# Package 'metaselection'

August 31, 2026

**Title** Meta-Analytic Selection Models for Dependent Effect Sizes

**Version** 0.3.0

**Description**
Fits a flexible class of p-value selection models for meta-analysis and meta-regression models, providing standard errors and confidence intervals based on either cluster-robust variance estimators (i.e., sandwich estimators) or cluster-level bootstrapping to handle dependent effect size estimates, as described in Pustejovsky, Citkowicz, and Joshi (2025) <DOI:10.31222/osf.io/qg5x6_v1> and Citkowicz, Pustejovsky, and Joshi (2026) <DOI:10.31222/osf.io/wjpxk_v1>. Supported models include generalizations of the step-function selection model as proposed by Vevea and Hedges (1995) <DOI:10.1007/BF02294384> and the beta-function selection model as proposed by Citkowicz and Vevea (2017) <DOI:10.1037/met0000119>.

**License** GPL (>= 3)

**Encoding** UTF-8

**Depends** R (>= 4.1.0)

**Imports** Formula, stats, utils, MASS, mvtnorm, optimx, nleqslv, purrr, future.apply, progressr, rlang, ggplot2 (>= 3.5.0), scales, Rdpack, simhelpers (>= 0.3.1)

**Suggests** testthat (>= 3.0.0), future, metafor (>= 4.8-0), metadat, clubSandwich, DescTools, knitr, rmarkdown, bookdown, dplyr

**RdMacros** Rdpack

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**LazyData** true

**URL** https://github.com/jepusto/metaselection

**Config/roxygen2/version** 8.0.0

**Language** en-US

**NeedsCompilation** no

**Author** James E. Pustejovsky [aut, cre] (ORCID:
<https://orcid.org/0000-0003-0591-9465>),
Megha Joshi [aut] (ORCID: <https://orcid.org/0000-0001-7936-076X>),
Martyna Citkowicz [aut] (ORCID:
<https://orcid.org/0000-0002-2048-0625>)

**Maintainer** James E. Pustejovsky <jepusto@gmail.com>

**Repository** CRAN

**Date/Publication** 2026-08-31 14:10:02 UTC

# Contents



---

`beta_fun` *Censor meta-analytic dataset based on the univariate beta-density model*

---

## Description

A functional that takes model parameters and returns a function that can be used to censor meta-analytic datasets according to the univariate beta-density model.

## Usage

```
beta_fun(
  delta_1 = 1,
  delta_2 = 1,
  trunc_1 = 0.025,
  trunc_2 = 0.975,
  renormalize = TRUE
)
```

## Arguments

| | |
|---|---|
| delta_1 | numeric value for the first parameter of the beta function. |
| delta_2 | numeric value for the second parameter of the beta function. |
| trunc_1 | numeric value between 0 and 1, below which p-values will be truncated. |
| trunc_2 | numeric value between 0 and 1, above which p-values will be truncated. |
| renormalize | logical indicating whether to normalize the beta function to have a maximum value of 1, with a default value of TRUE. |

## Value

A function that can be used to censor a meta-analytic dataset based on the univariate beta-density model.

---

# define_priors *Define prior penalty functions for selection model parameters*

---

## Description

Creates a set of priors for use in estimating selection models. beta parameters are assigned L-norm priors. log(tau) parameters are assigned independent log-gamma priors. log(lambda) parameters are assigned independent L-norm priors.

## Usage

```
define_priors(
  beta_mean = 0,
  beta_precision = 1/16,
  beta_L = 4,
  tau_mode = 0.2,
  tau_alpha = 1,
  lambda_mode = 0.5,
  lambda_precision = 1/54,
  lambda_L = 4
)
```

## Arguments

| | |
|---|---|
| beta_mean | numeric vector of prior means for beta (mean regression) parameters. |
| beta_precision | numeric vector of prior precisions for beta (mean regression) parameters. |
| beta_L | numeric vector of prior norms for beta (mean regression) parameters. |
| tau_mode | numeric vector of prior modes for tau (heterogeneity SD) regression parameters. |
| tau_alpha | numeric vector of prior precisions for tau (heterogeneity SD) regression parameters. |
| lambda_mode | numeric vector of prior modes for lambda (selection) parameters. |
| lambda_precision | numeric vector of prior precisions for lambda (selection) parameters. |
| lambda_L | numeric vector of prior norms for lambda (mean regression) parameters. |

## Value

An object of class "selmodel_prior" containing the following components:

log_prior function with arguments beta,gamma,zeta that returns the log of the prior density over these parameters.

score_prior function with arguments beta,gamma,zeta that returns the vector of scores for the prior density over these parameters.

hessian_prior function with arguments beta,gamma,zeta that returns the Hessian matrix of the prior density over these parameters.

## Examples

```
# set very informative priors on beta and lambda
strong_priors <- define_priors(
  beta_mean = 0.4, beta_precision = 40,
  lambda_mode = 0.2, lambda_precision = 40
)

# set standard normal prior on beta
weak_priors <- define_priors(
  beta_mean = 0, beta_precision = 1/2, beta_L = 2
)
```

---

# interleaved_learning *Interleaved Learning Meta-Analysis*

---

## Description

Meta-analytic dataset containing results of primary studies examining the effect of interleaved learning.

## Usage

```
interleaved_learning
```

## Format

A data frame with 238 rows and 21 variables:

**study** name of the study.

**esid** identifier for the effect size.

**g** effect size in form of Hedges' g.

**vg** corresponding variance of the effect size.

**sample_id** identifier for each sample within a study.

**article_num** identifier for the publication.

**item_num** code for items with (1) Paintings including mostly impressionistic paintings of different artists coded as 0; (2) Naturalistic photographs such as pictures of birds, butterflies coded as 3; (3) Artificial pictures, that is, pictures of artificial objects or creatures coded as 1; (4) Mathematical tasks, such as calculating the volume of geometric solids or the use of significance tests coded as 2; (5) Expository Texts, which included plain expository texts and combinations of texts and other media in a multimedia/interactive media environment coded as 5; (6) Words, such as names that belonged to different conceptual categories, pronunciation rules, or translations in different languages coded as 4; (7) Tastes such as liquids with different tastes coded as 6.

**age** mean age of the participants.

**grey_lit** indicator for grey literature with 1 indicating theses and dissertations and 0 indicating articles and conference papers.

**design** indicator for research design with 1 indicating within-participants design and 0 indicating between-participants design.

**students** indicator type of sample with 1 indicating samples with only students and 0 indicating all other types of samples.

**retention_interval** indicator for retention interval length with 1 indicating long (>= 20 min) intervals and 0 indicating short (< 20 min) intervals.

**intentionality** indicator for intentional learning designs with 1 indicating incidental learning designs and 0 indicating intentional learning designs.

**transfer_retention** indicator for type of tests with 1 indicating transfer tests and 0 indicating retention tests.

**simultaneity** indicator for type of presentation with 1 indicating simultaneous presentation and 0 indicating successive presentation.

**zmean_within** standardized ratings of similarity within categories.

**zmean_between** standardized ratings of similarity between categories.

**zmean_complex** standardized ratings of complexity.

**zmean_fam** standardized ratings of familiarity.

**zmean_cur** standardized ratings of curiosity.

**spaced** indicator for spacing with spaced indicating designs with spaced items (10 or 30 sec time intervals), nonspaced indicating designs with immediate succession of items (less than 2 sec), -1 indicating designs with distractors between presentation of items are coded as distract and items not included in the analysis of spacing between items.

## Source

OSF page for the project

## References

Brunmair M, Richter T (2019). "Similarity matters: A meta-analysis of interleaved learning and its moderators." *Psychological Bulletin*, **145**(11), 1029. doi:10.1037/bul0000209.

---

`n_ES_empirical` *Simulate empirical distribution of sample size and number of effect sizes*

---

## Description

A functional that takes in a dataset with empirical distribution of primary study sample sizes and number of effect sizes per primary study and returns a function that generates random samples from the dataset

## Usage

```
n_ES_empirical(dat)
```

## Arguments

| | |
|---|---|
| `dat` | a `data.frame` or `tibble` containing primary study sample sizes and number of effect sizes per primary study. |

## Value

A function that generates random samples from the input dataset.

## Examples

```
study_features <- n_ES_empirical(wwc_es)
study_features(m=3)
study_features(m=7)
```

## n_ES_param *Simulate empirical distribution of sample size and number of effect sizes*

### Description

A functional that takes in average sample size per primary study, average number of effect sizes per study, and the minimum sample size per study and returns a function to generate random samples of primary study sample sizes and numbers of effect sizes per study.

### Usage

```
n_ES_param(mean_N, mean_ES, min_N = 20L)
```

### Arguments

| | |
|---|---|
| mean_N | numeric value specifying the average sample size per primary study. |
| mean_ES | numeric value specifying the average number of effect sizes per primary study. |
| min_N | numeric value specifying the minimum sample size per study. |

### Value

A function that generates a `data.frame` with randomly generated sample size per primary study and number of effect sizes per study.

### Examples

```
study_features <- n_ES_param(mean_N = 40, mean_ES = 3, min_N = 10)
study_features(m = 3)
study_features(m = 5)
```

## practice_facilitation *Practice Facilitation Meta-Analysis*

### Description

Meta-analytic dataset containing results of primary studies examining the effect of practice facilitation on the uptake of evidence-based practices (EBPs) in primary care settings.

### Usage

```
practice_facilitation
```

## Format

A data frame with 23 rows and 3 variables:

**author** first author and publication year of primary study report.

**score** score on a scale from 0 to 12, for which higher scores correspond to higher quality of the study methods.

**design** study design, with CCT = controlled clinical trial, C-RCT = cluster randomized controlled trial, RCT = randomized controlled trial.

**allocation_concealed** indicator for allocation concealment.

**blinded** indicator for whether study was single- or double-blinded.

**intent_to_treat** indicator for whether study adhered to intent-to-treat principle.

**outcome** description of outcome measure.

**follow_up** months of follow-up.

**retention_pct** percentage of sample retained at follow-up.

**SMD** effect size in form of Hedges' g.

**SE** corresponding variance of the effect size.

## Source

Table 1 of Baskerville et al. (2012; doi:10.1370/afm.1312).

## References

Baskerville NB, Liddy C, Hogg W (2012). "Systematic review and meta-analysis of practice facilitation within primary care settings." *Annals of Family Medicine*, **10**(1), 63–74. ISSN 1544-1717, doi:10.1370/afm.1312.

---

`print.selmodel` *Print results from a* `selmodel` *object*

---

## Description

Print relevant results from a fitted `selmodel` object.

## Usage

```
## S3 method for class 'selmodel'
print(x, transf_gamma = TRUE, transf_zeta = TRUE, digits = 3, ...)
```

### Arguments

| | |
|---|---|
| x | fitted model of class "selmodel". |
| transf_gamma | logical with TRUE (the default) indicating that the heterogeneity parameter estimates (called gamma) should be transformed by exponentiating. |
| transf_zeta | logical with TRUE (the default) indicating that the selection parameter estimates (called zeta) should be transformed by exponentiating. |
| digits | minimum number of significant digits to be used, with a default of 3. |
| ... | further arguments passed to print.data.frame(). |

### Value

The method returns a data.frame containing parameter estimates, standard errors, p-values, and confidence intervals for model parameters.

### Examples

```
res_ML <- selection_model(
  data = self_control,
  yi = g,
  sei = se_g,
  cluster = studyid,
  steps = 0.025,
  estimator = "CML",
  bootstrap = "none"
)

print(res_ML)
print(res_ML, transf_gamma = FALSE, transf_zeta = FALSE)
```

---

## p_area *Calculate area under the selection weight function from a* selmodel *object*

---

### Description

Summarize the strength of selection by calculating the area under the selection weight function a selmodel object (excluding the area with weight fixed at 1). If the object has bootstrap replications, then a confidence interval will also be calculated.

### Usage

```
p_area(object, CI_type = NULL, conf_level = NULL, warn = TRUE)
```

## Arguments

| | |
|---|---|
| object | fitted model of class "selmodel". |
| CI_type | character string specifying the type of confidence interval to calculate, with options as in "selection_model". If NULL (the default), it will be inherited from object. |
| conf_level | desired coverage level for confidence intervals. If NULL (the default), it will be inherited from object, which has a default value of .95. |
| warn | logical controlling whether warnings are displayed, with a default of TRUE. |

## Value

A data.frame containing the estimated area under the selection weight function. If the input object includes bootstraps, then the returned data.frame also includes bootstrap confidence interval(s) for the area under the selection weight function.

## Examples

```
beta_noboot <- selection_model(
  data = practice_facilitation,
  yi = SMD,
  sei = SE,
  selection_type = "beta",
  steps = c(0.025,0.975)
)

p_area(beta_noboot)

step_boot <- selection_model(
  data = self_control,
  yi = g,
  sei = se_g,
  cluster = studyid,
  selection_type = "step",
  steps = c(0.025,0.50),
  estimator = "ARGL",
  bootstrap = "multinomial",
  CI_type = "normal",
  R = 9L
)

p_area(step_boot)
```

---

# r_meta *Generate meta-analytic data*

---

## Description

Generate meta-analytic correlated or correlated and hierarchical effects data with options to simulate selective outcome reporting

## Usage

```
r_meta(
  mean_smd,
  tau,
  omega,
  m,
  cor_mu,
  cor_sd,
  n_ES_sim,
  censor_fun = NULL,
  m_multiplier = 1,
  id_start = 0L,
  paste_ids = TRUE,
  include_sel_prob = FALSE
)
```

## Arguments

| | |
|---|---|
| mean_smd | numeric value indicating the true mean effect size. |
| tau | numeric value characterizing between-study heterogeneity in effects. |
| omega | numeric value characterizing within-study heterogeneity in effects. |
| m | numeric value of studies in the simulated meta-analysis. |
| cor_mu | numeric value indicating the average correlation between outcomes. |
| cor_sd | numeric value indicating standard deviation of correlation between outcomes. |
| n_ES_sim | a function used to simulate the distribution of primary study sample sizes and the number of effect sizes per study. |
| censor_fun | a function used to censor effects. The package provides functionals `step_fun()` and `beta_fun()` to censor effects based on step-function or beta-function models respectively. If `NULL` (the default), then all generated effect size estimates will be included (i.e., without censoring). |
| m_multiplier | numeric value for a multiplier to buffer the number of studies generated. |
| id_start | integer indicating the starting value for study id. |
| paste_ids | logical with `TRUE` (the default) indicating that the study id and effect size id should be pasted together. |
| include_sel_prob | logical with `TRUE` indicating that the returned dataset should include a variable `selection_prob` reporting the true probability of selection given the observed p-value. Default of `FALSE` indicates that the `selection_prob` variable should be omitted. |

## Value

A `data.frame` containing the simulated meta-analytic dataset.

## Examples

```
example_dat <- r_meta(
  mean_smd = 0,
  tau = .1, omega = .01,
  m = 50,
  cor_mu = .4, cor_sd = 0.001,
  censor_fun = step_fun(cut_vals = .025, weights = 0.4),
  n_ES_sim = n_ES_param(40, 3)
)
```

---

# selection_model *Estimate step or beta selection model*

---

## Description

Estimate step or beta selection model, with standard errors and confidence intervals based on either cluster-robust variance estimators (i.e., sandwich estimators) or cluster-level bootstrapping to handle dependent effect size estimates.

## Usage

```
selection_model(
  data,
  yi,
  vi,
  sei,
  ai,
  cluster,
  selection_type = c("step", "beta"),
  alternative = "greater",
  steps = NULL,
  mean_mods = NULL,
  var_mods = NULL,
  sel_mods = NULL,
  sel_zero_mods = NULL,
  priors = define_priors(),
  subset = NULL,
  estimator = "CML",
  vcov_type = "robust",
  CI_type = "large-sample",
  conf_level = 0.95,
  theta = NULL,
```

```
  optimizer = NULL,
  optimizer_control = list(),
  use_jac = NULL,
  bootstrap = "none",
  R = 1999,
  retry_bootstrap = 0L,
  valence_check = TRUE,
  ...
)
```

## Arguments

| | |
|---|---|
| data | data.frame or tibble containing the meta-analytic data. |
| yi | vector of effect sizes estimates. |
| vi | vector of sample variances. If vi is specified, then the sei argument must be omitted. |
| sei | vector of sampling standard errors. If sei is specified, then the vi argument must be omitted. |
| ai | optional vector of analytic weights. |
| cluster | vector indicating which observations belong to the same cluster. |
| selection_type | character string specifying the type selection model to estimate, with possible options "step" or "beta". |
| alternative | character string specifying the direction of the alternative hypothesis used in computing p-values for the observed effect sizes, with possible options "greater" (the default) or "less". |
| steps | If selection_type = "step", a numeric vector of one or more values specifying the thresholds (or steps) where the selection probability changes, with a default of steps = .025. If selection_type = "beta", then a numeric vector of two values specifying the thresholds beyond which the selection function is truncated, with a default of steps = c(.025, .975). |
| mean_mods | optional model formula for moderators related to average effect size magnitude. |
| var_mods | optional model formula for moderators related to effect size heterogeneity. |
| sel_mods | optional model formula for moderators related to the probability of selection. Only relevant for selection_type = "step". |
| sel_zero_mods | optional model formula for moderators related to the probability of selection for p-values below the lowest threshold value of steps. Only relevant for selection_type = "step". |
| priors | a selmodel_prior object that defines priors (i.e., penalty terms) for model parameters, with a default of define_priors(). Set to NULL to obtain unpenalized estimates. |
| subset | optional logical expression indicating a subset of observations to use for estimation. |

estimator vector indicating whether to use the composite marginal likelihood estimator (option "CML") or the augmented and reweighted Gaussian likelihood estimator (option "ARGL" or "ARGL-full"). If selection_type = "beta", only the composite marginal likelihood estimator, "CML", is available. For step function models, both estimators are available.

vcov_type character string specifying the type of variance-covariance matrix to calculate, with possible options "robust" for robust or cluster-robust standard errors, "model-based" for model-based standard errors, or "none".

CI_type character string or vector specifying the type of confidence interval to calculate, with possible options "large-sample" for large-sample normal interval (the default), "percentile" for a percentile interval, "BCa" for a bias-corrected-and-accelerated interval, "bias-corrected" for a bias-corrected percentile interval (without acceleration correction), "normal" for a standard normal interval, "basic" for a basic interval, "student" for a studentized interval, or "none". More than one type of interval can be computed by specifying a character vector with multiple options.

conf_level desired coverage level for confidence intervals, with the default value set to .95.

theta optional numeric vector of starting values to use in optimization routines.

optimizer character string indicating the optimizer to use. Ignored if estimator = "ARGL" or "ARGL-full".

optimizer_control
an optional list of control parameters to be used for optimization

use_jac logical indicating whether to use the Jacobian of the estimating equations for optimization. If NULL (the default), it will be reset to FALSE if estimator = "CML" or to TRUE if estimator = "ARGL"

bootstrap character string specifying the type of bootstrap to run, with possible options "none" (the default), "exponential" for the fractionally re-weighted cluster bootstrap, "multinomial" for a conventional clustered bootstrap, or , "two-stage" for a two-stage clustered bootstrap.

R number of bootstrap replications, with a default of 1999.

retry_bootstrap
number of times to re-draw a bootstrap sample in the event of non-convergence, with a default of 0.

valence_check logical value controlling whether to check that the valence of the median effect size estimate is consistent with the direction of the specified alternative. If TRUE (the default), a warning will be issued when most effect size estimates have the opposite sign of alternative. Set to FALSE to suppress the warning.

... further arguments passed to simhelpers::bootstrap_CIs.

## Value

An object of class "selmodel" containing the following components:

est data.frame with parameter estimates, standard errors, and confidence intervals. Note that the results do not include p-values so as to focus interpretation on the parameter estimates, rather than on the statistical significance of any given parameter.

`vcov` matrix containing the estimated variance-covariance matrix of the parameter estimates.

`method` character string indicating the optimization method used to solve for parameter estimates.

`info` further information about the optimization results.

`ll` log likelihood of the model evaluated at the reported parameter estimates.

`wpll` weighted partial log likelihood of the random effects model, with weights corresponding to inverse selection probabilities.

`n_clusters` number of independent clusters of effect sizes.

`n_effects` number of effect size estimates in the data.

`...` some additional elements containing information about the methods used to estimate the model.

## Examples

```
res_ML <- selection_model(
  data = self_control,
  yi = g,
  sei = se_g,
  cluster = studyid,
  steps = 0.025,
  estimator = "CML",
  bootstrap = "none"
)

res_ML
summary(res_ML)

# configure progress bar
progressr::handlers(global = TRUE)

res_hybrid <- selection_model(
  data = self_control,
  yi = g,
  sei = se_g,
  cluster = studyid,
  steps = 0.025,
  estimator = "ARGL",
  bootstrap = "multinomial",
  CI_type = "percentile",
  R = 19
)

res_hybrid
summary(res_hybrid)
```

`selection_plot` *Plot the selection weights implied by an estimated selection model.*

## Description

For a fitted model of class `"selmodel"`, create a plot of the selection weights implied by the model parameter estimates. If the model includes bootstrapped confidence intervals, then the plot will also display the selection weights implied by each bootstrap replicate of the parameter estimates.

## Usage

```
selection_plot(
  mod,
  limits = c(0, 1),
  pts = 200L,
  ref_pval = NULL,
  transform = "identity",
  expand = ggplot2::expansion(0, 0.01),
  ...
)

## S3 method for class 'selmodel'
selection_plot(
  mod,
  limits = c(0, 1),
  pts = 200L,
  ref_pval = NULL,
  transform = "identity",
  expand = ggplot2::expansion(0, 0.01),
  fill = "blue",
  alpha = 0.5,
  step_linetype = "dashed",
  ...
)

## S3 method for class 'boot.selmodel'
selection_plot(
  mod,
  limits = c(0, 1),
  pts = 200L,
  ref_pval = NULL,
  transform = "identity",
  expand = ggplot2::expansion(0, 0.01),
  color = "black",
  linewidth = 1.2,
  step_linetype = "dashed",
  draw_boots = TRUE,
```

```
  fill = "blue",
  alpha = 0.5,
  boot_color = "blue",
  boot_alpha = 0.1,
  ...
)
```

## Arguments

| | |
|---|---|
| mod | fitted model of class "selmodel". |
| limits | numeric vector of length 2 specifying the minimum and maximum p-values to plot. |
| pts | number of points for which to calculate selection weights, with a default of 200 points, evenly spaced between the specified limits. |
| ref_pval | numeric value of a p-value at which to standardize the weights. If not NULL, then a p-value of ref_pval will have selection weight of 1 and selection weights for all other p-values will be calculated relative to ref_pval. |
| transform | character string specifying the name of a transformation function or the transformation function itself, as defined in the scales package. The transform is passed to ggplot2::scale_x_continuous. The default transform is "identity". Other useful transforms for p-values are "sqrt" for square-root or "asn" for the arcsin square root. |
| expand | passed to the expand argument of ggplot2::scale_x_continuous. |
| ... | further arguments passed to ggplot2::scale_x_continuous. |
| fill | character string specifying the fill-color to use when mod does not include bootstrap replications, with a default of "blue". Passed to ggplot2::geom_area(). |
| alpha | numeric value specifying the opacity of the filled area plot, with a default of 0.5. Passed to ggplot2::geom_area(). Only used when mod does not include bootstrap replications. |
| step_linetype | character string specifying the type of line to draw to indicate p-value thresholds assumed in mod. |
| color | character string specifying the line color to use for drawing the estimated selection weights, with a default of "black". Passed to ggplot2::geom_line(). Only used when mod includes bootstrap replications. |
| linewidth | numeric value specifying the line width to use for drawing the estimated selection weights, with a default of 1.2. Passed to ggplot2::geom_line(). Only used when mod includes bootstrap replications. |
| draw_boots | logical value indicating whether to draw the selection weights for each bootstrap replication, with a default of TRUE. |
| boot_color | character string specifying the line color to use for drawing the selection weights of each bootstrap replication, with a default of "blue". Passed to ggplot2::geom_line(). Only used when mod includes bootstrap replications. |
| boot_alpha | numeric value specifying the opacity of the lines for drawing the selection weights of each bootstrap replication, with a default of "blue". Passed to ggplot2::geom_line(). Only used when mod includes bootstrap replications. |

## Value

A ggplot2 object.

## Examples

```
mod <- selection_model(
  data = self_control,
  yi = g,
  sei = se_g,
  cluster = studyid,
  steps = c(0.025, .5),
  estimator = "ARGL",
  bootstrap = "none"
)

selection_plot(mod, fill = "purple")

# rescale the horizontal axis using arc-sin square root
selection_plot(mod, fill = "purple", transform = "asn")


mod_boot <- selection_model(
  data = self_control,
  yi = g,
  sei = se_g,
  cluster = studyid,
  steps = c(0.025, .5),
  estimator = "ARGL",
  bootstrap = "multinomial",
  CI_type = "percentile",
  R = 9L
)

 selection_plot(mod_boot, transform = "sqrt")
 selection_plot(mod_boot, transform = "sqrt", draw_boots = FALSE) # turn off bootstrap lines
 selection_plot(mod_boot, transform = "sqrt", color = "red", boot_color = "orange") # change colors
```

---

# selection_wts *Calculate model-implied weights for specified p-values.*

---

## Description

Calculates the selection weights implied by an estimated model for a user-specified p-value or set of p-values.

## Usage

```
selection_wts(mod, pvals, ref_pval, ...)

## S3 method for class 'step.selmodel'
selection_wts(mod, pvals = NULL, ref_pval = NULL, bootstraps = TRUE, ...)

## S3 method for class 'beta.selmodel'
selection_wts(mod, pvals = NULL, ref_pval = NULL, bootstraps = TRUE, ...)
```

## Arguments

| | |
|---|---|
| mod | fitted model of class "selmodel". |
| pvals | numeric vector of p-values for which to calculate selection weights. |
| ref_pval | numeric value of a p-value at which to standardize the weights. If not NULL, then a p-value of ref_pval will have selection weight of 1 and selection weights for all other p-values will be calculated relative to ref_pval. |
| ... | further arguments passed to some methods. |
| bootstraps | If mod includes bootstrap replications, then setting bootstraps = TRUE will return selection weights for each bootstrap replication, in addition to the selection weights implied by the model parameter estimates. Ignored if mod does not include bootstrap replications. |

## Value

If mod does not include bootstrapped confidence intervals or if the argument bootstraps = FALSE, then selection_wts will return a data.frame containing the user-specified p-values and the selection weights implied by the estimated model parameters.

If mod does include bootstrapped confidence intervals (i.e., when inherits(mod, "boot.selmodel") is TRUE) and the argument bootstraps = TRUE, then selection_wts will return a list with two elements. The first element is a data.frame containing the user-specified p-values and the selection weights implied by the estimated model parameters. The second element is a data.frame containing the user-specified p-values and the selection weights implied by each bootstrap replicate of the model parameter estimates. The data.frame includes an additional variable, rep, identifying the bootstrap replicate.

## Examples

```
mod <- selection_model(
  data = self_control,
  yi = g,
  sei = se_g,
  cluster = studyid,
  steps = c(0.025, .5),
  estimator = "ARGL"
)

selection_wts(mod, pvals = seq(0, 1, 0.2))
```

```
mod_boot <- selection_model(
  data = self_control,
  yi = g,
  sei = se_g,
  cluster = studyid,
  steps = c(0.025, .5),
  estimator = "ARGL",
  bootstrap = "multinomial",
  CI_type = "percentile",
  R = 9
)

selection_wts(mod_boot, pvals = seq(0, 1, 0.2))
```

---

`self_control` *Self-Control Training Meta-Analysis*

---

## Description

Meta-analytic dataset containing results of primary studies examining the effect of self-control training.

## Usage

```
self_control
```

## Format

A data frame with 158 rows and 26 variables:

**studyid** identifier for the study.

**esid** identifier for the effect size.

**name** name of the study.

**g** effect size in form of Hedges' g.

**var_g** corresponding variance of the effect size.

**se_g** corresponding standard error of the effect size.

**outcome** name of the outcome measure.

**comparison** conditions compared.

**type_of_treatment** type of treatment/training.

**length_of_treatment** length of treatment coded in days.

**publication_status** indicator for whether the study was published.

**publication_status_new** indicator for whether the study was published (newer version).

**publication_year** year that the study was published.

**research_group** indicator for whether researchers of the study belonged to the strength model group.

**control_group_quality** quality of the control group with active indicating that the control group worked on some task.

**gender_ratio** percentage of males in the sample.

**type_of_outcome** type of outcome.

**subjectivity_of_outcome_measurement** indicator for the subjectivity of the outcome measure.

**lab_based_versus_real_world_behavior** indicator for whether the behavior measured was assessed in lab or in the real-world.

**stamina_versus_strength** indicator for whether the outcome was assessed with (Stamina) or without a preceding effortful task (Strength).

**pre_test_measurement** indicator for whether there was a pre and post measure of outcome or only post measure.

**sample_population** population examined in the study.

**sample_age** average age of the sample.

**attrition** percentage of attrition.

**participant_compensation** type of compensation received by the participants.

**self_control_potential** indicator for whether the outcome measure required utilization of maximum self-control potential.

## Source

OSF page for the project

## References

Friese M, Frankenbach J, Job V, Loschelder DD (2017). "Does self-control training improve self-control? A meta-analysis." *Perspectives on Psychological Science*, **12**(6), 1077–1099. doi:10.1177/1745691617697076.

---

`step_count_fun` *Censor meta-analytic dataset based on a multivariate step-function model*

---

## Description

A functional that takes in a single step value, a weight representing the selection probability for the upper interval of p-values, and a dependence parameter, and returns a function that can be used to censor meta-analytic datasets according to a multivariate step-function model.

## Usage

```
step_count_fun(cut_val = 0.025, weight = 1, psi = 0, renormalize = TRUE)
```

### Arguments

| | |
|---|---|
| cut_val | numeric value specifying the specifying the thresholds (or steps) where the selection probability changes. |
| weight | numeric value specifying the selection probability for the upper interval of p-values, i.e., for p-values larger than cut_val for an effect reported in a study with no p-values smaller than cut_val. |
| psi | numeric value controlling the degree of dependence in selection probabilities. |
| renormalize | logical indicating whether to normalize the step function to have a maximum value of 1, with a default value of TRUE. |

### Value

A function that can be used to censor a meta-analytic dataset based on a multivariate step-function model.

---

## step_fun *Censor meta-analytic dataset based on a univariate step-function model*

---

### Description

A functional that takes in cut values and weights representing selection probabilities for different intervals of p-values and returns a function that can be used to censor meta-analytic datasets according to the univariate step-function model.

### Usage

```
step_fun(cut_vals = 0.025, weights = 1, renormalize = TRUE)
```

### Arguments

| | |
|---|---|
| cut_vals | numeric vector of one or more values specifying the threshold (or step) where the selection probability changes. |
| weights | numeric vector of one or more values specifying the selection probabilities for different intervals of p-values; the intervals are determined by the cut_vals. |
| renormalize | logical indicating whether to normalize the step function to have a maximum value of 1, with a default value of TRUE. |

### Value

A function that can be used to censor a meta-analytic dataset based on the univariate step-function model.

summary.selmodel *Summarize results from a* selmodel *object*

## Description

Summarize relevant results from a fitted selection model.

## Usage

```
## S3 method for class 'selmodel'
summary(object, transf_gamma = TRUE, transf_zeta = TRUE, digits = 3, ...)
```

## Arguments

| | |
|---|---|
| object | fitted model of class "selmodel". |
| transf_gamma | logical with TRUE (the default) indicating that the heterogeneity parameter estimates (called gamma) should be transformed by exponentiating. |
| transf_zeta | logical with TRUE (the default) indicating that the selection parameter estimates (called zeta) should be transformed by exponentiating. |
| digits | minimum number of significant digits to be used, with a default of 3. |
| ... | further arguments passed to print.data.frame(). |

## Details

The function outputs a summary of a fitted selmodel object to the console. Output includes information about the number of clusters and number of effect size estimates used to fit the model, estimator and variance estimator settings, model fit information, and parameter estimates with associated uncertainty measures.

## Value

The method does not return an object.

## Examples

```
res_ML <- selection_model(
  data = self_control,
  yi = g,
  sei = se_g,
  cluster = studyid,
  steps = 0.025,
  estimator = "CML",
  bootstrap = "none"
)

summary(res_ML)
summary(res_ML, transf_gamma = FALSE, transf_zeta = FALSE)
```

`wwc_es` *What Works Clearinghouse sample size and effect size distribution data*

## Description

A dataset containing sample size and number of effect sizes seen in primary studies evaluated by the What Works Clearinghouse.

## Usage

```
wwc_es
```

## Format

A tibble with 615 rows and 2 variables:

**n** the sample size of the primary study.

**n_ES** number of effect sizes in the primary study.

## References

What Works Clearinghouse (2021). "Data from individual studies." https://ies.ed.gov/ncee/wwc/studyfindings. (visited on ).

# Index