Package ‘DPSynth’

September 28, 2026

Type Package

Title Differentially Private Synthetic Data with Guaranteed Utility

Version 0.1.0

Maintainer Mukul Bijalwan <mukulbijalwan555@gmail.com>

Description Differentially private (DP) synthetic data generation for tabular data. Provides DP Gaussian mixture models, DP Gaussian copulas, DP histogram marginals and a Private Aggregation of Teacher Ensembles (PATE) synthesizer for mixed-type data, together with a standardized utility evaluation framework (univariate fidelity, propensity score MSE, multivariate dependence, downstream task performance), empirical disclosure risk auditing (membership inference, attribute disclosure, record linkage) and privacy budget accounting (basic, advanced and Renyi DP composition). The implementation follows Dwork and Roth (2014) <doi:10.1561/0400000042> and Dwork et al. (2006) <doi:10.1007/11681878_14> for the DP mechanisms, Papernot et al. (2017) <doi:10.1145/3133956.3133982> for the PATE synthesizer, and Woo et al. (2009) <doi:10.2202/1557-4679.1203> for the disclosure risk evaluation framework.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

Imports stats, utils, MASS

Suggests testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

LazyData true

Config/testthat/edition 3

URL https://github.com/MukulBijalwan/DPSynth

BugReports https://github.com/MukulBijalwan/DPSynth/issues

NeedsCompilation no

Author Mukul Bijalwan [aut, cre],
Gunjan Aggarwal [aut],
Mukul Jain [aut]

Repository CRAN

Date/Publication 2026-09-28 08:00:08 UTC

Contents

`acs_pums_sample` *ACS PUMS-like sample*

Description

A synthetic microdata sample modeled on American Community Survey PUMS structure, used as an example in **DPSynth**.

Usage

```
acs_pums_sample
```

Format

A data frame with 500 rows and 5 variables:

age age in years

income annual income in USD

educ educational attainment

married married indicator

weeks_worked weeks worked last year

`adult_sample` *Anonymized UCI Adult subsample*

Description

A synthetic, anonymized subsample modeled on the UCI Adult census-income dataset, used as an example throughout **DPSynth**. No real personal records are included.

Usage

```
adult_sample
```

Format

A data frame with 500 rows and 8 variables:

age age in years

workclass employment sector

education_num years of education

marital_status marital status

hours_per_week working hours per week

sex sex

capital_gain capital gain in USD

income binary income class

audit_attribute_disclosure *Attribute disclosure risk audit*

Description

Assumes an attacker knows `known_vars` for a target record and uses nearest neighbours in the synthetic data to infer `sensitive_var`. Reports the normalized prediction error and the rate of high-confidence disclosures.

Usage

```
audit_attribute_disclosure(
  synth,
  orig,
  known_vars = NULL,
  sensitive_var = NULL,
  n_neighbors = 5
)
```

Arguments

`synth`	synthetic data frame
`orig`	original data frame
`known_vars`	column names the attacker observes; default: all common columns except `sensitive_var`
`sensitive_var`	sensitive column to predict; default: last column
`n_neighbors`	number of synthetic neighbours used

Value

list with `mean_disclosure_error` and `high_risk_rate`

audit_linkage_risk *Record linkage risk audit*

Description

Estimates how many real records can be uniquely re-identified by matching against the synthetic data: a real record is "linked" when its nearest synthetic neighbour is closer than a quantile threshold of the within-original distance distribution.

Usage

```
audit_linkage_risk(synth, orig, threshold_quantile = 0.05)
```

Arguments

synth — synthetic data frame

orig — original data frame

threshold_quantile — quantile of within-original NN distances used as the linkage threshold

Value

list with `linkage_rate`, `linked_records`, `threshold`

`audit_membership_risk` *Membership inference risk audit*

Description

Simulates the simplest membership inference attack: for each original record, compute the distance to its nearest synthetic record. Records whose nearest synthetic neighbour is unusually close are considered at risk of a correct "member" verdict.

Usage

```
audit_membership_risk(synth, orig, method = "nearest_neighbor")
```

Arguments

synth — synthetic data frame

orig — original data frame

method — currently only `"nearest_neighbor"`

Value

list with `risk_score`, `mean_min_distance` and `high_risk_records`

check_synth_budget *Check remaining privacy budget*

Description

Check remaining privacy budget

Usage

```
check_synth_budget(budget)
```

Arguments

budget	a synth_privacy_budget object

Value

named vector with remaining_epsilon, spent_epsilon and fraction_used; also prints a summary

clip_data *Clip data to bounds*

Description

Clamp every column of a data frame (or matrix) to user-specified bounds. Clamping bounds sensitivity and is a prerequisite for calibrated DP noise.

Usage

```
clip_data(data, bounds)
```

Arguments

data	data frame or matrix
bounds	named list with elements lower and upper (numeric vectors recycled across columns), or a 2-row matrix [lower; upper].

Value

data frame with values clipped to the bounds

dp_copula_synth *DP Gaussian copula synthesis*

Description

Separates the synthesis problem into (1) DP marginal distributions – per-column histograms released with Laplace noise – and (2) a DP dependence structure – a Gaussian copula correlation matrix estimated on rank-transformed pseudo-observations, perturbed with Gaussian noise and projected onto the positive-definite cone. Generation inverts the DP marginals through the DP copula (post-processing).

Usage

```
dp_copula_synth(
  data,
  n_synth = nrow(data),
  epsilon = 1,
  delta = 1e-06,
  n_bins = 20,
  copula_type = "gaussian"
)
```

Arguments

`data`	data frame (numeric, factor or character columns)
`n_synth`	number of synthetic records
`epsilon`	total privacy parameter epsilon
`delta`	privacy parameter delta
`n_bins`	number of histogram bins for numeric marginals
`copula_type`	currently only `"gaussian"`

Value

object of class `dp_copula` / `dp_synthetic`

dp_gmm_synth *DP Gaussian Mixture Model synthesis*

Description

Fits a Gaussian mixture model with diagonal covariances using the perturb-and-postprocess paradigm: sufficient statistics (component counts, means, variances) are perturbed with calibrated Laplace / Gaussian noise so that the released model satisfies (epsilon, delta)-DP. Sampling from the fitted model is post-processing and consumes no additional privacy budget.

Usage

```
dp_gmm_synth(
  data,
  n_synth = nrow(data),
  K = 3,
  epsilon = 1,
  delta = 1e-06,
  bounds = NULL,
  max_iter = 20
)
```

Arguments

`data`	data frame of purely numeric columns
`n_synth`	number of synthetic records to generate
`K`	number of mixture components
`epsilon`	privacy parameter epsilon (> 0)
`delta`	privacy parameter delta (in (0, 1))
`bounds`	optional list with `lower` / `upper` vectors used to clamp the data; by default taken from the data range (note: data-driven bounds are not themselves DP – supply public bounds for a formal guarantee)
`max_iter`	maximum EM iterations

Value

object of class `dp_gmm` / `dp_synthetic` with the synthetic data, model parameters and privacy parameters

---

`dp_marginals_synth` *DP marginal synthesis*

Description

Synthesizes each column independently from a differentially private histogram (numeric columns) or DP multinomial (categorical columns). No dependence structure is preserved; useful as a fast baseline and for consistent-marginal releases in official statistics.

Usage

```
dp_marginals_synth(data, n_synth = nrow(data), epsilon = 1, n_bins = 20)
```

Arguments

`data`	data frame
`n_synth`	number of synthetic records
`epsilon`	privacy parameter epsilon (split evenly across columns)
`n_bins`	number of bins for numeric columns

Value

object of class `dp_marginals` / `dp_synthetic`

`dp_pate_synth` *PATE synthetic data for discrete / mixed-type data*

Description

Partitions the private data into disjoint teacher datasets, fits a teacher model on each partition (no privacy cost), and releases each per-record query through a noisy aggregation of teacher votes. The privacy cost scales with the number of released records.

Usage

```
dp_pate_synth(data, n_synth = nrow(data), epsilon = 1, n_teachers = 10)
```

Arguments

`data`	data frame
`n_synth`	number of synthetic records
`epsilon`	total privacy parameter (spent across all released records via basic composition)
`n_teachers`	number of disjoint teacher partitions

Value

object of class `dp_pate` / `dp_synthetic`

`dp_synthesize` *Differentially private synthetic data generation*

Description

Top-level orchestrator that dispatches to a DP synthesizer, optionally evaluates utility and audits disclosure risk, and attaches a privacy report.

Usage

```
dp_synthesize(
  data,
  method = "copula",
  n_synth = nrow(data),
  epsilon = 1,
  delta = 1e-06,
  utility_eval = TRUE,
  risk_audit = TRUE,
  ...
)
```

Arguments

`data`	original data frame
`method`	one of `"copula"`, `"gmm"`, `"marginals"`, `"pate"`
`n_synth`	number of synthetic records
`epsilon`	privacy parameter
`delta`	privacy parameter
`utility_eval`	run `evaluate_utility`?
`risk_audit`	run disclosure-risk audits?
`...`	further arguments passed to the underlying synthesizer

Value

object of class `dp_synthesis_result`

See Also

`dp_copula_synth`, `dp_gmm_synth`, `dp_marginals_synth`, `dp_pate_synth`

Examples

```
set.seed(1)
dat <- data.frame(x = rnorm(200), y = rnorm(200) + 0.5 * rnorm(200))
res <- dp_synthesize(dat, method = "copula", epsilon = 2)
print(res)
```

`evaluate_downstream` *Downstream task performance (TSTR)*

Description

Trains the same linear model on the synthetic and the original data and compares out-of-sample RMSE on held-out real data. A `utility_ratio` close to 1 indicates good analytic utility.

Usage

```
evaluate_downstream(
  synth,
  orig,
  target_var,
  predictors = NULL,
  test_prop = 0.2
)
```

Arguments

`synth`	synthetic data frame
`orig`	original data frame
`target_var`	name of the numeric target column
`predictors`	optional predictor column names; default all others
`test_prop`	held-out fraction of the original data

Value

list with RMSEs and the utility ratio

`evaluate_multivariate` *Multivariate fidelity evaluation*

Description

Compares the correlation structure (numeric columns) of original and synthetic data via the relative Frobenius-norm distance between correlation matrices and Gaussian-approximated mutual information.

Usage

```
evaluate_multivariate(synth, orig)
```

Arguments

`synth`	synthetic data frame
`orig`	original data frame

Value

list with correlation distance and mutual-information summaries

`evaluate_propensity` *Propensity score utility (pMSE)*

Description

Fits a logistic model to distinguish synthetic from original records. Under perfect synthesis the expected pMSE is $c(1 - c)$ with $c = n_synth/(n_orig + n_synth)$; the reported ratio is 0 for indistinguishable data and grows as fidelity degrades.

Usage

```
evaluate_propensity(synth, orig)
```

Arguments

`synth`	synthetic data frame
`orig`	original data frame

Value

list with `pMSE`, `ratio` (standardised) and `c`

`evaluate_univariate` *Univariate fidelity evaluation*

Description

Compares original and synthetic columns with Kolmogorov-Smirnov statistics, Hellinger distance (histogram-based for numeric columns, Jensen-Shannon divergence for categorical columns) and relative moment errors.

Usage

```
evaluate_univariate(synth, orig)
```

Arguments

`synth`	synthetic data frame
`orig`	original data frame

Value

data frame of per-variable fidelity metrics

`evaluate_utility` *Standardized utility evaluation*

Description

Runs any combination of univariate, multivariate, propensity (pMSE) and downstream (TSTR) fidelity metrics and returns a `dp_utility_report`.

Usage

```
evaluate_utility(
  synthetic_data,
  original_data,
  metrics = c("univariate", "multivariate", "propensity"),
  target_var = NULL
)
```

Arguments

`synthetic_data`	synthetic data frame
`original_data`	original data frame
`metrics`	character vector of metric groups to run
`target_var`	optional target for the downstream metric

Value

object of class `dp_utility_report`

`gaussian_mech` *Gaussian mechanism*

Description

Release a statistic under (epsilon, delta)-DP by adding Gaussian noise calibrated to the L2 sensitivity via the classical analytic bound $\sigma = \Delta\sqrt{2\log(1.25/\delta)}/\epsilon$.

Usage

```
gaussian_mech(value, sensitivity, epsilon, delta = 1e-06)
```

Arguments

`value`	numeric statistic (scalar or vector)
`sensitivity`	L2 sensitivity of the statistic
`epsilon`	privacy parameter epsilon (> 0)
`delta`	privacy parameter delta (in (0, 1))

Value

numeric vector: noisy release of `value`

`new_synth_budget` *Create a privacy budget tracker*

Description

Create a privacy budget tracker

Usage

```
new_synth_budget(
  epsilon,
  delta = 0,
  accounting = c("basic", "advanced", "rdp")
)
```

Arguments

`epsilon`	total epsilon available
`delta`	total delta available
`accounting`	one of `"basic"`, `"advanced"`, `"rdp"`

Value

object of class `synth_privacy_budget`

`print.dp_synthesis_result` *Print a synthesis result*

Description

Print a synthesis result

Usage

```
## S3 method for class 'dp_synthesis_result'
print(x, ...)
```

Arguments

`x`	a `dp_synthesis_result`
`...`	unused

Value

No return value, called for side effects. The input `x` is returned invisibly.

`rlaplace` *Laplace noise*

Description

Draw `n` independent Laplace(`location`, `scale`) variates using inverse-CDF sampling. The Laplace mechanism adds `Lap(sensitivity / epsilon)` noise to achieve pure epsilon-DP.

Usage

```
rlaplace(n, location = 0, scale = 1)
```

Arguments

`n`	number of draws
`location`	location parameter
`scale`	scale parameter (must be > 0)

Value

numeric vector of length `n`

`spend_synth` *Spend privacy budget on a step*

Description

Records a DP release consuming `epsilon` (and optionally `delta`) and updates the accumulated spend under the configured composition rule. Issues a warning when the total budget is exceeded.

Usage

```
spend_synth(budget, epsilon, delta = 0, description = "")
```

Arguments

`budget`	a `synth_privacy_budget` object
`epsilon`	epsilon spent by this step
`delta`	delta spent by this step
`description`	free-text description of the step

Value

updated budget object

Index