

Package ‘weightr’

July 6, 2019

Title Estimating Weight-Function Models for Publication Bias

Version 2.0.2

Description Estimates the Vevea and Hedges (1995) weight-function model. By specifying arguments, users can also estimate the modified model described in Vevea and Woods (2005), which may be more practical with small datasets. Users can also specify moderators to estimate a linear model. The package functionality allows users to easily extract the results of these analyses as R objects for other uses. In addition, the package includes a function to launch both models as a Shiny application. Although the Shiny application is also available online, this function allows users to launch it locally if they choose.

URL <http://faculty.ucmerced.edu/jvevea/>,
<https://vevealab.shinyapps.io/WeightFunctionModel/>,
http://faculty.ucmerced.edu/jvevea/other/Vevea_and_Hedges_1995.pdf,
http://faculty.ucmerced.edu/jvevea/other/Vevea_and_Woods_Psy_Meth.pdf

Depends R (>= 3.3.0), ggplot2, scales

License GPL-2 | GPL-3

LazyData true

Suggests shiny (>= 0.12.2)

RoxygenNote 6.1.1.9000

NeedsCompilation no

Author Kathleen M. Coburn [aut, cre],
Jack L. Vevea [aut]

Maintainer Kathleen M. Coburn <kcoburn@ucmerced.edu>

Repository CRAN

Date/Publication 2019-07-06 18:00:10 UTC

R topics documented:

dat.bangertdrowns2004	2
dat.gatb	3
dat.smith	4
density	5
funnel	6
predict.weightfunct	7
print.weightfunct	8
shiny_weightr	9
weightfunct	9

Index	13
--------------	-----------

dat.bangertdrowns2004 *Studies on the Effectiveness of Writing-to-Learn Interventions*

Description

Results from 48 studies on the effectiveness of school-based writing-to-learn interventions on academic achievement.

Usage

```
dat.bangertdrowns2004
```

Format

A data frame; for documentation, see `dat.bangertdrowns2004` in Wolfgang Viechtbauer's R package `metafor`.

Details

This reproduced dataset and its documentation are credited to Wolfgang Viechtbauer and his `metafor` package (2010). Please see his package for details.

Source

Bangert-Drowns, R. L., Hurley, M. M., & Wilkinson, B. (2004). The effects of school-based writing-to-learn interventions on academic achievement: A meta-analysis. *Review of Educational Research*, 74, 29-58.

References

Bangert-Drowns, R. L., Hurley, M. M., & Wilkinson, B. (2004). The effects of school-based writing-to-learn interventions on academic achievement: A meta-analysis. *Review of Educational Research*, 74, 29-58.

Viechtbauer, W. (2010). Conducting meta-analysis in R with the `metafor` package. *Journal of Statistical Software*, 36(3), 1-48.

Examples

```
## Not run:
dat.bangertdrowns2004

# Extracting the effect sizes and sampling variances:
effect <- dat.bangertdrowns2004$yi
v <- dat.bangertdrowns2004$vi

# The weight-function model with no mean model:
weightfunct(effect, v)

# The weight-function model with a mean model:
weightfunct(effect, v, mods=~dat.bangertdrowns2004$info)

## End(Not run)
```

dat.gatb	<i>Studies of the Predictive Validity of the General Ability Subscale of the General Aptitude Test Battery (GATB)</i>
----------	---

Description

Results from 755 studies on the General Aptitude Test Battery's predictive validity of job performance (General Ability subscale).

Usage

```
dat.gatb
```

Format

A data frame containing the following columns:

z Fisher's z-transformed correlation coefficients

v corresponding sampling variance

Details

The General Aptitude Test Battery (GATB) is designed to measure nine cognitive, perceptual, and psychomotor skills thought relevant to the prediction of job performance. From 1947 to 1993, a total of 755 studies were completed in order to assess the validity of the GATB and its nine scales, and the GATB has been found to be a moderately valid predictor of job performance. This dataset consists of validity coefficients for the General Ability scale of the GATB.

Source

U.S. Department of Labor, Division of Counseling and Test Development, Employment and Training Administration. (1983a). The dimensionality of the General Aptitude Test Battery (GATB) and the dominance of general factors over specific factors in the prediction of job performance for the U.S. Employment Service (U.S. Employment Service Test Research Rep. No. 44). Washington, DC.

U.S. Department of Labor, Division of Counseling and Test Development, Employment and Training Administration. (1983b). Test validity for 12,000 jobs: An application of job classification and validity generalization analysis to the General Aptitude Test Battery (U.S. Employment Service Test Research Rep. No. 45). Washington, DC.

References

Vevea, J. L., Clements, N. C., & Hedges, L. V. (1993). Assessing the effects of selection bias on validity data for the General Aptitude Test Battery. *Journal of Applied Psychology*, 78(6), 981-987.

U.S. Department of Labor, Division of Counseling and Test Development, Employment and Training Administration. (1983a). The dimensionality of the General Aptitude Test Battery (GATB) and the dominance of general factors over specific factors in the prediction of job performance for the U.S. Employment Service (U.S. Employment Service Test Research Rep. No. 44). Washington, DC.

U.S. Department of Labor, Division of Counseling and Test Development, Employment and Training Administration. (1983b). Test validity for 12,000 jobs: An application of job classification and validity generalization analysis to the General Aptitude Test Battery (U.S. Employment Service Test Research Rep. No. 45). Washington, DC.

Examples

```
## Not run:
dat.gatb
effect <- dat.gatb$z
v <- dat.gatb$v
weightfunct(effect, v)

## End(Not run)
```

dat.smith

Studies From Smith, Glass, and Miller's (1980) Meta-Analysis of Psychotherapy Outcomes

Description

An arbitrary subset of 74 studies from a meta-analysis assessing the effectiveness of psychotherapy. Contains two moderator variables.

Usage

```
dat.smith
```

Format

A data frame containing the following columns:

`es` standardized mean difference effect sizes

`v` corresponding sampling variances

`age` continuous moderator representing clients' average age in years

`diagnosis` categorical moderator representing disorder for which clients were treated; 1 = complex phobia, 2 = simple phobia, 3 = other

Details

This dataset consists of an arbitrarily selected subset of 74 studies assessing the effectiveness of psychotherapy. Smith, Glass, and Miller (1980) published a meta-analysis designed to explore the current state of knowledge about psychotherapy effectiveness. Their original meta-analysis contains more than 1,700 effect sizes from 475 studies with multiple moderators and outcome measures. This subset is vastly simplified and intended solely for the purpose of demonstration.

Source

Smith, M. L., Glass, G. V., & Miller, T. I. (1980). Meta-analysis of psychotherapy. *American Psychologist*, 41, 165-180.

References

Smith, M. L., Glass, G. V., & Miller, T. I. (1980). Meta-analysis of psychotherapy. *American Psychologist*, 41, 165-180.

Examples

```
## Not run:
dat.smith
effect <- dat.smith$es
v <- dat.smith$v
weightfunct(effect, v)

## End(Not run)
```

density

Create a Density Plot

Description

This function allows you to create a plot displaying the unadjusted and adjusted densities of the specified model. Note that you must first specify a model using `weightfunct`.

Usage

```
density(x, ...)
```

Arguments

x an object of class `weightfunct`
... other arguments

Details

This function produces an approximate graphical illustration of the estimated unweighted and weighted densities. The unweighted density is represented by a dashed line and the weighted density by a solid line. For the unweighted density, the effect sizes are assumed to be normally distributed, with a mean equal to their unadjusted mean and a variance equal to their unadjusted variance component plus their individual sampling variances. This plot is an approximation because it is necessary to use a fixed sampling variance; here, we fix the sampling variance to the median of the distribution of sampling variances.

For the adjusted density, the expected density for effect sizes within each specified p-value interval is multiplied by the estimated weight for the corresponding interval. Greater density in an interval then represents a greater likelihood of effect-size survival. (Remember, of course, that the weight for the first interval is fixed to one, and other intervals should be interpreted relative to it.) Each discontinuity in the solid line, therefore, represents a p-value cutpoint.

Users may wonder why the adjusted density, or the solid line, sometimes falls outside of the unadjusted density, or the dashed line. In answer, recall that the mean and variance of the adjusted density also differ. Based on the severity of this difference, the adjusted density may fall outside of its unadjusted counterpart.

Examples

```
## Not run:  
test <- weightfunct(effect, v, steps)  
density(test)  
  
## End(Not run)
```

funnel

Create a Funnel Plot

Description

This function allows you to create a funnel plot using a vector of effect sizes and a vector of their corresponding sampling variances.

Usage

```
funnel(effect, v, type = "se", flip = FALSE)
```

Arguments

effect	a vector of meta-analytic effect sizes
v	a vector of sampling variances
type	v for variance or se for standard error; defaults to standard error
flip	FALSE (default) for a horizontal plot; TRUE for a vertical plot

Details

This funnel plot, by default, plots the effect sizes on the y-axis and the measure of study size (either variance or standard error) on the x-axis. If no asymmetry is present, the plot should resemble a horizontal funnel.

Users can choose either standard error (default) or sampling variance as a measure of study size. The choice is mostly arbitrary. In both cases, however, v must be a vector of variances, the same as that required by weightfunct. The conversion to standard error is automatic.

Examples

```
## Not run:
# Funnel plot using standard error (default):
funnel(effect, v)
# Funnel plot using sampling variance:
funnel(effect, v, type='v')

## End(Not run)
```

predict.weightfunct *Predicted Values for 'weightfunct' Objects*

Description

This function calculates predicted conditional means and their corresponding standard errors for objects of class weightfunct.

Usage

```
## S3 method for class 'weightfunct'
predict(object, values = NULL, ...)
```

Arguments

object	an object of class weightfunct
values	a vector or matrix specifying the values of the moderator variables for which predicted values should be calculated; defaults to NULL
...	other arguments

Details

predict(object) requires that the user specify a vector or matrix of predictor values. Without specifying values, the function will not work.

For models including y number of moderator variables, users should set values equal to a $k \times y$ matrix, where k is the number of rows of data (i.e., "new" studies). In the example code, for example, there are 3 moderator variables and one row of data, so values is a 1×3 matrix. The intercept is included by default.

Note that weightfunct handles categorical moderators automatically. To include them here, the appropriate contrast (dummy) variables must be explicitly specified. The contrasts function can help to understand the contrast matrix for a given factor.

Value

The function returns a list containing the following components: unadjusted, adjusted, and values. The values section simply prints the values matrix for verification. The unadjusted and adjusted sections print the conditional means for each row of new data, unadjusted and adjusted for publication bias (respectively), and their standard errors.

Examples

```
## Not run:
test <- weightfunct(effect, v, mods=~x1 + x2 + x3, steps)
values <- matrix(c(0,1,0),ncol=3) # An arbitrary set of 3 dummy-coded moderators
predict(test, values)

## End(Not run)
```

print.weightfunct *Print Model Results*

Description

This function allows you to print the model results.

Usage

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

Arguments

x an object of class weightfunct
 \dots other arguments

Examples

```
## Not run:  
print(weightfunct(d,v))  
  
## End(Not run)
```

shiny_weightr

Start weightr in Shiny

Description

This function allows you to launch the Shiny application locally.

Usage

```
shiny_weightr()
```

Examples

```
## Not run:  
library(shiny)  
shiny_weightr()  
  
## End(Not run)
```

weightfunct

Estimate the Vevea and Hedges (1995) Weight-Function Model

Description

This function allows the user to estimate the Vevea and Hedges (1995) weight-function model for publication bias.

Usage

```
weightfunct(effect, v, steps = c(0.025, 1), mods = NULL,  
            weights = NULL, fe = FALSE, table = FALSE, pval = NULL)
```

Arguments

<code>effect</code>	a vector of meta-analytic effect sizes.
<code>v</code>	a vector of meta-analytic sampling variances; needs to match up with the vector of effects, such that the first element in the vector of effect sizes goes with the first element in the vector of sampling variances, and so on.
<code>steps</code>	a vector of p-value cutpoints. The default only distinguishes between significant and non-significant effects ($p < 0.05$).
<code>mods</code>	defaults to NULL. A formula specifying the linear model.
<code>weights</code>	defaults to FALSE. A vector of prespecified weights for p-value cutpoints to estimate the Vevea and Woods (2005) model.
<code>fe</code>	defaults to FALSE. Indicates whether to estimate a fixed-effect model.
<code>table</code>	defaults to FALSE. Indicates whether to print a table of the p-value intervals specified and the number of effect sizes per interval.
<code>pval</code>	defaults to NULL. A vector containing observed p-values for the corresponding effect sizes. If not provided, p-values are calculated.

Details

This function allows meta-analysts to estimate both the weight-function model for publication bias that was originally published in Vevea and Hedges (1995) and the modified version presented in Vevea and Woods (2005). Users can estimate both of these models with and without predictors and in random-effects or fixed-effect situations. The function does not currently accommodate models without an intercept.

The Vevea and Hedges (1995) weight-function model is a tool for modeling publication bias using weighted distribution theory. The model first estimates an unadjusted fixed-, random-, or mixed-effects model, where the observed effect sizes are assumed to be normally distributed as a function of predictors. This unadjusted model is no different from the traditional meta-analytic model. Next, the Vevea and Hedges (1995) weight-function model estimates an adjusted model that includes not only the original mean model, fixed-, random-, or mixed-effects, but a series of weights for any pre-specified p-value intervals of interest. This produces mean, variance component, and covariate estimates adjusted for publication bias, as well as weights that reflect the likelihood of observing effect sizes in each specified interval.

It is important to remember that the weight for each estimated p-value interval must be interpreted relative to the first interval, the weight for which is fixed to 1 so that the model is identified. In other words, a weight of 2 for an interval indicates that effect sizes in that p-value interval are about twice as likely to be observed as those in the first interval. Finally, it is also important to remember that the model uses p-value cutpoints corresponding to one-tailed p-values. This allows flexibility in the selection function, which does not have to be symmetric for effects in the opposite direction; a two-tailed p-value of 0.05 can therefore be represented as $p < .025$ or $p > .975$.

After both the unadjusted and adjusted meta-analytic models are estimated, a likelihood-ratio test compares the two. The degrees of freedom for this test are equal to the number of weights being estimated. If the likelihood-ratio test is significant, this indicates that the adjusted model is a better fit for the data, and that publication bias may be a concern.

To estimate a large number of weights for p-value intervals, the Vevea and Hedges (1995) model works best with large meta-analytic datasets. It may have trouble converging and yield unreliable

parameter estimates if researchers, for instance, specify a p-value interval that contains no observed effect sizes. However, meta-analysts with small datasets are still likely to be interested in assessing publication bias, and need tools for doing so. Vevea and Woods (2005) attempted to solve this problem by adapting the Vevea and Hedges (1995) model to estimate fewer parameters. The meta-analyst can specify p-value cutpoints, as before, and specify corresponding fixed weights for those cutpoints. Then the model is estimated. For the adjusted model, only the variance component and mean model parameters are estimated, and they are adjusted relative to the fixed weights. For example, weights of 1 for each p-value interval specified describes a situation where there is absolutely no publication bias, in which the adjusted estimates are identical to the unadjusted estimates. By specifying weights that depart from 1 over various p-value intervals, meta-analysts can examine how various one-tailed or two-tailed selection patterns would alter their effect size estimates. If changing the pattern of weights drastically changes the estimated mean, this is evidence that the data may be vulnerable to publication bias.

For more information, consult the papers listed in the References section here. Also, feel free to email the maintainer of `weightr` at `kcoburn@ucmerced.edu`. The authors are currently at work on a detailed package tutorial, which we hope to publish soon.

Value

The function returns a list containing the following components: `output_unadj`, `output_adj`, `steps`, `mods`, `weights`, `fe`, `table`, `effect`, `v`, `npred`, `nsteps`, `p`, `XX`, `removed`.

The results of the unadjusted and adjusted models are returned by selecting the first (`[[1]]`) and second (`[[2]]`) elements of the list, respectively. The parameters can be obtained by `[[1]]$par` or `[[2]]$par`. The order of parameters is as follows: variance component, mean or linear coefficients, and weights. (Note that if `weights` are specified using the Vevea and Woods (2005) model, no standard errors, p-values, z-values, or confidence intervals are provided for the adjusted model, as these are no longer meaningful. Also note that the variance component is not reported for fixed-effect models.)

`unadj_est` the unadjusted model estimates

`adj_est` the adjusted model estimates

`steps` the specified p-value cutpoints

`mods` the linear model formula, if one is specified

`weights` the vector of weights for the Vevea and Woods (2005) model, if specified

`fe` indicates whether or not a fixed-effect model was estimated

`table` indicates whether a sample size table was produced

`effect` the vector of effect sizes

`v` the vector of sampling variances

`npred` the number of predictors included; 0 represents an intercept-only model

`nsteps` the number of p-value cutpoints

`p` a vector of p-values for the observed effect sizes

`XX` the model matrix; the first column of ones represents the intercept, and any other columns correspond to moderators

`removed` effect sizes with missing data are removed by listwise deletion; any removed are provided here. Defaults to `NULL`

References

- Coburn, K. M. & Vevea, J. L. (2015). Publication bias as a function of study characteristics. *Psychological Methods*, 20(3), 310.
- Vevea, J. L. & Hedges, L. V. (1995). A general linear model for estimating effect size in the presence of publication bias. *Psychometrika*, 60(3), 419-435.
- Vevea, J. L. & Woods, C. M. (2005). Publication bias in research synthesis: Sensitivity analysis using a priori weight functions. *Psychological Methods*, 10(4), 428-443.

Examples

```
## Not run:
# Uses the default p-value cutpoints of 0.05 and 1:

weightfunct(effect, v)

# Estimating a fixed-effect model, again with the default cutpoints:

weightfunct(effect, v, fe=TRUE)

# Specifying cutpoints:

weightfunct(effect, v, steps=c(0.01, 0.025, 0.05, 0.10, 0.20, 0.30, 0.50, 1.00))

# Including a linear model, where moderators are denoted as 'mod1' and 'mod2':

weightfunct(effect, v, mods=~mod1+mod2)

# Specifying cutpoints and weights to estimate Vevea and Woods (2005):

weightfunct(effect, v, steps=c(0.01, 0.05, 0.50, 1.00), weights=c(1, .9, .7, .5))

# Specifying cutpoints and weights while including a linear model:

weightfunct(effect, v, mods=~mod1+mod2, steps=c(0.05, 0.10, 0.50, 1.00), weights=c(1, .9, .8, .5))

## End(Not run)
```

Index

*Topic **datasets**

dat.bangertdrowns2004, [2](#)

dat.gatb, [3](#)

dat.smith, [4](#)

*Topic **weightr**

density, [5](#)

funnel, [6](#)

predict.weightfunct, [7](#)

print.weightfunct, [8](#)

shiny_weightr, [9](#)

weightfunct, [9](#)

dat.bangertdrowns2004, [2](#)

dat.gatb, [3](#)

dat.smith, [4](#)

density, [5](#)

funnel, [6](#)

predict.weightfunct, [7](#)

print.weightfunct, [8](#)

shiny_weightr, [9](#)

weightfunct, [9](#)