HLG MOS Synthetic Data Challenge

Report from the GOFCoE Team
(now renamed Smart Data Foundry)

Version 2.1
24–28 January 2022

Contents

1 GOFCoE Team **2**

2 Ten Steps to Synthesis **2**

3 Synthesis Methods **3**
3.1 Miró (Stochastic Solutions) . . . 3
3.2 GEMINAI (Diveplane) . . . 4
3.3 Synthetic Data Vault (SDV) from MIT . . . 4
3.4 Synthpop (University of Edinburgh) . . . 4

4 Synthesis Evaluation **5**
4.1 Utility Metrics . . . 5
4.1.1 Utility Metrics provided by GEMINAI (Diveplane): . . . 5
4.1.2 Utility Metrics provided by Synthpop . . . 7
4.2 Disclosure Risk (Privacy) Metrics . . . 8
4.2.1 Disclosure Risk Metrics provided by GEMINAI (Diveplane) . . . 8
4.2.2 Desirability for Privacy Metrics . . . 9
4.2.3 Disclosure Risk Metrics provided by Synthpop . . . 10

5 SAT-GPA dataset **10**
5.1 Metadata . . . 10
5.2 Data preparation . . . 10
5.3 Data exploration . . . 11
5.4 Synthesis Results . . . 11
5.4.1 Utility Metrics Table . . . 11
5.4.2 Privacy Metrics Table . . . 11
5.4.3 Results of Miró Synthesis . . . 12
5.4.4 Results of GEMINAI Synthesis . . . 13
5.4.5 Results of SDV Syntheses . . . 15
5.4.6 Results of Synthpop . . . 19

6 ACS dataset 20
6.1 ACS Metadata . . . 21
6.2 Data preparation . . . 21
6.2.1 Routine Preprocessing . . . 21
6.2.2 Income (“THICK” approach) . . . 22
6.2.3 POVERTY (“THICK” approach) . . . 22
6.2.4 Data Exploration . . . 23
6.3 Synthesis Results . . . 23
6.3.1 Utility Metrics Table . . . 23
6.3.2 Privacy Metrics Table . . . 23
6.3.3 Results of Miró Synthesis . . . 24
6.3.4 Results of GEMINAI (Diveplane) Synthesis . . . 25
6.3.5 Results of SDV Synthesis . . . 26
6.3.6 Results of Synthpop . . . 28

7 Comments and Recommendations 30

1 GOFCoE Team

The Global Open Finance Centre of Excellence (GOFCoE—now renamed Smart Data Foundry[1]) has a mission is to unlock the power of financial data to improve people’s lives, something that can only be achieved if we make respect for people’s privacy a central principle.

We want to generate safe synthetic datasets from real data, but we are relatively new to using learning-based data synthesisers.

The team consisted of:

- **Paola Arce**, Data Scientist
- **Victor Alfonzo Diaz**, Data Scientist
- **Euan Gardner**, Synthetic Data Specialist
- **Nick Radcliffe**, Chief Data Scientist GOFCoE (and Founder/Director, Stochastic Solutions Limited, principal author of Miró, see below)

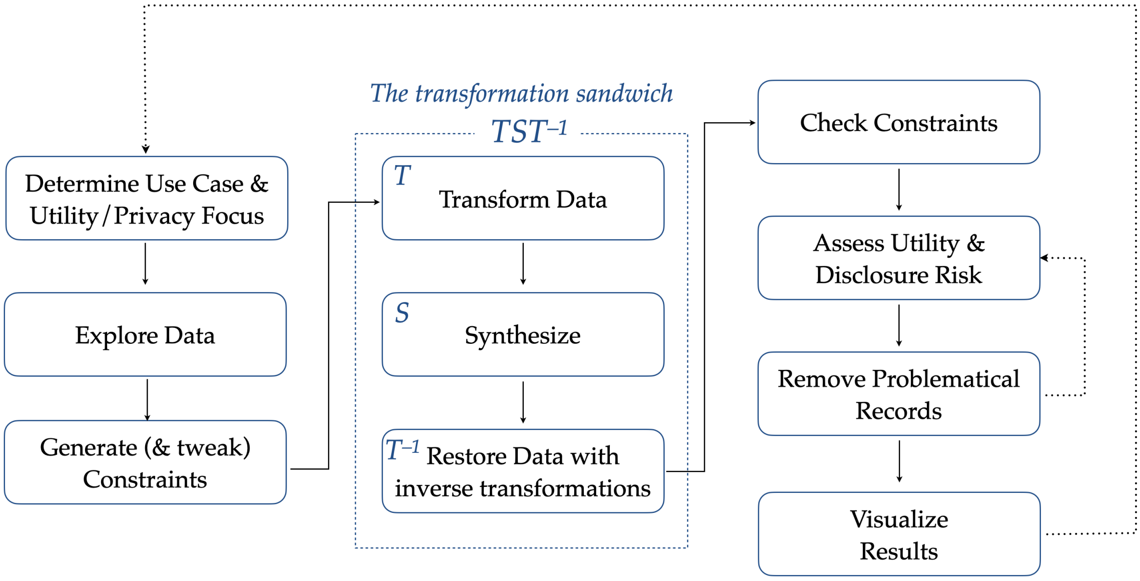
We have had excellent support from Diveplane, who produce GEMINAI, one of the systems we have used in the challenge.

2 Ten Steps to Synthesis

Our general approach is outlined in the diagram below, showing our 10-step process.

Notice particularly

[1] https://smartdatafoundry.com


The transformation sandwich
TST^{-1}
Determine Use Case & Utility/Privacy Focus
Explore Data
Generate (& tweak) Constraints
T Transform Data
S Synthesize
T^{-1} Restore Data with inverse transformations
Check Constraints
Assess Utility & Disclosure Risk
Remove Problematical Records
Visualize Results

Ten Steps to Synthesis

1. our use of (automatically generated) constraints from the real data, sometimes further tweaked manually, to characterize the data and the check some extra aspects of generated data

2. our frequent use of the "transformation sandwich", whereby we transform the data before synthesis and transform it back after synthesis. We often find this useful when there are strong dependencies between columns that cannot adequately be modelled or respected by synthesis methods

3 Synthesis Methods

As we mentioned previously we have tried different systems/methods to generate synthetic data. These are:

- Miró (Stochastic Solutions)
- GEMINAI (Diveplane)
- Synthetic Data Vault (MIT)
- Synthpop (Univeristy of Edinburgh)

A brief description of each system now follows.

3.1 Miró (Stochastic Solutions)

Miró is a less well-known commercial analytics package produced by Stochastic Solutions Limited[2] and licensed by GOFCoE. It is written in Python and makes heavy

[2] One of the team members, Nick Radcliffe, is a principal author of Miró, and founder director of Stochastic Solutions Limited

use of `numpy`. When Miró generates constraints (in a `.tdda` file) using its version of TDDA, it can also write out distribution information (based on binning each field appropriately). Miró then has the ability to generate data that is consistent (or, in a few cases, largely consistent) with both the constraints and distributions saved in the TDDA file. In most cases, fields are treated entirely independently, i.e. by default Miró makes *no attempt* to generate data that preserves any relationships between fields. It is mostly used to generate high-quality test data that respects many of the constraints of the original data, rather than rich synthetic data that preserves correlations and other inter-field relationships.

It is mostly included as a (very low) bar against which to compare other implementations, and as a very fast synthesis method. The nature of its synthesis also means that the data is knowably safe, in the sense that there cannot be (non-random) correlations between columns, so the only risks that should be possible in data generated in this way by Miró are risks of leaking particular sensitive values in isolation. These risks are typically easy to evaluate and to check for.

3.2 GEMINAI (Diveplane)

In this work we have used the GEMINAI [1] data synthesis software from Diveplane Inc. his is a commercial product, for which GOFCoE has recently purchased a (paid) licence. We also received support from Diveplane, helping us as we used the software in anger for the first time during this Challenge.

GEMINAI uses a proprietary synthesis method based on k-Nearest Neighbours and Differential Privacy, and provides a broad suite of assessment metrics covering both utility and disclosure risk, which we have used across the various datasets we have synthesized using all methods.

3.3 Synthetic Data Vault (SDV) from MIT

The Synthetic Data Vault is software developed by MIT. It is a suite of libraries for synthetic data generation that allows learn single-table, multi-table and time-series datasets to later on generate new synthetic data. It is set in the realm of *Deep Learning Synthesisers,* but also includes synthesis methods using copulas describing cumulative probability distributions.

SDV does also provide some assessment methods and mettrics, but we found them hard to use and not very useful.

3.4 Synthpop (University of Edinburgh)

The Synthpop [2] package for R, developed at Edinburgh University, allows to create synthetic versions of sensitive microdata. The package allows the synthesis process to be customised in many different ways according to the characteristics of the data being generated. There are default values for most of the parameters, but it is recommended to adjust parameters appropriately for the problem at hand to maximize the suitability the the synthetic data generated. It also includes utility and privacy metrics we are going to use in our methods evaluation.

4 Synthesis Evaluation

Several of the systems we have used provide Utility Metrics that can be applied to any synthetic data. We have used utility metrics provided by Diveplane (in their Data Quality Tool) and some by Synthpop.

4.1 Utility Metrics

4.1.1 Utility Metrics provided by GEMINAI (Diveplane):

All Diveplane's metrics are standardized to a 0–5 range, with 5 being the best, and are mostly described in terms of *desirability.* They are combined to provide overall metrics for privacy and utility using geometric means, as described below.

- **DescriptiveStatistics**: measures the desirability of the differences in basic statistics between the features of the original and generated data.
- **MMDStatistics**: measures the desirability of the joint distribution. If the `p-value` is greater than a significance value (e.g., 0.1), it is not possible to reject the null hypothesis (i.e., the distributions are similar.)
- **EnergyStatistics**: measures the desirability of joint distribution of continuous features. If the `p-value` is greater than a significance value, (e.g. 0.1) it's not possible to reject the null hypothesis (i.e. the distributions are similar.)
- **GTest**: measures the desirability of the marginal distribution for nominal features. If the `p-value` is greater than a significance value, (e.g. 0.1) it's not possible to reject the null hypothesis (i.e. the distributions are similar.)
- **MannWhitney**: measures the desirability of marginal distribution that measures central tendencies. If the `p-value` is greater than a significance value (e.g. 0.1), it is not possible to reject the null hypothesis (i.e. the distribution is similar).
- **ChiSquare**: measures the desirability of marginal distribution for nominal features. If the `p-value` is greater than a significance value, (e.g. 0.1) it's not possible to reject the null hypothesis (i.e. the distributions are similar.)
- **Kolmogorov-Smirnov**: measures the desirability of marginal distribution for continuous features. If the `p-value` is greater than a significance value (e.g., 0.1), it is not possible to reject the null hypothesis (i.e., the distributions are similar.)
- **RegressionComparison**: measures the desirability of a regression models using original and generated data and then compare performance.

  NOTE: By default, a single target (dependent variable) is chosen for regression, which defaults to the "last" (right-most) field in the dataset. But it can also be set. For the SAT-GPA data, this defaulted to `fy_gpa`, which is seems like a good choice. For ACS, we set it to `INCTOT` initially, but on reflection this was probably a poor choice (see below). For final evaluation we instead used `AGE`, which seems likely to have some relationship to most of the other variables, whereas `INCOME` can be modelled with quite high fidelity using only the components. Given that (as noted below), we give special treatment to those, this was not a good initial choice a modelling target.

GEMINAI's "Desirability" for Utility Metrics

Gemini defines an overall utility-specific **desirability** that summarizes the metrics given in Data Quality Tool of Diveplane, in order to understand the final score. They define *desirability* as how well is the accuracy of the generated data when compared to the original data. When there are fewer accuracy losses in the generated data, desirability goes up.

For different metrics the *desirability* is properly defined.

- **DescriptiveStatistics**: the measured desirability d is computed as follows:

  $$d = 5 - 5 \times \text{min}\left(1.0\,,\ \text{SMAPE}\left[\text{orig, gen}\right]\right)$$

  Here SMAPE is the *Symmetric Mean Absolute Percentage Error* between the original dataset A and the generated one B, and is define by

  $$\text{SMAPE}\left[\text{A, B}\right] = \text{min}\left(1.0\,,\frac{2}{n}\sum_{i=1}^{n}\frac{|B_i - A_i|}{|A_i| + |B_i| + \epsilon}\right)\,, \tag{1}$$

  where is assumed that the size n of both dataset is the same, and ϵ is a parameter to avoid division by zero setted by default to $1 \cdot 10^{-3}$. Desirability will be equal to 3 when there is a 40% difference between the original and generated data.

- For all marginal metrics **MMDStatistics**, **EnergyStatistics**, **ChiSquare**, **GTest**, and **KolmogorovSmirnov**

  we can write desirability d as the value of the stratified sigmoid function $\sigma'(p)$ as

  $$\sigma'(p) = \begin{cases} 0 + g(p) & ,\quad p < 0.001 \\ 1 + g(p - 0.001) & ,\ 0.001 \leq p < 0.001 \\ 2 + g(p - 0.01) & ,\ 0.01 \leq p < 0.001 \\ 3 + g(p - 0.005) & ,\ 0.005 \leq p < 0.01 \\ 4 + g(p - 0.1) & ,\ p \geq 0.1 \end{cases}\,,$$

  with p the `p-value` of the test, $\sigma(p)$ the sigmoid function and $g(p) \equiv \sigma\left(20p - 10\right)$. If there are insufficient categorical features to compute the metrics, $d = \epsilon$ setted by default to $1 \cdot 10^{-3}$.

- **MannWhitney**

  The desirability for this metric is given by

  $$d = 5 - 5 \times \text{SMAPE}\left[\text{MW}, \frac{n_{\text{orig}} \times n_{\text{gen}}}{2}\right]\,,$$

where MW is the score of the Mann-Whitney U-test. If there are insufficient continuous features to compute the metric, $d = \epsilon$ setted by default to $1 \cdot 10^{-3}$.

- **RegressionComparison**

  The desirability d is given by the geometric average of two different components:

  - The SMAPE between the original and generated ϵ^{R^2} scores, where R^2 is is the square of the coefficient of multiple correlation and $\epsilon = 0.5 \cdot 10^{-3}$.
  - The SMAPE between the original and generated $\frac{1}{\mathrm{RMSE} + \epsilon}$, with RMSE the root mean squared error of the regression and ϵ is equal to half the minimum non-zero gap or $1 \cdot 10^{-3}$ if such gap does not exits.

  Both components will range from 3-5 when the SMAPE $<= 0.2$ and 0-3 otherwise.

So Diveplane's **Overall Desirability** is a geometric average of the components, and like the components runs from 0 (terrible) to 5 (excellent). Note that, as a geometric mean, a single bad metric can have a large effect. Note also that many of the components are tests of univariate fidelity, rather than multivariate structure.

NOTE: In the case where both datasets (original and generated) differ in size a random selection will be made to match the sizes.

4.1.2 Utility Metrics provided by Synthpop

Synthpop provides three variations of the so-called Propensity Mean Square Error (pMSE).

- **pMSE**: Synthpop propensity mean squared error. More info here.
- **S_pMSE**: Synthpop pMSE standardized ratio. More info here.
- **PO50**: percentage over 50% of each synthetic data set where the model used correctly predicts whether real or synthetic. More info here.

The pMSE is the result of using a model (usually CART) to try to distinguish between real and synthetic records, with the ideal behaviour being that the tree fails completely, having the same proportions of synthetic and real records in every leaf node, indicating that it has completely failed to distinguish real and synthetic data. That would produce a pMSE of 0. As the model is more successful in splitting and synthetic records into separate nodes, the pMSE increases.

We consider this a very clever metric, because any way the model can separate the datasets, using any variable or combination of variables, leads to an increase in the pMSE value. The way CART operates is ideal for this purpose.

The possible downside is that it depends on the CART algorithm, which (inevitably) has some weaknesses. So it is possible that CART will fail to separate the two datasets when another method might do so, even easily. Nevertheless, we consider pMSE to be an extremely powerful, neutral and general purpose metric for assessing multivariate differences between datasets.

4.2 Disclosure Risk (Privacy) Metrics

Similarly, several of the systems we have used provide Disclosure Risk metrics that can be applied to any synthetic data. We have used privacy (disclosure risk) metrics provided by Diveplane (in their Data Quality Tool) and some by Synthpop.

4.2.1 Disclosure Risk Metrics provided by GEMINAI (Diveplane)

The metrics we have used (for all models) provided by from Diveplane are:

- **Distance Anonymity Preservation**: measures the desirability of the Distance Anonymity Preservation, which is computed by using the distances between the data points of a data set to the closest corresponding data point of another dataset to measure privacy, with a focus on the regions with the densest data. A value greater than 1.0 for minimum percentile is a strong indicator that privacy is being preserved even in the worst case, and any higher value means that privacy is even stronger. A value of less than 1.0 for minimum percentile indicates there might be a chance of privacy being compromised.

- **Density Anonymity Preservation**: measures the desirability of the Density Anonym-ity Preservation, which is computed by using the distances between the data points of a data set to the closest corresponding data point of another dataset to measure privacy, relative to the density.

- **k-anonymity**: measures the desirability of the computed k-anonymity, which happens when each equivalence class has at least k records to protect against identity disclosure.

- **KL-divergence**: measures the desirability of the Kullback–Leibler divergence. KL divergence values are a measure of how similar the distributions of the generated and original dataset are. KL-divergence values greater than 10^{-5} mean that the distribution of the original and the generated dataset are not overly similar, and is an indicator privacy is being preserved.

- **ℓ-Diversity**: measures the desirability of the L-Diversity, which happens when a distribution of a sensitive attribute in each equivalence class has at least ℓ "well represented" values to protect against attribute disclosure.

- **t-Closeness**: measures the desirability of t-closeness. An equivalence class is said to have t-closeness if the distance between the distribution of a sensitive attribute in this class and the distribution of the attribute in the whole table is no more than a threshold t. A table is said to have t-closeness if all equivalence classes have t-closeness.

The Diveplane *desirability* metric for privacy measures the overall privacy again on a zero (worst) to five (best) scale, again as a geometric average.

4.2.2 Desirability for Privacy Metrics

Similarly as before the desirability is defined for each of the metrics in privacy as

- **DistanceAnonymityPreservation**: The desirability d is the scale scores of geometric mean of min density ratios across all `p_norm` values.
- **DensityAnonymityPreservation**: The desirability d is the scale scores of geometric mean of min density ratios across all `p_norm` values.
- **AverageDensityAnonymityPreservation**: The desirability d is the scale scores of geometric mean of min density ratios across all `p_norm` values.
- **K-Anonymity**: The desirability is computed as ref-1

  $$d = \begin{cases} 5.0 & , \text{ all } k\text{-anonymity} > 1 \\ 3.0 - \left(\frac{\text{gen\_violations}}{n}\right) & , \text{ otherwise} \end{cases},$$

  where gen_violations is The number of violations in the generated data, and n is the total number of equivalence classes.
- **KLDivergence**: The desirability is compute as

  $$d = \log_{10}\left(0.001 \times \frac{\min D(K||L) + \epsilon}{0.0001}\right),$$

  where $D(K||L)$ is the the relative entropy between the discrete probability distribution of K and L, i.e. it is the expectation of the logarithmic difference between the probabilities K and L, where the expectation is taken using the probabilities K. Also, $\epsilon = 1 \cdot 10^{-9}$ which is only used to find d, and gen_violations is the number of violations of K-anonymity violation in the generated data. The ϵ mentioned in this paper used to avoid division by 0 while calculating KL Divergence is set to 0.001.
- **LDiversity**: The desirability is compute as this paper

  $$d = \begin{cases} 5.0 & , \text{ all } \ell\text{-diversity} > 1 \\ 3.0 - \left(\frac{\text{gen\_violations}}{n}\right) & , \text{ otherwise} \end{cases}.$$

  Here gen_violations is the number of violations of ℓ-diversity this paper in the generated data and n is the total number of equivalence classes.
- **TCloseness**:The desirability is compute as this paper

  $$d = \begin{cases} 5.0 & , \text{ all } t\text{-closeness} > 1 \\ 3.0 - \left(\frac{\text{gen\_violations}}{n}\right) & , \text{ otherwise} \end{cases},$$

  with gen_violations is the number of violations of KL-anonymity violation in the generated data and n is the total number of equivalence classes. We have removed t-**closeness** for ACS as it produces the same (awful) value of 0.001 for all methods, so has no discriminatory power. However, we need to understand the metric better to understand whether the poor value for all of our synthetic datasets is a genuine issue that should worry us, or a weakness or pathology in the measure that does not reflect a real problem.

So Diveplane's **Overall Desirability** is a geometric average of the components, and like the components runs from 0 (terrible) to 5 (excellent). Note that, as a geometric mean, a single bad metric can have a large effect. Note also that many of the components are tests of univariate fidelity, rather than multivariate structure.

NOTE: In the case where both datasets (original and generated) differ in size a random selection will be made to match the sizes.

4.2.3 Disclosure Risk Metrics provided by Synthpop

The metrics we have used (for all models) provided by Synthpop are:

- **replications**: number of duplicates in the synthetic data set(s). More info here.
- **uniques**: a number of unique individuals in the original data set. More info here.
- **per replications**: percentage of duplicates in the synthetic data set(s). More info here.

5 SAT-GPA dataset

This dataset contains SAT (United States Standardized university Admissions Test) and GPA (university Grade Point Average) data for 1000 students at an unnamed college. Educational Testing Service originally collected the data. It can be found here.

5.1 Metadata

Field	Description	Sensitive	Type	Range	To be synthetize	Confidential/Dependent
sex	Gender of the student	No	Categorical	1,2	Yes	No
sat_v	Verbal SAT percentile	Yes	Continuous	[0,100]	Yes	No
sat_m	Math SAT percentile	Yes	Continuous	[0,100]	Yes	No
sat_sum	Total of verbal and math SAT percentiles	Yes	Continuous	[0,200]	No	No
hs_gpa	High school grade point average	Yes	Continuous	[0,5]	Yes	No
fy_gpa	First year (college) grade point average	Yes	Continuous	[0,5]	Yes	Yes

5.2 Data preparation

The `sat_sum` column can be obtained adding up `sat_v` and `sat_m`, and the sum is consistent with the two components for every record in the real SAT data, so it was removed before applying the models.

The column was calculated after the rest synthetic data was produced.

5.3 Data exploration

As part of data exploration we used the Miró implementation of test-driven data analysis (TDDA) to generate constraints describing the input data. These constraints specify the names, types minimum and maximum values for each field, and can (though in this case didn't) also generate constraints on uniqueness of values in a field, whether missing values are allowed. For strings fields, it can also generate regular expression constraints or lists of allowed values, though again these were not relevant here.

We also looked generate a new field and generate constraints for that. (It should

```
sat_sum_less_v_less_m = sat_sum - sat_v - sat_m
```

always be zero.)

A white paper describing automatic constraint generation and verification of data using the constraints is available as Automatic Constraint Generation and Verification

5.4 Synthesis Results

The following tables give a summary of the performance of the methods using both Utility and Privacy Metrics provided by Diveplane Data Quality Tool and Synthpop package.

5.4.1 Utility Metrics Table

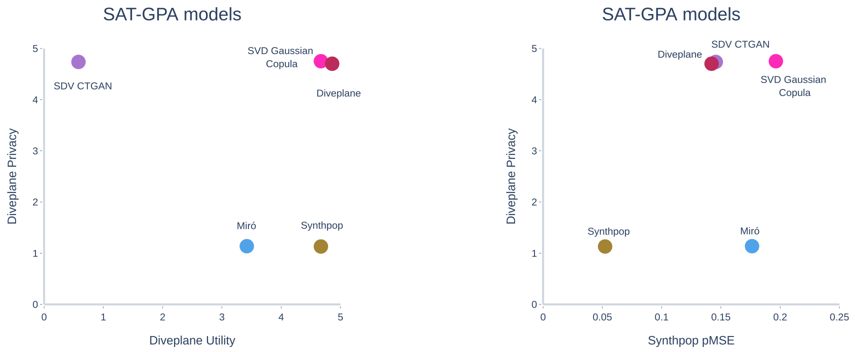
As mentioned before we have two sets of utility metrics

Metrics	Miró	GEMINAI	SDV CTGAN	SDV GCOPULA	Synthpop	Synthpop Smooth
Overall Desirability	3.37	4.831	0.796	4.616	4.69	4.796
DescriptiveStatistics	4.918	4.805	4.576	4.746	4.955	4.898
MMDStatistics	3.001	4.783	0.1	4.005	4.974	4.766
EnergyStatistics	1.001	4.294	1.001	4.931	4.001	4.769
GTest	5.0	5.0	0.101	4.007	4.005	4.201
MannWhitney	4.967	4.927	4.19	4.936	4.836	4.918
ChiSquare	5.0	5.0	4.745	5.0	5.0	5.0
KolmogorovSmirnov	4.619	5.0	0.101	4.998	5.0	5.0
RegressionComparison	1.964	4.88	1.73	4.45	4.905	4.866
pMSE	0.0523	0.067	0.176	0.196	0.1457	0.142
S_pMSE	1.662	1.896	4.780	5.375	3.868	3.968
P050	21.3	23.3	41.3	43.75	37.05	36.35

5.4.2 Privacy Metrics Table

NOTE: We are suspicious of some the individual metrics and in particular do not believe the KL-Divergence figure for Miró (which completely destroyed its otherwise good privacy metrics) identify a significant risk, though would want to investigate this further before any actual release.

Metrics	Miró	GEMINAI	SDV CTGAN	SDV GCOPULA	Synthpop	Synthpop Smooth
Overall Desirability	1.266	4.192	4.325	4.304	1.101	3.252
DistanceAnonymityPreservation	5.0	5.0	5.0	5.0	0.001	2.475
DensityAnonymityPreservation	3.426	3.408	3.623	3.699	1.774	1.487
AverageDensityAnonymityPreservation	4.411	4.196	4.538	4.295	3.367	3.059
KAnonymity	5.0	5.0	5.0	5.0	5.0	5.0
KlDivergence	0.001	4.596	5.0	5.0	4.728	4.959
LDiversity	5.0	5.0	5.0	5.0	5.0	5.0
TCloseness	2.766	2.769	2.754	2.757	2.773	2.757
`replications`	0	31	0	0	0	0
`uniques`	1000	1000	1000	1000	1000	1000
`per replications`	0	3.1	0	0	0	0



(a) SAT diveplane metrics

(b) SAT synthpop metrics

Figure 1: SAT-GPA metrics

As follow a brief description and comments on the observed performance results:

5.4.3 Results of Miró Synthesis

A graphical comparison between the real SAT data and that produced by Miró is available here:

- Profile of Miró Synthetic SAT Data vs. the Real Data

As can be clearly seen, the one-dimensional distributions are accurately preserved, and nearly constraints are respected (with the sole exception that two records are generated with a slightly out-of-range value for `sat_sum`, which we do not regard as a defect.

Also, as expected, the scatterplots comparing fields show no preservation of patterns between any of the pairs of fields, other than the `sat_sum`.

Diveplane's privacy metrics did down-weight the "privacy desirability" of the Miró-generated data quite heavily because two records were nearly identical to real records, but given the very small number of fields in the data, this is nor surprising, and given that we *know* the fields are generated independently and the records weren't outliers, we don't regard this as a material defect.

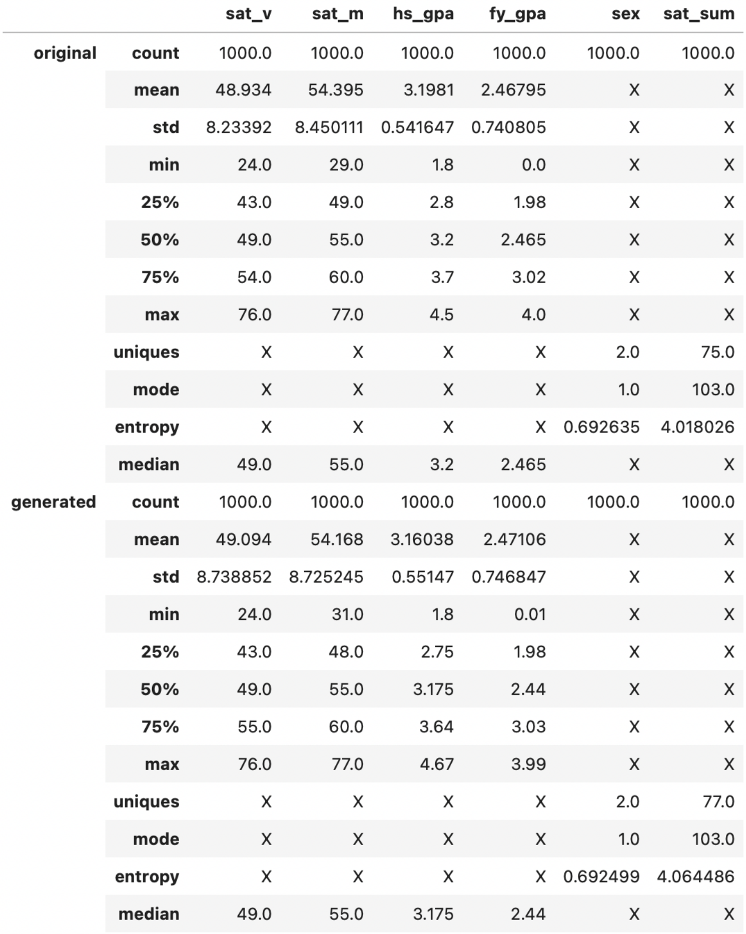
Miró's approach is clearly unsuitable for the three applications that require multi-column relationships to be preserved (testing our own specific analysis, teaching and release to the public) but is extremely well suited to testing technology, being computationally cheap, highly private and respecting the profiles of individual variables.

5.4.4 Results of GEMINAI Synthesis

In terms of the overall utility the synthesis done using Diveplane systems comes out well.

- Profile of GEMINAI Synthetic SAT Data vs. the Real Data

It preserves the descriptive statistical attributes of the original dataset as is shown in the summary of statistical attributes of both datasets:



		sat_v	sat_m	hs_gpa	fy_gpa	sex	sat_sum
original	count	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0
	mean	48.934	54.395	3.1981	2.46795	X	X
	std	8.23392	8.450111	0.541647	0.740805	X	X
	min	24.0	29.0	1.8	0.0	X	X
	25%	43.0	49.0	2.8	1.98	X	X
	50%	49.0	55.0	3.2	2.465	X	X
	75%	54.0	60.0	3.7	3.02	X	X
	max	76.0	77.0	4.5	4.0	X	X
	uniques	X	X	X	X	2.0	75.0
	mode	X	X	X	X	1.0	103.0
	entropy	X	X	X	X	0.692635	4.018026
	median	49.0	55.0	3.2	2.465	X	X
generated	count	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0
	mean	49.094	54.168	3.16038	2.47106	X	X
	std	8.738852	8.725245	0.55147	0.746847	X	X
	min	24.0	31.0	1.8	0.01	X	X
	25%	43.0	48.0	2.75	1.98	X	X
	50%	49.0	55.0	3.175	2.44	X	X
	75%	55.0	60.0	3.64	3.03	X	X
	max	76.0	77.0	4.67	3.99	X	X
	uniques	X	X	X	X	2.0	77.0
	mode	X	X	X	X	1.0	103.0
	entropy	X	X	X	X	0.692499	4.064486
	median	49.0	55.0	3.175	2.44	X	X

Figure 2: General descriptive statics comparison for synthesis with GEMINAI.

The one-dimensional distributions are well preserved

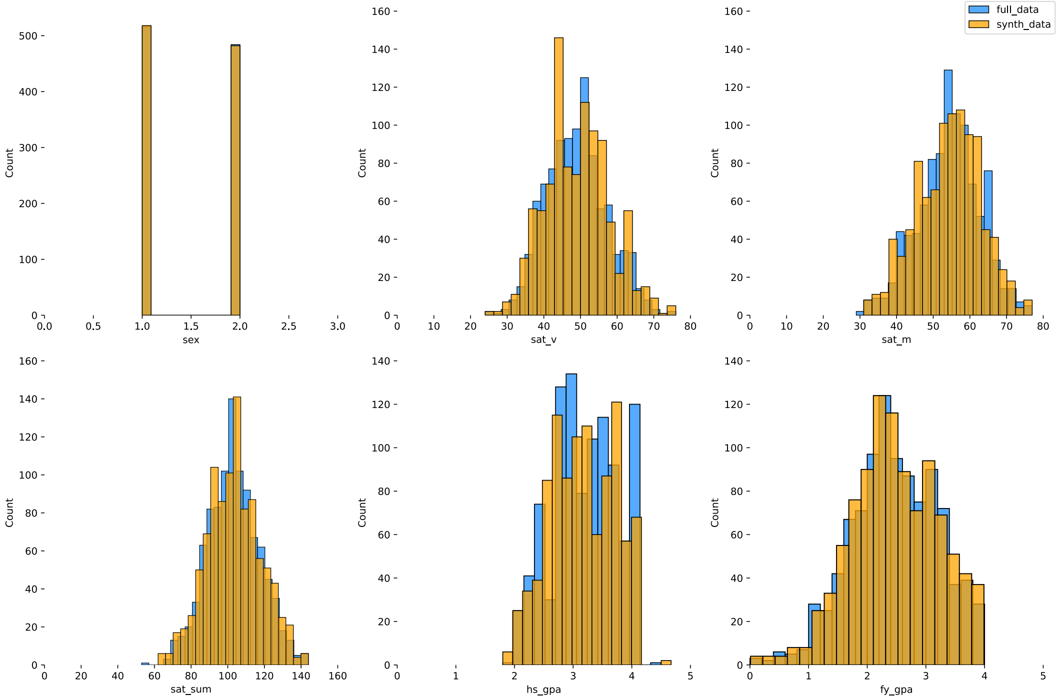
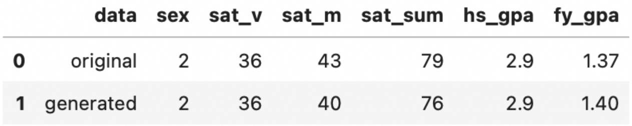


Figure 3: One-dimensional distribution GEMINAI synthetic data vs. Real data.

Similarly, we could see on the table that correlations between multi-columns were well preserved, in particular looking at the **RegressionComparison** test. The performance was really good been the second best out all the methods. In the regression test a multi-linear regression model was built using the both datasets to predict `fy_gpa`, the testing was done as explained on the section Utility Metrics. More regarding the correlations as well a graphical comparison between the real SAT data and that produced by Diveplane can be found in the Profile of GEMINAI Synthetic SAT Data vs. the Real Data.

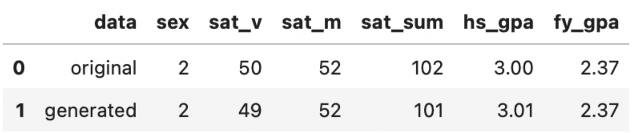
In terms of privacy, the method performed really well, however the data still suffer of possible disclosure risks. For example, the **Density Anonymity Preservation** metric had a lower performance finding a record that is potentially dangerous in terms of privacy.



	data	sex	sat_v	sat_m	sat_sum	hs_gpa	fy_gpa
0	original	2	36	43	79	2.9	1.37
1	generated	2	36	40	76	2.9	1.40

Figure 4: Records potentially dangerous pointed out by the **Density Anonymity Preservation**.

Similarly the **Distance Anonymity Preservation** pointed out the possible of a disclosure risk



	data	sex	sat_v	sat_m	sat_sum	hs_gpa	fy_gpa
0	original	2	50	52	102	3.00	2.37
1	generated	2	49	52	101	3.01	2.37

Figure 5: Records potentially dangerous pointed out by the **Distance Anonymity Preservation**.

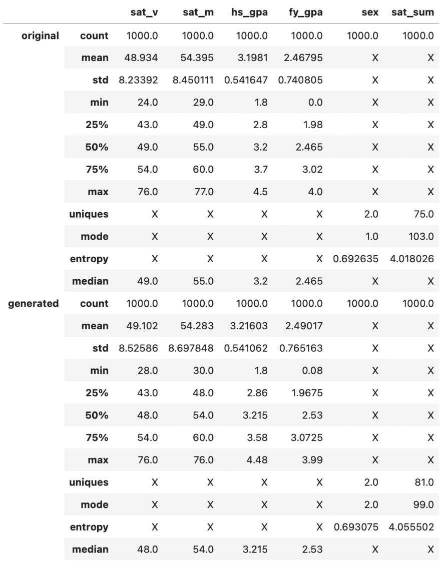
Overall we can say that the performance of Diveplane method was good, but still needs further audits in terms of privacy.

5.4.5 Results of SDV Syntheses

We tried two different synthesis methods one deep learning based using CTGAN and the other using Gaussian Copulas.

- **CTGAN**
  - Profile of SDV CT GAN Synthetic SAT Data vs. the Real Data

The utility of this method was the lowest of them. The standard deviation and the median of the model were a bit different in comparison with the original data., as is shown in the table below.



		sat_v	sat_m	hs_gpa	fy_gpa	sex	sat_sum
original	**count**	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0
	mean	48.934	54.395	3.1981	2.46795	X	X
	std	8.23392	8.450111	0.541647	0.740805	X	X
	min	24.0	29.0	1.8	0.0	X	X
	25%	43.0	49.0	2.8	1.98	X	X
	50%	49.0	55.0	3.2	2.465	X	X
	75%	54.0	60.0	3.7	3.02	X	X
	max	76.0	77.0	4.5	4.0	X	X
	uniques	X	X	X	X	2.0	75.0
	mode	X	X	X	X	1.0	103.0
	entropy	X	X	X	X	0.692635	4.018026
	median	49.0	55.0	3.2	2.465	X	X
generated	**count**	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0
	mean	49.102	54.283	3.21603	2.49017	X	X
	std	8.52586	8.697848	0.541062	0.765163	X	X
	min	28.0	30.0	1.8	0.08	X	X
	25%	43.0	48.0	2.86	1.9675	X	X
	50%	48.0	54.0	3.215	2.53	X	X
	75%	54.0	60.0	3.58	3.0725	X	X
	max	76.0	76.0	4.48	3.99	X	X
	uniques	X	X	X	X	2.0	81.0
	mode	X	X	X	X	2.0	99.0
	entropy	X	X	X	X	0.693075	4.055502
	median	48.0	54.0	3.215	2.53	X	X

Figure 6: General descriptive statics comparison for synthesis with SDV CTGAN

We saw that indeed, it did a poor job on preserving the one-dimensional distributions. In particular, the distribution of sex and `sat_v`.

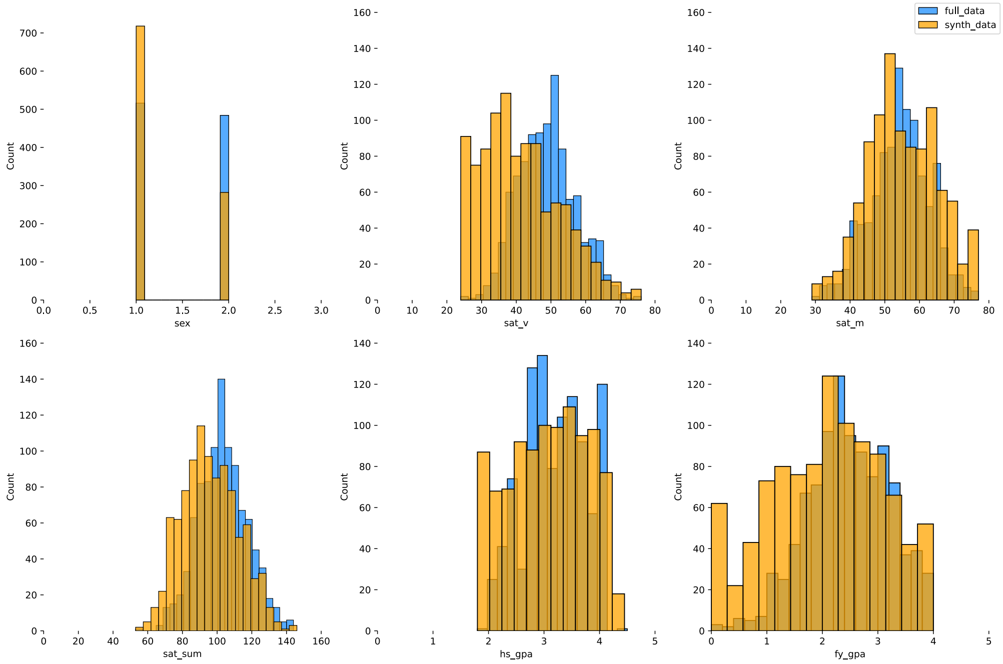
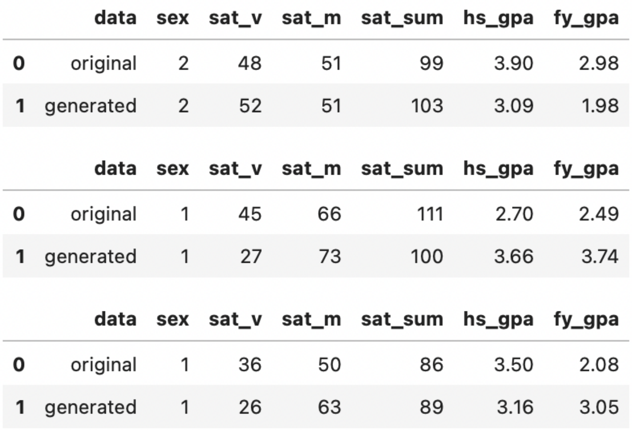


Figure 7: One-dimensional distribution SDV CTGAN synthetic data vs. Real data.

Similarly, we could see on the table that all the correlations between multi-columns were poorly preserved, again looking at the **RegressionComparison** test. The performance was poor, supporting again the statement of lack of correlation. More regarding the correlations as well a graphical comparison between the real SAT data and that produced by SDV CTGAN can be found in the Profile of SDV CTGAN Synthetic SAT Data vs. the Real Data.

In terms of privacy, the method performed really well, due to the lack of preserved correlations helped to the disclosure risk, supporting the trade-off curves between privacy and utility. However the data still suffer of possible disclosure risks. For example, the **Density Anonymity Preservation** and **Distance Anonymity Preservation** metrics pointed out this possible records that could be potentially dangerous in terms of privacy.



	data	sex	sat_v	sat_m	sat_sum	hs_gpa	fy_gpa
0	original	2	48	51	99	3.90	2.98
1	generated	2	52	51	103	3.09	1.98

	data	sex	sat_v	sat_m	sat_sum	hs_gpa	fy_gpa
0	original	1	45	66	111	2.70	2.49
1	generated	1	27	73	100	3.66	3.74

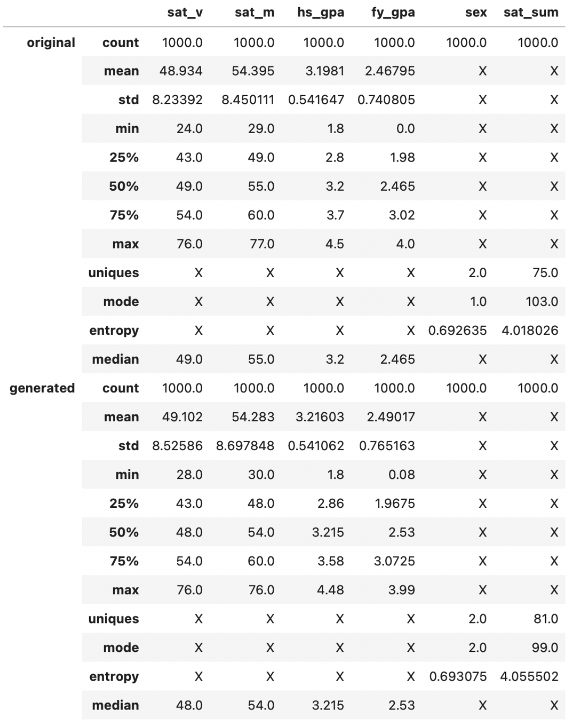
	data	sex	sat_v	sat_m	sat_sum	hs_gpa	fy_gpa
0	original	1	36	50	86	3.50	2.08
1	generated	1	26	63	89	3.16	3.05

Figure 8: Records potentially dangerous pointed out by **Density Anonymity Preservation** and **Distance Anonymity Preservation**.

Overall the method did poorly, one possible reason could be the size of the original dataset. Being a deep learning method the size of the input data in the training model has a strong impact on the final result.

- **Gaussian Copula**
  - Profile of SDV Gaussian Copula Synthetic SAT Data vs. the Real Data

This method was one of the best in terms of utility and it descriptive statistics



		sat_v	sat_m	hs_gpa	fy_gpa	sex	sat_sum
original	**count**	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0
	mean	48.934	54.395	3.1981	2.46795	X	X
	std	8.23392	8.450111	0.541647	0.740805	X	X
	min	24.0	29.0	1.8	0.0	X	X
	25%	43.0	49.0	2.8	1.98	X	X
	50%	49.0	55.0	3.2	2.465	X	X
	75%	54.0	60.0	3.7	3.02	X	X
	max	76.0	77.0	4.5	4.0	X	X
	uniques	X	X	X	X	2.0	75.0
	mode	X	X	X	X	1.0	103.0
	entropy	X	X	X	X	0.692635	4.018026
	median	49.0	55.0	3.2	2.465	X	X
generated	**count**	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0
	mean	49.102	54.283	3.21603	2.49017	X	X
	std	8.52586	8.697848	0.541062	0.765163	X	X
	min	28.0	30.0	1.8	0.08	X	X
	25%	43.0	48.0	2.86	1.9675	X	X
	50%	48.0	54.0	3.215	2.53	X	X
	75%	54.0	60.0	3.58	3.0725	X	X
	max	76.0	76.0	4.48	3.99	X	X
	uniques	X	X	X	X	2.0	81.0
	mode	X	X	X	X	2.0	99.0
	entropy	X	X	X	X	0.693075	4.055502
	median	48.0	54.0	3.215	2.53	X	X

Figure 9: General descriptive statics comparison for synthesis with SDV Gaussian Copula.

Its performance overall was really good preserving multi-table correlations and single column distribution.

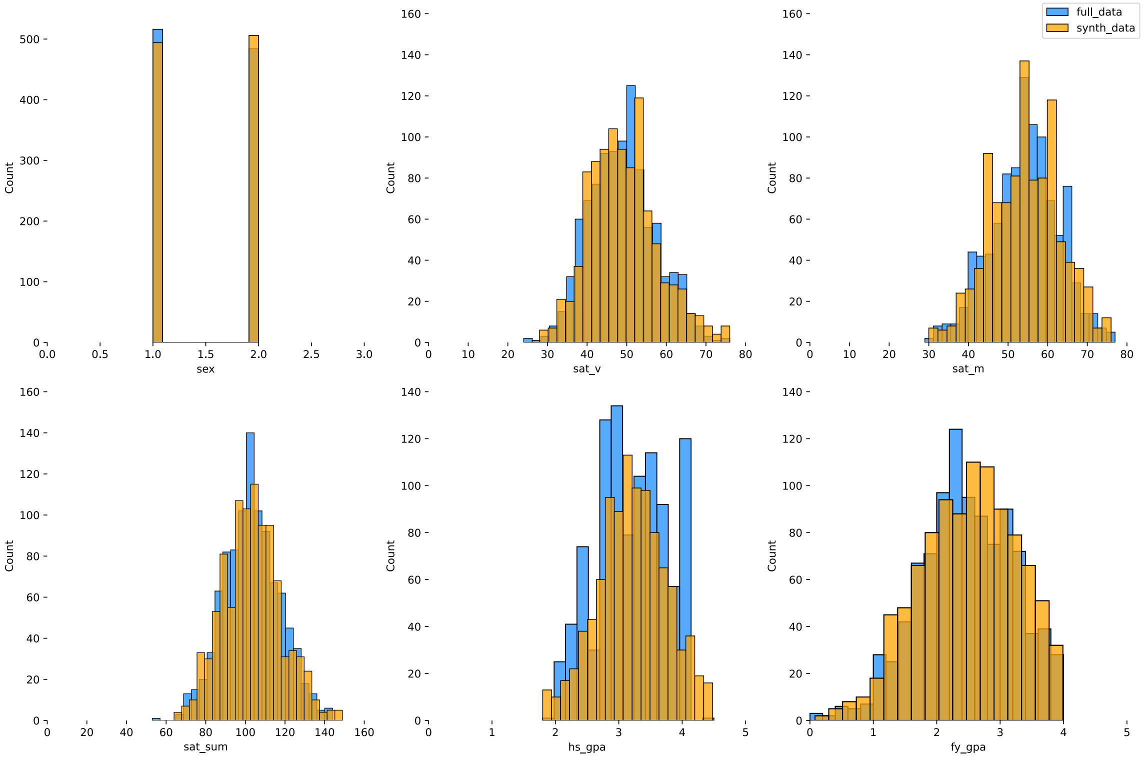
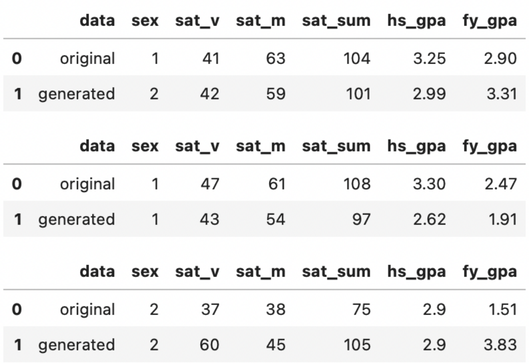


Figure 10: One-dimensional distribution SDV Gaussian Copula synthetic data vs. Real data.

with the worst preserved distribution being the hs_gpa.

In terms of **RegressionComparison** test, the performance was outstanding, supporting again the statement of well preserved correlations. More regarding the correlations as well a graphical comparison between the real SAT data and that produced by SDV CTGAN can be found in the Profile of SDV Gaussian Copula Synthetic SAT Data vs. the Real Data.

In terms of privacy, again was the one with the best score. However as all the synthesis methods the data still suffer of possible disclosure risks. For example, the **Density Anonymity Preservation** and **Distance Anonymity Preservation** metrics pointed out this possible records that could be potentially dangerous in terms of privacy



	data	sex	sat_v	sat_m	sat_sum	hs_gpa	fy_gpa
0	original	1	41	63	104	3.25	2.90
1	generated	2	42	59	101	2.99	3.31

	data	sex	sat_v	sat_m	sat_sum	hs_gpa	fy_gpa
0	original	1	47	61	108	3.30	2.47
1	generated	1	43	54	97	2.62	1.91

	data	sex	sat_v	sat_m	sat_sum	hs_gpa	fy_gpa
0	original	2	37	38	75	2.9	1.51
1	generated	2	60	45	105	2.9	3.83

Figure 11: Records potentially dangerous pointed out by **Density Anonymity Preservation** and **Distance Anonymity Preservation**.

Overall this was one of the fastest and easiest method to use and generate the synthetic data.

5.4.6 Results of Synthpop

Synthpop provides default parameters to synthetize the data:

- CART method used (except for the first variable)
- Synthesis incremental in the variables order presented in the dataset
- All previously synthesised variables used as predictors
- The `sex` variable was converted to factors

You can see a complete profile of Synthpop (default) Synthetic SAT Data vs. the Real Data

Additionally, an alternative version was produced called "Synthpop Smooth". The main changes introduced were:

- `hs_gpa` column moved to be second in the order.
- Smoothing option was activated for the variables: `sat_v`, `sat_m` and `fy_gpa`.
- Option `proper` was set to `TRUE`. More info here.

Synthpop visualisations:

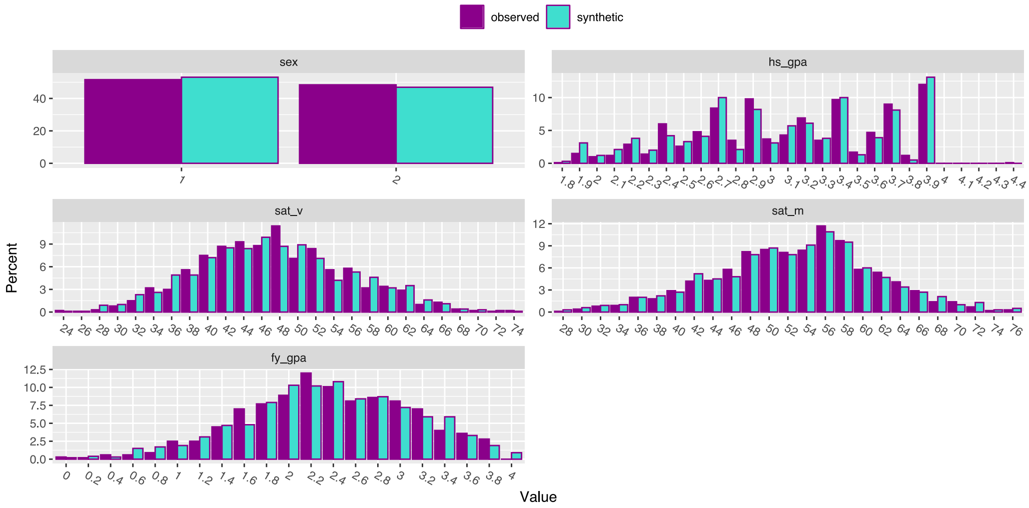
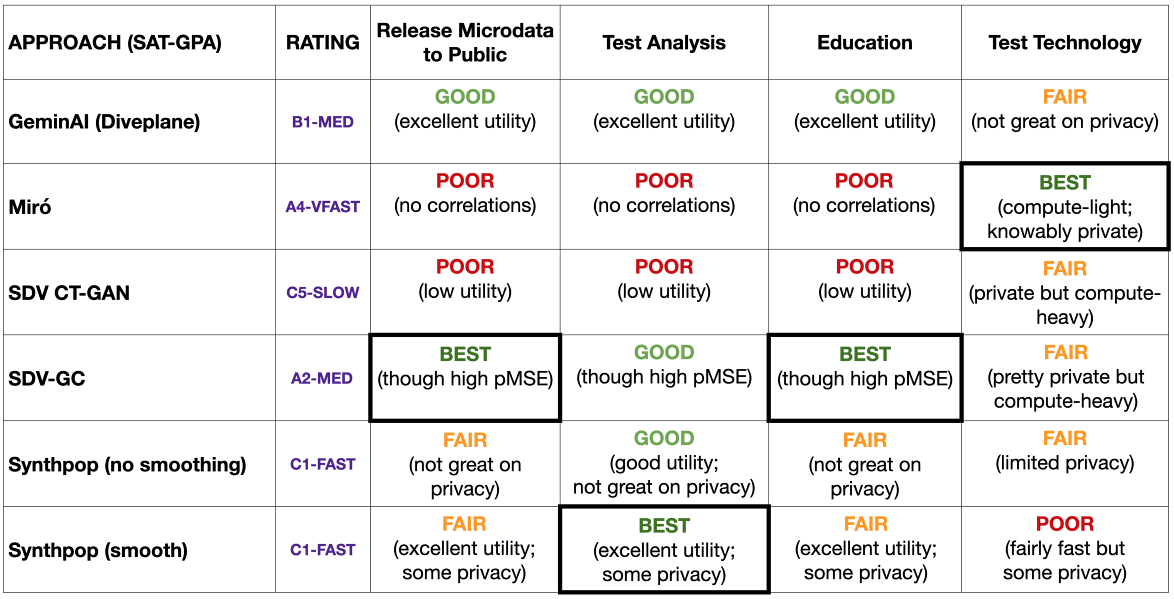


Figure 12: One-dimensional distribution Synthpop synthetic data vs. Real data.

Here more details about the results: Profile of Synthpop (Smooth) Synthetic SAT Data vs. the Real Data



APPROACH (SAT-GPA)	RATING	Release Microdata to Public	Test Analysis	Education	Test Technology
GeminAI (Diveplane)	B1-MED	GOOD (excellent utility)	GOOD (excellent utility)	GOOD (excellent utility)	FAIR (not great on privacy)
Miró	A4-VFAST	POOR (no correlations)	POOR (no correlations)	POOR (no correlations)	BEST (compute-light; knowably private)
SDV CT-GAN	C5-SLOW	POOR (low utility)	POOR (low utility)	POOR (low utility)	FAIR (private but compute-heavy)
SDV-GC	A2-MED	BEST (though high pMSE)	GOOD (though high pMSE)	BEST (though high pMSE)	FAIR (pretty private but compute-heavy)
Synthpop (no smoothing)	C1-FAST	FAIR (not great on privacy)	GOOD (good utility; not great on privacy)	FAIR (not great on privacy)	FAIR (limited privacy)
Synthpop (smooth)	C1-FAST	FAIR (excellent utility; some privacy)	BEST (excellent utility; some privacy)	FAIR (excellent utility; some privacy)	POOR (fairly fast but some privacy)

Figure 13: SAT models Summary and Overall Recommendations

6 ACS dataset

The main dataset includes survey data, including demographic and financial features, representing a subset of IPUMS American Community Survey data for Ohio and Illinois from 2012–2018. The data includes a large feature set of quantitative survey variables along with simulated individuals (with a sequence of records across years), time segments (years), and map segments (PUMA). Teams in this sprint are asked to produce a list of records (i.e. synthetic data) with corresponding time/map segments. There are 36 total columns in the ground truth data, including PUMA, YEAR, 33 additional survey features, and an ID denoting simulated individuals.

6.1 ACS Metadata

Field	Description	Sensitive	Type	Range	To be synthetize	Confidential/Dependent
PUMA	Public Use Microdata Area (PUMA) where the housing unit was located (map segment)	No	Categorical		Yes	No
YEAR	time	No	Continuous	[2012,2018]	Yes	No
HHWT	Indicates how many households in the U.S. population are represented by a given household	No	Continuous	0	No	No
GQ	Classifies all housing units	No	Categorical	1-6	Yes	No
PERWT	Indicates how many persons in the U.S. population are represented by a given person in an IPUMS sample	No	Continuous	¿ 0	No	No
SEX	male or female	No	Categorical	1-2	Yes	No
AGE	age	No	Continuous	[21,100]	Yes	No
MARST	marital status	No	Categorical	1-6	Yes	No
RACE	race	No	Categorical	1-9	Yes	No
HISPAN	origin	No	Categorical	0-4	Yes	No
CITIZEN	citizenship	No	Categorical	0-3	Yes	No
SPEAKENG	speaks only English	No	Categorical	1-9	Yes	No
HCOVANY	had any health insurance coverage	Yes	Categorical	1,2	Yes	No
HCOVPRIV	private health insurance	Yes	Categorical	1,2	Yes	No
HINSEMP	employer-provided	Yes	Categorical	1,2	Yes	No
HINSCAID	Medicaid or other government insurance	Yes	Categorical	1,2	Yes	No
HINSCARE	Medicare coverage	Yes	Categorical	1,2	Yes	No
EDUC	highest year of school or degree completed	No	Categorical	0-11	Yes	No
EMPSTAT	Employment status	Yes	Categorical	0-3	Yes	Yes
EMPSTATD	Employment status detailed	Yes	Categorical	10-15,20-22,30-34	Yes	Yes
LABFORCE	Labor force status	Yes	Categorical	1,2	Yes	Yes
WRKLSTWK	Worked last week	Yes	Categorical	1-3	Yes	Yes
ABSENT	Absent from work last week	Yes	Categorical	1-4	Yes	Yes
LOOKING	Looking for work	Yes	Categorical	1-3	Yes	Yes
AVAILBLE	Available for work	Yes	Categorical	1-5	Yes	Yes
WRKRECAL	Informed of work recall	Yes	Categorical	1-3	Yes	Yes
WORKEDYR	Worked last year	Yes	Categorical	1-3	Yes	Yes
INCTOT	Total personal income	Yes	Continuous		Yes	Yes
INCWAGE	Wage and salary income	Yes	Continuous	¿ 0	Yes	Yes
INCWELFR	Welfare (public assistance) income	Yes	Continuous	¿ 0	Yes	Yes
INCINVST	Interest, dividend, and rental income	Yes	Continuous		Yes	Yes
INCEARN	Total personal earned income	Yes	Continuous		Yes	Yes
POVERTY	Poverty status	Yes	Continuos	000, 001 -501	Yes	Yes
DEPARTS	Time of departure for work	No	Continuous	0 ¡, ¿ 2400	No	Yes
ARRIVES	Time of arrival at work	No	Continuous	0 ¡, ¿ 2400	No	Yes
sim_individual_id	Unique, synthetic ID	No	Continuous	¿ 0	No	No

6.2 Data preparation

6.2.1 Routine Preprocessing

In all cases:

- For many reasons, we restricted ourselves to a random sample of about 20k records from the sample. This allowed faster turn-around times, faster evaluation times (particularly when computing all-to-all distance) and use of more expensive methods. In reality, we would, of course, use all the data. In all cases, we generated the same number of synthetic records (just over 20k) as were in the original data.
- The unnamed index, `HHWT`, `PERWT` and `sim_individual_id` columns were removed
- We decomposed the `DEPARTS` column into two: `DEP-HOUR` (0-23) and `DEP-MINS` (0-59)
- transformed `ARRIVES` into the `JOURNEYTIME`, which is the time taken for the journey to work time in minutes.

After synthesis, we reconstitute `DEPARTS` and `ARRIVES`, but leave some of he other fields in for constraint checking.

In some cases (labelled **THICK**, where relevant) we also applied some transformation to the income and povery fields.

6.2.2 Income ("THICK" approach)

- We observe that INCTOT is almost, but not in all cases exactly, the sum of other components

$$\texttt{INCTOT} \simeq \texttt{INCINVST} + \texttt{INCWELFR} + \texttt{INCEARN}$$

- In many cases INCWAGE = INCEARN, and when they are not equal, INCEARN is usually higher.

- The trouble with this for synthesis is that many (most?) synthesis techniques will tend to struggle to make this approximate relationship work most of the time. As a result, we used following (**THICK**) approach:

  - derive the following:

```
earn_extra  = INCEARN - INCWAGE
missing     = INC_TOT - (INCINVST + INCWELFR + INCWAGE)

wageprop    = INCWAGE / INCTOT
welfprop    = INCWELFR / INCTOT
invprop     = INCINVST / INCTOT
extraprop   = earn_extra / INCTOT
missingprop = missing / INTTOT
```

  - These proportions sum to 1.0. We remove all the income components leaving only INCTOT and the prop fields above and use those for synthesis.

- After synthesis we reverse these transformations to recover the INCOME components in the real data with one twist:

  - before using the synthesized proportions, we compute the sum of all the props, and divide each one by that total, so that after this process they again sum to 1 (including the missingprop).

6.2.3 POVERTY ("THICK" approach)

- We observe that all the approaches except Miró had trouble synthesizing the POVERTY accurately. This is because the field is capped at 501 (representing more than 5 times the POVERTY threshold, with about 30% of records.

- We generate an extra boolean column, POVERTY501, which is 1 (true) if POVERTY is 501 and 0 (false) otherwise. During synthesis, we expect this column to be strongly correlated with largePOVERTY values.

- After synthesis, we simply replace any POVERTY value for which the synthesized POVERTY501 variable is 1 (true) with 501.

The dataset was initially prepared, these are the main steps:

- unnamed index and HHWT and PERWT and sim_individual_id columns were removed
- Split ARRIVES column into two: DEP-HOUR and DEP-MINS/
- transformed DEPARTS into the JOURNEYTIME, which is the journey to work time in minutes.

6.2.4 Data Exploration

As part of data exploration we used the Miró implementation of test-driven data analysis (TDDA) to generate constraints describing the input data. These constraints specify the names, types minimum and maximum values for each field, and can (though in this case didn't) also generate constraints on uniqueness of values in a field, whether missing values are allowed. For strings fields, it also generates regular expression constraints or lists of allowed values, or both.

6.3 Synthesis Results

The following tables give a summary of the performance of the methods using both Utility and Privacy Metrics

6.3.1 Utility Metrics Table

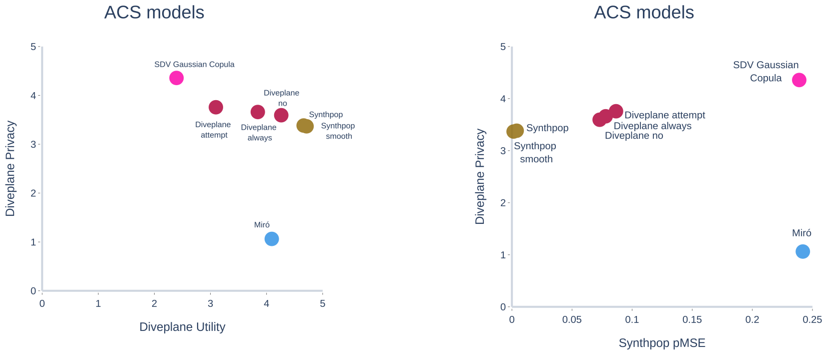
Metrics	Miró	GEMINAI always	GEMINAI attempt	GEMINAI no	SDV gcopula	Synthpop	Synthpop Smooth
Overall Desirability	4.093	3.844	3.097	4.262	2.395	4.661	4.713
DescriptiveStatistics	3.428	4.392	3.948	4.404	2.697	4.692	4.927
MMDStatistics	4.018	4.012	4.079	4.099	4.014	4.001	4.032
GTest	5.0	1.22	0.372	2.361	0.133	4.219	4.231
MannWhitney	4.221	4.846	4.748	4.853	4.65	4.87	4.982
ChiSquare	5.0	5.0	5.0	5.0	4.992	5.0	5.0
KolmogorovSmirnov	4.021	4.811	4.086	5.0	3.001	5.0	5.0
RegressionComparison	3.293	4.945	4.698	4.938	4.509	4.959	4.934
pMSE	0.242	0.078	0.0866	0.073	0.239	0.00398	0.00142
S_pMSE	188.070	69.057	74.0591	104.921	204.661	3.34067	1.20313
P050	49.204	23.999	26.1661	24.401	48.777	5.37888	3.35603

6.3.2 Privacy Metrics Table

NOTE: We are suspicious of some the individual metrics and in particular do not believe the KL-Divergence figure for Miró (which completely destroyed its otherwise good privacy metrics) identify a significant risk, though would want to investigate this further before any actual release.

Metrics	Miró	GEMINAI always	GEMINAI attempt	GEMINAI no	SDV gcopula	Synthpop	Synthpop Smooth
Overall Desirability	1.062	3.66	3.757	3.594	4.357	3.383	3.367
DistanceAnonymityPreservation	5.0	5.0	5.0	5.0	5.0	5.0	5.0
DensityAnonymityPreservation	3.817	3.334	3.334	3.294	3.746	3.404	3.533
AverageDensityAnonymityPreservation	5.0	4.291	4.285	4.244	4.88	4.262	4.252
KAnonymity	3.0	2.183	2.106	2.085	2.994	1.724	1.73
KlDivergence	0.001	3.077	3.74	2.959	5.0	2.396	2.242
LDiversity	5.0	5.0	5.0	5.0	5.0	5.0	5.0
`replications`	0	0	0	0	0	0	0
`uniques`	20360	20360	20360	20360	20360	20360	20360
`per replications`	0	0	0	0	0	0	0

Another way of visualising the results of utility and privacy metrics is shown in the figure below.



(a) ACS diveplane metrics (b) ACS synthpop metrics

Figure 14: ACS metrics

6.3.3 Results of Miró Synthesis

- **BEST** Profile of Miró (Wide, Thick) Synthetic ACS Data vs. the Real Data
- Profile of Miró Synthetic ACS Data vs. the Real Dataa

A graphical comparison between the real ACS data and that produced by two different Miró runs is available through the links above.

The first is a completely naïve approach, in which no transformations apart from the standard ones are applied; the second approach uses the **THICK** methodology described above (in which income components are handled as weights) and in which the income range is expanded in otder to reduce the risk of disclosure of the true maximum an minimum values. As can be clearly seen, the one-dimensional distributions are generally accurately preserved in both cases, as are most constraints. In the non-wide version, `DEPARTS` is sometimes out of range because there are no departure times later than 2345 in the real data. That is resolved in the `wide` version, where the allowed constraint range is increased (since 2345 should probably not be a hard limit). The `ARRIVES` maximum also goes out of range, above 2400. This is odd, but could easily be wrapped to turn 2604 into 0204, which would just represent a long journey, starting late. Those didn't occur in the real data, so they they could also be cured a different way.

The other constraint that failed frequently in the naïve version (about half the time) was a constraint on the `missing` income that we create as an extra check. This is the difference between the total of the relevant income components and the total income. This is much improved in the wide-thick version.

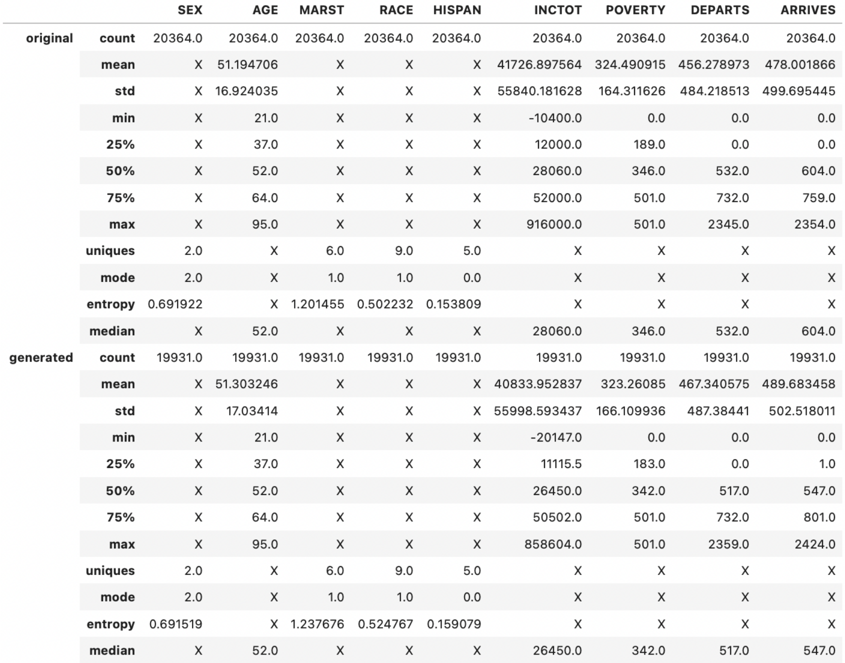
Also as expected, and again, following directly from Miró's synthesis of each field independently, when we look at the profiles that relate POVERTY to other variable, Miró shows no correlations, with essential flat breakdowns except where volumes are small.

Diveplane's privacy metrics downweight the "privacy desirability" of the Miró-generated data quite heavily on a couple of metrics, particularly KL-Divergence; this requires further investigation.

6.3.4 Results of GEMINAI (Diveplane) Synthesis

One of the useful settings of GEMINAI is given by the possibility to tune the generation of possible cases according the distance of the from the record to its nearest neighbour. These setting are summarize in the cases "always", "attempt", and "no". In terms of the overall utility the synthesis done using Diveplane systems comes out pretty good.

In terms of the overall utility the synthesis done using GEMINAI systems came out with good accuracy. It preserves the descriptive statistical attributes of the original dataset as is shown in the summary of statistical attributes of both datasets



		SEX	AGE	MARST	RACE	HISPAN	INCTOT	POVERTY	DEPARTS	ARRIVES
original	count	20364.0	20364.0	20364.0	20364.0	20364.0	20364.0	20364.0	20364.0	20364.0
	mean	X	51.194706	X	X	X	41726.897564	324.490915	456.278973	478.001866
	std	X	16.924035	X	X	X	55840.181628	164.311626	484.218513	499.695445
	min	X	21.0	X	X	X	-10400.0	0.0	0.0	0.0
	25%	X	37.0	X	X	X	12000.0	189.0	0.0	0.0
	50%	X	52.0	X	X	X	28060.0	346.0	532.0	604.0
	75%	X	64.0	X	X	X	52000.0	501.0	732.0	759.0
	max	X	95.0	X	X	X	916000.0	501.0	2345.0	2354.0
	uniques	2.0	X	6.0	9.0	5.0	X	X	X	X
	mode	2.0	X	1.0	1.0	0.0	X	X	X	X
	entropy	0.691922	X	1.201455	0.502232	0.153809	X	X	X	X
	median	X	52.0	X	X	X	28060.0	346.0	532.0	604.0
generated	count	19931.0	19931.0	19931.0	19931.0	19931.0	19931.0	19931.0	19931.0	19931.0
	mean	X	51.303246	X	X	X	40833.952837	323.26085	467.340575	489.683458
	std	X	17.03414	X	X	X	55998.593437	166.109936	487.38441	502.518011
	min	X	21.0	X	X	X	-20147.0	0.0	0.0	0.0
	25%	X	37.0	X	X	X	11115.5	183.0	0.0	1.0
	50%	X	52.0	X	X	X	26450.0	342.0	517.0	547.0
	75%	X	64.0	X	X	X	50502.0	501.0	732.0	801.0
	max	X	95.0	X	X	X	858604.0	501.0	2359.0	2424.0
	uniques	2.0	X	6.0	9.0	5.0	X	X	X	X
	mode	2.0	X	1.0	1.0	0.0	X	X	X	X
	entropy	0.691519	X	1.237676	0.524767	0.159079	X	X	X	X
	median	X	52.0	X	X	X	26450.0	342.0	517.0	547.0

Figure 15: General descriptive statics comparison for synthesis with GEMINAI always

The one-dimensional distributions are well preserved, showing only an small set of columns

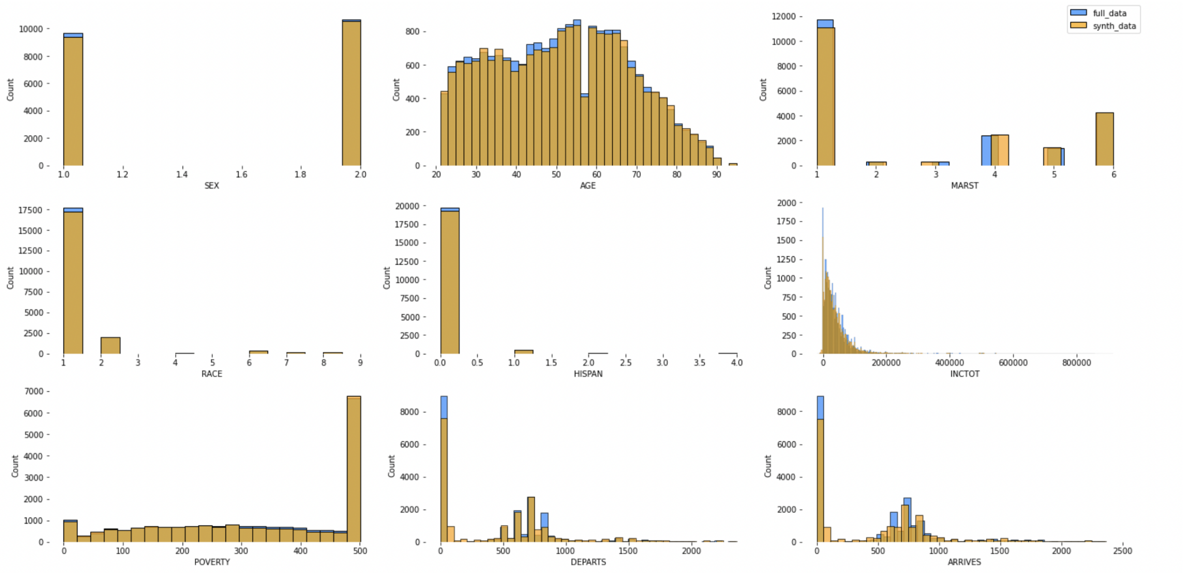


Figure 16: One-dimensional distribution GEMINAI synthetic always data vs. Real data.

For more information on the correlations, in the following profiles

- **BEST** Profile of GEMINAI Synthetic (Always, THICK) ACS Data vs. the Real Data
- Profile of GEMINAI Synthetic (Always) ACS Data vs. the Real Data
- Profile of GEMINAI Synthetic (Attempt, THICK) ACS Data vs. the Real Data
- Profile of GEMINAI Synthetic (Attempt) ACS Data vs. the Real Data
- Profile of GEMINAI Synthetic (No, THICK) ACS Data vs. the Real Data
- Profile of GEMINAI Synthetic (No) ACS Data vs. the Real Data

In terms of privacy, the "attempt" case was the best. It is worth mentioning that we have removed between 2% to 4% of the records which presented a higher disclosure risk. Overall combining all the metrics for GEMINAI, the case "always" was the one with higher desirability.

6.3.5 Results of SDV Synthesis

As before we wanted to tried two different methods one deep learning based using **CTGAN** and the other using **Gaussian Copulas**. However, we were unable to get SDV's Continuous GAN to complete on the 20k sample of the ACS data, so there are no results.

- **Gaussian Copula** It did very poorly on the categorical fields, hence the poor performance on the GTest. Here are some of the distributions:

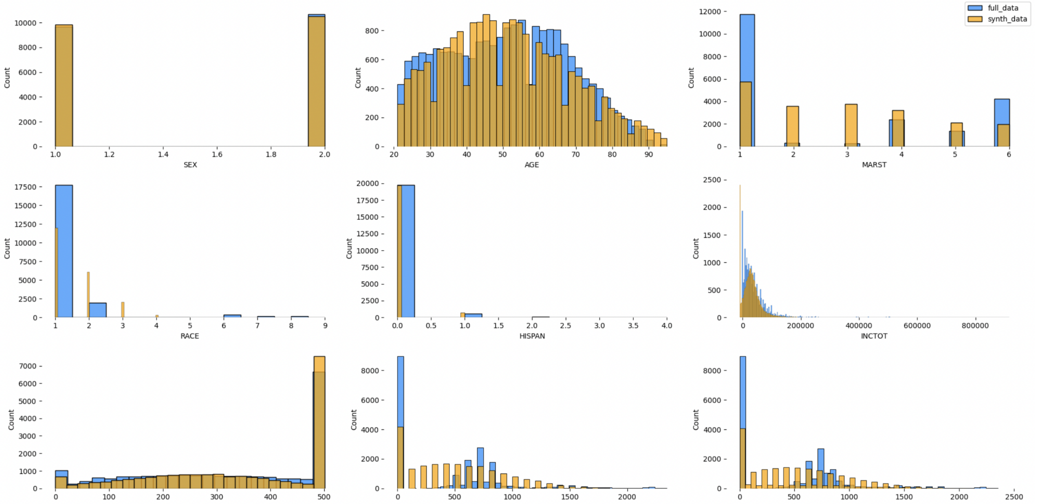
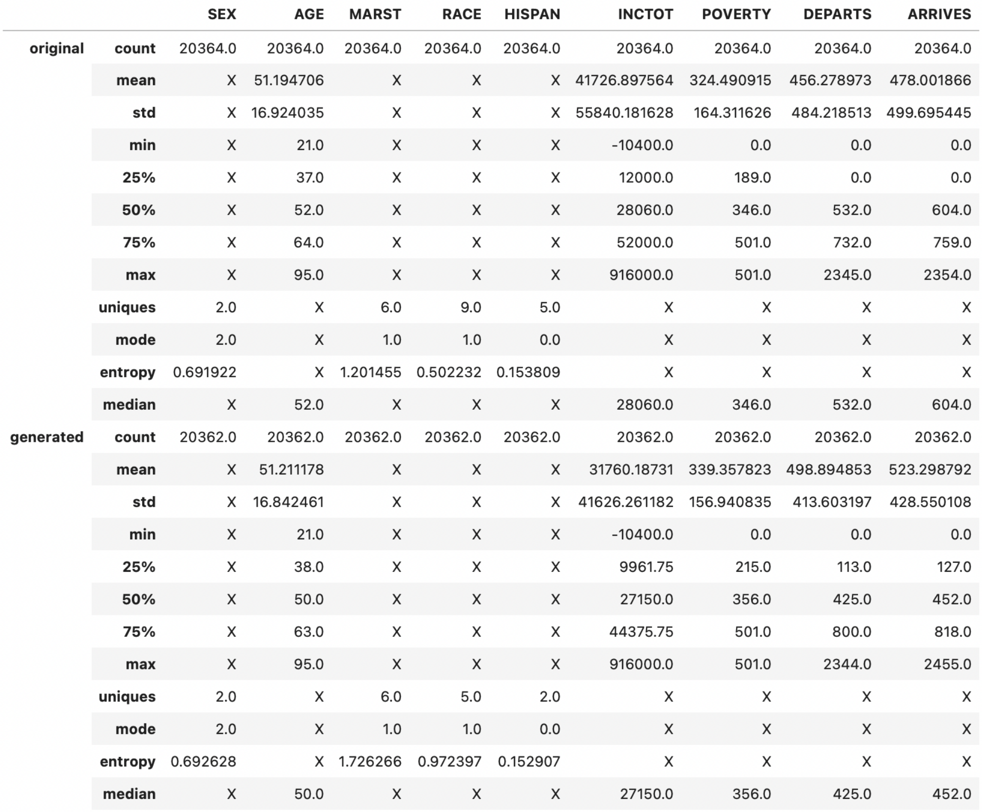


Figure 17: One-dimensional distribution SDV GCOPULA synthetic data vs. Real data.



		SEX	AGE	MARST	RACE	HISPAN	INCTOT	POVERTY	DEPARTS	ARRIVES
original	count	20364.0	20364.0	20364.0	20364.0	20364.0	20364.0	20364.0	20364.0	20364.0
	mean	X	51.194706	X	X	X	41726.897564	324.490915	456.278973	478.001866
	std	X	16.924035	X	X	X	55840.181628	164.311626	484.218513	499.695445
	min	X	21.0	X	X	X	-10400.0	0.0	0.0	0.0
	25%	X	37.0	X	X	X	12000.0	189.0	0.0	0.0
	50%	X	52.0	X	X	X	28060.0	346.0	532.0	604.0
	75%	X	64.0	X	X	X	52000.0	501.0	732.0	759.0
	max	X	95.0	X	X	X	916000.0	501.0	2345.0	2354.0
	uniques	2.0	X	6.0	9.0	5.0	X	X	X	X
	mode	2.0	X	1.0	1.0	0.0	X	X	X	X
	entropy	0.691922	X	1.201455	0.502232	0.153809	X	X	X	X
	median	X	52.0	X	X	X	28060.0	346.0	532.0	604.0
generated	count	20362.0	20362.0	20362.0	20362.0	20362.0	20362.0	20362.0	20362.0	20362.0
	mean	X	51.211178	X	X	X	31760.18731	339.357823	498.894853	523.298792
	std	X	16.842461	X	X	X	41626.261182	156.940835	413.603197	428.550108
	min	X	21.0	X	X	X	-10400.0	0.0	0.0	0.0
	25%	X	38.0	X	X	X	9961.75	215.0	113.0	127.0
	50%	X	50.0	X	X	X	27150.0	356.0	425.0	452.0
	75%	X	63.0	X	X	X	44375.75	501.0	800.0	818.0
	max	X	95.0	X	X	X	916000.0	501.0	2344.0	2455.0
	uniques	2.0	X	6.0	5.0	2.0	X	X	X	X
	mode	2.0	X	1.0	1.0	0.0	X	X	X	X
	entropy	0.692628	X	1.726266	0.972397	0.152907	X	X	X	X
	median	X	50.0	X	X	X	27150.0	356.0	425.0	452.0

Figure 18: General descriptive statics comparison for synthesis with SDV GCOUPLA

In terms of privacy, it was best, but we had to remove again between two to five records to get a better desirability. More on the multi-table correlations can be found in the following profile

- Profile of SDV Gaussian Copula (Thick) Synthetic ACS Data vs. the Real Data

- Profile of SDV Gaussian Copula Synthetic ACS Data vs. the Real Data

6.3.6 Results of Synthpop

- Profile of Synthpop (Thick) Synthetic ACS Data vs. the Real Data
- Profile of Synthpop Synthetic ACS Data vs. the Real Data
- Profile of Synthpop Smooth (Thick) Synthetic ACS Data vs. the Real Data
- Profile of Synthpop Smooth Synthetic ACS Data vs. the Real Data

As the other methods, Synthpop was used on a random sample of 20k records. The pre-process method is explained in the section 5.2.

Additionally to the default parameters we have introduced some changes in the data and the parameters to improve utility and privacy metrics:

- Categorical variables converted to factors
- CART method (except the first variable)
- Moved `PUMA` column at the end (so it is not used as a predictor), synthesis was done in the order of variables after this.
- Smoothing option set to "density" was applied to the proportions of the INCOME variables. For "density" smoothing a Gaussian kernel density estimator is applied with bandwidth selected using the Sheather-Jones `solve-the-equation` method. More info about the smoothing option here.
- All previously synthesized variables used as predictors
- Applied Statistical discosure control (sdc) to remove replicated records.

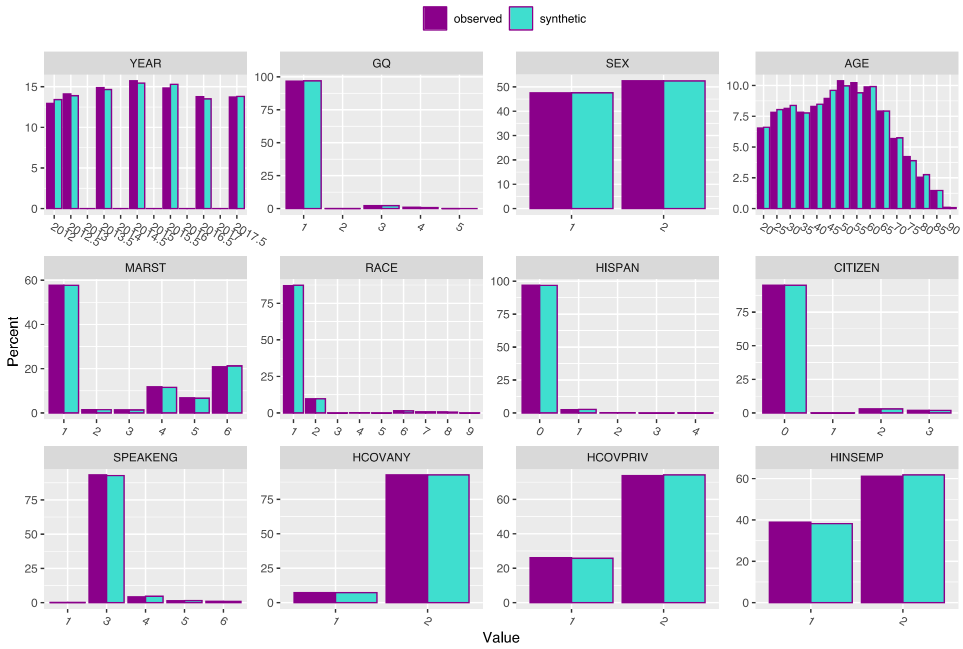


Figure 19: One-dimensional distribution Synthpop Smooth synthetic data vs. Real data.

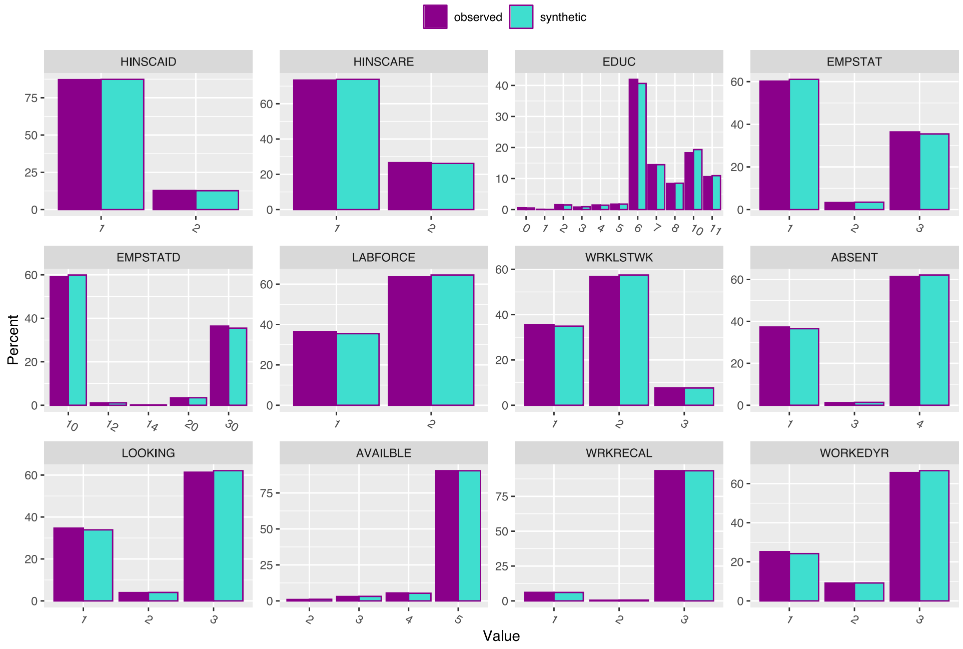


Figure 20: One-dimensional distribution Synthpop Smooth synthetic data vs. Real data.

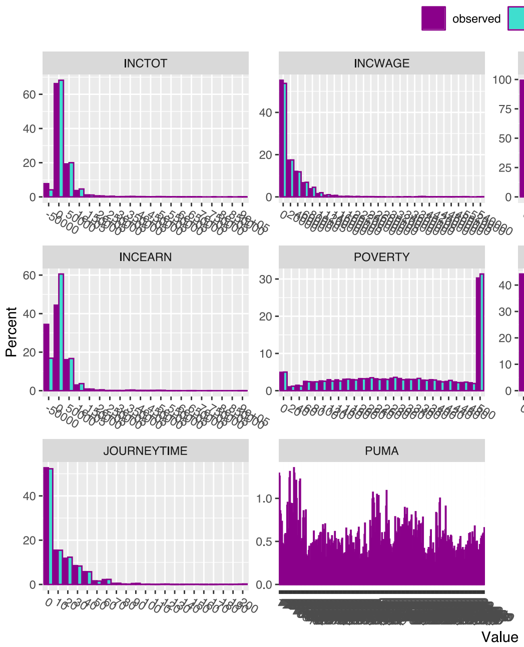
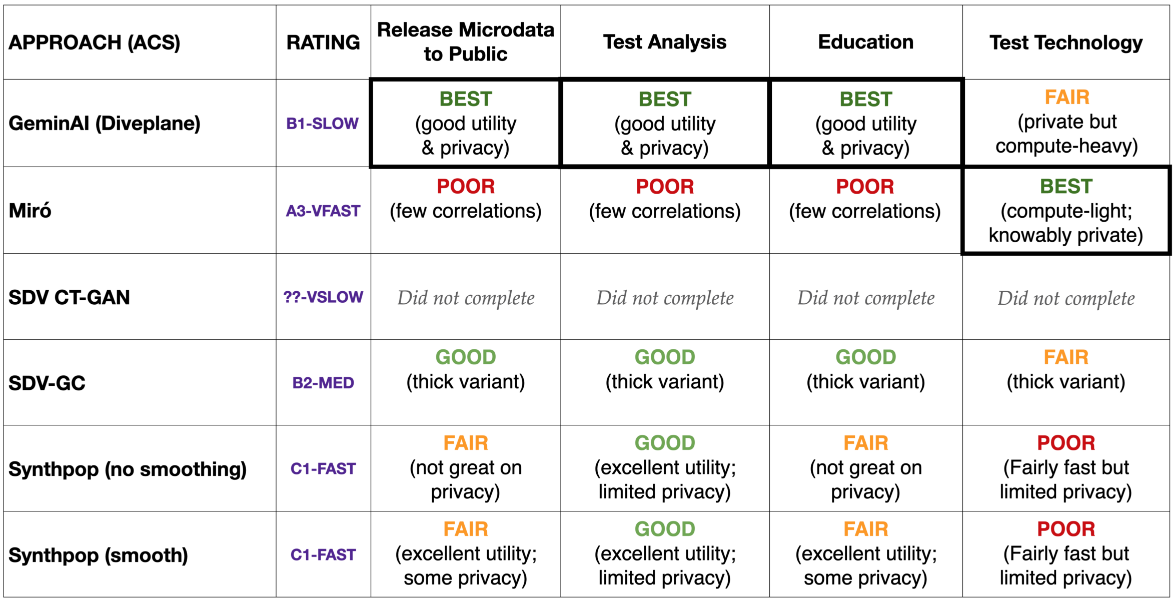


Figure 21: One-dimensional distribution Synthpop Smooth synthetic data vs. Real data.



APPROACH (ACS)	RATING	Release Microdata to Public	Test Analysis	Education	Test Technology
GeminAI (Diveplane)	B1-SLOW	BEST (good utility & privacy)	BEST (good utility & privacy)	BEST (good utility & privacy)	FAIR (private but compute-heavy)
Miró	A3-VFAST	POOR (few correlations)	POOR (few correlations)	POOR (few correlations)	BEST (compute-light; knowably private)
SDV CT-GAN	??-VSLOW	*Did not complete*	*Did not complete*	*Did not complete*	*Did not complete*
SDV-GC	B2-MED	GOOD (thick variant)	GOOD (thick variant)	GOOD (thick variant)	FAIR (thick variant)
Synthpop (no smoothing)	C1-FAST	FAIR (not great on privacy)	GOOD (excellent utility; limited privacy)	FAIR (not great on privacy)	POOR (Fairly fast but limited privacy)
Synthpop (smooth)	C1-FAST	FAIR (excellent utility; some privacy)	GOOD (excellent utility; limited privacy)	FAIR (excellent utility; some privacy)	POOR (Fairly fast but limited privacy)

Figure 22: ACS models: Overall Summary and Recommendations

7 Comments and Recommendations

Our thinking on the importance of the three dimensions for each of the proposed purposes is summarised in this table, with Public Release and Education having the same assumed priorities.

PURPOSE	**TOP PRIORITY**	**SECOND PRIORITY**	**LOWEST PRIORITY**
Release of Microdata to Public	PRIVACY	UTILITY	SPEED
(Internal) Test Analyses	UTILITY	PRIVACY	SPEED
Education	PRIVACY	UTILITY	SPEED
Test Technology	PRIVACY	SPEED	UTILITY

Despite a wealth of statistics indicating that much of the synthetic data we have generated is safe and useful, we would not be comfortable releasing it without further assessment. We would be more comfortable releasing synthetic SAT data than ACS data, and we would today have greatest confidence releasing data from Miró, where the nature of the synthesis intrinsically and knowably limits disclosure risk, notwithstanding one or two metrics casting doubt on this. Obviously, one of the participants is an author of Miró, so readers may conclude that this is bias. We would emphasize, however, that it is *only* for software testing that we would recommend Miró's simplistic approach, because the very same properties that give us confidence about releasing Miró data for testing make it utterly unsuited to fancier purposes.

References

[1] Christopher J. Hazard, Christopher Fusting, Michael Resnick, Michael Auerbach, Michael Meehan, & Valeri Korobov. "Natively Interpretable Machine Learning and Artificial Intelligence: Preliminary Results and Future Directions," https://arxiv.org/abs/1901.002469606387.

[2] Nowok, Beata and Raab, Gillian M. and Dibben Chris "synthpop: Bespoke Creation of Synthetic Data in R," https://www.jstatsoft.org/index.php/jss/article/view/v074i11.