

Package: mlmoderator (via r-universe)

October 7, 2026

Type Package

Title Probing, Plotting, and Interpreting Multilevel Interaction Effects

Version 0.3.0.9000

Description Provides a workflow for probing, plotting, and checking cross-level interaction effects in two-level mixed-effects models fitted with 'lme4' (Bates et al., 2015) [<doi:10.18637/jss.v067.i01>](https://doi.org/10.18637/jss.v067.i01). Implements simple slopes analysis following Aiken and West (1991, ISBN:9780761907121), Johnson-Neyman intervals following Johnson and Fay (1950) [<doi:10.1007/BF02288864>](https://doi.org/10.1007/BF02288864) and Bauer and Curran (2005) [<doi:10.1207/s15327906mbr4003_5>](https://doi.org/10.1207/s15327906mbr4003_5), and grand- or group-mean centering as described in Enders and Tofighi (2007) [<doi:10.1037/1082-989X.12.2.121>](https://doi.org/10.1037/1082-989X.12.2.121). Tests and intervals use Satterthwaite degrees of freedom via 'lmerTest' (Kuznetsova et al., 2017) [<doi:10.18637/jss.v082.i13>](https://doi.org/10.18637/jss.v082.i13) by default, with Kenward-Roger and between-cluster alternatives. Also provides confidence and new-cluster prediction intervals for simple slopes in random-slope models, contour plots of predicted outcomes over the predictor-by-moderator space, and leave-one-cluster-out influence diagnostics for the interaction.

License MIT + file LICENSE

Encoding UTF-8

Language en-US

LazyData true

Depends R (>= 4.1.0)

Imports lme4, lmerTest (>= 3.1-0), ggplot2, stats, utils, grDevices, rlang

Suggests testthat (>= 3.0.0), dplyr, tibble, knitr, rmarkdown, patchwork, pbkrtest, emmeans, mlmRev

Config/testthat/edition 3

VignetteBuilder knitr

URL <https://github.com/subirhait/mlmoderator>
BugReports <https://github.com/subirhait/mlmoderator/issues>
NeedsCompilation no
Config/roxygen2/version 8.0.0
Author Subir Hait [aut, cre] (ORCID:
 <https://orcid.org/0009-0004-9871-9677>)
Maintainer Subir Hait <haitsubi@msu.edu>
Config/pak/sysreqs cmake make
Repository <https://subirhait.r-universe.dev>
Date/Publication 2026-10-07 04:08:03 UTC
RemoteUrl <https://github.com/subirhait/mlmoderator>
RemoteRef HEAD
RemoteSha 1d5794a74de45b628880c96d33eb44e0c5b5e681

Contents

mlm_center	2
mlm_jn	4
mlm_plot	5
mlm_probe	7
mlm_sensitivity	9
mlm_summary	11
mlm_surface	12
mlm_variance_decomp	14
plot.mlm_jn	16
plot.mlm_sensitivity	17
plot.mlm_variance_decomp	18
school_data	18
Index	20

mlm_center	<i>Center variables for multilevel modeling</i>
------------	---

Description

Performs grand-mean centering, group-mean centering, or both (within-between decomposition) on one or more variables in a data frame. Group-mean centering is the standard preparation for cross-level interaction models.

Usage

```
mlm_center(
  data,
  vars,
  cluster = NULL,
  type = c("grand", "group", "both"),
  suffix_within = "_within",
  suffix_between = "_between"
)
```

Arguments

<code>data</code>	A data frame.
<code>vars</code>	Character vector of variable names to center.
<code>cluster</code>	Character scalar: name of the clustering variable (required when ‘type’ is “group” or “both”).
<code>type</code>	One of “grand”, “group”, or “both”. * “grand”: subtract the overall mean. * “group”: subtract the cluster mean (within-person / within-school centering). * “both”: return both the within-cluster-centered value *and* the cluster mean (between component), appended as new columns.
<code>suffix_within</code>	Suffix appended to within-centered variable names when ‘type = “both”’. Default is “_within”.
<code>suffix_between</code>	Suffix appended to between (cluster mean) variable names when ‘type = “both”’. Default is “_between”.

Value

The input data frame with new centered columns appended. Original columns are not modified.

Examples

```
data(school_data)

# Grand-mean center SES
d1 <- mlm_center(school_data, vars = "ses", type = "grand")
head(d1[, c("ses", "ses_c")])

# Group-mean center SES within schools
d2 <- mlm_center(school_data, vars = "ses", cluster = "school", type = "group")
head(d2[, c("ses", "ses_c")])

# Within-between decomposition
d3 <- mlm_center(school_data, vars = "ses", cluster = "school", type = "both")
head(d3[, c("ses", "ses_within", "ses_between")])
```

mlm_jn

*Johnson—Neyman interval for multilevel two-way interactions***Description**

Computes the Johnson—Neyman (JN) interval: the region(s) of the moderator ('modx') where the simple slope of 'pred' transitions between statistical significance and non-significance. Useful for identifying *exactly* at which moderator values an effect becomes significant.

Usage

```
mlm_jn(
  model,
  pred,
  modx,
  alpha = 0.05,
  modx.range = NULL,
  grid = 200L,
  df_method = c("satterthwaite", "kenward-roger", "between", "residual")
)
```

Arguments

model	An 'lmerMod' object with a two-way interaction between 'pred' and 'modx' in the fixed-effects structure.
pred	Character scalar. Focal predictor name.
modx	Character scalar. Moderator name.
alpha	Significance level. Default '0.05'.
modx.range	Numeric vector of length 2 giving the range over which to evaluate the slope. Defaults to the observed range of 'modx'.
grid	Integer. Number of points at which the slope is evaluated across the moderator range (used for plotting and to bracket the boundaries). Default '200'.
df_method	Denominator degrees of freedom for tests and intervals: * "satterthwaite" (default): Satterthwaite approximation via lmerTest . * "kenward-roger": Kenward-Roger approximation (requires pbkrtest); also uses the Kenward-Roger adjusted covariance, so standard errors can differ slightly. * "between": $J - q - 1$, where J is the number of clusters and q the number of cluster-level fixed effects (Snijders and Bosker, 2012). A simple, conservative choice for cross-level terms. * "residual": $N - p$, the rule used by mlmoderator before 0.3.0. It treats level-1 observations as independent and is anti-conservative for cross-level interactions, especially with few clusters; it is kept only for reproducing earlier results.

Details

A Johnson-Neyman boundary is a moderator value w at which $|t(w)| = t_{1-\alpha/2, df(w)}$, where $t(w) = (\hat{\beta}_1 + \hat{\beta}_3 w) / SE(w)$. With a constant number of degrees of freedom (`df_method = "between"` or `"residual"`) the boundaries are the real roots of a quadratic in w (Bauer and Curran, 2005), and these closed-form roots are returned. With Satterthwaite or Kenward-Roger degrees of freedom, which change with w , the boundaries are found by root-finding (`stats::uniroot()`) on the exact criterion, starting from brackets located on the grid.

Only boundaries inside `modx.range` are returned in `jn_bounds`. The closed-form roots outside that range are reported in `jn_bounds_all` when available, because a boundary outside the observed data is an extrapolation.

Value

An object of class `'mlm_jn'` with components: `* 'jn_bounds'`: numeric vector of moderator values inside `modx.range` where the slope crosses the significance threshold. `'NA'` if none. `* 'jn_bounds_all'`: for constant-df methods, all real roots of the Johnson-Neyman quadratic (possibly outside the data); `'NULL'` otherwise. `* 'slopes_df'`: data frame of slope estimates and significance across the grid. `* 'pred'`, `modx`, `alpha`, `modx.range`, `df_method`.

Examples

```
set.seed(1)
dat <- data.frame(
  y = rnorm(200), x = rnorm(200),
  m = rep(rnorm(20), each = 10),
  grp = factor(rep(1:20, each = 10))
)
dat$y <- dat$y + dat$x * dat$m
mod <- lme4::lmer(y ~ x * m + (1 | grp), data = dat,
  control = lme4::lmerControl(optimizer = "bobyqa"))
jn <- mlm_jn(mod, pred = "x", modx = "m")
print(jn)
```

Description

Creates a `'ggplot2'`-based interaction plot showing predicted values of the outcome across levels of `'pred'`, with separate lines for each selected value of `modx`. Confidence bands and raw data overlay are optional.

Usage

```
mlm_plot(
  model,
  pred,
  modx,
  modx.values = c("mean-sd", "quartiles", "tertiles", "custom"),
  at = NULL,
  interval = TRUE,
  conf.level = 0.95,
  points = FALSE,
  point_alpha = 0.3,
  colors = NULL,
  line_size = 1,
  x_label = NULL,
  y_label = NULL,
  legend_title = NULL,
  df_method = c("satterthwaite", "kenward-roger", "between", "residual"),
  modx.level = c("auto", "cluster", "observation")
)
```

Arguments

model	An 'lmerMod' object with a two-way interaction between 'pred' and 'modx' in the fixed-effects.
pred	Character scalar. Focal predictor (x-axis).
modx	Character scalar. Moderator (separate lines).
modx.values	Strategy for moderator values. Same options as 'mlm_probe()': "mean-sd", "quartiles", "tertiles", "custom".
at	Numeric vector of custom moderator values (used when 'modx.values = "custom"').
interval	Logical. Draw confidence bands? Default 'TRUE'.
conf.level	Confidence level for bands. Default '0.95'.
points	Logical. Overlay raw data points? Default 'FALSE'.
point_alpha	Transparency for raw data points. Default '0.3'.
colors	Character vector of colours for moderator lines. If 'NULL', uses a accessible default palette.
line_size	Line width for predicted lines. Default '1'.
x_label	Label for x-axis. Defaults to 'pred'.
y_label	Label for y-axis. Defaults to response variable name.
legend_title	Label for the legend. Defaults to 'modx'.
df_method	Denominator degrees of freedom for tests and intervals: * "satterthwaite" (default): Satterthwaite approximation via lmerTest . * "kenward-roger": Kenward-Roger approximation (requires pbkrtest); also uses the Kenward-Roger adjusted covariance, so standard errors can differ slightly. * "between": $J - q - 1$,

where J is the number of clusters and q the number of cluster-level fixed effects (Snijders and Bosker, 2012). A simple, conservative choice for cross-level terms. * "residual": $N - p$, the rule used by `mlmoderator` before 0.3.0. It treats level-1 observations as independent and is anti-conservative for cross-level interactions, especially with few clusters; it is kept only for reproducing earlier results.

`modx.level` How moderator values are summarised when choosing probe points. "auto" (default) uses one value per cluster when 'modx' is constant within clusters (a cluster-level moderator) and all observations otherwise; "cluster" and "observation" force either. Ignored when 'at' is supplied.

Value

A 'ggplot' object.

Examples

```
set.seed(1)
dat <- data.frame(
  y = rnorm(200),
  x = rnorm(200),
  m = rep(rnorm(20), each = 10),
  grp = factor(rep(1:20, each = 10))
)
dat$y <- dat$y + dat$x * dat$m
mod <- lme4::lmer(y ~ x * m + (1 | grp), data = dat,
  control = lme4::lmerControl(optimizer = "bobyqa"))
mlm_plot(mod, pred = "x", modx = "m")
mlm_plot(mod, pred = "x", modx = "m", modx.values = "quartiles")
```

mlm_probe

Probe simple slopes from a multilevel interaction

Description

Computes simple slopes of a focal predictor ('pred') at selected values of a moderator ('modx') from a two-level mixed-effects model fitted with [`lme4::lmer()`]. Returns estimates, standard errors, *t*-values, *p*-values, and confidence intervals in a tidy data frame.

Usage

```
mlm_probe(
  model,
  pred,
  modx,
  modx.values = c("mean-sd", "quartiles", "tertiles", "custom"),
  at = NULL,
```

```

conf.level = 0.95,
df_method = c("satterthwaite", "kenward-roger", "between", "residual"),
modx.level = c("auto", "cluster", "observation")
)

```

Arguments

<code>model</code>	An ‘lmerMod’ object containing a two-way interaction between ‘pred’ and ‘modx’ in the fixed-effects structure.
<code>pred</code>	Character scalar. Name of the focal predictor variable.
<code>modx</code>	Character scalar. Name of the moderator variable.
<code>modx.values</code>	Strategy for selecting moderator values. One of: * “mean-sd” (default): mean — 1 SD, mean, mean + 1 SD. * “quartiles”: 25th, 50th, 75th percentiles. * “tertiles”: 33rd and 67th percentiles. * “custom”: use values supplied via ‘at’.
<code>at</code>	Numeric vector of custom moderator values. Used when ‘modx.values = "custom"’, or to override any strategy.
<code>conf.level</code>	Confidence level for intervals. Default ‘0.95’.
<code>df_method</code>	Denominator degrees of freedom for tests and intervals: * “satterthwaite” (default): Satterthwaite approximation via lmerTest . * “kenward-roger”: Kenward-Roger approximation (requires pbkrtest); also uses the Kenward-Roger adjusted covariance, so standard errors can differ slightly. * “between”: $J - q - 1$, where J is the number of clusters and q the number of cluster-level fixed effects (Snijders and Bosker, 2012). A simple, conservative choice for cross-level terms. * “residual”: $N - p$, the rule used by mlmoderator before 0.3.0. It treats level-1 observations as independent and is anti-conservative for cross-level interactions, especially with few clusters; it is kept only for reproducing earlier results.
<code>modx.level</code>	How moderator values are summarised when choosing probe points. “auto” (default) uses one value per cluster when ‘modx’ is constant within clusters (a cluster-level moderator) and all observations otherwise; “cluster” and “observation” force either. Ignored when ‘at’ is supplied.

Value

An object of class ‘mlm_probe’ (a list) with components: * ‘slopes’: a data frame with columns ‘modx_value’, ‘slope’, ‘se’, ‘t’, ‘df’, ‘p’, ‘ci_lower’, ‘ci_upper’. * ‘df_method’: the degrees-of-freedom method used. * ‘pred’, ‘modx’: names of the predictor and moderator. * ‘modx.values’: the strategy used. * ‘conf.level’: the confidence level. * ‘model’: the original model (stored for downstream use).

References

- Satterthwaite, F. E. (1946). An approximate distribution of estimates of variance components. **Biometrics Bulletin**, 2(6), 110–114.
- Snijders, T. A. B., & Bosker, R. J. (2012). **Multilevel analysis** (2nd ed.). Sage.
- Kuznetsova, A., Brockhoff, P. B., & Christensen, R. H. B. (2017). lmerTest package: Tests in linear mixed effects models. **Journal of Statistical Software**, 82(13), 1–26. doi:10.18637/jss.v082.i13

Examples

```
set.seed(1)
dat <- data.frame(
  y = rnorm(200), x = rnorm(200),
  m = rep(rnorm(20), each = 10),
  grp = factor(rep(1:20, each = 10))
)
dat$y <- dat$y + dat$x * dat$m
mod <- lme4::lmer(y ~ x * m + (1 | grp), data = dat,
  control = lme4::lmerControl(optimizer = "bobyqa"))
mlm_probe(mod, pred = "x", modx = "m")
mlm_probe(mod, pred = "x", modx = "m", modx.values = "quartiles")
mlm_probe(mod, pred = "x", modx = "m", modx.values = "custom", at = c(-1, 0, 1))
```

mlm_sensitivity	<i>Leave-one-cluster-out influence diagnostics for a cross-level interaction</i>
-----------------	--

Description

Refits the model once for each cluster, leaving that cluster out, and records how the interaction coefficient changes. This answers a specific question: **is the interaction driven by a small number of clusters?**

Usage

```
mlm_sensitivity(
  model,
  pred,
  modx,
  alpha = 0.05,
  df_method = c("satterthwaite", "kenward-roger", "between", "residual"),
  verbose = FALSE,
  icc_range,
  icc_grid,
  loco = TRUE
)
```

Arguments

model	An ‘lmerMod’ object with a two-way interaction between ‘pred’ and ‘modx’.
pred	Character scalar. Focal predictor name.
modx	Character scalar. Moderator name.
alpha	Significance level used to record whether each refit keeps the interaction significant. Default ‘0.05’.

df_method	Denominator degrees of freedom for tests and intervals: * "satterthwaite" (default): Satterthwaite approximation via lmerTest . * "kenward-roger": Kenward-Roger approximation (requires pbkrtest); also uses the Kenward-Roger adjusted covariance, so standard errors can differ slightly. * "between": $J - q - 1$, where J is the number of clusters and q the number of cluster-level fixed effects (Snijders and Bosker, 2012). A simple, conservative choice for cross-level terms. * "residual": $N - p$, the rule used by mlmoderator before 0.3.0. It treats level-1 observations as independent and is anti-conservative for cross-level interactions, especially with few clusters; it is kept only for reproducing earlier results.
verbose	Logical. Print progress during refitting? Default 'FALSE'.
icc_range, icc_grid, loco	Defunct arguments from versions before 0.3.0 (see Details). Supplying 'icc_range' or 'icc_grid' gives a warning and is otherwise ignored; 'loco = FALSE' is no longer supported.

Details

Each refit uses [stats::update()] on the data the model was fitted to (via [lme4::getData()]), so it keeps the original formula, estimation method (REML or ML), weights, and control settings. Influence is measured by a DFBETA-type statistic,

$$\text{DFBETA}_j = (\hat{\beta}_{3(-j)} - \hat{\beta}_3) / SE(\hat{\beta}_3),$$

the change in the interaction when cluster j is omitted, in units of the full-data standard error. Clusters with $|\text{DFBETA}_j| > 2/\sqrt{J}$ are flagged, following the usual size-adjusted cutoff for DFBETAS (Belsley, Kuh, and Welsch, 1980). The cutoff is a screening heuristic, not a test.

****Changes in 0.3.0.**** Earlier versions also reported an "ICC-shift" analysis that rescaled the interaction's standard error by a design-effect ratio for hypothetical intraclass correlations. That formula applies to means under a random-intercept model, not to a cross-level interaction in a random-slope model, and the accompanying adjusted Johnson-Neyman boundary was not correct, so the analysis has been removed. The earlier leave-one-cluster-out refits used maximum likelihood even when the original model used REML, and refitted from the model frame, which failed silently for formulas with transformed variables; both are fixed.

****Scope.**** These are influence diagnostics. They do not address unmeasured confounding of the interaction.

Value

An object of class 'mlm_sensitivity' with components: * 'observed': list with the full-data interaction estimate, SE, df, p-value, and number of clusters. * 'loco': data frame with one row per cluster: 'cluster', 'n_obs', 'b_int', 'se_int', 'df', 'p_int', 'b_change', 'dfbeta', 'influential', 'same_sign', 'sig', and 'error' (the error message when a refit failed, otherwise 'NA'). * 'cutoff': the DFBETA flagging cutoff $2/\sqrt{J}$. * 'summary': list with the number of failed refits, the proportion of successful refits keeping the sign and significance of the interaction, and the range of refitted estimates. * Metadata: 'pred', 'modx', 'alpha', 'df_method', 'grp_name', 'int_term'.

References

Belsley, D. A., Kuh, E., & Welsch, R. E. (1980). *Regression diagnostics*. Wiley.

Examples

```

set.seed(42)
n_j <- 20; n_i <- 10
dat_small <- data.frame(
  y = rnorm(n_j * n_i),
  x = rnorm(n_j * n_i),
  m = rep(rnorm(n_j), each = n_i),
  grp = factor(rep(seq_len(n_j), each = n_i))
)
dat_small$y <- dat_small$y + 0.5 * dat_small$x * dat_small$m
mod_small <- lme4::lmer(y ~ x * m + (1 | grp), data = dat_small)

sens <- mlm_sensitivity(mod_small, pred = "x", modx = "m",
  df_method = "between")

sens
plot(sens)

```

mlm_summary

*Summary table for a multilevel moderation effect***Description**

Returns a consolidated summary of the moderation effect: the focal interaction coefficient, simple slopes at selected moderator values, and (optionally) the Johnson—Neyman interval. Designed for quick reporting and results sections.

Usage

```

mlm_summary(
  model,
  pred,
  modx,
  modx.values = c("mean-sd", "quartiles", "tertiles", "custom"),
  at = NULL,
  conf.level = 0.95,
  jn = TRUE,
  alpha = 0.05,
  df_method = c("satterthwaite", "kenward-roger", "between", "residual"),
  modx.level = c("auto", "cluster", "observation")
)

```

Arguments

model	An ‘lmerMod’ object with a two-way interaction between ‘pred’ and ‘modx’.
pred	Character scalar. Focal predictor name.
modx	Character scalar. Moderator name.

<code>modx.values</code>	Moderator value strategy. See <code>'mlm_probe()'</code> .
<code>at</code>	Optional numeric vector of custom moderator values.
<code>conf.level</code>	Confidence level. Default <code>'0.95'</code> .
<code>jn</code>	Logical. Include Johnson-Neyman region? Default <code>'TRUE'</code> .
<code>alpha</code>	Alpha for JN interval. Default <code>'0.05'</code> .
<code>df_method</code>	Denominator degrees of freedom for tests and intervals: * <code>"satterthwaite"</code> (default): Satterthwaite approximation via lmerTest . * <code>"kenward-roger"</code> : Kenward-Roger approximation (requires pbkrtest); also uses the Kenward-Roger adjusted covariance, so standard errors can differ slightly. * <code>"between"</code> : $J - q - 1$, where J is the number of clusters and q the number of cluster-level fixed effects (Snijders and Bosker, 2012). A simple, conservative choice for cross-level terms. * <code>"residual"</code> : $N - p$, the rule used by <code>mlmoderator</code> before 0.3.0. It treats level-1 observations as independent and is anti-conservative for cross-level interactions, especially with few clusters; it is kept only for reproducing earlier results.
<code>modx.level</code>	How moderator values are summarised when choosing probe points. <code>"auto"</code> (default) uses one value per cluster when <code>'modx'</code> is constant within clusters (a cluster-level moderator) and all observations otherwise; <code>"cluster"</code> and <code>"observation"</code> force either. Ignored when <code>'at'</code> is supplied.

Value

An object of class `'mlm_summary'` (a list) with components: * `'interaction'`: one-row data frame for the interaction term. * `'simple_slopes'`: data frame from `'mlm_probe()'`. * `'jn'`: output of `'mlm_jn()'` (or `'NULL'` if `'jn = FALSE'`). * Other metadata.

Examples

```
set.seed(1)
dat <- data.frame(
  y = rnorm(200), x = rnorm(200),
  m = rep(rnorm(20), each = 10),
  grp = factor(rep(1:20, each = 10))
)
dat$y <- dat$y + dat$x * dat$m
mod <- lme4::lmer(y ~ x * m + (1 | grp), data = dat,
  control = lme4::lmerControl(optimizer = "bobyqa"))
mlm_summary(mod, pred = "x", modx = "m")
```

mlm_surface

Contour plot of predicted outcomes over the predictor x moderator space

Description

Plots iso-outcome contour lines of $\hat{Y}(x, w)$ over the full joint space of ‘pred’ (x-axis) and ‘modx’ (y-axis). This is the most direct geometric representation of a two-way interaction:

Usage

```
mlm_surface(
  model,
  pred,
  modx,
  grid = 80L,
  n_contours = 10L,
  fill = TRUE,
  probe_lines = TRUE,
  x_label = NULL,
  y_label = NULL,
  legend_title = NULL
)
```

Arguments

model	An ‘lmerMod’ object with a two-way interaction between ‘pred’ and ‘modx’ in the fixed effects.
pred	Character scalar. Focal predictor (x-axis).
modx	Character scalar. Moderator (y-axis).
grid	Integer. Grid resolution (points per axis). Default ‘80’.
n_contours	Integer. Number of contour levels to draw. Default ‘10’.
fill	Logical. Fill contour bands with colour? Default ‘TRUE’.
probe_lines	Logical. Overlay horizontal lines at mean – 1 SD of ‘modx’? Default ‘TRUE’.
x_label	x-axis label. Defaults to ‘pred’.
y_label	y-axis label. Defaults to ‘modx’.
legend_title	Legend title. Defaults to the outcome variable name.

Details

*****No interaction**:** contour lines are perfectly straight and parallel — the effect of ‘pred’ does not depend on ‘modx’. *****Positive interaction**:** contour lines fan outward (rotate clockwise) — higher ‘modx’ steepens the ‘pred’ slope. *****Negative interaction**:** contour lines fan inward (rotate counter-clockwise).

The degree of non-parallelism among contours is a direct visual index of interaction strength: the larger β_3 , the more the lines rotate.

An optional overlay draws the three standard simple-slope evaluation lines (mean – 1 SD of ‘modx’) as horizontal reference lines, connecting the plot to ‘mlm_probe()’ output.

Predicted values are computed from fixed effects only (‘re.form = NA’), with all covariates held at their means or reference levels. The surface therefore represents the population-average predicted outcome, not any specific cluster.

****Reading the plot:**** Pick any contour line. Its slope in the ‘pred’ direction tells you how fast the outcome changes with ‘pred’ at that ‘modx’ value. If the contour slopes steeply up-right, ‘pred’ has a strong positive effect there. If contours become more horizontal as ‘modx’ increases, the ‘pred’ effect is weakening. If they rotate from positive to flat to negative, you have a sign-changing interaction — and the ‘modx’ value where they are perfectly horizontal is the Johnson-Neyman boundary.

Value

A ‘ggplot’ object.

Examples

```
set.seed(1)
dat <- data.frame(
  y = rnorm(200), x = rnorm(200),
  m = rep(rnorm(20), each = 10),
  grp = factor(rep(1:20, each = 10))
)
dat$y <- dat$y + dat$x * dat$m
mod <- lme4::lmer(y ~ x * m + (1 | grp), data = dat,
  control = lme4::lmerControl(optimizer = "bobyqa"))
mlm_surface(mod, pred = "x", modx = "m")
mlm_surface(mod, pred = "x", modx = "m", fill = FALSE, n_contours = 15)
```

mlm_variance_decomp	<i>Confidence and prediction intervals for simple slopes in a random-slope model</i>
---------------------	--

Description

In a model with a random slope for ‘pred’, two different questions can be asked about the slope at a given moderator value w :

Usage

```
mlm_variance_decomp(
  model,
  pred,
  modx,
  modx.values = c("mean-sd", "quartiles", "tertiles", "custom"),
  at = NULL,
  conf.level = 0.95,
  df_method = c("satterthwaite", "kenward-roger", "between", "residual"),
  modx.level = c("auto", "cluster", "observation")
)
```

Arguments

<code>model</code>	An ‘lmerMod’ object with a random slope for ‘pred’ and a two-way interaction between ‘pred’ and ‘modx’.
<code>pred</code>	Character scalar. Focal predictor name.
<code>modx</code>	Character scalar. Moderator name.
<code>modx.values</code>	Strategy for moderator values. See ‘mlm_probe()’.
<code>at</code>	Optional numeric vector of custom moderator values.
<code>conf.level</code>	Level for the confidence and prediction intervals. Default ‘0.95’.
<code>df.method</code>	Denominator degrees of freedom for tests and intervals: * “satterthwaite” (default): Satterthwaite approximation via lmerTest . * “kenward-roger”: Kenward-Roger approximation (requires pbkrtest); also uses the Kenward-Roger adjusted covariance, so standard errors can differ slightly. * “between”: $J - q - 1$, where J is the number of clusters and q the number of cluster-level fixed effects (Snijders and Bosker, 2012). A simple, conservative choice for cross-level terms. * “residual”: $N - p$, the rule used by mlmoderator before 0.3.0. It treats level-1 observations as independent and is anti-conservative for cross-level interactions, especially with few clusters; it is kept only for reproducing earlier results.
<code>modx.level</code>	How moderator values are summarised when choosing probe points. “auto” (default) uses one value per cluster when ‘modx’ is constant within clusters (a cluster-level moderator) and all observations otherwise; “cluster” and “observation” force either. Ignored when ‘at’ is supplied.

Details

1. ****How precisely is the average slope known?*** The confidence interval for $\beta_1 + \beta_3 w$ reflects only estimation uncertainty in the fixed effects (as in [mlm_probe()]).
2. ****What slope should be expected in a new cluster?*** The prediction interval adds the residual random-slope variance τ_{11} :

$$\hat{\beta}_1 + \hat{\beta}_3 w \pm t_{df} \sqrt{SE^2(w) + \hat{\tau}_{11}}.$$

Because ‘modx’ is in the model, τ_{11} is the slope variance that remains *after* the moderator: a significant interaction does not mean the moderator explains the between-cluster slope differences, and a large τ_{11} relative to the average slope means individual clusters can have slopes far from it.

The prediction interval treats $\hat{\tau}_{11}$ as known and assumes normally distributed random slopes, so it is approximate and too narrow when the number of clusters is small. The random slope is taken from the grouping factor whose random-effects terms include ‘pred’.

Versions before 0.3.0 reported ‘pct_random’, the ratio $\tau_{11}/(\tau_{11} + SE^2)$. It mixed a population variance with a sampling variance, so it grew with sample size without any change in the data-generating process; it has been removed.

Value

An object of class ‘mlm_variance_decomp’ (a list) with: * ‘decomp’: data frame with columns ‘modx_value’, ‘slope’, ‘se_fixed’ (SE of the average slope), ‘tau11’ (random-slope variance), ‘tau11_sd’ (its square root), ‘se_total’ ($\sqrt{SE^2 + \tau_{11}}$), ‘df’, ‘ci_lower’, ‘ci_upper’ (confidence interval for

the average slope), and 'pi_lower', 'pi_upper' (prediction interval for a new cluster). * 'tau11', 'tau11_sd': the random-slope variance and SD. * 'has_random_slope': logical; does the model include a random slope for 'pred'? * Metadata: 'pred', 'modx', 'conf.level', 'df_method', 'grp_name'.

Examples

```
set.seed(1)
dat <- data.frame(
  y = rnorm(200), x = rnorm(200),
  m = rep(rnorm(20), each = 10),
  grp = factor(rep(1:20, each = 10))
)
dat$y <- dat$y + dat$x * dat$m
mod <- lme4::lmer(y ~ x * m + (1 + x | grp), data = dat,
  control = lme4::lmerControl(optimizer = "bobyqa"))
vd <- mlm_variance_decomp(mod, pred = "x", modx = "m")
print(vd)
```

plot.mlm_jn

Plot a Johnson-Neyman interval from a multilevel model

Description

Creates a ggplot2 figure showing the simple slope of 'pred' across the full range of 'modx', with shading indicating regions of significance. A vertical dashed line marks each Johnson-Neyman boundary.

Usage

```
## S3 method for class 'mlm_jn'
plot(
  x,
  x_label = NULL,
  y_label = NULL,
  sig_color = "#2166AC",
  nonsig_color = "#D6604D",
  ...
)
```

Arguments

x	An 'mlm_jn' object from 'mlm_jn()'.
x_label	Label for the x-axis. Defaults to the moderator name.
y_label	Label for the y-axis. Defaults to "Simple slope of pred".
sig_color	Fill colour for the significant region. Default "#2166AC".
nonsig_color	Fill colour for the non-significant region. Default "#D6604D".
...	Ignored.

Value

A ‘ggplot’ object.

Examples

```
set.seed(1)
dat <- data.frame(
  y = rnorm(200), x = rnorm(200),
  m = rep(rnorm(20), each = 10),
  grp = factor(rep(1:20, each = 10))
)
dat$y <- dat$y + dat$x * dat$m
mod <- lme4::lmer(y ~ x * m + (1 | grp), data = dat,
  control = lme4::lmerControl(optimizer = "bobyqa"))
jn <- mlm_jn(mod, pred = "x", modx = "m")
plot(jn)
```

plot.mlm_sensitivity *Plot leave-one-cluster-out influence for a cross-level interaction*

Description

Plots the DFBETA of each cluster, ordered by size, with the flagging cutoff $\pm 2/\sqrt{J}$ as dashed lines.

Usage

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

Arguments

x An ‘mlm_sensitivity’ object.

... Ignored.

Value

A ggplot object.

```
plot.mlm_variance_decomp
```

Plot the variance decomposition of simple slopes

Description

Shows the simple slope at each moderator value as a point with two interval layers: an inner confidence interval (fixed-effect uncertainty only) and an outer prediction interval (fixed + random-slope variance). The gap between the two intervals is the contribution of random-slope heterogeneity.

Usage

```
## S3 method for class 'mlm_variance_decomp'
plot(x, x_label = NULL, y_label = NULL, ...)
```

Arguments

<code>x</code>	An 'mlm_variance_decomp' object.
<code>x_label</code>	x-axis label. Defaults to moderator name.
<code>y_label</code>	y-axis label. Defaults to "Simple slope of pred".
<code>...</code>	Ignored.

Value

A 'ggplot' object.

```
school_data
```

Simulated school achievement dataset

Description

A simulated two-level dataset with students nested within schools, designed to illustrate multilevel moderation analysis. The true data-generating model includes a cross-level interaction between student socioeconomic status and school climate.

Usage

```
school_data
```

Format

A data frame with 3,000 rows and 6 variables:

school A factor indicating the school identifier (1–100).

student An integer indicating the student identifier (1–3000).

math A numeric mathematics achievement score.

ses A numeric student socioeconomic status variable (standardized).

climate A numeric school climate rating (standardized level-2 variable).

gender A factor indicating student gender with levels "female" and "male".

Details

The data were generated from the model

$$math_{ij} = 50 + 1.5 ses_{ij} + 0.8 climate_j + 0.5 ses_{ij}climate_j + u_{0j} + u_{1j}ses_{ij} + e_{ij}.$$

The level-2 random effects were generated as $u_{0j} \sim N(0, 9)$ and $u_{1j} \sim N(0, 0.25)$, and the level-1 residuals were generated as $e_{ij} \sim N(0, 25)$.

Examples

```
data(school_data)
head(school_data)
str(school_data)
```

```
library(lme4)
mod <- lmer(math ~ ses * climate + gender + (1 + ses | school),
            data = school_data)
mlm_probe(mod, pred = "ses", modx = "climate")
```

Index

* datasets

school_data, [18](#)

mlm_center, [2](#)

mlm_jn, [4](#)

mlm_plot, [5](#)

mlm_probe, [7](#)

mlm_sensitivity, [9](#)

mlm_summary, [11](#)

mlm_surface, [12](#)

mlm_variance_decomp, [14](#)

plot.mlm_jn, [16](#)

plot.mlm_sensitivity, [17](#)

plot.mlm_variance_decomp, [18](#)

school_data, [18](#)