Package ‘MAGMA.R’

January 20, 2025

Title MAny-Group MAtching

Version 1.0.3

Description Balancing quasi-experimental field research for effects of covariates is fundamental for drawing causal inference. Propensity Score Matching deals with this issue but current techniques are restricted to binary treatment variables. Moreover, they provide several solutions without providing a comprehensive framework on choosing the best model. The MAGMA R-package addresses these restrictions by offering nearest neighbor matching for two to four groups. It also includes the option to match data of a 2x2 design. In addition, MAGMA includes a framework for evaluating the post-matching balance. The package includes functions for the matching process and matching reporting. We provide a tutorial on MAGMA as vignette. More information on MAGMA can be found in Feuchter, M. D., Urban, J., Scherrer V., Breit, M. L., and Preckel F. (2022) <https://osf.io/p47nc/>.

Depends R (>= 4.2.0), tidyverse (>= 2.0.0)

License GPL-3

Imports parallel (>= 4.2), doParallel (>= 1.0.17), foreach (>= 1.5.2), metafor (>= 4.4-0), robumeta (>= 2.1), psych (>= 2.3.9), ggplot2 (>= 3.4.4), janitor (>= 2.2.0), flextable (>= 0.9.4) , overlapping (>= 2.1), purrr (>= 1.0.1), tibble (>= 3.2.1), dplyr (>= 1.1.4), tidyselect (>= 1.2.0), stats (>= 4.2), rlang (>= 1.1.1), stddiff (>= 3.1)

LazyData true

Encoding UTF-8

RoxygenNote 7.2.3

Suggests knitr, rmarkdown

VignetteBuilder knitr

Author Julian Urban [aut, cre],
Markus D. Feuchter [aut],
Vsevolod Scherrer [aut],
Moritz L. Breit [aut],
Franzis Preckel [aut]

Maintainer Julian Urban <urbanj@uni-trier.de>

BugReports https://github.com/JulianUrban/MAGMA/issues

NeedsCompilation no

Repository CRAN

Date/Publication 2024-08-27 07:10:02 UTC

Contents

`adj_d_ratio` *adj_d_ratio*

Description

adjusted d-ratio with respect to sample size.

Usage

```
adj_d_ratio(input)
```

Arguments

`input`	An inner d object

Details

This function computed the adjusted d_ratio with respect to sample size.

Value

A vector containing the adjusted d-ratio in dependency of sample size.

Author(s)

Julian Urban

`Balance_extract` *Balance_extract*

Description

This function extracts the balance criteria or pairwise effects of a `Balance_MAGMA` result for a specified sample size.

Usage

```
Balance_extract(Balance, samplesize, effects = FALSE)
```

Arguments

`Balance`	A result of `Balance_MAGMA` See the function `Balance_MAGMA` for details.
`samplesize`	An integer indicating the sample size for which the balance criteria or pairwise effects should be extracted.
`effects`	Indicates whether balance criteria or pairwise effects should be extracted. The default value *FALSE* returns the balance criteria, while *TRUE* leads to the extraction of pairwise effects.

Details

Given a previous computed `Balance_MAGMA` object, this function enables the extraction of balance criteria or pairwise effects for any desired sample size. This makes it possible to independently check the balance for each possible sample size.

Value

Depends on the effects argument. If *FALSE*, it returns in a vector containing the balance criteria. If *TRUE*, it returns a vector containing all possible pairwise effects.

Author(s)

Julian Urban

Examples

```
# Defining the names of the metric and binary covariates
covariates_vector <- c("GPA_school", "IQ_score", "Motivation", "parents_academic", "gender")


# Estimating balance of a two-group matching using the data set
# 'MAGMA_sim_data'.
# Matching variable 'gifted_support' (received giftedness support yes or no)
Balance_gifted <- Balance_MAGMA(Data = MAGMA_sim_data[MAGMA_sim_data$step_gifted < 200, ],
                                group = "gifted_support",
                                covariates = covariates_vector,
                                step = "step_gifted")

# Balance criteria for 100 cases per group
# Balance criteria
Balance_100_gifted <- Balance_extract(Balance = Balance_gifted,
                                      samplesize = 100,
                                      effects = FALSE)
Balance_100_gifted

# Pairwise effects
Balance_100_gifted_effects <- Balance_extract(Balance = Balance_gifted,
                                              samplesize = 100,
                                              effects = TRUE)
Balance_100_gifted_effects


# 2x2 matching using the data set 'MAGMA_sim_data'
# Matching variables are 'gifted_support' (received giftedness support yes
# or no) and 'enrichment' (participated in enrichment or not)
# 'MAGMA_sim_data_gift_enrich' contains the result of the matching
# 2x2 matching is equivalent to four-group matching
MAGMA_sim_data_gift_enrich <- MAGMA(Data = MAGMA_sim_data,
                                    group = c("gifted_support", "enrichment"),
                                    dist = "ps_2x2",
                                    cores = 2)


# Estimating balance. Covariates same as above
Balance_2x2 <- Balance_MAGMA(Data = MAGMA_sim_data_gift_enrich,
                             group = c("gifted_support", "enrichment"),
                             covariates = covariates_vector,
                             step = "step") # Step created during matching
str(Balance_2x2)

# Balance criteria for 125 cases per group
# Balance criteria
Balance_125_2x2 <- Balance_extract(Balance = Balance_2x2,
                                   samplesize = 125,
                                   effects = FALSE)
Balance_125_2x2
```

```
# Pairwise effects
Balance_125_2x2_effects <- Balance_extract(Balance = Balance_2x2,
                                           samplesize = 125,
                                           effects = TRUE)
Balance_125_2x2_effects
```

Balance_MAGMA *Balance_MAGMA*

Description

This function computes all four balance criteria of 'MAGMA.R', namely *Pillai's Trace*, *d-ratio*, *mean g*, and *adjusted d-ratio*. The estimation considers the scale level of the variables. Balance estimation is performed across various sample sizes. See Details for more information.

Usage

```
Balance_MAGMA(
  Data,
  group,
  covariates,
  step = "step",
  verbose = TRUE,
  covariates_ordinal = NULL,
  covariates_nominal = NULL
)
```

Arguments

Data	A data frame containing at least the *grouping* variable, the *step* variable from the main MAGMA-function (or other matching algorithms), and all *covariates* of interest.
group	A character specifying the name of your grouping variable in data. Note that MAGMA can only match your data for a maximum of four groups. For matching over two grouping variables (e.g., 2x2 design) is possible by specifying group as a character vector with a length of two. In this case, each of the two grouping variables can only have two levels.
covariates	A character vector listing the names of all binary and metric covariates of interest.
step	A character specifying the step variable of the matching. Per default, it is set to *step*, which corresponds the resulting name of the main MAGMA function.
verbose	TRUE or FALSE indicating whether matching information should be printed to the console.
covariates_ordinal	A character vector listing the names of all ordinal covariates of interest.

`covariates_nominal`
A character vector listing the names of all nominal covariates of interest.

Details

This function computes all four balance criteria of 'MAGMA.R', namely Pillai's Trace, d-ratio, mean g, and adjusted d-ratio. This is an iterative process including more cases with each iteration according to the step variable. Thus, starting with cases having a small within-match distance, larger distances are included with increasing iterations. As a minimum the function specifies n >= 20 per group. This does not imply that balance criteria with such a small sample size can be estimated consistently. For Pillai's Trace a higher minimum sample size can be possible. It depends on the number of covariates to ensure a positive model identification. Missing data for Pillai's Trace are excluded listwise, while for the other balance criteria pairwise exclusion is applied.

Value

A list of length four containing all balance criteria and all pairwise effects with respect to group sample size.

Author(s)

Julian Urban

References

Pastore, M., Loro, P.A.D., Mingione, M., Calcagni, A. (2022). *overlapping: Estimation of Overlapping in Empirical Distributions*. R package version 2.1, (https://CRAN.R-project.org/package=overlapping).

Revelle, W. (2023). *psych: Procedures for Psychological, Psychometric, and Personality Research*. Northwestern University, Evanston, Illinois. R package version 2.3.6, (https://CRAN.R-project.org/package=psych).

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. Journal of Statistical Software, 36(3), 1-48. (doi:10.18637/jss.v036.i03)

Fisher, Z., Tipton, E., Zhipeng, H. (2023). *robumeta: Robust Variance Meta-Regression*. R package version 2.1, (https://CRAN.R-project.org/package=robumeta).

Examples

```
# Defining the names of the metric and binary covariates
covariates_vector <- c("GPA_school", "IQ_score", "Motivation", "parents_academic", "gender")


# Estimating balance of a two-group matching using the data set
# 'MAGMA_sim_data'.
# Matching variable 'gifted_support' (received giftedness support yes or no)
# Using subsample only for faster execution.
Balance_gifted <- Balance_MAGMA(Data = MAGMA_sim_data[MAGMA_sim_data$step_gifted < 200, ],
                                group = "gifted_support",
                                covariates = covariates_vector,
                                step = "step_gifted")
```

```
str(Balance_gifted)


# 2x2 matching using the data set 'MAGMA_sim_data'
# Matching variables are 'gifted_support' (received giftedness support yes
# or no) and 'enrichment' (participated in enrichment or not)
# 'MAGMA_sim_data_gift_enrich' contains the result of the matching
# 2x2 matching is equivalent to four-group matching
MAGMA_sim_data_gift_enrich <- MAGMA(Data = MAGMA_sim_data,
                                    group = c("gifted_support", "enrichment"),
                                    dist = "ps_2x2",
                                    cores = 2)


# Estimating balance. Covariates same as above
Balance_2x2 <- Balance_MAGMA(Data = MAGMA_sim_data_gift_enrich,
                             group = c("gifted_support", "enrichment"),
                             covariates = covariates_vector,
                             step = "step") # step created during matching
str(Balance_2x2)
```

---

build_value_matrix *build_value_matrix*

Description

prepares distance estimation.

Usage

```
build_value_matrix(input_list, rep_element, name_ps = "distance_ps")
```

Arguments

`input_list`	Data with PS
`rep_element`	sample sizes per group
`name_ps`	names of PS in data

Details

This function uses the PS inputs to prepare them for distance estimation.

Value

the input for distance estimation

cohen_d *cohen_d*

Description

This function estimates Cohen's d in MAGMA_desc.

Usage

```
cohen_d(Data, index_1, index_2)
```

Arguments

Data	A data frame that contains sample sizes, means, and standard deviations.
index_1	Number of group 1.
index_2	Number of group_2

Details

Inner function of MAGMA_desc that computes Cohen's d using the pooled SD.

Value

A vector of pairwise Cohen'ds.

Author(s)

Julian Urban

Density_overlap *Density_overlap*

Description

This function shows and quantifies the kernel density overlap of a variable for two or more groups.

Usage

```
Density_overlap(
  Data,
  variable,
  group,
  variable_name = NULL,
  group_labels = NULL,
  group_name = NULL,
```

```
    step_num = NULL,
    step_var = NULL,
    verbose = TRUE
  )
```

Arguments

`Data`	A data frame that contains the desired variable for density plotting as well as the specified grouping variable.
`variable`	A character specifying the variable for which the density should be plotted (e.g., "ps_gifted").
`group`	A character specifying the groups for which the density should be plotted. Can be an independent group comparison (e.g., comparing matched groups) or the comparison of pre and post matched samples.
`variable_name`	A character specifying the name to appear in the plot for the variable.
`group_labels`	A character vector specifying the labels for the groups to ppear in the legend of the plot.
`group_name`	A character specifying the name of the grouping variable to appear in the title of the legend.
`step_num`	An integer specifying the number of cases to be included per group in this post matching comparison. Is based on the step variable of MAGMA.
`step_var`	A character specifying the name of the step variable.
`verbose`	TRUE or FALSE indicating whether matching information should be printed to the console.

Details

This function enables the comparison of the density of variables for two or more groups. It plots the kernel density separately for each group and quantifies the amount of overlap.

Value

A plot showing the kernel density for a specified variable separately for specified groups and the quantification of this overlap.

Author(s)

Julian Urban

References

Pastore, M., Loro, P.A.D., Mingione, M., Calcagni, A. (2022). *overlapping: Estimation of Overlapping in Empirical Distributions*. R package version 2.1, https://CRAN.R-project.org/package=overlapping.

Examples

```
# Estimating density overlap using the data set 'MAGMA_sim_data
# Estimating density overlap for 'ps_gifted' (propensity scores for
# giftedness support)
# Defining plot aesthetics with 'group', 'variable_name', "group_lables',
# and 'group_name'
# Estimating pre-matching density overlap by not specifying 'step_num' and
# 'step_var'
Density_overlap(Data = MAGMA_sim_data,
variable = "ps_gifted",
group = "gifted_support",
step_num = NULL,
step_var = NULL,
variable_name = "Propensity Score",
group_labels = c("No Support", "Support"),
group_name = "Gifted Support")


# Estimating density overlap using the matched data set
#'MAGMA_sim_data_gifted'
# Estimating density overlap for 'ps_gifted' (propensity scores for
# giftedness support)
# Defining plot aesthetics with 'group', 'variable_name', 'group_lables',
# and 'group_name'
# Estimating post-matching overlap for 250 cases per group ('step_num')
# Name of the step variable is 'step'
Density_overlap(Data = MAGMA_sim_data,
variable = "ps_gifted",
group = "gifted_support",
step_num = 250,
step_var = "step_gifted",
variable_name = "Propensity Score",
group_labels = c("No Support", "Support"),
group_name = "Gifted Support")

# Estimating density overlap using the data set 'MAGMA_sim_data
# Estimating density overlap for 'teacher_ability_rating' (ability rated
# from teacher as below average, average, or above average)
# Defining plot aesthetics with 'group', 'variable_name', 'group_lables',
# and 'group_name'
# Estimating pre-matching density overlap by not specifying 'step_num' and
# 'step_var'
Density_overlap(Data = MAGMA_sim_data,
variable = "GPA_school",
group = "teacher_ability_rating",
variable_name = "School Achievement",
group_labels = c("Low", "Medium", "High"),
group_name = "Rating")
```

distance_estimator *distance_estimator*

Description

estimates distance in MAGMA.

Usage

```
distance_estimator(data, means, variance, cores, inp = NULL)
```

Arguments

data	A matrix containing all possible combinations.
means	A matrix containing all row means of all possible matches.
variance	A numeric indicating the variance of the propensity scores.
cores	An integer defining the number of cores used for parallel computation.
inp	input parameter for parallel distance computation.

Details

This function is an inner function of MAGMA. It estimates the distance of all possible matches.

Value

A matrix of distance for each case of each possible match.

Author(s)

Julian Urban

effect_nominal *effect_nominal*

Description

Computes descriptive statistics and effect size for nominal data

Usage

```
effect_nominal(Data, group, variable)
```

Arguments

`Data`	A data set
`group`	A character specifying the grouping variable
`variable`	Variables for which the effect should be estimated

Details

This function computes descriptive statistics and effect size for nominal data.

Value

A vector or matrix containing nominal effect sizes

Author(s)

Julian Urban

`effect_ordinal` *effect_ordinal*

Description

Computes descriptive statistics and effect size for ordinal data

Usage

```
effect_ordinal(Data, group, variable)
```

Arguments

`Data`	A data set
`group`	A character specifying the grouping variable
`variable`	Variables for which the effect should be estimated

Details

This function computes descriptive statistics and effect size for ordinal data.

Value

A vector or matrix containing ordinal effect sizes

Author(s)

Julian Urban

`initial_unbalance` *initial_unbalance*

Description

This function computes all four balance criteria of 'MAGMA.R,' namely *Pillai's Trace*, *d-ratiO*, *mean g*, and *adjusted d-ratio* for the unmatched data set. This enables comparison of initial unbalance with the balance after matching.

Usage

```
initial_unbalance(
  Data,
  group,
  covariates,
  verbose = TRUE,
  covariates_ordinal = NULL,
  covariates_nominal = NULL
)
```

Arguments

`Data`	A data frame containing at least the *grouping* variable and all *covariates* of interest.
`group`	A character specifying the name of your grouping variable in data. Note that MAGMA can only match your data for a maximum of 4 groups. For matching over two grouping variables (e.g., 2x2 design) is possible by specifying group as a character vector with a length of two. In this case each or the two grouping variables can only have two levels.
`covariates`	A character vector listing the names of all binary and metric covariates of interest.
`verbose`	TRUE or FALSE indicating whether matching information should be printed to the console.
`covariates_ordinal`	A character vector listing the names of all ordinal covariates of interest.
`covariates_nominal`	A character vector listing the names of all nominal covariates of interest.

Details

This function computes all four Balance criteria of 'MAGMA.R', namely Pillai's Trace, d-ratio, mean g, and adjusted d-ratio for the overall samples. Missing data for Pillai's Trace are excluded listwise, while for the other balance criteria pairwise exclusion is applied.

Value

A numeric vector of length 4 containing the balance criteria for the unmatched sample.

Author(s)

Julian Urban

References

Pastore, M., Loro, P.A.D., Mingione, M., Calcagni, A. (2022). *overlapping: Estimation of Overlapping in Empirical Distributions*. R package version 2.1, (https://CRAN.R-project.org/package=overlapping).

Revelle, W. (2023). *psych: Procedures for Psychological, Psychometric, and Personality Research*. Northwestern University, Evanston, Illinois. R package version 2.3.6, (https://CRAN.R-project.org/package=psych)

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. Journal of Statistical Software, 36(3), 1-48. (doi:10.18637/jss.v036.i03)

Fisher, Z., Tipton, E., Zhipeng, H. (2023). *robumeta: Robust Variance Meta-Regression*. R package version 2.1, (https://CRAN.R-project.org/package=robumeta).

Examples

```
# Defining covariates for balance estimation
covariates_vector <- c("GPA_school", "IQ_score", "Motivation", "parents_academic", "gender")

# Computing initial unbalance using the data set 'MAGMA_sim_data'
# Computing initial unbalance for the variable 'gifted_support' (received
# giftedness support yes or no)
unbalance_gifted <- initial_unbalance(Data = MAGMA_sim_data,
                                      group = "gifted_support",
                                      covariates = covariates_vector)
unbalance_gifted

# Computing initial unbalance using the data set 'MAGMA_sim_data'
# Computing initial unbalance for the variable 'teacher_ability_rating'
# (ability rated from teacher as below average, average, or above average)
unbalance_tar <- initial_unbalance(Data = MAGMA_sim_data,
                                   group = "teacher_ability_rating",
                                   covariates = covariates_vector)
unbalance_tar

# Computing initial unbalance using the data set 'MAGMA_sim_data'
# Computing initial unbalance for the variables 'gifted_support' (received
# giftedness support yes or no) and 'enrichment' (participated in enrichment
# or not)
unbalance_2x2 <- initial_unbalance(Data = MAGMA_sim_data,
                                   group = c("gifted_support", "enrichment"),
                                   covariates = covariates_vector)
unbalance_2x2
```

`inner_d` *inner_d*

Description

d-ratio and pairwise Cohen's d with respect to sample size.

Usage

```
inner_d(da, gr, co, st, co_ord = NULL, co_nom = NULL)
```

Arguments

`da`	Specifying the data frame or tibble with the data.
`gr`	A character vector specifying the IVs.
`co`	A character vector naming the DVs.
`st`	A character naming the variable for iteratively inclusion
`co_ord`	A character vector naming the ordinal DVs.
`co_nom`	A character vector naming the nominal DVs.

Details

This function computed the d-ratio and all pairwise effects with respect to sample size.

Value

A list of length two. The first element is a matrix including all pairwise effects. The second is a vector expressing d-ratio in dependency of sample size.

Author(s)

Julian Urban

`J_group_size` *J_group_size*

Description

inner function of `mean_g_meta`.

Usage

```
J_group_size(group_size)
```

Arguments

`group_size`	A numeric defining the max sample size for which J should be computed.

Details

This function computes J over samples sizes necessary for Hedges' g.

Value

A vector of J's in dependency of sample size.

Author(s)

Julian Urban

`MAGMA` *MAGMA*

Description

This function conducts many group matching for 2 to 4 groups. It augments the original data set by the relevant 'MAGMA.R' variables. For details, see below.

Usage

```
MAGMA(Data, group, dist, cores = 1, verbose = TRUE)
```

Arguments

`Data`	A data frame or tibble containing at least your grouping and distance variable. Data needs to be specified in your environment.
`group`	A character specifying the name of your grouping variable in the data. Note that MAGMA can only match your data for a maximum of 4 groups. Matching over two grouping variables (e.g., 2x2 Design) is possible by specifying group as a character vector with a length of two. In this case, each or the 2 grouping variables can only have two levels.
`dist`	A character specifying the name of your distance variable in data.
`cores`	An integer defining the number of cores used for parallel computation.
`verbose`	TRUE or FALSE indicating whether matching information should be printed to the console.

Details

This function conducts nearest neighbor many group matching. It is applicable for two to four groups or a 2x2 design. As output, this function augments your original data by the variables *weight*, *step*, *distance*, and *ID*. Weight indicates whether a case was matched. Step specifies the iteration in which a case was matched. It also shows which cases were matched together. Distance indicates the mean difference within a match. Since matches with a lower distance are matched in an earlier iteration, step and distance are strongly correlated. This function has some CPU and RAM load. In most four-group applications and three-group applications with large sample size, RAM may be not sufficient. Therefore MAGMA switches to random quasi-systematic matching. If this is the case, MAGMA informs you. The output of the function does not change, but balance might be slightly affected.

Value

Your input data frame augmented with matching relevant variables, namely *weight*, *step*, *distance*, and *ID*. In case of missing values on the distance or group variable, MAGMA excludes them for the matching process. The returned data set does not contain those excluded cases. For more information, see Details.

Author(s)

Julian Urban

Examples

```
# Running this code will take a while
# Two-group exact matching using the data set 'MAGMA_sim_data'
# Matching variable 'gifted_support' (received giftedness support yes or no)
# 'MAGMA_sim_data_gifted' contains the result of the matching
MAGMA_sim_data_gifted <- MAGMA(Data = MAGMA_sim_data,
                               group = "gifted_support",
                               dist = "ps_gifted",
                               cores = 1)
head(MAGMA_sim_data_gifted)


# Two-group exact matching using the data set 'MAGMA_sim_data'
# Matching variable 'teacher_ability_rating' (ability rated from teacher as
# below average, average, or above average)
# MAGMA_sim_data_tar' contains the result of the matching
# Cores per default = 1
MAGMA_sim_data_tar <- MAGMA(Data = MAGMA_sim_data,
                            group = "teacher_ability_rating",
                            dist = "ps_tar")
head(MAGMA_sim_data_tar)

# 2x2 matching using the data set 'MAGMA_sim_data'
# Matching variables are 'gifted_support' (received giftedness support yes
# or no) and 'enrichment' (participated in enrichment or not)
# 'MAGMA_sim_data_gift_enrich' contains the result of the matching
# 2x2 matching is equal to four-group matching
```

```
MAGMA_sim_data_gift_enrich <- MAGMA(Data = MAGMA_sim_data,
                                    group = c("gifted_support", "enrichment"),
                                    dist = "ps_2x2",
                                    cores = 2)
head(MAGMA_sim_data_gift_enrich)
```

MAGMA_desc *MAGMA_desc*

Description

This function provides pre- and post-matching descriptive statistics and effects.

Usage

```
MAGMA_desc(
  Data,
  covariates,
  group,
  step_num = NULL,
  step_var = NULL,
  filename = NULL,
  verbose = TRUE,
  covariates_ordinal = NULL,
  covariates_nominal = NULL
)
```

Arguments

`Data`	A data frame that contains the desired variable for density plotting as well as the specified grouping variable.
`covariates`	A character vector specifying the variable names of the binary and metric variables for which the descriptive statistics should be computed.
`group`	A character (vector) specifying the groups for which differentiated statistics should be computed.
`step_num`	An integer specifying the number of cases to be included per group in this post matching comparison (e.g., 100). If no value is specified, pre-matching statistics are computed. Is based on the step variable of MAGMA. Optional argument.
`step_var`	A character specifying the name of the step variable in the data set. If no value is specified, pre matching statistics are computed. Optional argument.
`filename`	A character specifying the filename that the resulting Word document with the Table should have. Optional argument.
`verbose`	TRUE or FALSE indicating whether matching information should be printed to the console.

covariates_ordinal
: A character vector specifying the variable names of the ordinal variables for which the descriptive statistics should be computed.

covariates_nominal
: A character vector specifying the variable names of the nominal variables for which the descriptive statistics should be computed.

Details

This function enables the computation of descriptive statistics for the overall sample and specified groups. Additional, pairwise effects according to the respective scale level are computed.

Value

A table of descriptive statistics and pairwise effects for pre- or post-matching samples.

Author(s)

Julian Urban

Examples

```
# Defining covariates
covariates_gifted <- c("GPA_school", "IQ_score", "Motivation", "parents_academic", "gender")

# Estimating pre-matching descriptive statistics and pairwise effects using
# the data set 'MAGMA_sim_data'
# Estimating statistics for grouping variable 'gifted support' (received
# giftedness support yes or no)
MAGMA_desc(Data = MAGMA_sim_data,
           covariates = covariates_gifted,
           group =  "gifted_support")


# Estimating post-matching descriptive statistics and pairwise effects using
# the data set 'MAGMA_sim_data'
# Estimating statistics for grouping variable 'gifted support' (received
# giftedness support yes or no)
# Estimating statistics for 100 cases per group
MAGMA_desc(Data = MAGMA_sim_data,
           covariates = covariates_gifted,
           group =  "gifted_support",
           step_num = 100,
           step_var = "step_gifted")
```

MAGMA_exact *MAGMA_exact*

Description

This function conducts exact many group matching for 2 to 4 groups. Exact means that only cases with the same value on the exact variable can be matched. It augments the original data set by relevant 'MAGMA.R' variables. For details, see below.

Usage

```
MAGMA_exact(Data, group, dist, exact, cores = 1, verbose = TRUE)
```

Arguments

`Data`	A data frame or tibble containing at least your grouping and distance variable. Data needs to be specified in your environment.
`group`	A character specifying the name of your grouping variable in the data. Note that MAGMA can only match your data for a maximum of 4 groups. Matching over two grouping variables (e.g., 2x2 Design) is possible by specifying group as a character vector with a length of two. In this case, each or the 2 grouping variables can only have two levels.
`dist`	A character specifying the name of your distance variable in data.
`exact`	A character specifying the name of the exact variable. Only cases with the same value on this variable can be matched.
`cores`	An integer defining the number of cores used for parallel computation.
`verbose`	TRUE or FALSE indicating whether matching information should be printed to the console.

Details

This function conducts nearest neighbor exact many group matching. It is applicable for two to four groups or a 2x2 design. As output, this function augments your original data by the variables *weight*, *step*, *distance*, and *ID*. Weight indicates whether a case was matched. Step specifies the iteration in which a case was matched. It also shows which cases were matched together. Distance indicates the mean difference within a match. Since matches with a lower distance are matched in an earlier iteration, step and distance are strongly correlated. Exact matching means that only cases with the same value on the exact variable can be matched. As example, only person of the same gender, the same school, or the same organization are possible matches. For standard matching, see `MAGMA`

Value

Your input data frame of valid cases augmented with matching relevant variables, namely *weight*, *step*, *distance*, and *ID*. In case of missing values on the distance or group variable, MAGMA_exact excludes them for the matching process. The returned data set does not contain those excluded cases. For more information, see Details.

Author(s)

Julian Urban

Examples

```
# Running this code will take a while
# Two-group exact matching using the data set 'MAGMA_sim_data'
# Matching variable 'gifted_support' (received giftedness support yes or no)
# 'MAGMA_sim_data_gifted_exact' contains the result of the matching
# Exact matching for 'enrichment' (participated in enrichment or not)
# Students that participated can only be matched with other
# students that participated and vice versa
MAGMA_sim_data_gifted_exact <- MAGMA_exact(Data = MAGMA_sim_data[c(1:20), ],
                                           group = "gifted_support",
                                           dist = "ps_gifted",
                                           exact = "enrichment",
                                           cores = 1)
head(MAGMA_sim_data_gifted_exact)


# Conducting three-group matching using the data set 'MAGMA_sim_data'
# Matching variable 'teacher_ability_rating' (ability rated from teacher as
# below average, average, or above average)
# 'MAGMA_sim_data_tar_exact' contains the result of the matching
# Exact matching for gender (male or female)
# Male students can only be matched to male students, female students can only
# be matched to female students
# Cores per default = 1
MAGMA_sim_data_tar_exact<- MAGMA_exact(Data = MAGMA_sim_data,
                                       group = "teacher_ability_rating",
                                       dist = "ps_tar",
                                       exact = "gender")
head(MAGMA_sim_data_tar_exact)

# 2x2 matching using the data set 'MAGMA_sim_data'
# Matching variables are 'gifted_support' (received giftedness support yes
# or no) and 'enrichment' (participated in enrichment or not)
# 'MAGMA_sim_data_gift_enrich_exact' contains the result of the matching
# 2x2 matching is equal to four-group matching
# Exact matching for for teacher rated ability (ability rated from teacher as
# below average, average, or above average)
# Below average students can only be matched to other below average rated
# students, average rated students can be matched with other average rated
# students, and above average rated students can only be matched to other
# above average rated students
MAGMA_sim_data_gift_enrich_exact <- MAGMA_exact(Data = MAGMA_sim_data,
                                              group = c("gifted_support", "enrichment"),
                                                dist = "ps_2x2",
                                                exact = "teacher_ability_rating",
                                                cores = 2)
head(MAGMA_sim_data_gift_enrich_exact)
```

`MAGMA_sim_data` *MAGMA simulated data set*

Description

The 'MAGMA.R' simulated data set contains 14 variables of 800 cases. It is used as example in the vignette and the help pages.

Usage

```
MAGMA_sim_data
```

Format

A data frame with 17 variables of 800 cases.

ID : Individual ID for each case

gender : Binary variable indicating gender of a participant

gifted_support : Binary variable that specifies whether a case received giftedness support (1) or not (0)

teacher_ability_rating : Three-step ordinal variable ranging from 1 to 3 indicating increasing eacher rated ability of a case

enrichment : Binary variable that indicates whether a case participated in an afternoon enrichment program

parents_academic : Binary variable that indicates whether at least one parent of a case has an academic background

GPA_school : Variable ranging from 1 to 6 that indicates a case's high school GPA. Lower values indicate higher achievement

IQ_score : Variable indicating the normed IQ score of a case

Motivation : The scale score of a case in a motivational questionnaire

college_GPA : variable ranging from 1 to 6 that indicates a case's college GPA. Lower values indicate higher achievement

support_enrichment : Multinomial variable representing the combination of gifted support and enrichment

ps_tar : Propensity score of twangs mnps function for teacher_ability_rating

ps_2x2 : Propensity score of twangs mnps function for support_enrichment

ps_gifted : Propensity score of twangs ps function for gifted support

step_gifted : Indicates step of MAGMA-matching for gifted support

weight_gifted : Indicates weight of MAGMA-matching for gifted support

distance_gifted : Indicates distance of MAGMA-matching for gifted support

Source

Simulated data

match_iterative *match_iterative*

Description

matches Cases iteratively during the matching process.

Usage

```
match_iterative(distance_input, output_list, rep_elements)
```

Arguments

distance_input	distance matrix to extract lowest distance
output_list	output where MAGMA results get stored
rep_elements	sample sizes per group

Details

This function conducts the matching process, by extracting the match with the lowest distance.

Value

A matched sample including the variables step, weight & distance

mean_g_meta *mean_g_meta*

Description

Mean standardized effect.

Usage

```
mean_g_meta(input, number_groups)
```

Arguments

input	An inner d object.
number_groups	A numeric specifying the number of groups.

Details

This function computes the mean effect. Method varies between two and many group matchings.

Value

A vector containing the mean g in dependency of sample size.

Author(s)

Julian Urban

`Pillai_iterativ` *Pillai_iterativ*

Description

Pillai's Trace with respect to sample size.

Usage

```
Pillai_iterativ(da, gr, co, st)
```

Arguments

`da`	Specifying the data frame or tibble with the data.
`gr`	A character vector specifying the IVs.
`co`	A character vector naming the DVs.
`st`	A character naming the variable for iteratively inclusion

Details

This function computes Pillai's Trace for increasing sample size.

Value

A vector containing Pillai's Trace in dependency of sample size. If two grouping variables were specified, the output is a matrix containing Pillai's Trace for both IVs and their ineraction.

Author(s)

Julian Urban

Plot_MAGMA *Plot_MAGMA*

Description

Plots for balance with respect to sample size.

Usage

```
Plot_MAGMA(
  Balance,
  criterion = c("Pillai", "d_ratio", "mean_g", "Adj_d_ratio")
)
```

Arguments

Balance	A result of Balance_MAGMA. Compare the function Balance_MAGMA.
criterion	A character vector specifying for which balance criteria a plot should be created. Default is all criteria.

Details

This function creates R-Plots using ggplot2 to show the balance trend over sample size.

Value

R Plots showing the balance trend over sample size.

Author(s)

Julian Urban

Examples

```
# This function bases on a MAGMA function as well as Balance_MAGMA
# To run examples, copy them into your console or script
# Defining the names of the metric and binary covariates
covariates_vector <- c("GPA_school", "IQ_score", "Motivation", "parents_academic", "gender")

#  Estimating balance of a two-group matching using the data set
# 'MAGMA_sim_data'.
# Matching variable 'gifted_support' (received giftedness support yes or no)
Balance_gifted <- Balance_MAGMA(Data = MAGMA_sim_data[MAGMA_sim_data$step_gifted < 150, ],
                                group = "gifted_support",
                                covariates = covariates_vector,
                                step = "step_gifted")

Plot_MAGMA(Balance = Balance_gifted,
           criterion = "Adj_d_ratio") #Using default to plot all criteria
```

```
# 2x2 matching using the data set 'MAGMA_sim_data'
# Matching variables are 'gifted_support' (received giftedness support yes
# or no) and 'enrichment' (participated in enrichment or not)
# 'MAGMA_sim_data_gift_enrich' contains the result of the matching
# 2x2 matching is equivalent to four-group matching
MAGMA_sim_data_gift_enrich <- MAGMA(Data = MAGMA_sim_data,
                                    group = c("gifted_support", "enrichment"),
                                    dist = "ps_2x2",
                                    cores = 2)


# Estimating balance. Covariates same as above
Balance_2x2 <- Balance_MAGMA(Data = MAGMA_sim_data_gift_enrich,
                             group = c("gifted_support", "enrichment"),
                             covariates = covariates_vector,
                             step = "step") #step created during matching

Plot_MAGMA(Balance = Balance_2x2,
           criterion = c("d_ration", "Adj_d_ratio"))
```

`row_nominal` *row_nominal*

Description

Computes descriptive statistics and effect size for nominal data

Usage

```
row_nominal(Data, group, variable)
```

Arguments

`Data`	A data set
`group`	A character specifying the grouping variable
`variable`	Variables for which the effect should be estimated

Details

This function computes descriptive statistics and effect size for nominal data.

Value

A vector containing the adjusted d-ratio in dependency of sample size.

Author(s)

Julian Urban

`row_ordinal` *row_ordinal*

Description

Computes descriptive statistics and effect size for ordinal data

Usage

```
row_ordinal(Data, group, variable)
```

Arguments

`Data`	A data set
`group`	A character specifying the grouping variable
`variable`	Variables for which the effect should be estimated

Details

This function computes descriptive statistics and effect size for ordinal data.

Value

A vector containing the adjusted d-ratio in dependency of sample size.

Author(s)

Julian Urban

`Table_MAGMA` *Table_MAGMA*

Description

This function prints an APA Table of the Balance criteria. It displays the balance criteria for four different sample sizes per group. In each scenario, one balance criteria has its optimal value. Thus, the table is a 4x5 table showing the four balance criteria and the respective sample size per group for the four scenarios.

Usage

```
Table_MAGMA(Balance, filename = NULL, verbose = TRUE)
```

Arguments

Balance	A result of Balance_MAGMA Compare the function Balance_MAGMA.
filename	Optional argument. A character specifying the filename that the resulting Word document with the table should have.
verbose	TRUE or FALSE indicating whether matching information should be printed to the console.

Details

This function creates an APA Table including the optimal models for each balance criterion, the other criteria for the respective sample size per group as well as the sample size itself. With an optional argument you can save a the APA table in Word.

Value

A 4x5 APA table showing the four balance criteria and the respective sample sizes per group for four scenarios. In each of these scenarios, one balance criteria has its optimal value. It can print a Word Document with this table.

Author(s)

Julian Urban

Examples

```
# This function bases on a MAGMA function as well as Balance_MAGMA
# Defining the names of the metric and binary covariates
covariates_vector <- c("GPA_school", "IQ_score", "Motivation", "parents_academic", "gender")


# Estimating balance of a two-group matching using the data set
# 'MAGMA_sim_data'.
# Matching variable 'gifted_support' (received giftedness support yes or no)
Balance_gifted <- Balance_MAGMA(Data = MAGMA_sim_data[MAGMA_sim_data$step_gifted < 200, ],
                                group = "gifted_support",
                                covariates = covariates_vector,
                                step = "step_gifted")

Table_MAGMA(Balance_gifted)


# 2x2 matching using the data set 'MAGMA_sim_data'
# Matching variables are 'gifted_support' (received giftedness support yes
# or no) and 'enrichment' (participated in enrichment or not)
# 'MAGMA_sim_data_gift_enrich' contains the result of the matching
# 2x2 matching is equivalent to four-group matching
MAGMA_sim_data_gift_enrich <- MAGMA(Data = MAGMA_sim_data,
                                    group = c("gifted_support", "enrichment"),
                                    dist = "ps_2x2",
                                    cores = 2)
```

```
# Estimating Balance. Covariates same as above
Balance_2x2 <- Balance_MAGMA(Data = MAGMA_sim_data_gift_enrich,
                             group = c("gifted_support", "enrichment"),
                             covariates = covariates_vector,
                             step = "step") #step created during matching

Table_MAGMA(Balance_2x2)
```

Index