

Package ‘multid’

September 23, 2025

Title Multivariate Difference Between Two Groups

Version 1.0.2

Description Estimation of multivariate differences between two groups (e.g., multivariate sex differences) with regularized regression methods and predictive approach. See Ilmarinen et al. (2023) <[doi:10.1177/08902070221088155](https://doi.org/10.1177/08902070221088155)>. Deconstructing difference score correlations (e.g., gender-equality paradox), see Ilmarinen & Lönnqvist (2024) <[doi:10.1037/pspp0000508](https://doi.org/10.1037/pspp0000508)>.

Includes also tools that help in understanding difference score reliability, conditional intra-class correlations, tail-dependency, and heterogeneity of variance estimates. Package development was supported by the Academy of Finland research grant 338891.

License GPL-3

Encoding UTF-8

BugReports <https://github.com/vjilmari/multid/issues>

LazyData true

RoxygenNote 7.3.2

Imports dplyr (>= 1.0.7), glmnet (>= 4.1.2), stats (>= 4.0.2), pROC (>= 1.18.0), lavaan (>= 0.6.9), emmeans (>= 1.6.3), lme4 (>= 1.1.27.1), quantreg (>= 5.88), lmerTest (>= 3.1.3), ggpubr (>= 0.6.0), ggplot2 (>= 3.4.4), rlang (>= 1.1.6)

Suggests knitr (>= 1.39), rmarkdown (>= 2.14), overlapping (>= 1.7), rio (>= 0.5.29)

VignetteBuilder knitr

Depends R (>= 3.5)

NeedsCompilation no

Author Ville-Juhani Ilmarinen [aut, cre] (ORCID:
<<https://orcid.org/0000-0001-9493-379X>>)

Maintainer Ville-Juhani Ilmarinen <vj.ilmarinen@gmail.com>

Repository CRAN

Date/Publication 2025-09-23 08:10:02 UTC

Examples

```
d <- list(
  X1 = rnorm(10, sd = 10),
  X2 = rnorm(100, sd = 7.34),
  X3 = rnorm(1000, sd = 6.02),
  X4 = rnorm(100, sd = 5.17),
  X5 = rnorm(10, sd = 4.56)
)
cvv(d)
```

cvv_manual	<i>Coefficient of variance variation from manual input sample sizes and variance estimates</i>
------------	--

Description

Calculates three different indices for variation between two or more variance estimates. VR = Variance ratio between the largest and the smallest variance. CVV = Coefficient of variance variation (Box, 1954). SVH = Standardized variance heterogeneity (Ruscio & Roche, 2012).

Usage

```
cvv_manual(sample_sizes, variances)
```

Arguments

`sample_sizes` Numeric vector of length > 1. Sample sizes used for each variance estimate.
`variances` Numeric vector of length > 1. Variance estimates.

Value

A vector including VR, CVV, and SVH.

References

Box, G. E. P. (1954). Some Theorems on Quadratic Forms Applied in the Study of Analysis of Variance Problems, I. Effect of Inequality of Variance in the One-Way Classification. *The Annals of Mathematical Statistics*, 25(2), 290–302.

Ruscio, J., & Roche, B. (2012). Variance Heterogeneity in Published Psychological Research: A Review and a New Index. *Methodology*, 8(1), 1–11. <https://doi.org/10.1027/1614-2241/a000034>

Examples

```
cvv_manual(sample_sizes=c(10,100,1000,75,3),
  variances=c(1.5,2,2.5,3,3.5))
```

ddsc_ml

*Deconstructing difference score correlation with multi-level modeling***Description**

Deconstructs a bivariate association between x and a difference score $y_1 - y_2$ with multi-level modeling approach. Within each upper-level unit (`lv12_unit`) there can be multiple observations of y_1 and y_2 . Can be used for either pre-fitted lmer-models or to long format data. A difference score correlation is indicative that slopes for y_1 as function of x and y_2 as function of x are non-parallel. Deconstructing the bivariate association to these slopes allows for understanding the pattern and magnitude of this non-parallelism.

Usage

```
ddsc_ml(
  model = NULL,
  data = NULL,
  predictor,
  moderator,
  moderator_values,
  DV = NULL,
  lv12_unit = NULL,
  re_cov_test = FALSE,
  var_boot_test = FALSE,
  boot_slopes = FALSE,
  nsim = NULL,
  level = 0.95,
  seed = NULL,
  covariates = NULL,
  scaling_sd = "observed"
)
```

Arguments

<code>model</code>	Multilevel model fitted with lmerTest.
<code>data</code>	Data frame.
<code>predictor</code>	Character string. Variable name of independent variable predicting difference score (i.e., x).
<code>moderator</code>	Character string. Variable name indicative of difference score components (w).
<code>moderator_values</code>	Vector. Values of the component score groups in moderator (i.e., y_1 and y_2).
<code>DV</code>	Character string. Name of the dependent variable (if model is not supplied as input).
<code>lv12_unit</code>	Character string. Name of the level-2 clustering variable (if model is not supplied as input).

re_cov_test	Logical. Significance test for random effect covariation? (Default FALSE)
var_boot_test	Logical. Compare variance by lower-level groups at the upper-level in a reduced model with bootstrap? (Default FALSE)
boot_slopes	Logical. Are bootstrap estimates and percentile confidence intervals obtained for the estimates presented in results? (Default FALSE)
nsim	Numeric. Number of bootstrap simulations.
level	Numeric. The confidence level required for the var_boot_test output (Default .95)
seed	Numeric. Seed number for bootstrap simulations.
covariates	Character string or vector. Variable names of covariates (Default NULL).
scaling_sd	Character string (either default "observed" or "model"). Are the simple slopes scaled with observed or model-based SDs?

Value

results	Summary of key results.
descriptives	Means, standard deviations, and intercorrelations at level 2.
vpc_at_moderator_values	Variance partition coefficients for moderator values in the model without the predictor and interactions.
model	Fitted lmer object.
reduced_model	Fitted lmer object without the predictor.
lv12_data	Data summarized at level 2.
ddsc_sem_fit	ddsc_sem object fitted to level 2 data.
re_cov_test	Likelihood ratio significance test for random effect covariation.
boot_var_diffs	List of different variance bootstrap tests.

Examples

```
## Not run:
set.seed(95332)
n1 <- 10 # groups
n2 <- 10 # observations per group
dat <- data.frame(
  group = rep(c(LETTERS[1:n1]), each = n2),
  w = sample(c(-0.5, 0.5), n1 * n2, replace = TRUE),
  x = rep(sample(1:5, n1, replace = TRUE), each = n2),
  y = sample(1:5, n1 * n2, replace = TRUE)
)
library(lmerTest)
fit <- lmerTest::lmer(y ~ x * w + (w | group),
  data = dat
)
round(ddsc_ml(model=fit,
  predictor="x",
  moderator="w",
```

```

moderator_values=c(0.5,-0.5))$results,3)

round(ddsc_ml(data=dat,
             DV="y",
             lvl2_unit="group",
             predictor="x",
             moderator="w",
             moderator_values=c(0.5,-0.5))$results,3)

## End(Not run)

```

ddsc_sem	<i>Deconstructing difference score correlation with structural equation modeling</i>
----------	--

Description

Deconstructs a bivariate association between x and a difference score $y_1 - y_2$ with SEM. A difference score correlation is indicative that slopes for y_1 as function of x and y_2 as function of x are non-parallel. Deconstructing the bivariate association to these slopes allows for understanding the pattern and magnitude of this non-parallelism.

Usage

```

ddsc_sem(
  data,
  x,
  y1,
  y2,
  center_yvars = FALSE,
  covariates = NULL,
  estimator = "ML",
  level = 0.95,
  sampling.weights = NULL,
  q_sesoi = 0,
  min_cross_over_point_location = 0,
  boot_ci = FALSE,
  boot_n = 5000,
  boot_ci_type = "perc"
)

```

Arguments

data	A data frame.
x	Character string. Variable name of independent variable.
y1	Character string. Variable name of first component score of difference score.

y2	Character string. Variable name of second component score of difference score.
center_yvars	Logical. Should y1 and y2 be centered around their grand mean? (Default FALSE)
covariates	Character string or vector. Variable names of covariates (Default NULL).
estimator	Character string. Estimator used in SEM (Default "ML").
level	Numeric. The confidence level required for the result output (Default .95)
sampling.weights	Character string. Name of sampling weights variable.
q_sesoi	Numeric. The smallest effect size of interest for Cohen's q estimates (Default 0; See Lakens et al. 2018).
min_cross_over_point_location	Numeric. Z-score for the minimal slope cross-over point of interest (Default 0).
boot_ci	Logical. Calculate confidence intervals based on bootstrap (Default FALSE).
boot_n	Numeric. How many bootstrap redraws (Default 5000).
boot_ci_type	If bootstrapping was used, the type of interval required. The value should be one of "norm", "basic", "perc" (default), or "bca.simple".

Value

descriptives	Means, standard deviations, and intercorrelations.
parameter_estimates	Parameter estimates from the structural equation model.
variance_test	Variances and covariances of component scores.
data	Data frame with original and scaled variables used in SEM.
results	Summary of key results.

References

- Edwards, J. R. (1995). Alternatives to Difference Scores as Dependent Variables in the Study of Congruence in Organizational Research. *Organizational Behavior and Human Decision Processes*, 64(3), 307–324.
- Lakens, D., Scheel, A. M., & Isager, P. M. (2018). Equivalence Testing for Psychological Research: A Tutorial. *Advances in Methods and Practices in Psychological Science*, 1(2), 259–269. <https://doi.org/10.1177/2515245918770963>

Examples

```
## Not run:
set.seed(342356)
d <- data.frame(
  y1 = rnorm(50),
  y2 = rnorm(50),
  x = rnorm(50)
)
ddsc_sem(
  data = d, y1 = "y1", y2 = "y2",
```

```

x = "x",
q_sesoi = 0.20,
min_cross_over_point_location = 1
)$results

## End(Not run)

```

diff_two_dep_cors	<i>Difference between two dependent Pearson's correlations (with common index)</i>
-------------------	--

Description

Calculates Cohen's q effect size statistic for difference between two correlations, r_{yx1} and r_{yx2} . Tests if Cohen's q is different from zero while accounting for dependency between the two correlations.

Usage

```
diff_two_dep_cors(data, y, x1, x2, level = 0.95, missing = "default")
```

Arguments

data	Data frame.
y	Character. Variable name of the common index variable.
x1	Character. Variable name.
x2	Character. Variable name.
level	Numeric. The confidence level required for the result output (Default .95)
missing	Character. Treatment of missing values (e.g., "ML", default = listwise deletion)

Value

Parameter estimates from the fitted structural path model.

Examples

```

set.seed(3864)
d<-data.frame(y=rnorm(100),x=rnorm(100))
d$x1<-d$x+rnorm(100)
d$x2<-d$x+rnorm(100)
diff_two_dep_cors(data=d,y="y",x1="x1",x2="x2")

```

d_pooled_sd	<i>Standardized mean difference with pooled standard deviation</i>
-------------	--

Description

Standardized mean difference with pooled standard deviation

Usage

```
d_pooled_sd(  
  data,  
  var,  
  group.var,  
  group.values,  
  rename.output = TRUE,  
  infer = FALSE  
)
```

Arguments

data	A data frame.
var	A continuous variable for which difference is estimated.
group.var	The name of the group variable.
group.values	Vector of length 2, group values (e.g. c("male", "female") or c(0,1)).
rename.output	Logical. Should the output values be renamed according to the group.values? Default TRUE.
infer	Logical. Statistical inference with Welch test? (default FALSE)

Value

Descriptive statistics and mean differences

Examples

```
d_pooled_sd(iris[iris$Species == "setosa" | iris$Species == "versicolor", ],  
  var = "Petal.Length", group.var = "Species",  
  group.values = c("setosa", "versicolor"), infer = TRUE  
)
```

D_regularized	<i>Multivariate group difference estimation with regularized binomial regression</i>
---------------	--

Description

Multivariate group difference estimation with regularized binomial regression

Usage

```
D_regularized(
  data,
  mv.vars,
  group.var,
  group.values,
  alpha = 0.5,
  nfolds = 10,
  s = "lambda.min",
  type.measure = "deviance",
  rename.output = TRUE,
  out = FALSE,
  size = NULL,
  fold = FALSE,
  fold.var = NULL,
  pcc = FALSE,
  auc = FALSE,
  pred.prob = FALSE,
  prob.cutoffs = seq(0, 1, 0.2),
  append.data = FALSE
)
```

Arguments

data	A data frame or list containing two data frames (regularization and estimation data, in that order).
mv.vars	Character vector. Variable names in the multivariate variable set.
group.var	The name of the group variable.
group.values	Vector of length 2, group values (e.g. c("male", "female") or c(0,1)).
alpha	Alpha-value for penalizing function ranging from 0 to 1: 0 = ridge regression, 1 = lasso, 0.5 = elastic net (default).
nfolds	Number of folds used for obtaining lambda (range from 3 to n-1, default 10).
s	Which lambda value is used for predicted values? Either "lambda.min" (default) or "lambda.1se".
type.measure	Which measure is used during cross-validation. Default "deviance".

rename.output	Logical. Should the output values be renamed according to the group.values? Default TRUE.
out	Logical. Should results and predictions be calculated on out-of-bag data set? (Default FALSE)
size	Integer. Number of cases in regularization data per each group. Default 1/4 of cases.
fold	Logical. Is regularization applied across sample folds with separate predictions for each fold? (Default FALSE, see details)
fold.var	Character string. Name of the fold variable. (default NULL)
pcc	Logical. Include probabilities of correct classification? Default FALSE.
auc	Logical. Include area under the receiver operating characteristics? Default FALSE.
pred.prob	Logical. Include table of predicted probabilities? Default FALSE.
prob.cutoffs	Vector. Cutoffs for table of predicted probabilities. Default seq(0,1,0.20).
append.data	Logical. If TRUE, the data is appended to the predicted variables.

Details

fold = TRUE will apply manually defined data folds (supplied with fold.var) for regularization and obtain estimates for each separately. This can be a good solution, for example, when the data are clustered within countries. In such case, the cross-validation procedure is applied across countries.

out = TRUE will use separate data partition for regularization and estimation. That is, the first cross-validation procedure is applied within the regularization set and the weights obtained are then used in the estimation data partition. The size of regularization set is defined with size. When used with fold = TRUE, size means size within a fold."

For more details on these options, please refer to the [vignette](#) and [README](#) of the multid package.

Value

D	Multivariate descriptive statistics and differences.
pred.dat	A data.frame with predicted values.
cv.mod	Regularized regression model from cv.glmnet.
P.table	Table of predicted probabilities by cutoffs.

References

Lönnqvist, J. E., & Ilmarinen, V. J. (2021). Using a continuous measure of genderedness to assess sex differences in the attitudes of the political elite. *Political Behavior*, 43, 1779–1800. doi:10.1007/s11109021096812

Ilmarinen, V. J., Vainikainen, M. P., & Lönnqvist, J. E. (2023). Is there a g-factor of genderedness? Using a continuous measure of genderedness to assess sex differences in personality, values, cognitive ability, school grades, and educational track. *European Journal of Personality*, 37, 313–337. doi:10.1177/08902070221088155

See Also[cv.glmnet](#)**Examples**

```

D_regularized(
  data = iris[iris$Species == "setosa" | iris$Species == "versicolor", ],
  mv.vars = c("Sepal.Length", "Sepal.Width", "Petal.Length", "Petal.Width"),
  group.var = "Species", group.values = c("setosa", "versicolor")
)$D

# out-of-bag predictions
D_regularized(
  data = iris[iris$Species == "setosa" | iris$Species == "versicolor", ],
  mv.vars = c("Sepal.Length", "Sepal.Width", "Petal.Length", "Petal.Width"),
  group.var = "Species", group.values = c("setosa", "versicolor"),
  out = TRUE, size = 15, pcc = TRUE, auc = TRUE
)$D

# separate sample folds
# generate data for 10 groups
set.seed(34246)
n1 <- 100
n2 <- 10
d <-
  data.frame(
    sex = sample(c("male", "female"), n1 * n2, replace = TRUE),
    fold = sample(x = LETTERS[1:n2], size = n1 * n2, replace = TRUE),
    x1 = rnorm(n1 * n2),
    x2 = rnorm(n1 * n2),
    x3 = rnorm(n1 * n2)
  )

# Fit and predict with same data
D_regularized(
  data = d,
  mv.vars = c("x1", "x2", "x3"),
  group.var = "sex",
  group.values = c("female", "male"),
  fold.var = "fold",
  fold = TRUE,
  rename.output = TRUE
)$D

# Out-of-bag data for each fold
D_regularized(
  data = d,
  mv.vars = c("x1", "x2", "x3"),
  group.var = "sex",
  group.values = c("female", "male"),
  fold.var = "fold",
  size = 17,

```

```
  out = TRUE,  
  fold = TRUE,  
  rename.output = TRUE  
)$D
```

example_big5

Example BIG5 personality dataset (random subset)

Description

A small subset of the BIG5 dataset from OpenPsychometrics, included as a fallback for vignettes. This dataset consists of 200 U.S. respondents: 100 men and 100 women.

Usage

```
example_big5
```

Format

A data frame with 200 rows and the same variables as the full BIG5 dataset.

gender Gender of respondent (1 = male, 2 = female)

country Country of respondent (here always "US")

age Age of respondent

A1 Agreeableness item 1

A2 Agreeableness item 2

A3 Agreeableness item 3

A4 Agreeableness item 4

A5 Agreeableness item 5

A6 Agreeableness item 6

A7 Agreeableness item 7

A8 Agreeableness item 8

A9 Agreeableness item 9

A10 Agreeableness item 10

C1 Conscientiousness item 1

C2 Conscientiousness item 2

C3 Conscientiousness item 3

C4 Conscientiousness item 4

C5 Conscientiousness item 5

C6 Conscientiousness item 6

C7 Conscientiousness item 7

C8 Conscientiousness item 8

- C9** Conscientiousness item 9
- C10** Conscientiousness item 10
- E1** Extraversion item 1
- E2** Extraversion item 2
- E3** Extraversion item 3
- E4** Extraversion item 4
- E5** Extraversion item 5
- E6** Extraversion item 6
- E7** Extraversion item 7
- E8** Extraversion item 8
- E9** Extraversion item 9
- E10** Extraversion item 10
- N1** Neuroticism item 1
- N2** Neuroticism item 2
- N3** Neuroticism item 3
- N4** Neuroticism item 4
- N5** Neuroticism item 5
- N6** Neuroticism item 6
- N7** Neuroticism item 7
- N8** Neuroticism item 8
- N9** Neuroticism item 9
- N10** Neuroticism item 10
- O1** Openness item 1
- O2** Openness item 2
- O3** Openness item 3
- O4** Openness item 4
- O5** Openness item 5
- O6** Openness item 6
- O7** Openness item 7
- O8** Openness item 8
- O9** Openness item 9
- O10** Openness item 10

Source

http://openpsychometrics.org/_rawdata/

ml_dadas

*Predicting algebraic difference scores in multilevel model***Description**

Decomposes difference score predictions to predictions of difference score components by probing simple effects at the levels of the binary moderator.

Usage

```
ml_dadas(
  model,
  predictor,
  diff_var,
  diff_var_values,
  scaled_estimates = FALSE,
  re_cov_test = FALSE,
  var_boot_test = FALSE,
  nsim = NULL,
  level = 0.95,
  seed = NULL,
  abs_diff_test = 0
)
```

Arguments

model	Multilevel model fitted with lmerTest.
predictor	Character string. Variable name of independent variable predicting difference score.
diff_var	Character string. A variable indicative of difference score components (two groups).
diff_var_values	Vector. Values of the component score groups in diff_var.
scaled_estimates	Logical. Are scaled estimates obtained? Does fit a reduced model for correct standard deviations. (Default FALSE)
re_cov_test	Logical. Significance test for random effect covariation? Does fit a reduced model without the correlation. (Default FALSE)
var_boot_test	Logical. Compare variance by lower-level groups at the upper-level in a reduced model with bootstrap? (Default FALSE)
nsim	Numeric. Number of bootstrap simulations.
level	Numeric. The confidence level required for the var_boot_test output (Default .95)
seed	Numeric. Seed number for bootstrap simulations.
abs_diff_test	Numeric. A value against which absolute difference between component score predictions is tested (Default 0).

Value

dadas	A data frame including main effect, interaction, regression coefficients for component scores, <i>dadas</i> , and comparison between interaction and main effect.
scaled_estimates	Scaled regression coefficients for difference score components and difference score.
vpc_at_reduced	Variance partition coefficients in the model without the predictor and interactions.
re_cov_test	Likelihood ratio significance test for random effect covariation.
boot_var_diffs	List of different variance bootstrap tests.

Examples

```
## Not run:
set.seed(95332)
n1 <- 10 # groups
n2 <- 10 # observations per group

dat <- data.frame(
  group = rep(c(LETTERS[1:n1]), each = n2),
  w = sample(c(-0.5, 0.5), n1 * n2, replace = TRUE),
  x = rep(sample(1:5, n1, replace = TRUE), each = n2),
  y = sample(1:5, n1 * n2, replace = TRUE)
)
library(lmerTest)
fit <- lmerTest::lmer(y ~ x * w + (w | group),
  data = dat
)

round(ml_dadas(fit,
  predictor = "x",
  diff_var = "w",
  diff_var_values = c(0.5, -0.5)
)$dadas, 3)

## End(Not run)
```

multid-deprecated

Deprecated functions in multid

Description

These functions still work but will be removed (defunct) in the next version.

Details

- `sem_dadas`: This function is deprecated, and will be removed in the next version of this package.
- `ml_dadas`: This function is deprecated, and will be removed in the next version of this package.

pcc

Returns probabilities of correct classification for both groups in independent data partition.

Description

Returns probabilities of correct classification for both groups in independent data partition.

Usage

```
pcc(data, pred.var, group.var, group.values)
```

Arguments

<code>data</code>	Data frame including predicted values (e.g., <code>pred.dat</code> from <code>D_regularized_out</code>).
<code>pred.var</code>	Character string. Variable name for predicted values.
<code>group.var</code>	The name of the group variable.
<code>group.values</code>	Vector of length 2, group values (e.g. <code>c("male", "female")</code> or <code>c(0,1)</code>).

Value

Vector of length 2. Probabilities of correct classification.

Examples

```
D_out <- D_regularized(
  data = iris[iris$Species == "versicolor" | iris$Species == "virginica", ],
  mv.vars = c("Sepal.Length", "Sepal.Width", "Petal.Length", "Petal.Width"),
  group.var = "Species", group.values = c("versicolor", "virginica"),
  out = TRUE,
  size = 15
)

pcc(
  data = D_out$pred.dat,
  pred.var = "pred",
  group.var = "group",
  group.values = c("versicolor", "virginica")
)
```

plot_ddsc

Plot deconstructed difference score correlation

Description

Plots the slopes for y_1 and y_2 by x , and a slope for $y_1 - y_2$ by x for comparison.

Usage

```
plot_ddsc(
  ddsc_object,
  diff_color = "black",
  y1_color = "turquoise",
  y2_color = "orange",
  x_label = NULL,
  y_labels = NULL,
  densities = TRUE,
  point_alpha = 0.5,
  dens_alpha = 0.75,
  col_widths = c(3, 1),
  row_heights = c(2, 1, 0.5),
  coef_locations = c(0/3, 1/3, 2/3),
  coef_names = c("b_11", "b_21", "r_x_y1-y2"),
  coef_text_size = 4,
  y_scale = "standardized",
  x_scale = "scaled",
  show_dens_x_labels = TRUE
)
```

Arguments

ddsc_object	An object produced by ddsc_sem function.
diff_color	Character. Color for difference score ($y_1 - y_2$). Default "black".
y1_color	Character. Color for difference score component y_1 . Default "turquoise".
y2_color	Character. Color for difference score component y_2 . Default "orange".
x_label	Character. Label for variable X . If NULL (default), variable name is used.
y_labels	Character vector. Labels for variable y_1 and y_2 . If NULL (default), variable names are used.
densities	Logical. Are y -variable densities plotted? Default TRUE.
point_alpha	Numeric. Opacity for data points (default 0.50)
dens_alpha	Numeric. Opacity for density distributions (default 0.75)
col_widths	Numeric vector. Widths of the plot columns: slope figures and density figures; default c(3, 1).

row_heights	Numeric vector. Heights of the plot rows: components, difference score, slope coeffs; default c(2, 1, 0.5).
coef_locations	Numeric vector. Locations for printed coefficients. Quantiles of the range of x-variable. Default c(0, 1/3, 2/3).
coef_names	Character vector. Names of the printed coefficients. Default c("b_11", "b_21", "r_x_y1-y2").
coef_text_size	Numeric. Text size of the printed coefficients. Default 4.
y_scale	Character. "Scaled"/"standardized" with harmonized SD "raw" (original scale). Default is "standardized".
x_scale	Character. "Standardized" or "raw" (original scale). Default is "standardized".
show_dens_x_labels	Logical. Show x-labels on the density plots. Default TRUE.

Examples

```

set.seed(342356)
d <- data.frame(
  y1 = rnorm(50),
  y2 = rnorm(50),
  x = rnorm(50)
)
fit <- ddsc_sem(
  data = d, y1 = "y1", y2 = "y2",
  x = "x"
)

plot_ddsc(fit, x_label = "X",
           y_labels = c("Y1", "Y2"))

```

qcc

Quantile correlation coefficient

Description

For computation of tail dependence as correlations estimated at different variable quantiles (Choi & Shin, 2022; Lee et al., 2022) summarized across two quantile regression models where x and y switch roles as independent/dependent variables.

Usage

```

qcc(
  x,
  y,
  tau = c(0.1, 0.5, 0.9),
  data,
  method = "br",

```

```

boot_n = NULL,
ci_level = 0.95
)

```

Arguments

x	Name of x variable. Character string.
y	Name of y variable. Character string.
tau	The quantile(s) to be estimated. A vector of values between 0 and 1, default c(.1,.5,.9). @seealso rq
data	Data frame.
method	The algorithmic method used to compute the fit (default "br"). @seealso rq
boot_n	Number of bootstrap redraws (default NULL = no bootstrap inference).
ci_level	Level for percentile bootstrap confidence interval. Numeric values between 0 and 1. Default .95.

Details

Note that when quantile regression coefficients for y on x and x on y have a different sign, the quantile correlation is defined as zero (see Choi & Shin, 2022, p. 1080).

Value

r	Pearson's correlation estimate for comparison.
rho_tau	Correlations at different tau values (quantiles).
r_boot_est	Pearson's correlation bootstrap estimates.
rho_tau_boot_est	Bootstrap estimates for correlations at different tau values (quantiles).

References

- Choi, J.-E., & Shin, D. W. (2022). Quantile correlation coefficient: A new tail dependence measure. *Statistical Papers*, 63(4), 1075–1104. <https://doi.org/10.1007/s00362-021-01268-7>
- Lee, J. A., Bardi, A., Gerrans, P., Sneddon, J., van Herk, H., Evers, U., & Schwartz, S. (2022). Are value–behavior relations stronger than previously thought? It depends on value importance. *European Journal of Personality*, 36(2), 133–148. <https://doi.org/10.1177/08902070211002965>

Examples

```

set.seed(2321)
d <- data.frame(x = rnorm(2000))
d$y <- 0.10 * d$x + (0.20) * d$x^2 + 0.40 * d$x^3 + (-0.20) * d$x^4 + rnorm(2000)
qcc_boot <- qcc(x = "x", y = "y", data = d, tau = 1:9 / 10, boot_n = 50)
qcc_boot$rho_tau

```

reliability_dms	<i>Reliability calculation for difference score variable that is a difference between two mean variables calculated over upper-level units (e.g., sex differences across countries)</i>
-----------------	---

Description

Calculates reliability of difference score (Johns, 1981) based on two separate ICC2 values (Bliese, 2000), standard deviations of mean values over upper-level units, and correlations between the mean values across upper-level units.

Usage

```
reliability_dms(  
  model = NULL,  
  data = NULL,  
  diff_var,  
  diff_var_values,  
  var,  
  group_var  
)
```

Arguments

model	Multilevel model fitted with lmer (default NULL)
data	Long format data frame (default NULL)
diff_var	Character string. A variable indicative of difference score components (two groups).
diff_var_values	Vector. Values of the component score groups in diff_var.
var	Character string. Name of the dependent variable or variable of which mean values are calculated.
group_var	Character string. Upper-level clustering unit.

Value

A vector including ICC2s (r11 and r22), SDs (sd1, sd2, and sd_d12), means (m1, m2, and m_d12), correlation between means (r12), and reliability of the mean difference variable.

References

Bliese, P. D. (2000). Within-group agreement, non-independence, and reliability: Implications for data aggregation and analysis. In K. J. Klein & S. W. J. Kozlowski (Eds.), *Multilevel theory, research, and methods in organizations: Foundations, extensions, and new directions* (pp. 349–381). Jossey-Bass.

Johns, G. (1981). Difference score measures of organizational behavior variables: A critique. *Organizational Behavior and Human Performance*, 27(3), 443–463. [https://doi.org/10.1016/0030-5073\(81\)90033-7](https://doi.org/10.1016/0030-5073(81)90033-7)

Examples

```
set.seed(4317)
n2 <- 20
n1 <- 200
ri <- rnorm(n2, m = 0.5, sd = 0.2)
rs <- 0.5 * ri + rnorm(n2, m = 0.3, sd = 0.15)
d.list <- list()
for (i in 1:n2) {
  x <- rep(c(-0.5, 0.5), each = n1 / 2)
  y <- ri[i] + rs[i] * x + rnorm(n1)
  d.list[[i]] <- cbind(x, y, i)
}

d <- data.frame(do.call(rbind, d.list))
names(d) <- c("x", "y", "centry")
reliability_dms(
  data = d, diff_var = "x",
  diff_var_values = c(-0.5, 0.5), var = "y", group_var = "centry"
)
```

sem_dadas

Predicting algebraic difference scores in structural equation model

Description

Predicting algebraic difference scores in structural equation model

Usage

```
sem_dadas(
  data,
  var1,
  var2,
  center = FALSE,
  scale = FALSE,
  predictor,
  covariates = NULL,
  estimator = "MLR",
  level = 0.95,
  sampling.weights = NULL,
  abs_coef_diff_test = 0
)
```

Arguments

data	A data frame.
var1	Character string. Variable name of first component score of difference score (Y_1).
var2	Character string. Variable name of second component score of difference score (Y_2).
center	Logical. Should var1 and var2 be centered around their grand mean? (Default FALSE)
scale	Logical. Should var1 and var2 be scaled with their pooled sd? (Default FALSE)
predictor	Character string. Variable name of independent variable predicting difference score.
covariates	Character string or vector. Variable names of covariates (Default NULL).
estimator	Character string. Estimator used in SEM (Default "MLR").
level	Numeric. The confidence level required for the result output (Default .95)
sampling.weights	Character string. Name of sampling weights variable.
abs_coef_diff_test	Numeric. A value against which absolute difference between component score predictions is tested (Default 0).

Value

descriptives	Means, standard deviations, and intercorrelations.
parameter_estimates	Parameter estimates from the structural equation model.
variance_test	Variances and covariances of component scores.
transformed_data	Data frame with variables used in SEM.
dadas	One sided dadas-test for positivity of $\text{abs}(b_{11}-b_{21})-\text{abs}(b_{11}+b_{21})$.
results	Summary of key results.

References

Edwards, J. R. (1995). Alternatives to Difference Scores as Dependent Variables in the Study of Congruence in Organizational Research. *Organizational Behavior and Human Decision Processes*, 64(3), 307–324.

Examples

```
## Not run:
set.seed(342356)
d <- data.frame(
  var1 = rnorm(50),
  var2 = rnorm(50),
  x = rnorm(50)
```

```

)
sem_dadas(
  data = d, var1 = "var1", var2 = "var2",
  predictor = "x", center = TRUE, scale = TRUE,
  abs_coef_diff_test = 0.20
)$results

## End(Not run)

```

value_correlation	<i>Testing and quantifying how much ipsatization (profile centering) influence associations between value and a correlate</i>
-------------------	---

Description

Testing and quantifying how much ipsatization (profile centering) influence associations between value and a correlate

Usage

```

value_correlation(
  data,
  rv,
  cf,
  correlate,
  scale_by_rv = FALSE,
  standardize_correlate = FALSE,
  estimator = "ML",
  level = 0.95,
  sampling.weights = NULL,
  sesoi = 0
)

```

Arguments

data	A data frame.
rv	Character string or vector. Variable name(s) of the non-ipsatized value variable(s) (raw value score).
cf	Character string. Variable name of the common factor that is used for ipsatizing raw value scores.
correlate	Character string. Name of the variable to which associations with values are examined.
scale_by_rv	Logical. Is standard deviation of the raw non-ipsatized value score used for scaling the common factor as well? (Default FALSE)
standardize_correlate	Logical. Should the correlate be standardized? (Default FALSE)

estimator	Character string. Estimator used in SEM (Default "ML").
level	Numeric. The confidence level required for the result output (Default .95)
sampling.weights	Character string. Name of sampling weights variable.
sesoi	Numeric. Smallest effect size of interest. Used for equivalence testing differences in ipsatized and non-ipsatized value associations (Default 0).

Value

parameter_estimates	Parameter estimates from the structural equation model.
transformed_data	Data frame with variables used in SEM (after scaling is applied).
results	Summary of key results.

Examples

```
## Not run:
set.seed(342356)
d <- data.frame(
  rv1 = rnorm(50),
  rv2 = rnorm(50),
  rv3 = rnorm(50),
  rv4 = rnorm(50),
  x = rnorm(50)
)
d$cf <- rowMeans(d[,c("rv1", "rv2", "rv3", "rv4")])
fit <- value_correlation(
  data = d, rv = c("rv1", "rv2", "rv3", "rv4"), cf = "cf",
  correlate = "x", scale_by_rv = TRUE,
  standardize_correlate = TRUE,
  sesoi = 0.10
)
round(fit$variability_summary, 3)
round(fit$association_summary, 3)

## End(Not run)
```

vpc_at

Variance partition coefficient calculated at different level-1 values

Description

Calculates variance estimates (level-2 Intercept variance) and variance partition coefficients (i.e., intra-class correlation) at selected values of predictor values in two-level linear models with random effects (intercept, slope, and their covariation).

Usage

```
vpc_at(model, lv11.var, lv11.values)
```

Arguments

model	Two-level model fitted with lme4. Must include random intercept, slope, and their covariation.
lv11.var	Character string. Level 1 variable name to which random slope is also estimated.
lv11.values	Level 1 variable values.

Value

Data frame of level 2 variance and std.dev. estimates at level 1 variable values, respective VPCs (ICC1s) and group-mean reliabilities (ICC2s) (Bliese, 2000).

References

Goldstein, H., Browne, W., & Rasbash, J. (2002). Partitioning Variation in Multilevel Models. *Understanding Statistics*, 1(4), 223–231. https://doi.org/10.1207/S15328031US0104_02

Bliese, P. D. (2000). Within-group agreement, non-independence, and reliability: Implications for data aggregation and analysis. In K. J. Klein & S. W. J. Kozlowski (Eds.), *Multilevel theory, research, and methods in organizations: Foundations, extensions, and new directions* (pp. 349–381). Jossey-Bass.

Examples

```
fit <- lme4::lmer(Sepal.Length ~ Petal.Length +
  (Petal.Length | Species),
  data = iris
)

lv11.values <-
  c(
    mean(iris$Petal.Length) - stats::sd(iris$Petal.Length),
    mean(iris$Petal.Length),
    mean(iris$Petal.Length) + stats::sd(iris$Petal.Length)
  )

vpc_at(
  model = fit,
  lv11.var = "Petal.Length",
  lv11.values = lv11.values
)
```

Index

* datasets

example_big5, [13](#)

cv.glmnet, [12](#)

cvv, [2](#)

cvv_manual, [3](#)

d_pooled_sd, [9](#)

D_regularized, [10](#)

ddsc_ml, [4](#)

ddsc_sem, [6](#)

diff_two_dep_cors, [8](#)

example_big5, [13](#)

ml_dadas, [15](#), [17](#)

multid-deprecated, [16](#)

pcc, [17](#)

plot_ddsc, [18](#)

qcc, [19](#)

reliability_dms, [21](#)

rq, [20](#)

sem_dadas, [17](#), [22](#)

value_correlation, [24](#)

vpc_at, [25](#)