Exploring the Limits of Differential Privacy

David D. Clark

Version 1.0 of September 5, 2023

1 Introduction

Differential Privacy (DP) is well-recognized as a useful and powerful privacy enhancing technology, but there are contexts in which it is less well suited, or where its applicability is less well understood. In this paper, I explore the use of DP in one such context—the protection of corporate proprietary information in computing aggregate industry-wide query results. Firms are often willing to allow data from several firms to be aggregated, provided that potentially harmful information about individual firms is protected. This is the challenge explored in this paper.

The protection of firm-level proprietary information is distinctive in two ways. First, the concerns about the potential harms from disclosure are often somewhat different in structure from the concerns about disclosure of PII (as I will illustrate). Second, the sample size (the number of firms providing their data for aggregation) is often small, and a small sample size is recognized as a challenge for DP (and other approaches to protecting privacy) in the tradeoff of privacy with utility.

In this exploration, I proceed pragmatically, by trying to understand the actual potential for harm from releasing the result of a query. One of the challenges in explaining DP is that the formal description of the protection provided by DP is in terms of a mathematical abstraction called privacy loss. My goal in this note is to avoid the mathematics as much as possible, and explore the utility (and limitations) of DP in an intuitive way based on visualisation of simulated results, but it is useful to understand the formal basis of DP.

There are a number of ways to think about (and use) DP, but the simple version I use in this paper is to imagine that there is a database with a number of records (each with a number of fields). Each record represents data that should be protected from release. DP provides protection by adding a degree of noise to the result of a query against the records, to make it difficult to reconstruct the actual values of any one record. Typically, a single record might map to an individual, and the goal is to protect the PII of each individual while returning a useful result from a query. In this exploration, each record would represent the data from one firm.

DP defines a tradeoff between privacy protection (the degree of privacy loss) and the utility of the result of a query. The tradeoff is specified by the value of

a parameter called ϵ. The underlying philosophy of DP is that (to the degree specified by ϵ) any one individual in the database will not suffer a loss of privacy as a result of a query (or set of queries) posed against the database. One way to think about this is that DP adds enough noise to the query result so that it will be sufficiently difficult for a "data intruder" to determine if any one record was actually included in the computation of the query result.

Here is the formal statement of the protection provided by any scheme that is consistent with the DP framework. For a query function M, that returns a noisy answer to a query of a database:

- for all subsets of the database d_1 and d_2 that differ by one record:
- for all subsets S of the range(M):

$$\frac{P(M(d_1) \in S}{P(M(d_2) \in S} < e^{\epsilon} \tag{1}$$

I will return to the implications of this formula in Section 5, but the basic concept is that the degree of privacy loss is related to the extent to which the answer to a query will change with the addition or removal of a single record (which might typically contains PII related to one individual).

2 An (over) simple example of a query

In this paper I look at issues of protecting firm-level confidentiality, not user PII. Consider the following simple query. The query asks 100 firms to report, for some well-known and widely used system or application, what fraction of these systems running in each firm have been upgraded to the latest security patch. The goal is to get a sense, industry-wide, of whether firms are making sure their systems are up to date with respect to security patches.

In response to this query, each firm returns a number that is a fraction between 0.0 and 1.0. Assume (to simplify the discussion) there is a trusted agent that receives all these numbers and computes a result that is then made public. One obvious result would be the average of the values. (If this were an actual query, we might ask whether we wanted to weight the samples based on the size of the firm, or perhaps ask a more complex question of the raw data, but since this is just a toy example, assume the query computes the mean of the values returned from each firm.)

The first question is whether releasing the mean of these values (with no noise added) can cause harm to the firms that provided the inputs. What if the mean is .5? On the average, the firms that contributed data to the result have upgraded half of their systems to the latest patch level. However, there might have been a wide range of individual results–some firms might have upgraded all of their systems, some might have upgraded none. Unless we assume that the data intruder knows about most of the firms and is trying to learn about the remaining firms he does not know about (I return to this assumption in

Section 3.3), it is unlikely that revealing that the average is .5 will harm any one firm.

However, depending on the answer, this harmless situation may not hold. What if the computed mean is 0.0? In other words, none of the firms have upgraded any of their systems. It must be true (from the math) that each of the firms returned the value 0.0 as their firm's response to the query. The release of the average will cause reputational harm to each of the firms that contributed data. Note that if the average had been 1.0, the firms might be very happy to reveal that result–it shows that they all did great. Actual harm (or the potential for actual harm) depends on the context of the query, not math.

I call this harm the *binomial pathology* of computing the mean, because by analogy to the binomial theorem, there are lots of ways to take 50 balls out of an urn with 100 balls in it, but only one way to take out 0 or to take out 100. As the returned value of the query gets closer to the minimum or maximum of the possible range for the mean, there are fewer and fewer data values that can yield the result.

What this simple example illustrates is that the harm from a given query may be *data dependent*. However, the philosophy of DP is that whatever mitigation against privacy loss is put in place (e.g., some level of noise based on ϵ), it must be applied in a way that is independent of the actual data on which the query is based, because the act of adding noise in a way that is dependent on the data would itself reveal information about the data, so the result could no longer be shown to comply with the definition of DP.

Before I get to the role that DP might play in mitigating this harm, note that this problem can arise in other contexts. Consider a census block where the average age is 45. There could be much younger and much older people contributing to that average, so we learn little about them as individuals. But if the average is 85, it is a good guess that most of the people in that census block are old. If there were people aged 50 in the block, there would have to be people that were 130 years old to balance the average. In real life they don't exist.

The data intruder, seeing that the average age is 85, can only guess about a given individual in the census block, but sometimes a guess is good enough to cause harm.

One conclusion that might be drawn from the above analysis is that computing the mean is a dangerous form of query, exactly because the lurking binomial pathology can occasionally arise and reveal firm-level (or PII) data. I return to this point in Section 6, but for the moment let's see if DP can provide useful protection in this context.

The philosophy of DP accepts harms to the individuals in the database from a query *so long as the degree of harm is independent of whether your data was in the query.* Consider a health-related query that tries to establish a link between smoking and cancer. If that linkage is accepted, you (as a smoker) might see your health insurance or life insurance rates go up. You were harmed by the result of the query. But your information need not have been in the data that was sampled for this harm to occur. So the difference between the harm you

suffered if you were or were not in the data is zero, which is why the approach is called *differential* privacy. You were harmed, but it made no difference whether your data was in the database. The smokers in the data were harmed, but so were the smokers not in the data.

In the example above, if the returned value for the 100 firms queried was 0.0, they are individually harmed, and as well, a broader presumption is created that other firms are probably not updating their system to current patch levels either. This kind of harm may not be acceptable in the case of corporate confidentiality. If we seek voluntary release of data (as opposed to data release that is compelled by regulation or law), the fear of this kind of harm may cause firms to refuse to release data. So the next question is whether DP can contribute to the mitigation of this sort of harm.

3 Adding noise, in the DP way

The general approach taken in DP is to add uncertainty to the result of a query, to make it harder to distinguish the contribution of each data item contributing to the result. While there are many approaches that are consistent with DP, in this paper I use a simple (and common) approach of adding noise to the result, drawn from a Laplace distribution with zero mean (so the noise added to the true answer is equally likely to be positive or negative). The magnitude of the noise (the width of the Laplace distribution) is determined by two factors: ϵ (as I discussed above), and a factor called *Global Sensitivity*. While ϵ gets all the attention in discussions of DP, the concept of Global Sensitivity is of great significance, and provides important intuition about how DP works.

3.1 Understanding Global Sensitivity

In the DP framework, Global Sensitivity is the maximum amount that a query result (in this case, mean) can change if one sample is removed from the set. The intuition is that DP tries to give a result such that the contribution of one sample to the result is bounded, so that the privacy loss to that sample from being added to the result is bounded.

Global Sensitivity is not based on the actual values in the current database, but on the worst possible values that might be there at some point. With 100 samples between 0 and 1, a worst case is where $(x_1...x_{99}) = 0$ and $x_{100} = 1$. In this case, the mean will either be 0, if x_{100} is omitted, or $\frac{x_{100}}{100}$, or .01. So the Global Sensitivity S is .01.[1]

There is much more to be said about Global Sensitivity, but now that we have a value, we can explore what the Laplace distribution will be.

[1] If you do the math, you will see that S is the same no matter what values you pick for $(x_1...x_{99})$.

3.2 The Laplace noise function

The zero-mean Laplace distribution is defined as follows: for a possible value x of noise to be added to the true value, the probability of adding that noise value is

$$P = \frac{1}{(2 * b)} * exp(-abs(\frac{(x)}{b}))$$

where b is defined as $\frac{S}{\epsilon}$.

A plot is an easy way to visualize what is happening.

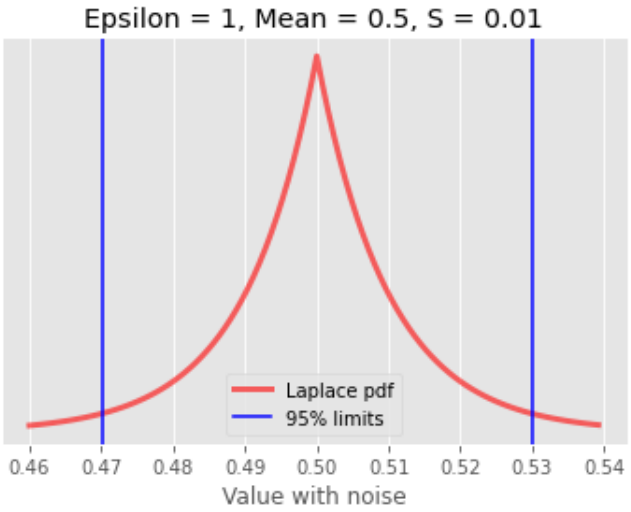

Figure 1: Noisy result of the computation of the mean

While we want to derive the required level of noise (or other harm mitigation) from an assessment of the real potential for harm (rather than a abstract privacy loss based on ϵ), this figure will help to show what DP is attempting to do.

With an epsilon of 1, the noisy result of the query (when the true mean is .5) is between .47 and .53 with 95% probability. The data analyst must decide if this degree of uncertainty renders the result useless, or is still informative. In the particular case where the query was the fraction of systems patched to the latest release, this level of imprecision is probably not important. So the answer is useful. But what harm is DP trying to prevent here? Where does the potential privacy (or confidentiality) loss arise?

3.3 The worst-case assumption

The philosophy of DP states that one must make a worst-case assumption about the prior knowledge of the data intruder. We must assess the potential loss of privacy, independent of what the data intruder knows. So consider the case of

a data intruder that happens to know the actual answer for 99 of the firms, and wants to learn the answer about the final firm. If no noise is added to the answer, then the intruder can easily reverse the computation of the mean and derive the answer for that firm. So DP adds noise to the result to limit the privacy loss.

Assume that the mean of the known 99 values was .5. If the remaining (unknown) value is 0.0, the mean of all the values would be .495. If the remaining value is 1.0, the mean would be .505. (Note that we have just recomputed the Global Sensitivity in this case–the difference is .01.) In these two extreme cases, what would the Laplace distributions be for the noisy answer?

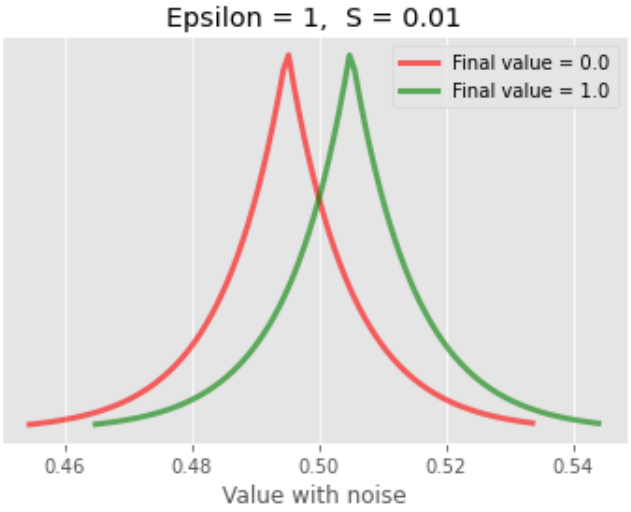

Figure 2: Range of noisy results depending on actual value of the remaining data item.

The data analyst is (as before) facing an 95% certainty of +/- .03 from the actual answer. What can the data intruder make of this situation? The data intruder does not know from which distribution (anywhere between the lowest and highest pictured here) the returned value came. All he sees is a single number. If that number happened to be .5, the value is equally likely to have come from the lowest and the highest alternative, so the intruder has learned nothing. However, if the answer were (for example) .48, it is much more likely that this result was from a distribution from at the lower range of the options. In other words, the intruder cannot guess the true value of the final value, knowing the other 99 values, but they may be able (for some noisy results) be able to guess that the remaining value is "lowish" or "highish". Whether this degree of guess is harmful is not a question of math, but must be answered from the actual context.

However, one obvious question we might ask is whether we need to address

the actual worst case. If the intruder knew all but 2 of the values, he would learn essentially nothing from the range of possible noisy answers. If we allow ourselves to relax the worst case assumption in assessing the potential for actual harm, we may get a more realistic assessment of what a data intruder can actually do, but the mathematical foundation of DP are of no help to us.

4 Addressing the binomial pathology

The previous illustrations showed the distribution of added noise if the actual mean was .5. What if it is 0.0?

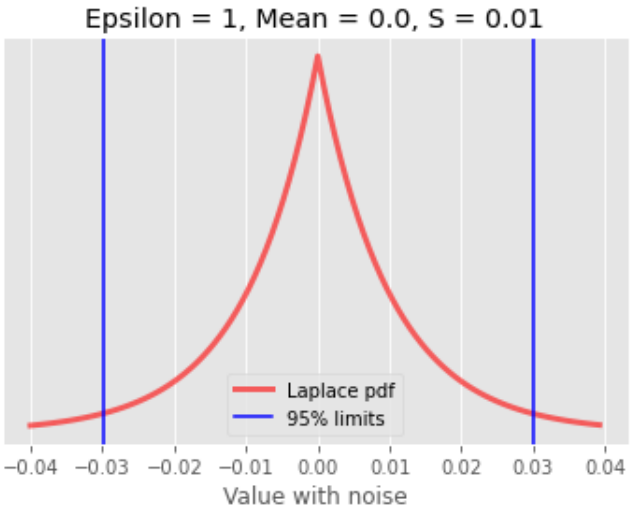

Figure 3: Range of noisy results if the true mean of the 100 values is 0.0.

First, the actual Laplace function may return values less than zero, which all parties (both analyst and intruder) will understand are not possible. The trusted analysist returning the result might choose to convert a value less than zero to zero, which will slightly change what the intruder can surmise. But with this degree of noise, what the intruder can reasonably *guess* is that the actual mean (that is, the fraction of systems that have been patched to the current release) is probably less than .03. He cannot know that the actual value was 0.0, but knowing that it is highly unlikely that the value was above .03 may be enough to cause harm. Would this degree of uncertainty allow any single firm that had contributed data to plausibly claim that while the overall number was really low, they had actually done a good job? Probably not. So it is not clear that adding noise in the DP manner did us any good. But this harm is not what DP claimed to protect us from. This is like the case of smoking and cancer. Some reputational harm may attach to firms of the sort surveyed, whether or not they were in the sample.

5 Making the problem worse

One of the problems with aggregating firm-level data is that there may not be many firms that contribute. The previous example specified 100 firms. What if there were only 10? In that case, the Global Sensitivity S would be 10 times greater, and if ϵ were still 1, the distribution of added noise would look like this.

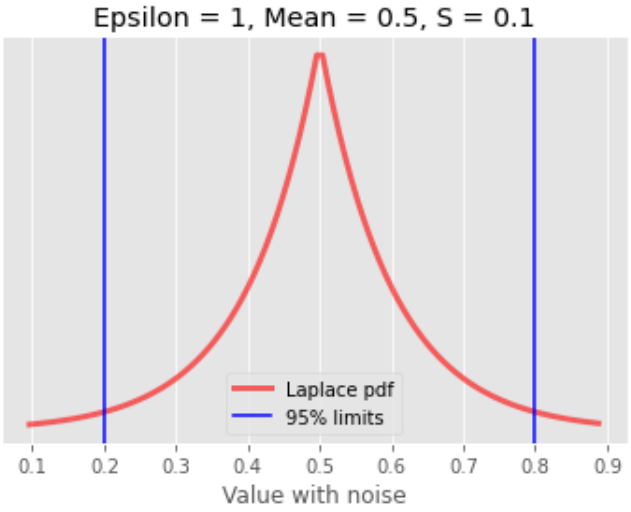

Figure 4: Range of noisy results if the true mean of the 10 values is 0.5.

This amount of noise, given a Global Sensitivity of .1, renders the result essentially useless. What can we do? One obvious answer is increase ϵ. If we increase ϵ to 10, the plot will be exactly the same as Figure 1. This should not be a surprise: if we increase ϵ by 10 and increase the Global Sensitivity by 10, the two changes cancel out. What is different is the view that the data intruder now has.

Because the Global Sensitivity has changed by a factor of 10, the curves representing the Laplace distribution for the minimum and maximum value of the one sample that the worst-case intruder does not know have moved 10 times further apart. In this case, while the intruder cannot guess an exact number for that remaining value, as Figure 5 shows, he can make a very good guess.

The definition of the privacy loss from DP in equation 1 makes clear that there is little protection with $\epsilon = 10$. It specifies that the ratio of the two probabilities (with d_1 differing from d_2 by one record) must differ by no more than e^{ϵ}, which in this case is 22,026. This is a mathematically well-formed way of saying that in the worst case the privacy loss is complete.

This does not mean that adding noise is a useless exercise with small samples. What it means is that we must step away from the strict, worse-case assumption of DP and accept a more pragmatic assessment of the capabilities

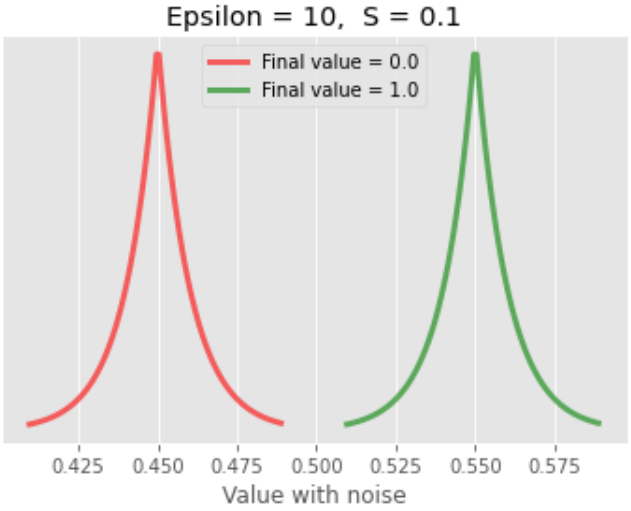

Figure 5: Range of noisy results depending on actual value of the final data item, with 10 samples.

and motivations of the data intruder. We can use the same method to add noise, so the result is nominally consistent with DP, but to assess harm we cannot rely on the concept of privacy loss.

6 Making the situation better

To this point, I have made two observations. First, DP was not designed to mitigate what I called the binomial pathology, and does so poorly. Second, as the sample size (the number of firms) shrinks, we must rely on a pragmatic, not worst-case, analysis of the capabilities and intentions of the intruder to assess potential harm.

But what can we do about the binomial pathology?

6.1 Sampling

One approach which has been used to try to protect individual entries in a database is to sample the database and compute the noisy result across a subset of the sample (in my example, the firms). In this case a firm can try to claim that the result may not apply to them because they may not even have been in the sample. In my simplistic example, would this be a plausible way to sidestep harm?

Sadly, probably not. Here, the sample size works against that claim. If there were 100 firms, and the trusted agent that computes and returns the noisy answer declares that it has used the data from only 90 of them, what

then? Statisticians, when looking at the behavior of firms, can often justify the assumption that the variables that define their behavior are i.i.d. In this case, using a sample of 90 to predict the behavior of a population of 100 is a highly robust statistical assumption. The more firms that are in the data, the stronger the justification for a statistical conclusion that a subsample is a robust predictor of the population. So sampling cannot help us here.

6.2 Change the query

The obvious way out of this dilemma is not to compute the mean, but to devise an alternative query that provides sufficient information for the needs of the data analyst but avoids pitfalls such as the binomial pathology. These alternative queries may also benefit from added noise, but with careful design can avoid dangerous results from low-probability data values.

One kind of query might be some form of quantile. For example, the query might be: "what fraction of the firms have patched more than 50% of their systems." This is essentially a histogram with two bins, and a count for each.

Of course, none of the firms might have patched more than 50% of their systems, so 100 would be in the lower bin, and none of them in the upper bin. Would this outcome represent a harm to an individual firm? None of the firms have patched more than 50%–that fact is known about each individual firm. But that might be a tolerable degree of reputational loss, since all the others are in the same boat. And any one firm could argue that they had done 49% of their systems.

If we want to further limit what the data intruder can know (or guess), we can add noise to the counts for each bin. If there were 50 firms that had patched less than 50% of their systems, the trusted agent might add noise and with some probability report 49 or 51, or with lower probability report 48 or 52.[2] The problem here is low counts. If there is a bin with one firm in it, and we added enough noise that the answer might be two or zero, that is a huge loss in the precision (and utility) of that number. And if we created more bins, the expected number of firms in each firm would be lower, so the degree of uncertainty in the results would go up. If there were only 10 firms in the data, and we split them up into more than a very few bins, the added noise would essentially render the results useless.[3]

There are other ways a query could be posed. We could ask the median of the percent of patched systems, looking across all the firms. If the median is 0.0, then at least half the firms have done nothing, but we know nothing about the other half. Again, we have to assess this query through the lens of the potential

[2]There is a DP method called *exponential* DP that always returns potentially valid answers, but would not return an answer like 49.3. On the other hand, reporting a fractional count is a good way to remind both the data analyst and the data intruder that noise has been added to the result.

[3]There is a DP-compliant scheme called DAWA that uses up part of the privacy loss budget to look at the actual data and construct bin sizes for a subsequent query that best avoids the "small bin count" problem. DAWA may be particularly useful in the case of queries with a small number of records (firms).

harm to the individual firms, and whether the use of DP could further mitigate these potential harms.[4]

6.3 Add noise based on the actual data

If it is necessary to use a query (such as mean) that has a low-probability data disclosure pathology, consider abandoning the logic of DP and adding noise (or more noise) only when the actual data triggers the pathology. Doing this completely steps outside the philosophy of DP, because the fact that additional noise has been added to a particular result (which has to be disclosed) itself reveals important facts about the data. In the example above of the 100 firms, the trusted agent could add additional noise as the true value of the result approaches 0.0. This would prevent a data intruder from making as precise guess, but would still make obvious that the number was unfortunately low.

Once one has committed to a query such as mean, there are no easy ways to mask the pathological outcomes. Refusing to return a result in that case itself reveals something about the data. Adding lots of noise reveals something about the data. Much better to reframe the query if that is possible, given the needs of the legitimate data analyst.

7 Conclusions

The astute reader may have observed that I could have written this paper from a different starting point, with a title such as: "The Hidden Perils of Computing a Mean," and gotten much of the way through the development without even mentioning DP. I chose this course through the material both to introduce the basic ideas of DP, and to point out that there are queries that can lead to harm that DP was not designed to prevent, and prevent only to a limited extent.

While most papers that discuss DP focus on ϵ, perhaps the most important element of a strategy for industry cooperation is to develop queries that provide sufficient utility but are free of low-probability disclosure of firm-specific information. Query design is a critical part of effective use of DP, but a part that is often not sufficiently discussed.

One of the problems that DP faces is to help practitioners map from a value of ϵ to a practical assessment of harm. The approach I have taken here (exploiting the simplicity of my example) is to start with an assessment of harm, and tune ϵ based on that assessment.

The DP framework provides assurance that a number of operations on the data, including forms of composition and post-processing, can be done in a way

[4] Median, if used in a simplistic way, also has a pathology, which we might call the *bimodal* pathology. If half the firms have done nothing and the other half have fully patched their systems, then the median would lie between the 50 0.0 values and the 50 1.0 values. If we consider adding DP noise, and derive the Global Sensitivity that results from removing one value, the resulting median value might jump between 0.0 and 1.0, which would imply such a large Global Sensitivity that the degree of noise needed would render the answer nonsense. Again, the solution is not to ask this exact sort of query.

that continues to provide a known privacy loss. If the decision about harm is based on a pragmatic assessment, even if the query used is nominally compliant with DP, one cannot rely on the DP framework to assess the consequences of further operations on the potential harm.

Adding noise to the result of a query can be an important method to reduce potential harm, even if the amount of noise added is not framed as a form of DP, but in that case there is no mapping from the noise to the formal DP specification. Pragmatic assessment of potential harm is a space that is fraught with failure.

It is critical that the amount of noise that has been added to a result be disclosed, both to inform the legitimate data analyst and as well a possible data intruder. A frustrating kind of harm can occur when an intruder does not understand how noise has limited his ability to draw a conclusion, draws a conclusion that is unjustified, but publicises it anyway, creating publicity that causes reputational harm that ideally would have been prevented by the added noise. Showing some form of error bars on an answer may be a way to make the point forcefully. However, error bars must not be presented in a way that reveals anything further about the actual data.