**Plan**

* ~~Workshops~~
* Read entire guide and all resources
* Read about libraries we need?

**Notes**

**Questions we want to answer**

1. How to decide on epsilon (larger the value of ε, the less private and the more accurate the output will be)
2. How to determine the accuracy of our synthetic data

**Numeric Columns**

**passenger\_count float64**

**trip\_distance float64**

**payment\_type int64**

**fare\_amount float64**

**extra float64**

**mta\_tax float64**

**tip\_amount float64**

**tolls\_amount float64**

**improvement\_surcharge float64**

**total\_amount float64**

**congestion\_surcharge float64**

**Airport\_fee float6**

**Categorical Columns**

**VendorID int64**

**RatecodeID float64**

**PULocationID int64**

**DOLocationID int64**

**??? Columns**

**tpep\_pickup\_datetime object**

**tpep\_dropoff\_datetime object**

**store\_and\_fwd\_flag object**

**To-do’s**

* **Data exploration:** 
  + “[sensitivity](https://becominghuman.ai/query-sensitivity-types-and-effects-on-differential-privacy-mechanism-c94fd14b9837)”
  + desired Epsilon
  + desired delta
  + [noise-adding mechanism](https://becominghuman.ai/differential-privacy-noise-adding-mechanisms-ede242dcbb2e)
  + decide what we want to learn
  + Bounds → lower and upper bounds are min and max of column
* **Data analysis**
  + Get statistics from data (mean…)
* **One-way marginal vs Two-way marginal**
* Decided: One-way marginal for every column to create new dataset
* Bounds are min and max for each column
* **Noise — Laplace**
  + [PyDP Laplace tutorial with math explained & example code](https://github.com/OpenMined/PyDP/blob/dev/examples/laplace_demo/laplace.ipynb)
  + Skeleton code/demos: [PyDP](https://github.com/OpenMined/PyDP/tree/dev/examples)
* **Pick an epsilon value**
* **Verify accuracy by comparing two databases**

One of the most commonly used mechanisms for adding noise is *Laplace mechanism*

Maybe one method of introducing noise? This is where Local DP comes in, instead of asking the exact amount, we ask the individuals to *add any random value (noise)* in the range of -100 to +100 to the amount they hold in their pockets and give us just the resultant **sum** of it. That is if ‘X’ had 30$ in his/her pocket by adding a random number say -10 to it, (30 + (-10) ), they give us *just the result*, which is 20$ in this case. Thus, preserving their individual privacy.

**Differential Privacy Synthetic data**

- produce a “de-identified” or “anonymized” version of data, redacted identifying info

- same schema and attempts to maintain properties of the original dataset (e.g., correlations between attributes) - but it provides a provable privacy guarantee

-**Generating**

- Build **probabilistic model** of the underlying population from which the original data is sampled

- newly generated data will retain all the properties of that population

- each data point rep “fake”, nonexistent individual

- epsilon → measure of accuracy

- histogram → Noisy → **1 way marginal** (distribution of eahc attibute , only consider one attribute of the data and ignores the correlation)or **two way margin**(distribution of both attributes simutaneously, but weaker signal relative to noise of each option)

- used to generate “fake” stae

<https://github.com/tensorflow/privacy> –

<https://www.tensorflow.org/responsible_ai/privacy/tutorials/classification_privacy>

<https://neptune.ai/blog/using-differential-privacy-to-build-secure-models-tools-methods-best-practices> <https://medium.com/@shaistha24/global-vs-local-differential-privacy-56b45eb22168> has a whole bunch of super simple examples

<http://www.gautamkamath.com/CS860notes/lec4.pdf> laplace mechanism overview

<https://programming-dp.com/ch5.html> sensitivity overview