# Chapter 7

# Differential Privacy: An Overview

```
require(smoothmest)
require(dplyr)
require(readr)
require(ggplot2)
cbPalette <- c("#999999",  "#E69F00", "#56B4E9",  "#009E73",
               "#CC79A7", "#D55E00",  "#F0E442", "#0072B2")
```

## 7.1 Introduction

Chapters 3 through 6 introduce the synthetic data approach to data confidentiality. As we have seen, after appropriate Bayesian synthesis models are developed and applied, data disseminators typically will evaluate the utility and disclosure risks of the generated synthetic datasets, and make release decision based on the trade-offs between the utility and disclosure risks. In particular, the evaluation of disclosure risks depends on the assumptions about intruder's knowledge and behavior, which means a different set of assumptions could present a different disclosure risks profile, potentially leading to a different release decision.

The computer science research community proposed a formal mathematical framework, called differential privacy, which provides privacy protection guarantees. Initially proposed by Dwork et al. (2006), differential privacy has gained momentum for the past decade or so, and is now being implemented in various settings, including government, academic, and industry.

In this chapter, we give an overview of differential privacy. Chapter 7.2 presents definitions and implications of key terms in differential privacy. Chapter 7.3

introduces the Laplace Mechanism, with working examples of adding Laplace noise to two numeric queries to a database: count and average. We present properties of differential privacy with working examples in 7.4.

This chapter focuses on the basics of differential privacy, which is non-Bayesian. We think introducing the basics is necessary - it helps deepen our understanding of differential privacy, and prepare us to look at the differential privacy from a Bayesian perspective in Chapters 8 and 9.

## 7.2 Definitions and Implications

### 7.2.1 Adding noise for privacy protection

At the core of differential privacy, is the idea of adding *noise* to the *output* of *queries* made to *databases*. The added noise is random. Furthermore, it depends on the predetermined *privacy budget* and the type of queries being made. In Chapter 7.2.2, we define the key terms in constructing differential privacy one-by-one. We provide a working example to a Consumer Expenditure Surveys (CE) sample, and discuss implications.

### 7.2.2 Definitions

Dwork and Roth (2014)

Think of databases as datasets, for data analysts to use for analysis. As a working example, the CE sample, first introduced in Chapter 4.1.1, is a sample of 994 consumer units (CU) from 2017 1st quarter, with the following information on each CU: urban / rural status, income before taxes in past 12 months, race category of the reference person, and total expenditures in last quarter.

In our settings, we assume databases are confidential, i.e. data analysts do not have access to them. The question becomes whether the data analyst can and how to get information of quantities of interest from a confidential database. Suppose we are the confidential database holder. Now the challenge is we as the database holder, whether we can and how to provide information to the data analyst, that is useful but does not compromise the confidentiality of each data entry in our database.

Suppose the data analyst wants to know quantities such as the number of rural CUs in this sample and the average income of this sample. To get to know these two quantities, count and average income, the analyst could send the following two queries to the database.

1. How many rural CUs are there in this sample?
2. What is the average income of this sample?

We note that for illustration purpose and providing more than one example, we mention two queries in our description. However, we assume that each time only one query is sent to the database by the data analyst.

**Definition 7.1.** Denote numeric queries as functions $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$, mapping databases to $k$ real numbers, $\mathbb{R}^k$.

As the database holder, how can we answer the two queries made by the data analyst? We could just give out the actual values, but would that compromise the confidentiality of any data entry in the database?

For example, if the richest person in the US happens to be in this database, the data entry will drive the average income astronomically high. When the data analyst receives such an extremely high average income output, it is natural to think that someone (maybe more than one) with extremely high income entry (even though might not be the richest one) is in the confidential database. This could be a confidentiality compromise.

As mentioned before, we add noise to the output for privacy protection. Adding noise under differential privacy considers the scenario where *two databases differ by one record*. To formally define differential privacy, let's first define the number of observations (rows) two databases differ, the Hamming distance.

**Definition 7.2.** Given databases $\mathbf{x}, \mathbf{y} \in \mathbb{N}^{|\mathcal{X}|}$, let $\delta(\mathbf{x}, \mathbf{y})$ denote the Hamming distance between $\mathbf{x}$ and $\mathbf{y}$ by:

$$\delta(\mathbf{x}, \mathbf{y}) = \#\{i : x_i \neq y_i\}. \tag{7.1}$$

For example, let $\mathbf{x}$ be the confidential CE sample, and $\mathbf{y}$ be the database where one data entry is deleted from $\mathbf{x}$. In this example, the Hamming distance between $\mathbf{x}$ and $\mathbf{y}$ is 1, therefore $\delta(\mathbf{x}, \mathbf{y}) = 1$. We note that the differed data entry can be anyone, not necessarily the record with the highest income. We further note that the "differ by one record" difference between $\mathbf{x}$ and $\mathbf{y}$ could be a removal of a single observation, or change of a single observation, among others.

Next, we define $\ell_1-$sensitivity, the magnitude a single individual's data can change the $\ell_1$ norm of the function $f$ in the worst case. The $\ell_1$ norm between $f(\mathbf{x})$ and $f(\mathbf{y})$ is the absolute difference between $f(\mathbf{x})$ and $f(\mathbf{y})$, denoted as $||f(\mathbf{x}) - f(\mathbf{y})||_1$.

**Definition 7.3.** The $\ell_1-$sensitivity of a function $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$ is:

$$\Delta f = \max_{\mathbf{x},\mathbf{y} \in \mathbb{N}^{|\mathcal{X}|}, \delta(\mathbf{x},\mathbf{y})=1} ||f(\mathbf{x}) - f(\mathbf{y})||_1. \tag{7.2}$$

Here, $\Delta f$ is the maximum change in the function $f$ on $\mathbf{x}$ and $\mathbf{y}$, where $\mathbf{x}, \mathbf{y} \in \mathbb{N}^{|\mathcal{X}|}$ and differ by a single observation (i.e. $\mathbf{x}, \mathbf{y} \in \mathbb{N}^{|\mathcal{X}|}, \delta(\mathbf{x}, \mathbf{y}) = 1$).

Continue with our CE sample example, where $\mathbf{x}$ is the confidential CE sample, $\mathbf{y}$ is the database where one data entry is deleted from $\mathbf{x}$, therefore $\delta(\mathbf{x}, \mathbf{y}) = 1$. Recall the two queries we receive from the data analyst:

1. How many rural CUs are there in this sample?
2. What is the average income of this sample?

For 1, the numeric query $f$ is the count of rural CUs. Since $\mathbf{y}$ is the database where one data entry is deleted from $\mathbf{x}$, the maximum change in $f$, the $\ell_1-$sensitivity, is 1. This is because the maximum change to the count of rural CUs by a single observation is 1: either the deleted data entry is a rural CU, resulting a change of -1 to $f$, or the deleted entry is an ubran CU, resulting a change of 0 to $f$. Therefore, for the count query when $\delta(\mathbf{x}, \mathbf{y}) = 1$, we have $\Delta f = 1$.

For 2, the numeric query $f$ is the average income. Let $n$ be the total number of observations, $a$ the lower bound and $b$ the upper bound of the income variable. Since $\mathbf{y}$ is the database where one data entry is deleted from $\mathbf{x}$, the maximum change in $f$, the $\ell_1-$sensitivity, is $\frac{b-a}{n}$. This is because the maximum change to the mean by a single observation is the range of the variable divided by the number of observations. Therefore, for the average query when $\delta(\mathbf{x}, \mathbf{y}) = 1$, we have $\Delta f = \frac{b-a}{n}$.

In sum, the $\ell_1$-sensitivity depends on the database and the query sent to the database by the data analyst. It is by definition, the magnitude a single individual's data can change the $\ell_1$ norm of the function $f$ in the worst case. For database holders like us, who are to provide privacy-guaranteed output for a query by adding noise to the output, intuitively the $\ell_1$-sensitivity is the noise in the output that we must introduce in order to hide a single individual's participation in the database.

Now we are ready to define $\epsilon-$differential privacy. In essence, we want to guarantee that a mechanism (aka technology) behaves similarly (i.e. giving similar outputs) on similar inputs (e.g. when two databases differ by one). One approach is to bound the log ratio of the probabilities of the outputs from above, which in turn gives an upper bound on the noise added to the output to preserve privacy.

**Definition 7.4.** A mechanism $\mathcal{M}$ with domain $\mathbb{N}^{|\mathcal{X}|}$ is $\epsilon-$differentially private for all $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$ and for all $\mathbf{x}, \mathbf{y} \in \mathbb{N}^{|\mathcal{X}|}$ such that $\delta(\mathbf{x}, \mathbf{y}) = 1$:

$$\left| \ln \left( \frac{Pr[\mathcal{M}(\mathbf{x}) \in \mathcal{S}]}{Pr[\mathcal{M}(\mathbf{y}) \in \mathcal{S}]} \right) \right| \leq \epsilon. \tag{7.3}$$

The ratio $\ln \left( \frac{Pr[\mathcal{M}(\mathbf{x}) \in \mathcal{S}]}{Pr[\mathcal{M}(\mathbf{y}) \in \mathcal{S}]} \right)$ the log of the ratio of the probability of the output undergone mechanism $\mathcal{M}$ from the database $\mathbf{x}$, and that from the database $\mathbf{y}$. $\mathbf{x}$ and $\mathbf{y}$ are similar inputs (differ by one record, as $\delta(\mathbf{x}, \mathbf{y}) = 1$), and the ratio

$\ln\left(\frac{Pr[\mathcal{M}(\mathbf{x})\in\mathcal{S}]}{Pr[\mathcal{M}(\mathbf{y})\in\mathcal{S}]}\right)$ can be considered as the difference in the outputs. Bounding it above by $\epsilon$, the privacy budget (Definition 7.5 below), $\epsilon-$differential privacy provides us a means to perturb the output by adding noise, so that similar inputs produce similar outputs under the mechanism $\mathcal{M}$.

**Definition 7.5.** The term $\epsilon$ is the privacy budget, that is to be spent by the database holder when answering queries.

### 7.2.3 Implications

With given privacy budget, we can then add noise according to the $\epsilon-$differential privacy definition to the output, in order to preserve privacy. Let's take a moment to understand the relationships among these quantities: database, query, sensitivity, privacy budget, and added noise. We will see these two important implications: the added noise is positively related to the sensitivity, while it is negatively related to the privacy budget.

#### 7.2.3.1 Sensitivity and added noise

Our Defintion 7.3 of $\ell_1-$sensitivity of query (function) $f$ is to capture the magnitude a single individual's data can change the $\ell_1$ norm of the query $f$ in the worse case, denoted as $\Delta f$. Furthermore, we emphasize that it depends on the database and the query sent to the database by the data analyst. That is, the $\ell_1-$sensitivity of a query (function) $f$, $\Delta f$, is determined and fixed given the database and the query. For example, for any count query, $\Delta f = 1$ (regardless of the database). However, for the average query, $\Delta f = \frac{b-a}{n}$, which depends on the database through: $n$ the number of observations, and $a$ and $b$, the lower and upper bounds of the variable of interest.

Intuitively, we can see that for a query $f$ with large $\ell_1-$sensitivity, $\Delta f$, larger noise is needed for the same level of privacy protection (i.e. given fixed privacy budget), and vice versa.

For example, if the data analyst sends two average queries to the CE sample:

1. What is the average income of this sample (income before taxes in past 12 months)?
2. What is the average expenditure of this sample (total expenditures in last quarter)?

Since the range of the variable income is larger than that of the variable expenditures, the $\ell_1-$sensitivity for the first query, denoted as $\Delta f_i$ ($i$ stands for income), is larger than that for the second query, denoted as $\Delta f_e$ ($e$ stands for expenditures). Therefore, the noised needs to be added to the average query to income, $f_i$, is larger than that for the average query to expenditures, $f_e$, if the same level of privacy protection is guaranteed.

In sum, the sensitivity and the added noise are positively related: given fixed privacy budget $\epsilon$, larger sensitivity results in larger added noise.

#### 7.2.3.2 Privacy budget and added noise

Our Definition 7.4 of $\epsilon-$differential privacy provides an upper bound on the noised necessary to be added to the output for privacy protection. The upper bound is $\epsilon$, the privacy budget (Definition 7.5). Unlike the sensitivity, the privacy budget does not depend on the database or the query sent to the database.

As can be seen in the setup in Equation (7.3), an increase of the privacy budget $\epsilon$ indicates less privacy protection, because we allow less similar outputs with an increased upper bound. On the other hand, a decrease of the privacy budget $\epsilon$ indicates more privacy protection, because we need more similar outputs with a decreased upper bound. In the extreme cases, roughly speaking, $\epsilon = 0$ indicates perfect privacy while $\epsilon = \infty$ indicates no privacy guarantee.

Subsequently, small privacy budget $\epsilon$ results in larger added noise, because we are less willing to disclose the true output value (e.g. $\epsilon = 0.1$), while large privacy budget $\epsilon$ results in smaller added noise, because we are more willing to disclose the true output value (e.g. $\epsilon = 1$).

In sum, the privacy budget and the added noise are negatively related: given fixed sensitivity $\Delta f$, larger privacy budget results in smaller added noise.

## 7.3 The Laplace Mechanism

The Laplace Mechanism is a mechanism that satisfies $\epsilon-$differential privacy. It adds noise to the output for privacy protection. The added noise is drawn from a Laplace distribution, whose parameters depend on the sensitivity and the privacy budget.

We introduce the Laplace distribution in Chapter 7.3.1, with visualizations of several example Laplace distributions and comparison of the Laplace distribution and the normal distribution. 7.3.2 describes how are the parameters of the Laplace distribution related to the sensitivity and the privacy budget, making connections to our previously discussed implications in Chapter 7.2.3. We explain why the Laplace Mechanism works in Chapter 7.3.3, and provide working examples of adding Laplace noise to numeric queries of the CE database in Chapter 7.3.4.

### 7.3.1 The Laplace distribution

**Definition 7.6.** A random variable has a Laplace$(\mu, s)$ distribution if its probability density function is

$$f(x \mid \mu, s) = \frac{1}{2s} \exp\left(-\frac{|x-\mu|}{s}\right) \tag{7.4}$$

$$= \frac{1}{2s} \begin{cases} \exp\left(-\frac{\mu - x}{s}\right) & \text{if } x < \mu; \\ \exp\left(-\frac{x-\mu}{s}\right) & \text{if } x \geq \mu, \end{cases} \tag{7.5}$$

where $\mu$ is a location parameter, and $s > 0$ is a scale parameter. When $\mu = 0, b = 1$, the positive half-line is an exponential distribution scaled by $\frac{1}{2}$.

Like the normal distribution, the Laplace distribution is symmetric, and centered at its location parameter $\mu$. The scale parameter $s$ controls its spread: larger $s$ indicates bigger spread.

We plot 4 Laplace densities with different values of $\mu$ and $b$. Note that the `dlaplace()` function is from the `rmulti` R package. It takes inputs `m` as $\mu$ and `s` as $s$.

```
require(rmutil)
ggplot(data.frame(x = c(-10, 10)), aes(x)) +
  stat_function(fun = dlaplace, args = list(m = 0, s = 0.1),
                aes(color = "Laplace(0, 0.1)")) +
  stat_function(fun = dlaplace, args = list(m = 0, s = 0.5),
                aes(color = "Laplace(0, 0.5)")) +
  stat_function(fun = dlaplace, args = list(m = 0, s = 1),
                aes(color = "Laplace(0, 1)")) +
  stat_function(fun = dlaplace, args = list(m = -5, s = 1),
                aes(color = "Laplace(-5,1)")) +
  scale_colour_manual(values = cbPalette) + ylab("Density") +
  theme_bw(base_size = 15, base_family = "")
```

Although the Laplace distribution is centered at its mean $\mu$ and symmetric, it has a different shape from the normal distribution. In fact, the Laplace distribution has fatter tails than normal distribution. Equation (7.3) indicates absolute difference from the mean, instead of the squared difference from the mean in the normal distribution, therefore the Laplace distribution has aftter tails.

To illustrate the comparison, we plot a Laplace distribution and a normal distribution, both have mean 0 and variance 1. Note that the variance of Laplace$(\mu, s)$ is $2s^2$, therefore $s = \sqrt{\frac{1}{2}}$ for a Laplace distribution with variance 1.

```
ggplot(data.frame(x = c(-10, 10)), aes(x)) +
  stat_function(fun = dlaplace, args = list(m = 0, s = sqrt(1/2)),
```

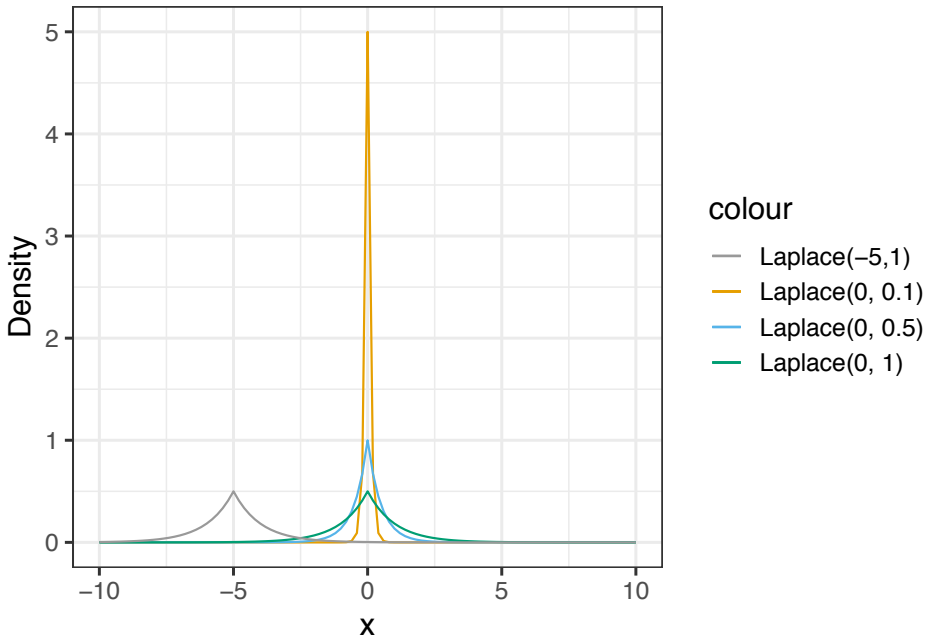


Figure 7.1: Density plots of four Laplace distributions.

```
                aes(color = "Laplace(0, sqrt(1/2))")) +
  stat_function(fun = dnorm, args = list(mean = 0, sd = 1),
                aes(color = "Normal(0, 1)")) +
  scale_colour_manual(values = cbPalette) + ylab("Density") +
  theme_bw(base_size = 15, base_family = "")
```

As illustrated in Figure 7.2, comparing to a normal distribution, the Laplace distribution with same mean and same variance is more concentrated arount its mean (more peaked with higher probability around its mean) and has fatter tails.

### 7.3.2 Laplace noise for privacy protection

As we know, the Laplace Mechanism adds noise to the output with $\epsilon-$differential privacy guarantee, and the noise is drawn from a Laplace distribution. Moreover, Chapter 7.2.3 demonstrates that the added noise is positively related to the sensitivity, and negatively related to the privacy budget. The sensitivity is dependent on the database and the query, while the privacy budget is independent of the database and the query.

For given sensitivity $\Delta f$ and privacy budget $\epsilon$, the added noise to the output of

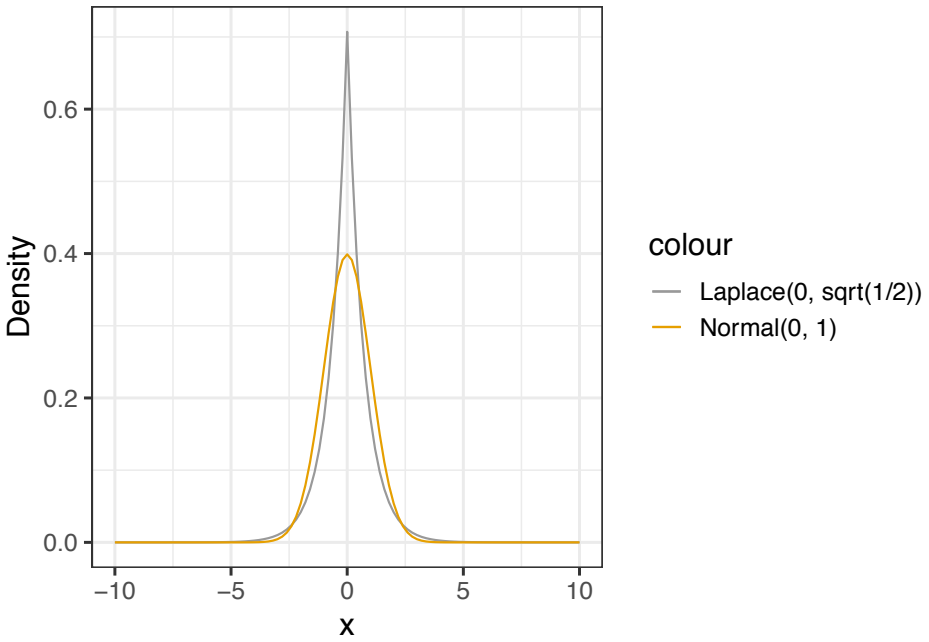


Figure 7.2: Density plots of a Laplace distribution and a normal distribution, both with mean 0 and variance 1.

a query sent to database $\mathbf{x}$, $X^*$ is drawn from a Laplace distribution with mean 0, and scale $\frac{\Delta f}{\epsilon}$:

$$X^* \sim \text{Laplace}\left(0, \frac{\Delta f}{\epsilon}\right). \tag{7.6}$$

The scale of a Laplace distribution controls its spread, and larger scale value indicates bigger spread. Therefore, if the added noise needs to be larger, we should draw it from a Laplace distribution with larger scale value, and vice versa.

The scale of a Laplace noise in the Laplace Mechanism is $\frac{\Delta f}{\epsilon}$, the ratio of the $\ell_1-$sensitivity and the privacy budget. The positive relation between the added noise and the sensitivity can be illustrated by fixing $\epsilon$, larger $\Delta f$ results in larger scale, indicating larger noise. The negative relation between the added noise and the privacy budget can be illustrated by fixing $\Delta f$, larger $\epsilon$ results in smaller scale, indicating smaller noise.

Formally, we define the Laplace Mechanism for a $k-$dimension query $f$ to database $\mathbf{x}$ as follows.

**Definition 7.7.** Given any function $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$, the Laplace mechanism is defined as

$$\mathcal{M}_L(\mathbf{x}, f(\cdot), \epsilon) = f(\mathbf{x}) + (X_1^*, \cdots, X_k^*), \tag{7.7}$$

where $X_i^*$ are *i.i.d.* random variables drawn from Laplace $\left(0, \frac{\Delta f}{\epsilon}\right)$.

Let's revisit our two example queries sent to the CE database by the data analyst:

1. How many rural CUs are there in this sample?
2. What is the average income of this sample?

For 1, the numeric count query $f$ is 1−dimension. Therefore, after calculating the sensitivity $\Delta f = 1$, and choosing the privacy budget $\epsilon$, the added noise to the output of the count query can be drawn from Laplace$\left(0, \frac{1}{\epsilon}\right)$. Note that with Hamming distance 1, i.e. two databases differ by one record $\delta(\mathbf{x}, \mathbf{y}) = 1$, the sensitivity $\Delta f = 1$ is for any count query.

For 2, the numeric average query $f$ is also 1−dimension. Note that its sensitivity depends on the database, i.e. we rely on the number of observations $n$ and the range of the numerical variable $b - a$, to calculate the sensitivity $\Delta f = \frac{b-a}{n}$. After choosing the privacy budget $\epsilon$, the added noise to the output of the average query can be drawn from Laplace$\left(0, \frac{(b-a)/n}{\epsilon}\right)$.

EXAMPLE OF A $k$-DIMENSION QUERY?

For ease of illustration, we assume that the two queries are sent separately, thus we consider $\epsilon$ as the privacy budget for each query. Later in Chapter 7.4, we will see that when multiple queries are sent to the same database at the same time, a composition theorem will be used to divide the privacy budget $\epsilon$ by the number of queries, and the resulting privacy budget will be used for each query.

### 7.3.3 Why Laplace Mechanism works

Before proving why the Laplace Mechanism works for the generic $k$−dimension query $f$, let's consider the count query example, where the data analyst is interested in the number of rural CUs in the CE database.

Let $\mathbf{x}$ be the CE database. Denote $\mathbf{s} \in \{0, 1\}^n$, the vector of 0's and 1's, where $s_i = 1$ indicates that CU $i$ is a rural CU, and $s_i = 0$ otherwise. The count query of the number of rural CUs can then be expressed as $f(\mathbf{x}) = \sum_i s_i$, the total number of 1's in the $\mathbf{s}$ vector. Following the Laplace Mechanism, we can add noise of $X^*$ to the output $f(\mathbf{x}) = \sum_i s_i$, where $X^*$ is drawn from the following Laplace distribution:

$$X^* \sim \text{Laplace}\left(0, \frac{1}{\epsilon}\right), \tag{7.8}$$

where $\epsilon$ is the pre-determined privacy budget.

*Proof*: For any real numbers $z, z'$, as draws from Laplace$\big(0, \frac{1}{\epsilon}\big)$, following the Laplace density in Equation (7.4), we have

$$\frac{p(z)}{p(z')} \propto \frac{\exp\left(-\frac{|z|}{1/\epsilon}\right)}{\exp\left(-\frac{|z'|}{1/\epsilon}\right)} \leq \exp\left(\frac{|z-z'|}{1/\epsilon}\right) \tag{7.9}$$

$$= \exp(\epsilon|z-z'|) = \exp(\epsilon|z-z'|). \tag{7.10}$$

For any two databses, $\mathbf{x}$ and $\mathbf{y}$ which differ by a single entry, the sums $f(\mathbf{x})$ and $f(\mathbf{y})$ differ by at most one, i.e. $|f(\mathbf{x}) - f(\mathbf{y})| \leq 1$. Thus, for noised-added output $z \in \mathbb{R}, z = f(\mathbf{x}) + X^*$, i.e. $X^* = f(\mathbf{x}) - z$.

$$\frac{Pr(f(\mathbf{x}) + X^* = z)}{Pr(f(\mathbf{y}) + X^* = z)} = \frac{p(z - f(\mathbf{x}))}{p(z - f(\mathbf{y}))} \leq \exp(\epsilon|f(\mathbf{x}) - f(\mathbf{y})|) \leq \exp(\epsilon). \tag{7.11}$$

Now we present the generic $k-$dimension query $f$ and prove why Laplace Mechanisms preserves $\epsilon-$differential privacy.

**Theorem 7.1.** *The Laplace Mechanism preserves $\epsilon-$differential privacy.*

*Proof: Let $\mathbf{x}, \mathbf{y} \in \mathbb{N}^{|\mathcal{X}|}$ with Haming distance 1, i.e. $\delta(\mathbf{x}, \mathbf{y}) = 1$, and let $f(\cdot)$ be some function $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$. Let $p_{\mathbf{x}}$ denote the probability density function of $\mathcal{M}_L(\mathbf{x}, f(\cdot), \epsilon)$, and let $p_{\mathbf{y}}$ denote the probability density function of $\mathcal{M}_L(\mathbf{y}, f(\cdot), \epsilon)$. We compare the two at some arbitrary output point $z \in \mathbb{R}^k$:*

$$\frac{p_{\mathbf{x}}(z)}{p_{\mathbf{y}}(z)} = \prod_{i=1}^{k} \left(\frac{\exp\left(-\frac{|f(\mathbf{x})_i - z_i|}{\Delta f/\epsilon}\right)}{\exp\left(-\frac{|f(\mathbf{y})_i - z_i|}{\Delta f/\epsilon}\right)}\right) \tag{7.12}$$

$$= \prod_{i=1}^{k} \exp\left(\epsilon \frac{|f(\mathbf{y})_i - z_i| - |f(\mathbf{x})_i - z_i|}{\Delta f}\right) \tag{7.13}$$

$$\leq \prod_{i=1}^{k} \exp\left(\epsilon \frac{|(f(\mathbf{x})_i - f(\mathbf{y})_i)|}{\Delta f}\right) \tag{7.14}$$

$$= \exp\left(\epsilon \frac{||f(\mathbf{x}) - f(\mathbf{y})||_1}{\Delta f}\right) \tag{7.15}$$

$$\leq \exp(\epsilon) \tag{7.16}$$

Equation (7.13) to Equation (7.14) follows from the triangle inequality, and Equation (7.15) to Equation (7.16) follows from the definition of sensitivity (Definition 7.3) and $\delta(\mathbf{x}, \mathbf{y}) = 1$, i.e. Hamming distance 1. That $\frac{p_{\mathbf{x}}(z)}{p_{\mathbf{y}}(z)} \geq \exp(-\epsilon)$ follows by symmetry. The proof is adopted from Dwork and Roth (2014).

### 7.3.4 Examples: adding Laplace noise to numeric queries

We now demonstrate how to add Laplace noise to numeric queries sent to the CE database. Table 7.1 shows the variables in the CE database.

Table 7.1. Variables used in the CE sample. Data taken from the 2017 CE public use microdata samples.

| Variable Name | Variable information |
|---|---|
| UrbanRural | Binary; the urban / rural status of CU: 1 = Urban, 2 = Rural. |
| Income | Continuous; the amount of CU income bfore taxes in past 12 months. |
| Race | Categorical; the race category of the reference person: 1 = White, 2 = Black, 3 = Native American, 4 = Asian, 5 = Pacific Islander, 6 = Multi-race. |
| Expenditure | Continuous; CU's total expenditures in last quarter. |

#### 7.3.4.1 Adding Laplace noise to a count query

In this example, we add Laplace noise to the count query about the number of rural CUs in the CE database.

##### 7.3.4.1.1 Calculate the true count of rural CUs

We first load the CE sample and create the indicator vector $\mathbf{s}$ by setting $s_i = 1$ if CU $i$ is a rural CU, and $s_i = 0$ otherwise.

```
CEdata <- read_csv("datasets/CEdata.csv")
CEdata$s[CEdata$UrbanRural == 2] <- 1
CEdata$s[CEdata$UrbanRural == 1] <- 0
```

Out of 994 CUs in the CE database, there are 51 rural CUs. This is the true count.

```
n_rural <- CEdata %>%
  summarize_at(vars(s), sum) %>%
  pull()
n_rural
```

```
## [1] 51
```

##### 7.3.4.1.2 Add Laplace noise to the true count

We know that the sensitivity for any count query with Hamming distance 1 (i.e. $\delta(\mathbf{x} - \mathbf{y}) = 1$) is $\Delta f = 1$.

```
Delta_f_count <- 1
```

We can use the `rlaplace()` function in the `rmulti` R package to generate a random draw from a Laplace distribution. We demonstrate using two different values of $\epsilon$: 0.1 and 1. Make sure to use the `set.seed()` function for reproducible results. To obtain integer counts, use the `round()` function to the noise-added ouptput.

```
require(rmutil)
set.seed(123)
epsilon1 <- 0.1
rlaplace(1, n_rural, Delta_f_count/epsilon1) %>%
  round()
```

```
## [1] 45
```

```
set.seed(123)
epsilon2 <- 1
rlaplace(1, n_rural, Delta_f_count/epsilon2) %>%
  round()
```

```
## [1] 50
```

With the true count of 51 rural CUs, we can see that smaller privacy budget adds more noise, $51 - 45 = 6$ (when $\epsilon = 0.1$) versus $51 - 50 = 1$ (when $\epsilon = 1$). These outcomes are in line with our previously discussed implications, that when fixing the sensitivity value, the added noise is negatively related to the privacy budget.

#### 7.3.4.2 Adding Laplace noise to an average query

In this example, we add Laplace noise to the averagey query about the average CU income in the CE database.

##### 7.3.4.2.1 Calculate the true average income of CUs

```
income_average <- CEdata %>%
  summarize_at(vars(Income), funs(mean))
income_average
```

```
## # A tibble: 1 x 1
##   Income
##    <dbl>
## 1 67593.
```

##### 7.3.4.2.2 Add Laplace noise to true average

We know that the sensitivity for an average query is $\Delta f = \frac{b-a}{n}$, where $n$ is the number of observations, $a$ is the lower bound and $b$ is the upper bound of the variable. Here, it is important to note that $a$ and $b$ should not depend on the sample. Rather, they should be the lower and upper bounds of the income variable itself, which is defined as "the amount of CU income before taxes in past 12 months". Clearly, different database holders might have different values for $a$ and $b$. For illustration purpose, suppose the lower bound of the income variale is \$0, and the upper bound of the income variable is \$1,000,000.

We assign the values to $a$, $b$, and calculate $\Delta f$ for the average query.

```
a <- 0
b <- 1000000
n <- nrow(CEdata)
Delta_f_average_income <- (b - a) / n
```

We again u se the `rlaplace()` function to generate a random draw from a Laplace distribution. Similar to the count query demonstrations, we use two different values of $\epsilon$: 0.1 and 1. Make sure to use the `set.seed()` function for reproducible results.

```
set.seed(123)
epsilon1 <- 0.1
rlaplace(1, income_average, Delta_f_average_income/epsilon1)
```

```
## [[1]]
## [1] 62028.67
```

```
set.seed(123)
epsilon2 <- 1
rlaplace(1, income_average, Delta_f_average_income/epsilon2)
```

```
## [[1]]
## [1] 67036.76
```

With the true income average of \$67,593, we again see that smaller privacy budget adds more noise, in line with the implication that when fixing the sensitivity value, the added noise is negatively related to privacy budget.

### 7.3.5 Additional resources about Laplace Mechanism

1. Sarathy and Muralidhar (2011): Evaluatng Laplace Noise Addition to Satisfy Differential Privacy for Numeric Data