Paper PP12

Creating Synthetic Data that Mimics Real World Data for Accurate Analysis Programming Development

Linshan Yuan, Johnson and Johnson, Spring House, USA
Weiwei Xu, Johnson and Johnson, Spring House, USA

ABSTRACT

Development of analysis programs in clinical trials often requires realistic test data before final study data becomes available. This paper presents effective methods for generating synthetic Anti-drug Antibody (ADA) data that preserves complex internal logical relationships. We developed a comprehensive approach utilizing the Synthetic Data Vault (SDV) framework combined with custom constraint implementations to generate realistic ADA datasets that maintain statistical properties and logical dependencies inherent in real-world clinical trial data. Our methodology demonstrates how synthetic data can significantly accelerate Tables, Figures, and Listings (TFL) generation while ensuring minimal code modifications when transitioning to actual study data. The approach is validated across multiple datasets, showing consistent preservation of complex relationships including conditional dependencies, monotonic constraints, and derived variable relationships.

INTRODUCTION

The development of analysis programs for clinical trials presents unique challenges that require data that accurately reflects the complex relationships, constraints, and statistical properties inherent in real study outcomes. This requirement stems from the need to validate analysis programs not just for technical correctness, but for their ability to handle the nuanced data patterns that emerge from clinical trial operations.

Anti-drug Antibody (ADA) assessments represent one of the more complex domains in clinical trial data management. ADA data involves multiple correlated test outcomes spanning both numeric and character results, with intricate dependencies between variables that must be preserved to ensure meaningful analysis validation. The SDTM IS (Immunogenicity Specimen) domain captures these assessments, but the relationships between classification results, titer values, neutralizing antibody status, and subject characteristics create a web of dependencies that simple random data generation cannot replicate.

The challenge is compounded by the timing of analysis program development. These programs must be written and validated before the actual study data becomes available, creating a chicken-and-egg problem where the validation of analysis logic depends on data that cannot be generated using the final analysis programs themselves. Traditional approaches have relied on either simplified dummy data that fails to exercise edge cases, or on historical data from previous studies that may not reflect the current study's design characteristics.

This work presents a systematic approach to generating synthetic ADA data that addresses these challenges. Our specific objectives include:

1. Preservation of Internal Logical Relationships: Develop methods to maintain the complex dependencies between ADA outcome on each sample, the corresponding titer values and neutralizing antibody status that exist in real clinical trial data.
2. Adaptability to Study Design: Create a framework that can be customized to align with specific study designs, dosing regimens, and investigated drug characteristics.
3. Validation of Analysis Programs: Demonstrate how synthetic data enables comprehensive testing and validation of analysis programs before final data availability.
4. Efficiency in Program Development: Quantify the acceleration in analysis program development achieved through the use of realistic synthetic data.

This paper aims to generate synthetic data for ADA assessments within the SDTM IS domain framework. We focus specifically on the relationships between ADA conclusion (positive/negative), subject peak titer values (SPTITER), subject ADA status (SUADAST), and neutralizing antibody results (NAB) and subject NAB status (SUNABST). The methodology is demonstrated through implementation using the Synthetic Data Vault (SDV) library in Python, with custom constraint classes that enforce domain-specific business rules.

METHODS AND APPROACH

Our approach to synthetic ADA data generation is built upon three fundamental principles: foundation in real-world data characteristics, preservation of internal logical relationships, and adaptability to specific study requirements. Rather than

generating data from scratch, we utilize real-world ADA data as the foundation, analyzing and extracting the statistical properties and logical patterns that characterize genuine clinical trial outcomes.
The framework begins with the extraction of relationship patterns from source ADA data. This involves identifying the statistical distributions of each variable, the correlations between variables, and the conditional dependencies that govern their relationships. For example, in ADA data, we observe that the relationship between ADA conclusion and titer values is not simply correlated but conditionally dependent: titer values only exist when ADA conclusion is positive, and their distribution varies based on the specific ADA outcome.

Once these patterns are extracted, the framework provides mechanisms to adapt them to specific study designs. This includes modifying the overall distributions to match expected enrollment figures, adjusting relationships to reflect the pharmacokinetic and pharmacodynamic characteristics of the investigated drug, and customizing output formats to match the specific analysis scheme requirements of the study.

DATA STRUCTURE AND VARIABLE DEFINITIONS

The synthetic ADA data model is structured around a set of core variables that capture the essential characteristics of ADA assessments.

```
Original dataframe shape: (10, 8)
================================================================================
     USUBJID  ADACNCLS  ADATTR  SPTITER   SUADAST       NAB   SUNABST  seq
0  dummy_002  Negative     NaN      NaN       NaN       NaN       NaN    1
1  dummy_002  Positive    20.0     20.0  Positive  Negative  Negative    2
2  dummy_002  Positive    10.0     20.0  Positive  Positive  Positive    3
3  dummy_002  Negative     NaN     20.0  Positive       NaN       NaN    4
4  dummy_002  Negative     NaN     20.0  Positive       NaN       NaN    5
5  dummy_002  Negative     NaN     20.0  Positive       NaN       NaN    6
6  dummy_002  Negative     NaN     20.0  Positive       NaN       NaN    7
7  dummy_002  Negative     NaN     20.0  Positive       NaN       NaN    8
8  dummy_002  Negative     NaN     20.0  Positive       NaN       NaN    9
9  dummy_002  Negative     NaN     20.0  Positive       NaN       NaN   10
```

The primary variables include:

USUBJID (Subject ID): A unique identifier for each subject in the dataset. In our implementation, this serves as a constant value across rows for single-subject demonstrations, but in full implementation would vary per subject.

ADACNCLS (ADA Conclusion): The primary variable indicating whether a subject tested positive or negative for anti-drug antibodies (ADA). This is a categorical variable with values typically limited to ‘Negative’ and ‘Positive’, though our framework supports extension to more granular classification schemes.

ADATTR (ADA Titer): The numeric titer value represents the strength of the ADA response. Values are typically drawn from a discrete set (10.0, 20.0, 40.0, 80.0 in our implementation as example) representing dilution factors, with special handling for missing values that indicate negative ADA outcome.

SPTITER (Subject Peak Titer): A derived variable that tracks the progression of ADA responses across study visits. These variable exhibits forward-fill behavior, maintaining the most recent non-missing titer value until a larger value is observed, reflecting the biological reality that ADA responses tend to persist and intensify rather than spontaneously decreasing.

SUADAST (Subject ADA Status): A status variable derived from SPTITER that indicates whether the subject currently has a detectable ADA response. This is positive when SPTITER is non-missing and negative when SPTITER is missing.

NAB (Neutralizing Antibody): A categorical variable indicating whether the ADA response includes neutralizing antibodies. This variable has a conditional dependency on ADACNCLS, being non-missing only when classification is positive.

SUNABST (Subject NAB Status): A status variable that mirrors NAB exactly, including missing value patterns. This ensures consistency between the underlying measurement and the subject status interpretation.

SEQ (Sequence): A sequential row number ensuring proper ordering of records within each subject’s data. The count aligns with the total number of samples collected for that subject in the study.

CONSTRAINT DEVELOPMENT

The integrity of synthetic ADA data depends on the accurate implementation of business rules that govern the relationships between variables. We implemented a comprehensive constraint framework using the SDV library's constraint template, adding custom constraint classes for domain-specific rules.

The most fundamental constraints in ADA data are conditional dependencies where the state of one variable determines the permissible values of another. We implemented two primary conditional constraint patterns:

Positive Conditional Constraint: When ADACNCLS equals 'Positive', ADATTR must be non-missing. This constraint enforces the logical requirement that positive ADA classifications must be supported by actual titer measurements. Without this constraint, synthetic data generators might produce positive classifications with missing titer values, which would never occur in actual clinical data.

Positive/Negative Conditional Constraint: When ADACNCLS equals 'Negative', ADATTR must be missing (NaN). This complementary constraint ensures that negative classifications are consistently represented with missing titer values, reflecting the clinical practice of not reporting quantitative results for negative screens.

Complex clinical data often exhibits chained dependencies where the state of one variable affects another, which in turn affects a third. We implemented chained conditional constraints to handle these scenarios:

ADACNCLS → ADATTR → SPTITER Chain: When ADATTR is non-missing, SPTITER must also be non-missing. This chain reflects the real-world relationship where titer measurements (ADATTR) determine the subject iteration status (SPTITER), which in turn affects downstream status variables.

SPTITER stays blank (NaN indicates blank) until the first non-missing ADATTR value appears at the row where ADATTR first becomes non-missing, SPTITER becomes non-missing with the same value SPTITER only updates when ADATTR has a value larger than current SPTITER When ADATTR is NaN or smaller, SPTITER maintains its last known value.

Example behavior:

ADATTR: [nan, 20.0, 10.0, nan, nan, nan, nan, nan, nan, nan]
SPTITER: [nan, 20.0, 20.0, 20.0, 20.0, 20.0, 20.0, 20.0, 20.0, 20.0]

Some relationships in ADA data are deterministic rather than probabilistic. We implemented fixed value constraints for these cases:

SUADAST Determination: When SPTITER is non-missing, SUADAST should be 'Positive'. When SPTITER is missing, SUADAST must be 'Negative'. This deterministic relationship ensures that subject status is always correctly derived from the underlying titer measurements.

NAB Default on Positive Outcome: When ADACNCLS is 'Positive', NAB is further to conduct to test for neutralizing anti-drug antibody, the NAB result could be either 'Positive' or 'Negative'. This ensures that positive ADA classifications always have a defined neutralizing antibody status, even when explicit measurement is unavailable.

Constraint Application Order

The order of constraint application is critical for achieving consistent results. We apply constraints in the following sequence:

1	Positive Conditional Constraints	ADACNCLS, ADATTR	Fills ADATTR values for Positive rows of ADACNCLS.
2	Chain Condition Constraints	ADATTR, SPTITER	Forward-fill and only updates when ADATTR has a value larger than current SPTITER.
3	Fixed Values Constraint	SPTITER, SUADAST	When SPTITER has non-missing values, SUADAST has Positive value only.
4	Negative Conditional Constraints	ADACNCLS, ADATTR	ADATTR is blank for Negative rows of ADACNCLS.
5	Positive Conditional Constraints	ADACNCLS, NAB	NAB has either Positive or Negative values and should not be blank for Positive rows of ADACNCLS.
6	Fixed Values Constraint	ADATTR	Value allowed: 10.0, 20.0, 40.0, 80.0
7	Equal Values Constraint	NAB, SUNABST	Ensures SUNABST equals NAB.

This reverse-dependency order ensures that downstream constraints are applied after upstream dependencies are established.

With above original tabulation given, synthesized tabulation data can be generated with any shape (different number of records) with above ADA rules carried:

Synthesized tabulation 1: 8 records

Synthetic dataframe shape: (8, 8)

	USUBJID	ADACNCLS	ADATTR	SPTITER	SUADAST	NAB	SUNABST	seq
0	dummy_002	Negative	NaN	NaN	Negative	NaN	NaN	1
1	dummy_002	Negative	NaN	NaN	Negative	NaN	NaN	2
2	dummy_002	Negative	NaN	NaN	Negative	NaN	NaN	3
3	dummy_002	Negative	NaN	NaN	Negative	NaN	NaN	4
4	dummy_002	Negative	NaN	NaN	Negative	NaN	NaN	5
5	dummy_002	Positive	10.0	10.0	Positive	Negative	Negative	6
6	dummy_002	Positive	20.0	20.0	Positive	Positive	Positive	7
7	dummy_002	Positive	20.0	20.0	Positive	Negative	Negative	8

Synthesized tabulation 2: 9 records

Synthetic dataframe shape: (9, 8)

	USUBJID	ADACNCLS	ADATTR	SPTITER	SUADAST	NAB	SUNABST	seq
0	dummy_002	Negative	NaN	NaN	Negative	NaN	NaN	1
1	dummy_002	Positive	10.0	10.0	Positive	Negative	Negative	2
2	dummy_002	Negative	NaN	10.0	Positive	NaN	NaN	3
3	dummy_002	Negative	NaN	10.0	Positive	NaN	NaN	4
4	dummy_002	Negative	NaN	10.0	Positive	NaN	NaN	5
5	dummy_002	Negative	NaN	10.0	Positive	NaN	NaN	6
6	dummy_002	Positive	10.0	10.0	Positive	Negative	Negative	7
7	dummy_002	Negative	NaN	10.0	Positive	NaN	NaN	8
8	dummy_002	Negative	NaN	10.0	Positive	NaN	NaN	9

Synthesized tabulation 3: 17 records

Synthetic dataframe shape: (17, 8)

	USUBJID	ADACNCLS	ADATTR	SPTITER	SUADAST	NAB	SUNABST	seq
0	dummy_002	Positive	10.0	10.0	Positive	Negative	Negative	1
1	dummy_002	Negative	NaN	10.0	Positive	NaN	NaN	2
2	dummy_002	Negative	NaN	10.0	Positive	NaN	NaN	3
3	dummy_002	Positive	20.0	20.0	Positive	Negative	Negative	4
4	dummy_002	Negative	NaN	20.0	Positive	NaN	NaN	5
5	dummy_002	Positive	10.0	20.0	Positive	Negative	Negative	6
6	dummy_002	Positive	10.0	20.0	Positive	Negative	Negative	7
7	dummy_002	Negative	NaN	20.0	Positive	NaN	NaN	8
8	dummy_002	Negative	NaN	20.0	Positive	NaN	NaN	9
9	dummy_002	Positive	10.0	20.0	Positive	Positive	Positive	10
10	dummy_002	Negative	NaN	20.0	Positive	NaN	NaN	11
11	dummy_002	Negative	NaN	20.0	Positive	NaN	NaN	12
12	dummy_002	Negative	NaN	20.0	Positive	NaN	NaN	13
13	dummy_002	Negative	NaN	20.0	Positive	NaN	NaN	14
14	dummy_002	Negative	NaN	20.0	Positive	NaN	NaN	15
15	dummy_002	Negative	NaN	20.0	Positive	NaN	NaN	16
16	dummy_002	Positive	10.0	20.0	Positive	Negative	Negative	17

One original tabulation data could create any number of tabulation with different shape and multiple unique original ADA tabulations could synthesize even more different tabulation data and provide the most various dummy ADA profiles.

DISCUSSION

The implementation of synthetic ADA data generation provides several significant benefits for clinical trial analysis programming:

Accelerated Development Timeline: By providing realistic test data before final study data availability, analysis programs can be developed, tested, and validated in advance. This shifts the critical path for analysis delivery earlier in the study timeline, reducing the time pressure that often accompanies database lock scenarios.

Comprehensive Edge Case Coverage: Synthetic data generation allows deliberate exploration of edge cases that might be rare or absent in actual study data. Programs can be tested against scenarios like all-positive populations, extreme effect sizes, or unusual variance patterns that might not occur in the actual study but could occur in future studies.

Reduced Rework During Transition: Analysis programs developed with synthetic data that accurately reflect real data patterns require minimal modification when actual study data becomes available. This contrasts with programs developed with simplified test data, which often require substantial restructuring to accommodate actual data patterns.

Safe Sharing for Programming Development: Synthetic data can be safely shared with programming teams, contract research organizations, and other stakeholders without the privacy concerns associated with actual subject data. This facilitates collaboration while maintaining compliance with data protection requirements.

CONCLUSION

This work demonstrates an effective methodology for generating realistic synthetic ADA data that preserves the complex logical relationships inherent in clinical trial immunogenicity assessments. By combining the statistical synthesis capabilities of SDV with custom constraint implementations, we created a framework that produces data suitable for rigorous analysis program validation.

A comprehensive constraint framework for ADA data that enforces all critical business rules including conditional dependencies, forward-fill behaviors, and deterministic derivations. Validation results confirming 100% constraint satisfaction across all generated datasets.

The adoption of synthetic data generation approaches like the one presented here can significantly improve the efficiency and reliability of clinical trial analysis programming, contributing to higher quality deliverables and more timely availability of study results.

REFERENCES

1. PHUSE Annual Conference 2026
2. Synthetic Data Vault (SDV) Documentation
3. CDISC SDTM Implementation Guide
4. CDISC ADaM Implementation Guide

CONTACT INFORMATION

Linshan Yuan
Johnson & Johnson Innovative Medicine
LYuan7@ITS.JNJ.com

Weiwei Xu
Johnson & Johnson Innovative Medicine
wxu52@ITS.JNJ.com