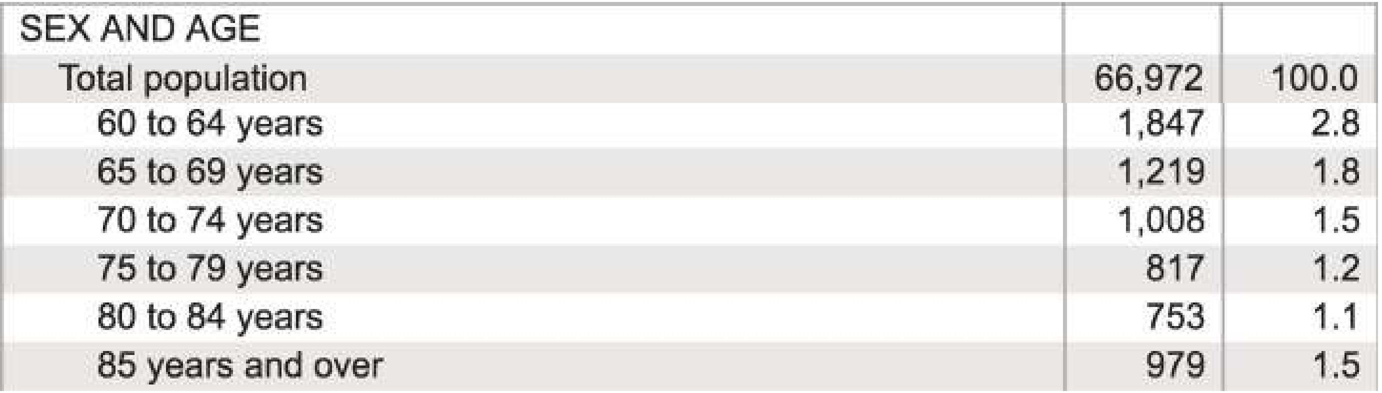
Data Hierarchies in Support of Disclosure Limitation

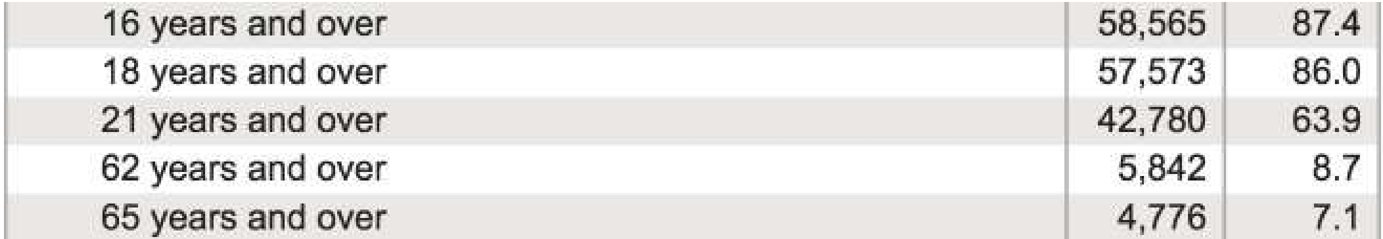
Keith Merrill, Brandeis University
Shawn Merrill, Purdue University

Transitioning to Differential Privacy

- Many Census products make implicit or explicit use of hierarchies, e.g. histogram or contingency table bins released at multiple levels



SEX AND AGE		
Total population	66,972	100.0
60 to 64 years	1,847	2.8
65 to 69 years	1,219	1.8
70 to 74 years	1,008	1.5
75 to 79 years	817	1.2
80 to 84 years	753	1.1
85 years and over	979	1.5



16 years and over	58,565	87.4
18 years and over	57,573	86.0
21 years and over	42,780	63.9
62 years and over	5,842	8.7
65 years and over	4,776	7.1

Transitioning to Differential Privacy

- These hierarchies are established by Subject Matter Experts (SMEs), and are frequently “inherited” from past years

- From a privacy perspective, data-dependent decisions leak (potentially unbounded) information, meaning that as we transition to a formal privacy definition (differential privacy), SMEs need to be experts in both their subject matter and the privacy literature.

- Can we eliminate the need for this?

Problems with Hierarchies

- Bins with small counts may also be washed out by noise, eliminating utility for end users interested in those groups.
    - There exist methods for dealing with this, e.g. introducing relative noise as opposed to absolute noise (like iReduct proposes, for instance), but a choice of bins which avoids the situation is still preferable.
- Data dependent hierarchies can leak information.
- Poorly chosen/naive hierarchies can result in semantically uninteresting bins:
    - Equal-width bins may group semantically dissimilar populations, e.g., a 16-20 year old bin would split the high school and college demographics, but combines portions of them.
    - Equal-depth bins may force dissimilar groups to be clumped together, e.g. in a college town, older working age adults and retirees may be grouped together.

Present Challenges

In the transition to provable privacy, two obvious questions need to be addressed:

1. Since every query now cuts into our privacy budget, when should we create a new table by combining smaller ones, as opposed to querying the data again?
   a. Combining results does not count against the budget, but does have much larger variance in the answers than running another query
2. Can we automate the process of hierarchy generation, to remove/lessen the need for SMEs to be savvy in the privacy literature?
   a. Certainly the answer is yes, but how to do this in a way which minimizes utility loss.

A Naive Approach to Problem 1

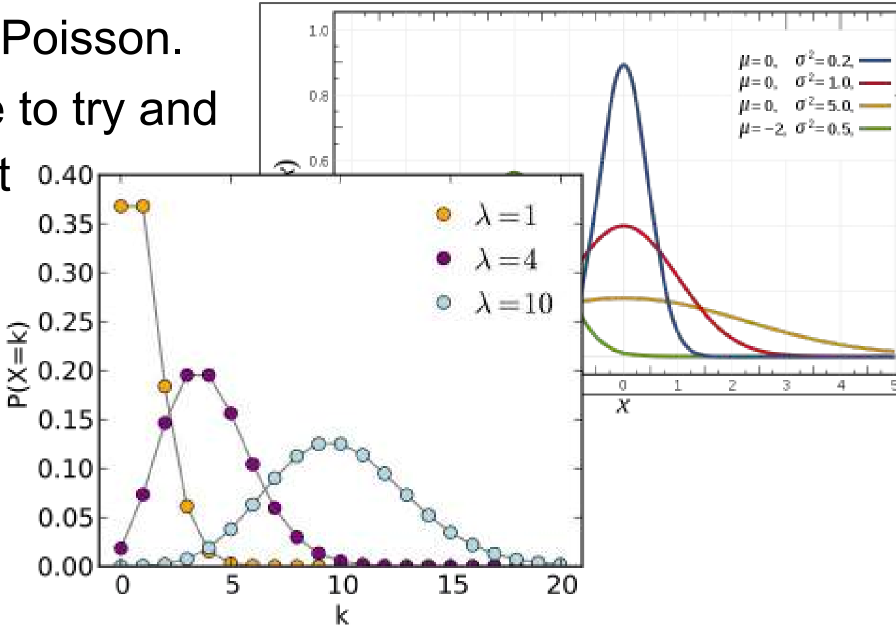
A solution should take into account 2 crucial ideas:

1. When creating a new table, we can combine values from smaller tables, or use complements:
    a. If we have results for [0-18], [18-21], [21-62], [62-65], [65+] and we want [0-21], [21+], we could either combine [21-62], [62-65], [65+] to compute [21+], or subtract [0-21] from the total population size. This latter option would have less variance.
2. Histogram queries have 1 sensitivity no matter how many boxes they have. We can use this to gain "free" queries.
    a. E.g. if we need to query [65+] (for medicare), we can query [<65] as well or we can query [0-18], [19-30], [31-65], etc. This allows for freedom to query other important sections of the data

By combining these rules we can choose a set of results to query that uses as little of the budget as possible while maintaining sufficient accuracy

Some Musings on Problem 2

- Write an algorithm which will make an educated guess about the underlying distribution of the population our sample was taken from. For example, age might be roughly a normal distribution, whereas number of Justin Bieber albums owned is probably more akin to Poisson.
- Query the data as few times as possible to try and determine a “good fit” for the data in that dimension.
  - We suggest potential bins for each candidate distribution and compute a chi squared test statistic. We select the distribution with the minimum value from among these.
  - Given that choice of distribution, we construct a hierarchy which conforms to our expectations.



Recognizing Distributions -- A First Step

- This technique requires touching the data a number of times, for each dimension. Can these queries be low sensitivity? How does that affect the prediction of the algorithm, and more importantly the utility of the released tables? How to measure the loss of utility?
- Have further experiments to run on how noisy the queries can be, but initial testing indicates that when a random sample is taken from a given population, the algorithm can correctly recognize which distribution was used, even for relatively small values of epsilon