

Package ‘drmeta’

September 29, 2026

Type Package

Title Design-Indexed Location-Scale Meta-Analysis

Version 0.2.3

Description Fits constrained and unrestricted meta-analytic location-scale models in which residual between-study heterogeneity is modeled as an exponential function of a prespecified design-robustness score. The package supports maximum-likelihood (ML) and restricted maximum-likelihood (REML) estimation, location moderators, the conventional random-effects model as a nested special case, exact estimation at the nonnegative scale-gradient boundary, design-indexed heterogeneity summaries, scale-attenuation measures, prediction of fitted heterogeneity, leave-one-out influence diagnostics, and parametric-bootstrap inference for the scale gradient. A grouped scale diagnostic checks whether a monotone curve misses an interior peak or trough. Because the scale-gradient null lies on the boundary of the constrained parameter space, standard chi-square likelihood-ratio references do not apply (Self and Liang, 1987, <doi:10.1080/01621459.1987.10478472>). The general location-scale parent model is described in Viechtbauer and Lopez-Lopez (2022, <doi:10.1002/jrsm.1562>). A scale model reweights studies and does not adjust the mean for design-linked bias.

License MIT + file LICENSE

URL <https://github.com/subirhait/drmeta>

BugReports <https://github.com/subirhait/drmeta/issues>

Encoding UTF-8

Depends R (>= 4.1.0)

Imports grDevices, graphics, parallel, stats, utils

Suggests knitr, metadat, metafor, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/markdown TRUE

NeedsCompilation no
Config/roxygen2/version 8.0.0
Author Subir Hait [aut, cre, cph] (ORCID:
 <<https://orcid.org/0009-0004-9871-9677>>)
Maintainer Subir Hait <haitsubi@msu.edu>
Repository CRAN
Date/Publication 2026-09-29 13:30:13 UTC

Contents

coef.drmeta	2
drmeta	3
drmeta_bootstrap_gamma	5
dr_from_design	6
dr_heterogeneity	7
dr_loo	8
dr_plot_vfun	8
dr_scale_attenuation	9
dr_scale_predict	10
dr_score	11
dr_shape_check	12
normalize_01	13
print.drmeta_bootstrap_gamma	13
print.dr_shape_check	14
Index	15

coef.drmeta	<i>S3 methods for drmeta objects</i>
-------------	--------------------------------------

Description

S3 methods for drmeta objects

Usage

```
## S3 method for class 'drmeta'  
coef(object, ...)  
  
## S3 method for class 'drmeta'  
vcov(object, ...)  
  
## S3 method for class 'drmeta'  
fitted(object, ...)  
  
## S3 method for class 'drmeta'
```

```

residuals(object, ...)

## S3 method for class 'drmeta'
print(x, ...)

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

## S3 method for class 'summary.drmeta'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

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

## S3 method for class 'drmeta'
logLik(object, REML = NULL, ...)

```

Arguments

<code>object, x</code>	A fitted <code>drmeta</code> object.
<code>...</code>	Ignored.
<code>digits</code>	Number of significant digits.
<code>parm</code>	Character or numeric selection of location parameters.
<code>level</code>	Confidence level.
<code>REML</code>	If TRUE, report the restricted log-likelihood.

Value

`coef` and `fitted` and `residuals` return numeric vectors; `vcov` returns the location covariance matrix; `confint` returns a matrix of confidence limits; `logLik` returns an object of class `logLik` carrying `df` and `nobs` attributes; `summary` returns an object of class `summary.drmeta`; the `print` methods return their argument invisibly.

<code>drmeta</code>	<i>Fit a design-indexed meta-analytic location-scale model</i>
---------------------	--

Description

Fit a design-indexed meta-analytic location-scale model

Usage

```

drmeta(
  yi,
  vi,
  dr,

```

```

    mods = NULL,
    slab = NULL,
    method = c("REML", "ML"),
    constrained = TRUE,
    gamma_max = 8,
    gamma_fixed = NULL,
    control = list(),
    .quiet = FALSE
  )

```

Arguments

<code>yi</code>	Numeric vector of study effect estimates.
<code>vi</code>	Numeric vector of sampling variances.
<code>dr</code>	Numeric design-robustness index in $[0, 1]$.
<code>mods</code>	Optional numeric vector or matrix of location moderators. An intercept is added automatically.
<code>slab</code>	Optional study labels. Defaults to the row numbers and is used by study-level diagnostics such as dr_loo .
<code>method</code>	Either "REML" or "ML".
<code>constrained</code>	If TRUE, impose $\gamma \geq 0$. If FALSE, allow either sign.
<code>gamma_max</code>	Finite optimization bound for $\text{abs}(\gamma)$. The default, 8, matches the range used in the simulation study reported in the accompanying manuscript. Refit under a wider bound whenever the estimate piles up near it.
<code>gamma_fixed</code>	Optional fixed value for γ , used for null fitting.
<code>control</code>	Optional list passed to <code>optim()</code> .
<code>.quiet</code>	If TRUE, suppress the design-diagnostic warnings about the distribution of <code>dr</code> . Intended for internal refitting, where the design index is identical across calls and the diagnostics would repeat without conveying new information. Errors and input validation are unaffected.

Value

An object of class `drmeta`: a list whose components include `beta` (location coefficients), `vcov`, `ci_beta`, `tau0sq`, `gamma`, `tau2i`, `sigma2`, `weights`, `residuals`, `slab`, `logLik`, `method`, `constrained`, `convergence`, and the supplied data.

Examples

```

path <- system.file("extdata", "bcg_design_robustness.csv", package = "drmeta")
bcg <- utils::read.csv(path)
fit <- drmeta(yi = bcg[["yi"]], vi = bcg[["vi"]], dr = bcg[["dr"]])
summary(fit)

```

drmeta_bootstrap_gamma

Parametric-bootstrap test of the constrained scale gradient

Description

Tests $H_0: \gamma = 0$ against $H_1: \gamma > 0$. The null lies on the boundary of the constrained parameter space, so the likelihood-ratio statistic does not have an asymptotic chi-square distribution and a parametric bootstrap is used instead. Data are simulated from the fitted null model, in which between-study heterogeneity is constant in the design-robustness index.

Usage

```
drmeta_bootstrap_gamma(
  object,
  B = 999,
  seed = NULL,
  parallel = FALSE,
  ncpus = 2L,
  tol = 1e-08
)
```

Arguments

object	A constrained drmeta fit.
B	Number of bootstrap samples.
seed	Optional random seed.
parallel	Logical; use mclapply on non-Windows platforms.
ncpus	Number of cores when parallel is TRUE.
tol	Tolerance below which the observed likelihood-ratio statistic is treated as exactly zero.

Details

Replicates in which either refit fails to converge are discarded rather than contributing a statistic computed at a non-optimal point. The p-value uses the number of usable replicates as its denominator, and the discarded count is returned so the loss is visible.

When the constrained estimate is already at the boundary, the null and alternative fits coincide and the observed statistic is zero by construction. Bootstrapping in that situation would compare zero against a simulated distribution with a large point mass at zero, and would return a p-value that looks like evidence but only records the proportion of replicates that also reached the boundary. The function therefore short-circuits, returns `p.value = 1` with `boundary = TRUE`, and runs no replicates. A boundary estimate means the data provide no support for a positive scale gradient; it is not a measured degree of evidence. Fit with `constrained = FALSE` to see whether the unrestricted

gradient is negative, which would contradict the substantive constraint rather than merely fail to support it.

When `parallel = TRUE`, reproducibility across cores requires the L'Ecuyer-CMRG generator, which this function sets and restores when a seed is supplied.

Value

An object of class `drmeta_bootstrap_gamma`, a list with components `statistic` (observed likelihood-ratio statistic), `p.value`, `B` (requested replicates), `B_used` (replicates in which both refits converged), `n_failed` (replicates discarded), `simulated` (the simulated statistics, with NA for failed replicates), `null` (the fitted null model), `alternative` (the supplied fit), and `boundary`, a logical flag that is TRUE when the constrained estimate already lies at $\gamma = 0$ and no bootstrap was run.

Examples

```
path <- system.file("extdata", "bcg_design_robustness.csv", package = "drmeta")
bcg <- utils::read.csv(path)
fit <- drmeta(yi = bcg[["yi"]], vi = bcg[["vi"]], dr = bcg[["dr"]])
drmeta_bootstrap_gamma(fit, B = 100, seed = 1)
```

dr_from_design

Map study design labels to illustrative design-robustness scores

Description

Provides a convenience mapping from design-type labels to numeric scores.

Usage

```
dr_from_design(
  design,
  custom_map = NULL,
  default_score = 0.25,
  warn_unknown = TRUE
)
```

Arguments

<code>design</code>	Character vector of design labels (case-insensitive).
<code>custom_map</code>	Optional named numeric vector overriding or extending the defaults.
<code>default_score</code>	Score assigned to unrecognised labels. Default 0.25.
<code>warn_unknown</code>	If TRUE, warn about unrecognised labels.

Details

This mapping is a demonstration default, not a validated measurement model and not a universal hierarchy of study quality. A design label does not determine the magnitude or direction of bias in a particular evidence base. For substantive work the index should be constructed for the specific synthesis, outcome, and identification problem at hand, ordinarily with [dr_score](#), and reported under alternative defensible codings.

Value

A numeric vector of scores in $[0, 1]$, the same length as design.

See Also

[dr_score](#)

Examples

```
dr_from_design(c("RCT", "DiD", "OLS", "IV", "matching"))
dr_from_design("house_method", custom_map = c(house_method = 0.65))
```

dr_heterogeneity

Classical heterogeneity diagnostics and design-indexed summaries

Description

Reports the classical fixed-effect residual Q statistic and I^2 alongside the fitted design-indexed scale parameters. Unlike version 0.1.0, this function does not return a design-explained variance decomposition; that quantity was not identified by the model and has been replaced by [dr_scale_attenuation](#).

Usage

```
dr_heterogeneity(object)
```

Arguments

object A fitted drmeta object.

Value

A list with components Q, df, p, I2, tau0sq, gamma, attenuation_observed, and tau2_by_study.

Examples

```
path <- system.file("extdata", "bcg_design_robustness.csv", package = "drmeta")
bcg <- utils::read.csv(path)
fit <- drmeta(yi = bcg[["yi"]], vi = bcg[["vi"]], dr = bcg[["dr"]])
dr_heterogeneity(fit)
```

dr_loo	<i>Leave-one-out influence diagnostics</i>
--------	--

Description

Refits the model with each study omitted in turn and records how the location and scale estimates move. Both components are reported, because a study can be uninfluential for the pooled mean while dominating the scale gradient.

Usage

```
dr_loo(object, slab = object$slab)
```

Arguments

object	A fitted drmeta object.
slab	Optional study labels. By default, labels stored in object\$slab are used; row-number labels are used for older objects without that component.

Value

A data frame with one row per omitted study, containing the study label, its design-robustness value, the leave-one-out estimate of the first location coefficient and its change from the full fit, the leave-one-out tau0sq and gamma with their changes, a converged flag, and a logical influential column flagging changes in the first location coefficient exceeding twice its full-model standard error.

Examples

```
path <- system.file("extdata", "bcg_design_robustness.csv", package = "drmeta")
bcg <- utils::read.csv(path)
fit <- drmeta(yi = bcg[["yi"]], vi = bcg[["vi"]], dr = bcg[["dr"]])
dr_loo(fit)
```

dr_plot_vfun	<i>Plot the fitted variance function</i>
--------------	--

Description

Draws the fitted residual between-study variance against the design index, with study-level fitted variances overlaid. By default the curve is drawn only over the observed range of the design index, since the fitted function outside that range is an extrapolation.

Usage

```
dr_plot_vfun(
  object,
  extrapolate = FALSE,
  col_curve = "#D6604D",
  col_pts = "#2166AC",
  xlab = "Design robustness",
  ylab = "Fitted tau^2",
  main = "Fitted variance function",
  ...
)
```

Arguments

object	A fitted drmeta object.
extrapolate	If TRUE, draw the curve over the whole interval $[0, 1]$ and shade the region beyond the observed support. Default FALSE.
col_curve	Colour of the fitted curve.
col_pts	Colour of the study-level points.
xlab, ylab, main	Axis labels and title.
...	Further arguments passed to plot .

Value

Invisibly, the data frame of plotted grid values.

Examples

```
path <- system.file("extdata", "bcg_design_robustness.csv", package = "drmeta")
bcg <- utils::read.csv(path)
fit <- drmeta(yi = bcg[["yi"]], vi = bcg[["vi"]], dr = bcg[["dr"]])
dr_plot_vfun(fit)
```

dr_scale_attenuation *Proportional attenuation in fitted heterogeneity*

Description

Returns the proportional reduction in fitted residual between-study heterogeneity between two design-robustness values, $A_r = 1 - \exp(-\gamma(d_H - d_L))$. This is a summary of the fitted scale function only. It is not a causal R^2 , not a proportion of heterogeneity explained, and not evidence that design quality caused a reduction in heterogeneity.

Usage

```
dr_scale_attenuation(object, d_low = min(object$dr), d_high = max(object$dr))
```

Arguments

object	A fitted drmeta object.
d_low	Lower DR value; defaults to the observed minimum.
d_high	Higher DR value; defaults to the observed maximum.

Value

A single numeric value. For a constrained fit it lies in $[0, 1]$; for an unrestricted fit with a negative gradient it is negative, indicating heterogeneity that increases with the design index.

Examples

```
path <- system.file("extdata", "bcg_design_robustness.csv", package = "drmeta")
bcg <- utils::read.csv(path)
fit <- drmeta(yi = bcg[["yi"]], vi = bcg[["vi"]], dr = bcg[["dr"]])
dr_scale_attenuation(fit)
dr_scale_attenuation(fit, d_low = 0.25, d_high = 0.75)
```

dr_scale_predict	<i>Predict the fitted between-study variance</i>
------------------	--

Description

Evaluates the fitted variance function $\tau^2(d) = \tau_0^2 \exp(-\gamma d)$ over a grid of design-robustness values. The default grid spans the observed range of the design index rather than the full interval $[0, 1]$, because values outside the observed support are extrapolations of the fitted scale model.

Usage

```
dr_scale_predict(
  object,
  dr = seq(min(object$dr), max(object$dr), length.out = 100)
)
```

Arguments

object	A fitted drmeta object.
dr	New design-robustness values in $[0, 1]$. Defaults to a grid spanning the observed range.

Value

A data frame with columns `dr` and `tau2`, giving the fitted residual between-study variance at each supplied value.

Examples

```
path <- system.file("extdata", "bcg_design_robustness.csv", package = "drmeta")
bcg <- utils::read.csv(path)
fit <- drmeta(yi = bcg[["yi"]], vi = bcg[["vi"]], dr = bcg[["dr"]])
head(dr_scale_predict(fit))
```

dr_score

*Build a design-robustness index from sub-scores***Description**

Combines user-supplied sub-score dimensions into a single index in $[0, 1]$ as a normalised weighted average. The components, their coding rules, and their weights should be prespecified and reported.

Usage

```
dr_score(..., weights = NULL, warn_range = TRUE)
```

Arguments

<code>...</code>	Named numeric vectors of equal length, each a sub-score in $[0, 1]$.
<code>weights</code>	Optional relative weights, one per sub-score. Defaults to equal weighting.
<code>warn_range</code>	If TRUE, warn when any value falls outside $[0, 1]$ before clipping.

Details

For confirmatory use, none of the sub-scores may be a function of the realized effect estimate, its standard error, its p-value, its confidence interval, or any outcome-dependent diagnostic. A score that uses such information defines an exploratory analysis, not a primary scale moderator. This function cannot verify that condition; it is the analyst's responsibility.

Value

A numeric vector in $[0, 1]$, carrying an attribute "subscores" that holds the clipped inputs and the resulting index.

Examples

```
dr_score(balance = c(0.9, 0.6, 0.4),
          overlap = c(0.8, 0.7, 0.5),
          weights = c(2, 1))
```

dr_shape_check

*Check whether a monotone scale model misses a grouped pattern***Description**

Fits a categorical scale model and a constant-scale model with `metafor::rma()`, using the location model already stored in a fitted `drmeta` object. This diagnostic is useful when the design score has a small number of ordered categories: an interior peak or trough cannot be represented by the package's monotone exponential variance function.

Usage

```
dr_shape_check(object, groups = object$dr, method = c("ML", "REML"))
```

Arguments

<code>object</code>	A fitted <code>drmeta</code> object.
<code>groups</code>	A vector defining scale groups. By default the observed design-robustness values are used. For an ordered interpretation, supply a factor whose levels are in the intended order.
<code>method</code>	Either "ML" or "REML". ML is the default for the likelihood-ratio comparison.

Details

The likelihood-ratio p-value is a descriptive model check. It uses the usual chi-square reference with one fewer degree of freedom than the number of groups. Estimates at or near zero can make that reference approximate. The diagnostic does not select a new primary score or alter the fitted `drmeta` model.

Value

An object of class `dr_shape_check`. It contains grouped variance estimates, the likelihood-ratio statistic and p-value, the inferred qualitative pattern, and the two fitted `metafor` models.

Examples

```
## Not run:
path <- system.file("extdata", "bcg_design_robustness.csv", package = "drmeta")
bcg <- utils::read.csv(path)
fit <- drmeta(bcg[["yi"]], bcg[["vi"]], bcg[["dr"]])
dr_shape_check(fit)

## End(Not run)
```

normalize_01

Rescale a numeric vector to the unit interval

Description

Rescale a numeric vector to the unit interval

Usage

```
normalize_01(x)
```

Arguments

x A numeric vector. Missing values are ignored during rescaling and preserved in the result.

Value

A numeric vector of the same length rescaled to $[0, 1]$. If all non-missing values are equal, a zero vector is returned.

Examples

```
normalize_01(c(2, 5, 8))
normalize_01(c(1, 1, 1))
```

```
print.drmeta_bootstrap_gamma
```

Print a parametric-bootstrap scale-gradient test

Description

Print a parametric-bootstrap scale-gradient test

Usage

```
## S3 method for class 'drmeta_bootstrap_gamma'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

Arguments

x An object of class `drmeta_bootstrap_gamma`.
digits Number of significant digits.
... Ignored.

Value

Invisibly returns `x`. Called for its printed output.

`print.dr_shape_check` *Print a grouped scale-shape diagnostic*

Description

Print a grouped scale-shape diagnostic.

Usage

```
## S3 method for class 'dr_shape_check'  
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

Arguments

<code>x</code>	A <code>dr_shape_check</code> object.
<code>digits</code>	Number of significant digits.
<code>...</code>	Ignored.

Value

Invisibly returns `x`.

Index

`coef.drmeta`, [2](#)
`confint.drmeta (coef.drmeta)`, [2](#)

`dr_from_design`, [6](#)
`dr_heterogeneity`, [7](#)
`dr_loo`, [4](#), [8](#)
`dr_plot_vfun`, [8](#)
`dr_scale_attenuation`, [7](#), [9](#)
`dr_scale_predict`, [10](#)
`dr_score`, [7](#), [11](#)
`dr_shape_check`, [12](#)
`drmeta`, [3](#)
`drmeta-methods (coef.drmeta)`, [2](#)
`drmeta_bootstrap_gamma`, [5](#)

`fitted.drmeta (coef.drmeta)`, [2](#)

`logLik.drmeta (coef.drmeta)`, [2](#)

`normalize_01`, [13](#)

`plot`, [9](#)
`print.dr_shape_check`, [14](#)
`print.drmeta (coef.drmeta)`, [2](#)
`print.drmeta_bootstrap_gamma`, [13](#)
`print.summary.drmeta (coef.drmeta)`, [2](#)

`residuals.drmeta (coef.drmeta)`, [2](#)

`summary.drmeta (coef.drmeta)`, [2](#)

`vcov.drmeta (coef.drmeta)`, [2](#)