Package: MAIHDA (via r-universe)

June 7, 2026

Type Package

Title Multilevel Analysis of Individual Heterogeneity and Discriminatory Accuracy

Version 0.1.11

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.2017.12.026> 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 (>= 3.4.0), 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, 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

Config/pak/sysreqs cmake make libicu-dev

Repository https://hdbt.r-universe.dev

Date/Publication 2026-06-07 19:33:43 UTC

RemoteUrl https://github.com/hdbt/maihda

RemoteRef HEAD

RemoteSha 22ef3faedfb921bc409e0bb667d542a86ac7b87e

Contents

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

Description

Calculates the proportional change in between-stratum variance (PCV) between two MAIHDA models. The PCV measures how much the between-stratum variance changes when moving from one model to another, and is calculated as: PCV = (Var_model1 - Var_model2) / Var_model1. (The function and result object retain the historical "pvc" naming; “PVC” and “PCV” refer to the same quantity.)

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 the PCV. 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 the proportional change in between-stratum variance when moving from model1 to model2: a positive value means model2 has lower between-stratum variance, a negative value means higher. It is the share of model1's between-stratum variance *explained* by model2 only in the canonical nested case, where model2 adds fixed-effect predictors to model1 on the same outcome, analytic sample and strata. The function does not require nesting, so for non-nested models the PVC is simply a model-dependent difference in variance, not an explained proportion.

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; for **lme4** models, compute parametric-bootstrap VPC confidence intervals. Default FALSE. It does not apply to **brms** models, which always return a posterior credible interval (so passing `bootstrap = TRUE` with brms models errors) – their interval is included regardless.
`n_boot`	Number of bootstrap samples if bootstrap = TRUE. Default is 1000.
`conf_level`	Confidence level for the VPC interval (lme4 bootstrap CI or brms credible interval). Default is 0.95.

Details

VPCs are only directly comparable when the models share an outcome, family/link, analytic sample, and strata – the canonical use is nested models (e.g. null vs covariate-adjusted) on the *same* data and strata, to show how the VPC attenuates. If the supplied models differ in any of these, `compare_maihda()` still returns the table but issues a single warning, because the VPCs are then not directly comparable.

Value

A `maihda_comparison` data frame of VPC/ICC by model. Interval columns (`ci_lower`/`ci_upper`) are included when any model supplies an interval – an lme4 bootstrap CI or a brms posterior credible interval.

Examples

```
# Canonical use: nested models on the SAME data and strata (null vs adjusted)
strata <- make_strata(maihda_sim_data, vars = c("gender", "race"))

null_model <- fit_maihda(health_outcome ~ 1 + (1 | stratum), data = strata$data)
adj_model  <- fit_maihda(health_outcome ~ age + (1 | stratum), data = strata$data)

# Compare without bootstrap
comparison <- compare_maihda(null_model, adj_model,
                             model_names = c("Null", "Adjusted"))

# Compare with bootstrap CI
comparison_boot <- compare_maihda(null_model, adj_model,
                                  model_names = c("Null", "Adjusted"),
                                  bootstrap = TRUE, n_boot = 500)
```

`compare_maihda_groups` *Compare MAIHDA Metrics Across Levels of a Grouping Variable*

Description

Fits a separate random-intercept MAIHDA model (intercept-only *random* effects; any fixed-effect covariates in `formula` are still used) within each level of a higher-level grouping variable (for example country, region, or survey wave) and reports how the variance partition coefficient (VPC/ICC) and the between-/within-stratum variance components differ across those groups.

Usage

```
compare_maihda_groups(
  formula,
  data,
  group,
```

```
  engine = "lme4",
  family = "gaussian",
  shared_strata = TRUE,
  min_group_n = 30,
  bootstrap = FALSE,
  n_boot = 1000,
  conf_level = 0.95,
  autobin = TRUE,
  ...
)
```

Arguments

`formula`	A model formula. Either the shorthand intersectional form `outcome ~ covars + (1 | var1:var2)` (strata are built automatically) or `outcome ~ covars + (1 | stratum)` when `data` already contains a `stratum` column from `make_strata`.
`data`	A data frame containing the variables in `formula` and the grouping variable.
`group`	Character string naming the grouping variable in `data` (e.g. `"country"`). A separate model is fitted for each non-missing level.
`engine`	Modeling engine, "lme4" (default) or "brms".
`family`	Model family. Default "gaussian". As in `fit_maihda`, a binary outcome is auto-detected once on the full data and switched to "binomial" (with a warning) so every group uses the same family.
`shared_strata`	Logical. When TRUE (default) intersectional strata are defined once on the full data so that a stratum denotes the same combination in every group; this makes the stratum *definitions* comparable across groups. Note that a group may still not contain every stratum, so two groups' VPCs can be estimated over different sets of populated strata – they are then not strictly directly comparable, and the function warns when this happens. When FALSE, strata are rebuilt independently within each group (stratum identities are then not comparable across groups at all).
`min_group_n`	Minimum size of the *analytic* sample a group must have – the rows that survive the model frame (covariate transformations applied, rows with a missing outcome/covariate dropped) – to be modelled. Groups with a smaller usable sample are skipped with a warning, even if they have more raw rows. Default 30.
`bootstrap`	Logical; compute per-group parametric-bootstrap VPC confidence intervals. lme4 engine only. Default FALSE.
`n_boot`	Number of bootstrap samples when `bootstrap = TRUE`. Default 1000.
`conf_level`	Confidence level for bootstrap intervals. Default 0.95.
`autobin`	Logical passed to `make_strata` controlling tertile binning of numeric grouping variables. Default TRUE.
`...`	Additional arguments passed to `fit_maihda` (and on to `lmer`/`glmer`).

Details

It estimates one VPC per group as a stratified analysis: each group is modelled independently. It is *not* a cross-classified model and does not adjust the strata for the grouping variable.

The VPC is the *share* of the unexplained variance that lies between strata, not the absolute magnitude of intersectional inequality. Because it is a ratio, a group's VPC can differ from another's because the between-stratum variance differs, because the within-stratum (residual) variance differs, or both – two groups with the same between-stratum variance can have very different VPCs. To compare the absolute amount of between-stratum (intersectional) variation across groups, read the returned `var_between` column alongside the VPC rather than treating a higher VPC as "more inequality".

It is **descriptive**: it reports each group's VPC (with an interval when available – an lme4 bootstrap CI or a brms credible interval) for side-by-side comparison, but does not test whether the VPCs differ between groups. The per-group intervals describe each group's own uncertainty; whether two intervals overlap is *not* a valid test of the difference between their VPCs, which would require modelling that difference directly.

Robustness: a group whose *analytic* sample (rows surviving the model frame) has fewer than `min_group_n` observations is always skipped with a warning. A group with fewer than two populated strata is also skipped (VPC is undefined with a single stratum) when the stratum membership is known before fitting – that is, when `shared_strata = TRUE` or `data` already carries a `stratum` column. Under `shared_strata = FALSE` strata are rebuilt inside each group, so a degenerate single-stratum group is instead reported with a "fit failed" status rather than a pre-fit skip. A singular fit yields a VPC of 0 rather than an error (unlike `calculate_pvc`). A hard fit failure in one group records `NA` and a status note without aborting the whole comparison.

Fit-quality diagnostics: for the `lme4` engine, groups whose model is singular or fails to converge keep a `status` of `"ok"` (the fit did complete) but are named in a single aggregated warning, because their VPC/ICC may be unreliable – a singular fit usually pins the between-stratum variance at the boundary, giving a VPC of 0.

Value

A `data.frame` of class `maihda_group_comparison` with one row per group and columns `group`, `n`, `n_strata`, `vpc`, `var_between`, `var_other`, `var_residual`, `status` (and `ci_lower`/`ci_upper` when `bootstrap = TRUE`). `n` is the analytic sample size used by the model (after dropping rows with a missing outcome/covariate) for both fitted and skipped groups, falling back to the raw row count only when the model frame cannot be built. `var_other` is the variance of any additional random effects and is 0 for the canonical single-stratum model. Groups that were skipped or failed have `NA` metrics and an explanatory `status`.

See Also

`compare_maihda` for comparing different models on the same data; `plot.maihda_group_comparison` for visualising the result.

Examples

```
data(maihda_country_data)
# How does gender x SES inequality in PISA math scores differ across countries?
cmp <- compare_maihda_groups(
```

```
  math ~ 1 + (1 | gender:ses),
  data = maihda_country_data,
  group = "country"
)
print(cmp)
plot(cmp, type = "vpc")
```

`compute_maihda_ternary_data`

Compute Ternary Data for MAIHDA Models

Description

Compute Ternary Data for MAIHDA Models

Usage

```
compute_maihda_ternary_data(
  model,
  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()‘.
`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, family object, or family function 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. When a two-level non-0/1 response is recoded (on either the auto-detected or an explicit `family = "binomial"` path), the mapping follows the usual convention – the first level becomes 0 (reference) and the second becomes 1 (the modeled event), where "first/second" means alphabetical order for a character outcome and the declared order for a factor. The chosen mapping is reported via a `message()` and stored on the result as `$response_recoding`; set the factor levels (or supply a 0/1 outcome) to control which level is the event. Although any valid family object is accepted for fitting, the MAIHDA variance summaries (`summary.maihda_model`, VPC/ICC, PCV) are only defined for `gaussian("identity")`, the binomial/Bernoulli families with a logit or probit link, and `poisson("log")`. Other families (for example `Gamma(link = "log")`) will fit, but `summary()` and the VPC/PCV helpers will stop with an "not implemented" error because no level-1 variance is defined for them.

`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
`response_recoding`	For a recoded two-level outcome, a data frame mapping each original level to its 0/1 value and role (reference/event); NULL when no recoding occurred
`diagnostics`	Fit-quality diagnostics (singular fit / convergence) for lme4 models, surfaced by the print and summary methods

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` *Run a Complete MAIHDA Analysis*

Description

A single high-level entry point that runs the standard MAIHDA workflow and returns one bundled object: it fits the multilevel model, summarises the variance partition (VPC/ICC) and components, and – when a higher-level grouping variable is supplied – also compares intersectional inequality across that variable’s levels.

Usage

```
maihda(
  formula,
  data,
  group = NULL,
  engine = "lme4",
  family = "gaussian",
  autobin = TRUE,
  shared_strata = TRUE,
  min_group_n = 30,
  bootstrap = FALSE,
  n_boot = 1000,
  conf_level = 0.95,
  ...
)
```

Arguments

`formula`	A model formula, using either the intersectional shorthand `outcome ~ covars + (1 | var1:var2)` or `... + (1 | stratum)` when `data` already has a `stratum` column from `make_strata`.
`data`	A data frame with the model variables (and the `group` variable, if used).
`group`	Optional character string naming a higher-level grouping variable (e.g. `"country"`). When supplied, `compare_maihda_groups` is run and attached to the result.
`engine`	Modeling engine, "lme4" (default) or "brms".
`family`	Model family. Default "gaussian". As in `fit_maihda`, a binary outcome is auto-detected when `family` is left at the default, and the same resolved family is then used for the group comparison so all models agree.
`autobin`	Logical passed to `make_strata`; tertile-bins numeric grouping variables. Default TRUE.
`shared_strata`	Logical, forwarded to `compare_maihda_groups` when `group` is supplied: build strata once on the full data so VPCs are comparable across groups (TRUE, default) or rebuild them within each group.
`min_group_n`	Minimum group size for the per-group comparison, forwarded to `compare_maihda_groups`. Default 30.
`bootstrap`	Logical; compute parametric-bootstrap VPC confidence intervals (lme4 only) for both the overall summary and the per-group comparison. Default FALSE.
`n_boot`	Number of bootstrap samples when `bootstrap = TRUE`.
`conf_level`	Confidence level for bootstrap intervals. Default 0.95.
`...`	Additional arguments passed to `fit_maihda` (and on to `lmer`/`glmer`).

Details

This is a convenience wrapper around `fit_maihda`, `summary.maihda_model` and `compare_maihda_groups`; it always returns the same `maihda_analysis` structure (the `groups` slot is simply `NULL` when `group` is not given), so downstream code never has to branch on the return type.

Value

An object of class `maihda_analysis`: a list with

`model`	the fitted `maihda_model` (see `fit_maihda`)
`summary`	the `maihda_summary` (VPC/ICC, variance components, stratum estimates)
`groups`	a `maihda_group_comparison` when `group` is supplied, otherwise `NULL`
`formula`, `group_var`, `call`	bookkeeping for printing

See Also

`fit_maihda` for the single-model fitter, `compare_maihda_groups` for the group comparison, and `summary.maihda_model` for the variance summary.

Examples

```
data(maihda_health_data)

# One call: fit + VPC summary
a <- maihda(BMI ~ Age + (1 | Gender:Race), data = maihda_health_data)
a
plot(a, type = "vpc")

# Add a higher-level grouping variable to also compare across its levels.
# maihda_country_data has a real country grouping (PISA achievement data):
data(maihda_country_data)
a2 <- maihda(math ~ 1 + (1 | gender:ses), data = maihda_country_data,
             group = "country")
a2
plot(a2, type = "group_vpc")
```

`maihda_country_data` *Cross-National Educational Achievement Data for MAIHDA*

Description

A cross-national dataset for demonstrating how Multilevel Analysis of Individual Heterogeneity and Discriminatory Accuracy (MAIHDA) can be used to compare intersectional inequality *across* a higher-level grouping variable (here, country) with `compare_maihda_groups` and `maihda`. Each row is a 15-year-old student; the intersectional strata are formed by `gender` and socioeconomic status (`ses`), and the outcome is the PISA mathematics score.

Usage

```
maihda_country_data
```

Format

A data frame with 3,600 rows (600 students in each of 6 countries) and 7 variables:

country Factor; one of Finland, Germany, United Kingdom, Italy, Japan, Mexico. The higher-level grouping variable.

gender Factor; student gender (female/male). A stratum dimension.

ses Factor; socioeconomic status as global tertiles (Low/Medium/High) of `escs`, computed on the pooled sample so a band means the same in every country. A stratum dimension.

escs Numeric; the PISA index of economic, social and cultural status (the continuous measure underlying `ses`).

math Numeric; PISA mathematics score (first plausible value). The primary outcome.

reading Numeric; PISA reading score (first plausible value).

low_math Factor; "Yes" if `math` is below 420 (PISA proficiency Level 2 baseline), else "No". A binary outcome for logistic examples.

Details

Intersectional inequality (the between-stratum share of variance, VPC/ICC) in mathematics achievement differs across the six countries, which is what makes the dataset a useful showcase for the group-comparison workflow.

The intersectional strata are `gender:ses` (2 x 3 = 6 strata). A canonical MAIHDA "null" model is `math ~ 1 + (1 | gender:ses)`; comparing its VPC across countries quantifies how much joint gender-by-class inequality in achievement varies between countries.

Note

This is a teaching/illustration dataset only. It uses a single PISA plausible value for each score and the analysis functions ignore the PISA survey weights and complex sampling design, so results are **not** survey-representative and should not be used for substantive cross-national inference.

Source

Derived from the OECD Programme for International Student Assessment (PISA) 2018 student questionnaire data (OECD (2019), *PISA 2018 Database*), accessed and cleaned via the **learningtower** R package (MIT licensed), `https://CRAN.R-project.org/package=learningtower`. A balanced random subsample of 600 complete-case students per country was taken (seed 2026). The data preparation script is in `data-raw/maihda_country_data.R`.

Examples

```
data(maihda_country_data)

# Compare intersectional (gender x SES) inequality across countries
analysis <- maihda(
  math ~ 1 + (1 | gender:ses),
  data = maihda_country_data,
  group = "country"
)
```

```
analysis
plot(analysis, type = "group_vpc")
```

`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.

Note

This is a teaching/illustration dataset only. It is a non-random subsample and the analysis functions ignore the NHANES survey weights and complex sampling design, so results are **not** survey-representative and should not be used for substantive population inference.

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, ...)
```

Arguments

`model`	A fitted MAIHDA model.
`...`	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. When this happens a `message()` is emitted, because the resulting strata are data-dependent (tertile cut-points depend on the sample) and a continuous variable placed in the grouping term is usually unintended. Set `autobin` = `FALSE` to disable, or bin the variable yourself for explicit, reproducible cut-points.

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_analysis` *Plot a MAIHDA Analysis*

Description

Dispatches to the model's plots (see `plot.maihda_model`) for the model-level types, and to the group comparison for `"group_vpc"` and `"group_components"` when `maihda` was called with a `group`.

Usage

```
## S3 method for class 'maihda_analysis'
plot(x, type = "all", ...)
```

Arguments

`x`	A `maihda_analysis` object from `maihda`.
`type`	One of the `plot.maihda_model` types ("all", "vpc", "obs_vs_shrunken", "predicted", "risk_vs_effect", "effect_decomp", "ternary", "prediction_deviation") or a group type ("group_vpc", "group_components"). Default "all".
`...`	Additional arguments passed to the underlying plot method.

Value

A ggplot2 object, or (for `type = "all"`) an invisible list of them.

`plot.maihda_comparison`

Plot a MAIHDA Model Comparison

Description

Plots VPC/ICC across the models compared by `compare_maihda`. Dispatched via `plot()` on the classed result.

Usage

```
## S3 method for class 'maihda_comparison'
plot(x, ...)
```

Arguments

`x`	A `maihda_comparison` object from `compare_maihda`.
`...`	Additional arguments (not used).

Value

A ggplot2 object.

Examples

```
strata <- make_strata(maihda_sim_data, vars = c("gender", "race"))

null_model <- fit_maihda(health_outcome ~ 1 + (1 | stratum), data = strata$data)
adj_model  <- fit_maihda(health_outcome ~ age + (1 | stratum), data = strata$data)

comparison <- compare_maihda(null_model, adj_model, bootstrap = TRUE)
plot(comparison)
```

`plot.maihda_group_comparison`

Plot a MAIHDA Group Comparison

Description

Visualises the output of `compare_maihda_groups` either as a point/forest plot of the VPC/ICC by group, or as stacked variance-composition bars (between- vs within-stratum share) by group. Dispatched via `plot()` on the classed result.

Usage

```
## S3 method for class 'maihda_group_comparison'
plot(x, type = c("vpc", "components"), ...)
```

Arguments

`x`	A `maihda_group_comparison` object from `compare_maihda_groups`.
`type`	Either "vpc" (default) for VPC by group with optional bootstrap confidence intervals, or "components" for stacked variance proportions.
`...`	Additional arguments (not used).

Value

A ggplot2 object.

Examples

```
data(maihda_health_data)
cmp <- compare_maihda_groups(BMI ~ Age + (1 | Gender:Race),
                             data = maihda_health_data, group = "Education")
plot(cmp, type = "vpc")
plot(cmp, type = "components")
```

`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. The y-axis (model-based estimate) includes the fixed effects, so for a covariate-adjusted model the distance from the diagonal reflects both shrinkage *and* covariate adjustment, not shrinkage alone; it is a pure shrinkage view only for an intercept-only (null) model • "predicted": Predicted values for each stratum with confidence intervals • "risk_vs_effect": Quadrant scatterplot of each stratum's mean predicted outcome against its random effect • "effect_decomp": Visualizes additive vs intersectional deviation from global mean • "ternary": Ternary diagnostic of the relative additive, intersectional, and uncertainty signals per stratum (a normalized-magnitude diagnostic, not a variance decomposition) • "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.maihda_ternary` *Plot MAIHDA Ternary Diagram*

Description

Renders the ternary decomposition produced by `compute_maihda_ternary_data`. Dispatched via `plot()` on the classed result.

Usage

```
## S3 method for class 'maihda_ternary'
plot(
  x,
  size_var = "n",
  color_var = "label",
  label_top_n = 5,
  label_by = c("interaction_signal", "uncertainty", "n"),
  alpha = 0.7,
  ...
)
```

Arguments

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

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 highlighting the most notable cases or strata. What "notable" means depends on the model type, and the labelled points are *not* statistical outliers in the regression-diagnostic sense:

- Gaussian and Poisson (and the ordinal `"expected_score"` mode): the cases/strata whose prediction sits furthest from the mean prediction (largest deviation), ranked by absolute deviation.
- Binomial: the cases/strata with the largest absolute deviance residual, i.e. where the observed 0/1 outcome is least consistent with the fitted probability (worst-fit points), ranked by $|devianceresidual|$.
- Ordinal `"surprise"` mode: the cases/strata with the highest surprise $-\log P(\text{observed category})$, i.e. the least probable observations under the model.

Usage

```
plot_prediction_deviation_panels(
  model,
  data = NULL,
  type = c("auto", "gaussian", "poisson", "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", "poisson", "binomial", or "ordinal".
`ordinal_mode`	For ordinal models: "surprise" (default, based on observation probability) or "expected_score".
`top_n_labels`	Number of points to label on the plot. The ranking metric depends on the model type (see Description): deviation from the mean prediction for Gaussian/Poisson and the ordinal expected-score mode, absolute deviance residual for binomial, and surprise for the ordinal surprise mode. 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. When `newdata` is supplied, the result is restricted to the strata present in `newdata` (and a stratum the model never saw is an error, as for "individual"); when `newdata` is `NULL`, every training stratum is returned.

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_analysis` *Print a MAIHDA Analysis*

Description

Print a MAIHDA Analysis

Usage

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

Arguments

`x`	A `maihda_analysis` object from `maihda`.
`...`	Additional arguments (not used).

Value

No return value, called for side effects.

`print.maihda_group_comparison` *Print method for MAIHDA group comparisons*

Description

Print method for MAIHDA group comparisons

Usage

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

Arguments

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

Value

No return value, called for side effects.

`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, adding each predictor one-by-one. The step-specific PCV is the change in between-stratum variance contributed by a predictor *given the variables already in the model*. Because the steps are sequential it is order-dependent: it reflects each variable's marginal, model-dependent change, not an order-invariant "unique" contribution.

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_analysis`

Summarize a MAIHDA Analysis

Description

Returns the variance summary (VPC/ICC, variance components, stratum estimates) of the fitted model. The per-group comparison, when present, is attached as the `"groups"` attribute.

Usage

```
## S3 method for class 'maihda_analysis'
summary(object, ...)
```

Arguments

`object`	A `maihda_analysis` object from `maihda`.
`...`	Additional arguments (not used).

Value

The `maihda_summary` for the fitted model.

`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 parametric bootstrap confidence intervals for VPC/ICC. Default is FALSE. Supported for lme4 models only; `brms` models always return a posterior credible interval (see Details), so `bootstrap = TRUE` is rejected for them.
`n_boot`	Number of bootstrap samples if bootstrap = TRUE. Default is 1000.
`conf_level`	Confidence level for the VPC/ICC interval – the lme4 bootstrap CI or the brms posterior credible interval. Default is 0.95.
`...`	Additional arguments (not currently used).

Value

A maihda_summary object containing:

`vpc`	Variance Partition Coefficient (ICC); for lme4 with `bootstrap = TRUE` and for all brms models this includes `ci_lower/ci_upper/conf_level`
`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
`diagnostics`	Fit-quality diagnostics (singular fit / convergence) carried over from the fitted model and reported by the print method

Interpreting the VPC/ICC

The VPC is the between-stratum variance divided by the total *unexplained* variance. For the canonical single-stratum model that denominator is between-stratum + residual, but if the model includes additional random effects (e.g. `(1 | site)`) their variance is included in the denominator too (between-stratum + other random effects + residual), so the VPC is the between-stratum *share* of all unexplained variance. It is a conditional/residual ICC that excludes variance captured by the fixed effects, so for models with covariates it is conditional on them. It is most commonly read from the null model `outcome ~ 1 + (1 | stratum)`, where it is the total between-stratum share. For non-Gaussian families the level-1 (residual) variance uses a latent/distributional approximation (e.g. $\pi^2/3$ for logistic), so the VPC is on that latent scale; for a *weighted* Gaussian model the level-1 variance is the mean conditional residual variance, $\overline{\sigma^2/w_i}$, since the per-observation residual variance is σ^2/w_i. The stratum random effects represent the total between-stratum deviation; they equal the *pure* intersectional (interaction) component only when the additive main effects of the strata variables are included in the model.

Note

For `lme4` models a VPC/ICC interval is obtained from a parametric bootstrap (`bootstrap = TRUE`). For `brms` models the VPC/ICC is summarised directly from the posterior draws: the reported estimate is the posterior median of the per-draw VPC ($E[\sigma^2]$-based, not the biased $E[\sigma]^2$) and the interval is a central credible interval at `conf_level` (default 95%), so no `bootstrap` argument is needed. The variance-components table reports the posterior-mean variance components, so the stratum proportion shown there may differ slightly from the headline VPC because the median of a ratio is not the ratio of means. For non-Gaussian `brms` families the level-1 (residual) variance uses the usual latent-scale approximation; for `poisson(log)` it is evaluated at the posterior-mean fitted values rather than per draw to avoid an expensive $ndraws \times nobs$ computation.

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