The Journal of Open Source Software

Metasynth: Simple Synthetic Tabular Data with Privacy Guarantees

Erik-Jan van Kesteren [1,2*¶], **Thom Volker** [1,3], **Ayoub Bagheri** [1], **Ron Scholten** [1], **Samuel Spithorst**[1], **and Raoul Schram** [1*]

1 Utrecht University, The Netherlands **2** ODISSEI: Open Data Infrastructure for Social Science and Economic Innovations, The Netherlands **3** Statistics Netherlands ¶ Corresponding author * These authors contributed equally.

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

We introduce `metasynth`, a python package that generates privacy-focused synthetic data. It allows owners of sensitive datasets to share data structure without privacy concerns. The software offers an auditable representation for synthesizing new data and supports initial exploration and analysis script development. `Metasynth` is extensible, allowing data owners to define privacy conditions through plug-ins. Core features include automatic distribution selection, intermediate representation creation, and data generation.

Statement of need

`Metasynth` is a python package for generating synthetic data with a focus on privacy and disclosure control. It is aimed at owners of sensitive datasets such as public organisations, research groups, or even individual researchers who want to improve the accessibility of their data for research and reproducibility by others. The goal of `metasynth` is to make it easy for data owners to share the structure and and approximation of the content of their data with others without any privacy concerns.

With this goal in mind, `metasynth` distinguishes itself from existing software for generating synthetic data (e.g., Nowok et al., 2016; Ping et al., 2017; Templ et al., 2017) by restricting itself to the "augmented plausible" category of synthetic data (Bates et al., 2019). This choice enables the software to generate synthetic data with **privacy or disclosure guarantees** through a plug-in system. Moreover, our system provides an **auditable and editable intermediate representation** in the form of a human- and machine-readable `.json` metadata file from which new data can be synthesized.

Through our focus on privacy and transparency, `metasynth` explicitly avoids generating synthetic data with high analytical validity. The data generated by our system is realistic in terms of data structure and plausible in terms of values for each variable, but any multivariate relation or conditional patterns are excluded. This has implications for how this synthetic data can be used: not for statistical analysis and inference but for initial exploration, analysis script development, and communication outside the data owner's institution. In the intended use case, an external researcher can make use of the synthetic data to assess the feasibility of their intended research before making the (often time-consuming) step of requesting access to the sensitive source data.

As mentioned before, the privacy capacities of `metasynth` are extensible through a plug-in system, recognizing that different data owners have different needs and definitions of privacy. A data owner can define under which conditions they would accept open distribution of their

The Journal of Open Source Software

synthetic data — be it based on differential privacy (Dwork, 2006), statistical disclosure control (Wolf, 2012), k-anonymity (Sweeney, 2002), or another specific definition of privacy. As part of the initial release of `metasynth`, we publish two proof-of-concept plugins: one following the disclosure control guidelines from Eurostat (Bond et al., 2015), and one based on the sample-and-aggregate technique for differential privacy (Dwork & Smith, 2010, p. 142).

Software description

At its core, `Metasynth` is designed to do three things:

1. Automatically select univariate distributions and fit them to a well-defined tabular dataset, possibly with privacy guarantees.
2. (optionally) Create an intermediate representation of the fitted model for auditing, editing, and exporting.
3. Generate new synthetic datasets based on the fitted model or its serialized representation.

In this section, we briefly describe each of these processes.

Distribution selection and fitting. For each data type supported by `metasynth`, there is a set of candidate distributions that can be fitted to that data type (see Table 1).

Table 1: Candidate distributions associated with data types in the core `metasynth` package.

variable type	data type	example	candidate distributions
continuous	float	1.0, 2.1, …	UniformDistribution, NormalDistribution
discrete	int	1, 2, …	DiscreteUniformDistribution
categorical	pl.Categorical	gender, country	MultinoulliDistribution
structured string	str	Room number A108, C122	RegexDistribution
unstructured string	str	Names, open answers	FakerDistribution, LLMDistribution
temporal	Date, Datetime	2021-01-13, 01:40:12	DateUniformDistribution

A basic example

```python
import polars as pl
from metasynth import MetaDataset, demo_file

# we use polars as the
df = pl.DataFrame(
    {
        "ID": [1, 2, 3, 4, 5],
        "fruits": ["banana", "banana", "apple", "apple", "banana"],
        "B": [5, 4, 3, 2, 1],
        "cars": ["beetle", "audi", "beetle", "beetle", "beetle"],
        "optional": [28, 300, None, 2, -30],
    }
)
```

```python
# convert appropriate columns to categorical
df = df.with_columns([
    pl.col("fruits").cast(pl.Categorical),
    pl.col("cars").cast(pl.Categorical),
])

# set A to unique and B to not unique
spec_dict = {
    "ID": {"unique": True},
    "B": {"unique": False}
}

# create metadataset
mds = MetaDataset.from_dataframe(df)

# write to json
mds.to_json("examples/basic_example.json")

# then, export json from secure environment

# outside secure environment, load json
mds_out = MetaDataset.from_json("examples/basic_example.json")

# create a fake dataset
mds_out.synthesize(10)
```


Acknowledgements

This research was conducted in whole or in part using ODISSEI, the Open Data Infrastructure for Social Science and Economic Innovations (https://ror.org/03m8v6t10)

The {metasynth} project is supported by the FAIR Research IT Innovation Fund of Utrecht University (as of March 2023)

References

Bates, A., Spakulová, I., Dove, I., & Mealor, A. (2019). *ONS methodology working paper series number 16—synthetic data pilot*.

Bond, S., Brandt, M., & Wolf, P. de. (2015). *Guidelines for output checking. eurostat*.

Dwork, C. (2006). Differential privacy. *International Colloquium on Automata, Languages, and Programming*, 1–12.

Dwork, C., & Smith, A. (2010). Differential privacy for statistics: What we know and what we want to learn. *Journal of Privacy and Confidentiality*, *1*(2).

Nowok, B., Raab, G. M., & Dibben, C. (2016). Synthpop: Bespoke creation of synthetic data in r. *Journal of Statistical Software*, *74*, 1–26.

Ping, H., Stoyanovich, J., & Howe, B. (2017). Datasynthesizer: Privacy-preserving synthetic datasets. *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*, 1–5.

Sweeney, L. (2002). K-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *10*(05), 557–570.

Templ, M., Meindl, B., Kowarik, A., & Dupriez, O. (2017). Simulation of synthetic complex data: The r package simPop. *Journal of Statistical Software*, *79*(10), 1–38.

Wolf, P.-P. de. (2012). *Statistical disclosure control*. Wiley & Sons, Chichester.