ChatGPT

Synthetic Tabular Data: A Comprehensive Overview

Generation Techniques

Synthetic tabular data can be generated using a variety of techniques that model the distribution of a real dataset and then sample new records from that model. The main categories of generation methods include **traditional statistical modeling**, **modern deep generative models** (especially GANs and VAEs), **differential privacy-based methods**, as well as **hybrid and emerging approaches**:

- **Statistical Modeling Approaches:** Early methods for synthetic data generation were rooted in statistical modeling. These methods estimate probability distributions from the real data and draw random samples to create new data [1]. For example, a **Gaussian copula** model can capture the joint distribution and dependencies among variables, then sample synthetic records that preserve correlation structure [2]. Another common approach is the **chained equations** (sequential modeling) method, where one models the conditional distribution of each column given others and generates data column-by-column [3]. Such methods are akin to techniques used in multiple imputation for missing data. They are relatively fast and ensure basic statistical properties (like means, variances, correlations) of the synthetic data align with the original.

- **Generative Adversarial Networks (GANs) and VAEs:** In recent years, deep generative models have become popular for tabular data. **GAN-based approaches** pit two neural networks against each other – a generator tries to create realistic data while a discriminator attempts to distinguish synthetic from real, and both improve through competition [4]. GANs have shown success in producing high-fidelity synthetic tables and have been applied in various domains (from **CTGAN** for generic tabular data to **medGAN** for healthcare records) [4] [5]. Alongside GANs, **Variational Autoencoders (VAEs)** have also been adapted for tabular synthesis. A notable example is the **Tabular VAE (TVAE)**, which learns a latent representation of the data and can sample new points from this latent space [6]. In some studies, VAE-based models like TVAE perform on par with or even better than GANs for certain tabular datasets [7]. Modern variations include **autoregressive models** and **normalizing flows**, and even using **energy-based models (EBMs)**, but GANs and VAEs remain the most widely used deep models [8].

- **Differential Privacy Techniques:** A critical concern for synthetic data is privacy – ensuring that no individual's real record can be reconstructed or re-identified from the synthetic output. **Differential privacy (DP)** provides a formal framework to bound privacy risk. Some synthetic data generators incorporate DP by adding carefully calibrated noise or using privacy-preserving training algorithms. For example, **PrivBayes** is a method that learns a Bayesian network from data with differential privacy and then samples from it [9]. The open-source **DataSynthesizer** toolkit implements this approach, using Laplace noise in the Bayesian network estimation to guarantee a DP privacy budget [9]. There are also DP-enhanced GANs, such as **DPGAN** and **PATE-GAN**, which inject noise during GAN training or use teacher-student frameworks to satisfy differential privacy [10]. These techniques

produce synthetic data that statistically resemble the original while provably limiting the disclosure of any one real record.

- **Hybrid and Emerging Approaches:** No single generation method works best for all cases, and recent research often combines techniques. Hybrid approaches might ensemble multiple generators or apply post-processing to fix issues. For instance, a platform called **STNG (Synthetic Tabular Neural Generator)** simultaneously runs multiple generators (copula models, GANs like CTGAN, and VAEs) and selects or ensembles them to improve overall quality [4] [11]. Another frontier is using **diffusion models** (which iteratively refine random noise into realistic samples) and even **Large Language Models (LLMs)** for tabular data – these have been explored to address the complex distributions in tables [12] [10]. For example, **TabDDPM** and **TabDiff** apply diffusion processes to tabular data, and some methods treat rows as sentences to leverage LLMs' generative power [10]. These emerging methods and hybrids often aim to overcome specific challenges (like heterogeneous data types, or the need for semantic constraints) that traditional GAN/VAE models struggle with. In practice, the choice of generation technique may depend on the data characteristics and requirements: simple statistical models can be effective for smaller or well-behaved data, whereas neural networks handle complex distributions at the cost of more computation, and DP methods are chosen when privacy guarantees are paramount.

Use Cases of Synthetic Tabular Data

Synthetic tabular data has become increasingly valuable across industries and research because it provides *realistic but not real* data, enabling many applications. Key use cases include:

- **Privacy-Preserving Data Sharing:** One of the foremost applications is sharing or releasing data that maintains the insights of real data without exposing sensitive information. For example, in healthcare and finance, strict regulations (HIPAA, GDPR, etc.) limit sharing of real person-specific data [13]. Synthetic datasets can mimic the statistical properties of patient or customer data while ensuring that no actual individual's records are present, thus greatly reducing privacy risks [13]. Organizations use this to **collaborate or monetize data** safely – e.g. sharing a synthetic version of a clinical database with external researchers or providing synthetic financial transaction data to data science teams, all without violating privacy.

- **Data Augmentation and ML Training:** Synthetic data is used to supplement real datasets in machine learning. When real data is scarce, imbalanced, or missing certain variations, synthetic samples can **augment the training set**. This is common in scenarios like rare disease modeling or fraud detection where positive examples are few – generative models can create additional realistic examples to balance classes and improve model training. Because synthetic records retain the statistical patterns of real ones [14], models trained on a mix of real and synthetic data can generalize better. In fact, synthetic data allows expanding dataset size virtually unlimitedly, which **boosts statistical power** and model performance in domains with limited data [14]. Some practitioners even use *only* synthetic data to train initial models when real data collection is difficult or expensive.

- **Testing and Validation of Systems:** Synthetic tabular data is extremely useful for **software testing, QA, and pipeline validation**. Instead of using production data (which may be sensitive or cumbersome to refresh), companies generate synthetic copies of their databases to test new

features, ETL pipelines, or analytics dashboards [15]. Because the synthetic data closely resembles the shape and edge cases of the real data, it can reveal bugs or performance issues in a safe manner. This is also applied in **data science model validation** – for instance, validating that a model or an AI service behaves correctly on realistic inputs without risking exposure of actual user data. Synthetic data thereby enables robust testing environments that mirror production, improving software quality and compliance [15].

- **Self-Service Analytics and Education:** In some organizations, analysts or external partners need access to data to generate insights (for business intelligence or research) but cannot be given the real data for privacy or security reasons. High-quality synthetic tabular data can serve as a *drop-in replacement* for real data in exploratory analysis and even analytics competitions [16]. This democratizes data access internally while ensuring compliance. Additionally, in educational contexts and training workshops, synthetic datasets (based on real data distributions) are used to teach data science techniques without legal constraints.

- **Compliance and Fairness Testing:** Synthetic data can help address bias and fairness by creating controlled variations of data. For example, one can generate synthetic individuals to test how an AI model performs on minority groups without having access to large real datasets for those groups. Some synthetic data generators (like MOSTLY AI's platform) include features to ensure **fair representation** of subpopulations in the generated data [17]. This helps in auditing AI systems for bias and ensuring models are robust across scenarios. Furthermore, regulators and auditors can use synthetic data to evaluate algorithms in a sandbox environment.

In summary, synthetic tabular data is used wherever *data availability* and *data privacy* are at odds. It unlocks data for AI development [15], accelerates innovation by removing bottlenecks in data access, and does so while protecting individual privacy and adhering to regulations. Many industries (finance, healthcare, marketing, public sector) are increasingly relying on synthetic data for these purposes.

Tools and Libraries for Synthetic Tabular Data

A number of tools and libraries – both open-source and commercial – have been developed to generate synthetic tabular data. These tools differ in their approach, features, and ease of use. **Table 1** summarizes some of the most widely used solutions and their key characteristics:

Tool / Library	Type	Key Features and Methods
Synthetic Data Vault (SDV) [18]	Open-source Python library (by MIT Data-to-AI Lab)	Supports multiple models (statistical **copulas** and deep generative models like **CTGAN**, **CopulaGAN**, **TVAE**) for single-table, multi-table, and time-series data [19] [20]. Includes an evaluation framework (SDMetrics/SDGym) to assess synthetic data quality [18] [21]. Provides constraints handling (so synthetic data obey real-world rules) and some anonymization controls. Large community and actively developed ecosystem.

Tool / Library	Type	Key Features and Methods
YData-Synthetic [22]	Open-source Python library (by YData)	Focuses on GAN-based generation for tabular data. Implements **CTGAN** (for conditional tabular GAN), **TimeGAN** for time-series, and even a faster Gaussian mixture model option [22]. Provides a user-friendly GUI (Streamlit app) for end-to-end synthetic data generation and comparison [23]. Integrates with YData-Profiling to compare real vs synthetic distributions easily [24]. Emphasizes ease of use and education for data scientists.
Gretel.ai Synthetics [25]	Open-source library + Cloud API (Commercial SaaS)	Offers an API-driven platform for developers [26]. Supports multiple data types (tabular, text, even image) via different models – e.g., an LSTM-based neural network for sequential/tabular data, **ACTGAN** (an improved CTGAN), and a **DGAN** for time-series [25]. Highly customizable with configuration options, and can integrate into ML pipelines via REST APIs. Gretel provides cloud services to train and host synthetic data models, but also open-sources many model implementations for local use.
MOSTLY AI Platform [27]	Commercial Synthetic Data Platform	Enterprise-focused tool specialized in **privacy-preserving synthetic data** for structured data. Uses proprietary deep generative models (not openly published) that aim to maximize fidelity to real data while guaranteeing privacy (no one-to-one mappings) [28]. Particularly popular in banking and healthcare sectors for its strong compliance features [29]. Offers rich GUI and automation, and includes features like ensuring fairness (bias mitigation) in generated data [17]. Benchmark studies by the company showed it produces very high-fidelity data (e.g., an average quality score of 0.97 vs 0.74–0.82 for open-source GAN models in one comparison) [30].
R synthpop**** [31]	Open-source R package	One of the earliest tools (since 2016) for tabular synthetic data. Uses **sequential modeling of each column** using classification or regression trees (CART) or other statistical models to preserve multivariate relationships (based on the approach by Nowok et al. 2016). Primarily designed for simpler use cases and **multiple imputation** style synthetic data generation. Widely used in academia and by statisticians for quick generation of synthetic versions of small datasets.

Tool / Library	Type	Key Features and Methods
DataSynthesizer [32] [9]	Open-source Python package (research)	Focuses on **privacy-preserving generation**. Implements three modes: **Random** (fully random baseline), **Independent** (preserve 1-dimensional distributions), and **Correlated** which uses the **PrivBayes** algorithm (a differentially-private Bayesian network) to model joint distribution [9]. Allows setting a privacy parameter (epsilon) to trade off fidelity and privacy. Useful for research and education on DP synthetic data, originally developed by the Data Responsibly lab (Ping et al. 2017).
SynthCity [33]	Open-source Python library (academic)	A newer library (open-sourced in 2023) providing a collection of advanced generative models for tabular data. Includes not only GANs and VAEs for generic data, but also specialized models for **imbalanced data, survival analysis**, and more [33]. It has an emphasis on modular evaluation – offering built-in metrics for quality and **privacy (e.g., re-identification risk)** [34]. Aims to be a one-stop framework for experimenting with various synthetic data algorithms and evaluating them on common benchmarks.

Table 1: Prominent tools for generating synthetic tabular data, with their nature and key features. (Open-source tools are typically libraries to be used in code, whereas commercial platforms provide UI and enterprise integration.)

Other notable names include **Hazy** (a UK-based tool focusing on synthetic data for financial services and other enterprise data), **Tonic.ai** and **Delphix** (platforms for creating de-identified or synthetic copies of production databases for testing), and healthcare-specific generators like **Syntegra** or **MDClone** which specialize in medical records. There are also domain-specific simulators (e.g., **Synthea** for detailed synthetic electronic health records [35]) – these generate data by simulating underlying processes rather than directly learning from a dataset. The landscape is rapidly evolving, but the tools listed in Table 1 are among the most widely used as of 2025. The choice often depends on whether one needs an off-the-shelf enterprise solution with privacy assurances (commercial) or flexibility for research and custom modeling (open-source).

Quality Evaluation of Synthetic Data

Evaluating the quality of synthetic tabular data is a multi-faceted process. A “good” synthetic dataset should *statistically resemble* the real data (fidelity), be *useful* for intended tasks (utility), and *preserve privacy*. There is usually a trade-off among these goals [36], and no single metric suffices to judge quality [37]. Instead, practitioners use a suite of metrics and visual assessments. Most evaluation frameworks (such as **SDMetrics** in SDV or the **SynthEval** toolkit [38]) organize metrics into three broad categories: **statistical similarity (fidelity)**, **utility for ML/analytics**, and **privacy protection** [39]. **Table 2** gives an overview of these categories and example metrics:

Evaluation Aspect	Example Metrics	Purpose
Statistical Similarity (Fidelity) [40] [41]	*Distribution distance tests:* e.g. **Kolmogorov-Smirnov (KS) test** for continuous features, **Chi-Square test** for categorical features. *Aggregate statistics:* comparing means, variances, or correlation matrices between real and synthetic data. *Coverage:* checking that synthetic data covers categories or ranges present in real data (and doesn't introduce implausible out-of-range values).	These metrics assess how closely the synthetic data matches the original data's distributions. For example, a KS two-sample test can check if each numeric column in synthetic data comes from the same distribution as in real data [41]. High fidelity means the synthetic dataset preserves one-dimensional and multi-dimensional patterns (e.g., proper marginal distributions and inter-feature correlations) of the real dataset. A low fidelity score indicates the synthetic data may have significant statistical differences, which could limit its usefulness.
Utility for ML / Analysis [42] [43]	*Predictive performance:* **Train on Synthetic, Test on Real (TSTR)** evaluation – train a model (classifier or regressor) on synthetic data and measure its accuracy on real data (and vice versa, TRTS). *Machine learning efficacy:* comparing metrics like AUC, F1, or prediction error when models are trained on synthetic data versus on real data [42]. *Feature importance similarity:* checking if models trained on synthetic data pick up similar important features or relationships as models trained on real data.	These metrics evaluate whether synthetic data is *as useful as real data* for downstream tasks. The intuition is that if an AI model trained on synthetic data performs well on a real test set, the synthetic data has retained the important signal for that task [44]. Similarly, if statistical analyses (like regressions) yield the same conclusions on synthetic vs real data, the synthetic data has high utility. A specialized metric called **Quality Score (QScore)** or similar can summarize utility by combining several predictive performance indicators [45]. High utility means analysts and models can trust results from synthetic data to reflect real-world outcomes.

Evaluation Aspect	Example Metrics	Purpose
Privacy Protection [46] [47]	*Nearest-neighbor distance (DCR):* measuring the distance from each synthetic record to the closest real record – large distances indicate synthetic samples are not simply copies. *Exact match rate:* checking if any synthetic record exactly matches a real record (which would be a privacy red flag) [47]. *Membership inference resistance:* training an attacker model to distinguish real data points from synthetic – if it fails, privacy is stronger. *Differential Privacy guarantee:* if a method is DP, reporting the epsilon value which quantitatively bounds privacy loss.	These metrics aim to ensure that the synthetic data **does not leak sensitive information** about individuals in the original dataset. For instance, a low **exact match** count and high **row novelty** (each synthetic row is unique and not a duplicate of real data) indicate good privacy [47]. Privacy metrics like *singling-out risk, linkability,* and *inference risk* (as defined in GDPR contexts) are used to simulate attacks on the synthetic data [48] [49]. If the synthetic data was generated with a formal differential privacy mechanism, the privacy loss parameter (ε) is reported to give a guaranteed upper bound on information leakage. In practice, there is a privacy-fidelity trade-off: overly privacy-preserving data (e.g., too much noise or too generic) may lose fidelity, so a balance is sought [36].

Table 2: Key dimensions and example metrics for evaluating synthetic tabular data quality. High-quality synthetic data should achieve a good balance: it should be statistically similar to real data and useful for analysis, **without** memorizing or revealing any real record.

Modern synthetic data platforms typically provide automated **quality reports** covering these metrics. For example, SDV's `sdmetrics` library computes a suite of over two dozen metrics and even produces an aggregate quality score between 0 and 1 [21]. In one instance, they use metrics like a KS test and a logistic detection classifier (which tries to distinguish real vs synthetic) to score the data – the closer the synthetic is to real (and the harder to tell apart), the higher the score [41]. Likewise, **SynthEval** (an open-source framework introduced in 2024) offers a configurable benchmark of many metrics to compare synthetic data generators on both utility and privacy fronts [38]. It's recommended to evaluate synthetic data on multiple datasets and metrics, as a generator might excel in one aspect (say, fidelity) but fail in another (privacy).

Finally, qualitative checks are also important. Domain experts often validate synthetic data by inspecting it for **plausibility and coherence** (e.g., no negative ages, no children with income, etc.). Some post-processing techniques can enforce such logical constraints [50]. The evaluation is closely tied to the intended use case: a synthetic dataset intended for developing a classifier should be judged largely on that classifier's accuracy, whereas a dataset meant for public release might be primarily judged on stringent privacy criteria. As of now, there is no single universal benchmark for synthetic data quality [51] – it requires a careful assessment along multiple dimensions to ensure the synthetic data is fit for purpose and respects the necessary privacy guarantees.

Sources: The information above is drawn from recent literature and industry reports on synthetic data generation and usage, including surveys of generative methods [4] [6], use cases documented by

synthetic data providers [15] [13], tool documentation and comparisons [18] [52], and established metrics from synthetic data evaluation frameworks [41] [47]. These sources are cited throughout the text to provide evidence and additional context for each point made.

---

[1] [2] [3] [4] [5] [6] [7] [11] [13] [14] [18] [31] [32] A novel and fully automated platform for synthetic tabular data generation and validation | Scientific Reports
https://www.nature.com/articles/s41598-024-73608-0?error=cookies_not_supported&code=d15cdb7a-d3f5-469c-880e-344dca087ed9

[8] [10] [12] [50] [2504.16506] A Comprehensive Survey of Synthetic Tabular Data Generation
https://ar5iv.labs.arxiv.org/html/2504.16506v2

[9] [38] DataSynthesizer: Privacy-Preserving Synthetic Datasets | Request PDF
https://www.researchgate.net/publication/317352353_DataSynthesizer_Privacy-Preserving_Synthetic_Datasets

[15] [16] [19] [20] [21] [28] [30] [41] [42] [44] SDV vs MOSTLY AI: Which synthetic data generator is better? - MOSTLY AI
https://mostly.ai/blog/sdv-vs-mostly-ai-synthetic-data-generators-comparison

[17] [26] [27] [29] [35] [52] Top 6 Synthetic Data Generation Tools [2025]
https://averroes.ai/blog/synthetic-data-generation-tools

[22] [23] [24] [25] [33] [34] The Top 5 Python Packages to Generate Realistic Synthetic Data
https://ydata.ai/resources/top-5-packages-python-synthetic-data

[36] [37] [39] [40] [43] [45] [46] [47] [48] [49] [51] How to evaluate synthetic data quality
https://syntheticus.ai/blog/how-to-evaluate-synthetic-data-quality