Package 'mars'

March 27, 2025

Title Meta Analysis and Research Synthesis

Version 0.2.2

Description Includes functions for conducting univariate and multivariate meta-analysis. This includes the estimation of the asymptotic variance-covariance matrix of effect sizes. For more details see Becker (1992) <doi:10.2307/1165128>, Cooper, Hedges, and Valentine (2019) <doi:10.7758/9781610448864>, and Schmid, Stijnen, and White (2020) <doi:10.1201/9781315119403>.

Depends R (>= 4.2)

Imports Matrix, corpcor, matrixcalc

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Suggests knitr, rmarkdown, dplyr, testthat (>= 3.0.0)

VignetteBuilder knitr

RoxygenNote 7.3.2

Config/testthat/edition 3

NeedsCompilation no

Author Brandon LeBeau [aut, cre] (<https://orcid.org/0000-0002-1265-8761>),
Ariel M. Aloe [aut],
Betsy Jane Becker [ctb],
Seohee Park [ctb],
Roberto C. Toro Rodriguez [ctb]

Maintainer Brandon LeBeau <lebebr01+mars@gmail.com>

Repository CRAN

Date/Publication 2025-03-27 18:00:05 UTC

Contents

---

bcg *bcg Data*

Description

This data ...

Usage

```
bcg
```

Format

A dataframe with 13 rows and 11 columns:

Trial Trial ID

Examples

```
bcg
```

---

becker09	*Becker 2009 data*

Description

This data contains data originally collected by Becker in 2009 on ...

Usage

```
becker09
```

Format

A dataframe with 10 rows and 9 columns:

ID Study ID

N Study sample size

Team ...

Cognitive_Performance Correlation between Cognitive and Performance

Somatic_Performance Correlation between Somatic and Performance

Selfconfidence_Performance Correlation between Self Confidence and Performance

Somatic_Cognitive Correlation between Somatic and Cognitive

Selfconfidence_Cognitive Correlation between Self Confidence and Cognitive

Selfconfidence_Somatic Correlation between Self Confidence and Somatic

Examples

```
becker09
```

---

c_mat_ft	*c mat ft*

Description

c mat ft

Usage

```
c_mat_ft(model_input, R)
```

Arguments

model_input	Model input as a character string. Multiple models need to be on their own line. Model syntax uses lavann like syntax, see details for more details about this syntax.
R	A correlation matrix, most likely this will be the average correlation matrix outputted from the metafor package.

Value

A list.

Examples

```
Br <- matrix(c(1.00000000, -0.09773331, -0.1755029,  0.3186775,
-0.09773331,  1.00000000,  0.5271873, -0.4175596,
-0.17550292,  0.5271872,  1.0000000, -0.4006848,
0.31867753, -0.41755963, -0.4006848,  1.0000000),
nrow = 4, byrow = TRUE)

colnames(Br) <- c("Performance",  "Self_confidence",  "Cognitive", "Somatic" )

rownames(Br) <- colnames(Br)

## Proposed path model
model <- "## Regression paths
Performance ~  Self_confidence  + Cognitive  + Somatic
Self_confidence ~ Cognitive + Somatic "

c_mat_ft(model, Br)
```

---

df_to_corr   *Helper function to read in data*

Description

This function builds a list of correlation matrices from data that are in flat files where a correlation matrix is coded into a single row, a common format for coding correlation matrices.

Usage

```
df_to_corr(data, variables, ID = NULL)
```

Arguments

data	Data frame input used to convert to correlation matrices.
variables	A character vector of variable names representing the columns to convert to a pairwise correlation matrix. Variable names take the following format: "column-name_rowname" where the name prior to the underscore is the column name and the name after the underscore is the row name.

ID   A variable name, as a character string, to use as names for the list elements.

Value

A list of correlation matrices.

Examples

```
becker09 <- read.csv(paste0(system.file('raw_data', package = 'mars'),
   '/Becker09.csv'))
becker09_list <- df_to_corr(becker09,
   variables = c('Cognitive_Performance',
                 'Somatic_Performance',
                 'Selfconfidence_Performance',
                 'Somatic_Cognitive',
                 'Selfconfidence_Cognitive',
                 'Selfconfidence_Somatic'),
     ID = 'ID')
```

---

estimation   *Master Estimation Function*

Description

The primary estimation function for conducting the optimization. The function is typically called through the mars function, but can be called here directly.

Usage

```
estimation(
  formula = NULL,
  effect_name = NULL,
  studyID = NULL,
  effectID = NULL,
  variance = NULL,
  data,
  estimation_method = "REML",
  optim_method = "L-BFGS-B",
  structure = "UN",
  varcov_type,
  weights = NULL,
  intercept = FALSE,
  N = NULL,
  missing = "remove",
  robustID = NULL,
  multivariate_covs = NULL,
  tol = 1e-10,
  ...
)
```

Arguments

`formula`	The formula used for specifying the fixed and random structure. Used for univariate and multilevel structures.
`effect_name`	Character string representing the name of the effect size column in the data.
`studyID`	Character string representing the study ID
`effectID`	Character string representing the effect size ID
`variance`	Character string representing the name of the variance of the effect size in the data.
`data`	Data used for analysis
`estimation_method`	
	Type of estimation used, either "REML" or "MLE", REML is the default
`optim_method`	Optimization method that is passed to the optim function. Default is 'L-BFGS-B'.
`structure`	Between studies covariance structure, default is "UN" or unstructured. See details for more specifics.
`varcov_type`	Type of variance covariance matrix computed. Default is 'cor_weighted' for correlations or 'smd_outcome' for standardized mean differences.
`weights`	User specified matrix of weights.
`intercept`	Whether a model intercept should be specified, default is FALSE meaning no intercept. See details for more information.
`N`	Character string representing the sample size of the studies.
`missing`	What to do with missing data, default is 'remove'
`robustID`	A character vector specifying the cluster group to use for computing the robust standard errors.
`multivariate_covs`	
	A one-sided formula to specify the covariates used in a multivariate analysis.
`tol`	Tolerance for estimating, passed to `optim`
`...`	Additional arguments to pass to `optim`.

Value

Output is a named list; The output returns the estimated parameters, fit statistics, estimation inputs.

`find_B`	*Estimate regression coefficients*

Description

Estimate regression coefficients

Usage

```
find_B(model_input, R)
```

Arguments

`model_input`	Model input as a character string. Multiple models need to be on their own line. Model syntax uses lavann like syntax, see details for more details about this syntax.
`R`	A correlation matrix, most likely this will be the average correlation matrix outputted from the metafor package.

Details

Coming soon.

The output will be the same length as the number of regression equations specified in the model_input argument.

Value

A list of parameter estimates

Examples

```
Br <-  matrix(c(1.00000000, -0.09773331, -0.1755029,  0.3186775,
-0.09773331,  1.00000000,  0.5271873, -0.4175596,
-0.17550292,  0.5271872,  1.0000000, -0.4006848,
0.31867753, -0.41755963, -0.4006848,  1.0000000),
nrow = 4, byrow = TRUE)

colnames(Br) <- c("Performance",  "Self_confidence",  "Cognitive", "Somatic" )

rownames(Br) <- colnames(Br)

## Proposed path model
model <- "## Regression paths
Performance ~  Self_confidence  + Cognitive  + Somatic
Self_confidence ~ Cognitive + Somatic "

find_B(model, Br)
```

---

find_reg_coef	*Find regression coefficients*

Description

Find regression coefficients

Usage

```
find_reg_coef(model_input, R, ...)
```

Arguments

`model_input`	Model input as a character string. Multiple models need to be on their own line. See details for specification.
`R`	A correlation matrix, most likely this will be the average correlation matrix outputted from the metafor package.
`...`	Currently not used

Value

A vector of regression coefficient estimates.

---

mars   *Master mars function*

Description

The primary function used for input and estimation. The function takes the data inputs and routes the estimation and structure type based on data structure. The function can handle univariate, multivariate, longitudinal, and multilevel meta-analytic models.

Usage

```
mars(
  data,
  studyID,
  effectID,
  sample_size,
  effectsize_type = NULL,
  formula = NULL,
  variable_names = NULL,
  effectsize_name = NULL,
  estimation_method = "REML",
  variance = NULL,
  varcov_type,
  weights = NULL,
  structure = "UN",
  intercept = FALSE,
  missing = "remove",
  optim_method = "L-BFGS-B",
  robustID = NULL,
  multivariate_covs = NULL,
  tol = 1e-10,
  ...
)
```

Arguments

`data`	Data used for analysis
`studyID`	Character string representing the study ID
`effectID`	Character string representing the effect size ID
`sample_size`	Character string representing the sample size of the studies.
`effectsize_type`	Type of effect size being analyzed
`formula`	The formula used for specifying the fixed and random structure. Used for univariate and multilevel structures.
`variable_names`	Vector of character strings representing the attributes with correlations. The attributes that are correlated should be separated by an underscore.
`effectsize_name`	Character string representing the name of the effect size column in the data.
`estimation_method`	Type of estimation used, either "REML" or "MLE", REML is the default
`variance`	Character string representing the name of the variance of the effect size in the data.
`varcov_type`	Type of variance covariance matrix computed. Default is 'cor_weighted' for correlations or 'smd_outcome' for standardized mean differences.
`weights`	User specified matrix of weights for analysis.
`structure`	Between studies covariance structure, default is "UN" or unstructured. See details for more specifics.
`intercept`	Whether a model intercept should be specified, default is FALSE meaning no intercept. See details for more information.
`missing`	Whether missing data should be removed, or kept. Default is removing.
`optim_method`	Optimization method that is passed to the optim function. Default is 'L-BFGS-B'.
`robustID`	A character vector specifying the cluster group to use for computing the robust standard errors.
`multivariate_covs`	A one-sided formula to specify the covariates used in a multivariate analysis.
`tol`	Tolerance of the optimization, default is 1E-10.
`...`	Not currently used.

Value

Returns a list of class mars; The returned object contains elements from the estimation.

---

math_correlations   *Math Correlations Data*

Description

This data contains correlations on math ...

Usage

```
math_correlations
```

Format

A nested list with correlations and sample size:

math_correlations A list of math correlations

sample_size Study sample size

Examples

```
math_correlations
```

---

model_fit   *Model fitting function*

Description

Model fitting function

Usage

```
model_fit(model_input, R, method_null = "sem", N)
```

Arguments

model_input	Model input as a character string. Multiple models need to be on their own line. See details for specification.
R	A correlation matrix, most likely this will be the average correlation matrix outputted from the metafor package.
method_null	Unsure
N	Sample size

Value

A list of fit indices.

Examples

```
Br <-  matrix(c(1.00000000, -0.09773331, -0.1755029,  0.3186775,
-0.09773331,  1.00000000,  0.5271873, -0.4175596,
-0.17550292,  0.5271872,  1.0000000, -0.4006848,
0.31867753, -0.41755963, -0.4006848,  1.0000000),
nrow = 4, byrow = TRUE)

colnames(Br) <- c("Performance",  "Self_confidence",  "Cognitive", "Somatic" )

rownames(Br) <- colnames(Br)

## Proposed path model
model <- "## Regression paths
Performance ~  Self_confidence  + Cognitive  + Somatic
Self_confidence ~ Cognitive + Somatic "

N <- 573
model_fit(model_input = model, R = Br,
        method_null = "sem", N)
model_fit(model_input = model, R = Br,
        method_null = "sem", N )
```

Mul_R2   *Mul R2*

Description

Mul R2

Usage

```
Mul_R2(model_input, R)
```

Arguments

model_input	Model input as a character string. Multiple models need to be on their own line. Model syntax uses lavann like syntax, see details for more details about this syntax.
R	A correlation matrix, most likely this will be the average correlation matrix out-putted from the metafor package.

Value

A vector of names

---

olkin_siotani   *Olkin & Siotani variance-covariance matrix*

Description

Computational function to compute the Olkin & Siotani (1976) variance-covariance matrix for correlation matrices. It allows the user to specify three different computations.

Usage

```
olkin_siotani(data, n, type = c("average", "weighted", "simple"))
```

Arguments

data	A correlation matrix or a list of correlation matrices.
n	Sample size
type	The type of Olkin and Siotani correction to make.

Details

The three possible computations that can be specified are:

- average: Average all the correlations element-wise to pool into a single correlation matrix. The variance-covariance is computed from the averaged correlation matrix, then divided by study specific sample sizes.

- weighted: Same as the average process-wise, but uses a weighted average to pool into a single correlation matrix.

- simple: Computes the variance-covariance for each individual correlation matrix, then divide these by the study specific sample sizes.

Value

List of matrices, same length as the number of studies or number of correlation matrices.

References

Becker, B. J. (1992). Using results from replicated studies to estimate linear models. Journal of Educational Statistics, 17(4), 341-362. Olkin, I. (1976). Asymptotic distribution of functions of a correlation matrix. Essays in provability and statistics: A volume in honor of Professor Junjiro Ogawa.

OS   *Olkin & Siotani 1976 variance-covariance numerator*

Description

Computation for the numerator of the variance-covariance matrix.

Usage

```
OS(data)
```

Arguments

data   A correlation matrix.

Value

A matrix representing the variance-covariance matrix.

---

path_model   *Path Model Function*

Description

This function fits the path model and returns adjusted standard errors.

Usage

```
path_model(
  mars_object,
  model,
  num_obs = NULL,
  adjust_se = TRUE,
  method_null = "sem",
  ...
)
```

Arguments

mars_object   The mars fitted object.

model   This is model syntax specified in the format by lavaan.

num_obs   Number of observations

adjust_se   Adjust the standard errors to reflect the ...

method_null   Unsure

...   Currently not used.

Details

The input is the coefficients and the variance covariance matrix returned from the [mars] function.

Value

List output with class path; The output is the parameter estimates from the fitted path model.

---

school   *School Data*

Description

This data ...

Usage

    school

Format

A dataframe with 56 rows and 5 columns:

district District ID

Examples

    school

---

smd_data   *SMD Simulated Data*

Description

This data ...

Usage

    smd_data

Format

A dataframe with 14 rows and 7 columns:

study Study ID

Examples

    smd_data

Index