

Package ‘hlmLab’

September 28, 2026

Type Package

Title Hierarchical Linear Modeling with Visualization and Decomposition

Version 0.2.0

Description Provides functions for visualization and decomposition in hierarchical linear models (HLM) for applications in education, psychology, and the social sciences. Includes variance decomposition for two-level and three-level data structures following Snijders and Bosker (2012, ISBN:9781849202015), intraclass correlation (ICC) estimation and design effect computation as described in Shrout and Fleiss (1979) <doi:10.1037/0033-2909.86.2.420>, and contextual effect decomposition via the Mundlak (1978) <doi:10.2307/1913646> specification distinguishing within- and between-cluster components, with the uncertainty of the contextual contrast obtained from the full fixed-effect covariance matrix. Teaching displays cover simulated intraclass correlations, partial pooling of cluster means, random-slope heterogeneity, and cross-level interaction with an observed Level-2 moderator, following Hofmann and Gavin (1998) <doi:10.1177/014920639802400504> and Hamaker and Muthen (2020) <doi:10.1037/met0000239>. Multilevel models are estimated using 'lme4' (Bates et al., 2015 <doi:10.18637/jss.v067.i01>). An optional 'Shiny' application enables interactive exploration of model components and parameter variation. The implementation follows the multilevel modeling framework of Raudenbush and Bryk (2002, ISBN:9780761919049).

License MIT + file LICENSE

Encoding UTF-8

Language en-US

Depends R (>= 4.1.0)

Imports dplyr, ggplot2 (>= 3.4.0), grDevices, grid, lme4, scales, stats

Suggests shiny, spelling, testthat (>= 3.0.0)

Config/testthat/edition 3

URL <https://github.com/subirhait/hlmLab>

BugReports <https://github.com/subirhait/hlmLab/issues>
Config/roxygen2/version 8.0.0
NeedsCompilation no
Author Subir Hait [aut, cre] (ORCID: <<https://orcid.org/0009-0004-9871-9677>>)
Maintainer Subir Hait <haitsubi@msu.edu>
Repository CRAN
Date/Publication 2026-09-28 04:50:02 UTC

Contents

hlm_context	2
hlm_context_plot	4
hlm_cross_level_plot	5
hlm_decompose	6
hlm_decompose_long	7
hlm_icc	8
hlm_icc_demo	10
hlm_icc_plot	11
hlm_random_slope_plot	12
hlm_shrinkage_plot	14
hlm_xint_geom	15
plot.hlm_context	16
plot.hlm_decompose	16
Index	18

hlm_context	<i>Contextual effect for a Level-1 predictor (Mundlak decomposition)</i>
-------------	--

Description

Given a multilevel model containing a within-cluster centered predictor and its cluster-mean counterpart (Mundlak specification), this function extracts the within- and between-cluster effects and computes the contextual effect (between minus within) together with its standard error and a confidence interval.

Usage

```
hlm_context(model, x_within, x_between, level = 0.95, check_centering = TRUE)
```

Arguments

model	A fitted lmerMod model.
x_within	Name of the within-cluster centered predictor (string).
x_between	Name of the cluster-mean predictor (string).
level	Confidence level for the reported intervals. Defaults to 0.95.
check_centering	Logical; if TRUE (the default), warn when x_within does not appear to be group-mean centered, because the difference between the two coefficients is then not the contextual effect.

Details

The contextual effect is a linear contrast of two fixed-effect coefficients, so its sampling variance is

$$Var(\beta_B - \beta_W) = Var(\beta_B) + Var(\beta_W) - 2Cov(\beta_B, \beta_W).$$

Since version 0.2.0 the covariance term is taken from the fixed-effect covariance matrix returned by `stats::vcov()` rather than assumed to be zero. In a random-intercept model with an exactly group-mean centered `x_within`, that covariance is zero by construction, so results are unchanged; it is generally nonzero when the model contains a random slope, when `x_within` is centered in some other way, or when the Mundlak specification uses the raw (uncentered) predictor.

Intervals use a normal approximation, which is the usual convention for `lme4` fixed effects because no denominator degrees of freedom are supplied.

Value

An object of class `hlm_context`, which is a tibble with:

effect_type	Within, between, or contextual.
estimate	Estimate of the effect.
se	Standard error; exact for the contextual contrast.
conf_low	Lower confidence limit.
conf_high	Upper confidence limit.

The covariance of the two coefficients is stored in the attribute `"cov_within_between"` and the confidence level in `"level"`.

References

Mundlak, Y. (1978). On the pooling of time series and cross section data. *Econometrica*, 46, 69-85.
[doi:10.2307/1913646](https://doi.org/10.2307/1913646)

Examples

```
# Build a small toy dataset (no external data needed)
set.seed(1)
n_schools <- 8
n_students <- 12
school_id <- rep(seq_len(n_schools), each = n_students)
SES_raw <- rnorm(n_schools * n_students)
SES_mean <- rep(tapply(SES_raw, school_id, mean), each = n_students)
SES_c <- SES_raw - SES_mean
math_score <- 50 + 2 * SES_c + 3 * SES_mean +
  rep(rnorm(n_schools, sd = 2), each = n_students) +
  rnorm(n_schools * n_students)
toy <- data.frame(math_score = math_score, SES_c = SES_c,
  SES_mean = SES_mean, school_id = school_id)

m <- lme4::lmer(math_score ~ SES_c + SES_mean + (1 | school_id), data = toy)
ctx <- hlm_context(m, x_within = "SES_c", x_between = "SES_mean")
ctx
hlm_context_plot(ctx)
```

hlm_context_plot

*Convenience wrapper to plot contextual effects***Description**

Convenience wrapper to plot contextual effects

Usage

```
hlm_context_plot(object)
```

Arguments

object An object of class hlm_context.

Value

A ggplot object.

Examples

```
set.seed(1)
school_id <- rep(seq_len(8), each = 12)
SES_raw <- rnorm(96)
SES_mean <- rep(tapply(SES_raw, school_id, mean), each = 12)
SES_c <- SES_raw - SES_mean
math_score <- 50 + 2 * SES_c + 3 * SES_mean +
  rep(rnorm(8, sd = 2), each = 12) + rnorm(96)
toy <- data.frame(math_score = math_score, SES_c = SES_c,
```

```

      SES_mean = SES_mean, school_id = school_id)
m <- lme4::lmer(math_score ~ SES_c + SES_mean + (1 | school_id), data = toy)
hlm_context_plot(hlm_context(m, "SES_c", "SES_mean"))

```

hlm_cross_level_plot *Cross-level interaction with an observed Level-2 moderator*

Description

Plots the fitted Level-1 association at selected values of an observed Level-2 moderator. Unlike [hlm_random_slope_plot](#), which shows unexplained slope heterogeneity, this display shows moderation by a measured cluster characteristic, which is what the term cross-level interaction denotes.

Usage

```

hlm_cross_level_plot(
  model,
  x_within,
  moderator,
  at = NULL,
  labels = NULL,
  n_points = 40,
  x_range = NULL,
  title = NULL,
  subtitle = NULL
)

```

Arguments

model	A fitted lmerMod model that includes a fixed-effect interaction between x_within and moderator.
x_within	Name of the Level-1 predictor (string).
moderator	Name of the Level-2 moderator (string). May be numeric or a factor.
at	Optional values of the moderator at which lines are drawn. For a numeric moderator the default is the mean and one standard deviation on either side; for a factor the default is every level.
labels	Optional character labels for the moderator values, used in the legend.
n_points	Number of points along the x-axis. Defaults to 40.
x_range	Optional length-2 numeric vector giving the plotting range of x_within. Defaults to the 0.02 and 0.98 quantiles of the observed predictor.
title, subtitle	Optional plot title and subtitle. NULL uses informative defaults; NA omits the element.

Details

Fitted values are obtained from `predict()` with the random effects excluded, so the lines describe the average cluster at each moderator value. Remaining numeric predictors are held at their sample means and remaining categorical predictors at their most frequent level. Variables must appear in the model formula under their own names; inline transformations such as `log(x)` are not supported.

Value

A ggplot object. The plotted line data are attached as the attribute "data".

See Also

[hlm_random_slope_plot](#)

Examples

```
set.seed(7)
n_schools <- 20
n_students <- 20
school_id <- rep(seq_len(n_schools), each = n_students)
school_resources <- rep(rnorm(n_schools), each = n_students)
SES_c <- rnorm(n_schools * n_students)
math_score <- 50 + 2 * SES_c + 1.5 * school_resources +
  0.8 * SES_c * school_resources +
  rep(rnorm(n_schools, sd = 1), each = n_students) +
  rnorm(n_schools * n_students)
toy <- data.frame(math_score = math_score, SES_c = SES_c,
                  school_resources = school_resources,
                  school_id = school_id)

m <- lme4::lmer(
  math_score ~ SES_c * school_resources + (1 | school_id),
  data = toy
)
hlm_cross_level_plot(m, x_within = "SES_c",
                    moderator = "school_resources")
```

hlm_decompose

Multilevel variance decomposition (within/between/longitudinal)

Description

Decomposes a continuous variable into between-cluster, and within-cluster components (2-level), or into between-cluster (B), between-person (P), and within-person (W) components for longitudinal 3-level data.

Usage

```
hlm_decompose(data, var, cluster, id = NULL, time = NULL)
```

Arguments

data	A data frame.
var	Name of the focal variable (string).
cluster	Name of the cluster ID variable (e.g., school/classroom).
id	Optional person ID variable for longitudinal data.
time	Optional time variable (not required for the algebra, but noted in the output for clarity).

Details

This is primarily a teaching tool: it shows how total variance is partitioned across levels.

Value

An object of class `hlm_decompose` with components:

data	Original data with added decomposition columns.
summary	A tibble summarizing variance components and shares.

Examples

```
# 2-level example (students in schools)
set.seed(2)
toy <- data.frame(
  math_score = rnorm(80, mean = 50, sd = 10),
  school_id = rep(letters[1:8], each = 10)
)
hlm_decompose(toy, var = "math_score", cluster = "school_id")

# 3-level longitudinal example (waves within students within schools)
toy3 <- data.frame(
  math_score = rnorm(120, 50, 10),
  school_id = rep(letters[1:4], each = 30),
  student_id = rep(seq_len(24), each = 5)
)
hlm_decompose(toy3, var = "math_score", cluster = "school_id",
  id = "student_id", time = NULL)
```

<code>hlm_decompose_long</code>	<i>Longitudinal three-level variance decomposition (B-P-W)</i>
---------------------------------	--

Description

Convenience wrapper around `hlm_decompose()` for 3-level data: clusters (e.g., schools), persons within clusters, and repeated measures within persons over time.

Usage

```
hlm_decompose_long(data, var, cluster, id, time = NULL)
```

Arguments

data	A data frame.
var	Name of the focal variable (string).
cluster	Cluster ID (e.g., school/classroom) (string).
id	Person ID within cluster (string).
time	Optional time variable (string); stored in the output for reference but not required for the algebra.

Value

An object of class `hlm_decompose` with between-cluster (B), between-person (P), and within-person (W) variance components and shares.

Examples

```
set.seed(4)
toy_long <- data.frame(
  math_score = rnorm(120, 50, 10),
  school_id = rep(letters[1:4], each = 30),
  student_id = rep(seq_len(24), each = 5),
  wave = rep(seq_len(5), times = 24)
)
hlm_decompose_long(data = toy_long,
  var = "math_score",
  cluster = "school_id",
  id = "student_id",
  time = "wave")
```

hlm_icc	<i>Intraclass correlation and design effect from a random-intercept model</i>
---------	---

Description

Computes the intraclass correlation (ICC) and, optionally, a design effect from a random-intercept multilevel model fitted with `lme4::lmer()`.

Usage

```
hlm_icc(model, cluster_size = NULL)
```

Arguments

<code>model</code>	A fitted <code>lmerMod</code> object with at least one random intercept.
<code>cluster_size</code>	Optional. A scalar average cluster size used to compute the design effect, or the string "auto" to take cluster sizes from the fitted model. If NULL, no design effect is computed.

Details

The ICC is a model-based share of variance, not a causal statement about clusters and not a fixed property of a population. With `cluster_size = "auto"` the observed cluster sizes are taken from the model frame and two design effects are reported: the familiar $1 + (\bar{m} - 1)ICC$ using the mean cluster size, and an unequal-size version that replaces $\bar{m}$ with $\sum n_j^2 / \sum n_j$. The second is the more appropriate teaching approximation when cluster sizes vary, and both are approximations rather than substitutes for fitting and diagnosing a multilevel model.

Value

An object of class `hlm_icc` with components:

<code>icc</code>	Estimated intraclass correlation.
<code>deff</code>	Design effect based on the supplied or mean cluster size.
<code>deff_unequal</code>	Design effect adjusted for unequal cluster sizes (only when <code>cluster_size = "auto"</code>).
<code>re_var</code>	Random intercept variance.
<code>resid_var</code>	Residual variance.
<code>n_clusters</code>	Number of clusters (when available).
<code>mean_cluster_size</code>	Mean cluster size (when available).

Examples

```
set.seed(3)
toy <- data.frame(
  math_score = rnorm(80, 50, 10),
  SES = rnorm(80),
  school_id = rep(seq_len(8), each = 10)
)
m <- lme4::lmer(math_score ~ SES + (1 | school_id), data = toy)
hlm_icc(m, cluster_size = 10)
hlm_icc(m, cluster_size = "auto")
```

Description

Simulates clustered data at several target intraclass correlations, holding the total variance and the overall mean fixed, and displays them side by side. The figure is a teaching device: it shows that the ICC is a share of variance rather than a test of whether clustering exists, and that the same outcome scale can look very different at different ICC values.

Usage

```
hlm_icc_demo(
  icc = c(0.05, 0.25, 0.6),
  n_clusters = 12,
  cluster_size = 20,
  grand_mean = 50,
  total_sd = 10,
  exact = TRUE,
  seed = NULL,
  show_cluster_means = TRUE,
  sort_clusters = TRUE
)
```

Arguments

icc	Numeric vector of target intraclass correlations, each strictly between 0 and 1. Defaults to <code>c(0.05, 0.25, 0.60)</code> .
n_clusters	Number of clusters per panel. Defaults to 12.
cluster_size	Number of observations per cluster. Defaults to 20.
grand_mean	Overall mean of the simulated outcome. Defaults to 50.
total_sd	Total standard deviation of the simulated outcome, held constant across panels. Defaults to 10.
exact	Logical; rescale the simulated components so that their realized sample standard deviations match the targets, making the data-generating ICC exactly the value in each panel label. Defaults to TRUE.
seed	Optional integer seed. When supplied, the state of the random number generator is restored on exit.
show_cluster_means	Logical; draw a horizontal marker at each cluster mean. Defaults to TRUE.
sort_clusters	Logical; order the clusters within each panel by their mean. Defaults to TRUE, which makes the contrast between panels legible: the sequence of cluster means is nearly flat at a low ICC and rises steeply at a high one. Set to FALSE to keep the simulated order.

Details

For each target value the between-cluster standard deviation is $\tau = \sigma_{total}\sqrt{ICC}$ and the within-cluster standard deviation is $\sigma = \sigma_{total}\sqrt{1 - ICC}$. With `exact = TRUE` the simulated cluster effects and residuals are rescaled so that their realized sample standard deviations match those targets, so each panel is generated at exactly the ICC in its label.

That is a statement about the data-generating values, not about what a model fitted to one panel will report. An estimated ICC carries estimation error, and that error is proportionally largest where the ICC is small: with a dozen clusters and a target of 0.05, the between-cluster variance is of the same order as the sampling variance of a cluster mean, and a single fitted panel can easily return 0.03 or 0.08. The panel labels are therefore the values used to generate the data. Fitting a model to the attached data is a useful follow-up exercise, but the estimate should be expected to differ.

Value

A ggplot object. The simulated data are attached as the attribute "data".

See Also

[hlm_icc](#), [hlm_icc_plot](#)

Examples

```
hlm_icc_demo(icc = c(0.05, 0.25, 0.60), n_clusters = 8,
             cluster_size = 15, seed = 2026)
```

hlm_icc_plot

Teaching plot for intraclass correlation (ICC)

Description

Visualizes the intraclass correlation by plotting the between- and within-cluster variance components as a stacked bar. This is intended as a teaching diagram to help students see how the ICC reflects the share of variance that lies between clusters.

Usage

```
hlm_icc_plot(model, cluster_size = NULL, labels = TRUE)
```

Arguments

<code>model</code>	A fitted <code>lmerMod</code> model with a random intercept.
<code>cluster_size</code>	Optional scalar giving the average cluster size, or "auto" to take cluster sizes from the fitted model; passed to hlm_icc to compute the design effect.
<code>labels</code>	Logical; print the component labels and percentages inside the bar. Defaults to TRUE.

Value

A ggplot object.

See Also

[hlm_icc](#), [hlm_icc_demo](#)

Examples

```
set.seed(3)
toy <- data.frame(
  math_score = rnorm(80, 50, 10),
  school_id = rep(seq_len(8), each = 10)
)
m0 <- lme4::lmer(math_score ~ 1 + (1 | school_id), data = toy)
hlm_icc_plot(m0, cluster_size = "auto")
```

hlm_random_slope_plot *Cluster-specific slopes from a random-slope model*

Description

Draws one fitted line per cluster from a model containing a random slope, together with the average fitted line. The display communicates unexplained heterogeneity in a Level-1 association across clusters.

Usage

```
hlm_random_slope_plot(
  model,
  x_within,
  cluster,
  n_points = 40,
  n_clusters = 20,
  select = c("spread", "random", "all"),
  x_range = NULL,
  hold_other = c("mean", "zero"),
  show_average = TRUE,
  title = NULL,
  subtitle = NULL
)
```

Arguments

model	A fitted lmerMod model with a random slope term of the form (x_within cluster).
x_within	Name of the Level-1 predictor with a random slope (string).

cluster	Name of the clustering factor (string).
n_points	Number of points along the x-axis. Defaults to 40.
n_clusters	Maximum number of clusters to display. Defaults to 20.
select	How clusters are chosen when there are more than n_clusters: "spread" (the default) selects clusters evenly across the distribution of conditional slopes, "random" samples them, and "all" displays every cluster. "random" uses the current state of the random number generator; call <code>set.seed()</code> beforehand for a reproducible figure.
x_range	Optional length-2 numeric vector giving the plotting range of x_within. Defaults to the 0.02 and 0.98 quantiles of the observed predictor.
hold_other	Either "mean" (the default) or "zero", controlling the value at which the remaining fixed-effect terms are held.
show_average	Logical; draw the average fitted line. Defaults to TRUE.
title, subtitle	Optional plot title and subtitle. NULL uses informative defaults; NA omits the element.

Details

A random slope is not a cross-level interaction. It shows that the Level-1 association varies across clusters without saying why. A cross-level interaction requires an observed Level-2 moderator; see [hlm_cross_level_plot](#).

All fixed-effect columns other than the intercept and x_within are held at their sample means when hold_other = "mean", so the lines are plotted on the scale of the observed outcome. Cluster-specific intercepts and slopes are conditional (empirical Bayes) quantities and are already shrunk toward the average; they should not be treated as precise rankings of individual clusters.

Value

A ggplot object. The plotted line data are attached as the attribute "data".

See Also

[hlm_cross_level_plot](#), [hlm_shrinkage_plot](#)

Examples

```
set.seed(42)
n_schools <- 10
n_students <- 15
school_id <- rep(seq_len(n_schools), each = n_students)
SES_c <- rnorm(n_schools * n_students)
u0 <- rep(rnorm(n_schools, sd = 0.5), each = n_students)
u1 <- rep(rnorm(n_schools, sd = 0.3), each = n_students)
math_score <- 50 + 2 * SES_c + u0 + u1 * SES_c +
  rnorm(n_schools * n_students, sd = 1)
toy <- data.frame(math_score = math_score, SES_c = SES_c,
  school_id = school_id)
```

```
m <- lme4::lmer(math_score ~ SES_c + (SES_c | school_id), data = toy)
hlm_random_slope_plot(m, x_within = "SES_c", cluster = "school_id")
```

hlm_shrinkage_plot *Partial pooling: raw cluster means and multilevel estimates*

Description

Compares each cluster's raw mean with its multilevel (empirical Bayes) estimate and draws an arrow between them. The figure makes partial pooling visible: small clusters are pulled further toward the overall mean than large clusters, and extreme raw means are moderated most.

Usage

```
hlm_shrinkage_plot(
  model,
  n_clusters = 25,
  select = c("spread", "random", "all"),
  style = c("arrows", "size"),
  title = NULL,
  subtitle = NULL
)
```

Arguments

model	A fitted lmerMod model with a random intercept, normally an unconditional model of the form $y \sim 1 + (1 \mid \text{cluster})$.
n_clusters	Maximum number of clusters to display. Defaults to 25.
select	How clusters are chosen when there are more than n_clusters: "spread" (the default) selects clusters evenly across the distribution of raw means, "random" samples them, and "all" displays every cluster. "random" uses the current state of the random number generator.
style	"arrows" (the default) places clusters on the vertical axis ordered by raw mean; "size" places cluster size on the horizontal axis, which shows directly that smaller clusters shrink more.
title, subtitle	Optional plot title and subtitle. NULL uses informative defaults; NA omits the element.

Details

The multilevel estimate for cluster j is $\hat{\beta}_0 + \hat{u}_{0j}$, which for an unconditional random-intercept model equals $\lambda_j \bar{Y}_j + (1 - \lambda_j) \hat{\beta}_0$ with reliability $\lambda_j = \tau^2 / (\tau^2 + \sigma^2 / n_j)$. The comparison is interpretable in that unconditional case; a warning is issued when the model contains additional fixed effects, because a raw cluster mean and an adjusted intercept are then no longer the same quantity.

Value

A ggplot object. A data frame with the raw mean, multilevel estimate, cluster size, and reliability for every displayed cluster is attached as the attribute "data".

See Also

[hlm_icc](#), [hlm_icc_demo](#)

Examples

```
set.seed(11)
cluster_size <- sample(3:40, 25, replace = TRUE)
school_id <- rep(seq_along(cluster_size), times = cluster_size)
u0 <- rep(rnorm(length(cluster_size), sd = 5), times = cluster_size)
math_score <- 50 + u0 + rnorm(length(school_id), sd = 10)
toy <- data.frame(math_score = math_score, school_id = school_id)

m0 <- lme4::lmer(math_score ~ 1 + (1 | school_id), data = toy)
hlm_shrinkage_plot(m0)
```

hlm_xint_geom

Geometry of a cross-level interaction (deprecated)

Description

Deprecated in hlmLab 0.2.0. The display produced by this function shows random-slope heterogeneity, which is not the same thing as a cross-level interaction, so the function was renamed. Use [hlm_random_slope_plot](#) instead; use [hlm_cross_level_plot](#) when the model contains an observed Level-2 moderator.

Usage

```
hlm_xint_geom(model, x_within, cluster, n_points = 20, n_clusters = 20)
```

Arguments

model	A fitted lmerMod model with a random slope term.
x_within	Name of the Level-1 predictor with a random slope (string).
cluster	Name of the clustering factor (string).
n_points	Number of points to plot along the x-axis. Defaults to 20.
n_clusters	Maximum number of clusters to display (sampled).

Value

A ggplot object, produced by [hlm_random_slope_plot](#).

Examples

```
set.seed(42)
school_id <- rep(seq_len(10), each = 15)
SES_c <- rnorm(150)
u0 <- rep(rnorm(10, sd = 0.5), each = 15)
u1 <- rep(rnorm(10, sd = 0.3), each = 15)
math_score <- 50 + 2 * SES_c + u0 + u1 * SES_c + rnorm(150, sd = 1)
toy <- data.frame(math_score = math_score, SES_c = SES_c,
                  school_id = school_id)
m <- lme4::lmer(math_score ~ SES_c + (SES_c | school_id), data = toy)
suppressWarnings(
  hlm_xint_geom(m, x_within = "SES_c", cluster = "school_id")
)
```

plot.hlm_context	<i>Plot method for hlm_context objects</i>
------------------	--

Description

Produces an error-bar plot of within, between, and contextual effects with confidence intervals. Intended as a teaching diagram.

Usage

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

Arguments

x	An object of class hlm_context.
...	Not used.

Value

A ggplot object.

plot.hlm_decompose	<i>Plot method for hlm_decompose objects</i>
--------------------	--

Description

Produces a simple bar chart of variance shares across components, suitable for teaching how variance is partitioned across levels.

Usage

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

Arguments

x	An object of class hlm_decompose.
...	Not used.

Value

A ggplot object.

Index

hlm_context, [2](#)
hlm_context_plot, [4](#)
hlm_cross_level_plot, [5](#), [13](#), [15](#)
hlm_decompose, [6](#)
hlm_decompose_long, [7](#)
hlm_icc, [8](#), [11](#), [12](#), [15](#)
hlm_icc_demo, [10](#), [12](#), [15](#)
hlm_icc_plot, [11](#), [11](#)
hlm_random_slope_plot, [5](#), [6](#), [12](#), [15](#)
hlm_shrinkage_plot, [13](#), [14](#)
hlm_xint_geom, [15](#)

plot.hlm_context, [16](#)
plot.hlm_decompose, [16](#)