Package: intervalpsych (via r-universe)

May 12, 2026

Title Analyzing Interval Data in Psychometrics

Version 0.1.0.9000

Maintainer Matthias Kloft <kloft.dev+intervalpsych@gmail.com>

Description Implements the Interval Consensus Model (ICM) for analyzing continuous bounded interval-valued responses in psychometrics using 'Stan' for Bayesian estimation. Provides functions for transforming interval data to simplex representations, fitting item response theory (IRT) models with isometric log-ratio (ILR) and sum log-ratio (SLR) link functions, and visualizing results. The package enables aggregation and analysis of interval-valued response data commonly found in psychological measurement and related disciplines. Based on Kloft et al. (2024) <doi:10.31234/osf.io/dzvw2>.

License GPL (>= 3)

Encoding UTF-8

URL https://matthiaskloft.github.io/intervalpsych/, https://github.com/matthiaskloft/intervalpsych

BugReports https://github.com/matthiaskloft/intervalpsych/issues

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Biarch true

Depends R (>= 4.1)

Imports dplyr, ggdist, ggokabeito, ggplot2, methods, posterior, purrr, Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), rstantools (>= 2.4)

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

SystemRequirements GNU make

LazyData true

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Config/Needs/website rmarkdown

Config/pak/sysreqs make

Repository https://matthiaskloft.r-universe.dev

Date/Publication 2025-12-13 20:35:37 UTC

RemoteUrl https://github.com/matthiaskloft/intervalpsych

RemoteRef HEAD

RemoteSha d59b08bad6b532799551abdd53fba7c46d130edc

Contents

`extract_consensus` *Extract consensus intervals from ICM Stan Fit Object*

Description

This function extracts parameter estimates for the consensus intervals from a fitted Interval Consensus Model Stan fit object of class `icm_stanfit`.

Usage

```
extract_consensus(icm_stanfit, print_summary = TRUE)
```

Arguments

`icm_stanfit`	An object of class `icm_stanfit` containing the fitted Stan model.
`print_summary`	A logical value indicating whether to print a summary of the extracted parameters. Default is `TRUE`.

Details

This function extracts parameter estimates for the consensus intervals from a fitted Interval Consensus Model Stan fit object of class `icm_stanfit`.

Value

A list containing:

`df_rvar`	A data frame with extracted posterior samples in the random variable datatype (see `posterior::rvar()`).
`summary`	A table with posterior medians and credible intervals for the consensus intervals.

Examples

```
# Create minimal example data
df_simplex <- data.frame(
  x1 = c(0.3, 0.4, 0.2, 0.5),
  x2 = c(0.3, 0.2, 0.4, 0.2),
  x3 = c(0.4, 0.4, 0.4, 0.3)
)
id_person <- c(1, 1, 2, 2)
id_item <- c(1, 2, 1, 2)

# Fit ICM model
fit <- fit_icm(df_simplex, id_person, id_item, n_chains = 1,
               iter_sampling = 100, iter_warmup = 100,
               refresh = 0)

# Extract consensus intervals
consensus <- extract_consensus(fit)
```

`fit_icm` *Fit the Interval Consensus Model*

Description

This function fits the Interval Consensus Model (ICM, Kloft et al., 2024) using Stan.

Usage

```
fit_icm(
  df_simplex,
  id_person,
  id_item,
  item_labels = NULL,
  link = "ilr",
  padding = 0,
  iter_sampling = 500,
  iter_warmup = 500,
  n_chains = 4,
  n_cores = 1,
  adapt_delta = 0.9,
  ...
)
```

Arguments

df_simplex	A dataframe containing the simplex data.
id_person	A vector of person indices.
id_item	A vector of item indices.
item_labels	A vector of item labels. Can be long format matching id_item or a vector of unique labels in ascending order. Default is NULL.
link	A character string specifying the link function. Options are "ilr" (Isometric Log-Ratio) or "slr" (Sum Log-Ratio). See also ilr() and slr() for details. Default is "ilr".
padding	Padding constant that was used to remove zero-components from the simplex. Default is 0. The model will reverse the padding when transforming results back to the original interval response scale. See also remove_zeros() for details.
iter_sampling	An integer specifying the number of sampling iterations used by rstan::sampling() Default is 500.
iter_warmup	An integer specifying the number of warmup iterations used by rstan::sampling() Default is 500.
n_chains	An integer specifying the number of Markov chains used by rstan::sampling() Default is 4.
n_cores	An integer specifying the number of cores to use used by rstan::sampling() Default is 1.
adapt_delta	A numeric value specifying the target acceptance rate used by rstan::sampling() Default is 0.9.
...	Additional arguments passed to the rstan::sampling() function.

Value

A fitted Stan model object of class `icm_stanfit` containing the following components:

stan_model The compiled Stan model object.

stan_fit The fitted Stan model with posterior samples for the model parameters (see below).

stan_data The data list passed to Stan.

item_labels A vector of item labels.

The `stan_fit` component contains posterior samples for these ICM parameters:

Person Parameters:
- `E_loc` - Person proficiency for location
- `E_wid` - Person proficiency for width
- `a_loc` - Person scaling bias for location
- `b_loc` - Person shifting bias for location
- `b_wid` - Person shifting bias for width
- `rho_E` - Correlation between person proficiencies for location and width

Item Parameters:
- `Tr_loc` - Item consensus location in the logit-transformed space
- `Tr_wid` - Item consensus width in the logit-transformed space
- `Tr_loc_splx` - Item consensus location in the simplex space
- `Tr_wid_splx` - Item consensus width in the simplex space
- `Tr_L` - Item consensus lower bound
- `Tr_U` - Item consensus upper bound
- `Tr_splx` - Item consensus simplex representation
- `lambda_loc` - Item discernibility for location
- `lambda_wid` - Item discernibility for width
- `omega` - Item residual correlations between location and width
- `rho_lambda` - Correlation between item discernibilities for location and width

Hyperparameters:
- `mu_E` - Means for person proficiencies
- `sigma_I` - Standard deviations for person parameters
- `sigma_lambda` - Standard deviations for item discernibilities

Posterior Predictive Checks:
- `Y_ppc_loc` - Predicted responses for location in the logit-transformed space
- `Y_ppc_wid` - Predicted responses for width in the logit-transformed space
- `Y_ppc_splx` - Predicted responses in simplex space
- `Y_ppc_loc_splx` - Predicted location responses in the simplex space (mean of lower and upper bound)
- `Y_ppc_wid_splx` - Predicted width responses in the simplex space

References

Kloft, M., Siepe, B. S., & Heck, D. W. (2024). The Interval Truth Model: A Consensus Model for Continuous Bounded Interval Responses. doi:10.31234/osf.io/dzvw2

Examples

```
# Create minimal example data
df_simplex <- data.frame(
  x1 = c(0.3, 0.4, 0.2, 0.5),
  x2 = c(0.3, 0.2, 0.4, 0.2),
```

```
  x3 = c(0.4, 0.4, 0.4, 0.3)
)
id_person <- c(1, 1, 2, 2)
id_item <- c(1, 2, 1, 2)

# Fit ICM model (reduce iterations for faster example)
fit <- fit_icm(df_simplex, id_person, id_item, n_chains = 1,
               iter_sampling = 100, iter_warmup = 100,
               refresh = 0)

# Print summary of the fit
summary(fit)
```

`ilr` *Log-Ratio transformations for interval responses*

Description

Transform interval responses from the simplex space to the unbounded space using either Isometric Log-Ratio (ILR) or Sum Log-Ratio (SLR) transformations, as described by Smithson & Broomell (2024). These transformations preserve the dimensional conceptualization of the interval responses in terms of a location and a width. See also `inv_ilr()`, `inv_slr()` for the inverse transformations.

ILR

The ILR transformation equations are:

$$x_{loc} = \sqrt{\frac{1}{2}} \log\left(\frac{x_1}{x_3}\right)$$

$$x_{wid} = \sqrt{\frac{2}{3}} \log\left(\frac{x_2}{\sqrt{x_1 x_3}}\right)$$

SLR

The SLR transformation equations are:

$$x_{loc} = \log\left(\frac{x_1}{x_3}\right)$$

$$x_{wid} = \log\left(\frac{x_2}{x_1 + x_3}\right)$$

where (x_1, x_2, x_3) is the interval response in the simplex format and (x_{loc}, x_{wid}) are the transformed values representing the unbounded location and width.

Usage

```
ilr(simplex)

slr(simplex)
```

Arguments

`simplex`	A numeric vector that is a 2-simplex (3 elements that sum to 1) or a dataframe where each of the rows is a 2-simplex.

Value

A numeric vector with 2 elements, the unbounded interval location and width, or a dataframe where each of the rows is a numeric vector with these 2 elements.

References

Smithson, M., & Broomell, S. B. (2024). Compositional data analysis tutorial. Psychological Methods, 29(2), 362–378.

See Also

`inv_ilr()`, `inv_slr()`

Examples

```
# Generate some simplex data
simplex <- data.frame(rbind(c(.1, .2, .7), c(.4, .5, .1)))

# ILR transformation
ilr(simplex)

# SLR transformation
slr(simplex)
```

---

inv_ilr *Inverse Log-Ratio transformations for interval responses*

Description

Transform unbounded data back to the simplex space using either Isometric Log-Ratio (ILR) or Sum Log-Ratio (SLR) inverse transformations, as described by Smithson & Broomell (2024). These transformations are the inverse transformations of `ilr()` and `slr()`.

Inverse ILR

The inverse ILR transformation equations are:

$$x_1 = \frac{\exp(\sqrt{2}x_{loc})}{\exp(\sqrt{2}x_{loc}) + \exp(\sqrt{\frac{3}{2}}x_{wid} + \frac{x_{loc}}{\sqrt{2}}) + 1}$$

$$x_2 = \frac{\exp(\sqrt{\frac{3}{2}}x_{wid} + \frac{x_{loc}}{\sqrt{2}})}{\exp(\sqrt{2}x_{loc}) + \exp(\sqrt{\frac{3}{2}}x_{wid} + \frac{x_{loc}}{\sqrt{2}}) + 1}$$

$$x_3 = \frac{1}{\exp(\sqrt{2}x_{loc}) + \exp(\sqrt{\frac{3}{2}}x_{wid} + \frac{x_{loc}}{\sqrt{2}}) + 1}$$

Inverse SLR

The inverse SLR transformation equations are:

$$x_1 = \frac{\exp(x_{loc})}{(\exp(x_{loc}) + 1)(\exp(x_{wid}) + 1)}$$

$$x_2 = \frac{\exp(x_{wid})}{\exp(x_{wid}) + 1}$$

$$x_3 = \frac{1}{(\exp(x_{loc}) + 1)(\exp(x_{wid}) + 1)}$$

where (x_{loc}, x_{wid}) are the unbounded interval location and width and (x_1, x_2, x_3) is the resulting interval response in the simplex format.

Usage

```
inv_ilr(bvn)

inv_slr(bvn)
```

Arguments

bvn	A numeric vector containing an unbounded interval location and width or a dataframe where each of the rows consists of such a vector.

Value

A numeric vector containing a 2-simplex or a dataframe where each of the rows consists of such a vector.

References

Smithson, M., & Broomell, S. B. (2024). Compositional data analysis tutorial. Psychological Methods, 29(2), 362–378.

See Also

ilr(), slr()

Examples

```
# Generate some unbounded data
bvn <- data.frame(rbind(c(0, .2), c(-2, .4)))

# Inverse ILR transformation
inv_ilr(bvn)
```

```
# Inverse SLR transformation
inv_slr(bvn)
```

`itvl_to_splx` *Convert from interval bounds to simplex*

Description

Convert interval responses from interval bounds format to compostional/simplex format. See also `splx_to_itvl()` for the inverse transformation.

Usage

```
itvl_to_splx(interval_bounds, min = NULL, max = NULL)
```

Arguments

`interval_bounds`	A vector of length 2 representing the lower and upper bounds of an interval response or a data frame where each row contains such a vector.
`min`	Minimum of the original response scale.
`max`	Maximum of the original response scale.

Value

A numeric vector representing a 2-simplex if input is a vector, or a data frame where each row is a 2-simplex if input is a data frame.

See Also

`splx_to_itvl()`

Examples

```
interval_responses <- data.frame(rbind(c(.1,.5), c(.4,.7)))
itvl_to_splx(interval_responses, min = 0, max = 1)
```

`plot.icm_stanfit` *Plot Method for icm_stanfit Objects*

Description

This function provides a plot method for objects of class `icm_stanfit`.

Usage

```
## S3 method for class 'icm_stanfit'
plot(x, ...)
```

Arguments

`x`	An object of class `icm_stanfit`.
`...`	Additional arguments passed to the `plot_consensus` function.

Value

A plot generated by the `plot_consensus()` function.

See Also

`plot_consensus()`

Examples

```
# Create minimal example data
df_simplex <- data.frame(
  x1 = c(0.3, 0.4, 0.2, 0.5),
  x2 = c(0.3, 0.2, 0.4, 0.2),
  x3 = c(0.4, 0.4, 0.4, 0.3)
)
id_person <- c(1, 1, 2, 2)
id_item <- c(1, 2, 1, 2)

# Fit ICM model
fit <- fit_icm(df_simplex, id_person, id_item, n_chains = 1,
               iter_sampling = 100, iter_warmup = 100,
               refresh = 0)

# Plot using S3 method
plot(fit)
```

plot_consensus *Plot ICM Consensus Intervals*

Description

Plot consensus intervals estimated by the Interval Consensus Model (ICM) via `fit_icm()`.

Usage

```
plot_consensus(icm_stanfit, method = "median_bounds", CI = 0.95)
```

Arguments

`icm_stanfit`	An object of class `icm_stanfit` containing the Stanfit results.
`method`	A character string specifying the plotting method. Options are "median_bounds" (default) or "draws_distribution".
`CI`	A numeric value specifying the confidence interval for the "draws_distribution" method. Default is 0.95. This can also be a vector of length 2 for multiple confidence intervals.

Details

If the method is "median_bounds", the function uses posterior medians for the lower and upper bounds of the consensus intervals.

If the method is "draws_distribution", the function computes a consensus distribution for each consensus interval by uniformly sampling one value from the interval range for each posterior draw. From this distribution, a density plot is generated. As a rough guideline, the number of draws for this method should be above 1000.

Value

A ggplot2 object depicting the consensus interval estimates.

Examples

```
# Create minimal example data
df_simplex <- data.frame(
  x1 = c(0.3, 0.4, 0.2, 0.5),
  x2 = c(0.3, 0.2, 0.4, 0.2),
  x3 = c(0.4, 0.4, 0.4, 0.3)
)
id_person <- c(1, 1, 2, 2)
id_item <- c(1, 2, 1, 2)

# Fit ICM model
fit <- fit_icm(df_simplex, id_person, id_item, n_chains = 1,
               iter_sampling = 100, iter_warmup = 100,
               refresh = 0)
```

```
# Plot consensus intervals using median bounds
plot_consensus(fit, method = "median_bounds")
```

`plot_intervals` *Plot Intervals*

Description

Plot intervals from a data frame of interval bounds.

Usage

```
plot_intervals(df_interval_bounds, item_labels = NULL)
```

Arguments

`df_interval_bounds`	A data frame with two columns: the lower and upper bounds of the intervals.
`item_labels`	An optional vector of labels for the items. Its length must match the number of rows in `df_interval_bounds`.

Value

A ggplot object depicting the intervals.

Examples

```
df <- data.frame(lower = c(0.1, 0.3, 0.5), upper = c(0.4, 0.6, 0.8))
labels <- c("Item 1", "Item 2", "Item 3")

plot_intervals(df, item_labels = labels)
```

plot_intervals_cumulative *Plot Cumulative Intervals*

Description

Generate a cumulative interval plot based on the provided lower and upper bounds, cluster IDs, and other optional parameters.

Usage

```
plot_intervals_cumulative(
  lower,
  upper,
  cluster_id,
  truth = NA,
  min,
  max,
  facet_wrap = NULL,
  weighted = FALSE,
  show_quantiles = TRUE,
  ncol = 3
)
```

Arguments

lower	A numeric vector of lower bounds.
upper	A numeric vector of upper bounds.
cluster_id	A vector of cluster IDs corresponding to the intervals.
truth	A numeric vector of ground truth values. Default is NA.
min	Numeric. The minimum value for the x-axis.
max	Numeric. The maximum value for the x-axis.
facet_wrap	A logical value indicating whether to use facet wrapping. Default is NULL.
weighted	A logical value indicating whether the intervals should be weighted by their width. If TRUE, values are sampled uniformly within each interval. If FALSE, values are gathered using the same step size for all intervals. Default is FALSE.
show_quantiles	A logical value indicating whether to show quantiles on the plot. Default is TRUE.
ncol	The number of columns for facet wrapping. Default is 3.

Value

A ggplot object depicting the cumulative intervals.

Examples

```
# Example data
lower_bounds <- c(0.01, 0.3, 0.02, 0.4)
upper_bounds <- c(0.5, 0.96, 0.6, 0.8)
cluster_ids <- c(1, 1, 2, 2)
truth_values <- c(0.3, 0.3, 0.6, 0.6)

# Create cumulative interval plot
plot_intervals_cumulative(
  lower = lower_bounds,
  upper = upper_bounds,
  cluster_id = cluster_ids,
  truth = truth_values,
  min = 0,
  max = 1,
  weighted = FALSE
)
```

---

quantifiers *Verbal Quantifier Data*

Description

A subset of data from the data collected by Kloft & Heck (2024) containing the probability interval judgments for verbal quantifiers. The dataset is in the long format, with responses for the lower and upper interval bounds in separate columns.

Usage

```
quantifiers
```

Format

A data frame with 3,344 rows and 10 columns:

id_person Unique identifier for each person

id_item Unique identifier for each item

name_ger German name of the quantifier

name_en English name of the quantifier

truth True value of the quantifier if applicable

scale_min Minimum value of the response scale

scale_max Maximum value of the response scale

width_min Minimum possible interval width of the response scale

x_L Lower bound of the interval jugdment

x_U Upper bound of the interval jugdment

Source

https://osf.io/67vyj/

References

Kloft, M., & Heck, D. W. (2024). Discriminant validity of interval response formats: Investigating the dimensional structure of interval widths. **Educational and Psychological Measurement, 0** (0). doi:10.1177/00131644241283400

---

remove_zeros *Remove Zeros from Simplex*

Description

Remove zero-components from interval data in the simplex format.

Usage

```
remove_zeros(simplex, method = "rescaling", padding = 0.01)
```

Arguments

simplex	A numeric simplex vector of length 3 where the elements sum to 1, or a numeric matrix or data frame where each row is a simplex vector.
method	A character string specifying the method to remove zeros. Currently, only "rescaling" is supported. Default is "rescaling".
padding	A numeric value to add to each element of the simplex when using the "rescaling" method. Default is 0.01.

Details

Rescaling

The rescaling methods adds a small value (padding) to each element of the simplex and then divides by the row sum to close the composition.

Value

A numeric matrix with the same dimensions as the input simplex, with zeros removed according to the specified method.

Examples

```
# Example usage:
simplex <- matrix(c(0.2, 0.3, 0.5, 0, 0.5, 0.5), nrow = 2, byrow = TRUE)
remove_zeros(simplex)
```

splx_to_itvl *Convert from simplex to interval bounds*

Description

Convert from simplex/compostional format to interval bounds format. See also `itvl_to_splx()` for the inverse transformation.

Usage

```
splx_to_itvl(simplex, min = NULL, max = NULL)
```

Arguments

`simplex`	A numeric vector that is a 2-simplex (3 elements that sum to 1) or a data frame where each of the rows is a 2-simplex.
`min`	Minimum of the original response scale.
`max`	Maximum of the original response scale.

Value

A numeric vector with 2 elements representing the lower and upper bounds of the interval response, or a data frame where each of the rows contains such a vector.

See Also

`itvl_to_splx()`

Examples

```
responses <- data.frame(rbind(c(.1,.5,.4), c(.3,.4,.3)))
splx_to_itvl(responses, min = 0, max = 1)
```

summary.icm_stanfit *Summarize ICM Stanfit Object*

Description

This function provides a summary method for objects of class `icm_stanfit`. via a wrapper around the `extract_consensus()` function.

Usage

```
## S3 method for class 'icm_stanfit'
summary(object, ...)
```

Arguments

object	An object of class icm_stanfit.
...	Additional arguments (currently not used).

Value

A table with posterior medians and credible intervals for the consensus intervals.

See Also

extract_consensus()

Examples

```
# Create minimal example data
df_simplex <- data.frame(
  x1 = c(0.3, 0.4, 0.2, 0.5),
  x2 = c(0.3, 0.2, 0.4, 0.2),
  x3 = c(0.4, 0.4, 0.4, 0.3)
)
id_person <- c(1, 1, 2, 2)
id_item <- c(1, 2, 1, 2)

# Fit ICM model
fit <- fit_icm(df_simplex, id_person, id_item, n_chains = 1,
               iter_sampling = 100, iter_warmup = 100,
               refresh = 0)

# Get summary
summary(fit)
```

---

theme_icm *Custom ggplot2 Theme for* intervalpsych

Description

Creates a custom ggplot2 theme for intervalpsych visualizations.

Usage

```
theme_icm(hide_axis_text_y = FALSE, base_size = 12)
```

Arguments

hide_axis_text_y	Logical. If TRUE, the y-axis text and ticks will be hidden. Default is FALSE.
base_size	Numeric. Base font size for the theme. Default is 12.

Value

A ggplot2 theme object.

Examples

```
# Create sample interval data
df_intervals <- data.frame(
  lower = c(0.1, 0.3, 0.2, 0.4),
  upper = c(0.5, 0.7, 0.6, 0.8)
)
item_labels <- c("Item A", "Item B", "Item C", "Item D")

# Basic usage
plot_intervals(df_intervals, item_labels) +
  theme_icm()

# Hide y-axis text
plot_intervals(df_intervals, item_labels) +
  theme_icm(hide_axis_text_y = TRUE)

# Custom base size
plot_intervals(df_intervals, item_labels) +
  theme_icm(base_size = 14)
```

Index