Package ‘MAIHDA’

July 2, 2026

Type Package

Title Multilevel Analysis of Individual Heterogeneity and Discriminatory Accuracy

Version 0.2.0

Description Tools for Multilevel Analysis of Individual Heterogeneity and Discriminatory Accuracy (MAIHDA) for intersectional inequality research. 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>. The package creates intersectional strata, fits multilevel MAIHDA models, estimates variance partition coefficients, proportional change in variance, stratum effects, and discriminatory-accuracy summaries, and provides diagnostic and presentation plots.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 4.1.0)

Imports lme4 (>= 1.1-27), reformulas, cli, ggplot2 (>= 3.4.0), dplyr (>= 1.0.0), tidyr (>= 1.1.0), generics, stats, rlang (>= 0.4.0), patchwork, ggrepel, tidyselect, tibble

Suggests shiny, bslib (>= 0.7.0), DT, future, promises, plotly, haven, shinyjs, shinycssloaders, brms (>= 2.15.0), WeMix (>= 4.0.0), ordinal, testthat (>= 3.0.0), knitr, MASS, rmarkdown

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 2026-07-02 08:00:02 UTC

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.

REML vs ML. `lmer` fits Gaussian models by REML, whose between-stratum variance estimate is *not* comparable across models with different fixed effects – exactly the canonical null-vs-adjusted PCV, where the adjusted model adds the dimensions’ main effects. `calculate_pvc()` therefore refits any REML `lmer` model with maximum likelihood (`refitML`) before reading the variances (and before the parametric bootstrap, so the interval matches), matching `maihda_ic` and `anova()`

on `lme4` models. Using REML estimates here biases the PCV (it overstates the residual between-stratum variance of the adjusted model). GLMM fits (`glmer`) and the brms/wemix/ordinal engines are already on the maximum-likelihood scale and are unaffected; single-model VPC/ICC summaries keep their REML fit, since that comparison-free quantity is not subject to the pitfall.

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. For negative-binomial models (`glmer.nb`) `refit()` holds the dispersion parameter theta fixed at its original estimate, so the interval is conditional on the estimated theta.

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,
  ic = TRUE
)
```

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.
`ic`	Logical; append relative-fit information criteria to the table for comparing model *structures*: `AIC`/`BIC` for the likelihood engines (lme4, ordinal) and `WAIC`/`LOOIC` for brms (see `maihda_ic`). Default TRUE. REML `lmer` fits are refitted with ML so AIC/BIC are comparable across different fixed effects. Set FALSE for the lean VPC-only table.

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. The same comparability caveat applies to the appended information criteria (see `maihda_ic`). In addition, when the appended criteria mix scales – likelihood `AIC`/`BIC` (lme4/ordinal) shown alongside Bayesian `WAIC`/`LOOIC` (brms), which can happen for a same-family lme4-vs-brms comparison that the family/link check does not flag – `compare_maihda()` warns, because those criteria are on different scales and are not comparable to each other.

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. When `ic = TRUE`, information-criteria columns (`AIC`/`BIC` or `WAIC`/`LOOIC`, whichever apply) are appended.

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. When the strata are defined by at least two dimensions it also fits the adjusted model (the dimensions' additive main effects) within each group and reports the per-group `pcv` – the proportional change in between-stratum variance, i.e. the additive share of that group's intersectional inequality.

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,
  decomposition = c("two-model", "crossed-dimensions"),
  sampling_weights = NULL,
  ...
)
```

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), "brms", "wemix" (the design-weighted fit; requires `sampling_weights` and is selected automatically when they are supplied with the default engine), or "ordinal" (cumulative link mixed model via `ordinal::clmm()`; selected automatically for an ordinal family or an ordered-factor outcome).
`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.
`decomposition`	Per-group additive-vs-interaction decomposition: the two-model null -> adjusted PCV (`"two-model"`, default) or the single crossed-dimensions model (`"crossed-dimensions"`; `"cross-classified"` is a deprecated alias that warns). The crossed-dimensions form requires `shared_strata = TRUE` and at least two stratum dimensions, and adds the `var_additive`, `var_interaction`, `additive_share` and `interaction_share` columns (in place of `pcv` / `var_between_adjusted`); `var_between` is then the total between-strata variance (additive + interaction). See `maihda` for the underlying model and its caveats.
`sampling_weights`	Optional name of a sampling-weight column in `data` for design-weighted per-group fits; see `fit_maihda`. The column is sliced with each group's rows, so every group is fitted with its own members' weights. Not compatible with `engine`

= `"lme4"`, `bootstrap = TRUE`, or (under the wemix engine) `decomposition = "crossed-dimensions"`.

`...`	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`). When the strata are defined by at least two dimensions, two further columns report the per-group null -> adjusted decomposition: `pcv` (proportional change in between-stratum variance when the dimensions' additive main effects are added; computed on the maximum-likelihood scale – see `calculate_pvc` – because REML variances are not comparable across the null vs. adjusted fixed effects), `var_between_adjusted` (a *derived* coherence quantity, reported as `var_between * (1 - pcv)` so it shares the scale of the REML `var_between`/`vpc` and the table satisfies `pcv = (var_between - var_between_adjusted) / var_between` exactly – it is **not** the adjusted fit's own variance), and `var_between_adjusted_ml` (the adjusted model's *actual* between-stratum variance, read straight off the adjusted fit on the same maximum-likelihood scale as the PCV; it differs from `var_between_adjusted` only by the small REML-vs-ML gap in the null variance). All three are `NA` for a group whose adjusted fit failed, and the columns are omitted entirely when the strata have a single dimension. `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")
```

`fit_maihda` *Fit MAIHDA Model*

Description

Fits a multilevel model for MAIHDA (Multilevel Analysis of Individual Heterogeneity and Discriminatory Accuracy) using lme4, brms, WeMix (for design-weighted (survey) data), or – for an ordered-factor outcome – a cumulative link mixed model via `ordinal::clmm()`.

Usage

```
fit_maihda(
  formula,
  data,
  engine = "lme4",
  family = "gaussian",
  autobin = TRUE,
  context = NULL,
  sampling_weights = NULL,
  id = NULL,
  time = NULL,
  time_degree = 1,
  interactions = FALSE,
  ...
)
```

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), "brms", "wemix" (design-weighted pseudo-maximum-likelihood via `WeMix::mix()`; requires `sampling_weights`), or "ordinal" (cumulative link mixed model via `ordinal::clmm()`; requires an ordinal family). When `sampling_weights` is supplied and `engine` is left at its default, the engine switches to "wemix" automatically (with a message); likewise an ordinal family (or an auto-detected ordered-factor outcome) switches the default engine to "ordinal".
`family`	Character string, family object, or family function specifying the model family. Common options: "gaussian", "binomial", "poisson", "negbinomial". Default is "gaussian". `family = "negbinomial"` fits an overdispersed count model with the dispersion parameter theta *estimated* from the data: lme4 via `lme4::glmer.nb()` and brms via its `shape` parameter (log link only; not supported by the wemix engine). A fixed-theta `MASS::negative.binomial(theta)` family object is also accepted with `engine = "lme4"` and is fitted with `glmer()`, honouring the supplied theta. `family = "ordinal"` (alias `"cumulative"`; or `maihda_cumulative("probit")` / `brms::cumulative()` for a non-logit link) fits a cumulative (proportional-odds) model for an *ordered-factor* outcome: `ordinal::clmm()` under the automatic "ordinal" engine, `brms::cumulative()` under `engine = "brms"`. The VPC/ICC lives on the latent scale (level-1 variance $\pi^2/3$ logit / 1 probit, as for binomial models) and response-scale predictions are *expected category scores* (categories scored 1..K in order). An ordered-factor outcome with 3+ levels under the default family selects this model automatically, with a warning. The logit and probit links are supported; `sampling_weights`, `context`, and lme4-style `weights`/`subset`/`offset` arguments are not available on the clmm path. 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, `poisson("log")`, and the negative binomial with a log link (level-1 variance `log(1 + 1/mu + 1/theta)`; Nakagawa, Johnson & Schielzeth 2017). 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.
`context`	Optional character vector naming one or more higher-level *context* columns in `data` (e.g. `"school"`, `"hospital"`, `"region"`). Each enters the model as a crossed intercept-only random effect alongside the intersectional stratum effect – `outcome ~ covars + (1 | stratum) + (1 | context)` – giving the *contextual cross-classified MAIHDA* of the literature (individuals cross-classified by stratum and place/institution). `summary.maihda_model` then partitions the unexplained variance into between-stratum vs. between-context vs. residual, and the headline VPC/ICC remains the between-stratum share (now net of the context). A context variable may not be a stratum dimension or `"stratum"` itself, and may not already appear as a fixed-effect term (its variance would then be absorbed by the fixed part). A context with few levels (say < 10) weakly identifies its variance and often yields a singular lme4 fit; the `brms` engine handles this better. Writing the random effect directly in the formula (`... + (1 | school)`) fits the same model but is summarised generically as "Other random effects"; only `context =` activates the labelled contextual partition. Not supported by the `wemix` engine.
`sampling_weights`	Optional single character string naming a numeric column of `data` holding individual *sampling* (survey/design) weights, for a **design-weighted MAIHDA** on complex-survey data (e.g. NHANES, PISA). Sampling weights are not the same thing as lme4's `weights=` (precision weights, which rescale the residual variance), so combining `sampling_weights` with `engine = "lme4"` is an error. Two engines support them: • `engine = "wemix"` (chosen automatically when `engine` is left at its default): weighted pseudo-maximum-likelihood via `WeMix::mix()` (Rabe-Hesketh & Skrondal 2006), the estimator used for NAEP/PISA analysis. The individual weights enter at level 1 unchanged and the level-2 (stratum) weights are 1, because intersectional strata are exhaustive population cells included with certainty. Supports `gaussian(identity)` and `binomial(logit)` models with the canonical single `(1 | stratum)` random intercept. Fixed-effect standard errors are design-consistent (sandwich); the VPC/PCV are reported as point estimates (no bootstrap – see `summary.maihda_model`). • `engine = "brms"`: the weights enter the model as likelihood weights (`y | weights(w)`), normalized to mean 1, giving a *pseudo-posterior*: point estimates target the population-weighted estimand but credible intervals are not design-based – interpret them cautiously. Full design-based (replicate-weight / linearised) variances for the variance components are not computed. Rows with a missing or non-positive sampling weight are dropped with a warning. The column names `.maihda_sw` (brms likelihood weight) and `.maihda_l2wt` (WeMix level-2 weight) are **reserved** for the design-weighted engines: they are written into the analytic data internally, so a model that supplies `sampling_weights` may not also reference a variable of either name in its formula (fitting would error). Default `NULL` (unweighted).

`id`	Optional single character string naming a person/unit identifier column for a **longitudinal (growth-curve) MAIHDA** on long-format data (one row per measurement occasion). Supplied together with `time`, it makes the model a 3-level growth curve – occasions within individuals (`id`) within intersectional strata – with a random intercept and slope on `time` at *both* the individual and stratum levels. The growth random effects are added automatically: write the strata shorthand `(1 | var1:var2)` (or `(1 | stratum)`) only, not the slopes. The between-stratum variance (and hence the VPC) then becomes a function of time; `summary.maihda_model` reports the time-varying VPC. Longitudinal fits are supported by `engine = "lme4"/"brms"` only (not `wemix/ordinal`), and are incompatible with `context` and `sampling_weights`. Default `NULL` (cross-sectional). See Bell, Evans, Holman & Leckie (2024).
`time`	Optional single character string naming a numeric measurement-time column (e.g. wave 0, 1, 2, ... or age), required for a longitudinal MAIHDA; see `id`. Default `NULL`.
`time_degree`	Polynomial degree of the growth curve when `time` is supplied: 1 (default) linear, 2 quadratic, etc. The brms engine supports degree 1 only.
`interactions`	Opt-in per-stratum interaction diagnostic (`maihda_interactions`), attached as the `interactions` slot and shown by `print()`. `FALSE` (default) skips it; `TRUE` computes it with the diagnostic's default correction (`adjust = "BH"`); or pass a `p.adjust` method name, including `"none"` for the uncorrected view. It is meaningful only on an *adjusted* model (the dimensions' main effects in the fixed part); on a null model `maihda_interactions` warns. This is the single-fit parallel to the default-on `interactions` of `maihda`.
`...`	Additional arguments passed to `lmer`/`glmer` (lme4), `brm` (brms), or `WeMix::mix()` (wemix; e.g. `nQuad`, `fast`). The lme4-style `weights`/`subset`/`offset` arguments are not supported by the wemix engine.

Value

A maihda_model object containing:

`model`	The fitted model object (lme4, brms, WeMix, or ordinal::clmm)
`engine`	The engine used ("lme4", "brms", "wemix", or "ordinal")
`sampling_weights`	The sampling-weight column name when supplied, NULL otherwise
`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
`context_vars`	The context variable name(s) when `context` was supplied, NULL otherwise
`interactions`	The `maihda_interactions` diagnostic when `interactions` is not `FALSE`, 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")

# Contextual cross-classified MAIHDA: strata crossed with a higher-level
# context (here country) -- the literature's cross-classified MAIHDA.
data(maihda_country_data)
model3 <- fit_maihda(math ~ 1 + (1 | gender:ses),
                     data = maihda_country_data,
                     context = "country")
summary(model3)  # between-stratum vs. between-country vs. residual
```

`maihda` *Run a Complete MAIHDA Analysis*

Description

A single high-level entry point that runs the standard two-model MAIHDA workflow and returns one bundled object. It fits the **null** model (covariates plus the intersectional random intercept, *excluding* the stratum dimensions' main effects) and the **adjusted** model (the null plus the additive main effects of the stratum-defining dimensions), summarises the variance partition (VPC/ICC) of the null model, and reports the **PCV** – the proportional change in between-stratum variance from the null to the adjusted model, i.e. the additive share of the intersectional inequality. When a higher-level grouping variable is supplied it also compares this decomposition across that variable's levels.

Usage

```
maihda(
  formula,
  data,
  group = NULL,
  context = NULL,
  engine = "lme4",
```

```
  family = "gaussian",
  decomposition = c("two-model", "crossed-dimensions", "longitudinal"),
  autobin = TRUE,
  shared_strata = TRUE,
  min_group_n = 30,
  bootstrap = FALSE,
  n_boot = 1000,
  conf_level = 0.95,
  response_vpc = FALSE,
  seed = NULL,
  sampling_weights = NULL,
  id = NULL,
  time = NULL,
  time_degree = 1,
  interactions = TRUE,
  ...
)
```

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`. The dimensions' additive main effects may be listed (the fully-specified adjusted model) or omitted (added automatically, with a message); see Details.
`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. `group` runs a *stratified* analysis (one independent model per level); to instead model a higher level *jointly*, crossed with the strata, use `context`. The two are different designs, so supplying both errors.
`context`	Optional character vector naming higher-level *context* column(s) in `data` (e.g. `"school"`, `"hospital"`, `"region"`), forwarded to `fit_maihda`. Each context enters every fitted model as a crossed random intercept – `outcome ~ covars + (1 | stratum) + (1 | context)` – the *contextual cross-classified MAIHDA* of the literature. The summaries then partition the unexplained variance into between-stratum vs. between-context vs. residual, the headline VPC/ICC becomes the between-stratum share *net of* the context, and the PCV decomposition is computed with the context partialled out (the context random intercept is carried by both the null and the adjusted model). Cannot be combined with `group`; a context with few levels weakly identifies its variance (consider `engine = "brms"`).
`engine`	Modeling engine, "lme4" (default), "brms", "wemix" (the design-weighted pseudo-maximum-likelihood fit; requires `sampling_weights` and is selected automatically when they are supplied with the default engine), or "ordinal" (cumulative link mixed model via `ordinal::clmm()`; selected automatically for an ordinal family or an ordered-factor outcome).
`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.
`decomposition`	How to decompose the intersectional inequality into additive and interaction parts. `"two-model"` (default) is the classic MAIHDA approach: a null model and an adjusted model (the dimensions' additive main effects as *fixed* effects), with the additive share read from the PCV (proportional change in between-stratum variance). `"crossed-dimensions"` fits a **single** model that enters each dimension's additive main effect as a *random* intercept – `outcome ~ covars + (1 | dim1) + ... + (1 | stratum)` – so each dimension's RE variance is its additive contribution and the intersection (`stratum`) RE variance is the interaction beyond additive; the additive and interaction *shares* of the total between-strata variance are read directly from that one fit. The two modes target the same scientific question with different estimators, so their additive shares are conceptually parallel but not numerically identical. The crossed-dimensions additive share is a partial-pooling estimate: dimensions with few levels (e.g. a binary sex variable, whose variance is estimated from two groups) are poorly identified and often give a singular lme4 fit – the `brms` engine handles this better. See Details. (`"cross-classified"` is accepted as a deprecated alias for `"crossed-dimensions"`, with a warning: in the MAIHDA literature "cross-classified" refers to the contextual stratum-by-place model, which this package fits via `context`.) `"longitudinal"` fits a 3-level **growth-curve** MAIHDA (requires `id` and `time`; selected automatically when they are supplied): a null and an adjusted growth model, where the adjusted model adds the dimensions' main effects *and their interactions with time* (`dim:time`). The between-stratum variance is then time-varying and the PCV is reported separately for the baseline (intercept) and the slope variance – the additive-vs-multiplicative split of the intersectional trajectory inequality (Bell, Evans, Holman & Leckie 2024). See `fit_maihda`.
`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.
`response_vpc`	Logical; for a binomial (lme4) outcome, also attach the response-scale VPC (`maihda_vpc_response`) to the model summaries. It is estimated by simulation, so it is opt-in (default `FALSE`) and uses `seed`. The discriminatory accuracy (AUC + MOR) is attached automatically for a binomial/Bernoulli outcome regardless of this flag (see Details).
`seed`	Optional integer seed for the response-scale VPC simulation.

`sampling_weights`	Optional name of a sampling-weight column for a **design-weighted MAIHDA** on complex-survey data; see `fit_maihda` for the full semantics (engine selection, the pseudo-likelihood weighting, and what is/is not design-based). Both the null and the adjusted model (and any per-group fits) use the same weights, so the PCV is a design-weighted decomposition. Not compatible with `engine = "lme4"`, `decomposition = "crossed-dimensions"` under the wemix engine, or `bootstrap = TRUE`.
`id`, `time`, `time_degree`	For a **longitudinal** MAIHDA: the person/unit identifier column, the numeric measurement-time column, and the growth-curve polynomial degree (1 = linear). Supplying `id`/`time` selects `decomposition = "longitudinal"`. See `fit_maihda` for the model structure; `group`, `context`, and `sampling_weights` are not supported alongside them. Default `NULL` (cross-sectional).
`interactions`	Whether to compute the per-stratum interaction diagnostic (`maihda_interactions`) on the adjusted / crossed-dimensions model and attach it as the `interactions` slot, surfaced in `print()`. `TRUE` (default) uses the diagnostic's own default correction (`adjust = "BH"`, FDR – since scanning all strata is a screening question); `FALSE` skips it; or pass a `p.adjust` method name – including `"none"` for the uncorrected, per-stratum individual-testing view. Uses `conf_level`. Not computed for a longitudinal decomposition. The computation is cheap (it reads the stratum estimates the summary already holds; no refit).
`...`	Additional arguments passed to `fit_maihda` (and on to `lmer`/`glmer`).

Details

Binomial companions. For a binary outcome the model summaries also carry the discriminatory accuracy (AUC / C-statistic and Median Odds Ratio) – the "DA" in MAIHDA – automatically, so the null model's strata-only AUC sits alongside its VPC; set `response_vpc = TRUE` to add the (simulation-based) response-scale VPC as well. These are read from `summary(x)` and the attached `summary_adjusted`, and the headline `print()` shows the null-model AUC.

This is a convenience wrapper around `fit_maihda`, `calculate_pvc`, `summary.maihda_model` and `compare_maihda_groups`. It is *intrinsically* a two-model decomposition and has no single-model mode – for a single fit (e.g. just the null-model VPC / discriminatory accuracy), call `fit_maihda` directly.

The dimensions' additive main effects. You may write them in the formula – the fully-specified, lme4-native adjusted model `outcome ~ covars + var1 + var2 + (1 | var1:var2)` – or omit them. Either way the null *excludes* the dimension main effects and the adjusted *includes* them: when the formula already lists them it is taken as the adjusted model and the null is derived by dropping them; when they are missing `maihda()` adds them to the adjusted model and emits a `message()` so the decomposition stays explicit. Only the *additive* main effects belong here: a fixed interaction among the stratum dimensions – `var1 * var2`, which R expands to `var1 + var2 + var1:var2` – duplicates the intersectional stratum random intercept (it absorbs the between-stratum variance into fixed cell means, which makes the PCV invalid), so it is **rejected with an error**. Write `var1 + var2`; the intersection is estimated by the stratum random effect (and quantified by `decomposition = "crossed-dimensions"` or `maihda_interactions`). The dimensions themselves are read from the random term: the shorthand `(1 | var1:var2)` and `make_strata` both record them, and a numeric dimension that `make_strata()` auto-binned enters the adjusted model as its reconstructed tertile factor (matching the strata), not as a linear term. Because `maihda()` is intrinsically a decomposition, it **errors** (rather than returning a null-only result) when it cannot build the adjusted model – when the dimensions cannot be recovered (a hand-built `stratum` column records none) or there is only one dimension (no intersection to decompose). Use `fit_maihda` for those single-model fits.

Value

An object of class `maihda_analysis`: a list with

`model`	the fitted `maihda_model` (see `fit_maihda`); the **null** model in `"two-model"` mode, or the single **crossed-dimensions** model in `"crossed-dimensions"` mode
`summary`	the model's `maihda_summary` (VPC/ICC, variance components, stratum estimates; plus the additive/interaction `decomposition` in crossed-dimensions mode, and the stratum-vs-context `context` partition when `context` is supplied)
`model_adjusted`	the fitted **adjusted** `maihda_model` (`"two-model"` and `"longitudinal"` modes; `NULL` otherwise)
`summary_adjusted`	the adjusted model's `maihda_summary`, or `NULL`
`pcv`	the proportional change in variance: the `pvc_result` from `calculate_pvc` in `"two-model"` mode, or a `maihda_long_pcv` (the intercept/slope and time-specific PCV) in `"longitudinal"` mode; `NULL` otherwise
`decomposition`	the additive/interaction partition (additive and interaction variances and shares, per-dimension variances; `"crossed-dimensions"` mode only, `NULL` otherwise)
`groups`	a `maihda_group_comparison` when group is supplied, otherwise `NULL`
`interactions`	the `maihda_interactions` diagnostic (per-stratum interaction BLUPs and flags) when `interactions` is not `FALSE`, otherwise `NULL`
`mode`	`"two-model"`, `"crossed-dimensions"`, or `"longitudinal"`
`context_vars`	the context variable name(s) when `context` was supplied, otherwise `NULL`
`formula`, `adjusted_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: null + adjusted fit, VPC summary, and PCV decomposition. Writing the
# dimensions' additive main effects (Gender + Race) gives the fully-specified
# adjusted model; maihda() derives the null by dropping them.
a <- maihda(BMI ~ Age + Gender + Race + (1 | Gender:Race), data = maihda_health_data)
a                              # VPC (null) and PCV (null -> adjusted)
a$pcv                          # proportional change in between-stratum variance
a$formula                      # null:     BMI ~ Age + (1 | stratum)
```

```
a$adjusted_formula            # adjusted: null + Gender + Race main effects

# Omitting them is equivalent -- maihda() adds them to the adjusted model and
# emits a message; the null and PCV are identical to the explicit form above.
a0 <- maihda(BMI ~ Age + (1 | Gender:Race), data = maihda_health_data)

plot(a, type = "vpc")          # null model
plot(a, type = "effect_decomp")# adjusted model (additive vs intersectional)

# Crossed-dimensions decomposition: one model, the dimensions' main effects entered
# as RANDOM intercepts. The additive and interaction shares of the between-strata
# variance are read directly from the single fit (no null/adjusted pair).
cc <- maihda(BMI ~ Age + (1 | Gender:Race), data = maihda_health_data,
             decomposition = "crossed-dimensions")
cc                                   # VPC and additive/interaction shares
cc$decomposition$additive_share      # crossed-dimensions analogue of the PCV
cc$formula                           # BMI ~ Age + (1|Gender) + (1|Race) + (1|stratum)

# 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")
plot(a2, type = "group_pcv")

# Contextual cross-classified MAIHDA: instead of one model per country (group=),
# model the strata CROSSED with country in a single fit. The summary partitions
# the unexplained variance into between-stratum vs. between-country vs. residual,
# and the PCV is computed with country partialled out.
a3 <- maihda(math ~ 1 + (1 | gender:ses), data = maihda_country_data,
             context = "country")
a3
a3$summary$context$vpc_context_total  # the country (general contextual) share
plot(a3, type = "context_vpc")
```

`maihda_auc` *Area under the ROC curve (C-statistic), rank-based*

Description

Computes the AUC / C-statistic as the Mann-Whitney U statistic: the probability that a randomly chosen case (`y == 1`) is assigned a higher predicted value than a randomly chosen non-case (`y == 0`), with ties counting as one half. This needs no external package. An AUC of 0.5 is chance; 1 is perfect separation.

Usage

```
maihda_auc(prob, y)
```

Arguments

`prob`	Numeric vector of predicted probabilities (or any score where larger means more case-like).
`y`	Observed binary outcome as 0/1 numeric or logical, the same length as `prob`.

Value

A single number in `[0, 1]`, or `NA_real_` if either class is absent.

References

Merlo, J., Wagner, P., Ghith, N., & Leckie, G. (2016). An original stepwise multilevel logistic regression analysis of discriminatory accuracy: the case of neighbourhoods and health. *PLOS ONE*, 11(4), e0153778.

Examples

```
maihda_auc(c(0.1, 0.4, 0.35, 0.8), c(0, 0, 1, 1))
```

`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 does **not** carry the PISA survey weights or complex sampling design, so results are **not** survey-representative and should not be used for substantive cross-national inference. (For your own survey data, the package supports design-weighted MAIHDA via the `sampling_weights` argument of `fit_maihda` / `maihda`.)

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_cumulative` *Cumulative (ordinal) family marker for MAIHDA models*

Description

Specifies a cumulative (proportional-odds) model for an ordinal outcome in `fit_maihda` / `maihda`, with a choice of link: `maihda_cumulative("logit")` (the default, equivalent to `family = "ordinal"`) or `maihda_cumulative("probit")`. It plays the role a `stats` family object plays for the other families – there is no cumulative family constructor in `stats`, and using `brms::cumulative()` would require brms for a frequentist fit.

Usage

```
maihda_cumulative(link = c("logit", "probit"))
```

Arguments

`link`	The cumulative link: `"logit"` (default) or `"probit"`. These are the links for which the latent-scale VPC is defined (level-1 variance $\pi^2/3$ and 1 respectively).

Value

A family marker list with elements `family = "cumulative"` and `link`.

See Also

`fit_maihda`

Examples

```
maihda_cumulative()
maihda_cumulative("probit")
```

`maihda_discriminatory_accuracy` *Discriminatory accuracy of a binary MAIHDA model*

Description

Bundles the individual-level discriminatory-accuracy summaries for a binomial MAIHDA model: the AUC / C-statistic (how well the model's predicted probabilities separate cases from non-cases) and the Median Odds Ratio. Applied to a strata-only (null) model, the AUC is the discriminatory accuracy of the intersectional strata themselves – Merlo's central quantity; comparing it with an adjusted model shows whether individual covariates beyond stratum membership sharpen classification. The AUC is computed for any binomial link; the Median Odds Ratio is reported only for the logit link and is `NA` otherwise (e.g. for a probit fit), since the MOR is an odds-ratio-scale quantity.

Aggregated-binomial fits are supported on both engines that fit them – an lme4 `cbind(success, failure)` response and a brms `y | trials(n)` response: the AUC is the count-weighted C-statistic over the implied individual-level 0/1 data, and `n_case` / `n_control` are the total successes / failures.

Usage

```
maihda_discriminatory_accuracy(model)
```

Arguments

`model`	A `maihda_model` from `fit_maihda` fitted with a `binomial` family – including an aggregated response (an lme4 `cbind(success, failure)` or a brms `y	trials(n)`) – or the `bernoulli` family a binary 0/1 outcome is fit with under `engine = "brms"`.

Value

An object of class `maihda_da`: a list with `auc`, `mor`, `n_case`, `n_control`, `family`, `link` and `engine`. `mor` is `NA` for a non-logit binomial link, where the AUC is still reported. For an aggregated-binomial fit `n_case` / `n_control` are the total successes / failures.

References

Merlo, J. (2018). Multilevel analysis of individual heterogeneity and discriminatory accuracy (MAIHDA) within an intersectional framework. *Social Science & Medicine*, 203, 74-80.

See Also

`maihda_auc`, `maihda_mor`

Examples

```
# Obese (Yes/No) by intersectional strata of Gender x Race
strata <- make_strata(maihda_health_data, vars = c("Gender", "Race"))
d <- maihda_health_data
d$stratum <- strata$data$stratum
m <- fit_maihda(Obese ~ (1 | stratum), data = d, family = "binomial")
maihda_discriminatory_accuracy(m)
```

`maihda_glance` *Glance at a MAIHDA model or analysis*

Description

glance methods that return the MAIHDA headline as a one-row `tibble`: the variance partition coefficient (VPC/ICC), and – for a `maihda_analysis` – the proportional change in variance (PCV), plus the additive/interaction shares and the discriminatory accuracy (AUC, MOR) for a binomial outcome. The layout is uniform across the lme4, brms, WeMix and ordinal engines. No other package emits this row from the underlying fit, because PCV needs the null+adjusted pair that only a `maihda_analysis` holds.

Usage

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

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

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

Arguments

`x`	A `maihda_summary`, `maihda_model`, or `maihda_analysis`.
`...`	Unused, for S3 consistency.

Value

A one-row `tibble`. `glance.maihda_analysis` adds `pcv` (with `pcv.conf.low`/`pcv.conf.high` when bootstrapped or from a brms posterior), the adjusted-model `auc.adjusted`, `nobs`, `family` and `mode` to the columns produced for a single summary.

See Also

`maihda_tidiers` for the per-estimate `tidy()` methods.

Examples

```
data("maihda_health_data")
a <- maihda(BMI ~ Age + Gender + Race + Education + (1 | Gender:Race:Education),
            data = maihda_health_data)
glance(a)
```

`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 does **not** carry the NHANES survey weights or complex sampling design, so results are **not** survey-representative and should not be used for substantive population inference. (For your own survey data, the package supports design-weighted MAIHDA via the `sampling_weights` argument of `fit_maihda` / `maihda`.)

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_ic` *Information criteria for MAIHDA models*

Description

Reports the relative-fit information criteria for one or more MAIHDA models, to help choose between model *structures* (different covariate sets, strata definitions, or families) – a question the VPC/ICC and PCV do not address. The criteria reported depend on the engine: **AIC** and **BIC** for the likelihood engines (`lme4`, and `ordinal::clmm`), and the Bayesian **WAIC** and **LOOIC** (leave-one-out information criterion) for `brms`. Lower is better for all four.

Usage

```
maihda_ic(..., model_names = NULL)
```

Arguments

`...`	One or more `maihda_model` objects (from `fit_maihda`) or `maihda_analysis` objects (from `maihda`). A `maihda_analysis` contributes its null model and, when present, its adjusted model as separate rows.
`model_names`	Optional character vector of names, one per `...` argument. A `maihda_analysis` argument's null/adjusted rows are suffixed from its name.

Details

REML vs ML. `lmer` fits Gaussian models by REML by default, and a REML log-likelihood (hence its AIC/BIC) is *not* comparable across models with different fixed effects – exactly the canonical MAIHDA null-vs-adjusted comparison. When more than one model is supplied, `maihda_ic()` therefore refits any REML `lmer` model with maximum likelihood (`refitML`) before computing AIC/BIC, matching the behaviour of `anova()` on `lme4` models; the `estimator` column records when this happened. For a single model the criterion is reported as fitted (the `estimator` column then reads `"REML"`).

Comparability. Like the VPC, information criteria are only comparable across models fitted to the *same* analytic sample (same rows and outcome). AIC/BIC additionally require the same response distribution – they are not comparable across families (e.g. a Gaussian vs a Poisson fit), nor between the likelihood engines and `brms` (AIC/BIC vs WAIC/LOOIC are different scales). `maihda_ic()` does not enforce this; `compare_maihda` warns when the supplied models differ in outcome, sample, or family.

Design-weighted fits. For the `wemix` (design-weighted) engine the criteria are reported as `NA`: a pseudo-likelihood with sampling weights does not define a standard AIC/BIC.

Value

A `data.frame` of class `maihda_ic` with one row per model and the columns that apply: `model`, `n` (analytic sample size), `estimator`, `df` (number of parameters; likelihood engines), `logLik`, `AIC`, `BIC` (likelihood engines), `WAIC`, `LOOIC` (brms), and – when more than one model is supplied – `delta` (the difference from the best model on the primary criterion: AIC for the likelihood engines, LOOIC for brms). Columns that are entirely `NA` across the supplied models are dropped.

See Also

`compare_maihda`, which reports these criteria alongside the VPC/ICC, and `calculate_pvc` for the variance decomposition.

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)

# AIC/BIC for two nested structures (REML lmer fits are ML-refitted first)
maihda_ic(null_model, adj_model, model_names = c("Null", "Adjusted"))

# Or straight from a one-call maihda() analysis (null + adjusted rows)
a <- maihda(health_outcome ~ age + gender + race + (1 | gender:race),
            data = maihda_sim_data)
maihda_ic(a)
```

`maihda_interactions` *Flag strata with credibly non-zero intersectional interaction*

Description

Reports, for each intersectional stratum, the **interaction** component of its outcome – the stratum random effect (BLUP) of an *adjusted* MAIHDA model, i.e. how far the stratum departs from the additive main-effects prediction of its defining dimensions – and **flags** the strata whose interaction is credibly different from zero. This is the heart of "where is there genuine intersectionality": a flagged stratum is one whose joint identity produces an outcome the additive parts do not.

Usage

```
maihda_interactions(object, conf_level = 0.95, adjust = "BH", rope = NULL, ...)
```

Arguments

`object`	A `maihda_analysis` from `maihda` (preferred – its adjusted / crossed-dimensions model is used automatically) or a `maihda_model` from `fit_maihda` (which should be the *adjusted* model; a null model is accepted but warned about).
`conf_level`	Confidence / credible level for the interval and the flag. Default 0.95.

`adjust`	Multiple-comparison adjustment for the per-stratum p-values (frequentist engines only): `"BH"` (default; false-discovery rate) or any method accepted by `p.adjust`, including `"none"` for the uncorrected, per-stratum individual-testing view. Ignored for `brms` (which uses the posterior tail directly; a message is shown only if you set it explicitly).
`rope`	Optional equivalence region (a "smallest interaction of interest") for an "is the interaction *negligible*?" reading (Schuirmann 1987; Kruschke 2018), on the link (latent) scale. `NULL` (default) gives only the usual zero-centred flag. A single positive number `d` means the symmetric region `c(-d, d)`; or supply `c(lower, upper)`. When set, the result gains a `decision` column classifying each stratum from its `conf_level` interval relative to the region: `"relevant"` (interval entirely outside it), `"negligible"` (entirely inside it), or `"inconclusive"` (straddling a bound).
`...`	Currently unused.

Details

It must be read off the adjusted model. Only when the dimensions' additive main effects are in the model (the *adjusted* model of the two-model decomposition, or the crossed-dimensions model) does the stratum random effect isolate the *pure interaction*. On a null model the stratum random effect is the total between-stratum deviation (additive + interaction), so passing one is flagged with a warning. The opposite mis-specification is flagged too: a model that adds a *fixed* interaction among the dimensions (e.g. `var1 * var2`) absorbs the intersectional effect into fixed cell means, so the stratum random effect is no longer the pure interaction. Passing a `maihda` result uses the right model automatically.

Frequentist vs. Bayesian evidence. For the frequentist engines (`lme4`, `wemix`, `ordinal`) the flag comes from the BLUP's conditional standard error: a Wald interval at `conf_level` and a two-sided p-value, with an optional multiplicity correction (`adjust`). For `brms` the full posterior is already available, so the *exact* posterior tail is used – a credible interval at `conf_level` and the probability of direction `pd = max(P(BLUP > 0), P(BLUP < 0))` (in `[0.5, 1]`; the sign is in `direction`) – and `adjust` is not applied (the Bayesian answer is multiplicity-free).

Multiplicity: partial pooling and a correction are different things, and the experts disagree.

- *Shrinkage (magnitude/sign).* The stratum BLUP is partially pooled, so extreme values are regularised toward the grand mean, attenuating exaggerated-magnitude and wrong-sign (Type M/S) error (Gelman & Carlin 2014). Gelman, Hill & Yajima (2012) argue this shrinkage *usually substitutes* for a classical multiple-comparisons correction (the problem can "disappear entirely" in the hierarchical model); on that view the flag/no-flag step itself is what to avoid – the null of an *exactly* zero interaction is rarely the question (McShane, Gelman et al. 2019) – so report the estimate and its interval.
- *Whether to correct.* If you do want an error-rate screen, whether a correction is warranted depends on the *inferential structure* of the claim – the joint hypothesis, not the number of strata (Rubin 2021). Each stratum as its own pre-specified hypothesis ("does *this* stratum interact?") is *individual* testing and needs none – **only** if you do not also read the flags collectively. Once the question is "is there an interaction *somewhere*?" – which an automated all-strata scan effectively is – it is *disjunction* testing and a correction applies.

`adjust = "BH"` **is the default**: fitting and flagging every stratum in one call is the disjunction/screening case, where controlling the expected *proportion* of false discoveries (FDR) is the appropriate goal. Pass `adjust = "none"` only when each stratum is a genuine, pre-specified individual hypothesis. The FDR choice (over family-wise `"bonferroni"`/`"holm"`) is this package's, matching that screening goal; it is not a recommendation of Rubin (2021), who raises FDR only to distinguish it from the family-wise rate. The flag itself is a Wald test on a shrunken BLUP whose conditional SE treats the variance components as known, so it (and any `adjust` on it) is an explicit, approximate *screen*, not a procedure inheriting an exact guarantee from the model. Lead with the interval (and, for `brms`, the probability of direction); the substantive question is often not whether an interaction differs from zero but whether it exceeds a smallest interaction of interest (an equivalence reading; Schuirmann 1987; Kruschke 2018), read from the interval.

The interaction is reported on the model's link (latent) scale – a log-odds deviation for a logistic model, etc. – because the additive/interaction split is only exact there.

Value

An object of class `maihda_interactions` (a data frame), one row per stratum, sorted flagged-first then by `abs(interaction)`. Columns common to every engine: `stratum`, `label`, `n` (stratum size), `interaction` (the BLUP), `lower`/`upper` (the interval), `flagged` (logical), and `direction` (`"above"`/`"below"` the additive expectation). Frequentist fits add `se` and `p_value` (and `p_adjusted` when `adjust != "none"`); `brms` adds `pd` (probability of direction, `max(P(>0), P(<0))` in `[0.5, 1]`). When `rope` is set, a `decision` column (`"relevant"`/`"negligible"`/`"inconclusive"`) is added. Attributes record `conf_level`, `adjust`, `rope`, `engine`, `model_type`, `n_strata`, `n_flagged`, `scale` and `singular`.

References

Evans, C. R., Williams, D. R., Onnela, J. P., & Subramanian, S. V. (2018). A multilevel approach to modeling health inequalities at the intersection of multiple social identities. *Social Science & Medicine*, 203, 64-73.

Merlo, J. (2018). Multilevel analysis of individual heterogeneity and discriminatory accuracy (MAIHDA) within an intersectional framework. *Social Science & Medicine*, 203, 74-80.

Gelman, A., Hill, J., & Yajima, M. (2012). Why we (usually) don't have to worry about multiple comparisons. *Journal of Research on Educational Effectiveness*, 5(2), 189-211.

Gelman, A., & Carlin, J. (2014). Beyond power calculations: assessing Type S (sign) and Type M (magnitude) errors. *Perspectives on Psychological Science*, 9(6), 641-651.

Rubin, M. (2021). When to adjust alpha during multiple testing: a consideration of disjunction, conjunction, and individual testing. *Synthese*, 199(3-4), 10969-11000. doi:10.1007/s11229021-032764

McShane, B. B., Gal, D., Gelman, A., Robert, C., & Tackett, J. L. (2019). Abandon statistical significance. *The American Statistician*, 73(sup1), 235-245.

Schuirmann, D. J. (1987). A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. *Journal of Pharmacokinetics and Biopharmaceutics*, 15(6), 657-680.

Kruschke, J. K. (2018). Rejecting or accepting parameter values in Bayesian estimation. *Advances in Methods and Practices in Psychological Science*, 1(2), 270-280.

See Also

`maihda`, `calculate_pvc`, `summary.maihda_model`; and `plot(..., highlight_interactions = TRUE)` to mark the flagged strata on the effect-decomposition / predicted / shrinkage plots.

Examples

```
data(maihda_health_data)
a <- maihda(BMI ~ Age + Gender + Race + (1 | Gender:Race),
            data = maihda_health_data)
maihda_interactions(a)                 # FDR-screened (default adjust = "BH")
maihda_interactions(a, adjust = "none") # uncorrected per-stratum individual view
maihda_interactions(a, rope = 0.1)      # equivalence: |interaction| within 0.1?
```

`maihda_long_data` *Simulated Longitudinal Data for MAIHDA*

Description

A simulated long-format (one row per person-occasion) panel for demonstrating **longitudinal / growth-curve MAIHDA** (`fit_maihda(id =, time =)` and `maihda(decomposition = "longitudinal")`). 600 individuals are each measured over five waves on a continuous wellbeing score, within 12 intersectional strata defined by gender x ethnicity x education. The between-stratum *trajectory* differences are constructed to be mostly additive (each dimension's main effect on the baseline level and on the rate of change) with one genuine multiplicative interaction, so the longitudinal PCV – a high but sub-1 `PCV_slope` – is demonstrable.

Usage

```
maihda_long_data
```

Format

A data frame with 3000 rows (600 persons x 5 waves) and 8 variables:

id Person identifier (level 2); repeated across waves.

wave Measurement occasion, 0 to 4 (the numeric time variable).

gender Gender (`Women`/`Men`); a stratum dimension.

ethnicity Ethnicity (`EthA`/`EthB`/`EthC`); a stratum dimension.

education Education (`Low`/`High`); a stratum dimension.

age Baseline age in years, a time-invariant covariate.

wellbeing Continuous wellbeing outcome (the growth-curve response).

low_wellbeing Binary companion outcome (1 = bottom 40% of wellbeing), for the logistic longitudinal path.

Source

Simulated for the purpose of the MAIHDA package. The growth structure follows the longitudinal MAIHDA of Bell, Evans, Holman & Leckie (2024) doi:10.1016/j.socscimed.2024.116955.

Examples

```
data(maihda_long_data)

# Time-varying VPC from a 3-level growth model:
m <- fit_maihda(wellbeing ~ wave + (1 | gender:ethnicity:education),
                data = maihda_long_data, id = "id", time = "wave")
summary(m)

# Additive-vs-multiplicative PCV (null vs adjusted growth model):
a <- maihda(wellbeing ~ wave + (1 | gender:ethnicity:education),
            data = maihda_long_data, id = "id", time = "wave",
            decomposition = "longitudinal")
a$pcv
```

`maihda_mor` *Median Odds Ratio (MOR) for a logistic MAIHDA model*

Description

The Median Odds Ratio translates the between-stratum variance of a logistic MAIHDA model onto the odds-ratio scale: the median relative change in the odds of the outcome when comparing two individuals from randomly chosen strata (higher- vs lower-risk). `MOR = exp(sqrt(2 * V_A) * qnorm(0.75))`, where `V_A` is the between-stratum (latent, logit-scale) variance. An MOR of 1 indicates no between-stratum heterogeneity. The MOR is defined only for the **logit** link (it is the median *odds* ratio); a non-logit binomial fit such as `probit` is rejected, because its latent variance is on a different scale and the `exp(...)` above would not be an odds ratio.

For a **cumulative-logit** (ordinal) MAIHDA model the same formula applies to the latent logit-scale between-stratum variance and is the *median cumulative odds ratio*: the median relative change in the odds of being at or below any given outcome category between two randomly chosen strata (under the model's proportional-odds assumption it is the same for every category split).

Usage

```
maihda_mor(model)
```

Arguments

`model`	A `maihda_model` from `fit_maihda` fitted with a `binomial` (lme4), `bernoulli` (brms), or `cumulative` (ordinal) family and a **logit** link.

Value

A single number (the MOR, ≥ 1), or `NA_real_` if the between-stratum variance is unavailable.

References

Larsen, K., & Merlo, J. (2005). Appropriate assessment of neighborhood effects on individual health: integrating random and fixed effects in multilevel logistic regression. *American Journal of Epidemiology*, 161(1), 81-88.

See Also

`maihda_discriminatory_accuracy`

`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_sparse_data` *Sparse Intersectional Data for Bayesian MAIHDA*

Description

A simulated cross-sectional dataset built to showcase **Bayesian (brms) MAIHDA for sparse intersections** – the regime where many intersectional strata each hold only a handful of individuals. There the maximum-likelihood (lme4) estimate of the *interaction* between-stratum variance collapses to a singular fit with no uncertainty, so the additive-vs-interaction split is both unstable and falsely precise; weakly-informative priors (‘engine = "brms"‘) regularise the variance off the boundary and return a calibrated credible interval.

Usage

```
maihda_sparse_data
```

Format

A data frame with 240 rows and 6 variables:

gender Strata dimension (Women/Men).

ethnicity Strata dimension (White/Black/Asian).

education Strata dimension (Low/High).

age_group Strata dimension (Young/Mid/Older).

y A continuous (Gaussian) outcome. True between-stratum VPC 0.26, of which 40% is the intersectional interaction.

event A binary outcome (No/Yes), ~46% "Yes". Its latent-scale between-stratum VPC is 0.31, again 40% interaction.

The exact generative truth is also attached as `attr(maihda_sparse_data, "truth")` (additive/interaction variances, shares, and VPCs for each outcome).

Details

The data carry a **known, non-trivial interaction** so the vignette can claim *recovery* rather than merely report numbers: 4 dimensions form 36 intersectional strata with deliberately skewed sizes (median 6 individuals, 12 of 36 cells below 5, two singletons), and the true interaction accounts for **40 between-stratum variance** on both outcomes. On the binary outcome a genuine 40 interaction is read by lme4 as roughly 3 that is purely a small-cell artifact.

Note

A purely illustrative dataset. The dimension labels are arbitrary and the interaction is constructed, not estimated from any real population – its only purpose is to make the sparse-cell behaviour of the ML and Bayesian estimators visible against a known answer.

Source

Simulated; see `data-raw/maihda_sparse_data.R`.

Examples

```
data(maihda_sparse_data)
attr(maihda_sparse_data, "truth")$gaussian$interaction_share  # 0.40

# ML over-shrinks the interaction under sparse cells (a singular fit):
# m_lme4 <- maihda(y ~ 1 + (1 | gender:ethnicity:education:age_group),
#                  data = maihda_sparse_data, decomposition = "crossed-dimensions")
#
# Weakly-informative priors regularise it and report honest uncertainty:
# m_brms <- maihda(y ~ 1 + (1 | gender:ethnicity:education:age_group),
#                  data = maihda_sparse_data, decomposition = "crossed-dimensions",
#                  engine = "brms",
#                  prior = brms::set_prior("normal(0, 0.5)", class = "sd"))
# See vignette("bayesian_sparse_maihda").
```

`maihda_table` *Canonical MAIHDA results table and ranked-strata table*

Description

Assembles the two standard MAIHDA write-up deliverables from a fitted analysis in one call: (a) a **model-results table** contrasting the null and adjusted models (intercept, between-stratum variance and SD, VPC/ICC, the PCV, and – for a binary outcome – the AUC and Median Odds Ratio), and (b) a **ranked-strata table** ordering the intersectional strata by their predicted outcome, so the best- and worst-off strata can be read directly. It introduces no new estimator: the model-results table reuses the quantities from `summary()` (calling `summary()` itself for a bare `fit_maihda` model), and the ranked-strata table reuses the same stratum predictions as `plot(type = "predicted")`, so the table agrees exactly with `summary()` and `plot()`.

Usage

```
maihda_table(
  x,
  n_strata = 10L,
  scale = c("response", "link"),
  which = c("null", "adjusted"),
  digits = 3,
  ...
)
```

Arguments

`x`	A `maihda_analysis` from `maihda` (the usual input), or a single `maihda_model` from `fit_maihda`.
`n_strata`	Number of strata to show at each end (top and bottom) in the printed ranked-strata table. The returned `$strata` holds all strata (`NULL` for a longitudinal fit, whose strata are trajectories rather than single ranked values – see `$strata` in Value). Default 10.
`scale`	Scale for the predicted stratum values: `"response"` (default) or `"link"`. For a cumulative (ordinal) model the response scale is the expected category score.
`which`	For a two-model analysis, which model's predictions to rank the strata by: `"null"` (default) or `"adjusted"`. Ignored for a crossed-dimensions analysis or a single model.
`digits`	Number of decimal places for the `print()` method. Default 3.
`...`	For a `maihda_model` input, additional arguments passed to `summary.maihda_model` (e.g. `bootstrap = TRUE`); ignored for a `maihda_analysis` input, whose summaries are already computed.

Details

The model-results table is mostly numeric and export-ready (e.g. `write.csv(maihda_table(a)$models, ...)` or pass it to `knitr::kable()`): statistics are rows, models are columns, and each estimate has accompanying `*_lower`/`*_upper` columns that hold the confidence/credible interval when one is available (the VPC bootstrap or posterior interval, and the bootstrap PCV interval) and `NA` otherwise. The intercept and the variance/SD rows are point estimates. The `print()` method renders the same table in the familiar "estimate [low, high]" layout.

For a `"crossed-dimensions"` analysis (one model, no null/adjusted pair) the results table has a single estimate column and gains "Additive share" / "Interaction share" rows instead of the PCV. For a contextual cross-classified analysis (`maihda(context = )`) it gains a "Context share (VPC)" row. A bare `fit_maihda` model is also accepted and yields a single-model table (no PCV).

The ranked-strata table ranks every stratum by its model-predicted outcome (on the `scale` requested), using the same stratum predictions as `plot(type = "predicted")`: the predicted value carries the conditional (random-effect) interval, and the stratum random effect (BLUP) is reported alongside it. By default the ranking uses the **null** model – the headline intersectional inequality (which strata fare best/worst overall); set `which = "adjusted"` to rank by the adjusted model instead. The full ranked table is returned in `$strata`; `print()` shows the top and bottom `n_strata`.

Value

An object of class `maihda_table`: a list with

`models`	a data frame of the model-results table (statistics in rows; one estimate column per model, each with `*_lower`/`*_upper` interval columns)
`strata`	a data frame of all strata ranked by predicted outcome, with `rank`, `stratum`, `label`, `n`, the predicted value and its conditional interval, and the stratum random effect and its interval. `NULL` when a single ranked value per stratum is not

	defined – in particular for a longitudinal (growth-curve) fit, whose strata are trajectories (see `strata_note`) – or if the stratum predictions could not otherwise be computed
`strata_note`	a character note explaining why `strata` is `NULL` (e.g. a longitudinal fit), or `NULL` when a ranked-strata table was produced
`models_note`	a character note (or `NULL`) flagging that the PCV was computed from maximum-likelihood-refitted between-stratum variances while the variance/VPC rows are each model's own (REML) estimate, so the PCV need not equal the reduction implied by the displayed variance rows; see `calculate_pvc`
`model_keys`, `model_labels`	the estimate-column keys and their display labels
`family`, `engine`, `mode`, `scale`, `ranked_by`, `n_obs`, `n_strata_total`, `context_vars`	metadata used by `print()`

See Also

`maihda`, `summary.maihda_model`, `calculate_pvc`, `maihda_discriminatory_accuracy`.

Examples

```
data(maihda_health_data)
a <- maihda(BMI ~ Age + Gender + Race + (1 | Gender:Race), data = maihda_health_data)

tab <- maihda_table(a)
tab                 # printed: model-results table + top/bottom strata
tab$models          # the numeric, export-ready results table
tab$strata          # all strata ranked by predicted BMI

# write.csv(tab$models, "results.csv", row.names = FALSE)
```

`maihda_tidiers` *Tidy a MAIHDA summary, model, or analysis*

Description

`tidy` methods that return the MAIHDA estimates as a tidy `tibble`, ready for downstream tables (`gt`, `flextable`) and `ggplot2`. They read the slots that `summary.maihda_model` already computes and add no new statistics.

Usage

```
## S3 method for class 'maihda_summary'
tidy(x, component = c("strata", "variance", "fixed"), ...)
```

```
## S3 method for class 'maihda_model'
tidy(x, component = c("strata", "variance", "fixed"), ...)

## S3 method for class 'maihda_analysis'
tidy(
  x,
  component = c("strata", "variance", "fixed"),
  which = c("null", "adjusted"),
  ...
)
```

Arguments

`x`	A `maihda_summary` (from `summary`), a `maihda_model` (from `fit_maihda`), or a `maihda_analysis` (from `maihda`).
`component`	Which estimates to return: `"strata"` (default) the stratum (intersection) random-effect estimates – one row per stratum, with `estimate`, `std.error` and `conf.low`/`conf.high`, plus the human-readable intersectional `label` when available. `"variance"` the variance-components table (between-stratum, any other random effects, residual, and total) with each component's variance, SD and proportion. `"fixed"` the fixed-effect estimates, in broom's `term`/`estimate`/`std.error` shape (with `conf.low`/`conf.high` for the brms engine).
`...`	Unused, for S3 consistency.
`which`	For a `maihda_analysis`, whether to tidy the `"null"` (default) or `"adjusted"` model's summary.

Value

A tibble. For `component = "strata"`: columns `stratum`, `label`, `estimate`, `std.error`, `conf.low`, `conf.high`. For `"variance"`: `component`, `variance`, `sd`, `proportion`. For `"fixed"`: `term`, `estimate`, `std.error`, `conf.low`, `conf.high`.

See Also

`glance.maihda_analysis` for the one-row model summary.

Examples

```
data("maihda_health_data")
m <- fit_maihda(BMI ~ Age + (1 | Gender:Race:Education), data = maihda_health_data)
tidy(m)                        # stratum estimates
tidy(m, component = "variance")
tidy(m, component = "fixed")
```

`maihda_upset_size` *Recommended Figure Size for the UpSet Stratum Plot*

Description

Computes sensible `width` and `height` (in inches) for `plot(object, type = "upset")`, so a knitr chunk or `ggsave()` call can size the figure to its content. The UpSet composite grows *taller* with the number of matrix rows (one per binary 0/1 dimension, one per level of a multi-level factor) and *wider* with the number of strata columns shown, so a single fixed size tends to crop or stretch it – particularly for multi-level designs (many rows) or a large `n_strata` (many columns; UpSet is an inherently wide format).

Usage

```
maihda_upset_size(object, n_strata = 50)
```

Arguments

`object`	A `maihda_model` from `fit_maihda` or a `maihda` analysis from `maihda`; it must carry the per-dimension stratum table from `make_strata`.
`n_strata`	Maximum number of strata the plot will show – pass the same value you give `plot()`. `NULL` means all strata. Default 50.

Value

A list with numeric `width` and `height` (inches) plus the `rows` (matrix rows) and `cols` (strata shown) they derive from.

See Also

`plot.maihda_model`

Examples

```
strata <- make_strata(maihda_sim_data, vars = c("gender", "race"))
model <- fit_maihda(health_outcome ~ (1 | stratum), data = strata$data)
sz <- maihda_upset_size(model, n_strata = 30)
ggplot2::ggsave(
  tempfile(fileext = ".png"),
  plot(model, type = "upset", n_strata = 30),
  width = sz$width, height = sz$height)
```

`maihda_vpc_response` *Response-scale VPC for a binomial MAIHDA model*

Description

Computes the variance partition coefficient on the response (probability) scale for a binomial MAIHDA model, using the simulation method of Goldstein, Browne & Rasbash (2002). Stratum random effects $u \sim N(0, \sigma_u^2)$ are simulated and converted to predicted probabilities $p = g^{-1}(\eta + u)$ (with η the fixed-part linear predictor); the VPC is then the between-stratum variance of p as a share of the total (between + the binomial within-stratum variance $\overline{p(1-p)}$).

Unlike the latent-scale VPC (fixed level-1 variance $\pi^2/3$ for the logit), the response-scale VPC depends on the overall outcome prevalence, so report it as a complement to – not a replacement for – the latent-scale value.

Usage

```
maihda_vpc_response(model, n_sim = 10000, seed = NULL)
```

Arguments

`model`	A binomial `maihda_model` (lme4 engine) from `fit_maihda`.
`n_sim`	Number of Monte-Carlo draws of the stratum random effect (>= 100). Default 10000.
`seed`	Optional integer seed for reproducibility.

Details

The fixed part η is collapsed to a single value – the mean linear predictor $\bar{\eta}$ over the analytic sample – before the random effect is simulated around it. The result is therefore a VPC *evaluated at the mean covariate profile* (a conditional-at-mean estimate), not one marginalised over the empirical covariate distribution. For the canonical strata-only (null) model η is constant (the intercept), so the two coincide and the value is exact. For an *adjusted* model (one with covariates) they can differ, because the inverse link is nonlinear and $g^{-1}(\bar{\eta}) \neq \overline{g^{-1}(\eta)}$: read the response-scale VPC from the null model, or interpret an adjusted value as conditional on the average covariate profile rather than as a covariate-averaged (marginal) VPC.

The method is binomial-link agnostic: it maps the simulated stratum effects through whichever inverse link the model uses (logit, probit, cloglog, ...), so a non-logit binomial fit is computed on its own scale rather than rejected. Only the family is required to be binomial.

Value

An object of class `maihda_vpc_response`: a list with `estimate`, `scale = "response"`, `method = "simulation"`, `n_sim`, `var_between` (the latent-scale between-stratum variance) and `lp_fixed` (the mean fixed-part linear predictor).

References

Goldstein, H., Browne, W., & Rasbash, J. (2002). Partitioning variation in multilevel models. *Understanding Statistics*, 1(4), 223-231.

See Also

`maihda_discriminatory_accuracy`, `summary.maihda_model`

Examples

```
strata <- make_strata(maihda_health_data, vars = c("Gender", "Race"))
d <- maihda_health_data
d$stratum <- strata$data$stratum
m <- fit_maihda(Obese ~ (1 | stratum), data = d, family = "binomial")
maihda_vpc_response(m, seed = 1)
```

`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.

When `autobin` discretises a numeric grouping variable `v`, the adjusted-model and prediction machinery later add an internal factor column named `.maihda_dim_<v>`; the `.maihda_dim_` prefix is therefore reserved. `make_strata()` errors if `data` already holds the `.maihda_dim_<v>` column for a variable it is about to auto-bin, so an existing user column is never silently overwritten (rename it, or pass `autobin = FALSE`).

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 each `type` to the model it is valid on. The VPC and shrinkage views (`"vpc"`, `"obs_vs_shrunken"`, `"predicted"`) use the **null** model. The additive-vs-intersectional views (`"effect_decomp"`, `"prediction_deviation"`) use the **adjusted** model, whose fixed effects carry the dimensions' additive part so the stratum random effect is the pure interaction; with fewer than two dimensions (no adjusted model) they fall back to the null model. Group types (`"group_vpc"`, `"group_components"`, `"group_between_variance"`, `"group_pcv"`, `"group_additive_share"`) use the group comparison when `maihda` was called with a `group`.

Usage

```
## S3 method for class 'maihda_analysis'
plot(
  x,
  type = "all",
  highlight_interactions = FALSE,
  only_flagged = FALSE,
  highlight_by = c("flag", "rope"),
  rope = NULL,
  select = c("order", "deviation"),
  ...
)
```

Arguments

`x`	A `maihda_analysis` object from `maihda`.
`type`	One of the model types ("all", "vpc", "obs_vs_shrunken", "predicted", "upset" (the UpSet-style alternative to "predicted"; forwards `quantity` via `...`), "effect_decomp", "prediction_deviation"), the contextual type ("context_vpc", a stratum-vs-context variance bar; requires `maihda(context = )`), a longitudinal type ("vpc_trajectory", "trajectories", "pcv_trajectory"; requires `decomposition = "longitudinal"`), or a group type ("group_vpc", "group_components", "group_between_variance", "group_pcv", "group_additive_share"). Default "all". For a longitudinal analysis "all" shows the VPC-over-time, the stratum trajectories, and the time-specific PCV.
`highlight_interactions`	Highlight strata with a credibly non-zero intersectional interaction on the BLUP-based views (see `maihda_interactions` and `plot`). `FALSE` (default), `TRUE` (computed from this analysis's adjusted / crossed-dimensions model), a multiple-testing method such as `"BH"`, or a `maihda_interactions` object. The flags are computed once from the correct (adjusted) model and reused across views.
`only_flagged`	Show only the highlighted strata on the `"predicted"` and `"obs_vs_shrunken"` views instead of dimming the rest (see `plot`). `FALSE` (default) keeps every stratum; `TRUE` restricts those views to the highlighted strata (those carrying a credibly non-zero interaction, or – under `highlight_by = "rope"` – those classified ROPE-`"relevant"`) and, when `highlight_interactions` is left `FALSE`, turns the highlight on with this analysis's stored diagnostic. `"effect_decomp"` ignores it (it stays highlighted in context).
`highlight_by`	Which interaction-screen column defines the highlighted strata: `"flag"` (default, the zero-centred `flagged` column) or `"rope"` (the equivalence `decision` column, highlighting the strata classified `"relevant"`). See `plot`. `"rope"` requires `rope` (or a `highlight_interactions` object built with one).
`rope`	Equivalence region forwarded to `maihda_interactions` when computing the screen, so `highlight_by = "rope"` has a decision column to read: a single positive `d` for the symmetric region `c(-d, d)` on the latent (link) scale, or `c(lower, upper)`. `NULL` (default) adds no equivalence classification.

`select`	When the `n_strata` cap drops strata on the `"predicted"` (or longitudinal `"trajectories"`) view, which to keep: `"order"` (default, first n_strata in stratum order) or `"deviation"` (the n_strata furthest from the reference, both tails). See `plot`.
`...`	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` as a point/forest plot of the VPC/ICC by group, as stacked variance-composition bars (between- vs within-stratum share) by group, as bars of the absolute between-stratum (intersectional) variance by group, or as bars of the additive share (PCV) by group. Dispatched via `plot()` on the classed result.

Usage

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

Arguments

`x`	A `maihda_group_comparison` object from `compare_maihda_groups`.
`type`	One of "vpc" (default) for VPC by group with optional bootstrap confidence intervals, "components" for stacked variance proportions (additive / interaction / residual for a crossed-dimensions comparison, between / other / residual otherwise), "between_variance" for the absolute between-stratum variance by group, "pcv" for the two-model additive share (null -> adjusted proportional change in between-stratum variance) by group, or "additive_share" for the crossed-dimensions additive share by group. The VPC is a *share* of the unexplained variance; "between_variance" shows the *magnitude* the ratio cannot convey (two groups with very different VPCs can share a between-stratum variance, and vice versa); "pcv" requires strata defined by at least two dimensions.
`...`	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(cmp, type = "between_variance")
```

`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", "upset", "effect_decomp",
   "prediction_deviation", "context_vpc", "vpc_trajectory", "trajectories"),
  summary_obj = NULL,
  n_strata = 50,
  highlight_interactions = FALSE,
  only_flagged = FALSE,
  highlight_by = c("flag", "rope"),
  rope = NULL,
  select = c("order", "deviation"),
  quantity = c("predicted", "interaction"),
  ...
)
```

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
- "upset": UpSet-style alternative to `"predicted"` – an intersection-size bar, a category matrix encoding each stratum's level on every dimension, and the predicted-value panel, all sharing one column order. Replaces the long intersectional axis labels with the matrix. Binary 0/1 dimensions show as a single present/absent row; multi-level factors get one row per level

- "effect_decomp": Visualizes additive vs intersectional deviation from global mean
- "prediction_deviation": Detailed deviation panels for individuals or strata
- "context_vpc": Stratum vs. context variance bars for a contextual cross-classified fit (`fit_maihda(context = )`); errors otherwise
- "vpc_trajectory": Time-varying VPC/ICC curve for a **longitudinal** fit (`fit_maihda(id =, time =)`); errors otherwise. For a longitudinal model `"vpc"` and `"all"` also route here
- "trajectories": Predicted per-stratum mean trajectories over time (longitudinal fits only)
- "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 the predicted plot. When there are more strata than this, the first `n_strata` (in stratum order) are shown and the plot caption notes how many were omitted. Default is 50. Use NULL for all strata.
`highlight_interactions`	Highlight the strata that carry a credibly non-zero intersectional interaction (from `maihda_interactions`) on the BLUP-based views (`"effect_decomp"`, `"predicted"`, `"obs_vs_shrunken"`); other views ignore it. `FALSE` (default) off; `TRUE` computes the flags with `maihda_interactions()` defaults; or pass a multiple-testing method such as `"BH"` or a `maihda_interactions` object to reuse a specific `conf_level`/ `adjust`. For the pure-interaction reading the model should be the adjusted (or crossed-dimensions) model – e.g. via `plot()` on a `maihda` analysis, which routes these views to the adjusted model automatically. Which column of the screen drives the highlight set is governed by `highlight_by`.
`only_flagged`	Show *only* the flagged strata rather than dimming the rest. `FALSE` (default) keeps every stratum (flagged ones highlighted); `TRUE` restricts the `"predicted"` and `"obs_vs_shrunken"` views to the highlighted strata (those carrying a credibly non-zero interaction, or – under `highlight_by = "rope"` – those classified ROPE-`"relevant"`), so a highlighted stratum is never hidden by the `n_strata` cap. When `TRUE` and `highlight_interactions` is left `FALSE`, the flags are computed with `maihda_interactions` defaults; pass `highlight_interactions` to choose the `conf_level`/`adjust`. A captioned empty panel is returned when no stratum is flagged. It does not apply to `"effect_decomp"` (whose waterfall exists to show each flagged stratum's place in the *full* distribution); that view stays highlighted. Independently of this argument, whenever interactions are highlighted the `n_strata` cap on `"predicted"` becomes flag-aware: every flagged stratum is kept and the remaining slots are filled according to `select`.
`highlight_by`	Which interaction-screen column defines the highlighted strata: `"flag"` (default) uses the zero-centred `flagged` column (credibly non-zero interaction), preserving the historical behaviour; `"rope"` uses the equivalence `decision` column, highlighting the strata classified `"relevant"` (interaction interval entirely outside the region of practical equivalence). `"rope"` requires a screen carrying a `decision` column: either pass `rope`, or supply a `maihda_interactions` object built with `rope`; otherwise it errors.

`rope`	Equivalence region (a "smallest interaction of interest") forwarded to `maihda_interactions` when the screen is computed here (i.e. when `highlight_interactions` is `TRUE` or a `p.adjust` method name). `NULL` (default) adds no equivalence classification; a single positive `d` means the symmetric region `c(-d, d)` on the latent (link) scale, or supply `c(lower, upper)`. Needed for `highlight_by = "rope"` unless the supplied `maihda_interactions` object already carries a `decision` column. Ignored when a precomputed `maihda_interactions` object is passed (its own `rope` is used).
`select`	When the `n_strata` cap must drop strata, which to keep: `"order"` (default; the first n_strata in stratum order, the historical behaviour) or `"deviation"` (the n_strata furthest from the reference line – largest `|predicted - reference|`, so the most extreme strata in *both* directions). Applies to `"predicted"` and, for a longitudinal fit, `"trajectories"` (where it keeps the strata whose trajectories swing furthest from the population curve). Flagged strata are always kept; this governs the fill and the unflagged case. The displayed x-axis stays in stratum order regardless, so `select` changes *which* strata appear, not their left-to-right order.
`quantity`	For `type = "upset"`, which quantity the bottom panel shows: `"predicted"` (default) the stratum's predicted value (fixed + random effect) against the across-strata reference line, or `"interaction"` the stratum random effect (the BLUP) against zero – the deviation from the model's fixed prediction, which is the *pure* intersectional interaction when the dimension main effects are in the model (the adjusted model). Ignored by the other plot types.
`...`	Additional arguments (not currently used).

Value

For a single `type`, a **ggplot2** object that you can extend with the usual `+` grammar (themes, `labs()`, added layers, or a replacement fill/colour scale). Two types return a richer object: `"prediction_deviation"` returns a **patchwork** of two panels (theme every panel at once with `& theme_*()`). `type = "all"` returns a named list of ggplot2 objects.

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")

# Single-type plots are ggplot objects -- restyle them with ggplot2:
plot(model, type = "vpc") +
  ggplot2::theme_classic() +
  ggplot2::labs(title = "Variance partition, restyled")

# Generate all plots (a named list); pick one out to restyle it:
plots <- plot(model)
plots$predicted + ggplot2::theme_bw()
```

`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"),
  allow_new_levels = FALSE,
  ...
)
```

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 a longitudinal (growth-curve) fit a stratum is a *trajectory*, so this returns the per-stratum trajectory parameters (baseline deviation, random intercept and random slope(s)) rather than a single random effect. 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". For a cumulative (ordinal) model the "link" scale is the latent location η and the "response" scale is the *expected category score* $\sum_k kP(Y = k)$ (categories scored 1..K in their declared order). For an aggregated-binomial fit (an lme4 `cbind(success, failure)` or a brms `success | trials(n)`) the "response" scale is the per-trial *probability* on both engines (not the expected success count).
`allow_new_levels`	Logical. By default (`FALSE`) a stratum in `newdata` that the model never saw – whether supplied directly as a `stratum` column or rebuilt from the grouping variables – is an error, for every engine, matching **lme4**'s default. Set `TRUE` to instead predict unseen strata with the stratum random effect dropped (treated as zero), while keeping any *other* random effect the row participates in (e.g. a contextual `(1 | school)` intercept from `fit_maihda(context = )`, or a longitudinal

	growth term) – the same behaviour as **lme4**'s `allow.new.levels`, which zeroes only the unseen level's effect and keeps seen ones. For the usual single-stratum model the stratum is the only random effect, so this is the *population-average* (fixed-effects-only) prediction. This affects `type = "individual"` only: a stratum-level prediction (`type = "strata"`) has no random effect to report for an unseen stratum, so unseen strata remain an error there regardless.
`...`	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_ic` *Print MAIHDA information criteria*

Description

Print MAIHDA information criteria

Usage

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

Arguments

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

Value

No return value, called for side effects.

`print.maihda_interactions` *Print a MAIHDA interaction diagnostic*

Description

Print a MAIHDA interaction diagnostic

Usage

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

Arguments

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

Value

No return value, called for side effects.

`print.maihda_long_pcv` *Print a longitudinal MAIHDA PCV*

Description

Print a longitudinal MAIHDA PCV

Usage

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

Arguments

`x`	A `maihda_long_pcv` object.
`...`	Unused.

Value

The object, invisibly.

`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_stepwise` *Print a stepwise MAIHDA table*

Description

Print a stepwise MAIHDA table

Usage

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

Arguments

`x`	A `maihda_stepwise` object from `stepwise_pcv`.
`...`	Additional arguments (not used).

Value

Invisibly, `x`.

`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.maihda_table` *Print a MAIHDA results table*

Description

Print a MAIHDA results table

Usage

```
## S3 method for class 'maihda_table'
print(x, digits = x$digits, ...)
```

Arguments

`x`	A `maihda_table` object from `maihda_table`.
`digits`	Decimal places (defaults to the value stored on `x`).
`...`	Additional arguments (not used).

Value

Invisibly, `x`.

`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",
  sampling_weights = NULL
)
```

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", "brms", "wemix", or "ordinal"). Default is "lme4"; switches to "wemix" automatically when `sampling_weights` is supplied, and to "ordinal" for an ordinal family or ordered-factor outcome.
`family`	Error distribution and link function. Default is "gaussian".
`sampling_weights`	Optional name of a sampling-weight column for design-weighted stepwise fits; see `fit_maihda`. The weight column joins the complete-case filter so every step uses the same analytic sample.

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.

For a binary outcome the table additionally tracks discriminatory accuracy (Merlo et al. 2016): `AUC` is each model's C-statistic and `Step_AUC` / `Total_AUC` are its *absolute* change (delta-AUC), in contrast to the *proportional* `Step_PCV` / `Total_PCV`. The `MOR` is reported for the logit link (`NA` otherwise) and is a monotone transform of the between-stratum variance already in `Variance`. For a design-weighted fit (`sampling_weights`) the AUC is the design-weighted (population) C-statistic. Reuses `maihda_discriminatory_accuracy` on each step's fitted model, so no additional models are fit. Note that adding a *stratum-defining* dimension (one already encoded by the strata) typically leaves the AUC essentially unchanged: it re-partitions the between-stratum variance (so the PCV and MOR move) but not the per-stratum predicted ranking the rank-based AUC depends on. The AUC trajectory is therefore most informative for individual-level covariates that vary *within* strata.

Value

A data.frame (class `maihda_stepwise`) showing the sequential models, the between-stratum variance at each step, and both the step-specific and total PCV. For a **binary** (binomial/Bernoulli) outcome it also carries the discriminatory-accuracy trajectory: `AUC` (the C-statistic of each step's model – step 0 is the strata-only discriminatory accuracy), `Step_AUC` and `Total_AUC` (the *absolute* change in AUC, delta-AUC, versus the previous step and versus the null), and `MOR` (the Median Odds Ratio, logit link only). These columns are absent for non-binary outcomes.

References

Merlo, J., Wagner, P., Ghith, N., & Leckie, G. (2016). An original stepwise multilevel logistic regression analysis of discriminatory accuracy: the case of neighbourhoods and health. *PLOS ONE*, 11(4), e0153778.

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,
  response_vpc = FALSE,
  seed = NULL,
  ...
)
```

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. For a negative-binomial model (`glmer.nb`) the bootstrap refits via `lme4::refit()`, which holds the dispersion parameter theta fixed at its original estimate, so the interval is conditional on the estimated theta (theta's own sampling uncertainty is not propagated). The `ordinal` (clmm) engine has no simulate/refit machinery, so `bootstrap = TRUE` is rejected there (use `engine = "brms"` for interval estimates).
`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.
`response_vpc`	Logical; for a binomial (lme4) model, also compute the response-scale VPC (`maihda_vpc_response`) and attach it as the `vpc_response` slot. It is estimated by simulation, so it is opt-in (default `FALSE`) and uses `seed` for reproducibility. Ignored for other families/engines.
`seed`	Optional integer seed for the response-scale VPC simulation when `response_vpc = TRUE`.
`...`	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`. For a contextual cross-classified fit this is the *between-stratum* share of all unexplained variance (net of the context)
`variance_components`	Data frame of variance components. For a contextual cross-classified fit (`fit_maihda(context = )`) each context appears as its own `Context: <name>` row

`context`	For a contextual cross-classified fit, the stratum vs. context partition: per-context variances and shares, the contexts' total share (`vpc_context_total`, with an interval when bootstrapped or for brms), and the between-stratum share (`vpc_stratum`); `NULL` otherwise
`discriminatory_accuracy`	For a binomial/Bernoulli outcome, the `maihda_da` object (AUC + MOR) from `maihda_discriminatory_accuracy`; `NULL` otherwise. Also `NULL` for a crossed-dimensions fit (whose single-stratum between-variance the MOR needs is not defined across crossed random effects) and for a contextual cross-classified fit (`fit_maihda(context = )`), where the AUC would be built from predictions that include the context random effects – a mismatch with the stratum-vs-context partition the summary reports
`vpc_response`	The response-scale VPC (`maihda_vpc_response`) when `response_vpc = TRUE` for a single-stratum binomial lme4 model; `NULL` otherwise (including for crossed-dimensions and contextual fits, whose partition the stratum-only simulation does not match)
`stratum_estimates`	Data frame of stratum-specific random effects with labels if available
`fixed_effects`	Fixed effects estimates
`thresholds`	For a cumulative (ordinal) clmm fit, the threshold (cut point) estimates with standard errors – the cumulative model's "intercepts"; NULL otherwise
`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 ($\pi^2/3$ for logistic, $\log(1+1/\mu)$ for Poisson per Stryhn et al. 2006, and $\log(1+1/\mu+1/\theta)$ for the negative binomial per Nakagawa, Johnson & Schielzeth 2017), 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)
```

`theme_maihda` *MAIHDA plot theme*

Description

A **ggplot2** theme that applies the MAIHDA brand identity to plot *chrome* only – navy titles and axis labels and soft slate gridlines. It encodes no data: the colourblind-safe data palettes used by the package's plotting functions are left untouched.

Usage

```
theme_maihda(base_size = 11, base_family = NULL)
```

Arguments

`base_size`	Base font size in points. Default ‘11‘.
`base_family`	Base font family. Defaults to ‘getOption("maihda.font", "")‘, i.e. the graphics device default unless you have opted into a brand font globally. Pass a string to force a family for a single plot.

Details

Built on [ggplot2::theme_minimal()], so it composes with additional ‘+ theme()‘ calls in the usual way (later settings win).

Value

A **ggplot2** theme object that can be added to a plot with ‘+‘.

Fonts

By default the theme uses the graphics device's default font, which is safe on every device (including the PostScript/PDF devices used by ‘R CMD check‘ and many rendering back-ends). To render figures in the brand font (Montserrat) – matching the hex logo – set ‘options(maihda.font = "Montserrat")‘ or pass ‘base_family = "Montserrat"‘, and use a graphics device that can resolve that family (e.g. **ragg** or **showtext**). Forcing a font that the active device cannot find produces "invalid font type" errors, which is why it is opt-in rather than automatic.

Examples

```
library(ggplot2)
ggplot(mtcars, aes(mpg, wt)) +
  geom_point(colour = "#0072B2") +
  labs(title = "MAIHDA brand theme", x = "MPG", y = "Weight") +
  theme_maihda()
```

Index