

Package: orchaRd (via r-universe)

July 25, 2026

Title Visualizing Meta-Analyses with Orchard Plots and Prediction Intervals

Version 2.2.1

Description Generates prediction intervals and orchard plots for meta-analytic and meta-regression models fitted with the 'metafor' package. Orchard plots augment classic forest plots by displaying individual effect sizes together with group means and their confidence and prediction intervals, providing an enhanced visualization of meta-analytic data for ecology, evolution, and beyond. Methods are described in Nakagawa et al. (2023) <[doi:10.1111/2041-210X.14152](https://doi.org/10.1111/2041-210X.14152)>.

Depends R (>= 4.2)

URL <https://daniel1noble.github.io/orchaRd/>

BugReports <https://github.com/daniel1noble/orchaRd/issues>

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Imports ape, dplyr, emmeans, ggbeeswarm, ggplot2, latex2exp, magrittr, MASS, metafor, progress, rWishart, stats, tester, utils

Suggests broom, glmmTMB, gt, knitr, lme4, metadat, moments, patchwork, PearsonDS, purrr, R.rsp, rmarkdown, testthat, tibble

Repository <https://daniel1noble.r-universe.dev>

Date/Publication 2026-07-24 23:52:20 UTC

RemoteUrl <https://github.com/daniel1noble/orchard>

RemoteRef HEAD

RemoteSha e2dccdb385d74926d193196a3e2f710007fde22c

Contents

.calc.kurtosis	3
.calc.skewness	3
.colour_blind_palette	4
.lnM	4
.MSb	5
.MSw	5
.r.to.zr	6
.safe_lnM_dep	6
.safe_lnM_indep	7
.sv_jack	8
.v_lnM	9
.zr.variance	9
bubble_plot	10
caterpillars	13
cor_diff	15
cvh1_ml	16
cvh2_ml	17
eklof	18
english	19
firstup	20
fish	20
geom_pub_stats_naka	21
geom_pub_stats_yang	22
get_data_raw	23
get_data_raw_cont	24
get_ints_dat	25
glmmTMB_to_rma	25
i2_ml	27
leave_one_out	28
lim	29
m1_ml	30
m2_ml	31
magnitude_effects	32
matrix_i2	34
ml_cvh1	34
ml_cvh2	35
ml_m1	35
ml_m2	36
mod_results	36
moment_effects	39
num_studies	40
orchard_leave1out	41
orchard_plot	43
phylo_matrix	46
pottier	46
pred_interval_esmeans	50

<i>.calc.kurtosis</i>	3
-----------------------	---

print.orchard	50
pub_bias_plot	51
R2_calc	53
r2_ml	54
ratio_i2	55
submerge	55
VCV_dARR	56
weighted_var	56
Zr_to_r	57

Index	58
--------------	-----------

<i>.calc.kurtosis</i>	<i>.calc.kurtosis</i>
-----------------------	-----------------------

Description

Computes the excess kurtosis of a numeric vector using small sample correction.

Usage

`.calc.kurtosis(x)`

Arguments

x A numeric vector.

Value

A numeric value representing the kurtosis estimate or variance.

<i>.calc.skewness</i>	<i>.calc.skewness</i>
-----------------------	-----------------------

Description

Computes the skewness of a numeric vector using small sample correction.

Usage

`.calc.skewness(x)`

Arguments

x A numeric vector.

Value

A numeric value representing the skewness estimate or variance.

<code>.colour_blind_palette</code>	<i>Colour blind palette</i>
------------------------------------	-----------------------------

Description

Colour blind palette

Usage

`.colour_blind_palette`

Format

An object of class character of length 20.

<code>.lnM</code>	<i>.lnM</i>
-------------------	-------------

Description

Computes the natural logarithm of the ratio of between and within group variances as a measure of the magnitude of difference.

Usage

`.lnM(x1, x2, sd1, sd2, n1, n2)`

Arguments

- | | |
|-----|--------------------------------|
| x1 | Mean of group 1. |
| x2 | Mean of group 2. |
| sd1 | Standard deviation of group 1. |
| sd2 | Standard deviation of group 2. |
| n1 | Sample size of group 1. |
| n2 | Sample size of group 2. |

Value

The natural logarithm of the ratio of between and within group variances.

.MSb

*.MSb***Description**

Computes the between group mean-squares difference.

Usage

```
.MSb(x1, x2, n1, n2, paired = FALSE)
```

Arguments

x1	Mean of group 1.
x2	Mean of group 2.
n1	Sample size of group 1.
n2	Sample size of group 2.
paired	Logical, whether the samples are paired. Default is FALSE.

Value

The between group mean-squares difference.

.MSw

*.MSw***Description**

Computes the within group mean-squares difference.

Usage

```
.MSw(sd1, sd2, n1, n2, paired = FALSE)
```

Arguments

sd1	Standard deviation of group 1.
sd2	Standard deviation of group 2.
n1	Sample size of group 1.
n2	Sample size of group 2.
paired	Logical, whether the samples are paired. Default is FALSE.

Value

The within group mean-squares difference.

`.r.to.zr`*Fisher Z-transformation of Correlation Coefficient*

Description

Converts a Pearson correlation coefficient r to Fisher's Z_r using the inverse hyperbolic tangent.

Usage

```
.r.to.zr(r)
```

Arguments

`r` A numeric vector of Pearson correlation coefficients.

Value

The Z_r -transformed correlation value(s).

`.safe_lnM_dep`*SAFE bootstrap : dependent samples*

Description

Computes a SAFE bootstrap estimate of the effect size and its variance for dependent samples.

Usage

```
.safe_lnM_dep(  
  x1bar,  
  x2bar,  
  sd1,  
  sd2,  
  n,  
  r,  
  min_kept = 2000,  
  chunk_init = 4000,  
  chunk_max = 2e+06,  
  max_draws = Inf,  
  patience_noaccept = 5,  
  seed = 565  
)
```

Arguments

<code>x1bar</code>	Mean of group 1.
<code>x2bar</code>	Mean of group 2.
<code>sd1</code>	Standard deviation of group 1.
<code>sd2</code>	Standard deviation of group 2.
<code>n</code>	Sample size (assumed equal for both groups).
<code>r</code>	Correlation between the two groups.
<code>min_kept</code>	Minimum number of accepted bootstrap samples to keep. Default is 2000.
<code>chunk_init</code>	Initial chunk size for bootstrap sampling. Default is 4000.
<code>chunk_max</code>	Maximum chunk size for bootstrap sampling. Default is 2e6.
<code>max_draws</code>	Maximum total number of bootstrap draws. Default is Inf.
<code>patience_noaccept</code>	Number of consecutive chunks with no accepted samples before stopping. Default is 5.
<code>seed</code>	Random seed for reproducibility. Default is 565.

Value

A list containing the point estimate, variance, number of kept samples, total draws, number of attempts, and status.

<code>.safe_lnm_indep</code>	<i>SAFE bootstrap : independent samples</i>
------------------------------	---

Description

Computes a SAFE bootstrap estimate of the effect size and its variance for independent samples.

Usage

```
.safe_lnm_indep(
  x1bar,
  x2bar,
  sd1,
  sd2,
  n1,
  n2,
  min_kept = 2000,
  chunk_init = 4000,
  chunk_max = 2e+06,
  max_draws = Inf,
  patience_noaccept = 5,
  seed = 565
)
```

Arguments

x1bar	Mean of group 1.
x2bar	Mean of group 2.
sd1	Standard deviation of group 1.
sd2	Standard deviation of group 2.
n1	Sample size of group 1.
n2	Sample size of group 2.
min_kept	Minimum number of accepted bootstrap samples to keep. Default is 2000.
chunk_init	Initial chunk size for bootstrap sampling. Default is 4000.
chunk_max	Maximum chunk size for bootstrap sampling. Default is 2e6.
max_draws	Maximum total number of bootstrap draws. Default is Inf.
patience_noaccept	Number of consecutive chunks with no accepted samples before stopping. Default is 5.
seed	Random seed for reproducibility. Default is 565.

Value

A list containing the point estimate, variance, number of kept samples, total draws, number of attempts, and status.

.sv_jack	<i>Jackknife Estimation of Skewness and Kurtosis for computing sampling variances</i>
----------	---

Description

Computes jackknife estimate and standard error for skewness and kurtosis.

Usage

```
.sv_jack(
  x,
  bias.correct = TRUE,
  return.replicates = FALSE,
  type = c("skew", "kurt")
)
```

Arguments

x	A numeric vector.
bias.correct	Logical, whether to apply bias correction. Default is TRUE.
return.replicates	Logical, whether to return jackknife replicates. Default is FALSE.
type	Character, either "skew" or "kurt" to specify the type of statistic.

Value

A list with estimate ('est'), bias-corrected estimate ('est_bc'), variance, and standard error.

`.v_lnM``.v_lnM`

Description

Computes the sampling variance of the natural logarithm of the ratio of between and within group variances.

Usage

```
.v_lnM(x1, x2, sd1, sd2, n1, n2, r = NULL)
```

Arguments

<code>x1</code>	Mean of group 1.
<code>x2</code>	Mean of group 2.
<code>sd1</code>	Standard deviation of group 1.
<code>sd2</code>	Standard deviation of group 2.
<code>n1</code>	Sample size of group 1.
<code>n2</code>	Sample size of group 2.
<code>r</code>	Optional correlation between groups for dependent samples. Default is NULL for independent samples.

Value

The sampling variance of the natural logarithm of the ratio of between and within group variances.

`.zr.variance`*Variance of Fisher Z-transformed Correlation*

Description

Computes the approximate sampling variance of Fisher's Zr given a sample size.

Usage

```
.zr.variance(n)
```

Arguments

<code>n</code>	Sample size (must be greater than 3).
----------------	---------------------------------------

Value

The variance of the Zr-transformed correlation.

bubble_plot	<i>bubble_plot</i>
-------------	--------------------

Description

Using a **metafor** model object of class `rma` or `rma.mv`, a results table of class `orchard`, or a raw `data.frame`, the `bubble_plot` function creates a bubble plot from slope estimates or raw effect sizes. When a raw `data.frame` is provided, only the data points are plotted (no model fit lines). In cases when a model includes interaction terms, this function creates panels of bubble plots.

Usage

```
bubble_plot(
  object,
  mod,
  group = NULL,
  xlab = "Moderator",
  ylab = "Effect size",
  N = NULL,
  alpha = 0.5,
  cb = TRUE,
  k = TRUE,
  g = FALSE,
  transfm = c("none", "tanh", "invlogit", "percent", "percentr"),
  n_transfm = NULL,
  est.lwd = 1,
  ci.lwd = 0.5,
  pi.lwd = 0.5,
  est.col = "black",
  ci.col = "black",
  pi.col = "black",
  point.size = c(1, 3.5),
  legend.pos = c("top.left", "top.right", "bottom.right", "bottom.left", "top.out",
    "bottom.out", "none"),
  k.pos = c("top.right", "top.left", "bottom.right", "bottom.left", "none"),
  condition.nrow = 2,
  condition.order = NULL,
  weights = "prop",
  by = NULL,
  at = NULL,
  yi = NULL,
  vi = NULL,
  stdy = NULL
)
```

Arguments

object	Model object of class <code>rma</code> , <code>rma.mv</code> , or <code>orchar</code> table of model results, or a <code>data.frame</code> containing raw effect sizes.
mod	The name of a continuous moderator, to be plotted on the x-axis of the bubble plot.
group	The grouping variable that one wishes to plot beside total effect sizes, <code>k</code> . This could be study, species, or any grouping variable one wishes to present sample sizes for. Not needed if an <code>orchar_plot</code> is provided with a <code>mod_results</code> object of class <code>orchar</code> .
xlab	Moderator label.
ylab	Effect size measure label.
N	The vector of sample size which an effect size is based on. Defaults to precision (the inverse of sampling standard error).
alpha	The level of transparency for pieces of fruit (effect size).
cb	If TRUE, it uses a colourblind-friendly palette of 20 colours (do not make this TRUE, when <code>colour = TRUE</code>).
k	If TRUE, it displays <code>k</code> (number of effect sizes) on the plot.
g	If TRUE, it displays <code>g</code> (number of grouping levels for each level of the moderator) on the plot.
transfm	If set to "tanh", a tanh transformation will be applied to effect sizes, converting <code>Zr</code> to a correlation or pulling in extreme values for other effect sizes (lnRR, lnCVR, SMD). "invlogit" can be used to convert lnRR to the inverse logit scale. "percentr" can convert to the percentage change scale when using response ratios and "percent" can convert to the percentage change scale of an log transformed effect size. Defaults to "none".
n_transfm	The vector of sample sizes for each effect size estimate. This is used when <code>transfm = "inv_ft"</code> . Defaults to NULL.
est.lwd	Size of the point estimate.
ci.lwd	Size of the confidence interval.
pi.lwd	Size of the prediction interval.
est.col	Colour of the point estimate.
ci.col	Colour of the confidence interval.
pi.col	Colour of the prediction interval.
point.size	Numeric vector of length 2, specifying the minimum and maximum point sizes for effect size bubbles. Defaults to <code>c(1, 3.5)</code> . Useful for controlling bubble size in small figures.
legend.pos	Where to place the legend, or not to include a legend ("none").
k.pos	The position of effect size number, <code>k</code> .
condition.nrow	Number of rows to plot condition variable.
condition.order	Order of the levels of the condition variable in the plot. Defaults to NULL.

weights	How to marginalize categorical variables; used when one wants marginalised means. The default is <code>weights = "prop"</code> , which weights means for moderator levels based on their proportional representation in the data. For example, if "sex" is a moderator, and males have a larger sample size than females, then this will produce a weighted average, where males are weighted more towards the mean than females. This may not always be ideal when, for example, males and females are typically roughly equally prevalent in a population. In cases such as these, you can give the moderator levels equal weight using <code>weights = "equal"</code> .
by	Character vector indicating the name that predictions should be conditioned on for the levels of the moderator.
at	List of levels one wishes to predict at for the corresponding variables in <code>by</code> . Used when one wants marginalised means. This argument can also be used to suppress levels of the moderator when argument <code>subset = TRUE</code> . Provide a list as follows: <code>list(mod = c("level1", "level2"))</code> .
yi	Character string. The name of the effect size column in the data.frame. Only used when object is a data.frame.
vi	Character string. The name of the sampling variance column in the data.frame. Only used when object is a data.frame.
stdy	Character string. The name of the study identifier column in the data.frame, used for computing k and g labels. Only used when object is a data.frame.

Value

Bubble plot

Author(s)

Shinichi Nakagawa - s.nakagawa@unsw.edu.au

Daniel Noble - daniel.noble@anu.edu.au

Examples

```
data(lim)
lim[, "year"] <- as.numeric(lim$year)
lim$vi <- 1 / (lim$N - 3)
model <- metafor::rma.mv(
  yi = yi, V = vi, mods = ~ Environment * year,
  random = list(~ 1 | Article, ~ 1 | Datapoint), data = na.omit(lim)
)
test <- orchaRd::mod_results(
  model, mod = "year", group = "Article",
  data = lim, weights = "prop", by = "Environment")
orchaRd::bubble_plot(
  test, mod = "year", group = "Article",
  legend.pos = "top.left")
# Or just using model directly
orchaRd::bubble_plot(
  model, mod = "year", legend.pos = "top.left",
```

```
group = "Article", weights = "prop",
by = "Environment")
```

caterpillars

caterpillars

Description

Using a **metafor** model object of class `rma` or `rma.mv` or a results table of class `orchard`, this function produces a caterpillar plot from mean effect size estimates for all levels of a given categorical moderator, their corresponding confidence intervals, and prediction intervals.

Usage

```
caterpillars(
  object,
  mod = "1",
  group,
  xlab,
  overall = TRUE,
  transfm = c("none", "tanh", "invlogit", "percent", "percentr", "inv_ft"),
  n_transfm = NULL,
  colerrorbar = "#00CD00",
  colpoint = "#FFD700",
  colpoly = "red",
  k = TRUE,
  g = TRUE,
  at = NULL,
  by = NULL,
  weights = "prop"
)
```

Arguments

<code>object</code>	Model object of class <code>rma.mv</code> , <code>rma</code> or <code>orchard</code> table of model results
<code>mod</code>	The name of a moderator variable. Otherwise, "1" for an intercept-only model.
<code>group</code>	The grouping variable that one wishes to plot beside total effect sizes, <code>k</code> . This could be <code>study</code> , <code>species</code> , or whatever other grouping variable one wishes to present sample sizes for.
<code>xlab</code>	The effect size measure label.
<code>overall</code>	Logical, indicating whether to re-label "Intrcpt" (the default label from <code>rma</code> or <code>rma.mv</code> intercept only models or meta-analyses) to "Overall". Defaults to <code>TRUE</code> .

transfm	If set to "tanh", a tanh transformation will be applied to effect sizes, converting Z_r to a correlation or pulling in extreme values for other effect sizes (lnRR, lnCVR, SMD). "invlogit" can be used to convert lnRR to the inverse logit scale. "percentr" can convert to the percentage change scale when using response ratios and "percent" can convert to the percentage change scale of an log transformed effect size. Defaults to "none".
n_transfm	The vector of sample sizes for each effect size estimate. This is used when transfm = "inv_ft". Defaults to NULL.
colerrorbar	Colour of the error bar in the caterpillars plot. Defaults to hex code - "#00CD00".
colpoint	Point estimate colour in the caterpillars plot. Defaults to hex code - "#FFD700".
colpoly	Polygon colour in the caterpillars plot. Defaults to "red".
k	If TRUE, it displays k (number of effect sizes) on the plot.
g	If TRUE, it displays g (number of grouping levels for each level of the moderator) on the plot.
at	Used when one wants marginalised means. List of levels one wishes to predict at for the corresponding variables in by, but is not presented in output. Provide a list as follows: list(mod = c("level1", "level2")).
by	Used when one wants marginalised means. Character vector indicating the name that predictions should be conditioned on for the levels of the moderator.
weights	Used when one wants marginalised means. How to marginalize categorical variables. The default is weights = "prop", which weights means for moderator levels based on their proportional representation in the data. For example, if "sex" is a moderator, and males have a larger sample size than females, then this will produce a weighted average, where males are weighted more towards the mean than females. This may not always be ideal. In the case of sex, for example, males and females are roughly equally prevalent in a population. As such, you can give the moderator levels equal weight using weights = "equal".

Value

Caterpillars plot

Author(s)

Shinichi Nakagawa - s.nakagawa@unsw.edu.au

Daniel Noble - daniel.noble@anu.edu.au

Examples

```
data(eklof)
eklof<-metafor::escalc(measure="ROM", n1i=N_control, sd1i=SD_control,
m1i=mean_control, n2i=N_treatment, sd2i=SD_treatment, m2i=mean_treatment,
data=eklof)
# Add the unit level predictor
eklof$Datapoint<-as.factor(seq(1, dim(eklof)[1], 1))
# fit a MLMR - accounting for some non-independence
eklof_MR<-metafor::rma.mv(yi=yi, V=vi, mods=~ Grazer.type-1, random=list(~1|ExptID,
```

```

~1|Datapoint), data=eklof)
results <- mod_results(eklof_MR, mod = "Grazer.type", group = "First.author")
caterpillars(results, mod = "Grazer.type",
group = "First.author", xlab = "log(Response ratio) (lnRR)", g = FALSE)

# Example 2
data(lim)
lim$vi<- 1/(lim$N - 3)
lim_MR<-metafor::rma.mv(yi=yi, V=vi, mods=~Phylum-1, random=list(~1|Article,
~1|Datapoint), data=lim)
results_lim <- mod_results(lim_MR, mod = "Phylum", group = "Article")
caterpillars(results_lim, mod = "Phylum",
group = "Article", xlab = "Correlation coefficient", transfm = "tanh")

```

cor_diff

*Correlation Difference***Description**

Computes the difference in correlation coefficients between two groups along with associated sampling variance for meta-analysis. The correlation and associated sample size or the data frame of the two variables can be provided

Usage

```
cor_diff(cor1 = NULL, cor2 = NULL, n1 = NULL, n2 = NULL, x1 = NULL, x2 = NULL)
```

Arguments

cor1	Numeric, the correlation coefficient for group 1.
cor2	Numeric, the correlation coefficient for group 2.
n1	Integer, the sample size for group 1.
n2	Integer, the sample size for group 2.
x1	A numeric vector for group 1.
x2	A numeric vector for group 2.

Value

A data frame with the difference in correlations and the sampling variances of the difference.

Examples

```

set.seed(982)
# Example with known correlations
cor_diff(0.5, 0.3, 100, 100)
cor_diff(0.2, 0.4, 40, 40)

```

cvh1_ml	<i>cvh1_ml</i>
---------	----------------

Description

CVH1 for mulilevel meta-analytic models, based on Yang et al. (2023). Under multilevel models, we can have multiple CVH1. TODO - we need to cite original CVH1 paper

Usage

```
cvh1_ml(model, boot = NULL)
```

Arguments

model	Model object of class <code>rma.mv</code> or <code>rma</code> . Currently only model objects using the <code>mods</code> argument work (e.g., <code>mod = ~1</code>).
boot	Number of simulations to run to produce 95 percent confidence intervals for I2. Default is <code>NULL</code> , where only the point estimate is provided.

Value

A data frame containing all the model results including mean effect size estimate, confidence, and prediction intervals

Author(s)

Shinichi Nakagawa - s.nakagawa@unsw.edu.au
Daniel Noble - daniel.noble@anu.edu.au

References

TODO

Examples

```
library(metafor)
# NOTE: boot is set LOW here for speed; use boot >= 1000 in practice.
data(english)
english <- escalc(measure = "SMD", n1i = NStartControl,
  sd1i = SD_C, m1i = MeanC, n2i = NStartExpt, sd2i = SD_E,
  m2i = MeanE, var.names = c("SMD", "vSMD"), data = english)
english_MA <- rma.mv(yi = SMD, V = vSMD,
  random = list(~1 | StudyNo, ~1 | EffectID), data = english)
cvh1_ml(english_MA)
cvh1_ml(english_MA, boot = 10)
```

cvh2_ml

*cvh2_ml***Description**

CVH2 for mulilevel meta-analytic models, based on Yang et al. (2023). Under multilevel models, we can have multiple CVH2. TODO - we need to cite original CVH2 paper

Usage

```
cvh2_ml(model, boot = NULL)
```

Arguments

model	Model object of class <code>rma.mv</code> or <code>rma</code> . Currently only model objects using the <code>mods</code> argument work (e.g., <code>mod = ~1</code>).
boot	Number of simulations to run to produce 95 percent confidence intervals for I2. Default is <code>NULL</code> , where only the point estimate is provided.

Value

A data frame containing all the model results including mean effect size estimate, confidence, and prediction intervals

Author(s)

Shinichi Nakagawa - s.nakagawa@unsw.edu.au

Daniel Noble - daniel.noble@anu.edu.au

References

TODO

Examples

```
library(metafor)
# NOTE: boot is set LOW here for speed; use boot >= 1000 in practice.
data(english)
english <- escalc(measure = "SMD", n1i = NStartControl,
  sd1i = SD_C, m1i = MeanC, n2i = NStartExpt, sd2i = SD_E,
  m2i = MeanE, var.names = c("SMD", "vSMD"), data = english)
english_MA <- rma.mv(yi = SMD, V = vSMD,
  random = list(~1 | StudyNo, ~1 | EffectID), data = english)
cvh2_ml(english_MA)
cvh2_ml(english_MA, boot = 10)
```

 eklof

 eklof

Description

Eklof et al. (2012) evaluated the effects of predation on benthic invertebrate communities. Using the log response ratio they quantified differences in abundance and/or biomass of gastropods and Amphipods in groups with and without predation in an experimental setting. Below we describe the variables used in our examples and analyses.

Format

A data frame :

ExptID Experiment / effect size ID

First.author First author of publication

Publication.year Year of publication

Journal Journal published within

Research.group Research group that data arose from

Grazer.type Type of grazer

mean_treatment Mean abundance/biomass of invertebrates in treatment group

SD_treatment Standard deviation in abundance/biomass of invertebrates in treatment group

N_treatment Sample size of invertebrate sample in treatment group

mean_control Mean abundance/biomass of invertebrates in control group

SD_control Standard deviation in abundance/biomass of invertebrates in control group

N_control Sample size of invertebrate sample in control group ...

References

Eklof J.S., Alsterberg C., Havenhand J.N., Sundback K., Wood H.L., Gamfeldt L. 2012. Experimental climate change weakens the insurance effect of biodiversity. *Ecology Letters*, 15:864-872. <https://doi.org/10.1111/j.1461-0248.2012.01810.x>

english

english

Description

English and Uller (2016) performed a meta-analysis on the effects of early life dietary restriction (a reduction in a major component of the diet without malnutrition; e.g. caloric restriction) on average age at death, using the standardised mean difference (often called *d*). In a subsequent publication, Senior et al. (2017) analysed this dataset for effects of dietary-restriction on among-individual variation in the age at death using the log coefficient of variation ratio. A major prediction in both English & Uller (2016) and Senior et al. (2017) was that the type of manipulation, whether the study manipulated quality of food versus the quantity of food, would be important.

Format

A data frame :

StudyNo Study ID

EffectID Effect size ID

Author First author of study

Year Year of study publication

Journal Research journal where study was published

Species Common name of species

Phylum Phylum of species

ExptLifeStage Life stage when manipulation was undertaken

ManipType Type of food manipulation

CatchUp Whether species exhibits catchup growth

Sex Sex of organisms in sample

AdultDiet Diet adults were provided and whether it was restricted (treatment) or the control

NStartControl Sample size of the control group

NStartExpt Sample size of the treatment group

MeanC Mean of the control group

MeanE Mean of the treatment/experimental group

SD_C Standard deviation of the control group

SD_E Standard deviation of the treatment/experimental group ...

References

English S, Uller T. 2016. Does early-life diet affect longevity? A meta-analysis across experimental studies. *Biology Letters*, 12: <http://doi:10.1098/rsbl.2016.0291>

firstup	<i>firstup</i>
---------	----------------

Description

Uppercase moderator names

Usage

```
firstup(x, upper = TRUE)
```

Arguments

x	a character string
upper	logical indicating if the first letter of the character string should be capitalized. Defaults to TRUE.

Value

Returns a character string with all combinations of the moderator level names with upper case first letters

Author(s)

Shinichi Nakagawa - s.nakagawa@unsw.edu.au
Daniel Noble - daniel.noble@anu.edu.au

fish	<i>fish</i>
------	-------------

Description

Fish data example.

Format

A data frame

deg_dif Temperature difference between treatments

es_ID Effect size ID

experimental_design Experimental design type

group_ID Group ID

lnrr Log Response ratio effect size

lnrr_vi Sampling variance for lnRR

mean_control Mean for control
mean_treat Mean for treatment
n_control Sample size for control
n_treat Sample size for treatment
paper_ID Paper ID
sd_control SD for control
sd_treat SD for treatment
species_ID Species ID
trait.type Type of trait effect size was calculated from
treat_end_days Duration of time developmental treatment was applied

References

Lim J.N., Senior A.M., Nakagawa S. 2014. Heterogeneity in individual quality and reproductive trade-offs within species. *Evolution*. 68(8):2306-18. doi: 10.1111/evo.12446

geom_pub_stats_naka *geom_pub_stats_naka*

Description

This function adds a corrected meta-analytic mean, sensu Nakagawa et al. 2022, confidence interval and text annotation to an intercept only orchard plot.

Usage

```
geom_pub_stats_naka(
  data,
  col = "blue",
  plotadj = -0.05,
  textadj = 0.05,
  branch.size = 1.2,
  trunk.size = 3
)
```

Arguments

<code>data</code>	The data frame containing the corrected meta-analytic mean and confidence intervals.
<code>col</code>	The colour of the mean and confidence intervals.
<code>plotadj</code>	The adjustment to the x-axis position of the mean and confidence intervals.
<code>textadj</code>	The adjustment to the y-axis position of the mean and confidence intervals for the text displaying the type of correction.
<code>branch.size</code>	Size of the confidence intervals.
<code>trunk.size</code>	Size of the mean, or central point.

Value

A list of ggplot2 objects to be added to the orchard plot.

Author(s)

Daniel Noble - daniel.noble@anu.edu.au

geom_pub_stats_yang *geom_pub_stats_yang*

Description

This function adds a corrected meta-analytic mean, sensu Yang et al. 2023, confidence interval and text annotation to an intercept only orchard plot.

Usage

```
geom_pub_stats_yang(
  data,
  col = "red",
  plotadj = -0.05,
  textadj = 0.05,
  branch.size = 1.2,
  trunk.size = 3
)
```

Arguments

data	The data frame containing the corrected meta-analytic mean and confidence intervals.
col	The colour of the mean and confidence intervals.
plotadj	The adjustment to the x-axis position of the mean and confidence intervals.
textadj	The adjustment to the y-axis position of the mean and confidence intervals for the text displaying the type of correction.
branch.size	Size of the confidence intervals.
trunk.size	Size of the mean, or central point.

Value

A list of ggplot2 objects to be added to the orchard plot.

Author(s)

Daniel Noble - daniel.noble@anu.edu.au

get_data_raw	get_data_raw
--------------	--------------

Description

Collects and builds the data used to fit the `rma.mv` or `rma` model in **metafor**.

Usage

```
get_data_raw(
  model,
  mod,
  group,
  N = NULL,
  at = NULL,
  subset = TRUE,
  upper = TRUE
)
```

Arguments

<code>model</code>	<code>rma.mv</code> object.
<code>mod</code>	the moderator variable.
<code>group</code>	The grouping variable that one wishes to plot beside total effect sizes, k . This could be study, species, or whatever other grouping variable one wishes to present sample sizes.
<code>N</code>	The name of the column in the data specifying the sample size, N . Defaults to <code>NULL</code> , so precision is plotted instead of sample size.
<code>at</code>	List of moderators. If <code>at</code> is equal to <code>mod</code> then levels specified within <code>at</code> will be used to subset levels when <code>subset = TRUE</code> . Otherwise, it will marginalise over the moderators at the specified levels.
<code>subset</code>	Whether or not to subset levels within the <code>mod</code> argument. Defaults to <code>FALSE</code> .
<code>upper</code>	Logical indicating if the first letter of the moderator levels should be capitalized. Defaults to <code>TRUE</code> .

Value

Returns a data frame

Author(s)

Shinichi Nakagawa - s.nakagawa@unsw.edu.au
 Daniel Noble - daniel.noble@anu.edu.au

Examples

```
data(fish)
warm_dat <- fish
model <- metafor::rma.mv(
  yi = lnrr, V = lnrr_vi,
  random = list(~1 | group_ID, ~1 | es_ID),
  mods = ~ experimental_design + trait.type +
    deg_dif + treat_end_days,
  method = "REML", test = "t", data = warm_dat,
  control = list(optimizer = "optim",
    optmethod = "Nelder-Mead"))
test <- get_data_raw(
  model, mod = "trait.type",
  group = "group_ID",
  at = list(trait.type = c("physiology",
    "morphology")))
test2 <- get_data_raw(model, mod = "1", group = "group_ID")
```

get_data_raw_cont

get_data_raw_cont

Description

Collects and builds the data used to fit the `rma.mv` or `rma` model in **metafor** when a continuous variable is fit within a model object.

Usage

```
get_data_raw_cont(model, mod, group, N = NULL, by)
```

Arguments

<code>model</code>	<code>rma.mv</code> object.
<code>mod</code>	the moderator variable.
<code>group</code>	The grouping variable that one wishes to plot beside total effect sizes, k . This could be study, species or whatever other grouping variable one wishes to present sample sizes.
<code>N</code>	The name of the column in the data specifying the sample size, N . Defaults to <code>NULL</code> so that precision is plotted instead of sample size.
<code>by</code>	Character name(s) of the 'condition' variables to use for grouping into separate tables.

Value

Returns a data frame

Author(s)

Shinichi Nakagawa - s.nakagawa@unsw.edu.au

Daniel Noble - daniel.noble@anu.edu.au

get_ints_dat

get_ints_dat

Description

This function extracts the corrected meta-analytic mean and confidence intervals from a model object.

Usage

```
get_ints_dat(model, type = c("bc", "br"))
```

Arguments

model	The rma model object containing the corrected meta-analytic mean and confidence intervals.
type	The type of correction to extract the corrected meta-analytic mean and confidence intervals from. "br" (i.e., Bias Robust) for Yang et al. 2023, "bc" (i.e., Bias-Corrected) for Nakagawa et al. 2023.

Value

A list containing the corrected meta-analytic mean and confidence intervals, and a label for the plot.

Author(s)

Daniel Noble - daniel.noble@anu.edu.au

glmmTMB_to_rma

Convert a glmmTMB model to an rma.mv-compatible object

Description

Wraps a fitted **glmmTMB** model in a light-weight object with class `c("rma.mv", "rma")` so it can be passed to **orchaRd** functions that operate on **metafor** model objects. The main use-case is multilevel meta-analysis fitted with **glmmTMB**, including models that use structured covariance terms such as `equalto()` and `propto()`.

Usage

```
glmmTMB_to_rma(
  model,
  yi,
  vi,
  data = NULL,
  V = NULL,
  measure = NULL,
  test = "z",
  ddf = NULL,
  study_col = NULL
)
```

Arguments

<code>model</code>	A fitted <code>glmmTMB</code> model object.
<code>yi</code>	Name of the effect-size column in data (character), or a numeric vector of effect sizes.
<code>vi</code>	Name of the sampling-variance column in data (character), or a numeric vector of sampling variances.
<code>data</code>	Optional data frame. Defaults to <code>model\$frame</code> .
<code>V</code>	Optional known sampling-error variance-covariance matrix. Supply the same matrix used in <code>equalto()</code> if you want <code>orchaRd::i2_m1(method = "matrix")</code> to work on the converted object.
<code>measure</code>	Character string labelling the effect-size metric. Stored as an attribute on <code>yi</code> . Defaults to "GEN".
<code>test</code>	Either "z" or "t".
<code>ddf</code>	Optional denominator degrees of freedom used when <code>test = "t"</code> .
<code>study_col</code>	Optional study-level grouping variable used for automatic <code>ddf</code> calculation when <code>test = "t"</code> and <code>ddf = NULL</code> .

Details

The returned object is intended for analytical summaries and plotting functions that only consume the fitted model components. Workflows that require refitting a **metafor** model, such as `leave_one_out()` or bootstrap paths inside some heterogeneity helpers, are not supported for converted objects.

Value

A list of class `c("rma.mv", "rma")` containing the slots that **orchaRd** and **metafor** summary helpers expect.

i2_ml

*i2_ml***Description**

I2 (I-squared) for mulilevel meta-analytic models, based on Nakagawa & Santos (2012). Under multilevel models, we can have multiple I2 (see also Senior et al. 2016). Alternatively, the method proposed by Wolfgang Viechtbauer can also be used.

Usage

```
i2_ml(model, method = c("ratio", "matrix"), boot = NULL)
```

Arguments

model	Model object of class <code>rma.mv</code> or <code>rma</code> . Currently only model objects using the <code>mods</code> argument work (e.g., <code>mod = ~1</code>).
method	Method used to calculate I2. Two options exist: a ratio-based calculation proposed by Nakagawa & Santos (<code>"ratio"</code>), or Wolfgang Viechtbauer's matrix method (<code>"matrix"</code>).
boot	Number of simulations to run to produce 95 percent confidence intervals for I2. Default is <code>NULL</code> , where only the point estimate is provided.

Value

A data frame containing all the model results including mean effect size estimate, confidence, and prediction intervals

Author(s)

Shinichi Nakagawa - s.nakagawa@unsw.edu.au
Daniel Noble - daniel.noble@anu.edu.au

References

Senior, A. M., Grueber, C. E., Kamiya, T., Lagisz, M., O'Dwyer, K., Santos, E. S. A. & Nakagawa S. 2016. Heterogeneity in ecological and evolutionary meta-analyses: its magnitudes and implications. *Ecology* 97(12): 3293-3299. Nakagawa, S, and Santos, E.S.A. 2012. Methodological issues and advances in biological meta-analysis. *Evolutionary Ecology* 26(5): 1253-1274.

Examples

```
library(metafor)
# NOTE: boot is set LOW here for speed; use boot >= 1000 in practice.
data(english)
english <- escalc(measure = "SMD", nli = NStartControl,
  sdli = SD_C, mli = MeanC, n2i = NStartExpt, sd2i = SD_E,
```

```
m2i = MeanE, var.names = c("SMD", "vSMD"), data = english)
english_MA <- rma.mv(yi = SMD, V = vSMD,
  random = list(~1 | StudyNo, ~1 | EffectID), data = english)
i2_ml(english_MA)
i2_ml(english_MA, method = "matrix")
i2_ml(english_MA, boot = 10)
```

leave_one_out	<i>Leave-One-Out Analysis for Meta-Analytic Models</i>
---------------	--

Description

Performs a leave-one-out analysis on a meta-analytic model from the **metafor** package by iteratively removing each level of a grouping variable and refitting the model.

Usage

```
leave_one_out(
  model,
  group,
  vcalc_args = NULL,
  robust_args = NULL,
  phylo_args = NULL
)
```

Arguments

model	A meta-analytic model fitted using metafor package functions such as <code>rma.mv()</code> .
group	A character string specifying the column name in <code>model\$data</code> that contains the grouping variable for which levels will be iteratively removed.
vcalc_args	Optional list of arguments for variance-covariance calculation using <code>metafor</code> 's <code>vcalc</code> function. Must include: • <code>vi</code>: Name of the variance column • <code>cluster</code>: Name of the clustering variable column • <code>obs</code>: Name of the observation ID column • <code>rho</code>: Correlation coefficient between effect sizes
robust_args	Optional list of arguments for robust variance estimation using <code>metafor</code> 's <code>robust</code> function. Must include: • <code>cluster</code>: Name of the clustering variable column • <code>clubSandwich</code>: Logical indicating whether to use <code>clubSandwich</code> method (optional)
phylo_args	Optional list of arguments for phylogenetic matrix calculation using <code>ape</code> 's <code>vcv</code> function. Must include: • <code>tree</code>: A phylogenetic tree object of class "phylo" • <code>species_colname</code>: Name of the column in model data linked to the tree tips

Value

An object of class "orchard" containing:

- **mod_table**: A data frame with model estimates from each leave-one-out iteration, with an additional column indicating which group was omitted.
- **data**: A data frame with effect sizes from each iteration, with an additional column indicating which group was omitted.
- **orig_mod_results**: The results from the original model without any omissions, as returned by `mod_results()`.

Author(s)

Facundo Decunta - fdecunta@agro.uba.ar

Examples

```
# Subset to a few studies so the leave-one-out refits run quickly
fish_sub <- fish[fish$paper_ID %in% unique(fish$paper_ID)[1:5], ]
res <- metafor::rma.mv(lnrr, lnrr_vi, random = ~ 1 | paper_ID, data = fish_sub)
loo_results <- leave_one_out(res, group = "paper_ID")

# With robust variance estimation
loo_robust <- leave_one_out(res, group = "paper_ID",
                           robust_args = list(cluster = "paper_ID"))
```

lim

lim

Description

Lim et al. (2014) meta-analysed the strength of correlation between maternal and offspring size within-species, across a very wide range of taxa. They found that typically there is a moderate positive correlation between maternal size and offspring size within species (i.e. larger mothers have larger offspring). Below we describe the variables use in our examples and analyses.

Format

A data frame :

Amniotes Amniote group

Article Study or research paper that data were collected from

Author Authors of article

Class Class of organism

Common.Name Common name of species

- Datapoint** Observation level ID that identified each unique datapoint
- Environment** Environment organism is found, wild or captive
- Family** Family of species
- Genus** Genus of species
- Maternal** Mother length, or mass
- N** Sample size used to estimated correlation
- Offspring**
- Order** Order of species
- Phylum** Phylum of the species
- Propagule**
- RU**
- Reproduction** Reproductive mode of species
- animal** Species name use for phylogenetic reconstruction
- year** year of study
- yi** Effect size correlation coefficient between maternal and offspring size within-species ...

References

Lim J.N., Senior A.M., Nakagawa S. 2014. Heterogeneity in individual quality and reproductive trade-offs within species. *Evolution*. 68(8):2306-18. doi: 10.1111/evo.12446

<i>m1_ml</i>	<i>m1_ml</i>
--------------	--------------

Description

M1 for mulilevel meta-analytic models, based on Yang et al. (2023). Under multilevel models, we can have multiple M1 - TODO - we need to cite original M1 paper

Usage

```
m1_ml(model, boot = NULL)
```

Arguments

- model** Model object of class *rma.mv* or *rma*. Currently only model objects using the *mods* argument work (e.g., *mod = ~1*).
- boot** Number of simulations to run to produce 95 percent confidence intervals for M1. Default is *NULL*, where only the point estimate is provided.

Value

A data frame containing all the model results including mean effect size estimate, confidence, and prediction intervals

Author(s)

Shinichi Nakagawa - s.nakagawa@unsw.edu.au
Daniel Noble - daniel.noble@anu.edu.au

References

TODO

Examples

```
library(metafor)
# NOTE: boot is set LOW here for speed; use boot >= 1000 in practice.
data(english)
english <- escalc(measure = "SMD", nli = NStartControl,
  sdli = SD_C, mli = MeanC, n2i = NStartExpt, sd2i = SD_E,
  m2i = MeanE, var.names = c("SMD", "vSMD"), data = english)
english_MA <- rma.mv(yi = SMD, V = vSMD,
  random = list(~1 | StudyNo, ~1 | EffectID), data = english)
m1_ml(english_MA)
m1_ml(english_MA, boot = 10)
```

m2_ml	<i>m2_ml</i>
-------	--------------

Description

M2 for mulilevel meta-analytic models, based on Yang et al. (2023). Under multilevel models, we can have multiple M2 - TODO - we need to cite original M2 paper

Usage

```
m2_ml(model, boot = NULL)
```

Arguments

- | | |
|-------|---|
| model | Model object of class rma.mv or rma. Currently only model objects using the mods argument work (e.g., mod = ~1). |
| boot | Number of simulations to run to produce 95 percent confidence intervals for M2. Default is NULL, where only the point estimate is provided. |

Value

A data frame containing all the model results including mean effect size estimate, confidence, and prediction intervals

Author(s)

Shinichi Nakagawa - s.nakagawa@unsw.edu.au

Daniel Noble - daniel.noble@anu.edu.au

References

TODO

Examples

```
library(metafor)
# NOTE: boot is set LOW here for speed; use boot >= 1000 in practice.
data(english)
english <- escalc(measure = "SMD", n1i = NStartControl,
  sd1i = SD_C, m1i = MeanC, n2i = NStartExpt, sd2i = SD_E,
  m2i = MeanE, var.names = c("SMD", "vSMD"), data = english)
english_MA <- rma.mv(yi = SMD, V = vSMD,
  random = list(~1 | StudyNo, ~1 | EffectID), data = english)
m2_ml(english_MA)
m2_ml(english_MA, boot = 10)
```

magnitude_effects

Magnitude Effects using SAFE Bootstrap

Description

Computes the magnitude effect size and its variance using the SAFE bootstrap method for either independent or dependent samples.

Usage

```
magnitude_effects(
  x1bar,
  x2bar,
  sd1,
  sd2,
  n1,
  n2,
  min_kept = 2000,
  chunk_init = 4000,
  chunk_max = 2e+06,
  max_draws = Inf,
  patience_noaccept = 5,
  paired = FALSE,
  r = NULL,
  seed = 565
)
```

Arguments

x1bar	Mean of group 1.
x2bar	Mean of group 2.
sd1	Standard deviation of group 1.
sd2	Standard deviation of group 2.
n1	Sample size of group 1.
n2	Sample size of group 2.
min_kept	Minimum number of accepted bootstrap samples to keep. Default is 2000.
chunk_init	Initial chunk size for bootstrap sampling. Default is 4000.
chunk_max	Maximum chunk size for bootstrap sampling. Default is 2e6.
max_draws	Maximum total number of bootstrap draws. Default is Inf.
patience_noaccept	Number of consecutive chunks with no accepted samples before stopping. Default is 5.
paired	Logical, whether the samples are paired. Default is FALSE.
r	Optional correlation between groups for dependent samples. Required if 'paired' is TRUE.
seed	Random seed for reproducibility. Default is 565.

Value

A list containing the point estimate, variance, number of kept samples, total draws, number of attempts, and status.

Examples

```
# min_kept/chunk_init are set LOW here for speed; use the
# defaults for real analyses.
# Independent samples
magnitude_effects(x1bar = 10, x2bar = 7, sd1 = 2, sd2 = 3,
                  n1 = 30, n2 = 30, min_kept = 200, chunk_init = 400)
# Dependent (paired) samples, given the correlation between groups
magnitude_effects(x1bar = 10, x2bar = 7, sd1 = 2, sd2 = 3,
                  n1 = 30, n2 = 30, paired = TRUE, r = 0.5,
                  min_kept = 200, chunk_init = 400)
```

matrix_i2

*matrix_i2***Description**

Calculated I2 (I-squared) for mulilevel meta-analytic models, based on a matrix method proposed by Wolfgang Viechtbauer.

Usage

```
matrix_i2(model)
```

Arguments

model Model object of class `rma.mv` or `rma`.

Value

A named numeric vector of I2 (I-squared) values expressed as percentages. The first element, `I2_Total`, is the total heterogeneity; each remaining element (named `I2_<level>` after a random-effect level of the model) is the heterogeneity attributable to that level.

Examples

```
library(metafor)
data(english)
english <- escalc(
  measure = "SMD", nli = NStartControl,
  sdli = SD_C, mli = MeanC,
  n2i = NStartExpt, sd2i = SD_E, m2i = MeanE,
  var.names = c("SMD", "vSMD"), data = english)
english_MA <- rma.mv(
  yi = SMD, V = vSMD,
  random = list(~1 | StudyNo, ~1 | EffectID),
  data = english)
matrix_i2(english_MA)
```

ml_cvh1

*ml_cvh1***Description**

Calculated CVH1 for mulilevel meta-analytic models

Usage

```
ml_cvh1(model)
```


Value

A named numeric vector of M1 heterogeneity values (a proportion-type index on the standard-deviation scale, bounded between 0 and 1). The first element, `M1_Total`, is the overall value across all random effects; each remaining element (named `M1_<level>` after a random-effect level of the model) is the value for that level.

<code>m1_m2</code>	<i>m1_m2</i>
--------------------	--------------

Description

Calculated CV for mulilevel meta-analytic models

Usage

`m1_m2(model)`

Arguments

`model` Model object of class `rma.mv` or `rma`.

Value

A named numeric vector of M2 heterogeneity values (a proportion-type index on the variance scale, bounded between 0 and 1). The first element, `M2_Total`, is the overall value across all random effects; each remaining element (named `M2_<level>` after a random-effect level of the model) is the value for that level.

<code>mod_results</code>	<i>mod_results</i>
--------------------------	--------------------

Description

Using a **metafor** model object of class `rma` or `rma.mv`, this function creates a table of model results containing the mean effect size estimates for all levels of a given categorical moderator, and their corresponding confidence and prediction intervals. The function is capable of calculating marginal means from meta-regression models, including those with multiple moderator variables of mixed types (i.e. continuous and categorical variables).

Usage

```
mod_results(
  model,
  mod = "1",
  group,
  N = NULL,
  weights = "prop",
  by = NULL,
  at = NULL,
  subset = FALSE,
  upper = TRUE,
  ...
)
```

Arguments

model	rma.mv model object
mod	Moderator variable of interest that one wants marginal means for. Defaults to the intercept, i.e. "1".
group	The grouping variable that one wishes to plot beside total effect sizes, k. This could be study, species, or any grouping variable one wishes to present sample sizes for.
N	The name of the column in the data specifying the sample size so that each effect size estimate is scaled to the sample size, N. Defaults to NULL, so that precision is used for scaling each raw effect size estimate instead of sample size.
weights	How to marginalize categorical variables. The default is weights = "prop", which weights moderator level means based on their proportional representation in the data. For example, if "sex" is a moderator, and males have a larger sample size than females, then this will produce a weighted average, where males are weighted more towards the mean than females. This may not always be ideal. In the case of sex, for example, males and females are roughly equally prevalent in a population. As such, you can give the moderator levels equal weight using weights = "equal".
by	Character vector indicating the name that predictions should be conditioned on for the levels of the moderator.
at	List of levels one wishes to predict at for the corresponding variables in by. Used when one wants marginalised means. This argument can also be used to suppress levels of the moderator when argument subset = TRUE. Provide a list as follows: list(mod = c("level1", "level2")).
subset	Used when one wishes to only plot a subset of levels within the main moderator of interest defined by mod. Default is FALSE, but use TRUE if you wish to subset levels of a moderator plotted (defined by mod) for plotting. Levels one wishes to plot are specified as a list, with the level names as a character string in the at argument. For subsetting to work, the at argument also needs to be specified so that the mod_results function knows what levels one wishes to plot.

upper	Logical, defaults to TRUE, indicating that the first letter of the character string for the moderator variable should be capitalized.
...	Additional arguments passed to <code>emmeans::emmeans()</code> .

Details

Prediction intervals with random-slope models (struct = "GEN" or "HCS"):

When your `rma.mv` model includes a random-slope term (e.g., `random = ~ moderator | study` with `struct = "GEN" or "HCS"`), prediction intervals require the full variance-covariance matrix of the random effects — not just the sum of the variance components. In a random-slope model the random intercept and slope are typically correlated, and a correct prediction interval at a given moderator value x must incorporate:

$$\text{Var}(u_{0j} + u_{1j}x) = \tau_0^2 + 2x\rho\tau_0\tau_1 + x^2\tau_1^2$$

where τ_0^2 and τ_1^2 are the random intercept and slope variances, and ρ is their correlation. Simply summing `tau2`, `gamma2`, and `sigma2` ignores the slope variance contribution and the covariance, producing intervals that may be too narrow or too wide depending on the moderator value.

`metafor::predict.rma()` does not currently return prediction intervals for models with GEN or HCS structures for this reason. `orchaRd` will issue a warning when it detects such a structure. Users should verify prediction intervals manually using the approach described by Pustejovsky (2024, R-sig-meta-analysis mailing list): <https://stat.ethz.ch/pipermail/r-sig-meta-analysis/2024-March/005152.html>.

A practical workaround is to re-centre the moderator at the value of interest (so the random slope term evaluates at zero) and then compare the resulting prediction interval with a manual calculation using the full variance-covariance matrix from the model.

Value

A data frame containing all the model results including mean effect size estimate, confidence and prediction intervals

Author(s)

Shinichi Nakagawa - s.nakagawa@unsw.edu.au

Daniel Noble - daniel.noble@anu.edu.au

Examples

```
# Simple eklof data
data(eklof)
eklof<-metafor::escalc(measure="ROM", n1i=N_control, sd1i=SD_control,
m1i=mean_control, n2i=N_treatment, sd2i=SD_treatment, m2i=mean_treatment, data = eklof)
# Add the unit level predictor
eklof$Datapoint<-as.factor(seq(1, dim(eklof)[1], 1))
# fit a MLMR - accounting for some non-independence
eklof_MR<-metafor::rma.mv(yi=yi, V=vi, mods=~ Grazer.type, random=list(~1|ExptID,
~1|Datapoint), data = eklof)
```

```

results <- mod_results(eklof_MR, mod = "Grazer.type", group = "ExptID")

# Fish example demonstrating marginalised means
data(fish)
model <- metafor::rma.mv(yi = lnrr, V = lnrr_vi,
  random = list(~1 | group_ID, ~1 | es_ID),
  mods = ~ trait.type + deg_dif,
  method = "REML", test = "t", data = fish)
overall <- mod_results(model, group = "group_ID")
across_trait <- mod_results(model, group = "group_ID", mod = "trait.type")
# Marginalised means, conditioning on levels of a continuous moderator
across_trait_by_deg <- mod_results(model, group = "group_ID",
  mod = "trait.type", at = list(deg_dif = c(5, 10, 15)), by = "deg_dif")

```

moment_effects	<i>moment_effects</i>
----------------	-----------------------

Description

Computes the effect estimate of the difference in skewness and kurtosis between two groups along with associated sampling variance for meta-analysis.

Usage

```
moment_effects(x1, x2, type = c("skew", "kurt"))
```

Arguments

x1	A numeric vector.
x2	A numeric vector.
type	Character, either "skew" or "kurt" to specify the type of moment effect.

Value

A data frame with the difference in moment effects and their variances.

Examples

```

set.seed(982)

# Just some random comparisons
moment_effects(rnorm(100), rnorm(100), type = "skew")
moment_effects(rnorm(40), rnorm(40), type = "skew")
moment_effects(rnorm(40), rnorm(40), type = "kurt")
moment_effects(rnorm(100), rnorm(100), type = "kurt")

# Comparisons with known moment differences
m1 <- c(mean = 0, variance = 1, skewness = 0, kurtosis = 2)

```

```

m2 <- c(mean = 0, variance = 1, skewness = 0, kurtosis = 4)

x1 <- PearsonDS::rpearson(1000, moments = m1)
x2 <- PearsonDS::rpearson(1000, moments = m2)
moment_effects(x1, x2, type = "skew") # ~0
moment_effects(x1, x2, type = "kurt") # ~-2

m1 <- c(mean = 0, variance = 1, skewness = 0, kurtosis = 3)
m2 <- c(mean = 0, variance = 1, skewness = 1, kurtosis = 3)
x1 <- PearsonDS::rpearson(1000, moments = m1)
x2 <- PearsonDS::rpearson(1000, moments = m2)
moment_effects(x1, x2, type = "skew") # ~-1
moment_effects(x1, x2, type = "kurt") # ~0

```

num_studies

num_studies

Description

Computes how many studies are in each level of categorical moderators of a `rma.mv` model object.

Usage

```
num_studies(data, mod, group)
```

Arguments

<code>data</code>	Raw data from object of class "orchard"
<code>mod</code>	Character string describing the moderator of interest.
<code>group</code>	A character string specifying the column name of the study ID grouping variable.

Value

Returns a table with the number of studies in each level of all parameters within a `rma.mv` or `rma` object.

Author(s)

Shinichi Nakagawa - s.nakagawa@unsw.edu.au

Daniel Noble - daniel.noble@anu.edu.au

Examples

```
data(fish)
warm_dat <- fish
model <- metafor::rma.mv(
  yi = lnrr, V = lnrr_vi,
  random = list(~1 | es_ID, ~1 | group_ID),
  mods = ~ experimental_design - 1,
  method = "REML", test = "t", data = warm_dat,
  control = list(optimizer = "optim",
    optmethod = "Nelder-Mead"))
res <- mod_results(model, mod = "experimental_design", group = "group_ID")
num_studies(res$data, moderator, stdy)
```

orchard_leave1out	<i>Orchard Plot for Leave-One-Out Analysis</i>
-------------------	--

Description

Plots the output of a leave-one-out analysis for a meta-analytic model. The plot displays the effect sizes, overall effect estimate (with confidence and prediction intervals), and optionally, the effect sizes left out ("ghost points") in each iteration. In addition, the original model's 95% confidence limits are shown as reference lines.

Usage

```
orchard_leave1out(
  leave1out,
  ylab = NULL,
  ci_lines = TRUE,
  ci_lines_color = "red",
  ghost_points = TRUE,
  angle = 0,
  g = FALSE,
  transfm = c("none", "tanh", "invlogit", "percent", "percentr"),
  ...
)
```

Arguments

<code>leave1out</code>	Output from <code>leave_one_out()</code> , containing the leave-one-out analysis results.
<code>ylab</code>	Optional label for the y-axis, which lists the elements left out.
<code>ci_lines</code>	Logical indicating whether to add horizontal reference lines (default is TRUE) representing the original model's 95% confidence intervals.
<code>ci_lines_color</code>	Character string specifying the color for the confidence interval lines (default is "red").

<code>ghost_points</code>	Logical indicating whether to add "ghost points" for the effect sizes that were left out. These are plotted as empty points (default is TRUE).
<code>angle</code>	Numeric value specifying the angle of the x-axis labels in degrees (default is 0).
<code>g</code>	Logical indicating whether to return the plot as a grob object (default is FALSE).
<code>transfm</code>	Character string specifying the transformation to apply to the data. Options are: "none" (default, no transformation), "tanh" (hyperbolic tangent), "invlogit" (inverse logit), "percent" (percentage), or "percentr" (percentage range).
<code>...</code>	Additional arguments passed to <code>orchard_plot</code> .

Details

This function is useful to see how sensitive is the overall effect size estimated by a meta-analytic model to exclusion of each element from a group.

The y-axis shows the elements from the group. For each one, the plot shows the result from the model that left-out that element. For each model, the plot shows the effect sizes, overall effect estimate with confidence and prediction intervals, and "ghost points". This "ghost points" are the effect sizes left out and they are shown as empty points. They can be omitted setting 'ghost_points' to FALSE.

The plot also shows 95 as red dotted lines. The arguments 'ci_lines' and 'ci_lines_color' allow to remove this lines or select their color.

The function internally sets a color palette for the plot. If more than 20 elements are in the group, a viridis palette is used. Otherwise, a color-blind friendly palette is applied by default.

Value

A ggplot2 object displaying the leave-one-out analysis with reference lines for the original model's confidence limits.

Author(s)

Facundo Decunta - fdecunta@agro.uba.ar

Examples

```
# Subset to a few studies so the leave-one-out refits run quickly
fish_sub <- fish[fish$paper_ID %in% unique(fish$paper_ID)[1:5], ]
res <- metafor::rma.mv(lnrr, lnrr_vi, random = ~ 1 | paper_ID, data = fish_sub)
loo_res <- orchaRd::leave_one_out(res, group = "paper_ID")
orchard_leave1out(leave1out = loo_res, xlab = "lnRR")
```

orchard_plot

orchard_plot

Description

Using a **metafor** model object of class `rma` or `rma.mv`, or a results table of class `orchard`, it creates an orchard plot from mean effect size estimates for all levels of a given categorical moderator, and their corresponding confidence and prediction intervals.

Usage

```
orchard_plot(
  object,
  mod = "1",
  group,
  xlab,
  N = NULL,
  alpha = 0.5,
  angle = 90,
  cb = TRUE,
  k = TRUE,
  g = TRUE,
  est = FALSE,
  mod.order = NULL,
  trunk.size = 0.5,
  branch.size = 1.2,
  twig.size = 0.5,
  point.size = c(1, 3.5),
  transfm = c("none", "tanh", "invlogit", "percent", "percentr", "inv_ft"),
  n_transfm = NULL,
  condition.lab = NULL,
  legend.pos = c("bottom.right", "bottom.left", "top.right", "top.left", "top.out",
    "bottom.out", "none"),
  k.pos = c("right", "left", "none"),
  k.size = 3.5,
  est.size = 3,
  refline.pos = 0,
  colour = FALSE,
  fill = TRUE,
  weights = "prop",
  by = NULL,
  at = NULL,
  upper = TRUE,
  flip = TRUE
)
```

Arguments

<code>object</code>	model object of class <code>rma.mv</code> , <code>rma</code> , or orchard table of model results.
<code>mod</code>	the name of a moderator. Defaults to "1" for an intercept-only model. Not needed if an <code>orchard_plot</code> is provided with a <code>mod_results</code> object of class <code>orchard</code> .
<code>group</code>	The grouping variable that one wishes to plot beside total effect sizes, k . This could be study, species, or any grouping variable one wishes to present sample sizes for. Not needed if an <code>orchard_plot</code> is provided with a <code>mod_results</code> object of class <code>orchard</code> .
<code>xlab</code>	The effect size measure label.
<code>N</code>	The name of the column in the data specifying the sample size so that each effect size estimate is scaled to the sample size, N . Defaults to <code>NULL</code> , so that precision is used for scaling each raw effect size estimate instead of sample size.
<code>alpha</code>	The level of transparency for effect sizes represented in the orchard plot.
<code>angle</code>	The angle of y labels. The default is 90 degrees.
<code>cb</code>	If <code>TRUE</code> , it uses 20 colour blind friendly colors.
<code>k</code>	If <code>TRUE</code> , it displays k (number of effect sizes) on the plot.
<code>g</code>	If <code>TRUE</code> , it displays g (number of grouping levels for each level of the moderator) on the plot.
<code>est</code>	If <code>TRUE</code> , it displays the mean estimate and confidence interval (e.g., $0.25 [0.10, 0.40]$) alongside or below the k labels on the plot. Defaults to <code>FALSE</code> .
<code>mod.order</code>	Order in which to plot the groups of the moderator when it is a categorical one. Should be a vector of equal length to number of groups in the categorical moderator, in the desired order (bottom to top, or left to right for flipped orchard plot)
<code>trunk.size</code>	Size of the mean, or central point.
<code>branch.size</code>	Size of the confidence intervals.
<code>twig.size</code>	Size of the prediction intervals.
<code>point.size</code>	Numeric vector of length 2, specifying the minimum and maximum point sizes for effect size bubbles. Defaults to <code>c(1, 3.5)</code> . Useful for controlling bubble size in small figures.
<code>transfm</code>	If set to "tanh", a tanh transformation will be applied to effect sizes, converting Z_r to a correlation or pulling in extreme values for other effect sizes (lnRR, lnCVR, SMD). "invlogit" can be used to convert lnRR to the inverse logit scale. "percentr" can convert to the percentage change scale when using response ratios and "percent" can convert to the percentage change scale of an log transformed effect size. Defaults to "none".
<code>n_transfm</code>	The vector of sample sizes for each effect size estimate. This is used when <code>transfm = "inv_ft"</code> . Defaults to <code>NULL</code> .
<code>condition.lab</code>	Label for the condition being marginalized over.
<code>legend.pos</code>	Where to place the legend. To remove the legend, use <code>legend.pos = "none"</code> .

<code>k.pos</code>	Where to put k (number of effect sizes) on the plot. Users can specify the exact position or they can use specify "right", "left", or "none". Note that numeric values (0, 0.5, 1) can also be specified and this would give greater precision.
<code>k.size</code>	Numeric, the font size for k (and g) labels on the plot. Defaults to 3.5.
<code>est.size</code>	Numeric, the font size for estimate and CI labels when <code>est = TRUE</code> . Defaults to 3.0.
<code>refline.pos</code>	Where to put the reference line. defaults to 0.
<code>colour</code>	Colour of effect size shapes. By default, effect sizes are colored according to the <code>mod</code> argument. If <code>TRUE</code> , they are colored according to the grouping variable
<code>fill</code>	If <code>TRUE</code> , effect sizes will be filled with colours. If <code>FALSE</code> , they will not be filled with colours.
<code>weights</code>	Used when one wants marginalised means. How to marginalize categorical variables. The default is <code>weights = "prop"</code> , which weights moderator level means based on their proportional representation in the data. For example, if "sex" is a moderator, and males have a larger sample size than females, then this will produce a weighted average, where males are weighted more towards the mean than females. This may not always be ideal. In the case of sex, for example, males and females are roughly equally prevalent in a population. As such, you can give the moderator levels equal weight using <code>weights = "equal"</code> .
<code>by</code>	Character vector indicating the name that predictions should be conditioned on for the levels of the moderator.
<code>at</code>	List of levels one wishes to predict at for the corresponding variables in 'by'. Used when one wants marginalised means. This argument can also be used to suppress levels of the moderator when argument <code>subset = TRUE</code> . Provide a list as follows: <code>list(mod = c("level1", "level2"))</code> .
<code>upper</code>	Logical, defaults to <code>TRUE</code> , indicating that the first letter of the character string for the moderator variable should be capitalized.
<code>flip</code>	Logical, defaults to <code>TRUE</code> , indicating whether the plot should be flipped.

Value

Orchard plot

Author(s)

Shinichi Nakagawa - s.nakagawa@unsw.edu.au

Daniel Noble - daniel.noble@anu.edu.au

Examples

```
data(eklof)
eklof<-metafor::escalc(measure="ROM", n1i=N_control, sd1i=SD_control,
m1i=mean_control, n2i=N_treatment, sd2i=SD_treatment, m2i=mean_treatment,
data=eklof)
# Add the unit level predictor
eklof$Datapoint<-as.factor(seq(1, dim(eklof)[1], 1))
```

```
# fit a MLMR - accounting for some non-independence
eklof_MR<-metafor::rma.mv(yi=yi, V=vi, mods=~ Grazer.type-1,
random=list(~1|ExptID, ~1|Datapoint), data=eklof)
results <- mod_results(eklof_MR, mod = "Grazer.type", group = "ExptID")
orchard_plot(results, mod = "Grazer.type",
group = "ExptID", xlab = "log(Response ratio) (lnRR)")
# or
orchard_plot(eklof_MR, mod = "Grazer.type", group = "ExptID",
xlab = "log(Response ratio) (lnRR)")

# Example 2
data(lim)
lim$vi<- 1/(lim$N - 3)
lim_MR<-metafor::rma.mv(yi=yi, V=vi, mods=~Phylum-1, random=list(~1|Article,
~1|Datapoint), data=lim)
orchard_plot(lim_MR, mod = "Phylum", group = "Article",
xlab = "Correlation coefficient", transfm = "tanh", N = "N")
```

phylo_matrix	<i>phylo_matrix</i>
--------------	---------------------

Description

Phylogenetic matrix used in Pottier et al. (2021) meta-analysis on sex-differences in acclimation

Format

A phylogenetic correlation matrix :

References

Pottier et al. 2022. Functional Ecology

pottier	<i>pottier</i>
---------	----------------

Description

Pottier et al. (2021) meta-analysis on sex-differences in acclimation

Format

A data frame :

initials
es_ID
study_ID
species_ID
population_ID
family_ID
shared_trt_ID
cohort_ID
note_ID
data_source
data_url
fig_file_name
data_type
data_file_name
peer.reviewed
ref
title
pub_year
journal
thesis_chapter
doi
citation
phylum
class
order
family
genus
species
genus_species
habitat
taxonomic_group
reproduction_mode
life_stage_manip
life_stage_tested
brought_common_temp

mobility_life_stage_manip
time_common_temp
common_temp
exp_design
origin_hatching
latitude
longitude
elevation
season
year
body_length
body_mass
age_tested
sex
housing_temp
incubation_independent
metric
endpoint
acc_temp_low
acc_temp_high
acc_temp_var
is_acc_temp_fluctuating
acc_duration
ramping
set_time
n_test_temp
n_replicates_per_temp
n_animals_per_replicate
humidity
oxygen
salinity
pH
photoperiod
gravity
starved
minor_concerns
major_concerns

notes_moderators
mean_HT_low
sd_HT_low
n_HT_low
mean_HT_high
sd_HT_high
n_HT_high
error_type
notes_es
exclude
n_trt
n_cohort
within_study_mean_low
within_study_mean_high
within_study_sd_low
within_study_sd_high
between_study_mean_low
between_study_mean_high
between_study_sd_low
between_study_sd_high
imputed
imputed_sd_low
imputed_sd_high
dARR
Var_dARR
precision
is_concern
search_string
unique_name
ott_id
phylogeny ...

References

Pottier et al. 2022. Functional Ecology

```
pred_interval_esmeans  pred_interval_esmeans
```

Description

Function to get prediction intervals (credibility intervals) from esmeans objects (**metafor**).

Usage

```
pred_interval_esmeans(model, mm, mod, ...)
```

Arguments

<code>model</code>	<code>rma.mv</code> object.
<code>mm</code>	result from <code>emmeans::emmeans</code> object.
<code>mod</code>	Moderator of interest.
<code>...</code>	other arguments passed to function.

Value

A data frame built from `summary()` of the `emmeans` object: one row per level of the moderator giving the estimated marginal mean and its confidence limits, plus two added columns, `lower.PI` and `upper.PI`, containing the lower and upper bounds of the prediction (credibility) interval.

Author(s)

Shinichi Nakagawa - s.nakagawa@unsw.edu.au
 Daniel Noble - daniel.noble@anu.edu.au

```
print.orchard          print.orchard
```

Description

Print method for class 'orchard'

Usage

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

Arguments

<code>x</code>	an R object of class <code>orchard</code>
<code>...</code>	Other arguments passed to <code>print</code>

Value

Returns a data frame

Author(s)

Shinichi Nakagawa - s.nakagawa@unsw.edu.au

Daniel Noble - daniel.noble@anu.edu.au

pub_bias_plot	<i>pub_bias_plot</i>
---------------	----------------------

Description

This function adds the bias corrected mean and confidence intervals to an existing orchard plot that displays the overall meta-analytic mean effect size.

Usage

```
pub_bias_plot(
  plot,
  fe_model,
  v_model = NULL,
  col = c("red", "blue"),
  plotadj = -0.05,
  textadj = 0.05,
  branch.size = 1.2,
  trunk.size = 3
)
```

Arguments

plot	The orchard plot object to add the bias corrected mean and confidence intervals to. This plot needs to be a plot that displays the raw and a single meta-analytic mean (overall mean), confidence interval and prediction interval.
fe_model	The meta-analytic model (rma object) produced from a two-step correction (sensu Yang et al. 2023) (Step 1: Fixed effect model) with a robust correction to correct the meta-analytic mean for publication bias (selection bias) and the dependency in the data.
v_model	An optional argument. The meta-analytic model (rma object) to deal with publication bias using the meta-regression approach proposed by Nakagawa et al. 2023. This model can have fixed and/or random effects, with one random effect being sampling error (se or v). The intercept from this model can be considered the corrected meta-analytic mean when sample size is infinite or sampling error is zero.
col	The colour of the mean and confidence intervals.

plotadj	The adjustment to the x-axis position of the mean and confidence intervals.
textadj	The adjustment to the y-axis position of the mean and confidence intervals for the text displaying the type of correction.
branch.size	Size of the confidence intervals.
trunk.size	Size of the mean, or central point.

Value

An orchard plot with the corrected meta-analytic mean and confidence intervals added.

Author(s)

Daniel Noble - daniel.noble@anu.edu.au

Examples

```
library(metafor)
# Data
data(english)
# We need to calculate the effect sizes, in this case d
english <- escalc(
  measure = "SMD",
  n1i = NStartControl, sd1i = SD_C, m1i = MeanC,
  n2i = NStartExpt, sd2i = SD_E, m2i = MeanE,
  var.names = c("SMD", "vSMD"),
  data = english)

# Our MLMA model
english_MA1 <- rma.mv(
  yi = SMD, V = vSMD,
  random = list(~1 | StudyNo, ~1 | EffectID),
  test = "t", data = english)

# Step 1: Fit the fixed effect model
english_MA2 <- rma.mv(
  yi = SMD, V = vSMD, data = english, test = "t")

english_MA3 <- rma(
  yi = SMD, vi = vSMD, data = english,
  test = "t", method = "FE")

# Step 2: Correct for dependency
english_MA2_1 <- robust(
  english_MA2, cluster = english$StudyNo)

# Step 3: Testing modified eggers
english_MA4 <- rma.mv(
  yi = SMD, V = vSMD, mod = ~vSMD,
  random = list(~1 | StudyNo, ~1 | EffectID),
  test = "t", data = english)
```

```
# Now plot the results
plot <- orchard_plot(
  english_MA1, group = "StudyNo",
  xlab = "Standardized Mean Difference")
plot2 <- pub_bias_plot(plot, english_MA2_1)
plot3 <- pub_bias_plot(
  plot, english_MA2_1, english_MA4)
```

R2_calc

R2_calc

Description

Calculated R2 (R-squared) for mixed (multilevel) models, based on Nakagawa & Schielzeth (2013).

Usage

```
R2_calc(model)
```

Arguments

model Model object of class `rma.mv` or `rma`.

Value

A named numeric vector of length two: `R2_marginal`, the proportion of variance explained by the fixed effects (moderators) alone, and `R2_conditional`, the proportion explained by both the fixed and random effects. Values range from 0 to 1.

Author(s)

Shinichi Nakagawa - s.nakagawa@unsw.edu.au

Daniel Noble - daniel.noble@anu.edu.au

Examples

```
data(lim)
lim$vi<-(1/sqrt(lim$N - 3))^2
lim_MR<-metafor::rma.mv(yi=yi, V=vi, mods=~Phylum-1,
  random=list(~1|Article, ~1|Datapoint), data=lim)
R2 <- R2_calc(lim_MR)
```

r2_ml

r2_ml

Description

R2 (R-squared) for mixed (multilevel) models, based on Nakagawa & Schielzeth (2013).

Usage

```
r2_ml(model, data, boot = NULL)
```

Arguments

model	Model object of class <code>rma.mv</code> or <code>rma</code> .
data	Data frame used to fit the <code>rma.mv</code> or <code>rma</code> model object
boot	The number of parametric bootstrap iterations, if desired. Defaults to <code>NULL</code> . A setting of 1000 is recommended as a minimum number of iterations.

Value

A data frame containing all model results, including: mean effect size estimate, confidence and prediction intervals, with estimates converted back to `r`.

Author(s)

Shinichi Nakagawa - s.nakagawa@unsw.edu.au

Daniel Noble - daniel.noble@anu.edu.au

References

Nakagawa, S, and Schielzeth, H. 2013. A general and simple method for obtaining R2 from generalized linear mixed-effects models. **Methods in Ecology and Evolution** 4(2): 133-142.

Examples

```
data(lim)
lim$vi <- (1/sqrt(lim$N - 3))^2
lim_MR <- metafor::rma.mv(
  yi = yi, V = vi, mods = ~ Phylum - 1,
  random = list(~1 | Article, ~1 | Datapoint),
  data = lim)
R2 <- r2_ml(lim_MR, data = lim)
```

ratio_i2

*ratio_i2***Description**

I2 (I-squared) for mulilevel meta-analytic models based on Nakagawa & Santos (2012). Under multilevel models, we can have a multiple I2 (see also Senior et al. 2016).

Usage

```
ratio_i2(model)
```

Arguments

`model` Model object of class `rma.mv` or `rma`.

Value

A named numeric vector of I2 (I-squared) values expressed as percentages. The first element, `I2_Total`, is the total heterogeneity; each remaining element (named `I2_<level>` after a random-effect level of the model) is the heterogeneity attributable to that level.

Examples

```
library(metafor)
data(english)
english <- escalc(measure = "SMD", nli = NStartControl,
sd1i = SD_C, m1i = MeanC, n2i = NStartExpt,
sd2i = SD_E, m2i = MeanE, var.names=c("SMD","vSMD"),data = english)
english_MA <- rma.mv(yi = SMD, V = vSMD,
random = list(~ 1 | StudyNo, ~ 1 | EffectID), data = english)
ratio_i2(english_MA)
```

submerge

*submerge***Description**

Merge two model results tables (orchard objects).

Usage

```
submerge(object1, object2, ..., mix = FALSE)
```

Arguments

- object1 object of class orchard.
- object2 object of class orchard.
- ... Other arguments passed to submerge.
- mix If TRUE, it will add the number to the moderator name.

Value

Returns a data frame.

Author(s)

Shinichi Nakagawa - s.nakagawa@unsw.edu.au
Daniel Noble - daniel.noble@anu.edu.au

VCV_dARR	<i>VCV_dARR</i>
----------	-----------------

Description

Sampling error matrix used in Pottier et al. (2021) meta-analysis on sex-differences in acclimation

Format

A sampling variance matrix :

References

Pottier et al. 2022. Functional Ecology

weighted_var	<i>weighted_var</i>
--------------	---------------------

Description

Calculate weighted variance

Usage

weighted_var(x, weights)

Arguments

- x A vector of tau2s to be averaged
- weights Weights, or sample sizes, used to average the variance

Value

Returns a vector with a single weighted variance

Author(s)

Shinichi Nakagawa - s.nakagawa@unsw.edu.au

Daniel Noble - daniel.noble@anu.edu.au

*Zr_to_r**Zr_to_r*

Description

Converts Zr back to r (Pearson's correlation coefficient)

Usage

```
Zr_to_r(df)
```

Arguments

df data frame of results of class 'orchard'

Value

A data frame containing all the model results including mean effect size estimate, confidence and prediction intervals with estimates converted back to r

Author(s)

Shinichi Nakagawa - s.nakagawa@unsw.edu.au

Daniel Noble - daniel.noble@anu.edu.au

Index

- * **intern**
 - .colour_blind_palette, [4](#)
 - .MSb, [5](#)
 - .MSw, [5](#)
 - .calc.kurtosis, [3](#)
 - .calc.skewness, [3](#)
 - .colour_blind_palette, [4](#)
 - .lnM, [4](#)
 - .r.to.zr, [6](#)
 - .safe_lnM_dep, [6](#)
 - .safe_lnM_indep, [7](#)
 - .sv_jack, [8](#)
 - .v_lnM, [9](#)
 - .zr.variance, [9](#)
- bubble_plot, [10](#)
- caterpillars, [13](#)
- cor_diff, [15](#)
- cvh1_ml, [16](#)
- cvh2_ml, [17](#)
- eklof, [18](#)
- english, [19](#)
- firstup, [20](#)
- fish, [20](#)
- geom_pub_stats_naka, [21](#)
- geom_pub_stats_yang, [22](#)
- get_data_raw, [23](#)
- get_data_raw_cont, [24](#)
- get_ints_dat, [25](#)
- glmmTMB_to_rma, [25](#)
- i2_ml, [27](#)
- leave_one_out, [28](#)
- lim, [29](#)
- m1_ml, [30](#)
- m2_ml, [31](#)
- magnitude_effects, [32](#)
- matrix_i2, [34](#)
- ml_cvh1, [34](#)
- ml_cvh2, [35](#)
- ml_m1, [35](#)
- ml_m2, [36](#)
- mod_results, [36](#)
- moment_effects, [39](#)
- num_studies, [40](#)
- orchard_leave1out, [41](#)
- orchard_plot, [43](#)
- phylo_matrix, [46](#)
- pottier, [46](#)
- pred_interval_esmeans, [50](#)
- print.orchard, [50](#)
- pub_bias_plot, [51](#)
- R2_calc, [53](#)
- r2_ml, [54](#)
- ratio_i2, [55](#)
- submerge, [55](#)
- VCV_dARR, [56](#)
- weighted_var, [56](#)
- Zr_to_r, [57](#)