

Open Diffix

Project and technical overview

Paul Francis, MPI-SWS

June 2022



Paul Francis

- Director at the Max Planck Institute for Software Systems
 - German government-funded research
mpi-sws.org
- Co-founder Aircloak GmbH
 - Builds data anonymization solution
aircloak.com
- Founding member Open Diffix project
 - “Open source” of Aircloak tech
open-diffix.org



MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS



open-
diffix



Talk overview

- Open Diffix project
 - Goals
 - History
- Current products
- Core anonymization tech overview
 - Diffix Elm
 - Evaluation of anonymization
- Demo of **Diffix for Desktop** and **Diffix for PostgreSQL**
- See open-diffix.org
- francis@mpi-sws.org

open- diffix

- Open software project for **strong** (GDPR-compliant) data **anonymization**
- Founded Jan 2021
 - Core tech from MPI-SWS and Aircloak GmbH
 - Open Diffix is simpler, easier to use
- Late 2021:
 - First product release: **Diffix for Desktop**
 - Partner with the German Research Institute for Public Administration (FÖV) on legal and compliance
- Mid 2022
 - Release **Diffix for PostgreSQL**
- Five people: myself, 3 engineers, David Wagner (FÖV)



MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS

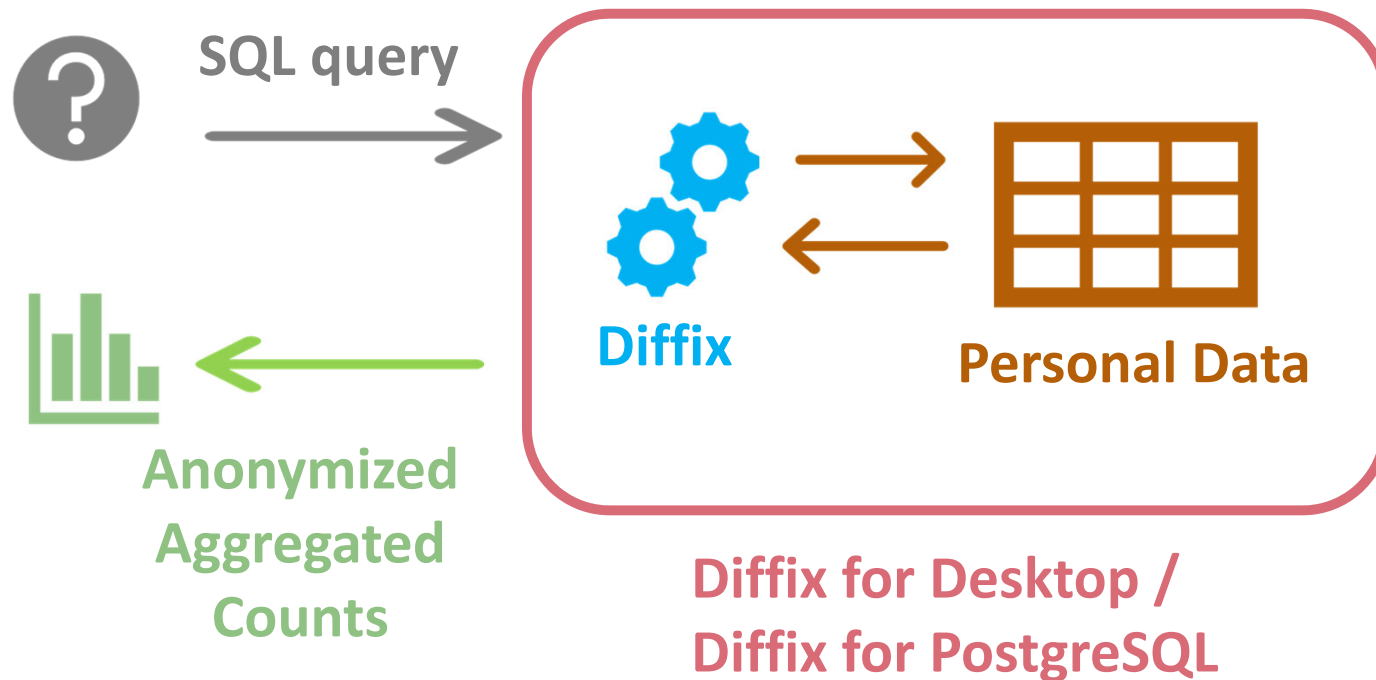


aircloak



Deutsches Forschungsinstitut
für öffentliche Verwaltung
German Research Institute
for Public Administration

Diffix is an anonymizing query engine





Diffix Elm anonymization

- Diffix Elm uses the **same kinds of mechanisms** as statistics offices
 - **Aggregation**
 - **Generalization**
 - **Noise**
 - **Suppression**
 - **Swapping**
- Diffix Elm effectively **automates** and **generalizes** what statistics office have successfully done for decades
- Adds additional mechanisms that defends against untrusted (malicious) analysts

Trusted and untrusted analyst modes



- Trusted analyst
 - Defends against *accidental* release of personal data
 - Easier to use because analyst can look at raw data
 - Analyst requires no knowledge of anonymization

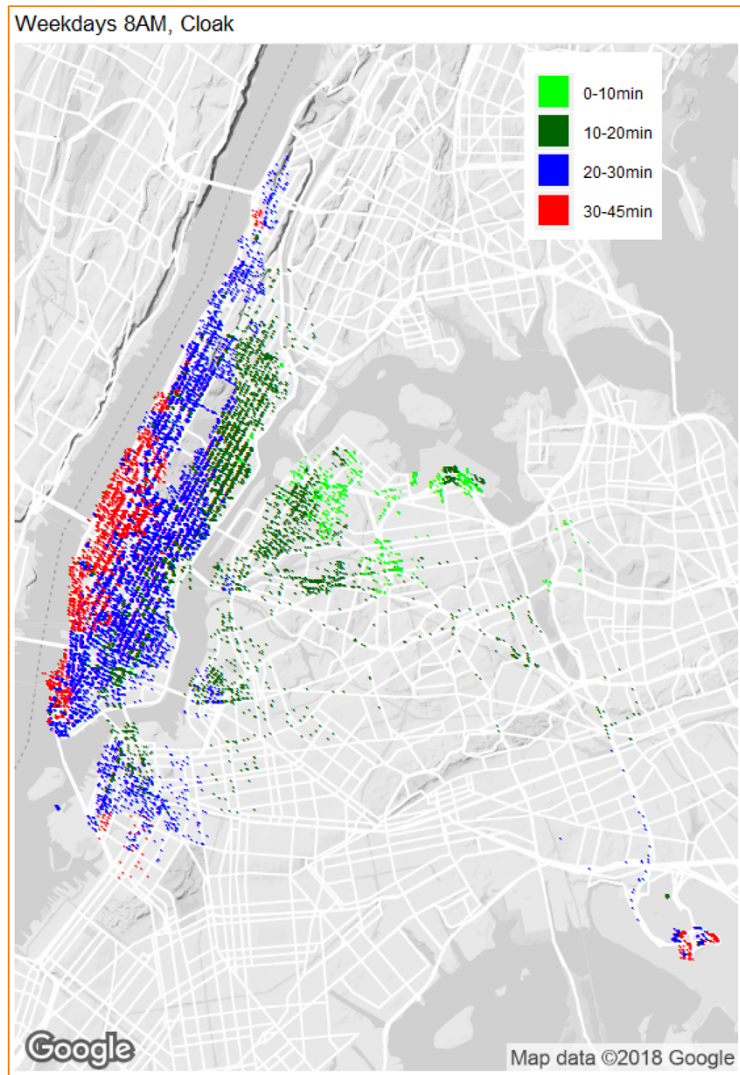


- Untrusted analyst
 - Defends against *intentional* release of personal data
 - Analyst can be malicious



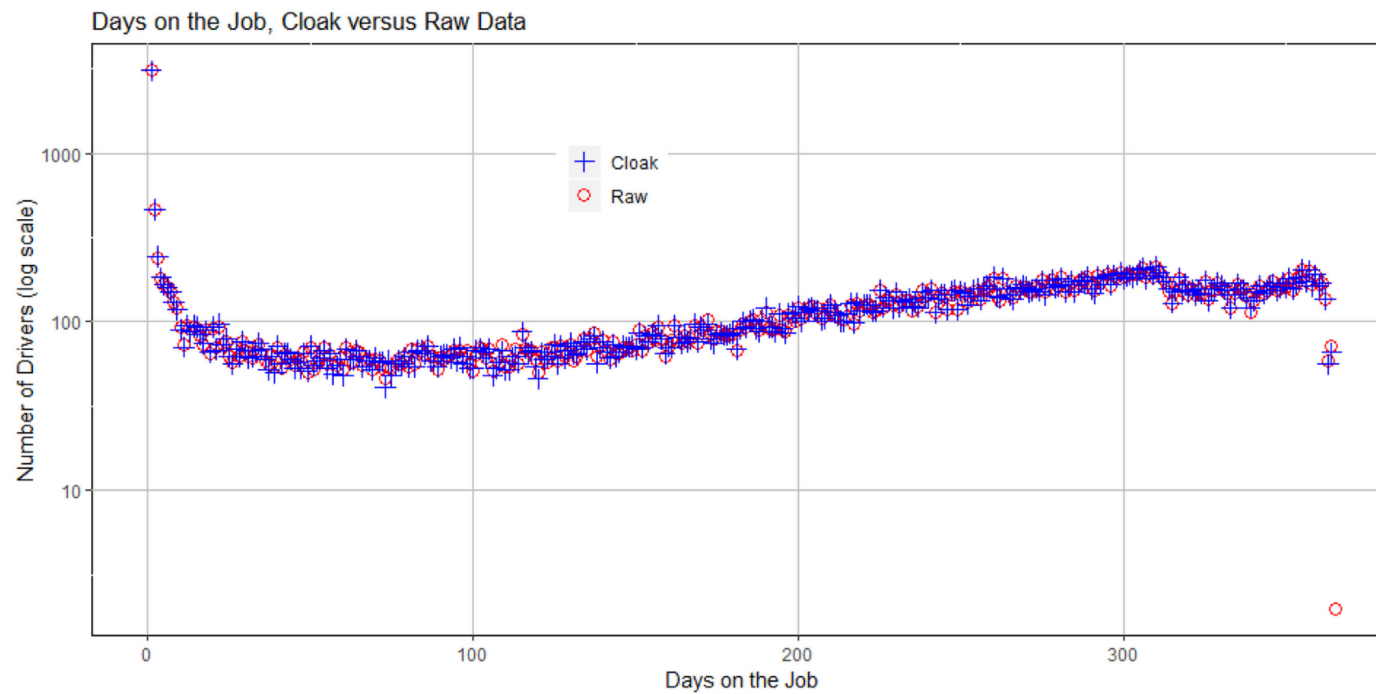
Licensing

- Business Source License (BSL1.1)
 - <https://github.com/diffix/desktop/blob/master/LICENSE.md>
- Free for any non-resale use, including commercial use
- Not free if resold substantially as is (anonymizing query interface)



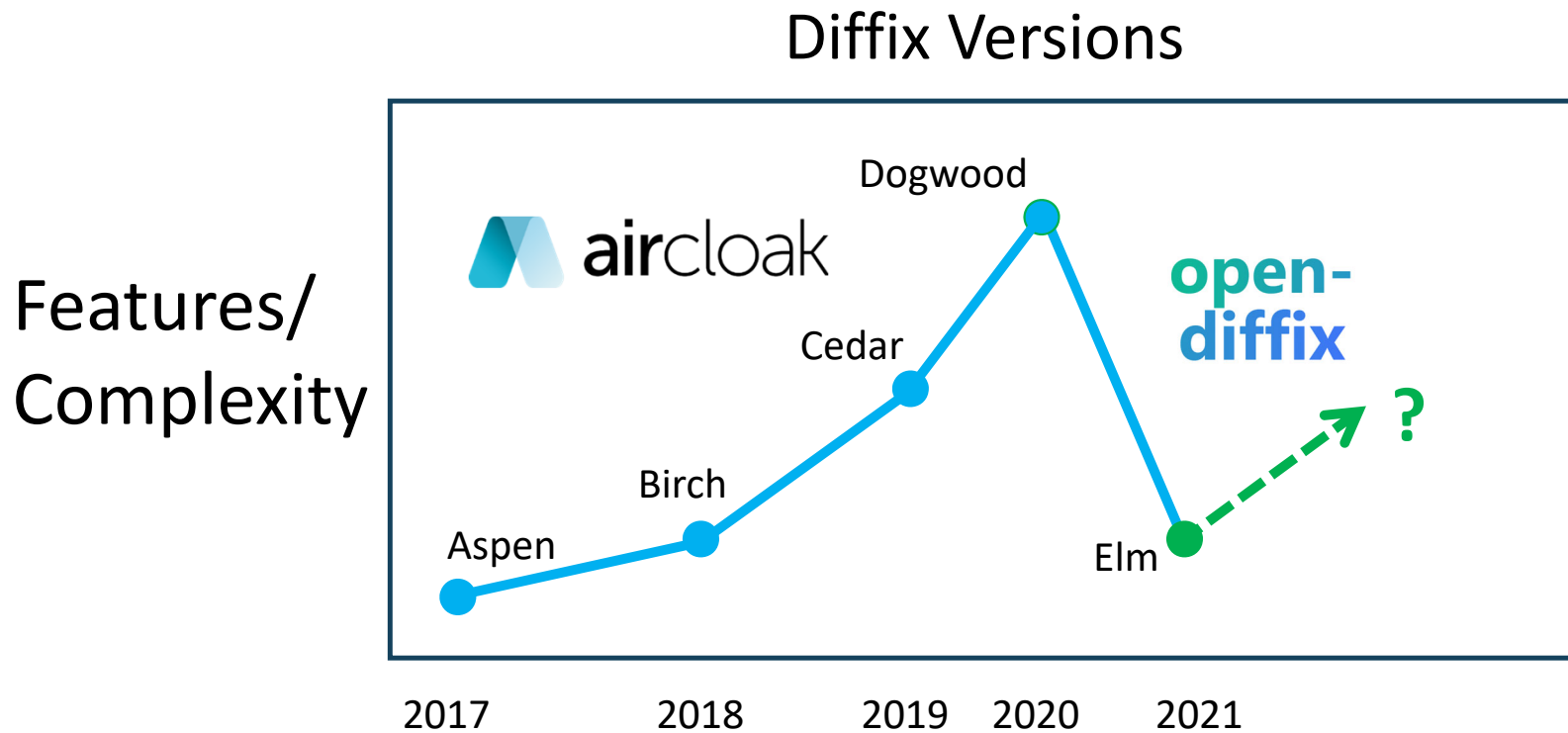
Taxi travel times to
La Guardia Airport

Comparison of Diffix and Raw Data





Currently looking for lighthouse projects



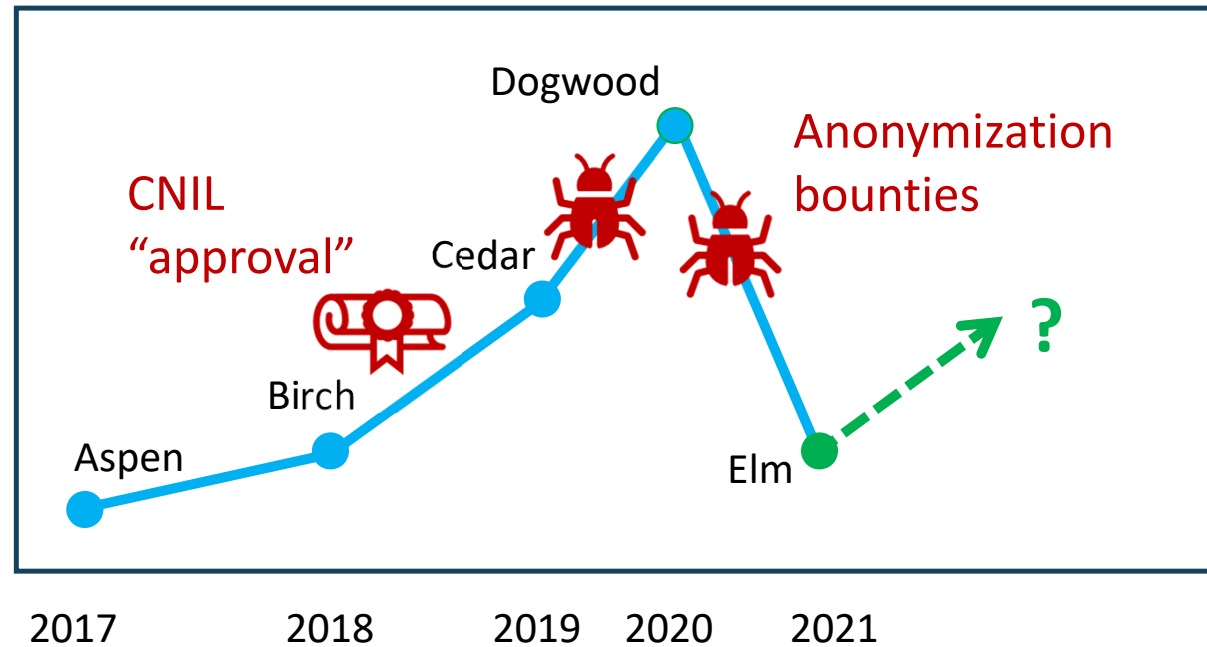
Difference between Aircloak and Open Diffix

Aircloak	Open Diffix
Proprietary (but transparent)	Source Available (BSL1.1)
SQL proxy in front of any DB	Integrated with DB (PostgreSQL or custom Desktop)
Untrusted (malicious) analyst	Two modes: trusted and untrusted analyst
Analyst can never view data	Trusted analyst can view data (ease-of-use)
Rich SQL (but complex/confusing)	Simplified (fewer features, easier to use)
High “start-up” effort	Low “start-up” effort (Diffix for Desktop)

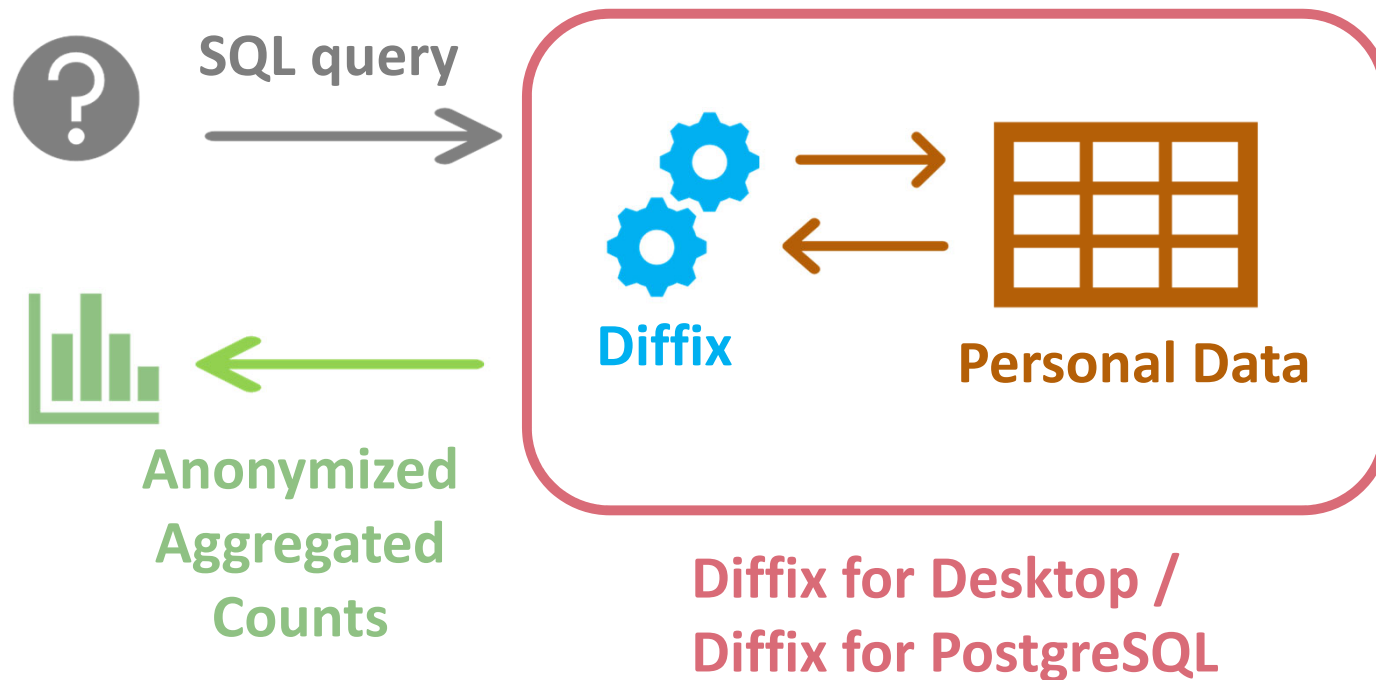


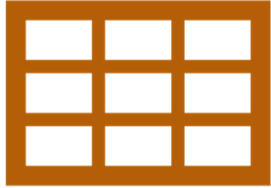
Building trust, minimizing risk

Features



Diffix is an anonymizing query engine





Personal Data

- Must be tabular
 - CSV (Diffix for Desktop)
 - PostgreSQL (Diffix for PostgreSQL)
- Data types: numeric or text
 - Other data types possible, for instance date and time
- Supports time-series data as well as non-time-series

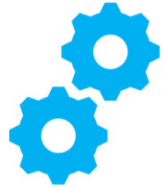
SQL query

- Supports simple queries
 - `SELECT column1, column2, count(*) FROM table`
- Supports counting rows and persons
 - `count(*)`, `count(column)`, `count(person_id)`
- Supports data aggregation as numeric ranges and substrings
- Future (3rd quarter 2022):
 - Other operations (sum, average, etc.)
 - Other aggregates
 - Other SQL functions (WHERE, JOIN, etc.)



Anonymized Aggregated Counts

- Counts have **noise** added
 - Roughly to within plus or minus 5 persons
- Bins are **suppressed** if too few persons in the bin
 - Avoid suppression by aggregating data



Software implementations

- .NET library (F#)
 - Thin command-line interface wrapped around library
 - Used as reference implementation (fast prototype)
 - Supports Diffix for Desktop
- Extension for PostgreSQL (C language)
 - All benefits of PostgreSQL (but with limited SQL syntax)
 - Diffix for PostgreSQL
- Current version is Diffix Elm



Personal
Data

relate	gender	age	marital_status	len_mar_stat	children_born
Head/Householder	Male	66	Widowed	99	0
Spouse	Female	46	Married, spouse present	21	11
Child	Female	3	Never married/single	99	0
Spouse	Female	22	Married, spouse present	5	1



SQL query

gender

☐

age

☒ Bin size:

marital_status

☒ Substring start: Substring length:

len_mar_stat

☐



Anonymized
Aggregated
Counts

age	marital_status	Count
20	Married, spouse abs...	180
20	Widowed	92
20	Divorced	12
25	Never married/single	3277
25	Married, spouse pre...	4992
25	Widowed	188
25	Married, spouse abs...	246
25	Divorced	34



Diffix for Desktop

- GUI-based application to run on desktop (Windows, Mac, Linux)
- Simple point-and-click operation (no SQL per se)
 - Import CSV
 - Select columns and bin sizes, examine data quality, repeat...
 - Export anonymized CSV
- Use-case is statistical data disclosure
 - Analyst with access to raw data wishes to release aggregate statistics
- GUI component compiled with core Diffix query engine



Diffix for PostgreSQL

- PostgreSQL extension
- Better scaling
- SQL API (limited SQL)



Diffix Fir and beyond

- Diffix Fir
 - JOIN
 - WHERE clauses (AND logic only)
 - New aggregates: sum(), average(), min(), max(), stddev()
- Beyond
 - More WHERE logic (OR, NOT)
 - Sub-queries
 - ???



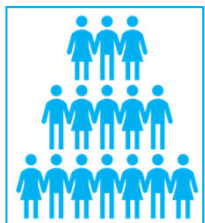
Diffix Elm anonymization

- A set of mechanisms
- Goal is not to “follow a model”, but to be strongly anonymous while giving adequate utility
- Note that all statistics offices take exactly this approach
 - UK Office for National Statistics
 - US Census Bureau
 -
- We measure anonymity using Precision Improvement for individual attacks



Diffix Elm anonymization

- Diffix Elm uses the **same kinds of mechanisms** as statistics offices
 - Aggregation
 - Generalization
 - Noise
 - Suppression
 - Swapping
- Diffix Elm effectively **automates** and **generalizes** what statistics office have successfully done for decades



Aggregation

- Output counts of things (bins), not microdata

person_id	age	sex
1	10	M
2	10	M
...
10	10	F
11	10	F
...
18	11	M
19	11	M
...
30	11	F
31	11	F
...

Original Data

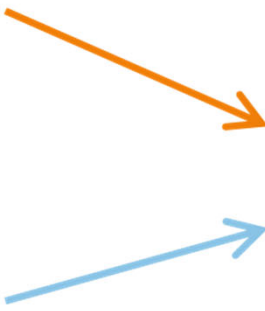
age	sex	count
10	M	9
10	F	8
11	M	12
11	F	11
...

Aggregated Data

Generalization

- 1957-12-14 → 1957
- SW1A 2AA → SW1
- Etc.
- Diffix Elm supports but doesn't enforce

person_id	date of birth
0	1989-10-24
1	1989-02-12
2	1989-08-14
...	...
18	1990-09-17
19	1990-11-01
20	1990-07-07
...	...



year of birth	count
1989	18
1990	26
...	...



Noise

- Distort counts
 - (Not column values)
- Sticky noise
 - “Same query same noise”
 - UK Office for National Statistics
 - Australian Bureau of Statistics
- Standard deviation = 1.5 for counting persons

year of birth	count		year of birth	count
1989	18	→	1989	19
1990	26	→	1990	23
...



Dynamic proportional noise

- Counting rows for time-series data
- Some persons contribute more than others
- Purpose of noise is to hide presence/absence of individual persons
- Per-bin noise is proportional to heavy contributors



Suppression

- Suppress bins pertaining to too few persons
 - Hides private column values
 - Tiny bins are easier to attack
- Suppression threshold itself is noisy
 - Increased uncertainty

Evaluating anonymity

We directly measure **intuitive** criteria:

- *What information can an analyst learn about a singled-out individual?*
 - **Precision:** What is the probability that the learned information is correct?
 - **Recall:** For what fraction of individuals can information be learned?
 - **Prior knowledge:** What does the analyst need to know in advance?
- Closely related to the three EU criteria

Evaluating anonymity

We directly measure **intuitive** criteria:

- *What information can an analyst learn about a singled-out individual?*
 - **Precision:** What is the probability that the learned information is correct?
 - **Recall:** For what fraction of individuals can information be learned?
 - **Prior knowledge:** What does the analyst need to know in advance?
- Closely related to the three EU criteria

Evaluating anonymity

- Design “all possible” attacks
 - Crowd-sourced: transparent publication, bounty programs
- Literally implement the attack, and measure each attack’s precision and recall
- Systems where *all known attacks* have low precision (improvement) **or** very low recall can be regarded as anonymous

Evaluating anonymity

- Design “all possible” attacks
 - Crowd-sourced: transparent publication, bounty programs
- Literally implement the attack, and measure each attack’s precision and recall
- Systems where *all known attacks* have low precision (improvement) **or** very low recall can be regarded as anonymous

Precision Improvement (not just precision)

- Any anonymized dataset has a certain expected precision
 - A prediction of gender=Male has an expected precision of 50%
 - A prediction of gender=Male given prostate cancer is near 100%
- We want to measure how much better we do than expected precision

Our measure of attack effectiveness (example using singling-out)

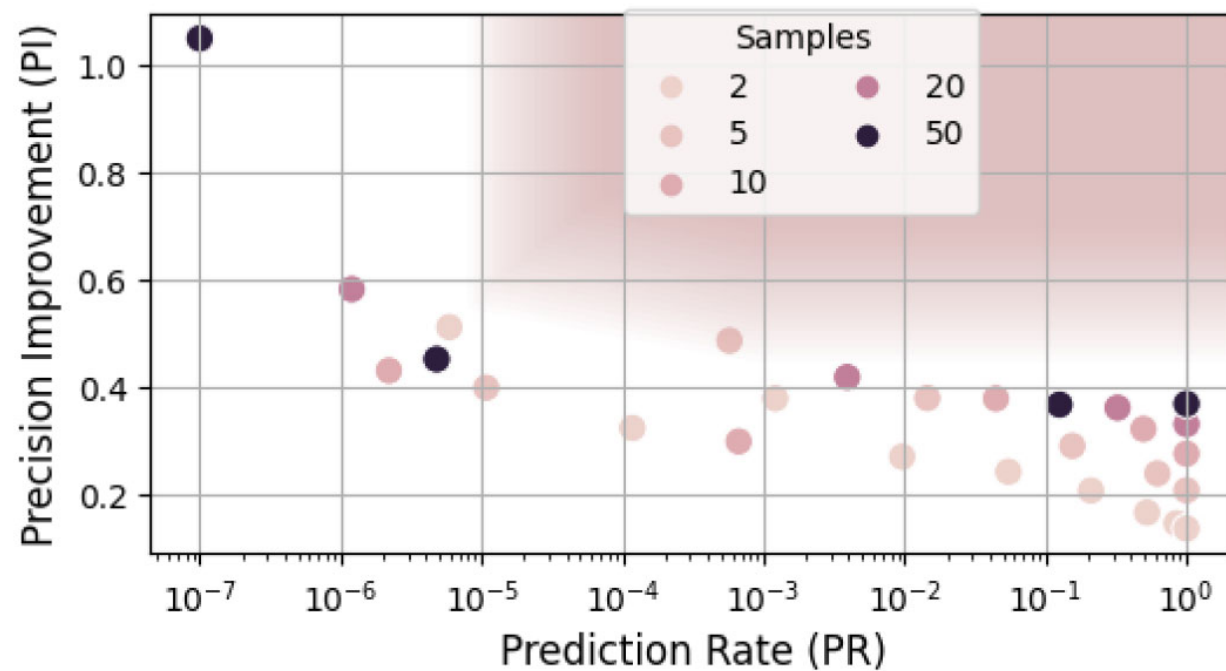
- Singling-out is one of three EU criteria for anonymity
- Singling-out occurs when a set of attributes can be associated with a single individual
 - “Exactly one individual is *male*, has bday *1957-12-14*, and zip code *67663*”
 - (Note not necessary to identify or “name” the individual)

PID	Zip	Birth	Gender	Lat	Lon	Time
...						
3X8YG82	67663	14.12.1957	M	40.1288	-73.4114	14:31:28
2N4XP31	92661	12.02.1994	M	41.2791	-72.6492	03:22:23
...						

Our measure of attack effectiveness (example using singling-out)

- Singling-out is one of three EU criteria for anonymity
- Singling-out occurs when a set of attributes can be associated with a single individual
 - “Exactly one individual is *male*, has bday *1957-12-14*, and zip code *67663*”
 - (Note not necessary to identify or “name” the individual)
- Run an attack making multiple singling-out ***claims***
- Each claim is either true or false
- ***The more true claims, the more effective the attack (precision improvement)***

Example of Precision Improvement measure



Evaluation (untrusted analyst mode)

- We (and others) have catalogued 37 different attacks
- 20 cannot be executed because of limited syntax of Diffix Elm
- 17 can be executed but are not effective
- Complete documentation on ArXiv
 - “Diffix Elm: Simple Diffix”
 - <https://arxiv.org/abs/2201.04351>

Summary of executable attacks
from
<https://arxiv.org/abs/2201.04351>

	Attack	PI / PR	PK Class	Conditions	Comments
5.3	Attribute value inspection	X			Must ensure that the issues described in Sections 6.2, 6.3, and 6.4 are addressed.
5.4	Unique Inference	VS		Com	May wish to inspect unique inference output bins with high AIDV counts that deviate from table-wide distribution (6.7).
5.5	Simple knowledge-based: Noise	W- VS	C	Com	
5.6	Simple knowledge-based: Suppression	W- VS	C	Com	May require XP or XXP level suppression
5.7	Averaging: naïve	X			
5.8	Averaging: different semantics, same result	X(T)		Com	Not an attack per se, but could partially reduce noise amount. Would not accidentally happen with trusted analyst.
5.9	LPR: randomness in column (UA-mode)	W- VS	B	Com	May want higher noise levels for untrusted analyst.
	(TA-mode)	X(T)	B	Com	Would not accidentally happen with trusted analyst.
5.10	LPR: aggregate combinations	VS			
5.11	Difference: positive AND, single victim	X	C	R	
5.12	Difference: positive AND, group of victims	VS	C	R	
5.13	Range creep with averaging (UA-mode)	X	A	Com	
	(TA-mode)	X(T)	A	Com	Would not accidentally happen with trusted analyst.
5.14	Salt: Dictionary attack on table	X(T)	X	Com	Morally equivalent to a password dictionary attack. Would not accidentally happen with trusted analyst.
5.15	Salt: Knowledge attack	X(T)	X		Requires knowledge of the secret salt. Would not accidentally happen with trusted analyst.
5.16	Access to multiple instances			X	Requires incorrect implementation of salt.
5.17	Incremental data update: difference	VS	A	Com	
5.18	Incremental data update: averaging	VS	A	VR	Depends on poor administration of data.
5.19	Detect outlier bucket	W	C	X	Only effective if learning one of a few distinct values. Data conditions can be detected and prevented in advance.

**open-
diffix**

open-diffix.org

francis@mpi-sws.org