

# SATGPA - Fully Conditional Specification (FCS)

## Evaluation Synthetic Data Creation

Steffen Moritz, Hariolf Merkle, Felix Geyer, Michel Reiffert, Reinhard Tent (DESTATIS)

January 31, 2022

- [Executive Summary](#)
- [Dataset Considerations](#)
- [Method Considerations](#)
- [Privacy and Risk Evaluation](#)
- [Utility Evaluation](#)
- [Tuning and Optimizations](#)

## Executive Summary

We applied **Fully Conditional Specification** (FCS) on the **SAT** dataset via the **synthpop** package. We tried several algorithm and parameter options, our final FCS model is implemented with **CART**. FCS seems to us like a very interesting option for certain use cases. Out of all different methods we tested (**FCS**, **IPSO**, **GAN**, **Simulation**, **Minutemen**) FCS produced the **best usability** results for **SAT**. Of course by providing good usability there is usually a **trade-off** with the privacy measures. Only **IPSO** was behind **FCS** in our main privacy metrics. However, overall the privacy measures were still acceptable for some use cases.

According to our utility measures, FCS is a **very suitable method** to generate synthetic data from the SAT dataset. At a first glance, the marginal distributions and Pearson correlation coefficients match nearly perfectly. This first impression is underpinned by further metrics. Especially the **S\_pMSE** for tables is often close to one. The **S\_pMSE** shows acceptable results for margins. The share of not significant Kolmogorov-Smirnov tests is 100%. The Mahalanobis distance of the regression parameters **never exceeded the critical value**. There is only some limited usability according to Mlodak's information loss criterion. Overall, we see **very satisfying results** regarding the utility measures.

### USE CASE RECOMMENDATIONS

Releasing_to_Public	Testing_Analysis	Education	Testing_Technology
NO	YES	NO	MAYBE

Since the usability results were especially good, FCS is interesting for every use case that requires high usability. Because of the trade-off with privacy we probably would only supply the FCS synthetic data to trusted partners. So **Testing Analysis**, where trusted researchers can develop and test their models before clearance for the actual microdata seems like a very good fit. **Releasing to Public** and **Education** mostly wouldn't fit because of privacy issues. Internal **Technology Testing** could be a possible use case, but for most of these testing cases there are probably easier options requiring less computational power to provide synthetic data.

## Dataset Considerations

When deciding, if data is released to the public it is of utmost importance to define, **which variables** are the **most relevant** in terms of **privacy and utility**. This process is very **domain and country specific**, since different areas of the world have different privacy legislation and feature specific overall circumstances. This step would require input and discussions with actual domain experts. Since we are foreign to US privacy law and there is no SAT equivalent in Germany, the assumptions made for the Synthetic Data Challenge are basically a **educated guess** from our side.

From a **utility perspective** it is important to know which variables and correlations are most interesting for actual users of the created synthetic dataset. Different use cases might require focus on different variables and correlations. We could not single out a most important variable, thus in our utility analysis we decided to focus on the overall SAT utility and **not to prioritize a specific variable**. From a data plausibility perspective it was essential to us, that the `sat_v + sat_m = sat_sum` stay consistent.

From a **privacy perspective** it has to be decided, which variables are **confidential** and which are **identifying**. As already mentioned, specifying this depends on multiple factors e.g. **regulations** or also **other public information**, that could be used for **de-anonymization**. For our analysis, we made the following assumptions: Feature `sex` is an identifying value. For the SAT percentiles (`sat_v`, `sat_m`, `sat_sum`) there can be argued in both directions, but we decided for it to be an identifying variable. The same holds for the grade point average `fy_gpa`. We assumed the grade point average to be more confidential than the SAT results. It is very likely for example that students who passed the SAT exchanged about their grades with their fellow students or that teachers know about the SAT results of their students. So these (older) information potentially can be used to **identify a student** within a dataset and find out about the **(newer) information of grade point average**. The calculus behind this decision is that the older information about a test, that is only used to get the college admission, is **less confidential than the newer information** about actual grades, which might give information about a student's current situation.

## Method Considerations

We decided to use the **FCS** method for multiple reasons. For one the use of the FCS method is fairly simple and straightforward, since no prior knowledge of the relation between the data is necessary to fit a first model. Secondly, the R package **synthpop** already comes with a good implementation of the method. Thirdly, and maybe most importantly, the method can be used for nearly all types of datasets and yield meaningful results. For our first approach we chose to use the default settings of the method, i.e. the order of synthesis of the variables in ascending order, and using the

**Classification and Regression Tree (CART)** machine learning model for each variable. We played around with different algorithm settings. In general, we also found the **pmm** algorithm an interesting option, for certain cases when a high plausibility of the created data is important. In the end, we chose to stick with **CART** since it featured the best results.

## Privacy and Risk Evaluation

### Disclosure Risk (R-Package: synthpop with own Improvements)

Our starting point was the **matching of unique records**, as described in the disclosure risk measures chapter of the starter guide. The synthpop package provides us with an easy-to-use implementation of this method: `replicated.uniques`. However, one downside of just using `replicated.uniques` is that it does **not consider almost exact matches in numeric variables**. Imagine a data set with information about the respondents' income. If there is a matching data point in the synthetic data set for a unique person in the original data set, that only differs by a slight margin, the original function would not identify this as a match. **Our solution** is to borrow the notion of the **p% rule** from **cell suppression methods**, which identifies a data point as critical, if one can guess the original values with **some error of at most p%**. Thus, **our improved risk measure** is able to evaluate disclosure risk in numeric data. Our Uniqueness-Measure for **"almost exact"** matches provides us with the following outputs:

- **Replication Uniques** | Number of unique records in the synthetic data set that replicates unique records in the original data set w.r.t. their quasi-identifying variables. In brackets, the proportion of replicated uniques in the synthetical data set relative to the original data set size is stated.
- **Disclosure in >= 1 CVar** | Number of replicated unique records in the synthetical data set that have a real disclosure risk in at least one confidential variable, i.e. there is at least one confidential variable where the record in the synthetical data set is "too close" to the matching unique record in the original data set. We identify two records as "too close" in a variable, if they differ in this variable by at most p%. In brackets, the described number is given in proportion to the original data set size.
- **Disclosure in 2 CVars** | Number of replicated unique records in the synthetical data set that have a real disclosure risk in both confidential variables, i.e. in both of the confidential variables the record in the synthetical data set is "too close" to the matching unique record in the original data set. We identify two records as "too close" in a variable, if they differ in this variable by at most p%. In brackets, the described number is given in proportion to the original data set size.

For our selected best parametrized solution in this method-category, we got the following results:

Replication.Uniques	Disclosure.in.>=1.CVar	Disclosure.in.2.CVars
93 (9.3%)	40 (4%)	7 (0.7%)

### Perceived Disclosure Risk (R-Package: synthpop)

Unique records in the synthetic dataset may be **mistaken for unique records** based on the fact, that **only the identifying variables match**. This can lead to problems, even if the associated confidential variables significantly differ from the original record. E.g. people might assume a certain income for a person, because they believe to have identified him from the identifying variables. Even if his real income **is not leaked** (as the confidential variables are different), this assumed (but wrong) information about him **might lead to disadvantages**. The **perceived risk** is measured by matching the unique records among the identifying variables, in our case `sex`, `sat_v`, `sat_m` and `sat_sum`. We applied the method `replicated.uniques` of the synthpop package. There is no fixed threshold that must not be exceeded in this measure, however, a smaller percentage of unique matches (referred to as Number Replications) is preferred to minimize the perceived disclosure risk.

These are the results variables for perceived disclosure risk:

- **Number Uniques** | Number of unique individuals in the original data set.
- **Number Replications** | The number of matching records in the synthetic data set (based only on identifying variables). This is the number of individuals, which might perceived as disclosed (real disclosures would also count into this metric)
- **Percentage Replications** | The calculated percentage of duplicates in the synthetic data

For our selected best parametrized solution in this method-category, we got the following results:

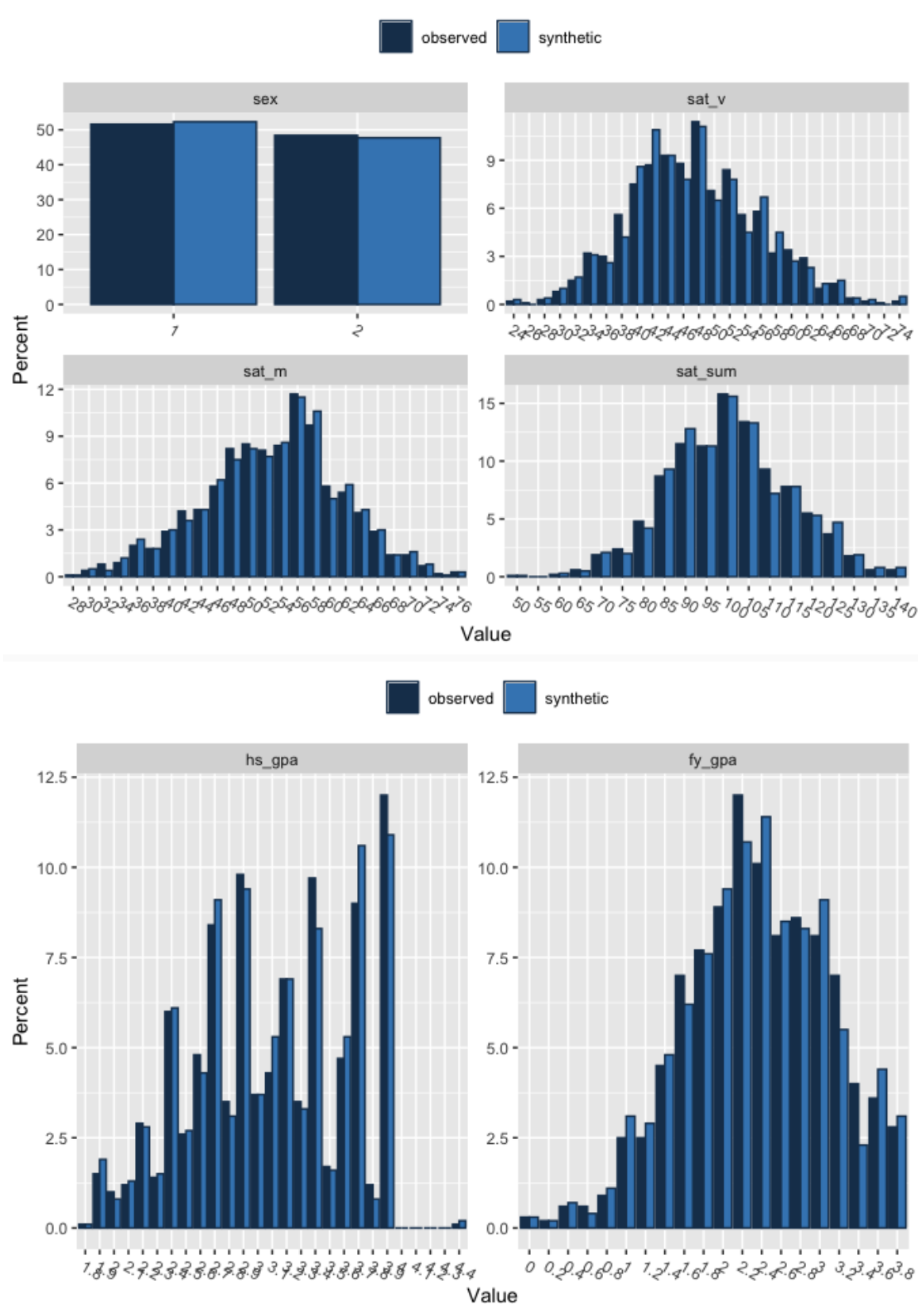
Metric	Number.Uniques	Number.Replications	Percentage.Replications
Perceived Risk	551	93	9.3

## Utility Evaluation

Different utility measures are applied in this section. These utility measures are the basis of the utility evaluation of the generated synthetic dataset. These are the utility measures for our selected dataset after optimizations and tuning.

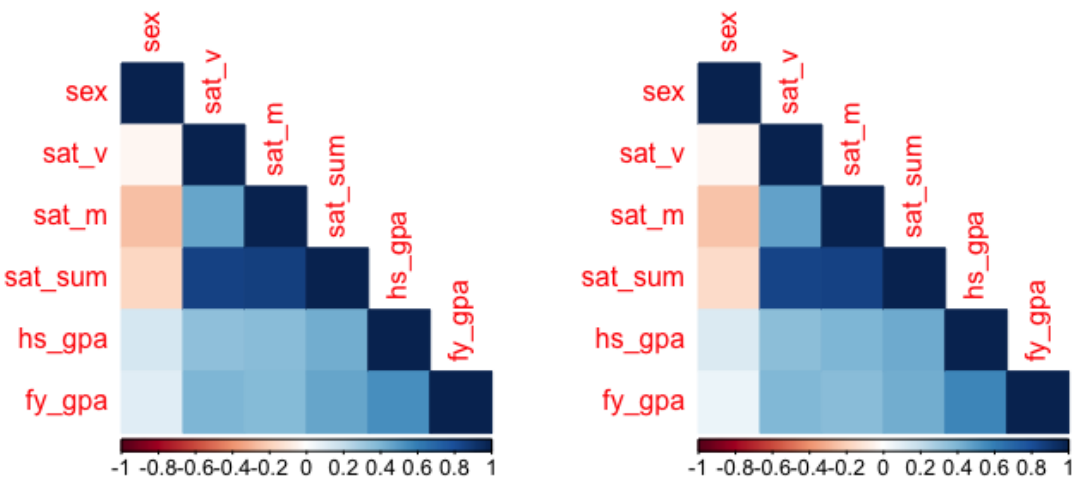
### Graphical Comparison for Margins (R-Package: synthpop)

The following histograms provide an ad-hoc overview on the marginal distributions of the original and synthetic dataset. Matching or close distributions are related to a high data utility.



### Correlation Plots for Graphical Comparison of Pearson Correlation

Synthetic Datasets should represent the dependencies of the original datasets. The following correlation plots provide an ad-hoc overview on the Pearson correlations of the original and synthetic dataset. The left plot shows the original correlation whereas the right plot provides the correlation based on the synthetic dataset.



### Kolmogorov-Smirnov Test

The Kolmogorov-Smirnov test is a classic way to compare (marginal) distributions. A significant result indicates that the two distributions are not identical. The following statistic show the share of variables in the synthetic dataset that have a p-value > 5%. (1: 100%, 0: 0%)

Mean_KS_not_signif
1

### Mahalanobis Distance for Regression Parameters

To assess testing analysis, coefficients and standard errors calculated based on synthetic dataset should lead to the same results when calculated on the original data. The evaluate differences and taking the variance covariance matrix into account, we calculated the mahalanobis distance between the respective regression parameters. Since we used several regression models, we present the mean of cases in which the mahalanobis distance exceeded the critrial value (0: excellent; 1: poor).

### Distributional Comparison of Synthesised Data (R-Package: synthpop) by (S\_)pMSE

Propensity scores are calculated on a combined dataset (original and synthetic). A model (here: CART) tries to identify the synthetic units in the dataset. Since both datasets should be identically structured, the pMSE should equal zero and the S\_pMSE (standardised) should equal one for perfect utility.

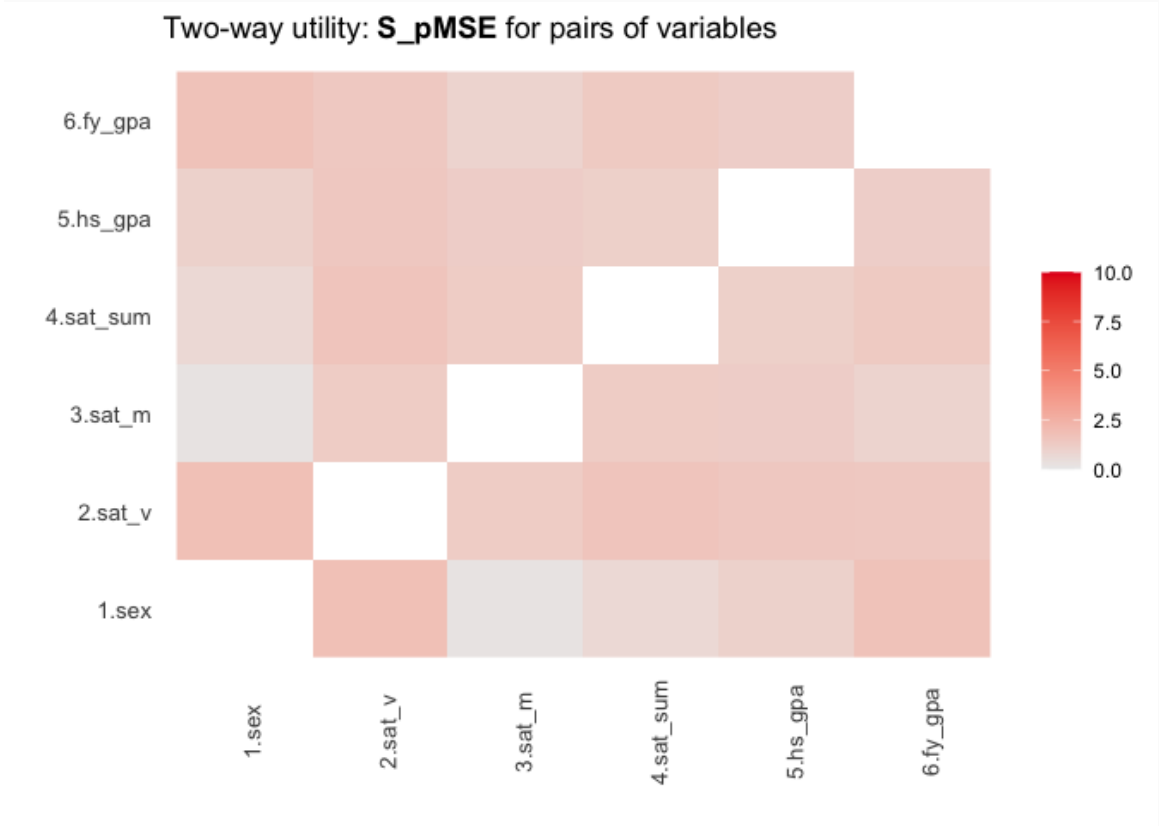
pMSE	S_pMSE	df
------	--------	----

	pMSE	S_pMSE	df
sex	0.0000123	0.1962986	1
sat_v	0.0007711	3.0844448	4
sat_m	0.0000272	0.1088852	4
sat_sum	0.0002773	1.1093592	4
hs_gpa	0.0000307	0.1227265	4
fy_gpa	0.0002054	0.8216316	4

pMSE	S_pMSE
0.0497517	1.445132

Two-way Tables Comparison of Synthesised Data (R-Package: synthpop) by (S\_)pMSE

Two-way tables are evaluated based on the original and the synthetic dataset. Here, tables/cells are also evaluated based on pMSE and S\_pMSE (see above). We also present the results for the mean absolute difference in densities (MabsDD).



	pMSE	S_pMSE	df
1.sex:2.sat_v	0.0010044	1.7856143	9
1.sex:3.sat_m	0.0000996	0.1770427	9
1.sex:4.sat_sum	0.0003686	0.6552470	9

	pMSE	S_pMSE	df
1.sex:5.hs_gpa	0.0005434	0.9659932	9
1.sex:6.fy_gpa	0.0009513	1.6912861	9
2.sat_v:3.sat_m	0.0018137	1.2091004	24
2.sat_v:4.sat_sum	0.0021920	1.5942083	22
2.sat_v:5.hs_gpa	0.0021391	1.4260514	24
2.sat_v:6.fy_gpa	0.0020734	1.3822533	24
3.sat_m:4.sat_sum	0.0017624	1.2260144	23
3.sat_m:5.hs_gpa	0.0017728	1.1818827	24
3.sat_m:6.fy_gpa	0.0013155	0.8769708	24
4.sat_sum:5.hs_gpa	0.0015831	1.0554303	24
4.sat_sum:6.fy_gpa	0.0020028	1.3351857	24
5.hs_gpa:6.fy_gpa	0.0017014	1.1342419	24

### Information Loss Measure Proposed by Andrzej Mlodak (R-Package: sdcMicro)

The value of this information loss criterion is between 0 (no information loss) and 1. It is calculated overall and for each variable.

Information.Loss
0.6254697

Individual Distances for Information Loss:

sex	sat_v	sat_m	sat_sum	hs_gpa	fy_gpa
0.479	0.83786	0.8433546	0.8951682	0.3124193	0.3850162

## Tuning and Optimizations

We tested different inputs for the parameter **method**, that enables to specify the algorithm used for synthesising (we stuck with CART in the end). In general, there are plenty of possible optimizations e.g. to the **visit.sequence** and a lot of other parameters (which we have not had time to optimize for). Most things we quickly tested (beyond the **method**) seemed like they did not have too much of an impact.