

Best practices for creating synthetic data: Balancing efficiency, utility, and privacy

Jonathan P. Latner, PhD

November 20, 2023

What is the story (paper idea)?

- What do we know?
 - ▶ According to the literature (Little et al., 2022 (WP); Danker and Ibrahim, 2022), Synthpop appears to be the 'winner,' high levels of utility, privacy, and computational efficiency
 - ▶ CTGAN appears to be the worst performing synthesizer
- What is the problem?
 - ▶ Synthpop
 - ★ Advantages may only be true in low dimensional data
 - ★ Synthpop may not follow the definition of synthetic data
 - ▶ Evaluations
 - ★ Compare packages, not methods
 - ★ Emphasize utility over privacy
- What do we show?
 - ▶ High dimensional data changes the story
 - ▶ We can make a better GAN (utility and efficiency)
 - ▶ 'Real' CART not as good as Synthpop (not done)
 - ▶ Role of privacy (not done)
- Whats the point/conclusion/contribution?
 - ▶ Not sure yet ...

What is synthetic data?

- ‘When Rubin (1993) introduced the idea of fully synthetic data, there was considerable appeal to releasing data that represented “no actual individual’s” responses, and skepticism regarding its feasibility. Subsequent research has adequately demonstrated the feasibility. However the basic question “How much protection does synthetic data methodology provide?” remained largely unanswered.’ (Abowd & Vilhuber, 2008)
- ‘The idea of synthetic data sets is similar. A statistical process is used to extract information from an actual set collected from a set of respondents and is reexpressed as a collection of artificial or synthetic data sets for public consumption. This allows wide dissemination of the informational content of the actual data set and, at the same time, limits the exposure to potential inadvertent or malicious disclosure of sensitive information about the respondents.’ (Raghunathan, 2021)
- A promising alternative to address the trade-off between broad data access and disclosure protection is the release of synthetic data. With this approach, **a model is fitted to the original data and draws from this model are used to replace the original values.** Depending on the desired level of protection, only some records (partial synthesis) or the entire dataset (full synthesis) are replaced by synthetic values. (Drechsler & Haensch, 2023)

What are the trade-offs when creating synthetic data?

- Privacy (risk) vs. utility
 - ▶ How do we measure privacy?
 - ▶ How do we measure utility?
- Computational efficiency (duration in time)
 - ▶ It takes a long time!
 - ▶ Role of hyperparameters results in better output, but takes even longer
- Methods vs. packages
 - ▶ Does the package do what it says (Synthpop)?
 - ▶ Is the package the best version of the method (CTGAN)?

Balancing risk vs. utility

Figure 1: Little et al., 2022 (WP), Table 2

Table 2: Synthetic data risk and utility (mean of 5 datasets), and comparable sample equivalence for each of the Census datasets.⁶

Data	Synthesizer	Overall Utility	Risk (marginal TCAP)	Sample Equiv. for Utility	Sample Equiv. for Risk
UK 1991	<i>CTGAN</i>	0.514	0.371	0.25% - 0.5%	2% - 3%
	<i>Synthpop</i>	0.774	0.516	10% - 20%	10% - 20%
	<i>DataSynthesizer:</i>				
	$\epsilon = 0.1$	0.330	0.043	<0.1%	<0.1%
	$\epsilon = 1$	0.416	0.303	<0.1%	0.1% - 0.25%
	$\epsilon = 10$	0.536	0.424	0.25% - 0.5%	5% - 10%
	No DP	0.643	0.440	1% - 2%	5% - 10%

Key finding: Synthpop is clear 'winner', with highest utility and risk, equivalent to releasing a 10-20% sample

Data dimensions

Table 1: JPL replication of UK 1991

Data	Rows	Columns	NumericVars	NonNumericVars
gb91ind	10427	12	1	11
gb91ind_full	104267	12	1	11

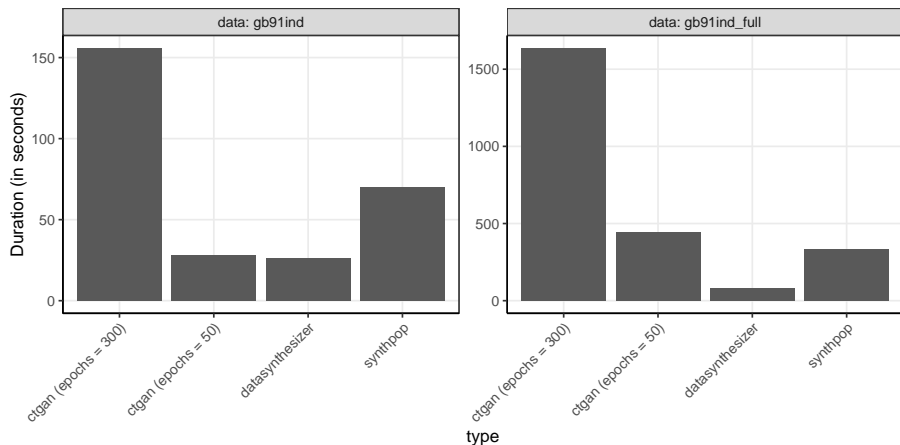
Figure 2: Little et al., 2022 (WP), Table 1

Table 1: Census Data Summary

Dataset	Sample size	#Total Variables	#Categorical	#Numerical
Canada 2011	32149	25	21	4
Fiji 2007	84323	19	18	1
Rwanda 2012	31455	21	20	1
UK 1991	104267	15	13	2

Duration in time (UK 1991)

Figure 3:



Whats the problem with Little et al., 2022 (WP)?

- ① All data sets are low dimensional (rows, obs, variable type)
- ② Privacy and utility overlap (TCAP/CIO from 1 regression)
 - ▶ Disadvantage: 1 regression seems limited
 - ▶ Advantage: Are more regressions better?
 - ▶ Other measure of utility is ROC (univ + bivar)
- ③ Synthpop is compared to Datasynthesizer and CTGAN using baseline hyperparameters
 - ▶ Not comparing apples to apples
- ④ **Synthpop appears to be the best, but this is a function of research choice (data, utility/privacy measures, and packages vs. methods)**

Whats the point?

- Efficiency – Datasynthesizer and CTGAN are more computationally efficient (duration) in data sets with higher dimensions
 - ▶ This matters because tuning is important
- Synthpop still superior with respect to utility, but ...
- Method vs. package
 - ▶ If we make a better GAN, then utility increases
 - ▶ Synthpop has high utility because it does not meet the definition of synthetic data. It does not draw from the model.
- How to measure privacy is still unresolved
 - ▶ Datasynthesizer is only package which provides a measure of privacy that can be adjusted
- **Under these conditions, Synthpop could be the worst synthesizer**

Whats is the goal?

- Compare/contrast synthesizers w/respect to:
 - ▶ Efficiency (computational duration)
 - ▶ Utility (multiple measures)
 - ▶ Privacy (not done yet)
 - ★ How to measure privacy in low and high dimensional data?
 - ★ is TCAP too specific?
 - ▶ Package vs. method
 - ★ CTGAN vs. GANs: Can we make a better GAN?
 - ★ Synthpop vs. CART: What if Synthpop sampled from predicted values?

Efficiency

24 data sets with simulated categorical variables

Data	Rows	Columns	NumericVars	NonNumericVars
sim_categorical_01	1000	10	0	10
sim_categorical_02	1000	10	0	10
sim_categorical_03	1000	10	0	10
sim_categorical_04	1000	10	0	10
sim_categorical_05	1000	15	0	15
sim_categorical_06	1000	15	0	15
sim_categorical_07	1000	15	0	15
sim_categorical_08	1000	15	0	15
sim_categorical_09	1000	20	0	20
sim_categorical_10	1000	20	0	20
sim_categorical_11	1000	20	0	20
sim_categorical_12	1000	20	0	20
sim_categorical_13	5000	10	0	10
sim_categorical_14	5000	10	0	10
sim_categorical_15	5000	10	0	10
sim_categorical_16	5000	10	0	10
sim_categorical_17	5000	15	0	15
sim_categorical_18	5000	15	0	15
sim_categorical_19	5000	15	0	15
sim_categorical_20	5000	15	0	15
sim_categorical_21	5000	20	0	20
sim_categorical_22	5000	20	0	20
sim_categorical_23	5000	20	0	20
sim_categorical_24	5000	20	0	20

5 data sets with simulated continuous data

Data	Rows	Columns	NumericVars	NonNumericVars
sim_continuous_01	50000	10	10	0
sim_continuous_02	50000	15	15	0
sim_continuous_03	50000	20	20	0
sim_continuous_04	100000	10	10	0
sim_continuous_05	100000	15	15	0
sim_continuous_06	100000	20	20	0
sim_continuous_07	200000	10	10	0
sim_continuous_08	200000	15	15	0
sim_continuous_09	200000	20	20	0

5 data sets with benchmarked data (i.e. 'real data')

Data are publicly available data used for benchmarking (Xu et al., 2019)

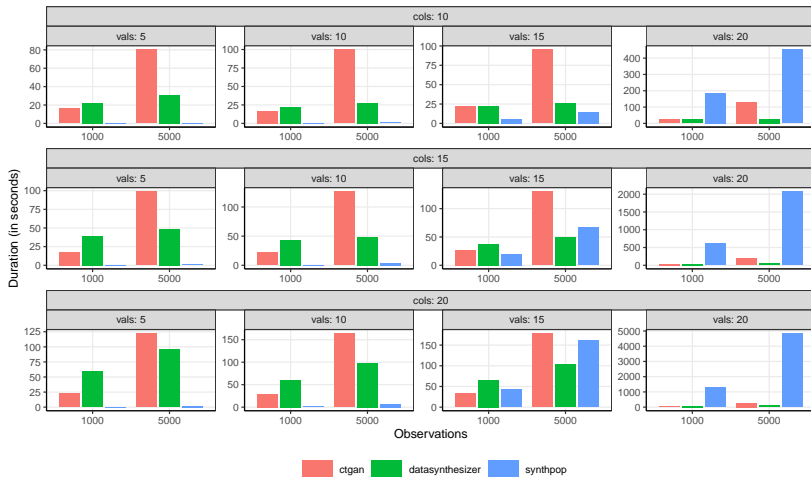
Data	Rows	Columns	NumericVars	NonNumericVars
adult	32561	15	6	9
grid	20000	2	2	0
gridr	20000	2	2	0
sd2011	5000	35	14	21
sd2011_small	5000	4	2	2

- Adult <http://archive.ics.uci.edu/ml/datasets/adult>
- grid: https://docs.sdv.dev/sdv/multi-table-data/data-preparation/loading-data#get_available_demos
- gridr: https://docs.sdv.dev/sdv/multi-table-data/data-preparation/loading-data#get_available_demos
- sd2011 (sd2011_small): synthpop

Categorical data

CTGAN =< Synthpop if (cols >= 20 | vals >= 20)

Figure 4:

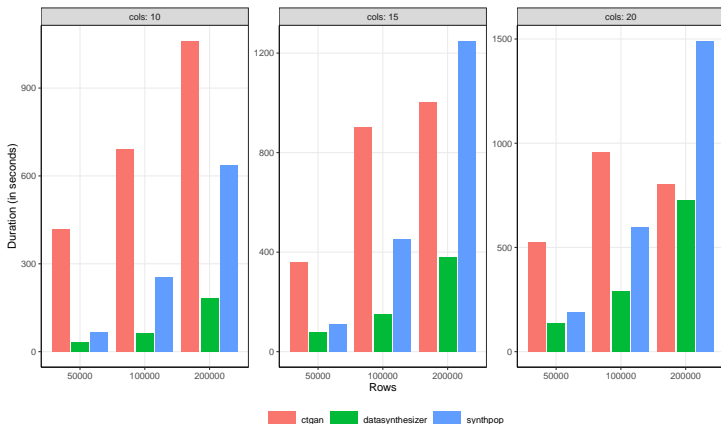


Continuous data

CTGAN =< Synthpop if (cols >= 15 | rows >= 200.000)

Datasynthesizer always the best

Figure 5:

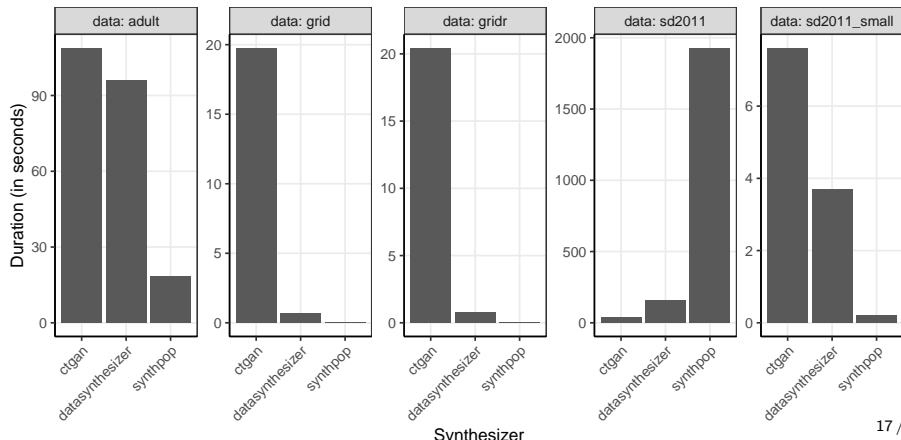


Note: CTGAN hyperparameter: batch size = 1000 and epochs = 50 (see figure 8)

Benchmark data

Synthpop always the best, except SD2011 (full size), which has 2x as many variables as adult, but 1/4th number of rows

Figure 6:



Summary of efficiency

- Synthpop is fastest in low dimensional data
- It does not take very high levels of dimensionality for CTGAN/Datasynthesizer to be faster
- In high dimensional data, datasynthesizer better than CTGAN

CTGAN vs. GANs

Figure 7: Duration (CTGAN) w/continuous data

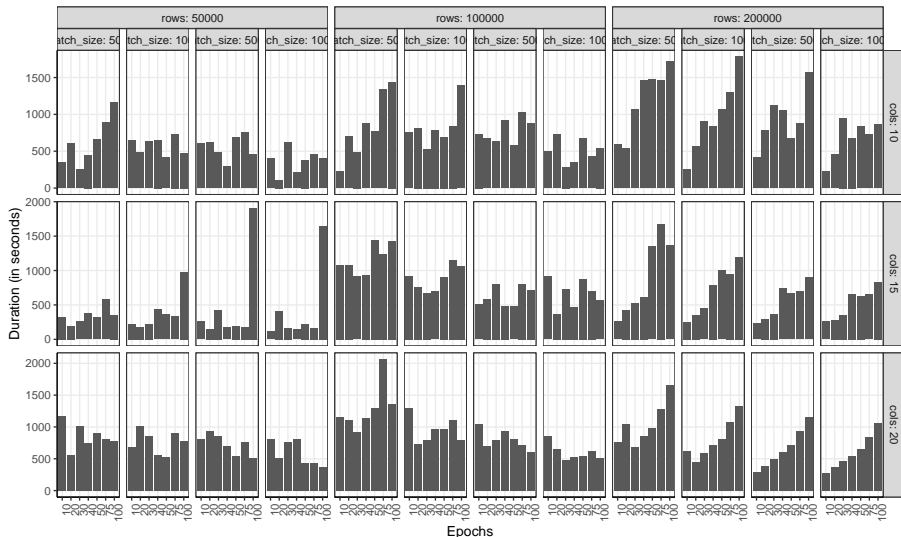
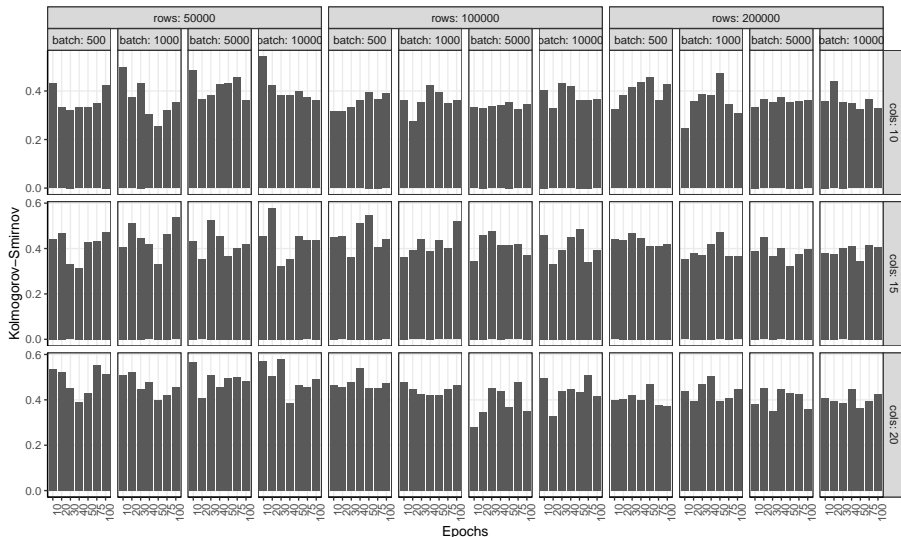


Figure 8: Kolmogorov-Smirnov utility measure (CTGAN)



Summary of GANs

- In CTAN, baseline is 300 epochs
- In single table, utility does not really change/improve after 50 epochs
- Utility of CTGAN is still worse than Datasynthesizer/Synthpop
- Can we create a better GAN?
 - ▶ answer: yes we can (Neuenhoffer)

Synthpop vs. cart

Methodology

<https://www.synthpop.org.uk/about-synthpop.html#methodology>

Consider as an example a default synthesis, i.e. synthesis with all values of all variables (Y_1, Y_2, \dots, Y_p) to be replaced. The first variable to be synthesised Y_1 cannot have any predictors and therefore its synthetic values are generated by random sampling with replacement from its observed values. **Then the distribution of Y_2 conditional on Y_1 is estimated and the synthetic values of Y_2 are generated using the fitted model and the synthesised values of Y_1 .** Next the distribution of Y_3 conditional on Y_1 and Y_2 is estimated and used along with synthetic values of Y_1 and Y_2 to generate synthetic values of Y_3 and so on. The distribution of the last variable Y_p will be conditional on all other variables. Similar conditional specification approaches are used in most implementations of synthetic data generation. They are preferred to joint modelling not only because of the ease of implementation but also because of their flexibility to apply methods that take into account structural features of the data such as logical constraints or missing data patterns.

This is consistent with the definition of synthetic data (Drechsler and Haensch, 2023)

Synthpop (syn.cart)

<https://rdrr.io/cran/synthpop/src/R/functions.syn.r>

```
1 synth.cart <- function(y, x, xp, smoothing = "", proper = FALSE,
2                       minbucket = 5, cp = 1e-08, ...)
3   fit <- rpart(y ~ ., data = as.data.frame(cbind(y, x)), method = "anova",
4             minbucket = minbucket, cp = cp, ...)
5   # get leaf number for observed data
6   leafnr <- floor(as.numeric(row.names(fit$frame[fit$where,])))
7   # replace yval with leaf number in order to predict later node number
8   # rather than yval (mean y for observations classified to a leaf)
9   fit$frame$yval <- as.numeric(row.names(fit$frame))
10  # predict leaf number
11  nodes <- predict(object = fit, newdata = xp)
12
13  ...
14
15  uniquenodes <- unique(nodes)
16  new <- vector("numeric", nrow(xp))
17  for (j in uniquenodes) {
18    donors <- y[leafnr == j] # values of y in a leaf
19    new[nodes == j] <- resample(donors, size = sum(nodes == j), replace = TRUE)
20  }
21
22
```

Method and code may not be the same

The code appears to indicate that the synthetic sample is drawn from the observed data in a predicted leaf, rather than the predicted y value.

Is this different than the methodological description?

Does this violate the definition of synthetic data, described above?

The point (and question): High levels of utility in synthpop may be the result of code that implements the method in a way that is not consistent with the definition of synthetic data

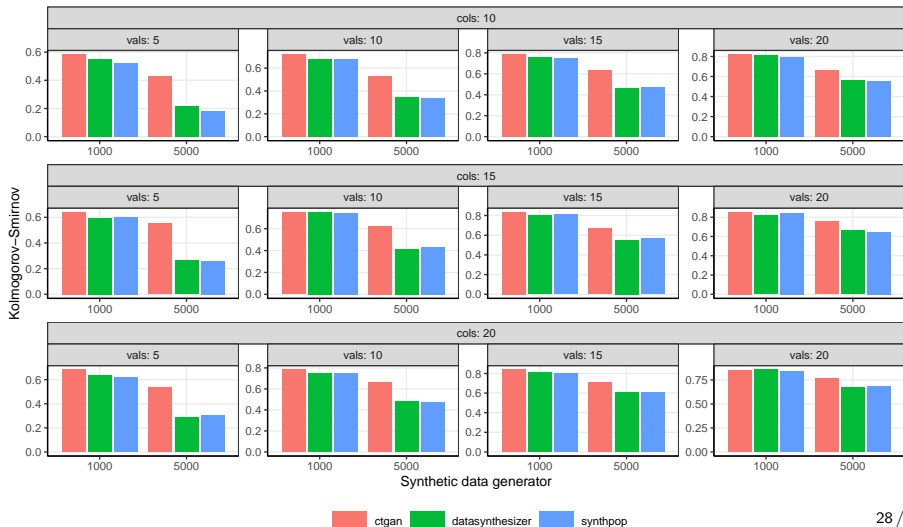
If true, then utility in Synthpop may be artificially high (which lowers the relative advantage to other data synthesizers)

More research is needed

Utility by synthesizer and data type

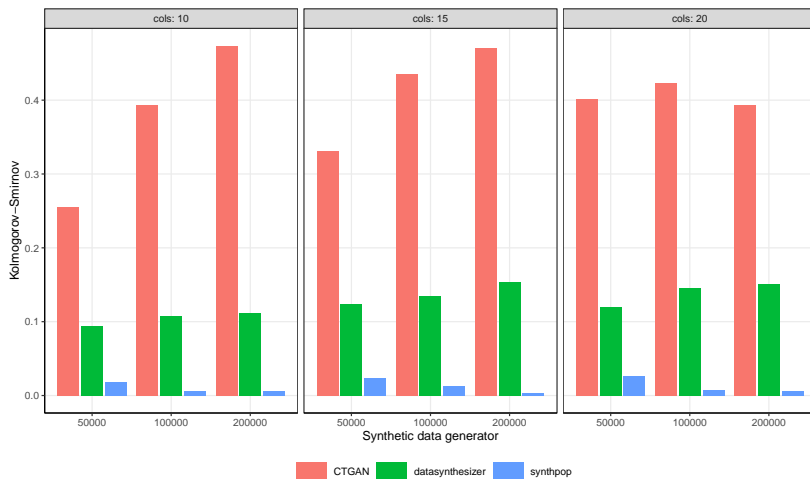
Categorical data

Figure 9:



Continuous data

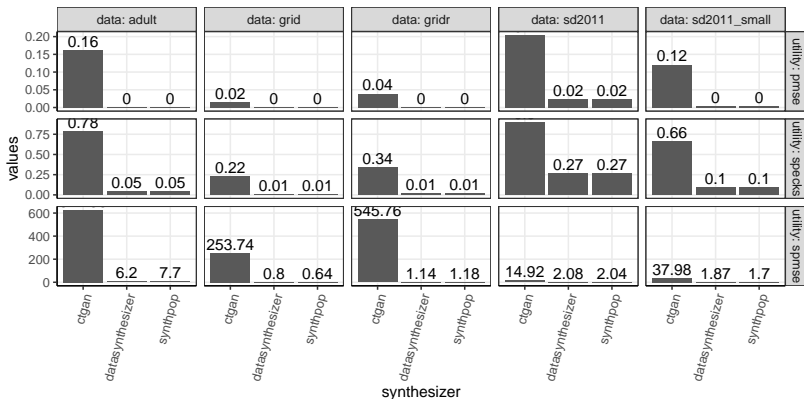
Figure 10:



Note: CTGAN hyperparameter: batch size = 1000 and epochs = 50 (see figure 8)

Benchmark data

Figure 11:



Privacy (not done yet)

Conclusion

What have we learned?

- Data dimensionality and Computational efficiency (duration in time)
 - ▶ Important because analyzing synthesizers takes lots of time
 - ▶ Synthpop
 - ★ In low dimensional data, synthpop performs well
 - ★ How does Synthpop get such high levels of utility? Sampling values vs. leafs?
 - ▶ CTGAN (GANs)
 - ★ In high dimensional data, has high efficiency, but still low utility
 - ★ Synthesizing single tables may not need many epochs (i.e. even better efficiency, even in low-dimensional data)
 - ★ CTGAN may not be a very good GAN for synthesizing single tables
 - ★ Can we make a better GAN?
 - ▶ Datasynthesizer appears to be high performing in both low and high-dimensional data

Next steps

- How do we measure privacy?
 - ▶ TCAP
 - ★ What about data with < 5 variables
 - ★ What about if we don't know the data
 - ★ Is 1 regression enough?
- How do we measure utility?
 - ▶ Datasynthesizer package has measures