IEOR 8100 / COMS 6998: Foundations of Data Privacy **Fall 2023**

Lecture 3: AboveNoisyThreshold & SparseVector

Lecturer: Rachel Cummings *Scribe: Rachel Cummings*

Today we're going to talk about the SparseVector Mechanism, which allows ϵ to degrade like $O(\ln(k)\epsilon)$, which means you can handle exponentially many more queries for the same fixed privacy budget, compared to Basic Composition. The catch is that you can't actually *answer* all these queries, but instead can identify a small number of "interesting" or "important" queries, and answer them.

The mechanism works by taking in a long stream of queries, adding Laplace noise to each answer, comparing each (noisy) answer against a noisy (Laplace) threshold, and only outputting the answer to those queries with values above the threshold.

This is especially useful if the analyst believes that only a small number of the queries will have large (i.e., interesting) answers. For example, a large set of queries of the form "What's the correlation between A and B?" where most things are close uncorrelated.

We'll start with just a simplified version of this called *AboveNoisyThreshold* [DNR+09, DNPR10], and then extend to the full SparseVector framework.

1 AboveNoisyThreshold

AboveNoisyThreshold (ANT) takes a stream of queries and halts after finding the first query with noisy answer above a noisy threshold. Specifically, it takes in a database x, a stream of sensitivity-1 queries, a threshold T, and privacy parameter ϵ. It will report back whether the noisy answer to each query was above or below a noisy version of the threshold, and it halted the first time it found an above-threshold query.

Algorithm 1 AboveNoisyThreshold($x, \{f_i\}, T, \epsilon$) [DNR+09, DNPR10]

Input: database x, adaptively chosen stream of sensitivity-1 queries $\{f_i\}$, threshold T, privacy parameter ϵ

Let $\hat{T} = T + Lap(\frac{2}{\epsilon})$
for each query f_i **do**
  Let $v_i = Lap(\frac{4}{\epsilon})$
  if $f_i(x) + v_i \geq \hat{T}$ **then**
    output $a_i = \top$
    Halt
  else
    output $a_i = \bot$
  end if
end for

Theorem 1 ([DNR+09, DNPR10] (also Theorem 3.23 in the textbook [DR14])). *AboveNoisyThreshold is* $(\epsilon, 0)$*-differentially private.*

We also want to make sure that the mechanism identifies the "right query". For that we'll need an accuracy guarantee. Note that we can't use our previous accuracy notions which say the answers provided by the mechanism are close to the true answers because AboveNoisyThreshold does not produce numeric answers.

Definition 2 (Accuracy). *A mechanism that outputs a stream of answers* $\{a_i\} \in \{\perp, \top\}^*$ *to a stream of* k *queries* $\{f_i\}$ *is* (α, β)*-accurate with respect to a threshold* T *if, with probability at least* $1 - \beta$*, the mechanism does not halt before* f_k*, and*

$$\forall a_i = \top : f_i(x) \geq T - \alpha$$
$$\forall a_i = \perp : f_i(x) \leq T + \alpha$$

This definition requires that with high probability, the mechanism produces an approximately correct output for all k queries.

Theorem 3 ([DNR+09, DNPR10] (also Theorem 3.24 in the textbook [DR14])). *For any sequence of* k *sensitivity-1 queries* $f_1, \ldots, f_k$ *s.t.* $|\{i < k : f_i(x) \geq T - \alpha\}| = 0$*, then AboveNoisyThreshold is* (α, β)*-accurate for any* $\beta > 0$ *and*

$$\alpha = \frac{8(\ln(k) + \ln(\frac{2}{\beta}))}{\epsilon}.$$

Note that this quantifier $|\{i < k : f_i(x) \geq T - \alpha\}| = 0$ requires that the only query close to being above threshold is possibly the last one. Without this condition, the algorithm would be required to halt before the k^{th} query with high probability, so it couldn't possibly satisfy the accuracy guarantee.

2 SparseVector Mechanism

Now we'll extend this to the Sparse Vector Mechanism, which can handle multiple above-threshold queries without halting, and it can produce numerical answers to those queries. This algorithm is effectively an adaptive composition of multiple calls to ANT and multiple calls to the Laplace Mechanism. It's going to call ANT, which will run until it finds a query that is above threshold. Then SparseVector will call the Laplace Mechanism to provide a noisy answer to the query that was found. Then it will repeat this process with a new call to ANT.

The SparseVector mechanism was developed in [DNR+09] and refined to its current form in [HR10, DNPR10]. It takes as input a database x, an adaptively chosen stream of sensitivity-1 queries $\{f_i\}$, a threshold T, a total number of numeric answers c, and privacy parameters (ϵ, δ). It outputs a stream of answers $\{a_i\} \in (\mathbb{R} \cup \{\perp\})^*$.

Now let's see the full SparseVector algorithm (Algorithm 2).

Algorithm 2 SparseVector$(x, \{f_i\}, T, c, \epsilon, \delta)$

SparseVector$(x, \{f_i\}, T, c, \epsilon, \delta)$:
Let $\epsilon_1 = \frac{8}{9}\epsilon$ and let $\epsilon_2 = \frac{2}{9}\epsilon$
if $\delta = 0$ **then**
    Let $\sigma(\epsilon) = \frac{2c}{\epsilon}$
else
    Let $\sigma(\epsilon) = \frac{\sqrt{32c \ln(2/\delta)}}{\epsilon}$
end if
Let $\hat{T}_0 = T + \text{Lap}(\sigma(\epsilon_1))$
Let count $= 0$
for each query f_i **do**
    Let $v_i = \text{Lap}(2\sigma(\epsilon_1))$
    if $f_i(x) + v_i \geq \hat{T}_{\text{count}}$ **then**
        Output $a_i = f_i(x) + \text{Lap}(\sigma(\epsilon_2))$
        Update count $=$ count $+ 1$ and $\hat{T}_{\text{count}} = T + \text{Lap}(\sigma(\epsilon_1))$
    else
        Output $a_i = \perp$
    end if
    if count $\geq c$ **then**
        Halt
    end if
end for

Now let's take a closer look at what's going on with this mechanism. The middle part looks a whole lot like AboveNoisyThreshold (ANT). SparseVector (SV) works by repeated calls to ANT as a subroutine up to c times, until our counter reaches the pre-set limit c. Instead of halting after finding an above-threshold query, SV calls the Laplace Mechanism as a subroutine to output a noisy answer to that query.

Notice that we re-draw fresh noise for every call to the Laplace Mechanism and ANT, so SV is really just an adaptive composition of these two mechanisms. We allocate our overally privacy budget ϵ between these two mechanisms, where ϵ_1 is our ANT privacy budget and ϵ_2 is our Laplace Mechanism privacy budget.

Depending on whether our overall privacy goal is $(\epsilon, 0)$-differential privacy or (ϵ, δ)-differential privacy, we'll have to set parameters within these subroutines differently. If we want $(\epsilon, 0)$-differential privacy, we'll end up using Basic Composition[1], so we'll set our privacy parameters so they sum to ϵ. If we want (ϵ, δ)-differential privacy, then we can use Advanced Composition, and we'll set our parameters according to the Corollary that we saw last time.

[1] Note that we only proved Basic Composition for non-adaptive mechanisms, but the result also holds for adaptive mechanisms. The Laplace Mechanism is used adaptively based on the results of the ANT mechanism.

2.1 SparseVector Privacy

Theorem 4 ([DNR+09, DNPR10])**.** *SparseVector is (ϵ, δ)-differentially private.*

Proof. Case $\delta = 0$:
We first consider the case where $\delta = 0$. SV consists of c runs of ANT, where each run is $(\frac{8}{9c}\epsilon, 0)$-differentially private, and c runs of the Laplace Mechanism, where each run is $(\frac{1}{9c}\epsilon, 0)$-differentially private. Then it will be straightforward to see through Basic Composition that SV is overall $(\epsilon, 0)$-differentially private. All that remains is to prove these claims about the subroutines.

Recall that ANT added $\text{Lap}(2/\epsilon)$ noise to the threshold and $\text{Lap}(4/\epsilon)$ noise to the query for overall ϵ-differential privacy. The subroutine in SV adds

$$\text{Lap}(\sigma(\epsilon_1)) = \text{Lap}\left(\frac{2c}{\epsilon_1}\right) = \text{Lap}\left(\frac{2c}{\frac{8}{9}\epsilon}\right) = \text{Lap}\left(\frac{2}{\frac{8}{9c}\epsilon}\right)$$

noise to the threshold and

$$\text{Lap}(2\sigma(\epsilon_1)) = \text{Lap}\left(\frac{4}{\frac{8}{9c}\epsilon}\right)$$

noise to the query. Therefore, each call to ANT is $(\epsilon', 0)$-differentially private for $\epsilon' = \frac{8}{9c}\epsilon$.

Recall that the Laplace Mechanism adds $\text{Lap}(\Delta f/\epsilon) = \text{Lap}(1/\epsilon)$ noise for our sensitivity-1 queries. The subroutine in SV adds

$$\text{Lap}(\sigma(\epsilon_2)) = \text{Lap}\left(\frac{2c}{\epsilon_2}\right) = \text{Lap}\left(\frac{2c}{\frac{2}{9}\epsilon}\right) = \text{Lap}\left(\frac{1}{\frac{1}{9c}\epsilon}\right)$$

noise. Therefore, each call to the Laplace Mechanism is $(\epsilon', 0)$-differentially private for $\epsilon' = \frac{1}{9c}\epsilon$.

Basic Composition gives that the overall privacy guarantee is:

$$c\left(\frac{8}{9c}\epsilon\right) + c\left(\frac{1}{9c}\epsilon\right) = \frac{8}{9}\epsilon + \frac{1}{9}\epsilon = \epsilon$$

so SV is $(\epsilon, 0)$-differentially private.

Case $\delta > 0$:
Now we address the case where $\delta > 0$. We will follow a similar structure, where we prove privacy guarantees of each subroutine, and then prove overall privacy through Advanced Composition this time.

Each run of ANT is $(\frac{8}{9\sqrt{8c\ln(2/\delta)}}\epsilon, 0)$-differentially private. Our subroutine adds

$$\text{Lap}(\sigma(\epsilon_1)) = \text{Lap}\left(\frac{\sqrt{32c\ln(2/\delta)}}{\epsilon_1}\right) = \text{Lap}\left(\frac{\sqrt{32c\ln(2/\delta)}}{\frac{8}{9}\epsilon}\right) = \text{Lap}\left(\frac{2}{\frac{8}{9\sqrt{8c\ln(2/\delta)}}\epsilon}\right)$$

noise to the threshold and

$$\mathrm{Lap}(2\sigma(\epsilon_1)) = \mathrm{Lap}\left(\frac{4}{\frac{8}{9\sqrt{8c\ln(2/\delta)}}\epsilon}\right)$$

noise to the answer. Therefore, each call to ANT is $(\epsilon', 0)$-differentially private for $\epsilon' = \frac{8}{9\sqrt{8c\ln(2/\delta)}}\epsilon$.

Recall the corollary of advanced composition from last time.

Corollary 5 (Advanced Composition). *If $\mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \to \mathcal{R}^k$ is a k-fold adaptive composition of $(\epsilon'/\sqrt{8k\ln(1/\delta')}, 0)$-differentially private mechanisms, then $\mathcal{M}$ is (ϵ', δ')-differentially private.*

Applying this result with $k = c, \epsilon' = \frac{8}{9}\epsilon, \delta' = \frac{\delta}{2}$, we see that these runs of ANT together are $(\frac{8}{9}\epsilon, \frac{\delta}{2})$-differentially private.

Each run of the Laplace Mechanism is $(\frac{1}{9\sqrt{8c\ln(2/\delta)}}\epsilon, 0)$-differentially private, and instantiating Corollary 5 with $\epsilon' = \frac{1}{9}\epsilon$ and $\delta' = \frac{\delta}{2}$ gives that these c runs of the Laplace Mechanism together are $(\frac{1}{9}\epsilon, \frac{\delta}{2})$-DP. Basic Composition of these two subroutines gives that SV is $(\frac{8}{9}\epsilon + \frac{1}{9}\epsilon, \frac{\delta}{2} + \frac{\delta}{2})$-differentially private, i.e., (ϵ, δ)-differentially private. □

2.2 SparseVector Accuracy

Before discussing the accuracy of SparseVector, let us recall the accuracy theorems for the two sub-routines used in the SparseVector algorithm: AboveNoisyThreshold and the Laplace Mechanism. Recall also that ANT outputs binary answers in $\{\perp, \top\}$, so the ANT accuracy corresponds to accurate comparison against a threshold.

Theorem 6 (ANT Accuracy). *For any sequence of k sensitivity-1 queries $\{f_1, f_2, \ldots f_k\}$ satisfying $|\{i < k : f_i(x) > T - \alpha\}| = 0$, then AboveNoisyThreshold is (α, β_{ANT})-accurate for*

$$\alpha = \frac{8(\log k + \log(2/\beta_{ANT}))}{\epsilon_{ANT}}.$$

Theorem 7 (Laplace Accuracy). *Let $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$ and let $\mathcal{M}_L(x, f, \epsilon_L)$ be the Laplace mechanism, then $\forall \beta_L \in (0, 1]$,*

$$\Pr\left[\|f(x) - M_L(x, f, \epsilon_L)\|_\infty \geq \log\left(\frac{k}{\beta_L}\right)\left(\frac{\Delta f}{\epsilon_L}\right)\right] \leq \beta.$$

To measure the accuracy of SparseVector, we have to modify our notion of accuracy from ANT to a setting where the algorithm also outputs numeric answers for some queries.

Definition 8 (Numeric Accuracy). *A mechanism that outputs a stream of answers $\{a_i\} \in \mathbb{R} \cup \{\perp\}$ to a stream of k queries $\{f_i\}$ is (α, β)-accurate with respect to a threshold T if with probability at least $1 - \beta$, the mechanism does not halt before f_k, and*

$$\forall a_i \in \mathbb{R} \quad |f_i(x) - a_i| < \alpha \text{ and}$$
$$\forall a_i = \perp, \quad f_i(x) \leq T + \alpha.$$

The first condition above is the same additive accuracy notion that we use with numeric outputs, e.g., for the Laplace mechanism: the answer produced by the mechanism should be within an additive α of the true answer to the query on the database. The second condition comes from ANT accuracy, where we only output whether a query answer is above or below a threshold: if the mechanism produces $\perp$, then the true query value should not be more than α above the threshold.

Theorem 9 (Sparse Vector Accuracy [DNR+09, DNPR10])**.** *For any sequence of k sensitivity-1 queries $\{f_1, f_2, \ldots f_k\}$ satisfying $|\{i<k \mid f_i(x) \geq T-\alpha\}| < c$, SparseVector is (α, β)-numeric accurate for*

$$\alpha = \begin{cases} \frac{9c\left(\log k+\log\left(\frac{4c}{\beta}\right)\right)}{\epsilon}, & \text{if } \delta = 0 \\ \frac{9(\log k+\log(4c/\beta))\sqrt{8c\log(2/\delta)}}{\epsilon}, & \text{if } \delta > 0 \end{cases}.$$

Note that the condition $|\{i<k \mid f_i(x) \geq T-\alpha\}| < c$ plays the same role as the analogous condition in ANT accuracy. Without this condition, the algorithm *should* provide numeric answers to more than c queries, which means that it *should* halt before the k-th query, which would violate the "no early halting" accuracy condition in Definition 8.

Proof. We will separately show that the two conditions required for accuracy are satisfied: (1) $f_i(x) \leq T + \alpha$ when $a_i = \perp$, and (2)$|f_i(x) - a_i| \leq \alpha$ when $a_i \in \mathbb{R}$.

Condition 1: $f_i(x) \leq T + \alpha$ when $a_i = \perp$.
If $\delta = 0$, instantiate Theorem 6 with $\beta_{ANT} = \frac{\beta}{2c}$ and $\epsilon_{ANT} = \frac{8}{9c}\epsilon$, to get that each of the c calls to ANT is $(\alpha_1, \frac{\beta}{2c})$-accurate for

$$\alpha_1 = \frac{9c\left(\log k + \log\left(\frac{4c}{\beta}\right)\right)}{\epsilon}.$$

If $\delta > 0$, instantiate Theorem 6 with $\beta_{ANT} = \beta/2c$ and $\epsilon_{ANT} = \frac{8}{9\sqrt{8c\log\left(\frac{2}{\delta}\right)}}\epsilon$, to get that each of the c runs of ANT is $(\alpha, \beta/2c)$-accurate for

$$\alpha_1 = \frac{9\left(\log k + \log\left(\frac{4c}{\beta}\right)\right)\sqrt{8c\log(2/\delta)}}{\epsilon}.$$

Condition 2: $|f_i(x) - a_i| \leq \alpha$ when $a_i \in \mathbb{R}$.
Instantiate Theorem 7 with $k = 1$, $\Delta f = 1$, $\beta_L = \beta/2c$. If $\delta = 0$, use $\epsilon_L = \frac{1}{9c}\epsilon$, and if $\delta > 0$, then use $\epsilon_L = \frac{1}{9\sqrt{8c\log(2/\delta)}}\epsilon$. This ensures that $\Pr[|f_i(x) - a_i| \geq \alpha_2] < \frac{\beta}{2c}$ for each $a_i \in \mathbb{R}$, for

$$\alpha_2 = \begin{cases} \frac{9c\log\left(\frac{2c}{\beta}\right)}{\epsilon}, & \text{if } \delta = 0 \\ \frac{9\log\left(\frac{2c}{\beta}\right)\sqrt{8c\log(2/\delta)}}{\epsilon}, & \text{if } \delta > 0 \end{cases}.$$

Note that $\alpha := \alpha_1 \geq \alpha_2$, for any value of δ_1. This can be seen by direct comparison on the two terms. Thus each of the two c-subroutines is $(\alpha, \frac{\beta}{2c})$-accurate. Taking a union bound over all $2c$ failure probabilities gives that SparseVector is (α, β)-accurate. □

3 SparseVector and the Reusable Holdout

Adaptive Data Analysis or the process of testing multiple adaptively chosen hypotheses reporting only the significant results, is problematic as the choice of future hypotheses to test are a function of the data, and the appropriate correction for multiple hypotheses is typically not performed. See Figure 1 for a hilarious but unfortunately (mostly) realistic depiction.

One way to overcome this is to pre-register all the hypotheses that are being evaluated before collecting data. However, this can be burdensome and is not always feasible.

It is common in fields like machine learning to split datasets into training and holdout sets. The training set is used for data exploration and to learn/fit model parameters, and the holdout set is used to validate the learned parameters or to adjust hyper-parameters of the method. However, once this set has been used to test hypothesis, it is no longer a fresh sample from the underlying population, because now all future hypotheses are functions of these data, and repeated use of this holdout would amount to adaptive data analysis.

Recent work [DFH$^+$15c, DFH$^+$15b, DFH$^+$15a] showed that differential privacy enabled the reuse of holdout datasets while still maintaining statistical validity. This work provided the Thresholdout algorithm, which is based on SparseVector.

3.1 Thresholdout

Consider a dataset $\mathcal{S} = \{x_1, \dots x_n\}$ where the data points are sampled i.i.d. from P, which is an unknown probability distribution over a data universe $\mathcal{X}$. Linear queries on such a dataset are of the form $\phi : \mathcal{X} \to [0, 1]$. We can define the expected value of this query on a dataset $\mathcal{S}$ as: $\mathbb{E}_{\mathcal{S}}[\phi] = \frac{1}{n} \sum_{i=1}^{n} \phi(x_i)$, which is a sensitivity-$\frac{1}{n}$ query. We can also define the value of this query on a population P as: $P[\phi] = \mathbb{E}_{x \sim P}[\phi(x)]$.

The goal of data analysis is to find a hypothesis ϕ for which $P[\phi]$ is close to $\mathbb{E}_{\mathcal{S}}[\phi]$. However, since the analyst does not have access to the true underlying population P, she will randomly partition her dataset $\mathcal{S}$ into a training set $\mathcal{S}_t$ and holdout set $\mathcal{S}_h$. She will treat $\mathcal{S}_h$ as a fresh sample drawn from P, and thus with high probability (assuming a sufficiently large sample), her findings on $\mathcal{S}_h$ will generalize to the underlying population. Specifically, she will train models on $\mathcal{S}_t$ using any method she wishes to identify a candidate hypothesis ϕ, and will then test whether $\mathbb{E}_{\mathcal{S}_h}[\phi]$ is close to $\mathbb{E}_{\mathcal{S}_t}[\phi]$. If it is, she will halt and claim a discovery, and if not, she will return to her test set $\mathcal{S}_t$ to search for a new hypothesis ϕ'.

To ensure statistical validity over the testing of multiple hypotheses, she will access her holdoutset set $\mathcal{S}_h$ only through a differentially private mechanism. Specifically, she will use an instantiation of SparseVector, with queries of the form: $f_\phi = |\mathbb{E}_{\mathcal{S}_h}[\phi] - \mathbb{E}_{\mathcal{S}_t}[\phi]|$ to identify whether this error is large. The formal presentation (following notation of [DFH$^+$15c]) is presented in Algorithm 3.

The main result in the works of [DFH$^+$15c, DFH$^+$15b, DFH$^+$15a] is that Thresholdout is differentially private and ensures statistical validity of the findings.

Theorem 10. *Thresholdout is* $(\frac{B}{\tau n}, 0)$*-differentially private, or* $\left(\frac{\sqrt{8B\log(2/\delta)}}{Tn}, \delta\right)$*-differentially*

Algorithm 3 Thresholdout($S_t, S_h, T, \tau, B, \{\phi_i\}$)

Input: training set S_t, holdout set S_h, threshold T, tolerance τ, budget B, stream of adaptively chosen hypotheses $\{\phi_i\}$
Set $\hat{T} = T + \mathit{Lap}(4\tau)$
for each function $\phi_i : \mathcal{X} \to [0,1]$ **do**
  if $B < 1$ **then**
    Output $\perp$
  else
    Sample $\xi \sim \mathit{Lap}(2\tau)$, $\gamma \sim \mathit{Lap}(4\tau)$, $\eta \sim \mathit{Lap}(8\tau)$
    if $|\mathbb{E}_{S_h}[\phi] - \mathbb{E}_{S_t}[\phi]| > \hat{T} + \eta$ **then**
      Output $a_i = \mathbb{E}_{S_h}[\phi] + \xi$
      $B = B - 1, \hat{T} = T + \gamma$
    else
      Output $a_i = \mathbb{E}_{S_t}[\phi]$
    end if
  end if
end for

private for any $\delta > 0$. For any adaptively chosen queries $\{\phi_1, \dots \phi_m\}$, Thresholdout satisfies that $\forall i \in [m]$ and $\forall t > 0$,

$$\Pr[|a_i - P[\phi_i]| > T + (t+1)\tau] < \beta + \exp(-t/8).$$

References

[DFH+15a] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. Generalization in adaptive data analysis and holdout reuse. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS '15, pages 2350–2358, 2015.

[DFH+15b] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. Preserving statistical validity in adaptive data analysis. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing*, STOC '15, pages 117–126, 2015.

[DFH+15c] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. The reusable holdout: Preserving validity in adaptive data analysis. *Science*, 349(6248):636–638, 2015.

[DNPR10] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. Differential privacy under continual observation. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, STOC '10, pages 715–724, 2010.

[DNR+09] Cynthia Dwork, Moni Naor, Omer Reingold, Guy N. Rothblum, and Salil Vadhan. On the complexity of differentially private data release: Efficient algorithms and hardness results. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, STOC '09, pages 381–390, 2009.

[DR14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(34):211–407, 2014.

[HR10] Moritz Hardt and Guy N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, FOCS '10, pages 61–70, 2010.

Figure 1: `https://xkcd.com/882/` Significance