

# Package: PUcopulaSynth (via r-universe)

October 7, 2026

**Type** Package

**Title** Partition-of-Unity Copula Fitting and Synthesis for R

**Version** 0.1.0

**Description** Fit multivariate distributions using a Partition-of-Unity copula dependence structure, estimate marginals, and generate synthetic data with factor pre/post-processing.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.3

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**URL** <https://github.com/amaendle/PUcopulaSynth>

**BugReports** <https://github.com/amaendle/PUcopulaSynth/issues>

**Imports** caret, PUcopula, logspline, dplyr, RANN, tidyr, tidyselect, stats

**Additional\_repositories** <https://amaendle.r-universe.dev/>

**Config/pak/sysreqs** libicu-dev

**Repository** <https://amaendle.r-universe.dev>

**Date/Publication** 2026-04-10 13:58:19 UTC

**RemoteUrl** <https://github.com/amaendle/PUcopulaSynth>

**RemoteRef** HEAD

**RemoteSha** f3ea55489dcf96ef1fa49ecc9383268ac2938025

Contents

|                                  |          |
|----------------------------------|----------|
| estimateMarginals . . . . .      | 2        |
| fitPUCopula . . . . .            | 3        |
| generateSynthetic . . . . .      | 3        |
| knnsmoother . . . . .            | 4        |
| postprocessData . . . . .        | 5        |
| preprocessData . . . . .         | 5        |
| save_original_varnames . . . . . | 6        |
| simulateCopula . . . . .         | 6        |
| <b>Index</b>                     | <b>7</b> |

---

|                   |                                 |
|-------------------|---------------------------------|
| estimateMarginals | <i>Estimate marginal models</i> |
|-------------------|---------------------------------|

---

Description

Fits logspline marginals for numeric/ordered variables, and empirical probability tables for binary or trivial variables. Optional k-NN smoothing is applied to numeric columns.

Usage

```
estimateMarginals(  
  data,  
  method = "spline",  
  k = NULL,  
  lbound = NULL,  
  ubound = NULL  
)
```

Arguments

|        |                                                                                   |
|--------|-----------------------------------------------------------------------------------|
| data   | Preprocessed data.frame                                                           |
| method | Character; for numeric and ordered factors (currently "spline")                   |
| k      | Numeric scalar, vector, or named list for k-NN smoothing                          |
| lbound | lower boundary of distribution passed to <a href="#">logspline::logspline()</a> . |
| rbound | upper boundary of distribution passed to <a href="#">logspline::logspline()</a> . |

Value

A named list of marginal models (each element has qfun)

---

fitPUCopula

*Fit a PUCopula model*


---

### Description

Fits a PUCopula model on a (preprocessed) data matrix with optional rank-binning and jitter for numeric variables.

### Usage

```
fitPUCopula(
  data,
  driver_strength_factor = 0.5,
  bin_size = 3,
  jitter = FALSE,
  family = "binom"
)
```

### Arguments

|                        |                                                                           |
|------------------------|---------------------------------------------------------------------------|
| data                   | Preprocessed data.frame (e.g., preprocessData()\$data)                    |
| driver_strength_factor | Numeric scalar or vector in (0,1] used to scale rows per variable         |
| bin_size               | Numeric scalar, vector, or named list with bin sizes                      |
| jitter                 | FALSE, numeric (single) or named list mapping variables to jitter factors |
| family                 | PUCopula family, e.g. "binom" or "nbinom"                                 |

### Value

A PUCopula::PUCopula model

---

generateSynthetic

*Generate synthetic data*


---

### Description

Combines a fitted PUCopula and marginal models to produce a synthetic data.frame. Optionally restores original factor structure, names and classes.

**Usage**

```
generateSynthetic(
  n,
  copula,
  marginals,
  original_levels = NULL,
  original_varnames = NULL,
  original_classes = NULL
)
```

**Arguments**

`n` Integer, number of rows to generate

`copula` A PUCopula model

`marginals` List of marginals from `estimateMarginals()`

`original_levels` Optional `preprocessData()$original_levels`

`original_varnames` Optional vector of original column names

`original_classes` Optional named vector of original classes

**Value**

Synthetic data.frame

---

|             |                                                                     |
|-------------|---------------------------------------------------------------------|
| knnsmoother | <i>k-NN smoother for numeric vectors (no DataSHIELD thresholds)</i> |
|-------------|---------------------------------------------------------------------|

---

**Description**

k-NN smoother for numeric vectors (no DataSHIELD thresholds)

**Usage**

```
knnsmoother(x, k = 3)
```

**Arguments**

`x` Numeric vector (NAs allowed).

`k` Integer neighbours in  $\setminus[1, N-1]$  for non-missing values.

**Value**

Numeric vector of same length with smoothed non-missing entries.

---

|                 |                                                          |
|-----------------|----------------------------------------------------------|
| postprocessData | <i>Restore original factor structure after synthesis</i> |
|-----------------|----------------------------------------------------------|

---

**Description**

Restore original factor structure after synthesis

**Usage**

```
postprocessData(data, cat_dummy_levels)
```

**Arguments**

|                  |                                                                  |
|------------------|------------------------------------------------------------------|
| data             | A data.frame containing dummy/encoded columns                    |
| cat_dummy_levels | The dummies element returned by <a href="#">preprocessData()</a> |

**Value**

A data.frame with factors restored and columns ordered like input

---

|                |                                              |
|----------------|----------------------------------------------|
| preprocessData | <i>Preprocess data before copula fitting</i> |
|----------------|----------------------------------------------|

---

**Description**

Converts multi-level unordered factors to dummy variables and tags remaining factor columns with '.oriname' while storing levels.

**Usage**

```
preprocessData(data)
```

**Arguments**

|      |              |
|------|--------------|
| data | A data.frame |
|------|--------------|

**Value**

A list with data (processed) and original\_levels (dummies + oriname)

---

`save_original_varnames`*Save original names and classes*

---

**Description**

Save original names and classes

**Usage**`save_original_varnames(data)``save_original_classes(data)`**Arguments**

`data`                      A data.frame

**Value**

Character vector (names) / Named character vector (classes)

---

`simulateCopula`*Draw from a fitted PUCopula*

---

**Description**

Draw from a fitted PUCopula

**Usage**`simulateCopula(model, n)`**Arguments**

`model`                      A PUCopula model from `fitPUCopula()`

`n`                              Number of rows

**Value**

Matrix of U(0,1) draws

# Index

`estimateMarginals`, [2](#)  
`estimateMarginals()`, [4](#)  
  
`fitPUCopula`, [3](#)  
`fitPUCopula()`, [6](#)  
  
`generateSynthetic`, [3](#)  
  
`knnsmoother`, [4](#)  
  
`logspline::logspline()`, [2](#)  
  
`postprocessData`, [5](#)  
`preprocessData`, [5](#)  
`preprocessData()`, [5](#)  
  
`save_original_classes`  
    `(save_original_varnames)`, [6](#)  
`save_original_varnames`, [6](#)  
`simulateCopula`, [6](#)