

Package ‘rmedsem’

September 16, 2026

Title Statistical Mediation Analysis for SEMs

Version 1.1.0

Description Conducts mediation analysis for structural equation models (SEM) estimated with 'lavaan', 'blavaan', 'cSEM', or 'modsem'. Implements the Baron and Kenny (1986) <doi:10.1037/0022-3514.51.6.1173> and Zhao, Lynch & Chen (2010) <doi:10.1086/651257> approaches to determine the presence and type of mediation. Supports covariance-based SEM, partial least squares SEM, Bayesian SEM, and moderated mediation and mediated moderation models. Tests indirect effects with the Sobel, Delta, Monte-Carlo, and bootstrap methods or, for Bayesian models, with posterior summaries and equal-tailed or highest density credible intervals. Reports the effect size measures RIT, RID, and Upsilon of Lachowicz, Preacher and Kelley (2018) <doi:10.1037/met0000165>. Results can be summarized, extracted with standard methods such as summary(), coef() and confint(), and plotted.

License MIT + file LICENSE

Encoding UTF-8

Imports lavaan, mvtnorm, ggplot2, stats

URL <https://github.com/ihrke/rmedsem>, <https://ihrke.github.io/rmedsem/>

BugReports <https://github.com/ihrke/rmedsem/issues>

Depends R (>= 4.1.0)

LazyData true

Suggests blavaan, cSEM, HDInterval, modsem, testthat (>= 3.0.0)

Config/Needs/website semPlot, rmarkdown

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Mehmet Mehmetoglu [aut] (ORCID:

<<https://orcid.org/0000-0002-6092-8551>>),

Matthias Mittner [aut, cre] (ORCID:

<<https://orcid.org/0000-0003-0205-7353>>),

Kjell Slupphaug [aut] (ORCID: <<https://orcid.org/0009-0005-8324-2834>>)

Maintainer Matthias Mittner <matthias.mittner@uit.no>
Repository CRAN
Date/Publication 2026-09-16 07:30:02 UTC

Contents

effect-sizes	2
hsbdemo	4
mchoice	5
plot.rmedsem	6
rmedsem	7
rmedsem-methods	12
workout	15
Index	17

effect-sizes	<i>Effect Sizes for Mediation Analysis</i>
--------------	--

Description

Extract the effect sizes computed by `rmedsem()` (argument `effect.size`).

Usage

```
RIT(res, ...)  
  
RID(res, ...)  
  
## Default S3 method:  
RIT(res, ...)  
  
## S3 method for class 'rmedsem'  
RIT(res, ...)  
  
## Default S3 method:  
RID(res, ...)  
  
## S3 method for class 'rmedsem'  
RID(res, ...)  
  
Upsilon(res, ...)  
  
## Default S3 method:  
Upsilon(res, ...)  
  
## S3 method for class 'rmedsem'  
Upsilon(res, adjusted = TRUE, ...)
```

Arguments

<code>res</code>	an <code>rmedsem</code> object
<code>...</code>	additional arguments (currently unused)
<code>adjusted</code>	logical; if TRUE (default), return the bias-adjusted estimator of Upsilon; if FALSE, the unadjusted estimator

Details

`RIT()` Ratio of the indirect to the total effect, $\text{lindirectl} / \text{ltotall}$, i.e., the proportion of the total effect that is mediated. Following Kenny (see <https://davidakenny.net/cm/mediate.htm>), it should only be interpreted if the total effect is not too small ($|\text{ltotall}| \geq 0.2$ for standardized coefficients).

`RID()` Ratio of the indirect to the direct effect, $\text{lindirectl} / \text{ldirectl}$.

`Upsilon()` The Upsilon effect size (Lachowicz, Preacher & Kelley, 2018), an R-squared-type measure of the variance in Y explained indirectly by X through M, computed from standardized coefficients.

`RIT()` and `RID()` give a warning (and the printed output of `rmedsem()` does not report them) if they should not be interpreted: `RIT` if the total effect is small ($|\text{ltotall}| < 0.2$), `RID` if the direct effect is not significant (p-value not below `p.threshold`), as the ratio is then unstable. Both also warn if the indirect effect is larger than the total effect.

Value

A numeric scalar.

References

Lachowicz, M. J., Preacher, K. J., & Kelley, K. (2018). A novel measure of effect size for mediation analysis. *Psychological Methods*, 23(2), 244–261. doi:10.1037/met0000165

Examples

```
mod.txt <- "
read ~ math
science ~ read + math
"

mod <- lavaan::sem(mod.txt, data = rmedsem::hsbdemo)
out <- rmedsem(mod, indep = "math", med = "read", dep = "science")
RIT(out)
RID(out)
Upsilon(out)
Upsilon(out, adjusted = FALSE)
```

hsbdemo

*High School and Beyond Demo Dataset***Description**

Demographic information and standardized test scores of 200 students from the High School and Beyond survey, as distributed by the UCLA Statistical Methods and Data Analytics group.

Usage

```
hsbdemo
```

Format

```
hsbdemo:
```

```
A data frame with 200 rows and 13 columns:
```

```
id Student ID
```

```
female Gender, "female" or "male"
```

```
ses Socio-economic status, "low", "middle" or "high"
```

```
schtyp School type, "public" or "private"
```

```
prog Type of program, "general", "academic" or "vocation"
```

```
read Reading score
```

```
write Writing score
```

```
math Math score
```

```
science Science score
```

```
socst Social studies score
```

```
honors Enrollment in honors program, "enrolled" or "not enrolled"
```

```
awards Number of awards
```

```
cid Class ID
```

Source

UCLA Statistical Methods and Data Analytics, <https://stats.oarc.ucla.edu/stat/data/hsbdemo.dta>

Examples

```
str(hsbdemo)
```

```
mod <- lavaan::sem("read ~ math\nscience ~ read + math", data = hsbdemo)
rmedsem(mod, indep = "math", med = "read", dep = "science")
```

mchoice

*Mate-Choice Survey Data from Trondheim***Description**

Data from a survey on mate-choice in Trondheim, collected in 2021 using a convenience sampling procedure. Participants rated how well physical features (`smv_attr_face`, `smv_attr_body`, `smv_sexy`) and personality features (`smv_kind` to `smv_sociable`) described them as romantic partners, answered items on self-esteem (`ses_*`) and reported how often they had experienced feelings related to mental well-being in the past two weeks (`mwb_*`). All items were measured on a scale from 1 to 5.

Usage

mchoice

Format

mchoice:

A data frame with 1067 rows and 15 columns:

smv_attr_face How well does this describe you as a partner? (attractive face), [1] very bad - [5] very well

smv_attr_body How well does this describe you as a partner? (attractive body), [1] very bad - [5] very well

smv_sexy How well does this describe you as a partner? (sexy), [1] very bad - [5] very well

ses_satis On the whole, I am satisfied with myself, [1] totally disagree - [5] totally agree

ses_qualities I feel that I have a number of good qualities, [1] totally disagree - [5] totally agree

ses_able_todo I am able to do things as well as most other people, [1] totally disagree - [5] totally agree

mwb_optimistic I have been feeling optimistic about the future, [1] never - [5] always

mwb_useful I have been feeling useful, [1] never - [5] always

mwb_energy I have had energy to spare, [1] never - [5] always

smv_kind How well does this describe you as a partner? (kind), [1] very bad - [5] very well

smv_caring How well does this describe you as a partner? (caring), [1] very bad - [5] very well

smv_understanding How well does this describe you as a partner? (understanding), [1] very bad - [5] very well

smv_make_laughh How well does this describe you as a partner? (make people laugh), [1] very bad - [5] very well

smv_funny How well does this describe you as a partner? (funny), [1] very bad - [5] very well

smv_sociable How well does this describe you as a partner? (sociable), [1] very bad - [5] very well

Source

Survey collected in Norway in 2021 using a convenience sampling procedure.

Examples

```
str(mchoice)

mod.txt <- "
  OwnLook  =~ smv_attr_face + smv_attr_body + smv_sexy
  SelfEst  =~ ses_satis + ses_qualities + ses_able_todo
  MentWell =~ mwb_optimistic + mwb_useful + mwb_energy
  SelfEst  ~ OwnLook
  MentWell ~ OwnLook + SelfEst
"

mod <- lavaan::sem(mod.txt, data = mchoice)
rmedsem(mod, indep = "OwnLook", med = "SelfEst", dep = "MentWell")
```

plot.rmedsem	<i>Plot rmedsem Results</i>
--------------	-----------------------------

Description

Visualize the results of `rmedsem()`.

Usage

```
## S3 method for class 'rmedsem'
plot(x, type = c("coef", "effect"), ...)

plot_effect(res, description = TRUE)

plot_coef(res)
```

Arguments

<code>x, res</code>	an <code>rmedsem</code> object
<code>type</code>	character; "coef" (default) for a coefficient plot or "effect" for an effect size plot
<code>...</code>	additional arguments passed to <code>plot_coef()</code> or <code>plot_effect()</code>
<code>description</code>	logical, whether to add a caption describing the proportion of the total effect that is mediated (default TRUE)

Details

`plot_coef()` A coefficient plot of the indirect effect (for each estimation method), the direct effect and the total effect, with their confidence (or credible) intervals.

`plot_effect()` A pie chart of the (absolute) indirect and direct effects, i.e., the proportion of the total effect that is mediated. Requires the effect sizes "RIT" and "RID".

`plot()` calls `plot_coef()` (`type = "coef"`) or `plot_effect()` (`type = "effect"`).

Value

a ggplot object

See Also

[rmedsem\(\)](#), [rmedsem-methods](#)

Examples

```
mod.txt <- "
read ~ math
science ~ read + math
"

mod <- lavaan::sem(mod.txt, data = rmedsem::hsbdemo)
out <- rmedsem(mod, indep = "math", med = "read", dep = "science")
plot(out)
plot(out, type = "effect")
plot_effect(out, description = FALSE)
```

rmedsem

Mediation Analysis for Structural Equation Models

Description

Tests the indirect effect of an independent variable X on a dependent variable Y through a mediator M in a fitted structural equation model (SEM), and determines the type of mediation using the Baron and Kenny (1986) and/or Zhao, Lynch & Chen (2010) approaches. Models estimated with **lavaan** (covariance-based SEM), **cSEM** (PLS-SEM), **blavaan** (Bayesian SEM) and **modsem** (models with latent interactions) are supported. The model must contain the regression paths X -> M, M -> Y and X -> Y.

Usage

```
rmedsem(
  mod,
  indep,
  med,
  dep,
  approach = c("bk", "zlc"),
  p.threshold = 0.05,
  effect.size = c("RIT", "RID", "upsilon"),
  ...
)

## Default S3 method:
rmedsem(mod, indep, med, dep, ...)
```

```
## S3 method for class 'blavaan'
rmedsem(
  mod,
  indep,
  med,
  dep,
  approach = c("bk", "zlc"),
  p.threshold = 0.05,
  effect.size = c("RIT", "RID", "upsilon"),
  ci.two.tailed = 0.95,
  hdi = FALSE,
  ...
)
```

```
## S3 method for class 'cSEMResults'
rmedsem(
  mod,
  indep,
  med,
  dep,
  approach = c("bk", "zlc"),
  p.threshold = 0.05,
  effect.size = c("RIT", "RID", "upsilon"),
  nbootstrap = 1000,
  ci.two.tailed = 0.95,
  seed = NULL,
  ...
)
```

```
## S3 method for class 'lavaan'
rmedsem(
  mod,
  indep,
  med,
  dep,
  approach = c("bk", "zlc"),
  p.threshold = 0.05,
  effect.size = c("RIT", "RID", "upsilon"),
  standardized = TRUE,
  mcreps = 5000,
  ci.two.tailed = 0.95,
  ...
)
```

```
## S3 method for class 'modsem'
rmedsem(
  mod,
  indep,
```

```

    med,
    dep,
    approach = c("bk", "zlc"),
    p.threshold = 0.05,
    effect.size = c("RIT", "RID", "upsilon"),
    moderator = NULL,
    standardized = TRUE,
    mcreps = 5000,
    ci.two.tailed = 0.95,
    ...
)

```

Arguments

mod	a fitted SEM: an object of class lavaan, cSEMResults, blavaan or modsem. blavaan models containing latent variables must be fitted with save.lvs = TRUE.
indep	a string, the name of the independent variable (X). For modsem models, this can be an interaction term such as "W:X".
med	a string, the name of the mediator (M)
dep	a string, the name of the dependent variable (Y)
approach	approach(es) to determine the type of mediation: "bk" (Baron and Kenny), "zlc" (Zhao, Lynch & Chen), or both (default). Ignored for blavaan models.
p.threshold	a number between 0 and 1, the p-value threshold for significance (default 0.05). A p-value equal to the threshold counts as not significant.
effect.size	character vector with the effect sizes to compute; one or more of "RIT" (ratio of indirect to total effect), "RID" (ratio of indirect to direct effect) and "upsilon" (Lachowicz et al., 2018); see effect-sizes .
...	additional arguments passed to methods (currently unused)
ci.two.tailed	a number between 0 and 1, the level of all confidence (or, for blavaan, credible) intervals (default 0.95)
hdi	(blavaan) a logical. If FALSE (default), equal-tailed credible intervals are computed; if TRUE, highest density intervals (requires the HDInterval package). Applies to the indirect, direct and total effects and to Upsilon.
nbootstrap	(cSEM) the number of bootstrap samples (default 1000)
seed	(cSEM) NULL (default) or a non-negative integer, the seed for the bootstrap. If NULL, the seed is drawn from R's random number generator, so that results can be reproduced with set.seed() .
standardized	(lavaan, modsem) a logical, whether to use standardized coefficients (default TRUE). cSEM and blavaan results are always standardized.
mcreps	(lavaan, modsem) the number of Monte-Carlo samples, a positive integer (default 5000)
moderator	(modsem) NULL (default) or a string, the name of the moderator W for moderated mediation. The model must contain an interaction of the moderator with indep and/or med.

Value

an object of class `c("rmedsem_<pkg>", "rmedsem")`, where `<pkg>` identifies the backend (lavaan, cSEM, blavaan or modsem). See [rmedsem-methods](#) for functions to print, summarize and extract results, [effect-sizes](#) for effect sizes and `plot.rmedsem()` for plots. The structure of the object is described in section 'Adding a backend'.

Backends

lavaan The indirect effect is tested with the Sobel, Delta and Monte-Carlo methods. The Zhao, Lynch & Chen approach is based on the Monte-Carlo test.

cSEMResults The model is re-estimated with `nbootstrap` bootstrap samples. The indirect effect is tested with the Sobel, Delta and bootstrap methods, and the Zhao, Lynch & Chen approach is based on the bootstrap test. Only single-group, first-order models are supported.

blavaan Estimates are based on the (standardized) posterior samples. The output reports posterior means, standard deviations, posterior probabilities of a positive and negative indirect effect, evidence ratios and credible intervals; the Baron and Kenny and Zhao, Lynch & Chen approaches are not applied.

modsem As for lavaan. In addition, moderated mediation (via `moderator`) and mediated moderation (an interaction term as `indep`) are supported.

Multi-group and multilevel models are not supported.

Adding a backend

Support for further model classes is added by writing a method `rmedsem.<class>()` that returns a list of class `c("rmedsem_<pkg>", "rmedsem")`. For the default `print.rmedsem()`, `plot.rmedsem()` and `as.data.frame.rmedsem()` methods to work, the list must contain the following elements:

package name of the estimating package (character).

standardized whether the coefficients are standardized (logical).

vars list with elements `indep`, `med` and `dep`.

est.methods character vector naming the estimation methods for the indirect effect, e.g. `c("sobel", "delta", "montc")`.

one element per entry in est.methods a named numeric vector with elements `coef`, `se`, `zval`, `pval`, `lower` and `upper`. The Baron and Kenny approach requires the element `sobel`.

zlc.method (optional) the entry of `est.methods` whose p-value is used for the Zhao, Lynch & Chen approach; defaults to the last entry of `est.methods`.

direct.effect named numeric vector with elements `coef`, `se`, `pval`, `lower` and `upper`.

total.effect named numeric vector with elements `coef`, `se`, `lower` and `upper`.

med.approach character vector, a subset of `c("bk", "zlc")`.

med.data list with elements `sig_thresh` (the p-value threshold), `coefs` and `pvals`; the latter two are lists with elements `moi` ($X \rightarrow M$), `dom` ($M \rightarrow Y$) and `doi` ($X \rightarrow Y$).

effect.size list with (a subset of) elements `RIT`, `RID` and `upsilon`, as returned by the built-in backends.

nobs (optional) number of observations, used by `stats::nobs()`.

`ci.level` (optional) level of the stored intervals, used by `stats::confint()` and `summary()`; defaults to 0.95.

`ci.type` (optional) label of the stored intervals, "CI" (default) or "HDI".

A backend whose output does not fit this scheme can provide its own `print.rmedsem_<pkg>()` method, either replacing the default output (as for `lavaan`) or extending it via `NextMethod()` (as for `modsem`).

References

- Baron, R. M., & Kenny, D. A. (1986). The moderator-mediator variable distinction in social psychological research: Conceptual, strategic, and statistical considerations. *Journal of Personality and Social Psychology*, 51(6), 1173–1182. doi:[10.1037/00223514.51.6.1173](https://doi.org/10.1037/00223514.51.6.1173)
- Lachowicz, M. J., Preacher, K. J., & Kelley, K. (2018). A novel measure of effect size for mediation analysis. *Psychological Methods*, 23(2), 244–261. doi:[10.1037/met0000165](https://doi.org/10.1037/met0000165)
- Zhao, X., Lynch, J. G., & Chen, Q. (2010). Reconsidering Baron and Kenny: Myths and truths about mediation analysis. *Journal of Consumer Research*, 37(2), 197–206. doi:[10.1086/651257](https://doi.org/10.1086/651257)

Examples

```
## lavaan: observed variables
mod.txt <- "
read ~ math
science ~ read + math
"

mod <- lavaan::sem(mod.txt, data = rmedsem::hsbdemo)
out <- rmedsem(mod, indep = "math", med = "read", dep = "science")
out

# Zhao, Lynch & Chen approach only, unstandardized coefficients
rmedsem(mod, indep = "math", med = "read", dep = "science",
        approach = "zlc", standardized = FALSE, mcreps = 5000)

## cSEM
if (requireNamespace("cSEM", quietly = TRUE)) {
  model <- "
    OwnLook =~ smv_attr_face + smv_attr_body + smv_sexy
    SelfEst  =~ ses_satis + ses_qualities + ses_able_todo
    MentWell =~ mwb_optimistic + mwb_useful + mwb_energy
    SelfEst  ~ OwnLook
    MentWell ~ OwnLook + SelfEst
  "

  mod <- cSEM::csem(rmedsem::mchoice, model)
  # small number of bootstrap samples to keep the example fast
  rmedsem(mod, indep = "OwnLook", med = "SelfEst", dep = "MentWell",
          nbootstrap = 200)
}

## modsem: mediated moderation and moderated mediation
if (requireNamespace("modsem", quietly = TRUE)) {
```

```

m <- "
OwnLook =~ smv_attr_face + smv_attr_body + smv_sexy
SelfEst =~ ses_satis + ses_qualities + ses_able_todo
MentWell =~ mwb_optimistic + mwb_useful + mwb_energy
smv =~ smv_kind + smv_caring + smv_understanding +
      smv_make_laughh + smv_funny + smv_sociable
SelfEst ~ OwnLook + smv + smv:OwnLook
MentWell ~ OwnLook + SelfEst + smv + smv:OwnLook
"

est <- modsem::modsem(m, data = rmedsem::mchoice, method = "lms")

# mediated moderation
rmedsem(est, indep = "smv:OwnLook", med = "SelfEst", dep = "MentWell")

# moderated mediation
rmedsem(est, indep = "OwnLook", med = "SelfEst", dep = "MentWell",
        moderator = "smv")
}

## blavaan
if (requireNamespace("blavaan", quietly = TRUE)) {
  # blavaan's fitting functions need the package to be attached
  library(blavaan)
  # short single chain to keep the example fast; use more chains and
  # iterations in practice
  bmod <- bsem(mod.txt, data = rmedsem::hsbdemo, n.chains = 1,
              burnin = 500, sample = 500, seed = 1,
              bcontrol = list(refresh = 0))
  rmedsem(bmod, indep = "math", med = "read", dep = "science")

  # highest density intervals instead of equal-tailed intervals
  if (requireNamespace("HDInterval", quietly = TRUE))
    rmedsem(bmod, indep = "math", med = "read", dep = "science", hdi = TRUE)
}

```

Description

Print, summarize and extract the results of `rmedsem()`.

Usage

```

## S3 method for class 'rmedsem'
summary(object, ...)

## S3 method for class 'summary.rmedsem'

```

```

print(x, digits = 3, ...)

## S3 method for class 'rmedsem'
coef(object, method = NULL, ...)

## S3 method for class 'rmedsem'
confint(object, parm, level = NULL, method = NULL, ...)

## S3 method for class 'rmedsem'
nobs(object, ...)

## S3 method for class 'rmedsem'
as.data.frame(x, ...)

## S3 method for class 'rmedsem'
print(x, digits = 3, indent = 3, ...)

## S3 method for class 'rmedsem_blavaan'
print(x, digits = 3, indent = 3, ...)

## S3 method for class 'rmedsem_modsem'
print(x, digits = 3, indent = 3, ci_moderation = FALSE, ...)

```

Arguments

<code>object</code>	an <code>rmedsem</code> object
<code>...</code>	additional arguments (currently unused)
<code>x</code>	an <code>rmedsem</code> object; for <code>print.summary.rmedsem()</code> a <code>summary.rmedsem</code> object
<code>digits</code>	an integer, the number of decimal places to print
<code>method</code>	estimation method for the indirect effect, one of <code>object\$est.methods</code> (e.g., "sobel", "delta", "montc", "boot" or "bayes"); see section 'Extracting results' for the default
<code>parm</code>	character vector; a subset of <code>c("indirect", "direct", "total")</code>
<code>level</code>	the confidence level. The intervals are computed by <code>rmedsem()</code> (argument <code>ci.two.tailed</code>), so <code>level</code> can only be used to check that the stored intervals have the requested level.
<code>indent</code>	an integer, the number of spaces to indent
<code>ci_moderation</code>	a logical, whether to print confidence intervals for the moderation effects (moderated mediation with <code>modsem</code> only)

Value

`print()` returns `x` invisibly.

`summary()` returns an object of class `summary.rmedsem`, a list with elements

`package`, `vars`, `standardized`, `nobs`, `ci.level` copied from `object`.

`ci.type` type of the intervals: "CI" or, for blavaan models fitted with `hdi = TRUE`, "HDI".

`p.threshold` the p-value threshold (NULL for blavaan models).

`effects` a data frame with columns `effect`, `method`, `estimate`, `se`, `zval`, `pval`, `lower` and `upper`; NA where a quantity is not available. For blavaan models, `pval` is the posterior probability of the opposite sign.

`mediation` a list with elements `bk` and `zlc` giving the type of mediation (NULL if the approach was not requested). `bk` is one of "none", "complete" or "partial"; `zlc` is one of "indirect-only", "direct-only", "no-effect", "complementary" or "competitive".

`zlc.method` the estimation method used for the Zhao, Lynch & Chen approach.

`effect.size` a named numeric vector with the requested effect sizes (RIT, RID, `upsilon` (adjusted) and `upsilon.unadjusted`).

`effect.size.problems` a named character vector with an entry for each of RIT and RID that should not be interpreted (see [effect-sizes](#)), describing the reason; empty if there is none.

`coef()` returns a named numeric vector with elements `indirect`, `direct` and `total`.

`confint()` returns a matrix with one row per effect and columns giving the lower and upper limits, labelled by their probabilities (e.g., "2.5 %" and "97.5 %") or, for highest density intervals, "lower" and "upper".

`nobs()` returns an integer.

`as.data.frame()` returns a data frame with one row per estimation method of the indirect effect and columns `package`, `method` and the estimates stored for that method (`coef`, `se`, `zval`, `pval`, `lower` and `upper`; blavaan results additionally contain the posterior probabilities `pvpos` and `pvneg` and the evidence ratios `ERpos` and `ERneg`).

Printing and summarizing

`print()` gives a detailed, step-by-step description of the results: a table of the tests of the indirect effect (one column per estimation method), the steps and conclusions of the Baron and Kenny and/or Zhao, Lynch & Chen approaches, and the effect sizes. For blavaan models, the table reports posterior summaries instead; for moderated mediation with `modsem`, the moderation effects are printed in addition.

`summary()` collects the same results in compact form: a table of the indirect (for each estimation method), direct and total effects, the type of mediation, and the effect sizes.

Extracting results

`coef()` returns the estimated indirect, direct and total effects, `confint()` their confidence (or, for blavaan models, credible) intervals, and `nobs()` the number of observations. `as.data.frame()` returns the estimates of the indirect effect for all estimation methods.

The indirect effect is estimated with several methods (see `object$est.methods`), which all give the same point estimate but different standard errors and intervals. By default, `coef()` and `confint()` use the method that also underlies the Zhao, Lynch & Chen approach: "montc" (Monte-Carlo) for lavaan and `modsem`, "boot" (bootstrap) for cSEM and "bayes" for blavaan.

Extending the printed output

`print.rmedsem()` handles all backends that provide the elements described in section 'Adding a backend' of `rmedsem()`. Backends that need a different output provide a method for their subclass, either replacing the default output (`print.rmedsem_blavaan()`) or extending it with `NextMethod()` (`print.rmedsem_modsem()`).

See Also

`rmedsem()`, `effect-sizes`, `plot.rmedsem()`

Examples

```
mod.txt <- "
read ~ math
science ~ read + math
"

mod <- lavaan::sem(mod.txt, data = rmedsem::hsbdemo)
out <- rmedsem(mod, indep = "math", med = "read", dep = "science")

# detailed output
print(out)

# compact summary and its elements
s <- summary(out)
s
s$mediation
s$effects

# extract estimates
coef(out)
confint(out)
confint(out, parm = "indirect", method = "sobel")
nobs(out)
as.data.frame(out)
```

workout

Fitness Center Survey Data from Trondheim

Description

Data from a survey among the members of a fitness center in Trondheim, Norway. Participants rated how important different reasons for working out are to them and how well two physical features describe them. All items except age are measured on a scale from 1 to 6. The data contain missing values; 187 of the 246 rows are complete.

Usage

```
workout
```

Format

workout:

A data frame with 246 rows and 12 columns:

age Age in years

lweight How important is the following to you to work out: to lose weight, [1] not important at all - [6] very important

calories How important is the following to you to work out: to burn calories, [1] not important at all - [6] very important

cweight How important is the following to you to work out: to control my weight, [1] not important at all - [6] very important

body How important is the following to you to work out: to have a good body, [1] not important at all - [6] very important

appear How important is the following to you to work out: to improve my appearance, [1] not important at all - [6] very important

attract How important is the following to you to work out: to look more attractive, [1] not important at all - [6] very important

muscle How important is the following to you to work out: to develop my muscles, [1] not important at all - [6] very important

strength How important is the following to you to work out: to get stronger, [1] not important at all - [6] very important

endur How important is the following to you to work out: to increase my endurance, [1] not important at all - [6] very important

face How well does the following describe you as a person: attractive face, [1] very badly - [6] very well

sexy How well does the following describe you as a person: sexy, [1] very badly - [6] very well

Source

Survey among the members of a fitness center in Trondheim, Norway.

Examples

```
str(workout)

mod.txt <- "
  Attractive =~ face + sexy
  Appearance =~ body + appear + attract
  Muscle      =~ muscle + strength + endur
  Appearance ~ Attractive + age
  Muscle      ~ Appearance + Attractive + age
"

mod <- lavaan::sem(mod.txt, data = workout)
rmedsem(mod, indep = "Attractive", med = "Appearance", dep = "Muscle")
```

Index

- * **datasets**
 - hsbdemo, [4](#)
 - mchoice, [5](#)
 - workout, [15](#)
- as.data.frame.rmedsem
 (rmedsem-methods), [12](#)
- as.data.frame.rmedsem(), [10](#)
- coef.rmedsem(rmedsem-methods), [12](#)
- confint.rmedsem(rmedsem-methods), [12](#)
- effect-sizes, [2](#), [9](#), [10](#), [14](#), [15](#)
- hsbdemo, [4](#)
- mchoice, [5](#)
- NextMethod(), [11](#), [15](#)
- nobs.rmedsem(rmedsem-methods), [12](#)
- plot.rmedsem, [6](#)
- plot.rmedsem(), [10](#), [15](#)
- plot_coef(plot.rmedsem), [6](#)
- plot_effect(plot.rmedsem), [6](#)
- print.rmedsem(rmedsem-methods), [12](#)
- print.rmedsem(), [10](#)
- print.rmedsem_blavaan
 (rmedsem-methods), [12](#)
- print.rmedsem_modsem(rmedsem-methods),
 [12](#)
- print.summary.rmedsem
 (rmedsem-methods), [12](#)
- RID(effect-sizes), [2](#)
- RIT(effect-sizes), [2](#)
- rmedsem, [7](#)
- rmedsem(), [2](#), [3](#), [6](#), [7](#), [12](#), [13](#), [15](#)
- rmedsem-methods, [7](#), [10](#), [12](#)
- stats::confint(), [11](#)
- stats::nobs(), [10](#)
- summary(), [11](#)
- summary.rmedsem(rmedsem-methods), [12](#)
- Upsilon(effect-sizes), [2](#)
- workout, [15](#)