Actuarial Innovation and Technology

Health Care Cost Trends

AI-Based Synthetic Medical Claims Data Generation: A Practical Guide for Actuaries

MAY | 2026

SOA Research INSTITUTE

AI-Based Synthetic Medical Claims Data Generation

A Practical Guide for Actuaries

AUTHORS Shea Parkes, FSA, MAAA

Zihua She

Jianxi Su, PhD, FSA

Xiao Wang, PhD

SPONSOR Actuarial Innovation and Technology Strategic Research Program

CONTENTS

Executive Summary **4**

Section 1 **Introduction** **5**

Section 2 **Potential Applications of Synthetic Medical Data** **8**
2.1 Structure-Focused Applications 8
2.2 Analytics-Focused Applications 8

Section 3 **Overview of the DE-SynPUF Data** **10**

Section 4 **Pretrained LLM-Based Synthetic Data for Training and Development** **11**

Section 5 **Generative Models for Statistically Representative Synthetic Data** **14**
5.1 Chain-Type Generative Framework 14
5.2 Variational Autoencoders 15
5.3 Generative Adversarial Networks 17
5.4 Transformers 18

Section 6 **An Illustrative Example** **22**
6.1 Modeling and Generating the Beneficiary Summary File 22
6.2 Modeling and Generating the Prescription Drug Event File 24

Section 7 **Privacy Considerations** **31**
7.1 The Origins of Privacy Risks in Synthetic Data 31
7.2 A Simple Verification Method 31
7.3 More Advanced Privacy Frameworks 32

Section 8 **Conclusion and Future Research** **34**

Section 9 **Supplemental Sample Prompts** **35**
9.1 Prompts for Beneficiary Summary File 35
9.2 Prompts for Prescription Drug Event File 37

Section 10 **Acknowledgments** **40**

About The Society of Actuaries Research Institute **41**

AI-Based Synthetic Medical Claims Data Generation

A Practical Guide for Actuaries

Executive Summary

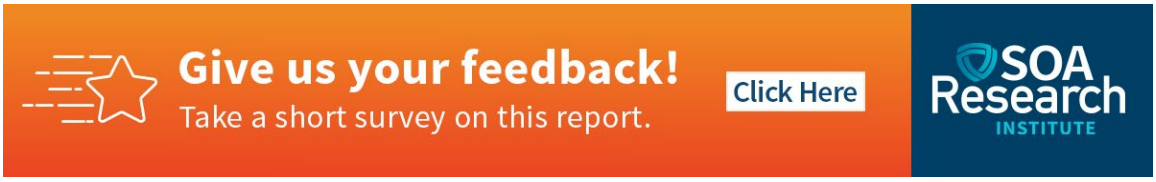
Medical claims data form the backbone of healthcare and insurance analytics. However, their sensitive nature raises significant privacy and regulatory concerns, limiting access to such data and impeding research, model development, and innovation across the actuarial and healthcare communities. Recent advances in artificial intelligence (AI) have opened new opportunities to overcome these long-standing barriers through the generation of synthetic medical data that preserve the statistical and structural characteristics of real data while containing practically no information about actual individuals.

This report presents a practical framework for actuaries to leverage generative AI techniques to create realistic, privacy-preserving medical claims data. Specifically, a chain-type generative framework is proposed for mirroring the hierarchical structure of healthcare data. Within the proposed framework, static beneficiary attributes are first simulated using tabular generative models, and then dynamic, time-dependent records are modeled using transformer-based architectures.

Two complementary approaches are demonstrated. For structure-focused applications, pretrained large language models (LLMs) such as ChatGPT can generate structurally coherent synthetic datasets that reproduce the schema of real medical claims data. For analytics-focused applications where statistical fidelity is essential, customized generative models are employed to replicate the joint distributions and temporal dependencies underlying the original datasets. A case study is provided to demonstrate the practical usefulness of the proposed framework for generating synthetic data with high structural and statistical fidelity.

Synthetic data generation techniques offer a transformative path for overcoming long-standing challenges related to privacy and data access. By enabling the safe sharing of realistic, privacy-preserving datasets, this approach opens new avenues for collaboration across organizational boundaries, including between research teams, industry stakeholders, and between academia and industry. Practitioners will still need to carefully review results for privacy concerns, especially when including patients with rare diseases and/or treatments in their training data. This report presents a meaningful step toward responsible data sharing, accelerating methodological advances, and fostering impactful applications in underwriting, health policy design, and beyond.

It is noteworthy that although this project is motivated by medical claims data, the proposed methodology and modeling tools are, in principle, broadly applicable to other types of insurance data with similar or simpler structures, such as life insurance and property insurance policy or claims data.



Section 1 Introduction

Medical claims data which contain patients' demographic information as well as healthcare records, including diagnoses, treatments, procedures, and costs, can offer valuable insights into healthcare utilization patterns, treatment outcomes, and the overall effectiveness of medical interventions. For the health insurance industry, medical claims data are particularly vital as they form the backbone of risk assessment, pricing strategies, and cost management. It can also help insurers detect fraudulent claims, manage provider networks, and design more effective healthcare plans.

However, as a society, the importance of treating medical claims data with a high level of privacy has been well recognized. Knowledge of an individual's healthcare history, while informing accurate predictions of future outcomes, could be exploited to bias decisions against that individual. Failing to uphold the confidentiality of medical claims data may lead to discrimination in various aspects of patients' lives. For this reason, access to real medical claims data are often limited due to privacy concerns and regulatory restrictions. In the United States, only companies with very specific business use cases are allowed access to healthcare data, but always with strong protections afforded by the Health Insurance Portability and Accountability Act (HIPAA).[1]

Synthetic data has emerged as a promising solution to address the privacy challenges that limit the availability and accessibility of medical claims data. High-quality synthetic data can retain the properties of the original dataset, while largely not being linked to any real individual records, thus preserving patient privacy.[2] Synthetic medical claims data holds great promise for enhancing insurers' ability to analyze healthcare trends, develop models, and conduct studies without compromising patient privacy, though whether these potentials can be fully realized remains to be seen.

The development of synthetic data can broadly be classified into two approaches. The first approach, often considered more naive or conventional, involves techniques such as de-identification, variable modification, or random reshuffling of data fields.[3] These methods aim to obscure direct identifiers (e.g., names, IDs, addresses) while retaining some analytical utility. However, they often fail to provide sufficient privacy protection and can also compromise the integrity of the data. In particular, variable modification or reshuffling can contaminate the joint distribution and correlation structure across variables, which are often essential for downstream tasks such as predictive modeling or risk adjustment. For example, the relationship between chronic conditions and treatment costs may be lost or distorted if diagnosis codes and claim amounts are randomly perturbed or mismatched. Additionally, these methods do not eliminate the risk of re-identification, especially in datasets with rare conditions or distinctive combinations of attributes. One illustrative case is that for a large family covered under a group insurance plan, even after removing direct identifiers, the family's collective profile may remain recognizable to individuals familiar with their circumstances, potentially exposing sensitive information through indirect inference.

In contrast, the second approach is more sophisticated and relies on applying some (statistical) algorithms to learn the underlying distribution of the original data. Once a model is trained, new synthetic data points can be simulated from this learned structure. Compared to other naive methods, such as de-identification, variable modification, or random reshuffling, discussed in the first approach, model-based generation is more compatible with formal privacy

[1] HIPAA is a U.S. federal law enacted in 1996 that establishes national standards for the protection of sensitive patient health information. For more information, see https://www.hhs.gov/hipaa.

[2] If building models from real patients' data, careful review should be completed on any generated synthetic data to ensure no training patient was exactly recreated.

[3] De-identification removes or obscures direct personal identifiers (e.g., names, Social Security numbers, dates of birth) from a dataset. Variable modification perturbs or recodes individual field values (e.g., rounding ages, generalizing ZIP codes) to reduce identifiability. Random reshuffling reassigns field values across records at random so that no single record retains its original combination of attributes. While these methods reduce re-identification risk to varying degrees, they do not provide formal mathematical privacy guarantees and may distort analytical properties of the data.

frameworks, such as differential privacy,[4] which offer mathematical guarantees against re-identification risks. Moreover, model-based generation allows for the preservation of complex relationships within the data, including interactions between patient demographics, healthcare utilization patterns, and cost trajectories. These structural relationships are essential for supporting actuarial modeling, forecasting, and decision-making. Still, it is important to always confirm that no generated patient is an exact copy of a real patient used to train the generative models.

In light of its practical advantages and theoretical rigor, this study adopts the model-based approach as its central focus. Unless otherwise specified, any reference to "synthetic data" throughout the remainder of this document is intended to mean data generated using such model-based approaches.

That said, modeling medical claims data are particularly challenging under the traditional statistical paradigm. On the one hand, there exist complex and nonlinear dependencies among variables in the data. For example, a beneficiary's demographic and socioeconomic characteristics can significantly influence the types of medical conditions they develop, the treatments and medications they receive, and the associated costs. These relationships often involve interactions and non-additive effects that are difficult to capture using traditional models like linear regression or generalized linear models, which typically assume simple parametric relationships between predictors and outcomes.

On the other hand, each participant in the dataset has medical history typically consisting of a variable-length sequence of healthcare events, such as physician visits, diagnoses, treatment, medication, and claims. Not only does the length of these sequences vary widely across individuals, but there is also meaningful temporal dependence within each sequence. For example, a prior healthcare event or diagnosis may affect subsequent medication patterns or follow-up care. Capturing this kind of heterogeneous, individual-specific trajectory of healthcare events poses significant challenges for traditional statistical models, which are often designed for fixed-length and independently observed records.

Recent advances in generative AI present a strong potential for using the technology to handle the story-like nature of longitudinal medical claims. At the core of these techniques are models capable of learning from complex, sequential, and context-rich data that is variable in length. Just as large language models (LLMs) generate coherent sentences by predicting the next word based on prior context, it is posited that generative models for healthcare data can simulate sequences of medical events by learning from a patient's historical records. This intuitive parallel between language and healthcare trajectories suggests that generative AI may be particularly well-suited for modeling individual medical histories in ways that traditional statistical approaches often struggle to achieve.

The purpose of this report is to illustrate how modeling techniques emerging from recent advances in generative AI can be applied to simulate realistic medical claims data. In particular, this report proposes a chain-type simulation framework specifically tailored for the structure of healthcare claims data. Variables involving medical claims data for an individual can typically be divided into two categories:

- Static variables: The first category consists of variables that are either inherently fixed (sex, race, enrollment age, place of birth, or genetic predispositions) or remain relatively constant over a given time period (e.g., chronic conditions, BMI, smoking status, disability status, or socioeconomic indicators). For this type of variable, it is reasonable to assume independent records across individuals, with uniform

[4] Formal privacy frameworks provide mathematical guarantees on the degree to which an individual's data can influence the output of a model or analysis. The most widely studied example is differential privacy, introduced by Dwork et al. (2006); see Dwork, C., McSherry, F., Nissim, K., & Smith, A. (2006). *Calibrating noise to sensitivity in private data analysis. Theory of Cryptography*, 265–284. https://doi.org/10.1007/11681878_14.

structure that does not evolve over time. In practice, this corresponds to the beneficiary summary file, which contains demographic and overall health status information for each policyholder.

- Dynamic variables: The second category consists of variables that accumulate over time, such as those found in inpatient, outpatient, and prescription claims. These include diagnosis codes, procedure codes, prescription drug identifiers, dates of service, days supplied, and claim costs. Unlike static variables, dynamic variables form variable-length sequences for each individual and exhibit temporal dependencies.

Given the structural differences between the two variable types, it is fair to argue that modeling and simulating the static variables is much more manageable than handling the dynamic ones. This report's chain-type framework addresses this by first generating a synthetic policyholder pool based on the static variables in the beneficiary summary file, and then conditional on each simulated profile, constructing an individualized sequence of dynamic variables representing that person's medical journey. This sequential, layered approach enables the selection of appropriate generative AI methods for each task, thereby reducing the risk of model misspecification, overfitting, and inflated uncertainty, potentially compromising both the credibility and usability of the simulated datasets. In particular, overfitting not only degrades generalization performance but also increases privacy risk, as models that memorize training data are more likely to leak sensitive individual-level information into the synthetic output.

It is important to clarify the scope and intent of the proposed synthetic data generation framework. The methodology developed in this study is not designed to generate arbitrary claims data for arbitrary populations. Rather, its sole objective is to learn and replicate the statistical and structural properties inherent in the input data on which the generator is trained. Consequently, any biases, coverage limitations, or population-specific characteristics that may be present in the original dataset such as demographic imbalances, selection effects, or benefit design constraints, will essentially be reflected in the synthetic output, ideally to a comparable extent when the generator is well specified and properly trained. In this sense, the synthetic data in the context of this report is intended as a privacy-preserving proxy for the input data, not as a mechanism for creating representative data beyond the support or quality of the original sample. If the training data are themselves well-represented and broadly reflective of the target population, the resulting synthetic data will inherit those favorable properties; conversely, if the input data exhibit sampling bias, the synthetic data will reproduce that bias rather than correct it. The primary goal of this framework is therefore not data creation in the abstract, but creating data with specific characteristics to facilitate responsible data sharing for legitimate business and research purposes, such as model development and validation, vendor benchmarking, educational training, regulatory testing environments, and cross-organizational collaboration where access to real claims data are restricted.

The remainder of this report is organized as follows. Section 2 reviews several potential applications of synthetic medical claims data, which can be broadly categorized into two classes. In the first class of applications, where data structure and formatting are the primary focus and preserving the statistical properties of the original data is less critical. Section 4 demonstrates how such synthetic data can be generated using pretrained LLMs. In the second class of applications, where synthetic data is intended for use in healthcare and insurance analytics, preserving the underlying statistical relationships is essential. To this end, Section 5 reviews generative AI models particularly well-suited for this purpose. An illustrative case study is presented in Section 6, providing step-by-step guidance for adopting these techniques. Throughout the report, all discussions and demonstrations are motivated by and based on the SynPUF dataset,[5] which is summarized in Section 3.

[5] The CMS DE-SynPUF dataset is publicly available from the Centers for Medicare & Medicaid Services at: https://www.cms.gov/data-research/statistics-trends-and-reports/medicare-claims-synthetic-public-use-files/cms-2008-2010-data-entrepreneurs-synthetic-public-use-file-de-synpuf.

Section 2 Potential Applications of Synthetic Medical Data

Depending on the purpose and required fidelity, the potential applications of synthetic data can be broadly categorized into two classes. These two categories differ in their objectives, synthetic data quality requirements, and implications for model design. Each class is elaborated in the subsections below.

2.1 STRUCTURE-FOCUSED APPLICATIONS

In the first class of applications, the primary focus is on data structure, format, and realism at the surface level, rather than on preserving the deeper statistical properties of the original dataset. In such cases, synthetic data serves as a sandbox environment, offering a safe and realistic alternative to real data that enables development, testing, or training activities without exposing sensitive patient information.

Key examples in this application category include:

- Training students and junior analysts on how to work with real-world data pipelines, perform joins across multiple healthcare files (e.g., beneficiary, inpatient, and prescription files).
- Demonstrating data tools or platforms (e.g., dashboards, ETL processes, SQL queries) in environments where production data cannot be used.
- Creating mock datasets for software development, vendor testing, or product demos in insurance or health-tech contexts.

Because these applications are often qualitative or engineering-oriented, exact statistical fidelity is not required. This allows for more flexible and lightweight data generation methods, such as prompting pretrained LLMs to fabricate claims-like datasets based on schema templates. A discussion of this approach is provided in Section 4.

2.2 ANALYTICS-FOCUSED APPLICATIONS

In contrast, the second class of applications involves quantitative analytics, where retaining the statistical and temporal characteristics of the original medical claims data are crucial. These applications rely on synthetic data that closely mirrors the real data's distributional properties, relationships, and dependencies, especially when used for modeling, forecasting, or evaluating actuarial assumptions.

Examples in this application class include:

- Training and validating predictive models, such as risk adjustment, cost forecasting, and pricing models, where the distributional characteristics and relationships among variables play a critical role and serve as key inputs to the process.
- Conducting actuarial simulations to evaluate strategic, financial, or regulatory scenarios by applying various shocks to test the resilience of insurance plans, pricing models, or financial projections, and to support decision-making. For example, actuaries may apply cost inflation shocks by increasing synthetic claim amounts, utilization shocks by raising the frequency of medical events, or morbidity shocks by modifying chronic condition profiles. Additional applications include simulating the effects of extremal events, assessing the impact of plan redesigns, and evaluating stop-loss or reinsurance coverage under stressed conditions.
- Supporting model governance, vendor evaluation, and regulatory filing through privacy-preserving testing environments. Specifically, synthetic data enables insurers to regularly validate internal models for accuracy, bias, and stability, and to compare the performance of external vendors' predictive tools under identical data conditions. This promotes transparency, supports defensible decision-making, and facilitates reproducible demonstrations for regulators when real claims data cannot be shared.

These applications rely on synthetic data generated by models that learn and replicate the underlying data-generating process, capturing both marginal distributions and complex dependencies. Section 5 will review such generative AI models designed to support this type of high-fidelity data generation.

Section 3 Overview of the DE-SynPUF Data

To ensure that all illustrations and case studies in this report are reproducible and publicly accessible, the analysis is based on the CMS DE-SynPUF dataset (2008–2010 version), released by the Centers for Medicare & Medicaid Services.[6] Although the SynPUF dataset is itself a form of synthetic data, it remains highly valuable for the purposes of illustration and education. The SynPUF dataset retains the general structure, coding logic, and data formats of real Medicare claims; thus, it is an effective stand-in for demonstrating how to build model-based synthetic data generators in a realistic setting.

Moreover, although the DE-SynPUF is labeled as "synthetic," it is not a model-based synthetic dataset in the sense emphasized in this report. Instead, it represents a conventional synthetic approach constructed through de-identification, variable suppression, random shuffling, and value perturbation techniques designed to reduce re-identification risk while maintaining general utility. This work is not intended as simply using AI to "learn from another synthetic dataset." Instead, the SynPUF data serves as a proxy for real-world data, which is used to demonstrate workflows, modeling decisions, and technical challenges that would arise in actual applications.

The SynPUF dataset contains several claim types, but this report focuses on two core files:

- The Beneficiary Summary (BS) File, which includes static variables such as age, gender, race, and indicators for chronic conditions.
- The Prescription Drug Event (PDE) File, which records dynamic prescription events, such as drug name, fill date, quantity dispensed, and costs.

This study focuses exclusively on data from the year 2008 to simplify illustrations. While inpatient and outpatient claims files are also available, their structural complexity makes them less suitable for this pedagogical purpose and thus are excluded from this report. After the necessary data processing, the data volume used to train the synthetic data generator in this illustrative example includes 72,848 beneficiaries with 1,995,229 prescription drug records.

[6] Source: https://www.cms.gov/data-research/statistics-trends-and-reports/medicare-claims-synthetic-public-use-files/cms-2008-2010-data-entrepreneurs-synthetic-public-use-file-de-synpuf.

Section 4 Pretrained LLM-Based Synthetic Data for Training and Development

In certain applications, synthetic data are required not for statistical modeling or inference, but rather for system testing, software development, or educational demonstrations. In these cases, the goal is to reproduce the structural and logical patterns of real-world datasets without preserving their statistical properties. Such data are particularly valuable for verifying data pipelines, testing interface compatibility, and demonstrating analytic workflows while eliminating privacy or compliance concerns, since the content is entirely artificial.

Pretrained LLMs can be effectively employed to generate highly structured and internally coherent synthetic datasets that mimic the schema of the real-world medical claim data, such as the official CMS data. In this report, detailed prompts have been developed to instruct a pretrained LLM to generate synthetic BS data consistent with the format of the CMS DE-SynPUF BS file, and PDE data aligned with the format of the CMS DE-SynPUF prescription drug file.

Two complementary sets of prompts are developed to operationalize a coherent two-step procedure for generating these files in a consistent and relational manner. The first set of prompts instructs the LLM to generate the BS file, while the second focuses on producing the PDE file. The beneficiary prompt serves as the foundation for establishing a synthetic population whose demographic, coverage, and chronic condition attributes define the basis for subsequent prescription activity. The prescription prompt then builds on this foundation and generates multiple synthetic drug events for each beneficiary to capture longitudinal utilization patterns.

In the first step, the LLM is guided by a prompt specifying the creation of synthetic BS data consistent with the structure and logic of the CMS DE-SynPUF Beneficiary file. The prompt requires the output to include exactly 32 columns, beginning with a header row that lists each field name in the precise CMS order. These variables collectively describe a beneficiary's demographic profile, vital status, state and county of residence, Medicare coverage months, chronic condition indicators, and annual reimbursement amounts across inpatient, outpatient, and carrier service categories. The prompt further defines explicit generation rules for each field to ensure internal realism and data integrity. Dates of birth and death are required to follow chronological order, with death dates omitted for beneficiaries presumed alive. Coverage months are generated as integer values between zero and twelve, and chronic condition indicators are coded as binary responses following CMS conventions. Reimbursement amounts for inpatient, outpatient, and carrier services are represented as non-negative dollar values with two-decimal precision and magnitudes consistent with typical annual healthcare costs.

In the second step, the model is guided by a corresponding set of prompts designed to generate synthetic PDE data consistent with the structure and logic of the CMS DE-SynPUF PDE file. This second set of prompts extends the relational framework established in the BS file by generating multiple prescription records for each synthetic patient, using the shared DESYNPUF_ID field as the linking key. As a result, each beneficiary may be associated with several distinct drug events, thereby capturing the longitudinal nature of medication utilization patterns within the synthetic population.

The PDE prompt specifies the creation of exactly eight variables: DESYNPUF_ID, PDE_ID, SRVC_DT, PROD_SRVC_ID, QTY_DSPNSD_NUM, DAYS_SUPLY_NUM, PTNT_PAY_AMT, and TOT_RX_CST_AMT. Each variable mirrors its CMS counterpart in both naming and format. These fields together describe the prescription-level attributes of a drug claim, including the beneficiary identifier, a unique transaction ID, the date of service, the dispensed product code, the quantity and duration of supply, and the associated financial details. The prompt also defines explicit generation rules to ensure internal consistency and realism. For example, SRVC_DT values are formatted as eight-digit dates within a plausible multi-year window (e.g., 2008–2010), while product identifiers (PROD_SRVC_ID) follow an eleven-digit numeric structure resembling the National Drug Code used in real CMS data. Quantities dispensed and days supplied are generated as positive integers within typical ranges and are designed to be correlated, reflecting real

dispensing behavior. Cost variables are formatted as dollar amounts with two decimal precision, with the total drug cost always greater than or equal to the patient payment amount.

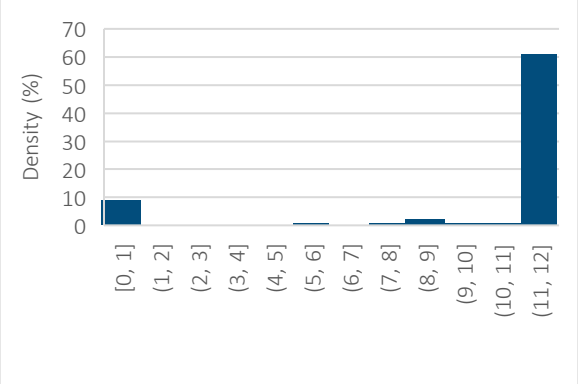
To emulate real-world data imperfections, the prompt allows occasional missing values in selected numeric fields such as quantity dispensed, days supplied, patient payment, or total cost, while ensuring that key identifiers and dates remain complete. Logical and numeric consistency checks are incorporated to maintain coherence across records, preventing duplicate PDE identifiers, enforcing positive values for quantities and costs, and ensuring that higher dispensed quantities correspond to proportionally larger costs.

Both sets of prompts produce outputs directly in CSV format, beginning with a single header line followed by any user-specified number of data records. Each record is comma-separated and formatted to ensure immediate compatibility with analytic workflows, database ingestion processes, and software testing environments designed to handle the original data. The complete prompt specifications for generating the BS and PDE files are provided in Section 9. As an illustration, Figure 1 and Figure 2 present examples of synthetic records for representative variables from the BS and PDE files, respectively. The comparison between the distributions observed in the real CMS data and those in the LLM-generated synthetic data reveals noticeable differences, underscoring that the latter are designed only to reproduce structural patterns rather than replicate true statistical relationships.

Figure 1

ILLUSTRATION OF THE DIFFERENCE BETWEEN THE REAL AND SYNTHETIC DISTRIBUTIONS OF TOTAL MONTHS OF MEDICARE PART A (HOSPITAL INSURANCE) COVERAGE DURING THE REFERENCE YEAR

HISTOGRAM OF THE NUMBER OF ELIGIBILITY MONTHS IN THE ORIGINAL DATA



HISTOGRAM OF THE NUMBER OF ELIGIBILITY MONTHS IN THE GENERATED DATA

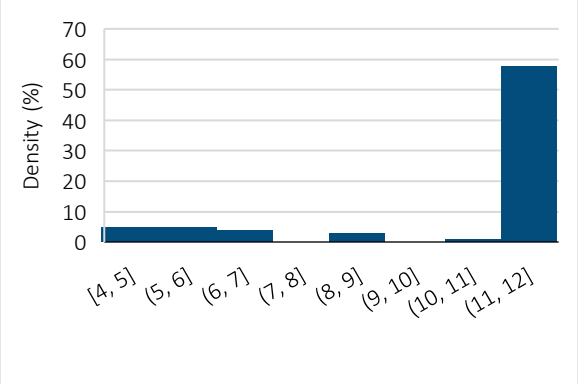
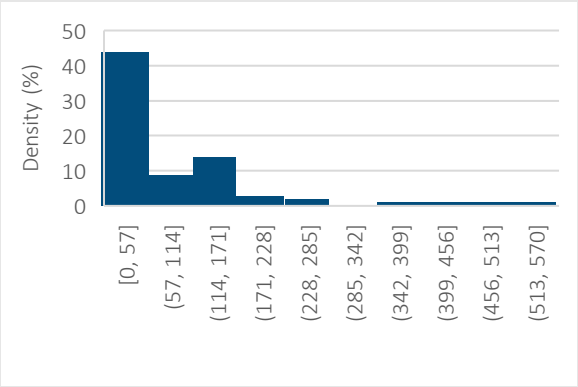


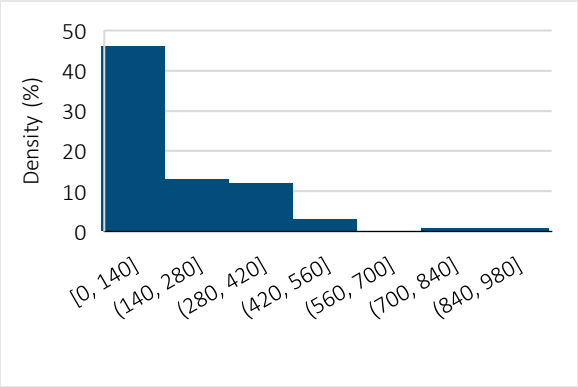
Figure 1

ILLUSTRATION OF THE DIFFERENCE BETWEEN THE REAL AND SYNTHETIC DISTRIBUTIONS OF PER-BENEFICIARY AGGREGATE DRUG COST DURING THE REFERENCE YEAR

HISTOGRAM OF THE AMOUNT OF TOTAL COST IN THE ORGINAL DATA



HISTOGRAM OF THE AMOUNT OF TOTAL COST IN THE GENERATED DATA



Section 5 Generative Models for Statistically Representative Synthetic Data

As discussed in earlier sections, many real-world applications such as risk adjustment, pricing, and healthcare policy simulation require synthetic data that faithfully preserves the statistical properties of the original dataset. Unlike conventional methods that rely on de-identification and reshuffling, model-based synthetic data generation aims to learn the underlying data-generating process, capturing both marginal distributions and complex relationships among variables. This section introduces three generative modeling frameworks that are considered particularly well-suited for producing high-fidelity, statistically representative synthetic medical claims data.

5.1 CHAIN-TYPE GENERATIVE FRAMEWORK

To address the structural complexity of medical claims data, this report proposes a chain-type generative framework that mirrors the natural process by which such data is generated. Specifically, the framework begins by modeling and simulating the static beneficiary-level information, such as demographic and health status variables. Then, conditional on these synthetic profiles, the framework proceeds to simulate individualized sequences of medical events that reflect each patient's longitudinal experience. This sequential setup decomposes the synthetic data generation task into two stages, each of which is addressed using a model tailored to the specific data structure and complexity.

In the first stage, the static component is generated, which comprises variables that are either fixed or exhibit slow, predictable changes, such as age, sex, race, chronic condition indicators, and socioeconomic characteristics. These variables are typically stored in the BS file and are treated as independent records across individuals. For this stage, generative models tailored to structured tabular data, particularly Variational Autoencoders (VAEs) and tabular Generative Adversarial Networks (GANs), can be applied. These models are well-suited to capturing the marginal distributions and inter-variable dependencies within the static beneficiary file.

In the second stage, the framework models and generates the dynamic component, which comprises medical activity records such as inpatient visits, outpatient events, or prescription fills, which are longitudinal, variable in length, and unfold over time for each simulated beneficiary. These synthetic sequences are generated conditionally on the static profiles created in the first stage. Given the temporal and contextual dependencies inherent in such data, the framework adopts generative models specifically designed for sequence modeling, particularly transformer-based architectures, to simulate each patient's personalized medical journey.

The justification for adopting this chain-type structure is twofold. On the one hand, it mirrors the natural process by which real-world medical claims data are generated, thereby enhancing both the realism of the synthetic dataset and the transparency of the simulation process. On the other hand, it enables each generative model to focus on a specific, well-scoped subtask. This modularity not only increases flexibility but also makes the overall modeling process more tractable and easier to calibrate, helping to reduce the risk of overfitting and model misalignment. In the subsections that follow, three classes of generative models are described, which can be used within this framework, and explain how they are adapted to each stage of the synthetic data pipeline.

5.2 VARIATIONAL AUTOENCODERS

A core mechanism underlying many generative AI models is the use of low-dimensional latent representations to capture the essential structure of high-dimensional data. This idea is closely tied to the manifold hypothesis,[7] which posits that real-world data existing in high-dimensional space actually lies on or near a lower-dimensional manifold governed by a smaller set of underlying factors. For example, the distribution of patient characteristics or disease progressions may vary along just a few interpretable axes (e.g., age, comorbidity burden, and socioeconomic status), even though the observed data may span hundreds of variables.

Two widely used approaches built upon this principle are VAEs and GANs, which will be discussed in more detail one by one in the sections that follow. VAEs are discussed first. They operate by encoding input data into a probability distribution over a low-dimensional latent space that is easy to sample from. To reconstruct or synthesize realistic data, new data instances are then generated by decoding samples drawn from this latent space.

To explain the methodology, let x be the observed data, which stands in a high-dimensional space and introduce a latent variable z, chosen to be lower-dimensional which can capture the essential structure underlying the data x. The distribution of the latent variables, also known as the prior, $p(z)$, is often chosen to be simple and tractable to simulate, most commonly a standard multivariate Gaussian $N(0, I)$.

The aim of the model is then to uncover the essential factors of variation embedded in this latent space and establish their connection to the observed data. To achieve this goal, VAEs employ two major components: an encoder which compresses data into latent codes, and a decoder which reconstructs data from latent codes. The encoder, parameterized by ϕ, maps the input data x into an approximate posterior distribution over the latent variables, $q_\phi(z|x)$. The approximate posterior distribution reflects that given a data x, what values of latent variable z are most likely responsible for generating it. It is typically modeled as a conditional Gaussian distribution with mean and variance that are themselves outputs of a neural network with parameters ϕ. Concretely, the encoder takes an observation data point x as input, and outputs the parameters of the approximate posterior distribution of z, such that:

$$q_\phi(z|x) \sim N(z \mid \mu_\phi(x), \sigma_\phi^2(x)\, I\,).$$

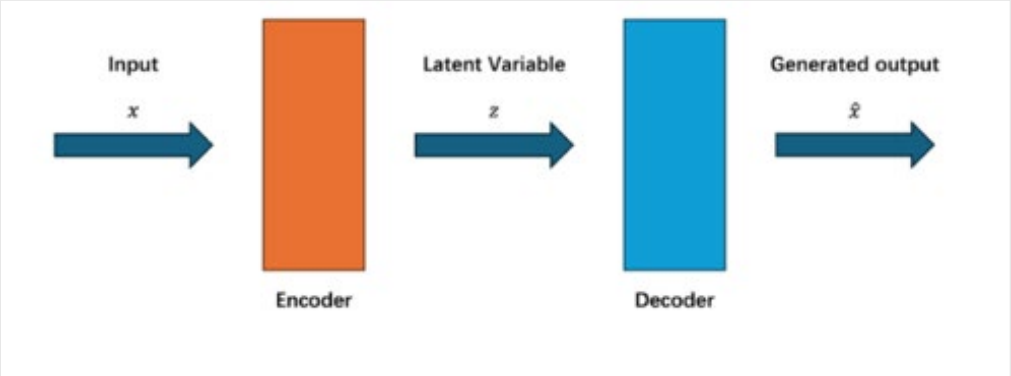
The decoder is another neural network parameterized by θ that maps latent representation z back into the data space, thereby reconstructing the original input or generating new synthetic instances. Given a latent variable z, the decoder outputs the parameters of a conditional distribution for x, which is commonly modeled as a Gaussian:

$$p_\theta(x|z) \sim N(x \mid \ \mu_\theta(x), \sigma_\theta^2(x)\, I).$$

Figure 3 provides a schematic overview of the VAE architecture and illustrates how the encoder maps data into the latent space and how the decoder reconstructs data from latent representations.

[7] The manifold hypothesis is the widely used idea that high-dimensional real-world data tend to concentrate on or near a lower-dimensional manifold. Although it is not usually attributed to a single definitive origin paper, a commonly cited formal discussion appears in Fefferman, Mitter, and Narayanan. See Charles Fefferman, Santosh Mitter, and Hrushikesh Narayanan, "Testing the Manifold Hypothesis," *Journal of the American Mathematical Society* 29, no. 4 (2016): 983–1049, https://doi.org/10.1090/jams/852https://pubs.ams.org/journals/jams/2016-29-04/S0894-0347-2016-00852-4.

Figure 3
ILLUSTRATION OF THE ARCHITECTURE OF A VAE



To train a VAE, it is natural to maximize the log-likelihood of the observed data which can be written as:

$$\log p_\theta(x) = \log \int p(z)\, p_\theta(x|z)\, dz.$$

However, this data likelihood is intractable because it involves integrating over latent variables in a high-dimensional space, and the resulting integral has no closed form expression. Consequently, direct maximization of the likelihood becomes computationally infeasible.

To get around the issue, rewrite the observed data likelihood as:

$$\begin{aligned}\log p_\theta(x) &= E_{z\sim q_\phi(z|x)}\,(\log p_\theta(x)) \\ &= E_{z\sim q_\phi(z|x)}\,(\log p_\theta(x,z) - \log p_\theta(z|x)) \\ &= E_{z\sim q_\phi(z|x)}\,(\log p_\theta(x|z) + \log p(z) - \log p_\theta(z|x)) \\ &= E_{z\sim q_\phi(z|x)}\,(\log p_\theta(x|z)) - D_{KL}\left(q_\phi(z|x)\,||\,p_\theta(z)\right) + D_{KL}\left(q_\phi(z|x)\,||\,p_\theta(z|x)\right),\end{aligned}$$

where $D_{KL}(p||q) = \int_{-\infty}^{+\infty} p(x) \times (\log p(x) - \log q(x))dx$ represents the Kullback–Leibler (KL) divergence between two probability distributions p and q. Among the three terms in the expression, the first term, $E_{z\sim q_\phi(z|x)}\,(\log p_\theta(x|z))$ represents how the original data would be reconstructed from the latent codes. This term is tractable because latent variables z can be conveniently sampled from the approximate posterior $q_\phi(z|x)$, and the distribution of the decoder, $p_\theta(x|z)$ has an explicit form. With the reparameterization trick for Gaussian distributions, the expectation in the first term becomes differentiable with respect to both ϕ and θ, which enables users to apply gradient descent to efficiently estimate these parameters.

The second term, $-D_{KL}\left(q_\phi(z|x)\,||\,p_\theta(z)\right)$ acts as a regularizer that measures how far the approximate posterior is from the prior distribution. Since both p and q are chosen to be Gaussian distributions, the KL divergence has a closed-form expression, which makes the second term straightforward to compute during training.

That said, the third term, $D_{KL}\left(q_\phi(z|x)\,||\,p_\theta(z|x)\right)$ is the KL divergence between the approximate posterior and the true posterior. This term is intractable because the involved true posterior $p_\theta(z|x)$ has no closed-form solution. For this reason, the third term is dropped, leaving a tractable Evidence Lower Bound (ELBO) that serves as a surrogate objective for training VAEs. Namely, the objective function for training a VAE is given by:

$$\mathcal{L}(x,\theta,\phi) = \mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}\big(q_\phi(z|x)||p_\theta(z)\big).$$

After training, new data can be generated by first sampling a latent variable z from the prior distribution $p(z)$, and then passing it through the trained decoder, $p_{\hat{\theta}}(x|z)$.

5.3 GENERATIVE ADVERSARIAL NETWORKS

GANs are another class of powerful generative models that approach data synthesis from a game-theoretic perspective. Different from VAEs, which explicitly model a likelihood function, GANs rely on an adversarial setup between two neural networks: a generator and a discriminator. The generator starts with random noise sampled from a simple distribution (e.g., standard Gaussian) and transforms it into synthetic data. The discriminator, on the other hand, is trained to distinguish real data samples from synthetic ones. During training, the two networks are pitted against each other in a minimax optimization problem, where the generator tries to produce data convincing enough to fool the discriminator, and the discriminator strives to correctly classify inputs as real or fake. Through this competitive dynamic, both models improve over time, with the generator learning to capture the distribution of real data, while the discriminator becomes increasingly good at detecting slight differences between real and synthetic data.

To formalize the description of a GAN, the following key components are defined:

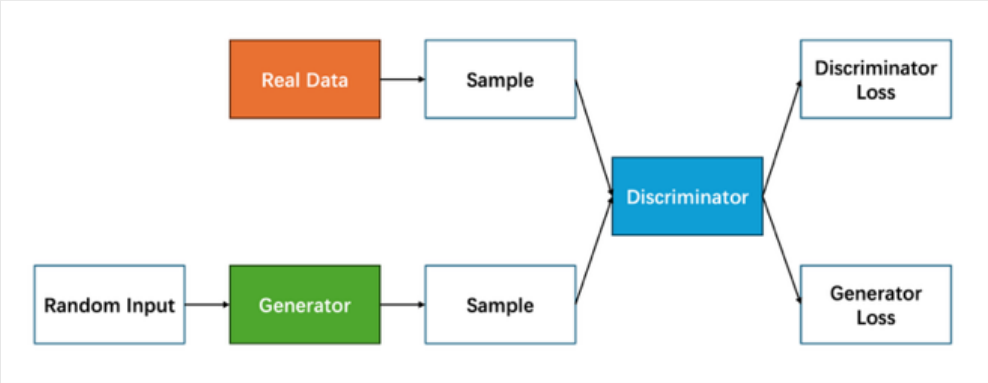
- $p_{data}(x)$ denotes the distribution of real data x;
- $p(z)$ represents a simple prior distribution over the latent variable z, typically chosen as a standard multivariate Gaussian;
- The generator, $G_{\theta_g}(z)$ is a neural network with parameter θ_g that transforms a latent variable z into a synthetic data point.
- The discriminator, $D_{\theta_d}(x)$ is another neural network with parameter θ_d that outputs a value between zero and one, which indicates the probability that the input x comes from the real data distribution rather than being generated.

With these definitions in place, the training process is cast as a two-player minimax game with the following objective:

$$\min_{\theta_g}\max_{\theta_d}[\mathbb{E}_{x\sim p_{data}(x)}\log D_{\theta_d}(x) + \mathbb{E}_{z\sim p(z)}\log(1 - D_{\theta_d}\left(G_{\theta_g}(z)\right)],$$

Here, the discriminator aims to maximize its ability to correctly classify real data as real (first term) and fake data as fake (second term), while the generator seeks to minimize the second term by producing synthetic data that the discriminator cannot distinguish from real data. Figure 4 illustrates the adversarial training architecture of a GAN.

Figure 4
ILLUSTRATION OF THE ARCHITECTURE OF A GAN



To conclude the two subsections on VAEs and GANs, it is noteworthy that both methods are powerful yet fundamentally different. VAEs represent a probabilistic framework that enables efficient inference of latent variables and stable optimization. Moreover, the continuous and well-behaved latent space of VAEs allows for meaningful interpolation and counterfactual simulation, which can be useful in domains like healthcare, finance, and insurance. However, their reliance on maximizing a lower bound of the likelihood often results in generated samples that are perceptually less sharp and less realistic. In contrast, GANs aim to produce realistic data through an adversarial training process between a generator and a discriminator. This framework typically yields sharper and more realistic outputs but suffers from training instabilities such as vanishing gradients[8] and mode collapse.[9] Ultimately, the choice between VAEs and GANs depends on the modeling goal. If the priority is interpretability and learning structured latent representations, VAEs are often more suitable. On the other hand, if the objective is to capture complex joint distributions and generate high-fidelity synthetic data, GANs may offer better performance.

5.4 TRANSFORMERS

VAEs and GANs discussed thus far are well-suited for modeling data when the observations can be treated as independent and identically distributed, which is a reasonable assumption for beneficiary information data such as age, sex, comorbidities, and socioeconomic attributes. These models effectively capture complex marginal distributions and inter-variable dependencies without needing to account for temporal structure. However, for claims data such as prescription records, which are inherently sequential and exhibit strong temporal and contextual dependencies, models designed for independent data fall short. In such cases, transformer-based architectures have emerged as a natural and useful choice due to their ability to model long-range dependencies and contextual patterns within variable-length sequences.

Transformers represent a more recent advancement in generative modeling. Originally developed for natural language processing, they have gained traction in domains involving sequential and structured data, including financial transactions, manufacturing logs, consumer behaviors, and healthcare records. Compared to traditional machine learning methods, a key innovation of transformer models lies in their self-attention mechanism, which

[8] In the context of GANs, vanishing gradients often occur when the discriminator becomes too strong too quickly. If the discriminator easily distinguishes real from fake data, the generator receives little useful gradient information to improve, causing its training to stall. This disrupts the adversarial balance and can prevent the generator from learning to produce realistic outputs.
[9] Mode collapse refers to a failure mode where the generator learns to produce only a limited set of outputs regardless of the input noise. Instead of capturing the full diversity of the target data distribution, the generator collapses to generating a few "safe" outputs that consistently fool the discriminator, resulting in low variety in the synthetic data.

allows each element in a sequence to dynamically attend to all other elements at once, capturing both local and long-range dependencies efficiently.

In the self-attention mechanism, each element in a sequence tries to understand how much attention to give to every other record when forming its contextual representation. The mathematical foundation of the self-attention mechanism in transformers is the scaled dot-product attention function, which quantifies how strongly each element in a sequence attends to every other element. To this end, each element in the sequence is transformed into three distinct vector representations, which are collectively organized into the matrices Q, K, and V, corresponding respectively to the following:

- Query (Q): The element's "question," representing what it is looking for;
- Key (K): The element's "label," representing what information it offers;
- Value (V): The element's actual information content.

Given a sequence represented by queries Q, keys K, and values V, the dot-product attention function is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V,$$

where d_k is the dimension of the keys used for scaling. The softmax function takes a list of real numbers, say $\boldsymbol{v} = (v_1, \ldots v_n)$, and then converts them into positive weights that sum to 1. That is $\text{softmax}(\boldsymbol{v}) = (p_1, \ldots, p_n)$ such that:

$$p_i = \frac{e^{v_i}}{\sum_{j=1}^{n} e^{v_j}}.$$

Intuitively, the dot-product attention function captures the similarity between queries and keys to determine the weights for the values, so that the model can learn the relations across the entire sequence.

To better understand the intuition behind the mathematical formulation of the self-attention mechanism, consider a hypothetical patient's sequence of prescriptions, summarized in Table 1. For each record, its relevant information such as drug type, fill date, quantity, days supplied, and cost, is first converted into a vector representation x_i, $i = 1,2,3$. When a transformer model processes this sequence, it generates three distinct representations for each x_i:

$$q_i = W^Q x_i,\ k_i = W^K x_i, \text{ and } v_i = W^V x_i,$$

where W^Q, W^K, and W^V are parameter matrices learned during model training. These matrices transform the same input vector into three different roles, query, key, and value, that correspond to the conceptual functions described earlier.

Table 1

A HYPOTHESIZED PRESCRIPTION RECORD.

Record	Drug name	Fill Date	Quantity	Days Supplied	Total Cost
1	Metformin	Jan 1	30	30	$10
2	Lisinopril	Feb 1	30	30	$12
3	Atorvastatin	Mar 1	90	90	$25

Now, consider the second record in the sequence, where the prescribed drug is Lisinopril. To determine how much attention this record pays to Metformin and Atorvastatin, the transformer formulates this by taking Lisinopril's query vector q_2 and comparing it with the key vectors $k_1, k_2,$ and $k_3$corresponding to all records in the sequence. These comparisons quantify how relevant each record is to understanding the current one. For example, since diabetes (Metformin) and hypertension (Lisinopril) often co-occur, the similarity measure $q_2 \cdot k_1$ is expected to be relatively high, indicating a strong association. In contrast, cholesterol medication (Atorvastatin) is less directly related to hypertension, so the similarity $q_2 \cdot k_3$ would likely be lower, reflecting weaker relevance.

Thus far, the amount of attention the Lisinopril record gives to each of the other prescriptions has been determined. To aggregate the relevant information, the value component comes into play. Specifically, the actual information content vectors v_i, $i = 1,2,3$, are combined to form an updated representation for each record. The output corresponding to the i-th record, originally represented by x_i, is computed as:

$$\text{Output}_i = p_{i,1}v_1 + p_{i,2}v_2 + p_{i,3}v_3,$$

where $p_{i,j}$ denotes the attention weight from record i to record j, derived from the softmax of the scaled dot products between the query q_i and all keys k_j. Intuitively, this weighted summation allows each prescription record to update its representation by selectively incorporating information from other records in the sequence, with greater emphasis placed on those deemed more relevant through the attention weights.

The attention mechanism discussed thus far can be regarded as a single-head attention, in which the model behaves like a single "mind" focusing on one specific type of relationship within the data. In the previous illustrative example, this relationship corresponds to the clinical co-occurrence between hypertension and diabetes. However, in many real-world applications, data often exhibit multiple kinds of dependencies that coexist along different dimensions. In the context of the prescription example, additional relationships may include temporal patterns, such as how far apart prescriptions are refilled, and cost-related dependencies, which capture associations among prescriptions based on their cost characteristics.

To capture different types of dependencies simultaneously, the single-head attention mechanism can be generalized to multi-head attention, which can be viewed as involving multiple independent "minds" that examine the same sequence data from different perspectives. Specifically, after transforming the input sequence into three global matrices Q, K, and V, each head is defined as:

$$\text{head}_h = \text{Attention}\left(QW_h^Q, KW_h^K, VW_h^V\right), \quad h = 1, \dots, H.$$

Each head processes the same information Q, K, and V but through its own lens, which is formulated by the projection matrices W_l^Q, W_l^K, W_l^V, which project the representations into distinct subspaces. This design enables each head to specialize in capturing a particular type of dependency or relational pattern within the data. The choice of the number of heads H involves balancing model capacity, computational cost, and interpretability.

After all H heads compute their respective attention outputs, these outputs are concatenated and linearly transformed to form the final multi-head attention output:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_H)\, W^O,$$

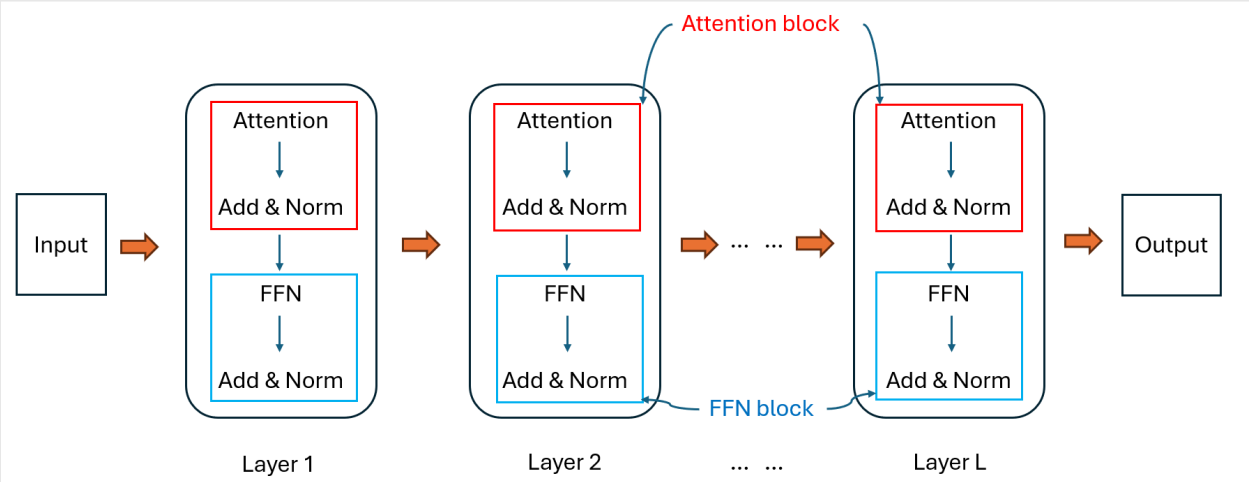
where W^O is an additional projection matrix. This step integrates the diverse information captured by different heads into a unified representation, which is then passed to the subsequent block of the Transformer for further refinement and stabilization of the learned patterns. Beyond the attention mechanism, each element in the sequence (e.g., each prescription record in the illustrative example) is processed through a position-wise feed-forward network (FFN). Each FFN operates independently on its position, and it applies nonlinear transformations to enrich the linear aggregations of information produced by the attention mechanism.

Another important architectural component is the use of residual connections and layer normalization. Residual connections add the input of a block (e.g., attention or FFN block) back to its output, thus preserving the original information while incorporating new features learned by the sub-layer. This design alleviates a common problem of vanishing gradient in training deep architectures. Layer normalization is then applied to the combined output to rescale representations to a consistent range, which can improve numerical stability and convergence speed.

Collectively, a Transformer is composed of multiple stacked layers, each containing one attention block and one FFN block. While the layers share the same architectural design, they do not share parameters; each layer has its own independently learned weights, allowing different layers to capture dependencies at varying levels of abstraction. Each layer takes the output of the previous layer as input and passes it sequentially through these two sub-components. By iterating the attention–FFN pair across many layers, the Transformer progressively constructs finer, more contextual representations, resulting in a hierarchical understanding of the patterns and dependencies in sequential data.

Figure 5 summarizes this multi-layer, two-block architecture of the Transformer described so far.

Figure 5
ILLUSTRATION OF THE STACKED-LAYER STRUCTURE IN A TRANSFORMER



Once trained, the Transformer can generate synthetic sequential data in an autoregressive manner.
In the context of prescription record generation, the process begins with an initial token containing a beneficiary's background information. The Transformer then predicts the conditional distribution of the next record and samples the first synthetic prescription from it. This newly generated record is appended to the input sequence, and the model continues predicting subsequent prescription events one step at a time, finally leading to a realistic, temporally coherent sequence of prescription behaviors.

Section 6 An Illustrative Example

This section aims to illustrate how the generative models discussed in Section 5 are adopted to construct a customized data generator for simulating synthetic Beneficiary Summary (BS) and Prescription Drug Event (PDE) files using the chain-type generation framework proposed in Subsection 5.1. In contrast to the synthetic data generated by pretrained LLMs described in Section 4, the customized generator presented here is designed to capture and reproduce the statistical properties and dependencies inherent in the original datasets. As a result, the synthetic data produced through this framework are not only structurally faithful to the CMS DE-SynPUF schema but also statistically representative of real-world patterns, making them suitable for applications such as model development, predictive validation, and actuarial or risk analytics.

It is noteworthy that like most simulation-based approaches, synthetic data generation is inherently stochastic. The objective of the proposed framework is not to produce a single "best" synthetic dataset, but to learn an underlying probability distribution that captures the statistical structure of the input data. Individual synthetic datasets represent Monte Carlo realizations from this learned distribution and may therefore differ from one another. Evaluation is accordingly conducted at the distributional and aggregate levels, rather than at the level of individual realizations, which is consistent with standard practice in actuarial simulation and stochastic modeling.

6.1 MODELING AND GENERATING THE BENEFICIARY SUMMARY FILE

The first step of the proposed chain-type generation framework focuses on modeling the variables contained in the BS file and subsequently simulating a synthetic population of beneficiaries to form the initial simulation pool. Recall that the BS file contains a heterogeneous collection of variables describing each beneficiary's background information including:

- Demographics: age, gender, race, and geographic indicators.
- Enrollment details: monthly indicators of Medicare Part A, Part B, and Part D coverage.
- Chronic conditions: the indicator for each specific illness.

These variables vary in type. Namely, some are discrete categorical variables (e.g., sex, race, chronic condition flags, and state codes), while others are continuous numerical variables (e.g., age and coverage months). Moreover, dependencies may exist among variables—for instance, certain chronic conditions often co-occur, and their prevalence may vary systematically with demographic attributes or insurance coverage duration. Capturing the joint relationships among all variables is nevertheless essential for producing realistic synthetic data that preserve the structural and dependency patterns of the original dataset.

It is important to note that reimbursement-related variables are intentionally excluded from the first stage, as they are modeled and generated in the subsequent stage. Including these variables prematurely would transform the second stage into a constrained simulation problem, requiring the generated reimbursement amounts to match previously simulated totals. Such an approach would be computationally demanding, unnecessary, and less natural to interpret.

The Tabular Variational Autoencoder (TVAE) and Conditional Tabular Generative Adversarial Network (CTGAN) methods are used, which have been implemented in the open-source Python package Synthetic Data Vault (SDV)[10] to build a generator for the BS file. Both models are extensions of the VAE and GAN architectures discussed in Subsections 5.2 and 5.3 respectively, and are specifically adapted for handling tabular data that contain a mixture of

[10] Source: https://docs.sdv.dev/sdv.

categorical and numerical variables. To be more specific, the standard VAE and GAN architectures are primarily designed for continuous data, and they struggle when applied to tabular data, which often include discrete categorical features and imbalanced category frequencies. To address these limitations, the TVAE generalizes the standard VAE by introducing data-type-specific transformations prior to encoding. These transformations include one-hot encodings for categorical columns and variable-specific normalization for numerical columns. The decoder is then trained to jointly reconstruct the original mixed representations. Once a TVAE is trained, the decoder can sample from the latent space and generate new, statistically consistent synthetic records.

Similarly, the CTGAN extends the standard GAN framework to handle mixed-type tabular data through two key modifications. First, it introduces a conditional generation mechanism, allowing the generator to produce synthetic samples conditioned on specific column values. This conditional sampling can help ensure that underrepresented categories are appropriately modeled and reproduced, thus enhancing the robustness of the generator. Second, CTGAN employs a mode-specific normalization strategy by representing each continuous variable as a mixture of Gaussian distributions rather than a single standard Gaussian. This modification enables the generator to more accurately capture the multimodal and irregular shapes of real-world continuous variable distributions.

The quality of the synthetic BS data generated by the TVAE and CTGAN models is assessed across four diagnostic dimensions defined in the SDV evaluation, all can be implemented using the SDV package:

- Data validation: Measures the proportion of synthetic records that conform to the schema and logical constraints inferred from the original dataset. A record is valid if all its fields have appropriate types, formats, and allowable values (e.g., coverage months within 0–12, valid sex, and race codes). This is evaluated automatically by the SDV package. Specifically, SDV first infers a metadata schema from the real dataset, which captures information such as data types, value ranges, categorical levels, date formats, and logical relationships. Then the package checks whether each synthetic record satisfies these same constraints. The final score represents the percentage of synthetic records that pass all schema-level validations.
- Data structure: Examines whether the synthetic dataset preserves the structural integrity of the original data, including column presence, order, and data types. SDV evaluates this by comparing the inferred metadata schema of the real dataset with that of the synthetic dataset, ensuring that all expected columns are present, correctly typed (e.g., categorical, numerical, or datetime), and arranged in the same order. A dataset achieves a perfect 100% data structure score when there are no missing or extra columns, and every field matches the schema-defined data type and format. Compared with data validation, which assesses record-level conformity of individual field values, the data structure metric evaluates dataset-level consistency.
- Column shapes: Evaluates how well the univariate marginal distributions of individual variables in the synthetic data match those in the real data. This metric captures the model's ability to reproduce realistic frequency patterns for both categorical and numerical variables. For numerical columns, SDV applies statistical distance measures such as the Kolmogorov–Smirnov statistic, which quantifies the maximum difference between the cumulative distribution functions of the real and synthetic data. For categorical columns, SDV assesses the relative frequency distributions, e.g., via total variation distance or Jensen–Shannon divergence. Each column's score is normalized between zero and one, and the overall column shapes score is computed as the average similarity across all columns. A higher value indicates that the model has more accurately reproduced the marginal distributions of the original dataset, while a lower value suggests distortions or smoothing in individual variable distributions.
- Column pair trends: Assesses the preservation of bivariate relationships between variables, such as correlations among numerical fields or co-occurrence patterns among categorical attributes (e.g., relationships between age and chronic conditions). Admittedly, this approach captures only pairwise dependence and does not fully account for higher-order interactions involving three or more variables

simultaneously. Nonetheless, it provides an informative and computationally efficient proxy for assessing relational fidelity, as preserving key pairwise relationships generally implies that the synthetic dataset maintains the broader multivariate structure of the original data.

Table 2 presents the diagnostic evaluation results for the synthetic BS data generated by the TVAE and CTGAN models. Both models achieved perfect scores in data validity and data structure, which indicates that every synthetic record conforms fully to the inferred schema and that the overall dataset layout, column types, and ordering are perfectly aligned with the original data.

In terms of column shapes and column pair trends, the TVAE achieved slightly higher scores (93.93% and 89.07%) than the CTGAN (91.53% and 86.53%), indicating somewhat better preservation of both marginal distributions and pairwise dependencies. Interestingly, this superior performance of the TVAE is somewhat unexpected, since VAEs rely on an approximate likelihood-based formulation, which often underfits fine-grained data features compared to the adversarial optimization used in GANs. This result indicates that the TVAE's latent-variable framework was particularly effective at modeling the structured dependencies in the BS data. Nevertheless, the performance differences between the two models are modest, and both exhibit strong fidelity in replicating the statistical structure of the original BS data.

Table 2
SUMMARY OF QUALITY EVALUATION METRICS FOR THE SYNTHETIC BS DATA GENERATED BY THE TVAE AND CTGAN MODELS

	Data Validity	Data Structure	Column Shapes	Column Pair Trends
TVAE	100.0%	100.0%	93.93%	89.07%
CTGAN	100.0%	100.0%	91.53%	86.53%

6.2 MODELING AND GENERATING THE PRESCRIPTION DRUG EVENT FILE

Next, the PDE file is modeled using a Transformer-based generative framework described in Subsection 5.4. Suppose that there are N beneficiaries, let the dataset be denoted by:

$$D = \{(b_i, P_i)\}_{i=1}^{N},$$

where

- $b_i \in B$ represents the feature vector for beneficiary i, and
- $P_i = \{p_{i,1}, p_{i,2}, \dots, p_{i,T_i}\}$ denotes the prescription sequence of length T_i, where each $p_{i,j}$ is the j-th drug event for beneficiary i.

Each prescription record $p_{i,j}$ consists of the following variables:

- $d_{i,j} \in \{1,2, \dots, D\}$ denotes the drug ID number;
- $q_{i,j} \in \{1,2, \dots, Q\}$ denotes the drug quantity dispensed;
- $s_{i,j} \in \{1,2, \dots, S\}$ denotes the days supply;
- $pay_{i,j} > 0$ denotes the patient payment amount;
- $cost_{i,j} > 0$ denotes the total cost of prescription;
- $g_{i,j} > 0$ denotes the gap days since the previous prescription event, created during the data pre-processing step.

The objective is to use a Transformer to model the conditional distribution $p_\theta(P_i|b_i)$, and thereby generate realistic prescription sequences conditioned on beneficiary attributes.

Transformer-based sequence models operate on discrete tokens and require input variables to be expressed in a categorical form. The application of Transformers to the modeling of prescription sequences requires discretizing all continuous-valued variables such as patient payment, total drug cost, and gap days. This is achieved via quantile-based binning using the empirical cumulative distribution function. Specifically, for a given continuous variable x, define a set of K bin boundaries according to:

$$b_k = \hat{F}^{-1}\left(\frac{k}{K}\right), k = 0,1,\dots,K,$$

where $\hat{F}^{-1}$ denotes the empirical quantile function estimated from the training data. Then, each observed value $x$is mapped to a discrete token through the following binning function:

$$\psi(x) = \arg\max_k\{k: x \le b_{k+1}\}.$$

It is noteworthy that the choice of the desired number of discrete bins K reflects a trade-off between modeling accuracy and computational complexity. A larger K leads to finer discretization, which better preserves the variability and resolution of the original continuous variable. However, this comes at the cost of increased vocabulary size for the Transformer model, which can complicate learning and increase training time. Correspondingly, the tokenized sequence for each beneficiary i is constructed by concatenating their demographic and clinical features with their longitudinal prescription events:

$$s_i = \left[BOS, b_i^{tok}, p_{i,1}^{tok}, p_{i,2}^{tok}, \dots, p_{i,T_i}^{tok}, EOS\right], \ \ i = 1,\dots,N.$$

Here,

- BOS and EOS are special tokens denoting the beginning and end of a sequence, respectively;
- b_i^{tok} is the discretized, tokenized representation of the beneficiary's features, including age, sex, race, and chronic condition flags;
- $p_{i,j}^{tok}$ represents the tokenized form of the j-th prescription event for beneficiary i, structured as

$$p_{i,j}^{tok} = \left[PRES, DRUG, d_{i,j}, QTY, q_{i,j}, DAYS, s_{i,j}, PAY, \psi\left(pay_{i,j}\right), COST, \psi\left(cost_{i,j}\right), GAP, \psi\left(g_{i,j}\right)\right].$$

In the sequence construction, explicit variable markers (e.g., DRUG, QTY, PAY) are introduced to precede each field within a prescription record. These markers serve a critical role by informing the model what type of variable is being generated at each step. This approach can be viewed as a deterministic, schema-aware analogue to traditional positional embeddings, which encode where a token appears in the sequence but remain agnostic to its semantic role. In contrast, positional embeddings encode relative location within a sequence, and they do not carry explicit semantic information about the type or role of each token. As a result, the model struggles to distinguish between structurally similar tokens that appear at different locations across sequences of varying lengths. By anchoring generation to variable-specific markers, this approach provides clearer inductive bias and improves the model's ability to learn structured output patterns. Empirically, this structured schema-aware tokenization strategy improves training stability and generation fidelity.

To train a Transformer to the tokenized data, the following configuration is adopted:

- Embedding dimension[11]: $d = 256$;
- Number of layers: $L = 6$;
- Number of attention heads: $H = 8$;
- FFN dimension: $d_f = 1025$;
- Maximum sequence length: $T_{max} = 1024$.

These hyperparameters were selected to strike a practical balance between modeling accuracy and computational feasibility. In particular, based on unreported preliminary analysis, the chosen embedding dimension, number of layers, and attention heads are sufficient to capture the key temporal and cross-variable dependencies in prescription sequences, while keeping the overall model size manageable so that training can be completed within hours using the computational resources available to the research team.

After the Transformer is built, given beneficiary features b, prescription events can be generated autoregressively via the algorithm described in Algorithm 1.

[11] Before inputting tokens into a Transformer, each discrete token is mapped to a continuous vector via an embedding lookup. The embedding dimension is the number of components in that vector. The values in this vector are learned, meaning that the model figures out how similar or different tokens should be to help with accurate prediction. A larger embedding dimension can capture more complex relationships and offer richer representations but also increases computational cost and risk of overfitting. However, a smaller embedding dimension is more efficient but may lack representational capacity for complex data.

Algorithm

SUMMARY OF THE AUTOREGRESSIVE GENERATIVE ALGORITHM OF THE TRANSFORMER ARTICHETURE

Algorithm 1 Prescription Sequence Generation

———————————————————————————————

1: **Input**: Beneficiary features b, model p_θ, temperature τ

2: **Output**: Prescription sequence P

3: $s \leftarrow [BOS, b^{tok}, SEP]$

4: cumulative_days $\leftarrow 0$

5: while cumulative_days < 365 and $|s| < T_{max}$ do

6: $s \leftarrow s \cup [PRES]$

7: **for** var in [DRUG, QTY, DAYS, PAY, COST, GAP] **do**

8: $s \leftarrow s \cup [var]$

9: $h \leftarrow Model(s)$

10: $p \leftarrow softmax((h-1)/\tau)$

11: $v \leftarrow sample(p)$ with top-k/top-p filtering

12: $s \leftarrow s \cup [v]$

13: **if** var = GAP **then**

14: cumulative_days ←cumulative_days + decode(v)

15: **end if**

16: **end for**

17: **end while**

18: $s \leftarrow s \cup [EOS]$

For ease of reading, the following table summarizes the symbols and terms used in Algorithm 1.

Table 3
SUMMARY AND DEFINITIONS OF SYMBOLS AND TERMS USED IN ALGORITHM 1

Term	Explanation
b, b^{tok}	Vector of beneficiary-level demographic and clinical features (e.g., age, sex, chronic condition indicators) and its tokenized version
p_θ	Trained Transformer model with parameters θ
τ	Temperature parameter used in softmax sampling to control randomness in generation
P	Generated prescription sequence for a beneficiary, consisting of a sequence of prescription events
s	Current token sequence constructed autoregressively during generation
BOS	Beginning-of-sequence token
SEP	Separator token distinguishing beneficiary-level tokens from prescription-level tokens
$PRES$	Marker token indicating the start of a new prescription event
EOS	End-of-sequence token
cumulative_days	Running total of elapsed days implied by the generated prescription gaps, used to enforce a one-year observation window
T_{max}	Maximum allowable token sequence length to prevent unbounded generation
[DRUG,QTY,DAYS,PAY,COST,GAP]	Ordered set of variable-specific markers indicating which prescription attribute is being generated: drug identifier, quantity dispensed, days supplied, patient payment amount, total prescription cost, number of days since the previous prescription event
$h \leftarrow Model(s)$	Model output logits for the next token, conditional on the current token sequence s
$p \leftarrow softmax((h-1)/\tau)$	Probability distribution over the token vocabulary after temperature scaling
v	Token sampled from p using top-k and/or top-p filtering, which is a sampling strategy that restricts candidate tokens to the most probable ones in order to reduce unlikely or unstable generations
decode(v)	Mapping from a discrete token v back to its original numerical value

To evaluate the quality of the trained Transformer in capturing the statistical properties underlying prescription dynamics, its ability to generate realistic synthetic prescription sequences conditional on real beneficiary features is assessed. Specifically, for each real beneficiary profile, the trained model is used to simulate a corresponding prescription sequence and compute summary-level quantities of interest. These include the total number of

prescription events, total drug costs, and the cumulative gap days between consecutive prescriptions. Then the mean and standard deviation of these summary statistics between the real and synthetic datasets are compared.

The rationale for this evaluation is twofold. First, these summary-level quantities are not directly modeled or predicted by the Transformer; rather, they emerge from the sequence of simulated granular events such as drug type, quantity, supply duration, and inter-event gap time. Therefore, accurate reproduction of these aggregate measures serves as a strong indicator that the model has successfully learned the fine-grained temporal and distributional patterns governing prescription behavior. Second, the ability to preserve both first-order (i.e., mean) and second-order (i.e., standard deviation) statistics suggests that the synthetic data retains not only central tendencies but also variability reflective of the real-world data.

Table 4 presents a comparison of key utility metrics between real and synthetic data. Across all metrics, the synthetic data demonstrates strong fidelity to the real data. The average number of prescriptions and total cost in the synthetic sample are slightly lower than those of the real data, while the standard deviation of prescription counts is slightly higher, suggesting a modest increase in variability among synthetic patients. The metrics for gap days are particularly well aligned, with both mean and standard deviation closely matching those in the real data. This is expected because gap days are directly generated as part of each prescription event by the Transformer model, making them explicitly modeled at the token level. In contrast, the number of prescriptions per beneficiary and the total cost are not explicitly modeled; rather, they emerge as aggregates across a sequence of individually generated events. As such, small discrepancies in token-level predictions (e.g., drug quantity, days supplied, or per-event cost) can accumulate over the full sequence, leading to slightly larger differences in these aggregate statistics. Nonetheless, these deviations remain modest, suggesting that the Transformer has effectively learned the joint structure needed to produce realistic prescription trajectories.

Table 4

COMPARISON OF KEY PER-BENEFICIARY SUMMARY STATISTICS BETWEEN REAL AND SYNTHETIC DATA TO ASSESS UTILITY PERFORMANCE

Metric	Real Data	Synthetic Data
Avg. Number of Prescriptions	54.0	50.4
Std. Number of Prescriptions	14.1	18.5
Avg. Total Cost	57.0	52.1
Std. Total Cost	87.9	78.9
Avg. Number of Gap Data	6.6	6.4
Std. Number of Gap Data	6.3	6.6

Finally, a downstream application is considered to further illustrate the high fidelity of the synthetic data. This serves two complementary purposes. On one hand, synthetic data are often created with the intent to support downstream analytical or operational tasks, so evaluating their performance in such settings is a natural and practical benchmark. On the other hand, while global statistical similarity measures assess the overall quality of synthetic data, their alignment with downstream tasks can serve as a more targeted, task-specific validation. Importantly, discrepancies in global quality may not necessarily translate into downstream performance degradation, or conversely, minor mismatches could be amplified, depending on the sensitivity and nature of the downstream application. Therefore, the example reported here should be interpreted as a local validation of utility, rather than a general guarantee across all possible applications.

The downstream application considered here is the use of generalized linear models (GLMs) to identify and quantify the significance of risk drivers for two key outcomes: the per-beneficiary number of prescription events and the total prescription costs. This task reflects a common actuarial and health analytics use case.

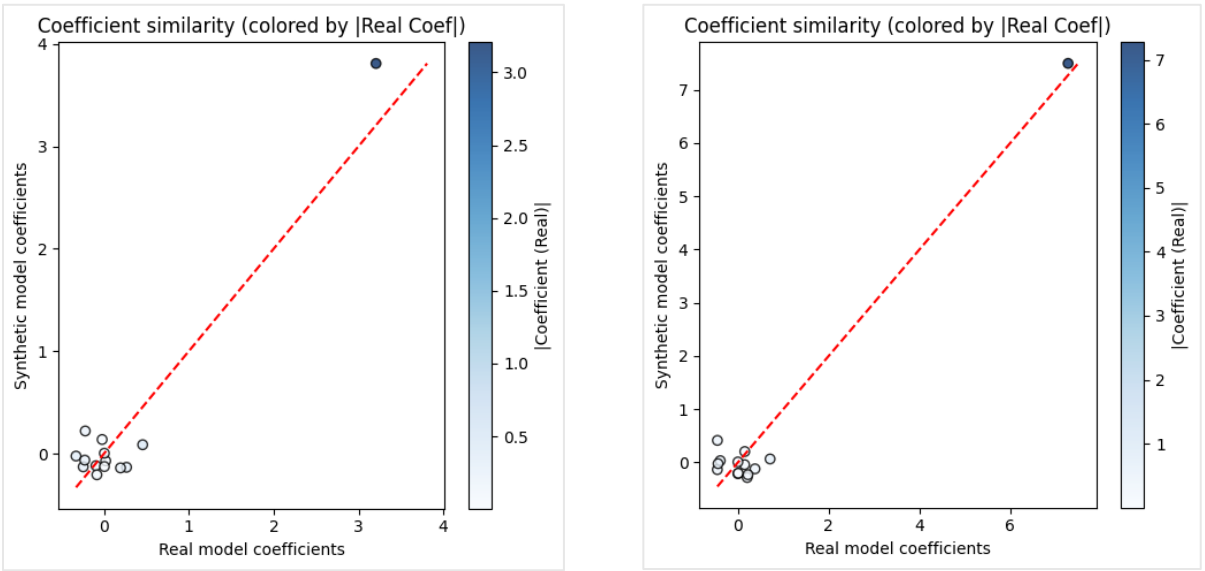
Specifically, demographic and clinical features, such as age, sex, and chronic condition indicators, were used as predictors, drawn from the beneficiary file. The corresponding response variables were derived by aggregating the prescription file at the beneficiary level, which yields each individual's total number of prescriptions and total drug costs over a one-year period. To model the count of prescriptions, a Poisson regression is applied with a log link, which is well-suited for non-negative count data. For total drug cost, a Gamma regression is applied with a log link, appropriate for modeling continuous, positive, and right-skewed cost outcomes.

Figure 6 presents a side-by-side comparison of model coefficients estimated from real and synthetic data for the count outcome and cost outcome. In both panels, a red dashed 45-degree line indicates perfect alignment between the two models. Points closely clustered along this line suggest strong agreement in the direction and magnitude of estimated effects across the real and synthetic datasets. For both outcomes, the coefficients derived from synthetic data closely mirror those from the real data. Minor deviations for smaller coefficients are expected and generally tolerable, particularly given that such effects are statistically weaker and more sensitive to sampling noise.

The evaluation checks conducted throughout this study provide compelling and multi-faceted evidence that the synthetic data preserves both global distributional properties and local relational structures critical to downstream analytical tasks. This suggests that synthetic data not only replicate overarching statistical trends but also faithfully encode the subtle feature-outcome relationships necessary for risk modeling and actuarial decision-making. Together, these findings affirm the high utility of Transformer-based synthetic data generation in actuarial and health analytics workflows.

Figure 6

COEFFICIENT ALIGNMENT BETWEEN REAL DATA MODELS AND SYNTHETIC DATA MODELS FOR THE TOTAL NUMBER OF PRESCRIPTIONS (LEFT PANEL) AND THE TOTAL ANNUAL PRESCRIPTION DRUG COST (RIGHT PANEL)



Note: Each point represents an estimated coefficient from the Poisson regression trained on the real and synthetic datasets. Points are colored based on the absolute magnitude of the coefficient estimated from the real data.

Section 7 Privacy Considerations

As discussed in the introduction, the development of synthetic data, whether medical claims data or other forms of insurance data, is often motivated by the need to reduce privacy risks associated with sharing sensitive individual-level information. That said, privacy protection is not automatically guaranteed in synthetic data generation. Typically, some degree of data fidelity or model flexibility inevitably needs to be sacrificed to ensure privacy in practice, underscoring the reality that there is an inherent trade-off between utility and privacy. Moreover, as generative modeling techniques become increasingly powerful, the potential privacy risks associated with synthetic data can become more pronounced if not carefully managed. For this reason, this section is devoted to helping readers develop a high-level understanding of the key considerations involved in identifying, assessing, and mitigating inherent privacy risks when using synthetic data in practice.

7.1 THE ORIGINS OF PRIVACY RISKS IN SYNTHETIC DATA

Privacy concerns in synthetic data generation primarily arise from the risk that information about individuals in the original training dataset may be inadvertently revealed through the synthetic output. Such risks typically stem from overfitting, where a generative model memorizes specific training records rather than learning the underlying data-generating distribution. When overfitting occurs, synthetic records may closely resemble, or in extreme cases replicate, real individuals, thereby increasing the risk of re-identification or attribute inference.

These risks are particularly salient in medical claims data, which often contain high-dimensional feature spaces, rare disease indicators, and unusual utilization patterns. Individuals with rare conditions or distinctive combinations of attributes are inherently more identifiable, and generative models trained on limited or unbalanced samples may disproportionately encode such patterns. For example, a dataset may contain a small number of unusually large families covered under the same insurance plan, which exhibit distinctive utilization patterns and correlated claims across family members. A generative model trained on such data may reproduce similarly structured family-level patterns in the synthetic output. This makes these large-family cases indirectly recognizable to individuals familiar with the original population. Importantly, because the proposed framework is designed to replicate the statistical properties of the input data, any privacy vulnerabilities present in the original dataset, such as small cell counts or extreme outliers, can propagate into the synthetic data if not explicitly addressed.

The aforementioned challenges are further compounded by the highly complex and black-box nature of modern generative models. The lack of interpretability makes it difficult to directly verify whether a trained generator has overfitted specific individuals or to precisely quantify the resulting privacy risks. As a result, assessing and validating privacy protection in synthetic data remains a challenging problem, and the methodological frameworks for doing so are still under active development. At present, the literature is far from providing a mature or comprehensive understanding of how overfitting, model complexity, and privacy risk interact in high-dimensional generative settings.

7.2 A SIMPLE VERIFICATION METHOD

A commonly adopted, intuitive approach to assessing privacy risk is to verify that no synthetic record is an exact copy of any training record. In practice, this may involve comparing synthetic and real records field by field and confirming that no complete matches exist. For tabular data, this check is straightforward to implement and provides a basic sanity check against direct memorization.

While such verification is useful as an initial screening step, the authors emphasize that the absence of exact duplicates is a necessary but not sufficient condition on its own for ensuring privacy safety. A primary limitation of simple duplication checks is that privacy risk can arise from variables that are continuous rather than discrete (e.g., age or cost). As a result, even when no exact copies exist, synthetic records may still lie very close to real records in the feature space, which can facilitate membership inference or attribute inference attacks.

In the illustrative example in Section 6, this type of verification is implemented by explicitly checking for exact matches between the real and synthetic records. No identical records were found, which indicates that the trained generator did not directly replicate any individual record from the training data. Moreover, because the prescription sequences were generated using discrete tokenization, this matching check can be interpreted more broadly as a range-based similarity test, where two records that are identical at the token level necessarily fall within the same discretized bins for all variables. As a result, the absence of exact token-level matches also implies that no synthetic record lies within the same discretized range across all fields as any real record. This provides a slightly stronger safeguard than an exact-value comparison for continuous data.

7.3 MORE ADVANCED PRIVACY FRAMEWORKS

In the related literature, more rigorous privacy measurement frameworks have been developed. At a high level, such formal approaches for assessing privacy risk can be classified into the following categories:

- Distance-based metrics: These methods quantify the similarity between synthetic and real records by applying statistical or embedding-based distance measures. One common example is the nearest-neighbor distance metric, in which, for each synthetic record, the distance (e.g., Euclidean, Mahalanobis, or Gower distance) to its closest real record is computed. Extremely small minimum distances may indicate elevated disclosure risk.
- Information-theoretic metrics: These approaches measure how much information about the original data can be inferred from the synthetic output. A typical example is the use of mutual information or related entropy-based measures to quantify the dependence between real and synthetic datasets. Higher inferred information content suggests greater potential for privacy leakage.
- Differential privacy: The framework ensures that the output of a data analysis or model training procedure is nearly indistinguishable whether or not any single individual's data is included in the input dataset. This guarantee is formalized through a privacy parameter, which bounds the influence that the presence or absence of one individual can have on the model's behavior. Smaller values of the privacy parameter correspond to stronger privacy protection but typically require greater modification during the training process.

For readers seeking a more rigorous treatment of these privacy measurement frameworks, readers are directed to the following representative references: for distance- or similarity-based assessments of memorization and data-copying in generative models, see Meehan, Chaudhuri, and Dasgupta (2020):[12] for information-theoretic perspectives on privacy-preserving disclosure, see Rassouli, Rosas, and Gündüz (2019);[13] and for differential privacy, see Dwork and Roth (2014).[14]

To enhance privacy protection in synthetic data generation, several practical strategies can be adopted within the proposed framework:

- Data preprocessing: Suppressing or aggregating rare categories and truncating extreme values that pose elevated re-identification risk, particularly for small subpopulations or outlier behaviors.

---

[12] . Meehan, K. Chaudhuri, and S. Dasgupta, "A Non-Parametric Test to Detect Data-Copying in Generative Models," *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)* (2020), https://arxiv.org/abs/2004.05675.

[13] B. Rassouli, F. E. Rosas, and D. Gündüz, "Data Disclosure under Perfect Sample Privacy," *IEEE Transactions on Information Forensics and Security* (2019), https://doi.org/10.1109/TIFS.2019.2954652.

[14] C. Dwork and A. Roth, "The Algorithmic Foundations of Differential Privacy," *Foundations and Trends in Theoretical Computer Science* 9, nos. 3–4 (2014): 211–407, https://doi.org/10.1561/0400000042.

- Controlled model capacity: Limiting the complexity of the generative model so that it is expressive enough to learn population-level patterns, while avoiding excessive capacity that may lead to memorization of individual trajectories.
- Early stopping: Monitoring training performance and terminating model training before overfitting occurs, thereby reducing the likelihood that the model memorizes specific training records.
- Noise injection: Introducing controlled randomness during model training or data generation (e.g., adding noise to gradients, latent representations, or outputs) to weaken the direct influence of any single training record on the synthetic data.

It is very important for the reader to understand that a fundamental and unavoidable trade-off exists between statistical fidelity and privacy protection. Models that closely replicate fine-grained patterns in the data, particularly rare events or extreme behaviors, tend to offer higher analytical utility but also pose greater privacy risks. Conversely, stronger privacy constraints often require smoothing, noise addition, or suppression of detailed structure, which can reduce the usefulness of the synthetic data for downstream modeling and decision-making.

The discussion in this section does not seek to eliminate this trade-off, if it is even possible to do so, but rather to make it explicit. The appropriate balance between statistical fidelity and privacy protection depends on the application context. For example, exploratory analysis, education, and software testing may tolerate lower fidelity in exchange for stronger privacy safeguards, whereas model development and validation may prioritize statistical realism while relying on governance controls to manage residual privacy risk.

Section 8 Conclusion and Future Research

This report demonstrates an actuarial application of AI methods for modeling and generating synthetic medical claim data. A flexible, chain-based framework is proposed, which enables the identification and customization of generative models tailored to diverse data types, ensuring both usability and performance across varying application needs.

For scenarios where preserving the exact statistical properties of the original data is not critical, synthetic data can be generated using pretrained LLMs to maximize privacy and accessibility. Conversely, in applications where statistical fidelity is essential such as in downstream modeling or risk factor analysis, this report presents model choices that are structurally aligned with the data and capable of capturing complex relationships. These modeling strategies are illustrated using the SynPUF BS and PDE files to showcase how synthetic data can be constructed with meaningful realism. This report also outlines a suite of validation methods to assess the utility and fidelity of the synthetic data. Empirical results suggest that the generated datasets maintain a satisfactory degree of statistical integrity.

Synthetic data generation techniques offer a transformative path for overcoming long-standing challenges related to privacy and data access. By enabling the safe sharing of realistic, privacy-preserving datasets, this approach opens new avenues for collaboration across organizational boundaries, including between research teams, industry stakeholders, and academia. Practitioners are cautioned to always check if any generated data recreated any patient from the training data. This report represents a meaningful step toward responsible data sharing, accelerating methodological advances, and fostering impactful applications in underwriting, health policy design, and beyond.

The analytics-focused applications outlined in Section 2.2 also point to several important directions for future research. While synthetic data have shown strong potential for supporting predictive modeling, actuarial simulation, and model governance, further work is needed to better understand the conditions under which synthetic data can reliably preserve the statistical and temporal structures required for these tasks. Ongoing research is actively examining issues such as robustness of downstream analytics to Monte Carlo variability, stability across repeated synthetic data generations, and the propagation of bias from training data into analytic outcomes. In addition, continued development of evaluation frameworks, governance practices, and privacy–utility trade-offs will be essential for expanding the responsible use of synthetic data in analytics-focused actuarial applications.

This report establishes a practical and operational solution for developing a synthetic medical claims data generator, which represents a fundamental first stepping stone for continued research in this area. Building on this foundation, several important directions merit further exploration. One future research direction is to more systematically investigate how the developed synthetic data can be used to realize the analytics-focused applications outlined in Section 2.2, including predictive modeling, actuarial simulation, and product design, and to better understand the conditions under which synthetic data can reliably support these downstream tasks. A second direction is to develop more comprehensive and rigorous frameworks for characterizing the trade-off between statistical utility and privacy protection, together with clearer governance mechanisms for assessing and managing privacy risk in practice. Finally, the proposed framework can be further strengthened for group health settings, where patients from the same family or similar occupational backgrounds may exhibit shared dependencies and correlated utilization patterns. In such cases, a fully Transformer-based or hierarchy-aware generative approach may offer a promising avenue for explicitly modeling group-level dependence structures while maintaining flexibility and scalability.

Synthetic data generated using the techniques described here are likely not appropriate for informing clinical guidance.

Section 9 Supplemental Sample Prompts

This section presents the sample prompts developed to instruct the pretrained LLM in generating structurally faithful synthetic datasets, including the BS and PDE files, which together replicate the schema and relational organization of the CMS DE-SynPUF data.

9.1 PROMPTS FOR BENEFICIARY SUMMARY FILE

You are a data generator tasked with creating realistic synthetic Beneficiary Summary data in CSV format, similar in structure to the CMS DE-SynPUF Beneficiary file.

Requirements and context:

- Structure & Columns:

Include 32 columns exactly. The first line of the output must be the header row (column names). Follow it with N rows of data (where N is the number of synthetic beneficiaries requested). The columns are:

DESYNPUF_ID, BENE_BIRTH_DT, BENE_DEATH_DT, BENE_SEX_IDENT_CD, BENE_RACE_CD, BENE_ESRD_IND, SP_STATE_CODE, BENE_COUNTY_CD, BENE_HI_CVRAGE_TOT_MONS, BENE_SMI_CVRAGE_TOT_MONS, BENE_HMO_CVRAGE_TOT_MONS, PLAN_CVRG_MOS_NUM, SP_ALZHDMTA, SP_CHF, SP_CHRNKIDN, SP_CNCR, SP_COPD, SP_DEPRESSN, SP_DIABETES, SP_ISCHMCHT, SP_OSTEOPRS, SP_RA_OA, SP_STRKETIA, MEDREIMB_IP, BENRES_IP, PPPYMT_IP, MEDREIMB_OP, BENRES_OP, PPPYMT_OP, MEDREIMB_CAR, BENRES_CAR, PPPYMT_CAR

(Ensure this header is output exactly as above, with identical spelling and order.)

Variable definitions and generation rules:

- DESYNPUF_ID

Unique synthetic beneficiary identifier. Use a 16-character alphanumeric string. Some IDs may appear in other CMS-like synthetic files to allow linkage.

- BENE_BIRTH_DT

Beneficiary date of birth in YYYYMMDD format. Must be realistic (e.g., beneficiaries mostly born 1900–1965).

- BENE_DEATH_DT

Beneficiary date of death in YYYYMMDD format, or blank if still alive. Must be $\geqslant$ BENE_BIRTH_DT. For realism, only a subset of records should include death dates.

- BENE_SEX_IDENT_CD

Beneficiary sex. Codes: 1 = Male, 2 = Female.

- BENE_RACE_CD

Beneficiary race/ethnicity. Codes: 1 = White, 2 = Black, 3 = Other, 5 = Hispanic.

- BENE_ESRD_IND

End-stage renal disease indicator. Codes: 0 = No ESRD, Y = Yes.

- SP_STATE_CODE

State of residence. Use two-digit SSA state codes (01–54, excluding gaps such as 40, 48).

- BENE_COUNTY_CD

SSA county code within the state. Numeric string (e.g., "001", "045"). Should be consistent with the state.

- BENE_HI_CVRAGE_TOT_MONS

Total months of Part A coverage. Integer 0–12.

- BENE_SMI_CVRAGE_TOT_MONS

Total months of Part B coverage. Integer 0–12.

- BENE_HMO_CVRAGE_TOT_MONS

Total months of HMO coverage. Integer 0–12.

- PLAN_CVRG_MOS_NUM

Total months of Part D plan coverage. Integer 0–12.

- SP_ALZHDMTA, SP_CHF, SP_CHRNKIDN, SP_CNCR, SP_COPD, SP_DEPRESSN, SP_DIABETES, SP_ISCHMCHT, SP_OSTEOPRS, SP_RA_OA, SP_STRKETIA

Chronic condition indicators. Binary codes: 1 = Yes, 2 = No.

- MEDREIMB_IP, BENRES_IP, PPPYMT_IP

Annual inpatient Medicare reimbursement, beneficiary responsibility, and primary payer reimbursement. Dollar amounts with 2 decimals (e.g., 1234.56).

- MEDREIMB_OP, BENRES_OP, PPPYMT_OP

Annual outpatient reimbursement fields. Dollar amounts with 2 decimals.

- MEDREIMB_CAR, BENRES_CAR, PPPYMT_CAR

Annual carrier (physician/supplier) reimbursement fields. Dollar amounts with 2 decimals.

Consistency checks:

- Completeness:
- No missing values except that BENE_DEATH_DT may be blank for alive beneficiaries.
- All other fields must be populated.

- Logical consistency:
- BENE_DEATH_DT ⩾ BENE_BIRTH_DT (if present).
- Coverage months (BENE_HI_CVRAGE_TOT_MONS, etc.) must be integers 0–12.
- Chronic condition indicators must be coded as 1 or 2.
- Reimbursement amounts (MEDREIMB_*, BENRES_*, PPPYMT_*) must be ⩾ 0.
- Range realism:
- Birth years should produce plausible Medicare ages (mostly 65+ during 2008–2010).
- Death dates should fall within a realistic observation window.
- Dollar amounts should reflect plausible healthcare costs (e.g., thousands to tens of thousands annually).

Output format:

- Provide the output as CSV text only: first the single header line, and then N data lines following it.
- Separate fields with commas.
- Do not include explanations, bullet points, or formatting outside of the CSV itself.
- Ensure the CSV is properly structured and ready for use.

9.2 PROMPTS FOR PRESCRIPTION DRUG EVENT FILE

You are a data generator tasked with creating **realistic synthetic Prescription Drug Events (PDE) data** in CSV format.

Requirements and context:

- **Structure & Columns:**

Include **8 columns exactly**. The first line of the output must be the header row (column names). Follow it with `N` rows of data (where `N` is the number of synthetic claims requested). The columns are:

```

DESYNPUF_ID, PDE_ID, SRVC_DT, PROD_SRVC_ID, QTY_DSPNSD_NUM, DAYS_SUPLY_NUM, PTNT_PAY_AMT, TOT_RX_CST_AMT

```

- (Ensure this header is output **exactly as above**, with identical spelling and order.)

Variable definitions and generation rules:

- **DESYNPUF_ID**

A synthetic patient identifier. Use a **16-character alphanumeric string** (e.g., “001AB23CD45E678F”). Ensure that some IDs repeat across different rows to reflect that a single beneficiary may have multiple prescription events. This ID carries no inherent patient information and serves only as a linking key.

- **PDE_ID** (CCW Part D Event Number)

A unique identifier for each synthetic drug event. Use a random **12–15 character alphanumeric string**. No inherent meaning is embedded, but each PDE_ID must be unique within the dataset.

- **SRVC_DT** (RX Service Date)

The date the prescription was filled. Use **YYYYMMDD** format. Dates should fall within a plausible multi-year period (e.g., 2008–2010) and respect chronological realism (e.g., no future dates).

- **PROD_SRVC_ID** (Product Service ID)

A synthetic **11-digit numeric code** resembling a National Drug Code (NDC11). These should vary across rows and can repeat (as multiple patients may receive the same drug).

- **QTY_DSPNSD_NUM** (Quantity Dispensed)

The quantity dispensed, expressed as an integer number of units (e.g., 10, 30, 90). Must be positive. For realism, typical values are in the range 1–200.

- **DAYS_SUPLY_NUM** (Days Supply)

The number of days the dispensed medication is intended to last. Integer values typically between 1 and 365. Often correlated with QTY_DSPNSD_NUM (e.g., 30 tablets → ~30 day supply).

- **PTNT_PAY_AMT** (Patient Pay Amount)

The dollar amount paid by the patient that is **not reimbursed by third parties** (copayments, coinsurance, deductibles). Use a decimal with 2 digits (e.g., 5.00, 20.75). Must be $\geqslant 0$.

- **TOT_RX_CST_AMT** (Gross Drug Cost)

The total gross cost of the drug (sum of ingredient cost, dispensing fee, taxes, etc.). Use a decimal with 2 digits. Must be **greater than or equal to PTNT_PAY_AMT**.

Missing values:

- Allow occasional missing values ("") in **QTY_DSPNSD_NUM, DAYS_SUPLY_NUM, PTNT_PAY_AMT**, and **TOT_RX_CST_AMT** to reflect real-world data imperfections.

- Do **not** generate missing values for **DESYNPUF_ID, PDE_ID, SRVC_DT, PROD_SRVC_ID** (these are required for every record).

Consistency checks:

- **Uniqueness:**

  - Each PDE_ID must be unique.

  - Rows should not be exact duplicates.

- **Numeric constraints:**

  - QTY_DSPNSD_NUM and DAYS_SUPLY_NUM must be positive integers.

  - PTNT_PAY_AMT ⩾ 0.00.

  - TOT_RX_CST_AMT ⩾ PTNT_PAY_AMT.

- **Logical consistency:**

  - Higher QTY_DSPNSD_NUM generally corresponds to higher DAYS_SUPLY_NUM.

  - Cost amounts should scale plausibly with quantity.

Output format:

- Provide the output as **CSV text only**: first the single header line, and then `N` data lines following it.

- Separate fields with commas.

- Do **not** include explanations, bullet points, or formatting outside of the CSV itself.

- Ensure the CSV is properly structured and ready for use.

Section 10 Acknowledgments

The authors' deepest gratitude goes to those without whose efforts this project could not have come to fruition: the volunteers who generously shared their wisdom, insights, advice, guidance, and arm's-length review of this study prior to publication. Any opinions expressed may not reflect their opinions nor those of their employers. Any errors belong to the authors alone.

Project Oversight Group members:

Priya Dwarakanath, FSA, FIA, MAAA

Greg Fann, FSA, MAAA

Ka Wah Michael Fung, FSA, CERA

Xinyi Hu, ASA, MAAA

Sarah Lopez

Shisheng Qian, FSA, CERA

Norm Storwick, FSA, MAAA

At the Society of Actuaries Research Institute:

Joe Alaimo, ASA, ACIA

Korrel Crawford, Sr. Research Administrator

About The Society of Actuaries Research Institute

Serving as the research arm of the Society of Actuaries (SOA), the SOA Research Institute provides objective, data-driven research bringing together tried and true practices and future-focused approaches to address societal challenges and your business needs. The Institute provides trusted knowledge, extensive experience and new technologies to help effectively identify, predict and manage risks.

Representing the thousands of actuaries who help conduct critical research, the SOA Research Institute provides clarity and solutions on risks and societal challenges. The Institute connects actuaries, academics, employers, the insurance industry, regulators, research partners, foundations and research institutions, sponsors and non-governmental organizations, building an effective network which provides support, knowledge and expertise regarding the management of risk to benefit the industry and the public.

Managed by experienced actuaries and research experts from a broad range of industries, the SOA Research Institute creates, funds, develops and distributes research to elevate actuaries as leaders in measuring and managing risk. These efforts include studies, essay collections, webcasts, research papers, survey reports, and original research on topics impacting society.

Harnessing its peer-reviewed research, leading-edge technologies, new data tools and innovative practices, the Institute seeks to understand the underlying causes of risk and the possible outcomes. The Institute develops objective research spanning a variety of topics with its strategic research programs: aging and retirement; actuarial innovation and technology; mortality and longevity; diversity, equity and inclusion; health care cost trends; and catastrophe and climate risk. The Institute has a large volume of topical research available, including an expanding collection of international and market-specific research, experience studies, models and timely research.

Society of Actuaries Research Institute
8770 W Bryn Mawr Ave, Suite 1000
Chicago, IL 60631
www.SOA.org