

# Package: probmed (via r-universe)

June 12, 2026

**Title** Probabilistic Effect Sizes for Causal Mediation Analysis

**Version** 0.2.0

**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

**Roxygen** list(markdown = TRUE)

**Suggests** alabama, knitr, lavaan, MASS, mediation, rmarkdown, testthat  
( $\geq 3.0.0$ ), quarto

**Config/testthat/edition** 3

**Imports** medfit ( $\geq 0.3.0$ ), methods, S7 ( $\geq 0.2.1$ ), stats

**Remotes** data-wise/medfit@v0.3.0

**VignetteBuilder** knitr, quarto

**RoxygenNote** 7.3.3

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

**Date/Publication** 2026-06-12 09:58:21 UTC

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

**RemoteRef** dev

**RemoteSha** 7f7ab7667b5bc33fecfbe035c61d23f83fb4cea3

## Contents

extract_mediation . . . . .	2
plot-PmedResult . . . . .	2
pmed . . . . .	3
pmed-formula . . . . .	4

pmed-MediationData . . . . .	6
pmed-ParallelMediationData . . . . .	6
PmedResult . . . . .	6
print-PmedResult . . . . .	8
summary-PmedResult . . . . .	8

<b>Index</b>	<b>9</b>
--------------	----------

---

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

---

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

---

### Description

Plot Bootstrap Distribution

pmed

*Compute P\_med: Probabilistic Effect Size for Mediation***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:
	<ul style="list-style-type: none"> <li>• A formula for the outcome model (most common)</li> <li>• A <code>medfit::MediationData</code> object from <code>extract_mediation()</code></li> </ul>
...	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  $P_{\text{med}}$  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 != 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  $P_{\text{med}}$  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 = "probedmed") for the estimand.

---

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

---

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

---

**Description**

Print PmedResult

---

summary-PmedResult	<i>Summary Method</i>
--------------------	-----------------------

---

**Description**

Summary Method

# Index

`extract_mediation`, [2](#), [2](#), [3](#)

`plot-PmedResult`, [2](#)

`pmed`, [3](#)

`pmed-formula`, [4](#)

`pmed-MediationData`, [6](#)

`pmed-ParallelMediationData`, [6](#)

`PmedResult`, [6](#)

`print-PmedResult`, [8](#)

`summary-PmedResult`, [8](#)