

# Package ‘idealstan’

February 9, 2018

**Type** Package

**Title** Bayesian IRT Ideal Point Models with 'Stan'

**Version** 0.2.7

**Date** 2018-02-07

**BugReports** <https://github.com/saudiwin/idealstan/issues>

**Description** Offers item-response theory (IRT) ideal-point scaling/dimension reduction methods that incorporate additional response categories and missing/censored values, including absences and abstentions, for roll call voting data (or any other kind of binary or ordinal item-response theory data). Full and approximate Bayesian inference is done via the 'Stan' engine ([www.mc-stan.org](http://www.mc-stan.org)).

**License** GPL

**Depends** R (>= 3.1), Rcpp (>= 0.12.7)

**Imports** rstan (>= 2.13.2), rstantools (>= 1.1.0), methods (>= 3.3.1), dplyr, tidyr, stringr, bayesplot, ggplot2, lazyeval, rlang, shinystan

**Suggests** pscl, loo, knitr, rmarkdown

**LinkingTo** StanHeaders (>= 2.13.1), rstan (>= 2.13.2), BH (>= 1.62.0.1), Rcpp (>= 0.12.7), RcppEigen (>= 0.3.2.9.0)

**RoxygenNote** 6.0.1

**LazyData** false

**VignetteBuilder** knitr

**SystemRequirements** GNU make

**NeedsCompilation** yes

**Author** Robert Kubinec [aut, cre],  
Jonah Gabry [ctb],  
Ben Goodrich [ctb],  
Trustees of Columbia University [cph]

**Maintainer** Robert Kubinec <[rmk7xy@virginia.edu](mailto:rmk7xy@virginia.edu)>

**Repository** CRAN

**Date/Publication** 2018-02-09 11:16:51 UTC

## R topics documented:

idealdata-class . . . . .	2
idealstan . . . . .	3
idealstan-class . . . . .	3
id_estimate . . . . .	3
id_extract . . . . .	8
id_extract,idealstan-method . . . . .	9
id_log_lik . . . . .	10
id_log_lik,idealstan-method . . . . .	10
id_make . . . . .	11
id_plot . . . . .	13
id_plot,idealstan-method . . . . .	13
id_plot_all_hist . . . . .	14
id_plot_compare . . . . .	15
id_plot_legis . . . . .	15
id_plot_ppc . . . . .	17
id_plot_ppc,idealstan-method . . . . .	18
id_plot_rhats . . . . .	18
id_plot_sims . . . . .	19
id_post_pred . . . . .	19
id_post_pred,idealstan-method . . . . .	20
id_sim_coverage . . . . .	20
id_sim_gen . . . . .	21
id_sim_resid . . . . .	22
id_sim_rmse . . . . .	22
launch_shinystan . . . . .	23
launch_shinystan,idealstan-method . . . . .	23
release_questions . . . . .	24
senate114 . . . . .	24
senate114_fit . . . . .	24
summary,idealstan-method . . . . .	25
<b>Index</b>	<b>26</b>

---

idealdata-class	<i>Data and Identification for id_estimate</i>
-----------------	--

---

### Description

idealdata objects contain the relevant legislator/bill (person/item) matrix of data along with slots containing information about the kind of identification used in the estimation.

### See Also

[id\\_make](#) to create an idealdata object suitable for estimation with id\_estimate.

---

idealstan	idealstan package
-----------	-------------------

---

## Description

R Interface to Stan for Item-Response Theory Ideal Point Models

## Details

See the README on [GitHub](#)

## References

1. Clinton, J., Jackman, S., & Rivers, D. (2004). The Statistical Analysis of Roll Call Data. *The American Political Science Review*, 98(2), 355-370. doi:10.1017/S0003055404001194
2. Bafumi, J., Gelman, A., Park, D., & Kaplan, N. (2005). Practical Issues in Implementing and Understanding Bayesian Ideal Point Estimation. *Political Analysis*, 13(2), 171-187. doi:10.1093/pan/mpi010

---

idealstan-class	Results of <code>id_estimate</code> function
-----------------	--

---

## Description

The idealstan objects store the results of estimations carried out by the `id_estimate` function. These objects include the full results of Bayesian sampling performed by the `stan` function in the `rstan` package.

---

id_estimate	Estimate an idealstan model
-------------	-----------------------------

---

## Description

This function will take a pre-processed idealdata vote/score matrix and run one of the available IRT models on the data using Stan's MCMC engine.

**Usage**

```
id_estimate(idealdata = NULL, model_type = 2, use_subset = FALSE,
  sample_it = FALSE, subset_group = NULL, subset_person = NULL,
  sample_size = 20, nchains = 4, niters = 2000, use_vb = FALSE,
  nfix = 3, restrict_params = "person", pin_vals = NULL,
  restrict_ind_high = NULL, restrict_ind_low = NULL,
  restrict_type = "constrain_twoway", fixtype = "vb",
  warmup = floor(niters/2), ncores = 4, auto_id = FALSE,
  discrim_reg_sd = 3, discrim_miss_sd = 3, person_sd = 1,
  diff_reg_sd = 4, diff_miss_sd = 4, restrict_sd = 4,
  restrict_low_bar = 0, restrict_high_bar = 0, restrict_alpha = 4,
  restrict_beta = 1, ...)
```

**Arguments**

idealdata	An object produced by the <a href="#">id_make</a> containing a score/vote matrix for use for estimation & plotting
model_type	An integer reflecting the kind of model to be estimated. See below.
use_subset	Whether a subset of the legislators/persons should be used instead of the full response matrix
sample_it	Whether or not to use a random subsample of the response matrix. Useful for testing.
subset_group	If person/legislative data was included in the <a href="#">id_make</a> function, then you can subset by any value in the \$group column of that data if use_subset is TRUE.
subset_person	A list of character values of names of persons/legislators to use to subset if use_subset is TRUE and person/legislative data was included in the <a href="#">id_make</a> function with the required \$person.names column
sample_size	If sample_it is TRUE, this value reflects how many legislators/persons will be sampled from the response matrix
nchains	The number of chains to use in Stan's sampler. Minimum is one. See <a href="#">stan</a> for more info.
niters	The number of iterations to run Stan's sampler. Shouldn't be set much lower than 500. See <a href="#">stan</a> for more info.
use_vb	Whether or not to use Stan's variational Bayesian inference engine instead of full Bayesian inference. Pros: it's much faster. Cons: it's not quite as accurate. See <a href="#">vb</a> for more info.
nfix	An integer specifying the number of parameters to constrain (for both high and low) if fixtype is set to 'vb'
restrict_params	Determines which parameters in the model are used for identification. If fixtype is 'vb', can be either 'person' (person/legislator ideal points) or 'items' (discrimination parameters for items/bills). Otherwise, the value must be one of the available classes of parameters: 'person' for persons/legislators, 'discrim_miss' for the missing-inflated item/bill discrimination parameters, or 'discrim_reg' for the non-inflated item/bill discrimination parameters.

pin_vals	If <code>fixtype</code> is set to 'pinned', then a vector of numeric values of which to pin the legislators/persons or bills/items should be given.
restrict_ind_high	If <code>fixtype</code> is not "vb", the particular indices of legislators/persons or bills/items to constrain high
restrict_ind_low	If <code>fixtype</code> is not "vb", the particular indices of legislators/persons or bills/items to constrain low. (Note: not used if values are pinned).
restrict_type	Determines whether constraints will only be set on one direction 'constrain_oneway' or both high/low constrains 'constrain_twoway'. Ignored if <code>fixtype</code> is equal to 'pinned'.
fixtype	Sets the particular kind of identification used on the model, could be one of 'vb', 'constrained' or 'pinned'. See details for more information.
warmup	The number of iterations to use to calibrate Stan's sampler on a given model. Shouldn't be less than 100. See <a href="#">stan</a> for more info.
ncores	The number of cores in your computer to use for parallel processing in the Stan engine. See <a href="#">stan</a> for more info.
auto_id	If TRUE, a variational Bayesian version of your model will be run using <a href="#">vb</a> in a non-identified form. Depending on what is set via <code>restrict_type</code> and <code>restrict_params</code> , bills/items or persons/legislators will be used to constrain and identify the model.
discrim_reg_sd	Set the prior standard deviation of the bimodal prior for the discrimination parameters for the non-inflated model.
discrim_miss_sd	Set the prior standard deviation of the bimodal prior for the discrimination parameters for the inflated model.
person_sd	Set the prior standard deviation for the legislators (persons) parameters
diff_reg_sd	Set the prior standard deviation for the bill (item) intercepts for the non-inflated model.
diff_miss_sd	Set the prior standard deviation for the bill (item) intercepts for the inflated model.
restrict_sd	Set the prior standard deviation for constrained parameters
restrict_low_bar	Set the constraint threshold for constrained parameters (parameter must be lower than this bar and no greater than zero)
restrict_high_bar	Set the constraint threshold for constrained parameters (parameter must be higher than this bar and no less than zero)
restrict_alpha	This is the scale (alpha) parameter passed to the gamma prior if exactly two item/person parameters are constrained, each high or low. The gamma prior pushes these two polar parameters apart. A higher value will push these two poles farther apart, which will help identification.
restrict_beta	The beta (shape) parameter passed to the gamma prior if exactly two item/person parameters are constrained, each high or low. The gamma prior pushes these two polar parameters apart.

... Additional parameters passed on to Stan's sampling engine. See [stan](#) for more information.

## Details

To run an IRT ideal point model, you must first pre-process your data using the [id\\_make](#) function. Be sure to specify the correct options for the kind of model you are going to run: if you want to run an ordinal and/or an inflated model, the data needs to be processed differently. As of this version of [idealstan](#), the following model types are available:

1. IRT 2-PL (binary response) ideal point model, no absence inflation
2. IRT 2-PL model (binary response) with absence inflation
3. Ordinal IRT (rating scale) model incorporating abstentions as middle category, no absence inflation
4. Ordinal IRT (rating scale) model incorporating abstentions as middle category with absence inflation

Additional models are available but have not yet been tested. You can find them by searching the included [stan](#) files for more info.

## Value

A fitted [idealstan](#) object that contains posterior samples of all parameters either via full Bayesian inference or a variational approximation if `use_vb` is set to `TRUE`. This object can then be passed to the plotting functions for further analysis.

## Identification

Identifying IRT models is challenging, and ideal point models are still more challenging because the discrimination parameters are not constrained. As a result, more care must be taken to obtain estimates that are the same regardless of starting values. The parameter `fixtype` enables you to change the type of identification used. The default, `'vb'`, does not require any further information from you in order for the model to be fit. In this version of identification, an unidentified model is run using variational Bayesian inference (see [vb](#)). If `restrict_type` is set to `'constrain_oneway'`, then the `nfix` highest legislators/persons (if `restrict_params` is `'person'`) or bills/items (if `restrict_params` equals `'items'`) are used to constrain and identify the model. If `restrict_type` is set to `'constrain_twoway'`, then `nfix` highest and lowest legislators/persons or bills/items are used to constrain and identify the model. In addition, if `fixtype='vb'` is used, `auto_id` can be set to `TRUE`. This will run additional variational Bayesian models using the identification achieved and will see if the signs of the estimated parameters are at least 90 percent in the same direction. If so, the model is considered identified. If not, the function will re-run and will increase `nfix` by one to see if that will identify the model, ad infinitum. If `fixtype` is set to `'constrained'`, then identification is achieved by constraining a specified list of legislators/persons or bills/items. If `restrict_type` is `'constrain_oneway'`, then the indices of all constrained legislators/persons or bills/items should be included as the row or column indices of these parameters in the response matrix as a vector in `restrict_ind_high`. If `restrict_type` is `'constrain_twoway'`, then the indices of high constrained parameters should go in `restrict_ind_high` and the indices of low constrained parameters in `restrict_ind_low`. The numbers of high and low constrained parameters should be equal. To pick the parameter to constrain, set `restrict_params` to `'person'`



```

        ncores=2,
        nchains=2)

# We can now see how well the model recovered the true parameters

id_sim_coverage(bin_irt_2pl_abs_est) %>%
  bind_rows(.id='Parameter') %>%
  ggplot(aes(y=avg,x=Parameter)) +
  stat_summary(fun.args=list(mult=1.96)) +
  theme_minimal()

## End(Not run)

# In most cases, we will use pre-existing data
# and we will need to use the id_make function first
# We will use the full rollcall voting data
# from the 114th Senate as a rollcall object

data('senate114')

# Running this model will take at least a few minutes, even with
# variational inference (use_vb=T) turned on
## Not run:

to_idealstan <- id_make(score_data = senate114,
                       ordinal = FALSE,
                       include_pres=FALSE)

sen_est <- id_estimate(senate_data,
                      model_type = 2,
                      use_vb = TRUE,
                      restrict_type='constrain_oneway',
                      restrict_params='person',
                      restrict_ind_high = which(row.names(senate114$votes[-1,])=='SASSE (R NE)'),
                      auto_id=FALSE,
                      fixtype='constrained')

# After running the model, we can plot
# the results of the person/legislator ideal points

id_plot_legis(sen_est)

## End(Not run)

```

---

id\_extract

*Generic Method for Extracting Posterior Samples*


---

### Description

This is a generic function.



**Usage**

```
id_extract(object, ...)
```

**Arguments**

object	A fitted idealstan object
...	Other arguments passed on to underlying functions

---

```
id_extract,idealstan-method
```

*Extract [stan](#) joint posterior distribution from idealstan object*

---

**Description**

This convenience function allows you to extract the underlying [rstan](#) posterior estimates for the full parameters estimates of the idealstan model object. See [extract](#) for the underlying function and more options.

You can use this function to access a matrix or array of the full posterior estimates of each of the parameters in an idealstan object. There are available options to pick certain parameters of the model, such as the person (legislator) ideal points or item (bill) discrimination scores. Alternatively, you can leave the `extract_type` option blank and receive a list of all of the available parameters. Please note that the list of parameters do not have particularly informative names.

All parameters are returned in the order in which they were input into the [id\\_make](#) function.

**Usage**

```
## S4 method for signature 'idealstan'
id_extract(object, extract_type = "persons", ...)
```

**Arguments**

object	A fitted idealstan object (see <a href="#">id_estimate</a> )
extract_type	Can be one of 'persons' for person/legislator ideal points, 'reg_discrim' for non-inflated item (bill) discrimination scores, 'reg_diff' for non-inflated item (bill) difficulty scores, 'miss_discrim' for inflated item (bill) discrimination scores, and 'miss_diff' for inflated item (bill) difficulty scores.
...	Any additional arguments passed on to the <a href="#">extract</a> function.

---

id\_log\_lik

*Generic Method for Extracting Log Likelihood from Stan Objects*


---

**Description**

This function is a generic that is used to match the functions used with `loo` to calculate Bayesian information criteria on models.

**Usage**

```
id_log_lik(object, ...)
```

**Arguments**

object	A fitted idealstan object
...	Other arguments passed onto underlying functions

---

id\_log\_lik,idealstan-method

*Extract Log-Likelihood of the Posterior*


---

**Description**

This function returns a matrix of an S by N matrix of the log density of posterior draws from a fitted idealstan model, where S is the size of the posterior sample and N is the total number of parameters in the idealstan model. This matrix can then be used to fit an information criterion to assess model fit, see the `loo` package for details.

**Usage**

```
## S4 method for signature 'idealstan'
id_log_lik(object, ..., draws = 100,
  sample_scores = NULL)
```

**Arguments**

object	A fitted idealstan object
...	Other arguments passed on to underlying function (currently unused)
draws	The number of draws to use from the total number of posterior draws (default is 100).
sample_scores	In addition to reducing the number of posterior draws used to calculate the posterior predictive distribution, you can sample from the scores/votes themselves. To do so, set <code>sample_scores</code> to the number of scores/votes to sample.

---

id_make	<i>Create data to run IRT model</i>
---------	-------------------------------------

---

### Description

To run an IRT model using `idealstan`, you must first process your data using the `id_make` function.

### Usage

```
id_make(score_data = NULL, simul_data = NULL, person_cov = NULL,
        item_cov = NULL, item_cov_miss = NULL, person_data = NULL,
        item_data = NULL, miss_val = NA, high_val = 3L, low_val = 1L,
        middle_val = 2L, ordinal = TRUE, time = NULL,
        outcome_label_type = "votes", exclude_level = NULL, inflate = TRUE,
        simulation = FALSE, include_pres = FALSE)
```

### Arguments

<code>score_data</code>	Your person-item (legislator-bill) matrix in which persons (legislators) are in rows and items (bills) are in columns. The cells of the matrix should contain consecutive integers $1, \dots, K$ for ordinal $K$ outcomes or integers $[0, 1]$ for a binary outcome in which 0 equals no (or incorrect) and 1 equals yes (or correct). If absences/missing data are included, they should be coded as the highest category $K + 1$ for ordinal outcomes and 2 for binary outcomes.
<code>simul_data</code>	Optionally, data that has been generated by the <code>id_sim_gen</code> function.
<code>person_cov</code>	A matrix of person (legislator) covariates for hierarchical modeling. If multiple time points are used, then should be a multi-column matrix with one column for each time point.
<code>item_cov</code>	A one-column matrix of item (bill) covariates for hierarchical modeling of item (bill) discrimination parameters in the non-inflated model, i.e., non-missing model
<code>item_cov_miss</code>	A one-column matrix of item (bill) covariates for hierarchical modeling of item (bill) discrimination parameters in the inflated model (missing data model)
<code>person_data</code>	An optional data frame of information about the persons (legislators). Should include a <code>person.names</code> column containing names of persons (legislators) and optionally a <code>group</code> column with names of any groupings of the persons (legislators), such as parties or blocs (used for visualization).
<code>item_data</code>	An optional data frame of item/bill labels and other information used for visualization. Should include <code>item.names</code> column containing item/bill labels; other columns are optional.
<code>miss_val</code>	The value (numeric or character) that indicate missing data/absences in the data. If there are multiple possible values, pass along a numeric or character vector of all such values. If missing data is coded as <code>NA</code> , simply leave this parameter at the default, <code>NA</code> .

high_val	The value (numeric or character) that indicate the highest ordinal outcome possible, such as yes in a vote dataset or correct in a test examination. If there are multiple possible values, pass along a numeric or character vector of all such values.
low_val	The value (numeric or character) that indicates the lowest ordinal outcome possible, such as no votes in a vote dataset or incorrect in a test examination. If there are multiple possible values, pass along a numeric or character vector of all such values.
middle_val	The value (numeric or character) that indicate values between the lowest and highest categories, such as abstention in voting data or "Neither Agree nor Disagree" in likert scales. If there are multiple possible values, pass along a numeric or character vector of all such values in correct order (lower to higher values). If there are no middle values (binary outcome), set to NULL.
ordinal	Whether or not the data contain ordinal responses. If TRUE, middle values/abstentions are used as a middle category in constructing the outcome. Otherwise the response is assumed to be binary (yes/no) or (correct/incorrect).
time	An optional integer vector of length equal to the number of columns in the response matrix (i.e., the number of items or bills) that indicates for each item/bill which time point that item/bill belongs to. This time vector is only used for models in which person/legislator ideal points are allowed to vary over time.
outcome_label_type	Whether to use pre-set labels for the outcome values. If set to bills, the default, will label the outcome as c('No', 'Abstain', 'Yes', 'Absent') if there are 3 possible categories. Otherwise, either pass NULL to this option to use integers for labels or a character vector equal to the number of categories in the outcome. Used for visualization.
exclude_level	A vector of any values that should be treated as NA in the response matrix. Unlike the middle_val parameter, these values will be dropped from the data before estimation rather than modeled explicitly.
inflate	If TRUE, the score matrix is set up to enable modeling of missing data/absences (miss_val) as an inflation model in <a href="#">id_estimate</a>
simulation	If TRUE, simulated values are saved in the idealdata object for later plotting with the <a href="#">id_plot_sims</a> function
include_pres	If FALSE and score_data is a rollcall object, drop the first row which often represents tiebreaker votes cast by the Vice President in the US Senate.

## Value

A idealdata object that can then be used in the [id\\_estimate](#) function to fit a model.

## Examples

```
# You can either use a pscl rollcall object or a vote/score matrix
# where persons/legislators are in the rows
# and items/bills are in the columns

library(dplyr)
```

```
# First, using a rollcall object with the 114th Senate's rollcall votes:

data('senate114')

to_idealstan <- id_make(score_data = senate114,
ordinal = FALSE,
include_pres=FALSE)
```

---

id\_plot *Generic Function for Plotting idealstan objects*

---

### Description

This generic function will run all the plotting functions associated with fitted idealstan objects.

### Usage

```
id_plot(object, ...)
```

### Arguments

object	An idealstan object
...	Other options passed onto the underlying plot function

---

id\_plot,idealstan-method  
*Plot Results of [id\\_estimate](#)*

---

### Description

This function allows you to access the full range of plotting options for fitted idealstan models.

### Usage

```
## S4 method for signature 'idealstan'
id_plot(object, plot_type = "legislators", ...)
```

### Arguments

object	A fitted idealstan object
plot_type	Specify the plot as a character string. Currently 'legislators' for legislator/person ideal point plot and 'histogram' for a histogram of model estimates for given parameters.
...	Additional arguments passed on to the underlying functions. See individual function documentation for details.

**Details**

id\_plot is a wrapper function that can access the various plotting functions available in the idealstan package. Currently, the options are limited to a plot of legislator/person ideal points with bills/item midpoints as an optional overlay. Additional plots will be available in future versions of idealstan.

**Value**

A `ggplot` object

**See Also**

[id\\_plot\\_legis](#) for a legislator/person ideal point plot, [id\\_plot\\_all\\_hist](#) for a standard histogram plot, [id\\_plot\\_compare](#) for an ideal point plot of two different models of the same data, [id\\_plot\\_rhats](#) for a histogram of Rhat values, [id\\_plot\\_sims](#) for plotting true versus estimated values, [id\\_estimate](#) for how to estimate an idealstan object.

---

id_plot_all_hist	<i>Density plots of Posterior Parameters</i>
------------------	--

---

**Description**

This function produces density plots of the different types of parameters in an idealstan model: item (bill) difficulty and discrimination parameters, and person (legislator) ideal points.

**Usage**

```
id_plot_all_hist(object, params = "person", param_labels = NULL,
  dens_type = "all", return_data = FALSE, func = median, ...)
```

**Arguments**

object	A fitted idealstan object
params	Select the type of parameter from the model to plot. 'person' for person/legislator ideal points, 'miss_diff' and 'miss_discrim' for difficulty and discrimination parameters from the missing/inflated item/bill parameters, and 'regular_diff' and 'regular_discrim' for difficulty and discrimination parameters from the non-missing/non-inflated item/bill parameters.
param_labels	A vector of labels equal to the number of parameters. Primarily useful if return_data is TRUE.
dens_type	Can be 'all' for showing 90 Or to show one of those posterior estimates at a time, use 'high' for 90 'low' for 10 in func (median by default).
return_data	Whether or not to return the plot as a ggplot2 object and the data together in a list instead of plotting.
func	The function to use if 'dens_type' is set to 'function'.
...	Other options passed on to the plotting function, currently ignored.

---

id_plot_compare	<i>Function to compare two fitted idealstan models by plotting ideal points. Assumes that underlying data is the same for both models.</i>
-----------------	--

---

### Description

Function to compare two fitted idealstan models by plotting ideal points. Assumes that underlying data is the same for both models.

### Usage

```
id_plot_compare(model1 = NULL, model2 = NULL, scale_flip = FALSE,
  return_data = FALSE, labels = NULL, hjust = -0.1, palette = "Set1",
  color_direction = 1, text_size_label = 2, rescale = FALSE)
```

### Arguments

model1	The first model to compare
model2	The second model to compare
scale_flip	This parameter is set to true if you have two models that are reflected around the ideal point axis. This can happen as a result of identification and is harmless.
return_data	Whether to return the underlying data
labels	TRUE or FALSE, whether to use labels for points
hjust	The horizontal adjustment of point labels
palette	colorbrewer palette name
color_direction	Whether to reverse the color scale
text_size_label	Size of point labels
rescale	Whether to rescale the estimates from two models so they will match regardless of arbitrary scale shifts in the ideal points

---

id_plot_legis	<i>Plot Legislator/Person and Bill/Item Ideal Points</i>
---------------	--

---

### Description

This function can be used on a fitted idealstan object to plot the relative positions and uncertainties of legislator/persons and bills/items.

**Usage**

```
id_plot_legis(object, return_data = FALSE, item_plot = NULL,
  text_size_label = 2, text_size_group = 2.5, point_size = 1,
  hjust_length = -0.7, person_labels = TRUE, group_labels = F,
  person_ci_alpha = 0.1, show_score = NULL, abs_and_reg = "both",
  show_true = FALSE, group_color = TRUE, hpd_limit = 10,
  group_overlap = FALSE, sample_persons = NULL, ...)
```

**Arguments**

object	A fitted idealstan object
return_data	If true, the calculated legislator/bill data is returned along with the plot in a list
item_plot	The column index of the bill/item midpoint to overlay on the plot
text_size_label	ggplot2 text size for legislator labels
text_size_group	ggplot2 text size for group text used for points
point_size	If person_labels and group_labels are set to FALSE, controls the size of the points plotted.
hjust_length	horizontal adjustment of the legislator labels
person_labels	if TRUE, use the person.names column to plot labels for the person (legislator) ideal points
group_labels	if TRUE, use the group column to plot text markers for the group (parties) from the person/legislator data
person_ci_alpha	The transparency level of the dot plot and confidence bars for the person ideal points
show_score	Show only person/legislator ideal points that have a certain score/vote category from the outcome (character string)
abs_and_reg	Whether to show 'both' absence and regular item/bill midpoints if the model is absence-inflated, the default, or 'Absence Points' for only the absence midpoints or 'Vote Points' for only the non-inflated midpoints
show_true	Whether to show the true values of the legislators (if model has been simulated)
group_color	If TRUE, give each group/bloc a different color
hpd_limit	The greatest absolute difference in high-posterior density interval shown for any point. Useful for excluding imprecisely estimated persons/legislators from the plot. Leave NULL if you don't want to exclude any.
group_overlap	Whether to prevent the text from overlapping itself (ggplot2 option)
sample_persons	If you don't want to use the full number of persons/legislators from the model, enter a proportion (between 0 and 1) to select only a fraction of the persons/legislators.
...	Other options passed on to plotting function, currently ignored



**Details**

This plot shows the distribution of ideal points for the legislators/persons in the model. It will plot them as a vertical dot plot with associated high-density posterior interval (10% to 90%). In addition, if the column index for a bill/item from the response matrix is passed to the `item_plot` option, then an item/bill midpoint will be overlain on the ideal point plot, showing the point at which legislators/persons are indifferent to voting/answering on the bill/item. Note that because this is an ideal point model, it is not possible to tell from the midpoint itself which side will be voting which way. For that reason, the legislators/persons are colored by their votes/scores to make it clear.

**Examples**

```
# To demonstrate, we load a fitted idealstan object based on the 114th Senate
data('senate114_fit')

id_plot_legis(senate114_fit)

# We can overlap the bill/item midpoints to show
# where the persons/legislators are indifferent to responding positively

id_plot_legis(senate114_fit, item_plot=5)
```

---

id\_plot\_ppc

*Generic Method for Plotting Posterior Predictive Distribution*


---

**Description**

This function is a wrapper around [ppc\\_bars](#) that enables the plotting of the posterior predictive distribution from [id\\_post\\_pred](#) against the original data and for the distribution for individual persons/legislators and bills/items.

**Usage**

```
id_plot_ppc(object, ...)
```

**Arguments**

<code>object</code>	A fitted <code>idealstan</code> object
<code>...</code>	Other arguments passed on to underlying functions

---

 id\_plot\_ppc, idealstan-method

*Plot Posterior Predictive Distribution for idealstan Objects*


---

### Description

This function is a wrapper around [ppc\\_bars](#) that plots the posterior predictive distribution derived from [id\\_post\\_pred](#) against the original data. You can also specify a legislator/person or bill/item by specifying the index of each in the original score/vote matrix. Only person or items can be specified, not both.

### Usage

```
## S4 method for signature 'idealstan'
id_plot_ppc(object, ppc_pred = NULL, person = NULL,
            item = NULL, ...)
```

### Arguments

object	A fitted idealstan object
ppc_pred	The output of the <a href="#">id_post_pred</a> function on a fitted idealstan object
person	The indices of the rows (persons/legislators) around which to compare the posterior prediction
item	The indices of the columns (items/bills) around which to compare the posterior prediction
...	Other arguments passed on to <a href="#">ppc_bars</a>

---

 id\_plot\_rhats

*Plotting Function to Display Rhat Distribution*


---

### Description

This plotting function displays a histogram of the Rhat values of all parameters in an idealstan model.

### Usage

```
id_plot_rhats(obj)
```

### Arguments

obj	A fitted idealstan object.
-----	----------------------------

---

id_plot_sims	<i>This function plots the results from a simulation generated by <a href="#">id_sim_gen</a>.</i>
--------------	---

---

**Description**

This function plots the results from a simulation generated by [id\\_sim\\_gen](#).

**Usage**

```
id_plot_sims(sims, type = "RMSE")
```

**Arguments**

sims	A fitted <code>idealstan</code> object that has true data generated by <a href="#">id_sim_gen</a>
type	Type of analysis of true versus fitted values, can be 'RMSE', 'Residuals' or 'Coverage'

---

id_post_pred	<i>Generic Method for Obtaining Posterior Predictive Distribution from Stan Objects</i>
--------------	---

---

**Description**

This function is a generic that is used to match the functions used with [ppc\\_bars](#) to calculate the posterior predictive distribution of the data given the model.

**Usage**

```
id_post_pred(object, ...)
```

**Arguments**

object	A fitted <code>idealstan</code> object
...	All other parameters passed on to the underlying function.

**Value**

`posterior_predict` methods should return a  $D$  by  $N$  matrix, where  $D$  is the number of draws from the posterior predictive distribution and  $N$  is the number of data points being predicted per draw.

---

id\_post\_pred, idealstan-method

*Posterior Prediction for idealstan objects*

---

### Description

This function will draw from the posterior predictive distribution of the outcome, i.e., all the scores or votes that are used to create the idealstan model.

You can then use functions such as [ppc\\_bars](#) to see how well the model does returning the correct number of categories in the score/vote matrix. Also see `help("posterior_predict", package = "rstanarm")`

### Usage

```
## S4 method for signature 'idealstan'
id_post_pred(object, draws = 100,
  sample_scores = NULL, ...)
```

### Arguments

object	A fitted idealstan object
draws	The number of draws to use from the total number of posterior draws (default is 100).
sample_scores	In addition to reducing the number of posterior draws used to calculate the posterior predictive distribution.
...	Any other arguments passed on to posterior_predict (currently none available)

---

id\_sim\_coverage

*Function that computes how often the true value of the parameter is included within the 95/5 high posterior density interval*

---

### Description

Function that computes how often the true value of the parameter is included within the 95/5 high posterior density interval

### Usage

```
id_sim_coverage(obj, rep = 1, quantiles = c(0.95, 0.05))
```

### Arguments

obj	A fitted idealstan object with true data generated by <a href="#">id_sim_gen</a>
rep	How many times the models were fitted on new data, currently can only be 1
quantiles	What the quantile coverage of the high posterior density interval should be

---

id_sim_gen	<i>Simulate IRT ideal point data</i>
------------	--------------------------------------

---

### Description

A function designed to simulate absence-inflated data with either binary or ordinal outcomes.

### Usage

```
id_sim_gen(num_person = 20, num_bills = 50, absence_discrim_sd = 2,
  absence_diff_mean = 0.5, reg_discrim_sd = 2, diff_sd = 0.25,
  ideal_pts_sd = 1, prior_type = "gaussian", ordinal = TRUE,
  ordinal_outcomes = 3, graded_response = FALSE, absence = TRUE)
```

### Arguments

num_person	The number of persons/personlators
num_bills	The number of items/bills
absence_discrim_sd	The SD of the discrimination parameters for the absence model
absence_diff_mean	The mean intercept for the absence model; increasing it will lower the total number of absences
reg_discrim_sd	The SD of the discrimination parameters for the non-inflated model
diff_sd	The SD of the difficulty parameters (bill/item intercepts)
ideal_pts_sd	The SD for the person/personlator ideal points
prior_type	The statistical distribution that generates the data. Currently only 'gaussian' is supported.
ordinal	Whether the data should have binary (FALSE) or ordinal (TRUE) responses
ordinal_outcomes	If ordinal is TRUE, an integer giving the total number of categories
graded_response	Not currently implemented
absence	If TRUE, an absence-inflated dataset is produced.

### Details

This function produces simulated data that matches (as closely as possible) the models used in the underlying Stan code. Currently the simulation can produce absence-inflated and non-absence-inflated binary and ordinal ideal point models.

### Value

The results is a `idealdata` object that can be used in the `id_estimate` function to run a model. It can also be used in the simulation plotting functions.

**See Also**

[id\\_plot\\_sims](#) for plotting fitted models versus true values.

---

id_sim_resid	<i>Residual function for checking estimated samples compared to true simulation scores Returns a data frame with residuals plus quantiles.</i>
--------------	--

---

**Description**

Residual function for checking estimated samples compared to true simulation scores Returns a data frame with residuals plus quantiles.

**Usage**

```
id_sim_resid(obj, rep = 1)
```

**Arguments**

obj	A fitted <code>idealstan</code> object with true data from <a href="#">id_sim_gen</a>
rep	Over how many replicates to calculate residuals? Currently can only be 1

---

id_sim_rmse	<i>RMSE function for calculating individual RMSE values compared to true simulation scores Returns a data frame with RMSE plus quantiles.</i>
-------------	---

---

**Description**

RMSE function for calculating individual RMSE values compared to true simulation scores Returns a data frame with RMSE plus quantiles.

**Usage**

```
id_sim_rmse(obj, rep = 1)
```

**Arguments**

obj	A fitted <code>idealstan</code> object with true data from <a href="#">id_sim_gen</a>
rep	Over how many replicates to calculate RMSE? Currently can only be 1

---

launch_shinystan	<i>Generic Method to Use shinystan with idealstan</i>
------------------	---

---

**Description**

A generic function for launching [launch\\_shinystan](#).

**Usage**

```
launch_shinystan(object, ...)
```

**Arguments**

object	A fitted idealstan object.
...	Other arguments passed on to underlying function

---

launch_shinystan, idealstan-method	<i>Function to Launch Shinystan with an idealstan Object</i>
------------------------------------	--

---

**Description**

This wrapper will pull the rstan samples out of a fitted idealstan model and then launch [launch\\_shinystan](#). This function is useful for examining convergence statistics of the underlying MCMC sampling.

**Usage**

```
## S4 method for signature 'idealstan'
launch_shinystan(object, pars = c("L_free",
  "sigma_reg_free", "sigma_abs_free", "restrict_high", "restrict_low",
  "restrict_ord", "steps_votes", "steps_votes_grm"), ...)
```

**Arguments**

object	A fitted idealstan object
pars	A character vector of parameters to select from the underlying rstan model object
...	Other parameters passed on to <a href="#">shinystan</a>

**See Also**

[shinystan](#)

---

release_questions	<i>Function that provides additional check questions for package release</i>
-------------------	--

---

**Description**

Function that provides additional check questions for package release

**Usage**

```
release_questions()
```

---

senate114	<i>Rollcall vote data for 114th Senate</i>
-----------	--

---

**Description**

This rollcall vote object (see [rollcall](#)) contains the full voting records for the 114th Senate in the US Congress. The data can be pre-processed via the [id\\_make](#) function for estimation. See package vignette for details.

**Usage**

```
senate114
```

**Format**

A roll call vote object that is a list with votes as the legislator-bill roll call vote matrix.

---

senate114_fit	<i>Fitted idealstan model for 114th Senate Data</i>
---------------	---

---

**Description**

This R object is a fitted `idealstan` object that can be used for testing and illustratory purposes.

**Usage**

```
senate114_fit
```

**Format**

A fitted `idealstan` object



---

`summary,idealstan-method`*Posterior Summaries for fitted idealstan object*

---

**Description**

This function produces quantiles and standard deviations for the posterior samples of `idealstan` objects.

**Usage**

```
## S4 method for signature 'idealstan'  
summary(object)
```

**Arguments**

`object` An `idealstan` object fitted by `id_estimate`

**Value**

A `tibble` data frame with parameters as rows and descriptive statistics as columns

# Index

## \*Topic **datasets**

senate114, [24](#)  
senate114\_fit, [24](#)

extract, [9](#)

ggplot, [14](#)

id\_estimate, [3](#), [3](#), [9](#), [12–14](#), [21](#), [25](#)

id\_extract, [8](#)

id\_extract, idealstan-method, [9](#)

id\_log\_lik, [10](#)

id\_log\_lik, idealstan-method, [10](#)

id\_make, [2](#), [4](#), [6](#), [7](#), [9](#), [11](#), [24](#)

id\_plot, [13](#)

id\_plot, idealstan-method, [13](#)

id\_plot\_all\_hist, [14](#), [14](#)

id\_plot\_compare, [14](#), [15](#)

id\_plot\_legis, [7](#), [14](#), [15](#)

id\_plot\_ppc, [17](#)

id\_plot\_ppc, idealstan-method, [18](#)

id\_plot\_rhats, [14](#), [18](#)

id\_plot\_sims, [12](#), [14](#), [19](#), [22](#)

id\_post\_pred, [17](#), [18](#), [19](#)

id\_post\_pred, idealstan-method, [20](#)

id\_sim\_coverage, [20](#)

id\_sim\_gen, [11](#), [19](#), [20](#), [21](#), [22](#)

id\_sim\_resid, [22](#)

id\_sim\_rmse, [22](#)

idealdata-class, [2](#)

idealstan, [3](#), [6](#)

idealstan-class, [3](#)

idealstan-package (idealstan), [3](#)

launch\_shinystan, [23](#), [23](#)

launch\_shinystan, idealstan-method, [23](#)

loo, [10](#)

posterior\_predict, [7](#)

ppc\_bars, [17–20](#)

release\_questions, [24](#)

rollcall, [24](#)

rstan, [9](#)

senate114, [24](#)

senate114\_fit, [24](#)

shinystan, [23](#)

stan, [3–6](#), [9](#)

summary, [7](#)

summary, idealstan-method, [25](#)

tibble, [25](#)

vb, [4–6](#)