

Package ‘matchednull’

August 21, 2026

Title Matched-Null Tests for Cluster-Count Claims

Version 0.2.1

Description Builds matched nulls for cluster-count claims: synthetic twins of a dataset that preserve every marginal distribution and the full correlation matrix 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 adds tail dependence to the null, so that an apparent excess of clusters can be checked against a heavier-tailed alternative before it is read as evidence of types. Implements the matched-null procedure of Meng (2026) “Types Without Taxa” <[doi:10.17605/OSF.IO/2EKCG](https://doi.org/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 2026-08-21 08:20:03 UTC

Contents

copula_null	2
matched_null_test	3
Index	6

copula_null	<i>Draw one matched null ("null twin") of a dataset</i>
-------------	---

Description

Builds a synthetic twin of `x` that preserves every marginal distribution exactly and the correlation matrix to within sampling error, while containing no cluster structure by construction. With `copula = "gaussian"` (the default) all dependence in the twin is Gaussian. Clustering found in real data but not in its twins must come from structure beyond the margins and covariance; clustering found equally in both was never more than the data's shape.

Usage

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

Arguments

<code>x</code>	A numeric matrix or data frame (rows = observations, columns = variables), complete cases only, at least two rows and two columns.
<code>copula</code>	Dependence family of the twin: "gaussian" (default) or "t". The Gaussian family is the null of the method; "t" supplies a second, deliberately blinded null for stress-testing an exceedance the Gaussian twins have already licensed, and does not replace it.
<code>df</code>	Degrees of freedom of the t copula, a single positive number (default 8; used only when <code>copula = "t"</code>). Smaller is heavier-tailed; <code>df = 3</code> is a hard stress test, <code>df = 8</code> a moderate one.
<code>ridge</code>	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, 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; the rank correlation is matched exactly and the Pearson correlation to within sampling error.

With `copula = "t"` the same construction is driven by a multivariate t sample instead: margins and correlations are preserved 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. Its use is as a stress test: a verdict of "exceeds the Gaussian null" that a t twin reproduces was heavy-tailed dependence, not types.

Value

A numeric matrix of the same dimensions as `x`: one matched-null draw. Each column contains exactly the values of the corresponding column of `x`, rearranged.

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:
round(cor(x) - cor(twin), 2)

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

matched_null_test	<i>Test a cluster count against matched nulls</i>
-------------------	---

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, which share the data's margins and covariance but contain no cluster structure, produce. 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

<code>x</code>	A numeric matrix or data frame, complete cases only.
<code>cluster_fn</code>	A function taking a data matrix and returning a single number: the statistic to be tested, typically the selected number of clusters (but any scalar summary of clustering strength works).
<code>R</code>	Number of matched-null twins to draw (default 200).
<code>probs</code>	Lower and upper quantiles of the null distribution used for the interval verdict (default <code>c(.025, .975)</code>).
<code>copula, df</code>	Dependence family of the twins and <code>t</code> degrees of freedom, passed to <code>copula_null()</code> . Defaults to "gaussian", which is the null of the method; <code>copula = "t"</code> adds a second, tighter null rather than substituting for it, and <code>df</code> is fixed by the caller, never estimated from the data.
<code>ridge</code>	Passed to <code>copula_null()</code> .
<code>parallel</code>	Evaluate the twin loop through future.apply , so that the backend is whatever <code>future::plan()</code> 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 <code>parallel = FALSE</code> stream, which is left byte-identical to earlier versions, so a seeded analysis should not switch settings part-way through.

Details

A verdict of "exceeds the null" under the default Gaussian twins licenses only "structure beyond margins and correlations", it does not by itself license types: heavy-tailed dependence also exceeds a Gaussian null. To separate the two, rerun with `copula = "t"` (`df` 8, then 3). A result that survives the `t` twins as well is harder to attribute to tails; a result the `t` twins reproduce was tail dependence, not types. The Gaussian run remains the test, and a result quoted without its Gaussian verdict is not a matched-null result.

Value

An object of class "matched_null_test": a list with the real statistic (`real`), the null draws (`null`), the null interval (`interval`), a one-sided Monte Carlo p-value for exceedance (`p_exceed`), the interval verdict (`within`), and the null 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)

# stress test of an exceedance verdict against heavier tails:
# matched_null_test(x, pick_k, R = 30, copula = "t", df = 8)
}
```

Index

`copula_null`, [2](#)

`copula_null()`, [3](#), [4](#)

`matched_null_test`, [3](#)