Responsible Data Science

Differential privacy

March 23, 2022

Prof. Julia Stoyanovich

Center for Data Science &
Computer Science and Engineering
New York University

query sensitivity & composition

Query sensitivity

The ℓ_1 sensitivity of a query ***q***, denoted ***Δq***, is the maximum difference in the result of that query on a pair of neighboring databases

$$\Delta q = \max_{D,D'} \left| q(D) - q(D') \right|$$

lower ε = stronger privacy

- Example 1: counting queries
  - "How many elements in ***D*** satisfy property ***P*** ?" **What's *Δq* ?**
  - "What fraction of the elements in ***D*** satisfy property ***P*** ?"
- Example 2: max / min
  - "What is the maximum employee salary in ***D*** ?" **What's *Δq* ?**

Intuition: for a given ε, the higher the sensitivity, the more noise we need to add to meet the privacy guarantee

Query sensitivity

The ℓ_1 sensitivity of a query ***q***, denoted ***Δq***, is the maximum difference in the result of that query on a pair of neighboring databases

$$\Delta q = \max_{D,D'} \left| q(D) - q(D') \right|$$

query *q*	**query sensitivity *Δq***
select count(*) from D	1
select count(*) from D where sex = Male and age > 30	?

r/ai

Query sensitivity

The ℓ_1 sensitivity of a query ***q***, denoted ***Δq***, is the maximum difference in the result of that query on a pair of neighboring databases

$$\Delta q = \max_{D,D'} \left| q(D) - q(D') \right|$$

query *q*	**query sensitivity *Δq***
select count(*) from D	1
select count(*) from D where sex = Male and age > 30	1
select MAX(salary) from D	?

r/ai

Query sensitivity

The ℓ_1 sensitivity of a query ***q***, denoted ***Δq***, is the maximum difference in the result of that query on a pair of neighboring databases

$$\Delta q = \max_{D,D'} \left| q(D) - q(D') \right|$$

query *q*	**query sensitivity *Δq***
select count(*) from D	1
select count(*) from D where sex = Male and age > 30	1
select MAX(salary) from D	*MAX(salary)-MIN(salary)*
select gender, count(*) from D group by gender	?

r/ai

Query sensitivity

The ℓ_1 sensitivity of a query ***q***, denoted ***Δq***, is the maximum difference in the result of that query on a pair of neighboring databases

$$\Delta q = \max_{D,D'} \left| q(D) - q(D') \right|$$

query *q*	query sensitivity *Δq*
select count(*) from D	1
select count(*) from D where sex = Male and age > 30	1
select MAX(salary) from D	*MAX(salary)-MIN(salary)*
select gender, count(*) from D group by gender	1 (disjoint groups, presence or absence of one tuple impacts only one of the counts)

r/ai

Query sensitivity

The ℓ_1 sensitivity of a query ***q***, denoted ***Δq***, is the maximum difference in the result of that query on a pair of neighboring databases

$$\Delta q = \max_{D,D'} \left| q(D) - q(D') \right|$$

query *q*	**query sensitivity *Δq***
select gender, count(*) from D group by gender	**1** (**disjoint groups**, presence or absence of one tuple impacts only one of the counts)
an arbitrary list of *m* counting queries	?

Query sensitivity

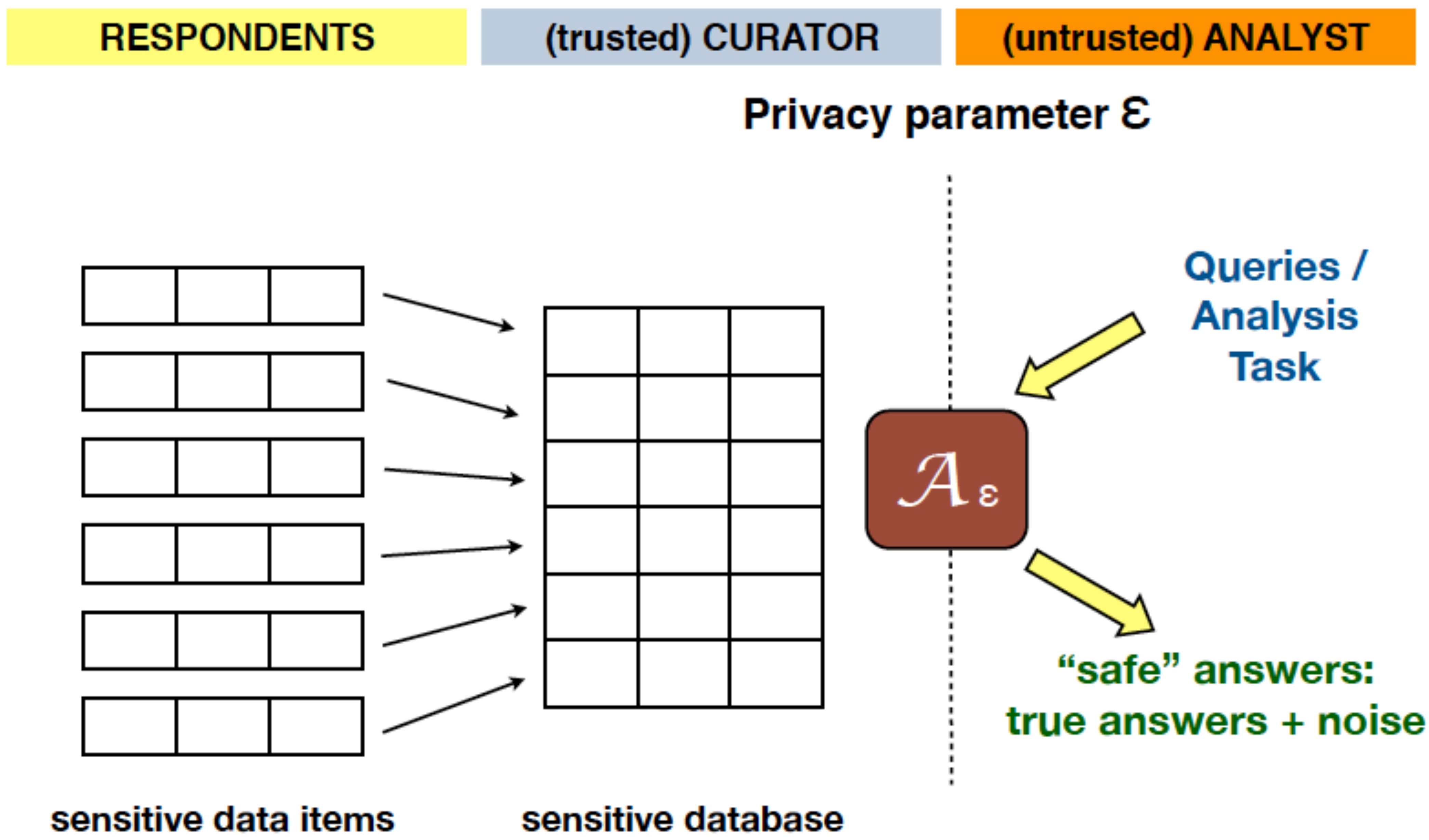
The ℓ_1 sensitivity of a query ***q***, denoted ***Δq***, is the maximum difference in the result of that query on a pair of neighboring databases

$$\Delta q = \max_{D,D'} \left| q(D) - q(D') \right|$$

query *q*	**query sensitivity *Δq***
select gender, count(*) from D group by gender	**1** (**disjoint groups**, presence or absence of one tuple impacts only one of the counts)
an arbitrary list of *m* counting queries	***m*** (no assumptions about the queries, and so a single individual may change the answer of **every query** by 1)

r/ai

Adding noise

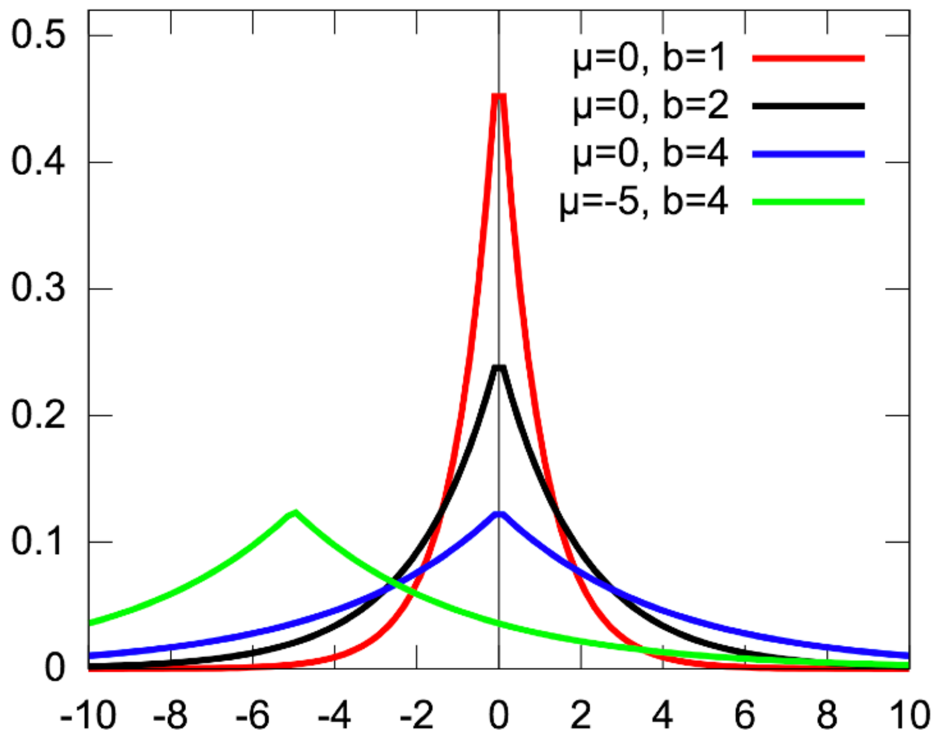


Adding noise

Use the **Laplace mechanism** to answer ***q*** in a way that's ***ε***-differentially private

$$M(\varepsilon): q(D) + Lap\left(\frac{\Delta q}{\varepsilon}\right)$$

The Laplace distribution, centered at 0 with scale ***b***, denoted ***Lap(b)***, is the distribution with probability density function:



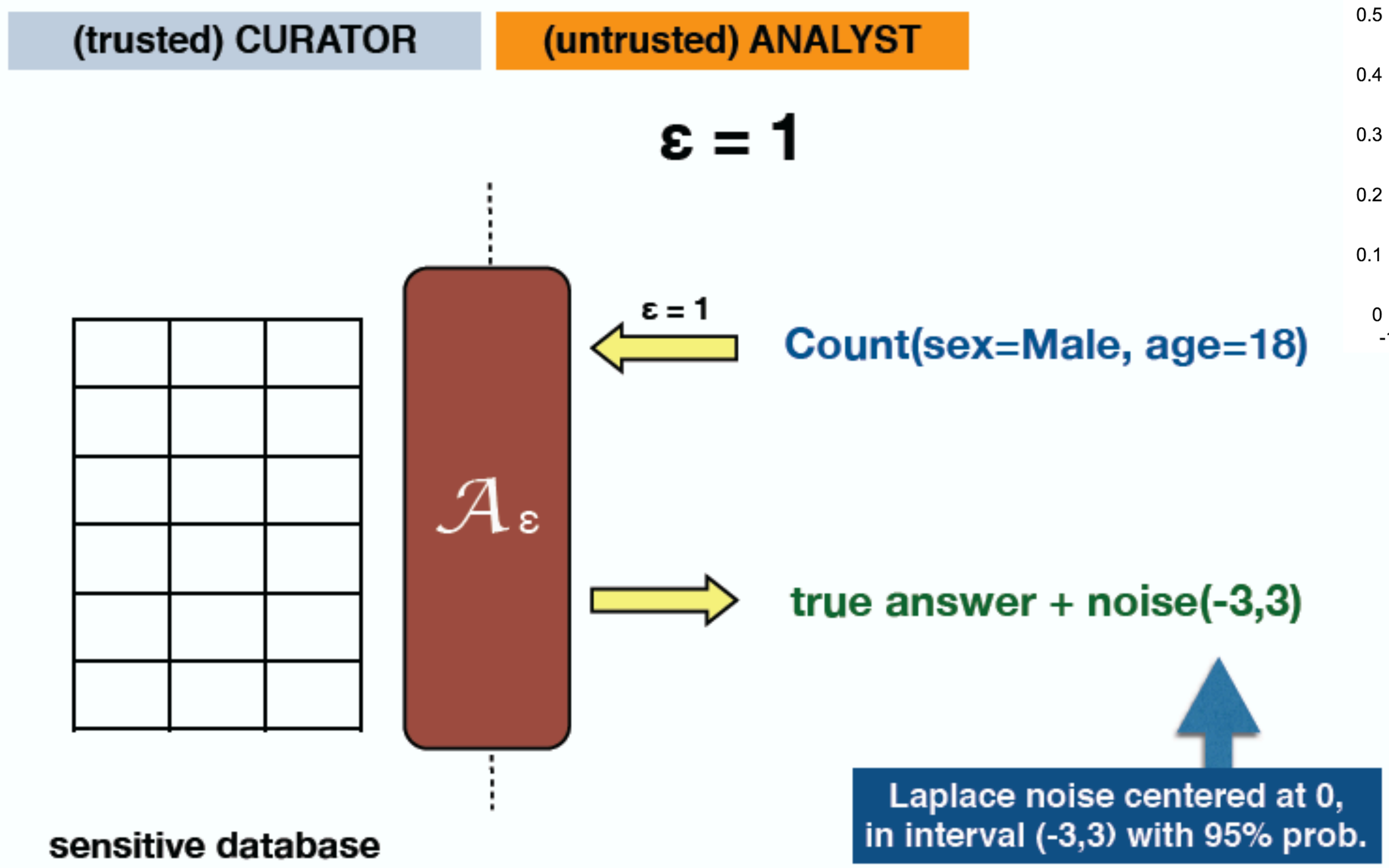
https://en.wikipedia.org/wiki/Laplace_distribution

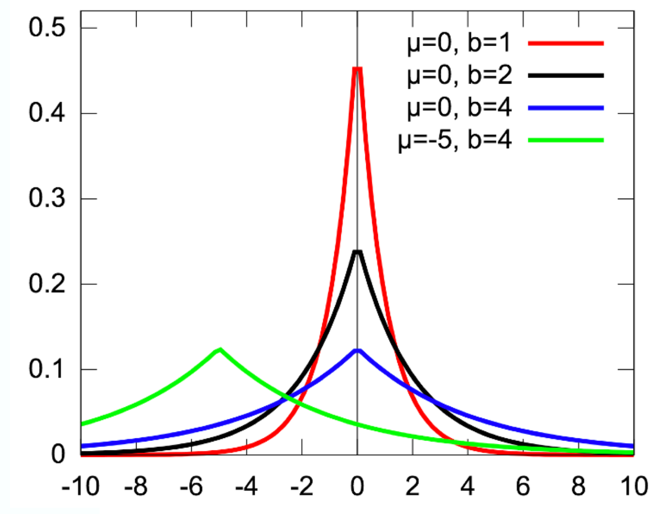
fix sensitivity ***Δq***, verify that more noise is added for lower ***ε***

lower ε = stronger privacy

r/ai

Adding noise





slide by Gerome Miklau

Query sensitivity

The ℓ_1 sensitivity of a query ***q***, denoted ***Δq***, is the maximum difference in the result of that query on a pair of neighboring databases

$$\Delta q = \max_{D,D'} \left| q(D) - q(D') \right|$$

query *q*	query sensitivity *Δq*
parallel composition	
select gender, count(*) from D group by gender	**1** (**disjoint groups**, presence or absence of one tuple impacts only one of the counts)
sequential composition	
an arbitrary list of *m* counting queries	***m*** (no assumptions about the queries, and so a single individual may change the answer of **every query** by 1)

r/ai

Sequential composition

- Consider 4 queries executed in sequence
  - Q1: select count(*) from D under $\boldsymbol{\varepsilon_1 = 0.5}$
  - Q2: select count(*) from D where sex = Male under $\boldsymbol{\varepsilon_2 = 0.2}$
  - Q3: select count(*) from D where sex = Female under $\boldsymbol{\varepsilon_3 = 0.25}$
  - Q4: select count(*) from D where age > 20 under $\boldsymbol{\varepsilon_4 = 0.25}$

- $\boldsymbol{\varepsilon = \varepsilon_1 + \varepsilon_2 + \varepsilon_3 + \varepsilon_4 = 1.2}$ That is: all queries together are ε-differentially private for ε =1.2. **Can we make a stronger guarantee?**
- This works because **Laplace noise is additive**

More generally: set a **cumulative privacy budget**, and split it between all queries, pre-processing, other data manipulation steps of the pipeline

r/ai

Parallel composition

- If the inputs are disjoint, then the result is ε-differentially private for $\varepsilon = \max(\boldsymbol{\varepsilon_1, \ldots, \varepsilon_k})$
  - Q1: select count(*) from D under $\boldsymbol{\varepsilon_1 = 0.5}$
  - Q2: select count(*) from D where sex = Male under $\boldsymbol{\varepsilon_2 = 0.2}$
  - Q3: select count(*) from D where sex = Female under $\boldsymbol{\varepsilon_3 = 0.25}$
  - Q4: select count(*) from D where age > 20 under $\boldsymbol{\varepsilon_4 = 0.25}$
- $\boldsymbol{\varepsilon = \varepsilon_1 + \max(\varepsilon_2, \varepsilon_3) + \varepsilon_4 = 1}$ That is: all queries together are ε-differentially private for ε =1.

r/ai

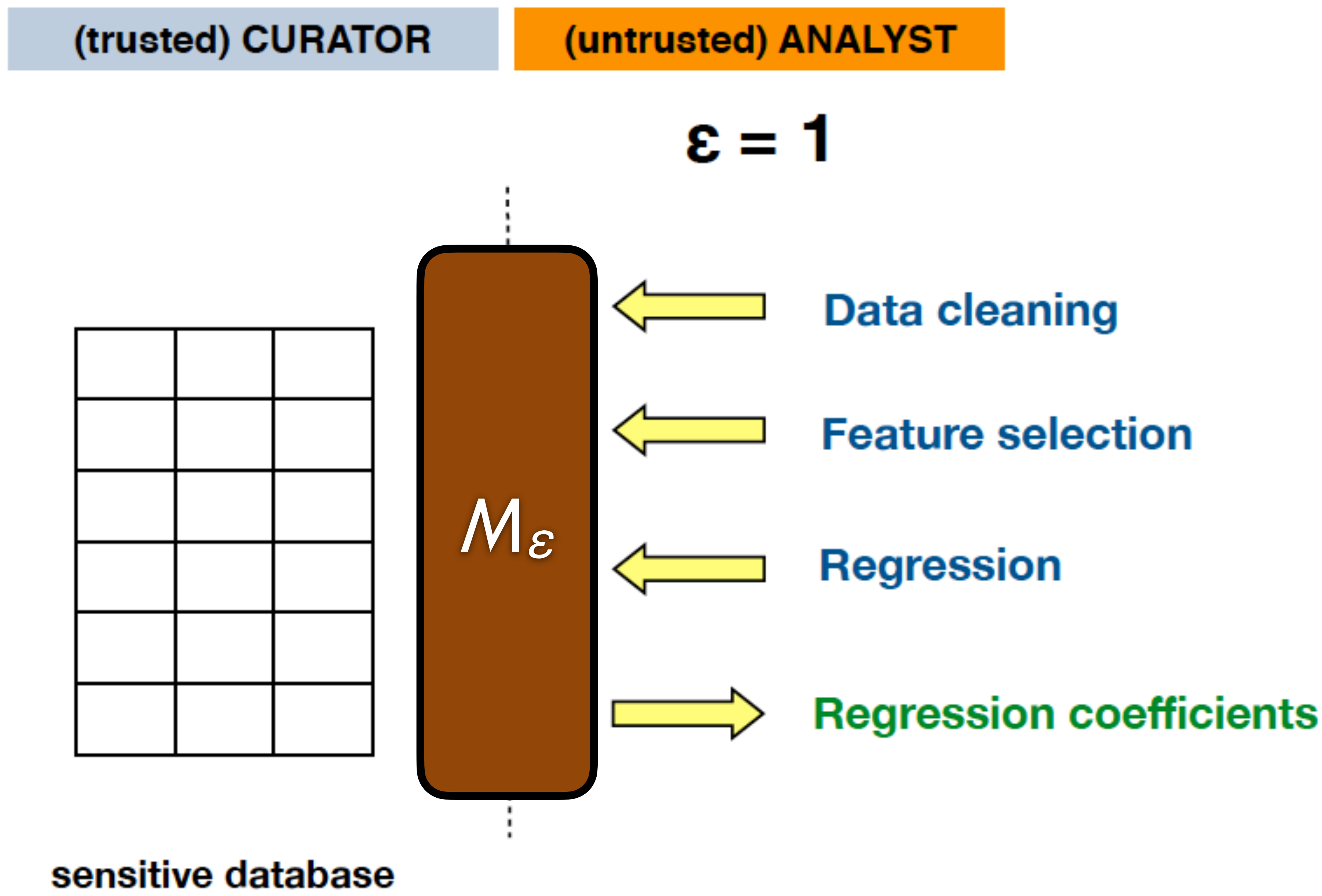
Composition and consistency

- Consider again 4 queries executed in sequence
  - Q1: select count(*) from D under $\varepsilon_1 = 0.5$ returns **2005**
  - Q2: select count(*) from D where sex = Male under $\varepsilon_2 = 0.2$ returns **1001**
  - Q3: select count(*) from D where sex = Female under $\varepsilon_3 = 0.25$ returns **995**
  - Q4: select count(*) from D where age > 20 under $\varepsilon_4 = 0.25$ returns **1789**

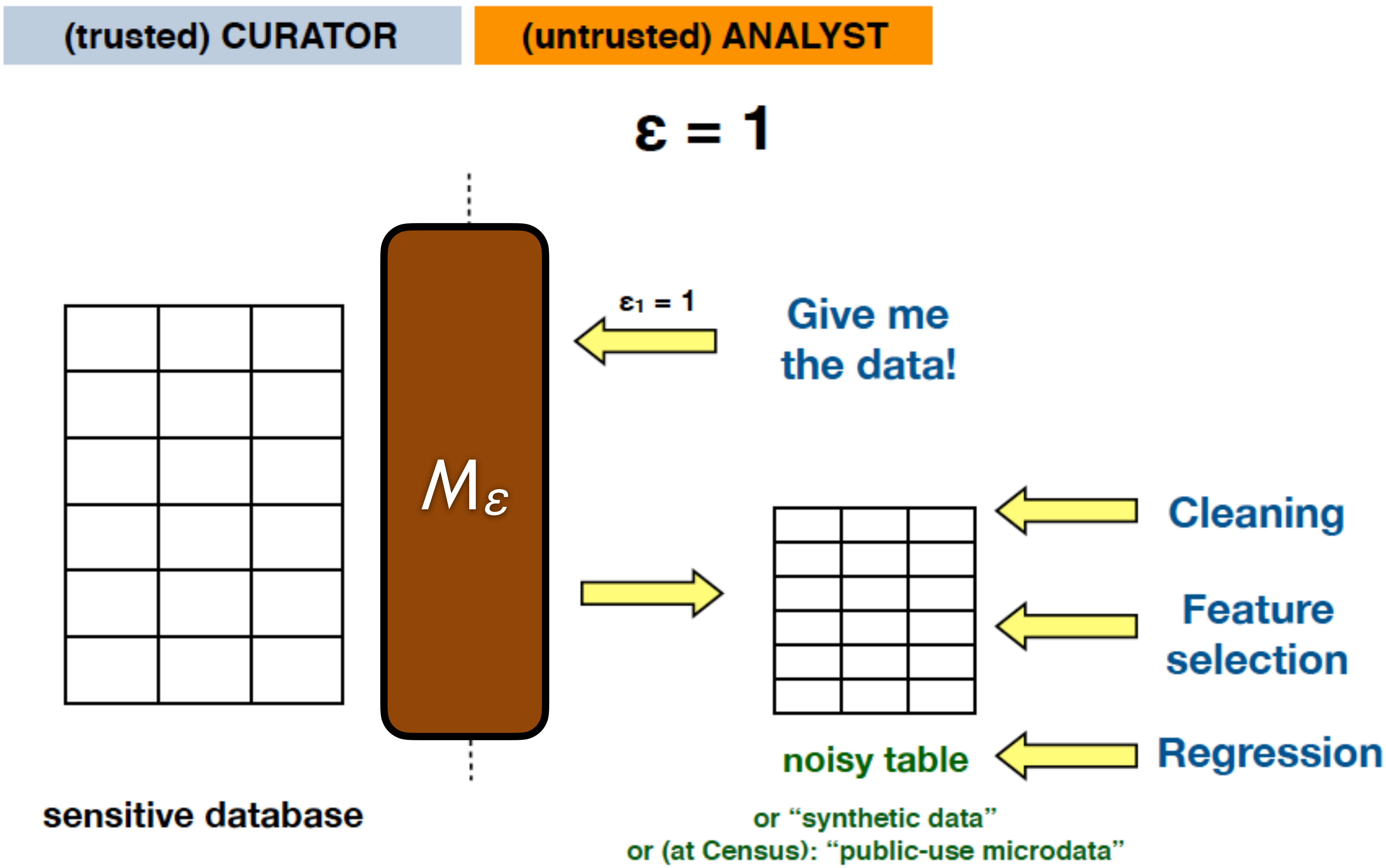
Assuming that there are 2 genders in D, Male and Female, there is **no database consistent with these statistics**!

Also don't want any negative counts + may want to impose datatype checks, e.g., no working adults with age = 5 etc.

Entire workflow must be DP



Privacy-preserving synthetic data



DP synthetic data generation

DP synthetic data

Lots of advantages

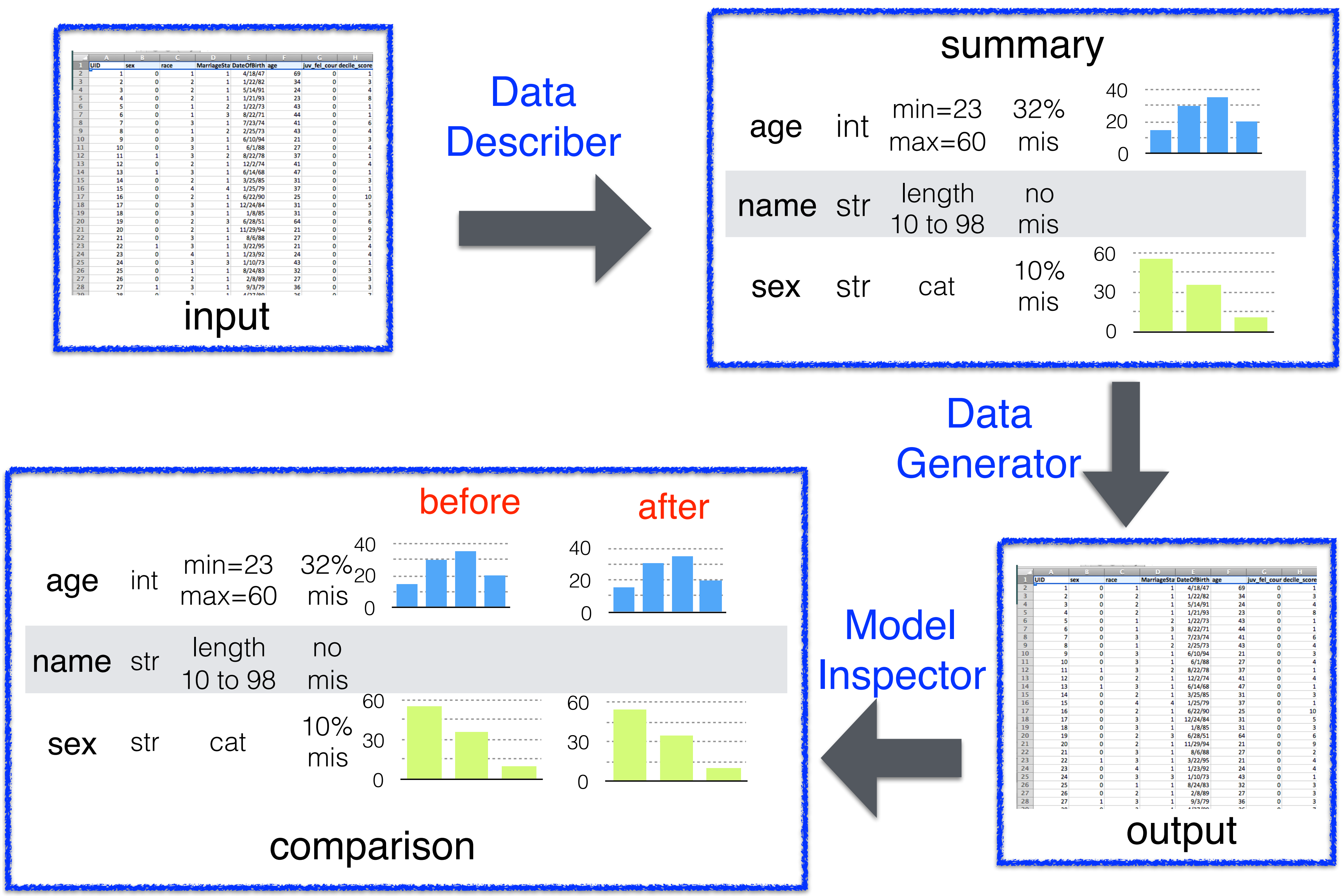
- Consistency is not an issue
- Analysts can treat synthetic data as a regular dataset, run existing tools
- No need to worry about the privacy budget
- Can answer as many queries as they want, and any kind of a query they want, including record-level queries

What's the catch?

Recall the Fundamental Law of Information Recovery. It tells us that we cannot answer all these queries accurately and still preserve privacy!

Therefore, when releasing synthetic data, we need to document it with which queries it supports well

Data Synthesizer



[Ping, Stoyanovich, Howe *2017*] http://demo.dataresponsibly.com/synthesizer/

Data Synthesizer

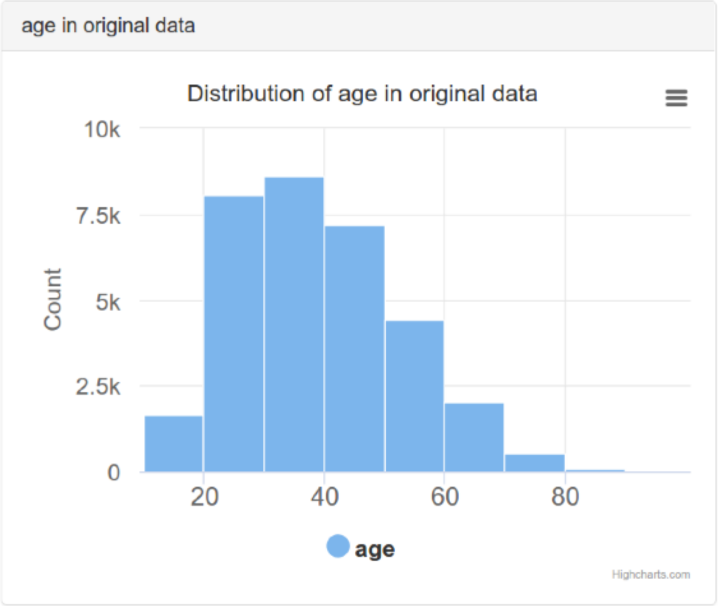
- Main goal: **usability first**
  - user is the data owner
  - the tool picks up data types from the input file: categorical / string / numerical (integer, float) / date-time
  - the tool computes the frequency of missing values per attribute
  - user can then inspect the result, over-ride what was learned about an attribute, e.g., whether it's categorical, or what its datatype is
- The tool generates an output dataset of a specified size, in one of three modes
  - **random** - type-consistent random output
  - **independent attribute** - learn a noisy histogram for each attribute
  - **correlated attribute** - learn a noisy Bayesian network (BN)

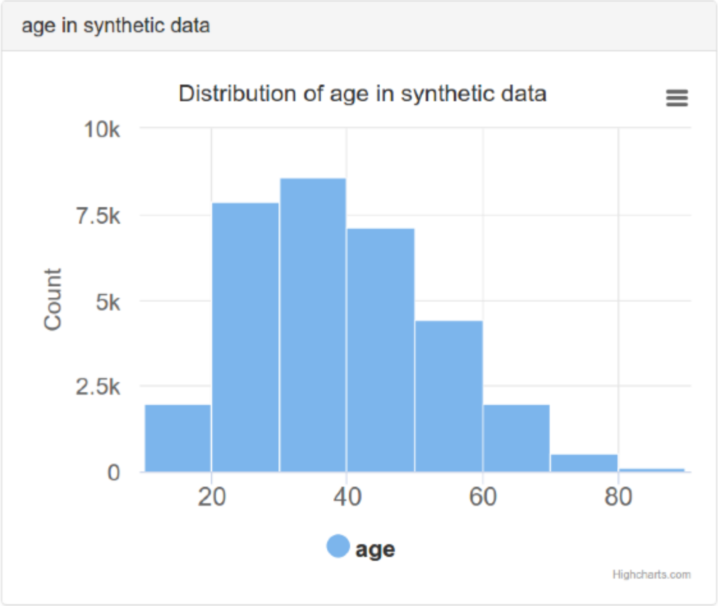
[Ping, Stoyanovich, Howe *2017*] http://demo.dataresponsibly.com/synthesizer/

Data Synthesizer: Independent attributes

Given the over-all privacy budget ε, and an input dataset of size n. Allocate ε/d of the budget to each attribute A_i in $\{A_1, .., A_d\}$. Then for each attribute:

- Compute the i^{th} histogram with t bins (t=20 by default), with query q_i
- The sensitivity Δq_i of this (or any other) histogram query is **2/n** **Why?**
- So, each bin's noisy probability is computed by adding $Lap\left(\frac{2d}{\varepsilon n}\right)$

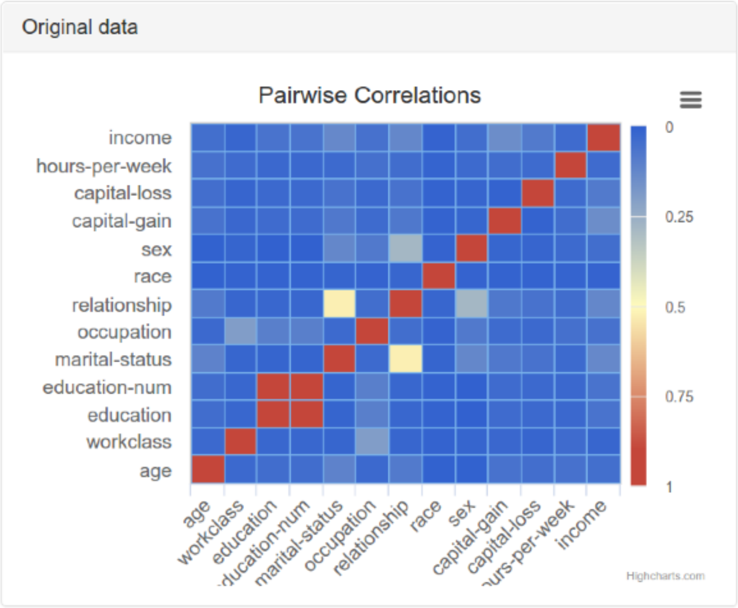


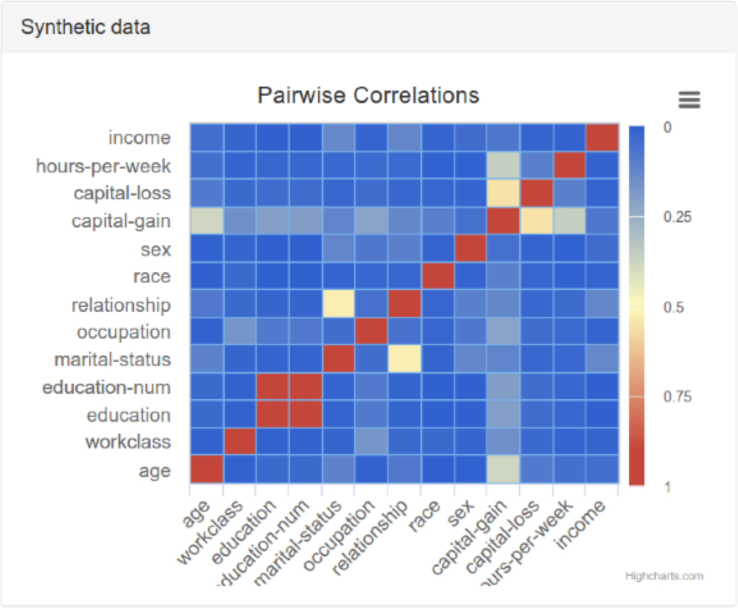


[Ping, Stoyanovich, Howe *2017*] http://demo.dataresponsibly.com/synthesizer/

Data Synthesizer: Correlated attributes

- Learn a differentially private Bayesian network (BN)
- Use the method called **PrivBayes** [Zhang, Cormode, Procopiuc, Srivastava, Xiao, *2016*]
- Privacy budget is split equally between (a) network structure computation and (b) populating the conditional probability tables of each BN node
- User inputs privacy budget **ε** and the maximum number of parents for a BN node ***k*** - you'll play with these settings as part of HW2
- The tool treats a missing attribute value as one of the values in the attribute's domain (not shown in the examples in the next two slides)

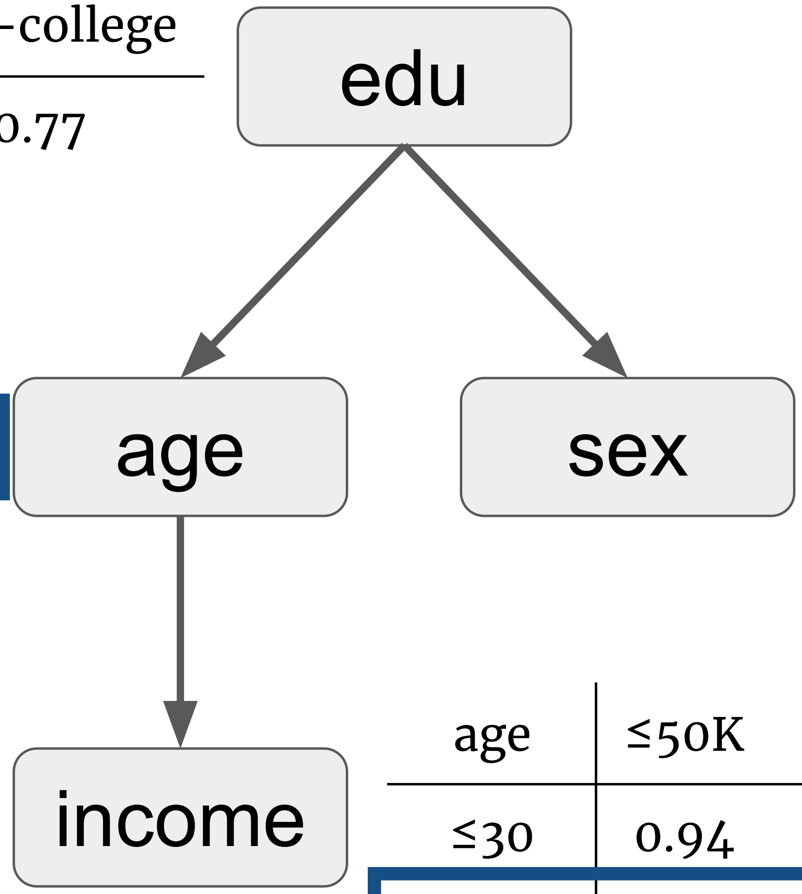




Data Synthesizer: Correlated attributes

K=1 not a causal DAG, a regular Bayesian network!

college	non-college
0.23	0.77



edu	≤30	30~50	>50
college	0.24	0.56	0.20
non-college	0.35	0.45	0.20

edu	female	male
college	0.30	0.70
non-college	0.34	0.66

age	≤50K	>50K
≤30	0.94	0.07
30~50	0.67	0.33
>50	0.68	0.32

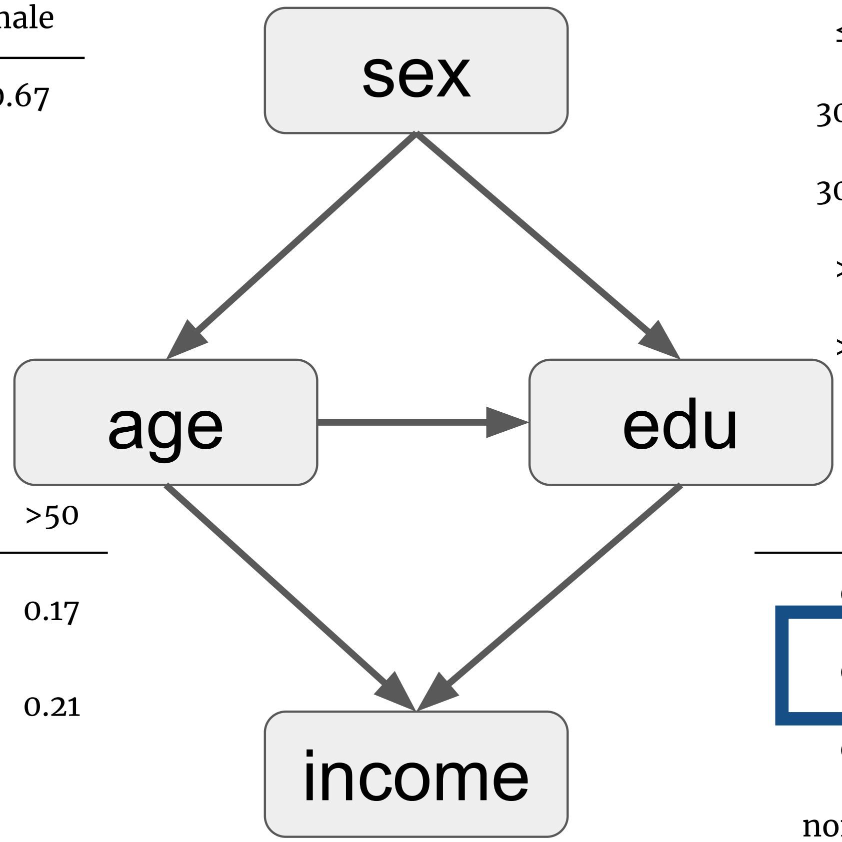
[Ping, Stoyanovich, Howe *2017*] http://demo.dataresponsibly.com/synthesizer/

Data Synthesizer: Correlated attributes

not a causal DAG, a regular Bayesian network!

K=2

female	male
0.33	0.67



age	sex	college	non-college
≤30	female	0.18	0.82
≤30	male	0.16	0.84
30~50	female	0.25	0.75
30~50	male	0.28	0.72
>50	female	0.17	0.83
>50	male	0.25	0.75

sex	≤30	30~50	>50
female	0.40	0.43	0.17
male	0.29	0.59	0.21

edu	age	≤50K	>50K
college	≤30	0.83	0.17
college	30~50	0.45	0.55
college	>50	0.41	0.59
non-college	≤30	0.96	0.04
non-college	30~50	0.76	0.24
non-college	>50	0.75	0.25

[Ping, Stoyanovich, Howe *2017*] http://demo.dataresponsibly.com/synthesizer/