

Package ‘dgo’

July 17, 2018

Title Dynamic Estimation of Group-Level Opinion

Version 0.2.15

Date 2018-07-16

Description Fit dynamic group-level item response theory (IRT) and multilevel regression and poststratification (MRP) models from item response data. dgo models latent traits at the level of demographic and geographic groups, rather than individuals, in a Bayesian group-level IRT approach developed by Caughey and Warshaw (2015) <doi:10.1093/pan/mpu021>. The package also estimates subpopulations' average responses to single survey items with a dynamic MRP model proposed by Park, Gelman, and Bafumi (2004) <doi:10.11126/stanford/9780804753005.003.0011>.

Maintainer James Dunham <james@jdunham.io>

License GPL-3

URL <https://jdunham.io/dgo/>

BugReports <https://github.com/jamesdunham/dgo/issues>

Depends dgodata, R (>= 3.2.2), rstan (>= 2.15.1)

LazyData true

Imports assertthat, data.table, ggplot2, lubridate, methods, R6, survey

Suggests knitr, reshape2, rmarkdown, testthat

Collate 'aggregate_item_responses.r' 'assertions.r' 'class-control.r' 'class-dgo_fit.r' 'class-dgirt_fit.r' 'constants.r' 'class-dgirtin.r' 'class-dgmrp_fit.r' 'dgirt.r' 'dichotomize_item_responses.r' 'methods-control.r' 'methods-dgirtfit-plot.r' 'methods-dgirtfit-poststratify.r' 'methods-dgirtfit.r' 'methods-dgirtin.r' 'name_helpers.r' 'package.R' 'rake_partial.r' 'restrict_input_data.r' 'reweight_item_responses.r' 'shape.r' 'shape_hierarchical.r' 'toy_dgirt_in.r' 'toy_dgirtfit.r' 'validate_dgirtIn.r' 'validate_input_data.r'

RoxygenNote 6.0.1

NeedsCompilation yes

Author James Dunham [aut, cre],
Devin Caughey [aut],
Christopher Warshaw [aut]

Repository CRAN

Date/Publication 2018-07-17 12:20:04 UTC

R topics documented:

dgirt	2
dgirtfit-class	4
dgirtin-class	4
dgirt_fit-class	5
dgirt_plot	6
dgmrp_fit-class	8
dgo	8
dgo_fit-class	8
poststratify	9
shape	10
show,dgo_fit-method	13
toy_dgirtfit	14
toy_dgirt_in	15

Index 16

dgirt	<i>Fit a dynamic group IRT or single-issue MRP model</i>
-------	--

Description

dgirt and dgmrp make calls to [stan](#) with the Stan code and data for their respective models.

Usage

```
dgirt(shaped_data, ..., separate_t = FALSE, delta_tbar_prior_mean = 0.65,
      delta_tbar_prior_sd = 0.25, innov_sd_delta_scale = 2.5,
      innov_sd_theta_scale = 2.5, version = "2017_01_04",
      hierarchical_model = TRUE, model = NULL)
```

```
dgmrp(shaped_data, ..., separate_t = FALSE, delta_tbar_prior_mean = 0.65,
       delta_tbar_prior_sd = 0.25, innov_sd_delta_scale = 2.5,
       innov_sd_theta_scale = 2.5, version = "2017_01_04_singleissue",
       model = NULL)
```

Arguments

shaped_data	Output from shape .
...	Further arguments, passed to stan .
separate_t	Whether smoothing of estimates over time should be disabled. Default FALSE.
delta_tbar_prior_mean	Prior mean for delta_tbar, the normal weight on theta_bar in the previous period. Default 0.65.
delta_tbar_prior_sd	Prior standard deviation for delta_bar. Default 0.25.
innov_sd_delta_scale	Prior scale for sd_innov_delta, the Cauchy innovation standard deviation of nu_geo and delta_gamma. Default 2.5.
innov_sd_theta_scale	Prior scale for sd_innov_theta, the Cauchy innovation standard deviation of gamma, xi, and if constant_item is FALSE the item difficulty diff. Default 2.5.
version	The name of the dgo model to estimate, or the path to a .stan file. Valid names for dgo models are "2017_01_04", "2017_01_04_singleissue". Ignored if argument model is used.
hierarchical_model	Whether a hierarchical model should be used to smooth the group IRT estimates. If set to FALSE, the model will return raw group-IRT model estimates for each group. Default TRUE.
model	A Stan model object of class stanmodel to be used in estimation. Specifying this argument avoids repeated model compilation. Note that the Stan model object for a model fitted with dgirt() or dgmrp() can be found in the stanmodel slot of the resulting dgirt_fit or dgmrp_fit object.

Details

The user will typically pass further arguments to [stan](#) via the ... argument, at a minimum iter and cores.

By default dgirt and dgmrp override the [stan](#) default for its pars argument to specify typical parameters of interest. They also set iter_r to 1L.

Important: the dgirt model assumes consistent coding of the polarity of item responses for identification.

Value

A [dgo_fit-class](#) object that extends [stanfit-class](#).

dgirtfit-class	<i>A class for fitted dynamic group IRT models</i>
----------------	--

Description

dgo 0.2.8 deprecated the dgirtfit class and replaced it with the [dgirt_fit](#) class.

dgirtin-class	<i>A class for data ready to model</i>
---------------	--

Description

shape() generates objects of class dgirtIn for modeling with dgirt() and dgrp().

Usage

```
summary(object, ...)

## S4 method for signature 'dgirtIn'
summary(object, ...)

print(x, ...)

## S4 method for signature 'dgirtIn'
print(x, ...)

get_item_names(x)

## S4 method for signature 'dgirtIn'
get_item_names(x)

get_n(x, by = NULL, aggregate_name = NULL)

## S4 method for signature 'dgirtIn'
get_n(x, by = NULL, aggregate_name = NULL)

get_item_n(x, by = NULL, aggregate_data = FALSE)

## S4 method for signature 'dgirtIn'
get_item_n(x, by = NULL, aggregate_data = FALSE)

## S4 method for signature 'dgirtIn'
show(object)
```

Arguments

object	An object of class dgirtIn as returned by shape.
...	Unused.
x	An object of class dgirtIn as returned by shape.
by	The name of a grouping variable.
aggregate_name	If specified get_n will operate on the table passed to shape as aggregate_data instead of on the individual data and count nonmissingness in the given variable.
aggregate_data	If specified get_item_n will operate on the table passed to shape as aggregate_data instead of on the individual data.

Value

A list of item names.

Examples

```

data(toy_dgirt_in)
get_item_names(toy_dgirt_in)
get_n(toy_dgirt_in)
get_n(toy_dgirt_in, by = "year")
get_n(toy_dgirt_in, by = "source")
get_item_n(toy_dgirt_in)
get_item_n(toy_dgirt_in, by = "year")
data(toy_dgirt_in)
get_item_names(toy_dgirt_in)
# respondent count
data(toy_dgirt_in)
get_n(toy_dgirt_in)

# respondent count by year
get_n(toy_dgirt_in, by = "year")

# respondent count by year and survey identifier
get_n(toy_dgirt_in, by = c("year", "source"))

data(toy_dgirt_in)
get_item_n(toy_dgirt_in)
get_item_n(toy_dgirt_in, by = "year")

```

dgirt_fit-class

A class for fitted dynamic group IRT models

Description

`dgirt` returns a fitted model object of class `dgirt_fit`, which inherits from `dgo_fit`.

Details

dgo 0.2.8 deprecated the dgirtfit class and replaced it with the [dgirt_fit](#) class.

Slots

dgirt_in [dgirtin-class](#) data used to fit the model.

See Also

[dgmpr_fit](#) [dgo_fit](#)

Examples

```
data(toy_dgirtfit)
# summarize the fitted results
summary(toy_dgirtfit, pars = 'xi')

# get posterior means with a convenience function
get_posterior_mean(toy_dgirtfit, pars = 'theta_bar')

# generally apply functions to posterior samples after warmup; n.b.
# `as.array` is iterations x chains x parameters so `MARGIN = 3` applies
# `FUN` over iterations and chains
apply(as.array(toy_dgirtfit, pars = 'xi'), 3, mean)

# access the posterior samples
head(as.data.frame(toy_dgirtfit, pars = 'theta_bar'))
```

dgirt_plot

Plot estimates and diagnostic statistics

Description

dgirt_plot plots estimates from a dgo model. plot_rhats plots split R-hats.

Usage

```
dgirt_plot(x, ...)

## S4 method for signature 'dgo_fit'
dgirt_plot(x, y_fun = "median", y_min = "q_025",
  y_max = "q_975", pars = "theta_bar")

## S4 method for signature 'data.frame'
dgirt_plot(x, group_name, time_name, geo_name,
  y_fun = "median", y_min = "q_025", y_max = "q_975")

## S4 method for signature 'dgo_fit,missing'
```

```

plot(x, y, ...)

plot_rhats(x, ...)

## S4 method for signature 'dgo_fit'
plot_rhats(x, pars = "theta_bar", facet_vars = NULL,
           shape_var = NULL, color_var = NULL, x_var = NULL)

```

Arguments

x	A dgo_fit-class object.
...	Further arguments to dgirt_plot .
y_fun	Summary function to be plotted as y.
y_min	Summary function giving the ymin argument for a geom_pointrange object.
y_max	Summary function giving the ymax argument for a geom_pointrange object.
pars	Selected parameter.
group_name	A discrete grouping variable that will be passed to the color argument of aes.
time_name	A time variable with numeric values that will be plotted on the x axis.
geo_name	A variable representing local areas that will be used in faceting.
y	Ignored.
facet_vars	Optionally, one or more variables passed to facet_wrap
shape_var, color_var, x_var	Optionally, a variable passed to the shape, color, or x arguments of aes_string, respectively.

Examples

```

## Not run:
data(toy_dgirtfit)
dgirt_plot(toy_dgirtfit)
dgirt_plot(toy_dgirtfit, y_min = NULL, y_max = NULL)
p <- dgirt_plot(toy_dgirtfit)
p %>% ylab("posterior median")

## End(Not run)
## Not run:
data(toy_dgirtfit)
ps <- poststratify(toy_dgirtfit, annual_state_race_targets, strata_names =
                  c("state", "year"), aggregated_names = "race3")
dgirt_plot(ps, group_name = NULL, time_name = "year", geo_name = "state")

## End(Not run)
## Not run:
data(toy_dgirtfit)
plot(toy_dgirtfit)

## End(Not run)

```

```
## Not run:
data(toy_dgirtfit)
plot_rhats(toy_dgirtfit)
plot_rhats(toy_dgirtfit, facet_vars = "race3") +
  scale_x_continuous(breaks = seq.int(2006, 2008))

## End(Not run)
```

dgmrp_fit-class *A class for fitted dynamic group MRP models*

Description

`dgmrp` returns a fitted model object of class `dgmrp_fit`, which inherits from `dgo_fit`.

Slots

`dgirt_in` `dgirtin-class` data used to fit the model.

See Also

`dgirt_fit` `dgo_fit`

dgo *dgo: Dynamic Estimation of Group-level Opinion*

Description

Fit dynamic group-level IRT and MRP models from individual or aggregated item response data. This package handles common preprocessing tasks and extends functions for inspecting results, poststratification, and quick iteration over alternative models.

dgo_fit-class *A class for fitted models*

Description

`dgo_fit` is a superclass for `dgirt_fit` and `dgmrp_fit` that inherits from the `stanfit-class` in the `rstan` package.

Slots

`dgirt_in` `dgirtin-class` data used to fit the model.
`call` The function call that returned the `dgo_fit` object.

See Also

`dgmrp_fit` `dgo_fit`

Description

This function reweights and aggregates estimates from `dgirt` for strata defined by modeled variables. The names of each of the model's time, geographic, and demographic grouping variables can be given in either the `strata_names` or `aggregated_names` argument. The result has estimates for the strata indicated by the `strata_names` argument, aggregated over the variables specified in `aggregated_names`. `poststratify` requires a table given as `target_data` with population proportions for the interaction of the variables given in `strata_names` and `aggregated_names`.

Usage

```
poststratify(x, target_data, strata_names, aggregated_names,
            proportion_name = "proportion", ...)

## S4 method for signature 'dgo_fit'
poststratify(x, target_data, strata_names, aggregated_names,
            proportion_name = "proportion", pars = "theta_bar")

## S4 method for signature 'data.frame'
poststratify(x, target_data, strata_names,
            aggregated_names, proportion_name = "proportion")
```

Arguments

<code>x</code>	A <code>data.frame</code> or <code>dgo_fit</code> object.
<code>target_data</code>	A table giving the proportions contributed to strata by the interaction of <code>strata_names</code> and <code>aggregated_names</code> .
<code>strata_names</code>	Names of variables whose interaction defines population strata.
<code>aggregated_names</code>	Names of variables to be aggregated over in poststratification.
<code>proportion_name</code>	Name of the column in <code>target_data</code> that gives strata proportions.
<code>...</code>	Additional arguments to methods.
<code>pars</code>	Selected parameter names.

Value

A table of poststratified estimates.

Examples

```
## Not run:
data(toy_dgirtfit)

# the stratifying variables should uniquely identify proportions in the
# target data; to achieve this, sum over the other variables
targets <- aggregate(proportion ~ state + year + race3, targets, sum)

# the dgirtfit method of poststratify takes a dgirtfit object, the target
# data, the names of variables that define population strata, and the names
# of variables to be aggregated over
post <- poststratify(toy_dgirtfit, targets, c("state", "year"), "race3")

## End(Not run)
```

shape

Prepare data for modeling

Description

This function shapes data for use in a dgirt or dgmpr model. Most arguments give the name or names of key variables in the data. These arguments end in `_name` or `_names` and should be character vectors.

Usage

```
shape(item_data = NULL, item_names = NULL, time_name, geo_name,
      group_names = NULL, id_vars = NULL, time_filter = NULL,
      geo_filter = NULL, min_t_filter = 1L, min_survey_filter = 1L,
      survey_name = NULL, modifier_data = NULL, modifier_names = NULL,
      t1_modifier_names = NULL, standardize = TRUE, target_data = NULL,
      raking = NULL, max_raked_weight = NULL, weight_name = NULL,
      proportion_name = "proportion", aggregate_data = NULL,
      aggregate_item_names = NULL, constant_item = TRUE, ...)
```

Arguments

<code>item_data</code>	A table in which items appear in columns and each row represents an individual's responses in some time period and local geographic area.
<code>item_names</code>	Item response variables.
<code>time_name</code>	A time variable with numeric values.
<code>geo_name</code>	A geographic variable representing local areas.
<code>group_names</code>	Discrete grouping variables, usually demographic. Using numeric variables is allowed but not recommended.
<code>id_vars</code>	Additional variables that should be included in the result, other than those specified elsewhere.

<code>time_filter</code>	A numeric vector giving possible values of the time variable. Observed and unobserved time periods can be given. Defaults to observed values.
<code>geo_filter</code>	A character vector giving values of the geographic variable. Defaults to observed values.
<code>min_t_filter</code>	An integer minimum of time period appearances for included items.
<code>min_survey_filter</code>	An integer minimum of survey appearances for included items.
<code>survey_name</code>	A survey identifier.
<code>modifier_data</code>	Table giving characteristics of local geographic areas in time periods. See details below.
<code>modifier_names</code>	Variables giving modifiers of geographic hierarchical parameters in <code>modifier_data</code> .
<code>t1_modifier_names</code>	Variables to be used instead of those in <code>modifier_names</code> , only in the first period.
<code>standardize</code>	Whether to standardize the variables given by <code>modifier_names</code> and <code>t1_modifier_names</code> to be zero-mean and unit-variance for performance gains. (For discussion see the Stan Language Reference section "Standardizing Predictors and Outputs.")
<code>target_data</code>	A table giving population proportions for groups by local geographic area and time period. See details below.
<code>raking</code>	A formula or list of formulas specifying the variables on which to rake survey weights.
<code>max_raked_weight</code>	A maximum over which raked weights will be trimmed. Only applied after raking. To trim unraked weights, manipulate the input data directly.
<code>weight_name</code>	A variable giving survey weights.
<code>proportion_name</code>	The variable giving population proportions for strata in <code>target_data</code> .
<code>aggregate_data</code>	A table of trial and success counts by group and item. See details below.
<code>aggregate_item_names</code>	A subset of values of the <code>item</code> variable in <code>aggregate_data</code> , for restricting the aggregate data.
<code>constant_item</code>	Whether item difficulty parameters should be constant over time.
<code>...</code>	Further arguments.

Value

An object of class `dgirtIn` expected by `dgirt` and `dgmrp`.

Item Response Data

Individual-level data giving item responses is expected as argument `item_data`. Required arguments `time_name` and `geo_name` give the names of variables in `item_data` that indicate time period and local geographic area. Optional argument `group_names` gives other respondent characteristics to be modeled. `item_data` is optional if argument `aggregate_data` is used. Note that the `dgirt()` model assumes consistent coding of the polarity of item responses for identification.

Modifier Data

Data for modeling geographic hierarchical parameters can be given with argument `modifier_data`, in which case argument `modifier_names` is required and arguments `t1_modifier_names` and `standardize` are optional.

Aggregate Item Response Data

`shape()` aggregates the individual-level item response data given as `item_data` for modeling. Data already aggregated to the group level can be provided with argument `aggregate_data`.

The data given by `aggregate_data` must be in a long table of trial and success counts indexed by item, group, and time period. The variable names given by arguments `group_names`, `geo_name`, and `time_name` should exist in `aggregate_data`. Three fixed variable names must also appear in `aggregate_data`: `item` giving item identifiers, `n_grp` giving counts of item-response trials, and `s_grp` giving counts of item-response successes. These counts should be adjusted consistently with the transformations applied during the aggregation by `shape()` of the individual `item_data`.

Reweighting

Use argument `target_data` to adjust the weighting of groups toward population targets via raking, using an adaptation of [rake](#). To adjust existing survey weights in `item_data`, provide argument `weight_name`. Otherwise, observations in `item_data` will be assigned equal starting weights. Argument `raking` defines strata. If you pass it a list of formulas like `list(~ x, ~ y)`, raking is first over `x`, then over `y`. Given an additive formula like `~ x + y`, raking is over the combinations of `x` and `y`. So, `list(~ x, ~ y + z)` is first over `x`, then over `y-z` pairs. Argument `proportion_name` is optional.

Restrictions

For convenience, data in `item_data`, `modifier_data`, `aggregate_data`, and `target_data` can be restricted (subsetting) row-wise to the time periods given by argument `time_filter` and the local geographic areas given by argument `geo_filter`.

Data can also be filtered column-wise to retain item variables that appear in a minimum of time periods, using argument `min_t_filter`, or a minimum of surveys, with argument `min_survey_filter`. Argument `survey_name` is required when filtering by survey.

If both row-wise and column-wise restrictions are specified, `shape` iterates over them until they leave the data unchanged.

Examples

```
# model individual item responses
shaped_responses <- shape(opinion, item_names = "abortion", time_name =
  "year", geo_name = "state", group_names = "race3")

# summarize result)
summary(shaped_responses)

# check sparseness of data to be modeled
get_item_n(shaped_responses, by = "year")
```

show,dgo_fit-method print *method* for dgo_fit-class objects

Description

print method for dgo_fit-class objects
 get_elapsed_time: extract chain run times from dgo_fit-class objects
 summary method for dgo_fit-class objects
 summarize method for dgo_fit-class objects
 as.data.frame method for dgo_fit-class objects
 rhats: extract split R-hats from dgo_fit-class objects

Usage

```
## S4 method for signature 'dgo_fit'
show(object)

## S4 method for signature 'dgo_fit'
print(x, ...)

print.dgo_fit(x, ...)

## S4 method for signature 'dgo_fit'
get_elapsed_time(object, ...)

## S4 method for signature 'dgo_fit'
summary(object, ..., verbose = FALSE)

## S4 method for signature 'dgo_fit'
get_posterior_mean(object, pars = "theta_bar", ...)

summarize(x, ...)

## S4 method for signature 'dgo_fit'
summarize(x, pars = "theta_bar", funs = c("mean", "sd",
  "median", "q_025", "q_975"))

## S3 method for class 'dgo_fit'
as.data.frame(x, ..., pars = "theta_bar",
  keep.rownames = FALSE)

rhats(x, ...)

## S4 method for signature 'dgo_fit'
rhats(x, pars = "theta_bar")
```

Arguments

object	A dgo_fit-class object
x	A dgo_fit-class object
...	Further arguments to stanfit-class methods.
verbose	Whether to show the full output from the rstan method.
pars	Parameter name(s)
funs	Quoted names of summary functions. 'q_025' is accepted as shorthand for 'function(x) quantile(x, .025)', and similarly 'q_975'.
keep.rownames	Whether to retain original parameter names with numeric indexes, as output from RStan.

Value

A table giving split R-hats for model parameters

Examples

```
data(toy_dgirtfit)
summarize(toy_dgirtfit)
data(toy_dgirtfit)
# access posterior samples
as.data.frame(toy_dgirtfit, pars = 'theta_bar')
data(toy_dgirtfit)
rhats(toy_dgirtfit)
```

toy_dgirtfit	<i>A minimal example of a fitted model</i>
--------------	--

Description

`dgirt` returns a `dgirtfit`-class object that extends `stanfit-class`. `toy_dgirtfit` is a minimal `dgirtfit` object for use in examples.

Usage

```
toy_dgirtfit
```

Format

A `dgirtfit-class` object.

toy_dgirt_in	<i>A minimal example of shaped data</i>
--------------	---

Description

`shape` returns a `dgirtin-class` object used with `dgirt` for DGIRT modeling. `toy_dgirt_in` is a minimal `dgirtin-class` object for use in examples.

Usage

```
toy_dgirt_in
```

Format

A `dgirtin-class` object.

Index

as.data.frame.dgo_fit
 (show,dgo_fit-method), 13

dgirt, 2, 5, 11, 14, 15
dgirt_fit, 4, 6, 8
dgirt_fit (dgirt_fit-class), 5
dgirt_fit-class, 5
dgirt_plot, 6, 7
dgirt_plot, data.frame-method
 (dgirt_plot), 6
dgirt_plot, dgo_fit-method (dgirt_plot),
 6
dgirtfit, 14
dgirtfit (dgirtfit-class), 4
dgirtfit-class, 4
dgirtin-class, 4
dgirtin-class, (dgirtin-class), 4
dgirtin-method, (dgirtin-class), 4
dgmpr, 8, 11
dgmpr (dgirt), 2
dgmpr_fit, 6, 8
dgmpr_fit (dgmpr_fit-class), 8
dgmpr_fit-class, 8
dgo, 8
dgo-package (dgo), 8
dgo_fit, 5, 6, 8
dgo_fit (dgo_fit-class), 8
dgo_fit-class, 8

get_elapsed_time, dgo_fit-method
 (show,dgo_fit-method), 13
get_item_n (dgirtin-class), 4
get_item_n, (dgirtin-class), 4
get_item_n, dgirtin-method
 (dgirtin-class), 4
get_item_names (dgirtin-class), 4
get_item_names, (dgirtin-class), 4
get_item_names, dgirtin-method
 (dgirtin-class), 4
get_n (dgirtin-class), 4

get_n, (dgirtin-class), 4
get_n, dgirtin-method (dgirtin-class), 4
get_posterior_mean, dgo_fit-method
 (show,dgo_fit-method), 13

plot, dgo_fit, missing-method
 (dgirt_plot), 6
plot_rhats (dgirt_plot), 6
plot_rhats, dgo_fit-method (dgirt_plot),
 6
poststratify, 9
poststratify, data.frame-method
 (poststratify), 9
poststratify, dgo_fit-method
 (poststratify), 9
print (dgirtin-class), 4
print, dgirtin-method (dgirtin-class), 4
print, dgo_fit-method
 (show,dgo_fit-method), 13
print.dgirtin, (dgirtin-class), 4
print.dgo_fit (show,dgo_fit-method), 13

rake, 12
rhats (show,dgo_fit-method), 13
rhats, dgo_fit-method
 (show,dgo_fit-method), 13
rstan, 8

shape, 3, 10, 15
show, dgirtin-method (dgirtin-class), 4
show, dgo_fit-method, 13
stan, 2, 3
summarize (show,dgo_fit-method), 13
summarize, dgo_fit-method
 (show,dgo_fit-method), 13
summary (dgirtin-class), 4
summary, dgirtin-method (dgirtin-class),
 4
summary, dgo_fit-method
 (show,dgo_fit-method), 13

toy_dgirt_in, [15](#)
toy_dgirtfit, [14](#)