A Matched-Null Test of Whether the Clusters a Pipeline Reports Are Real, With an Application to Personality Types

Miura Meng — Graduate School of Education, University of Pennsylvania

2026-07-11

Abstract

A clustering algorithm applied to continuous data will report some number of clusters, or "types," whether or not any exist, and a selection criterion will usually prefer more than one. This tutorial introduces matchednull, an R package implementing a general test of whether a reported cluster count describes the population or only the shape of the data. The test builds a matched null: a synthetic twin of the dataset that preserves every variable's distribution exactly and the full correlation matrix, while containing no clusters by construction. Because the null is defined on the data rather than on any algorithm, the user's own pipeline runs unchanged on data and twins alike, and the real count is read against the twins' distribution. In calibration on typeless data with skewed margins, BIC selection and the bootstrap likelihood-ratio test report spurious structure in every dataset while the matched-null test fires in none; in positive controls the test detects types built into synthetic data, including types that leave every marginal distribution untouched. A worked example applies the full workflow, preregistered and run on held-out data, to three public personality datasets totaling roughly one million respondents: the number of "types" selected swings from one to ten with a single covariance setting, and by the registered statistic it never exceeds what the typeless twin produces, under a Gaussian mixture and under k-means alike. The tutorial closes with the test's operating boundaries and what exceeding the null does and does not mean.

Keywords: cluster analysis, matched null, mixture models, taxometrics, R package

The problem

Run a clustering method on continuous data and it will hand back clusters. Give a Gaussian mixture model a single smooth cloud of points and it still reports two or three or ten components, because each added component trims the fit a little and the selection criterion takes the trade. The same holds for k-means with any count heuristic, for latent class and latent profile models, and for hierarchical methods cut at a chosen height. A clean partition on its own therefore establishes very little, and the announcement that "the data contain k clusters" quietly mixes a fact about people with a fact about the procedure.

The announcement is common. Clinical subtypes, learner profiles, market segments, and data-driven personality types are all, statistically, the same claim: a clustering pipeline selected some number of groups, and the number is offered as a property of the population. The best-known recent case is in personality, where a widely cited paper reported four personality types in large trait datasets (Gerlach et al., 2018). That mixture models over-extract components when continuous data depart from normality is well established (Bauer & Curran, 2003), so the reported number may describe the respondents, or only the method. The problem this tutorial addresses is how to tell.

What is needed is a null: a version of the data that contains no clusters, against which the real count can be read. The nulls in common use set the bar too low or answer a different question. Shuffling every association out of the data (Gerlach et al., 2018) destroys the correlations, and correlated data clear that bar without containing any groups; a published comment made this criticism but did not supply the null it called for (Katahira et al., 2020). The gap statistic compares the observed clustering to a uniform reference with no dependence at all (Tibshirani et al., 2001), which correlated

data clear as easily. The bootstrap likelihood-ratio test (McLachlan, 1987) asks whether $k+1$ components fit better than k within an assumed parametric family, which means committing to that family rather than holding the data's own structure fixed. SigClust (Liu et al., 2008) tests whether any two-group split beats a single Gaussian, so it speaks to whether clustering exists rather than to how many, and its Gaussian assumption is the very one that fails on skewed data. Taxometric procedures (Meehl, 1995; Ruscio et al., 2006) ask the categorical-versus-continuous question directly and well, but from a fixed indicator set; they do not say how many clusters there are or how that number moves when the model changes.

The test presented here fills that gap with a matched null borrowed from the surrogate-data tradition of nonlinear time-series analysis (Schreiber & Schmitz, 2000; Theiler et al., 1992): fix everything innocent about the dataset, randomize only the thing in question, and ask whether the real statistic exceeds its surrogate distribution. For a claim about clusters, the innocent features are each variable's distribution and the correlations among variables; the thing in question is everything beyond them, which is where latent groups live. The matchednull package implements this construction, the count test around it, and the follow-up checks that separate "more structure than the null" from "categories of people." The package is available on GitHub at https://github.com/haomeng797-ship-it/matchednull and has been submitted to CRAN.

The idea: a null twin

The null is a Gaussian copula (Sklar, 1959). It keeps every trait's distribution and the whole covariance matrix and scrambles only what is left, the higher-order dependence, so a cluster counts as real only if it survives a world that already shares the data's margins and correlations.

Pictured as a three-story building, the data hold their marginals, each variable's histogram, on the ground floor, and their pairwise correlations on the second; everything from the third floor up is higher-order dependence, where any latent types would live, since groups within a dataset show up as conditional dependencies beyond the linear correlation. The copula null rebuilds the ground floor and the second floor exactly as in the real data and leaves every floor above empty: once the correlation matrix is fixed, its Gaussian dependence adds no further joint structure. If clustering finds as many components in a building whose upper floors are empty as it does in the real data, those components were never on the upper floors of the real data either; they were the shape of the lower two all along.

The construction is a single rank transformation:

Algorithm 1. Gaussian-copula matched null.

Input: a real $n \times p$ data matrix X.

1. Take the correlation matrix $\mathrm{C} = \mathrm{cor}(\mathrm{X})$ and its Cholesky factor L (adding a 10^{-6} ridge to the diagonal if C is not positive definite).
2. Draw Z, an $n \times p$ matrix of independent standard-normal values, and set $\mathrm{Y} = \mathrm{ZL}$, so that Y carries the same correlations as X.
3. For each variable j, replace column j of Y with the sorted real values of X_j, laid down in the rank order of Y_j.

Output: X^*, whose every marginal matches X exactly, whose covariance matches to within sampling error, whose dependence is Gaussian, and which has no clustering built in.

Step 3 is the rank reordering of Iman & Conover (1982); it matches the normal-scores rank correlation exactly, so the Pearson covariance is reproduced up to sampling error (in the datasets below, the largest correlation the twin misses is about 0.009).

The test, stated formally

For readers who want the construction and the test stated precisely, four displays suffice. By Sklar's theorem (Sklar, 1959), any joint distribution F with margins $F_1, \dots, F_p$ decomposes into those margins and a copula C that carries all of the dependence,

$$F(x_1, \dots, x_p) \;=\; C\big(F_1(x_1), \dots, F_p(x_p)\big)\,.$$

The matched null replaces the data's copula with the Gaussian copula indexed by the data's own correlation matrix $\widehat{\Sigma}$, while keeping the empirical margins $\widehat{F}_1, \dots, \widehat{F}_p$:

$$F_0 \;=\; C^{\text{Gauss}}_{\widehat{\Sigma}}\big(\widehat{F}_1, \dots, \widehat{F}_p\big)\,, \qquad C^{\text{Gauss}}_{\Sigma}(\mathrm{u}) \;=\; \Phi_\Sigma\big(\Phi^{-1}(u_1), \dots, \Phi^{-1}(u_p)\big)\,,$$

where Φ_Σ is the multivariate normal distribution with correlation matrix Σ and Φ^{-1} the standard-normal quantile function. Because a Gaussian copula is fully indexed by its correlation matrix, F_0 is the unique member of this family that agrees with the data on both margins and correlations and carries no dependence beyond them; Algorithm 1 samples from F_0. The null hypothesis is that the data are distributed as F_0, that is, that nothing beyond the margins and correlations, where any latent groups would have to live, shapes the joint distribution.

The test statistic is the count functional of the user's pipeline. In the worked example it is the median selected count over a set $\mathcal{M}$ of mixture parameterizations,

$$T(\mathrm{X}) \;=\; \mathrm{median}_{m\in\mathcal{M}}\;\hat{k}_m(\mathrm{X}), \qquad \hat{k}_m(\mathrm{X}) \;=\; \arg\max_{k\in\{1,\dots,K\}} \mathrm{BIC}_m(k;\mathrm{X}),$$

but any scalar functional of a clustering, from any pipeline, may stand in for T. With $\mathrm{X}_1^*, \dots, \mathrm{X}_R^*$ independent draws from F_0 at the matched sample size, the one-sided Monte Carlo p-value for exceedance is

$$p \;=\; \frac{1+\sum_{r=1}^{R} \mathbb{1}\{T(\mathrm{X}_r^*) \geq T(\mathrm{X})\}}{R+1},$$

and the interval verdict reads $T(\mathrm{X})$ against the $[.025,\ .975]$ quantiles of the null draws; throughout, "exceeds the null" means $T(\mathrm{X})$ above the .975 quantile, the upper end of that interval. The construction is exact in the margins, and the correlations are reproduced up to the sampling error of the rank map; both conventions, the added-one Monte Carlo p and the quantile interval, follow standard surrogate-data practice (Theiler et al., 1992).

Why this null and not a richer one

One design choice deserves stating up front, because it governs everything the test can and cannot do. The Gaussian copula is the minimal null for the role, not the most faithful one, and minimality is what the test needs. Its dependence is exhausted by the correlation matrix, so every higher-order dependency in the real data must come from the data rather than the null. A richer copula would defeat the purpose: a t copula adds tail dependence, and a vine copula can fit almost any dependence

structure at all, so using either as the null would absorb into it the very structure the test is meant to detect. The choice also has an information-theoretic characterization: among all distributions with a given covariance, the multivariate normal has maximum entropy (Cover & Thomas, 2006), so on the normal-scores scale the twin carries the least structured dependence consistent with the data's margins and correlations, and anything the real data show beyond it is structure the data themselves supply. The flip side, taken up in the final section, is that the test reads any dependence beyond the linear as a departure, whether or not it is categorical; deciding whether a departure is categorical is the job of the follow-up checks, not of the count.

Using matchednull

Installation

```
# install.packages("remotes")
remotes::install_github("haomeng797-ship-it/matchednull")
library(matchednull)
```

The twin, in one line

`copula_null()` builds the synthetic twin of Algorithm 1:

```
set.seed(1)
x <- matrix(rnorm(400 * 3), 400, 3) %*%
  chol(matrix(c(1, .5, .3, .5, 1, .4, .3, .4, 1), 3, 3))
twin <- copula_null(x)
```

```
all(sort(twin[, 1]) == sort(x[, 1]))   # margins: identical, value for value
round(cor(x) - cor(twin), 2)           # correlations: equal to sampling error
```

The twin reuses the real values, reshuffled, so every histogram is exactly the real one; the correlations are reproduced; and everything above them is plain Gaussian dependence.

The test: your pipeline against its twins

`matched_null_test()` takes the data and the user's own pipeline, wrapped as a function that returns one number, typically the selected number of clusters. It runs the identical pipeline on the real data and on `R` twins, and asks whether the real answer stands out:

```
library(mclust)
# your pipeline: the number of components BIC selects
pick_k <- function(d) Mclust(d, G = 1:10, verbose = FALSE)$G

set.seed(20260620)
matched_null_test(x, pick_k, R = 200)
#> Matched-null test (200 null twins)
#>   real statistic:      1
#>   null interval:       [1, 1]
#>   p (real >= nulls):   1
#>   verdict:             null-like (within the twins' interval)
```

The output reports the real count, the null's 95 percent interval, a one-sided Monte Carlo p-value

for exceedance, and the verdict. Here the verdict is "null-like": the pipeline finds no more clusters in the real data than in its twins, because the data, though correlated, are a single population, and everything the pipeline could see was carried by the margins and correlations. (A p of 1 is not an error: it says the real statistic sits at or below every null draw, the strongest null-like reading the test can return.)

For adapting the call beyond the example, the full signature is `matched_null_test(x, cluster_fn, R = 200, probs = c(.025, .975), ridge = 1e-6)`. `x` is a numeric matrix or data frame with complete cases: remove or impute missing values first, since the twin is built from the observed values and the correlation matrix, and neither is defined under NA. `cluster_fn` is the pipeline function, the second argument above; it may return any scalar summary of clustering strength, not only a selected count. `R` is the number of twins; `probs` sets the quantiles of the interval verdict, so a stricter or looser read than 95 percent is one argument away; `ridge` is a small diagonal added when the correlation matrix is not positive definite. The returned object is a list of class `"matched_null_test"` whose components are directly usable in code, not only in the printed summary: `$real` (the statistic on the real data), `$null` (all `R` null draws), `$interval`, `$p_exceed`, and `$within`. The pipeline is run exactly as supplied: an error raised inside `cluster_fn` on some twin propagates rather than being skipped, so a fragile pipeline should catch its own exceptions.

A detection, for contrast

To see the test fire, hide two genuine groups where the null cannot reach them. The two components below share identical standard-normal margins on every variable; they differ only in the ori-

entation of their correlations. No histogram, mean, or variance carries any trace of the grouping; it lives entirely in the dependence.

```
set.seed(42)

z <- sample(2, 400, replace = TRUE)

x_types <- matrix(rnorm(400 * 4), 400, 4)

L1 <- chol(matrix(c(1,  .85,  .85, 1), 2, 2))

L2 <- chol(matrix(c(1, -.85, -.85, 1), 2, 2))

x_types[z == 1, 1:2] <- x_types[z == 1, 1:2] %*% L1

x_types[z == 1, 3:4] <- x_types[z == 1, 3:4] %*% L1

x_types[z == 2, 1:2] <- x_types[z == 2, 1:2] %*% L2

x_types[z == 2, 3:4] <- x_types[z == 2, 3:4] %*% L2



set.seed(20260620)

matched_null_test(x_types, pick_k, R = 200)

#> Matched-null test (200 null twins)

#>   real statistic:      2

#>   null interval:       [1, 1]

#>   p (real >= nulls):   0.005

#>   verdict:             exceeds the null (beyond margins + covariance)
```

The twins of these data have the same (near-zero) pooled correlations and the same margins, but no groups, and the pipeline finds one component in every one of them; in the real data it finds two. "Exceeds the null" means exactly this: the clustering draws on structure beyond what the margins

and correlations supply. Whether that structure is categorical is the next question, not the same one, and it is answered by the signature checks of the workflow below.

Any engine, one changed line

The null is defined on the data, not on the algorithm, so `cluster_fn` can wrap anything that returns a scalar: a k-means heuristic, a hierarchical cut, or a published typology's exact workflow, run unchanged.

```
library(cluster)

# a different engine: k-means, count chosen by the gap statistic

pick_k_km <- function(d) {

  g <- clusGap(d, kmeans, K.max = 10, B = 50, spaceH0 = "scaledPCA",

               nstart = 10, iter.max = 30)

  maxSE(g$Tab[, "gap"], g$Tab[, "SE.sim"], method = "Tibs2001SEmax")

}

set.seed(20260620)

matched_null_test(x, pick_k_km, R = 200)

#> Matched-null test (200 null twins)

#>   real statistic:      1

#>   null interval:       [1, 1]

#>   p (real >= nulls):   1

#>   verdict:             null-like (within the twins' interval)
```

Switching engines changes one line, and on the typeless data of the first example the verdict is the

same. The question, whether the real count exceeds what the data's own margins and correlations already produce, stays the same whatever the engine; engines differ only in what they are able to see (a centroid method like k-means, for instance, has little power for the dependence-defined groups of the second example, which is a property of the engine, not of the test).

The full workflow

The count test sits inside a five-step workflow. The steps around it are not decoration; each one guards a boundary the count test alone cannot.

1. Inspect the margins for pronounced multimodality, the one regime where the null is blind (the reason is given in the final section). Granular scales are tie-broken with a small jitter before a formal test such as Hartigan's dip (Hartigan & Hartigan, 1985).
2. Draw the twins with `copula_null()`.
3. Run the identical clustering pipeline, whatever it is, on the real data and on every twin.
4. Compare the real count with the twins' distribution (`matched_null_test()`), always at a matched sample size: selected counts grow with n, so real data and null must be compared at the same n.
5. Read the categorical signatures of the selected solution: the share of respondents assigned to a component with high posterior confidence, the share of the fit improvement carried by the first split, and, where several indicators of one construct are available, the taxometric Comparison Curve Fit Index (Ruscio et al., 2006) through RTaxometrics (Ruscio, 2023). A higher count is not the same as categories; these signatures are what separate the two.

In code, the two steps that bracket the count test:

```
# Step 1: margins first. Tie-break granular scales, then a formal test.
apply(x, 2, function(v) diptest::dip.test(jitter(v, amount = .5))$p.value)


# Step 5: categorical signatures, where several indicators of one
# construct are available (here, the facets of one trait).
RTaxometrics::RunCCFIProfile(as.data.frame(x_facets), num.p = 11,
                             n.pop = 5000, n.samples = 18, graph = 0)
```

Reproducibility is deliberately left to the caller: set a seed before calling.

Does the test work?

A null result means nothing unless the test detects types that are genuinely present, and a detection means nothing if the test also fires on data with no types. Both directions were checked, with the same fourteen-model Gaussian-mixture pipeline used in the worked example below (fit with mclust; Scrucca et al. (2016)).

It detects real types, including invisible ones

Two regimes of synthetic data with known structure (five variables, 4,000 cases, forty nulls per setting; Figure 1). In the dependence regime, two components share identical standard-normal margins on every variable and differ only in the sign of their within-pair correlations, so the type signal lives entirely in the joint structure and nothing leaks into the histograms. The test detects this kind, increasingly with signal strength: as the correlation difference grows, the real median count sepa-

rates from the null, rising from one to two and beyond, while the null median stays at one. This regime matters because a central form of the type hypothesis, types that differ in the configuration of traits rather than the level of any one, produces exactly this structure.

In the separation regime, components differ in their means, so the separation enters the histograms. At moderate separations the test detects the structure; at the largest separation tested, three standard deviations, it does not, because a separation that large is carried entirely by the margins and covariance, which the null preserves. That boundary is real and is discussed in the final section; the workflow's first step exists precisely because of it.

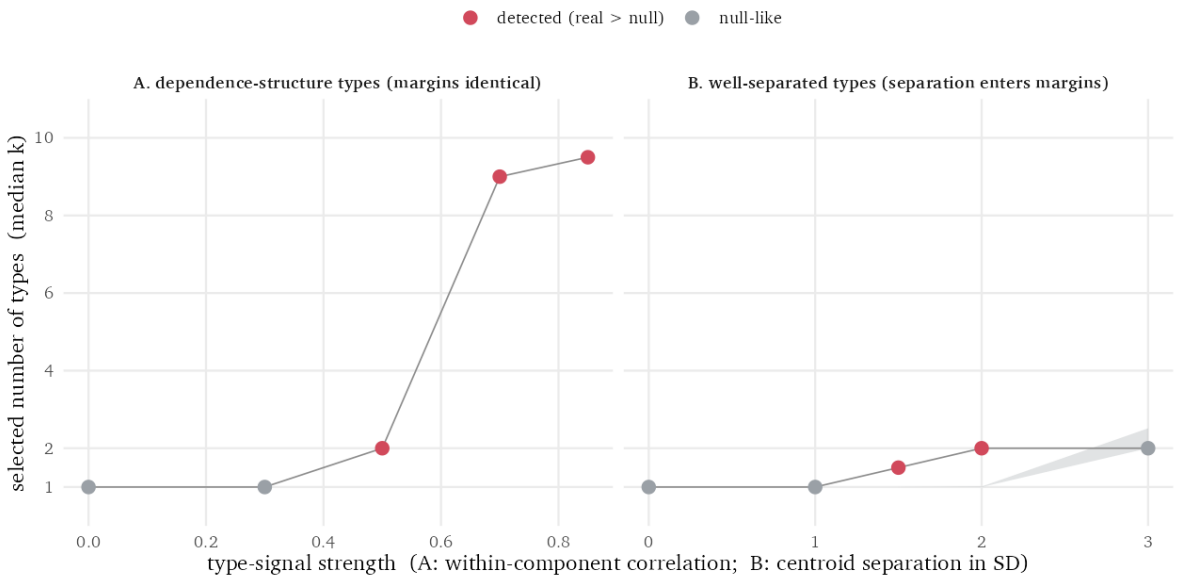

Figure 1: Positive controls on synthetic data with known structure. Grey band: the 95% interval of the matched-null median count. Points: the real median count, red where it exceeds the null. Left: types defined only by dependence (identical margins, opposite correlation orientation) are detected, increasingly so with signal strength. Right: types separated in their means are detected at moderate separation and are reproduced by the null only at the largest separation, where the separation is fully carried by the margins and covariance the null preserves.

It stays quiet where standard tools do not

The sharper check is calibration on typeless data, in the regime known to break standard criteria: non-normal margins (Bauer & Curran, 2003). One hundred datasets were drawn from a Gaussian with the correlation matrix of real personality data, and one hundred more with the same Gaussian copula but skewed margins (chi-square, beta, and squared-uniform transforms). No types exist in either set.

Table 1. False-positive calibration on typeless data (100 datasets per row). The test fires when the real fourteen-model median exceeds the nulls' 97.5th percentile, the upper end of the 95% interval. Table S5 reports the full calibration ladder, adding the gap statistic and the non-Gaussian-dependence scenarios.

Scenario (typeless)	Matched-null test fired	BIC selects k > 1	Mean BIC k	bLRT selects k > 1
Gaussian margins	0%	0%	1.0	0%
Skewed margins	0%	100%	9.7	100%

On the skewed but typeless data, BIC selection reported more than one component in every one of the hundred datasets, 9.7 components on average of a possible ten, and the bootstrap likelihood-ratio test did the same in every dataset tested. The matched-null test fired in none. This is the failure mode that produces published "types" from typeless data, and the matched null is built to absorb it: the twin inherits every skewed margin exactly, so skew alone can no longer look like structure.

The verdict does not depend on the engine

Because the null is data-level, the same test was run with two engines of different paradigm and count rule on the personality data of the next section: the Gaussian mixture above, and k-means

with a gap-statistic count. The two engines report very different raw counts, as they should, but the verdict, real count within the null, is the same for every dataset under both (Table S3 in the supplement). The framework's algorithm-agnostic claim is borne out in practice, not only in principle.

A worked example: are there personality types?

The workflow was applied, under a preregistered decision rule and on held-out data, to three large public personality datasets: the IPIP-NEO-120 (Johnson, 2014) (410,376 respondents), an IPIP fifty-item Big-Five inventory (Goldberg, 1999) (603,322), and the IPIP-HEXACO (Lee & Ashton, 2004) (22,734). Dimensional structure in personality is already well established (Haslam et al., 2012; Haslam et al., 2020); the datasets serve here as a realistic, high-stakes demonstration, since the "types" claim was made of data exactly like these (Gerlach et al., 2018). The full analysis, including the preregistration, seed, and code, is public (see the Open Practices Statement); this section walks the workflow's steps in order.

Step 1, margins. Unimodality tests on the raw scores reject only because summed Likert scales are granular; after a tie-breaking jitter, 64 of the 65 trait margins retain unimodality (Hartigan & Hartigan, 1985). No trait shows the pronounced multimodality that would mark strongly separated mean-type groups, so the data sit in the regime where the count test has power.

Steps 2 and 3, twins and pipeline. Two hundred twins per dataset, each at the matched sample size (8,000 cases at the domain level, 5,000 at the facet level). The pipeline is the fourteen-parameterization Gaussian-mixture grid: mclust's covariance codes from spherical (EII) to fully ellipsoidal (VVV), one to ten components, count selected by BIC, summarized by the preregistered median across the fourteen.

The grid itself answers a first question: how much of a reported count is the analyst's covariance assumption? All of it can be. The selected count ran from two to ten in the NEO-120 domains and across the full range, one to ten, in its thirty facets (Figure 2; Table S1). Constrained parameterizations pile up components to approximate a correlated continuous cloud; flexible ones absorb the same cloud into one or two. A setting that encodes no claim about respondents moves the answer across nearly the entire reportable range.

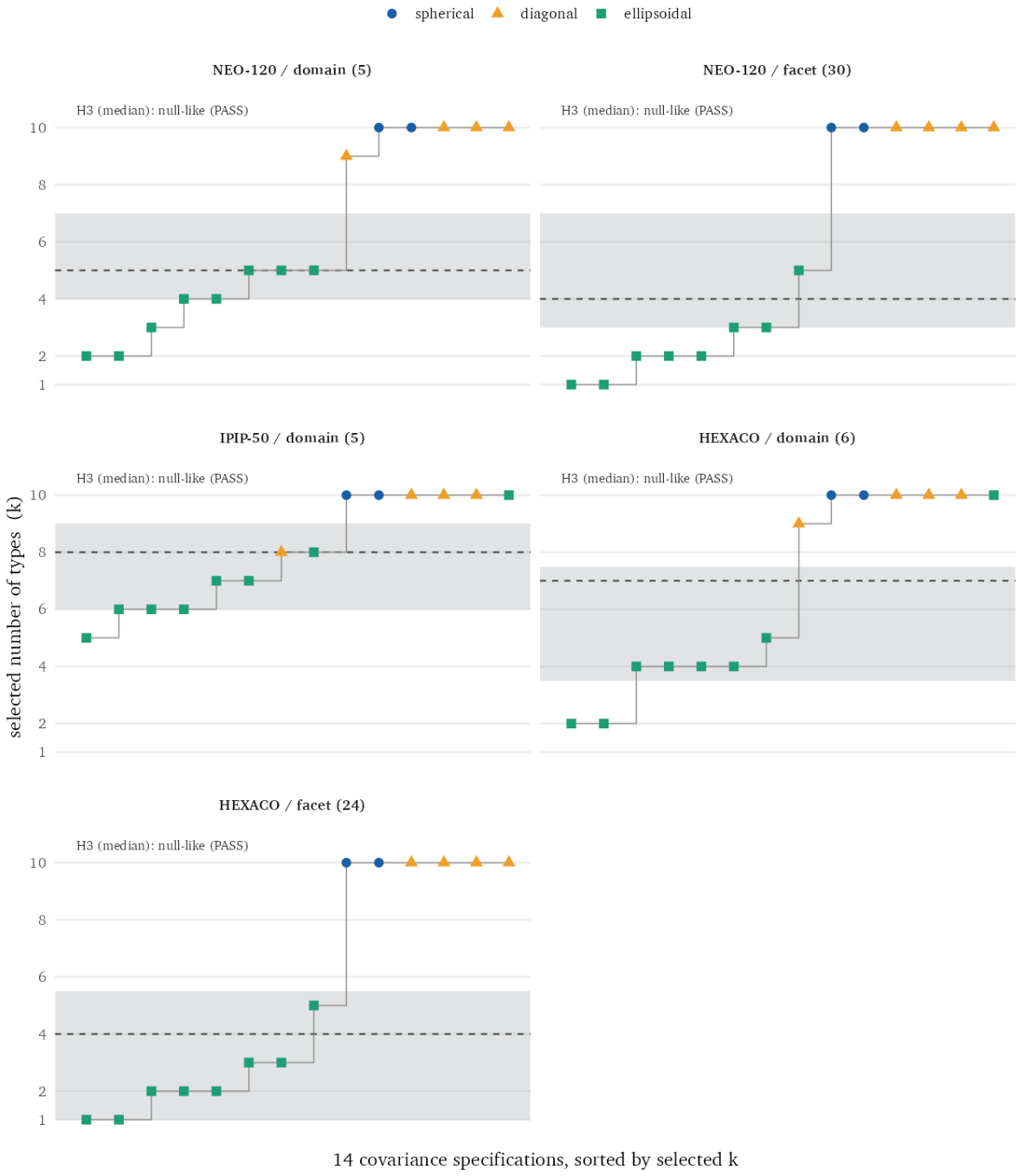

Figure 2: The BIC-selected number of mixture components ("types") across the fourteen covariance parameterizations, on held-out data at the matched sample size. Each point is one parameterization, colored and shaped by covariance family and sorted by selected count. The grey band marks the 95% interval of the matched-null median count; the dashed line is the real median. A real median inside the band indicates a typical count indistinguishable from data with no latent types.

Step 4, the count against the null. In all five analyses the real median falls within the null's 95

percent interval (Table 2). In the NEO-120 domains the real median is five components against a null interval of [4.0, 7.0]; the same holds everywhere (one-sided Monte Carlo p from .11 to .80). The data do carry structure the twin cannot reproduce: the BIC improvement of the best mixture over one component exceeds the null's 95th percentile in every dataset (Table 2, ΔBIC). But that improvement certifies only non-Gaussianity, skew or mild nonlinearity or uneven density, and it does not raise the number of types above what a typeless population yields under the identical procedure.

Table 2. Real data compared with the matched null, by analysis. The clustering gain (ΔBIC of the best mixture over a single component) exceeds the null's 95th percentile in every analysis, so real structure beyond the linear dependence is present; the median count lies within the null's 95% interval in every analysis, so that structure does not raise the number of types.

Slice	n	Median k	Null 95%	One-sided p	ΔBIC	Null 95th
NEO-120 dom.	8,000	5	[4.0, 7.0]	.78	717	488
NEO-120 fac.	5,000	4	[3.0, 7.0]	.80	4,013	1,122
IPIP-50 dom.	8,000	8	[6.0, 9.0]	.40	1,369	915
HEXACO dom.	8,000	7	[3.5, 7.5]	.11	816	221
HEXACO fac.	5,000	4	[1.0, 5.5]	.34	2,701	344

Step 5, categorical signatures. None appears. No trait's taxometric index reaches the categorical threshold (mean CCFI .375 across NEO-120 traits, .430 across HEXACO; highest single trait .535, in the ambiguous band; Figure 3; Table S2). Under the selected solutions, fewer than half of respondents in every dataset are assigned to any component with posterior confidence above .8, and most of the fit improvement is concentrated in the first binary split (Table 3), the signature of a partitioned continuum rather than of separable groups.

Table 3. The BIC-selected mixture solution for each dataset and two properties of it. The solution

shown is the single parameterization with the highest BIC over the fourteen-model grid; its k is a property of that one model and need not equal the cross-grid median count of Table 2. Confident assignment: the proportion of respondents with maximum posterior membership probability above .8. First-split share: the proportion of the total BIC gain contributed by the one-to-two-component division.

Slice	Model	k	Confident assign. (>.8)	First-split share
NEO-120 dom.	VEE	5	16.0%	90.2%
NEO-120 fac.	VEE	5	48.9%	88.7%
IPIP-50 dom.	VVE	8	10.6%	59.4%
HEXACO dom.	VVE	4	24.0%	86.6%
HEXACO fac.	VEE	3	46.0%	98.9%

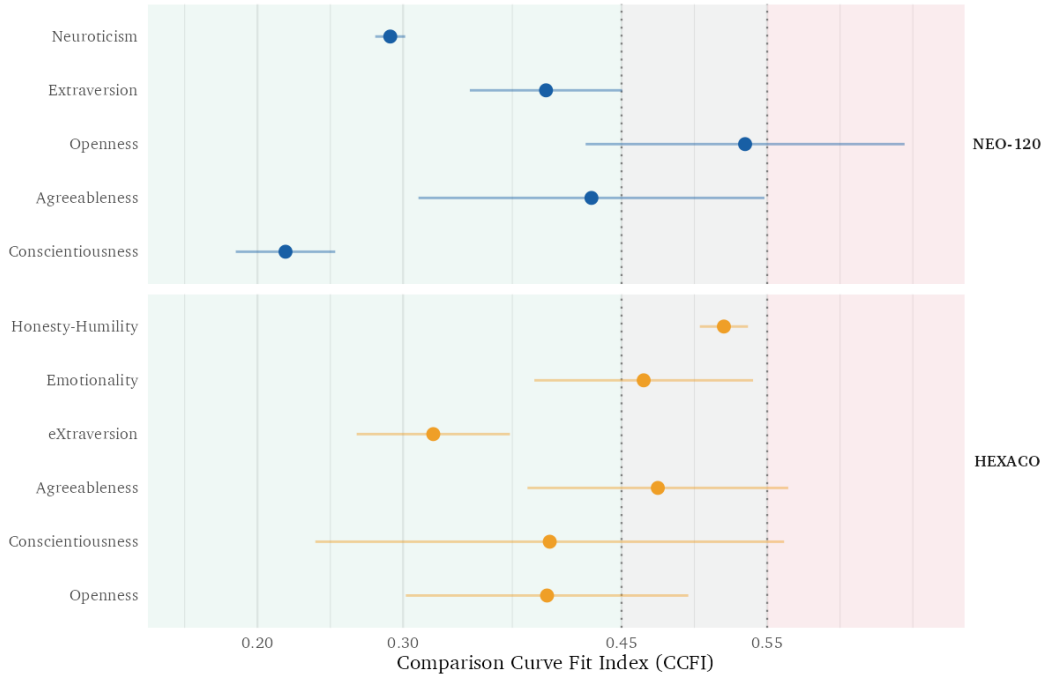

Figure 3: CCFI by trait. Each point is the mean of the two preregistered taxometric procedures, MAMBAC and MAXEIG; the segment marks their range. Values below .45 read dimensional, .45 to .55 ambiguous, above .55 categorical. No trait's mean reaches the categorical threshold.

The workflow's verdict, in one sentence: the "types" these datasets produce are the shape of their margins and correlations, reproduced in full by a twin that contains no types, under two different

clustering engines, with no categorical signature in any trait. The substantive conclusion is not new; what the workflow adds is that the claim was made to fail in a specific, preregistered way, and did not.

Practical guidance: where the test has power, and what its verdicts mean

A tool is used well only when its boundaries are as clear as its output. Four are worth knowing before applying this one.

Strong mean separation is the null's blind spot, and the workflow's first step is the guard. Suppose two groups differ only in their average levels, far enough apart that the separation shows in the histograms and the correlations. The null preserves both, so its twins reproduce the same partition, and the count test goes quiet: the three-standard-deviation case in Figure 1. The reason is a fact about the data rather than a defect of the test. From margins and correlations alone, two well-separated groups and a single unclustered population with the same bimodal margins are the same object; the information that would separate those descriptions is not present in the margins and correlations. A method that reports the separated pair anyway is supplying the missing information by assumption, and the calibration shows what that assumption costs: committed to Gaussian components, the bootstrap likelihood-ratio test renders a skewed single group as several, in every typeless dataset tested. The practical rule is simple. A separation strong enough to defeat the null is strong enough to bend at least one margin visibly bimodal, which is exactly what step 1 inspects; a dataset that passes the margin check is in the regime where the count test speaks with authority.

"Exceeds the null" means structure, not types. The null fixes the linear dependence and nothing above it, so any departure, categorical or not, can register. This is by design, and it has a measurable consequence: on typeless data whose *dependence* is non-Gaussian (a t copula with tail dependence, correlation matrix from real personality data), the count test fires nearly always (99 to 100 percent across tail-dependence strengths; Table S5), just as BIC and the bootstrap test do. The two-stage design is what keeps this from being a false alarm: the count answers "is there structure beyond margins and correlations," and step 5 answers "is it categorical." On those same t-copula data the structure is elliptical and groupless, which is what the categorical signatures then report. Read the verdicts in their order; an exceedance alone licenses interest, not a typology.

Power declines as irrelevant dimensions are added. With the type signal confined to a fixed set of four variables, detection of dependence-only types weakens as noise dimensions grow: strong at five variables, reduced at ten, and present only for the strongest signal at twenty (Table S4). Where the substantive claim concerns a low-dimensional trait space, as in the worked example, this costs little; users testing high-dimensional data should concentrate the analysis on the variables the typology is actually about, or expect the test to be conservative.

Fix the summary statistic in advance. A count must be summarized across specifications, and the choice matters: in the worked example the per-model counts are right-censored at the grid ceiling, so their mean is pulled upward by saturated parameterizations while the median is not, and the two can disagree about the verdict. The median was preregistered here for that reason, and the divergence is itself an instance of the analytic flexibility the procedure is meant to surface. Whatever statistic is chosen, choose it before the data are seen, and always compare real data and twins at the same sample size, since selected counts grow with n.

Conclusion

The number of clusters a pipeline reports is not, by itself, a property of the people it describes; it becomes one only when it exceeds what the data's own margins and correlations already produce, and when the excess carries categorical signatures. matchednull makes that check routine: one function builds the twin, one function runs any pipeline against it, and the surrounding workflow separates structure from categories. In the demonstration, a million-respondent "types" claim came apart into a covariance setting and a typeless twin; the next such claim, from any pipeline, can be given the same afternoon of scrutiny.

Open Practices Statement

All materials are public. The analysis code, the three public datasets' sources, and the rendered manuscript are available at https://github.com/haomeng797-ship-it/types-without-taxa; the matchednull package is available at https://github.com/haomeng797-ship-it/matchednull and has been submitted to CRAN. The confirmatory analysis of the worked example, its decision rule, its specification grid, and the analysis code were preregistered and archived on the Open Science Framework before the held-out data were opened, as a frozen, time-stamped registration with a persistent identifier (https://doi.org/10.17605/OSF.IO/2EKCG; analysis seed 20260620). The simulations reported in the validation and practical-guidance sections were not preregistered.

Declarations

Funding. This work received no external funding.

Conflicts of interest. The author declares no competing interests.

Ethics approval. The study is a secondary analysis of publicly available, de-identified datasets and involved no new data collection; the University of Pennsylvania Institutional Review Board determined that it did not require review.

Consent to participate / consent for publication. Not applicable (no new data were collected from human participants).

Availability of data and materials. The three datasets are public; their sources and citations are documented in the project repository.

Code availability. The matchednull R package and the complete analysis code are available at the repositories given in the Open Practices Statement.

Authors' contributions. The sole author designed the method, performed all analyses, wrote the software, and wrote the manuscript.

Supplementary tables

Table S1. The number of components BIC selected under each of the fourteen mclust covariance parameterizations, at the matched sample size (8,000 cases for domain analyses, 5,000 for facet analyses). The median of each column is the count statistic reported in Table 2.

Model	NEO-120 dom.	NEO-120 fac.	IPIP-50 dom.	HEXACO dom.	HEXACO fac.
EII	10	10	10	10	10
VII	10	10	10	10	10
EEI	10	10	10	10	10
VEI	9	10	10	9	10
EVI	10	10	10	10	10
VVI	10	10	8	10	10
EEE	5	2	7	10	5

Model	NEO-120 dom.	NEO-120 fac.	IPIP-50 dom.	HEXACO dom.	HEXACO fac.
EVE	4	3	7	5	3
VEE	5	5	10	4	3
VVE	3	3	8	4	2
EEV	5	1	6	4	1
VEV	2	2	6	2	2
EVV	4	1	5	4	1
VVV	2	2	6	2	2
Median	5	4	8	7	4

Table S2. Comparison Curve Fit Index for each trait under MAMBAC and MAXEIG, their mean (the statistic plotted in Figure 3), and the supplementary L-Mode curve. Values below .45 indicate dimensional structure, above .55 categorical, ambiguous between. The L-Mode curve is supplementary rather than preregistered (the registered statistic is the MAMBAC-MAXEIG mean); it slightly exceeds .55 for one trait (Emotionality, .559) without changing any registered verdict.

Trait	MAMBAC	MAXEIG	Mean	L-Mode
NEO-120 Neuroticism	.281	.301	.291	.316
NEO-120 Extraversion	.450	.346	.398	.385
NEO-120 Openness	.644	.425	.535	.364
NEO-120 Agreeableness	.548	.311	.430	.491
NEO-120 Conscientiousness	.185	.253	.219	.391
HEXACO Honesty-Humility	.537	.504	.520	.541
HEXACO Emotionality	.390	.540	.465	.559
HEXACO eXtraversion	.373	.268	.321	.344
HEXACO Agreeableness	.385	.564	.475	.471
HEXACO Conscientiousness	.240	.562	.401	.542
HEXACO Openness	.302	.496	.399	.546

Table S3. The count test under a second clustering engine. Each held-out domain dataset was tested with two engines of different paradigm and count rule: the Gaussian mixture of the worked example (median BIC-selected components over the fourteen covariance models; 200 null replicates) and k-means with a gap-statistic count (40 null replicates, an independently drawn subsample of the same size, n = 8,000). The k-means arm is a robustness demonstration rather than a registered analysis; the comparison is qualitative, each engine's real count against its own null, and the verdict

is unchanged under the smaller replicate count and the independent subsample.

Dataset	Gaussian mixture: real (null 95%)	k-means: real (null 95%)
NEO-120 dom.	5 ([4.0, 7.0])	1 ([1.0, 2.0])
IPIP-50 dom.	8 ([6.0, 9.0])	1 ([1.0, 2.0])
HEXACO dom.	7 ([3.5, 7.5])	1 ([1.0, 2.0])

Table S4. Power of the count test as irrelevant dimensions are added. Data as in the dependence regime of Figure 1: two components with identical standard-normal margins whose within-pair correlations (dimensions 1-4 only) differ in sign by 2*rho*; dimensions 5 through *p* are independent noise. n = 4,000; 40 null replicates per cell; the entry is the real median count against the null's 95% interval, with detection marked *.

	rho = 0	.30	.50	.70	.85
p = 5	1 [1, 1]	1 [1, 1]	2 [1, 1]*	9 [1, 1]*	9.5 [1, 1]*
p = 10	1 [1, 1]	1 [1, 1]	1 [1, 1]	2 [1, 1]*	2 [1, 1]*
p = 20	1 [1, 1]	1 [1, 1]	1 [1, 1]	1 [1, 1]	2 [1, 1]*

Table S5. What the count test reads as a departure: calibration on typeless data whose margins or dependence are progressively less Gaussian. Each row: 100 datasets (n = 2,000, five variables, correlation matrix from the real NEO-120 domains), 40 copula nulls each; the test fires when the real fourteen-model median exceeds the nulls' 97.5th percentile (nominal rate 2.5%). Rows 1-2 change only the margins, which the null matches exactly, so the test stays quiet where BIC and the bootstrap LRT do not. Rows 3-4 change the dependence itself (a t copula with tail dependence; margins standard normal), which no margin- and correlation-matched twin can carry, so the test fires by design: the departure is real, and whether it is categorical is settled by the signature checks (on these elliptical, groupless data, they read continuous). The bootstrap LRT was run on 30 datasets per scenario.

Typeless scenario	Matched-null test fired	BIC k > 1	gap k > 1	bLRT k > 1
Gaussian copula, normal margins	0%	0%	1%	0%
Gaussian copula, skewed margins	0%	100%	1%	100%
t copula (df = 8), normal margins	99%	99%	8%	100%
t copula (df = 3), normal margins	100%	100%	11%	100%

References

Bauer, D. J., & Curran, P. J. (2003). Distributional assumptions of growth mixture models: Implications for overextraction of latent trajectory classes. *Psychological Methods*, *8*(3), 338–363. https://doi.org/10.1037/1082-989X.8.3.338

Cover, T. M., & Thomas, J. A. (2006). *Elements of information theory* (2nd ed.). Wiley.

Gerlach, M., Farb, B., Revelle, W., & Nunes Amaral, L. A. (2018). A robust data-driven approach identifies four personality types across four large data sets. *Nature Human Behaviour*, *2*(10), 735–742. https://doi.org/10.1038/s41562-018-0419-z

Goldberg, L. R. (1999). A broad-bandwidth, public domain, personality inventory measuring the lower-level facets of several five-factor models. In I. Mervielde, I. Deary, F. De Fruyt, & F. Ostendorf (Eds.), *Personality psychology in Europe* (Vol. 7, pp. 7–28). Tilburg University Press.

Hartigan, J. A., & Hartigan, P. M. (1985). The dip test of unimodality. *Annals of Statistics*, *13*(1), 70–84. https://doi.org/10.1214/aos/1176346577

Haslam, N., Holland, E., & Kuppens, P. (2012). Categories versus dimensions in personality and

psychopathology: A quantitative review of taxometric research. *Psychological Medicine*, *42*(5), 903–920. https://doi.org/10.1017/S0033291711001966

Haslam, N., McGrath, M. J., Viechtbauer, W., & Kuppens, P. (2020). Dimensions over categories: A meta-analysis of taxometric research. *Psychological Medicine*, *50*(9), 1418–1432. https://doi.org/10.1017/S003329172000183X

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. https://doi.org/10.1080/03610918208812265

Johnson, J. A. (2014). Measuring thirty facets of the five factor model with a 120-item public domain inventory: Development of the IPIP-NEO-120. *Journal of Research in Personality*, *51*, 78–89. https://doi.org/10.1016/j.jrp.2014.05.003

Katahira, K., Kunisato, Y., Yamashita, Y., & Suzuki, S. (2020). Commentary: A robust data-driven approach identifies four personality types across four large data sets. *Frontiers in Big Data*, *3*, 8. https://doi.org/10.3389/fdata.2020.00008

Lee, K., & Ashton, M. C. (2004). Psychometric properties of the HEXACO personality inventory. *Multivariate Behavioral Research*, *39*(2), 329–358. https://doi.org/10.1207/s15327906mbr3902_8

Liu, Y., Hayes, D. N., Nobel, A., & Marron, J. S. (2008). Statistical significance of clustering for high-dimension, low-sample-size data. *Journal of the American Statistical Association*, *103*(483), 1281–1293. https://doi.org/10.1198/016214508000000454

McLachlan, G. J. (1987). On bootstrapping the likelihood ratio test statistic for the number of components in a normal mixture. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *36*(3), 318–324. https://doi.org/10.2307/2347790

Meehl, P. E. (1995). Bootstraps taxometrics: Solving the classification problem in psychopathology. *American Psychologist*, *50*(4), 266–275. https://doi.org/10.1037/0003-066X.50.4.266

Ruscio, J. (2023). *RTaxometrics: Taxometric analysis* (Version 3.2.1) [Computer software]. https://CRAN.R-project.org/package=RTaxometrics

Ruscio, J., Haslam, N., & Ruscio, A. M. (2006). *Introduction to the taxometric method: A practical guide*. Lawrence Erlbaum Associates.

Schreiber, T., & Schmitz, A. (2000). Surrogate time series. *Physica D: Nonlinear Phenomena*, *142*(3–4), 346–382. https://doi.org/10.1016/S0167-2789(00)00043-9

Scrucca, L., Fop, M., Murphy, T. B., & Raftery, A. E. (2016). Mclust 5: Clustering, classification and density estimation using Gaussian finite mixture models. *The R Journal*, *8*(1), 289–317. https://doi.org/10.32614/RJ-2016-021

Sklar, A. (1959). Fonctions de répartition à n dimensions et leurs marges. *Publications de l'Institut de Statistique de l'Université de Paris*, *8*, 229–231.

Theiler, J., Eubank, S., Longtin, A., Galdrikian, B., & Farmer, J. D. (1992). Testing for nonlinearity in time series: The method of surrogate data. *Physica D: Nonlinear Phenomena*, *58*(1–4), 77–94. https://doi.org/10.1016/0167-2789(92)90102-S

Tibshirani, R., Walther, G., & Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B*, *63*(2), 411–423. https://doi.org/10.1111/1467-9868.00293