System evaluation of PSI, a Differential Privacy prototype

Marcelo Novaes
marcelonovaes@dcc.ufba.br
Privacy Tools for Sharing Research Data
Harvard University
Cambridge, Massachusetts

Mentors: Georgios Kellaris and Marco Gaboardi

August 15, 2016

This document provides a summary of the evaluation of the PSI ("A Private data Sharing Interface") prototype regarding accuracy, efficiency, and security. It explains metrics, experiments and also describes the testing suite developed in R language.

1 Introduction

1.1 Background

Researchers in all experimental and empirical fields are increasingly expected to widely share the data behind their published research, to enable other researchers to verify, replicate, and extend their work [11]. Indeed, data-sharing is now often mandated by funding agencies and journals. However, many of these datasets contain sensitive data about human subjects, which cannot be shared without really strong privacy protections. Traditional "deidentification" methods seem to provide weak privacy protections, once examples of re-identified people in anonymized datasets permeates many different areas such as hospital health care records and search logs in the web [1, 7].

1.2 The Private data Sharing Interface (PSI - Ψ)

The Privacy Tools for Sharing Research Data group at Harvard is developing the PSI ("A Private data Sharing Interface") [11]. It is a system which enables researchers in social sciences and other fields to share and explore privacy-sensitive datasets. It relies on the strong privacy protection of Differential Privacy, instead of the traditional "deidentification" methods.

1.3 Project goal

This research project goal was to evaluate the PSI prototype regarding accuracy, efficiency, and security. Specifically, we evaluated the current PSI prototype [11],[10] in terms of:

- Accuracy: between statistics implemented in PSI and their non-private. We performed the evaluation on different datasets for different parameter values. Also, we bootstrapped different datasets varying the sample size and number of variables. Our focus was measuring accuracy by distance functions, but we also check the coverage (considering confidence intervals).

- Efficiency: the performance of the budgeting tool and the statistics release. We give a list of suggested optimizations for bottlenecks (Histogram routine and the optimal composition routine). Also, we checked if the algorithms consisting the budgeting tool and the statistic release run in a reasonable time frame.
- Security: Review of possible attacks on past differentially private systems [4, 6, 7] and PSI.

2 Main contributions and derivables

This project brought to the Privacy Tools group two main contributions:

- Reports evaluating the system
- The PSI evaluation package

2.1 Reports evaluating the system

A general evaluation of the system is composed by reports attached as appendices in this document.

- If not answer, at least we can now reason better about questions considered important by the group
  - (Primary) "Can we release all the statistics with meaningful accuracy given a specific budget?"
  - (Primary) "Given dataset parameters (sample size, number of variables,...), can we estimate how accurate PSI would be?"
  - (Secondary) "When decide for a post-processing"
  - (Secondary) "How does the bins/granularity affects our accuracy".
- We can make estimations
  - Given release properties, we can estimate accuracy and efficiency. This estimations can help data depositors to budget their privacy across the statistics.
- We identified the main bottlenecks and suggested optimizations (regarding efficiency)

2.2 The PSI evaluation package

- Experiment suite in R that generates accuracy metrics, plots and sanity checking, portable for different datasets.
- Additional features
  - In one of our experiments, it takes any measure as an input. As long as the user provides an equation and a standardized name for it. The experiment is called "Release statistics".
  - Reduce budget effort of the users, suggesting, by simple rules (the rules can be improved as a future work), statistics for dataset variables.

3 Measuring Accuracy

We evaluated accuracy by:

- Measuring accuracy error metrics.
- Showing intuitive ways to visualize the differentially private values.

We rely on the study that was done last year in the Privacy Tools for Sharing Research Data Project. In particular, we rely on Jessica Bu and Caper Gooden works. Jessica Bu, on [3], shows intuitive visualizations for Means, Histograms, and CDFs in terms of theoretical bounds for the error (e.g. confidence intervals). We tried to create visualizations considering the experimental data instead of the theoretical bounds. Caper Gooden, on [5], compares, using similar accuracy metrics, differentially private algorithms and their non-private version. We extend these comparisons to different metrics, multiple parameter variations, different datasets and variations of the same dataset (bootstrapped size and variables number variation).

3.1 Accuracy error metrics

The Mean Absolute Error (MAE) measures the absolute difference between the actual value (non-DP) and the differentially private one. The Mean Relative Error (MRE) computes also the difference, but it divides by the original value in order to derive the relative distance. The Mean Squared Error (MRE) returns the square of the difference, resembling the variance of the error (it is actually the composition of variance and bias). The Root of the Mean Squared Error puts gives an interpretable result of the MSE. We also report other suggested metrics: L1, L2 and L-infinity norm. We give the exact equations below, where $actual_i$ is the actual value and $released_i$ is its differentially private counterpart.

MAE = $\frac{1}{n}\sum_{i=1}^{n}|released_i - actual_i|$

MRE = $\frac{1}{n}\sum_{i=1}^{n}|\frac{released_i - actual_i}{actual_i}|$

MSE = $\frac{1}{n}\sum_{i=1}^{n}(released_i - actual_i)^2$

After a first set of evaluations, other accuracy error metrics were proposed:

RMSE = $\sqrt{(\frac{1}{n}\sum_{i=1}^{n}(released_i - actual_i)^2)}$

L1 norm = $MAE * dsize$, where dsize is the domain size*

L2 norm = $\sqrt{(MSE * \sqrt{(dsize)})}$, where dsize is the domain size*

L-infinity norm = for i from 1 to domain size: $max((released_i - actual_i)^2)$

Specifically for the "Release statistics" experiment, the system is also flexible to accept a metric you want, as long you provide the equation and a standardized name for it.The domain size for Histograms as for the CDFs are the number of bins.

3.2 Splitting the global ε among the statistics

We considered two ways of splitting the global epsilon value among the desired statistics. Consider the following example in the Public Use Microdata Samples (PUMS) from California. Suppose the requested statistics are the ones marked with X in the Figure 1.

	Mean	CDFs	Histogram
puma	-	-	x
sex	-	-	x
age	x	x	x
educ	-	-	x
income	x	x	x
latino	-	-	x
black	-	-	x
asian	-	-	x
married	-	-	x

x: Modified histogram to bin

Figure 1: Required statistics from PUMS California dataset

We have 9 variables of interest and 13 required statistics.
Let $\varepsilon = 0.1$, consider the different splitting methods.

- Approach 1: we split ε evenly among all the statistics released. In our example, each statistic would have local epsilon: $\varepsilon/13 \approx 0.0077$
- Approach 2: we split ε evenly among all the variables. Then for each variable, splitting its budget for all its requested statistics. In our example, each variable would have $\varepsilon/9 \approx 0.0111$, but the statistics of CDF and Income would have $0.0111/3 \approx 0.0037$.
- Approach 3: Fixing accuracy values, forcing the optimal composition. This approach is not in the R package, it might be possible use it through the 2014 GUI.R interface.

3.3 Examples

A specific value for at least one of our metrics (e.g. MRE 10%), for a data depositor, can possibly infer that the releases have meaningful accuracy. So, we check in the data/plots for each privacy parameter, if each of the released statistics meets this requirement (e.g. Figure 3). Also, we represented intuitively, using a boxplot, how a statistic changes according to a parameter variation (Figure 2).

3.4 Budget parameters variation

Consider the budgeting tool (Figure 4). The definitions of the parameters given by the PSI interface are the following:

- ε (global): from definition of differential privacy. Smaller values correspond to more privacy protection.
- δ (global): from definition of differential privacy. Smaller values correspond to more privacy protection.

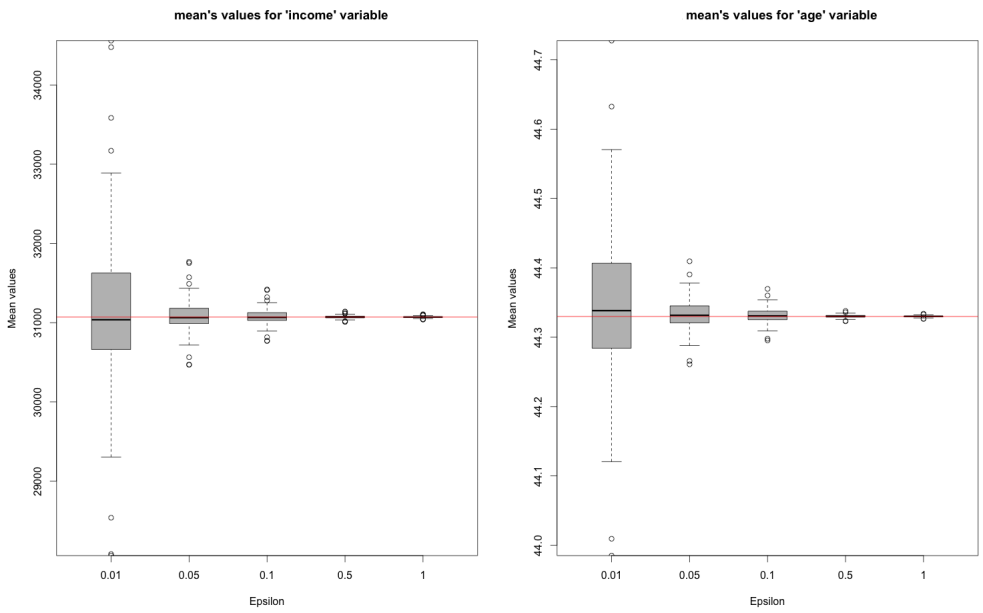


Figure 2: Intuitive representation for the Mean statistic for different global epsilons

- Secrecy of the sample: determines if the dataset is a random and secret sample from a larger population of known size. In this context, secret means that the choice of the people in the sample has not been revealed. If it holds, we can improve the accuracy of your statistics without changing the privacy guarantee.
- Functioning epsilon: when using secrecy of the sample, you get a boost in epsilon. This value can only be edited by changing the epsilon or secrecy of the sample fields.
- Sample size: number of records of a dataset.
- Variables: variables of interest in a study, usually represented by columns of dataframes loaded in the system.
- β (global): alpha level used in the prototype in order to define expected accuracy (in terms of confidence intervals).

The experimental parameter values are independent of the dataset. We run the experiment for each parameter **using the following range:**

3.4.1 Internal parameter variation

- ε: $\{0.01, 0.05, 0.1, 0.5, 1.0\}$
- Secrecy of the sample: $\{1\%, 3\%, 5\%, 100\%\}$
- δ : $2^{-}20$ (fixed)
- β : 0.05 (fixed)
- Granularity variations (e.g. for income at PUMS California, $\{\$1, \$10, \$100, \$1000\}$)

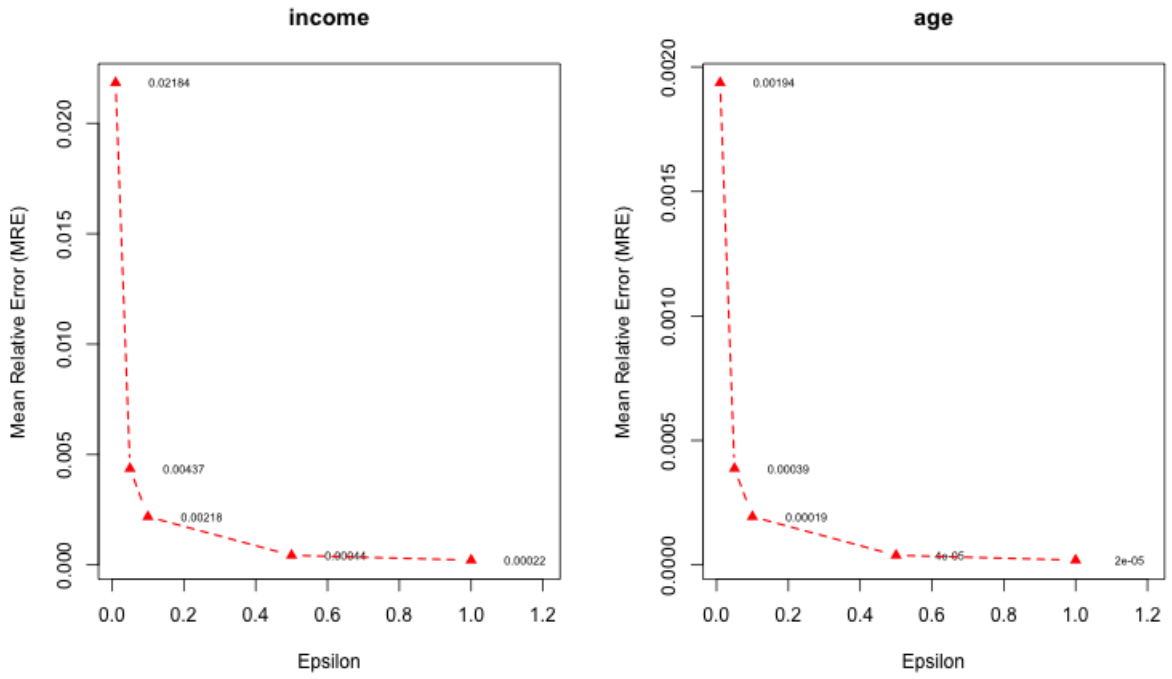


Figure 3: Mean's Mean Relative Error (MRE)

3.4.2 External parameter variation

- Sample size: $\{10^3, 10^4, 10^5, 10^6\}$
- Number of variables: $\{5, 10, 20, 50\}$
- ε: $\{0.01, 0.05, 0.1, 0.5, 1.0\}$

Different datasets were bootsptrapped (using R function *sample*()) in order to provide this experiment. A result can be seen in the Figure 5.

3.5 Experiments

The following experiments related to accuracy are implemented in the **tests.R:** file. We run each experiment 100 times.

Experiment 0: Release statistics: we do a **one time release** and report all accuracy metrics in the system (this is the test which accepts any metric you add on it).

Experiments 1-2: Accuracy for Mean: values of MAE, MRE, MSE, and for each value, accuracy as (MAE/range):

1. for different global epsilon values
2. for different values of secrecy of the sample

Experiments 3-4:Accuracy for Histograms: values of mae, mre, mse, and for each value, accuracy as specified in the theoretical way (MAE/n):

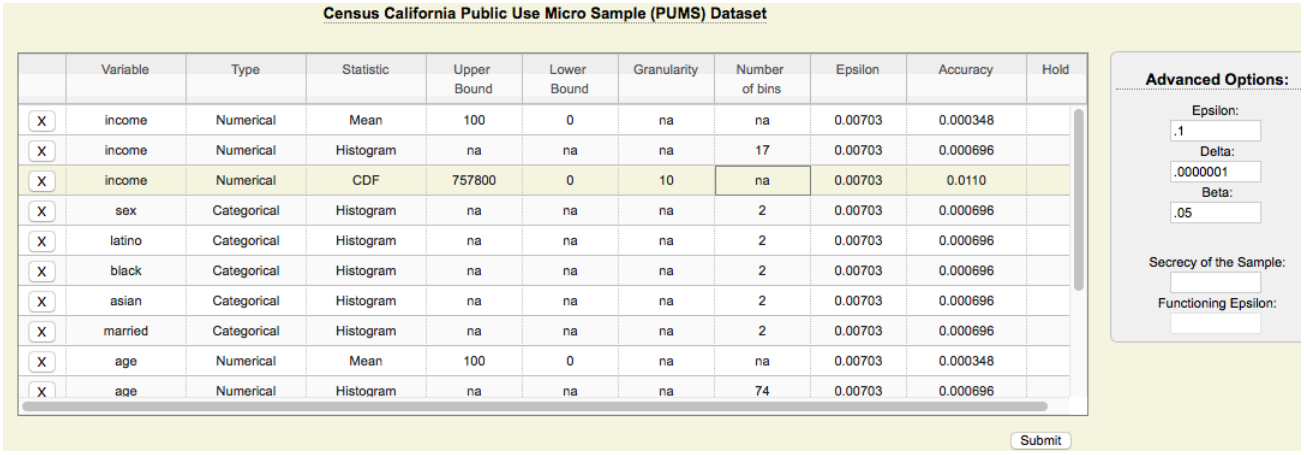


Figure 4: Budgeting tool interface

3. for different global epsilon values
4. for different values of secrecy of the sample

Experiments 5-6: Accuracy for CDFs

5. for different global epsilon values
6. for different values of secrecy of the sample

Experiments 7-8: Accuracy for CDF post-processing

7. for different global epsilon values
8. for different values of secrecy of the sample

Experiments 1-2: Intuitive plots: For the Mean, the experiment uses a boxplot for each of the 100x values and a line with the actual mean. We do not have a Histogram and CDF intuitive representation. Daniel Muise has an approach to Histograms.

9. for different global epsilon values
10. for different secrecy of the sample

Experiment 12: Granularity variations

Using make_acc_ref.R you can use the experiments above to generate results for external parameter variation.

4 Measuring Efficiency

In terms of efficiency, we mainly targeted two aspects:

- The statistics should be generated in a reasonable time frame.
- The composition routines (e.g. advanced composition) should be fast enough, without delaying the budgeting step.

We use as metric the total amount of time elapsed until the end of the computation. Furthermore, in order to define what is “slow”, based on [8], we use the following acceptance time:

- 0.1-1 sec for any user operation
- 10 seconds for any other computation (e.g. statistics release)

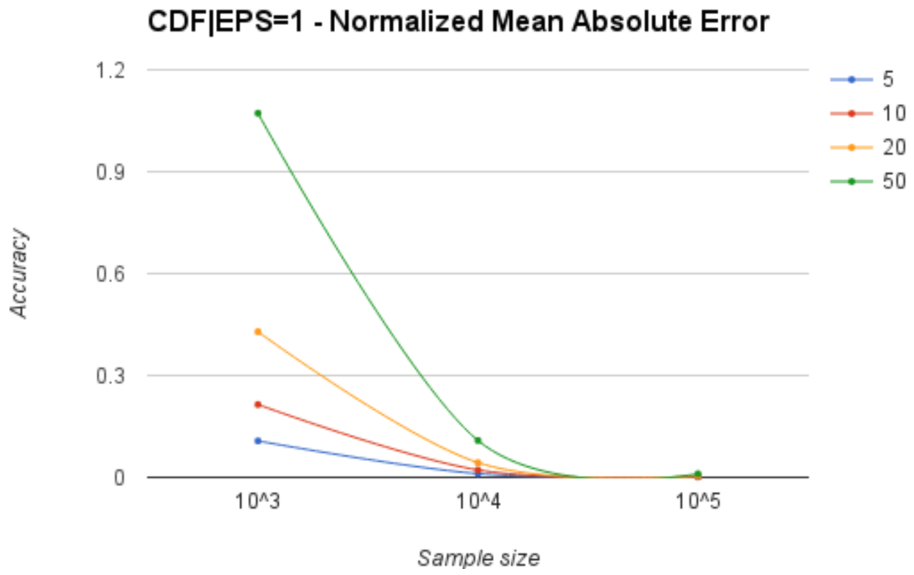


Figure 5: Analyzing the relation between sample size, number of variables and global epsilon

The overall result after experiments are PSI efficiency is not a problem. The algorithms are fast and composition routine does not delay the overall process. Even more, optimizations were proposed in case of future problems with the actual algorithms. All the releases tested meet the 10 seconds. The operations also meet 0.1-1 seconds. The past interface in one of the configurations tested, presented delay. However, it is not caused by the number of variables when using composition as we thought it was. It appeared to be quite randomly. If the group decides to use the past interface again or if the same delay appears in the new interface (the group haven't reported anything yet), it might be a future work.

After the tests it does not seem to be caused by the composition routine though. Finally, we should consider the total time it takes to the user do the whole budget step. However great part of it relies in the new interface of the system, what is a out of the scope of this project.

4.1 Experiments

The following experiments relating to efficiency are implemented in the **efficiency.R:** file.

Experiment 0: Compute and analyze the elapsed time of the differentially private univariate statistics routines.
Experiment 1: Compute and analyze all the elapsed time of the advanced composition for ε-update when adding a new statistic to be released (not in the R file, but in the Efficiency Report)
Experiment 2: Compute and analyze elapsed time of statistics release of actual datasets and releases.

4.2 Suggested optimizations on identified bottlenecks

- Histogram routine. Suggested optimization is described in the efficiency document. It is basically changes the current R function used $match()$, for another R function, $hist()$, making the necessary changes.

- The aggregate cubic time algorithm in the number of variables used by the composition to update the ε. Improvements, as the case when epsilons are split evenly, were introduced over the summer by the graduate student Jack Murtagh. Even more optimizations were proposed, but the running time is already fast enough to do not do an premature optimization. If someday necessary, the optimizations can be found in the Efficiency document.

5 Security

Regarding security, the work done was:

- Review of attacks on previous DP systems also considering PSI

As general comments, we also recommend:

- Once the interface is done, before release it, check if it meets a web server security, such as bayse2004security.
- Study R security packages such as RAppArmor [9].

6 The R package: psieval

The evaluation suite is composed by four main components: load.R, func.R, do.R, config_dataset.R. The load.R is responsible for helping load datasets and for releases descriptions, it also loads differentially private routines. The func.R is responsible for compute metrics and actually call the differentially private routines. The do.R is responsible for execute the tests, varying privacy parameters, filtering variables to be tested and saving the results. The config_dataset.R

load.R:

- **Function to load datasets** - $load\_datasets()$
  - input: a binary variable to tell us if the datafile is local or in a dataverse repository
  - output: dataset as a dataframe and the number of samples
- **Function to load differentially private routines** - $load\_dp\_routines()$: loads required Differentially Private routines currently in the system (Mean, Histogram, CDF, and CDF Post-processing).
  - input: none
  - output: none
  - side effect: load all dp routines
- **Function to request statistics**: hard coded (if time permits, a command line interface for the user) uses a function to set the seeds - $set\_seeds()$: set a seed for each statistic in order to make experiment results reproducible

func.R:

- **Function to compute accuracy for the DP CDF post-processing** $get\_post\_proc\_accuracy$

  - input: variable, binary variable for secrecy of the sample, local eps, files, iterations (default 120)
  - output: none
  - side effect: write all accuracy metrics for the CDF post-processing methods in the files passed as arguments.

- **Function to compute accuracy for the DP statistics** *get_stat_acc*
  - input: variables, string with statistic name, local epsilon, filename, iterations, delta
  - output: accuracy metrics for the statistics
- **Function to normalize the MAE to the theoretical routines**- *get_acc_from_mae*
  - input: value representing the MAE computed, statistic string, number of samples, vrange
  - output: get accuracy as in the theoretical returned in the (getAccuracy() method)
- **Function to compute the accuracy for the DP statistics for different granularity**- *get_stat_accuracy_gran*
  - input: variable, epsilon, statistic, array with granularity values, number of iterations
  - output: accuracy metrics for the statistics varying the granularity
- **Function to retrieve the metric function** *get_metric*
  - input: string specifying an Accuracy metric
  - output: metric function
- **Functions comparing statistics on raw data with the DP ones**
  - **Histogram:***get_histogram*
    - input: variable, granularity
    - output: non-private histogram
  - **CDF:***get_CDF*
    - input: variable, granularity
    - output: non-private cdf
  - **Number of bins:***nbins*()
    - input: data and granularity
    - output: get number of bins
- **Functions to compute the metrics**
  - **All error metrics cited in 3.1**
    - input: original value and DP value
    - output: result of the metric equation
- **Function to write the released statistics** *release_stats*:
  - input: variables, epsilon, secrecy of the sample, granularity, number of iterations, file, delta, beta and id of the file
  - output: all metrics of the system and general information about a one iteration of each variation
- **Function to plot the accuracy statistics** - *plot_stat_accuracy*
  - input: dataset variable, local epsilon, metric, a label for x- plotting, binary variable for secrecy of the sample, flag if it should plot the theoretical values, delta and number of iterations.
  - output: accuracy line plotting and if flag for theoretical plot is on, theoretical line plotting

References

[1] Michael Barbaro, Tom Zeller & Saul Hansell (2006): *A face is exposed for AOL searcher no. 4417749*. New York Times 9(2008), p. 8For.

[2] Gail Zemanek Bayse (2004): *A Security Checklist for Web Application Design*. SANS Institute, Practical Assignment, Version 1.

[3] Jessica Bu (2015): *Visualization of Uncertainty in Differential Privacy*. Available at `http://privacytools.seas.harvard.edu/training-students-researchers`.

[4] Úlfar Erlingsson, Vasyl Pihur & Aleksandra Korolova (2014): *Rappor: Randomized aggregatable privacy-preserving ordinal response*. In: Proceedings of the 2014 ACM SIGSAC conference on computer and communications security, ACM, pp. 1054–1067.

[5] Caper Gooden (2015): *Testing Usability of Differentially Private Estimates*. Available at `http://privacytools.seas.harvard.edu/training-students-researchers`.

[6] Frank D McSherry (2009): *Privacy integrated queries: an extensible platform for privacy-preserving data analysis*. In: Proceedings of the 2009 ACM SIGMOD International Conference on Management of data, ACM, pp. 19–30.

[7] Prashanth Mohan, Abhradeep Thakurta, Elaine Shi, Dawn Song & David Culler (2012): *GUPT: privacy preserving data analysis made easy*. In: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, ACM, pp. 349–360.

[8] Jakob Nielsen (1994): *Usability engineering*. Elsevier.

[9] Jeroen Ooms (2013): *The RAppArmor Package: Enforcing Security Policies in R Using Dynamic Sandboxing on Linux*. Journal of Statistical Software 55(7), pp. 1–34. Available at `http://www.jstatsoft.org/v55/i07/`.

[10] Privacy Tools for Sharing Research Data Group (2015): *PSI prototype*. Available at `https://beta.dataverse.org/custom/DifferentialPrivacyPrototype/UI/code/interface.html`.

[11] Privacy Tools for Sharing Research Data Group (April, 2016): *PSI - A private data Sharing Interface*. Not in circulation.

Marcelo Novaes, Summer Intern 2016

PSI experiments - Specific Accuracy Evaluation

PUMS California

Dataset setting

Number of statistics	Number of variables	Sample size
9	13	~1.2M

User statistics release (assumption)

	Mean	CDFs	Histogram
puma	-	-	x
sex	-	-	x
age	x	x	x
educ	-	-	x
income	x	x	x
latino	-	-	x
black	-	-	x
asian	-	-	x
married	-	-	x

x: Modified histogram to bin

Assumptions

No hold applied, i.e. without fixing accuracy
Default #bins: (puma,17); (educ,16)

Parameters variation

Delta variation: 2^-20
Epsilon variation: 0.01, 0.05, **0.1**, 0.5, 1
Secrecy of the sample: **100%**, 1% (12239.92) 3% (36719.76) 5% (61199.6)
Income granularity 1,10, **100**,1000

Metrics

Mean Absolute Error; Mean Relative Error; Mean Squared Error

Classification

Seems to have meaningful accuracy

As the first experiment, PUMS was analyzed only considering MRE <= 10% as having a possible objective function which would tell the user he has meaningful accuracy.

Seems not to have meaningful accuracy

Summary comments

- **All basic statistics are OK (MRE < 10%) for both approaches.**
- **Histograms are really good for most of the variables, because they are binary variables (two bins). Even for histograms with larger number of bins, it STILL MEETS the 10% MRE for "age" that has 76 bins and for "income" that depending of its granularity has 778, 7779, 77781 or 777801 bins.**
- **All the statistics are bounded by the theoretical values. For Means, I would say the bounds are the exactly 95%. However, the exactly 95% percentile sometimes does not coincide with the experimental 95% on Histogram and CDFs for the 100 iterations.**
- **Most of post-processing over the CDFs did not meet the (MRE < 10%). The necessity for evaluate the post-processing is that many current methods in the system can be obtained only by the CDF-post-processing. After , we moved then to a separated experiment, and we focused here o**

System problems solved during tests

- **Solved**
  - **The Differentially Private Histogram assumes the data to be categoric (it does not bin any data). It had to be modified.**
    - **For continuous data, it had to be binned, for different granularities.**

- **Stack overflow in the CDF .cpp file for a big number of bins (income with granularity 1). Variables changed to heap (dynamic arrays).**
- **None of the Differentially Private implementation statistics deal with NAs, NaNs. It had to be done.**
- **The DP Histogram returned only the used bins**
  - **It makes the error lower because it adds less noise**
  - **However, the algorithm removing the unused bins is not Differentially Private**

- **Not solved**
  - **Bounds are not too tight. Histograms and CDFs do not achieve always the 95% confidence interval. Usually it is an upper bound, achieving for the theoretical 95%, a 100%**

Summary :

Approach 1: budget epsilon evenly among all statistics

ALL STATS - GLOBAL EPSILON VARIATION

	0.01	0.05	0.1	0.5	1	file_id; plot_id
PUMA HIST						11; 11p2
SEX HIST						11; 11p0
AGE MEAN						3; 3
AGE CDF						13; -
AGE HIST						11; 11p0
EDUC HIST						11; 11p2
INC MEAN						3; 3
INC CDF						13; -
INC HIST						11; 11p0
LATINO HIST						11; 11p0
BLACK HIST						11; 11p1
ASIAN HIST						11; 11p1
MARRIED HIST						11; 11p2

ALL STATS - SECRECY OF THE SAMPLE VARIATION

	1%	3%	5%	100%	file_id; plot_id
PUMA HIST					12; 12p2
SEX HIST					12; 12p0
AGE MEAN					4; 4
AGE CDF					14; -
AGE HIST					12; 12p0
EDUC HIST					12; 12p2
INC MEAN					4; 4
INC CDF					14; -
INC HIST					12; 12p0
LATINO HIST					12; 12p1
BLACK HIST					12; 12p1
ASIAN HIST					12; 12p1
MARRIED HIST					12; 12p2

INCOME - GRANULARITY VARIATION

	1	10	100	1000	file_id; plot_id
INC HIST					18; -

INC CDF						17; -

Approach 2: budget epsilon evenly among all variables. Then split each variable epsilon among its statistics.

The results form the second approach are in the folder /approach-2/. The file_id and plot_id are the same as above.

ALL STATS - GLOBAL EPSILON VARIATION

	0.01	0.05	0.1	0.5	1
PUMA HIST					
SEX HIST					
AGE MEAN					
AGE CDF					
AGE HIST					
EDUC HIST					
INC MEAN					
INC CDF					
INC HIST					
LATINO HIST					
BLACK HIST					
ASIAN HIST					
MARRIED HIST					

ALL STATS - SECRECY OF THE SAMPLE VARIATION

	100%	5%	3%	1%
PUMA HIST				
SEX HIST				
AGE MEAN				
AGE CDF				
AGE HIST				
EDUC HIST				
INC MEAN				
INC CDF				
INC HIST				
LATINO HIST				
BLACK HIST				
ASIAN HIST				
MARRIED HIST				

INCOME - GRANULARITY VARIATION

	1	10	100	1000
INC HIST				
INC CDF				

CLASSIFYING ACCURACY ON POST-PROCESSING

It was just an experiment, done for the first dataset: PUMS - California. Then, accepting suggestions, we considered just the actual releases, and moved the post-processing evaluation to its own document. Doing a deeper analysis.

Approach 1:

POST-PROCESSING (GLOBAL EPSILON VARIATION)

	0.01	0.05	0.1	0.5	1	file_id; plot_id
INC Mean						15; -
INC Variance						15; -
INC Stand. Dev.						15; -

INC Kurtosis						15; -
INC Skewness						15; -
INC Histogram						15; -
INC Percentil*						15; -
INC Mode*						15; -
INC Zeros*						15 ;-

	0.01	0.05	0.1	0.5	1	file_id; plot_id
AGE Mean						15; -
AGE Variance						15; -
AGE Stand.Dev.						15; -
AGE Kurtosis						15; -
AGE Skewness						15; -
AGE Histogram						15; -
AGE Percentil*						15; -
AGE Mode*						15; -
AGE Zeros*						15 ;-

POST-PROCESSING (SECRECY OF THE SAMPLE VARIATION)

	100%	5%	3%	1%	file_id; plot_id
INC Mean					
INC Variance					
INC Standard deviation					
INC Kurtosis					
INC Skewness					
INC Histogram					
INC Percentil*					
INC Mode*					
INC Zeros*					

	100%	5%	3%	1%	file_id; plot_id
AGE Mean					16; -
AGE Variance					16; -
AGE Standard deviation					16; -
AGE Kurtosis					16; -
AGE Skewness					16; -
AGE Histogram					16; -
AGE Percentil*					16; -
AGE Mode*					16; -
AGE Zeros*					16; -

Approach 2:

POST-PROCESSING (GLOBAL EPSILON VARIATION)

	0.01	0.05	0.1	0.5	1
INC Mean					
INC Variance					
INC Standard Dev.					
INC Kurtosis					
INC Skewness					
INC Histogram					
INC Percentil*					

INC Mode*					
INC Zeros*					

	0.01	0.05	0.1	0.5	1
AGE Mean					
AGE Variance					
AGE Standard Dev.					
AGE Kurtosis					
AGE Skewness					
AGE Histogram					
AGE Percentil*					
AGE Mode*					
AGE Zeros*					

POST-PROCESSING (SECRECY OF THE SAMPLE VARIATION)

	100%	5%	3%	1%
INC Mean				
INC Variance				
INC Standard deviation				
INC Kurtosis				
INC Skewness				
INC Histogram				
INC Percentil*				
INC Mode*				
INC Zeros*				

	100%	5%	3%	1%
AGE Mean				
AGE Variance				
AGE Standard deviation				
AGE Kurtosis				
AGE Skewness				
AGE Histogram				
AGE Percentil*				
AGE Mode*				
AGE Zeros*				

Notes for the post-processing part:

- **MRE definition changed before/after post-processing**
  - before: divide by max(original,1) after: divide by the non-zero minimum absolute value.
  - It is due the Skewnewss, Kurtosi, Percentil, Range Queries, that have decimal values original values.
- **Range Queries not in the table**
  - There is a function which compares one differentially private range query result against the original one. However, it is not a good approach to evaluate it. A good one would be see the average error of all possible range queries and compare it in some way. Possible future work.

Full experiment informations

Release file [Hard coded]**:**
https://github.com/IQSS/PrivateZelig/tree/evaluation/experiments/datasets/pum_rel.R

Plots:
https://github.com/IQSS/PrivateZelig/tree/evaluation/experimenpums-california/plots/released

Text:

https://github.com/IQSS/PrivateZelig/tree/evaluation/experimenpums-california/outputs/released

Compulsory Voting Accuracy Evaluation

Dataset setting

Number of statistics	Number of variables	Sample size
16	10	~1.8M

User statistics release (assumption)

	Mean	CDFs	Histogram
fv	x	x	x
mob	x	x	x
dob	x	x	x
yob	-	-	x
marital_status	-	-	x
education	-	-	x
sexo	-	-	x
uf	-	-	x
turnout	-	-	x
treat	-	-	x

Assumptions

No hold applied, i.e. without fixing accuracy
Default #bins: yob = 3; sexo = 3; uf = 27; marital_status: 6; fv = 729; dob = 31

Parameters variation

Delta variation: 2^-20
Epsilon variation: 0.01, 0.05, **0.1**, 0.5, 1
Secrecy of the sample: **100%**, 1% (12239.92) 3% (36719.76) 5% (61199.6)
Income granularity 1,10, **100**,1000

Metrics in the files

Mean Absolute Error; Mean Relative Error; Mean Squared Error

Classification

Seems to have meaningful accuracy

- **MRE < 10% are green and no text in the cell**
- **MRE <= 10% <= 40% green and text in the cell with actual value.**

Seems to have not meaningful accuracy

System problems solved during tests

- Solved
  - NaN's and NA's had to be handled

Summary :

Approach 1: budget epsilon evenly among all statistics

ALL STATS - GLOBAL EPSILON VARIATION

	0.01	0.05	0.1	0.5	1	file_id; plot_id
FV HIST						11; 11p
MOB HIST						11; 11p
DOB HIST						11; 11p
YOB HIST						11; 11p
M_STATUS HIST						11; 11p
EDUC HIST						
SEXO HIST						11; 11p
UF HIST						11; 11p
TURNOUT HIST						11; 11p
TREAT HIST						11; 11p
FV CDF						13; -
MOB CDF						13; -
DOB CDF						13; -

FV Mean						3; 3
MOB Mean						3; 3
DOB Mean						3; 3

ALL STATS - SECRECY OF THE SAMPLE VARIATION

	1%	3%	5%	100%	file_id; plot_id
FV HIST					12; 12p2
MOB HIST					12; 12p0
DOB HIST					4; 4
YOB HIST					14; -
M_STATUS HIST					12; 12p0
EDUC HIST					12; 12p2
SEXO HIST					4; 4
UF HIST					14; -
TURNOUT HIST					12; 12p0
TREAT HIST					12; 12p1
FV CDF					12; 12p1
MOB CDF					12; 12p1
DOB CDF					12; 12p2
FV Mean					
MOB Mean					
DOB Mean					

Approach 1: budget epsilon evenly among all statistics

ALL STATS - GLOBAL EPSILON VARIATION

	0.01	0.05	0.1	0.5	1	file_id; plot_id
FV HIST	3.77 MRE	0.91 MRE	0.18 MRE			11; 11p
MOB HIST						11; 11p
DOB HIST						11; 11p
YOB HIST						11; 11p
M_STATUS HIST						11; 11p
SEXO HIST						11; 11p
UF HIST						11; 11p
TURNOUT HIST						11; 11p
TREAT HIST						11; 11p
FV CDF						13; -
MOB CDF						13; -
DOB CDF						13; -
FV Mean						3; 3
MOB Mean						3; 3
DOB Mean						3; 3

EDUC MRE acted weird and was removed

ALL STATS - SECRECY OF THE SAMPLE VARIATION

	1%	3%	5%	100%	file_id; plot_id
FV HIST				0.18 MRE	12; 12p2
MOB HIST					12; 12p0
DOB HIST					4; 4
YOB HIST					14; -
M_STATUS HIST					12; 12p0
EDUC HIST					12; 12p2
SEXO HIST					4; 4
UF HIST					14; -
TURNOUT HIST					12; 12p0

TREAT HIST					12; 12p1
FV CDF					12; 12p1
MOB CDF					12; 12p1
DOB CDF					12; 12p2
FV Mean					
MOB Mean					
DOB Mean					

Full experiment informations

Release file [Hard coded]**:**
https://github.com/IQSS/PrivateZelig/tree/evaluation/experiments/datasets/comp_rel.R
Plots:
https://github.com/IQSS/PrivateZelig/tree/evaluation/experiments/comp-voting/plots/released
Text:
https://github.com/IQSS/PrivateZelig/tree/evaluation/experiments/comp-voting/outputs/released

PSI experiments - Pew Global

Dataset setting

Number of statistics	Number of variables	Sample size
21	21	48k (48643)

User statistics release (assumption)

Variables	Histograms
Q91A **to** Q91J **(10 variables)**	x
Q93A **to** Q93J **(10 variables)**	x
Q94	x

Other assumptions

No hold applied, i.e. without fixing accuracy

Parameters variation

Delta variation: 2^-20
Epsilon variation: 0.01, 0.05, **0.1**, 0.5, 1
Secrecy of the sample: **100%**, 1% (12239.92) 3% (36719.76) 5% (61199.6)
Income granularity 1,10, **100**,1000

System problems solved during tests

- Solved
    - NaN's and NA's had to be handled

Summary :

Simple Classification

Seems to have meaningful accuracy

- **Normalized** Mean **Absolute** Error <= 10%

Seems not to have meaningful accuracy

Approach 1: budget epsilon evenly among all statistics

ALL STATS - GLOBAL EPSILON VARIATION

	0.01	0.05	0.1	0.5	1	file_id; plot_id
Q91A						11; 11p0
Q91B						11; 11p0
Q91C						11; 11p0
Q91D						11; 11p1
Q91E						11; 11p1
Q91F						11; 11p1
Q91G						11; 11p1
Q91H						11; 11p2
Q91I						11; 11p2
Q91J						11; 11p2
Q93A						11; 11p3
Q93B						11; 11p3
Q93C						11; 11p3
Q93D						11; 11p4
Q93E						11; 11p4
Q93F						11; 11p4
Q93G						11; 11p5
Q93H						11; 11p5
Q93I*						11; 11p5
Q93J						11; 11p6
Q94						11; 11p6

ALL STATS - SECRECY OF THE SAMPLE VARIATION

	100%	5%	3%	1%	file_id; plot_id
Q91A					12; 12p0
Q91B					12; 12p0
Q91C					12; 12p0

Q91D					12; 12p1
Q91E					12; 12p1
Q91F					12; 12p1
Q91G					12; 12p1
Q91H					12; 12p2
Q91I					12; 12p2
Q91J					12; 12p2
Q93A					12; 12p3
Q93B					12; 12p3
Q93C					12; 12p3
Q93D					12; 12p4
Q93E					12; 12p4
Q93F					12; 12p4
Q93G					12; 12p5
Q93H					12; 12p5
Q93I					12; 12p5
Q93J					12; 12p6
Q94					12; 12p6

Stronger Classification

Seems to have meaningful accuracy

- **Normalized** Mean Absolute Error <= 10%
- Mean Relative Error <= 40% (Written)
    - Mean Relative Error <=10% (Not written)

Seems not to have meaningful accuracy

ALL STATS - GLOBAL EPSILON VARIATION

	0.01	0.05	0.1	0.5	1	file_id; plot_id
Q91B				MRE 0.17		11; 11p0
Q91E				MRE 0.21		
Q91F						11; 11p1
Q91G						11; 11p1
Q91H						11; 11p2
Q91I						11; 11p2
Q91J				MRE 0.12		11; 11p2
Q93A						11; 11p3
Q93B						11; 11p3
Q93C			MRE 0.30			11; 11p3
Q93D			MRE 0.33			11; 11p4
Q93E				MRE 0.2		11; 11p4
Q93F			MRE 0.23			11; 11p4
Q93G			MRE 0.38	MRE 0.25		11;
Q93H						11;

MRE acted weird on Q91A, Q91C, Q91D, Q93I, Q94 so they were removed. Lacking Q91e, Q93j, Q94 graphics.

ALL STATS - SECRECY OF THE SAMPLE VARIATION

	100%	5%	3%	1%	file_id; plot_id
Q91A					12; 12p0
Q91B					12; 12p0
Q91C					12; 12p0
Q91D					12; 12p1
Q91E					
Q91F					12; 12p1
Q91G					12; 12p1
Q91H					12; 12p2
Q91I					12; 12p2
Q91J					12; 12p2
Q93A					12; 12p3
Q93B					12; 12p3
Q93C	MRE 0.3				12; 12p3
Q93D	MRE 0.33				12; 12p4
Q93E					12; 12p4
Q93F	MRE 0.23				12; 12p4
Q93G	MRE 0.38				12; 12p5
Q93H					12; 12p5
Q93I					12; 12p5
Q93J					12;
Q94					12;

Lacking Q91e, Q93j, Q94 graphics.

Full experiment informations

Release file [Hard coded]**:**
https://github.com/IQSS/PrivateZelig/tree/evaluation/experiments/datasets/comp_rel.R
Plots:
https://github.com/IQSS/PrivateZelig/tree/evaluation/experiments/comp-voting/plots/released
Text:
https://github.com/IQSS/PrivateZelig/tree/evaluation/experiments/comp-voting/outputs/released

Marcelo Novaes, Summer Intern 2016

PSI experiments - Differentially Private bounds

Experiment done in order to produce an Accuracy Reference Card in the future and to meet the Extensional Work proposed by Salil Vadhan, and later by James Honaker in his lecture about “Statistical Weights and Measuring Usefulness”

Introduction

The objective of this document is exploit patterns among a chosen epsilon, number of variables, sample size and when decide for save budget, generating Means and Histograms from the CDF . It can be represented by the question “Given dataset parameters (sample size, number of variables,...), can we estimate how accurate PSI would be?”. The aiming is help data depositors when deciding about budget allocation among the statistics.

Parameters variation

- **Global Epsilon**:
  - 0.01, 0.05, 0.5, 1
- **Number of variables**
  - 5, 10, 20, 50
- **Sample size**
  - 10^3, 10^4, 10^5, 10^6

Process to generate the variations

- Generating different sample sizes: done via statistical bootstrap, using sample() R function.
- Generating different number of variables: changed the variable that conceptually counted this number. It affected in the local epsilon. It was not necessary add an actual variable to the dataset.
- Generating different global epsilons: normal variation that had been done for other experiments.

Fixed parameters

Delta: 2^-20
Granularities: income, 100. Age, 1.

Results

It is summarized the results considering the Normalized Mean Absolute Error. For other metrics, you should access one of the following folders:

- <dataset>-1k
- <dataset>-10k
- <dataset>-100k

Inside them there will be two folders

- outputs

- plots

Inside any of them there are the folders: **5 10 20 50**.

Assumptions

No hold applied, i.e. without fixing accuracy
Default #bins: 779 for income (considering the default granularity 100) and 76 for age (default granularity 1)

Datasets

The summarized results (plots below) are from the following dataset:

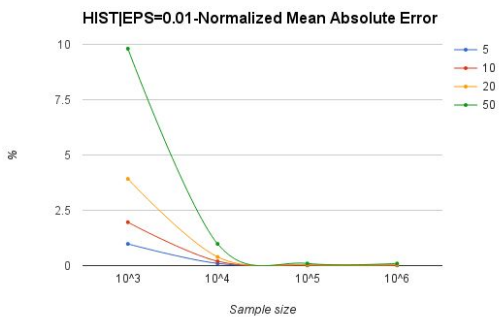
- PUMS California already in the system

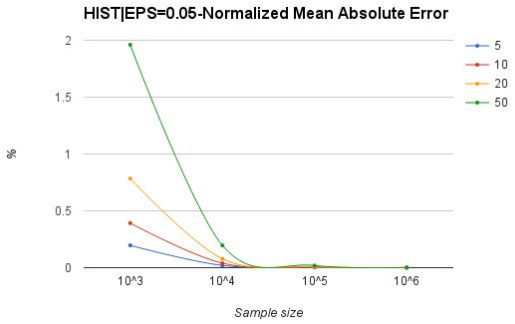
It was also generated data for other four datasets:

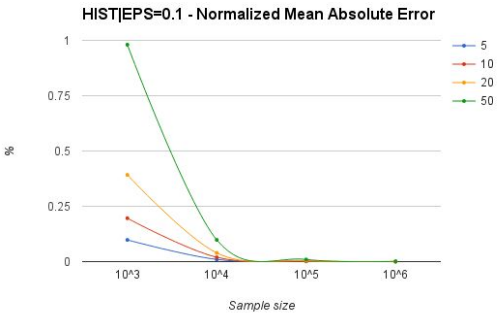
- “Compulsory Voting Can Increase Political Inequality: Evidence from Brazil”
- “Social Pressure and Voter Turnout: Evidence from a Large-Scale Field Experiment”
- “Do Perceptions of Ballot Secrecy Influence Turnout? Results from a Field Experiment”
- “Many in Emerging and Developing Nations Disconnected from Politics”

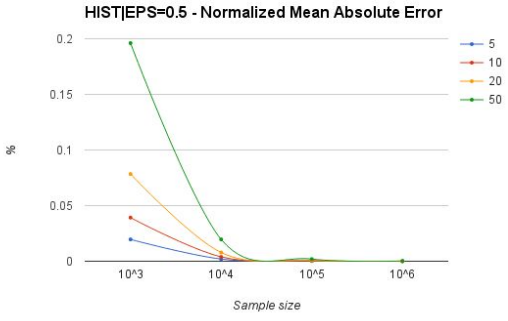
SUMMARIZATION

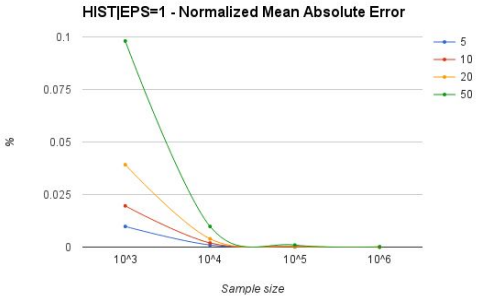
- Histograms



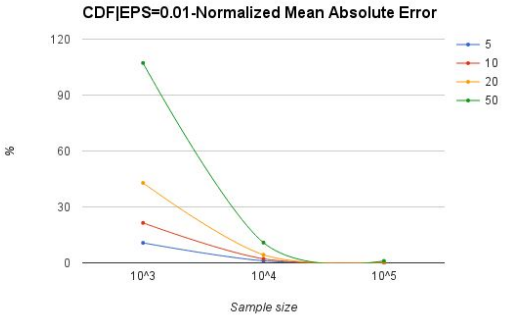


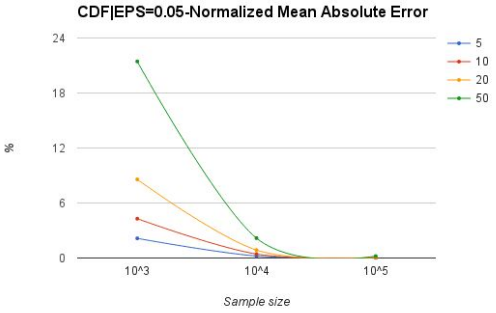


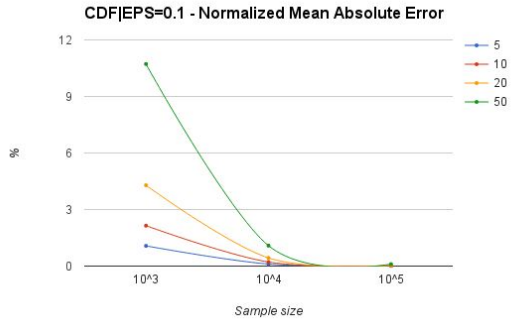


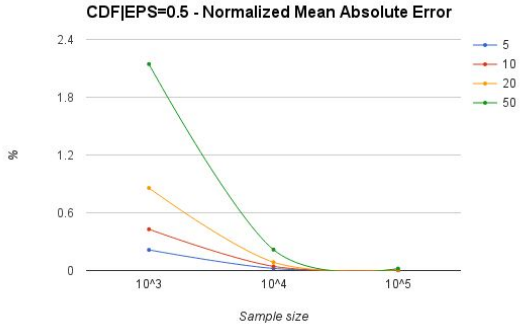

HIST|EPS=1 - Normalized Mean Absolute Error
0.1
0.075
0.05
0.025
0
%
10^3
10^4
10^5
10^6
Sample size
5
10
20
50

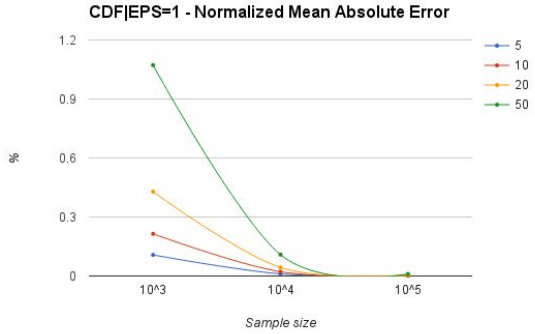
- CDFs


CDF|EPS=0.01-Normalized Mean Absolute Error
120
90
60
30
0
%
10^3
10^4
10^5
Sample size
5
10
20
50

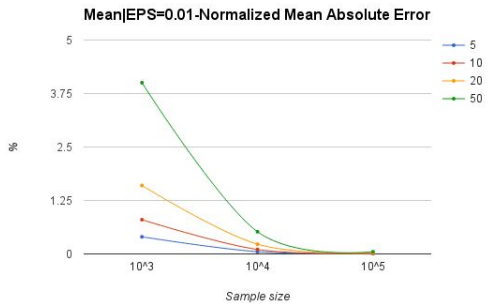

CDF|EPS=0.05-Normalized Mean Absolute Error
24
18
12
6
0
%
10^3
10^4
10^5
Sample size
5
10
20
50

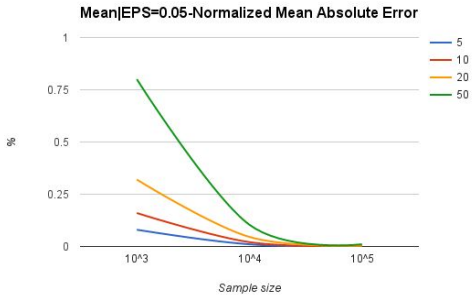

CDF|EPS=0.1 - Normalized Mean Absolute Error
12
9
6
3
0
%
10^3
10^4
10^5
Sample size
5
10
20
50

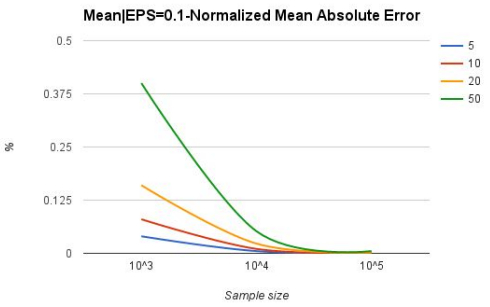

CDF|EPS=0.5 - Normalized Mean Absolute Error
2.4
1.8
1.2
0.6
0
%
10^3
10^4
10^5
Sample size
5
10
20
50

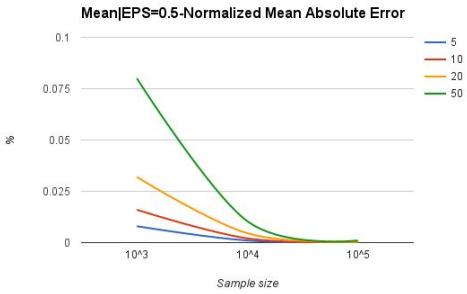

CDF|EPS=1 - Normalized Mean Absolute Error
1.2
0.9
0.6
0.3
0
%
10^3
10^4
10^5
Sample size
5
10
20
50

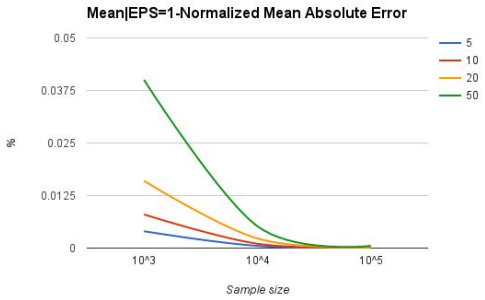
- Mean











Conclusions

As we have discussed so far during the summer, there is no a global limit for how much utility is acceptable to be lost. It highly depend on the analysis that is going to be done in the dataset. However, it follows a guidance considering 10% of the mean absolute error normalized by the range. For the four datasets took from the replication group, they seem to be a reasonable choice.

Global epsilon decision

Number of variables

	5	10	20	50
10^3	1	1*	**> 1**	**> 1**
10^4	0.1	0.5	0.5	1

10^5	0.01	0.05	0.05	0.1
10^6	0.01	0.01	0.01	0.01

*20%

By being conservative, we mean we consider the CDF accuracy as a process decision, because they introduce more noise than the Histograms and Means algorithms. If you want to release for example only histograms and means, you probably can choose lower epsilons.

The Global Epsilon table was created based on this CDF summary. The other results, summaries for Means and Histograms, also each variable result, can be found based in the files in the github branch /evaluation/. There is a sheet called “Extension work - summary table” that can be requested to the project coordinator.

EPS=0.01	**10^3**	**10^4**	**10^5**	**10^6**
5	10.720938	1.0839323	0.10447983	0.008718306
10	21.441876	2.1678651	0.20895966	0.017436612
20	42.883745	4.3357295	0.41791925	0.03487322
50	107.20938	10.839323	1.0447983	0.08718306
EPS=0.05	**10^3**	**10^4**	**10^5**	**10^6**
5	2.1441876	0.21678651	0.020895966	0.0017436612
10	4.288375	0.433573	0.04179193	0.0034873225
20	8.5767505	0.86714605	0.083583855	0.006974646
50	21.441876	2.1678651	0.20895966	0.017436612
EPS=0.1	**10^3**	**10^4**	**10^5**	**10^6**
5	1.0720938	0.10839323	0.010447983	0.0008718306
10	2.1441876	0.21678651	0.020895966	0.0017436612
20	4.288375	0.433573	0.04179193	0.0034873225
50	10.720938	1.0839323	0.10447983	0.008718306
EPS=0.5	**10^3**	**10^4**	**10^5**	**10^6**
5	0.21441876	0.021678651	0.0020895966	0.00017436612
10	0.4288375	0.043357295	0.004179193	0.00034873225
20	0.85767505	0.086714605	0.0083583855	0.0006974646
50	2.1441876	0.21678651	0.020895966	0.0017436612

EPS=1	**10^3**	**10^4**	**10^5**	**10^6**
5	0.10720938	0.010839323	0.0010447983	0.00008718306
10	0.21441876	0.021678651	0.0020895966	0.00017436612
20	0.4288375	0.043357295	0.004179193	0.00034873225
50	1.0720938	0.10839323	0.010447983	0.0008718306

We expect these results can help data depositors budget their global epsilon. For example, if they have a dataset with 16 statistics and sample size ~10^6 (as the PUMS), we could recommend for example 0.05, as expressed in the last bullet point.

EPS=0.01	**10^3**	**10^4**	**10^5**	**10^6**
5	0.9804047778	0.09804047778	0.009804047778	0.009804047778
10	1.960809667	0.1960809667	0.01960809667	0.01960809667
20	3.921619222	0.3921619222	0.03921619222	0.03921619222
50	9.804047778	0.9804047778	0.09804047778	0.09804047778
EPS=0.05	**10^3**	**10^4**	**10^5**	**10^6**
5	0.1960809667	0.01960809667	0.001960809667	0.0001960809667
10	0.3921619222	0.03921619222	0.003921619222	0.0003921619222
20	0.7843238333	0.07843238333	0.007843238333	0.0007843238333
50	1.960809667	0.1960809667	0.01960809667	0.001960809667
EPS=0.1	**10^3**	**10^4**	**10^5**	**10^6**
5	0.09804047778	0.009804047778	0.0009804047778	0.00009804047778
10	0.1960809667	0.01960809667	0.001960809667	0.0001960809667
20	0.3921619222	0.03921619222	0.003921619222	0.0003921619222
50	0.9804047778	0.09804047778	0.009804047778	0.0009804047778
EPS=0.5	**10^3**	**10^4**	**10^5**	**10^6**
5	0.01960809667	0.001960809667	0.0001960809667	0.00001960809667
10	0.03921619222	0.003921619222	0.0003921619222	0.00003921619222
20	0.07843238333	0.007843238333	0.0007843238333	0.00007843238333
50	0.1960809667	0.01960809667	0.001960809667	0.0001960809667
EPS=1	**10^3**	**10^4**	**10^5**	**10^6**

5	0.009804047778	0.0009804047778	0.00009804047778	0.000009804047778
10	0.01960809667	0.001960809667	0.0001960809667	0.00001960809667
20	0.03921619222	0.003921619222	0.0003921619222	0.00003921619222
50	0.09804047778	0.009804047778	0.0009804047778	0.00009804047778

Finally, we also provide a table **not upper bounded by the CDFs accuracy** (CDFs will not meet the 10% mean absolute error), but by the Histograms accuracy (which bounds hold for Means also).

Number of variables

	5	10	20	50
10^3	0.05*	0.1*	0.5	0.5*
10^4	0.01	0.01*	0.05	0.05
10^5	0.01	0.01	0.01	0.01
10^6	0.01	0.01	0.01	0.01

*20% mean absolute error normalized by the range, all the others are 10%

Finally, this table was created taking the mean of the accuracy of statistics released with the same type. It is biased by the Histograms and CDFs number of bins on each release.

Comments

- Global epsilon has a linear relationships with accuracy.
- Number of variables has also a linear relationship with accuracy
    - The accuracy is given in terms of local epsilon. This relationship seems to be linear for all routines
    - The local epsilon is given by a linear computation of the global epsilon (evenly splitted, it is the default of the system and we assume no accuracy was fixed by the user). So, the accuracy is given by the local epsilon that is linear.

Possible future paths

Another good parameter that could be analyzed is the **number of binary variables**. One might find a way to integrate it with the other variations: number of variables, sample size and global epsilon.

Marcelo Novaes, Summer Intern 2016

PSI experiments - (1) When decide for a post-processing? (2) When encoding a new granularity to make DP better?

Post-processing Analysis and Granularity/Number of bins analysis

Introduction

The objective of this document is exploit patterns among a chosen epsilon, number of variables, sample size and when decide for save budget, generating Means and Histograms from the CDF. The secondary goal is help data depositors during the privacy budget.

EXAMPLE 1: PUMS CALIFORNIA

We compare two releasing approaches for all numeric non-categorical variables:
Approach 1: Request only CDF and generates Means and Histograms.
Approach 2: Request CDF, Means and Histograms as requested statistics.

Procedure

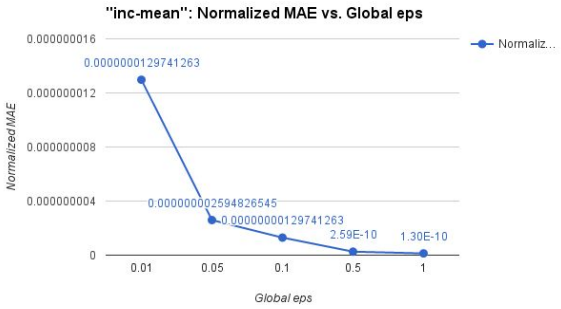
- Each metric point is a metric average for **100 iterations**

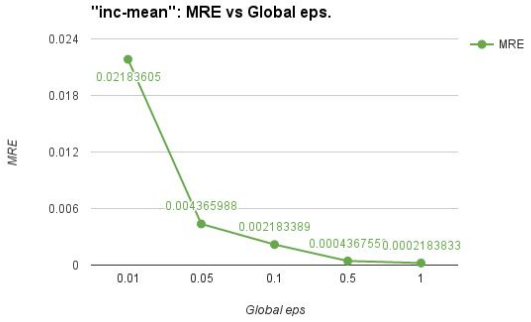
Remark

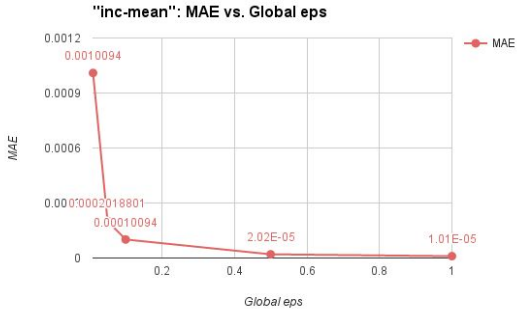
Analyzing the results, this test shows inconsistencies and therefore the data must be **revised**. Some of the inconsistencies are increasing error for epsilon 0.5 and huge difference between mean absolute error normalized values on different variables.

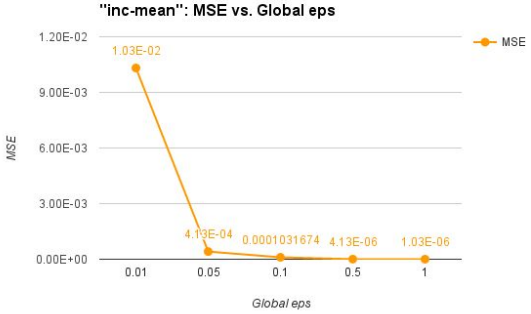
1) ORIGINAL EPSILON USING POST-PROCESSING

Mean Income

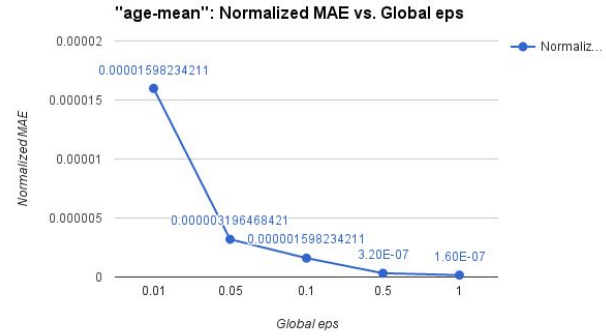


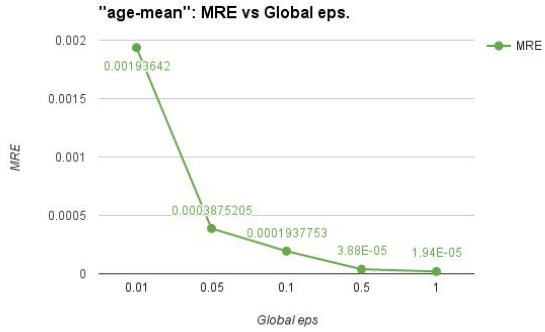


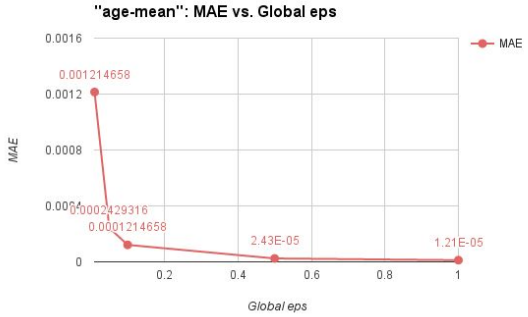


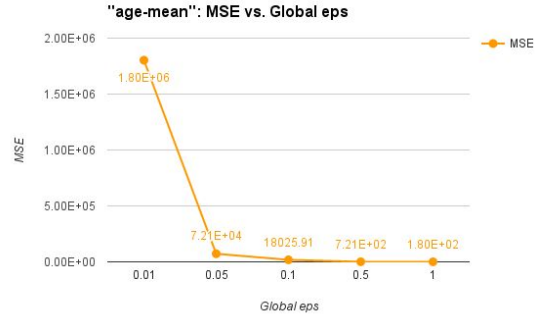


Mean Age

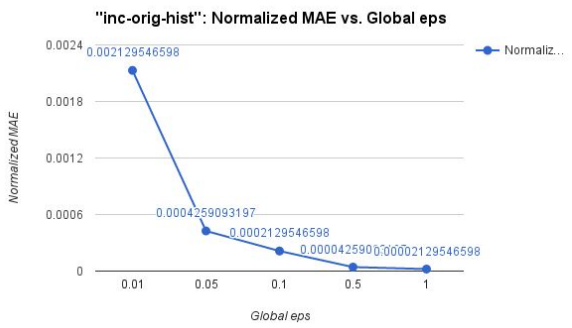


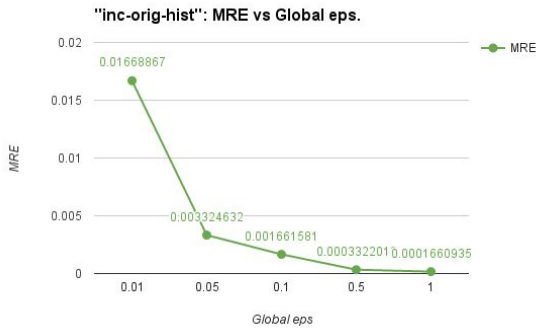


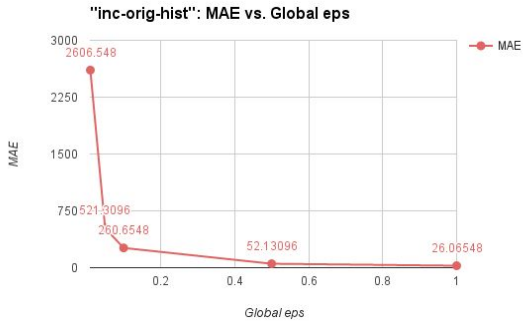


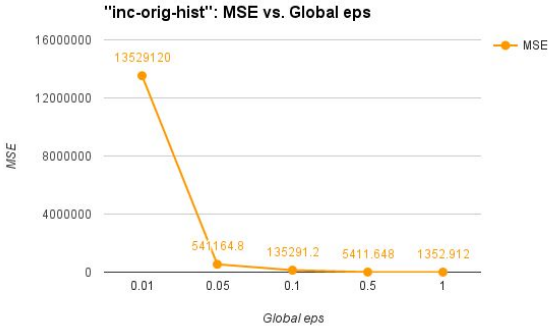


Histogram Income

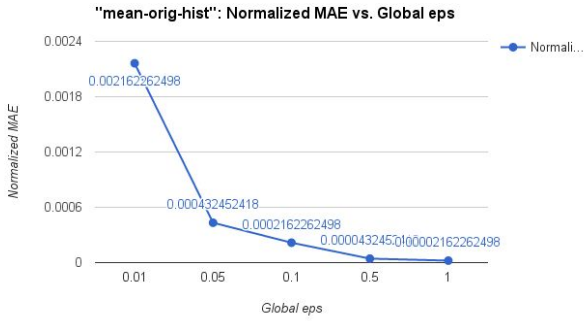


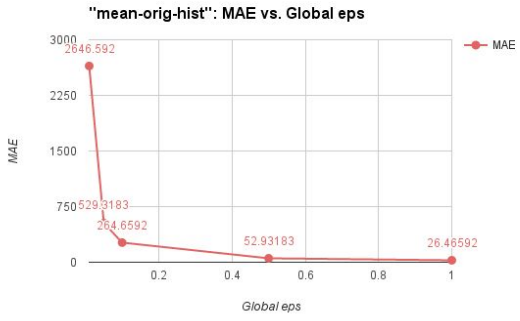


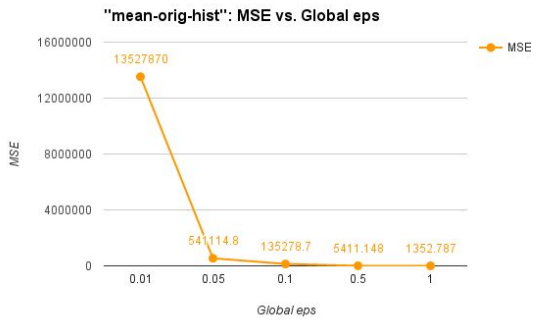


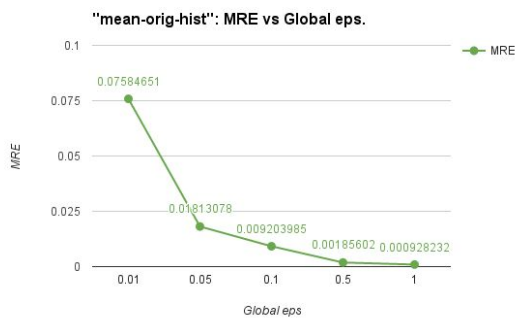


Mean Income



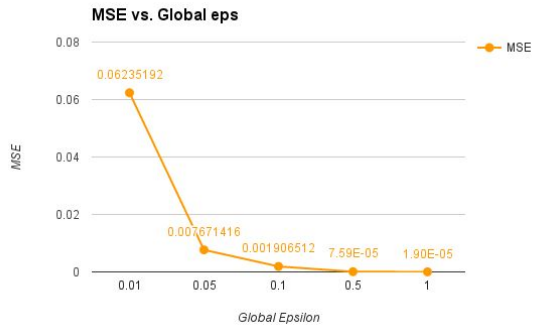


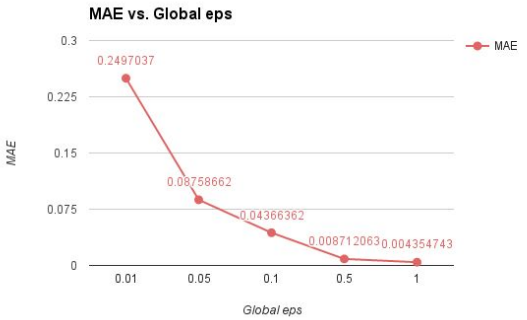


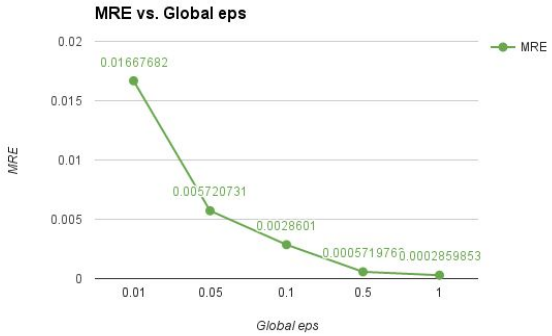


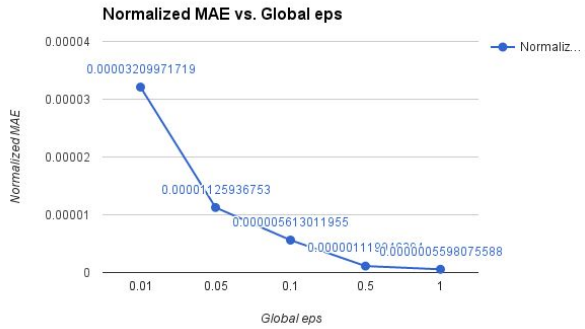
2) SAVING EPSILON USING POST-PROCESSING

Histogram (from CDF) Income

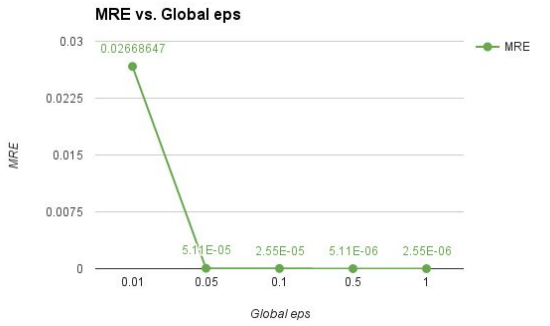


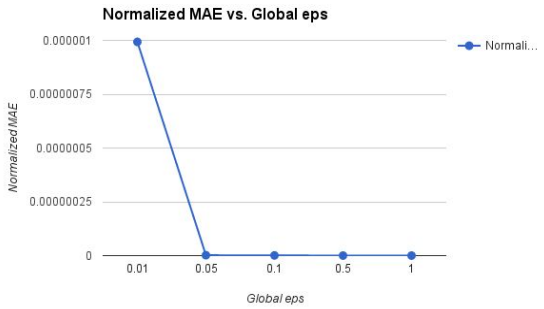


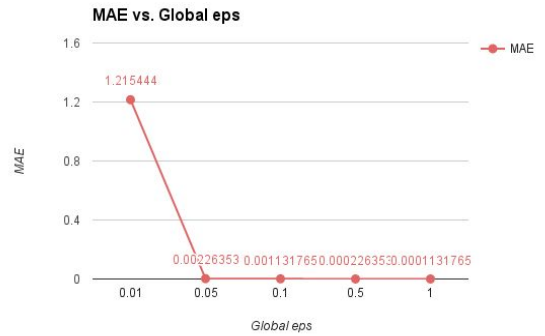


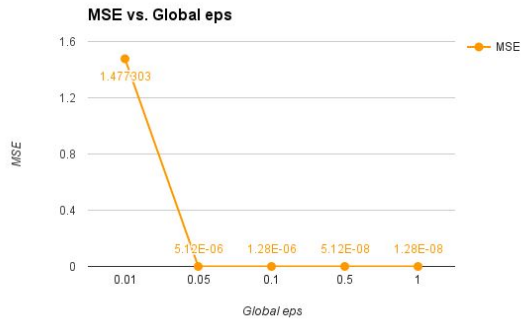


Mean (from CDF) Income

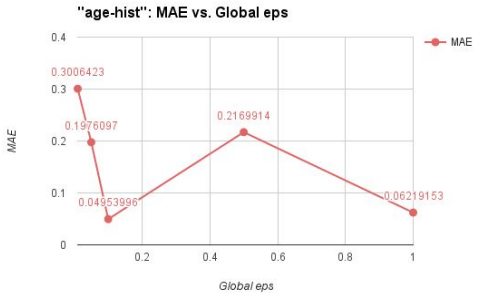


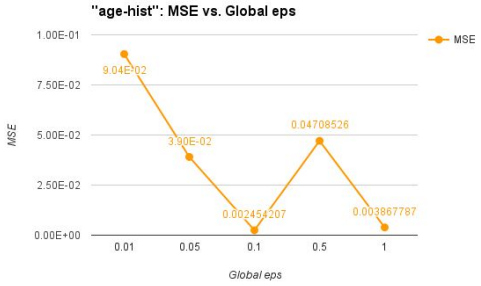


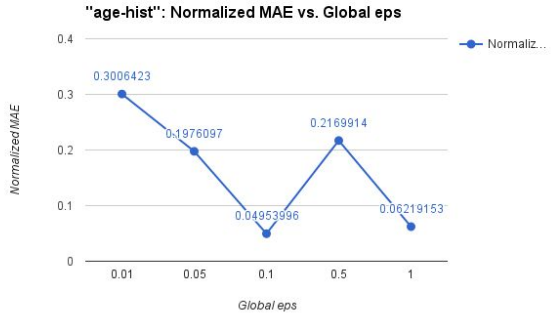


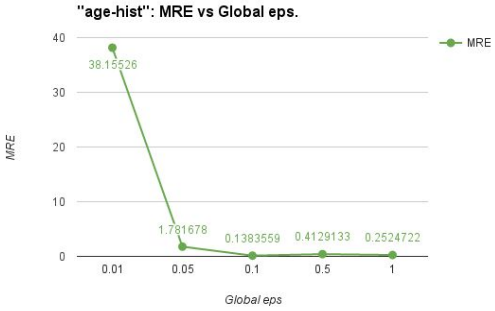


Histogram (from CDF) Age (bug at epsilon=0.5?)

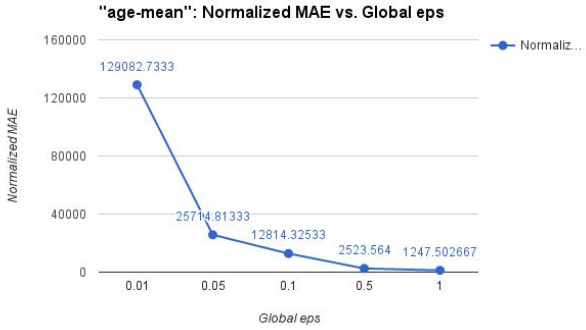


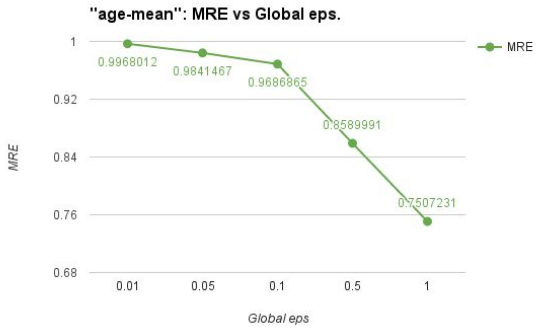


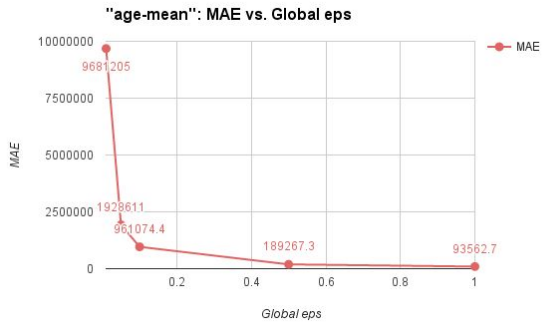


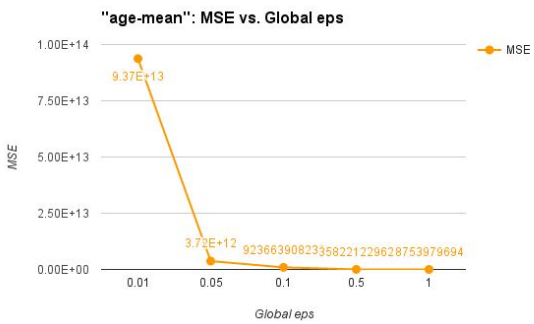


Mean (from CDF) Age









Consequence on other released statistics

- Increase of a value “x” on each local statistic, where “x” depend of the global epsilon (look the table)
- By now the post-processing can reduce at max 2 statistics per variable (Mean and Histogram). Assuming the user did not have any accuracy preferences, the system that would split the global epsilon evenly between the 13 statistics (on PUMS), splits between 9 (on PUMS).

Number of statistics (Original release)	Number of Statistics (Using post-processing)
13	9

Global eps

	0.01	0.05	0.1	0.5	1.0
Original local epsilon	0.00077	0.00385	0.00769	0.03846	0.07692
New local epsilon	0.00111	0.00555	0.01111	0.05555	0.11111
Gain (in terms of eps)	0.00034	0.0017	0.00342	0.01709	0.03419

- How much was the **gain in terms of accuracy** depend of the statistic method (See experiment about **fixing accuracy for different statistics**). In the case of PUMS, all other released statistics were **Histograms.**

EXAMPLE 2: SOCIAL PRESSURE

Variables considered to be not categorical and ordered (CDF post-processing can be used):

- Yob: year of birth.
- Cluster: identification of records household values.

Number of statistics (Original release)	Number of Statistics (Using post-processing)
20	16

	0.01	0.05	0.1	0.5	1.0
Original local epsilon	0.0005	0.0025	0.005	0.025	0.05

New local epsilon	0.000625	0.003125	0.00625	0.03125	0.0625
Gain (in terms of eps)	0.000125	0.000625	0.00125	0.00625	0.0125

EXAMPLE 3: COMP VOTING

Variables considered to be not categorical and ordered (CDF post-processing can be used):

- Mob: month of birth
- Dob: day of birth
- Fv

Number of statistics (Original release)	Number of Statistics (Using post-processing)
16	10

	0.01	0.05	0.1	0.5	1.0
Original local epsilon	0.000625	0.003125	0.00625	0.03125	0.0625
New local epsilon	0.001	0.005	0.01	0.05	0.1
Gain (in terms of eps)	0.000375	0.001875	0.00375	0.01875	0.0375

OTHER DATASETS ASSUMPTIONS EXAMPLE
(Using quick-start statistics or provided cookbook)

In green, it is highlighted the maximum gain on each epsilon in the datasets **of the replication group (collected by Ana, Clara and Grace)** using post-processing.

	Stats non-post	Stats post-proc	0.01	0.05	0.1	0.5	1.0
PUMS	13	9	0.00034	0.0017	0.00342	0.01709	0.03419
SOCIALP	20	16	0.000125	0.000625	0.00125	0.00625	0.0125
COMPV	16	10	0.000375	0.001875	0.00375	0.01875	0.0375

Huff and Tingley (2015)	12	10	0.0001667	0.000833	0.001667	0.00833	0.01667

For most of the other datasets collected, marked as “Survey based” by the replication group, we assume there is only one variable possible to be done post-processing (such as weights). So, the epsilon saved would be:

Nyhan et al. (2012)	40	38	0.00001315789474	0.00006578947368	0.0001315789474	0.0006578947368	0.001315789474
Panagopoulos	22	20	0.00004545454545	0.0002272727273	0.0004545454545	0.002272727273	0.004545454545
Mousseau (2011)	36	34	0.00001633986928	0.00008169934641	0.0001633986928	0.0008169934641	0.001633986928
Hiromi Ono	25	23	0.00003478260870	0.00017391304345	0.00034782608700	0.0017391304345	0.003478260870
Pew	210	208	0.00000045787545 79	0.00000228937728 9	0.00000457875457 9	0.0000228937728 9	0.00004578754579
Broockman	47	45	0.00000945626477 5	0.00004728132388	0.00009456264775	0.0004728132388	0.00094562647 75
Kinder and Ryan (2016)	4320	4318	0.00000000107216 99	0.00000000536084 9502	0.00000001072169 9	0.0000000536084 9502	0.00000010721699
Huff and Tingley (2015)	19	17	0.00006191950464	0.00030959752332	0.00061919504640	0.0030959752332	0.006191950464
Pew	270	268	0.00000027639579 88	0.00000138197899 4	0.00000276395798 8	0.0000138197899 4	0.00002763957988
Lopez	12	10	0.00016666666667	0.00083333333333	0.00166666666667	0.00833333333333	0.01666666667
Sanchez	12	10	0.00016666666667	0.00083333333333	0.00166666666667	0.00833333333333	0.01666666667
Hanmer et al	41	39	0.00001250781739	0.00006253908693	0.00012507817390	0.0006253908693	0.001250781739
Pew	140	138	0.00000103519666 87	0.00000517598343 7	0.00001035196687	0.0000517598343 7	0.00010351966 87
Fraga	27	25	0.00002962962963	0.00014814814814 81	0.00029629629630	0.0014814814814 81	0.002962962963
Enchautegi	32	30	0.00002083333333	0.00010416666667	0.00020833333333	0.0010416666667	0.002083333333

Winters et al	58	56	0.000006157635468	0.00003078817734	0.000061576354 68	0.00030788177 34	0.000615763 5468
Caumont	16	14	0.00008928 571429	0.000446428571 4	0.0008928571 429	0.00446428571 4	0.008928571 429
Fuse	15	13	0.00010256 41026	0.000512820512 8	0.0010256410 26	0.00512820512 8	0.010256410 26
Plutzer et al. (2016)	148	146	0.00000092 55831174	0.000004627915 587	0.0000092558 31174	0.00004627915 587	0.000092558 31174
Goldin	29	27	0.00002554 278416	0.000127713920 8	0.0002554278 416	0.00127713920 8	0.002554278 416
Plutzer and Berkman (2011)	90	88	0.00000252 5252525	0.000012626262 63	0.0000252525 2525	0.00012626262 63	0.000252525 2525

CONCLUSIONS & OTHER RESEARCH QUESTIONS

Experiments related with this

1) This experiment might lead us to another question: “What are the numeric data properties which the desired post-processed statistic is as good as the requested statistic”. It might vary for the desired post-processing method and for the measure of goodness or usefulness.

2) “What is the fixed accuracy for CDF I need to have an accurate specific post-processed routine”**.** Suppose the question above want to be answered by an user. The general way was define a global accuracy which would always satisfy the “accurate” property e.g. a defined accuracy that meets the MRE < 10%. We have seen there is no a satisfactory function to it, it depends of the analysis in the data.

So, instead of it, we define the steps the data depositor should analyze to answer the question. For a specific use, the user will:

1. Define the accuracy he/she needs for the post-processed statistic (according to one or more metrics defined by the system).
2. See the how his/her post-processed statistic accuracy changes in terms of the CDF accuracy (What is the decreasing function behaviour, it will vary depending of the CDF configurations such as number of bins) - table*
3. Fix this accuracy in the CDF.

When change the granularity and number of bins on Histograms/CDF in order to use DP

Motivation

Data depositors might face a situation which they have to **shrink the range** of their data or **shrink number of bins** of an Histogram or CDF in order to avoid inaccurate differentially private results. These two scenarios are not disjoint, changing granularity we might shrink both: range and number of bins. PSI provides granularity changing. A simple example would be after getting an estimative that differentially private histograms would be too noise (age with granularity 1 year) to a possible less noisy encoding (age with granularity 10 years). A 10 years range for each bin would shrink the range, so probably reducing a noise based in the sensitivity. Also, reducing the number of bins, it reduces the error on post-processing operations that depends on the full histogram. Different from the experiments of Caper Gooden, we are not estimating different ranges for the same data, but actually encoding a new granularity in the data. An important remark is that changing the granularity or number of bins cannot always be used. Considering for example, we are studying specific ranges of ages 1-21, 21-65, 65+, a different age binarization can mess up the social science analysis. A focus on this variation was partly motivated by discussions with Clara Wang, a 2016 summer intern.

EXAMPLE 1: PUMS CALIFORNIA

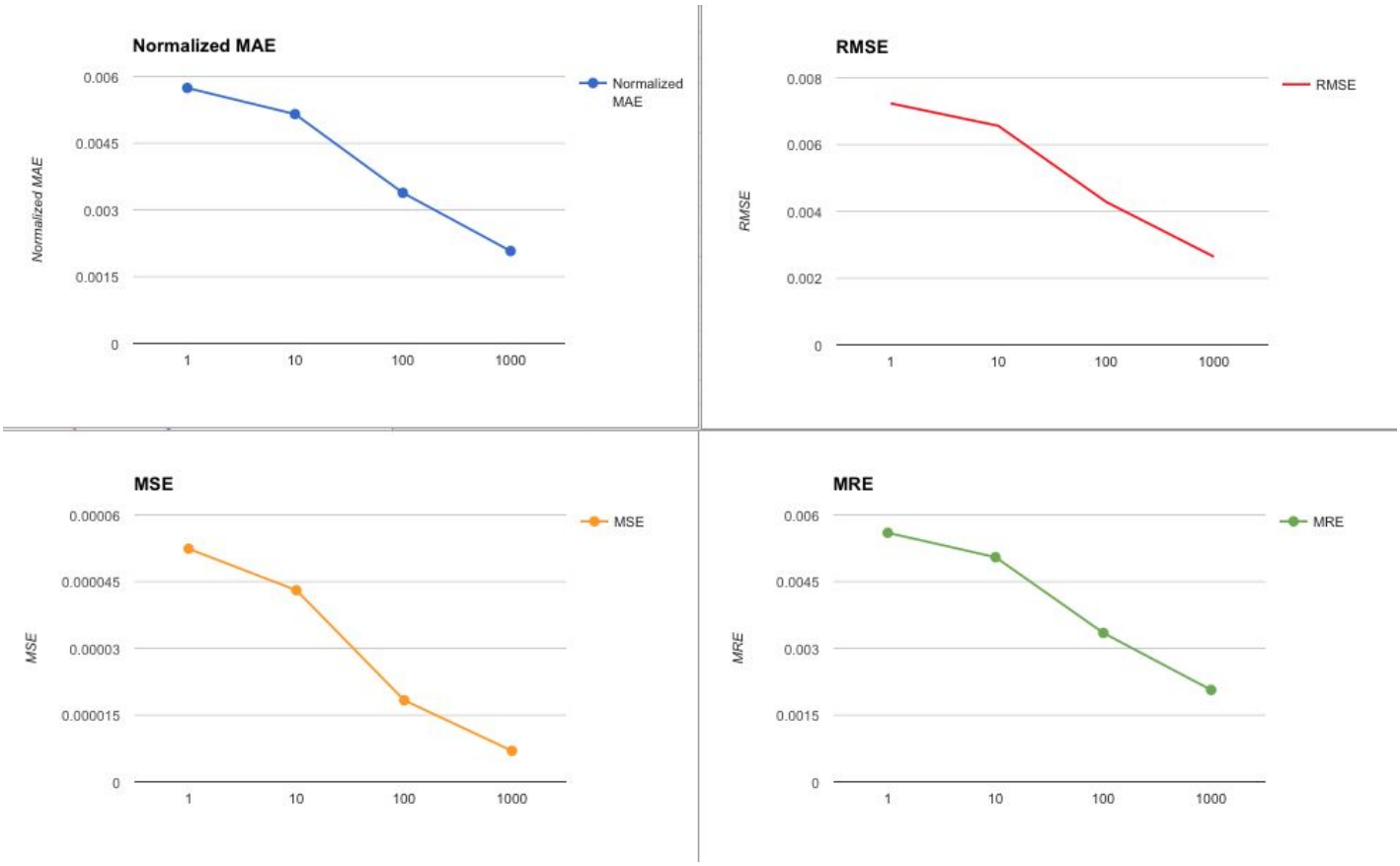
Actual release example

Income:

Granularity	1	10	100	1000
Number of bins	777801	77781	7779	778
Normalized MAE	0.005743614	0.00515572	0.003386653	0.002078574
RMSE	0.07482489559	0.0710684107	0.05786804818	0.04546308393
MRE	0.005598765	0.005050719	0.003348711	0.002066892
MSE	5.24E-05	4.31E-05	1.84E-05	6.99E-06

Range is [0,Number of bins -1]
Accuracy metrics results as **average of 100 iterations**

After budget, the local epsilon was: ~ 0.01 (actual: 0.01111111)



Future possible experiment

For the usual epsilon space, consider granularities such as will reduce the number of bins to a specific percent value.

	Reduces 50% bins	Reduces 25% bins	Reduces 15% bins	Reduces 5% bins
Different variables from different datasets	acc	acc	acc	acc

PSI experiments - Efficiency

Each differentially private routine, full releases and composition theorem analysis

SUMMARY 1: STATISTICS RELEASE (PER STATISTIC)

Procedure/Metric

1. Call the statistic **20 times**
2. Get the **elapsed time** of each call
3. Remove the **outliers**
4. Get the **median**

Results in **sec**.

Means

Sample size	< 200K	200K	2M	1B
	Negligible	0.003	0.041	0.24
Result		**3ms**	**41 ms**	**240 ms**

Histograms

Bins/Sample size	20K	200K	2M	1B
2 (Binary)	0.002	0.016	0.237	1.0825
20	0.002	0.016	0.2605	1.1415
50	0.002	0.018	0.2465	1.114
250	0.002	0.021	0.269	1.2335
1K	0.002	0.019	0.283	1.175
Result	**2 ms**	**18ms**	**259 ms**	**1.15 sec**

CDFs

Bins/Sample size	20K	200K	2M	1B
2 (Binary)	0.011	0.013	0.035	0.136

20	0.011	0.013	0.035	0.136
50	0.0105	0.013	0.035	0.136
250	0.011	0.0124	0.035	0.136
1K	0.011	0.013	0.035	0.136
Result	**11 ms**	**13ms**	**35 ms**	**130 ms**

Machine used

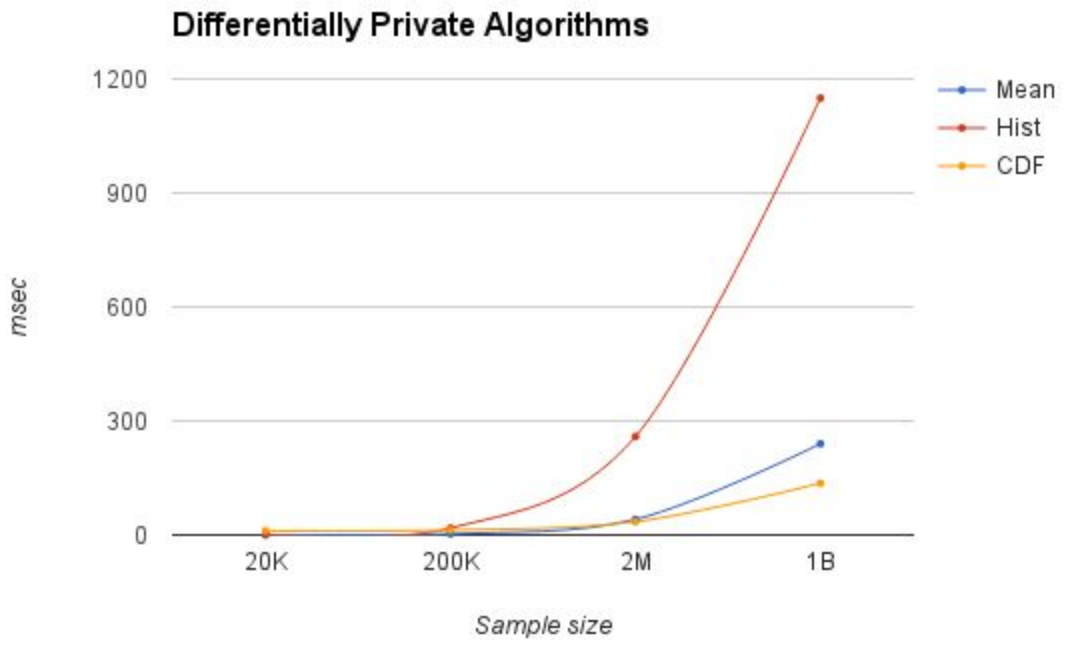
- iMac (21.5-inch, Late 2013), OSX Yosemite, 10.10.5
  - Processor: 2.7 GHz Intel Core i5
  - Memory: 8 GB 1600 MHz DDR3

Assumptions

- Already binned data for CDF
- Data categorical for Histogram

Conclusions

- Number of bins does not affect the running time significantly
- Histogram takes more time, but we expect it faster (~3x), after optimized



SUMMARY 2: STATISTICS RELEASE (FULL RELEASES)

		Dataset properties			
	Released statistics	Number of variables	Number of **numeric** variables used	Sample size	Elapsed time
PUMS California	13	9	2	1223992	2.603 sec
Pew Global	21	21	0	48643	0.060 sec
Ballot Secrecy	31	31	0	894791	5.183 sec
Social Pressure	20	16	2	344084	1.739 sec
Compulsory Voting	16	10	3	1797225	4.522 sec

Considerations regarding the usual parameter space

By the usual parameter space I'm considering the one collected by the replication group. It is just an estimation. Considering some properties below about a subset of datasets (at the time I collected, they were all of them which were described as containing univariate statistics: 30) collected by Clara Wang, Grace Rehaut and Ana Oaxaca. The Compulsory voting dataset is the 2nd largest dataset in terms of Sample Size and works really well. Considering the parameter space of the, PSI would perform well for all of them also. For the worst case in the list, it depends of the number of statistics which would be chosen from the variables, but probably it would take between 15-30 sec, what is really good given the dimensions of the dataset.

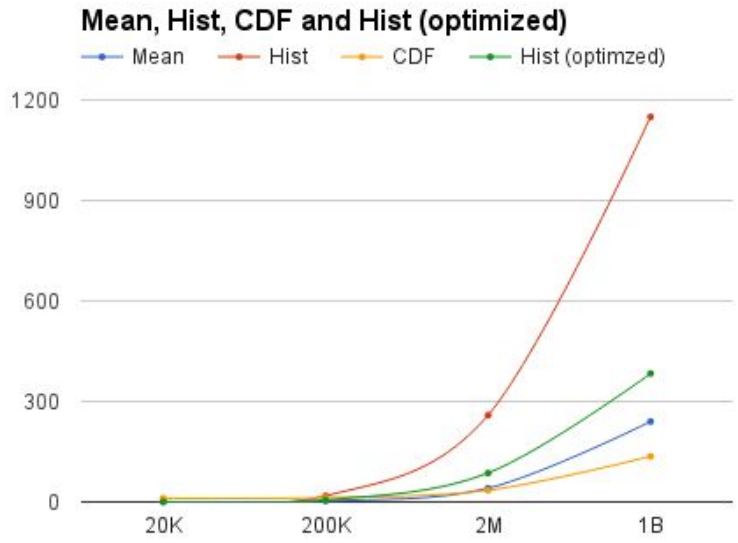
	Sample size	Number of variables
Mean (no outliers)	27,897	35.9
Median	4,000	29
Worst case in the list	80,000	800

Very bad case (#variables) in the list	1,800,000	21
Very bad case (sample size) in the list	4240	4318

OPTIMIZATION ON HISTOGRAM

Histograms: Use hist() instead of match()

- Map categorical data to numeric numbers
  - Call hist()
  - Map back in the correct order
- Handle missing data
  - The original match() assumed all data was categorical, so NA and NaN's were categories. It already handled missing data correctly.
  - preProcHist() function created to deal with it: it counts NA and NaN, add them to the final computation as additional bins.
- Use hist() breaks to specify the bins
  - Remeber that hist() has breaks with default (begin,end] range. Suppose we want to create 10 bins, from 1 to 10, in the traditional approach using match(), we would pass 1:10 command. Now, we would pass 0:10, so the bins would be (0,1], (1,2], (2,3] … (9,10]. So we would have the same bins 1 to 10.



EXPERIMENT: APPROXIMATE COMPOSITION (BUDGET INTERFACE)

Test R composition routine

Test: composition routine takes **60 ms** for a small error of 0.01, while a simples O(n^3) **50ms**.

approxComp with err=0.1	approxComp with err=0.01	Simple O(n^3)
1 ms	60ms	50 ms

The update still call it ~ 30 times, total 2 seconds. What would be acceptable, even though the desired would be 0.1-1sec as commented in the Design Document.

Composition	Simplest O(n^3) algorithms

source("./dpmodules/Jack/CompositionTheorems.R")
eps_list <- runif(n = 50,min = 0.01,max=0.03)
delta_global <- 2e-20
del_list <- rep(delta_global/100, times = 50)
params <- cbind(eps_list, del_list)
approxComp(params,delta_global)
	# simple 3-loop with 4 different constants
	x3 <- function(size) {
	a <- b <- c <- d<- 0
	for (i in 1:size){
	a <- a + 1
	for (j in 1:size){
	b <- b + 1
	for (k in 1:size) {
	c <- c + 1
	d <- d + 1
	} } } }

Interface tests

The interface was tested in three computers. It follows their basic configurations, in the firefox and chrome web browsers:

- MacBook Pro
    - Processor: 2.5GHz Intel Core i7-4870HQ
    - Memory: 16GB DDR3 SDRAM 1600MHz
    - Results
        - No delay at all.
        - More than 50 variables were added, fixing accuracy (forcing optimal composition)
        - A video with the test can be found in the google drive.

2. MacBook Air
    - Processor: i5
    - Memory: 4 GB RAM
    - Results
        - There were random delays. Sometimes it was pretty fast. Sometimes it had delay. Not closely related with the number of variables. We could not figure it out what was causing it. It seems it was related with refreshing the page and new tests.

Possible optimizations

it follows some possible optimizations, some of them reported by from during a meeting with Jack Murtagh.

- Do not update every time to the user the new epsilons when a new variable comes in.
    - Update for each fixed number of variable additions.
    - Put a javascript timer.
    - Ask for the data depositor how many variables we can expect to be requested, so we can plan better ways to update it.
- Using a faster method (achieving an accuracy lower than the expected) during budget. Then, when the user press the “submit”, we call the actual approx. composition and compute the new global epsilon of all the variables before use the DP routines.
- Maybe in the future it could be a good approach, use the fact that we are requesting different kind of statistics to do the composition. For example, assume already that we just have univariate statistics: mean, histograms and csfs. Then, try to improve the approximate composition results.

Helpful comments about the approximate composition

- If the epsilons are all splitted evenly (by now it means the user did not fixed any accuracy), it calls a homogenous method which is O(1), the whole composition considering the update is O(k), where k is the number of variables.
- Find the actual optimal composition is a non-polynomial problem, similar to find the number of solutions for knapsack problem. The approximate composition is a O(k^3), where k is the number of variables.
- The algorithm can be found at http://arxiv.org/pdf/1507.03113.pdf, section 5.

Future possible measures

- Once implemented the new histogram method, compute
    - Bin the data
    - Transform data from numeric to character when necessary

Review - Marcelo Novaes, Summer intern 2016

Attacks on Differentially Private systems
Summary of attacks and PSI relationship

Introduction

- RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response (RAPPOR) is a Google technology for crowdsourcing statistics from end-user client software [2]. As multiple statistics are collected multiple-times over the same users, it uses two layers of Randomized Response. The first one, called Permanent Randomized response is responsible for the insertion of the noisy in the data. The second one, called Instantâneos Randomized Response is responsible for protection against longitudinal attacks. The paper is from 2014.

- GUPT: The Privacy Preserving Data Analysis Made Easy is system from the U.C. Berkeley that guarantees differential privacy to programs not developed with privacy in mind. The paper is from 2012. An important fact is that they consider aging in their system degrading privacy of a data over time. A good analysis regarding side-channel attacks is done in the paper based in [1].

- PINQ: The Privacy Integrated Queries paper published by the Microsoft Research worker in 2009. It describes a system which the data analyst should “write computer programs to distill the data to manageable aggregates on which they base further analysis.” Even though it had problems with side-channel attacks later in the time the paper was published, this is one of the main precursors for the new differentially private systems.

In some way, GUPT is the differential private system more close PSI. While PINQ and RAPPOR are also differentially private system, they have more distinct characteristics. In the one side, PINQ provides differential privacy through a programming API and that assume trusted programs from analysts. In the other side, RAPPOR is focused in the intense collection of data from the same users and techniques to guarantee privacy on it.

Side-channel attacks

In case we allow the analyst to run their own computations over the data (e.g. compositions of predefined transformations) and then add sufficient noise, three main side-channel attacks are found in the literature: privacy budget, state and timing attacks.

Description according to [1]:

Privacy budget attacks	When encountering a specific record, the analyst program could inadvertently or proposital exhaust the privacy budget.
Protection against state attack	When encountering a specific record, the analyst program could change some internal state. Later he could try to read the internal state and see if the record was in or out the output received.
Protection against timing attack	Encountering a specific record, the analyst program could run in an infinite loop.

Based in the Section 7.2.2 of [2], the PSI were introduced in the comparison.

	GUPT	PINQ	Airavat	PSI
Protection against privacy budget attack	Yes	No	Yes	Yes*
Protection against state attack	Yes	No	No	Yes*
Protection against timing attack	Yes	No	No	Yes*

On PSI, these attacks could be applied in the Query interface. However, the computations that the analyst are able to ask for are **predefined by the system**. So **the system is protect against all of them**.

*It clearly limits the power of the system, what will be analyzed in another moment.

No side-channel attacks:

Global budget attack	An analyst could try to exhaust all the budget of the database. It could be done trying to: ● Making additional queries in his session ● Creating faking accounts to get more budget ● Colliding data with other real users

On PSI, the global budget attack is related to the Query Interface. The protection against privacy budget attack is done by ensuring that each analyst has a fixed budget and that they do not collide information. This last part must be more carefully analyzed once publish a result is a way to collide information. As the prototype is not yet in production, it was not analyzed.

Longitudinal Attacks:

Longitudinal attacks are the focus in the [2] paper. The main concern is protect against the

	RAPPOR	PSI
Protection against window (longitudinal) attacks	Yes	Yes*

Sufficient noise is suppose to be in the Differential Privacy statistics released by the PSI such that it does not reveal any specific information. However, when collecting data multiple times over the same individuals.

*Experiments are necessary to show it.

Other questions regarding security:

Aging: It is important decide if “aging”, as a database property, will be taking into account. If it will be considered, how to evaluate it (maybe data in the past not considered sensitive can become sensitive?). PINQ, for example, takes it explicitly into account.

Privacy techniques used

	Laplace Noise	Randomized Response
RAPPOR		x
PINQ	x	
GUPT	x	
PSI	x	

Bibliography

[1] Haeberlen, Andreas, Benjamin C. Pierce, and Arjun Narayan. "Differential Privacy Under Fire." *USENIX Security Symposium*. 2011.

[2] Mohan, Prashanth, et al. "GUPT: privacy preserving data analysis made easy." *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. ACM, 2012.

[3] Erlingsson, Úlfar, Vasyl Pihur, and Aleksandra Korolova. "Rappor: Randomized aggregatable privacy-preserving ordinal response." *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*. ACM, 2014.

[4] McSherry, Frank D. "Privacy integrated queries: an extensible platform for privacy-preserving data analysis." *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*. ACM, 2009.

Review - Marcelo Novaes, Summer Intern 2016

Experiments on GUPT, PINQ and RAPPOR

Database description used in the experiments:

	GUPT	RAPPOR	PINQ
Database	Life Science	Distributed information of chrome users	Web search logs
Description	Top 10 principal components of chemical/biological compounds.	They define a set of properties that a user has and encoded them to strings (e.g. “This user has the specific process X running on his computer”	Contains users IP address, localization, and query text.
Database Size	~ 30,000 entries ● First set of experiments : 26,733 entries ● Second: 32,561	~14,000,000	~648,615 (inferred by the Table 1 in section 4.1)

RAPPOR:

- Utility:Statistical inference between processes (e.g. relationship between a non-malicious and a malicious process), detect the frequency of processes
- Statistical methods: Learning underlying distribution; estimated frequency and p-values.

PINQ:

- Utility: Frequency of terms searched by distinct users; Distribution of latitude-longitude coordinates of a specific term (used same size blocks; aggregate analysis by dense regions spare regions)
- Statistical methods supported: K-means clustering, perceptron classification, contingency table measurement, and association rule mining.

GUPT:

- Utility: statistical inference, classify which compostos were cancerings
- Statistical methods used: clustering (using k-means) and logistic regression.

[1] Mohan, Prashanth, et al. "GUPT: privacy preserving data analysis made easy." *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. ACM, 2012.

[2] Erlingsson, Úlfar, Vasyl Pihur, and Aleksandra Korolova. "Rappor: Randomized aggregatable privacy-preserving ordinal response." *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*. ACM, 2014.

[3] McSherry, Frank D. "Privacy integrated queries: an extensible platform for privacy-preserving data analysis." *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*. ACM, 2009.