Package 'robmed'

November 5, 2018

Type Package

Title (Robust) Mediation Analysis

Version 0.3.0

Date 2018-11-05

Depends R (>= 3.2.0), ggplot2 (>= 0.9.3), robustbase (>= 0.92-7)

Imports boot (>= 1.3-1), quantreg (>= 5.36), shiny (>= 1.1.0)

Suggests MASS, dplyr, tidyr, testthat

Description Perform mediation analysis via a (fast and robust) bootstrap test.

License GPL (>= 2)

LazyData yes

LazyLoad yes

Author Andreas Alfons [aut, cre]

Maintainer Andreas Alfons <alfons@ese.eur.nl>

Encoding UTF-8

RoxygenNote 6.1.0

NeedsCompilation no

Repository CRAN

Date/Publication 2018-11-05 17:00:03 UTC

R topics documented:

robmed-package 2
BSG2014 4
coef.test_mediation 6
confint.test_mediation 7
cov_control 9
cov_Huber 10
cov_ML 11
fit_mediation 12
fortify.test_mediation 15

plot_mediation 16
p_value 18
reg_control 20
retest 21
robmed-deprecated 22
run_shiny_app 24
summary.test_mediation 24
test_mediation 26
weights.cov_Huber 29

Index **31**

`robmed-package` *(Robust) Mediation Analysis*

Description

Perform mediation analysis via a (fast and robust) bootstrap test.

Details

The DESCRIPTION file:

Package:	robmed
Type:	Package
Title:	(Robust) Mediation Analysis
Version:	0.3.0
Date:	2018-11-05
Depends:	R (>= 3.2.0), ggplot2 (>= 0.9.3), robustbase (>= 0.92-7)
Imports:	boot (>= 1.3-1), quantreg (>= 5.36), shiny (>= 1.1.0)
Suggests:	MASS, dplyr, tidyr, testthat
Description:	Perform mediation analysis via a (fast and robust) bootstrap test.
License:	GPL (>= 2)
LazyData:	yes
LazyLoad:	yes
Authors@R:	c(person("Andreas", "Alfons", email = "alfons@ese.eur.nl", role = c("aut", "cre")))
Author:	Andreas Alfons [aut, cre]
Maintainer:	Andreas Alfons <alfons@ese.eur.nl>
Encoding:	UTF-8
RoxygenNote:	6.1.0

Index of help topics:

```
BSG2014                 Business simulation game data
coef.test_mediation     Coefficients in (robust) mediation analysis
confint.test_mediation
                        Confidence intervals for (robust) mediation
```

	analysis
cov_Huber	Huber M-estimator of location and scatter
cov_ML	Maximum likelihood estimator of mean vector and covariance matrix
cov_control	Tuning parameters for Huber M-estimation of location and scatter
fit_mediation	(Robustly) fit a mediation model
fortify.test_mediation	Convert (robust) mediation analysis results into a data frame for plotting
p_value	p-Values for (robust) mediation analysis
plot_mediation	Plot (robust) mediation analysis results
reg_control	Tuning parameters for MM-regression
retest	Retest for mediation
robmed-deprecated	Deprecated functions in package 'robmed'
robmed-package	(Robust) Mediation Analysis
run_shiny_app	Shiny app: simulation for mediation analysis with outliers
summary.test_mediation	Summary of results from (robust) mediation analysis
test_mediation	(Robust) mediation analysis
weights.cov_Huber	Robustness weights of Huber M-estimation of location and scatter

Author(s)

Andreas Alfons [aut, cre]

Maintainer: Andreas Alfons <alfons@ese.eur.nl>

References

Alfons, A., Ates, N.Y. and Groenen, P.J.F. (2018) A robust bootstrap test for mediation analysis. *ERIM Report Series in Management*, Erasmus Research Institute of Management. URL https://hdl.handle.net/1765/109594.

Examples

```
## compare bootstrap methods for mediation analysis on simulated data
## Not run:
run_shiny_app()

## End(Not run)

## reproduce empirical examples
## Not run:
demo("case1")
demo("case2")
demo("case3")
```

```
## End(Not run)
```

BSG2014 *Business simulation game data*

Description

The data were collected from 354 senior business administration students during a business simulation game at a Western European University.

The game was played for a total of 12 rounds (i.e., two separate games of 6 rounds) as part of the capstone strategy class. Students were randomly assigned to teams of four, and surveyed in three waves: prior to, during, and after the simulation game (with different variables being surveyed in the different waves).

The 354 students formed 92 teams, and the responses of individual students were aggregated to the team level. Leaving out teams with less than 50 percent response rate yields $n = 89$ teams. Only a small subset of the collected variables are included here.

Usage

```
data("BSG2014")
```

Format

A data frame with 89 observations on the following 7 variables.

`ProcessConflict` Process conflict was measured with the three item scale of Jehn (199? - CITATION missing) and responses were aggregated.

`SharedExperience` Teams were randomly formed, no prior shared group experience is expected and shared group experience and training is developed during the first game for the second game. Team performance in the first game, which was determined by objective performance measures, is a good proxy for the level of shared group experience and training.

`TaskConflict` Task conflict was operationalized with the intra-group conflict scale of Jehn (1995). Five items on the presence of conflict were rated on a 5-point Likert scale (1 = none, 5 = a lot) and aggregated.

`TeamCommitment` Team commitment was measured by four items based on Mowday, Steers & Porter (1979) and responses were aggregated.

`TeamPerformance` Team performance in the second game was measured subjectively by the team members' perceptions of the team's functioning. Hackman's (1986) Likert scale items were thereby used to operationalize team performance.

`TMS` Transactive memory systems (TMS) are defined as shared systems that people in relationships develop for encoding, storing, and retrieving information about different substantive domains. TMS was operationalized with Lewis' (2003) 15-item scale that measures the three sub-dimensions of TMS (credibility, specialization and coordination). Team members responded on a 5-point scale (1 = strongly disagree, 5 = strongly agree). Following Lewis (2003), the three sub dimensions were aggregated to form the TMS construct.

`ValueDiversity` Value diversity was operationalized with the short version of Schwartz's Value Survey (SVS) to measure team members' individual values (Lindeman & Verkasalo, 2005). The responses were aggregated with the average of the coefficient of variations of each value dimension among team members.

References

Hackman, J.R. (1986) The Psychology of Self-Management in Organizations. In Pallack, M.S and Perloff, R.O. (Eds.), *Psychology and Work: Productivity, Change, and Employment*, 89–136. Washington, DC: American Psychological Association.

Jehn, K.A. (1995) A Multi-Method Examination of the Benefits and Detriments of Intra-Group Conflict. *Administrative Science Quarterly*, **40**(2), 256–285.

Lewis, K. (2003) Measuring Transactive Memory Systems in the Field: Scale Development and Validation. *Journal of Applied Psychology*, **88**(4), 587–604.

Lindeman, M. and Verkasalo, M. (2005) Measuring Values With the Short Schwartz's Value Survey. *Journal of Personality Assessment*, **85**(2), 170–178.

Mowday, R.T., Steers, R.M. and Porter, L.W. (1979) The Measurement of Organizational Commitment. *Journal of Vocational Behavior*, **14**(2), 224–47.

Examples

```
data("BSG2014")
summary(BSG2014)

## scatterplot matrices for three illustrative mediation analyses

# empirical case 1
x <- "SharedExperience"
y <- "TeamPerformance"
m <- "TMS"
plot(BSG2014[, c(x, y, m)], pch = 21, bg = "black")

# empirical case 2
x <- "ValueDiversity"
y <- "TeamCommitment"
m <- "TaskConflict"
plot(BSG2014[, c(x, y, m)], pch = 21, bg = "black")

# empirical case 3
x <- "ValueDiversity"
y <- "TeamPerformance"
m <- "ProcessConflict"
plot(BSG2014[, c(x, y, m)], pch = 21, bg = "black")
```

`coef.test_mediation` *Coefficients in (robust) mediation analysis*

Description

Extract coefficients from models computed in (robust) mediation analysis.

Usage

```
## S3 method for class 'test_mediation'
coef(object, parm = NULL, ...)

## S3 method for class 'boot_test_mediation'
coef(object, parm = NULL,
  type = c("boot", "data"), ...)

## S3 method for class 'fit_mediation'
coef(object, parm = NULL, ...)
```

Arguments

`object`	an object inheriting from class `"test_mediation"` containing results from (robust) mediation analysis, or an object inheriting from class `"fit_mediation"` containing a (robust) mediation model fit.
`parm`	an integer, character or logical vector specifying the coefficients to be extracted, or `NULL` to extract all coefficients.
`...`	additional arguments are currently ignored.
`type`	a character string specifying whether to extract the means of the bootstrap distribution (`"boot"`; the default), or the coefficient estimates based on the full data set (`"data"`).

Value

A numeric vector containing the requested coefficients.

Author(s)

Andreas Alfons

See Also

`test_mediation`, `fit_mediation`, `confint`

Examples

```
data("BSG2014")

# fit robust mediation model and extract coefficients
fit <- fit_mediation(BSG2014,
                     x = "ValueDiversity",
                     y = "TeamCommitment",
                     m = "TaskConflict")
coef(fit)

# run fast and robust bootstrap test and extract coefficients
test <- test_mediation(fit)
coef(test, type = "data")  # from orignal sample
coef(test, type = "boot")  # means of bootstrap replicates
```

`confint.test_mediation` *Confidence intervals for (robust) mediation analysis*

Description

Extract or compute confidence intervals for coefficients from (robust) mediation analysis.

Usage

```
## S3 method for class 'boot_test_mediation'
confint(object, parm = NULL,
  level = NULL, other = c("boot", "theory"), ...)

## S3 method for class 'sobel_test_mediation'
confint(object, parm = NULL,
  level = 0.95, ...)
```

Arguments

`object`	an object inheriting from class `"test_mediation"` containing results from (robust) mediation analysis.
`parm`	an integer, character or logical vector specifying the coefficients for which to extract or compute confidence intervals, or `NULL` to extract or compute confidence intervals for all coefficients.
`level`	for the `"boot_test_mediation"` method, this is ignored and the confidence level of the bootstrap confidence interval for the indirect effect is used. For the other methods, the confidence level of the confidence intervals to be computed. The default is to compute 95% confidence intervals.

other	a character string specifying how to compute the confidence interval of the effects other than the indirect effect(s). Possible values are "boot" (the default) to compute bootstrap confidence intervals using the normal approximation (i.e., to assume a normal distribution of the corresponding effect with the standard deviation computed from the bootstrap replicates), or "theory" to compute confidence intervals via statistical theory (e.g., based on a t-distribution the coefficients are estimated via regression). Note that this is only relevant for mediation analysis via a bootstrap test, where the confidence interval of the indirect effect is always computed via a percentile-based method due to the asymmetry of its distribution.
...	additional arguments are currently ignored.

Value

A numeric matrix containing the requested confidence intervals.

Author(s)

Andreas Alfons

See Also

test_mediation, coef

Examples

```
data("BSG2014")

# run fast and robust bootstrap test
robust_boot <- test_mediation(BSG2014,
                              x = "ValueDiversity",
                              y = "TeamCommitment",
                              m = "TaskConflict",
                              robust = TRUE)
confint(robust_boot, other = "boot")

# run standard bootstrap test
standard_boot <- test_mediation(BSG2014,
                                x = "ValueDiversity",
                                y = "TeamCommitment",
                                m = "TaskConflict",
                                robust = FALSE)
confint(standard_boot, other = "theory")
```

cov_control *Tuning parameters for Huber M-estimation of location and scatter*

Description

Obtain a list with tuning paramters for cov_Huber.

Usage

```
cov_control(prob = 0.95, max_iterations = 200, tol = 1e-07)
```

Arguments

prob	numeric; probability for the quantile of the χ^2 distribution to be used as cutoff point in the Huber weight function (defaults to 0.95).
max_iterations	an integer giving the maximum number of iterations in the iteratively reweighted algorithm.
tol	a small positive numeric value to be used to determine convergence of the iteratively reweighted algorithm.

Value

A list with components corresponding to the arguments.

Author(s)

Andreas Alfons

References

Huber, P.J. (1981) *Robust statistics*. John Wiley & Sons.

See Also

cov_Huber

Examples

```
data("BSG2014")

# run fast and robust bootstrap test
ctrl <- cov_control(prob = 0.95)
test <- test_mediation(BSG2014,
                       x = "ValueDiversity",
                       y = "TeamCommitment",
                       m = "TaskConflict",
                       method = "covariance",
                       control = ctrl)
```

```
summary(test)
```

`cov_Huber` *Huber M-estimator of location and scatter*

Description

Compute a Huber M-estimator of location and scatter, which is reasonably robust for a small number of variables.

Usage

```
cov_Huber(x, control = cov_control(...), ...)
```

Arguments

`x`	a numeric matrix or data frame.
`control`	a list of tuning parameters as generated by `cov_control`.
`...`	additional arguments can be used to specify tuning parameters directly instead of via `control`.

Details

An iterative reweighting algorithm is used to compute the Huber M-estimator. The Huber weight function thereby corresponds to a convex optimization problem, resulting in a unique solution.

Value

An object of class `"cov_Huber"` with the following components:

`center`	a numeric vector containing the location vector estimate.
`cov`	a numeric matrix containing the scatter matrix estimate.
`prob`	numeric; probability for the quantile of the χ^2 distribution used as cutoff point in the Huber weight function.
`weights`	a numeric vector containing the relative robustness weights for the observations.
`tau`	numeric; correction for Fisher consistency under multivariate normal distributions.
`converged`	a logical indicating whether the iterative reweighting algorithm converged.
`iterations`	an integer giving the number of iterations required to obtain the solution.

Author(s)

Andreas Alfons

References

Huber, P.J. (1981) *Robust statistics*. John Wiley & Sons.

Zu, J. and Yuan, K.-H. (2010) Local influence and robust procedures for mediation analysis. *Multivariate Behavioral Research*, **45**(1), 1–44.

See Also

cov_control, test_mediation, fit_mediation

Examples

```
data("BSG2014")

# define variables
x <- "ValueDiversity"
y <- "TeamCommitment"
m <- "TaskConflict"

# compute Huber M-estimator
cov_Huber(BSG2014[, c(x, y, m)])
```

---

cov_ML *Maximum likelihood estimator of mean vector and covariance matrix*

Description

Compute the maximum likelihood estimator of the mean vector and the covariance matrix.

Usage

```
cov_ML(x, ...)
```

Arguments

x	a numeric matrix or data frame.
...	additional arguments are currently ignored.

Value

An object of class "cov_ML" with the following components:

center	a numeric vector containing the mean vector estimate.
cov	a numeric matrix containing the covariance matrix estimate.
n	an integer giving the number of observations.

Author(s)

Andreas Alfons

References

Zu, J. and Yuan, K.-H. (2010) Local influence and robust procedures for mediation analysis. *Multivariate Behavioral Research*, **45**(1), 1–44.

See Also

test_mediation, fit_mediation

Examples

```
data("BSG2014")

# define variables
x <- "ValueDiversity"
y <- "TeamCommitment"
m <- "TaskConflict"

# compute Huber M-estimator
cov_ML(BSG2014[, c(x, y, m)])
```

---

fit_mediation *(Robustly) fit a mediation model*

Description

(Robustly) estimate the effects in a mediation model.

Usage

```
fit_mediation(data, x, y, m, covariates = NULL,
  method = c("regression", "covariance"), robust = TRUE,
  median = FALSE, control, ...)
```

Arguments

data	a data frame containing the variables.
x	a character string, an integer or a logical vector specifying the column of data containing the independent variable.
y	a character string, an integer or a logical vector specifying the column of data containing the dependent variable.
m	a character, integer or logical vector specifying the columns of data containing the hypothesized mediator variables.

covariates	optional; a character, integer or logical vector specifying the columns of data containing additional covariates to be used as control variables.
method	a character string specifying the method of estimation. Possible values are "regression" (the default) to estimate the effects via regressions, or "covariance" to estimate the effects via the covariance matrix. Note that the effects are always estimated via regressions if more than one hypothesized mediator is supplied in m, or if control variables are specified via covariates.
robust	a logical indicating whether to robustly estimate the effects (defaults to TRUE).
median	a logical indicating if the effects should be estimated via median regression (defaults to FALSE). This is ignored unless method is "regression" and robust is TRUE.
control	a list of tuning parameters for the corresponding robust method. For robust regression (method = "regression", robust = TRUE and median = FALSE), a list of tuning parameters for lmrob as generated by reg_control. For Huberized covariance matrix estimation (method = "covariance" and robust = TRUE), a list of tuning parameters for cov_Huber as generated by cov_control. No tuning parameters are necessary for median regression (method = "regression", robust = TRUE and median = TRUE).
...	additional arguments can be used to specify tuning parameters directly instead of via control.

Details

If method is "regression", robust is TRUE and median is FALSE (the defaults), the effects are estimated via robust regressions with lmrob.

If method is "regression", robust is TRUE and median is TRUE, the effects are estimated via median regressions with rq. Unlike the robust regressions above, median regressions are not robust against outliers in the explanatory variables.

If method is "covariance" and robust is TRUE, the effects are estimated based on a Huber M-estimator of location and scatter. Note that this covariance-based approach is less robust than the approach based on robust regressions described above.

Value

An object inheriting from class "fit_mediation" (class "reg_fit_mediation" if method is "regression" or "cov_fit_mediation" if method is "covariance") with the following components:

a	a numeric vector containing the point estimates of the effect of the independent variable on the proposed mediator variables.
b	a numeric vector containing the point estimates of the direct effect of the proposed mediator variables on the dependent variable.
c	numeric; the point estimate of the direct effect of the independent variable on the dependent variable.
c_prime	numeric; the point estimate of the total effect of the independent variable on the dependent variable.

`fit_mx` an object of class "lmrob" or "lm" containing the estimation results from the regression of the proposed mediator variable on the independent variable, or a list of such objects in case of more than one hypothesized mediator (only "reg_fit_mediation").

`fit_ymx` an object of class "lmrob" or "lm" containing the estimation results from the regression of the dependent variable on the proposed mediator and independent variables (only "reg_fit_mediation").

`fit_yx` an object of class "lm" containing the estimation results from the regression of the dependent variable on the independent variable (only "reg_fit_mediation" and if robust is FALSE).

`cov` an object of class "cov_Huber" or "cov_ML" containing the covariance matrix estimates (only "cov_fit_mediation").

`x, y, m, covariates` character vectors specifying the respective variables used.

`data` a data frame containing the independent, dependent and proposed mediator variables, as well as covariates.

`robust` a logical indicating whether the effects were estimated robustly.

`median` a logical indicating whether the effects were estimated via median regression (only "reg_fit_mediation").

`control` a list of tuning parameters used (only if robust is TRUE).

Author(s)

Andreas Alfons

References

Alfons, A., Ates, N.Y. and Groenen, P.J.F. (2018) A robust bootstrap test for mediation analysis. *ERIM Report Series in Management*, Erasmus Research Institute of Management. URL https://hdl.handle.net/1765/109594.

Yuan, Y. and MacKinnon, D.P. (2014) Robust mediation analysis based on median regression. *Psychological Methods*, **19**(1), 1–20.

Zu, J. and Yuan, K.-H. (2010) Local influence and robust procedures for mediation analysis. *Multivariate Behavioral Research*, **45**(1), 1–44.

See Also

test_mediation

lmrob, lm, cov_Huber, cov_ML

Examples

```
data("BSG2014")
fit <- fit_mediation(BSG2014,
                     x = "ValueDiversity",
                     y = "TeamCommitment",
```

```
                    m = "TaskConflict")
test <- test_mediation(fit)
summary(test)
```

`fortify.test_mediation` *Convert (robust) mediation analysis results into a data frame for plotting*

Description

Supplement the estimated coefficients with other useful information for informative visualization of the (robust) mediation analysis results. It is thereby possible to construct data frames for dot plots of selected coefficients, as well as density plots of the indirect effect.

Usage

```
## S3 method for class 'boot_test_mediation'
fortify(model, data, method = c("dot",
  "density"), parm = NULL, ...)

## S3 method for class 'sobel_test_mediation'
fortify(model, data, method = c("dot",
  "density"), parm = NULL, level = 0.95, ...)

## S3 method for class 'list'
fortify(model, data, ...)
```

Arguments

`model`	an object inheriting from class `"test_mediation"` containing results from (robust) mediation analysis.
`data`	for the `"boot_test_mediation"` method, this is currently ignored. For the `"sobel_test_mediation"` method, this is an optional numeric vector containing the x-values at which to evaluate the assumed normal density from Sobel's test (only used in case of a density plot). The default is to take 100 equally spaced points between the estimated indirect effect $\pm$ three times the standard error according to Sobel's formula.
`method`	a character string specifying for which plot to construct the data frame. Possible values are `"dot"` for a dot plot of selected coefficients, or `"density"` for a density plot of the indirect effect(s).
`parm`	a character string specifying the coefficients to be included in a dot plot. The default is to include the direct and the indirect effect(s).
`...`	additional arguments are currently ignored.
`level`	numeric; the confidence level of the confidence intervals from Sobel's test to be included in a dot plot. The default is to include 95% confidence intervals.

Value

A data frame containing the necessary data for the selected plot, as well as additional information stored in attributes.

Author(s)

Andreas Alfons

See Also

test_mediation, plot_mediation

Examples

```
data("BSG2014")

# run fast and robust bootstrap test
test <- test_mediation(BSG2014,
                       x = "ValueDiversity",
                       y = "TeamCommitment",
                       m = "TaskConflict")

# data for dot plot
dot <- fortify(test, method = "dot")
plot_mediation(dot)

# data for density plot
density <- fortify(test, method = "density")
plot_mediation(density)
```

---

plot_mediation *Plot (robust) mediation analysis results*

Description

Produce dot plots of selected coefficients from regression models computed in (robust) mediation analysis, or density plots of the indirect effect.

Usage

```
plot_mediation(object, ...)

## S3 method for class 'boot_test_mediation'
plot_mediation(object, method = c("dot",
  "density"), parm = NULL, ...)

## S3 method for class 'sobel_test_mediation'
```

```
plot_mediation(object, data,
  method = c("dot", "density"), parm = c("c", "ab"), level = 0.95,
  ...)

## S3 method for class 'list'
plot_mediation(object, data, method = c("dot", "density"),
  parm = NULL, level = 0.95, ...)

## Default S3 method:
plot_mediation(object, mapping = attr(object,
  "mapping"), facets = attr(object, "facets"), ...)

## S3 method for class 'test_mediation'
autoplot(object, ...)

## S3 method for class 'test_mediation'
plot(x, ...)
```

Arguments

object, x	an object inheriting from class "test_mediation" containing results from (robust) mediation analysis. For plot_mediation, a list of such objects may be supplied as well.
...	additional arguments to be passed to and from methods.
method	a character string specifying which plot to produce. Possible values are "dot" for a dot plot of selected coefficients, or "density" for a density plot of the indirect effect(s).
parm	a character string specifying the coefficients to be included in a dot plot. The default is to include the direct and the indirect effect(s).
data	an optional numeric vector containing the x-values at which to evaluate the assumed normal density from Sobel's test (only used in case of a density plot). The default is to take 100 equally spaced points between the estimated indirect effect $\pm$ three times the standard error according to Sobel's formula.
level	numeric; the confidence level of the confidence intervals from Sobel's test to be included in a dot plot. The default is to include 95% confidence intervals.
mapping	an aesthetic mapping to override the default behavior (see aes or aes_).
facets	a faceting formula to override the default behavior (only used in case of a dot plot). If supplied, facet_wrap or facet_grid is called depending on whether the formula is one-sided or two-sided.

Value

An object of class "ggplot" (see ggplot).

Author(s)

Andreas Alfons

See Also

test_mediation, fortify

Examples

```
data("BSG2014")

# run fast and robust bootstrap test
robust_boot <- test_mediation(BSG2014,
                              x = "ValueDiversity",
                              y = "TeamCommitment",
                              m = "TaskConflict",
                              robust = TRUE)

# create plots for robust bootstrap test
plot(robust_boot, method = "dot")
plot(robust_boot, method = "density")

# run standard bootstrap test
standard_boot <- test_mediation(BSG2014,
                                x = "ValueDiversity",
                                y = "TeamCommitment",
                                m = "TaskConflict",
                                robust = FALSE)

# compare robust and standard tests
tests <- list(Robust = robust_boot, Standard = standard_boot)
plot_mediation(tests, method = "dot")
plot_mediation(tests, method = "density")
```

---

p_value *p-Values for (robust) mediation analysis*

Description

Estimate or extract the p-values for indirect effects in (robust) mediation analysis.

Usage

```
p_value(object, ...)

## S3 method for class 'boot_test_mediation'
p_value(object, digits = 4L, ...)

## S3 method for class 'sobel_test_mediation'
p_value(object, ...)
```

Arguments

`object`	an object inheriting from class `test_mediation` containing results from (robust) mediation analysis.
`...`	additional arguments are currently ignored.
`digits`	an integer determining the number of digits of the p-values to be computed. The default is to compute 4 digits after the comma.

Details

For bootstrap tests, the p-value is estimated as the smallest significance level α for which the $(1-\alpha)*100\%$ confidence interval obtained from the bootstrapped distribution of the indirect effect does not contain 0.

This is a simple implementation, where each digit after the comma is determined via a grid search. Hence computation time can be long if confidence intervals are computed via the bias-corrected and accelerated method (`"bca"`).

For Sobel tests, the p-value is already stored in the object returned by `test_mediation` and simply extracted.

Value

A numeric vector containing the p-values for the indirect effect(s).

Author(s)

Andreas Alfons

See Also

`test_mediation`, `boot.ci`

Examples

```
data("BSG2014")

## Not run:
# BCa intervals are recommended, but take a while to run
test_bca <- test_mediation(BSG2014,
                           x = "ValueDiversity",
                           y = "TeamCommitment",
                           m = "TaskConflict",
                           type = "bca")
p_value(test_bca)

## End(Not run)
```

reg_control *Tuning parameters for MM-regression*

Description

Obtain a list with tuning paramters for lmrob.

Usage

```
reg_control(efficiency = 0.85, max_iterations = 200, tol = 1e-07,
  seed = NULL)
```

Arguments

efficiency	a numeric value giving the desired efficiency (defaults to 0.85 for 85% efficiency).
max_iterations	an integer giving the maximum number of iterations in various parts of the algorithm.
tol	a small positive numeric value to be used to determine convergence in various parts of the algorithm.
seed	optional initial seed for the random number generator (see .Random.seed).

Value

A list of tuning parameters as returned by lmrob.control.

Note

This is a simplified wrapper function for lmrob.control, as the latter requires detailed knowledge of the MM-type regression algorithm. Currently only 95%, 90%, 85% (the default) and 80% efficiency are supported. For other values, please specify the corresponding tuning parameters in lmrob.control directly.

Author(s)

Andreas Alfons

References

Salibian-Barrera, M. and Yohai, V.J. (1987) A fast algorithm for S-regression estimates. *Journal of Computational and Graphical Statistics*, **15**(2), 414–427.

Yohai, V.J. (1987) High breakdown-point and high efficiency estimates for regression. *The Annals of Statistics*, **15**(20), 642–656.

See Also

lmrob, lmrob.control

Examples

```
data("BSG2014")

# run fast and robust bootstrap test
ctrl <- reg_control(efficiency = 0.95)
test <- test_mediation(BSG2014,
                       x = "ValueDiversity",
                       y = "TeamCommitment",
                       m = "TaskConflict",
                       control = ctrl)
summary(test)
```

`retest` *Retest for mediation*

Description

Reperform a (fast and robust) bootstrap test or Sobel's test for the indirect effect(s) based on results from (robust) mediation analysis.

Usage

```
retest(object, ...)

## S3 method for class 'boot_test_mediation'
retest(object, alternative = c("twosided",
  "less", "greater"), level = 0.95, type = c("bca", "perc"), ...)

## S3 method for class 'sobel_test_mediation'
retest(object, alternative = c("twosided",
  "less", "greater"), ...)
```

Arguments

`object`	an object inheriting from class `"test_mediation"` containing results from (robust) mediation analysis.
`...`	additional arguments to be passed down to methods.
`alternative`	a character string specifying the alternative hypothesis in the test for the indirect effect. Possible values are `"twosided"` (the default), `"less"` or `"greater"`.
`level`	numeric; the confidence level of the confidence interval in the bootstrap test. The default is to compute a 95% confidence interval.
`type`	a character string specifying the type of confidence interval to be computed in the bootstrap test. Possible values are `"bca"` (the default) for the bias-corrected and accelerated bootstrap, or `"perc"` for the percentile bootstrap.

Value

An object of the same class as `object` with updated test results (see `test_mediation`).

Author(s)

Andreas Alfons

See Also

`test_mediation`

Examples

```
data("BSG2014")

# run fast and robust bootstrap test
test <- test_mediation(BSG2014,
                       x = "ValueDiversity",
                       y = "TeamCommitment",
                       m = "TaskConflict")
summary(test)

# now compute 97.5% confidence interval
retest(test, level = 0.975)
```

`robmed-deprecated` *Deprecated functions in package* **robmed**

Description

These functions are provided for compatibility with older versions only, and may be defunct as soon as the next release.

Usage

```
fitMediation(x, y, m, covariates = NULL, data, ...)

regControl(efficiency = 0.85, maxIterations = 200, tol = 1e-07,
  seed = NULL)

covControl(prob = 0.95, maxIterations = 200, tol = 1e-07)

covHuber(...)

covML(...)

testMediation(x, ...)
```

```
## Default S3 method:
testMediation(x, y, m, covariates = NULL, data, ...)

## S3 method for class 'fit_mediation'
testMediation(x, ...)

plotMediation(...)
```

Arguments

x	either a numeric vector containing the independent variable, or (if data is supplied) a character string, an integer or a logical vector specifying the corresponding column of data.
y	either a numeric vector containing the dependent variable, or (if data is supplied) a character string, an integer or a logical vector specifying the corresponding column of data.
m	either a numeric vector or data frame containing the proposed mediator variables, or (if data is supplied) a character, integer or logical vector specifying the corresponding columns of data.
covariates	optional; either a numeric vector or data frame containing additional covariates to be used as control variables, or (if data is supplied) a character, integer or logical vector specifying the corresponding columns of data.
data	an optional data.frame containing the variables.
...	arguments to be passed down
efficiency	a numeric value giving the desired efficiency (defaults to 0.85 for 85% efficiency).
maxIterations	an integer giving the maximum number of iterations in iterative algorithms.
tol	a small positive numeric value to be used to determine convergence in various parts of the algorithm.
seed	optional initial seed for the random number generator (see .Random.seed).
prob	numeric; probability for the quantile of the χ^2 distribution to be used as cutoff point in the Huber weight function (defaults to 0.95).

Author(s)

Andreas Alfons

See Also

Deprecated

fit_mediation, test_mediation, plot_mediation

reg_control, cov_control

cov_Huber, cov_ML

run_shiny_app *Shiny app: simulation for mediation analysis with outliers*

Description

Compare various bootstrap methods for mediation analysis on simulated data.

Usage

```
run_shiny_app()
```

Details

The default settings follow the simulation design of Zu & Yuan (2010). You can adjust the total number of observations, the values of the coefficients in the mediation model, the number of outliers, as well as the expected distance of the outliers from the main point cloud.

As this simulation is just for illustration, the bootstrap procedures use only 1000 replicates. For each selected methods, the bootstrap distribution of the indirect effect is shown together with a shaded area representing the 95% confidence interval.

Author(s)

Andreas Alfons

See Also

test_mediation

Examples

```
## Not run:
run_shiny_app()

## End(Not run)
```

summary.test_mediation *Summary of results from (robust) mediation analysis*

Description

Summarize results from (robust) mediation analysis for proper interpretation.

Usage

```
## S3 method for class 'boot_test_mediation'
summary(object, other = c("boot",
  "theory"), ...)

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

Arguments

object	an object inheriting from class "test_mediation" containing results from (robust) mediation analysis.
other	a character string specifying how to summarize the effects other than the indirect effect(s). Possible values are "boot" (the default) to compute significance tests using the normal approximation of the bootstrap distribution (i.e., to assume a normal distribution of the corresponding effect with the standard deviation computed from the bootstrap replicates), or "theory" to compute significance tests via statistical theory (e.g., t-tests if the coefficients are estimated via regression). Note that this is only relevant for mediation analysis via a bootstrap test, where significance of the indirect effect is always assessed via a percentile-based confidence interval due to the asymmetry of its distribution.
...	additional arguments are currently ignored.

Value

An object of class "summary_test_mediation" with the following components:

object	the object passed to the summary method, which contains the results from testing the indirect effect.
summary	an object containing all necessary information to summarize the effects other than the indirect effect.

Author(s)

Andreas Alfons

See Also

test_mediation

Examples

```
data("BSG2014")
test <- test_mediation(BSG2014,
                       x = "ValueDiversity",
                       y = "TeamCommitment",
                       m = "TaskConflict")
summary(test)
```

`test_mediation` *(Robust) mediation analysis*

Description

Perform (robust) mediation analysis via a (fast and robust) bootstrap test or Sobel's test.

Usage

```
test_mediation(data, ...)

## Default S3 method:
test_mediation(data, x, y, m, covariates = NULL,
  test = c("boot", "sobel"), alternative = c("twosided", "less",
  "greater"), R = 5000, level = 0.95, type = c("bca", "perc"),
  method = c("regression", "covariance"), robust = TRUE,
  median = FALSE, control, ...)

## S3 method for class 'fit_mediation'
test_mediation(data, test = c("boot", "sobel"),
  alternative = c("twosided", "less", "greater"), R = 5000,
  level = 0.95, type = c("bca", "perc"), ...)

robmed(..., test = "boot", method = "regression", robust = TRUE,
  median = FALSE)

indirect(..., test = "boot", method = "regression", robust = FALSE,
  median = FALSE)
```

Arguments

`data`	a data frame containing the variables. Alternatively, this can be a mediation model fit as returned by `fit_mediation`.
`...`	additional arguments to be passed down. For the bootstrap tests, those can be used to specify arguments of `boot`, for example for parallel computing.
`x`	a character string, an integer or a logical vector specifying the column of `data` containing the independent variable.
`y`	a character string, an integer or a logical vector specifying the column of `data` containing the dependent variable.
`m`	a character, integer or logical vector specifying the columns of `data` containing the hypothesized mediator variables.
`covariates`	optional; a character, integer or logical vector specifying the columns of `data` containing additional covariates to be used as control variables.

test	a character string specifying the test to be performed for the indirect effect. Possible values are "boot" (the default) for the bootstrap, or "sobel" for Sobel's test. Currently, Sobel's test is not implemented for more than one hypothesized mediator variable.
alternative	a character string specifying the alternative hypothesis in the test for the indirect effects. Possible values are "twosided" (the default), "less" or "greater".
R	an integer giving the number of bootstrap replicates. The default is to use 5000 bootstrap replicates.
level	numeric; the confidence level of the confidence interval in the bootstrap test. The default is to compute a 95% confidence interval.
type	a character string specifying the type of confidence interval to be computed in the bootstrap test. Possible values are "bca" (the default) for the bias-corrected and accelerated bootstrap, or "perc" for the percentile bootstrap.
method	a character string specifying the method of estimation for the mediation model. Possible values are "regression" (the default) to estimate the effects via regressions, or "covariance" to estimate the effects via the covariance matrix. Note that the effects are always estimated via regressions if more than one hypothesized mediator is supplied in m, or if control variables are specified via covariates.
robust	a logical indicating whether to perform a robust test (defaults to TRUE).
median	a logical indicating if the effects should be estimated via median regression (defaults to FALSE). This is ignored unless method is "regression" and robust is TRUE.
control	a list of tuning parameters for the corresponding robust method. For robust regression (method = "regression", robust = TRUE and median = FALSE), a list of tuning parameters for lmrob as generated by reg_control. For Huberized covariance matrix estimation (method = "covariance" and robust = TRUE), a list of tuning parameters for cov_Huber as generated by cov_control. No tuning parameters are necessary for median regression (method = "regression", robust = TRUE and median = TRUE).

Details

If method is "regression", robust is TRUE and median is FALSE (the defaults), the tests are based on robust regressions with lmrob. The bootstrap test is thereby performed via the fast and robust bootstrap.

Note that the regression estimator implemented in lmrob can be seen as weighted least squares estimator, where the weights are dependent on how much an observation is deviating from the rest. The trick for the fast and robust bootstrap is that on each bootstrap sample, first a weighted least squares estimator is computed (using those robustness weights from the original sample) followed by a linear correction of the coefficients. The purpose of this correction is to account for the additional uncertainty of obtaining the robustness weights.

If method is "regression", robust is TRUE and median is TRUE, the tests are based on median regressions with rq and the standard bootstrap (). Unlike the robust regressions described above, median regressions are not robust against outliers in the explanatory variables, and the standard bootstrap can suffer from oversampling of outliers in the bootstrap samples.

If method is "covariance" and robust is TRUE, the tests are based on a Huber M-estimator of location and scatter. For the bootstrap test, the M-estimates are used to first clean the data via a transformation. Then the standard bootstrap is performed with the cleaned data. Note that this covariance-based approach is less robust than the approach based on robust regressions described above. Furthermore, the bootstrap does not account for the variability from cleaning the data.

robmed is a wrapper function for performing robust mediation analysis via regressions and the fast and robust bootstrap.

indirect is a wrapper function for performing non-robust mediation analysis via regressions and the bootstrap (inspired by Preacher & Hayes' SPSS macro INDIRECT).

Value

An object inheriting from class "test_mediation" (class "boot_test_mediation" if test is "boot" or "sobel_test_mediation" if test is "sobel") with the following components:

ab	a numeric vector containing the point estimates of the indirect effects.
ci	a numeric vector of length two or a matrix of two columns containing the bootstrap confidence intervals for the indirect effects (only "boot_test_mediation").
reps	an object of class "boot" containing the bootstrap replicates of the effects (only "boot_test_mediation").
se	numeric; the standard error of the indirect effect according to Sobel's formula (only "sobel_test_mediation").
statistic	numeric; the test statistic for Sobel's test (only "sobel_test_mediation").
p_value	numeric; the p-value from Sobel's test (only "sobel_test_mediation").
alternative	a character string specifying the alternative hypothesis in the test for the indirect effects.
R	an integer giving the number of bootstrap replicates (only "boot_test_mediation").
level	numeric; the confidence level of the bootstrap confidence interval (only "boot_test_mediation").
type	a character string specifying the type of bootstrap confidence interval (only "boot_test_mediation").
fit	an object inheriting from class "fit_mediation" containing the estimation results for the direct effect and the total effect in the mediation model.

Note

For the fast and robust bootstrap, the simpler correction of Salibian-Barrera & Van Aelst (2008) is used rather than the originally proposed correction of Salibian-Barrera & Zamar (2002).

Author(s)

Andreas Alfons

References

Alfons, A., Ates, N.Y. and Groenen, P.J.F. (2018) A robust bootstrap test for mediation analysis. *ERIM Report Series in Management*, Erasmus Research Institute of Management. URL https://hdl.handle.net/1765/109594.

Preacher, K.J. and Hayes, A.F. (2004) SPSS and SAS procedures for estimating indirect effects in simple mediation models. *Behavior Research Methods, Instruments, & Computers*, **36**(4), 717–731.

Preacher, K.J. and Hayes, A.F. (2008) Asymptotic and resampling strategies for assessing and comparing indirect effects in multiple mediator models. *Behavior Research Methods*, **40**(3), 879–891.

Salibian-Barrera, M. and Van Aelst, S. (2008) Robust model selection using fast and robust bootstrap. *Computational Statistics & Data Analysis*, **52**(12), 5121–5135

Salibian-Barrera, M. and Zamar, R. (2002) Bootstrapping robust estimates of regression. *The Annals of Statistics*, **30**(2), 556–582.

Sobel, M.E. (1982) Asymptotic confidence intervals for indirect effects in structural equation models. *Sociological Methodology*, **13**, 290–312.

Yuan, Y. and MacKinnon, D.P. (2014) Robust mediation analysis based on median regression. *Psychological Methods*, **19**(1), 1–20.

Zu, J. and Yuan, K.-H. (2010) Local influence and robust procedures for mediation analysis. *Multivariate Behavioral Research*, **45**(1), 1–44.

See Also

`fit_mediation`

`coef`, `confint`, `fortify` and `plot` methods, `p_value`

`boot`, `lmrob`, `lm`, `cov_Huber`, `cov_ML`

Examples

```
data("BSG2014")
test <- test_mediation(BSG2014,
                       x = "ValueDiversity",
                       y = "TeamCommitment",
                       m = "TaskConflict")
summary(test)
```

`weights.cov_Huber` *Robustness weights of Huber M-estimation of location and scatter*

Description

Extract (relative) robustness weights of a Huber M-estimate of location and scatter.

Usage

```
## S3 method for class 'cov_Huber'
weights(object, type = c("consistent", "relative"),
  ...)
```

Arguments

object	an object inheriting from class "cov_Huber" containing Huber M-estimates of location and scatter.
type	a character string specifying the type of robustness weights to be extracted. Possible values are "consistent" and "relative". The former can be used for a robust transformation of the data such that the covariance matrix of the transformed data is Fisher consistent. Observations that are not downweighted in general receive a weight larger than 1. The latter are useful for interpretation, as observations that are not downweighted receive a relative weight of 1.
...	additional arguments are currently ignored.

Value

A numeric vetor containing the requested robustness weights.

Author(s)

Andreas Alfons

References

Zu, J. and Yuan, K.-H. (2010) Local influence and robust procedures for mediation analysis. *Multivariate Behavioral Research*, **45**(1), 1–44.

See Also

cov_Huber

Examples

```
data("BSG2014")

# define variables
x <- "ValueDiversity"
y <- "TeamCommitment"
m <- "TaskConflict"

# compute Huber M-estimator
S <- cov_Huber(BSG2014[, c(x, y, m)])
weights(S, type = "relative")
```

Index

∗Topic **datasets**
BSG2014, 4
∗Topic **documentation**
run_shiny_app, 24
∗Topic **hplot**
plot_mediation, 16
∗Topic **multivariate**
cov_control, 9
cov_Huber, 10
cov_ML, 11
fit_mediation, 12
retest, 21
robmed-deprecated, 22
test_mediation, 26
∗Topic **package**
robmed-package, 2
∗Topic **regression**
reg_control, 20
∗Topic **utilities**
coef.test_mediation, 6
confint.test_mediation, 7
fortify.test_mediation, 15
p_value, 18
summary.test_mediation, 24
weights.cov_Huber, 29
.Random.seed, *20*, *23*

aes, *17*
aes_, *17*
autoplot.test_mediation *(*plot_mediation*)*, 16

boot, *26*, *28*, *29*
boot.ci, *19*
BSG2014, 4

coef, *8*, *29*
coef.boot_test_mediation *(*coef.test_mediation*)*, 6
coef.fit_mediation *(*coef.test_mediation*)*, 6
coef.test_mediation, 6
confint, *6*, *29*
confint.boot_test_mediation *(*confint.test_mediation*)*, 7
confint.sobel_test_mediation *(*confint.test_mediation*)*, 7
confint.test_mediation, 7
cov_control, 9, *10*, *11*, *13*, *23*, *27*
cov_Huber, *9*, 10, *13*, *14*, *23*, *27*, *29*, *30*
cov_ML, 11, *14*, *23*, *29*
covControl *(*robmed-deprecated*)*, 22
covHuber *(*robmed-deprecated*)*, 22
covML *(*robmed-deprecated*)*, 22

Deprecated, *23*

facet_grid, *17*
facet_wrap, *17*
fit_mediation, *6*, *11*, *12*, 12, *23*, *26*, *28*, *29*
fitMediation *(*robmed-deprecated*)*, 22
fortify, *18*, *29*
fortify.boot_test_mediation *(*fortify.test_mediation*)*, 15
fortify.list *(*fortify.test_mediation*)*, 15
fortify.sobel_test_mediation *(*fortify.test_mediation*)*, 15
fortify.test_mediation, 15

ggplot, *17*

indirect *(*test_mediation*)*, 26

lm, *14*, *29*
lmrob, *13*, *14*, *20*, *27*, *29*
lmrob.control, *20*

p_value, 18, *29*
plot, *29*

plot.test_mediation *(*plot_mediation*)*, 16
plot_mediation, *16*, 16, *23*
plotMediation *(*robmed-deprecated*)*, 22
print.boot_test_mediation
        *(*test_mediation*)*, 26
print.cov_Huber *(*cov_Huber*)*, 10
print.cov_ML *(*cov_ML*)*, 11
print.fit_mediation *(*fit_mediation*)*, 12
print.sobel_test_mediation
        *(*test_mediation*)*, 26

reg_control, *13*, 20, *23*, *27*
regControl *(*robmed-deprecated*)*, 22
retest, 21
robmed *(*test_mediation*)*, 26
robmed-deprecated, 22
robmed-package, 2
rq, *13*, *27*
run_shiny_app, 24

summary.boot_test_mediation
        *(*summary.test_mediation*)*, 24
summary.cov_fit_mediation
        *(*fit_mediation*)*, 12
summary.reg_fit_mediation
        *(*fit_mediation*)*, 12
summary.sobel_test_mediation
        *(*summary.test_mediation*)*, 24
summary.test_mediation, 24

test_mediation, *6–8*, *11, 12*, *14–19*, *21–25*,
        26
testMediation *(*robmed-deprecated*)*, 22

weights.cov_Huber, 29