Package 'latent2likert'

June 24, 2024

Type Package

Title Converting Latent Variables into Likert Scale Responses

Version 1.2.1

Description Effectively simulates the discretization process inherent to Likert scales while minimizing distortion. It converts continuous latent variables into ordinal categories to generate Likert scale item responses. Particularly useful for accurately modeling and analyzing survey data that use Likert scales, especially when applying statistical techniques that require metric data.

License MIT + file LICENSE

URL https://lalovic.io/latent2likert/

BugReports https://github.com/markolalovic/latent2likert/issues/

Depends R (>= 3.5)

Imports graphics, mvtnorm, sn, stats, utils

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.3.1

NeedsCompilation no

Author Marko Lalovic [aut, cre]

Maintainer Marko Lalovic <marko@lalovic.io>

Repository CRAN

Date/Publication 2024-06-24 15:50:02 UTC

Contents

discretize_density 2
estimate_params 4
part_bfi 5
plot_likert_transform 6
response_prop 7
rlikert 7
simulate_likert 9

Index **11**

`discretize_density` *Discretize Density*

Description

Transforms the density function of a continuous random variable into a discrete probability distribution with minimal distortion using the Lloyd-Max algorithm.

Usage

```
discretize_density(density_fn, n_levels, eps = 1e-06)
```

Arguments

`density_fn`	probability density function.
`n_levels`	cardinality of the set of all possible outcomes.
`eps`	convergence threshold for the algorithm.

Details

The function addresses the problem of transforming a continuous random variable X into a discrete random variable Y with minimal distortion. Distortion is measured as mean-squared error (MSE):

$$\mathrm{E}\left[(X-Y)^2\right] = \sum_{k=1}^{K} \int_{x_{k-1}}^{x_k} f_X(x)\,(x-r_k)^2\,dx$$

where:

f_X is the probability density function of X,

K is the number of possible outcomes of Y,

x_k are endpoints of intervals that partition the domain of X,

r_k are representation points of the intervals.

This problem is solved using the following iterative procedure:

1. Start with an arbitrary initial set of representation points: $r_1 < r_2 < \cdots < r_K$.

2. Repeat the following steps until the improvement in MSE falls below given ε.
3. Calculate endpoints as $x_k = (r_{k+1} + r_k)/2$ for each $k = 1, \ldots, K-1$ and set x_0 and x_K to $-\infty$ and ∞, respectively.
4. Update representation points by setting r_k equal to the conditional mean of X given $X \in (x_{k-1}, x_k)$ for each $k = 1, \ldots, K$.

With each execution of step (3) and step (4), the MSE decreases or remains the same. As MSE is nonnegative, it approaches a limit. The algorithm terminates when the improvement in MSE is less than a given $\varepsilon > 0$, ensuring convergence after a finite number of iterations.

This procedure is known as Lloyd-Max's algorithm, initially used for scalar quantization and closely related to the k-means algorithm. Local convergence has been proven for log-concave density functions by Kieffer. Many common probability distributions are log-concave including the normal and skew normal distribution, as shown by Azzalini.

Value

A list containing:

prob discrete probability distribution.

endp endpoints of intervals that partition the continuous domain.

repr representation points of the intervals.

dist distortion measured as the mean-squared error (MSE).

References

Azzalini, A. (1985). A class of distributions which includes the normal ones. *Scandinavian Journal of Statistics* **12(2)**, 171–178.

Kieffer, J. (1983). Uniqueness of locally optimal quantizer for log-concave density and convex error function. *IEEE Transactions on Information Theory* **29**, 42–47.

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory* **28 (2)**, 129–137.

Examples

```
discretize_density(density_fn = stats::dnorm, n_levels = 5)
discretize_density(density_fn = function(x) {
  2 * stats::dnorm(x) * stats::pnorm(0.5 * x)
}, n_levels = 4)
```

`estimate_params` *Estimate Latent Parameters*

Description

Estimates the location and scaling parameters of the latent variables from existing survey data.

Usage

```
estimate_params(data, n_levels, skew = 0)
```

Arguments

`data`	survey data with columns representing individual items. Apart from this, `data` can be of almost any class such as "data.frame" "matrix" or "array".
`n_levels`	number of response categories, a vector or a number.
`skew`	marginal skewness of latent variables, defaults to 0.

Details

The relationship between the continuous random variable X and the discrete probability distribution p_k, for $k = 1, \ldots, K$, can be described by a system of non-linear equations:

$$p_k = F_X\left(\frac{x_{k-1} - \xi}{\omega}\right) - F_X\left(\frac{x_k - \xi}{\omega}\right) \quad \text{for } k = 1, \ldots, K$$

where:

F_X is the cumulative distribution function of X,

K is the number of possible response categories,

x_k are the endpoints defining the boundaries of the response categories,

p_k is the probability of the k-th response category,

ξ is the location parameter of X,

ω is the scaling parameter of X.

The endpoints x_k are calculated by discretizing a random variable Z with mean 0 and standard deviation 1 that follows the same distribution as X. By solving the above system of non-linear equations iteratively, we can find the parameters that best fit the observed discrete probability distribution p_k.

The function `estimate_params`:

- Computes the proportion table of the responses for each item.
- Estimates the probabilities p_k for each item.
- Computes the estimates of ξ and ω for each item.
- Combines the estimated parameters for all items into a table.

Value

A table of estimated parameters for each latent variable.

See Also

discretize_density for details on calculating the endpoints, and part_bfi for example of the survey data.

Examples

```
data(part_bfi)
vars <- c("A1", "A2", "A3", "A4", "A5")
estimate_params(data = part_bfi[, vars], n_levels = 6)
```

---

part_bfi *Agreeableness and Gender Data*

Description

This dataset is a cleaned up version of a small part of bfi dataset from psychTools package. It contains responses to the first 5 items of the agreeableness scale from the International Personality Item Pool (IPIP) and the gender attribute. It includes responses from 2800 subjects. Each item was answered on a six point Likert scale ranging from 1 (very inaccurate), to 6 (very accurate). Gender was coded as 0 for male and 1 for female. Missing values were addressed using mode imputation.

Usage

```
data(part_bfi)
```

Format

An object of class "data.frame" with 2800 observations on the following 6 variables:

A1 Am indifferent to the feelings of others.

A2 Inquire about others' well-being.

A3 Know how to comfort others.

A4 Love children.

A5 Make people feel at ease.

gender Gender of the respondent.

Source

International Personality Item Pool (https://ipip.ori.org)

https://search.r-project.org/CRAN/refmans/psychTools/html/bfi.html

References

Revelle, W. (2024). Psych: Procedures for Psychological, Psychometric, and Personality Research. *Evanston, Illinois: Northwestern University.* https://CRAN.R-project.org/package=psych

Examples

```
data(part_bfi)
head(part_bfi)
```

---

plot_likert_transform *Plot Transformation*

Description

Plots the densities of latent variables and the corresponding transformed discrete probability distributions.

Usage

```
plot_likert_transform(n_items, n_levels, mean = 0, sd = 1, skew = 0)
```

Arguments

n_items	number of Likert scale items (questions).
n_levels	number of response categories for each Likert item. Integer or vector of integers.
mean	means of the latent variables. Numeric or vector of numerics. Defaults to 0.
sd	standard deviations of the latent variables. Numeric or vector of numerics. Defaults to 1.
skew	marginal skewness of the latent variables. Numeric or vector of numerics. Defaults to 0.

Value

NULL. The function produces a plot.

Examples

```
plot_likert_transform(n_items = 3, n_levels = c(3, 4, 5))
plot_likert_transform(n_items = 3, n_levels = 5, mean = c(0, 1, 2))
plot_likert_transform(n_items = 3, n_levels = 5, sd = c(0.8, 1, 1.2))
plot_likert_transform(n_items = 3, n_levels = 5, skew = c(-0.5, 0, 0.5))
```

response_prop *Calculate Response Proportions*

Description

Returns a table of proportions for each possible response category.

Usage

```
response_prop(data, n_levels)
```

Arguments

data	numeric vector or matrix of responses.
n_levels	number of response categories.

Value

A table of response category proportions.

Examples

```
data <- c(1, 2, 2, 3, 3, 3)
response_prop(data, n_levels = 3)

data_matrix <- matrix(c(1, 2, 2, 3, 3, 3), ncol = 2)
response_prop(data_matrix, n_levels = 3)
```

rlikert *Generate Random Responses*

Description

Generates an array of random responses to Likert-type questions based on specified latent variables.

Usage

```
rlikert(size, n_items, n_levels, mean = 0, sd = 1, skew = 0, corr = 0)
```

Arguments

size	number of observations.
n_items	number of Likert scale items (number of questions).
n_levels	number of response categories for each item. Integer or vector of integers.
mean	means of the latent variables. Numeric or vector of numerics. Defaults to 0.
sd	standard deviations of the latent variables. Numeric or vector of numerics. Defaults to 1.
skew	marginal skewness of the latent variables. Numeric or vector of numerics. Defaults to 0.
corr	correlations between latent variables. Can be a single numeric value representing the same correlation for all pairs, or an actual correlation matrix. Defaults to 0.

Value

A matrix of random responses with dimensions `size` by `n_items`. The column names are `Y1`, `Y2`, `...`, `Yn` where `n` is the number of items. Each entry in the matrix represents a Likert scale response, ranging from 1 to `n_levels`.

Examples

```
# Generate responses for a single item with 5 levels
rlikert(size = 10, n_items = 1, n_levels = 5)

# Generate responses for three items with different levels and parameters
rlikert(
  size = 10, n_items = 3, n_levels = c(4, 5, 6),
  mean = c(0, -1, 0), sd = c(0.8, 1, 1), corr = 0.5
)

# Generate responses with a correlation matrix
corr <- matrix(c(
  1.00, -0.63, -0.39,
  -0.63, 1.00, 0.41,
  -0.39, 0.41, 1.00
), nrow = 3)
data <- rlikert(
  size = 1000, n_items = 3, n_levels = c(4, 5, 6),
  mean = c(0, -1, 0), sd = c(0.8, 1, 1), corr = corr
)
```

`simulate_likert` *Simulate Likert Scale Item Responses*

Description

Simulates Likert scale item responses based on a specified number of response categories and the centered parameters of the latent variable.

Usage

```
simulate_likert(n_levels, cp)
```

Arguments

`n_levels`	number of response categories for the Likert scale item.
`cp`	centered parameters of the latent variable. Named vector including mean (`mu`), standard deviation (`sd`), and skewness (`skew`). Skewness must be between -0.95 and 0.95.

Details

The simulation process uses the following model detailed by Boari and Nai-Ruscone. Let X be the continuous variable of interest, measured using Likert scale questions with K response categories. The observed discrete variable Y is defined as follows:

$$Y = k, \quad \text{if} \quad x_{k-1} < X \leq x_k \quad \text{for} \quad k = 1, \ldots, K$$

where x_k, $k = 0, \ldots, K$ are endpoints defined in the domain of X such that:

$$-\infty = x_0 < x_1 < \cdots < x_{K-1} < x_K = \infty.$$

The endpoints dictate the transformation of the density f_X of X into a discrete probability distribution:

$$\Pr(Y = k) = \int_{x_{k-1}}^{x_k} f_X(x)\, dx \quad \text{for} \quad k = 1, \ldots, K.$$

The continuous latent variable is modeled using a skew normal distribution. The function `simulate_likert` performs the following steps:

- Ensures the centered parameters are within the acceptable range.
- Converts the centered parameters to direct parameters.
- Defines the density function for the skew normal distribution.
- Computes the probabilities for each response category using optimal endpoints.

Value

A named vector of probabilities for each response category.

References

Boari, G. and Nai Ruscone, M. (2015). A procedure simulating Likert scale item responses. *Electronic Journal of Applied Statistical Analysis* **8(3)**, 288–297. doi:10.1285/i20705948v8n3p288

See Also

`discretize_density` for details on how to calculate the optimal endpoints.

Examples

```
cp <- c(mu = 0, sd = 1, skew = 0.5)
simulate_likert(n_levels = 5, cp = cp)
cp2 <- c(mu = 1, sd = 2, skew = -0.3)
simulate_likert(n_levels = 7, cp = cp2)
```

Index

∗ **datasets**
part_bfi, 5

discretize_density, 2, *5*, *10*

estimate_params, 4

part_bfi, *5*, 5
plot_likert_transform, 6

response_prop, 7
rlikert, 7

simulate_likert, 9