Package 'edstan'

January 31, 2017

Type Package

Title Stan Models for Item Response Theory

Version 1.0.6

Date 2017-01-30

Author Daniel C. Furr

Maintainer Daniel C. Furr <danielcfurr@berkeley.edu>

Description Provides convenience functions and pre-programmed Stan models related to item response theory. Its purpose is to make fitting common item response theory models using Stan easy.

License BSD_3_clause + file LICENSE

Depends R (>= 3.0.2), rstan (>= 2.10)

Imports ggplot2

Suggests knitr, rmarkdown

VignetteBuilder knitr

LazyData true

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2017-01-31 06:51:10

R topics documented:

edstan-package *Stan for item response theory*

Description

edstan attempts to make easy the fitting of standard item response theory models using **rstan**.

Details

A user will generally want to use the following functions (in order) to fit a model:

1. `irt_data` to format the data,
2. `irt_stan` to fit a model, and
3. `print_irt_stan` to view some results.

Additionally, `labelled_integer` is some times helpful for data formatting and `stan_columns_plot` creates a plots of convergence and other statistics by parameter vector. The package also includes six Stan models (see `irt_stan` for a list) and two example datasets (`aggression` and `spelling`).

It is expected that once a user is comfortable fitting pre-defined **edstan** models, they will write their own Stan models and fit them with `stan`, for which `irt_stan` is a wrapper.

See Also

Case studies for each of the **edstan** models have been published.

Rasch and two-parameter logistic models

http://mc-stan.org/documentation/case-studies/rasch_and_2pl.html

(Generalized) partial credit model

http://mc-stan.org/documentation/case-studies/pcm_and_gpcm.html

(Generalized) rating scale model

http://mc-stan.org/documentation/case-studies/rsm_and_grsm.html

aggression *Verbal aggression data*

Description

Item response data regarding verbal agression from 316 persons and 24 items. Participants were instructed to imagine four frustrating scenarios in which either another or oneself is to blame. For each scenario, they responded "yes", "perhaps", or "no" regarding whether they would react by cursing, scolding, and shouting. They also responded whether they would want to engage in those three behaviors, resulting in a total six items per scenario. An example item is, "A bus fails to stop for me. I would want to curse."

Usage

```
aggression
```

Format

A long-form data.frame (one row per item response) with the following columns:

person Integer person identifier.

item Integer item identifier.

poly Original, polytomous response. 0 indicates "no", 1 "perhaps", and 3 "yes".

dich Dichotomized response. 0 indicates "no" and 1 indicates "perhaps" or "yes".

description Brief description of the item.

anger Trait anger score for a person.

male Indicator for whether person is male.

do Indicator for whether item concerns actually doing the behavior instead of wanting to do it.

other Indicator for whether item concerns another person being to blame instead of self to blame.

scold Indicator for whether item concerns scolding behavior instead of cursing or shouting.

shout Indicator for whether item concerns shouting behavior instead of cursing or scolding.

Source

Vansteelandt, K. (2000). Formal models for contextualized personality psychology. Unpublished doctoral dissertation. K. U. Leuven, Belgium.

References

De Boeck, P. and Wilson, M. (2004) *Explanatory Item Response Models*. New York: Springer.

`irt_data` *Create a Stan data list from an item response matrix or from long-form data.*

Description

Create a Stan data list from an item response matrix or from long-form data.

Usage

```
irt_data(response_matrix = matrix(), y = integer(), ii = integer(),
  jj = integer(), covariates = data.frame(), formula = ~1)
```

Arguments

`response_matrix`
: An item response matrix. Columns represent items and rows represent persons. NA may be supplied for missing responses. The lowest score for each item should be 0, with exception to rating scale models. `y`, `ii`, and `jj` should not be supplied if a response matrix is given.

`y`
: A vector of scored responses for long-form data. The lowest score for each item should be 0, with exception to rating scale models. NAs are not permitted, but missing responses may simply be ommitted instead. Required if `response_matrix` is not supplied.

`ii`
: A vector indexing the items in `y`. This must consist of consecutive integers starting at 1. `labelled_integer` may be used to create a suitable vector. Required if `response_matrix` is not supplied.

`jj`
: A vector indexing the persons in `y`. This must consist of consecutive integers starting at 1. `labelled_integer` may be used to create a suitable vector. Required if `response_matrix` is not supplied.

`covariates`
: An optional data frame containing (only) person-covariates. It must contain one row per person or be of the same length as `y`, `ii`, and `jj`. If it contains one row per person, it must be in the same order as the response matrix (or `unique(jj)`). If it has a number of columns equal to the length of `y`, `ii`, and `jj`, it must be in the same order as `jj` (for example, it may be a subset of columns from the same data frame that contains `y`, `ii`, and `jj`).

`formula`
: An optional formula for the latent regression that is applied to `covariates`. The left side should be blank (for example, `~ v1 + v2`). By default it includes only a model intercept, interpretable as the mean of the person distribution. If set to `NULL`, then `covariates` is used directly as the design matrix for the latent regression.

Value

A data list suitable for `irt_stan`.

See Also

See `labelled_integer` for a means of creating appropriate inputs for `ii` and `jj`. See `irt_stan` to fit a model to the data list.

Examples

```
# For a response matrix ("wide-form" data) with person covariates:
spelling_list <- irt_data(response_matrix = spelling[, 2:5],
                          covariates = spelling[, "male", drop = FALSE],
                          formula = ~ 1 + male)

# Alternatively, the same may be created by:
W <- cbind(intercept = 1, spelling[, "male"])
spelling_list <- irt_data(response_matrix = spelling[, 2:5],
                          covariates = W,
```

```
                          formula = NULL)

# For long-form data (one row per item-person pair):
agg_list_1 <- irt_data(y = aggression$poly,
                       ii = aggression$item,
                       jj = aggression$person)

# Add a latent regression and use labelled_integer() with the items
agg_list_2 <- irt_data(y = aggression$poly,
                       ii = labelled_integer(aggression$description),
                       jj = aggression$person,
                       covariates = aggression[, c("male", "anger")],
                       formula = ~ 1 + male*anger)
```

`irt_stan` *Estimate an item response model with Stan*

Description

Estimate an item response model with Stan

Usage

```
irt_stan(data_list, model = "", ...)
```

Arguments

`data_list`	A Stan data list created with `irt_data`.
`model`	The file name for one of the provided .stan files, or alternatively, a user-created .stan file that accepts `data_list` as input data. The ".stan" file extension may be omitted. Defaults to either "rasch_latent_reg.stan" or "pcm_latent_reg.stan".
`...`	Additional options passed to `stan`. The usual choices are `iter` for the number of iterations and `chains` for the number of chains.

Details

The following table lists the models inlcuded in **edstan** along with the associated *.stan* files. The file names are given as the `model` argument.

Model	**File**
Rasch	*rasch_latent_reg.stan*
Partial credit	*pcm_latent_reg.stan*
Rating Scale	*rsm_latent_reg.stan*
Two-parameter logistic	*2pl_latent_reg.stan*
Generalized partial credit	*gpcm_latent_reg.stan*
Generalized rating Scale	*grsm_latent_reg.stan*

Three simplified models are also available: *rasch_simple.stan*, *pcm_simple.stan*, *rsm_simple.stan*. These are (respectively) the Rasch, partial credit, and rating scale models omitting the latent regression. There is no reason to use these instead of the models listed above, given that the above models allow for rather than require the inclusion of covariates for a latent regression. Instead, the purpose of the simplified models is to provide a straightforward starting point researchers who wish to craft their own Stan models.

Value

A stanfit-class object.

See Also

See stan, for which this function is a wrapper, for additional options. See irt_data and labelled_integer for functions that facilitate creating a suitable data_list. See print_irt_stan and print.stanfit for ways of getting tables summarizing parameter posteriors.

Examples

```
# List the Stan models included in edstan
folder <- system.file("extdata", package = "edstan")
dir(folder, "\\.stan$")

# List the contents of one of the .stan files
rasch_file <- system.file("extdata/rasch_latent_reg.stan",
                          package = "edstan")
cat(readLines(rasch_file), sep = "\n")

## Not run:
# Fit the Rasch and 2PL models on wide-form data with a latent regression

spelling_list <- irt_data(response_matrix = spelling[, 2:5],
                          covariates = spelling[, "male", drop = FALSE],
                          formula = ~ 1 + male)

rasch_fit <- irt_stan(spelling_list, iter = 300, chains = 4)
print_irt_stan(rasch_fit, spelling_list)

twopl_fit <- irt_stan(spelling_list, model = "2pl_latent_reg.stan",
                      iter = 300, chains = 4)
print_irt_stan(twopl_fit, spelling_list)


# Fit the rating scale and partial credit models without a latent regression

agg_list_1 <- irt_data(y = aggression$poly,
                       ii = labelled_integer(aggression$description),
                       jj = aggression$person)

fit_rsm <- irt_stan(agg_list_1, model = "rsm_latent_reg.stan",
                    iter = 300, chains = 4)
print_irt_stan(fit_rsm, agg_list_1)
```

```
fit_pcm <- irt_stan(agg_list_1, model = "pcm_latent_reg.stan",
                    iter = 300, chains = 4)
print_irt_stan(fit_pcm, agg_list_1)


# Fit the generalized rating scale and partial credit models including
# a latent regression

agg_list_2 <- irt_data(y = aggression$poly,
                       ii = labelled_integer(aggression$description),
                       jj = aggression$person,
                       covariates = aggression[, c("male", "anger")],
                       formula = ~ 1 + male*anger)

fit_grsm <- irt_stan(agg_list_2, model = "grsm_latent_reg.stan",
                     iter = 300, chains = 4)
print_irt_stan(fit_grsm, agg_list_2)

fit_gpcm <- irt_stan(agg_list_2, model = "gpcm_latent_reg.stan",
                     iter = 300, chains = 4)
print_irt_stan(fit_grsm, agg_list_2)

## End(Not run)
```

`labelled_integer` *Transform a vector into consecutive integers*

Description

Transform a vector into consecutive integers

Usage

```
labelled_integer(x = vector())
```

Arguments

x	A vector, which may be numeric, string, or factor.

Value

A vector of integers corresponding to entries in `x`. The lowest value will be 1, and the greatest value will equal the number of unique elements in `x`. The elements of the recoded vector are named according to the original values of `x`. The result is suitable for the `ii` and `jj` options for `irt_data`.

Examples

```
x <- c("owl", "cat", "pony", "cat")
labelled_integer(x)

y <- as.factor(x)
labelled_integer(y)

z <- rep(c(22, 57, 13), times = 2)
labelled_integer(z)
```

`print_irt_stan` *View a table of selected parameter posteriors after using* `irt_stan`

Description

View a table of selected parameter posteriors after using `irt_stan`

Usage

```
print_irt_stan(fit, data_list = NULL, ...)
```

Arguments

`fit`	A `stanfit-class` object created by `irt_stan`.
`data_list`	An optional Stan data list created with `irt_data`. If provided, the printed posterior summaries for selected parameters are grouped by item. Otherwise, ungrouped results are provided, which may be preferred, for example, for the Rasch or rating scale models.
`...`	Additional options passed to `print`.

Examples

```
# Make a suitable data list:
spelling_list <- irt_data(response_matrix = spelling[, 2:5],
                          covariates = spelling["male", drop = FALSE],
                          formula = ~ 1 + male)

## Not run:
# Fit a latent regression  2PL
twopl_fit <- irt_stan(spelling_list, model = "2pl_latent_reg.stan",
                      iter = 300, chains = 4)

# Get a table of parameter posteriors
print_irt_stan(twopl_fit, spelling_list)
# Or
print_irt_stan(twopl_fit)

## End(Not run)
```

spelling *Spelling data*

Description

Item response data regarding student spelling performance on four words: *infidelity*, *panoramic*, *succumb*, and *girder*. The sample includes 284 male and 374 female undergraduate students from the University of Kansas. Each item was scored as either correct or incorrect.

Usage

```
spelling
```

Format

A wide-form data.frame (one row per person) with the following columns:

male Indicator for whether person is male.

infidelity Indicator for whether person spelled *infidelity* correctly.

panoramic Indicator for whether person spelled *panoramic* correctly.

succumb Indicator for whether person spelled *succumb* correctly.

girder Indicator for whether person spelled *girder* correctly.

Source

Thissen, D., Steinberg, L. and Wainer, H. (1993). Detection of Differential Item Functioning Using the Parameters of Item Response Models. In *Differential Item Functioning*, edited by Holland. P. and Wainer, H., 67-114. Hillsdale, NJ: Lawrence Erlbaum.

stan_columns_plot *View a plot of summary statistics after using* irt_stan

Description

View a plot of summary statistics after using irt_stan

Usage

```
stan_columns_plot(fit, stat = "Rhat", ...)
```

Arguments

fit	A stanfit-class object created by irt_stan or stan.
stat	A string for the statistic from the summary method for a stanfit object to plot. The default is "Rhat" but could, for example, be "mean" or "n_eff".
...	Additional options (such as pars or use_cache), passed to the summary method for a stanfit object. Not required.

Value

A ggplot object.

See Also

See stan_rhat, which provides a histogram of Rhat statistics.

Examples

```
# Make a suitable data list:
spelling_list <- irt_data(response_matrix = spelling[, 2:5],
                          covariates = spelling[, "male", drop = FALSE],
                          formula = ~ 1 + male)

## Not run:
# Fit a latent regression  2PL
twopl_fit <- irt_stan(spelling_list, model = "2pl_latent_reg.stan",
                      iter = 300, chains = 4)

# Get a plot showing Rhat statistics
rhat_columns(twopl_fit)

# Get a plot showing number of effective draws
rhat_columns(twopl_fit, stat = "n_eff")

## End(Not run)
```

Index