

Package: spectralDEMATEL (via r-universe)

October 8, 2026

Title Spectral Diagnostics for DEMATEL Influence Matrices

Version 0.6.0

Description Computes the spectral diagnostics of a DEMATEL influence matrix: coupling and the total-effect multiplier, indirect and mode dominance, three readings of hierarchy, the entry and accumulation profiles, the eigenvalue condition number, a permutation surrogate baseline, and the closed-form derivative of the dominant eigenvalue in every off-diagonal entry. The package computes and returns data. It does not draw, read files, or print, so it can be driven equally from a script, a batch job, or a browser-side application compiled to WebAssembly.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

Imports stats, utils

Suggests testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

Roxygen list(markdown = TRUE)

LazyData true

URL <https://github.com/pdelias/spectralDEMATEL>

BugReports <https://github.com/pdelias/spectralDEMATEL/issues>

Config/roxygen2/version 8.0.0

Repository <https://pdelias.r-universe.dev>

Date/Publication 2026-08-09 08:29:17 UTC

RemoteUrl <https://github.com/pdelias/spectralDEMATEL>

RemoteRef HEAD

RemoteSha e1e6af59390c6ee7476b21bed5b673fdc17a17fc

Contents

assumption_checks	2
dematel	3
is_irreducible	4
measurement_stability	5
sensitivity_matrix	6
spectral_diagnostics	8
structural_type	10
surrogate_ensemble	12
surrogate_position	13
tradeoff_residual	15
type_stability	16
worked_matrices	17
Index	18

assumption_checks	<i>Check a matrix against the assumptions the diagnostics rest on</i>
-------------------	---

Description

Every condition is returned as a row of data: a verdict, an identifier stable enough to branch on, a reason written for a user rather than a developer, and where possible the factors at fault. Nothing here stops, warns or prints.

Usage

```
assumption_checks(A, type = c("A", "T"))
```

Arguments

- | | |
|------|---|
| A | The matrix to check. Anything may be passed; a non-matrix fails the first check rather than raising an error. |
| type | Whether A is a direct influence matrix ("A") or an already-published total-relation matrix ("T"). Two checks do not apply to a total-relation matrix and are reported as skipped. |

Value

- A data frame of class `spectral_checks`, one row per check, in a fixed order, with columns:
- check Stable identifier. Branch on this, never on reason.
 - verdict One of "pass", "warn", "fail", "skipped".
 - reason Plain-language explanation, and what to do about it.
 - value The number behind the verdict, or NA where there is none.
 - factors List column of integer indices of the factors at fault, empty where the check is not about particular factors.

The four verdicts

pass and fail are the assumption holding or not. warn is the case a binary verdict cannot express: the matrix is in scope, but it is close enough to a boundary that the reading is worth less than it looks. Strong connectivity is yes or no, but a graph one edge from disconnection behaves differently from a dense one, and numerical conditioning degrades long before any check trips.

skipped means the check could not be evaluated because something it depends on failed first — there is no point testing connectivity on a matrix that is not square. It is not a pass, and an interface must not render it as one.

A failure is not a refusal

The engine computes diagnostics for a matrix that fails these checks, and `spectral_diagnostics()` returns the checks alongside the numbers so the two cannot be separated. One of the two matrices in `worked_matrices` fails strong connectivity and its diagnostics are published. Deciding what to do about a failure belongs to whoever is calling this.

See Also

`spectral_diagnostics()`, which returns these alongside the numbers.

Examples

```
checks <- assumption_checks(worked_matrices$resilience_capabilities)
checks[checks$verdict != "pass", c("check", "verdict", "reason")]

# Which factor is at fault:
checks$factors[checks$check == "strong_connectivity"][[1]]
```

dematel	<i>Normalise a direct influence matrix and compute the total-relation matrix</i>
---------	--

Description

Divides A by the larger of its maximum row sum and its maximum column sum to obtain the normalised matrix D, then sums the influence propagated over paths of every length to obtain the total-relation matrix $T = D(I - D)^{-1}$.

Usage

```
dematel(A)
```

Arguments

A A square, non-negative numeric matrix of direct influences.

Value

A list with components A (the input), s (the normalising constant), D (the normalised matrix) and T (the total-relation matrix). Returns NULL invisibly for any input it cannot process.

Why it returns NULL instead of failing

A user pasting an asymmetric matrix is not an exceptional condition, it is Tuesday. Every rejection here is silent and recoverable, and `assumption_checks()` is where the reason lives — as data, in plain language, naming the factors at fault. Nothing in this package raises a condition for something a user can produce.

NULL is returned when A is not a numeric matrix, is not square, holds a missing, infinite or negative entry, is entirely zero, or meets the degenerate case below.

The degenerate case

When every row and column total is the same, the spectral radius of D is exactly 1, $I - D$ is singular and T is undefined (Lee et al. 2013). A matrix in which every pair was rated identically always meets it.

See Also

`assumption_checks()`, which reports why an input was rejected.

Examples

```
A <- matrix(c(0, 2, 0,
              0, 0, 3,
              4, 0, 0), 3, 3, byrow = TRUE)
m <- dematel(A)
m$s
round(m$T, 4)

# Degenerate input returns NULL rather than failing:
is.null(dematel(matrix(2, 4, 4)))
```

is_irreducible

Strong connectivity of the influence graph (assumption A2)

Description

Tests whether the directed graph whose edges are the positive entries of A is strongly connected, that is, whether every factor can reach every other factor along some directed path. This is assumption A2, and it is what makes the dominant eigenvector unique and strictly positive.

Usage

```
is_irreducible(A)
```

Arguments

A A square numeric matrix. Only the sign pattern is used.

Value

TRUE if the influence graph is strongly connected, FALSE otherwise.

Failure is common and is not fatal

A single factor that dispatches nothing gives an all-zero row and fails the test. One of the two matrices in [worked_matrices](#) fails it, and its diagnostics are published regardless. Use `assumption_checks()` to learn *which* factors are stranded.

Examples

```
cycle <- matrix(c(0, 1, 0,
                  0, 0, 1,
                  1, 0, 0), 3, 3, byrow = TRUE)
is_irreducible(cycle)

# A factor that influences nothing breaks strong connectivity.
sink_node <- cycle
sink_node[3, ] <- 0
is_irreducible(sink_node)
```

measurement_stability *Does the type survive the noise expert ratings carry?*

Description

Perturbs the influence ratings by a stated tolerance and recomputes the structural type. Expert judgements on a four- or five-point scale carry at least half a point of noise, and a classification that does not survive that is not a classification.

Usage

```
measurement_stability(
  A,
  tolerance = 0.5,
  B = 200,
  seed = NULL,
  perturb = c("nonzero", "all"),
  coupling_cut = STRUCTURE_CUTS$coupling,
  hierarchy_cut = STRUCTURE_CUTS$hierarchy
)
```

Arguments

A	A square, non-negative numeric matrix of direct influences.
tolerance	Half-width of the perturbation, in rating-scale points. The default 0.5 is the noise a four- or five-point expert scale is generally taken to carry.
B	Number of perturbed matrices to draw.
seed	Optional integer, so a result can be reproduced.
perturb	Which entries to disturb. "nonzero", the default, leaves recorded zeros alone; "all" disturbs every off-diagonal entry.
coupling_cut, hierarchy_cut	Passed to <code>structural_type()</code> .

Value

A list, or NULL when the matrix cannot be diagnosed: `observed_type`, `share_same` (proportion of perturbed matrices keeping that type), `type_table`, `B_admissible` (draws that could be diagnosed at all), `tolerance` and `perturb`.

Why recorded zeros are left alone by default

A rating of zero is a judgement that there is no influence between two factors, which is a different kind of statement from a rating of one carrying measurement error. Disturbing zeros upward also fills in the matrix, which destroys the sparsity the connectivity checks are about. `perturb = "all"` is available for anyone who disagrees, and the choice is reported back.

Perturbed values are clamped to $[0, \max(A)]$: a rating cannot go negative, and nothing should exceed the top of the scale the analyst used.

Examples

```
ms <- measurement_stability(worked_matrices$fefo_stock_control,
                           B = 40, seed = 1)
ms$observed_type
ms$share_same
```

sensitivity_matrix	<i>Per-link sensitivity of the dominant eigenvalue</i>
--------------------	--

Description

Returns $\partial \lambda_{max} / \partial a_{ij}$ for every off-diagonal entry of the direct influence matrix, split into the two effects that produce it.

Usage

```
sensitivity_matrix(A)
```

Arguments

A A square, non-negative numeric matrix of direct influences.

Details

This moves the question from "which factor matters" to "which relationship, if strengthened or weakened, moves the system most", which is much closer to what an intervention actually is. The derivative is signed: entries with a negative derivative *lower* coupling when strengthened, and are the levers for a system someone wants to calm rather than drive.

Value

A list of three n by n matrices, or NULL when `dematel()` found the degenerate case. Diagonals are NA throughout, since a self-loop is not an actionable link.

local The effect of the entry on the propagation structure, holding the normalising constant fixed. Always positive.

normalization The effect of the entry on the normalising constant s . Non-zero only in the row or column that currently sets s , and always negative there.

total Their sum, which is the quantity to rank on.

Why the split is not decoration

s is the larger of the maximum row sum and the maximum column sum. Increasing an entry that sits in the row setting s raises s too, which shrinks the whole normalised matrix. The two effects oppose each other, and in that row the normalisation effect can win — so raising an influence rating there *lowers* the dominant eigenvalue. A user shown only the total will read that as a bug. Showing the components names it as a mechanism.

Never show this without the condition number

These are first-order estimates. `ev_condition` from `spectral_diagnostics()` bounds how far they can be trusted, and large values mean the derivative is locally uninformative. The ranking without it misleads, so the two ship together or neither ships.

Examples

```
A <- matrix(c(0, 2, 1,
              3, 0, 1,
              1, 2, 0), 3, 3, byrow = TRUE)
s <- sensitivity_matrix(A)

# The single most effective link to strengthen:
which(s$total == max(s$total, na.rm = TRUE), arr.ind = TRUE)

# How far to trust that: 1 is ideal, large is uninformative.
spectral_diagnostics(A)$ev_condition
```

spectral_diagnostics *Spectral diagnostics for one DEMATEL system*

Description

Computes the full diagnostic set from a single influence matrix: how strongly the system amplifies its own influence, whether one propagation pattern dominates, how concentrated the entry points are, where influence enters and where it accumulates, and how far a first-order sensitivity estimate can be trusted.

Usage

```
spectral_diagnostics(A, type = c("A", "T"), checks = TRUE)
```

Arguments

A	A square numeric matrix. Either a direct influence matrix (type = "A", the default) or an already-published total-relation matrix (type = "T").
type	Which matrix A is. "A" normalises and builds T through <code>dematel()</code> . "T" takes A as the total-relation matrix and recovers the spectrum of D through the inverse Moebius map $\mu = \lambda/(1 + \lambda)$.
checks	Whether to run <code>assumption_checks()</code> and attach the result. Defaults to TRUE and should stay that way for anything a person will read. FALSE returns the numbers alone, or NULL if they cannot be computed, and exists for inner loops such as <code>surrogate_ensemble()</code> .

Value

With checks = TRUE, a named list that always has the same shape, whatever was passed in. With checks = FALSE, the same list without checks and engine_version, or NULL when the matrix is inadmissible.

n Number of factors.

mu_max **Coupling**, in $(0, 1)$. The spectral radius of the normalised matrix, so the distance to criticality. Higher means more strongly coupled.

multiplier Total-to-direct multiplier, $1/(1 - \mu_{max})$. Always greater than 1.

lambda_max Dominant eigenvalue of the total-relation matrix.

indirect_dominant TRUE when indirect effects exceed direct ones, which happens exactly when $\lambda_{max} > 1$, equivalently when $\mu_{max} > 0.5$.

dominance **Mode dominance**, $|\lambda_2|/\lambda_{max}$ in $[0, 1)$. Low means a single propagation mode; high means a second mode competes with the first.

hierarchy_sd Hierarchy as the standard deviation of the unit-norm right eigenvector. **HIGH means concentrated.**

hierarchy_pr Hierarchy as the participation ratio divided by n, in $(0, 1]$. **LOW means concentrated** — the opposite direction to the other two.

`hierarchy_gini` Hierarchy as the Gini coefficient of the same vector. **HIGH means concentrated.** Size-free.

`entry_points` Right eigenvector, unit 2-norm. Weights each factor by the influence it injects: where to apply pressure.

`accumulation` Left eigenvector, unit 2-norm. Weights each factor by what accumulates there: where effects land.

`ev_condition` Eigenvalue condition number, $1/|v'u|$. At least 1 by construction. Bounds how far the first-order sensitivity estimates of `sensitivity_matrix()` can be trusted; large values mean the derivative is locally uninformative.

`checks` The `assumption_checks()` table for this matrix.

`engine_version` The package version that produced the diagnosis.

The numbers never travel without their checks

A diagnosis computed on a matrix that failed a check has to carry that fact with it, so `checks` is part of the returned object rather than something the caller was told separately and may not have kept. When the matrix is inadmissible every numeric field is NA and `checks` says why — the shape does not change, so a batch job over a hundred matrices gets a rectangular result and never a failure.

`engine_version` is recorded because a metric definition has moved here before. When one moves again, old diagnoses must remain interpretable rather than silently mean something new.

The three hierarchy readings run in different directions

`hierarchy_sd` and `hierarchy_gini` are high when influence enters at a few factors. `hierarchy_pr` is **low** when influence enters at a few factors. This project has inverted the direction once already, in code that had been reviewed, and the mistake produced entirely plausible output. Anything displaying one of these must state which reading it used and which way it runs.

`hierarchy_sd` is the reading the source paper's corpus analysis uses throughout, so it is the one behind every corpus-level result including the coupling-hierarchy trade-off.

Mode dominance uses the largest modulus

λ_2 is the largest-modulus eigenvalue below the dominant one, not the second largest real part. For a non-negative T the negative end of the spectrum usually carries the larger modulus, so taking the real part understates the ratio nearly everywhere — while passing every plausibility check.

See Also

`assumption_checks()` for the checks alone, `sensitivity_matrix()` for where to intervene, `surrogate_ensemble()` for a baseline to read these numbers against.

Examples

```
A <- matrix(c(0, 2, 0,
              0, 0, 3,
              4, 0, 0), 3, 3, byrow = TRUE)
d <- spectral_diagnostics(A)
d$mu_max
```

```

d$multiplier
d$indirect_dominant

# Anything that failed, and why:
subset(d$checks, verdict != "pass", c(check, verdict))

# Inadmissible input keeps the shape and explains itself:
bad <- spectral_diagnostics(matrix(2, 4, 4))
bad$mu_max
bad$checks$reason[bad$checks$check == "totals_vary"]

```

structural_type	<i>Name a system's structural type, and say how firmly</i>
-----------------	--

Description

Places a diagnosed system on the (coupling, hierarchy) map and returns the type, the margin to each cut in that axis's own units, and the intervention logic the source paper pairs with the type.

Usage

```

structural_type(
  d,
  coupling_cut = STRUCTURE_CUTS$coupling,
  hierarchy_cut = STRUCTURE_CUTS$hierarchy
)

```

Arguments

d	A diagnostics list from spectral_diagnostics() .
coupling_cut	Where amplified meets dampened. The default 0.5 is the indirect-dominance threshold and follows from the algebra; changing it makes the type mean something other than what the corpus results mean.
hierarchy_cut	Where hierarchical meets diffuse. The default 0.10 is a recommendation, not a fitted constant, and a user may move it.

Value

A list, or NULL when the diagnostics could not be computed:

type	One of "diffuse-amplified", "hierarchical-amplified", "hierarchical-dampened", "diffuse-dampened".
coupling_margin	Signed distance from coupling_cut. Positive is amplified.
hierarchy_margin	Signed distance from hierarchy_cut. Positive is hierarchical.
nearest_cut	Which boundary the system sits closest to, once each margin is expressed in units of that axis's corpus spread.

`nearest_margin_scaled` That distance, in corpus interquartile ranges. Small means the type is a close call.

`corpus_share` Proportion of the 117 reference systems of this type, so a user in an uncommon corner is told so.

`corpus_multiplier` Median total-to-direct multiplier within the type across the corpus.

`intervention_logic` The source paper's wording for this type.

`caveat` What that pairing is and is not. Show it wherever the logic is shown.

`cuts` The two cuts actually used, and whether each was the default.

`hierarchy_reading` Always "hierarchy_sd". See below.

Only the standard-deviation reading classifies

The 0.10 cut is calibrated against `hierarchy_sd`, and every corpus-level result is expressed in it. `hierarchy_gini` and `hierarchy_pr` measure the same idea on different scales – and the participation ratio runs the opposite way – so a cut of 0.10 means nothing on either. Show all three readings by all means; classify on one.

The margin is the point

A system at coupling 0.51 and one at 0.95 are both amplified, and the confidence in the advice is not remotely the same. This function returns the distance and leaves the wording to the caller, because how much to hedge is a presentation decision.

The advice is a hypothesis

`intervention_logic` is quoted from the source paper, where each pairing is explicitly a hypothesis derived from structure and not a validated result. An interface that renders it as an instruction will be read as stronger than the evidence. `caveat` exists to be shown, not stored.

See Also

[`type\_stability\(\)`](#) for whether the type survives moving the cut, [`tradeoff\_residual\(\)`](#) for how the system sits against the corpus trade-off.

Examples

```
d <- spectral_diagnostics(worked_matrices$fefo_stock_control)
s <- structural_type(d)
s$type
s$coupling_margin
s$intervention_logic

# An uncommon corner is worth knowing about:
structural_type(spectral_diagnostics(worked_matrices$resilience_capabilities))$corpus_share
```

surrogate_ensemble	<i>Permutation surrogate baseline</i>
--------------------	---------------------------------------

Description

Shuffles the off-diagonal entries of A while holding the number of factors, the density and the exact multiset of entry values fixed, then recomputes the diagnostics. Repeating this gives a null distribution against which an observed diagnostic can be read.

Usage

```
surrogate_ensemble(A, B = 200, seed = NULL, max_attempts = 50L * B)
```

Arguments

A	A square, non-negative numeric matrix of direct influences.
B	Ensemble size requested. Draws that fail strong connectivity or hit the degenerate case are rejected and redrawn.
seed	Optional integer. Sets the random seed before drawing, so a figure can be reproduced. Passing NULL leaves the caller's random state alone.
max_attempts	Maximum number of shuffles to draw before giving up, default 50 * B. Reaching it returns a short ensemble rather than continuing; see the section below.

Details

The question it answers is the one a single number cannot: how much of the structure is something the analyst built, and how much follows from the entry distribution on its own. A system whose coupling sits in the middle of its own surrogate ensemble has coupling because of how its ratings were distributed, not because of how its factors were connected.

Value

A data frame with columns `mu_max`, `lambda_max`, `dominance`, `hierarchy_sd`, `hierarchy_pr` and `hierarchy_gini`, carrying three attributes: `requested` (the B asked for), `attempts` (shuffles drawn) and `complete` (whether B rows were reached). Returns NULL if no draw was admissible, matching [spectral_diagnostics\(\)](#) on inadmissible input.

The row count is not guaranteed to be B. Read `complete` before reporting anything derived from the ensemble.

Cost

This is the one part of the package that is not instant. Each draw is an eigendecomposition plus a connectivity check, so B = 200 is two hundred of each. At DEMATEL sizes that is still fast, but it is the only function here that should be run on demand rather than eagerly.

When rejection dominates

A matrix sparse enough that most shuffles disconnect the graph rejects most draws. A directed cycle is the extreme case: its edges are exactly as many as strong connectivity needs, so almost every rearrangement of them breaks it.

The original implementation looped until it had B admissible draws with no cap, which on such a matrix does not terminate in any useful time. In a script that is a long wait; compiled to WebAssembly it is a frozen browser tab with no way to cancel, reachable by pasting a sparse matrix.

So the loop is bounded. Hitting the bound is not an error and does not throw: the function returns the draws it did get and says so in complete, on the principle that a short null distribution the caller knows about is more useful than no answer and far more useful than a hang. $50 * B$ tolerates an acceptance rate down to about 2% before it bites.

Examples

```
set.seed(1)
A <- matrix(sample(0:4, 36, replace = TRUE), 6, 6)
diag(A) <- 0
storage.mode(A) <- "double"
if (is_irreducible(A)) {
  ens <- surrogate_ensemble(A, B = 20, seed = 42)
  observed <- spectral_diagnostics(A)$mu_max
  mean(ens$mu_max >= observed) # where the observation sits in the null
}
```

surrogate_position	<i>Where the observed diagnostics sit in their own surrogate ensemble</i>
--------------------	---

Description

Shuffles the off-diagonal entries, holding the number of factors, the density and the exact multiset of values fixed, and asks where the real system falls in the resulting null. A diagnosis is worth less if the same numbers would arise from the rating distribution on its own.

Usage

```
surrogate_position(
  A,
  B = 200,
  seed = NULL,
  coupling_cut = STRUCTURE_CUTS$coupling,
  hierarchy_cut = STRUCTURE_CUTS$hierarchy,
  max_attempts = 50L * B
)
```

Arguments

A	A square, non-negative numeric matrix of direct influences.
B	Ensemble size. 200 is the paper's figure; lower it when a person is waiting.
seed	Optional integer, so a figure in a paper can be reproduced.
coupling_cut, hierarchy_cut	Passed to structural_type() when classifying each draw.
max_attempts	Maximum number of shuffles drawn before giving up, default $50 * B$, passed to surrogate_ensemble() . Reaching it yields a short ensemble and <code>complete = FALSE</code> rather than an error.

Details

This is the comparison the source paper could not run across its corpus, because the precomputed spectral file holds no raw matrices. A user of this package has their own matrix, so they can.

Value

A list, or NULL when the matrix cannot be diagnosed:

`metrics` Data frame, one row per diagnostic, with the observed value, the ensemble median, minimum and maximum, `share_ge` (the proportion of draws at or above the observed value) and `outside` (TRUE when the observation falls beyond the whole ensemble).

`type` The observed structural type.

`type_share` Proportion of surrogate draws that produce the same type. High means the type follows from the rating distribution rather than from the structure the analyst built.

`type_table` Counts of every type across the ensemble.

`B` Ensemble size actually used.

`requested` Ensemble size asked for.

`complete` TRUE when B reached requested.

`attempts` Shuffles drawn, accepted and rejected together.

Returns NULL when the matrix is inadmissible, when it is not irreducible, or when no shuffle of it was admissible.

Read complete before quoting a share

A sparse matrix rejects most shuffles, so the ensemble can come back shorter than requested; see [surrogate_ensemble\(\)](#). A `share_ge` computed over seven draws and one computed over two hundred are not the same claim, and B alone does not distinguish them from what the caller asked for.

Reading share_ge

It is a position, not a p-value. `share_ge = 0` means the observation exceeds every draw; `share_ge = 1` means every draw exceeds it. Values near 0.5 mean the shuffle reproduces the observation, so that diagnostic says more about how the ratings were distributed than about how the factors were connected.

type_share is weak evidence for a common type

Nearly 70% of the reference corpus is diffuse-amplified, so a shuffled matrix usually lands there too. A type_share near 1 for a common type therefore means very little – it says the null also produces that type, not that the observation is unremarkable. The per-metric positions are the informative part. type_share earns its keep when the observed type is an uncommon one, where a high value is a genuine warning that the rating distribution alone would have put the system in that corner.

See Also

[surrogate_ensemble\(\)](#) for the raw draws.

Examples

```
sp <- surrogate_position(worked_matrices$fefo_stock_control, B = 40, seed = 1)
sp$metrics[, c("metric", "observed", "share_ge", "outside")]
sp$type_share
```

tradeoff_residual	<i>Where the system sits against the corpus coupling-hierarchy trade-off</i>
-------------------	--

Description

Coupling and hierarchy oppose each other across the reference corpus, coupling accounting for about 59% of the variation in hierarchy. A system's residual from that relation says whether it has unusually concentrated leverage for how strongly it is coupled.

Usage

```
tradeoff_residual(d)
```

Arguments

d A diagnostics list from [spectral_diagnostics\(\)](#).

Value

A list, or NULL when the diagnostics could not be computed: expected (hierarchy predicted from this system's coupling), residual, residual_sd (that residual in units of the corpus residual standard deviation), direction ("above" or "below" the line), and caveat.

A supplement, never a substitute

Used on its own to assign types, the trade-off is wrong for 19 of the 117 reference systems, and wrong where it counts: 10 systems with a dominant point of entry would be told to spread their effort, and all 8 without one would be sent to look for one. Read this beside the computed hierarchy, never instead of it.

Examples

```
d <- spectral_diagnostics(worked_matrices$resilience_capabilities)
r <- tradeoff_residual(d)
round(r$residual_sd, 2)
r$direction
```

type_stability

Does the type survive moving the hierarchy cut?

Description

The hierarchy cut is a recommendation rather than a fitted constant, so the honest question is not "which side is this system on" but "does the answer change across the range anyone might reasonably choose". For one matrix that is a handful of classifications and a sentence.

Usage

```
type_stability(
  d,
  hierarchy_cuts = c(0.075, 0.09, 0.1, 0.11),
  coupling_cut = STRUCTURE_CUTS$coupling
)
```

Arguments

d A diagnostics list from `spectral_diagnostics()`.

hierarchy_cuts Cuts to try. The default spans the range the source paper's supplement re-runs its whole corpus assignment across.

coupling_cut Held fixed: it is not a free parameter.

Value

A list, or NULL when the diagnostics could not be computed:

by_cut Data frame of cut and the resulting type.

stable TRUE when every cut gives the same type.

flips_between The two adjacent cuts the type changes between, or NA when it does not change.

Examples

```
d <- spectral_diagnostics(worked_matrices$fefo_stock_control)
s <- type_stability(d)
s$stable
s$by_cut
```

worked_matrices

Two transcribed DEMATEL influence matrices

Description

The two worked systems from the source paper's Section 5, transcribed from their published tables and verified against those papers' own printed total-relation matrices. They are the package's regression fixtures and the worked examples in the vignette.

Usage

```
worked_matrices
```

Format

A named list of two numeric matrices. Each carries a `doi` attribute and a `label` attribute.

`fefo_stock_control` 13 factors, DOI 10.1016/j.jafr.2025.101848. Strongly coupled ($\mu_{\max}$ 0.828), diffuse, single-mode. Has a **non-zero diagonal**, which makes it a useful case for input-form detection.

`resilience_capabilities` 15 factors, DOI 10.1007/s12063-024-00470-8. Weakly coupled ($\mu_{\max}$ 0.351), hierarchical. **Fails strong connectivity**, stranding four factors: row 14 is blank so factor 14 dispatches nothing, column 7 is blank so factor 7 receives nothing, and factors 8 and 4 are reachable only through 7. The source paper reports its diagnostics regardless, which is why an assumption check has to be a returned verdict rather than a refusal.

Details

Between them they straddle every threshold the diagnostics use: coupling above and below 0.5, hierarchy above and below 0.10, indirect dominance both ways, and strong connectivity both ways.

Source

Transcribed in the JDS paper repository, `data/raw/matrices/`.

Examples

```
names(worked_matrices)
attr(worked_matrices$fefo_stock_control, "doi")

d <- spectral_diagnostics(worked_matrices$fefo_stock_control)
d$mu_max

# The other one is not strongly connected:
is_irreducible(worked_matrices$resilience_capabilities)
```

Index

* datasets

worked_matrices, [17](#)

assumption_checks, [2](#)

assumption_checks(), [4](#), [8](#), [9](#)

dematel, [3](#)

dematel(), [7](#), [8](#)

is_irreducible, [4](#)

measurement_stability, [5](#)

sensitivity_matrix, [6](#)

sensitivity_matrix(), [9](#)

spectral_diagnostics, [8](#)

spectral_diagnostics(), [3](#), [7](#), [10](#), [12](#), [15](#), [16](#)

structural_type, [10](#)

structural_type(), [6](#), [14](#)

surrogate_ensemble, [12](#)

surrogate_ensemble(), [8](#), [9](#), [14](#), [15](#)

surrogate_position, [13](#)

tradeoff_residual, [15](#)

tradeoff_residual(), [11](#)

type_stability, [16](#)

type_stability(), [11](#)

worked_matrices, [3](#), [5](#), [17](#)