Testing Private Regression Algorithms

Andreea Antuca, Haoqing Wang

Mentors: James Honaker, Vishesh Karwa, Or Sheffet

Contents

1 Introduction

1.1 Project Goals

Linear regression models allow researchers to investigate multivariate relationships in data and are at the base of more complex regression analysis. Incorporating a tool for linear regression in the Privacy Tools project will greatly benefit researchers who are examining potential datasets for their research. This project involves the design, implementation, and testing of multiple algorithms for linear regression. The goal of the project is to develop tools that enable us to run regressions as well as statistical tests on the differentially private regression results. In addition, we are interested in finding theoretical guarantees on the results of these tests and a thorough empirical analysis of the techniques available.

A number of possible mechanisms hold the promise of providing useful regression-style results for researchers investigating private data. Judging the practical performance of these mechanisms involves understanding where and why these mechanisms succeed or fail.

1.2 Summary of Tests

The main goal of this summer was to create a testing framework to test existing mechanisms and potential new ones. The framework constructs Monte Carlo datasets across ranges of parameters and runs our differentially private algorithms on the datasets. Then, statistical tests may be performed on the output of these algorithms to examine the utility of the algorithms. Our testing framework can help compare the inferences made based on these noisy results with the inferences that would have been made had the original dataset been studied instead, using traditional algorithms for regression that do not attempt to preserve privacy.

This summer, we focused on research and implementation of existing algorithms, attempting to gain a greater understanding of these algorithms. Given a dataset sampled from a bounded d-dimensional space, we ran differentially-private linear regressions and examined their inferential properties. Our focus is on the approximation of the t-values used for a simple acceptance or rejection of the null hypothesis, "the beta coefficient is not significantly different than zero," which is one of the first things a researcher examines when performing a regression. Typically, we will examine results at the 5% significance level.

Given this, we define a correct t-value to be a t-value with the same sign as the population correlation and with an absolute value above 1.96. For a correlation of zero, a correct t-value is one with an absolute value is below 1.96. An incorrect t-value has been defined in this report as a t-value of opposite sign as the population correlation and with an absolute value greater than 1.96. For the correlation of zero, an inccorect t-value has been defined as an absolute t-value greater than 1.96. The four algorithms that we tested are all based on a differentially private approximation of the second-moment matrix of the data, from which we can obtain β coefficients and standard errors. This allowed us to experiment with each algorithms ability to give the correct statistical inference. We studied the performance of such algorithms and analyzed their error on multiple axes:

1. the distance from the "true" correlation coefficient,
2. the magnitude of the standard error, as compared to the magnitude of the standard error for the noiseless regression,
3. difference in t-values,
4. probability of outputting the correct statistical inference.

1.3 Summary of Results

As theoretically established, the algorithms converge to the true means of the correlation coefficients. However, some of the functionality of estimators is lost. In ordinary least-squares linear regression, let f be the computation that gives $beta_i$ through finding X^TX, where X is the dataset, and let g be the computation that outputs σ_i, where σ_i is the standard error of β_i. Let $\tilde{f}$ and $\tilde{g}$ be the same two computations done with differential privacy. Then, we see empirically that $E[\tilde{f}] = E[f]$, and $E[\tilde{g}] = E[g]$, that they are both good point estimates of the sample. However, in ordinary least-squares, we have that there is the key inferential relationship between β_i and its standard error, but this relationship appears to break when we attempt to naively use the standard error calculated from the differentially private X^TX. The variance in $\tilde{f}$ is considerably higher than what is predicted by our naive version of$\tilde{g}$, and it requires additional work to improve our computation of $\tilde{g}$.

2 Testing Framework and Functions

This section will provide a brief overview of the testing framework written by Haoqing and the plotting function written by Andreea.

2.1 Framework Description

The testing framework was created with the above questions in mind. The framework itself accepts a matrix for input and outputs a matrix giving population βs, sample βs, sample standard errors, sample t-values, differentially private βs, differentially private standard errors, and differentially private t-values. The framework does this by running differentially private algorithms over a synthetic dataset generated as specified many times. The following is the list of columns for the input matrix; any of these columns could be varied and tested using the framework. Each row would have one entry in each of these sections. The simulation function is called `simulFixedD`, named as the dimensionality of the dataset must be fixed throughout the tests done in one run of the function.

1. `i`: indicator showing whether the current row of the matrix should instruct the framework to create a new dataset (if `i` $= 2$) or keep using the previous dataset if one exists (if `i` $= 1$)
2. `T`: number of times to run differentially private algorithm based on the instructions given in the current row
3. `n`: number of observations, or rows, in synthetic dataset
4. `d`: number of variables, or columns, in synthetic dataset
5. `restrict`: indicator of whether we should truncate (`restrict` $= 1$) or censor (`restrict` $= 2$) values outside of the permissible ranges for each column of the dataset
6. `epsilon`: value of ϵ, the privacy budget in differential privacy
7. `delta`: value of δ, the "probability of disaster" in differential privacy
8. `alg`: number of algorithm to use, as according to the enumeration above
9. `vars`: `d` columns indicating whether each variable is dependent (1) or independent (2)
10. `means`: `d` columns giving the mean of each column of the dataset to be generated
11. `sigma`: Σ, the $d \times d$ symmetric variance-covariance matrix from which the dataset will be generated according to a Multivariate Normal distribution, written as a vector of length $\frac{d^2+d}{2}$
12. `ranges`: `2d` columns giving the permitted minimum and maximum of each column in the dataset

Along with the testing framework, we have written a series of functions to provide various plots for the outputs of the testing framework. These functions could be equipped to the testing framework and used by others for convenience. Of course, another user could simply use or create their own preferred suite of graphing functions instead.

The following gives the columns for the results matrix. Note that many of the columns are identical to those in the parameter matrix, allowing one to save only the results matrix.

1. `i`
2. `t`: tth iteration of the differentially private algorithm, based on instructions identical to those in the current row

3. `n`
4. `d`
5. `restrict`
6. `epsilon`
7. `delta`
8. `alg`
9. `means`
10. `sigma`
11. `ranges`
12. `popBeta`: `d` columns, population values of β_i, when working with a synthetic dataset
13. `sampleBeta`: `d` columns, the sample β_i coefficients
14. `dpBeta`: `d` columns, the differentially private estimates of the β_i coefficients
15. `sampleSte`: `d` columns, the sample standard errors associated with the β_i
16. `dpSte`: `d` columns, a computation of standard error for the differentially private algorithm
17. `sampleT`: `d` columns, the sample t-value associated with the β_i
18. `dpT`: `d` columns, the differentially private estimate of the t-value

Occasionally, we may choose to add additional instructions to the input. This would be done by giving a second function parameter called `params`, in addition to the original parameter matrix. The details of these instructions are described in the comments accompanying our code, as these instructions were mostly written to allow some specific tests we desired, without changing the function entirely.

2.2 Functions for Plotting

We wrote a function that plots the three statistical estimates for each combination of n and ϵ. The function allows the researcher to look at only one algorithm or at many algorithms in the same time. Below is a partial example of what the function displays when used for 4 algorithms at the same time.

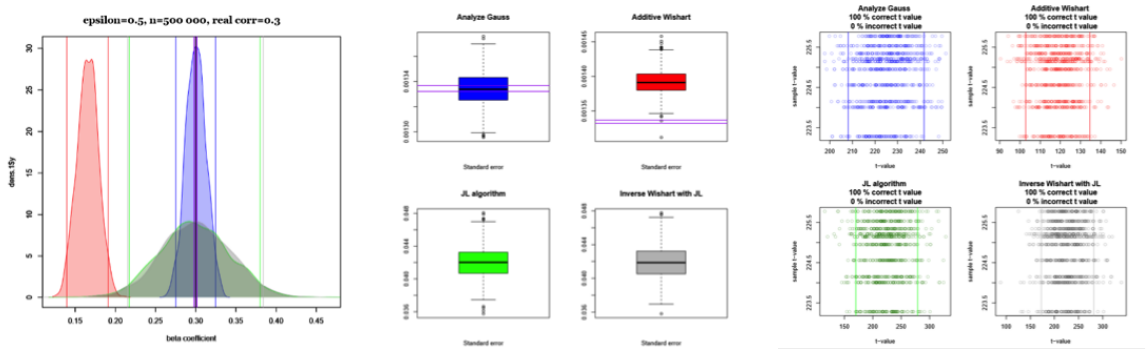


Figure 1: β densities, separated standard errors, separated t-values

For each combination of parameters, we have six sets of graphs:

- an overlay of the densities of a particular β coefficient for all algorithms,

- separate plots of the densities of the β coefficients,
- an overlay of the boxplots of the standard errors,
- separate boxplots of the standard errors for each algorithm,
- an overlay of the plots of the t-values,
- and separate plots of the t-values for each algorithm.

Each algorithm is depicted by the same colour in all graphs. The purple lines represent the range of the estimates from the many samples we created. The black line (possible covered by the purple lines) represents the real correlation in the population, available for our Monte Carlo simulations.

By looking at the overlaid plots, one can immediately compare the algorithms against the others. By looking at the separate densities, one can see more clearly the performance of the algorithm compared to the desired results.

3 Basics Results and Analysis

3.1 Scope of Tests

We examine the key question: How does each algorithm perform when compared to the other algorithms and their noiseless counterparts?

Given the importance of the β_i values and their associated standard errors in a regression, we wanted to visualize the range of results our differentially private algorithm could output. In addition, we want to investigate the t-statistic, the ratio of beta coefficient to its standard error, used in hypothesis testing where the null hypothesis is $H_0 : \beta_i = 0$. The larger the t-value, the more confident we are that the β coefficient is significant.

3.2 Experimental Setting

We wanted to greatly focus on the effect of several factors such as n, the number of observations in the dataset and the algorithm used (Analyze Gauss, Additive Wishart, Johnson-Lindenstrauss, Johnson-Lindenstrauss with Inverse Wishart). Thus, we hold all other factors constant, and we list their values below.

- T: We run each differentially private algorithm 100 times.
- d: We hold d, the dimensionality of the dataset, constant at 5 with the last of these variables being independent.
- The vector of means is identically 0.
- The Σ matrix we use is the following:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0.3 \\ 0 & 1 & 0 & 0 & 0.1 \\ 0 & 0 & 1 & 0 & -0.1 \\ 0 & 0 & 0 & 1 & 0 \\ 0.3 & 0.1 & -0.1 & 0 & 1 \end{pmatrix}.$$

  Note that this represents one independent variable strongly correlated with the dependent variable, one independent variable weakly positively correlated with the dependent variable, one independent variable weakly negatively correlated with the dependent variable, and one independent variable uncorrelated with the dependent variable. Also, none of the independent variables are correlated with each other.
- We choose to *censor* rather than to *truncate* the data.
- $\epsilon = 0.5$.
- $\delta = e^{-10}$.
- We restrict each variable to the range $[-4, 4]$.

To examine the performance of the four algorithms mentioned before, we run multiple simulations using the setting above. The experimental setting uses population parameters given by the researcher and draws independent samples according to them. Then, all algorithms are applied to each sample and the results are registered. This allows researchers to see how much noise the algorithm introduces in the regression results compared to the imprecision of

the sample. To keep a record of the sample variability, all the graphs presented here depict the range of sample results with purple lines.

3.3 Basic Comparisons

The following graphs depict the case in which we test the 4 algorithms using $\epsilon = 0.5$, $n = 500000$, and a population correlation of 0.3.

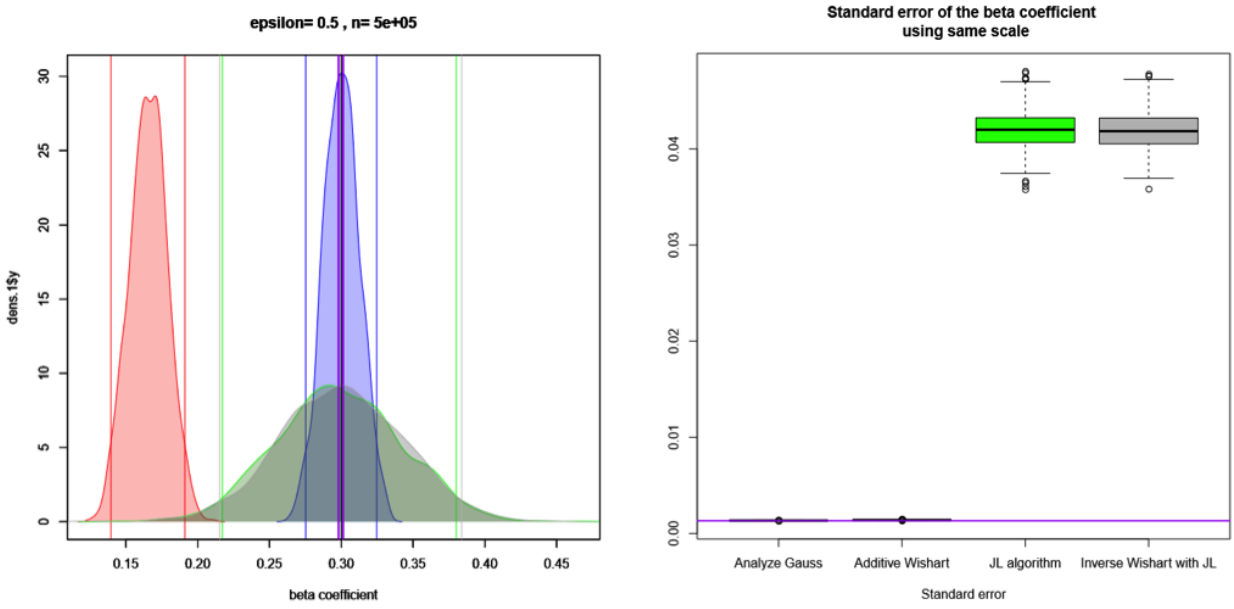


Figure 2: βs and standard errors for 4 algorithms

We notice that the 4 algorithms we tested each have a different behavior. While Analyze Gauss (blue), Johnson-Lindenstrauss transform (green) and Inverse Wishart Sampling (gray) are centered on the true correlation of 0.3 with different spreads, Additive Wishart (red) appears to have a bias in the negative direction. This behavior is consistent across the board.

Because in these graphs, the number of observations in the data set is very large ($n = 500000$), we have no negative outputs for the differentially private estimate of β. However, if n is a smaller number, we encounter a larger spread of the differentially private estimate of β. For example, the algorithms may output values over the range $[-40, 40]$.

Turning our attention to the standard error, we notice that Analyze Gauss and Additive Wishart have smaller standard errors while the other two algorithms have larger ones. One notable fact about the standard error resulting from our simulations is that it converges to the sample standard error as n increases. We thus see that these estimates of standard error are poor, as we would expect a greater standard error than the sample standard error, as the randomness of the differentially private algorithm should increase the spread of our β values. This can also be seen visually on the graphs: the purple lines in the β graphs are very close

to 0.3, but the distribution of differentially private β have a much greater variability which is not reflected in their corresponding standard errors.

Therefore, the standard error associated with each β computed from the differentially private second-moment matrix is inconsistent with the typical interpretation of standard error. Hence the t-values that we obtain using these standard errors are biased and possibly unsuitable for computing hypothesis testing. This happens with a greater frequency at small values of n.

Despite the naivety of these values for standard error, we will use them to provide a statistical claim regarding the null hypothesis, $H_0 : \beta_i = 0$. Understandably, we may make some unjustified claims due to the nature of the standard error calculation here, but we will be able to make at least some preliminary observations regarding the algorithms tested. We now analyze our algorithms one-by-one using this method.

3.3.1 Analyze Gauss

In general, for sets of graphs in these sections, the first graph will be for $n = 1000$, the second graph will be for $n = 50000$, and the third graph will be for $n = 500000$. Exceptions will be noted clearly in captions. We begin by examining graphs for the differentially private βs outputted.

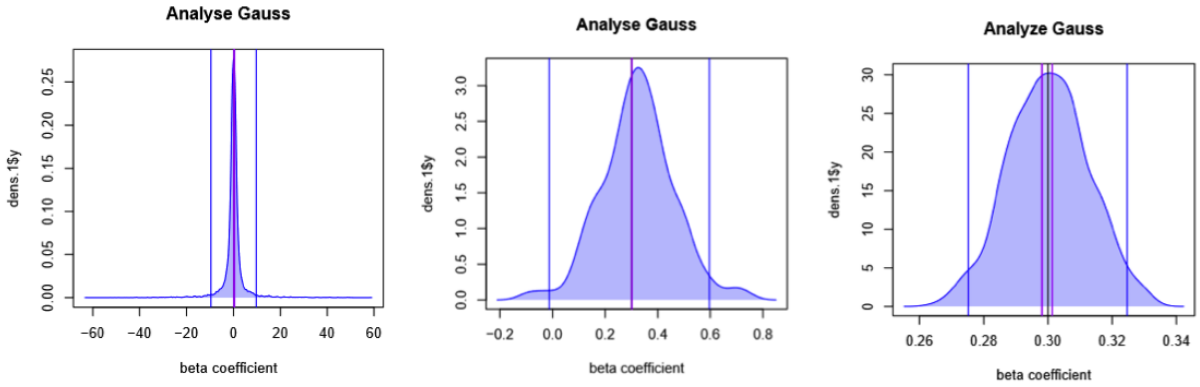


Figure 3: Analyze Gauss βs

The performance of the Analyze Gauss algorithm depends greatly on the number of observations in the dataset. Although we do not focus on ϵ here, there is a high dependence of the value of ϵ as well. The section on Analyze Gauss and positive semi-definite matrices will look more in-depth at this issue, as we hypothesize that the need to post-process certain outputted matrices for low values of n is greatly detrimental to the performance of the algorithm. We note here that the results obtained from Analyze Gauss under a certain threshold are entirely unreliable as they output unreasonable βs and standard errors. In the first graph the results range from -60 to 60 for a population parameter of 0.3. Even for $n = 50000$, we still see that more than 2.5% of the results obtained are less than 0. At large values of n, the possible values of β outputted by the mechanism have small variance and are very close to the true correlation.

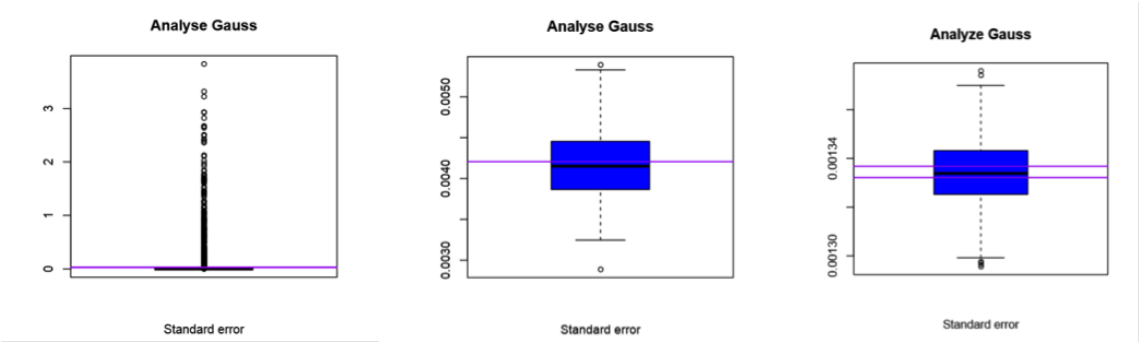


Figure 4: Analyze Gauss standard errors

For Analyze Gauss, as with the other algorithms, we see that the standard error associated with β is smaller when we have larger values of n. It in fact converges to the sample standard error. However, as previously mentioned, the functionality and meaning of the standard error has been lost here. This can be seen in the t-values below.

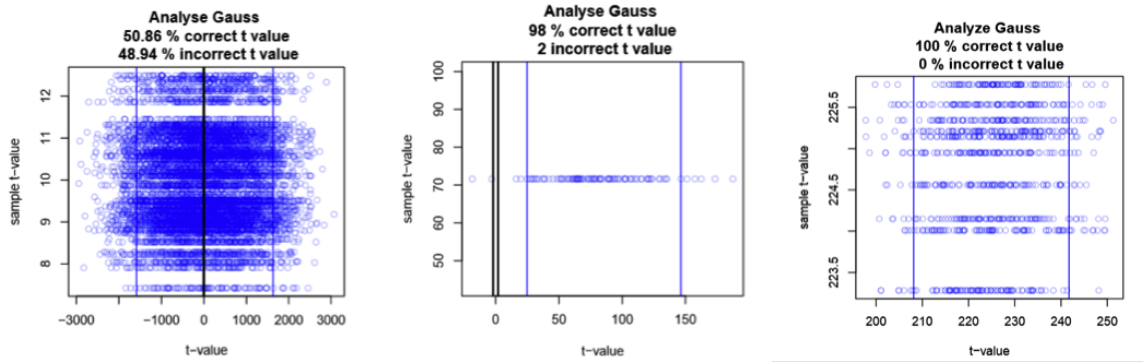


Figure 5: Analyze Gauss t-values

We notice that there are many incorrect t-values for small n, with the number of incorrect t-values decreasing as n increases. Note that these incorrect t-values sometimes allow us to make a very certain declaration in the "wrong" direction. For example, a t-value less than -1000 would lead to the conclusion that $\beta < 0$, essentially with absolute certainty. It would be more reasonable that we would obtain more t-values in the range $(-1.96, 1.96)$, as these t-values would prevent us from making any conclusion regarding the data, but the presence of t-values of large magnitude in both directions is quite disastrous and needs to be addressed.

3.3.2 Additive Wishart

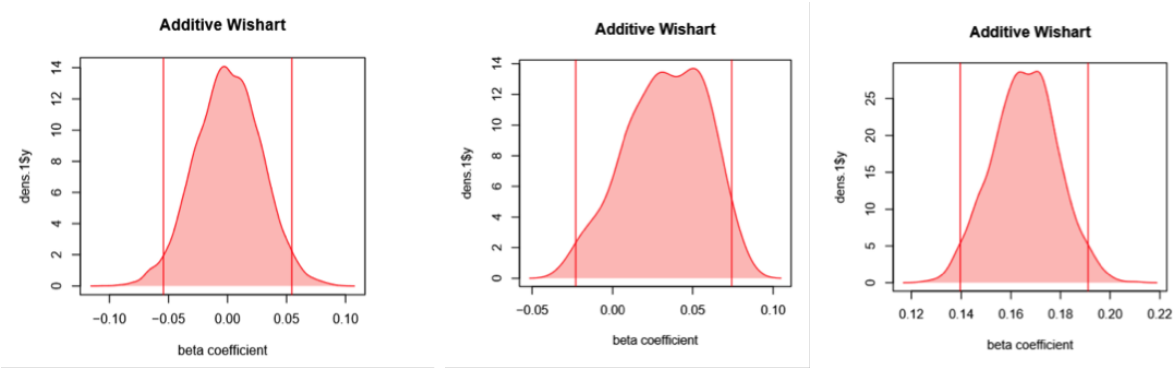


Figure 6: Additive Wishart βs

Without post-processing, we see here that the Additive Wishart mechanism outputs βs that are not reliable. This is theoretically confirmed, as the differentially private estimates of β are less than the actual value of β in expectation. Thus, some post-processing of the results here is needed.

Compared to other algorithms, Additive Wishart outputs differentially private βs that are less variable than the one obtained with Johnson-Lindenstrauss or Inverse Wishart Sampling mechanisms. If post-processing is done effectively, Additive Wishart could be a very useful differentially private mechanism as we see here that the spread of the βs themselves are quite tight.

Figure 7: Additive Wishart standard errors

The standard errors of Additive Wishart tend to be higher than the sample ones. However, they also converge to values close to the sample standard errors as n increases. They are comparable to the ones obtained using Analyze Gauss. Unfortunately, just as with Analyze Gauss, these also lack functionality for the purposes of hypothesis testing.

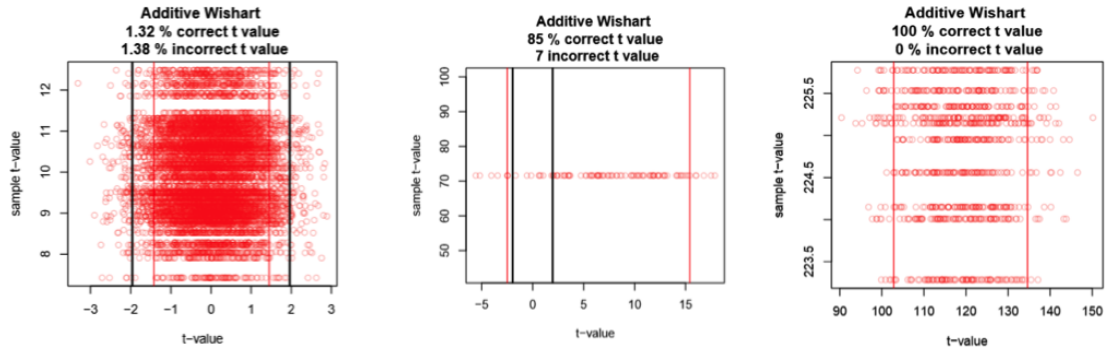


Figure 8: Additive Wishart t-values

Additive Wishart is more conservative in terms of correct t-values than Analyze Gauss. However, these t-values are highly unreliable as the estimate for β is biased and our computation for standard error is poorly chosen.

3.3.3 Johnson-Lindenstrauss

For the next two algorithms, we do not have many additional comments, so we simply present the graphs below.

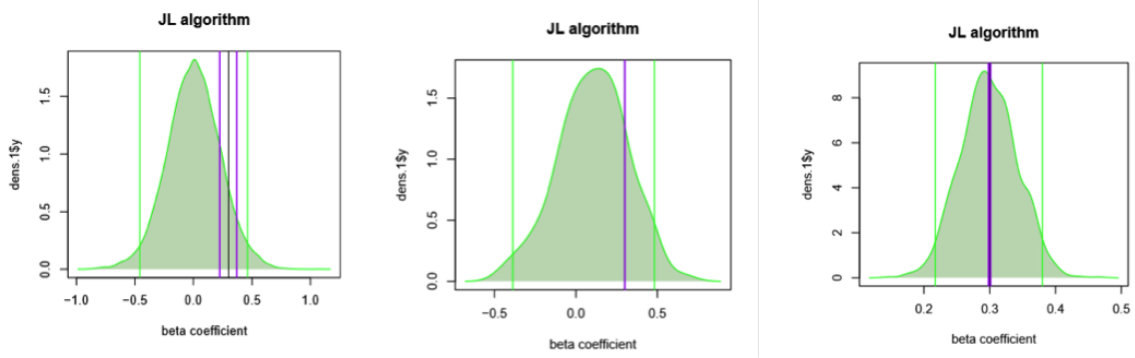


Figure 9: Johnson-Lindenstrauss βs

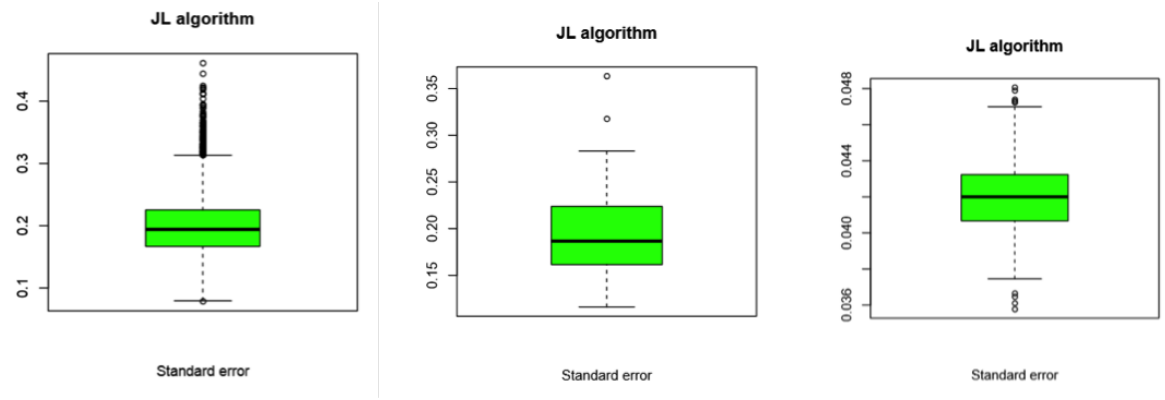


Figure 10: Johnson-Lindenstrauss standard errors

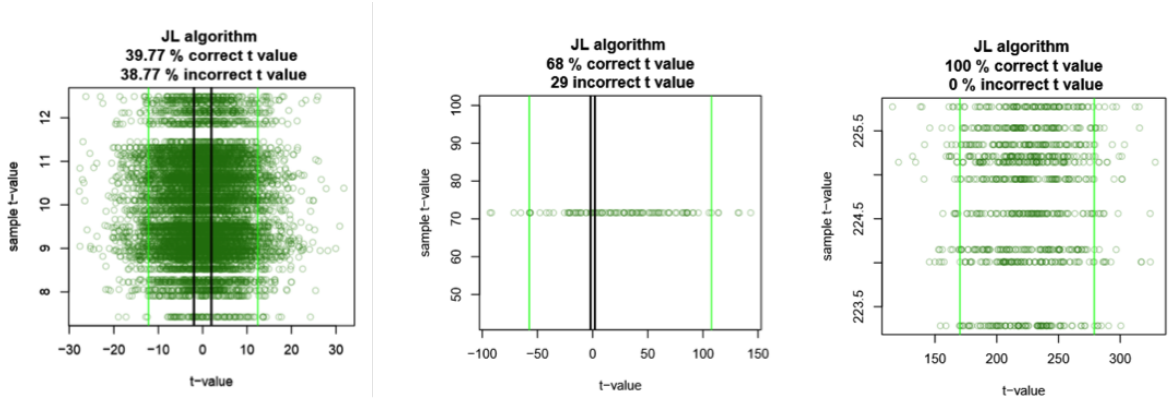


Figure 11: Johnson-Lindenstrauss t-values

3.3.4 Johnston-Lindenstrauss with Inverse Wishart

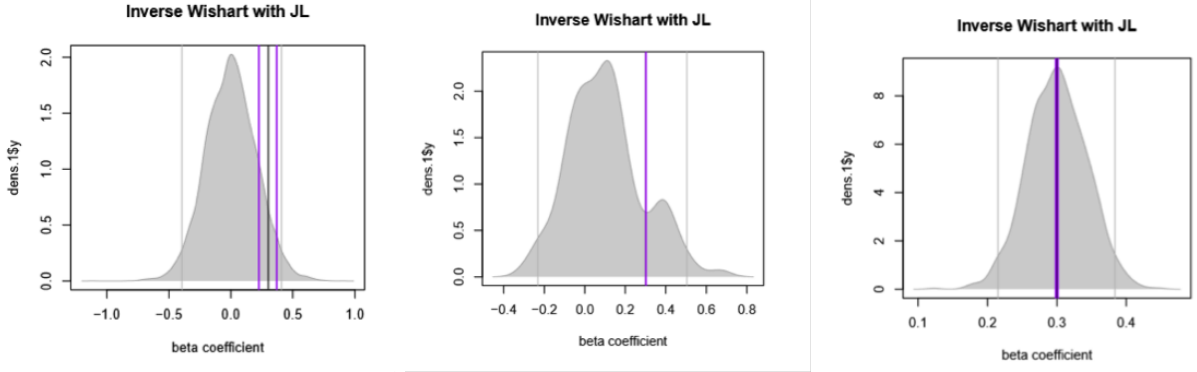


Figure 12: Johnson-Lindenstrauss with Inverse Wishart βs

Figure 13: Johnson-Lindenstrauss with Inverse Wishart standard errors

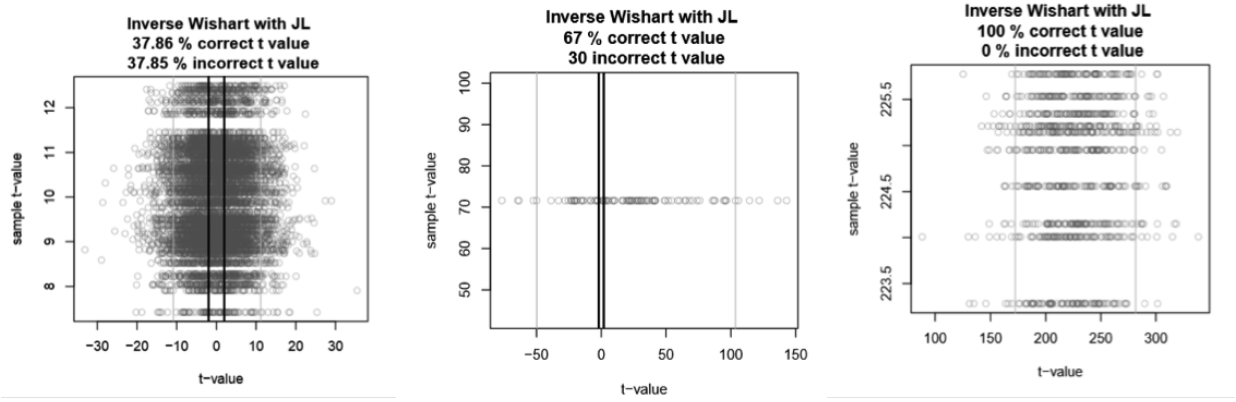


Figure 14: Johnson-Lindenstrauss with Inverse Wishart t-values

Note that these previous two algorithms are more conservative in their t-values than the first two algorithms. However, we still would not like t-values of -10, for example, when our true correlation is positive.

4 Sample Size Tests

In this section, we perform tests related to sample size. We find the minimum sample size s required to have 95% of our t-values be "correct." That is, these t-values should have the same sign as the true correlation and be significant if the true correlation is significant (which, in our tests, it is).

4.1 Sample Size and Proportion of Correct T-Values

As usual, we take an empirical approach to answer this question. We run simulations for different values of n, varying the algorithm used and the strength of the correlation in the population. We continue to use the parameters from before: $\epsilon = 0.5$, $\delta = e^{-10}$, true correlation of 0.3, and the same mean and Σ as earlier. The below graph compares sample size and proportion of "correct" t-values for all 4 algorithms.

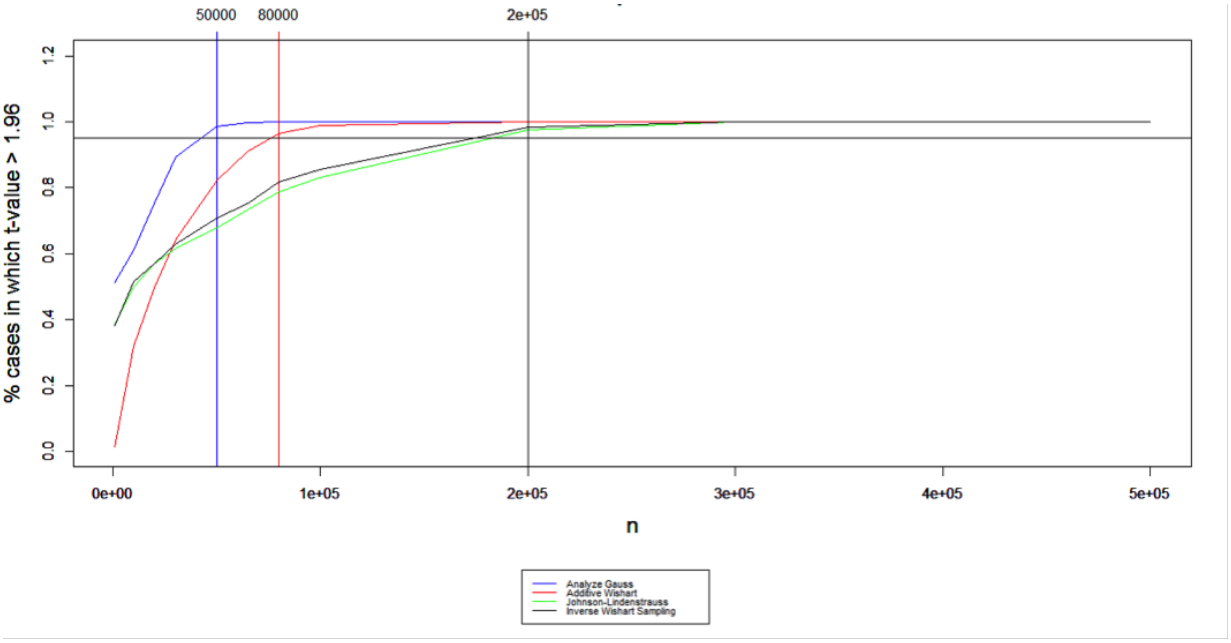


Figure 15: Sample size and proportion of correct t-values for all algorithms

Using our naive standard error computation, we have the following ranges for the minimum sample size s required to get the desired 95% correctness of our algorithm. Note that these ranges exist as we performed tests over a fairly coarse scale.

1. Analyze Gauss: $40000 < s < 50000$.
2. Additive Wishart: $65000 < s < 80000$.
3. Johnson-Lindenstrauss: $100000 < s < 200000$.
4. Johnson-Lindenstrauss with Inverse Wishart: $100000 < s < 200000$.

Given that many of the datasets used in social science have few observations, there needs to be work in improving the performance of these algorithms on smaller datasets.

4.2 Minimum s as a Function of True Correlation

Continuing to use our previous value of Σ, we now examine each of the independent variables, with true correlations of 0.1, -0.1, and 0, in addition to applying the hypothesis test to the intercept, which should be 0, to find the breakpoints for s. These tests yielded the following graphs:

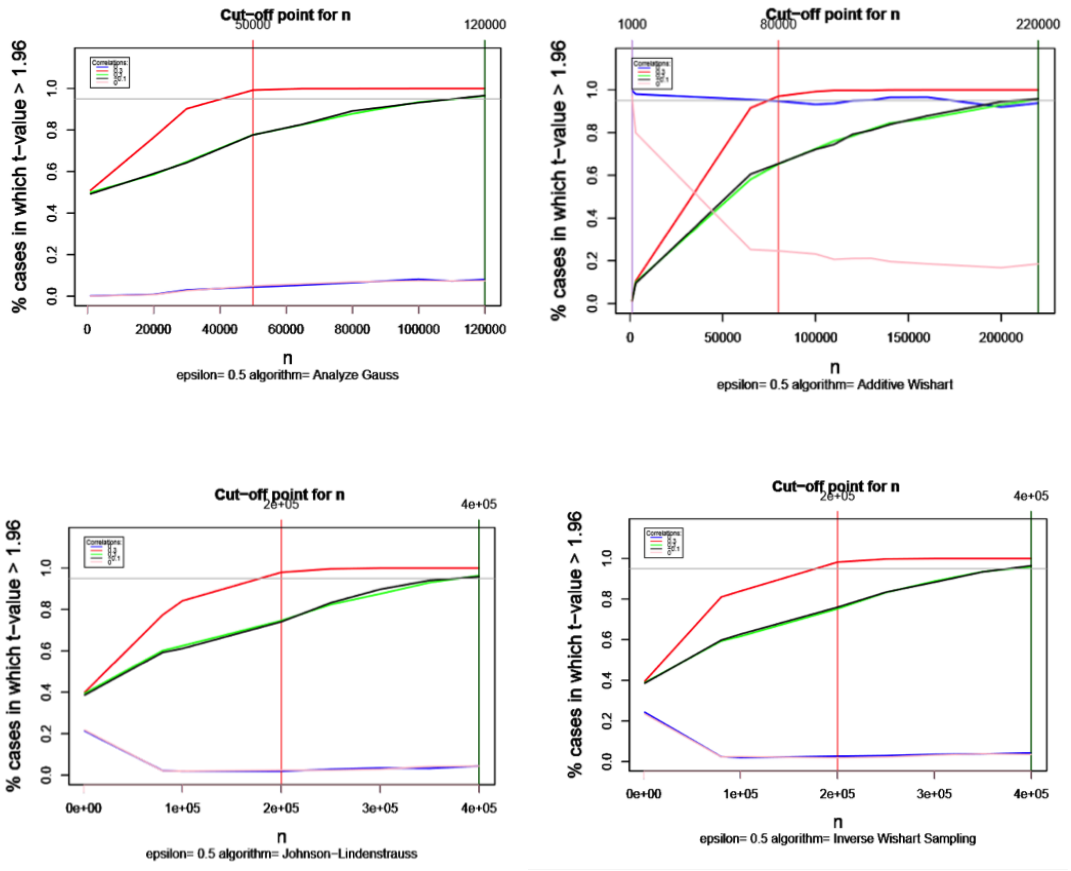


Figure 16: Variables with correlations of 0.1, -0.1, 0, and 0 (intercept), starting from the top left and moving clockwise

We make the following observations regarding the above graphs:

- Stronger correlations appear easier to identify in smaller samples regardless of the algorithm used.
- The behavior of the case of 0 is different for each algorithm.
- In the 0 cases, Analyze Gauss has the same percentage of correctly identified t-values for both the intercept and the uncorrelated independent variable.
- Additive Wishart identifies the t-value of the intercept in more than 80% of cases, regardless of the value of n. This is due to the fact that one column of 1 was added to the variance-covariance matrix without noise. The percentage of correct t-values for the correlation of 0 are initially very high, but oddly, they quickly drop as n increases.
- The Johnson-Lindenstrauss algorithms have the same percentage of correctly identified t-values for the intercept and for the 0 correlation independent variable. An interesting

result is that at $n = 1000$, around 20% of the simulations outputted a correct t-value, only to drop afterwards towards 0. These effects should be examined in greater detail.

5 Correcting Estimates of Standard Error

As we saw in a previous section, the differentially private standard errors obtained are naive and lose their functionality. This is extremely important as it affects our ability of conducting statistically appropriate hypothesis testing for the correlation coefficients and very often outputs an inexistent correlation as being correct.

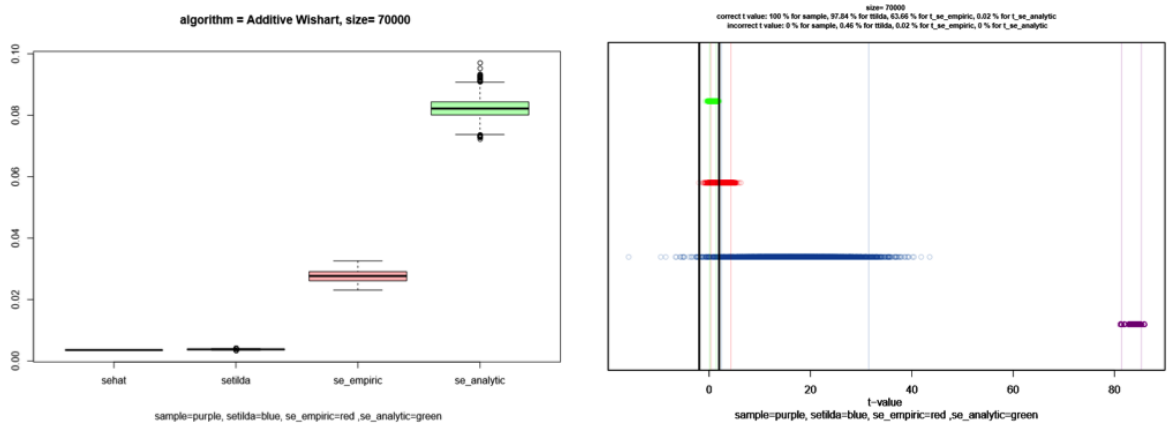
New methods of computing this estimate need to be determined. At present, we have alternatives for the following algorithms: Additive Wishart, Johnson-Lindenstrauss, and Johnson-Lindenstrauss using the Inverse Wishart distribution. However, we do not know how to do it for Analyze Gauss, and the solutions for Johnson-Linderstrauss and Inverse Wishart Sampling exist but have not been implemented. The following section will focus on the experiments done using an alternative standard error for Additive Wishart mechanism and involves using an upper theoretical bound on the error of the differentially private βs.

5.1 Alternative Standard Errors for Additive Wishart

Ideally, the standard error estimate will reflect the standard deviation present in the differentially private β coefficients while taking into account the sampling error as well. This will allow a statistically valid hypothesis test. A theoretical solution found for the Additive Wishart algorithm involves adding an upper bound for the error due to the differentially private algorithm to the naively computed differentially private standard error.

In addition to this solution, which we call the analytical standard error, we computed empirically an ideal standard error by using the standard deviation in the β estimates generated from several iterations of the differentially private algorithm. Note that this represents an ideal, as it cannot be done in a differentially private setting.

For our testing parameters, we use our previous values of $\epsilon = 0.5$ and $\delta = e^{-10}$. Additionally, we keep using the means and Σ that we have used previously. This time, we use $n = 70000$ and $n = 500000$ as our sample sizes, and we run the algorithm 100 times on 100 iterations of generating data and performing the differentially private algorithm on the generated data. This tests yielded the graphs below.



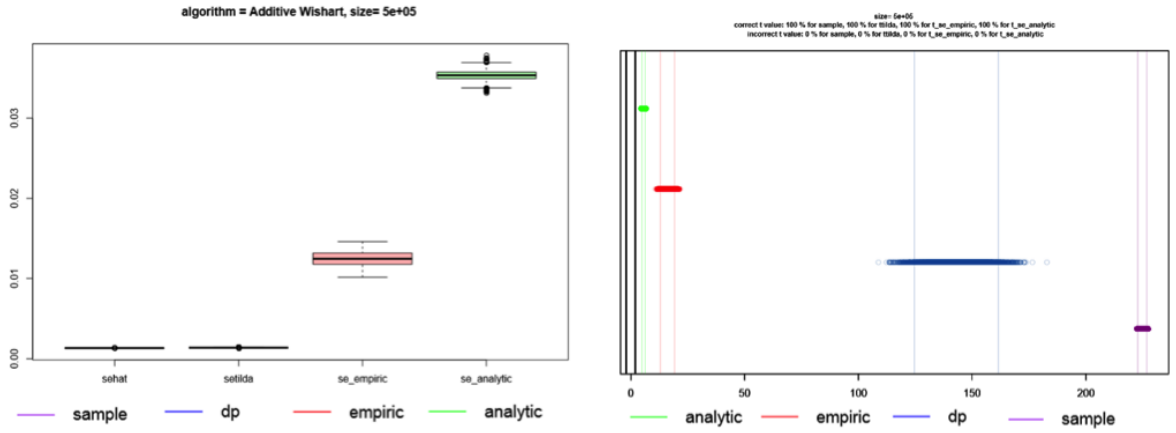


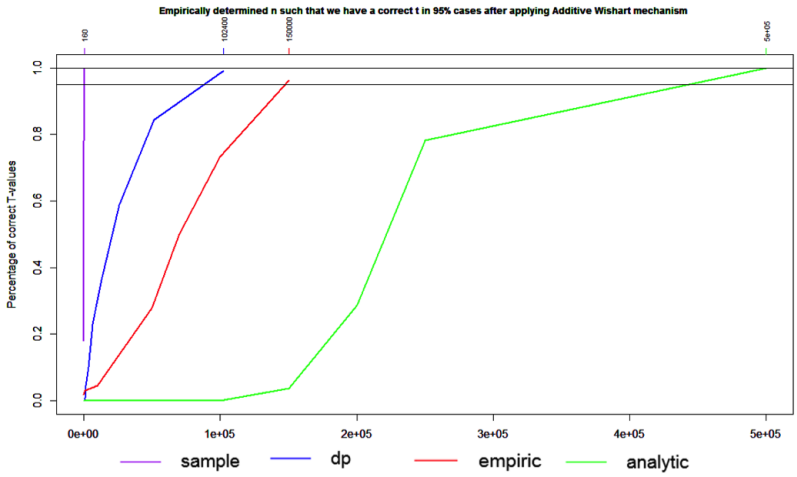
Figure 17: Various standard errors and t-values for $n = 70000$ (top) and $n = 500000$ (bottom)

We see that the sample is always correctly identifying the correlation. Note that all these results are based on the Additive Wishart computation without post-processing to adjust for bias. We note the following observations:

- At $n = 70000$, the naive differentially private t-values are correct 97.84% of the time. The "ideal" empirical standard error is correct 63.66% of the time, and the upper bound is only correct with observed proportion 0.02%.
- At $n = 500000$, all types of standard error produce all correct t-values at differeing ranges.

An important outcome of the comparison is that we notice the empirical standard error is far more conservative than the naive computation. This indicates that the latter is not reliable. However, the upper bound applied to the naive standard error seems to be too conservative in databases. Thus, it appears necessary to find a better computation of standard error or to somehow incorporate the empirical computation into the outputted results without the loss of privacy budget.

We now find the minimum value of n required to get 95% correct t-values.



The previous graph is constructed using all the parameters specified in the Basic Comparisons section. One can notice that the analytic standard error could be useful in very large datasets; otherwise, it is very conservative. The red line, our "ideal case" requires around 150000 observations in a dataset in order to produce correct hypothesis tests.

This section shows that future work is needed to find a better standard error computation for the Additive Wishart algorithm. This extends to other algorithms as well, as we do not have proper standard errors, or even estimates for standard error, that takes into account the randomness in the differentially private algorithm in addition to the randomness caused by sampling.

5.2 Effective Sample Size for Analyze Gauss

Unlike for Additive Wishart, it is very difficult to find an analytical bound on the error of Analyze Gauss. However, we would still like to find a way to report a standard error similar in error to the empirical standard error of the β coefficients given by the Analyze Gauss algorithm. One way we attempted to do this was to find a relationship between empirical standard error and the "expected" standard error for a sample size n.

Mathematically, the formula for standard error for a coefficient β_i is

$$\mathrm{SE}_{\beta_i} = \frac{\mathrm{S}_{y,y}}{\sqrt{n}}(X^T X)^{-1}_{i,i}$$

where S is the swept second-moment matrix of the data, X is the matrix of the data itself, and y is the index of the dependent variable. We can compute from the population variance-covariance matrix Σ expected values for all of these terms other than n. Thus, for each sample size n, we call the standard deviation in the set of β_is reported after some large number of trials $\mathrm{SE}_{\mathrm{emp}}$ and set this equal to

$$\mathrm{SE}_{\mathrm{emp}} = \frac{S_{y,y}}{n_{\mathrm{eff}}}(X^T X)^{-1}_{i,i}.$$

Then, we solve for n and report the values.

The parameters for this test were the same as the usual ones: $\epsilon = 0.5$, $\delta = e^{-10}$, and $d = 5$ with our usual means of 0 and usual value of Σ. In particular, we focus on the β corresponding to the independent variable with true correlation 0.3. To compute $\mathrm{SE}_{\mathrm{emp}}$, we generated 100 samples of data from each size of n and performed the differentially private algorithm once per sample of data. However, this was quite computationally intensive for large values of n, so we only collected a small set of preliminary data that we display in the table following.

n	n_{eff}
10^2	$7.25 \cdot 10^{-2}$
10^3	$7.55 \cdot 10^{-2}$
10^4	$4.77 \cdot 10^{-2}$
10^5	$3.71 \cdot 10^{2}$
10^6	$3.65 \cdot 10^{4}$
10^7	$3.33 \cdot 10^{6}$

Note here that for small n, the empirical standard error of β is extremely large, causing even the effective sample size to be mostly meaningless. Otherwise, if more tests were done with improved granularity, it may be possible to find an empirical relationship between n and n_{eff}, most likely depending on the values of ϵ and the true correlation as well.

If such a relationship were found, it could be used as an explanation for researchers unfamiliar with differential privacy to explain how our addition of randomness here affects the original sample. Furthermore, such a relationship could be used to release an estimated standard error so researchers could at least use such a value for preliminary hypothesis testing.

6 Additional Tests

6.1 Positive Semidefiniteness of Matrices in Analyze Gauss

The results obtained in our initial analysis of Analyze Gauss pointed to a very bad performance at low values of n. We began to examine what aspect of this algorithm caused this poor performance. One possible answer was the need to post-process many matrices outputted for low values of n, as these matrices do not satisfy the required mathematical property of positive semidefiniteness for second-moment matrices. Thus, this yields the following questions:

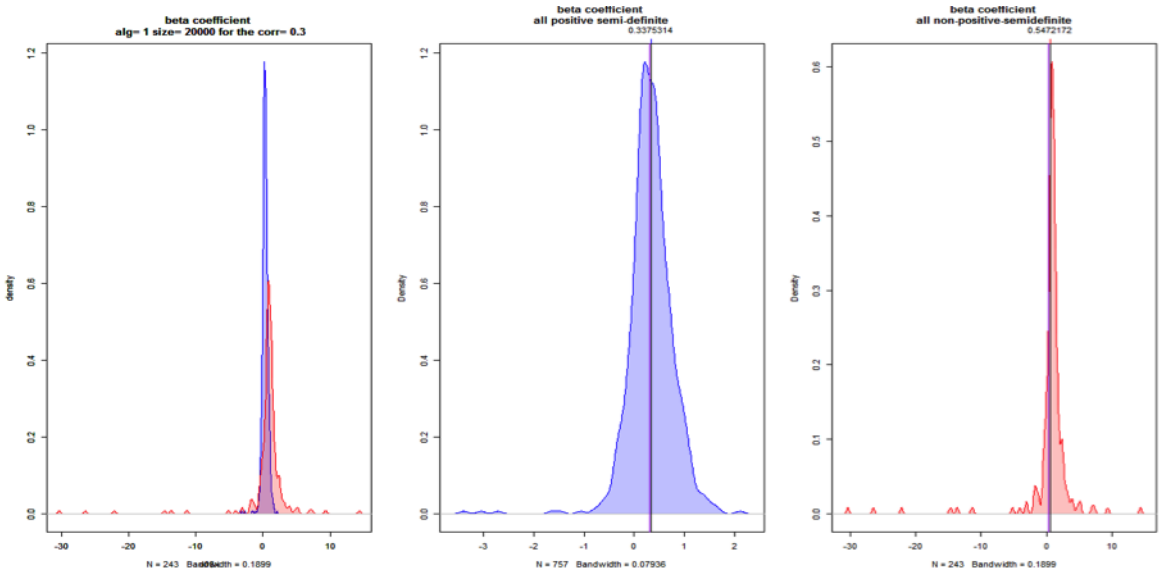
1. How does the need to post-process affect the performance of the Analyze Gauss algorithm?
2. What is the size of the dataset for which all differentially private second-moment matrices are positive semidefinite?

6.1.1 Post-Processing vs Performance

We begin by examining the first question. A slight extension to the testing framework allowed the results matrix to display whether the result in a particular row of the matrix required post-processing. Additionally, we created a function that separates the regression results for matrices that are positive semidefinite from those that required post-processing.
The function then graphs the βs of these results separately. The cases that did not require post-processing are blue, and those that did are red. The function then outputs the following three graphs:

1. a graph with overlaid densities for both cases,
2. a density graph for the matrices that did not require post-processing, and
3. a density graph for the matrices that did require post-processing.

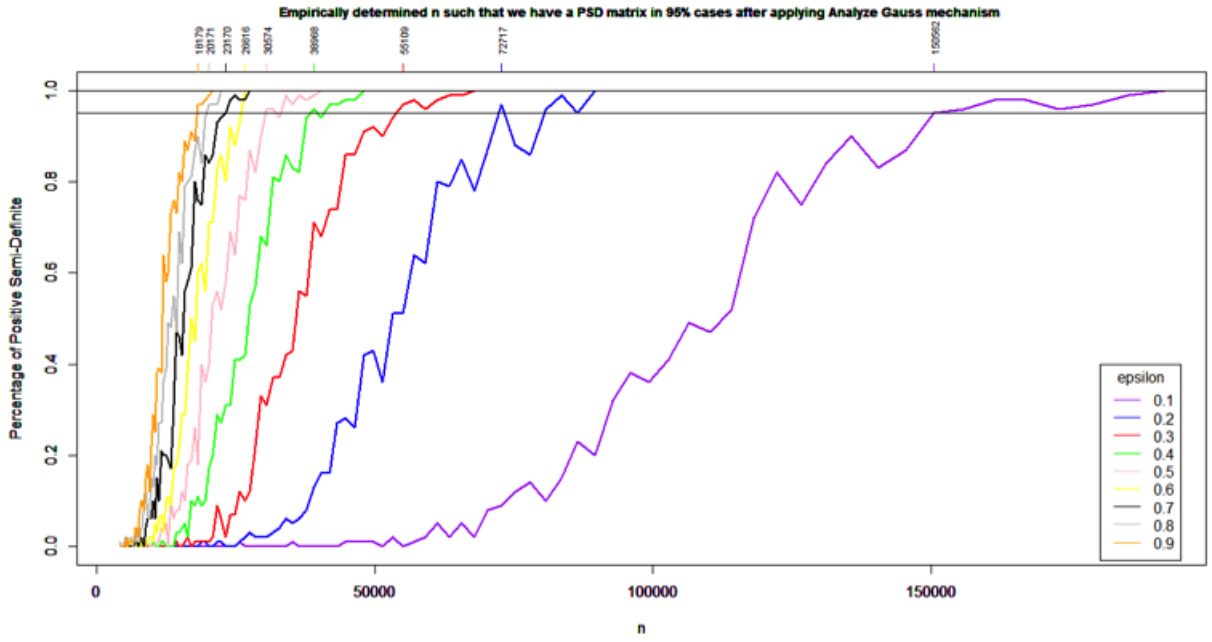
Additionally, we used $n = 20000$, $\epsilon = 0.5$, and $\delta = e^{-10}$. We used means of 0 and the Σ that we typically work with, focusing on the variable with true correlation 0.3.



When looking at the scale of the last two plots, we can see that the large variance in the β coefficient comes from those cases when post-processing was applied. At the bottom of the graphs, we can see how many cases are used for each density. In the example above, we had 22.6% cases where post processing was applied, and these are the simulations that lead to an overall bad performance of the algorithm. The mean for the cases not requiring post-processing is 0.33, very close to the true mean of 0.3,, while the mean for the cases requiring post-processing is 0.55. In this case, we can notice that the density of the β coefficients for the red density is to the right of the blue one, suggesting that the median is even further away from 0.55.

6.1.2 Minimum n for No Post-Processing Required

Knowing that Analyze Gauss performs poorly for small datasets, we would like to locate a breakpoint at which we can be reasonably certain that post-processing does not occur. In particular, we are interested in the value of n at which post-processing does not occur at least in 95% of cases. To this end, we created a function that empirically determines this n. We used this function for various epsilon values and then plotted the results in one graph, which we present below.



We also present this information in the table following. ϵ and n denote their usual values, and p denotes the proportion of matrices that did not require post-processing at those levels of ϵ and n. These values of p are not exactly p as we tested values of n for fixed ϵ without using very fine granularity.

ϵ	n	p
0.1	150562	0.95
0.2	72717	0.97
0.3	55109	0.97
0.4	38968	0.96
0.5	30574	0.96
0.6	26616	1.00
0.7	23170	0.95
0.8	20171	0.97
0.9	18179	0.97

We can see through the table and graph that it appears the condition $\epsilon n > 15000$ is a good approximation of when we can be reasonably certain that outputted matrices are not being post-processed. Below these values of ϵn, the results of Analyze Gauss cannot be treated as nearly as reliable, and another algorithm should be used.

6.1.3 Potential Solutions

In the present form, Analyze Gauss is unusable at small values of n. One potential solution is to simply run the algorithm until a matrix that does not require post-processing is obtained. This expends a privacy budget of 2ϵ rather than the typical ϵ when dealing with $\delta = 0$, and more theoretical analysis is necessary to determine the effect of a non-zero δ. This solution has the flaw of requiring a much greater computation time if the algorithm must be run many times before obtaining a single matrix that is positive semidefinite.

Alternatively, other methods of post-processing could be explored. Presently, the algorithm adds cI_k to the $k \times k$ matrix outputted by Analyze Gauss, where $c = 1+$ (mininum eigenvalue of matrix) and I_k is the $k \times k$ identity matrix. Alternative methods of post-processing involving expected noise, for example, have been suggested but not yet implemented. It is possible that one of these methods does not affect the values of β as drastically as the present method of post-processing does. The testing of these methods is an important step that should be taken in the future.

6.2 Decreasing Dimensionality by Ignoring Variables

It is plausible to believe that a researcher studying a set of say, 10 variables, would be completely uninterested in several of the variables. The algorithms we have for performing regressions simply return the second-moment matrix of the data, which allows the researchers to perform any of the possible regressions on a dataset. However, as the researcher may not be interested in some of these variables, we could perhaps optimize our privacy budget by simply ignoring several of the variables.

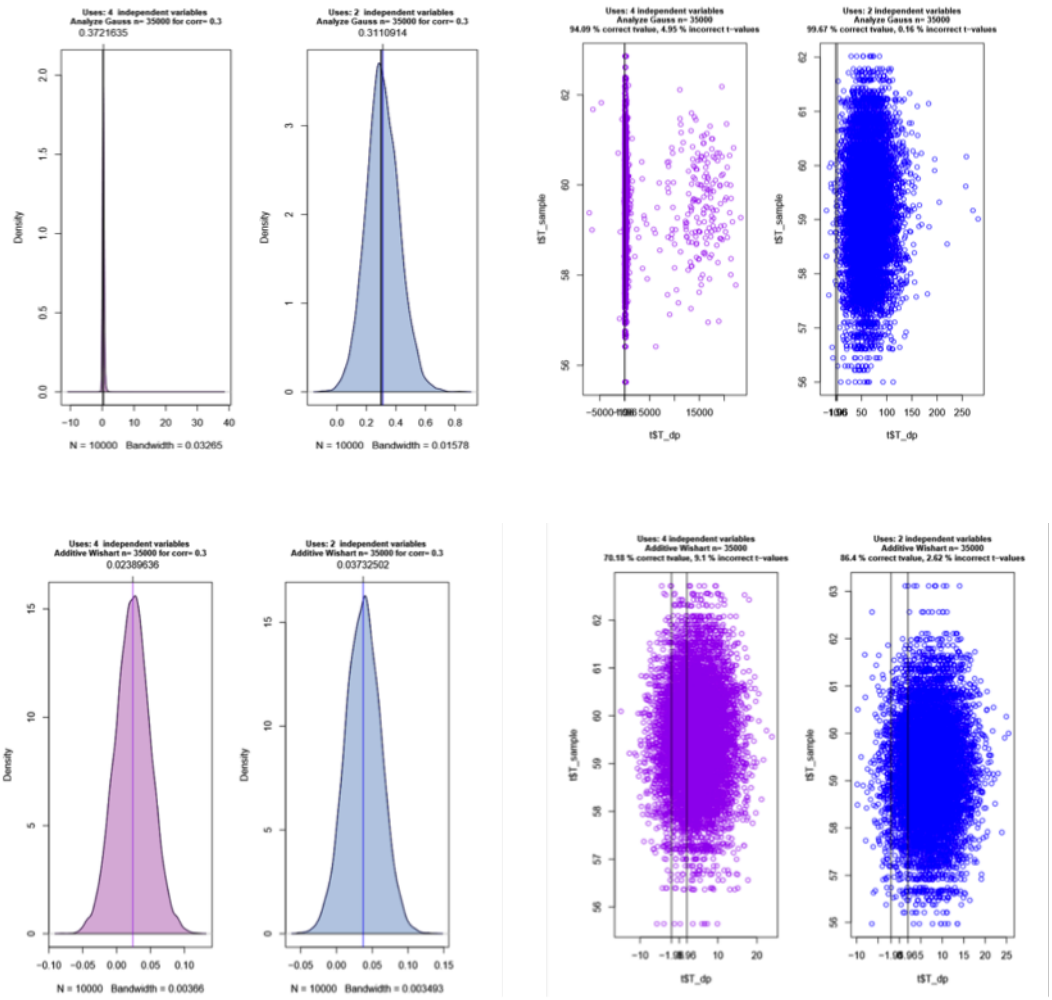
We work in the following setting: We first consider the situation we have before, with $\epsilon = 0.5$, $\delta = e^{-10}$, and our previous value of Σ with means of 0. Then, we compare this with the case

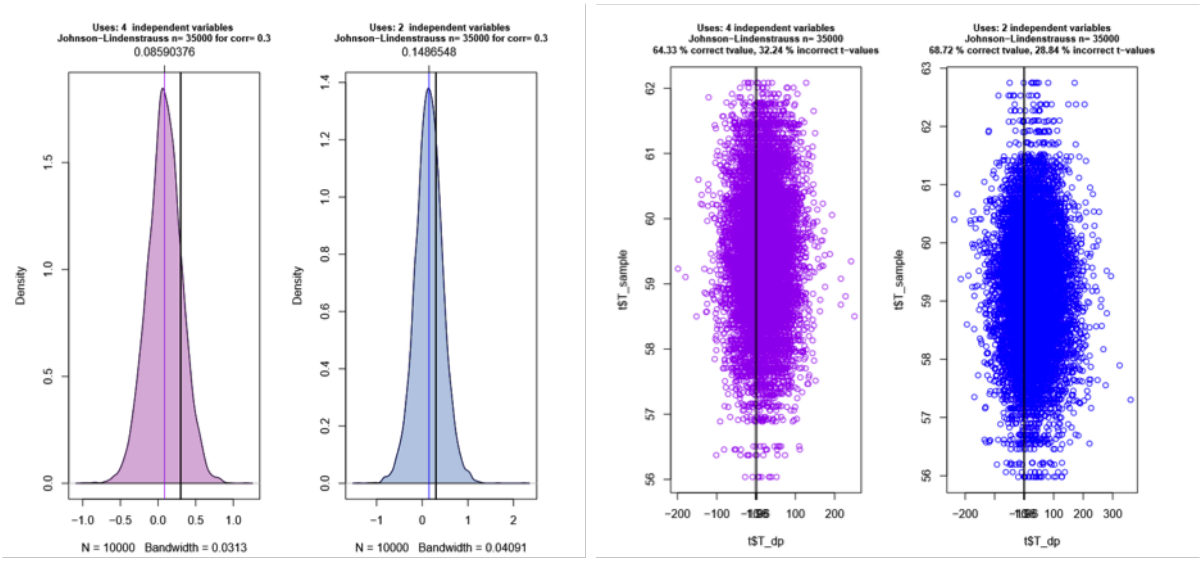
with $d = 3$ and

$$\Sigma = \begin{pmatrix} 1 & 0 & 0.3 \\ 0 & 1 & 0 \\ 0.3 & 0 & 1 \end{pmatrix}.$$

Essentially, the independent variables with true correlations of 0.1 and -0.1 have been removed. In both cases, we examine the value and standard error of the β associated with the variable with a true correlation of 0.3. The sample size we use here is $n = 35000$.

We display the graphs following here. The graphs display the results for the Analyze Gauss, Additive Wishart, and Johnson-Lindenstrauss algorithms in that order. In each pair of graphs, the graph on the left is for 4 variables, and the graph on the right is for 2 variables. The left pair of graphs shows βs, and the right pair of graphs shows standard errors.





As theoretically suggested, the spread of βs is much tighter in the case where we have only 2 independent variables instead of 4. This could be a significant improvement for researchers who are certain that they can exclude some of the potential independent variables from their analysis. For future analysis, we could examine the difference between 10 variables and 2 variables, for example, to see more drastic effects.

7 Conclusion and Future Work

In each subsection of the report, we have highlighted the points to be researched in the future. Here, we will mention several potential areas to focus on for the continuation of the testing of these algorithms, with the focus of eventually being able to use these algorithms on real data.

Overall, one of the major points we realized through our tests is that hypothesis testing fails completely after differential privacy. The framework showed that the differentially private mechanism works well for point estimates of β but not for inferential statistics. In this direction, some progress has been made in finding alternative ways of calculating the standard error. However, it is far from being solved and should be a major focus of improving these algorithms in the future. Releasing a tool for differentially private linear regression would crucially need a viable method of hypothesis testing.

Secondly, the first two algorithms, Analyze Gauss and Additive Wishart, need improved methods of post-processing. For Analyze Gauss, other methods of post-processing positive semidefinite matrices should be tested to determine which is effective. For Additive Wishart, the impact of post-processing to correct for bias in the expected value of β should be tested as well.

Thirdly, the hypothesized relationship of ϵn being directly related to the performance of our algorithms was confirmed in several cases, such as with the positive semidefiniteness of matrices outputted by Analyze Gauss. This will help researchers establish an intuitive understanding when choosing the privacy budget and considering the value of ϵ.

The continued study and testing of these areas in particular would be greatly helpful for the project and for the ultimate goal of creating a working tool for researchers. The continuation of the work we began this summer would hopefully resolve many of the issues we encountered and tweak these algorithms for the better.