Package 'MAIHDA'

May 16, 2026

Type Package

Title Multilevel Analysis of Individual Heterogeneity and Discriminatory Accuracy

Version 0.1.8

Description Provides a comprehensive toolkit for conducting Multilevel Analysis of Individual Heterogeneity and Discriminatory Accuracy (MAIHDA). Methods are described in Merlo (2018) <doi:10.1016/j.socscimed.2017.12.018> and Evans et al. (2018) <doi:10.1016/j.socscimed.2017.11.011>. Automatically generates intersectional strata, fits analytical models, extracts statistics, and produces visualizations.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 4.1.0)

Imports lme4 (>= 1.1-27), reformulas, ggplot2 (>= 3.3.0), dplyr (>= 1.0.0), tidyr (>= 1.1.0), stats, rlang (>= 0.4.0), patchwork, ggrepel, tidyselect, tibble

Suggests shiny, bslib, DT, future, promises, plotly, haven, shinyjs, brms (>= 2.15.0), testthat (>= 3.0.0), shinytest2, knitr, MASS, rmarkdown, boot (>= 1.3-20), ggtern

VignetteBuilder knitr

URL https://github.com/hdbt/MAIHDA, https://hdbt.github.io/MAIHDA/

BugReports https://github.com/hdbt/MAIHDA/issues

RoxygenNote 7.3.3

NeedsCompilation no

Author Hamid Bulut [aut, cre]

Maintainer Hamid Bulut <me@hamidbulut.com>

Repository CRAN

Date/Publication 2026-05-16 16:10:02 UTC

Contents

`calculate_pvc` *Calculate Proportional Change in Between-Stratum Variance (PVC)*

Description

Calculates the proportional change in between-stratum variance (PVC) between two MAIHDA models. The PVC measures how much the between-stratum variance changes when moving from one model to another, and is calculated as: PVC = (Var_model1 - Var_model2) / Var_model1

Usage

```
calculate_pvc(
  model1,
  model2,
  bootstrap = FALSE,
  n_boot = 1000,
  conf_level = 0.95
)
```

Arguments

model1	A maihda_model object from fit_maihda(). This is the reference model (typically a simpler or baseline model).
model2	A maihda_model object from fit_maihda(). This is the comparison model (typically a more complex model with additional predictors).
bootstrap	Logical indicating whether to compute bootstrap confidence intervals for PVC. Default is FALSE.
n_boot	Number of bootstrap samples if bootstrap = TRUE. Default is 1000.
conf_level	Confidence level for bootstrap intervals. Default is 0.95.

Details

The PVC is interpreted as the proportional reduction (or increase if negative) in between-stratum variance when moving from model1 to model2. A positive PVC indicates that model2 explains some of the between-stratum variance present in model1, while a negative PVC suggests that model2 has more unexplained between-stratum variance.

When bootstrap = TRUE, the function uses a parametric bootstrap: it simulates new responses from model2 and refits both models with lme4::refit() for each simulated response to obtain confidence intervals for the PVC estimate.

Value

A list containing:

pvc	The estimated proportional change in variance
var_model1	Between-stratum variance from model1
var_model2	Between-stratum variance from model2
ci_lower	Lower bound of confidence interval (if bootstrap = TRUE)
ci_upper	Upper bound of confidence interval (if bootstrap = TRUE)
bootstrap	Logical indicating if bootstrap was used

Examples

```
# Create strata and fit two models
strata_result <- make_strata(maihda_sim_data, c("gender", "race"))
model1 <- fit_maihda(health_outcome ~ age + (1 | stratum), data = strata_result$data)
model2 <- fit_maihda(health_outcome ~ age + gender + (1 | stratum), data = strata_result$data)

# Calculate PVC without bootstrap
pvc_result <- calculate_pvc(model1, model2)
print(pvc_result$pvc)

# Calculate PVC with bootstrap CI
# pvc_boot <- calculate_pvc(model1, model2, bootstrap = TRUE, n_boot = 500)
# print(pvc_boot)
```

compare_maihda *Compare MAIHDA Models*

Description

Compares variance partition coefficients (VPC/ICC) across multiple MAIHDA models, with optional bootstrap confidence intervals.

Usage

```
compare_maihda(
  ...,
  model_names = NULL,
  bootstrap = FALSE,
  n_boot = 1000,
  conf_level = 0.95
)
```

Arguments

...	Multiple maihda_model objects to compare.
model_names	Optional character vector of names for the models.
bootstrap	Logical indicating whether to compute bootstrap confidence intervals. Default is FALSE.
n_boot	Number of bootstrap samples if bootstrap = TRUE. Default is 1000.
conf_level	Confidence level for bootstrap intervals. Default is 0.95.

Value

A data frame comparing VPC/ICC across models with optional confidence intervals.

Examples

```
# Create strata and models using simulated data
strata_1 <- make_strata(maihda_sim_data, vars = c("gender", "race"))
strata_2 <- make_strata(maihda_sim_data, vars = c("gender", "race", "education"))

model1 <- fit_maihda(health_outcome ~ age + (1 | stratum), data = strata_1$data)
model2 <- fit_maihda(health_outcome ~ age + gender + (1 | stratum), data = strata_2$data)

# Compare without bootstrap
comparison <- compare_maihda(model1, model2,
                             model_names = c("Base", "With Gender"))

# Compare with bootstrap CI
comparison_boot <- compare_maihda(model1, model2,
                                  model_names = c("Base", "With Gender"),
```

```
                                bootstrap = TRUE, n_boot = 500)
```

`compute_maihda_ternary_data` *Compute Ternary Data for MAIHDA Models*

Description

Compute Ternary Data for MAIHDA Models

Usage

```
compute_maihda_ternary_data(
  model,
  summary_obj = NULL,
  scale = c("link", "response"),
  reference_values = NULL,
  uncertainty_method = c("auto", "se", "ci_width", "posterior_sd"),
  include_na_strata = FALSE,
  verbose = TRUE
)
```

Arguments

`model`	A fitted MAIHDA model object from ‘fit_maihda()‘.
`summary_obj`	Optional output from ‘summary()‘.
`scale`	Character, either "link" or "response".
`reference_values`	List or data.frame of reference values for covariates.
`uncertainty_method`	Character indicating how to extract uncertainty. "auto" uses conditional standard errors for lme4 models and posterior standard deviations for brms models. "ci_width" uses the 95% interval width.
`include_na_strata`	Logical, whether to include strata with missing data.
`verbose`	Logical, whether to print messages.

Value

A tidy tibble with ternary coordinates.

`fit_maihda` *Fit MAIHDA Model*

Description

Fits a multilevel model for MAIHDA (Multilevel Analysis of Individual Heterogeneity and Discriminatory Accuracy) using either lme4 or brms.

Usage

```
fit_maihda(
  formula,
  data,
  engine = "lme4",
  family = "gaussian",
  autobin = TRUE,
  ...
)
```

Arguments

`formula`	A formula specifying the model. Can include a random effect for stratum (e.g., `outcome ~ fixed_vars + (1 | stratum)`) or can directly specify the intersection variables to be used for forming strata (e.g., `outcome ~ fixed_vars + (1 | var1:var2:var3)`). If variables other than "stratum" are provided in the random effect, `make_strata` will be called internally to compute the strata and the formula will be updated.
`data`	A data frame containing the variables in the formula.
`engine`	Character string specifying which engine to use: "lme4" (default) or "brms".
`family`	Character string or family object specifying the model family. Common options: "gaussian", "binomial", "poisson". Default is "gaussian". If the outcome variable appears to be binary and the default family is used, the function will automatically switch to "binomial", recode two-level responses to 0/1 for `glmer()`, and issue a warning.
`autobin`	Logical indicating whether numeric variables used only for automatic strata creation should be binned by `make_strata`. Default is TRUE.
`...`	Additional arguments passed to `lmer`/`glmer` (lme4) or `brm` (brms).

Value

A maihda_model object containing:

`model`	The fitted model object (lme4 or brms)
`engine`	The engine used ("lme4" or "brms")
`formula`	The model formula

data	The data used for fitting
family	The family used
strata_info	The strata information from make_strata() if available, NULL otherwise

Examples

```
# Standard approach: manually create strata first
strata_result <- make_strata(maihda_sim_data, vars = c("gender", "race", "education"))
model <- fit_maihda(health_outcome ~ age + (1 | stratum),
                    data = strata_result$data,
                    engine = "lme4")

# Simplified approach: specify stratifying variables directly in the grouping structure
# The function internally calls make_strata() to create intersectionals
model2 <- fit_maihda(health_outcome ~ age + (1 | gender:race:education),
                     data = maihda_sim_data,
                     engine = "lme4")
```

---

maihda_health_data *NHANES Health Data Subset for MAIHDA Use*

Description

A pedagogical subset of the National Health and Nutrition Examination Survey (NHANES) dataset, serving as a real-world example for Multilevel Analysis of Individual Heterogeneity and Discriminatory Accuracy (MAIHDA). Contains selected records demonstrating intersectional demographic health inequalities.

Usage

```
maihda_health_data
```

Format

A data frame with 3,000 rows and 7 variables:

BMI Body Mass Index (kg/m^2), a continuous outcome variable.

Obese Factor indicating obesity status (No/Yes).

Age Age in years at screening, a continuous covariate.

Gender Gender of the participant (male/female).

Race Self-reported race/ethnicity.

Education Educational attainment level.

Poverty Poverty to income ratio, a continuous covariate. Some values may be missing.

Source

Derived from the NHANES R package. Original data collected by the Centers for Disease Control and Prevention (CDC).

Examples

```
data(maihda_health_data)

# Example usage:
# strata_result <- make_strata(maihda_health_data, vars = c("Gender", "Race", "Education"))
# model <- fit_maihda(BMI ~ Age + (1 | stratum), data = strata_result$data)
```

---

maihda_sim_data *Simulated Health Data for MAIHDA Use*

Description

A simulated dataset for demonstrating Multilevel Analysis of Individual Heterogeneity and Discriminatory Accuracy (MAIHDA).

Usage

```
maihda_sim_data
```

Format

A data frame with 500 rows and 7 variables:

id Unique participant identifier.

gender Gender of the participant.

race Simulated race/ethnicity category.

education Educational attainment level.

age Age in years, a continuous covariate.

health_outcome A continuous simulated health outcome.

binary_outcome A binary version of the health outcome.

Source

Simulated for the purpose of the MAIHDA package.

Examples

```
data(maihda_sim_data)
strata_result <- make_strata(maihda_sim_data, vars = c("gender", "race", "education"))
```

maihda_ternary_plot *Generate Ternary Plot from MAIHDA Model*

Description

Generate Ternary Plot from MAIHDA Model

Usage

```
maihda_ternary_plot(model, summary_obj = NULL, ...)
```

Arguments

model	A fitted MAIHDA model.
summary_obj	Optional output from summary_maihda.
...	Additional arguments passed to compute_maihda_ternary_data and plot_maihda_ternary.

Value

A list containing data and plot.

make_strata *Create Strata from Multiple Variables*

Description

This function creates strata (intersectional categories) from multiple categorical variables in a dataset.

Usage

```
make_strata(data, vars, sep = " × ", min_n = 1, autobin = TRUE)
```

Arguments

data	A data frame containing the variables to create strata from.
vars	Character vector of variable names to use for creating strata.
sep	Separator to use between variable values when creating stratum labels. Default is " \u00d7 " (a mathematical multiplication sign).
min_n	Minimum number of observations required for a stratum to be included. Strata with fewer observations will be coded as NA. Default is 1.
autobin	Logical indicating whether to automatically bin numeric grouping variables with more than 10 unique values into 3 categories (tertiles). Default is TRUE.

Details

If any of the specified variables has a missing value (NA) for a given observation, that observation will be assigned to the NA stratum (stratum = NA), rather than creating a stratum that includes the missing value.

The strata_info data frame is also attached as an attribute to the data, which allows fit_maihda() to automatically capture stratum labels for use in plots and summaries.

Value

A list with two elements:

`data`	The original data frame with an added 'stratum' column. The strata_info is also attached as an attribute for use by fit_maihda()
`strata_info`	A data frame with information about each stratum including counts and the combination of variable values

Examples

```
# Create strata from gender and race variables
result <- make_strata(maihda_sim_data, vars = c("gender", "race"))
print(result$strata_info)
```

---

plot.maihda_model *Plot MAIHDA Model Results*

Description

Creates various plots for visualizing MAIHDA model results including variance partition coefficient comparisons, observed vs. shrunken estimates, and predicted subgroup values with confidence intervals.

Usage

```
## S3 method for class 'maihda_model'
plot(
  x,
  type = c("all", "vpc", "obs_vs_shrunken", "predicted", "risk_vs_effect",
    "effect_decomp", "ternary", "prediction_deviation"),
  summary_obj = NULL,
  n_strata = 50,
  ...
)
```

Arguments

x	A maihda_model object from `fit_maihda()`.
type	Character string specifying plot type: • "vpc": Variance partition coefficient visualization • "obs_vs_shrunken": Observed vs. shrunken stratum means • "predicted": Predicted values for each stratum with confidence intervals • "risk_vs_effect": Quadrant scatterplot comparing overall risk to intersectional effect • "effect_decomp": Visualizes additive vs intersectional deviation from global mean • "ternary": Ternary plot analyzing the dimensional breakdown of variance • "prediction_deviation": Detailed deviation panels for individuals or strata • "all": Generate all available plots (default if not specified)
summary_obj	Optional maihda_summary object from `summary()`. If NULL, will be computed.
n_strata	Maximum number of strata to display in predicted plot. Default is 50. Use NULL for all strata.
...	Additional arguments (not currently used).

Value

A ggplot2 object, or a list of ggplot2 objects if type = "all".

Examples

```
strata_result <- make_strata(maihda_sim_data, vars = c("gender", "race"))
model <- fit_maihda(health_outcome ~ age + (1 | stratum), data = strata_result$data)

# VPC plot
plot(model, type = "vpc")

# Generate all plots
plots <- plot(model)
```

---

plot_comparison *Plot Model Comparison*

Description

Creates a plot comparing VPC/ICC across multiple models.

Usage

```
plot_comparison(comparison_df)
```

Arguments

comparison_df	A data frame from compare_maihda().

Value

A ggplot2 object.

Examples

```
# Create strata and models using simulated data
strata_1 <- make_strata(maihda_sim_data, vars = c("gender", "race"))
strata_2 <- make_strata(maihda_sim_data, vars = c("gender", "race", "education"))

model1 <- fit_maihda(health_outcome ~ age + (1 | stratum), data = strata_1$data)
model2 <- fit_maihda(health_outcome ~ age + gender + (1 | stratum), data = strata_2$data)

comparison <- compare_maihda(model1, model2, bootstrap = TRUE)
plot_comparison(comparison)
```

---

plot_maihda_ternary *Plot MAIHDA Ternary Diagram*

Description

Plot MAIHDA Ternary Diagram

Usage

```
plot_maihda_ternary(
  ternary_data,
  size_var = "n",
  color_var = "label",
  label_top_n = 5,
  label_by = c("interaction_signal", "uncertainty", "n"),
  alpha = 0.7
)
```

Arguments

ternary_data	Data output from compute_maihda_ternary_data.
size_var	Column name for point sizing.
color_var	Column name for point colors.
label_top_n	Number of top strata to label.
label_by	Variable used to determine top strata.
alpha	Point transparency.

Value

A plot object.

`plot_prediction_deviation_panels` *Plot Prediction Deviation Panels*

Description

Creates an advanced, publication-ready two-panel dashboard for visualizing predicted values and identifying deviant cases in linear, binomial, or ordinal models.

Usage

```
plot_prediction_deviation_panels(
  model,
  data = NULL,
  type = c("auto", "gaussian", "binomial", "ordinal"),
  ordinal_mode = c("surprise", "expected_score"),
  top_n_labels = 5,
  strata_info = NULL
)
```

Arguments

`model`	A fitted model object (e.g., from 'lm()', 'glm()', 'MASS::polr()', or 'lme4::glmer()').
`data`	The original data frame used to fit the model. If 'NULL', attempts to extract from the model.
`type`	Model type: "auto" (default), "gaussian", "binomial", or "ordinal".
`ordinal_mode`	For ordinal models: "surprise" (default, based on observation probability) or "expected_score".
`top_n_labels`	Number of extreme/deviant cases to label on the plot. Default is 5.
`strata_info`	Optional data frame of strata labels, generally extracted from 'maihda_model' objects.

Value

A 'patchwork' object containing two 'ggplot2' panels.

predict_maihda *Predict from MAIHDA Model*

Description

Makes predictions from a fitted MAIHDA model, either at the stratum level or individual level.

Usage

```
predict_maihda(
  object,
  newdata = NULL,
  type = c("individual", "strata", "response", "link"),
  scale = c("response", "link"),
  ...
)
```

Arguments

object	A maihda_model object from fit_maihda().
newdata	Optional data frame for making predictions. If NULL, uses the original data from model fitting.
type	Character string specifying prediction type: • "individual": Individual-level predictions including random effects • "strata": Stratum-level predictions (random effects only) For backward compatibility, "link" or "response" may also be passed here and will be interpreted as individual-level predictions on that scale.
scale	Character string specifying the prediction scale for individual-level predictions: "response" (default) or "link".
...	Additional arguments passed to predict method of underlying model.

Value

Depending on type:

- For "individual": A numeric vector of predicted values on the requested scale
- For "strata": A data frame with stratum ID and predicted random effect

Examples

```
strata_result <- make_strata(maihda_sim_data, vars = c("gender", "race"))
model <- fit_maihda(health_outcome ~ age + (1 | stratum), data = strata_result$data)

# Individual predictions
pred_ind <- predict_maihda(model, type = "individual")
```

```
# Stratum predictions
pred_strata <- predict_maihda(model, type = "strata")
```

`print.maihda_model` *Print method for maihda_model*

Description

Print method for maihda_model

Usage

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

Arguments

`x`	A maihda_model object
`...`	Additional arguments

Value

No return value, called for side effects.

`print.maihda_strata` *Print method for maihda_strata objects*

Description

Print method for maihda_strata objects

Usage

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

Arguments

`x`	A maihda_strata object
`...`	Additional arguments (not used)

Value

No return value, called for side effects.

print.maihda_summary *Print method for maihda_summary objects*

Description

Print method for maihda_summary objects

Usage

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

Arguments

x	A maihda_summary object
...	Additional arguments (not used)

Value

No return value, called for side effects.

print.pvc_result *Print method for PVC results*

Description

Print method for PVC results

Usage

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

Arguments

x	A pvc_result object
...	Additional arguments

Value

No return value, called for side effects.

run_maihda_app *Run MAIHDA Shiny Application*

Description

Launches a Shiny graphical user interface that exposes core functions of the MAIHDA package, allowing for visual data exploration, model fitting, and performance visualization.

Usage

```
run_maihda_app()
```

Value

No return value, called to launch the shiny app.

Examples

```
## Not run:
run_maihda_app()

## End(Not run)
```

stepwise_pcv *Stepwise Proportional Change in Variance (PCV)*

Description

Estimates the proportional change in variance (PCV) sequentially by fitting intermediate (partially-adjusted) models. It adds each predictor variable one-by-one to gauge its unique contribution in explaining between-stratum inequalities.

Usage

```
stepwise_pcv(data, outcome, vars, engine = "lme4", family = "gaussian")
```

Arguments

data	Data frame with observations. Ensure ‘make_strata()‘ was run first so the ‘stratum‘ variable exists.
outcome	Character string; the dependent variable.
vars	Character vector; predictors (strata groupings & covariates) to add sequentially to the model.
engine	Modeling engine ("lme4" or "brms"). Default is "lme4".
family	Error distribution and link function. Default is "gaussian".

Details

All models are fit on the complete cases for ‘outcome‘, ‘stratum‘, and all variables in ‘vars‘ so that each sequential variance comparison uses the same analytic sample.

Value

A data.frame showing the sequential models, the between-stratum variance at each step, and both the step-specific and total PCV.

Examples

```
strata_result <- make_strata(maihda_sim_data, c("gender", "race"))
stepwise_pcv(strata_result$data, "health_outcome", c("gender", "race", "age"))
```

---

summary.maihda_model *Summarize MAIHDA Model*

Description

Provides a summary of a MAIHDA model including variance partition coefficients (VPC/ICC) and stratum-specific estimates.

Usage

```
## S3 method for class 'maihda_model'
summary(object, bootstrap = FALSE, n_boot = 1000, conf_level = 0.95, ...)
```

Arguments

object	A maihda_model object from fit_maihda().
bootstrap	Logical indicating whether to compute bootstrap confidence intervals for VPC/ICC. Default is FALSE. Currently supported for lme4 models only.
n_boot	Number of bootstrap samples if bootstrap = TRUE. Default is 1000.
conf_level	Confidence level for bootstrap intervals. Default is 0.95.
...	Additional arguments (not currently used).

Value

A maihda_summary object containing:

vpc	Variance Partition Coefficient (ICC) with optional CI
variance_components	Data frame of variance components
stratum_estimates	Data frame of stratum-specific random effects with labels if available
fixed_effects	Fixed effects estimates
model_summary	Original model summary

Examples

```
strata_result <- make_strata(maihda_sim_data, vars = c("gender", "race"))
model <- fit_maihda(health_outcome ~ age + (1 | stratum), data = strata_result$data)
summary_result <- summary(model)

# With bootstrap CI
# summary_boot <- summary(model, bootstrap = TRUE, n_boot = 50)
```

Index