

Package ‘hbamr’

September 25, 2023

Title Hierarchical Bayesian Aldrich-McKelvey Scaling via 'Stan'

Version 1.2.0

Description Perform hierarchical Bayesian Aldrich-McKelvey scaling using Hamiltonian Monte Carlo via 'Stan'. Aldrich-McKelvey ('AM') scaling is a method for estimating the ideological positions of survey respondents and political actors on a common scale using positional survey data. The hierarchical versions of the Bayesian 'AM' model included in this package outperform other versions both in terms of yielding meaningful posterior distributions for respondent positions and in terms of recovering true respondent positions in simulations. The package contains functions for preparing data, fitting models, extracting estimates, plotting key results, and comparing models using cross-validation. The default model is described in Bølstad (2023) <[doi:10.1017/pan.2023.18](https://doi.org/10.1017/pan.2023.18)>.

License GPL (>= 3)

URL <https://github.com/jbolstad/hbamr/>

Depends R (>= 3.4.0)

Imports dplyr, ggplot2, loo, matrixStats, methods, parallel,
pbmcapply, plyr, RColorBrewer, Rcpp (>= 1.0.7), RcppParallel
(>= 5.1.4), rlang, rstan (>= 2.26.1), rstantools (>= 2.2.0),
stats, tidy

Suggests data.table, knitr, rmarkdown

LinkingTo BH (>= 1.66.0), Rcpp (>= 1.0.7), RcppEigen (>= 0.3.3.9.1),
RcppParallel (>= 5.1.4), rstan (>= 2.26.1), StanHeaders (>= 2.26.22)

Biarch true

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

SystemRequirements GNU make

VignetteBuilder knitr

NeedsCompilation yes

Author Jørgen Bølstad [aut, cre] (<<https://orcid.org/0000-0002-7623-5741>>)

Maintainer Jørgen Bølstad <jorgen.bolstad@stv.uio.no>

Repository CRAN

Date/Publication 2023-09-25 15:00:02 UTC

R topics documented:

hbamr-package	2
fbam	3
get_est	4
get_plot_data	5
hbam	6
hbam_cv	9
LC1980	12
LC2012	12
plot_over_self	13
plot_respondents	14
plot_stimuli	15
prep_data	15
prep_data_cv	17
Index	18

hbamr-package	<i>Hierarchical Bayesian Aldrich-McKelvey Scaling via Stan</i>
---------------	--

Description

Fit hierarchical Bayesian Aldrich-McKelvey (HBAM) models using a form of Hamiltonian Monte Carlo via Stan. Aldrich-McKelvey (AM) scaling is a method for estimating the ideological positions of survey respondents and political actors on a common scale using positional survey data. The hierarchical versions of the Bayesian AM model included in this package outperform other versions by a considerable margin both in terms of yielding meaningful posterior distributions for respondent positions and in terms of recovering true respondent positions in simulations. The package contains functions for preparing data, fitting models, extracting estimates, plotting key results, and comparing models using cross-validation.

Author(s)

Jørgen Bølstad

References

- Bølstad, Jørgen (2023). Hierarchical Bayesian Aldrich-McKelvey Scaling. *Political Analysis*. doi:10.1017/pan.2023.18.
- Stan Development Team (2023). RStan: the R interface to Stan. R package version 2.21.8. <https://mc-stan.org>.

See Also

- <https://github.com/jbolstad/hbamr/>

fbam

*Fit the FBAM_MINI model using optimization***Description**

Fit a simplified Bayesian Aldrich-McKelvey model with fixed hyperparameters using optimization via rstan.

Usage

```
fbam(
  self = NULL,
  stimuli = NULL,
  allow_miss = 2,
  req_valid = NA,
  req_unique = 2,
  prep_data = TRUE,
  data = NULL,
  seed = sample.int(.Machine$integer.max, 1),
  ...
)
```

Arguments

self	A numerical vector of N ideological self-placements. Any missing data must be coded as NA. This argument will not be used if the data have been prepared in advance via the prep_data function.
stimuli	An $N \times J$ matrix of numerical stimulus placements, where J is the number of stimuli. Any missing data must be coded as NA. This argument will not be used if the data have been prepared in advance via the prep_data function.
allow_miss	Integer specifying how many missing stimulus positions to be accepted for an individual still to be included in the analysis. This argument will not be used if the data have been prepared in advance via the prep_data function. Defaults to 2.
req_valid	Integer specifying how many valid observations to require for a respondent to be included in the analysis. The default is req_valid = J - allow_miss, but if specified, req_valid takes precedence. This argument will not be used if the data have been prepared in advance via the prep_data function.
req_unique	Integer specifying how many unique positions on the ideological scale each respondent is required to have used when placing the stimuli in order to be included in the analysis. The default is req_unique = 2. This argument will not be used if the data have been prepared in advance via the prep_data function.

prep_data	Logical: Should the data be prepared before fitting the model? (Or have the data been prepared in advance via the prep_data function? If so, set prep_data = FALSE.)
data	List of data that have been prepared in advance via the prep_data function. Only applicable when prep_data = TRUE.
seed	A positive integer specifying an optional seed for reproducibility. If this argument is not supplied, a random seed will be generated and the function will produce slightly different results on each run.
...	Arguments passed to rstan::optimizing.

Value

A list produced by rstan::optimizing.

Examples

```
# Loading ANES 2012 data:
data(LC2012)

self <- LC2012[, 2]
stimuli <- LC2012[, -c(1:2)]

# Fitting the FBAM_MINI model:
fit_fbam_mini <- fbam(self, stimuli)

# Obtaining point estimates for the latent stimulus positions:
theta_est <- get_est(fit_fbam_mini, par = "theta")
```

get_est	<i>Extract summaries of marginal posterior distributions from an HBAM model or point estimates from an FBAM model</i>
---------	---

Description

For objects produced by hbam, this function is a wrapper for rstan::summary. For objects produced by fbam it offers a way to extract point estimates.

Usage

```
get_est(
  object,
  par = "theta",
  probs = c(0.025, 0.5, 0.975),
  simplify = TRUE,
  ...
)
```

Arguments

object	An instance of class <code>stanfit</code> produced by <code>hbam</code> , or a list produced by <code>fbam</code> .
par	Character: Name of the parameter type to be extracted. Typically "theta" (stimuli positions) or "chi" (respondent positions).
probs	A numeric vector of quantiles of interest for summarizing <code>stanfit</code> objects. The default is <code>c(0.025, 0.50, 0.975)</code> .
simplify	Logical: Should the returned object be simplified by dropping the Monte Carlo standard error and the posterior standard deviation? Defaults to <code>TRUE</code> .
...	Other arguments are passed on to <code>rstan::summary</code> when summarizing <code>stanfit</code> objects.

Value

A tibble containing summaries of marginal posterior distributions. For objects produced by `fbam`, only maximum a posteriori estimates are returned.

get_plot_data	<i>Extract data for plotting results from an HBAM model.</i>
---------------	--

Description

Extract data for plotting results from an HBAM model.

Usage

```
get_plot_data(object, n_draws = 15, seed = 1)
```

Arguments

object	An instance of class <code>stanfit</code> produced by <code>hbam</code> or a list produced by <code>fbam</code> .
n_draws	Integer specifying the number of posterior draws to use when illustrating the uncertainty of the population distribution. This only applies for <code>stanfit</code> objects.
seed	A positive integer specifying an optional seed for reproducibility. The seed is used to select respondent position draws for illustrating uncertainty. This only applies for <code>stanfit</code> objects.

Value

A list of three tibbles: The first element contains the posterior mean stimulus positions, as well as the x- and y-values of the posterior modes (which can be useful for labeling the distributions). The second element contains the posterior draws for the stimulus positions (which can be used to calculate marginal posterior densities). The third element contains the selected number of posterior draws for each respondent (which form the key ingredient for `plot_respondents`).

 hbam

Fit an HBAM model

Description

Fit a Hierarchical Bayesian Aldrich-McKelvey model using automatically tuned Hamiltonian Monte Carlo (NUTS) sampling via `rstan`.

Usage

```

hbam(
  self = NULL,
  stimuli = NULL,
  model = "HBAM",
  allow_miss = 2,
  req_valid = NA,
  req_unique = 2,
  prefs = NULL,
  prep_data = TRUE,
  data = NULL,
  chains = 4,
  cores = parallel::detectCores(logical = FALSE),
  warmup = 1000,
  iter = 4000,
  thin = 3,
  control = list(adapt_delta = 0.6),
  seed = sample.int(.Machine$integer.max, 1),
  ...
)

```

Arguments

<code>self</code>	A numerical vector of N ideological self-placements. Any missing data must be coded as <code>NA</code> . This argument will not be used if the data have been prepared in advance via the <code>prep_data</code> function.
<code>stimuli</code>	An $N \times J$ matrix of numerical stimulus placements, where J is the number of stimuli. Any missing data must be coded as <code>NA</code> . This argument will not be used if the data have been prepared in advance via the <code>prep_data</code> function.
<code>model</code>	Character: Name of the model to be used. One of: "HBAM", "HBAM_2", "HBAM_NE", "HBAM_HM", "HBAM_MINI", "HBAM_0", "HBAM_R", "HBAM_R_MINI", "FBAM_MINI", or "BAM". Defaults to "HBAM".
<code>allow_miss</code>	Integer specifying how many missing stimulus positions to be accepted for an individual still to be included in the analysis. This argument will not be used if the data have been prepared in advance via the <code>prep_data</code> function. Defaults to 2.

req_valid	Integer specifying how many valid observations to require for a respondent to be included in the analysis. The default is req_valid = J - allow_miss, but if specified, req_valid takes precedence. This argument will not be used if the data have been prepared in advance via the prep_data function.
req_unique	Integer specifying how many unique positions on the ideological scale each respondent is required to have used when placing the stimuli in order to be included in the analysis. The default is req_unique = 2. This argument will not be used if the data have been prepared in advance via the prep_data function.
prefs	An $N \times J$ matrix of numerical stimulus ratings or preference scores. These data are only required by the "HBAM_R" and "HBAM_R_MINI" models and will be ignored when fitting other models.
prep_data	Logical: Should the data be prepared before fitting the model? (Or have the data been prepared in advance via the prep_data function? If so, set prep_data = FALSE.)
data	List of data that have been prepared in advance via the prep_data function. Only applicable when prep_data = TRUE.
chains	A positive integer specifying the number of Markov chains. Defaults to 4.
cores	The number of cores to use when executing the Markov chains in parallel. By default, all detected physical cores will be used if chains is equal to or higher than the number of cores.
warmup	A positive integer specifying the number of warmup (aka burn-in) iterations per chain. If step-size adaptation is on (which it is by default), this also controls the number of iterations for which adaptation is run (and hence these warmup samples should not be used for inference). The number of warmup iterations should be smaller than iter.
iter	A positive integer specifying the number of iterations for each chain (including warmup).
thin	A positive integer specifying the period for saving samples.
control	A named list of parameters to control the sampler's behavior. See the details in the documentation for the control argument in the stan function in the rstan package.
seed	A positive integer specifying an optional seed for reproducibility. If this argument is not supplied, a random seed will be generated and the function will produce slightly different results on each run.
...	Arguments passed to rstan::sampling.

Details

This package provides several alternative models, which can be specified using the names below. Users who are unsure which model to use are advised to use the default HBAM model. If speed or sampling diagnostics are an issue, HBAM_MINI may provide a useful alternative.

HBAM is the default model, which allows for scale flipping and employs hierarchical priors on all individual level parameters. It also models heteroskedastic errors that vary by both individual and stimuli. This model was introduced by Bølstad (2023).

HBAM_2 uses different hyperpriors for the shifting parameters of respondents with different self-placements. In particular, the model estimates a separate mean hyperparameter for each self-placement. This model avoids shrinking the shifting parameters toward a common population mean, and may therefore fit better than HBAM if there are clear differences in average shifting across the scale of self-placements. However, this model also tends to run slower than the standard model.

HBAM_NE models the self-placements as if they contain no error. The latent respondent positions are not treated as parameters, but rather calculated as function of the self-placements and other individual level parameters. The respondents positions are not given a prior, which means the model relies on the likelihood function and the prior on beta to yield meaningful results for these positions. By assuming no error in the self-placements, the model may underestimate the uncertainty in estimated respondents positions, while otherwise yielding very similar results to the standard HBAM model. In contrast to the standard model, the estimated respondent positions from this model will not exhibit any shrinkage, which for some purposes may be desirable, as the results may better represent the true distances between respondents and stimuli. This model also runs somewhat faster than the standard HBAM model.

HBAM_HM assumes the prediction errors in the stimuli placements to be homoskedastic. This simplified model should normally produce very similar results to the HBAM model, and it runs somewhat faster.

HBAM_MINI combines the characteristics of HBAM_NE and HBAM_HM: It models the self-placements as if they contain no error and assumes the prediction errors in the stimuli placements to be homoskedastic. This is the simplest model provided in this package that still retains all key features of the HBAM model. It is also the fastest HBAM variant in this package – sampling about twice as fast as the standard HBAM model for the dataset analyzed here (while yielding very similar point estimates). For large datasets, this model may provide a reasonable compromise between model complexity and estimation speed.

HBAM_0 does not allow for scale flipping. This may be useful if there are truly zero cases of scale flipping in the data. Such scenarios can be created artificially, but may also arise in real data. For example, expert surveys appear unlikely to contain many instances of scale flipping.

HBAM_R incorporates the rationalization component of the ISR model by Bølstad (2020). This model requires additional data to be supplied to the `prep_data()` function: An $N \times J$ matrix of stimuli ratings from the respondents, supplied as the argument `pref`. The rationalization part of the model is simplified relative to the original ISR model: The direction in which respondents move disfavored stimuli is estimated as a common expectation for each possible self-placement on the scale.

HBAM_R_MINI combines the features of the HBAM_R model with the light-weight features of the HBAM_MINI model to achieve faster sampling compared to HBAM_R.

FBAM_MINI is a version of the HBAM_MINI model with fixed hyperparameters to allow fitting via optimization rather than MCMC – which can be useful for large data sets. The hyperparameters have been set to realistic values based on analyses of ANES data. As with the other models, scale-dependent priors are automatically adjusted to the length of the survey scale.

BAM is an unpooled model, similar to the JAGS version introduced by Hare et al. (2015). This model is mainly provided to offer a baseline for model comparisons. While it is simple and fast, this model tends to overfit the data and produce invalid posterior distributions for some respondent positions (Bølstad 2023).

Some of these models can also be used in situations where self-placements are not available and the only goal is to estimate stimulus positions. This can be achieved by supplying a vector of zeros (or

random data) instead of real self-placements: `self = rep(0, nrow(stimuli))`). The HBAM_NE and HBAM_MINI models are then the relevant alternatives, as the other HBAM variants will include superfluous parameters (and will not sample properly with zero variance in the supplied self-placement data).

Value

An object of S4 class `stanfit`.

References

- Bølstad, Jørgen (2023). Hierarchical Bayesian Aldrich-McKelvey Scaling. *Political Analysis*. doi:10.1017/pan.2023.18.
- Bølstad, Jørgen (2020). Capturing Rationalization Bias and Differential Item Functioning: A Unified Bayesian Scaling Approach. *Political Analysis* 28(3): 340–355.
- Hare, Christopher et al. (2015). Using Bayesian Aldrich-McKelvey Scaling to Study Citizens' Ideological Preferences and Perceptions. *American Journal of Political Science* 59(3): 759–774.

Examples

```
# Loading and re-coding ANES 1980 data:
data(LC1980)
LC1980[LC1980 == 0 | LC1980 == 8 | LC1980 == 9] <- NA

# Making a small subset of the data for illustration:
self <- LC1980[1:100, 1]
stimuli <- LC1980[1:100, -1]

# Fitting the HBAM_MINI model, obtaining 1000 draws:
fit_hbam_mini <- hbam(self, stimuli, model = "HBAM_MINI",
  warmup = 500, iter = 1000, chains = 2, thin = 1)

# Preparing the data before fitting, requiring complete responses:
dat <- prep_data(self, stimuli, allow_miss = 0)
fit_hbam_mini <- hbam(data = dat, prep_data = FALSE, model = "HBAM_MINI",
  warmup = 500, iter = 1000, chains = 2, thin = 1)

# Obtaining posterior summaries for the latent stimulus positions:
theta_est <- get_est(fit_hbam_mini, par = "theta")
```

hbam_cv

Perform K-fold cross-validation for an HBAM-type model

Description

This function performs a K-fold cross-validation for an HBAM-type model in order to estimate the expected log pointwise predictive density for a new dataset (ELPD).

Usage

```

hbam_cv(
  self = NULL,
  stimuli = NULL,
  model = "HBAM",
  allow_miss = 0,
  req_valid = NA,
  req_unique = 2,
  prefs = NULL,
  prep_data = TRUE,
  data = NULL,
  K = 10,
  chains = 2,
  cores = parallel::detectCores(logical = FALSE),
  warmup = 1000,
  iter = 4000,
  thin = 1,
  control = list(adapt_delta = 0.6),
  seed = 1
)

```

Arguments

<code>self</code>	A numerical vector of N ideological self-placements. Any missing data must be coded as NA. This argument will not be used if the data have been prepared in advance via the <code>prep_data</code> function.
<code>stimuli</code>	An $N \times J$ matrix of numerical stimulus placements, where J is the number of stimuli. Any missing data must be coded as NA. This argument will not be used if the data have been prepared in advance via the <code>prep_data</code> function.
<code>model</code>	Character: Name of the model to be used. Defaults to "HBAM".
<code>allow_miss</code>	Integer specifying how many missing stimulus positions to be accepted for an individual still to be included in the analysis. This argument will not be used if the data have been prepared in advance via the <code>prep_data</code> function. Defaults to 0.
<code>req_valid</code>	Integer specifying how many valid observations to require for a respondent to be included in the analysis. The default is <code>req_valid = J - allow_miss</code> , but if specified, <code>req_valid</code> takes precedence. This argument will not be used if the data have been prepared in advance via the <code>prep_data</code> function.
<code>req_unique</code>	Integer specifying how many unique positions on the ideological scale each respondent is required to have used when placing the stimuli in order to be included in the analysis. The default is <code>req_unique = 2</code> . This argument will not be used if the data have been prepared in advance via the <code>prep_data</code> function.
<code>prefs</code>	An $N \times J$ matrix of numerical stimulus ratings or preference scores. These data are only required by the "HBAM_R" and "HBAM_R_MINI" models and will be ignored when fitting other models.

LC1980

1980 Liberal-Conservative Scales

Description

Liberal-Conservative 7-point scales from the 1980 National Election Study. Includes (in order) self-placement, and rankings of Carter, Reagan, Kennedy, Anderson, Republican party, Democratic Party. Stored as a matrix of integers. The numbers 0, 8, and 9 are considered to be missing values.

Usage

```
data(LC1980)
```

Format

An object of class `matrix` (inherits from `array`) with 888 rows and 7 columns.

Source

American National Election Studies: www.electionstudies.org/.

This dataset was originally part of the `basicspace` package under the same name ("LC1980").

Examples

```
data(LC1980)
LC1980[LC1980 == 0 | LC1980 == 8 | LC1980 == 9] <- NA
head(LC1980)
```

LC2012

2012 Liberal-Conservative Scales

Description

Liberal-Conservative 7-point scales from the 2012 National Election Study. Includes (in order) original case id, self-placement, and rankings of Obama, Romney, Democratic Party, Republican party. Missing values are coded as NA.

Usage

```
data(LC2012)
```

Format

An object of class `data.frame` with 5914 rows and 6 columns.

Source

American National Election Studies: www.electionstudies.org/.

Examples

```
data(LC2012)
head(LC2012)
```

plot_over_self	<i>Plot individual parameter estimates over self-placements</i>
----------------	---

Description

Create a boxplot of individual parameter estimates from an HBAM model over self-placements

Usage

```
plot_over_self(
  object,
  data,
  par = "chi",
  estimate = "median",
  names = NULL,
  parlabel = NULL,
  fill = "#2166AC",
  color = "#053061",
  width = 0.7,
  alpha = 0.5,
  outlier.size = 0.3,
  median_color = "black",
  median_lwd = 0.7
)
```

Arguments

object	An object of class stanfit produced by hbam, a list produced by fbam, or a list of such objects, which will produce a faceted plot.
data	The list of data that was used to produce the object(s).
par	Character: Name of the parameter to be plotted. One of the following: "alpha", "beta", "abs_beta", "lambda", "chi", and "eta". Defaults to "chi".
estimate	Character: Specifying which type of posterior point estimate to use. One of "median" and "mean". Defaults to "median". This only applies for stanfit objects.
names	An optional character vector of model names of same length as the supplied list of models.

parlabel	An optional character containing an alternative label for the parameter (will be parsed if passed as an expression).
fill	Fill color of boxes. Passed on to geom_boxplot.
color	Color of outer lines. Passed on to geom_boxplot.
width	Width of boxes. Passed on to geom_boxplot.
alpha	Number in [0,1]: Inverse level of transparency for fill color. Passed on to geom_boxplot.
outlier.size	Size of dots representing outliers. Passed on to geom_boxplot.
median_color	Color of solid line representing the median.
median_lwd	Thickness of solid line representing the median.

Value

A ggplot object.

plot_respondents	<i>Plot estimated respondent positions</i>
------------------	--

Description

Plot the distribution of estimated respondent positions from an HBAM model.

Usage

```
plot_respondents(
  object,
  inc_stimuli = TRUE,
  n_draws = 15,
  color = "#053061",
  fill = "#2166AC",
  alpha_color = 0.6,
  alpha_fill = 0.7/n_draws,
  seed = 1
)
```

Arguments

object	An instance of class stanfit produced by hbam.
inc_stimuli	Logical: Should estimated stimulus positions also be shown?
n_draws	Integer specifying the number of posterior draws to use when illustrating the uncertainty of the population distribution. Defaults to 15.
color	Color of lines illustrating uncertainty.
fill	Fill color for density plots.
alpha_color	Number in [0,1]: Inverse level of transparency for line color.

alpha_fill	Number in [0,1]: Inverse level of transparency for fill color.
seed	A positive integer specifying an optional seed for reproducibility. The seed is used to select respondent position draws for illustrating uncertainty.

Value

A ggplot object.

plot_stimuli	<i>Plot estimated stimulus positions</i>
--------------	--

Description

Plot marginal posterior distributions of stimulus positions from an HBAM model

Usage

```
plot_stimuli(object, rev_color = FALSE, alpha = 0.55)
```

Arguments

object	An instance of class stanfit produced by hbam.
rev_color	Logical: Display low positions as red and high positions as blue.
alpha	Number in [0,1]: Inverse level of transparency for fill color.

Value

A ggplot object.

prep_data	<i>Prepare data to fit an HBAM model</i>
-----------	--

Description

This function prepares data to fit a hierarchical Bayesian Aldrich-McKelvey (HBAM) model. It can be run ahead of fitting the models, or it can be run implicitly as part of a single function call to fit the models using hbam. It applies a set of inclusion criteria, performs any necessary data transformation, and returns a list of data suited for sampling in rstan. The data provided to prep_data can be centered, but they do not have to be: The function will detect un-centered data and attempt to center these automatically, assuming that the highest and lowest observed values in the data mark the extremes of the scale.

Usage

```
prep_data(
  self,
  stimuli,
  prefs = NULL,
  allow_miss = 2,
  req_valid = NA,
  req_unique = 2,
  B = NULL
)
```

Arguments

<code>self</code>	A numerical vector of N ideological self-placements. Any missing data must be coded as NA.
<code>stimuli</code>	An $N \times J$ matrix of numerical stimulus placements, where J is the number of stimuli. Any missing data must be coded as NA.
<code>prefs</code>	An $N \times J$ matrix of numerical stimulus ratings or preference scores. These data are only required by the "HBAM_R" and "HBAM_R_MINI" models and will be ignored when fitting other models.
<code>allow_miss</code>	Integer specifying how many missing stimulus positions to be accepted for an individual still to be included in the analysis. This argument will not be used if the data have been prepared in advance via the <code>prep_data</code> function. Defaults to 2.
<code>req_valid</code>	Integer specifying how many valid observations to require for a respondent to be included in the analysis. The default is <code>req_valid = J - allow_miss</code> , but if specified, <code>req_valid</code> takes precedence.
<code>req_unique</code>	Integer specifying how many unique positions on the ideological scale each respondent is required to have used when placing the stimuli in order to be included in the analysis. The default is <code>req_unique = 2</code> .
<code>B</code>	Integer specifying the upper bound of the survey scale after centering. If not supplied, this information will be inferred from the data.

Value

A list of data to be used by `hbam`. The returned list includes the logical vector `keep`, which identifies the rows in the original data that have been kept for further analysis. The stimuli data are stored in a vector as a long-form sparse matrix. If the stimuli data include column-names, these will be preserved for later use.

Examples

```
# Loading and re-coding ANES 1980 data:
data(LC1980)
LC1980[LC1980 == 0 | LC1980 == 8 | LC1980 == 9] <- NA
self <- LC1980[, 1]
stimuli <- LC1980[, -1]
```



```

# Prepare data for model fitting, using defaults:
dat <- prep_data(self, stimuli)

# Prepare data for model fitting, using using alternative settings:
dat2 <- prep_data(self, stimuli, allow_miss = 0, req_unique = 3)

# Obtain the data that are included in the analysis:
self2 <- self[dat2$keep]
stimuli2 <- stimuli[dat2$keep, ]

```

```
prep_data_cv
```

Prepare data for a K-fold cross-validation of an HBAM model

Description

This function turns data prepared for hbam into a list of K versions, where each version includes a different vector identifying holdout-data.

Usage

```
prep_data_cv(data, K = 10, seed = 1)
```

Arguments

data	A list of data produced by prep_data.
K	An integer above 2, specifying the number of folds to use in the analysis. Defaults to 10.
seed	An integer passed on to set.seed before creating the folds to increase reproducibility. Defaults to 1.

Value

A list of K data objects where each version includes a different vector identifying holdout-data.

Examples

```

# Loading and re-coding ANES 1980 data:
data(LC1980)
LC1980[LC1980 == 0 | LC1980 == 8 | LC1980 == 9] <- NA
self <- LC1980[, 1]
stimuli <- LC1980[, -1]
dat <- prep_data(self, stimuli)

# Prepare data for cross-validation:
dat_cv <- prep_data_cv(dat, K = 10)

```

Index

* datasets

LC1980, [12](#)

LC2012, [12](#)

fbam, [3](#)

get_est, [4](#)

get_plot_data, [5](#)

hbam, [6](#)

hbam_cv, [9](#)

hbamr (hbamr-package), [2](#)

hbamr-package, [2](#)

LC1980, [12](#)

LC2012, [12](#)

plot_over_self, [13](#)

plot_respondents, [14](#)

plot_stimuli, [15](#)

prep_data, [15](#)

prep_data_cv, [17](#)