Package 'dgo'

December 27, 2016

Title Dynamic Estimation of Group-level Opinion

Version 0.2.7

Date 2016-10-12

Description Fit dynamic group-level IRT 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. Under active development.

Maintainer James Dunham <james@jdunham.io>

License GPL-3

URL https://github.com/jamesdunham/dgo

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

Depends R (>= 3.2.2), Rcpp (>= 0.11.0), rstan (>= 2.8.1)

LazyData true

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

Suggests knitr, rmarkdown, testthat

LinkingTo StanHeaders (>= 2.8.0), rstan (>= 2.8.1), BH (>= 1.58.0), Rcpp (>= 0.11.0), RcppEigen

RcppModules stan_fit42016_09_14_mod, stan_fit42016_04_20_mod, stan_fit42015_12_16_mod

NeedsCompilation yes

Collate 'RcppExports.R' 'aggregate_item_responses.r' 'assertions.r' 'class-control.r' 'class-dgirtfit.r' 'constants.r' 'class-dgirtin.r' 'data-opinion.r' 'data-states.r' 'data-targets.r' 'require_namespace.r' 'shape.r' 'data-toy_dgirt_in.r' 'dgirt.r' 'data-toy_dgirtfit.r' 'dgmrp.r' 'dichotomize_item_responses.r' 'expand_rownames.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' 'stanmodels.R' 'validate_dgirtIn.r' 'validate_input_data.r' 'wrangle.r' 'zzz.R'

RoxygenNote 5.0.1

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

R topics documented:

dgirt — dgirt*: fit a DGIRT model*

Description

dgirt makes a call to stan with the Stan code and data for a DGIRT model.

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 = "2016_09_14")
```

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 version of the DGIRT model to use.

Details

The user will typically pass further arguments to `stan` via the `...` argument, at a minimum `iters` and `cores`.

By default `dgirt` overrides the `stan` default for its `pars` argument to specify typical DGIRT parameters of interest. `dgirt` also sets `iter_r` to `1L`.

Value

A `dgirtfit-class` object that extends `stanfit-class`.

See Also

`dgirt` expects `shaped_data` created by `shape` and returns an object of class `dgirtfit-class`. `dgirtfit-class` `shape`

`dgirtfit-class` *Class* `dgirtfit`*: a class for fitted DGIRT modelsf*

Description

Fitting a dgirt model results in a `dgirtfit` object that inherits from `rstan`'s `stanfit-class`. `rstan` methods will be dispatched (only) if a `dgirtfit` method does not exist.

Slots

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

See Also

`stanfit-class` `dgirtin-class`

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
```

```
as.array(toy_dgirtfit, pars = 'theta_bar')
as.data.frame(toy_dgirtfit, pars = 'theta_bar')
extract(toy_dgirtfit, pars = 'theta_bar')
```

`dgirtin-class` *Class* `dgirtIn`*: data prepared for modeling with* `dgirt`

Description

shape generates objects of class `dgirtIn` for modeling with `dgirt`.

Summarize DGIRT Data

Get Items Names in DGIRT Data.

Count Respondents in DGIRT Data.

Count Respondents for Items in DGIRT Data

Show Summary of DGIRT Data

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_n‘ will operate on the table passed to ‘shape‘ as ‘aggregate_data‘ instead of on the individual data.

Value

A list of item names.

See Also

‘get_item_n, get_item_names‘

Examples

```
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")
get_item_names(toy_dgirt_in)
# respondent count
get_n(toy_dgirt_in)

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

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

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

`dgirt_plot` `dgirt_plot`: *plot* `dgirtfit`-*class objects*

Description

`dgirt_plot`: plot `dgirtfit`-class objects

`dgirt_plot`: plot `data.frame` objects

`plot`: plot method for `dgirtfit`-class objects

This function plots R-hats from a dgirt model.

Usage

```
dgirt_plot(x, ...)

## S4 method for signature 'dgirtfit'
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_names, time_name, geo_name,
  y_fun = "median", y_min = "q_025", y_max = "q_975")

## S4 method for signature 'dgirtfit,missing'
plot(x, y, ...)

plot_rhats(x, ...)

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

Arguments

`x` A `dgirtfit`-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_names` Discrete grouping variables, if any, which will be used as the `color` argument in `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

```
dgirt_plot(toy_dgirtfit)
dgirt_plot(toy_dgirtfit, y_min = NULL, y_max = NULL)
p <- dgirt_plot(toy_dgirtfit)
p %+% ylab("posterior median")
data(state_year_targets)
ps <- poststratify(toy_dgirtfit, state_year_targets, strata_names =
                   c("state", "year"), aggregated_names = "race")
dgirt_plot(ps, group_names = NULL, time_name = "year", geo_name = "state")

plot(toy_dgirtfit)
```

```
plot_rhats(toy_dgirtfit)
plot_rhats(toy_dgirtfit, facet_vars = c("race", "state")) +
  scale_x_continuous(breaks = seq.int(2006, 2008))
```

---

dgmrp — dgmrp*: fit a single-issue MRP model, with hierarchical covariates*

Description

dgmrp makes a call to stan with the Stan code and data for a dgmrp model.

Usage

```
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 = "2016_12_24")
```

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 version of the DGIRT model to use.

Details

The user will typically pass further arguments to stan via the ... argument, at a minimum iters and cores.

By default dgmrp overrides the stan default for its pars argument to specify typical DGIRT parameters of interest. dgmrp also sets iter_r to 1L.

Value

A dgirtfit-class object that extends stanfit-class.

See Also

dgmrp expects shaped_data created by shape and returns an object of class dgirtfit-class. dgirtfit-class shape

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

Description

Fit dynamic group-level IRT 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.

expand_rownames expand_rownames*: expand parameter descriptions in rownames*

Description

Move rownames that describe parameters (e.g. xi[2009]) to columns.

Usage

```
expand_rownames(x, time_name, geo_name, group_names)
```

Arguments

x	A table with rownames in the format param[group1__groupK,t] or param[t].
time_name	A name for any resulting time variable.
geo_name	A name for any resulting geographic variable.
group_names	Names for any resulting group variables.

Details

It should rarely be necessary to call expand_rownames directly. But elements extracted from dgirtfit-class objects may have rownames of the format param[group1__groupK,t] for parameters indexed by group and time period, or param[t] for parameters indexed by time period. expand_rownames moves this information to columns whose names are given by the col_names argument. The rownames in their original format will appear in another column called rn.

Value

x with additional columns (see details).

See Also

dgirtfit-class

opinion — dgirt *example data: item responses*

Description

A table of item responses given by survey respondents and their characteristics. Data are from the Cooperative Congressional Election Study (CCES), 2006-2008

Usage

```
opinion
```

Format

A `data.frame` with 78,709 observations of 10 variables.

See Also

http://projects.iq.harvard.edu/cces/data

Examples

```
opinion
```

poststratify — poststratify*: reweight and aggregate estimates*

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,
  prop_name = "proportion", single_issue = "F", ...)

## S4 method for signature 'dgirtfit'
poststratify(x, target_data, strata_names,
  aggregated_names, prop_name = "proportion", single_issue = "F",
  pars = "theta_bar")

## S4 method for signature 'data.frame'
poststratify(x, target_data, strata_names,
  aggregated_names, prop_name = "proportion", single_issue = "F",
  pars = "theta_bar")
```

Arguments

x A data.frame or dgirtfit object.

target_data A table giving the proportions contributed to strata by the interaction of strata_names and aggregated_names.

strata_names Names of variables whose interaction defines population strata.

aggregated_names
Names of variables to be aggregated over in poststratification.

prop_name Name of the column in target_data that gives strata proportions.

single_issue Flag for whether DGO ran a single-issue manifest variable model. If "T", apply pnorm to convert results to response scale.

... Additional arguments to methods.

pars Selected parameter names.

Value

A table of poststratified estimates.

Examples

```
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 + race, 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"), "race")
```

---

shape   shape: *prepare data for modeling with* dgirt

Description

This function shapes various kinds of data for use in a dgirt model. Most arguments give the name or names of key variables in the data; they end in _name or _names and should be character vectors. Some others implement preprocessing and modeling choices.

Usage

```
shape(item_data, item_names, time_name, geo_name, group_names = NULL,
  weight_name, raking = NULL, survey_name, modifier_data = NULL,
  target_data = NULL, aggregate_data = NULL, aggregate_item_names = NULL,
  id_vars = NULL, ...)
```

Arguments

item_data A table in which items appear in columns and each row represents an individual's responses in some time period and local geographic area.

item_names Individual item responses. These variables should be integers or ordered factors in the data.

time_name A time variable with numeric values.

geo_name A geographic variable representing local areas.

group_names Discrete grouping variables, usually demographic. Using numeric variables is allowed but not recommended.

weight_name A variable giving survey weights.

raking A formula or list of formulas specifying the variables on which to rake survey weights.

survey_name A survey identifier.

modifier_data Table giving characteristics of local geographic areas in time periods. See details below.

target_data A table giving population proportions for groups by local geographic area and time period. See details below.

aggregate_data A table of trial and success counts by group and item. See details below.

aggregate_item_names
A subset of values of the item variable in aggregate_data, for restricting the aggregate data.

id_vars Additional variables that should be included in the result, other than those specified elsewhere.

... Further arguments for more complex models, input data, and preprocessing.

Value

An object of class dgirtIn expected by dgirt.

Modifier Data

Geographic hierarchical parameters can be modeled with modifier_data. These arguments are also required:

modifier_names: Modifiers of geographic hierarchical parameters, e.g. median household income in each local-area and time-period combination.

t1_modifier_names: Modifiers to be used instead of those in modifier_names, only in the first period.

standardize: Whether to standardize hierarchical modifier data to be zero-mean and unit-variance for performance gains. For discussion see the Stan Language Reference section "Standardizing Predictors and Outputs."

Aggregate Data

Specifying aggregate_data requires no additional arguments; instead, we make many assumptions about the data. This implementation is likely to change in the future.

aggregate_data is expected to be a long table of trial and success counts by group and item. Some variable names given for item_data are expected in the table of aggregates: group_names, geo_name, and time_name. Three fixed variable names are also expected in aggregate_data: item giving item identifiers, n_grp giving adjusted counts of item-response trials, and s_grp giving adjusted counts of item-response successes. The counts should be adjusted consistently with the transformations applied to the individual item_data.

Preprocessing

If target_data is specified shape will adjust the weighting of groups toward population targets via raking. This relies on an adaptation of rake. The additional required arguments are target_proportion_name and raking.

shape can restrict data row-wise in item_data, modifier_data, and aggregate_data to that within specified time periods (time_filter) and local geographic areas (geo_filter). Data can also be filtered for sparsity, to keep items that appear in a minimum of time periods or surveys. This is a column-wise operation. If both row-wise and column-wise restrictions are specified, shape iterates over them until they leave the data unchanged.

Target Data

target_proportion_name The variable giving population proportions for strata.

raking A formula or list of formulas specifying the variables on which to rake.

geo_filter A character vector giving values of the geographic variable. Defaults to observed values.

time_filter A numeric vector giving possible values of the time variable. Observed and unobserved time periods can be given. Defaults to observed values.

min_survey_filter An integer minimum of survey appearances for included items. Defaults to 1.

min_t_filter An integer minimum of time period appearances for included items. Defaults to 1.

Modeling Choices

Optional. Most arguments like this one are now in the dgirt signature, but constant_item affects the shape of the data. It may move to dgirt in the future.

constant_item Whether item difficulty parameters should be constant over time. Default TRUE.

See Also

dgirtin-class dgirtfit-class

Examples

```
# model individual item responses
data(opinion)
opinion$respondent = 1:nrow(opinion)
shaped_responses <- shape(opinion,
                          item_names = "Q_cces2006_gaymarriageamendment",
                          time_name = "year",
                          geo_name = "state",
                          group_names = "race",
                          weight_name = "weight",
                          survey_name = "source",
                          id_vars = 'respondent')
# summarize result)
summary(shaped_responses)
```

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

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

Description

print method for dgirtfit-class objects

get_elapsed_time: extract chain run times from dgirtfit-class objects

summary method for dgirtfit-class objects

summarize method for dgirtfit-class objects

as.data.frame method for dgirtfit-class objects

rhats: extract split R-hats from dgirtfit-class objects

Usage

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

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

print.dgirtfit(x, ...)

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

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

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

summarize(x, ...)

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

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

rhats(x, ...)

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

Arguments

`object`	A `dgirtfit-class` object
`x`	A `dgirtfit-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

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

`states` `dgirt` *example data: state demographics*

Description

A table giving demographic characteristics of American states 2006-2008.

Usage

```
states
```

Format

A `data.frame` with 153 observations of 7 variables.

Examples

```
states
```

`state_year_targets` `dgirt` *example data: state-year population targets*

Description

A table giving the population proportions of state-years by race, 2006-2008.

Usage

```
state_year_targets
```

Format

A `data.frame` with 459 observations of 4 variables.

Examples

```
head(state_year_targets)
```

`targets` `dgirt` *example data: U.S. population targets*

Description

A table giving U.S. population proportions by year for combinations of demographic variables. Data are from the U.S. Census. When using this table to adjust survey weights via the `target_data` arguments of `shape`, note that its proportions sum to to 1 within years.

Usage

```
targets
```

Format

A `data.frame` with 18,360 observations of 6 variables.

See Also

`http://www.census.gov/`

Examples

```
targets
```

toy_dgirtfit *Class `dgirtfit`: a minimal example object*

Description

`dgirt` returns a `dgirtfit`-class object that extends `stanfit-class` from `rstan`. `toy_dgirtfit` is a minimal `dgirtfit` object, mostly for use in development.

Usage

```
toy_dgirtfit
```

Format

A `dgirtfit-class` object.

Examples

```
toy_dgirtfit
```

toy_dgirt_in *Class `dgirtIn`: a minimal example object*

Description

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

Usage

```
toy_dgirt_in
```

Format

A `dgirtin-class` object.

Examples

```
toy_dgirt_in
```

Index