**CS1699:**

**Privacy in the**

**Electronic Society**

Project 3

*Henri Frelin*

4 - 20 - 2019

W0: The dataset I chose is from Kaggle, and it contains heart disease information from a University of California health study. The data includes points such as age, sex, chest pain, blood pressure, electrocardiographic results, depression, and even number of colored vessels on body. I chose this dataset because many of these datapoints would be very specific to the individual, and from a medical perspective, identifiers such as resting blood sugar when combined with physical markings on the body can lead to very revealing quasi-identifiers. This dataset is intended for determining the health of the heart or risk for heart disease based on these variables and combinations thereof. The dataset has 14 individual variables and 304 total entries. <https://www.kaggle.com/ronitf/heart-disease-uci>

W1: The release of this dataset could potentially provide information that would later be used in conjunction with other quasi-identifiers to expose potentially sensitive medical information. The potential for such an attack is made possible by the presence of quasi-identifiers such as age, sex, and physical markings on the body in conjunction with variables that expose heart issues and potential heart disease in the individuals. The danger does not come solely from the release of this dataset, but rather, this dataset could be used in conjunction with a secondary medical record or external dataset that contains the names of individuals, and perhaps a matching variable or combination of quasi-identifiers. Examples of quasi-identifiers that could be used to create such a link in this scenario are age, sex, fasting blood sugar, resting blood sugar, and number of vessels exposed on body. These variables would, in conjunction, create a unique fingerprint or identity that could be used in conjunction with an external dataset to potentially link and expose the sensitive data such as heart disease.

C2: *See P3script.py and C2.csv for code and resulting dataset*

W3: The changes that have been made to the dataset lie in the most revealing quasi-identifiers. Those being age, exact blood pressure in mm/Hg, and exact serum cholesterol in mg/dl. These quasi-identifiers were particularly important to obscure with at least 2(k)-anonymity by means of grouping and generalization because of their ability to be linked back to a particular individual. Each one of these individually may be interchangeable with another example of a patient in a hospital’s medical record, but when combined together, they could be used to correctly identify a large percentage of independent individuals. I have used a python dataframe masking function to add ranges and group each one of these quasi-identifiers to obscure the data, but chose to leave other, less revealing variables in their original state as to preserve the utility of the dataset. This dataset was not particularly threatening on its own, but rather when used as a supplementary quasi-identifier with other medical records, so I would say that with the addition of these generalizations there is a very small likelihood of sensitive information being revealed. Along with decreased likelihood of sensitive medical data leaking, there is only a minor impact on utility of the data. For example, if a data analyst studying heart disease factors wanted to analyze the chest pain of a user in a 10 year age bracket in comparison to their exact maximum heart rate, this is still possible with the altered dataset. However, the utility is slightly limited in the sense that once can no longer choose to analyze only 32 year olds, or resting blood pressure with higher precision than 30 mm Hg.

C4: See *P3script.py* for algorithm finding *“What percentage of males between the age of 40-50 have cholesterol between 200 & 300”* algorithm implementation

W5: The algorithm that derives the particular insight, “*What percentage of males between the age of 40-50 have cholesterol between 200 & 300*?”, can be seen in *P3script.py*, and it functions by creating a new datafame, then finding the total count of all sexes, then determining what percentage are male. Next, it queries through the dataframe object for subjects that satisfy both constraints of the question. We then count the results, divide by the total males in the population, and multiply by 100 to attain a percentage. An adversary would easily be able to differentiate between this dataset D and another altered by one value D’, because there is no attempt to mask what this simple algorithm is computing. For example, if we add or subtract a male that satisfies this particular query, then the percentage that the algorithm produces will certainly be lower in D’ than in D.

C6: See *P3script.py* for algorithm implementing a Laplace Distribution Variable to satisfy differential privacy. When run, you can choose b in 1/b-differential privacy through the text prompt.

W7: To attain 1/b differential privacy, I am adding noise to the output of the algorithm in the form of a Laplacian noise value. To determine the correct scale parameter for in the Laplacian distribution, I tested the impact of removing a single user to determine the maximum amount by which a subject could affect the output of the algorithm. It was determined that the maximum effect that a single user can have on the output of the algorithm is *25.604% - 25.243% = .361%.* With this result in mind, I use .361 \* b as the input scale parameter for the Laplacian distribution, where b represents 1/b-differential privacy. Running the script will prompt you to enter the variable b at your own discretion to test the output, allowing you to choose the level of differential privacy. This algorithm respects user privacy by using the maximum impact that a user can have on the original dataset by being removed and adding noise to the output of the algorithm to account for this potential difference. This prevents an adversary from having the ability to differentiate between the two logical datasets D and D’, and subsequently prevents attacks such as reverse engineering the algorithm that produces the result to de-anonymize.