

Package: localhierarchy (via r-universe)

June 2, 2026

Type Package

Title An R package to facilitate fitting of global and local Bayesian hierarchical models

Version 0.91

Date 2025-11-21

Description The localhierarchy package implements a flexible set up to estimate parameters hierarchically and to carry out local model fits, whereby higher-level parameters are fixed.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Roxygen list(markdown = TRUE)

Depends R (>= 4.1.0)

Imports dplyr, ggplot2, posterior, purrr, rlang, stringr, tibble, tidyr, truncnorm, tidybayes, readr, glue

URL <https://github.com/AlkemaLab/localhierarchy>,
<https://alkemalab.github.io/localhierarchy/>

BugReports <https://github.com/AlkemaLab/localhierarchy/issues>

Suggests bayesplot, cmdstanr, knitr, quarto, rmarkdown

Config/Needs/website quarto

VignetteBuilder knitr

Config/pak/sysreqs libicu-dev libx11-dev

Repository <https://alkemalab.r-universe.dev>

Date/Publication 2026-06-02 03:07:50 UTC

RemoteUrl <https://github.com/AlkemaLab/localhierarchy>

RemoteRef HEAD

RemoteSha 5cc0bf1d4cdc6452c7518d60a33f95d8463a2833

Contents

check_nas	2
fit_model_localhierarchy	3
get_geo_unit_index_data	5
get_posterior_summaries_localhierarchy	6
hierarchical_data	6
hierarchical_model_matrix	7
hierarchical_param_stan_data	8
plot_muraw_localhierarchy	9
plot_posterior_summaries_localhierarchy	9
plot_prior_post_sigmas_localhierarchy	10
posterior_summary_hierparam_localhierarchy	11
simulate_multilevel_data	12

Index	13
--------------	-----------

check_nas	<i>Check for missing values</i>
-----------	---------------------------------

Description

This function checks if there are any NAs in a specified column of a data frame.

Usage

```
check_nas(data, column)
```

Arguments

data	tibble or data frame
column	column to check for NAs

Value

stops if there are NAs in the column or if the column does not exist

`fit_model_localhierarchy`*Fit localhierarchy example model*

Description

This function fits the localhierarchy example model

Usage

```
fit_model_localhierarchy(  
  survey_df,  
  y = "y",  
  area = "iso",  
  area_select = NULL,  
  runstep,  
  global_fit = NULL,  
  hierarchical_level = c("intercept", "subcluster", "iso"),  
  add_subnational_hierarchy = "subnat",  
  use_globalsubnat_fromnat = TRUE,  
  mu_isvector = FALSE,  
  chains = 4,  
  iter_sampling = 300,  
  iter_warmup = 150,  
  compile_model = TRUE,  
  force_recompile = FALSE,  
  seed = 1234,  
  refresh = 200,  
  adapt_delta = 0.9,  
  max_treedepth = 14  
)
```

Arguments

<code>survey_df</code>	tibble with survey data
<code>y</code>	column name of outcome, defaults to <code>y</code> .
<code>area</code>	column name of the area of each observation (used as geounit; iso or subnational region)
<code>area_select</code>	area name to use for local run (eg iso code or subnat region name)
<code>runstep</code>	Type of run, defines which model fitting step to perform (see Details for options).
<code>global_fit</code>	optional global fit object, used to obtain fixed values to use for some parameters in the current fit (see Details).
<code>hierarchical_level</code>	vector specifying hierarchical structure used for mu

<code>add_subnational_hierarchy</code>	level that's added to the hierarchy for subnational, defaults to <code>subnat</code>
<code>use_globalsubnat_fromnat</code>	Logical, whether in a local subnational run, to use the global fit derived from national data if TRUE and local subnat run, <code>global_fit</code> needs to contain object <code>fit_globalsubnat_fromnat</code>
<code>mu_isvector</code>	Logical, TRUE if <code>mu</code> is a vector, defaults to FALSE
	Settings for sampling
<code>chains</code>	number of chains to run
<code>iter_sampling</code>	number of posterior samples to draw
<code>iter_warmup</code>	number of warmup iterations
<code>compile_model</code>	boolean indicator of whether to compile the Stan model
<code>force_recompile</code>	boolean indicator of whether to force recompilation of the Stan model
<code>seed</code>	random seed
<code>refresh</code>	number of iterations between progress updates
<code>adapt_delta</code>	target acceptance rate for the No-U-Turn Sampler
<code>max_treedepth</code>	maximum tree depth for the No-U-Turn Sampler

Details

The `fit_model_localhierarchy` function fits the toy example for hierarchical models/seq fitting. The argument `runstep` determines the type of run to perform. The following run steps are supported:

- `"global_national"`: Fit the global model.
- `"local_national"`: Fit the model to data from a single country, using a `global_national` fit.
- `"global_subnational"`: Fit the model to global database with subnational data, using a `global_national` fit (NISE modeling).
- `"local_subnational"`: Fit the model to subnational data from a single country or region, using a `global_subnational` fit.

The options are also explained in the article with the package.

Details on hierarchical set ups used The package allows the structure of the hierarchical prior to be configured by the user through the `hierarchical_level` argument. These arguments expect a character vector that specifies a nesting hierarchical structure. Each element of the vector must be either `"intercept"` or a column name in the survey dataset, where `"intercept"` will add a global intercept for the parameter. The vector must be in descending order in terms of the hierarchy: that is, it starts with `"intercept"` and proceeds down the hierarchy.

For example, suppose we are fitting country-level data, where the dataset has columns `"name_country"`, `"name_sub_region"`, and `"name_region"` containing the name of the country, sub-region, and region that each observation belongs to. To specify that parameter `mu` should be fitted with a hierarchical model in which countries are nested within sub-regions within regions within world, we would use the argument `hierarchical_level = c("intercept", "name_region", "name_sub_region", "name_country")`.

Optionally, model parameters can be fixed to values from a previous model fit provided via the `global_fit` argument. In a typical use case, the `global_fit` will have been fit to data from many geographic units (e.g., all countries), while the current fit uses data from a smaller number of locations. When using a `global_fit` to fix parameter values, what exactly is fixed is determined by the `runstep` and `global_fit` combi.

Value

List that contains samples, `stan_data`, other information relevant to model fit (arguments), and for global fits, point estimates of relevant parameters (`post_summ`). For subnational global fits, the list includes `fit_globalsubnat_fromnat`, which is the global fit with additional subnational sigmas added to the `postsum` object.

`get_geo_unit_index_data`

Get unique hierarchical levels from survey data

Description

Get unique hierarchical levels from survey data and assign index "c" based on area.

Usage

```
get_geo_unit_index_data(data, hierarchical_levels, area)
```

Arguments

<code>data</code>	survey data of interest that contains the <code>hierarchical_column_names</code>
<code>hierarchical_levels</code>	vector that contains one of more specs of <code>hierarchical_level</code> (incl intercept)
<code>area</code>	unit of analysis (eg country or subnational region)

Value

tibble with unique hierarchical levels (one column for each level) and index "c" assigned based on area

Examples

```
survey_dat <- tibble::tibble(subcluster = c("A", "A", "B", "B"),
  iso = c("iso1", "iso2", "iso3", "iso4"))
get_geo_unit_index_data(survey_dat,
  hierarchical_levels = c("intercept", "subcluster", "iso"),
  area = "iso")
```

```
get_posterior_summaries_localhierarchy
```

Get posterior summaries for use in local models

Description

Get posterior summaries for a set of parameters for use in local model

Usage

```
get_posterior_summaries_localhierarchy(
  fit,
  params = c("mu_raw", "mu_sigma", "nonse")
)
```

Arguments

fit	fit object to summarize
params	vector of parnames to summarize (w/o index)

Value

tibble with variable (parname with index used in model) and postmean (posterior mean)

```
hierarchical_data
```

Create hierarchical data structure for a hierarchical model.

Description

Create hierarchical data structure for a hierarchical model.

Usage

```
hierarchical_data(geo_unit_index_data, hierarchical_level)
```

Arguments

geo_unit_index_data	tibble with unique hierarchical levels (one column for each level)
hierarchical_level	vector with hierarchical spec

Value

list with model matrix and info on start and end indices associated with different levels

hierarchical_model_matrix
Hierarchical model matrix

Description

Create list with information including a model matrix, associated with the `geo_unit_data` and `hierarchical_levels`

Usage

```
hierarchical_model_matrix(geo_unit_index_data, hierarchical_level)
```

Arguments

`geo_unit_index_data`
tibble with unique hierarchical levels (one column for each level, one row per combi)

`hierarchical_level`
vector with hierarchical spec from highest to lowest levels

Value

A list with components `modelmatrix`, `assign`, and `index`:

- `modelmatrix`: A matrix where each column refers to one eta (number of rows equals number of lowest level units).
- `assign`: Integer vector of length equal to the number of etas; 0 for the intercept, then a hierarchical level index (starting at 2) for each eta level.
- `index`: A tibble with `n_eta` rows, and columns:
 - `i`: Index of eta
 - `column`: Hierarchical level
 - `level`: Name of hierarchical level

All components are associated with the provided `geo_unit_data` and `hierarchical_levels`.

 hierarchical_param_stan_data

Stan data for hierarchical parameters

Description

Set up data for Stan related to hierarchical parameters

Usage

```
hierarchical_param_stan_data(
  param_name,
  param_data,
  global_fit = NULL,
  hierarchical_terms_fixed,
  hierarchical_sigmas_fixed
)
```

Arguments

param_name	The name of the parameter we are working with, e.g. "mu"
param_data	Data structures as constructed by hierarchical_data for that parameter
global_fit	an optional "global" fit that will be used to extract parameter estimates for any specified hierarchical units to fix. Defaults to NULL, no fixing.
hierarchical_terms_fixed	character vector specifying hierarchical levels for which the terms should be fixed (subset of hierarchical_levels). Used only if global fit is provided.
hierarchical_sigmas_fixed	character vector specifying hierarchical levels for which the terms should be fixed (subset of hierarchical_levels, intercept is always included) Used only if global fit is provided.

Value

named list with Stan data relevant to the hierarchical set up for this parameter, e.g. if the param_name is "mu", these will be mu_n_terms, mu_n_sigma, mu_re_start, mu_re_end, mu_model_matrix, mu_n_terms_fixed, m

 plot_muraw_localhierarchy

Plot prior and posterior densities of location parameters

Description

This function plots the posterior densities of location (`mu_raw`) parameters, with priors added.

Usage

```
plot_muraw_localhierarchy(
  fit,
  parname,
  morethan1param = FALSE,
  nresultsperpage = 30
)
```

Arguments

<code>fit</code>	List that includes " <code>parname</code> "_raw_estimate and stan_data
<code>parname</code>	Selected parameter name (example: mu)
<code>morethan1param</code>	Logical, does parname refer to more than 1 parameter (a vector)
<code>nresultsperpage</code>	Number of results per page in summary plots

Value

lists with list 'summary_plots' and list 'plots_allmuraw'. summary plots gives summary CIs, nresultsperpage at a time. plots_allmuraw gives all the individual plots of each mu_raw, with density per chain and prior added. If `morethan1param = TRUE`, then each list contains a list per parameter k

 plot_posterior_summaries_localhierarchy

Plot summaries of hierarchical parameters

Description

Display outputs from posterior_summary_hierparam_localhierarchy

Usage

```
plot_posterior_summaries_localhierarchy(
  res,
  hierarchy_select = NULL,
  areas_select = NULL,
  res2 = NULL,
  modelname1 = "model 1",
  modelname2 = "model 2",
  k_select = NULL,
  dodge = position_dodge(width = 0.5)
)
```

Arguments

res	output from posterior_summary_hierparam_localhierarchy
hierarchy_select	optional, what hierarchical level to show? if NULL, all levels are shown
areas_select	optional: specific areas in a level to filter by, allowed only if one hierarchical level is selected
res2	optional, output from posterior_summary_hierparam_localhierarchy for comparison
modelname1	label for res
modelname2	label for res2
k_select	optional, if res contains k, which k values to show? if NULL, all k values are shown
dodge	used for offsetting plots, default is 0.5

Value

ggplot object

plot_prior_post_sigmas_localhierarchy

Plot prior and posteriors of sigmas from hierarchical models

Description

This function plots the prior and posterior densities of sigma_estimate parameters.

Usage

```
plot_prior_post_sigmas_localhierarchy(fit, parname)
```

Arguments

fit List, needs to include parname_sigma_estimate and stan_data
 parname Selected parameter name (example: "mu")

Value

Plot with density of sigma_estimate and prior added

posterior_summary_hierparam_localhierarchy

Calculate posterior summaries for hierarchical parameters

Description

Calculate posterior summaries for hierarchical parameters

Usage

```
posterior_summary_hierparam_localhierarchy(  
  fit,  
  parname,  
  morethan1param = FALSE,  
  hierarchical_levels = fit$hierarchical_level  
)
```

Arguments

fit needs to include parname_star
 parname selected parameter name (example: "mu")
 morethan1param does paramname refer to more than 1 parameter (a vector)
 hierarchical_levels specifies the names of the hierarchical levels (defaults to fit\$hierarchical_level)

Value

list with summaries of mu for each hierarchical level (units with each level) these mus are obtained by summing up all relevant etas for morethan1param, each level has a list where k refers to the index

```
simulate_multilevel_data
  Simulate multilevel data
```

Description

Simulates a nested multilevel data structure. Lowest level can be used as observational error.

Usage

```
simulate_multilevel_data(
  n_levels = 3,
  n_units_perlevel = rep(25, n_levels),
  sigma_perlevel = rep(1, n_levels),
  mu_global = 0,
  add_data = TRUE,
  n_data = 10,
  sigma_y = 0.05,
  add_data1levelup = FALSE,
  seed = 12345
)
```

Arguments

<code>n_levels</code>	Number of hierarchical levels (default: 3).
<code>n_units_perlevel</code>	A vector specifying the number of units at each level (default: <code>rep(25, n_levels)</code>).
<code>sigma_perlevel</code>	A vector specifying the standard deviation of etas at each level (default: <code>rep(1, n_levels)</code>).
<code>mu_global</code>	Global mean for the outcome variable (default: 0).
<code>add_data</code>	Logical indicating whether to add observations (default: TRUE). If TRUE, adds column y.
<code>n_data</code>	Number of observations to add at the lowest level (default: 10).
<code>sigma_y</code>	Standard deviation of the added observations (default: 0.05)
<code>add_data1levelup</code>	Logical indicating whether to add observations one level up e.g. at national level if lowest level is subnational (default: FALSE).
<code>seed</code>	Random seed for reproducibility (default: 1234).

Value

A tibble with columns for each level, `mu_global`, `etas`, and their total labeled `y`

Index

`check_nas`, [2](#)

`fit_model_localhierarchy`, [3](#)

`get_geo_unit_index_data`, [5](#)

`get_posterior_summaries_localhierarchy`,
[6](#)

`hierarchical_data`, [6](#)

`hierarchical_model_matrix`, [7](#)

`hierarchical_param_stan_data`, [8](#)

`plot_muraw_localhierarchy`, [9](#)

`plot_posterior_summaries_localhierarchy`,
[9](#)

`plot_prior_post_sigmas_localhierarchy`,
[10](#)

`posterior_summary_hierparam_localhierarchy`,
[11](#)

`simulate_multilevel_data`, [12](#)