ABV - Indian Institute of Information Technology and Management Gwalior

Information System Security

Thesis Report

An Empirical Approach to protect Data with Differential Privacy

Prepared by:

Shivansh Srivastava (2018BCS-053)

@sastava007

Submitted to:

Dr. Debanjan Sadhya

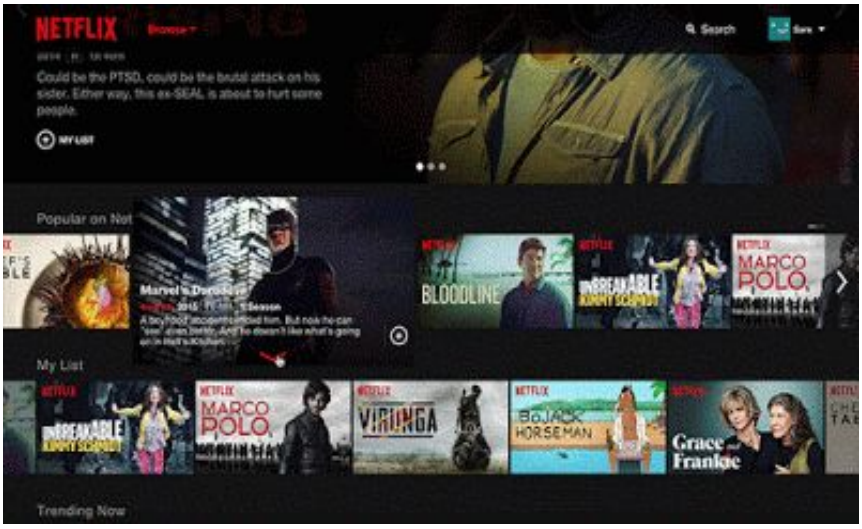
Overview

In the age of big data and machine learning, there is a strong demand for large-scale, well-annotated datasets to make better analysis and improve user experience. At the same time, privacy concerns and violations are more and more in the spotlight of society and regulators.

The problem of statistical disclosure control revealing accurate statistics about a population while preserving the privacy of individuals has a venerable history. This has been experienced many times in the past where it was possible to *reverse-engineer* many of the records in the source dataset with enough computation power and time to search all possible records the source dataset could hold, and discover that only certain data records could possibly generate in the values given in the tables.

Need of Anonymization

Netflix in 2006, announced a 1 million dollar challenge for improving their recommendation engine, together with releasing 100 million anonymized movie ratings. Although the data sets were constructed to preserve customer privacy, two researchers from The University of Texas, Austin were able to identify individual users by matching the data sets with film ratings on the IMDb.



Openly sharing customer data in 2006 was not a good idea, over the past years, the explosion in data volumes met a poorly regulated market, with little sanctions being imposed, and thus allowing excessive misuse of personal data.

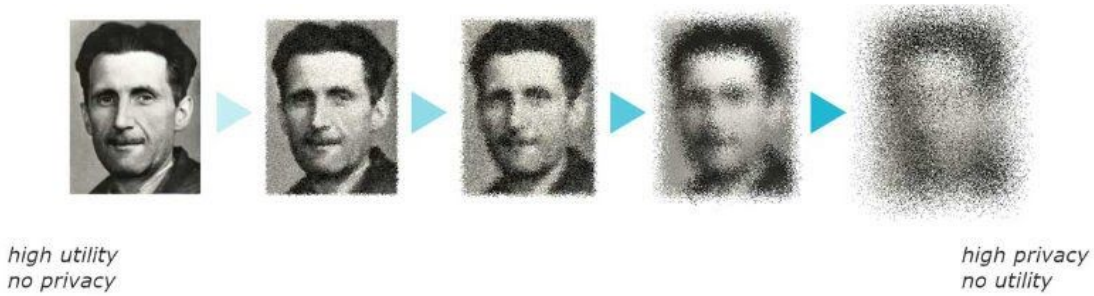
Introduction to Differential Privacy

Differential privacy is a data anonymization technique that's used by major tech companies to collect and share aggregate information about user habits while maintaining the privacy of individual users.

Differential Privacy is among the most innovative methods of cybersecurity that allows data analysts to build accurate models without sacrificing the privacy of the individual subjects by introducing randomness into the process of data retrieval. Practically, it enables us to quantify the level of privacy of a database by making the tradeoff between the amount of noise added and accuracy.

Trade-off: Privacy vs. Accuracy

Typically, differential privacy works by adding some noise to the data (Laplacian mechanism). The amount of added noise is a trade-off, adding more noise makes the data more anonymous, but it also makes the data less useful. In differential privacy, this trade-off is known as a privacy **budget** which is formally controlled using a parameter called epsilon **(ε)**.



Requirements Analysis and Specification

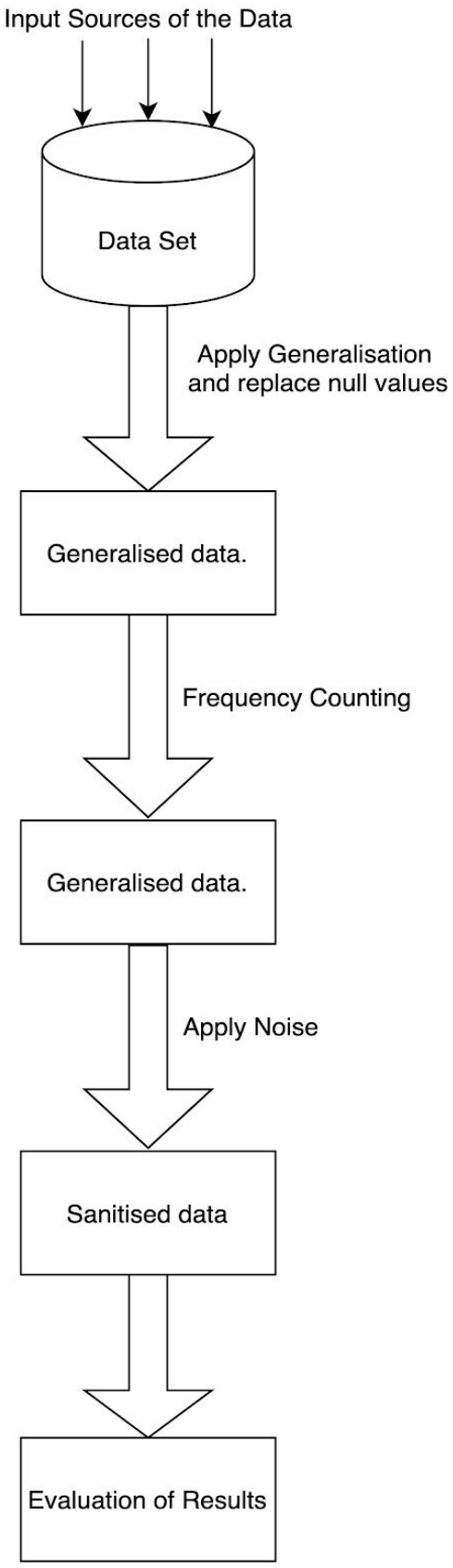
We are making this application in python running on a Google Colab notebook. The basic requirements for this project are:

1. Hardware Requirements
    a. A system with moderate computational capabilities and having a high bandwidth internet connection.
2. Software Requirements
    a. A common web browser in the latest version. (Google Chrome or Mozilla Firefox)
    b. Python 3
    c. Anaconda Navigator
    d. PIP
3. Libraries and other tools
    a. jupyter
    b. jupyter-client
    c. jupyter-console
    d. jupyter-core
    e. jupyterlab
    f. jupyterlab-pygments
    g. jupyterlab-server
    h. matplotlib
    i. numpy
    j. opendp-smartnoise
    k. pandas
    l. scipy
    m. seaborn
    n. z3-solver

System Architecture and Methodology

There are various steps involved in making this project. First, the data is preprocessed and generalized into various categories and numerical values. The null values are replaced with the values that appear most often in the attributes of the dataset.

Then we are counting the frequency of values after applying Differential Privacy using the opendp-smartnoise function dp_histogram. The comparison is made on the new differentially private data and on the actual data.



Data Flow Diagram of the proposed system

Case Study

An analysis of Mental Health in Tech Survey with/ without preserving data Privacy

The purpose of this demo is to showcase the utility of **OpenDP** differential privacy framework by making statistical queries to data with and without privacy-preserving mechanisms. As we compare query results side-by-side, we show that conclusions about the data are similar in both settings: without a privacy-preserving mechanism, and with differential privacy mechanism.

1. Data Set

The mental health in tech survey data set is released by **OSMI** which consists of 27 questions, answered by 1,259 volunteers. The data used in the analysis were preprocessed i.e the original age, gender, and country variables were mapped into categories for the analysis.

Overview of each attribute

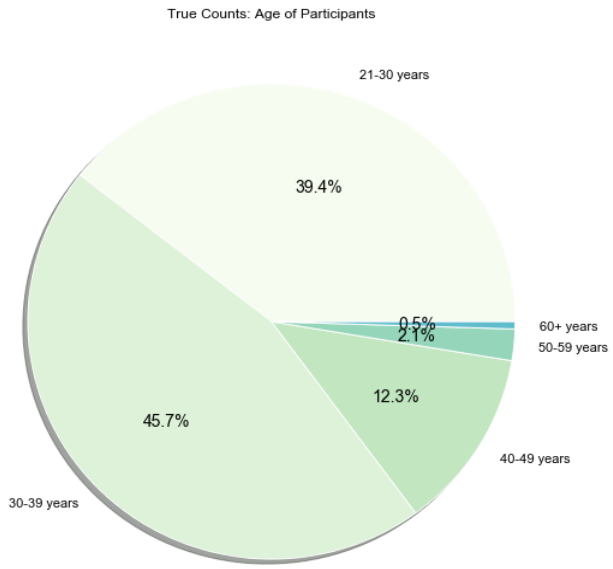
- **age**: age of the participant. Categorized as follows: 21-30yo (0), 31-40yo (1), 41-50yo (2), 51-60yo (3), 60yo+ (4).
- **gender**: gender declared by the participant. Categorized as follows: Male/Man (1), Female/Woman(2), all other inputs (0).
- **country**: participant's country of residence. Categorized as follows: United States (1), United Kingdom (2), Canada (3), other countries (0).
- **remote_work**: Binary value that indicates if participant works remotely more than 50% of the time.
- **family_history**: Binary value that indicates if the participant has a family history of mental illness.
- **treatment**: Binary value that indicates if the participants have sought treatment for mental illness.

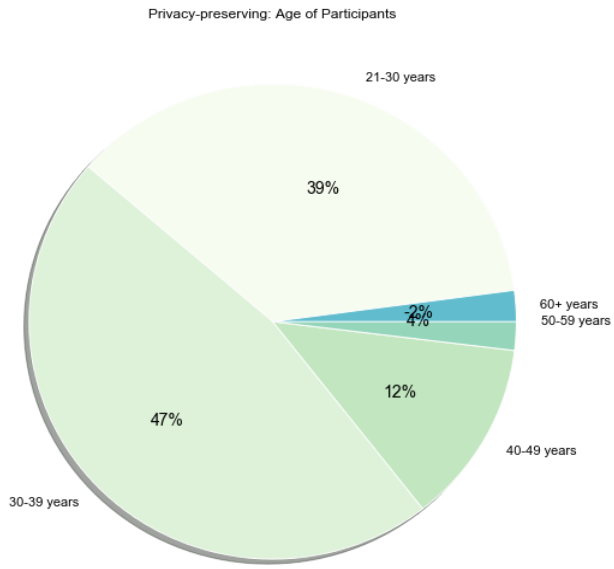
Now, we will make statistical queries on different variables to generate a comparative analysis of results obtained from data with/ without differential privacy mechanisms.

1. Age

The true age distribution is: **[478, 554, 149, 26, 6]**

The age histogram obtained using DP is: **[458 555 148 48 25]**

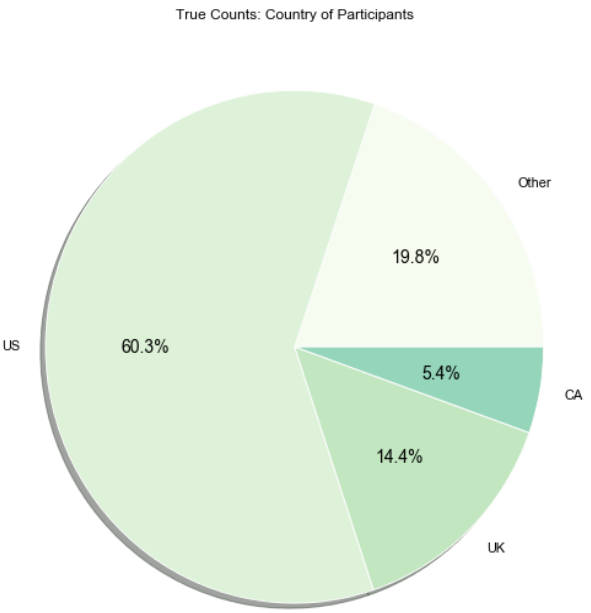


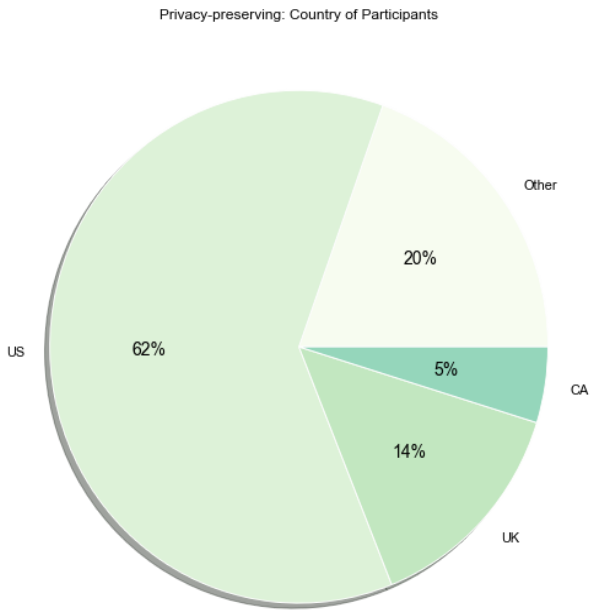


2. Country

The true country distribution is: **[240, 732, 175, 66]**

The country histogram obtained using DP is: **[257, 810, 186, 63]**

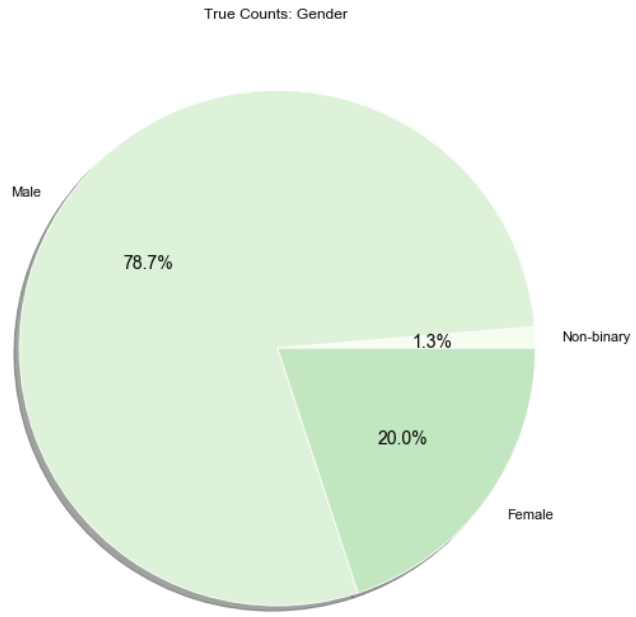


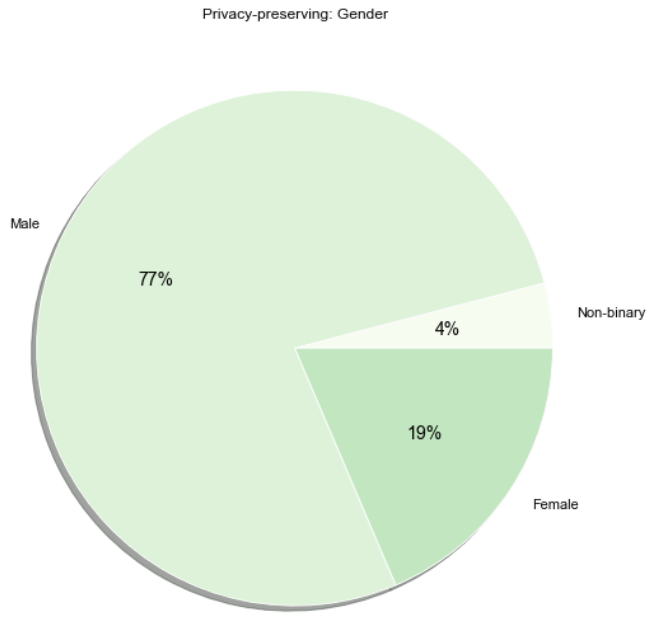


3. Gender

The true gender distribution is: **[16, 955, 242]**

The gender histogram obtained using DP is: **[52, 990, 238]**

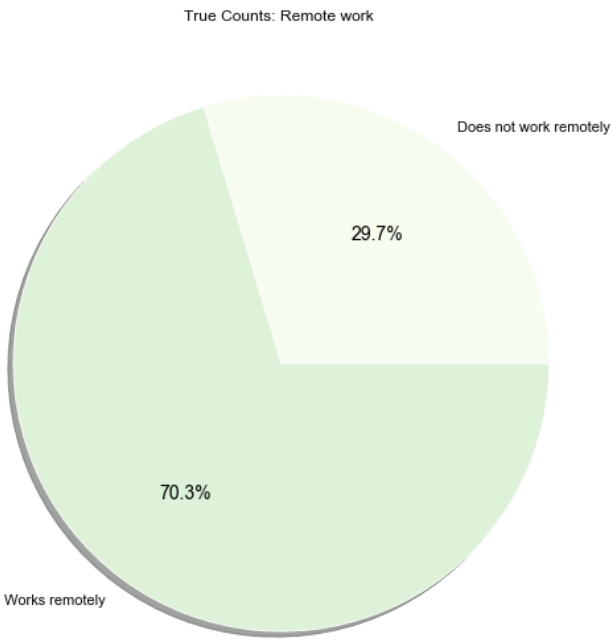


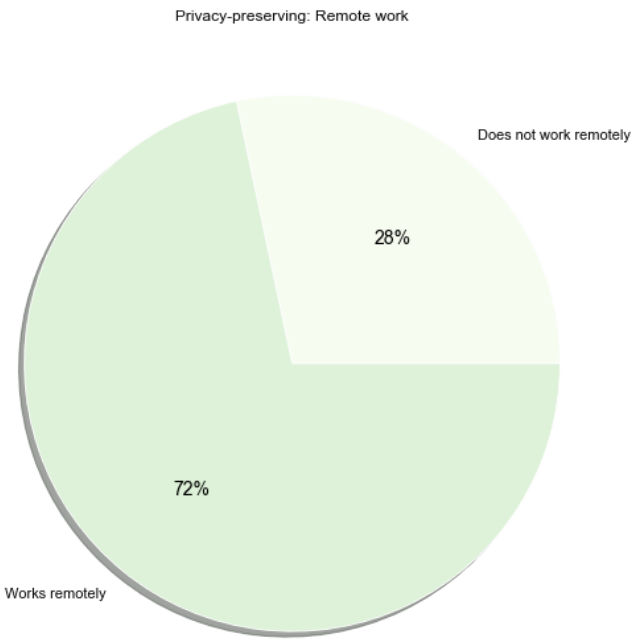


4. Remote Work

The true remote work distribution is: **[360, 853]**

The remote work histogram obtained using DP is: **[355, 898]**

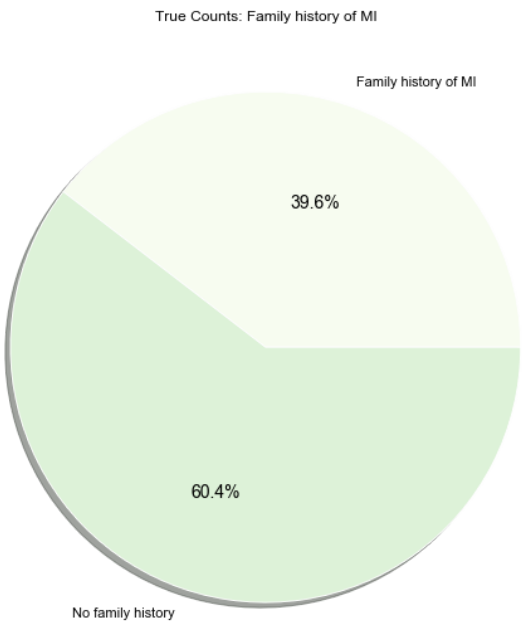


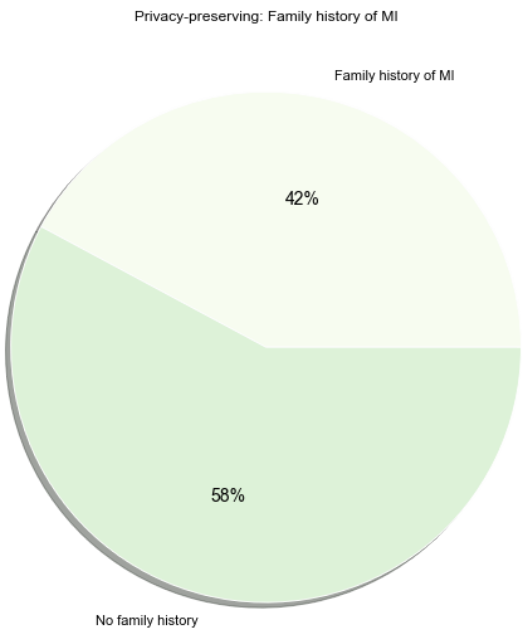


5.Family History

The true count of participants with a family history of mental illness is: **[480, 733]**

The family history histogram obtained using DP is: **[517, 709]**

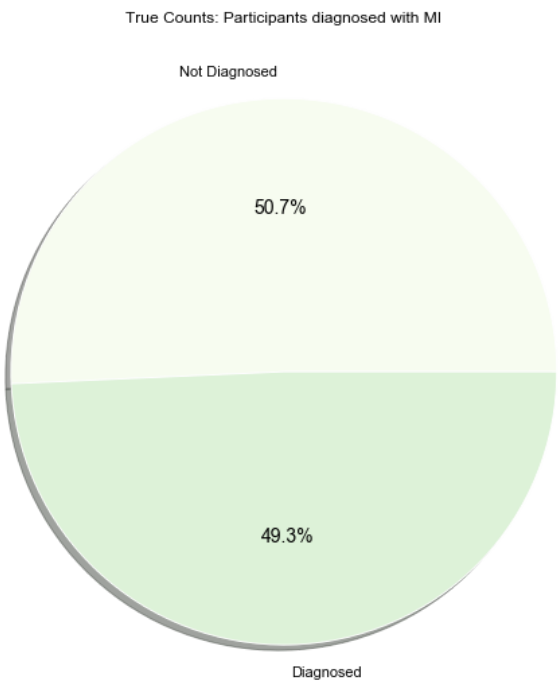


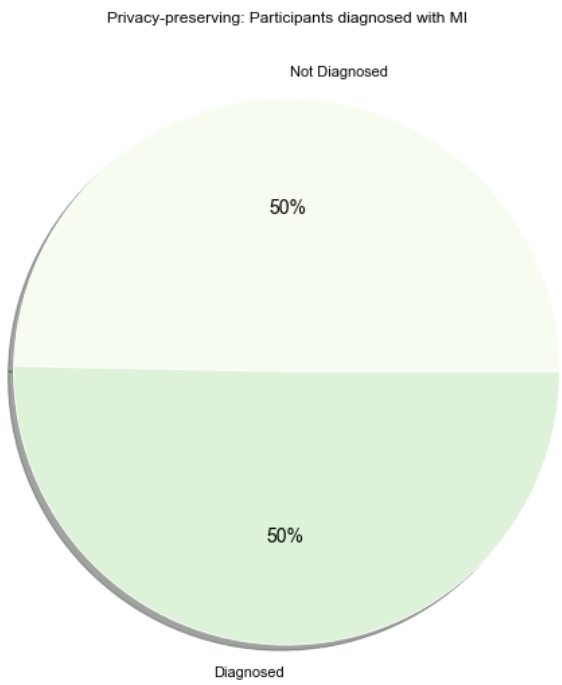


6.Treatment

The true count of participants diagnosed with mental illness is: **[619, 598]**

The mental illness obtained using DP is: **[604, 612]**





Usefulness of Differential Privacy in handling the attack on an individual's data

In this demo, we will examine perhaps the simplest possible attack on an individual's private data and what the differential privacy can do to mitigate it. We are considering a dataset of 10,000 people having attributes like (*name, sex, age, education, income, married, race*).

```
person of interest:

sex           0.0
age          45.0
educ          6.0
income     6000.0
married       1.0
race          1.0
Name: 0, dtype: float64
```

Consider an attacker who knows everything about the data except for the person of interest's (POI) income, which is considered private. They can back out the individual's income very easily, just by asking for the mean overall mean income.

```
POI_income = overall_mean * n_obs - known_mean * known_obs
```

But if the attackers were made to interact with the data through differential privacy and were given a privacy budget of **ε = 1**. Now, they should use tighter data bounds than they know are actually in the data in order to get a less noisy estimate and need to update their known_mean accordingly.

Upon executing the code in the attached Jupyter notebook the result we got are as follows:

```
Known Mean Income: 26886.001600160016
Observed Mean Income: 26883.930944271226
Estimated POI Income: 6179.4427122677835
True POI Income: 6000.0
```

Conclusion

In this project, we've reviewed the theory of differential privacy and seen how it can be used to quantify privacy. Differential privacy is a powerful tool for quantifying and solving practical problems related to privacy.

Through our first simulation on *mental health data*, we've shown that Differential privacy guarantees that anyone seeing the result of a differentially private analysis will essentially make the same inference about any individual's private information, whether or not that individual's private information is included in the input to the analysis.

Whereas in our second case study of *finding the income of POI*, we've seen that DP provides a provable guarantee of privacy protection against a wide range of privacy attacks (include differencing attacks, linkage attacks, and reconstruction attacks).