Package 'matchednull'

November 2, 2026

Title Matched-Null Tests for Cluster-Count Claims

Version 0.2.2

Description Builds matched nulls for cluster-count claims: synthetic twins of a dataset that preserve every marginal distribution exactly and the correlation matrix approximately, while containing no cluster structure by construction. A reported number of clusters or ``types'' can then be tested against what the data's own margins and covariance already produce, using any clustering pipeline. A t-copula option is a sensitivity check that adds tail dependence to the reference, so an apparent excess of clusters can be rerun against a heavier-tailed reference; neither outcome of that rerun establishes that types exist. Implements the matched-null procedure of Meng (2026) ``Types Without Taxa'' <doi:10.17605/OSF.IO/2EKCG>.

License MIT + file LICENSE

Encoding UTF-8

Imports stats

Suggests mclust, testthat (>= 3.0.0), knitr, rmarkdown, future.apply, future

VignetteBuilder knitr

Config/testthat/edition 3

URL https://haomeng797-ship-it.github.io/matchednull/, https://github.com/haomeng797-ship-it/matchednull

BugReports https://github.com/haomeng797-ship-it/matchednull/issues

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Miura Meng [aut, cre] (ORCID: <https://orcid.org/0009-0004-1522-1997>)

Maintainer Miura Meng <haomeng797@gmail.com>

Repository CRAN

Date/Publication 2026-11-02 01:00:02 UTC

Contents

copula_null 2
matched_null_test 3

Index **7**

`copula_null` *Draw one matched null ("null twin") of a dataset*

Description

Builds a synthetic twin of `x` that preserves every marginal distribution exactly and reproduces the correlation matrix approximately, while containing no cluster structure by construction. Each column of the twin is a permutation of the corresponding column of `x`, so no marginal test can tell them apart. The correlations are only approximately reproduced: the observed Pearson correlation matrix serves as the latent correlation of the copula, and once the non-normal empirical margins are restored by rank reordering the output Pearson correlations are not guaranteed to equal it. The gap is not sampling noise alone: it carries a random component from the finite draw and, for skewed or heavy-tailed margins, a systematic component as well. With `copula = "gaussian"` (the default) all dependence in the twin is Gaussian. Clustering found in real data but not in its twins indicates structure this reference does not carry; clustering found equally in both is no detected excess under this reference, which is not evidence that clusters are absent.

Usage

```
copula_null(x, copula = c("gaussian", "t"), df = 8, ridge = 1e-06)
```

Arguments

`x`	A numeric matrix or data frame (rows = observations, columns = variables), complete cases only, at least two rows and two columns.
`copula`	Dependence family of the twin: `"gaussian"` (default) or `"t"`. The Gaussian family is the reference model of the method; `"t"` is a sensitivity variant with heavier-tailed dependence, run alongside the Gaussian twins rather than in place of them, and it does not by itself establish or rule out clusters.
`df`	Degrees of freedom of the t copula, a single positive number (default 8; used only when `copula = "t"`). Smaller is heavier-tailed; `df = 3` is a heavy-tailed variant, `df = 8` a moderate one.
`ridge`	Small value added to the diagonal of the correlation matrix only if it is not positive definite (default `1e-6`).

Details

The construction is a rank reordering in the tradition of Iman and Conover (1982): draw a Gaussian sample with the data's correlation matrix as its latent correlation, then replace each column with the sorted real values laid down in the rank order of the Gaussian column. Every real value is reused

exactly once per column, which is why the margins match exactly. When the observed column has no ties the sorted values are laid down in a strictly increasing map of the Gaussian column, so the twin's rank correlations are those of the Gaussian sample. Tied observed values are laid down together, the twin carries those ties, and the rank pattern then depends on how the ties are broken, so the equality no longer holds. The twin's Pearson correlations are close to, but not in general equal to, the observed ones.

With `copula = "t"` the same construction is driven by a multivariate t sample instead: margins are preserved exactly and correlations approximately as before, but the twin also carries tail dependence, so extreme values across variables arrive together, the more strongly the smaller `df`. A t twin is still a single population with no clusters. It is a sensitivity variant of the reference model with heavier-tailed dependence: a departure from the Gaussian twins that the t twins reproduce is consistent with tail dependence, and one they do not reproduce is harder to attribute to tails. Neither outcome by itself establishes or rules out clusters.

Value

A numeric matrix of the same dimensions as `x`: one matched-null draw. Each column is a permutation of the corresponding column of `x`, so the margins are exact; the correlations are approximate.

References

Iman, R. L., & Conover, W. J. (1982). A distribution-free approach to inducing rank correlation among input variables. *Communications in Statistics - Simulation and Computation, 11*(3), 311-334.

Examples

```
set.seed(1)
x <- matrix(rnorm(200 * 3), 200, 3) %*% chol(matrix(c(1, .5, .3,
                                                      .5, 1, .4,
                                                      .3, .4, 1), 3, 3))
twin <- copula_null(x)
# margins identical:
all(sort(twin[, 1]) == sort(x[, 1]))
# correlations: close, not exact
round(cor(x) - cor(twin), 2)

# a heavier-tailed twin, as a sensitivity variant:
stress <- copula_null(x, copula = "t", df = 3)
all(sort(stress[, 1]) == sort(x[, 1]))
```

`matched_null_test` *Test a cluster count against matched nulls*

Description

Runs the user's own clustering pipeline on the real data and on R matched-null twins of it (see copula_null()), and asks whether the real result exceeds what the twins produce. The twins are a reference model fitted to the data: every twin column is a permutation of the observed values, so the margins are preserved exactly; the correlations are reproduced approximately (see copula_null() for why); and the twins contain no cluster structure by construction. The pipeline is supplied as a function, so any procedure that returns a number, a BIC-selected mixture, a k-means heuristic, a published typology's own workflow, can be tested unchanged.

Usage

```
matched_null_test(
  x,
  cluster_fn,
  R = 200,
  probs = c(0.025, 0.975),
  copula = c("gaussian", "t"),
  df = 8,
  ridge = 1e-06,
  parallel = FALSE
)
```

Arguments

x	A numeric matrix or data frame, complete cases only.
cluster_fn	A function taking a data matrix and returning one finite number: the statistic to be tested, typically the selected number of clusters (but any scalar summary of clustering strength works). A return value that is missing, non-finite, non-numeric, or not of length one is an error, on the real data and on every twin.
R	Number of matched-null twins to draw (default 200). The result always holds exactly R twin statistics.
probs	Lower and upper quantiles of the twin statistics that define the reference interval (default c(.025, .975)).
copula, df	Dependence family of the twins and t degrees of freedom, passed to copula_null(). Defaults to "gaussian", which is the reference model of the method; copula = "t" is a sensitivity variant with heavier-tailed dependence, run in addition to the Gaussian test rather than in place of it, and df is fixed by the caller, never estimated from the data.
ridge	Passed to copula_null().
parallel	Evaluate the twin loop through **future.apply**, so that the backend is whatever future::plan() the caller has set. Because the twins are independent and the null construction itself is a negligible share of the cost, the speed-up is close to the cost of the user's own pipeline divided by the number of workers. Seeded results are reproducible and do not depend on how many workers run them. They do, however, differ from the parallel = FALSE stream, which is left byte-identical to earlier versions, so a seeded analysis should not switch settings part-way through.

Details

`cluster_fn` must return one finite number on every call. If it returns `NA`, `NaN`, `Inf`, a non-numeric value, or a vector of the wrong length, on the real data or on any twin, the test stops with an error naming the offending call. Invalid twins are never dropped and `R` is never quietly reduced, because an exceedance computed from a subset of the twins would not be the one that was asked for. Errors raised inside `cluster_fn` propagate unchanged.

`p_exceed` is a plug-in Monte Carlo exceedance quantity: one plus the number of twins whose statistic reaches the real value, divided by `R + 1`, under a reference fitted to the same data. It is not an exact finite-sample p-value. The reference's correlations are estimated from `x` and reproduced only approximately, so `p_exceed` is a comparison against that fitted reference, not a test with guaranteed size.

Reading the outcome. A real statistic above the twins' interval means the data depart from this reference model, one that carries their margins, approximately their correlations, and no cluster structure. It does not by itself establish that clusters exist: structure the reference does not carry, heavy-tailed dependence included, can produce the same departure, and whether a particular departure registers depends on the statistic, the pipeline and the sample size. A real statistic inside the interval means no detected excess under this reference; it is not evidence that clusters are absent. A statistic below the interval is reported as a diagnostic (the pipeline finds less on the data than on its twins), not as a verdict.

`copula = "t"` is a sensitivity variant. It gives the twins heavier-tailed dependence at a `df` fixed by the caller (8, then 3, is a reasonable pair), so a rerun shows whether a departure from the Gaussian twins persists once tail co-movement is built into the reference. A departure that persists is harder to attribute to tails; one that disappears is consistent with tail dependence. Neither outcome by itself establishes or rules out clusters. The Gaussian run is the primary analysis and the t run is reported alongside it, not in its place.

Value

An object of class `"matched_null_test"`: a list with the real statistic (`real`), the `R` twin statistics (`null`), the reference interval (`interval`, the `probs` quantiles of `null`), the plug-in Monte Carlo exceedance quantity (`p_exceed`, computed as `(1 + sum(null >= real)) / (R + 1)`; not an exact finite-sample p-value, see Details), whether the real statistic lies inside the interval (`within`), the requested `R`, and the reference family used (`copula`, with `df` when it is `"t"`). Reproducibility is the caller's: set a seed before calling.

Examples

```
if (requireNamespace("mclust", quietly = TRUE)) {
  suppressPackageStartupMessages(library(mclust))
  # a pipeline: how many components does BIC select?
  pick_k <- function(d) Mclust(d, G = 1:5, verbose = FALSE)$G
  set.seed(42)
  x <- matrix(rnorm(500 * 4), 500, 4)  # typeless data
  matched_null_test(x, pick_k, R = 30)

  # sensitivity variant with heavier-tailed dependence in the reference:
  # matched_null_test(x, pick_k, R = 30, copula = "t", df = 8)
}
```

Index

copula_null, 2
copula_null(), *4*

matched_null_test, 3