Package ‘drmeta’

April 8, 2026

Title Design-Robust Meta-Analysis via Variance-Function Models

Version 0.1.0

Description Implements Design-Robust Meta-Analysis (DR-Meta), a variance-function random-effects framework in which between-study heterogeneity is modelled as a function of a study-level design robustness index, allowing heterogeneity to depend systematically on study quality or design strength rather than being treated as a single nuisance parameter. The package provides profiled restricted maximum likelihood (REML) estimation of the overall effect and variance-function parameters, study-specific weights, heterogeneity diagnostics (tau-squared, I-squared), influence and leave-one-out analysis, and graphical tools including forest plots and influence plots. The DR-Meta framework nests classical fixed-effects and standard random-effects meta-analysis as special cases, making it a strict generalisation of existing approaches.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports grDevices, graphics, stats, utils

Suggests ggplot2 (>= 3.4.0), knitr, metafor (>= 4.0.0), rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

URL https://github.com/causalfragility-lab/drmeta

BugReports https://github.com/causalfragility-lab/drmeta/issues

NeedsCompilation no

Author Subir Hait [aut, cre] (ORCID: <https://orcid.org/0009-0004-9871-9677>)

Maintainer Subir Hait <haitsubi@msu.edu>

Repository CRAN

Date/Publication 2026-04-08 18:40:03 UTC

Contents

`drmeta-package` *drmeta: Design-Robust Meta-Analysis*

Description

Fits the DR-Meta variance-function random-effects model (Hait, 2025), where between-study heterogeneity is a monotone-decreasing function of each study's design robustness index.

Author(s)

Maintainer: Subir Hait `<haitsubi@msu.edu>` (ORCID)

See Also

Useful links:

- `https://github.com/causalfragility-lab/drmeta`
- Report bugs at `https://github.com/causalfragility-lab/drmeta/issues`

`coef.drmeta` *Extract Coefficients from a drmeta Object*

Description

Returns a named numeric vector of the three model parameters: `mu`, `tau0sq`, and `gamma`.

Usage

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

Arguments

`object`	A fitted `"drmeta"` object.
`...`	Ignored.

Value

A named numeric vector of length 3 with the estimated model parameters: `mu` (pooled effect estimate), `tau0sq` (baseline between-study variance at DR = 0), and `gamma` (variance-function decay rate).

`confint.drmeta` *Confidence Interval for a drmeta Object*

Description

Returns a data frame with the estimate and 95% confidence interval for the pooled effect $\hat{\mu}$.

Usage

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

Arguments

`object`	A fitted `"drmeta"` object.
`parm`	Ignored (only `mu` is returned).
`level`	Confidence level. Default 0.95.
`...`	Ignored.

Value

A data frame with one row (`mu`) and three columns: `estimate` (the pooled effect $\hat{\mu}$), `lower`, and `upper` (confidence interval bounds at the requested `level`, default 95\

`drmeta` *Fit a Design-Robust Meta-Analysis (DR-Meta) Model*

Description

Fits a random-effects meta-analysis model in which between-study heterogeneity is a monotone-decreasing function of each study's design robustness index $\mathrm{DR}_i \in [0, 1]$. Studies with higher design robustness receive less heterogeneity weight, implementing Proposition 1 of Hait (2025).

Usage

```
drmeta(
  yi,
  vi,
  dr = NULL,
  vfun = c("exponential", "linear"),
  method = c("REML", "ML"),
  slab = NULL,
  control = list()
)
```

Arguments

`yi`	Numeric vector of k effect-size estimates.
`vi`	Numeric vector of k sampling variances (all must be > 0).
`dr`	Numeric vector of k design robustness indices in the interval $[0, 1]$. If `NULL`, a warning is issued and all studies are assigned $\mathrm{DR}_i = 0.5$.
`vfun`	Variance function: `"exponential"` (default) or `"linear"`.
`method`	Estimation method: `"REML"` (default) or `"ML"`.
`slab`	Optional character vector of study labels.
`control`	List of control arguments passed to `stats::optim`.

Value

An object of class `"drmeta"` (a named list). Key components: `mu` (pooled estimate), `se`, `ci.lb`, `ci.ub`, `zval`, `pval`, `tau0sq`, `gamma`, `tau2_i`, `sigma2_i`, `weights`, `loglik`, `reml_loglik`, `AIC`, `BIC`, `k`, `yi`, `vi`, `dr`, `slab`, `vfun`, `method`, `converged`, `optim_out`, `call`.

References

Hait, S. (2025). Design-Robust Meta-Analysis: A Variance-Function Framework for Causal Credibility.

See Also

`dr_heterogeneity`, `dr_loo`, `dr_pub_bias`, `dr_forest`, `dr_score`, `dr_from_design`

Examples

```
set.seed(42)
k  <- 20
dr <- runif(k, 0.1, 0.9)
vi <- runif(k, 0.01, 0.05)
tau2_true <- 0.04 * exp(-2 * dr)
yi <- rnorm(k, 0.3, sqrt(vi + tau2_true))

fit <- drmeta(yi, vi, dr)
print(fit)
summary(fit)
```

`dr_forest` *Forest Plot for DR-Meta*

Description

Draws a forest plot for a fitted `"drmeta"` model using base graphics. Studies are ordered by design robustness (largest DR at top by default). Point sizes are proportional to DR-Meta weights; a vertical reference line and summary diamond are included.

Usage

```
dr_forest(
  object,
  order_by = c("dr", "yi", "none"),
  xlab = "Effect size",
  main = "DR-Meta Forest Plot",
  col_point = "#2166AC",
  col_diamond = "#D6604D",
  col_dr = "#4DAC26",
  show_dr = TRUE,
  xlim = NULL,
  cex_study = 0.8,
  ...
)
```

Arguments

`object`	A fitted `"drmeta"` object.
`order_by`	Character: `"dr"` (default, sort by DR_i descending), `"yi"` (sort by effect size), or `"none"` (original order).
`xlab`	X-axis label. Default `"Effect size"`.
`main`	Plot title.
`col_point`	Colour for study-level estimate points. Default `"#2166AC"`.

`col_diamond`	Colour for the summary diamond. Default `"#D6604D"`.
`col_dr`	Colour for the DR bar on the left. Default `"#4DAC26"`.
`show_dr`	Logical. If `TRUE` (default), displays a coloured DR bar indicating design robustness strength.
`xlim`	Numeric vector of length 2 for x-axis limits. If `NULL`, computed automatically.
`cex_study`	Scaling factor for study labels. Default 0.8.
`...`	Further graphical arguments (ignored).

Value

Invisibly returns the data frame used for plotting (ordered studies).

Examples

```
set.seed(42)
k <- 10
dr <- runif(k, 0.1, 0.9)
vi <- runif(k, 0.01, 0.05)
yi <- rnorm(k, 0.3, sqrt(vi + 0.04 * exp(-2 * dr)))
fit <- drmeta(yi, vi, dr)
dr_forest(fit)
```

`dr_from_design` *Design Robustness from Study Design Type*

Description

Maps a vector of study design type labels to a numeric design robustness score in $[0, 1]$, using a pre-specified hierarchy of causal credibility. This is a convenient starting point for operationalising the DR index when only design type is available.

Usage

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

Arguments

`design`	Character vector of design type labels (case-insensitive).
`custom_map`	Optional named numeric vector to override or add design types, e.g. `c(my_design = 0.65)`.
`default_score`	Numeric score assigned to unrecognised design labels. Default is 0.25 (conservative).
`warn_unknown`	Logical. If `TRUE` (default), warns about unrecognised labels.

Details

The default hierarchy follows the causal inference literature (Rubin, 2008; Rosenbaum, 2010; Imbens & Rubin, 2015):

Design type label	Default DR score
`"rct"`	1.00
`"rd"`, `"rdd"`	0.75
`"iv"`	0.75
`"did"`, `"diff_in_diff"`	0.60
`"matching"`, `"psm"`	0.50
`"ipw"`, `"propensity"`	0.45
`"regression"`, `"ols"`	0.25
`"cross_section"`	0.20
`"descriptive"`	0.10

Users can override or extend this table via the `custom_map` argument.

Value

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

See Also

`dr_score`, `drmeta`

Examples

```
designs <- c("RCT", "DiD", "OLS", "IV", "matching", "unknown_design")
dr_from_design(designs)

# Custom override
dr_from_design(designs, custom_map = c(unknown_design = 0.35))
```

`dr_funnel` *Funnel Plot for DR-Meta*

Description

Creates a funnel plot for a fitted `"drmeta"` model. The horizontal axis shows effect-size estimates; the vertical axis shows standard errors. Point sizes are proportional to DR-Meta weights and point colour encodes design robustness.

Usage

```
dr_funnel(
  object,
  contours = TRUE,
  xlab = "Effect size",
  ylab = "Standard error",
  main = "DR-Meta Funnel Plot",
  col_low = "#D6604D",
  col_high = "#2166AC"
)
```

Arguments

`object`	A fitted `"drmeta"` object.
`contours`	Logical. If `TRUE` (default), adds 95\ contours around the pooled estimate.
`xlab`	X-axis label.
`ylab`	Y-axis label (default: reversed SE axis).
`main`	Plot title.
`col_low`	Colour for low-DR studies. Default `"#D6604D"`.
`col_high`	Colour for high-DR studies. Default `"#2166AC"`.

Value

Invisibly returns `NULL`.

Examples

```
set.seed(42)
k <- 15
dr <- runif(k)
vi <- runif(k, 0.01, 0.06)
yi <- rnorm(k, 0.3, sqrt(vi + 0.04 * exp(-1.5 * dr)))
fit <- drmeta(yi, vi, dr)
dr_funnel(fit)
```

`dr_heterogeneity` *Heterogeneity Diagnostics for DR-Meta*

Description

Computes a suite of heterogeneity statistics for a fitted `"drmeta"` model, including Cochran's Q (with DR-Meta weights), I^2, H^2, the design-residual variance decomposition of Proposition 6 (Hait, 2025), and per-study contributions to Q.

Usage

```
dr_heterogeneity(object)
```

Arguments

`object`	A fitted `"drmeta"` object from `drmeta`.

Value

A list with three elements:

`summary` A one-row data frame with: `k`, `Q`, `df`, `pval`, `tau2_mean` (mean design-specific heterogeneity), `I2`, `H2`.

`decomposition` A one-row data frame with the Proposition 6 decomposition: `tau2_design_explained`, `tau2_residual`, `tau2_total`, `R2_DR` (proportion explained by design).

`contributions` A data frame with per-study Q contributions: `study`, `DR`, `tau2_i`, `q_i`, `pct_Q`.

Design-Residual Decomposition (Proposition 6)

The total between-study heterogeneity can be decomposed as:

$$\tau^2_{\text{total}} = \mathbb{E}[\tau^2(\mathrm{DR}_i)] + \mathrm{Var}(u_i \mid \mathrm{DR}_i),$$

where the first term is the **design-explained** heterogeneity (captured by the variance function) and the second is the **design-residual** heterogeneity (unexplained by DR). This decomposition is analogous to R-squared in meta-regression.

The design-explained proportion is

$$R^2_{\mathrm{DR}} = \frac{\mathbb{E}[\tau^2(\mathrm{DR}_i)]}{\tau^2_{\text{total}}}.$$

References

Hait, S. (2025). *Design-Robust Meta-Analysis: A Variance-Function Framework for Causal Credibility*. Proposition 6.

Examples

```
set.seed(42)
k <- 15
dr <- runif(k, 0.1, 0.9)
vi <- runif(k, 0.01, 0.05)
tau2_true <- 0.04 * exp(-2 * dr)
yi <- rnorm(k, 0.3, sqrt(vi + tau2_true))

fit <- drmeta(yi, vi, dr)
het <- dr_heterogeneity(fit)
het$summary
het$decomposition
het$contributions
```

`dr_loo` *Leave-One-Out Influence Diagnostics for DR-Meta*

Description

For each study, re-fits the DR-Meta model after excluding that study and records how the pooled estimate, confidence interval, variance-function parameters, and heterogeneity change. Studies with large absolute $\Delta\hat{\mu}$ or large shifts in $\hat{\tau}_0^2/\hat{\gamma}$ are considered influential.

Usage

```
dr_loo(object, parallel = FALSE, mc.cores = NULL)
```

Arguments

`object`	A fitted `"drmeta"` object from `drmeta`.
`parallel`	Logical. If `TRUE` and the **parallel** package is available, uses `parallel::mclapply` for the LOO loop (Unix/macOS only). Default `FALSE`.
`mc.cores`	Integer. Number of cores for parallel execution. Default is `parallel::detectCores() - 1`.

Value

A list with:

`summary` A data frame with one row per study and columns: `study`, `DR`, `est_loo` (LOO pooled estimate), `ci.lb_loo`, `ci.ub_loo`, `tau0sq_loo`, `gamma_loo`, `delta_mu` (change in estimate), `delta_tau0sq`, `delta_gamma`, `influential` (logical: |delta_mu| > 2 * SE of full model).

`full` The original full-model `"drmeta"` object.

Examples

```
set.seed(7)
k  <- 12
dr <- runif(k, 0.1, 0.9)
vi <- runif(k, 0.01, 0.05)
tau2_true <- 0.04 * exp(-2 * dr)
yi <- rnorm(k, 0.3, sqrt(vi + tau2_true))

fit <- drmeta(yi, vi, dr)
loo <- dr_loo(fit)
loo$summary
```

`dr_plot` *Weight Diagnostic Plot for DR-Meta*

Description

Plots DR-Meta study weights against design robustness (DR_i), illustrating the monotone ordering of Lemma 3 (Hait, 2025). An overlaid curve shows the theoretical weight function holding sampling variance at its median value.

Usage

```
dr_plot(
  object,
  col_pts = "#2166AC",
  col_curve = "#D6604D",
  xlab = expression(paste("Design robustness (", DR[i], ")")),
  ylab = "DR-Meta weight (normalised, %)",
  main = "DR-Meta Weight vs Design Robustness",
  show_labels = TRUE,
  ...
)
```

Arguments

`object`	A fitted `"drmeta"` object.
`col_pts`	Colour for individual study points. Default `"#2166AC"`.
`col_curve`	Colour for the theoretical weight curve. Default `"#D6604D"`.
`xlab`	X-axis label.
`ylab`	Y-axis label.
`main`	Plot title.
`show_labels`	Logical. If `TRUE` (default), labels outlier points.
`...`	Additional graphical arguments passed to `plot`.

Value

Invisibly returns a data frame with study-level weight information.

Examples

```
set.seed(42)
k <- 12
dr <- runif(k, 0.1, 0.9)
vi <- runif(k, 0.01, 0.05)
yi <- rnorm(k, 0.3, sqrt(vi + 0.04 * exp(-2 * dr)))
fit <- drmeta(yi, vi, dr)
dr_plot(fit)
```

`dr_plot_vfun` *Variance Function Plot for DR-Meta*

Description

Plots the fitted variance function $\tau^2(\mathrm{DR};\,\hat{\psi})$ as a curve from DR = 0 to DR = 1, with study-level $\hat{\tau}_i^2$ overlaid as points.

Usage

```
dr_plot_vfun(
  object,
  col_curve = "#D6604D",
  col_pts = "#2166AC",
  xlab = expression(paste("Design robustness (", DR[i], ")")),
  ylab = expression(paste("Between-study variance ", tau^2)),
  main = "DR-Meta Variance Function"
)
```

Arguments

`object`	A fitted `"drmeta"` object.
`col_curve`	Line colour for variance function. Default `"#D6604D"`.
`col_pts`	Point colour for study-level $\hat{\tau}_i^2$. Default `"#2166AC"`.
`xlab`	X-axis label.
`ylab`	Y-axis label.
`main`	Plot title.

Value

Invisibly returns a data frame with the plotting grid.

Examples

```
set.seed(1)
fit <- drmeta(rnorm(12, 0.3), runif(12, 0.01, 0.04), runif(12))
dr_plot_vfun(fit)
```

`dr_pub_bias` *Publication Bias Assessment for DR-Meta (PET/PEESE/Egger/Funnel)*

Description

Performs a suite of publication-bias and small-study-effects tests adapted for DR-Meta weights. PET and PEESE regressions use $w_i = 1/\hat{\sigma}_i^2 = 1/(v_i + \hat{\tau}^2(\mathrm{DR}_i))$ as regression weights, so precision is design-adjusted.

Usage

```
dr_pub_bias(object, test = c("PET", "PEESE", "Egger"), alpha = 0.05)
```

Arguments

`object`	A fitted `"drmeta"` object from `drmeta`.
`test`	Character vector of tests to run; any subset of `c("PET", "PEESE", "Egger")`. Default: all three.
`alpha`	Significance level for PET/PEESE intercept test. Default 0.05.

Details

PET (Precision Effect Test): regresses y_i on $se_i = \sqrt{v_i}$, with DR-Meta weights. A significant slope implies small-study bias; the intercept estimates the publication-bias-corrected effect.

PEESE (Precision Effect Estimate with Standard Error): regresses y_i on v_i, with DR-Meta weights. Generally preferred when the true effect is non-zero.

Egger test: Egger-type regression using the standardised effect y_i/se_i on $1/se_i$, with DR-Meta weights.

Funnel asymmetry: classic funnel plot with DR-Meta summary.

Value

A list with elements:

`PET` Summary of PET regression (data frame with intercept, slope, SE, z, p).

`PEESE` Summary of PEESE regression.

`Egger` Summary of Egger regression.

`recommendation` Character string: use PET intercept if PET slope is significant and effect is small; use PEESE intercept otherwise.

Examples

```
set.seed(99)
k  <- 20
dr <- runif(k, 0.1, 0.9)
vi <- runif(k, 0.005, 0.08)
# Introduce small-study effect: smaller studies overestimate
yi <- rnorm(k, 0.3 + 0.5 * sqrt(vi), sqrt(vi + 0.03 * exp(-1.5 * dr)))
fit <- drmeta(yi, vi, dr)
pb  <- dr_pub_bias(fit)
pb$PET
pb$PEESE
pb$recommendation
```

`dr_score` *Construct a Design Robustness Index (DR_i)*

Description

Computes a scalar design robustness index $\mathrm{DR}_i \in [0,1]$ for each study by forming a weighted composite of user-supplied sub-scores. This is the recommended way to operationalise the DR index described in Section 3.1 of Hait (2025).

Usage

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

Arguments

`...`	Named numeric vectors, each of length k (number of studies), representing individual sub-score dimensions. Each element must lie in $[0,1]$. Names are used in the returned data frame.
`weights`	Optional numeric vector of the same length as the number of sub-score arguments, giving the relative importance of each dimension. Defaults to equal weighting.
`warn_range`	Logical. If `TRUE` (default), warns when any input value lies outside $[0,1]$ before clipping.

Details

Sub-scores are first clipped to $[0,1]$ and then combined as a normalised weighted average:

$$\mathrm{DR}_i = \sum_j \tilde{w}_j \, s_{ij}, \qquad \tilde{w}_j = w_j / \sum_j w_j.$$

The result is therefore guaranteed to lie in $[0,1]$.

Typical sub-score dimensions for quasi-experimental studies include:

- **Balance**: covariate balance between treatment and control (e.g., standardised mean difference < 0.1 scores 1.0).
- **Overlap**: common-support / propensity-score overlap.
- **Design**: study design type — see `dr_from_design`.
- **Transparency**: pre-registration, data/code availability.

Value

A numeric vector of length k with values in $[0, 1]$. The vector carries an attribute `"subscores"` containing a data frame of the clipped sub-scores and the final DR index.

See Also

`dr_from_design`, `drmeta`

Examples

```
k <- 5
balance  <- c(0.9, 0.6, 0.4, 0.8, 0.3)
overlap  <- c(0.8, 0.7, 0.5, 0.9, 0.4)
design   <- c(1.0, 0.5, 0.5, 0.75, 0.25)

# Equal weights
dr <- dr_score(balance = balance, overlap = overlap, design = design)
dr

# Down-weight transparency
transp <- c(1, 0, 0, 1, 0)
dr_w <- dr_score(balance = balance, overlap = overlap,
                 design = design, transparency = transp,
                 weights = c(2, 2, 3, 1))
dr_w
```

`dr_variance` *Evaluate the DR-Meta Variance Function*

Description

Evaluates $\tau^2(\mathrm{DR}; \psi)$ for a grid of DR values or for the study-level DR indices from a fitted `"drmeta"` model. Useful for visualising how heterogeneity varies with design robustness.

Usage

```
dr_variance(dr, tau0sq = NULL, gamma = NULL, vfun = c("exponential", "linear"))
```

Arguments

`dr`	Numeric vector of design robustness values in $[0, 1]$, or a fitted `"drmeta"` object (in which case `tau0sq`, `gamma`, and `vfun` are extracted automatically and `dr` from the model is used).
`tau0sq`	Scalar $\hat{\tau}_0^2$. Ignored if `dr` is a `"drmeta"` object.
`gamma`	Scalar $\hat{\gamma}$. Ignored if `dr` is a `"drmeta"` object.
`vfun`	Variance function: `"exponential"` or `"linear"`. Ignored if `dr` is a `"drmeta"` object.

Value

A numeric vector of $\tau^2(\mathrm{DR}_i)$ values.

See Also

`drmeta`, `dr_weights`

Examples

```
# Evaluate on a grid
grid <- seq(0, 1, by = 0.1)
tau2 <- dr_variance(grid, tau0sq = 0.04, gamma = 1.5)
plot(grid, tau2, type = "l", xlab = "DR", ylab = expression(tau^2))

# Extract from a fitted model
set.seed(1)
fit <- drmeta(yi = rnorm(10, 0.3), vi = runif(10, 0.01, 0.05),
              dr = runif(10))
dr_variance(fit)
```

`dr_weights` *Compute DR-Meta Study Weights*

Description

Returns the DR-Meta inverse-total-variance weights

$$w_i = \frac{1}{v_i + \hat{\tau}^2(\mathrm{DR}_i)},$$

given estimated variance-function parameters. Optionally normalises weights to sum to 1 or to 100.

Usage

```
dr_weights(
  vi,
  dr,
  tau0sq,
  gamma,
  vfun = c("exponential", "linear"),
  normalise = c("none", "sum1", "pct")
)
```

Arguments

`vi`	Numeric vector of sampling variances ($v_i > 0$).
`dr`	Numeric vector of design robustness indices in $[0, 1]$.
`tau0sq`	Non-negative scalar: estimated baseline heterogeneity $\hat{\tau}_0^2$.
`gamma`	Non-negative scalar: estimated variance-function slope $\hat{\gamma}$.
`vfun`	Variance function: `"exponential"` (default) or `"linear"`.
`normalise`	Character: `"none"` (raw weights, default), `"sum1"` (divide by sum so weights sum to 1), or `"pct"` (multiply by 100 after normalising to sum to 1).

Details

This function is primarily a utility for diagnostics and visualisation; weights are also returned as part of the `"drmeta"` object produced by `drmeta`.

Value

A numeric vector of weights, the same length as `vi`.

See Also

`drmeta`, `dr_variance`

Examples

```
vi  <- c(0.02, 0.03, 0.015, 0.025, 0.01)
dr  <- c(0.9,  0.4,  0.7,   0.2,   1.0)
dr_weights(vi, dr, tau0sq = 0.04, gamma = 1.5)
dr_weights(vi, dr, tau0sq = 0.04, gamma = 1.5, normalise = "pct")
```

`fitted.drmeta` *Fitted Values for a drmeta Object*

Description

Returns a vector of length k where every element equals the pooled estimate $\hat{\mu}$ (the model has a single intercept, so all fitted values are identical).

Usage

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

Arguments

`object`	A fitted `"drmeta"` object.
`...`	Ignored.

Value

A numeric vector of length k where every element equals the pooled estimate $\hat{\mu}$. Because DR-Meta has a single intercept, all studies share the same fitted value.

`logLik.drmeta` *Log-Likelihood for a drmeta Object*

Description

Log-Likelihood for a drmeta Object

Usage

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

Arguments

`object`	A fitted `"drmeta"` object.
`REML`	Logical. If `TRUE`, returns the REML log-likelihood. Default `FALSE` (ML).
`...`	Ignored.

Value

An object of class `"logLik"`. The numeric value is the maximised log-likelihood (ML or REML, depending on `REML`). The object carries two attributes: `df` (number of parameters, always 3: `mu`, `tau0sq`, `gamma`) and `nobs` (number of studies k).

`normalize_01` *Normalise a Numeric Vector to the $[0, 1]$ Interval*

Description

Linearly rescales a numeric vector to the $[0, 1]$ interval. Useful for standardising individual subscore components before aggregation with `dr_score`.

Usage

```
normalize_01(x)
```

Arguments

`x`	A numeric vector. `NA` values are ignored during rescaling.

Value

A numeric vector rescaled to $[0, 1]$. If all non-missing values are equal, returns a zero vector (to avoid division by zero).

Examples

```
normalize_01(c(2, 5, 8))   # returns c(0, 0.5, 1)
normalize_01(c(1, 1, 1))   # returns c(0, 0, 0)
```

`print.drmeta` *Print Method for drmeta Objects*

Description

Print Method for drmeta Objects

Usage

```
## S3 method for class 'drmeta'
print(x, digits = 4, ...)
```

Arguments

`x`	A fitted `"drmeta"` object.
`digits`	Number of significant digits. Default 4.
`...`	Ignored.

Value

Invisibly returns the original `drmeta` object `x`, unchanged. This function is called for its side effect of printing a formatted summary of the fitted DR-Meta model to the console.

`residuals.drmeta` *Residuals for a drmeta Object*

Description

Residuals for a drmeta Object

Usage

```
## S3 method for class 'drmeta'
residuals(object, type = c("raw", "standardised"), ...)
```

Arguments

`object`	A fitted `"drmeta"` object.
`type`	`"raw"` (default) or `"standardised"`.
`...`	Ignored.

Value

A numeric vector of length k of residuals. When `type = "raw"` (default), returns observed minus fitted values ($y_i - \hat{\mu}$). When `type = "standardised"`, each residual is divided by $\sqrt{\hat{\sigma}_i^2}$ (the square root of the total study variance under the fitted model).

`summary.drmeta` *Summary Method for drmeta Objects*

Description

Summary Method for drmeta Objects

Usage

```
## S3 method for class 'drmeta'
summary(object, digits = 4, ...)
```

Arguments

`object`	A fitted `"drmeta"` object.
`digits`	Number of significant digits. Default 4.
`...`	Ignored.

Value

Invisibly returns the fitted `drmeta` object `object`, unchanged. Called for its side effect of printing a detailed formatted summary — including the pooled estimate, confidence interval, z-test, variance-function parameters, and model fit statistics — to the console.

Index