CSCI 2390 Assignment 3: Differential Privacy in Practice

Question 1: *Look at the result of this count query. Note that it does not include any name, email, or other personally identifiable information. What can you nevertheless learn about the TA's musical tastes? What possible genres might they have chosen? Alternatively, what genres is it impossible for them to have chosen?*

We can learn that the TA's music tastes must be in the defined music categories listed or that they preferred not to answer. So, the music genre the TA chose must be Classical, House, Country, Hip Hop, Pop, Rock, Metal, or they preferred not to answer. Alternatively, it would be impossible for them to have chosen genres like Indie, Folk, or Gospel. If we knew more about the TA's age, we could narrow down music genres that they more likely have or haven't chosen. For example, someone on the younger end around 19 years old may be unlikely to have had their chosen genre be Metal while it may be likely if the TA is around 30 years old.

Question 2: What did you find out about the TA? Are your findings consistent with Question 1? Combine the two together to learn the TA's exact age.

I found out that the TA, Kinan Bab, appears to have posted on facebook quotes by Metallica and once updated his cover photo to be Sleep - Dopesmoker, which suggests that his favorite genre is likely Metal. This is consistent with Question 1, so the TA is likely 29 or 30. His birth year was also 1994 on Facebook, which corresponds to him being 28 or 29 years old. Putting these two together, the TA's exact age would be 29.

Question 3: Identify the TA's favorite color. What is it? How easy or obvious is this to do, and why?

The TA's favorite color is thus Black. This was pretty easy/obvious, as there were only 2 29 year-old responses, with their music genre choices being Metal and 'Prefer not to answer' for Black and Yellow respectively. Since we already established that Kinan's favorite genre is likely Metal, and other Black-ish image posts on Facebook support this assumption, we can identify Kinan's favorite color as Black.

Question 4: What information can you learn about the TA's favorite sport from the above query?

Based on the above query alone, we see that ages of 25 or more who responded with their favorite music genre as Metal also chose Soccer as their favorite sport. So, we would likely identify the TA's favorite sport to be Soccer.

Question 5: What is our TA's favorite sport?

The TA's favorite sport is Soccer. We know this as the person aged 29 selected their favorite sport to be Soccer. Also, even those close to 29 (26 and 30) chose soccer, so we can be even more confident about the TA's favorite sport.

Question 6: Run `dp.py` several times varying the epsilon privacy parameter for different values between 10 and 0.01, like so:

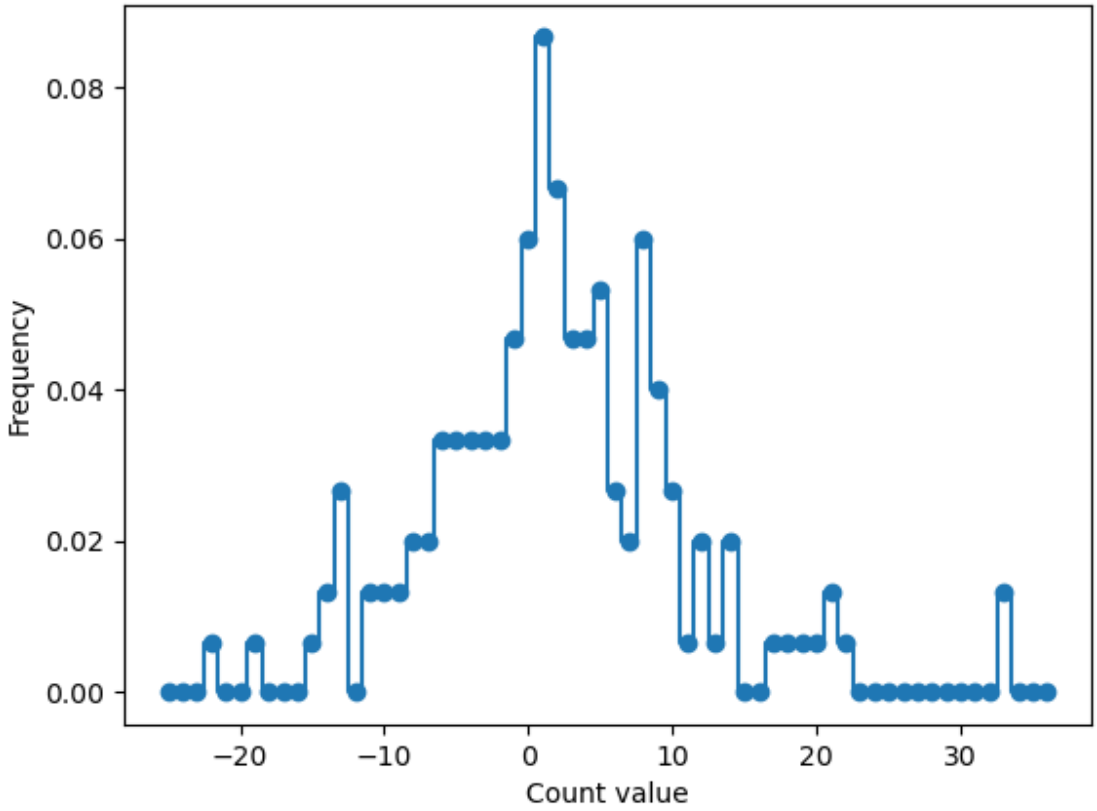
```
$ python3 dp.py 0.01
[...]
$ python3 dp.py 0.1
[...]
[etc.]
```

What happens when the privacy parameter grows larger or smaller? How does that affect privacy?

As the epsilon privacy parameter gets smaller and smaller, the differences in count values for each demographic appear much larger and with greater variation. As the privacy parameter grows larger, it becomes more similar to the actual distribution. This suggests that as the privacy parameter grows smaller, we have higher privacy and more noise, and when it gets bigger we have less privacy and noise.

Question 7: Look at the plot generated with privacy parameter epsilon = 0.5. What is the most likely value? What is the expected (i.e., average) value? How do they relate to the actual value (i.e., the query excuted without any noise via `client.py`)? How does the plot change for different values of the privacy parameter?
Please include the generated plot for epsilon = 0.5 in your submission.

The most likely value looks like it's slightly above 0, at a value of 1. The expected value also appears to be around this value of 1, but slightly higher due to the slight skew. Compared to the actual value, the expected/most likely value is similar, as most of the unique age-music taste-combination people have only a count of 1. However, a noticeable difference is that the actual value doesn't have values that are negative. Generated plot for epsilon = 0.5:



Question 8: Run the composition attack against the average age grouped by programming experience. What can you deduce from the exposed averages about the programming experience level of our TA? How confident are you in what you have deduced? Are there scenarios where they might be wrong?

From the exposed averages about the programming experience level of our TA, I found that the AVG age for people with programming experience of greater than 10 years was 28 and that the other age categories were far away from 28 (even those with 8-10 years of programming experience had only an average of 23). Based on this, we can be quite confident that our TA has had more than 10 years of experience, as we had concluded earlier that his age was 29.

Question 9: Reuse your composition attack from question 8 to compute the exact non-noised counts per programming experience level. Deduce the programming experience level of our TA, with high confidence, by looking at both the exposed counts

and the previously exposed averages. Now summarize everything you've learned about the TA!

When we look at both the exposed counts and the previously exposed averages then we can deduce with even more high confidence that it is likely that our TA had more than 10 years of experience, as the count of the exposed programmers with more than 10 years was 4, which suggests a relatively low amount of people with more than 10 years of experience. This makes it much more likely that Kinan has more than 10 years of experience when we consider that those with more than 10 years of experience were also relatively much older and that those in this category were few.

Question 10: Does the class you implemented suffice to truly enforce that a dataset is never used beyond a certain privacy budget? Can developers intentionally or unintentionally over-use the dataset beyond the privacy budget? At a very high level, how would you design a different privacy budget enforcment mechanism that does not suffer these drawbacks?

The implemented class helps provide a basic mechanism to track and enforce privacy budgets for each query made on the dataset, but has several limitations and potential drawbacks. First of all, the current class uses a fixed privacy cost of EPSILON = 0.5 , which would not be practical in many realistic scenarios in which certain queries can leak more privacy and access more sensitive data than others. Additionally, developers can unintentionally or intentionally over-use the dataset beyond its intended privacy budget by simply directly calling the ‘avg’, ‘count’, or ‘count0’ functions from the client without going through the BudgetTracker. Developers may also unintentionally or intentionally create multiple instances of the privacy budget, leading to over-use.

To remedy these drawbacks at a high level, we can design a different privacy budget enforcement mechanism in which the raw query functions ‘avg’, ‘count’, and ‘count0’ are both encapsulated and hidden from direct access, making it that any access must adjust the privacy budget accordingly. Additionally, if we consider that multiple developers may be accessing the dataset, we may not want each developer to create their own BudgetTracker as this could lead to each developer being under budget but the sum total being over budget, and should thus implement some centralized component responsible for executing queries, enforcing privacy budges, and logging/query monitoring. We can also consider more advanced features like budget enforcement middleware or query-wrapper, rate limiting / request throttling mechanism to avoid intentional/malicious timing attacks, or even some sort of token-based encrypted authorization to validate privacy access requests.