

Package: probmed (via r-universe)

October 14, 2026

Title Probabilistic Effect Sizes for Causal Mediation Analysis

Version 0.3.0

Description Compute P_med, a scale-free probabilistic effect size for causal mediation analysis.

License MIT + file LICENSE

URL <https://data-wise.github.io/probmed/>,
<https://github.com/data-wise/probmed>

BugReports <https://github.com/data-wise/probmed/issues>

Encoding UTF-8

Language en-US

Roxygen list(markdown = TRUE)

Suggests alabama, EValue (>= 4.1.0), knitr, lavaan, MASS, mediation, rmarkdown, testthat (>= 3.0.0), quarto

Config/testthat/edition 3

Imports medfit (>= 0.3.0), methods, S7 (>= 0.2.1), stats

Remotes data-wise/medfit@v0.3.0

VignetteBuilder knitr, quarto

RoxygenNote 7.3.3

Repository <https://data-wise.r-universe.dev>

Date/Publication 2026-06-22 04:46:18 UTC

RemoteUrl <https://github.com/Data-Wise/probmed>

RemoteRef main

RemoteSha 0486b012b20938cac1ae9703c5e971503f086d7a

Contents

extract_mediation	2
GaugePmedResult	3
incr_pmed	4
IncrPmedResult	5
plot-PmedResult	6
pmed	6
pmed-formula	8
pmed-MediationData	9
pmed-ParallelMediationData	10
pmed_sensitivity	10
PmedResult	12
PmedSensitivity	13
print-PmedResult	14
sobol_from_theta	14
sobol_pmed	15
SobolPmedResult	17
summary-PmedResult	20
ward_residual	20
Index	23

extract_mediation	<i>Re-export extract_mediation from medfit</i>
-------------------	--

Description

Generic function to extract mediation structure from fitted models. This function is provided by the medfit package and re-exported here for convenience.

Arguments

- object Fitted model object
- ... Additional arguments passed to methods

Value

A medfit::MediationData object

See Also

[extract_mediation](#) for full documentation

GaugePmedResult

*Gauge-Calibrated Proportion Mediated Result***Description**

S7 class for the gauge-calibrated proportion mediated. Reports the interventional proportion mediated $P_{\text{med}} = \text{IIE} / \text{OE}$ alongside the **gauge residual** $W = R / \text{OE}$, where $R = \text{OE} - \text{IDE} - \text{IIE}$ is the non-decomposability (treatment-by-mediator interaction) term. A W significantly different from zero signals that the additive split of the overall effect fails and the single-number P_{med} is unreliable.

Usage

```
GaugePmedResult(
  p_med = integer(0),
  p_med_ci = integer(0),
  p_med_fieller = numeric(0),
  fieller_type = NA_character_,
  W = integer(0),
  W_ci = integer(0),
  W_se = integer(0),
  W_p = integer(0),
  OE = integer(0),
  IDE = integer(0),
  IIE = integer(0),
  R = integer(0),
  theta = integer(0),
  method = character(0),
  n = integer(0),
  ci_level = integer(0),
  se_method = "analytic",
  reps = 1L,
  call = NULL
)
```

Arguments

<code>p_med</code>	Numeric: interventional proportion mediated, IIE / OE .
<code>p_med_ci</code>	Numeric length-2: confidence interval for <code>p_med</code> .
<code>p_med_fieller</code>	Numeric: Fieller confidence set for the ratio <code>p_med</code> (may be unbounded; empty if not requested).
<code>fieller_type</code>	Character: type of Fieller set returned (e.g. bounded, unbounded, or empty); NA if not computed.
<code>W</code>	Numeric: gauge residual R / OE .
<code>W_ci</code>	Numeric length-2: confidence interval for <code>W</code> .

W_se	Numeric: standard error of W.
W_p	Numeric: two-sided p-value for $H_0: W = 0$.
OE, IDE, IIE, R	Numeric: overall, interventional direct, interventional indirect effects and the remainder.
theta	Numeric length-4: corner means $\theta(a, a')$.
method	Character: estimation method.
n	Integer: sample size.
ci_level	Numeric: confidence level.
se_method	Character: "analytic" (influence-function se) or "bootstrap" (nonparametric, valid but mildly conservative).
reps	Integer: number of repeated cross-fitting fold draws averaged for the point estimate.
call	Call: original call.

incr_pmed

Incremental Mediated Elasticity Curve

Description

Estimate the incremental mediated elasticity $P_{med}^{\Delta}(\Delta)$ – the derivative-scale proportion mediated as a function of the treatment-tilt factor Δ . The estimator is cross-fitted and built on the same triply-robust corner pseudo-outcomes as [ward_residual\(\)](#); standard errors use the ratio-identity influence function for med / tot at each Δ .

Usage

```
incr_pmed(
  object,
  deltas = c(1/3, 1/2, 1, 2, 3),
  covars = "C",
  K = 5L,
  ci_level = 0.95,
  seed = 1L,
  ...
)
```

Arguments

object	A data frame with columns A (binary treatment), M (mediator), Y (outcome), and the covariates named in covars.
deltas	Numeric: tilt factors at which to evaluate the curve.
covars	Character: covariate column names (default "C").
K	Integer: number of cross-fitting folds (default 5).
ci_level	Numeric: confidence level (default 0.95).
seed	Integer: random seed for fold assignment (default 1).
...	Unused.

Details

At each δ the tilted assignment probability is $q = \delta * g / (\delta * g + 1 - g)$ with derivative $q' = g(1 - g) / (\delta * g + 1 - g)^2$, where $g = P(A = 1 | C)$. The direct and mediated elasticities are weighted contrasts of the four corner pseudo-outcomes by q and q' ; by Theorem 1 (multivariate chain rule) they sum to the total elasticity exactly, so $P_{\text{med}}^{\delta} = \text{med} / (\text{dir} + \text{med})$.

Inference. Point estimates use the tilt-derivative weight $q' = g(1 - g) / (\delta * g + 1 - g)^2$ (Kennedy 2019, Corollary 2, Term I). Standard errors use the full efficient influence function for the ratio $P_{\text{med}} = \text{med} / \text{tot}$, which adds the g -score correction $(dq/dg) * (A - g(C)) * (\text{dir} * \gamma_{\text{med}} - \text{med} * \gamma_{\text{dir}}) / \text{tot}^2$ with $dq/dg = \delta / (\delta * g + 1 - g)^2$ (Term II). Term II is mean-zero (by $E[A - g(C) | C] = 0$), so it does not shift the point estimate but restores Neyman orthogonality w.r.t. the propensity score, ensuring the CI is consistent under nonparametric estimation of g .

Value

An [IncrPmedResult](#) object whose `@curve` slot holds one row per δ with the direct, mediated, and total elasticities, the proportion mediated P_{med} , and its standard error and Wald interval.

Examples

```
set.seed(1)
n <- 1500
C <- rnorm(n)
A <- rbinom(n, 1, plogis(-0.2 + 0.8 * C))
M <- 0.6 * A + 0.4 * C + rnorm(n)
Y <- 0.5 * A + 0.7 * M + 0.3 * C + rnorm(n)
d <- data.frame(A, M, Y, C)
incr_pmed(d, deltas = c(0.5, 1, 2))
```

IncrPmedResult

Incremental Mediated Elasticity Result

Description

S7 class for the **incremental mediated elasticity** $P_{\text{med}}^{\delta}(\delta)$ (Paper 2 of the `P_med-for-modern-estimands` program). Rather than a single number, the estimand is a *curve*: the share of the marginal (derivative-scale) total effect that flows through the mediator when the treatment-assignment odds are tilted by a factor δ .

Writing $\Theta(\delta_1, \delta_2)$ for the mean outcome under a δ_1 -tilt of the direct pathway and a δ_2 -tilt of the mediated pathway, the elasticities are $\theta'_{\text{dir}} = d/d \delta_1 \Theta$, $\theta'_{\text{med}} = d/d \delta_2 \Theta$, and $\theta'_{\text{tot}} = d/d \delta \Theta(\delta, \delta)$. The multivariate chain rule gives the **exact decomposition** $\theta'_{\text{tot}} = \theta'_{\text{dir}} + \theta'_{\text{med}}$ (remainder $R(\delta) = 0$ for every δ), and $P_{\text{med}}^{\delta}(\delta) = \theta'_{\text{med}} / \theta'_{\text{tot}}$.

Under a linear-Gaussian model with no $A * M$ interaction the curve is flat and equals the classical proportion mediated; it bends with δ only when an interaction is present.

Usage

```
IncrPmedResult(  
  curve = data.frame(),  
  deltas = integer(0),  
  method = character(0),  
  n = integer(0),  
  ci_level = integer(0),  
  call = NULL  
)
```

Arguments

curve	Data frame: one row per delta, columns delta, dir, med, tot, Pmed, se, lo, hi.
deltas	Numeric: the tilt factors at which the curve was evaluated.
method	Character: estimation method.
n	Integer: sample size.
ci_level	Numeric: confidence level.
call	Call: original call.

plot-PmedResult	<i>Plot Bootstrap Distribution</i>
-----------------	------------------------------------

Description

Plot Bootstrap Distribution

pmed	<i>Compute P_med: Probabilistic Effect Size for Mediation</i>
------	---

Description

Compute P_{med} , a scale-free probabilistic effect size for mediation analysis, along with the traditional Indirect Effect ($a \times b$). Provides point estimates and bootstrap confidence intervals.

P_{med} represents $P(Y(x, M(x)) > Y(x, M(x^*))) + \frac{1}{2}P(Y(x, M(x)) = Y(x, M(x^*)))$: the probability that the outcome under treatment with the mediator at its treated level exceeds the outcome under treatment with the mediator at its control level (a one-half correction is added for ties). Both potential outcomes hold treatment at x , so the direct effect cancels and P_{med} reflects mediation only; values > 0.5 indicate positive mediation.

Usage

```
pmed(object, ...)
```

Arguments

object Either:

- A formula for the outcome model (most common)
- A `medfit::MediationData` object from `extract_mediation()`

... Additional arguments passed to methods (see method documentation)

Value

`PmedResult` object containing:

- `estimate`: `P_med` point estimate
- `ci_lower`, `ci_upper`: Confidence interval bounds
- `ie_estimate`: Indirect Effect point estimate
- `ie_ci_lower`, `ie_ci_upper`: IE confidence interval
- `boot_estimates`: Bootstrap distribution (if applicable)
- `method`: Inference method used

Methods

Available inference methods via `method` argument:

- `"parametric_bootstrap"` (default): Fast, assumes normality
- `"nonparametric_bootstrap"`: Robust to assumptions, slower
- `"plugin"`: Point estimate only, no CI

See Also

[extract_mediation](#) for extracting from fitted models

Examples

```
# Basic example with formula interface
set.seed(123)
n <- 200
data <- data.frame(X = rnorm(n), C = rnorm(n))
data$M <- 0.5 * data$X + 0.3 * data$C + rnorm(n)
data$Y <- 0.4 * data$M + 0.2 * data$X + 0.2 * data$C + rnorm(n)

# Compute with parametric bootstrap
result <- pmed(
  Y ~ X + M + C,
  formula_m = M ~ X + C,
  data = data,
  treatment = "X",
  mediator = "M",
  method = "parametric_bootstrap",
  n_boot = 500
)
```

```

print(result)

## Not run:
# View bootstrap distribution
summary(result)
plot(result)

## End(Not run)

```

pmed-formula

Compute P_med from Formula

Description

Compute P_med from Formula

Arguments

object	Formula for outcome model ($Y \sim X + M + C$)
formula_m	Formula for mediator model ($M \sim X + C$)
data	Data frame
treatment	Character: treatment variable name
mediator	Character: mediator variable name
family_y	Family for outcome model (default: gaussian())
family_m	Family for mediator model (default: gaussian())
x_ref	Reference treatment value (default: 0)
x_value	Treatment value (default: 1)
method	Inference method: "parametric_bootstrap", "nonparametric_bootstrap", "plugin", "mbco"
n_boot	Number of bootstrap samples (default: 1000)
ci_level	Confidence level (default: 0.95)
seed	Random seed for reproducibility
...	Additional arguments

Details

method = "mbco" returns a Model-Based Constrained Optimization interval (Tofighi & Kelley, 2020): a likelihood-ratio interval for P_med and for the indirect effect $a*b$, obtained by inverting the constrained-likelihood test rather than by resampling. It is deterministic (no n_boot, no seed) and supports a Gaussian outcome and mediator, with covariates, and any treatment contrast $x_{ref} \neq x_{value}$. For binary or other non-Gaussian models, use the bootstrap methods.

For method = "mbco", the converged flag reflects the **P_med** interval only. The indirect-effect interval is reported separately and may be NA on a degenerate design (e.g. a non-finite delta-method scale for $a*b$) even when the P_med interval converges; check ie_ci_lower / ie_ci_upper directly.

Value

PmedResult object

Examples

```
# Toy example: Simple mediation model
# Generate data where X affects Y through M
set.seed(123)
n <- 100
data <- data.frame(
  X = rnorm(n),
  C = rnorm(n)
)
data$M <- 0.5 * data$X + 0.3 * data$C + rnorm(n, sd = 0.5)
data$Y <- 0.4 * data$M + 0.2 * data$X + 0.2 * data$C + rnorm(n, sd = 0.5)

# Compute P_med using plugin estimator (fast, no CI)
result_plugin <- pmed(
  Y ~ X + M + C,
  formula_m = M ~ X + C,
  data = data,
  treatment = "X",
  mediator = "M",
  method = "plugin"
)
print(result_plugin)

# With parametric bootstrap for confidence intervals
result_boot <- pmed(
  Y ~ X + M + C,
  formula_m = M ~ X + C,
  data = data,
  treatment = "X",
  mediator = "M",
  method = "parametric_bootstrap",
  n_boot = 200, # Use more (e.g., 1000+) in practice
  seed = 456
)
print(result_boot)
```

pmed-MediationData

*Compute P_med from MediationData***Description**

Compute P_med from MediationData

pmed-ParallelMediationData

Compute parallel (joint) P_med from a ParallelMediationData object

Description

Computes the joint P_med for k parallel mediators (all moved together from their control to their treated levels). Gaussian outcome and mediators only. See `vignette("parallel-mediation", package = "probmmed")` for the estimand.

pmed_sensitivity

M–Y Confounding Sensitivity for a Proportion-Mediated Estimand

Description

A shared, minimal sensitivity helper for the P_med program (Papers 1–3). All three estimands have the ratio form $P_{\text{med}} = \text{indirect} / \text{total}$. The single shared point of failure is assumption **A4** (no exposure-induced mediator–outcome confounding): it biases the cross-world corner mean that enters the *indirect* (numerator) component, leaving the *total* effect (denominator), identified from observable corner means, robust. This function propagates a stated numerator bias *b* to P_med, reporting $P_{\text{med}}(b) = (\text{indirect} + b) / \text{total}$ over a grid, the **tipping bias** that zeroes the indirect effect, and (optionally) the bias that pushes P_med to a user threshold. Gives each paper one honest sensitivity row.

Usage

```
pmed_sensitivity(
  object = NULL,
  indirect = NULL,
  total = NULL,
  bias_grid = NULL,
  threshold = NA_real_,
  eval = FALSE,
  se = NA_real_,
  scale = NA_character_,
  ...
)
```

Arguments

object	A P_med-family result (e.g. GaugePmedResult) or any object exposing indirect/total components. Ignored if both indirect and total are supplied directly.
indirect, total	Numeric: supply the components directly instead of (or to override) extraction from object.

bias_grid	Numeric: numerator biases b to evaluate. Default spans $\pm indirect $ on an 11-point grid.
threshold	Numeric or NA: a P_{med} value whose tipping bias to report (e.g. 0 to ask "how much bias makes the share vanish"). Default NA (skip).
evaluate	Logical: also compute an EValue E-value summary if the package is installed (default FALSE).
se	Numeric or NA: standard error of indirect, used only for the E-value CI bound when evaluate = TRUE.
scale	Character: label for the P_{med} scale (free text).
...	Unused.

Details

The bias model is an additive offset on the *mean-based* indirect component. This is exact for the difference-scale (Paper 1) and incremental (Paper 2) shares, whose numerators are differences of (cross-world) means. For variance-share estimands (Paper 3, Sobol $P_{med}^{\{\sigma^2\}}$) the numerator is a variance component, not a mean; the additive-offset model is **not** the right object there. Use this helper for mean-based shares; for Paper 3, supply a pre-computed numerator perturbation if you have one, and prefer the fuller Cinelli–Hazlett-style contour (planned) for a principled variance-component bias map.

The optional E-value summary (evaluate = TRUE) is a coarse RR-oriented robustness number; it is reported only as a familiar single figure and is guarded behind the **EValue** package (Suggests).

Value

A [PmedSensitivity](#) object.

Examples

```
## From components (any paper):
pmed_sensitivity(indirect = 0.30, total = 0.50, threshold = 0)
```

```
## From a Paper 1 fit:
set.seed(1)
n <- 400; C <- rnorm(n)
A <- rbinom(n, 1, plogis(0.5 * C))
M <- 0.5 * A + 0.3 * C + rnorm(n)
Y <- 0.4 * A + 0.6 * M + 0.2 * C + rnorm(n)
fit <- ward_residual(data.frame(A, M, Y, C))
pmed_sensitivity(fit, threshold = 0)
```

PmedResult

*P_med Result Class***Description**

S7 class for P_med computation results.

Usage

```
PmedResult(
  estimate = integer(0),
  ci_lower = integer(0),
  ci_upper = integer(0),
  ci_level = integer(0),
  method = character(0),
  n_boot = NA_integer_,
  boot_estimates = numeric(0),
  ie_estimate = NA_real_,
  ie_ci_lower = NA_real_,
  ie_ci_upper = NA_real_,
  ie_boot_estimates = numeric(0),
  x_ref = integer(0),
  x_value = integer(0),
  source_extract = NULL,
  converged = logical(0),
  call = NULL
)
```

Arguments

estimate	Numeric: P_med point estimate
ci_lower	Numeric: lower bound of confidence interval
ci_upper	Numeric: upper bound of confidence interval
ci_level	Numeric: confidence level (e.g., 0.95)
method	Character: inference method used
n_boot	Integer: number of bootstrap samples (NA if not bootstrap)
boot_estimates	Numeric vector: bootstrap distribution (empty if not bootstrap)
ie_estimate	Numeric: Indirect Effect (NIE) point estimate
ie_ci_lower	Numeric: lower bound of NIE confidence interval
ie_ci_upper	Numeric: upper bound of NIE confidence interval
ie_boot_estimates	Numeric vector: bootstrap distribution of NIE
x_ref	Numeric: reference treatment value

x_value	Numeric: treatment value for contrast
source_extract	medfit::MediationData object: source of the estimates
converged	Logical: did computation converge
call	Call object: original function call

PmedSensitivity	<i>P_med Sensitivity Result (M–Y Confounding)</i>
-----------------	---

Description

S7 class holding a one-row sensitivity analysis for a proportion-mediated estimand under violation of assumption **A4** (no exposure-induced mediator–outcome confounding). Every estimand in the P_med family shares the ratio form $P_{med} = \text{indirect} / \text{total}$. A4 failure biases the **cross-world** corner mean that enters the *indirect* (numerator) component, while the *total* (denominator) effect is identified from observable corner means and is robust to A4. The object therefore reports how P_med moves as the numerator bias b ranges over a stated set, plus the **tipping bias** that drives the indirect component (or P_med) through a threshold.

Usage

```
PmedSensitivity(
  p_med = integer(0),
  indirect = integer(0),
  total = integer(0),
  bias_grid = integer(0),
  p_med_bias = integer(0),
  tipping_indirect = integer(0),
  tipping_threshold = NA_real_,
  threshold = NA_real_,
  eval = numeric(0),
  scale = NA_character_,
  call = NULL
)
```

Arguments

p_med	Numeric: the point estimate indirect / total (b = 0).
indirect	Numeric: the indirect (mediated) component, the numerator.
total	Numeric: the total effect, the denominator.
bias_grid	Numeric: the grid of numerator biases b evaluated.
p_med_bias	Numeric: P_med at each b, i.e. (indirect + b)/total.
tipping_indirect	Numeric: bias b at which indirect + b = 0 (the indirect effect vanishes). NA if total == 0.

tipping_threshold	Numeric: bias b at which P_med hits threshold; NA if not requested.
threshold	Numeric: the P_med target used for tipping_threshold (NA if not requested).
evaluate	Numeric: optional E-value summary (length-2: estimate and CI bound) from the EValue package; numeric(0) if not computed.
scale	Character: label of the P_med scale (e.g. "Delta", "incremental"); free text supplied by the caller.
call	Call: the originating call.

See Also

[pmed_sensitivity\(\)](#)

print-PmedResult	<i>Print PmedResult</i>
------------------	-------------------------

Description

Print PmedResult

sobol_from_theta	<i>Map corner means to the Sobol variance decomposition</i>
------------------	---

Description

Closed-form map from the four corner means $\theta(a, a')$ to the Sobol / functional-ANOVA variance components and the variance-scale proportion mediated. Used internally by [sobol_pmed\(\)](#) and exported as a helper for computing the analytic (population) truth from known corner means.

Usage

sobol_from_theta(theta, pd = 0.5, pm = 0.5)

Arguments

theta	Numeric length-4: corner means named "11", "10", "01", "00" (i.e. $\theta(a, a')$ for (a, a') in the four corners).
pd, pm	Numeric: tilt probabilities for the direct and mediated pathways (Bernoulli design points); defaults 0.5.

Value

A named numeric vector with elements Vd, Vm, Vdm, VT, Pmed_sobol (= Vm / VT), S1_med (= Vm / VT), ST_med (= (Vm + Vdm) / VT), and Rint (the interaction remainder).

Examples

```
th <- c("11" = 1.32, "10" = 0.5, "01" = 0.42, "00" = 0)
sobol_from_theta(th)
```

sobol_pmed

Sobol / Variance-Scale Proportion Mediated

Description

Estimate the **Sobol proportion mediated** $P_{\text{med}}^{\{\sigma^2\}} = V_{\text{med}} / V_T$ – the share of the total interventional outcome variance carried by the mediated pathway, i.e. the first-order Sobol (functional-ANOVA) sensitivity index of the outcome with respect to the mediator. The estimator is cross-fitted and built on the same triply-robust corner pseudo-outcomes as [ward_residual\(\)](#) and [incr_pmed\(\)](#); the standard error uses the delta method through the quadratic variance components and the ratio identity. See [SobolPmedResult](#) for the variance decomposition and the boundary remedy.

Usage

```
sobol_pmed(
  object,
  pd = 0.5,
  pm = 0.5,
  covars = "C",
  K = 5L,
  seed = 1L,
  ci_level = 0.95,
  warn_boundary = TRUE,
  boundary_test = c("split", "plugin", "none"),
  procedure = c("B", "A"),
  reps = 1L,
  se_method = c("analytic", "bootstrap"),
  B = 200L,
  ...
)
```

Arguments

object	A data.frame with columns A (binary treatment), M (mediator), Y (continuous outcome), and the covariates named in covars.
pd, pm	Numeric: tilt probabilities for the direct and mediated pathways (Bernoulli design points); defaults 0.5.
covars	Character vector of covariate column names. Default "C".
K	Integer: number of cross-fitting folds (default 5).
seed	Integer: RNG seed for fold assignment and the boundary split (default 1).

ci_level	Numeric: confidence level (default 0.95).
warn_boundary	Logical: warn when the boundary test does not reject and the reported interval falls back to the Procedure-A upper bound (default TRUE).
boundary_test	Character: boundary test for $H_0: V_{\text{med}} = 0$. "split" (default) is the Williamson sample-split test on Δ_m ; "plugin" is the (over-rejecting) plug-in Wald test, diagnostic only.
procedure	Character: interval procedure for ci. "B" (default) is the sample-split CI (image of the Δ_m CI); "A" is the legacy gated interval (non-uniform near the boundary). See SobolPmedResult .
reps	Integer: repeated cross-fitting draws averaged for the point estimate (default 1). Use $\text{reps} > 1$ near the boundary for a reproducible, lower-variance estimate (it removes the ~80% fold-split variance).
se_method	Character: "analytic" (default) or "bootstrap". The bootstrap gives a valid (conservative) se near the non-regular boundary, where the analytic se is anti-conservative; off the boundary the analytic se is calibrated.
B	Integer: bootstrap resamples when $\text{se_method} = \text{"bootstrap"}$ (default 200).
...	Unused.

Details

At the null $V_{\text{med}} = 0$ (equivalently $\Delta_m = 0$) the squared-variance influence function degenerates and the symmetric Wald interval is non-regular. A Williamson et al. (2021) sample-split test on the regular contrast Δ_m decides the boundary. The default reported interval is **Procedure B** ($\text{procedure} = \text{"B"}$): the image of the regular Δ_m Wald CI under $\delta \rightarrow (c_m/V_T) \delta^2$ (continuous mapping, no gate). Its validity inherits that of the input Δ_m CI – so near the boundary use $\text{reps} > 1$ and $\text{se_method} = \text{"bootstrap"}$ for a valid interval (the analytic Δ_m se is anti-conservative there; see the boundary coverage section). **Procedure A** ($\text{procedure} = \text{"A"}$) is the legacy gated rule (Wald on rejection, one-sided $[0, P_{\text{med_upper}}]$ otherwise), pointwise- but **not** uniformly-valid across the near-null transition (Leeb-Potscher pre-test under-coverage). The boundary machinery recurses through a plain internal fitter (not this generic), so the point estimate, fold draw, and split are deterministic in seed. See the boundary coverage section in [SobolPmedResult](#).

Value

A [SobolPmedResult](#) object.

References

Williamson, B. D., Gilbert, P. B., Carone, M., & Simon, N. (2021). Nonparametric variable importance assessment using machine learning techniques. *Biometrics*, 77(1), 9–22.

Examples

```
set.seed(1)
n <- 1500; C <- rnorm(n)
A <- rbinom(n, 1, plogis(-0.2 + 0.8 * C))
M <- 0.6 * A + 0.4 * C + rnorm(n)
```

```
# strong mediation (off-boundary): Wald interval reported
Y <- 0.5 * A + 0.7 * M + 0.3 * C + rnorm(n)
sobol_pmed(data.frame(A, M, Y, C))
```

SobolPmedResult

Sobol / Variance-Scale Proportion Mediated Result

Description

S7 class for the **Sobol (functional-ANOVA) proportion mediated** $P_{\text{med}}^{\{\sigma^2\}} = V_{\text{med}} / V_T$ (Paper 3 of the P_{med} -for-modern-estimands program). On the variance scale the proportion mediated is the first-order Sobol sensitivity index of the outcome with respect to the mediated pathway: the share of the total interventional outcome variance V_T explained by the mediated component V_{med} . Unlike the additive split $TE = NDE + NIE$, the variance decomposition $V_T = V_{\text{dir}} + V_{\text{med}} + V_{\text{int}}$ always holds (the interaction variance V_{int} is a separate non-negative term), so $P_{\text{med}}^{\{\sigma^2\}}$ is well defined even under treatment-by-mediator interaction.

Usage

```
SobolPmedResult(
  p_med = integer(0),
  se = integer(0),
  ci = integer(0),
  ci_wald = integer(0),
  ci_A = NA_real_,
  ci_B1 = NA_real_,
  ci_B2 = NA_real_,
  procedure = "B",
  boundary = logical(0),
  Pmed_upper = integer(0),
  S1_med = integer(0),
  ST_med = integer(0),
  Vd = integer(0),
  Vm = integer(0),
  Vdm = integer(0),
  VT = integer(0),
  Dm = integer(0),
  se_Dm = integer(0),
  vmed_split_p = NA_real_,
  vmed_split_reject = NA_integer_,
  theta = integer(0),
  method = character(0),
  se_method = "analytic",
  reps = 1L,
  n = integer(0),
  ci_level = integer(0),
```

```

    call = NULL
)

```

Arguments

p_med	Numeric: Sobol proportion mediated $V_{\text{med}} / V_{\text{T}}$.
se	Numeric: standard error of p_med (delta-method, ratio identity).
ci	Numeric length-2: reported confidence interval, selected by procedure – Procedure B (default, uniformly valid) or Procedure A (gated).
ci_wald	Numeric length-2: ungated symmetric Wald interval (always the Wald interval, even at the boundary).
ci_A	Numeric length-2: Procedure-A (gated, pre-test) interval – the Wald interval off-boundary, or $[0, P_{\text{med_upper}}]$ at the boundary. Pointwise- but not uniformly-valid (see Coverage caveat).
ci_B1	Numeric length-2: Procedure-B interval from the same-sample Delta_m Wald CI mapped through delta $\rightarrow (c_m/V_T) \text{ delta}^2$ (efficient).
ci_B2	Numeric length-2: experimental Procedure-B variant from a single decorrelated split (Delta_m point on one half, se on the independent half) mapped through the same square. Intended to break Delta_m/se coupling, but the half-sample centre makes it high-variance and it did not improve near-null coverage in simulation – a K-fold-decorrelated se keeping the full-sample point is the open fix. Not recommended; use ci (B1).
procedure	Character: "B" (default, uniform sample-split CI) or "A" (legacy gated). Controls which interval is returned in ci.
boundary	Logical: TRUE when the split test does not reject $H_0: V_{\text{med}} = 0$ (the symmetric Wald CI is non-regular there).
Pmed_upper	Numeric: one-sided Procedure-A upper bound for p_med.
S1_med	Numeric: first-order Sobol index $V_{\text{med}} / V_{\text{T}}$ (equals p_med).
ST_med	Numeric: total Sobol index $(V_{\text{med}} + V_{\text{int}}) / V_{\text{T}}$.
Vd, Vm, Vdm, VT	Numeric: direct, mediated, interaction, and total variance components.
Dm	Numeric: mediated antecedent contrast Delta_m.
se_Dm	Numeric: standard error of Delta_m.
vmed_split_p	Numeric: p-value of the sample-split boundary test of $H_0: V_{\text{med}} = 0$.
vmed_split_reject	Integer: 1L if the boundary test rejects, 0L otherwise (NA when the test was not run).
theta	Numeric length-4: corner means theta(a, a').
method	Character: estimation method.
se_method	Character: "analytic" (default, influence-function se) or "bootstrap" (non-parametric resample-and-refit se; valid but conservative near the boundary). See the boundary coverage section.
reps	Integer: number of repeated cross-fitting fold draws averaged for the point estimate (default 1); reps > 1 removes the near-boundary fold-split Monte-Carlo variance (mean DML aggregation).

n	Integer: sample size.
ci_level	Numeric: confidence level.
call	Call: original call.

Details

Writing the four corner means as $\theta(a, a') = E[Y(a, M(a'))]$, the contrasts are the direct antecedent Δ_d , the mediated antecedent Δ_m , and the interaction remainder $R = \theta_{11} - \theta_{10} - \theta_{01} + \theta_{00}$. With Bernoulli tilt variances $c_d = p_d(1 - p_d)$ and $c_m = p_m(1 - p_m)$, $V_{dir} = c_d \Delta_d^2$, $V_{med} = c_m \Delta_m^2$, $V_{int} = c_d c_m R^2$, and $V_T = V_{dir} + V_{med} + V_{int}$. The estimator is the cross-fitted one-step estimator built on the shared triply-robust corner pseudo-outcomes (the same engine as `ward_residual()` and `incr_pmed()`); the delta method propagates the corner influence functions through the quadratic V-components and the ratio identity to a standard error for $P_{med}^{\sigma^2}$.

Boundary remedy. Because $V_{med} = c_m \Delta_m^2$ is a squared quantity, its influence function $\phi_{\{V_{med}\}} = 2 c_m \Delta_m \phi_{\{\Delta_m\}}$ degenerates as $\Delta_m \rightarrow 0$, deflating the Wald standard error and collapsing coverage at the null $V_{med} = 0$. The null is equivalent to $\Delta_m = 0$, which has a regular root-n influence function. Following Williamson et al. (2021, *Biometrics*), the boundary is tested with a sample-split statistic that estimates Δ_m on one half and its standard error on the independent other half (so $z \sim N(0, 1)$ even at the boundary).

Two interval procedures. The default reported ci is **Procedure B**: the image of the regular two-sided Δ_m Wald CI under the map $\delta \rightarrow (c_m/V_T) \delta^2 (ci_{B1}/ci_{B2})$. Because it is the continuous image of a uniformly-valid CI for the *regular* parameter Δ_m , it is uniformly valid in Δ_m – there is no pre-test, hence no boundary pathology (subject to the near-null se caveat below). The reported ci is ci_{B1} , the same-sample image (centered at the full Δ_m , so it always brackets p_{med}); ci_{B2} is an *experimental* single-split variant that did not improve coverage (see its note). **Procedure A** (procedure = "A", ci_A) is the legacy *gated* rule: Wald on rejection, one-sided $[0, P_{med_upper}]$ otherwise.

Coverage near the boundary – mechanism and remedies (A-15). Two distinct issues; the second is now decomposed (reproducible fixed-seed variance decomposition) into three findings, each with a concrete remedy.

(1) *Procedure A is not uniformly valid.* As a pre-test (gating) rule A is only pointwise valid: near the boundary it routes downward-selected Δ_m to the contracting one-sided bound, a Leeb-Potscher post-selection effect. Prefer B, which does not gate.

(2) *The near-null Δ_m standard error.* Three pinned facts:

- **Fold-split Monte-Carlo variance dominates.** A single cross-fitting fold partition contributes ~80% of $\text{Var}(\Delta_m\text{-hat})$ in the near-null regime (~45% at ordinary effect sizes), persistent across $n = 2000$ to 8000 . This is an algorithmic nuisance, not sampling information: it inflates the estimator's variance and makes the single-split point estimate seed-dependent. *Remedy:* $\text{reps} > 1$ (repeated cross-fitting – the corner influence matrix is averaged over reps independent fold draws), which removes it and yields a reproducible, lower-variance point estimate (mean DML aggregation).
- **Δ_m is approximately normal at the null.** Recomputing coverage with the oracle Monte-Carlo SD gives ~0.95, so the Wald *shape* is correct for the regular contrast Δ_m : the under-coverage is a wrong interval *width*, not a wrong *shape* (there is no boundary non-normality for Δ_m itself).

- **A residual analytic-se bias remains.** Even after fold noise is removed, the analytic influence-function se for `Delta_m` is ~0.8x its true sampling SD near the null (structural; persistent in `n`); it is well calibrated (ratio ~1) at ordinary effect sizes. So the *default* analytic near-null interval – and its Procedure-B image – covers only ~0.85. *Remedy:* `se_method = "bootstrap"` (resample rows, refit the cross-fit estimator), which recovers a valid but mildly **conservative** se (~1.25x the true SD, coverage ~0.97 near null) and is calibrated off-boundary.

Practical guidance. Off the boundary the analytic default is calibrated and fast. Near the boundary (the split test does not reject), pass `reps > 1` for a reproducible point and `se_method = "bootstrap"` for a valid interval (conservative there – it trades width for guaranteed coverage). `V_T` is well calibrated throughout. The full coverage grid across the `Delta_m` transition is validated by a separate large simulation. Use `procedure = "A"` only to reproduce the legacy gated behaviour.

References

Williamson, B. D., Gilbert, P. B., Carone, M., & Simon, N. (2021). Nonparametric variable importance assessment using machine learning techniques. *Biometrics*, 77(1), 9–22.

summary-PmedResult	Summary Method
--------------------	----------------

Description

Summary Method

ward_residual	Gauge-Calibrated Proportion Mediated
---------------	--------------------------------------

Description

Estimate the interventional proportion mediated together with the gauge residual $W = R / OE$ that flags non-decomposability (treatment-by-mediator interaction). Uses a cross-fitted one-step estimator built on the triply-robust efficient influence functions of the four corner means $\theta(a, a') = E[Y(a, M(a'))]$.

Usage

```
ward_residual(  
  object,  
  covars = "C",  
  K = 5L,  
  ci_level = 0.95,  
  seed = 1L,  
  fieller = TRUE,  
  reps = 1L,  
  se_method = c("analytic", "bootstrap"),  
  B = 200L,
```

```

    a0 = 0,
    a1 = 1,
    ...
  )

```

Arguments

object	A data.frame with columns A (binary treatment), M (mediator), Y (outcome), and the covariates named in covars.
covars	Character vector of covariate column names. Default "C".
K	Integer: number of cross-fitting folds (default 5).
ci_level	Numeric: confidence level (default 0.95).
seed	Integer: RNG seed for fold assignment.
fieller	Logical: also compute the Fieller confidence set for the ratio $P_{med} = IIE/OE$, which is unbounded when the total effect OE is not significant (default TRUE).
reps	Integer: number of repeated cross-fitting fold draws averaged for the point estimate (default 1). $reps > 1$ averages the corner influence matrix over independent fold assignments, removing the fold-split component of the variance; the analytic se then adds the residual fold Monte-Carlo variance of the averaged point.
se_method	Character: "analytic" (default, influence-function se, symmetric Wald CI) or "bootstrap". The analytic se for the ratios W and P_{med} is right-skewed with a median below the empirical SD, so the symmetric Wald CI is mildly anti-conservative (sub-nominal coverage ≈ 0.85 -0.90). Under "bootstrap" the CIs for W and P_{med} are the tail-aware percentile intervals of the nonparametric bootstrap (resample rows, refit) – the appropriate construction for a skewed ratio (widening a symmetric se fails to cover it); W_se/ p_{med} se are still reported as bootstrap dispersion summaries. $reps > 1$ and <code>se_method = "bootstrap"</code> compose. Bootstrap validity follows from the estimator being a Neyman-orthogonal cross-fit (DML-type) functional (Lin et al. 2026); it requires the ratio to be regular, i.e. OE bounded away from 0 – see the Fieller diagnostic for the near-null case.
B	Integer: number of bootstrap resamples when <code>se_method = "bootstrap"</code> (default 200). Cost is B (x reps) refits.
a0, a1	Reference and comparison exposure levels (defaults 0/1). A may use any two-level coding (factor, {1, 2}, {-1, 1}); it is recoded to the binary indicator ($A == a1$). The gauge $W = R/OE$ is invariant to swapping $a0/a1$ (both R and OE flip sign). An A with more than two levels is an error, not a silent subset: restricting to { $a0, a1$ } would shift the covariate-averaging population and estimate a different (sub-population) gauge. Filter to the two intended levels first. Multi-valued / continuous exposures are future work.
...	Unused.

Value

A [GaugePmedResult](#) object.

Examples

```
set.seed(1)
n <- 800; C <- rnorm(n)
A <- rbinom(n, 1, plogis(-0.2 + 0.8 * C))
M <- 0.6 * A + 0.4 * C + rnorm(n)
Y <- 0.5 * A + 0.7 * M + 0.8 * A * M + 0.3 * C + rnorm(n)
ward_residual(data.frame(A, M, Y, C))
```

Index

`extract_mediation`, [2](#), [2](#), [7](#)

`GaugePmedResult`, [3](#), [10](#), [21](#)

`incr_pmed`, [4](#)

`incr_pmed()`, [15](#), [19](#)

`IncrPmedResult`, [5](#), [5](#)

`plot-PmedResult`, [6](#)

`pmed`, [6](#)

`pmed-formula`, [8](#)

`pmed-MediationData`, [9](#)

`pmed-ParallelMediationData`, [10](#)

`pmed_sensitivity`, [10](#)

`pmed_sensitivity()`, [14](#)

`PmedResult`, [12](#)

`PmedSensitivity`, [11](#), [13](#)

`print-PmedResult`, [14](#)

`sobol_from_theta`, [14](#)

`sobol_pmed`, [15](#)

`sobol_pmed()`, [14](#)

`SobolPmedResult`, [15](#), [16](#), [17](#)

`summary-PmedResult`, [20](#)

`ward_residual`, [20](#)

`ward_residual()`, [4](#), [15](#), [19](#)