





# Metasynth: Transparent Generation of Synthetic Tabular Data with Privacy Guarantees

Erik-Jan van Kesteren<sup>1,2,\*</sup>, Thom Volker<sup>1,3</sup>, Ayoub Bagheri<sup>1</sup>, Ron Scholten<sup>1</sup>, Samuel Spithorst<sup>1</sup>, and Raoul Schram<sup>1\*</sup>

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

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

## Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Open Journals](#) 

## Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

## Summary

We introduce metasynth, a Python package designed to transparently generate privacy-preserving synthetic data. This tool aids data owners in sharing their data's structure and approximated content without compromising privacy. Metasynth focuses on the 'augmented plausible' category of synthetic data, balancing transparency with privacy through its plug-in system. While the analytical validity of the generated data is intentionally limited, its potential uses include exploratory analyses, script development, and external communication. The software is flexible, scalable, and easily extended to meet diverse privacy needs.



Figure 1: Logo of the metasynth project.

## 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, and 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 and 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 relations or conditional patterns are excluded. This has implications for how this synthetic data can be used: not for statistical analysis and inference, but rather 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 for the final analysis.

As mentioned before, the privacy capacities of metasynt 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 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 metasynt, 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).

## 47 Software features

48 At its core, Metasynth is designed for three functions, which are briefly described in this section:

- 49 1. **Estimation:** Automatically select univariate distributions and fit them to a well-defined  
50 tabular dataset, possibly with privacy guarantees.
- 51 2. **(De)serialization:** Create an intermediate representation of the fitted model for auditing,  
52 editing, and exporting.
- 53 3. **Generation:** Generate new synthetic datasets based on the fitted model or its serialized  
54 representation.

## 55 Estimation

The generative model for multivariate datasets in metasynth makes the simplifying assumption of marginal independence: each column is considered separately, just as is done in e.g., naïve Bayes classifiers (Hastie et al., 2009). Formally, this leads to the following generative model for the  $K$ -variate data  $\mathbf{x}$ :

$$p(\mathbf{x}) = \prod_{k=1}^K p(x_k) \quad (1)$$

There are many advantages to this naïve approach when compared to more advanced generative models: it is transparent and explainable, it is able to flexibly handle data of mixed types, and it is computationally scalable to high-dimensional datasets. As mentioned before, the tradeoff is the limited analytical validity when the independence assumption does not hold: in the synthetic data, the expected value of correlations, regression parameters, and other measures of association is 0.

For each data type supported by metasynth, there is a set of candidate distributions that can be fitted to that data type (see Table [Table 1](#)). To estimate the generative model of Equation [Equation 1](#), for each variable the software fits all compatible candidate distributions — by default with maximum likelihood estimation — and then selects the one with the lowest AIC ([Akaike, 1973](#)).

**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

## 71 A basic example

72 First, we create an example dataset using polars ([Vink et al., 2023](#)), the data frame library  
73 used internally in metasynth.

```
import polars as pl

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],
    }
)

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

74 Then, a model can be fitted and the intermediate representation can be stored as follows:

```
from metasynth import MetaDataset

# Ensure that the column "ID" has unique values
# when data is synthesized later
mds_spec = {
    "ID": {"unique": True}
}

# create metadataset
mds = MetaDataset.from_dataframe(df, spec=mds_spec)
```

```
# write to json
mds.to_json("metasynth_example.json")

75 After this, the json can be inspected or audited, potentially edited, and subsequently exported
76 from the secure environment. Then, outside the secure environment, the following code can
77 be run to synthesize new data:

# load json into a metadataset object
mds_out = MetaDataset.from_json("metasynth_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

- Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. *2nd International Symposium on Information Theory*, 267–281.
- Bates, A., Spakulová, I., Dove, I., & Meador, 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).
- Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (Vol. 2). Springer.
- 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.
- Vink, R., Gooijer, S. de, Beedie, A., Gorelli, M. E., Zundert, J. van, Hulselmans, G., Grinstead, C., Santamaria, M., Heres, D., ibENPC, Magarick, J., Leitao, J., Marshall, Wilksch, M., Heerden, M. van, Borchert, O., Jermain, C., Peek, J., Russell, R., ... Robert. (2023). *Pola-rs/polars: Python polars 0.18.8* (py-0.18.8). Zenodo. <https://doi.org/10.5281/zenodo.8167449>
- Wolf, P.-P. de. (2012). *Statistical disclosure control*. Wiley & Sons, Chichester.