Use queries of sums over random subsets to reconstruct individual data.
Importantly, the members of the subset are reported in each sum.
Received the Aircloak Bounty ($5000) for reidentifying challenge data in the Diffix commercial system.
Regression Based Reconstruction

\[ y_i = \beta_1 x_{1,i} + \beta_2 x_{2,i} + \ldots + \beta_N x_{N,i} + \epsilon_i \]

Here:

- \( N \) is the Number of people in the database
- \( i \) is query index
- \( y_i \) is \( i \)-th query release
- \( x_{h,i} \) is a \( \{0, 1\} \)-indicator of whether person \( h \) was included in query \( i \)
- \( \beta_h \) is \( h \)’s sensitive data
- \( \epsilon_i \) is the noise added to the \( i \)-th query
 Regression Based Reconstruction

\[ y_i = \beta_1 x_{1,i} + \beta_2 x_{2,i} + \ldots + \beta_N x_{N,i} + \epsilon_i \]

\[ 7 = 1 \cdot 1 + 0 \cdot 1 + 1 \cdot 0 + 0 \cdot 0 + \ldots + 0 \cdot 1 + 2 \]
\[ 4 = 1 \cdot 0 + 0 \cdot 1 + 1 \cdot 1 + 0 \cdot 1 + \ldots + 0 \cdot 1 + (−1) \]
\[ 6 = 1 \cdot 0 + 0 \cdot 0 + 1 \cdot 0 + 0 \cdot 1 + \ldots + 0 \cdot 0 + 1 \]

Here:

- \( N \) is the Number of people in the database
- \( i \) is query index
- \( y_i \) is \( i \)-th query release
- \( x_{h,i} \) is a \{0, 1\}-indicator of whether person \( h \) was included in query \( i \)
- \( \beta_h \) is \( h \)'s sensitive data
- \( \epsilon_i \) is the noise added to the \( i \)-th query
Regression Based Reconstruction

\[ y_i = \beta_1 x_{1,i} + \beta_2 x_{2,i} + \ldots + \beta_N x_{N,i} + \epsilon_i \]

\[ 7 = 1 \cdot 1 + 0 \cdot 1 + 1 \cdot 0 + 0 \cdot 0 + \ldots + 0 \cdot 1 + 2 \]
\[ 4 = 1 \cdot 0 + 0 \cdot 1 + 1 \cdot 1 + 0 \cdot 1 + \ldots + 0 \cdot 1 + (-1) \]
\[ 6 = 1 \cdot 0 + 0 \cdot 0 + 1 \cdot 0 + 0 \cdot 1 + \ldots + 0 \cdot 0 + 1 \]

Here:

- **N** is the Number of people in the database
- **i** is query index
- **y_i** is the **i**-th query release
- **x_{h,i}** is a \{0, 1\}-indicator of whether person **h** was included in query **i**
- **\beta_h** is **h**’s sensitive data
- **\epsilon_i** is the noise added to the **i**-th query
Regression Based Reconstruction

Find $\hat{\beta}_1, \ldots, \hat{\beta}_n$ s.t.:

$$\hat{\beta} = \text{argmin} \left[ \sum_i (y_i - \hat{y}_i)^2 \right]$$

where

$$\hat{y}_i = \hat{\beta}_1 x_{1,i} + \hat{\beta}_2 x_{2,i} + \ldots + \hat{\beta}_N x_{N,i}$$

In R see:

`lm()`

In Python see for example:

`linear_model.LinearRegression()`

from scikit-learn.
Example

From `regressionAttack.r`:

Reconstruction of Latino Variable

- fraction ones correct: 1
- fraction zeros correct: 1

Reconstruction of Latino Variable

- fraction ones correct: 0.53
- fraction zeros correct: 0.38
Demonstrates feasibility of Database Reconstruction Attacks (DRAs) on small Census blocks by using their released aggregate statistics.

- 1.5M census blocks with between 1 and 7 residents
- Each released statistic provides a constraint, and some blocks have only one possible dataset that satisfy all constraints.
### Example Dataset

**TABLE 1: FICTIONAL STATISTICAL DATA FOR A FICTIONAL BLOCK**

<table>
<thead>
<tr>
<th>STATISTIC</th>
<th>GROUP</th>
<th>AGE COUNT</th>
<th>MEDIAN</th>
<th>MEAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1A</td>
<td>total population</td>
<td>7</td>
<td>30</td>
<td>38</td>
</tr>
<tr>
<td>2A</td>
<td>female</td>
<td>4</td>
<td>30</td>
<td>33.5</td>
</tr>
<tr>
<td>2B</td>
<td>male</td>
<td>3</td>
<td>30</td>
<td>44</td>
</tr>
<tr>
<td>2C</td>
<td>black or African American</td>
<td>4</td>
<td>51</td>
<td>48.5</td>
</tr>
<tr>
<td>2D</td>
<td>white</td>
<td>3</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>3A</td>
<td>single adults</td>
<td>[D]</td>
<td>[D]</td>
<td>[D]</td>
</tr>
<tr>
<td>3B</td>
<td>married adults</td>
<td>4</td>
<td>51</td>
<td>54</td>
</tr>
<tr>
<td>4A</td>
<td>black or African American female</td>
<td>3</td>
<td>36</td>
<td>36.7</td>
</tr>
<tr>
<td>4B</td>
<td>black or African American male</td>
<td>[D]</td>
<td>[D]</td>
<td>[D]</td>
</tr>
<tr>
<td>4C</td>
<td>white male</td>
<td>[D]</td>
<td>[D]</td>
<td>[D]</td>
</tr>
<tr>
<td>4D</td>
<td>white female</td>
<td>[D]</td>
<td>[D]</td>
<td>[D]</td>
</tr>
<tr>
<td>5A</td>
<td>persons under 5 years</td>
<td>[D]</td>
<td>[D]</td>
<td>[D]</td>
</tr>
<tr>
<td>5B</td>
<td>persons under 18 years</td>
<td>[D]</td>
<td>[D]</td>
<td>[D]</td>
</tr>
<tr>
<td>5C</td>
<td>persons 64 years or over</td>
<td>[D]</td>
<td>[D]</td>
<td>[D]</td>
</tr>
</tbody>
</table>

*Note: Married persons must be 15 or over*
SAT Solvers state the feasibility of a solution to a series of logical formulae, and find a solution if one exists.

PicoSAT has R and Python bindings.

In R:
\texttt{install.packages("rpicosat")}

In Python:
\texttt{pip install pycosat}
Conjunctive Normal Form

Conjunctive Normal Form (CNF) is expressed as *conjunctions of disjunctions*, that is, clauses entirely composed of OR $\lor$, each of which are bound together by AND $\land$.

Negations of literals are allowed.

Examples:

- $(A \lor B) \land (\neg C \lor D)$
- $(A \lor B) \land (C \lor D \lor E \lor F)$
- $(A \lor B) \land (C)$
- $(A)$

Construction:

- $A \rightarrow B$ is expressed as $(\neg A \lor B)$
- $A \leftrightarrow B$ is expressed as $(\neg A \lor B) \land (A \lor \neg B)$
<table>
<thead>
<tr>
<th>Sex</th>
<th>Married</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>
### Dataset

<table>
<thead>
<tr>
<th>Actual</th>
<th>Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sex</td>
<td>Married</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>
### Dataset

<table>
<thead>
<tr>
<th>Actual</th>
<th>Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sex</td>
<td>Married</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>

Release:  
\[ \sum \text{Sex} = 2, \]
\[ \sum \text{Married} = 3, \]
\[ \sum \text{Sex} \cdot \text{Married} = 1. \]
Challenge

Transform into CNF:
\((A \land F) \lor (B \land G) \lor (C \land H) \lor (D \land I) \lor (E \land J)\)

What about:
\((A \land F) \oplus (B \land G) \oplus (C \land H) \oplus (D \land I) \oplus (E \land J)\)
Challenge

Transform into CNF:

\[(A \land F) \lor (B \land G) \lor (C \land H) \lor (D \land I) \lor (E \land J)\]

What about:

\[(A \land F) \oplus (B \land G) \oplus (C \land H) \oplus (D \land I) \oplus (E \land J)\]

(requires \(2^4\) clauses)