q(d') \end{aligned} \tag{4.3}$$

From these above, the β-smooth sensitivity is:

$$S^{\star}_{q,\beta}(d) = max_{d'}(LS_q(d') exp(-m\beta))$$

Where d, d′ differ by m entries.

For completeness, we show how to derive the smooth sensitivity of the median function q_{50}. The same algorithm can be used for any percentile. Applying this framework to the query of the median:

$$LS_{q_{med}}(d) = max(e_M - e_{M-1}, e_{M+1} - e_M) \textit{ for } M = \frac{N+1}{2} \tag{4.4}$$

This implies that

$$S^{*}_{q_{med},\beta}(d) = max_{k=0,\ldots,n} \left(exp(-k\beta) max_{t=0,\ldots,k+1}(e_{M+t} - e_{M+t-k-1})\right) \tag{4.5}$$

More generally, for any percentile X, $LS_{q_X}(d) = max(e_M - e_{M-1}, e_{M+1} - e_M) for M = \frac{(N+1)X}{100}$, which implies the β-smooth sensitivity for percentile X is:

$$S^{*}_{q_X,\beta}(d) = max_{k=0,\ldots,n} \left(exp(-k\beta) max_{t=0,\ldots,k+1}(e_{M+t} - e_{M+t-k-1})\right) \tag{4.6}$$

The smooth sensitivity value for a percentile X from earnings list E is defined as $S^{*}_{x,\beta}(E)$.

If the true percentile of the earnings list is $P_x(E)$, then it is protected in the following way:

$$\tilde{P}_x(E) = P_x(E) + \eta \frac{S^{*}_{x,\beta}(E)}{\epsilon/16}$$

[19]When using the smooth sensitivity algorithm with a privacy budget of ϵ, the researcher can allocate privacy loss differently depending on the goal of the output. If the researcher desires accuracy to be similar across the three queries, then he could allocate more privacy loss to more sensitive queries. If he instead desires to allocate privacy loss equally, then $\epsilon_x = \epsilon/3$.

[20]Formally, if for some query $q(.)$, and neighboring datasets d and d', $log(\frac{Pr(q(d)=X)}{Pr(q(d')=X)} < \epsilon$, then $q(.)$ is ϵ-differentially private.

Where η is drawn from the distribution $h(y) = \frac{1}{1+|y|^4}$ and $\beta = \epsilon_x/4$. According to Lemma 2.5 of Nissim, Raskhodnikova and Smith (2007), the output $\tilde{P}_x(E)$ is ϵ_x-differentially private.

5. Evaluation

This section describes the experiments we ran on the simulated earnings data, and proceeds as follows. First, we describe the accuracy measures we use. Second, we describe the algorithms we compared and the range of parameter settings we tested. We then present the results of our experiments.

5.1. **Data For Simulations.** The data we use for the experiments in this paper are constructed using the protected histograms from the PSEO as inputs.[21] That is, we take the differentially private histograms, and generate individual observations based on the counts in each bin. For example, if a bin has a count of 5, we generate 5 observations by randomly drawing earnings from a uniform distribution between the two bin edges. This approach allows us to run simulations on a dataset that has similar statistical properties to the underlying data.

5.2. **Error Measures.** We use relative accuracy to assess the quality of the protected data, where relative accuracy is defined as below:

$$RelAccuracy = 1 - \frac{|Protected - True|}{True} \tag{5.1}$$

Where $True$ is the true value (a percentile of the distribution), and $Protected$ is the value after applying the differentially-private algorithm. The numerator of the second term is the absolute value of the L1 error, which we scale by the true value, so that the accuracy values are expressed as a percent deviation from the true value. For interpretation, an algorithm that always has an output of 0 will have a relative error of 1 and a relative accuracy of 0.

This notion of relative accuracy fits well with the utility of the data. As students and policy-makers evaluate the data, they care about how close the reported values are to the truth. Our measure of accuracy also scales the absolute difference, since errors of $10,000 have different implications for outcomes for lower earnings majors than higher earnings majors (that is, similar magnitude errors are more costly from a utility perspective if the true earnings are smaller)

Finally, note that in our setting, the range of earnings values is strictly above $10,000, which means that minimum value of $True$ for any percentile is never 0, so $RelAccuracy$ is always defined.

[21] We use the March 2018 vintage of the PSEO

5.3. **Algorithms Compared.** We compare two different histogram algorithms, which provide the same guarantee of privacy. First, the PSEO approach, which is a histogram where the bins are percentiles of a log-normal distribution (referred to as "Log-Normal" hereafter). Second, a histogram where the bins are uniformly distributed across the range of earnings values (referred to as "Even" hereafter). To compare these approaches to an existing method of releasing percentiles, we also evaluate the accuracy of the smooth sensitivity algorithm in our setting.

5.4. **Parameter Values.** We run our experiments on a number of different combinations of parameter values.

- Bins: 10 - 30
- Epsilon: 0.1 - 3, in steps of 0.1
- Percentiles: 25th, 50th and 75th

For each of the combinations above, we draw noise 20 times and calculate the average relative accuracy.

5.5. **Results of Experiments.** We summarize the results of these simulations in Figures 1-3. In the following figures, we only show the accuracy results for the 50th percentile of earnings; Figures 4 and 5 display results for the 25th and 75th percentiles, respectively. Figure 1 shows the relative accuracy for a number of bin counts (10,15,20,25), comparing the three candidate algorithms.[22] What we find is that the log-normal histogram scheme is strictly better than the evenly-spaced scheme, with significant reductions in the error. Strikingly, for most values of ϵ, the log-normal approach is also more accurate than the smooth sensitivity approach.[23]

Figure 2 fixes the bin count constant at 20, and shows the accuracy measure by cell size categories. The results show that for every cell bin, the log-normal approach is more accurate. Our results show that in each cell-size bin, for the smooth sensitivity algorithm to have similar accuracy, the privacy loss parameter has to be much higher. This is true even for large cells, which are much less sensitive.

There are two main reasons driving the lower accuracy for the smooth sensitivity approach. First, for smaller cells, the local sensitivity (LS) can be quite large. Second, the distribution from which draws are made has very fat tails, and combined with the large values of LS, causes the protected percentiles to differ considerably from the truth.

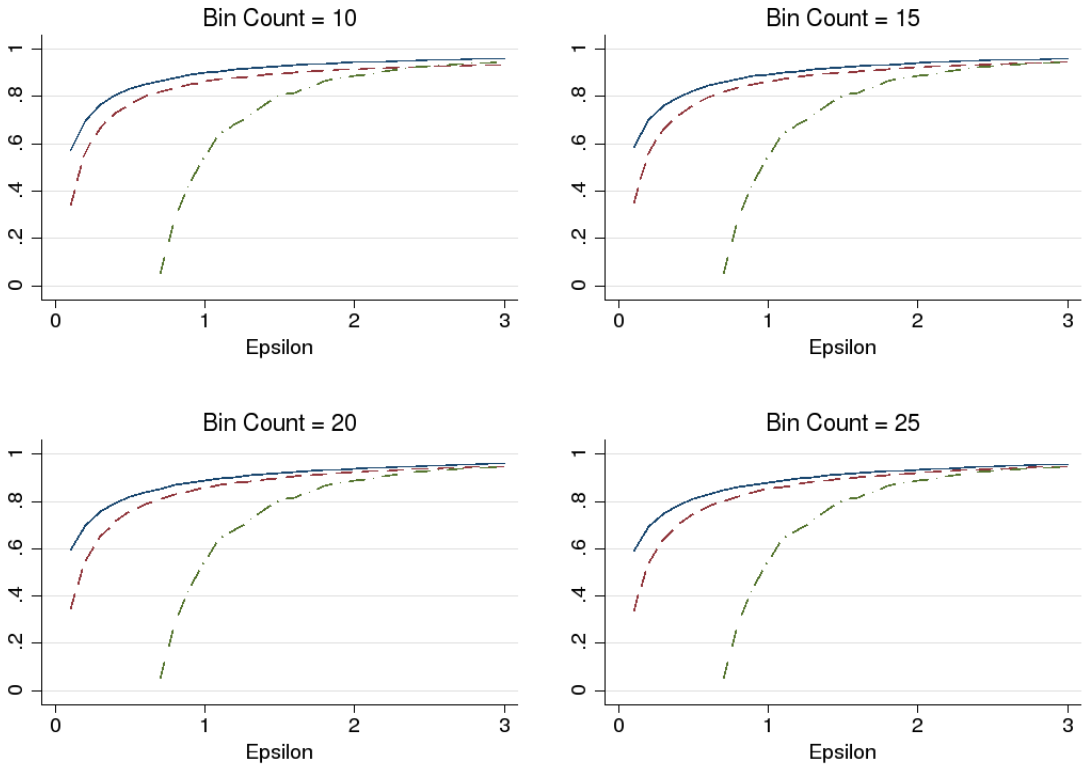
Given that the log-normal approach is clearly more accurate, the next decision to make in protecting the data is how many bins the histogram should have; that is, how is the relative accuracy related to the number of bins used. Theoretically, the effect is ambiguous. Having more bins means that the bins are less wide, which should increase the accuracy within a bin. However, having more bins also increases the total noise infused into the data, as well as the share of each bin's count that is noise, since geometric noise is drawn for each bin.

To test the relationship empirically, Figure 3 graphs the relationship between bins and accuracy. It appears that accuracy is decreasing in bin size, and that the noise effect

[22]Since the smooth sensitivity approach does not depend on the bin count, these are invariant.

[23]Figures showing the accuracy for the 25th and 75th percentiles are in the appendix. We also include an additional error measure, which sums up the L1 errors across the percentiles; this measure also shows that the log-norm approach is more accurate.

FIGURE 1. Relative Accuracy by Histogram Method, 50th Percentile



Notes: Log-Normal is Solid Blue; Even is Dashed Red; Smooth-sensitivity is Dash-dot Green

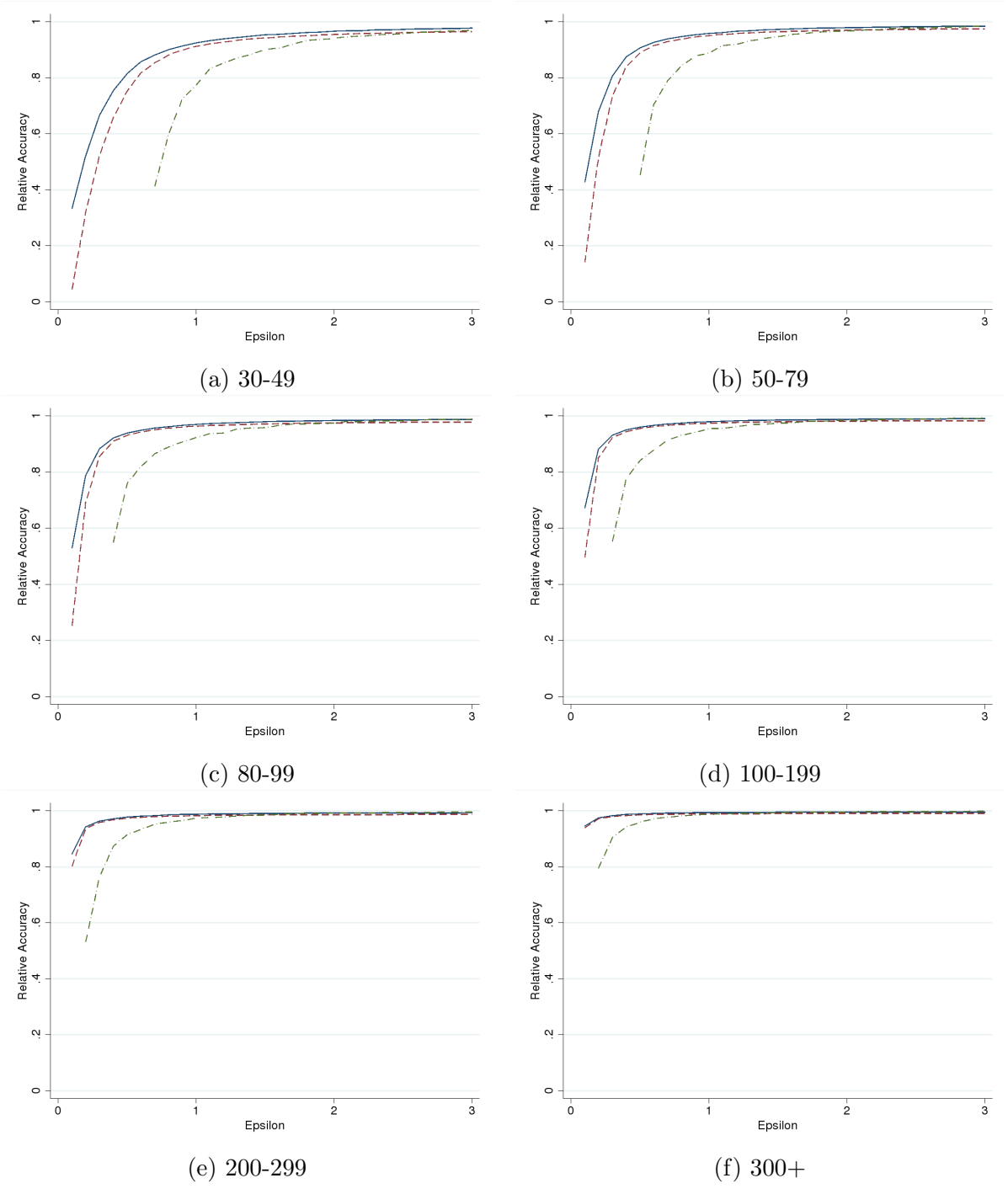
dominates the width effect. However, the effects are relatively small, such that the accuracy only decreases by about 2 percentage points from 10 to 30 bins.[24]

One thing to note here is that in our comparisons above, for the smooth sensitivity each percentile is calculated as a separate query, which implies that the required privacy loss is much higher for the smooth sensitivity approach.

Formally, note that $RelAccuracy$ is a function of ϵ. Now consider two different sets of privacy loss parameters, $(\epsilon_H(y))$ and $(\epsilon_{SS}(y))$, such that for a given yth percentile, $RelAccuracy(\epsilon_w(y)) > a$, where a is an accuracy level (e.g., $a = 0.9$ is 90% accuracy). To guarantee that all percentiles have at least an accuracy above a in the histogram approach, a practicioner must use a total privacy loss budget $\epsilon^*_H = max_{y \in Y}[\epsilon_H(y)]$ if using the histogram approach, or $\epsilon^*_{SS} = \sum_{y \in Y} \epsilon_{SS}(y)$ if using the smooth sensitivity approach (because each percentile is a distinct query). Our results show that for most values of ϵ, the histogram approach is more accurate, and there is no additional privacy loss from calculating additional percentiles.

[24]For reference, the PSEO uses 21 bins in its protection system. These bins are shown in the appendix.

FIGURE 2. Relative Accuracy by Cell Size



(a) 30-49

(b) 50-79

(c) 80-99

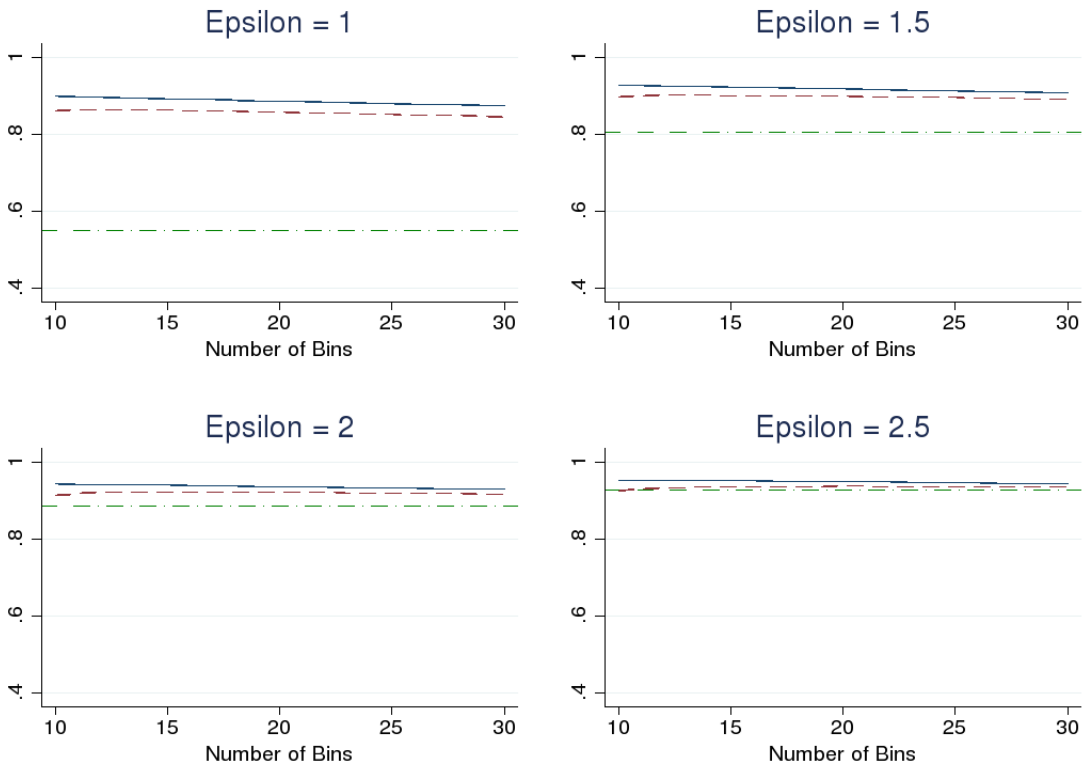
(d) 100-199

(e) 200-299

(f) 300+

Notes: Log-Normal is Solid Blue; Even is Dashed Red; Smooth-sensitivity is Dash-dot Green

FIGURE 3. Relative Accuracy by Bin Count



Notes: Log-Normal is Solid Blue; Even is Dashed Red; Smooth-sensitivity is Dash-dot Green

6. Conclusion

In the world of readily available microdata for analysis, statistical agencies need to take confidentiality seriously. Increasingly, outside parties have access to a large share of the microdata used in the production of statistics, which makes protecting the data with conventional methods much more difficult.

In 2018, the U.S. Census Bureau released the Post-Secondary Employment Outcomes, which uses differential privacy to protect the underlying microdata while releasing detailed information on the distribution of earnings for graduates.

In this paper, we describe the method we use to protect the data, and compare our method to other potential methods of protecting the data in a differentially private way. We find that it yields significant improvements over previous methods for protecting percentiles.

Our method for releasing detailed distributional characteristics of earnings is easily generalized to other settings, and we believe that it can be used for other settings where distributions characteristics are of interest. For example, there is a lot of interest in releasing statistics on earnings and wealth inequality at the national and local levels; our paper proposes one approach to releasing these statistics using differential privacy.

References

Abowd, John M., Bryce E. Stephens, Lars Vilhuber, Fredrik Andersson, Kevin L. McKinney, Marc Roemer, and Simon Woodcock. 2009. "The LEHD Infrastructure Files and the Creation of the Quarterly Workforce Indicators." In *Producer Dynamics: New Evidence from Micro Data. NBER Chapters*, 149–230. National Bureau of Economic Research, Inc. https://ideas.repec.org/h/nbr/nberch/0485.html.

Dwork, Cynthia. 2006. "Differential Privacy." *ICALP'06*, 1–12. Berlin, Heidelberg:Springer-Verlag. https://doi.org/10.1007/11787006_1 .

Dwork, Cynthia, and Aaron Roth. 2014. "The Algorithmic Foundations of Differential Privacy." *Found. Trends Theor. Comput. Sci.*, 9(3–4): 211–407. https://doi.org/10.1561/0400000042 .

Dwork, Cynthia, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. "Calibrating Noise to Sensitivity in Private Data Analysis." *TCC'06*, 265–284. Berlin, Heidelberg:Springer-Verlag. https://doi.org/10.1007/11681878_14 .

Foote, Andrew, Ashwin Machanavajjhala, and Kevin McKinney. 2019. "Code for Releasing Earnings under Differential Privacy (Version v2.0.0)." https://doi.org/10.5281/zenodo.3516706 .

Ghosh, A., T. Roughgarden, and M. Sundararajan. 2012. "Universally Utility-maximizing Privacy Mechanisms." *SIAM Journal on Computing*, 41(6): 1673–1693. https://doi.org/10.1137/09076828X .

Hay, Michael, Vibhor Rastogi, Gerome Miklau, and Dan Suciu. 2010. "Boosting the Accuracy of Differentially Private Histograms Through Consistency." *Proc. VLDB Endow.*, 3(1-2): 1021–1032. https://doi.org/10.14778/1920841.1920970 .

Li, Amy Y. 2018. "Lessons Learned: A Case Study of Performance Funding in Higher Education." Third Way.

Nissim, Kobbi, Sofya Raskhodnikova, and Adam Smith. 2007. "Smooth Sensitivity and Sampling in Private Data Analysis." *STOC '07*, 75–84. New York, NY, USA:ACM. https://doi.org/10.1145/1250790.1250803 .

Smith, Adam. 2011. "Privacy-preserving statistical estimation with optimal convergence rates." 813–822, ACM.

Varian, Hal R. 2010. *Intermediate Microeconomics: A Modern Approach.* . eighth ed., New York:W.W. Norton & Co.

Wasserman, Larry, and Shuheng Zhou. 2010. "A Statistical Framework for Differential Privacy." *Journal of the American Statistical Association*, 105(489): 375–389. https://doi.org/10.1198/jasa.2009.tm08651 .

Appendix A. Proof of Sensitivity of Count

Consider a dataset D, and a neighboring dataset D' which differs by one observation. Furthermore, consider a count query $q_c(.)$ on a dataset, which returns the number of observations with certain attributes, which we will refer to as X. Now consider the cases below:

$$|q_c(D) - q_c(D')| = \begin{cases} 1, \text{if the differing observation has the attributes } X \\ 0, otherwise. \end{cases} \tag{A.1}$$

In the case of the count query, $S(q_c) = 1$. Therefore, for any count query $q_c(d)$, if we draw $\zeta \sim Lap(1/\epsilon)$, then $\tilde{q}_c(d) = q_c(d) + \zeta$ is ϵ-differentially private.

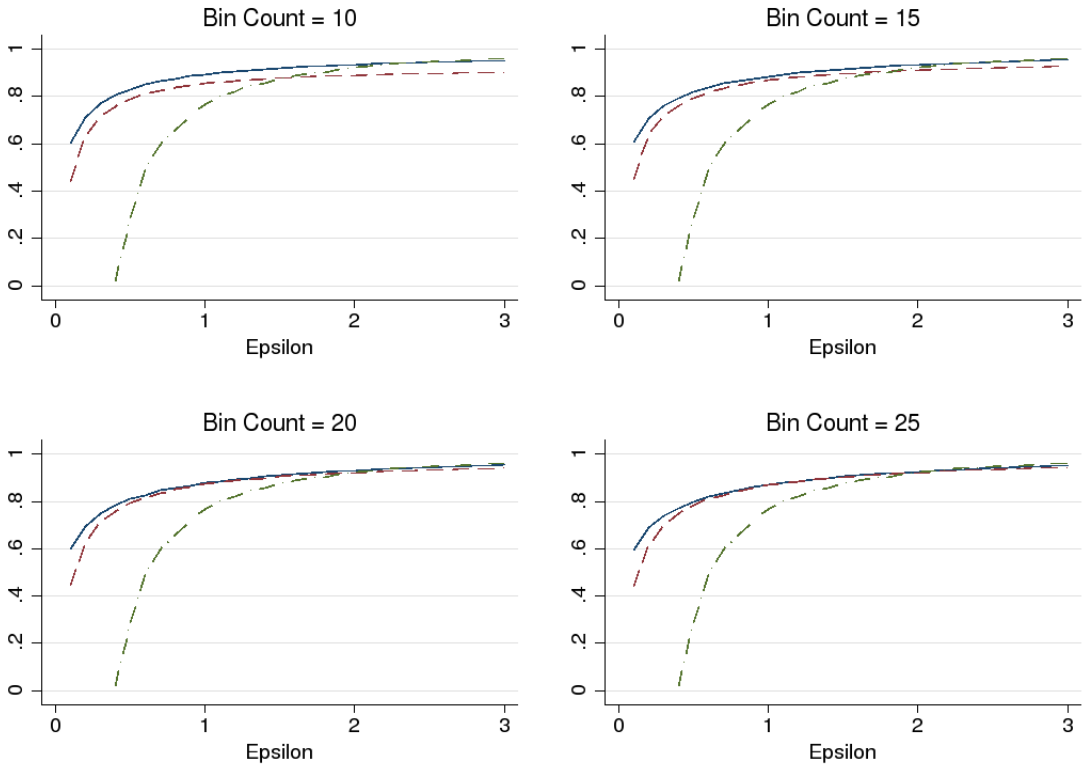
Appendix B. Tables and Figures Appendix

Table 1. Histogram bin values

Bin	Lower Bound	Upper Bound
1	10000	17403
2	17403	22876
3	22876	27512
4	27512	31857
5	31857	36128
6	36128	40449
7	40449	44914
8	44914	49605
9	49605	54609
10	54609	60027
11	60027	65982
12	65982	72639
13	72639	80226
14	80226	89080
15	89080	99735
16	99735	113106
17	113106	130970
18	130970	157509
19	157509	207050
20	207050	262475
21	262475	614597

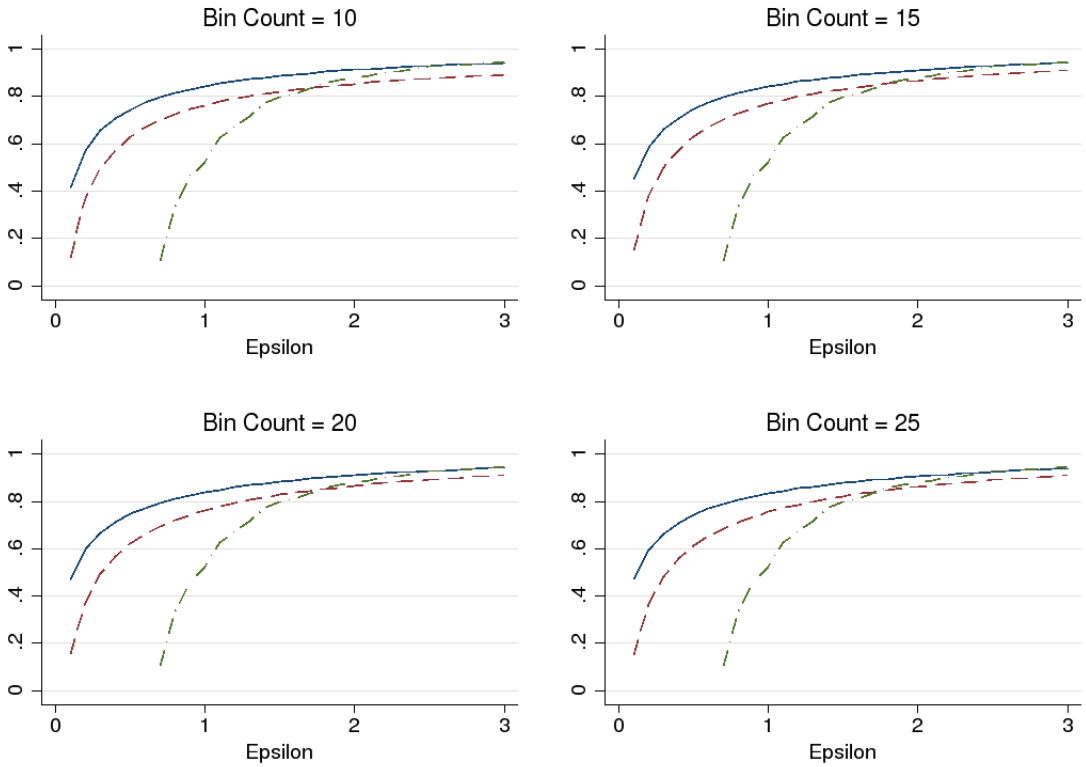
Notes: Except for the lowest value, these are all percentiles from a log normal distribution with mean 11.003 and standard deviation 0.753. Any observation will be classified into the final bin (21) if it has a value above 262475. For purposes of calculating the percentiles, we use the upper bound value for bin 21 of 614597, which is the 99.9th percentile of the log normal distribution.

FIGURE 4. Relative Accuracy by Histogram Method, 25th Percentile



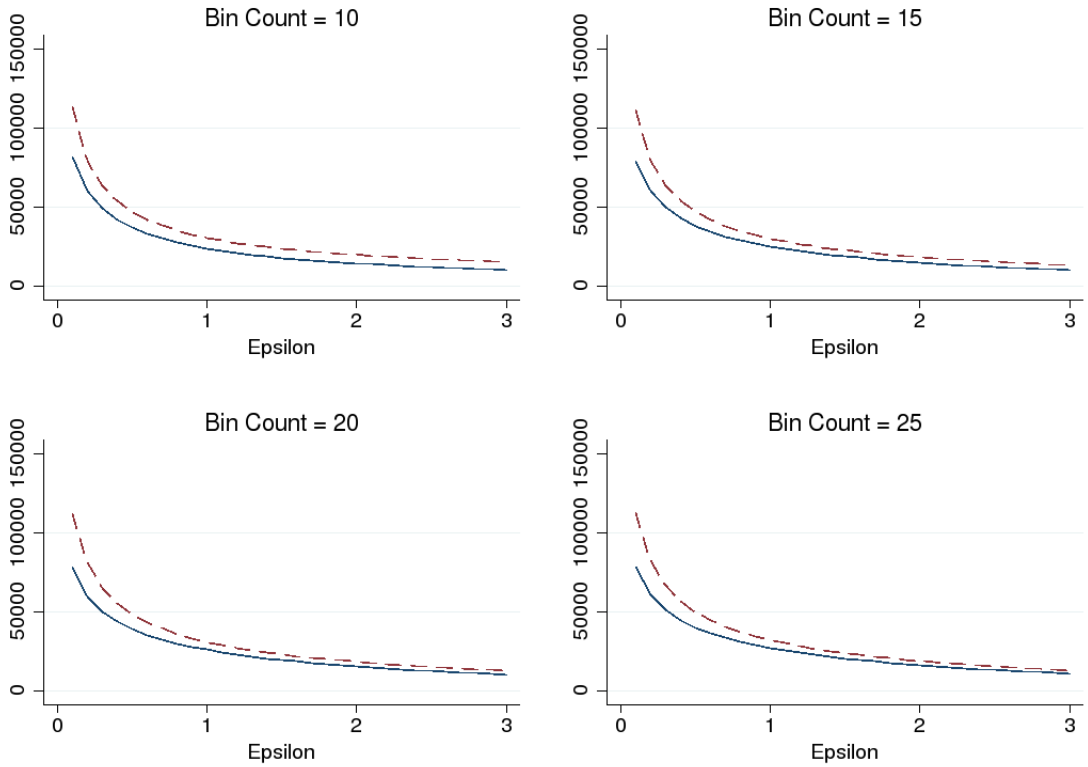
Notes: Log-Norm is Solid Blue; Even is Dashed Red; Smooth-sensitivity is Dash-dot Green

FIGURE 5. Relative Accuracy by Histogram Method, 75th Percentile



Notes: Log-Norm is Solid Blue; Even is Dashed Red; Smooth-sensitivity is Dash-dot Green

FIGURE 6. Overall L1 Error by Histogram Method



Notes: Log-Norm is Solid Blue; Even is Dashed Red. Error measure is the sum of the three L1 error measures across the 25th, 50th and 75th percentile measures.