STATS/BIODS 352: Practical Privacy-preservation with Differential Privacy and OpenDP

James Honaker
james@hona.kr

Meta and School of Engineering & Applied Sciences
Harvard University

April 7, 2022

- **Privacy Attacks**
  - Linkage Attacks
  - Reconstruction Attacks
- **Differential Privacy**
  - Definition
  - Adding Noise (Laplace mechanism)

– WEEK 2 –

- **More Foundations**
  - Choosing Privately (Exponential mechanism)
  - Differentially Private Optimization (DP-SGD)
- **Implementations**
  - OpenDP (Open source library of DP components for statistical releases)
  - Opacus (Open source DP optimization for PyTorch ML)

Amelia

Zelig

Zelig

PSI (Ψ)

Private data Sharing Interface

SmartNoise

OpenDP

OpenDP

Privacy Attacks

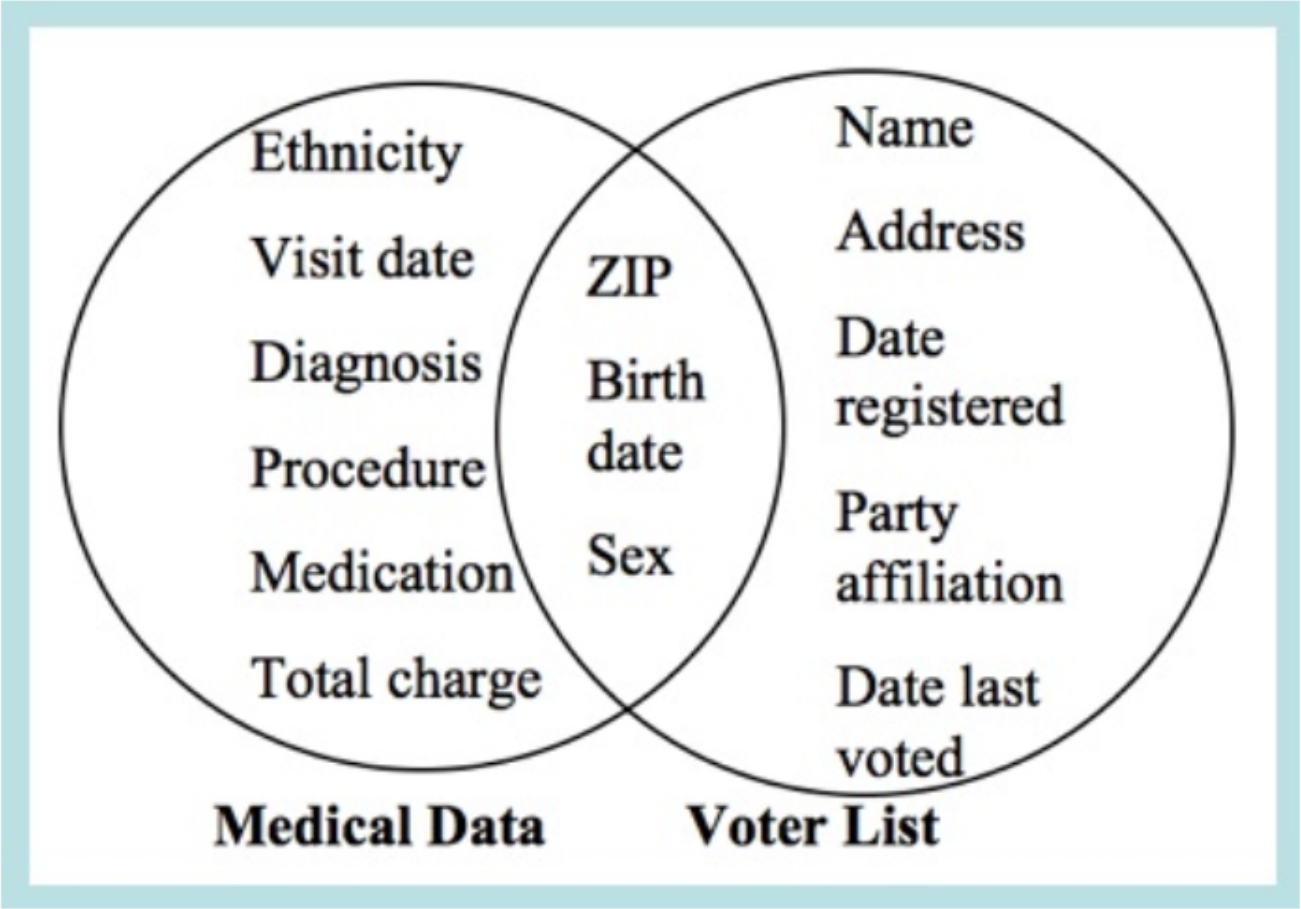
PII as Privacy Preservation

- A common technique for dealing with sensitive data is to strip out *personally identifiable information* (PII) and consider the data anonymized.
- For example, HIPAA safe harbor.

PII as Privacy Preservation

- A common technique for dealing with sensitive data is to strip out *personally identifiable information* (PII) and consider the data anonymized.
- For example, HIPAA safe harbor.
- Anonymization techniques for data releases are generally open to reidentification attacks (Sweeney 1997, 2000, Narayanan & Shimatikov 2008);

Linkage Attack



[Sweeney `97]

Aggregation as Privacy Preservation

- Data owners often release aggregate statistics under the assumption that the crowd provides anonymity.

Aggregation as Privacy Preservation

- Data owners often release aggregate statistics under the assumption that the crowd provides anonymity.
- aggregated statistics can not have any privacy guarantee (Dinur and Nissim 2003) - fingerprinting;

Aggregation as Privacy Preservation

- Data owners often release aggregate statistics under the assumption that the crowd provides anonymity.
- aggregated statistics can not have any privacy guarantee (Dinur and Nissim 2003) - fingerprinting;
- statistical estimates can leak individual information (Ullman and Steinke 2013) - time variance.
- the release of gradients from optimization steps (e.g., SGD) can enable reconstruction of the underlying data.

Difference Attack

- Let's say the mean age of 21 people in this presentation is 29.95

Difference Attack

- Let's say the mean age of 21 people in this presentation is 29.95
- Then say the mean age of 20 people watching this lecture is 29.10

Difference Attack

- Let's say the mean age of 21 people in this presentation is 29.95
- Then say the mean age of 20 people watching this lecture is 29.10
- Then my age is (29.95 - 29.10)*20 = 47.

Difference Attack

- Let's say the mean age of 21 people in this presentation is 29.95
- Then say the mean age of 20 people watching this lecture is 29.10
- Then my age is (29.95 - 29.10)*20 = 47.
- Sophisticated difference attacks are increasingly possible as:
  - more statistics are released of the data
  - statistics are released repeatedly over time
  - or users can generate their own queries
  - machine learning models overfit or memorize data

Regression Based Reconstruction Attack

$$a_j = x_1 d_{1,j} + x_2 d_{2,j} + \ldots + x_p d_{p,n} + \epsilon_i$$

Here:

p	is the number of people in the database
n	is the number of queries
i	is a person index
j	is query index
a_j	is j-th query release
$d_{i,j}$	is a $\{0,1\}$-indicator $i \in S - j$
x_h	is h's sensitive data
ϵ_i	is the noise/error of the i-th query
r_i	is the residual/error of the i-th prediction

Regression Based Reconstruction Attack

$$
\begin{aligned}
a_j &= x_1 d_{1,j} + x_2 d_{2,j} + \ldots + x_p d_{p,n} + \epsilon_i \\
7 &= 1{\cdot}1 + 0{\cdot}1 + 1{\cdot}0 + 0{\cdot}0 + \ldots + 0{\cdot}1 + 2 \\
4 &= 1{\cdot}0 + 0{\cdot}1 + 1{\cdot}1 + 0{\cdot}1 + \ldots + 0{\cdot}1 + (-1) \\
6 &= 1{\cdot}0 + 0{\cdot}0 + 1{\cdot}0 + 0{\cdot}1 + \ldots + 0{\cdot}0 + 3
\end{aligned}
$$

Here:

p	is the number of people in the database
n	is the number of queries
i	is a person index
j	is query index
a_j	is j-th query release
$d_{i,j}$	is a $\{0,1\}$-indicator $i \in S - j$
x_h	is h's sensitive data
ϵ_i	is the noise/error of the i-th query
r_i	is the residual/error of the i-th prediction

Regression Based Reconstruction Attack

$$
\begin{aligned}
a_j &= x_1 d_{1,j} + x_2 d_{2,j} + \ldots + x_p d_{p,n} + \epsilon_i \\
7 &= 1{\cdot}1 + 0{\cdot}1 + 1{\cdot}0 + 0{\cdot}0 + \ldots + 0{\cdot}1 + 2 \\
4 &= 1{\cdot}0 + 0{\cdot}1 + 1{\cdot}1 + 0{\cdot}1 + \ldots + 0{\cdot}1 + (-1) \\
6 &= 1{\cdot}0 + 0{\cdot}0 + 1{\cdot}0 + 0{\cdot}1 + \ldots + 0{\cdot}0 + 3
\end{aligned}
$$

Here:

p	is the number of people in the database
n	is the number of queries
i	is a person index
j	is query index
a_j	is j-th query release
$d_{i,j}$	is a $\{0,1\}$-indicator $i \in S - j$
x_h	is h's sensitive data
ϵ_i	is the noise/error of the i-th query
r_i	is the residual/error of the i-th prediction

Regression Based Reconstruction Attack

$$a_j = \hat{x}_1 d_{1,j} + \hat{x}_2 d_{2,j} + \ldots + \hat{x}_p d_{p,n} + r_i$$

Here:

p	is the number of people in the database
n	is the number of queries
i	is a person index
j	is query index
a_j	is j-th query release
$d_{i,j}$	is a $\{0,1\}$-indicator $i \in S - j$
x_h	is h's sensitive data
ϵ_i	is the noise/error of the i-th query
r_i	is the residual/error of the i-th prediction

Regression Based Reconstruction Attack

$$a_j = \hat{x}_1 d_{1,j} + \hat{x}_2 d_{2,j} + \ldots + \hat{x}_p d_{p,n} + r_i$$

$$7 = 0.92 \cdot 1 + 0.11 \cdot 1 + 1.07 \cdot 0 + -0.08 \cdot 0 + \ldots + 0.07 \cdot 1 + 5.71$$

$$4 = 0.92 \cdot 0 + 0.11 \cdot 1 + 1.07 \cdot 1 + -0.08 \cdot 1 + \ldots + 0.07 \cdot 1 + 2.31$$

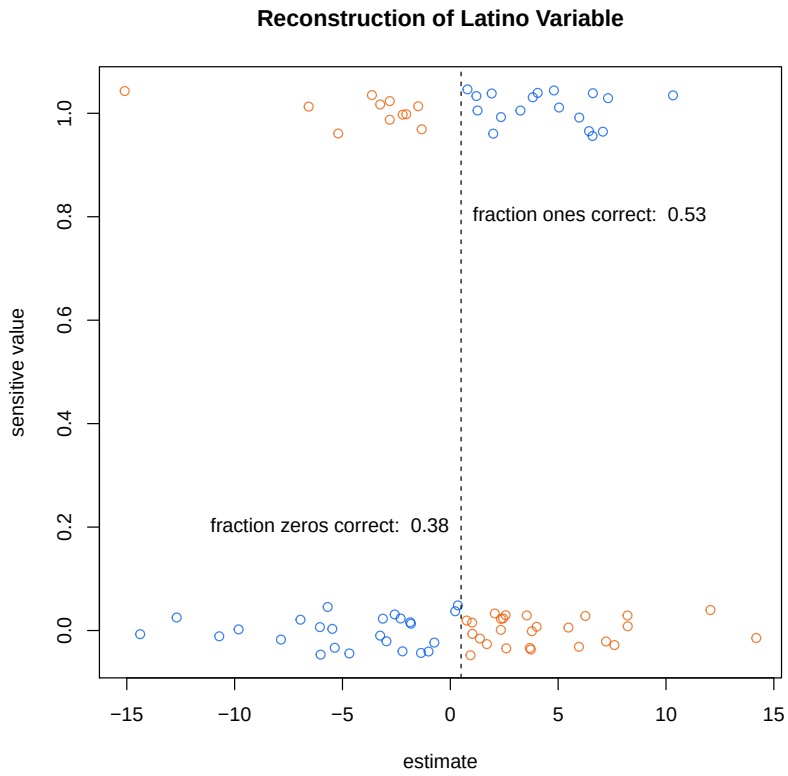
$$6 = 0.92 \cdot 0 + 0.11 \cdot 0 + 1.07 \cdot 0 + -0.08 \cdot 1 + \ldots + 0.07 \cdot 0 - 1.04$$

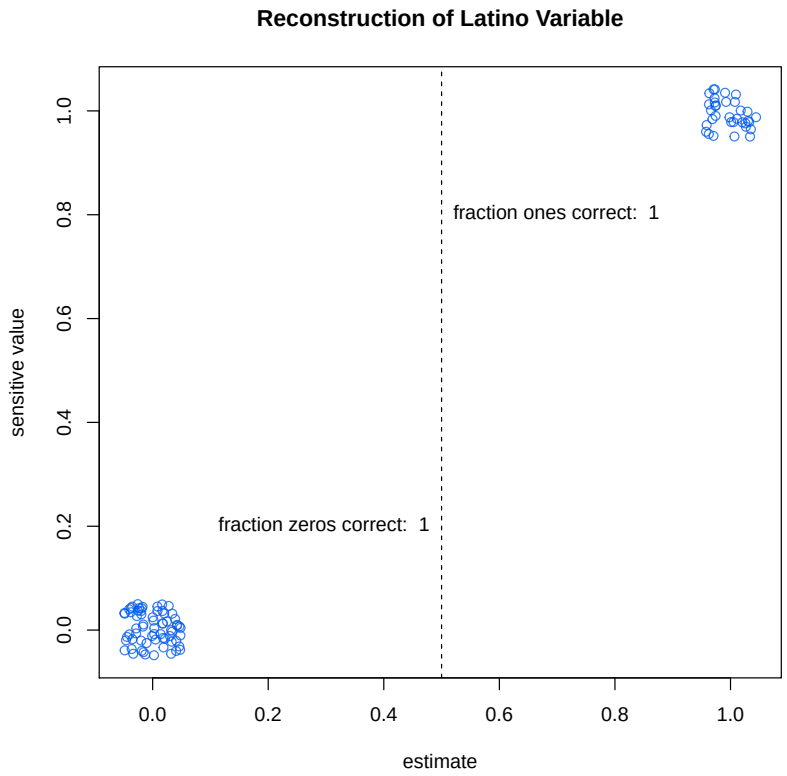
Here:

p	is the number of people in the database
n	is the number of queries
i	is a person index
j	is query index
a_j	is j-th query release
$d_{i,j}$	is a $\{0,1\}$-indicator $i \in S - j$
x_h	is h's sensitive data
ϵ_i	is the noise/error of the i-th query
r_i	is the residual/error of the i-th prediction

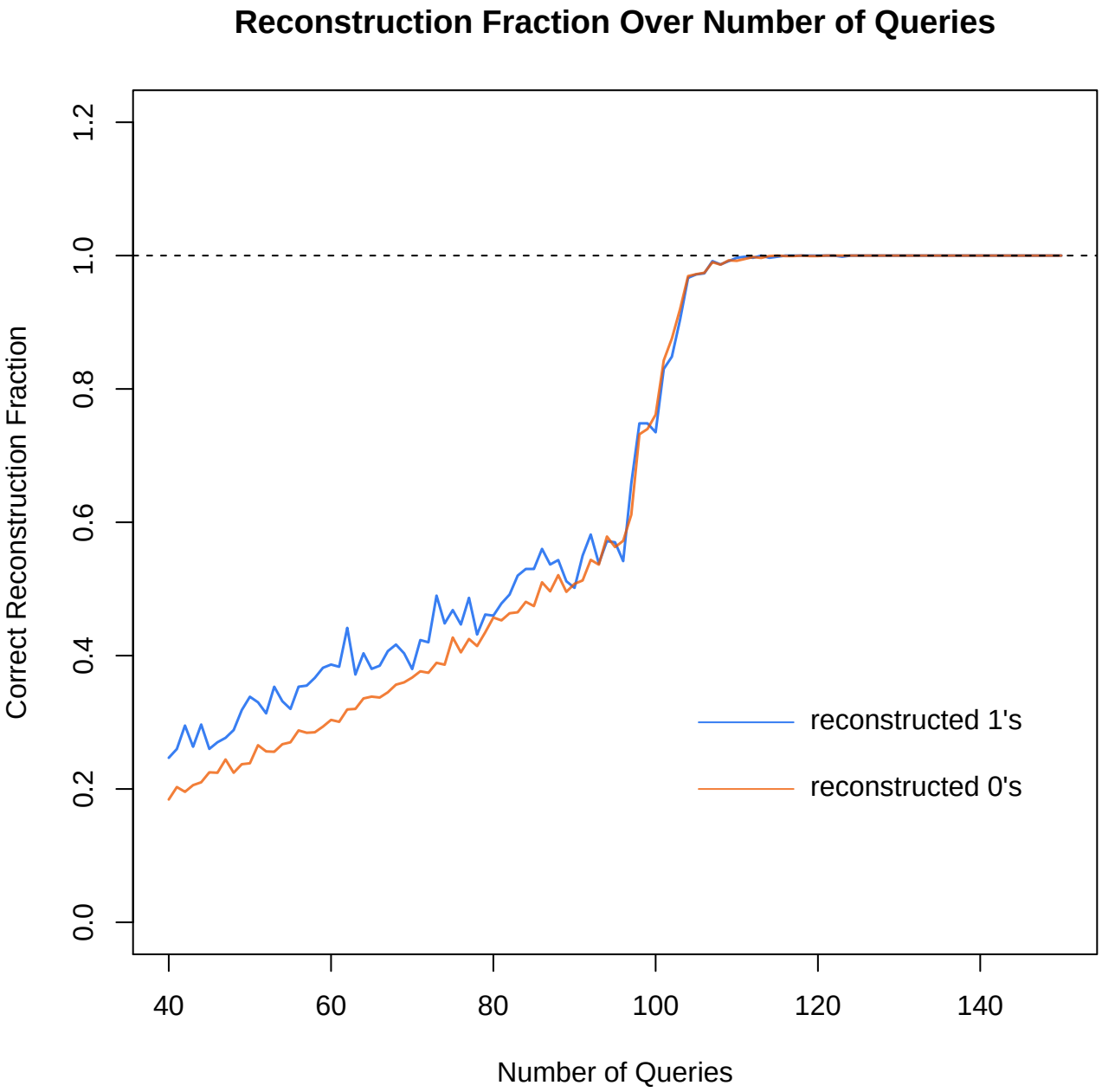
Example

From `regressionAttack.r`:

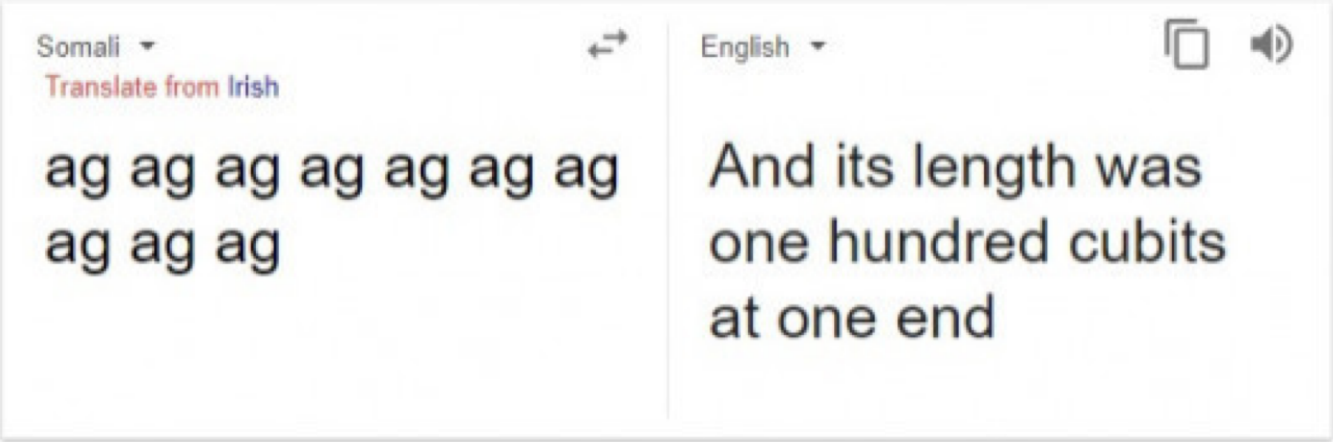




Example: Normal Errors



Memorization and Membership attacks



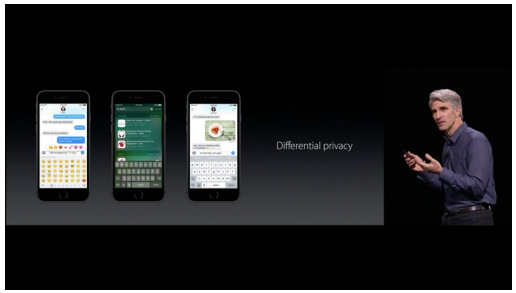
Screenshot Ilya Mironov

Differential Privacy

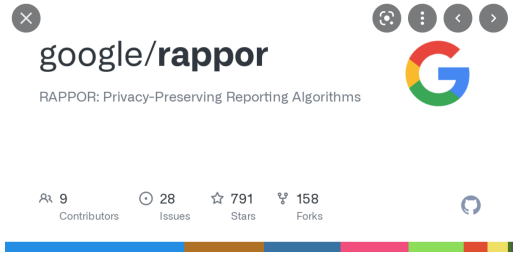
Cryptographic Solutions

Model	Utility	Privacy	Who Holds Data?
Differential Privacy	statistical analysis of dataset	individual-specific info	trusted curator
Secure Function Evaluation	any query desired	everything other than result of query	original users (or semi-trusted delegates)
Fully Homomorphic (or Functional) Encryption	any query desired	everything (except possibly result of query)	untrusted server

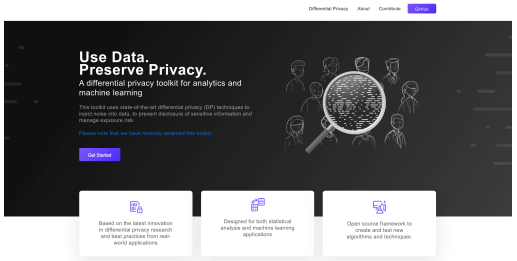
Differential Privacy Deployed



Apple



Google



Microsoft

United States™ Census Bureau

Census Bureau

Uber

Opacus

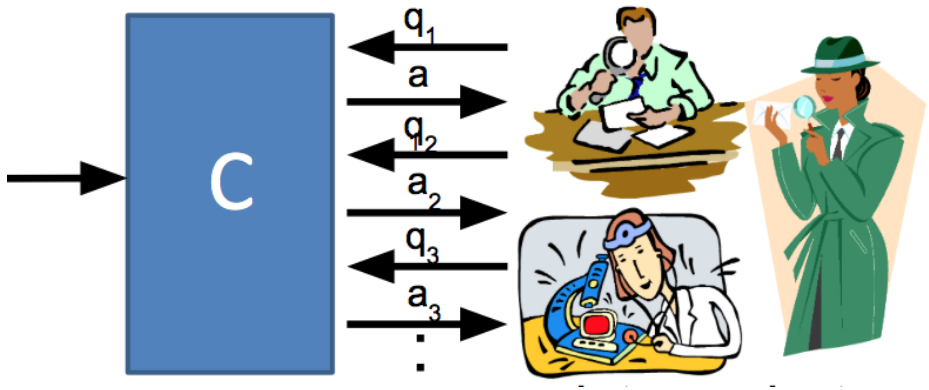
Train PyTorch models with Differential Privacy

Meta

Differential Privacy

[Dinur-Nissim '03,+Dwork, Dwork-Nissim '04,
Blum-Dwork-McSherry-Nissim '05, Dwork-McSherry-Nissim-Smith '06]

Sex	Blood		HIV?
F	B		Y
M	A		N
M	O		N
M	O		Y
F	A		N
M	B		Y

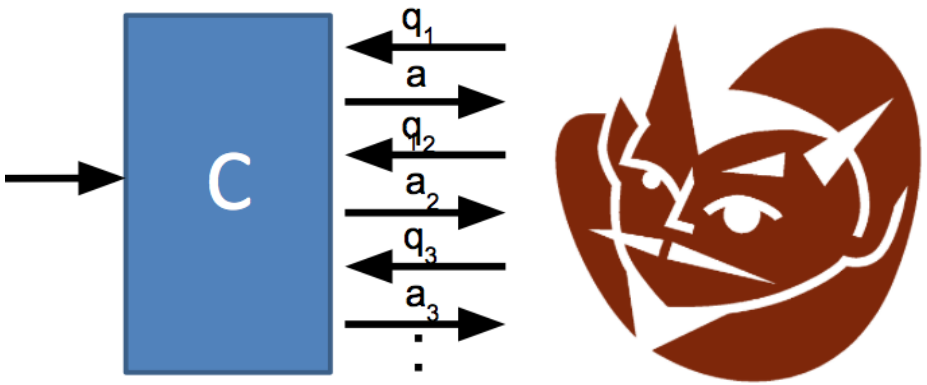


Data Repository Curator Analysts

Differential Privacy

[Dinur-Nissim '03,+Dwork, Dwork-Nissim '04,
Blum-Dwork-McSherry-Nissim '05, Dwork-McSherry-Nissim-Smith '06]

Sex	Blood		HIV?
F	B		Y
M	A		N
M	O		N
M	O		Y
F	A		N
M	B		Y

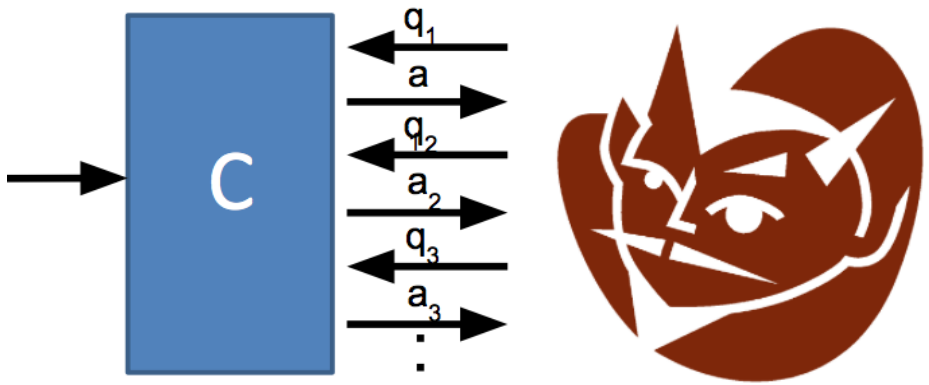


Data Repository Curator Adversary

Differential Privacy

[Dinur-Nissim '03,+Dwork, Dwork-Nissim '04,
Blum-Dwork-McSherry-Nissim '05, Dwork-McSherry-Nissim-Smith '06]

Sex	Blood		HIV?
F	B		Y
M	A		N
M	O		N
M	O		Y
F	A		N
M	B		Y

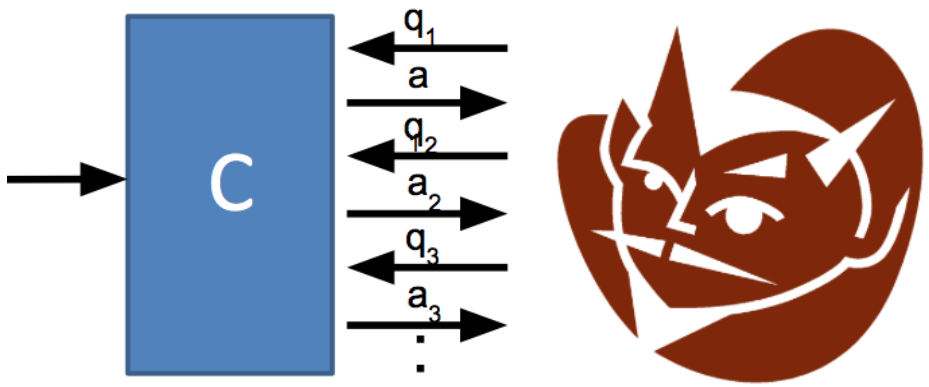


Data Repository Curator Adversary

Differential Privacy

[Dinur-Nissim '03,+Dwork, Dwork-Nissim '04,
Blum-Dwork-McSherry-Nissim '05, Dwork-McSherry-Nissim-Smith '06]

Sex	Blood		HIV?
F	B		Y
M	O		N
M	O		Y
F	A		N
M	B		Y

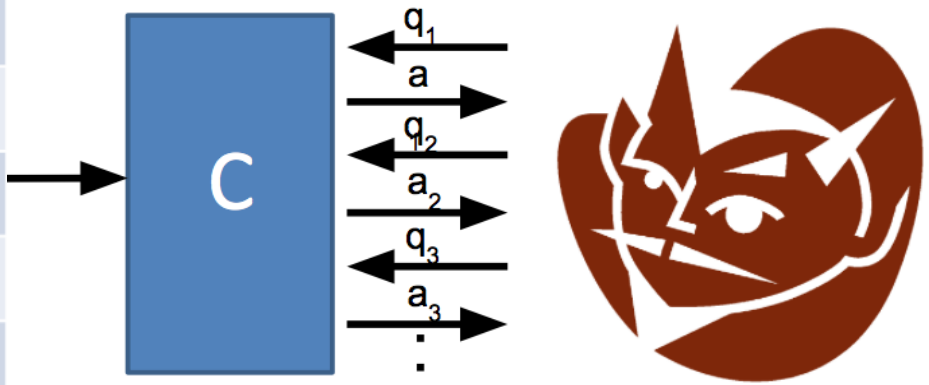


Data Repository Curator Adversary

Differential Privacy

[Dinur-Nissim '03,+Dwork, Dwork-Nissim '04,
Blum-Dwork-McSherry-Nissim '05, Dwork-McSherry-Nissim-Smith '06]

Sex	Blood		HIV?
F	B		Y
F	A		Y
M	O		N
M	O		Y
F	A		N
M	B		Y



Data Repository Curator Adversary

Differential Privacy

M is ϵ-DP if

$$Pr[M(D, q) \in T] \leq (1 + \epsilon) Pr[M(D', q) \in T], \qquad \forall\, T, q.$$

Differential Privacy

M is ϵ-DP if

$$Pr[M(D, q) \in T] \leq (1 + \epsilon) Pr[M(D', q) \in T], \qquad \forall\, T, q.$$

- D, D' Neighbouring datasets

Differential Privacy

M is ϵ-DP if

$$Pr[M(D,q) \in T] \leq e^{\epsilon} Pr[M(D',q) \in T], \qquad \forall\, T, q.$$

- D, D' Neighbouring datasets
- M Mechanism that Maps from data to result

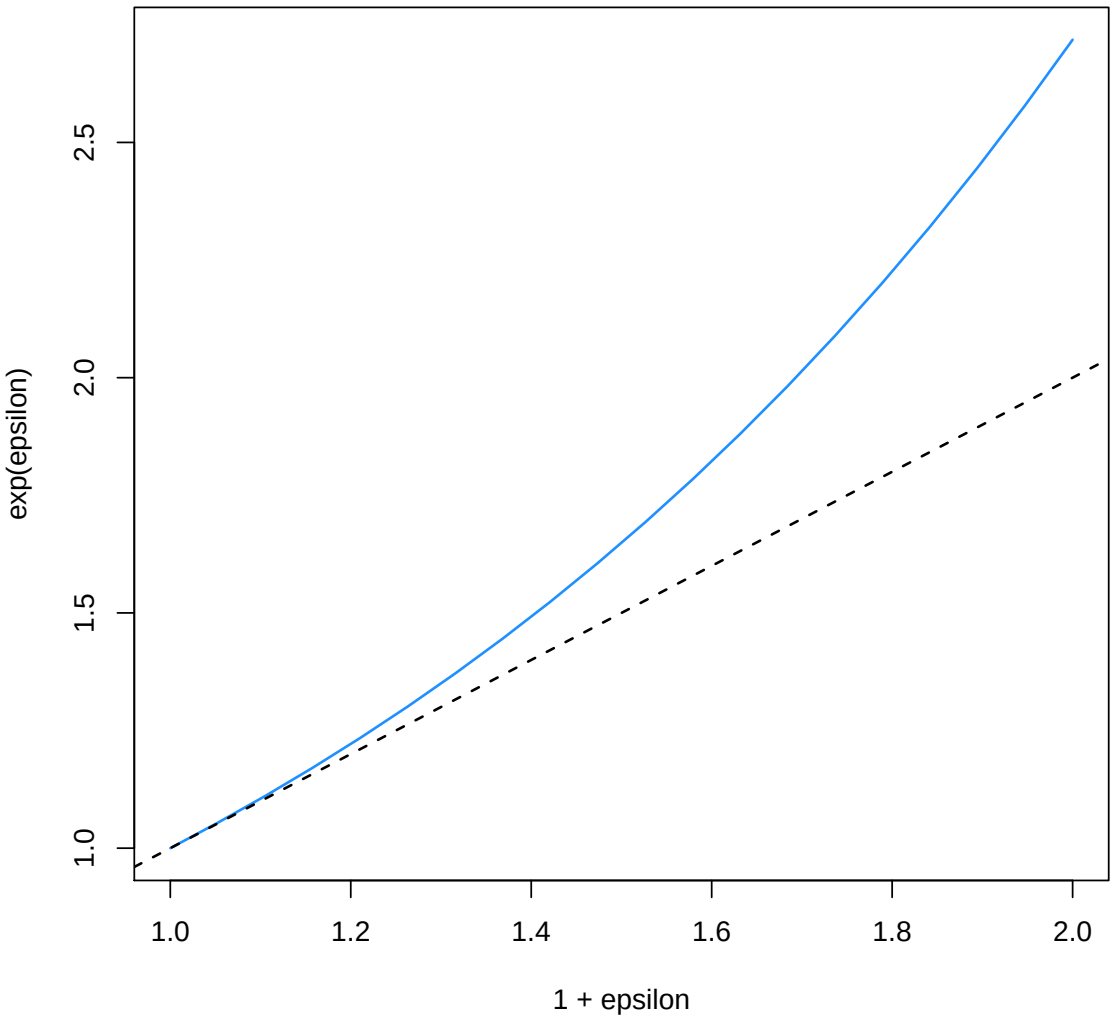
Differential Privacy

M is ϵ-DP if

$$Pr[M(D,q) \in T] \leq e^{\epsilon} Pr[M(D',q) \in T], \qquad \forall\, T, q.$$

- D, D' Neighbouring datasets
- M Mechanism that Maps from data to result
- q Query
- T Set providing a decision rule

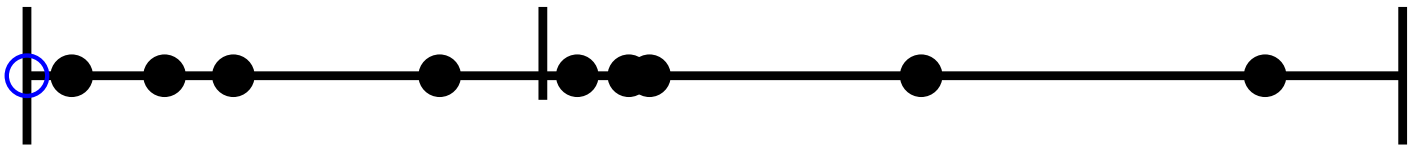
e^{ϵ} vs. $(1+\epsilon)$

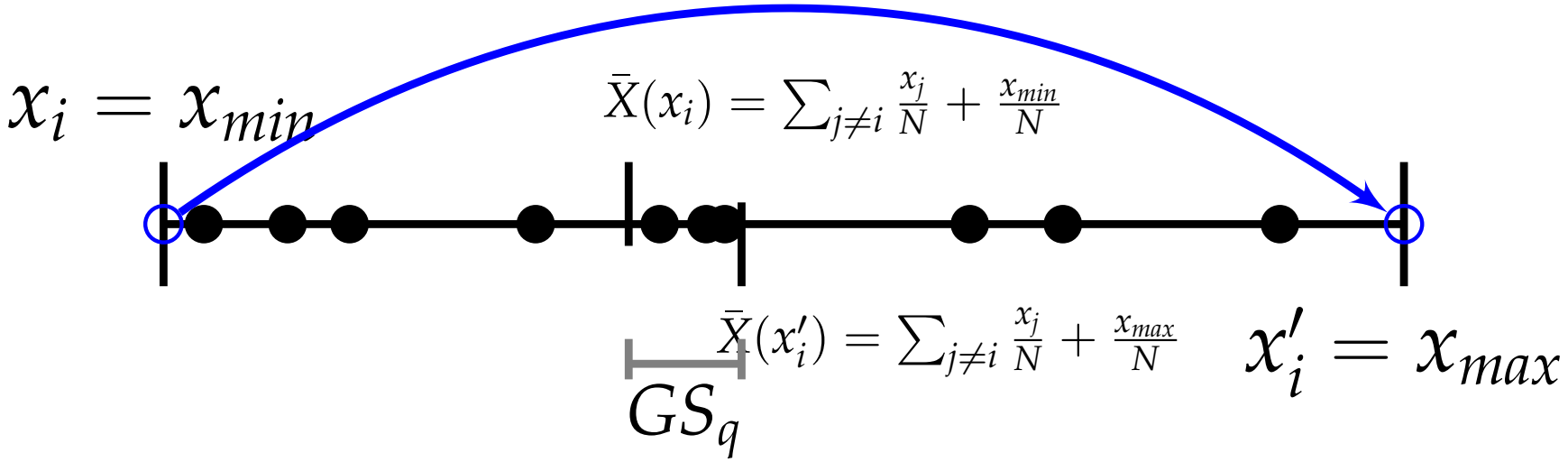


see `expEpsilon.r`

$x_i = x_{min}$

$\bar{X}(x_i) = \sum_{j \neq i} \frac{x_j}{N} + \frac{x_{min}}{N}$




$x_i = x_{min}$
$\bar{X}(x_i) = \sum_{j \neq i} \frac{x_j}{N} + \frac{x_{min}}{N}$
$\bar{X}(x_i') = \sum_{j \neq i} \frac{x_j}{N} + \frac{x_{max}}{N}$
$x_i' = x_{max}$
GS_q

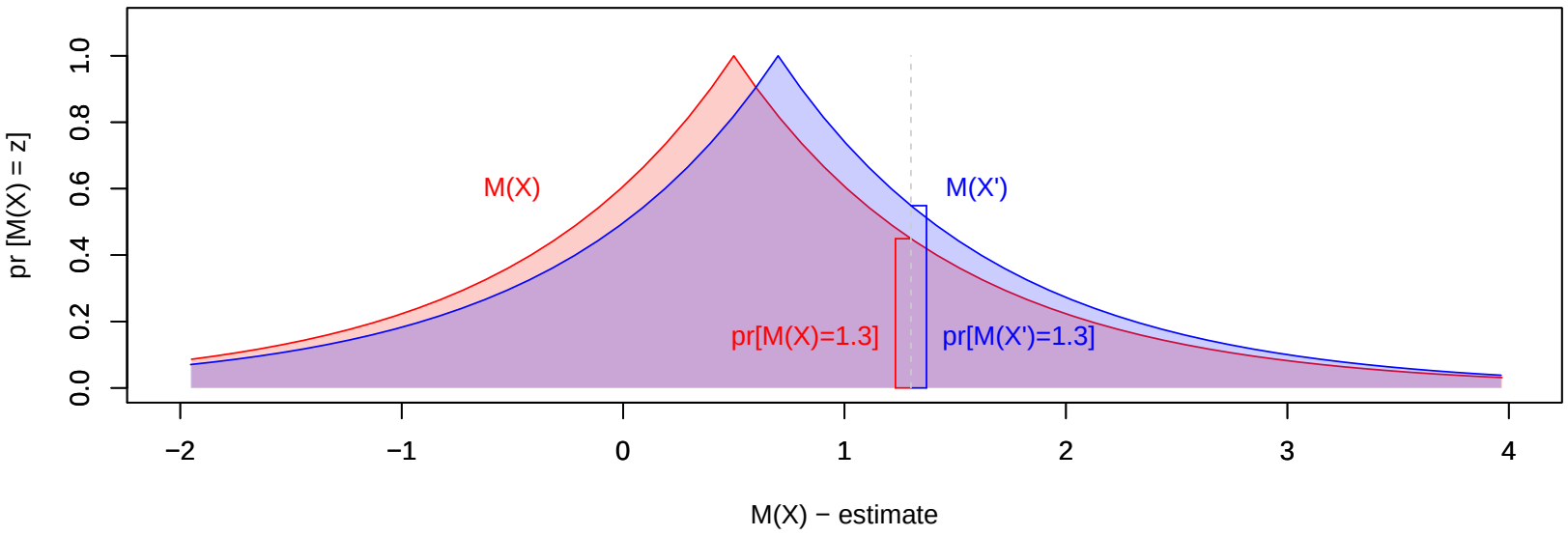
Bounding Data

Action	Term	Domain	Library
Recode	Clip	Signal Processing	NumPy, pandas
	Clamp	Graphics, Geospatial	Torch
	Censor	Statistics	
	Top-code	Surveys	

Bounding Data

Action	Term	Domain	Library
Recode	Clip	Signal Processing	NumPy, pandas
	Clamp	Graphics, Geospatial	Torch
	Censor	Statistics	
	Top-code	Surveys	
Remove	Truncate	Statistics	
	Trim	Many	
Constrain	Winsorize	Statistics	

Meme Gautam Kamath


M(X)
M(X')
pr[M(X)=1.3]
pr[M(X')=1.3]
pr [M(X) = z]
0.0
0.2
0.4
0.6
0.8
1.0
−2
−1
0
1
2
3
4
M(X) − estimate

The Laplace has density over y:

$$f_{Laplace}(y|s,\mu) = \mathrm{Lap}(s,\mu) = \frac{1}{2s}\exp\Big(-\frac{|y-\mu|}{s}\Big)$$

The Laplace has density over y:

$$f_{Laplace}(y|s) = \mathrm{Lap}(s) = \frac{1}{2s} \exp\Big(- \frac{|y|}{s} \Big)$$

We were given the theorem:

$$M(x, q) = q(x) + \mathrm{Lap}(GS_q/\epsilon)$$

The Laplace has density over y:

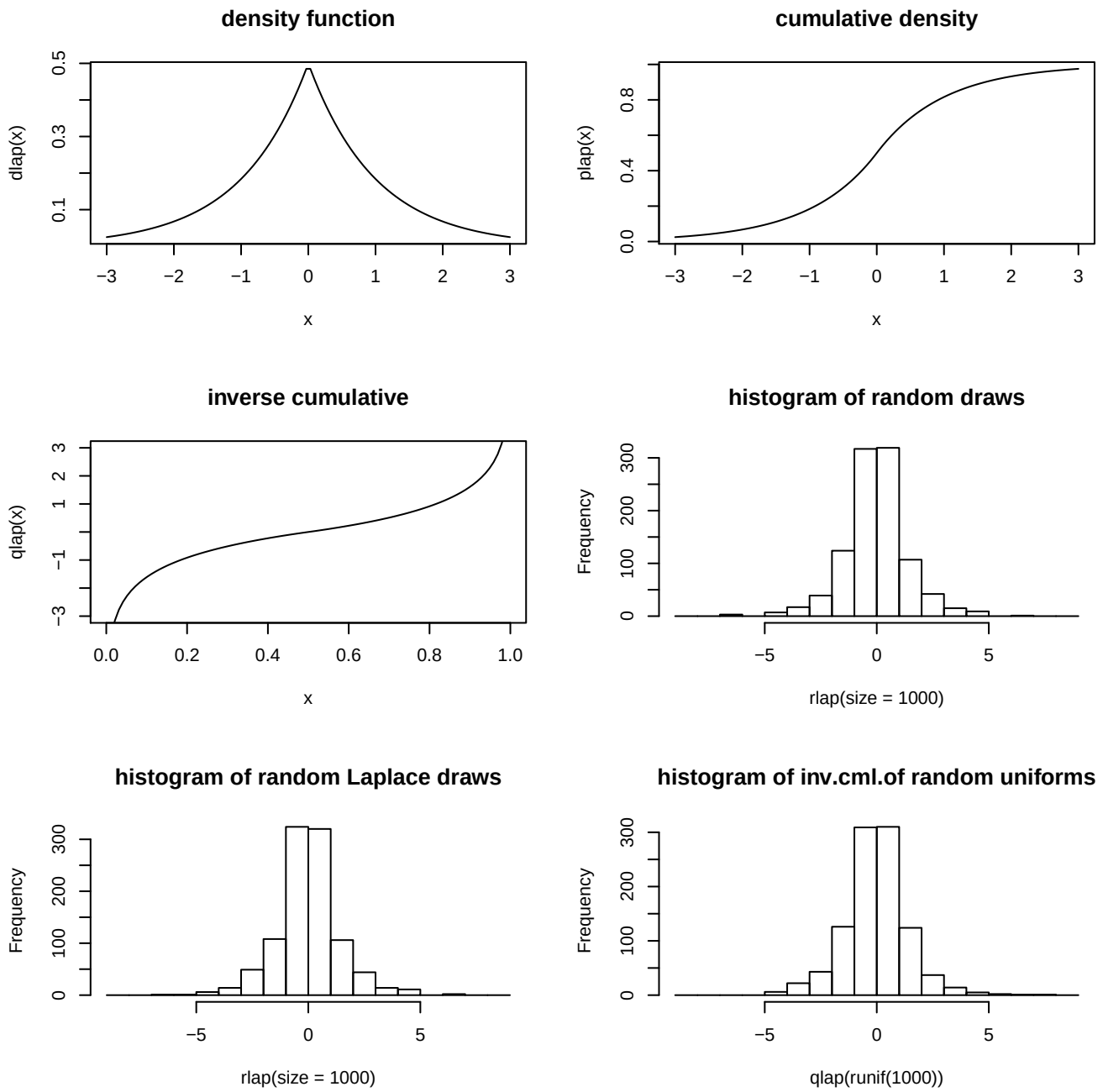
$$f_{Laplace}(y|s) = \text{Lap}(s) = \frac{1}{2s} \exp\Big(- \frac{|y|}{s} \Big)$$

We were given the theorem:

$$M(x, q) = q(x) + \text{Lap}(GS_q/\epsilon)$$

So our differentially private mean, $M(X)$, which combines the "true" sample mean with Laplace noise, becomes:

$$M(x) = \bar{x} + Z; \qquad Z \sim \text{Lap}(s = GS_q/\epsilon)$$

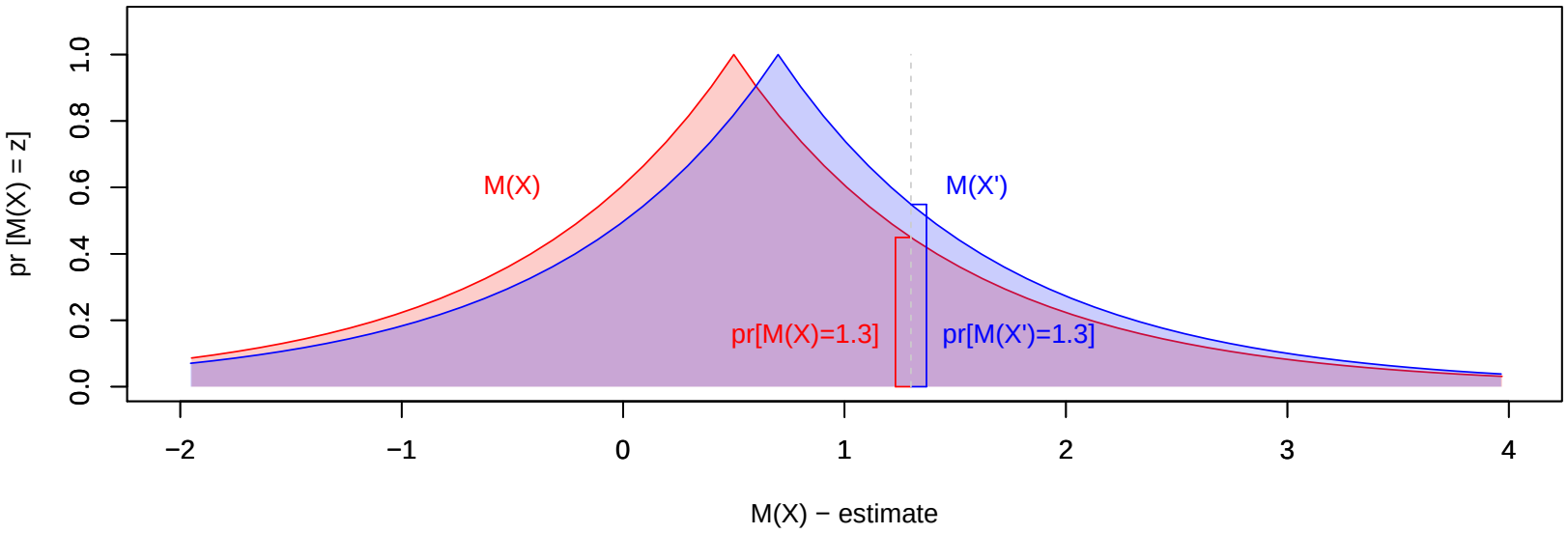


see `showLaplaceDistributions.r`

$$\frac{pr[M(x) = t]}{pr[M(x') = t]} = \frac{e^{\frac{-\epsilon|\bar{x}-t|}{GS_q}}}{e^{\frac{-\epsilon|\bar{x}'-t|}{GS_q}}} = e^{\frac{\epsilon|\bar{x}'-t|-\epsilon|\bar{X}-t|}{GS_q}} = e^{\frac{\epsilon|\bar{x}'-\bar{x}|}{GS_q}} \leq e^{\epsilon}$$

since we know $GS_q \geq |\bar{x}' - \bar{x}|$ by the def. of sensitivity. Thus we meet the original definition:

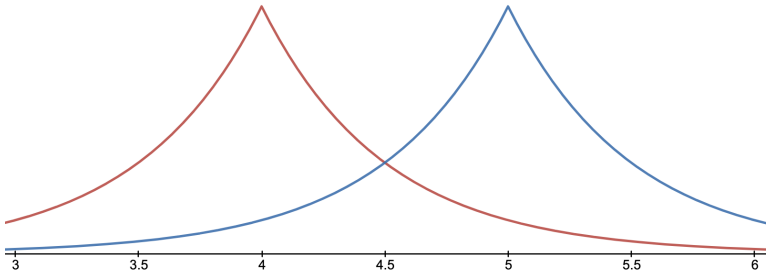
$$Pr[M(x) = t] \leq e^{\epsilon} Pr[M(x') = t]$$



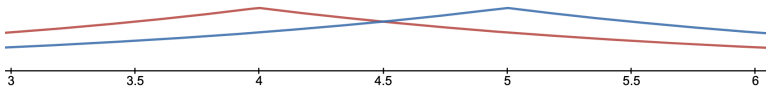
Two Laplace distributions, for two adjacent datasets x and x'. The definition of ϵ-differential privacy requires the ratio of $M(x)/M(x')$ is not greater than e^{ϵ} for all points along the x-axis. Thus for any realized output z (for example here, $z = 1.3$) we can not determine that x or x' were more likely to have produced z.

Privacy vs Accuracy

More **noise** gives more **privacy** (smaller ϵ) and more **error**

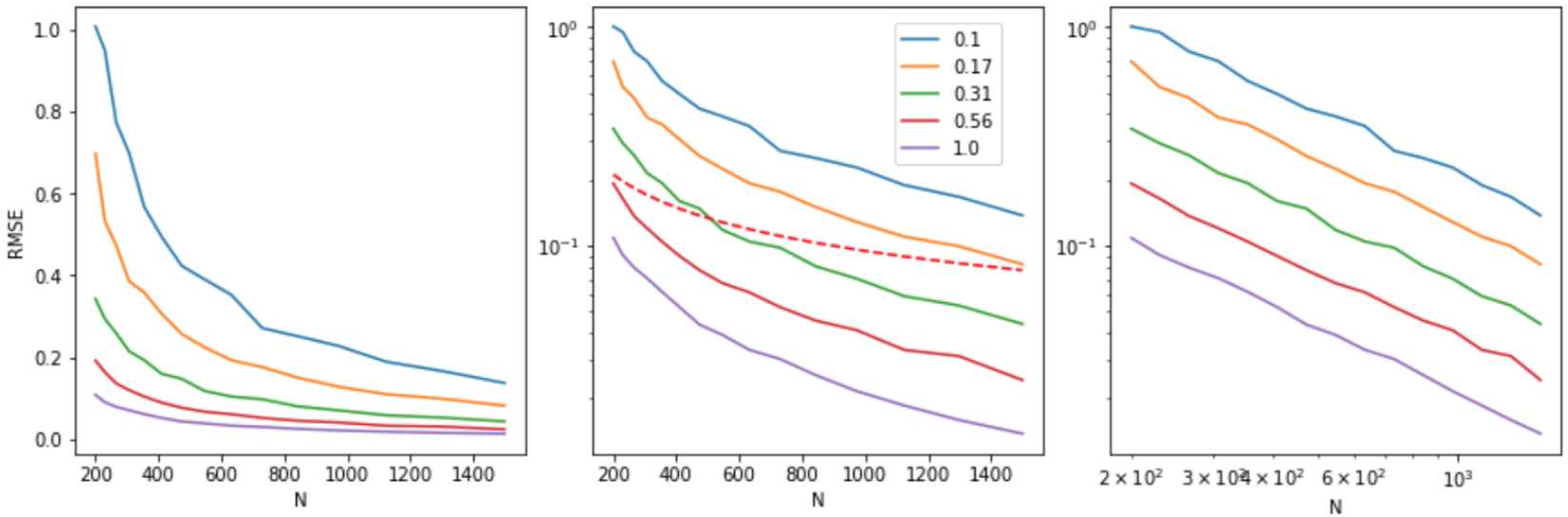


$\epsilon = 2$



$\epsilon = 0.5$

Error from Laplace over ϵ, N



see `laplace_mechanism_and_opendp.ipynb`

Key Differential Privacy Properties:

1. **Post-processing:** Any function on a DP release is still DP
2. **Programability:** Any combination of DP releases is also DP.
3. **Composition:** Privacy loss can be summed across DP releases.

Basic Composition

How to count privacy for multiple queries?
Claim: If $\forall i = 1, \ldots, k$, $A_i :\rightarrow \mathbb{R}^d$ satisfies (ϵ, δ)-DP, then $B(D) = A_1(D), A_2(D), \ldots, A_k(D)$ satisfies $(k\epsilon, k\delta)$-DP.
Example:

- q_1 : histogram of the age, (ϵ, δ)-DP.
- q_2: proportion of the male, (ϵ, δ)-DP.

$q = (q_1, q_2)$ satisfies $(2\epsilon, 2\delta)$-DP.

Advanced Composition

Can we do better? Yes!
Claim: If $\forall i = 1, \ldots, k$, $A_i :\rightarrow \mathbb{R}^d$ satisfies (ϵ, δ)-DP, then $B(D) = A_1(D), A_2(D), \ldots, A_k(D)$ satisfies

$$\epsilon\sqrt{k \ln(1/\delta')} + e^{\epsilon} - 1k, k\delta + \delta'\text{-DP},$$

for any $\delta' > 0$.

Remark 1: when ϵ is small, privacy increases with a sublinear rate!
This is quite common in machine learning.
Remark 2: still loose for specific algorithms, e.g., Gaussian mechanism.

Group Privacy

What if neighbouring datasets may have multiple different rows?

Claim: if A is (ϵ, δ)-DP with respect to $D \sim D'$,
A is $(t\epsilon, te^{t\epsilon}t\delta)$-DP with respect to $D \sim^t D'$ (i.e., t changes).

Application: user-level DP vs event-level DP.

User-level DP vs Event-level DP

Context: multiple rows in a dataset D can belong to one user.

- User-level DP: $D \sim D'$ where one user (perhaps multiple rows) is different.
- Event-level DP: $D \sim D'$ where one row is different.

Relationship: User-level DP $\geq$ Event-level DP.
Claim: suppose one user has at most t events. If A is (ϵ, δ) event-level DP, then A is $(t\epsilon, te^{t\epsilon}t\delta)$ user-level DP.
An immediate corollary of group differential privacy!

Essential Concepts

- **Privacy Loss Definition** - Measure of closeness of distribns

Essential Concepts

- **Privacy Loss Definition** - Measure of closeness of distribns
- **Sensitivity** - Worst case change in a function

Essential Concepts

- **Privacy Loss Definition** - Measure of closeness of distribns
- **Sensitivity** - Worst case change in a function
- **Privacy Loss Parameter** - Formal measure of privacy loss (e.g., ϵ, δ, ρ)

Essential Concepts

- **Privacy Loss Definition** - Measure of closeness of distribns
- **Sensitivity** - Worst case change in a function
- **Privacy Loss Parameter** - Formal measure of privacy loss (e.g., ϵ, δ, ρ)
- **Privacy Loss Budget** - Bound on cumulative privacy loss

Essential Concepts

- **Privacy Loss Definition** - Measure of closeness of distribns
- **Sensitivity** - Worst case change in a function
- **Privacy Loss Parameter** - Formal measure of privacy loss (e.g., ϵ, δ, ρ)
- **Privacy Loss Budget** - Bound on cumulative privacy loss
- **Composition** - Parameters across releases add up

Essential Concepts

- **Privacy Loss Definition** - Measure of closeness of distribns
- **Sensitivity** - Worst case change in a function
- **Privacy Loss Parameter** - Formal measure of privacy loss (e.g., ϵ, δ, ρ)
- **Privacy Loss Budget** - Bound on cumulative privacy loss
- **Composition** - Parameters across releases add up
- **Post Processing** - Any function on a DP release is still DP

Pereira et al. US Broadband Coverage Data Set: A DP Data Release

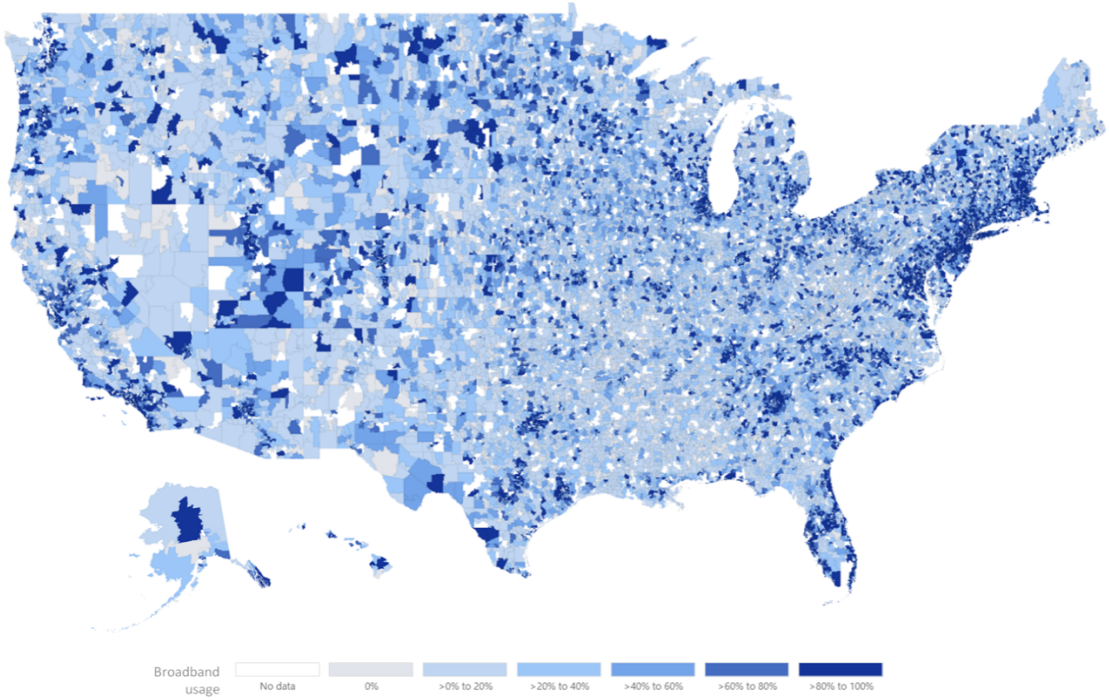
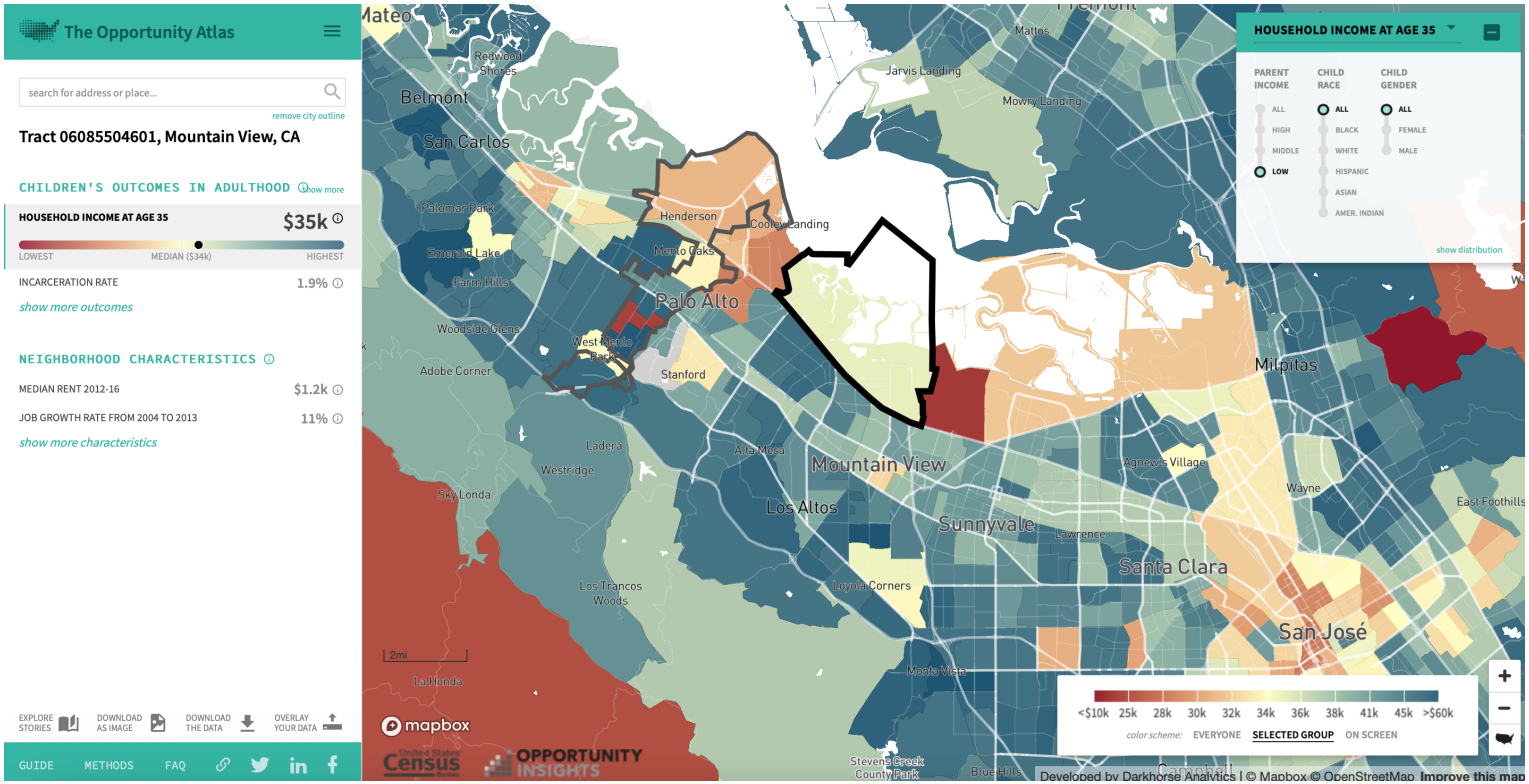


Fig. 1. Map of the United States by postal codes with indicators of broadband coverage.

From Microsoft Services, we query the following Windows telemetry data:
– L_z: Counts of devices connecting to Microsoft Services with internet speed lower than 25Mbps in zip code z.
– H_z: Counts of devices connecting to Microsoft Services with internet speed greater or equal than 25Mbps in zip code z.
Additionally, we query Microsoft Services for the following data:
– M_z: Counts of devices utilizing Microsoft Services in zip code z.
– O_z: Counts of devices not utilizing Microsoft Services in zip code z.


The Opportunity Atlas
search for address or place...
remove city outline
Tract 06085504601, Mountain View, CA
CHILDREN'S OUTCOMES IN ADULTHOOD
show more
HOUSEHOLD INCOME AT AGE 35
$35k
LOWEST
MEDIAN ($34k)
HIGHEST
INCARCERATION RATE
1.9%
show more outcomes
NEIGHBORHOOD CHARACTERISTICS
MEDIAN RENT 2012-16
$1.2k
JOB GROWTH RATE FROM 2004 TO 2013
11%
show more characteristics
EXPLORE STORIES
DOWNLOAD AS IMAGE
DOWNLOAD THE DATA
OVERLAY YOUR DATA
GUIDE
METHODS
FAQ
HOUSEHOLD INCOME AT AGE 35
PARENT INCOME
ALL
HIGH
MIDDLE
LOW
CHILD RACE
ALL
BLACK
WHITE
HISPANIC
ASIAN
AMER. INDIAN
CHILD GENDER
ALL
FEMALE
MALE
show distribution
<$10k 25k 28k 30k 32k 34k 36k 38k 41k 45k >$60k
color scheme: EVERYONE SELECTED GROUP ON SCREEN
Belmont
San Carlos
Palo Alto
Mountain View
Los Altos
Sunnyvale
Santa Clara
San José
Milpitas
Stanford
Menlo Oaks
Henderson
Cooley Landing
Jarvis Landing
Mowry Landing
Alta Mesa
Agnews Village
Wayne
East Foothills
Lawrence
Loyola Corners
Monta Vista
Westridge
Ladera
Sky Londa
Los Trancos Woods
La Honda
Adobe Corner
Woodside Glens
Palomar Park
Emerald Lake
Farm Hills
Redwood Shores
2mi
mapbox
Census
OPPORTUNITY INSIGHTS
Developed by Darkhorse Analytics | © Mapbox © OpenStreetMap Improve this map

Implementations

OpenDP

OpenDP

A **community effort** to build a **trustworthy** and **open-source** suite of differential privacy tools that can be **easily adopted** by custodians of sensitive data to make it available for **research and exploration** in the public interest.

Why?

- Channel our collective advances on science & practice of DP
- Enable wider adoption of DP
- Address high-demand, compelling use cases
- Provide a starting point for custom DP solutions
- Identify important research directions for the field

Project site: `http://opendp.org`

Library Architecture

OpenDP

A Programming Framework that:

1. allows arbitrary distance relations to define privacy-loss across a library of data transforms and privacy measures, and concretely reason about and compose total privacy loss,
2. centrally enables interactive queries such as sparse vector and matrix mechanism,
3. can integrate with trusted compute resources (eg. SQL, Spark) for performance at scale.

Library Language Choice

OpenDP

Library implemented in Rust:

- Rust is a fast-growing, imperative functional language.
- high performance, reliable, good memory safety and type safety.
- structures, generics, higher-order functions, and values are immutable by default.
- It has a strong ownership model that gives memory safety.

Collectively, this makes Rust code easier to reason about concretely.

Opacus

Train PyTorch models with Differential Privacy

INTRODUCTION | GET STARTED | TUTORIALS

Key Features

Scalable

Vectorized per-sample gradient computation that is 10x faster than microbatching

Built on PyTorch

Supports most types of PyTorch models and can be used with minimal modification to the original neural network.

Extensible

Open source, modular API for differential privacy research. Everyone is welcome to contribute.

https://opacus.ai

Opacus for PyTorch

Write out a standard PyTorch model:

```
import torch
import torch.nn as nn
import torch.nn.functional as F

class ExampleLogisticModule(nn.Module):
    def __init__(self, input_size):
        super().__init__()
        self.linear = nn.Linear(input_size, 1)

    def forward(self, x):
        x = self.linear(x)
        x = torch.sigmoid(x)
        return x[:,0]
```

Opacus for PyTorch

Write out a standard PyTorch model:

```
import torch
import torch.nn as nn
import torch.nn.functional as F

class ExampleLogisticModule(nn.Module):
    def __init__(self, input_size):
        super().__init__()
        self.linear = nn.Linear(input_size, 1)

    def forward(self, x):
        x = self.linear(x)
        x = torch.sigmoid(x)
        return x[:,0]
```

Swap out the optimizer for DP:

```
from opacus import PrivacyEngine

privacy_engine = PrivacyEngine()
model, optimizer, data_loader = privacy_engine.make_private(
    module=model,
    optimizer=optimizer,
    data_loader=data_loader,
    noise_multiplier=1.0,
    max_grad_norm=0.5,
)
```