As an initial exploration of these ideas, we have developed a tool named DataSynthesizer that generates structurally and statistically similar synthetic datasets based on real, private datasets. Given a sensitive dataset, DataSynthesizer infers the domain of each attribute and derives a probabilistic model of the attribute values, possibly adding noise to ensure differential privacy. The description of this derived model is saved in a *dataset description file*. The tool then generates synthetic datasets of arbitrary size by sampling from the stored distribution.

DataSynthesizer can operate in one of three modes, which differ in how the probabilistic model is derived: In *correlated attribute mode*, we learn a differentially private Bayesian network capturing the correlation structure between attributes, then draw samples from this model to construct the result dataset. In cases where the correlated attribute mode is too computationally expensive or when there is insufficient data to derive a reasonable model, one can use *independent attribute mode*. In this mode, a histogram is derived for each attribute, noise is added to the histogram to achieve differential privacy, and then samples are drawn for each attribute. Finally, for cases of extremely sensitive data, one can use *random* mode that simply generates type-consistent random values for each attribute. We give a brief overview of the implementation of the tool, and of the kinds of user interaction it supports, in Section 2.

We envision various extensions to this basic approach. For example, joining multiple sensitive datasets requires care: the join of two synthetic datasets does not necessarily have the same properties as a synthetic dataset derived from the join of two rel datasets. But in many cases, joins between the real data are expressly forbidden: linking education and housing datasets is important to understand the effects of homelessness on graduation rates, but FERPA laws preclude this kind of linking. Beyond linking, we see value in mixing real data and fake data in order to adjust statistical properties or ensure anonymity requirements, adversarially generating fake datasets to assess bias and accuracy of external models, and even generating complete"cities" of fake data based on the real data exhaust from city operations. In the latter case, we see the resulting interconnected datasets as a research instrument that can attract researchers who may otherwise be turned off by the administrative hurdles in getting access to data. We describe all of these extensions in more detail in Section 3.

We briefly survey related work in Section 4 and conclude in Section 5.

2 DATASYNTHESIZER: SAFE TABULAR DATA

We instantiate our vision in an open-source tool called DataSynthesizer, which takes a private dataset as input and generates synthetic datasets that can be shared safely. DataSynthesizer generates fake data using state-of-the-art differential privacy mechanisms [8]. Differential privacy is a family of techniques that guarantee that the output of an algorithm is statistically indistinguishable on a pair of *neighboring* databases — a pair of databases that differ by only one row. That is, the presence or absence of a single individual in the input to the algorithm will be undetectable when one looks at the output.

An important property of differential privacy is that its effectiveness degrades with repeated queries [11]. To prevent leaking private information through adversaries repeatedly sending data generation requests, the system administrator can assign a unique random seed for each person who requires a synthetic dataset.

2.1 Overview of the implementation

We now briefly describe the implementation of DataSynthesizer; see Ping et al. [20] for additional details.

DataSynthesizer is implemented in Python 3. The data owner can interact with the tool through Jupyter notebooks or through a Web-based UI. DataSynthesizer assumes that the private dataset is presented in CSV format. The tool is designed to work with minimal input from the user. For example, it is not necessary to specify data types of the attributes, the tool determines these automatically.

The input dataset is first processed by the DataDescriber module, which infers attribute data types and estimates their distributions. For each attribute identified as categorical (one with a small number of distinct values, such as gender, ethnicity and education level), DataDescriber computes the frequency distribution of each distinct value. For non-categorical numerical and datetime attributes, DataDescriber derives an equi-width histogram to represent the distribution. For non-categorical string attributes, their minimum and maximum lengths are recorded. This information is recorded in a dataset description, which is then used to generate fake datasets. Additional processing steps may be required depending on the specific mode of operation chosen by the data owner. There are three such modes, which we describe next.

Random mode. When invoked in random mode, DataGenerator generates type-consistent random values for each attribute. If an attribute is of type string, then a random string is generated with length that falls within the observed range of lengths in the dataset.

Independent attribute mode. When invoked in independent attribute mode, DataDescriber implements a differentially private mechanism by adding controlled noise into the learned per-attribute distributions (histograms). The noise is from a Laplace distribution with location 0 and scale $\frac{1}{n\epsilon}$, where n is the size of the input, denoted $Lap(\frac{1}{n\epsilon})$, setting $\epsilon = 0.1$ by default. When Laplace noise is added to histogram frequencies, the value may become negative. In that case the value is reset to 0 [24]. To generate a synthetic privacy-preserving dataset, the DataGenerator module is invoked and generates a synthetic dataset by sampling. Each row is sampled independently. The value of each attribute in each row is sampled independently from the corresponding noisy histogram using uniform sampling.

Correlated attribute mode. Attribute values are often correlated, e.g., *age* and *income* of a person. When invoked in correlated attribute mode, DataDescriber uses the GreedyBayes algorithm to construct Bayesian networks (BN) to model correlated attributes [24].

The Bayesian network gives the sampling order for generating attribute values, see Figures 5 and 6 for examples. The distribution from which a dependent attribute is sampled is called a conditioned distribution. When constructing a noisy conditioned distribution, $Lap(\frac{4(d-k)}{n\cdot\epsilon})$ is injected to preserve privacy. Here, d is the number of attributes, k is the maximum number of parents of a BN node,

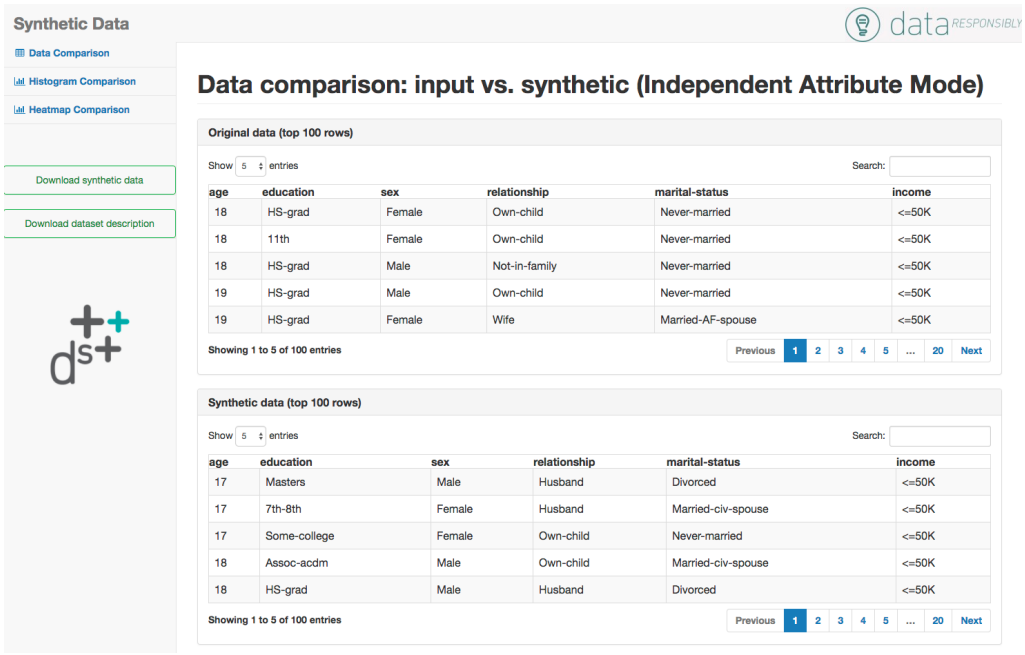

Figure 1: Data comparison

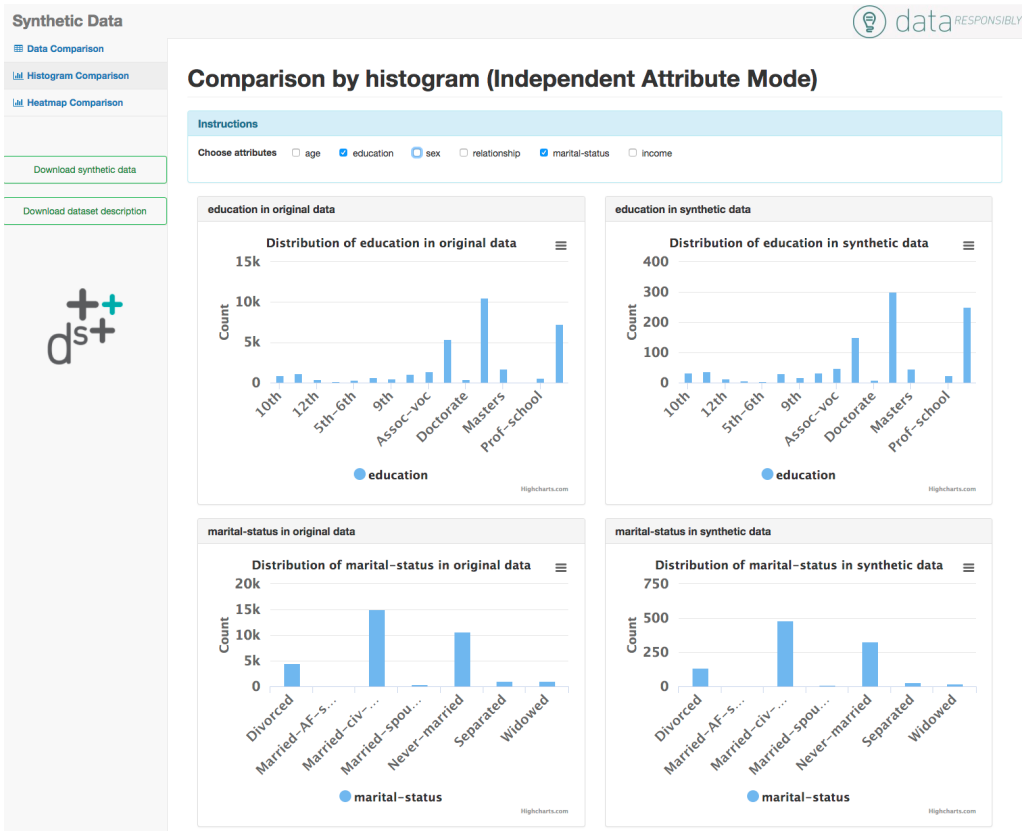

Figure 2: Histogram comparison

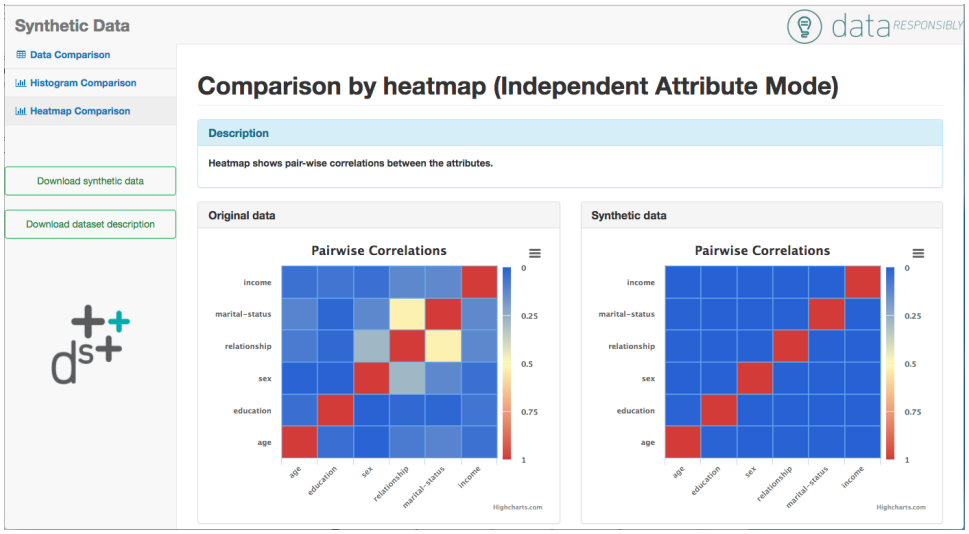

Figure 3: Pair-wise correlations: independent mode.

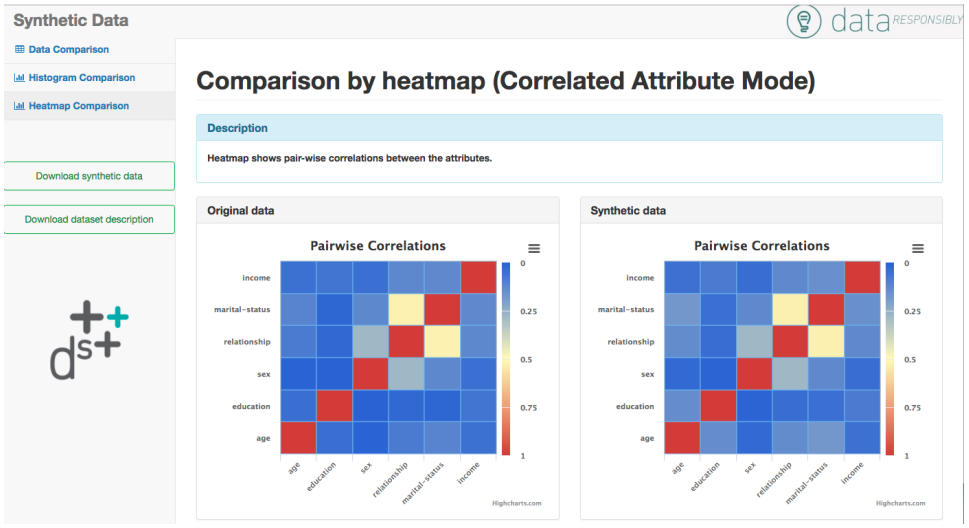

Figure 4: Pair-wise correlations: correlated attribute mode.

and n is the number of tuples in the input dataset. We construct conditional distributions according to Algorithm 1 of [24].

The parents of a dependent attribute can be categorical or numerical, whose distributions are modeled by bar charts and histograms, respectively. The conditions for this dependent attribute are the legal values of the categorical parents and the intervals of the numerical parents. Here, the intervals are formed in the same way as the unconditioned distributions of the parent attributes. For example, the *age* attribute has intervals {[10, 20),[20, 30),[30, 40)} in its unconditioned distribution. Assume *education* only depends on *age*. Its conditioned distributions will be under the same intervals, i.e., age $\in [10, 20)$, age $\in [20, 30)$ and age $\in [30, 40)$ respectively.

2.2 Interacting with DataSynthesizer

DataSynthesizer provides several built-in functions to inspect the similarity between the private input dataset and the output synthetic dataset.

With the *Data Comparison* view (Figure 1), the data owner can quickly test whether the tuples in the synthetic dataset are detectable by inspecting and comparing the raw data.

With the *Comparison by histogram* view (Figure 2), the user can compare the estimates of the per-attribute probability distributions in the input dataset to those in the synthetic dataset, with the expectation that these histograms will be similar in independent attribute and correlated attribute modes (as shown in Figure 2), but that they would be dis-similar in random attribute mode.

With the *Comparison by heatmap* view (Figures 3 and 4), the user can inspect pair-wise attribute correlations in the original dataset, and compare these to the correlations in the synthetic dataset. We quantify correlations using mutual information (MI), a common measure of mutual dependence between two random variables.

Consider, for example, the comparison by heatmap in independent attribute mode presented in Figure 3. Blue grid cells correspond to little or no correlation (MI close to 0), yellow cells correspond to moderate correlation (MI around 0.5), while red cells correspond to strong correlation (MI close to 1). Observe that marital status and relationship exhibit moderate correlation in the original dataset but that they do not correlate in the synthetic dataset (the corresponding cells are yellow on the left side of Figure 3 and blue on the right side of this figure). This effect is expected, since independent attribute mode removes any correlations between attributes.

Next, consider Figure 4, which presents heatmaps for correlated attribute mode. Observe that marital status and relationship exhibit a similar level of correlation in the synthetic dataset as they did in the original.

3 EXTENSIONS AND DISCUSSION

We see synthetic data as a fundamental component of "people analytics," where sensitive, private data must be used to make high-risk decisions. Beyond the capabilities of the current DataSynthesizer tool, we envision a number of usage scenarios and corresponding extensions; we describe these extensions in this section.

3.1 Enabling collaboration

As described in the introduction, our primary motivating use case is to reduce friction in the early stages of collaboration between data providers and outside data scientists. Our hypothesis is if data scientists are allowed to "get their hands dirty" with synthetic data, they are more likely to internalize the problem being solved and develop effective solutions more efficiently. In our own experience, we find that "whiteboard" solutions designed prior to seeing the data often become irrelevant once the data is available — attributes are different than we expected, data sizes are too small to train advanced models, biases in the data prevent certain kinds of analyses from taking place, inconsistent values complicate debugging (e.g., the string "N/A" in a column of integers). Exposure to these challenges early helps shape the conversation and reduce effort as data sharing agreements are being prepared.

3.2 Fake Linked Data

The value of the municipal datasets that motivate our approach enjoys a network effect: each pair of datasets enables new insights. For example, in the City of Seattle, a study is underway to determine the effect of housing instability on high school graduation rates: Do children who endure periods of homelessness graduate on time? Although the question is simple, it involves linking two extremely sensitive datasets: student data (protected by FERPA and requiring consent of parents to use) and homelessness data (protected by HIPAA and local privacy norms). In fact, the typical agreements governing the use of each of these datasets explicitly forbid linking them with any other datasets, and these typical agreements must be revised on a case by case basis to enable such studies.

To bootstrap collaborations over linked data, we might like to use the same approach we have described: generate fake education data, and then generate fake homelessness data. But this naïve will not work: synthetic records will not necessarily link with other synthetic records in a statistically similar way as in the real data. An apparent solution is to simply join the two datasets, then generate a synthetic dataset from the result. But this approach entails one data provider sharing their data with another provider, which is explicitly what we need to avoid.

To solve this problem, we need to estimate the statistical properties of the joined dataset, and use that information to guide the data synthesis process independently for the education and homelessness data. We observe that there are three classes of joined tuples: housing records H that have no corresponding education records, education records E that have no corresponding housing record, and linked pairs of housing and education records HE. We want to estimate the number in each category, $|H|$, $|E|$, $|HE|$.

To produce these estimates without sharing information, we can use locality sensitive hashing techniques [17] to independently map education tuples and housing tuples into a common space. Locality sensitive hashing algorithms have the property that similar inputs are mapped to similar outputs, without coordination. For example, integers can be mapped to their most significant n bits. For structured data, one simple approach is to concatenate the values in the tuples, split this long string into n-grams, sort the n-grams lexicographically, then truncate the sorted list to retain the first k n-grams. This way, similar tuples will map to similar sequences of n-grams.

Using this approach (or more sophisticated approaches that make use of domain knowledge), we can independently map tuples from different providers into a shared space to determine how likely they are to match, and therefore estimate the counts $|H|$, $|E|$, $|HE|$. Recall that $|H|$ is the number of housing tuples for which no nearby education tuples exist, and $|E|$ is the number of education tuples for which no nearby housing tuples exist. We can assume the remaining tuples join to produce linked pairs. Armed with these estimates, we can generate ids for education and housing tuples to ensure that an appropriate number of joined tuples are produced. Further, we can generate attribute values guided by the same LSH techniques to ensure that joined tuples share similar values.

3.3 Mixing Real and Synthetic Data

In many data sharing situations, data must be aggregated as an attempt at anonymization. Although aggregation approaches typically offer limited formal protection in practical cases [6, 7], they are often written into data sharing policies that must be obeyed.

For example, energy providers are strongly incentivized to deliver upgrades designed to improve efficiency. But assessing the efficacy of these upgrades using consumption data, normalized by weather and project specifications, is difficult. Once again, the problem is to share sensitive data: the energy consumption of customers is a signal-rich resource that could be used for "cybercasing," for example to predict when people leave their homes. To mitigate such risks, the rules that govern data sharing are designed to prevent disambiguation of aggregate measures. For instance, a geographic

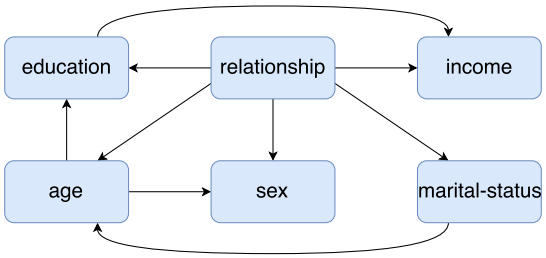

Figure 5: Bayesian network: Adult Income [15].

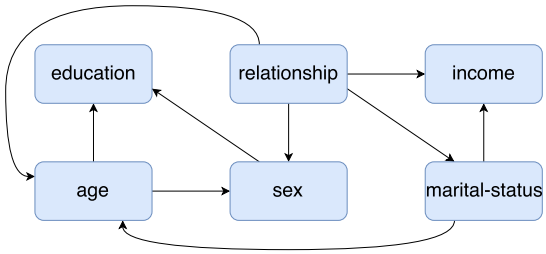

Figure 6: Bayesian network: synthetic.

estimate of energy usage must aggregate no fewer than 100 consumers, and no one consumer can represent more than 10% of the total usage. The final calculation must require a certain number of days of recorded usage data.

We see a novel use case for synthetic data to "fill out" the aggregates to meet anonymity requirements, as a kind of tuple-level imputation process.

Synthetic data may also be mixed with real data to repair global statistical anomalies, a kind of tuple-level imputation. For example, as reported in a recent New York Times article on urban homelessness [21]: "Last year, the total number of sheltered and unsheltered homeless people in the city was 75,323, which included 1,706 people between ages 18 and 24. The actual number of young people is significantly higher, according to the service providers, who said the census mostly captured young people who received social services." This representativeness gap can be filled with synthetic data to help data scientists triage their methods, or obscure the fact that the gap exists, in case data collection activities are sensitive.

3.4 Adversarial Fake Data Generation

Data providers are reluctant to share data for more than just privacy reasons. As decisions are shifted from humans to algorithms, the opportunity for, and impact of, discrimination becomes more acute. To earn the trust of data providers and demonstrate that proposed methods are robust to biased data, we envision generating intentionally biased datasets to explore "corner cases." Consider for example a hiring scenario. A cluster of job applicants who are similar in terms of experience, skills, and current position should tend to all receive offers to interview. If an African-American candidate in the cluster does not receive an offer when Caucasian candidates do, there is evidence that individual fairness is violated [9], and, specifically in this case, that there is *disparate treatment* — an illegal practice of treating an entity differently based on a protected characteristic such as race or gender. This situation is illustrated in Figure 7. Beyond disparate treatment, adversarial synthetic datasets can be generated to test more general cases of violation of individual and group fairness, under different interpretations of these measures [9, 25].

Testing fairness and bias properties of algorithmic decision-making systems is particularly important in cases where black-box third party tools are used, and where intervening on the inputs and analyzing the impact of the interventions on the outputs is one of only a handful of methods to infer system behavior. In enacting such interventions, it is particularly important to generate inputs

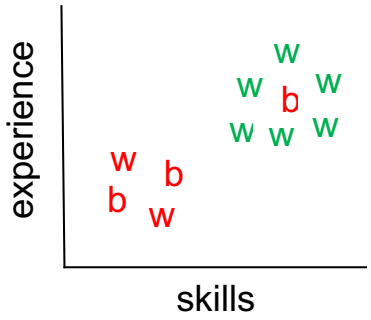

Figure 7: An illustration of a pathological dataset to evaluate disparate treatment. The symbol w represents a white candidate and the symbol b represents a black candidate. Red indicates that a hypothetical model rejected the candidate and green indicates that a hypothetical model accepted the candidate. At lower left, candidates are both low-skill and low-experience, and all are rejected. At upper right, a cluster of predominately white candidates and one black candidate receives inconsistent outcomes: if the one black candidate is rejected while similar candidates are accepted, there is evidence of disparate treatment. Generating specific situations to test a model's response is a role that synthetic data generation can play.

that are realistic, and that systematically explore cases that may not have been present in the actual historical data that was used to train the model.

Existing approaches for this problem rely on random sampling of input data to measure the response of black box models [5, 23], but random sampling cannot necessarily generate the pathological datasets that may occur in practice.

Beyond statistical bias, the ability to generate pathological datasets in terms of scale, anomalous values, and unexpected correlations can aid in debugging and stress-testing of external models, in the same way that benchmark datasets can help expose problems in, say, database systems [1].

To generate these pathological datasets, we can make a three extensions to the existing DataSynthesizer tool: First, we can allow users to edit the distribution derived from the real data to produce extreme values. Second, to allow even more precise control, we can design preconfigured pathological distributions to simulate, for

example, individual fairness situations. That is, using the annotations on the original data to distinguish protected attributes from non-protected attributes, we can generate clusters of similar tuples intentionally. Third, to assess systematic robustness (as opposed to statistical robustness) we can intentionally inject pathological values into attributes — missing values, inconsistent types, and extreme values (say, an age of -2 years).

3.5 Synthetic Cities: Comprehensive Interconnected Administrative Datasets

We envision combining all these techniques to generate, and incrementally improve, an administrative projection of entire virtual city to support research without data sharing encumbrances. Unlike population synthesis approaches in urban planning [10] and economics [2] which use agent-based models to study the emergent dynamics of an entire city from the ground up, our approach focuses on modeling *only the administrative data that would result from the dynamics of the city*, which provides a more realistic way of evaluating solutions. Since in practice researchers will typically only have access to the administrative data, models developed based on untestable assumptions about human behavior are difficult to evaluate, and interventions based on these models are difficult to trust. But the approaches are ultimately complementary, since artificial administrative data can be used to evaluate agent-based models, and agent-based models can be used as a source of artificial administrative data when the true datasets are not available.

4 RELATED WORK

In our work on DataSynthesizer we leverage recent advances in practical differential privacy [12] and privacy-preserving generation of synthetic datasets [16, 24]. In particular, we make use of the privacy-preserving learning of the structure and conditional probabilities of a Bayesian network in PrivBayes [24], and are inspired in our implementation by the work on DPBench [12].

Other recent approaches in privacy-preserving data generation include the work on plausible deniability in data synthesis [4], on perturbed Gibbs sampling for private data release [18, 19], and on sampling from differentially private copula functions [14].

Data sharing systems, including SQLShare [13] and DataHub [3], aim to facilitate collaborative data analysis, but do not incorporate privacy preserving features or purport to manage sensitive data. We see these systems efforts as a potential delivery vector for DataSynthesizer capabilities.

5 TAKE-AWAY MESSAGES

In this paper, we argued that the generation and use of synthetic data is a critical ingredient in facilitating collaborations involving sensitive data. The cost of establishing formal data sharing agreements limits the impact of these ad hoc collaborations in government, social sciences, health, or other areas where data is heavily encumbered by privacy rules.

A good fake dataset has two properties: it is representative of the original data, and it provides strong guarantees against privacy violations.

We discussed several use cases for fake data generation, and presented DataSynthesizer, a privacy-preserving synthetic data generator for tabular data. Given a dataset, DataSynthesizer can derive a structurally and statistically similar dataset, at a configurable level of statistical fidelity, while ensuring strong privacy guarantees. DataSynthesizer is designed with usability in mind. The system supports three intuitive modes of operation, and requires minimal input from the user.

We see fake data generators like DataSynthesizer used in a variety of application contexts, both as stand-alone libraries and as components of more comprehensive data sharing platforms. As part of ongoing work, we are studying how best to deliver these features to data owners, and determining how additional requirements can be met.

DataSynthesizer is open source, and is available for download at https://github.com/DataResponsibly. The work on DataSynthesizer is part of the Data, Responsibly project, and is a component of the Fides framework [22], which operationalizes responsibility in data sharing, integration, analysis and use.

REFERENCES

[1] Arvind Arasu, Raghav Kaushik, and Jian Li. 2011. Data Generation Using Declarative Constraints. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data (SIGMOD '11)*. ACM, New York, NY, USA, 685–696. DOI: http://dx.doi.org/10.1145/1989323.1989395

[2] Robert L. Axtell. 2016. 120 Million Agents Self-Organize into 6 Million Firms: A Model of the U.S. Private Sector. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems (AAMAS '16)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 806–816. http://dl.acm.org/citation.cfm?id=2937029.2937042

[3] Anant P. Bhardwaj and others. 2015. DataHub: Collaborative Data Science & Dataset Version Management at Scale. In *CIDR*.

[4] Vincent Bindschaedler, Reza Shokri, and Carl A. Gunter. 2017. Plausible Deniability for Privacy-Preserving Data Synthesis. *PVLDB* 10, 5 (2017), 481–492. http://www.vldb.org/pvldb/vol10/p481-bindschaedler.pdf

[5] Anupam Datta, Shayak Sen, and Yair Zick. 2016. Algorithmic Transparency via Quantitative Input Influence: Theory and Experiments with Learning Systems. In *IEEE SP*. 598–617. DOI: http://dx.doi.org/10.1109/SP.2016.42

[6] Yves-Alexandre de Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. 2013. Unique in the Crowd: The privacy bounds of human mobility. *Scientific Reports* 3 (2013).

[7] Irit Dinur and Kobbi Nissim. 2003. Revealing information while preserving privacy. In *Proceedings of the Twenty-Second ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 9-12, 2003, San Diego, CA, USA*. 202–210. DOI: http://dx.doi.org/10.1145/773153.773173

[8] Cynthia Dwork and others. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *TCC*.

[9] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S. Zemel. 2012. Fairness through awareness. In *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*. 214–226. DOI: http://dx.doi.org/10.1145/2090236.2090255

[10] Bilal Farooq, Michel Bierlaire, Ricardo Hurtubia, and Gunnar Flötteröd. 2013. Simulation based population synthesis. *Transportation Research Part B: Methodological* 58, C (2013), 243–263. http://EconPapers.repec.org/RePEc:eee:transb:v:58:y:2013:i:c:p:243-263

[11] Andreas Haeberlen and others. 2011. Differential Privacy Under Fire. In *USENIX Security*.

[12] Michael Hay and others. 2016. Principled Evaluation of Differentially Private Algorithms using DPBench. In *SIGMOD*.

[13] Shrainik Jain, Bill Howe, and Ed Lazowska. 2016. SQLShare: Results from a Multi-Year SQL-as-a-Service Experiment. In *SIGMOD*. ACM, New York, NY, USA.

[14] Haoran Li, Li Xiong, Lifan Zhang, and Xiaoqian Jiang. 2014. DPSynthesizer: Differentially Private Data Synthesizer for Privacy Preserving Data Sharing. *PVLDB* 7, 13 (2014), 1677–1680. http://www.vldb.org/pvldb/vol7/p1677-li.pdf

[15] M. Lichman. 2013. UCI Machine Learning Repository. (2013). http://archive.ics.uci.edu/ml

[16] Wentian Lu and others. 2014. Generating private synthetic databases for untrusted system evaluation. In *ICDE*.

[17] Dat Ba Nguyen. 2012. *Efficient Entity Disambiguation via Similarity Hashing*. Master's thesis. Universität des Saarlandes, Saarbruecken, Germany.

[18] Yubin Park and Joydeep Ghosh. 2014. PeGS: Perturbed Gibbs Samplers that Generate Privacy-Compliant Synthetic Data. *Trans. Data Privacy* 7, 3 (2014), 253–282. http://www.tdp.cat/issues11/abs.a154a13.php

[19] Yubin Park, Joydeep Ghosh, and Mallikarjun Shankar. 2013. Perturbed Gibbs Samplers for Generating Large-Scale Privacy-Safe Synthetic Health Data. In *IEEE International Conference on Healthcare Informatics, ICHI 2013, 9-11 September, 2013, Philadelphia, PA, USA*. 493–498. DOI: http://dx.doi.org/10.1109/ICHI.2013.76

[20] Haoyue Ping, Julia Stoyanovich, and Bill Howe. 2017. DataSynthesizer: Privacy-Preserving Synthetic Datasets. In *Proceedings of the 29th International Conference on Scientific and Statistical Database Management, Chicago, IL, USA, June 27-29, 2017*. 42:1–42:5. DOI: http://dx.doi.org/10.1145/3085504.3091117

[21] Nikita Stewart. 2016. Homeless Young People of New York, Overlooked and Underserved. *The New York Times* (Feb. 6 2016). https://www.nytimes.com/2016/02/06/nyregion/young-and-homeless-in-new-yorkoverlooked-and-underserved.html

[22] Julia Stoyanovich, Bill Howe, Serge Abiteboul, Gerome Miklau, Arnaud Sahuguet, and Gerhard Weikum. 2017. Fides: Towards a Platform for Responsible Data Science. In *Proceedings of the 29th International Conference on Scientific and Statistical Database Management, Chicago, IL, USA, June 27-29, 2017*. 26:1–26:6. DOI: http://dx.doi.org/10.1145/3085504.3085530

[23] Florian Tramèr, Vaggelis Atlidakis, Roxana Geambasu, Daniel J. Hsu, Jean-Pierre Hubaux, Mathias Humbert, Ari Juels, and Huang Lin. 2015. Discovering Unwarranted Associations in Data-Driven Applications with the FairTest Testing Toolkit. *CoRR* abs/1510.02377 (2015). http://arxiv.org/abs/1510.02377

[24] Jun Zhang and others. 2014. PrivBayes: private data release via Bayesian networks. In *SIGMOD*.

[25] Indre Zliobaite. 2015. A survey on measuring indirect discrimination in machine learning. *CoRR* abs/1511.00148 (2015). http://arxiv.org/abs/1511.00148