Package 'support.BWS'

May 9, 2026

Type Package

Title Tools for Case 1 Best-Worst Scaling

Version 0.4-6

Date 2023-03-29

Author Hideo Aizaki

Maintainer Hideo Aizaki <azk-r@spa.nifty.com>

Description Provides basic functions that support an implementation of object case (Case 1) best-worst scaling: a function for converting a two-level orthogonal main-effect design/balanced incomplete block design into questions; two functions for creating a data set suitable for analysis; a function for calculating count-based scores; a function for calculating shares of preference; and a function for generating artificial responses to questions. See Louviere et al. (2015) <doi:10.1017/CBO9781107337855> for details on best-worst scaling, and Aizaki and Fogarty (2023) <doi:10.1016/j.jocm.2022.100394> for the package.

URL http://lab.agr.hokudai.ac.jp/spmur/
http://lab.agr.hokudai.ac.jp/nmvr/

License GPL (>= 2)

Suggests DoE.base, crossdes, survival, mlogit, gmnl, apollo

NeedsCompilation no

Repository CRAN

Date/Publication 2023-03-29 07:40:02 UTC

Contents

`support.BWS-package` *Tools for Case 1 best-worst scaling*

Description

The package provides three basic functions that support an implementation of object case (Case 1) best-worst scaling: one for converting a two-level orthogonal main-effect design/balanced incomplete block design into questions; one for creating a data set suitable for analysis; and one for calculating count-based scores.

Details

Object case (or Case 1) best-worst scaling (BWS), or maximum difference scaling (MaxDiff) (Finn and Louviere 1992) is a stated preference method. After listing the items (objects) evaluated by respondents, a number of different subsets of the items are constructed from the list using the design of experiments. Each of the subsets is presented as a choice set to respondents, who are then asked to select the best (or most important) item and the worst (or least important) item in the choice set. This question is repeated until all the subsets are evaluated.

There are two methods to construct choice sets for object case BWS: one uses a two-level orthogonal main-effect design (OMED) (Finn and Louviere, 1992) and the other uses a balanced incomplete block design (BIBD) (Auger et al., 2007). The first method uses a two-level OMED with T columns, where T is the total number of items evaluated: each column corresponds to an item and each row corresponds to a question. There are two values in the two-level OMEDs (e.g., 1 and 2): one value is interpreted as an item being "absent" from the corresponding column and the other as being "present." In this way, we can decide which items are assigned to each question: for example, if a row in a two-level OMED contains a value of 2 (which means "present" here) in the 1st, 5th, and 8th columns, and a value of 1 in the other columns, these three items are presented in a question corresponding to the row.

The second method uses a BIBD, which is a category of designs in which a subset of treatments is assigned to each block. The features of a BIBD are expressed by "number of treatments (items)," "size of a block (number of items per question)," "number of blocks (questions)," "number of replications of each treatment (item)," and "frequency that each pair of treatments (items) appears in the same block (question)." Each row corresponds to a question; the number of columns is equal to the number of items per question; and the level values correspond to item identification numbers. For example, assume that there are seven items, ITEM1, ITEM2, ..., and ITEM7, and a BIBD with seven rows, four columns, and seven level values (1, 2, ..., 7). Under these assumptions, if a row in the BIBD contains values of 1, 4, 6, and 7, a set containing ITEM1, ITEM4, ITEM6, and ITEM7 is presented in a question corresponding to the row.

There are two approaches to analyzing responses to object case BWS questions: a counting approach and a modeling approach. The counting approach calculates several types of scores on the basis of the number of times (the frequency or count) item i is selected as the best (B_{in}) and the worst (W_{in}) among all the questions for respondent n. These scores are roughly divided into two categories: disaggregated (individual-level) scores and aggregated (total-level) scores (Finn and Louviere, 1992; Lee et al., 2007; Cohen, 2009; Mueller and Rungie, 2009).

The first category includes a disaggregated BW score and its standardized score:

$$BW_{in} = B_{in} - W_{in},$$

$$std.BW_{in} = \frac{BW_{in}}{r},$$

where r is the frequency with which item i appears across all questions.

The frequency with which item i is selected as the best across all questions for N respondents is defined as B_i. Similarly, the frequency with which item i is selected as the worst is defined as W_i (i.e., $B_i = \sum_{n=1}^{N} B_{in}$, $W_i = \sum_{n=1}^{N} W_{in}$). The second category includes the aggregated versions of BW_{in} and $std.BW_{in}$, as well as the square root of the ratio of B_i to W_i and its standardized score:

$$BW_i = B_i - W_i,$$

$$std.BW_i = \frac{BW_i}{Nr},$$

$$sqrt.BW_i = \sqrt{\frac{B_i}{W_i}},$$

$$std.sqrt.BW_i = \frac{sqrt.BW_i}{max.sqrt.BW_i},$$

where $max.sqrt.BW_i$ is the maximum value of $sqrt.BW_i$.

The modeling approach uses discrete choice models to analyze responses. In the package, this approach is based on the understanding of respondents' behavior in the following situation (Finn and Louviere, 1992; Auger et al., 2007). Suppose that m items exist in a choice set (a question). The number of possible pairs in which item i is selected as the best and item j is selected as the worst ($i \neq j$) from m items is $m \times (m-1)$. Respondents are assumed to have a utility (v) for each item. Further, they are assumed to select item i as the best and item j as the worst because the difference in utility between i and j represents the greatest utility difference (the maxdiff model). Under these assumptions, the probability of selecting item i as the best and item j as the worst is expressed as a conditional logit model:

$$Pr(i,j) = \frac{\exp(v_i - v_j)}{\sum_{k=1}^{m} \sum_{l=1,k\neq l}^{m} \exp(v_k - v_l)}.$$

A share of preference for item i (SP_i) based on the conditional logit model choice rule is as follows (Cohen, 2003; Cohen and Neira, 2004; Lusk and Briggeman, 2009):

$$SP_i = \frac{\exp(v_i)}{\sum_{t=1}^{T} \exp(v_t)}.$$

Version 0.2-0 and later versions of the package are also available for the marginal and marginal sequential models (Hensher et al., 2015, Appendix 6B) in the modeling approach.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Numbers JP25450341, JP16K07886, and JP20K06251.

Author(s)

Hideo Aizaki

References

Aizaki H, Fogarty J (2023) R packages and tutorial for case 1 best-worst scaling. *Journal of Choice Modelling*, **46**, 100394.

Aizaki H, Nakatani T, Sato K (2014) *Stated Preference Methods Using R*. CRC Press.

Auger P, Devinney TM, Louviere JJ (2007) Using best-worst scaling methodology to investigate consumer ethical beliefs across countries. *Journal of Business Ethics*, **70**, 299–326.

Cohen E (2009) Applying best-worst scaling to wine marketing. *International Journal of Wine Business Research*, **21**(1), 8–23.

Cohen SH (2003) Maximum difference scaling: Improved measures of importance and preference for segmentation. *Sawtooth Software Research Paper Series*, 1–17. https://sawtoothsoftware.com/resources/technical-papers/maximum-difference-scaling-improved-measures-of-importance-and-pref

Cohen S, Neira L (2004) Measuring preference for product benefits across countries: Overcoming scale usage bias with maximum difference scaling. *Excellence in International Research*, 1–22.

Finn A, Louviere JJ (1992) Determining the appropriate response to evidence of public concern: The case of food safety. *Journal of Public Policy & Marketing*, **11**(2), 12–25.

Hensher DA, Rose JM, Greene WH (2015) *Applied Choice Analysis*. 2nd edition. Cambridge University Press.

Hess S, Palma D (2019a) Apollo: a flexible, powerful and customisable freeware package for choice model estimation and application. *Journal of Choice Modelling*, **32**, 100170.

Hess S, Palma D (2019b) Apollo version 0.0.9, user manual, http://www.apollochoicemodelling.com/.

Lee JA, Soutar GN, Louviere J (2007) Measuring values using best-worst scaling: The LOV example. *Psychology & Marketing*, **24**(12), 1043–1058.

Lusk JL, Briggeman BC (2009) Food values. *American Journal of Agricultural Economics*, **91**(1), 184–196.

Louviere JJ, Flynn TN, Marley AAJ (2015) *Best-Worst Scaling: Theory, Methods and Applications*. Cambridge University Press.

Mueller S, Rungie C (2009) Is there more information in best-worst choice data?: Using the attitude heterogeneity structure to identify consumer segments. *International Journal of Wine Business Research*, **21**(1), 24–40.

`bws.apollo` *Converting a dataset into that for Apollo*

Description

This function converts a dataset generated using `bws.dataset` into that in a format suitable for a modeling analysis using the **apollo** package.

Usage

```
bws.apollo(data, detail = FALSE)
```

Arguments

data	A data frame containing the output from the function bws.dataset.
detail	A logical value: if TRUE, the dataset is returned in a detailed format; and if FALSE (default), the dataset is returned in a simple format.

Details

This function converts a dataset generated using bws.dataset into that in a format suitable for the modeling analysis using the **apollo** package. The **apollo** is a comprehensive package for choice modelling. The format of a dataset suitable for **apollo** differs from that for clogit in **survival**, mlogit in **mlogit**, and gmnl in **gmnl**. The former assumes that a row corresponds to a question, while the latter assumes that a row corresponds to an alternative. Therefore, the output from bws.dataset, which is suitable for clogit, mlogit, and gmnl, has to be converted into a dataset for the modeling analysis using **apollo**.

Value

The function bws.apollo returns a data frame that contains variables id, Q, and RES, as well as independent variables in a simple format. While the variable id is assigned to indivID used in the section "Definition of core settings" of Apollo's model estimation script, the variable RES is assigned to choiceVar used in the section "Likelihood definition". The serial number of alternatives is appended to the tail of the independent variable name: e.g., Item1.1 for the first alternative, Item1.2 for the second alternative, and Item1.3 for the third alternative. The independent variables are also used in the section "Likelihood definition". The detailed format dataset includes variables that are not used for the analysis with Apollo. For details on Apollo's model estimation script, refer to the vignette (https://cran.r-project.org/package=apollo) and the the user manual (Hess and Palma 2019b).

Author(s)

Hideo Aizaki

See Also

support.BWS-package, bws.dataset, clogit, mlogit, gmnl, apollo_estimate, apollo_mnl

---

bws.count *Calculating count-based BW scores*

Description

This function calculates various BW scores on the basis of a counting approach.

Usage

```
bws.count(data, cl = 1)

## S3 method for class 'bws.count'
print(x, digits = max(3, getOption("digits") - 3), scientific = FALSE, ...)

## S3 method for class 'bws.count2'
plot(x, score = c("bw", "b", "w"), pos = 1, xlab = NULL, ylab = NULL, subset, ...)

## S3 method for class 'bws.count2'
barplot(height, score = c("bw", "b", "w", "sbw"), mfrow = NULL, mean = FALSE,
error.bar = NULL, conf.level = 0.95, subset, sort = FALSE, ...)

## S3 method for class 'bws.count2'
sum(x, ...)

## S3 method for class 'bws.count2'
mean(x, ...)

## S3 method for class 'bws.count2'
summary(object, sort = FALSE, subset, ...)

## S3 method for class 'summary.bws.count2'
print(x, digits = max(3, getOption("digits") - 3), scientific = FALSE, ...)
```

Arguments

data	A data frame containing the output from bws.dataset().
cl	A value describing the S3 class of the object created by this function: 1 for the S3 class "bws.count", and 2 for the S3 class "bws.count2".
x, height, object	An object of the S3 class "bws.count" or "bws.count2".
digits	The number of significant digits. See the format function.
scientific	Scores are encoded in scientific format. See the format function.
score	A character showing a type of the output from this function: "b" is assigned to this argument when the output is based on best scores, "w" is assigned when it is based on worst scores, "bw" is assigned when it is based on best-minus-worst scores, or "sbw" is assigned when it is based on standardized best-minus-worst scores.
pos	A value showing a position of labels for points in the plot. See the argument pos of the text function.
xlab	A character showing a label for the x axis.
ylab	A character showing a label for the y axis.
mfrow	A two-element vector c(nr, nc): bar plots will be drawn in an nr-by-nc array on the device by row.

mean	A logical value denoted by TRUE when drawing a bar plot of the aggregated standardized BW scores or FALSE (default) when drawing bar plots of B, W, or BW scores.
error.bar	A character shoiwng a type of error bar adding on the bar plot of the aggregated standardized BW scores: "sd" for the standard deviation; "se" for the standard error; "ci" for the confidence interval; and NULL (default) for none.
conf.level	A value showing the confidence level when adding the confidence interval on the bar plot using error.bar.
subset	A logical expression indicating a subset of observations to be used.
sort	A logical value denoted by TRUE when sorting barplots or items according to mean scores or FALSE (default).
...	Arguments passed to a function used internally.

Details

This function calculates various BW scores on the basis of the counting approach. For details on the scores, refer to support.BWS-package.

When using this function with the argument cl = 1, it returns an object of the S3 class "bws.count", containing disaggregated scores and aggregated scores in list format. The first category includes disaggregated best (B), worst (W), best-minus-worst (BW), and standardized BW scoers. The second category includes aggregated B, W, BW, and standardized BW scores as well as the square root of the ratio of the aggregated B to the aggregated W and its standardized scores. The generic function print() is available for the S3 class "bws.count". The print() shows a summary of disaggregated scores and a table of aggregated scores.

When using this function with the argument cl = 2, it returns an object of the S3 class "bws.count2", which inherits from the S3 class "data.frame", including disaggregated B, W, BW, and standardized BW scores, respondent identification number variable, and respondent characteristic variables. The generic functions such as plot(), barplot(), sum(), mean(), and summary() are available for the S3 class "bws.count2". The plot() draws the relationship between means and standard deviations of B, W, or BW scores. The barplot() draws the bar plot of the aggregated standardized BW scores or the bar plots of B, W, or BW scores. The sum() returns the aggregated B, W, and BW scores in data frame format. The mean() returns means of B, W, BW, and standardized BW scores in data frame format. The summary() calculates (1) aggregated B, W, BW, and standardized BW scores, (2) item ranks based on the BW score, (3) means of B, W, BW, and standardized BW scores, and (4) the square root of the aggregated B to the aggregated W and its standardized scores.

Value

The output from bws.count() with the argument cl = 1 is an object of the S3 class "bws.count", containing three components:

A list disaggregate contains five objects related to disaggregated scores.

ID	A vector showing the respondent's identification number.
B	A matrix showing the number of times item i is selected as the best by each respondent.
W	A matrix showing the number of times item i is selected as the worst by each respondent.

BW	A matrix showing the difference between B and W for item i per respondent.
stdBW	A matrix showing standardized BW.

A data frame aggregate contains aggregated scores across all respondents.

B	A variable showing the number of times item i is selected as the best across all respondents.
W	A variable showing the number of times item i is selected as the worst across all respondents.
BW	A variable showing the difference between B and W for item i across all respondents.
stdBW	A variable showing standardized BW.
sqrtBW	A variable showing the square root of the ratio of B to W for item i across all respondents.
std.sqrtBW	A variable showing the standardized sqrtBW.

A list information contains basic information related to the BWS questions.

nrespondents	A variable showing the number of respondents.
nitems	A variable showing the number of items.
fitem	A variable showing the frequency of each item in the choice sets.
vnames	A variable showing the names of each item.

The output from bws.count() with the argument cl = 2, which is an object of the S3 class "bws.count2", is a data frame containing respondent identification number variable, B score variables, W score variables, BW score variables, standardized BW score variables, and respondent characteristic variables. These scores are calculated by each respondent.

Note that the S3 class "bws.count" would be replaced by the S3 class "bws.count2" in future.

Author(s)

Hideo Aizaki

See Also

support.BWS-package, bws.dataset

Examples

```
## See examples in bws.dataset()
```

bws.dataset *Creating a data set suitable for case 1 best-worst scaling analysis*

Description

This function creates a data set used for `bws.count()` in **support.BWS** and `clogit()` in **survival**.

Usage

```
bws.dataset(data = NULL, response.type = 1,
            choice.sets, design.type = 1,
            item.names = NULL, row.renames = TRUE,
            id = NULL, response = NULL,
            model = "maxdiff", delete.best = FALSE,
            version = NULL, respondent.dataset = NULL)
```

Arguments

`data`	A data frame including responses to BWS questions (i.e., a survey/respondent data set).
`response.type`	A value describing the format of the response variables: `1` if the response variables follow a row number format, and `2` if they follow an item number format.
`choice.sets`	A data frame or matrix containing choice sets.
`design.type`	A value describing how to design the choice sets: `1` if the design assigned to `choice.sets` is a two-level OMED, and `2` if it is a BIBD.
`item.names`	A vector containing the names of items: if it takes `NULL`, default item names (i.e., `ITEM1`, `ITEM2`, ...) are used in the resultant data set.
`row.renames`	A logical variable describing whether or not the row names of a data set created by this function are changed. If `TRUE`, integer values are assigned to the row names starting from `1`. If `FALSE`, the row names are not changed.
`id`	A character showing the name of the respondent identification number variable used in the respondent data set.
`response`	A vector containing the names of response variables in the respondent data set, showing the best and worst items selected in each BWS question.
`model`	A character showing a type of data set created by this function: `"maxdiff"` for the maxdiff model; `"marginal"` for the marginal model; and `"sequential"` for the marginal sequential model.
`delete.best`	A logical value denoted by `TRUE` when deleting an item selected as the best in the worst choice set (that is, using a marginal sequential model) or `FALSE` when not doing so. The argument is deprecated. Please use the argument `model` instead.
`version`	A character showing the name of the version variable used in the respondent dataset and choice sets.
`respondent.dataset`	A data frame containing a respondent data set. The argument is deprecated. Please use the argument `data` instead.

Details

The respondent data set, in which each row corresponds to a respondent, has to be organized by users. The data set must contain the id variable in the first column, denoting the respondent's identification number, and the response variables in the subsequent columns, each indicating which items are selected as the best and the worst for each question. Although the names of the id and response variables are up to the discretion of the user, the response variables must be constructed such that the best alternates with the worst by question. For example, when there are seven BWS questions, the variables are `B1 W1 B2 W2 ... B7 W7`; here, `Bi` and `Wi` show which items are selected as the best and the worst in the `i`th question.

There are two types of data format related to response variables: one is a row number format, and the other is an item number format. In the former, the row numbers of the items selected as the best and the worst are stored in the response variables. In the latter, item numbers are stored in the response variables.

The arguments `choice.sets`, `design.type`, and `item.names` are the same as those in the `bws.questionnaire()` function. However, `item.names` can take `NULL` (default), when default item names (i.e., `ITEM1`, `ITEM2`, ...) are used in the resultant data set. Further, the order of questions in the choice sets has to be the same as that in the respondent data set.

The argument `version` is set when two or more versions of the choice sets are used. The version variable must be included in the respondent dataset and choice sets. The variable denotes the serial integer number of versions starting from 1.

Note that this function in version 0.2-0 and later versions of the package can create a data set for the marginal (sequential) model as well as that for the maxdiff model: `"maxdiff"` is assigned to the argument `model` when the maxdiff model is used for the analysis; `"marginal"` is assigned when the marginal model is used; and `"sequential"` is assigned when the marginal sequential model is used. Furthermore, the argument `delete.best` is deprecated: please use the argument `model = "sequential"` instead.

Value

This function returns a data set in data frame format for the maxdiff model or one for the marginal (sequential) model. The data set for the maxdiff model contains the following variables:

`ID`	A respondent's identification number, assigned according to the id variable in the respondent data set.
`Q`	A serial number of BWS questions.
`PAIR`	A serial number of possible pairs of the best and worst items for each question.
`BEST`	An item number treated as the best in the possible pairs of the best and worst items for each question.
`WORST`	An item number treated as the worst in the possible pairs of the best and worst items for each question.
`RES.B`	An item number selected as the best by respondents.
`RES.W`	An item number selected as the worst by respondents.
`ITEMj`	State variables related to the possible pairs of the best and worst items: `1` if item `j` is treated as the best in the possible pair, `-1` if item `j` is treated as the worst in

the possible pair, and 0 otherwise. These variables are used as independent variables in the model formula of the clogit function in **survival** when analyzing responses to BWS questions.

RES — Responses to BWS questions: TRUE if a possible pair of the best and worst items is selected by respondents and FALSE otherwise. This variable is used as a dependent variable in the model formula of the clogit function in **survival** when analyzing responses to BWS questions.

STR — A stratification variable identifying each combination of respondent and question. This variable is also used in formula of the clogit function with the strata function.

The data set for the marginal (sequential) model contains the variables ID, Q, RES.B, RES.W, and STR mentioned above and the following variables:

ALT — A serial number of alternatives for each question.

BW — A state variable that takes the value of 1 for the possible best items and -1 for the possible worst items.

Item — An item number treated as the possible best or worst items for each question.

RES — Response to BWS questions: TRUE if a possible best or worst item is selected by respondents and FALSE otherwise. This variable is used as a dependent variable in the model formula of the clogit function in **survival** when analyzing responses to BWS questions.

ITEMj — State variables that takes the value of 1 for the possible best items and -1 for the possible worst items. These variables are used as independent variables in the model formula of the clogit function in **survival** when analyzing responses to BWS questions.

Furthermore, the resultant data set includes respondent characteristic variables when the respondent data set has those variables.

Author(s)

Hideo Aizaki

See Also

support.BWS-package, oa.design, find.BIB, isGYD, clogit, strata

Examples

```
## Not run:
## load packages
require(DoE.base) # include oa.design() used to generate a two-level OMED
require(crossdes) # include find.BIB() used to generate a BIBD
require(survival) # include clogit() used to analyze responses

if(getRversion() >= "3.6.0") RNGkind(sample.kind = "Rounding")

## example 1: BWS using a two-level OMED
```

```
## suppose that ten respondents answered twelve BWS questions valuing nine items

# create a two-level OMED with nine factors
set.seed(123) # set seed for random number generator
des1 <- oa.design(nfactors = 9, nlevels = 2)
des1 # resultant design with twelve rows, nine columns, and level values of 1 and 2

# set item names, in which the order of elements corresponds to
#  the order of columns in des1
items1 <- LETTERS[1:9] # item names are "A", "B", ..., "I"

# create questions for BWS
bws.questionnaire(
 choice.sets = des1,
 design.type = 1, # OMED
 item.names = items1)

# set a respondent data set in a row number format
res1 <- data.frame(
 ID  = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10), # id variable
 B1  = c(1, 1, 1, 1, 3, 5, 1, 1, 1, 1),  # best item in question 1
 W1  = c(5, 5, 3, 5, 4, 1, 4, 2, 4, 3),  # worst item in question 1
 B2  = c(1, 3, 1, 4, 1, 2, 3, 2, 1, 2),  # best item in question 2
 W2  = c(5, 5, 5, 5, 3, 5, 1, 4, 4, 5),  # worst item in question 2
 B3  = c(1, 2, 1, 2, 4, 1, 1, 3, 1, 3),  # best item in question 3
 W3  = c(4, 3, 3, 3, 3, 4, 4, 2, 3, 4),  # worst item in question 3
 B4  = c(2, 1, 3, 5, 2, 3, 1, 1, 2, 5),  # best item in question 4
 W4  = c(4, 4, 5, 3, 5, 5, 3, 5, 4, 3),  # worst item in question 4
 B5  = c(2, 3, 3, 2, 2, 2, 2, 1, 3, 2),  # best item in question 5
 W5  = c(4, 4, 4, 4, 3, 4, 4, 4, 4, 3),  # worst item in question 5
 B6  = c(2, 1, 1, 3, 3, 3, 1, 1, 1, 1),  # best item in question 6
 W6  = c(1, 2, 3, 2, 1, 2, 3, 2, 2, 3),  # worst item in question 6
 B7  = c(3, 3, 1, 1, 3, 6, 1, 2, 1, 7),  # best item in question 7
 W7  = c(9, 6, 8, 4, 8, 2, 6, 5, 4, 6),  # worst item in question 7
 B8  = c(2, 1, 2, 2, 2, 1, 1, 3, 1, 1),  # best item in question 8
 W8  = c(3, 3, 3, 3, 4, 4, 4, 4, 3, 4),  # worst item in question 8
 B9  = c(2, 1, 3, 1, 4, 2, 3, 4, 1, 1),  # best item in question 9
 W9  = c(3, 2, 2, 3, 3, 3, 2, 2, 3, 4),  # worst item in question 9
 B10 = c(1, 1, 1, 1, 1, 1, 1, 4, 3, 3),  # best item in question 10
 W10 = c(4, 2, 2, 4, 4, 3, 4, 2, 4, 4),  # worst item in question 10
 B11 = c(2, 1, 3, 3, 3, 2, 1, 2, 2, 4),  # best item in question 11
 W11 = c(1, 4, 4, 1, 1, 4, 4, 4, 1, 1),  # worst item in question 11
 B12 = c(2, 2, 1, 1, 1, 1, 3, 2, 1, 2),  # best item in question 12
 W12 = c(3, 3, 2, 3, 3, 2, 2, 3, 3, 3))  # worst item in question 12

# create a data set for the maxdiff model analysis
#  by combining the choice sets and respondent data set
dat1 <- bws.dataset(
 respondent.dataset = res1,
 response.type = 1, # row number format
 choice.sets = des1,
 design.type = 1)   # OMED
```

```
# analyze responses to BWS questions
# counting approach
bws1 <- bws.count(dat1)
# modeling approach
# note: ITEM5 is excluded from fr1 to normalize its coefficient to zero
fr1 <- RES ~ ITEM1 + ITEM2 + ITEM3 + ITEM4 + ITEM6 + ITEM7 +
             ITEM8 + ITEM9 + strata(STR)
clg1 <- clogit(formula = fr1, data = dat1)
clg1


## example 2: BWS using a balanced incomplete block design
## suppose that ten respondents answered seven BWS questions valuing seven items

# create a BIBD with seven items, four items per question, and seven questions
set.seed(123) # set seed for random number generator
des2 <- find.BIB(trt = 7, k = 4, b = 7)
isGYD(des2)  # check whether the design is a BIBD
des2 # resultant design with seven rows, four columns, and level values ranging from 1 to 7

# set item names, in which the order of element corresponds to
#  the order of level values in des2
items2 <- LETTERS[1:7]

# create questions for BWS
bws.questionnaire(
 choice.sets = des2,
 design.type = 2, # BIBD
 item.names = items2)

# set a respondent data set in a row number format
res2 <- data.frame(
 ID = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10), # id variable
 B1 = c(2, 1, 2, 4, 2, 2, 2, 1, 2, 1),  # best item in question 1
 W1 = c(3, 4, 3, 3, 1, 3, 3, 2, 3, 3),  # worst item in question 1
 B2 = c(4, 3, 3, 3, 3, 1, 1, 2, 1, 1),  # best item in question 2
 W2 = c(3, 2, 4, 1, 2, 3, 3, 4, 2, 4),  # worst item in question 2
 B3 = c(3, 1, 1, 1, 1, 1, 1, 1, 2, 1),  # best item in question 3
 W3 = c(1, 4, 2, 2, 4, 4, 2, 3, 3, 3),  # worst item in question 3
 B4 = c(2, 2, 1, 3, 2, 2, 2, 2, 4, 1),  # best item in question 4
 W4 = c(4, 4, 3, 4, 1, 3, 4, 1, 2, 4),  # worst item in question 4
 B5 = c(1, 3, 2, 1, 3, 2, 1, 1, 1, 1),  # best item in question 5
 W5 = c(3, 1, 4, 4, 1, 4, 3, 2, 4, 3),  # worst item in question 5
 B6 = c(2, 1, 1, 3, 2, 4, 4, 3, 3, 3),  # best item in question 6
 W6 = c(3, 2, 3, 4, 3, 2, 2, 4, 4, 2),  # worst item in question 6
 B7 = c(2, 1, 3, 1, 3, 2, 3, 3, 2, 2),  # best item in question 7
 W7 = c(4, 4, 4, 4, 4, 1, 4, 1, 4, 4))  # worst item in question 7

# create a data set for the maxdiff model analysis
#  by combining the choice sets and respondent data set
dat2 <- bws.dataset(
 respondent.dataset = res2,
 response.type = 1,   # row number format
```

```
 choice.sets = des2,
 design.type = 2,     # BIBD
 item.names = items2) # state variables are labeled using item names

# analyze responses to BWS questions
# counting approach
bws2 <- bws.count(dat2)
bws2
# the argument cl is set to 2 to generaet a data set
#  of the S3 class 'bws.count2'
bws2.2 <- bws.count(dat2, cl = 2)
plot(bws2.2, score = "bw")
barplot(bws2.2, score = "bw")
sum(bws2.2)
summary(bws2.2)
# modeling approach
# note: D is excluded from fr2 to normalized its coefficient to zero
fr2 <- RES ~ A + B + C + E + F + G + strata(STR)
clg2 <- clogit(fr2, data = dat2)
clg2
bws.sp(clg2, base = "D", order = TRUE)

## End(Not run)
```

`bws.questionnaire` *Converting a two-level OMED/BIBD into BWS questions*

Description

This function converts a two-level orthogonal main-effect design (OMED)/balanced incomplete block design (BIBD) into a series of BWS questions.

Usage

```
bws.questionnaire(choice.sets, design.type, item.names)
```

Arguments

`choice.sets`	A data frame or matrix containing choice sets.
`design.type`	A value describing how to design the choice sets: `1` if the design assigned to `choice.sets` is a two-level OMED, and `2` if it is a BIBD.
`item.names`	A vector containing the names of items shown in the questions.

Details

A two-level OMED/BIBD is assigned to `choice.sets`, and may be generated by R functions (e.g., the `oa.design` function in **DoE.base**; the `find.BIB` function in **crossdes**) or copied manually from text books/web sites related to the design of experiments.

When the design is a two-level OMED, each row corresponds to a question and each column corresponds to an item. The level values in the design have to be 1 and 2. The former corresponds to an item being "absent" from a column and the latter corresponds to the item being "present." The correspondence between item names and columns is defined and assigned to the argument `item.names`: the order of names in the vector assigned to `item.names` corresponds to the order of columns (from left to right) in the choice sets assigned to `choice.sets`.

When the design is a BIBD, each row corresponds to a question and the number of columns is equal to the number of items per question. The level values in the design have to be serial integer values, starting from 1: each value corresponds to an item. The correspondence between item names and level values is defined and assigned to the argument `item.names`: the order of names in `item.names` corresponds to the order of level values in the design (i.e., the j th item in `item.names` corresponds to the level value of j in the design).

Value

BWS questions converted from the design are returned.

Author(s)

Hideo Aizaki

See Also

`bws.dataset`, `oa.design`, `find.BIB`

Examples

```
## See examples in bws.dataset()
```

`bws.response` *Generating artificial responses to Case 1 best-worst scaling questions*

Description

This function synthesizes responses to Case 1 best-worst scaling (BWS) questions on the basis of a maximum difference model.

Usage

```
bws.response(design, item.names = NULL, b, n, detail = FALSE,
             seed = NULL)
```

Arguments

design	A matrix or data frame containing a balanced incomplete block design (BIBD).
item.names	A character vector containing the names of items: if NULL (default), default names (i.e., ITEM1, ITEM2, ...) are used in the resultant dataset. The argument is valid for the dataset in a detailed format.
b	A vector containing parameters of independent variables in the model. The vector is used to calculate utilities for alternatives.
n	An integer value showing the number of respondents in the resultant dataset.
detail	A logical variable: if TRUE, the dataset is returned in a detailed format; and if FALSE (default), the dataset is returned in a simple format.
seed	Seed for a random number generator.

Details

This function synthesizes responses to Case 1 BWS questions on the basis of a maximum difference model. The model assumes that there are m items to be evaluated, and that k ($< m$) items exist in a choice set (a question). The number of possible pairs where item i is selected as the best and item j is selected as the worst ($i \neq j$) from k items is given by $k \times (k-1)$. The model also assumes that the respondents select item i as the best and item j as the worst because the difference in utility between items i and j is the highest among all of the $k \times (k-1)$ differences in utility. The systematic component of the utility is assumed to be a linear additive function of the item variables. If the error component of the utility is assumed to be an independently, identically distributed type I extreme value, the probability of selecting item i as the best and item j as the worst is expressed as a conditional logit model.

Given the parameter values assigned to the argument b, and the choice sets assigned to the argument design, the function bws.response calculates the utility for the items. The parameter values assigned to the argument b are set as a numerical vector where the elements correspond to the parameters of item variables. For example, assume that seven items A, B, C, D, E, F, and G exist in the list for evaluation, and their corresponding dummy-coded item variables D_A, D_B, D_C, D_D, D_E, and D_G (item F is assumed to be the reference item) have parameter values of 0.5, 1.2, 1.6, 1.8, 2.1, and 0.9, respectively. A vector assigned to the argument b is c(0.5, 1.2, 1.6, 1.8, 2.1, 0, 0.9), where the sixth element (i.e., item F) corresponds to the reference level, and thus has a value of 0. After calculating the utility values (by adding the calculated values of the systematic component of the utility and random numbers generated from a type I extreme value distribution), the function bws.response finds the pair with the highest difference in utility from the $k \times (k-1)$ differences in utility.

Value

The function bws.response returns a data frame that contains synthesized responses to Case 1 BWS questions, in either a detailed or a simple format. The detailed format dataset contains the following variables, as well as independent variables according to the argument item.names.

id	An identification number of artificial respondents.
Q	A serial number of questions.
PAIR	A serial number of possible pairs of the best and worst items for each question.

BEST	An item number treated as the best in the possible pairs of the best and worst items for each question.
WORST	An item number treated as the worst in the possible pairs of the best and worst items for each question.
RES	Responses to BWS questions, taking the value of 1 if a possible pair of the best and worst items is selected by the synthesized respondents and 0 otherwise.
STR	A stratification variable used to identify each combination of respondent and question.

The simple format dataset contains the following variables.

id	An identification number of artificial respondents.
Bi	A variable describing the row number of the item that is selected as the best in the i-th BWS question (see the help for bws.dataset for a row number format). The serial number of questions is appended to the tail of the variable name (e.g., B1 for the first question, B2 for the second question, and B3 for the third question).
Wi	A variable describing the row number of the item that is selected as the worst in the i-th BWS question (see the help for bws.dataset for a row number format). The serial number of questions is appended to the tail of the variable name (e.g., W1 for the first question, W2 for the second question, and W3 for the third question).

The detailed format dataset includes a dependent variable and independent variables for the analysis, and thus is available for discrete choice analysis functions such as the function clogit in the **survival** package. On the other hand, the simple format dataset only contains variables that correspond to responses to BWS questions, as well as id variable. It must be converted using the function bws.dataset in the package for the analysis. For details, see the Example section.

See Also

support.BWS-package, bws.dataset

Examples

```
# The following lines of code synthesize responses to Case 1 BWS questions,
# return them in detailed and simple format, and then fit the models using
# the function clogit in the survival package. The questions evaluate seven
# items. The choice sets consist of seven questions with four items each.
# The function find.BIB in the crossdes package creates the corresponding
# BIBD with seven treatments, seven blocks, and size four. The systematic
# component of the utility for items is the same as that explained in the
# Details section.

## Not run:
# Load packages
library(survival)
library(crossdes)

# Generate BIBD
```

```
set.seed(123)
bibd <- find.BIB(trt = 7, b = 7, k = 4)
isGYD(bibd)
bibd

# Synthesize responses to Case 1 BWS questions
b <- c(0.5, 1.2, 1.6, 1.8, 2.1, 0, 0.9)
items = c("A", "B", "C", "D", "E", "F", "G")
dat.detail <- bws.response(
  design = bibd, item.names = items,
  b = b, n = 100,
  detail = TRUE, seed = 123)
str(dat.detail)
dat.simple <- bws.response(
  design = bibd,
  b = b, n = 100,
  detail = FALSE, seed = 123)
str(dat.simple)

# Convert dat.simple into dataset for the analysis
response.vars <- colnames(dat.simple)[-1]
dat.simple.md <- bws.dataset(
  respondent.dataset = dat.simple,
  response.type = 1,
  choice.sets = bibd,
  design.type = 2,
  item.names = items,
  id = "id",
  response = response.vars,
  model = "maxdiff")

# Fit conditional logit models
mf <- RES ~ A + B + C + D + E + G + strata(STR)
out.detail <- clogit(mf, dat.detail)
out.simple <- clogit(mf, dat.simple.md)
out.simple
all.equal(coef(out.detail), coef(out.simple))

## End(Not run)
```

---

bws.sp *Calculating shares of preference*

Description

This function calculates shares of preference for each item based on the conditional logit model choice rule.

Usage

```
bws.sp(object, base, coef = NULL, order = FALSE, decreasing = FALSE, ...)
```

Arguments

`object`	An output from the function `clogit` in **survival** or vector/matrix/data frame containing estimates.
`base`	A character showing the base item.
`coef`	A vector containing the names of item variables used in the model.
`order`	A logical value denoted by `TRUE` when the resultant shares are sorted or `FALSE` when not doing so.
`decreasing`	A logical value denoted by `TRUE` when the sort order is decreasing, or `FALSE` when it is increasing.
`...`	Arguments passed to a function used internally.

Details

This function calculates a share of preference for item i. For details on the shares of preference, refer to `support.BWS-package`. Althoug this function is developed for the function `clogit` in the **survival**, it may be available for other functions regarding discrete choice models. This function assumes a simple model without covariates.

Value

A vector or data frame containing the calculated shares of preference for each item is returned.

Author(s)

Hideo Aizaki

See Also

`support.BWS-package`, `bws.dataset`, `clogit`

Examples

```
## See examples in bws.dataset()
```

`fruit` *Synthetic respondent data set: consumers' preferences for fruits*

Description

Data set artificially created for an example based on a BIBD. This example illustrates consumers' preferences for seven fruits: apple, orange, grapes, banana, peach, melon, and pear.

Usage

```
data(fruit)
```

Format

A data frame with 100 respondents on the following 15 variables.

`ID` Identification number of respondents.

`B1` Item selected as the best in question 1.

`W1` Item selected as the worst in question 1.

`B2` Item selected as the best in question 2.

`W2` Item selected as the worst in question 2.

`B3` Item selected as the best in question 3.

`W3` Item selected as the worst in question 3.

`B4` Item selected as the best in question 4.

`W4` Item selected as the worst in question 4.

`B5` Item selected as the best in question 5.

`W5` Item selected as the worst in question 5.

`B6` Item selected as the best in question 6.

`W6` Item selected as the worst in question 6.

`B7` Item selected as the best in question 7.

`W7` Item selected as the worst in question 7.

Author(s)

Hideo Aizaki

See Also

`bws.dataset`, `find.BIB`

Examples

```
# The following BIBD is generated using find.BIB()
# in the crossdes package:
#  set.seed(123)
#  find.BIB(trt = 7, k = 4, b = 7)
sets.fruit <- cbind(
  c(1,2,2,1,1,3,1),
  c(4,3,4,2,3,5,2),
  c(6,4,5,5,4,6,3),
  c(7,6,7,6,5,7,7))
items.fruit <- c(
  "Apple",
  "Orange",
  "Grapes",
  "Banana",
  "Peach",
  "Melon",
  "Pear")
```

```
bws.questionnaire(
  choice.sets = sets.fruit,
  design.type = 2,
  item.names = items.fruit)
data(fruit)
data.fruit <- bws.dataset(
  respondent.dataset = fruit,
  response.type = 1,
  choice.sets = sets.fruit,
  design.type = 2,
  item.names = items.fruit)
count.fruit <- bws.count(data = data.fruit)
count.fruit
```

`mfa` *Synthetic respondent data set: citizens' preferences for the multifunctionality of agriculture*

Description

Data set artificially created for an example based on a two-level OMED. This example illustrates citizens' preferences for the multifunctionality of agriculture: landscape, biodiversity, water use, land conservation, flood control, rural viability, food security, animal welfare, and cultural heritage.

Usage

```
data(mfa)
```

Format

A data frame with 100 respondents on the following 25 variables.

`ID` Identification number of respondents.

`B1` Item selected as the best in question 1.

`W1` Item selected as the worst in question 1.

`B2` Item selected as the best in question 2.

`W2` Item selected as the worst in question 2.

`B3` Item selected as the best in question 3.

`W3` Item selected as the worst in question 3.

`B4` Item selected as the best in question 4.

`W4` Item selected as the worst in question 4.

`B5` Item selected as the best in question 5.

`W5` Item selected as the worst in question 5.

`B6` Item selected as the best in question 6.

`W6` Item selected as the worst in question 6.

B7 Item selected as the best in question 7.

W7 Item selected as the worst in question 7.

B8 Item selected as the best in question 8.

W8 Item selected as the worst in question 8.

B9 Item selected as the best in question 9.

W9 Item selected as the worst in question 9.

B10 Item selected as the best in question 10.

W10 Item selected as the worst in question 10.

B11 Item selected as the best in question 11.

W11 Item selected as the worst in question 11.

B12 Item selected as the best in question 12.

W12 Item selected as the worst in question 12.

Author(s)

Hideo Aizaki

See Also

bws.dataset, oa.design

Examples

```
# The following OA is generated using oa.design()
# in the DoE.base package:
#  set.seed(123)
#  oa.design(nfactors = 9, nlevels = 2)
sets.mfa <- cbind(
  c(1,2,1,2,2,1,2,2,1,1,1,2),
  c(2,1,2,1,2,1,2,2,1,1,2,1),
  c(1,2,1,1,2,1,2,1,2,2,2,1),
  c(1,2,2,2,1,2,2,1,1,1,2,1),
  c(2,2,2,1,1,1,2,1,2,1,1,2),
  c(1,1,2,2,1,1,2,2,2,2,1,1),
  c(2,1,1,2,2,2,2,1,2,1,1,1),
  c(2,1,1,2,1,1,2,1,1,2,2,2),
  c(2,2,1,1,1,2,2,2,1,2,1,1))
items.mfa <- c(
  "Landscape",
  "Biodiversity",
  "Water use",
  "Land conservation",
  "Flood control",
  "Rural viability",
  "Food security",
  "Animal welfare",
  "Cultural heritage")
bws.questionnaire(
```

```
  choice.sets = sets.mfa,
  design.type = 1,
  item.names = items.mfa)
data(mfa)
data.mfa <- bws.dataset(
  respondent.dataset = mfa,
  response.type = 1,
  choice.sets = sets.mfa,
  design.type = 1,
  item.names = items.mfa)
count.mfa <- bws.count(data = data.mfa)
count.mfa
```

`ricebws1` *Consumers' preferences for rice characteristics*

Description

This dataset contains responses to Case 1 BWS questions about consumers' preferences for rice characteristics.

Usage

```
data(ricebws1)
```

Format

A data frame with 90 respondents on the following 18 variables.

`id` Identification number of respondents.

`b1` Item selected as the best in BWS question 1.

`w1` Item selected as the worst in BWS question 1.

`b2` Item selected as the best in BWS question 2.

`w2` Item selected as the worst in BWS question 2.

`b3` Item selected as the best in BWS question 3.

`w3` Item selected as the worst in BWS question 3.

`b4` Item selected as the best in BWS question 4.

`w4` Item selected as the worst in BWS question 4.

`b5` Item selected as the best in BWS question 5.

`w5` Item selected as the worst in BWS question 5.

`b6` Item selected as the best in BWS question 6.

`w6` Item selected as the worst in BWS question 6.

`b7` Item selected as the best in BWS question 7.

`w7` Item selected as the worst in BWS question 7.

age Respondents' age: 1 = <40; 2 = 40-<60; 3 = >=60

hp Highest price of rice per 5 kg that respondents have purchased for the last six months: 1 = < 1600 JPY; 2 = 1600-<2100; 3 = >=2100

chem Respondents' valuation of rice grown with low-chemicals: 1 if respondents value low-chemical rice and 0 otherwise

Author(s)

Hideo Aizaki

See Also

bws.dataset, find.BIB

Examples

```
# Respondents were asked to select their most and least important
# characteristics of rice when purchasing rice. Rice characteristics
# were assumed to be place of origin, variety, price, taste, safety,
# wash-free rice, and milling date. BWS questions were created from
# a balanced incomplete block design (BIBD) with seven treatments
# (items), four columns (four items per question), and seven rows
# (seven questions).

# Generate the BIBD using find.BIB() in the crossdes package:
require("crossdes")
set.seed(8041)
bibd.ricebws1 <- find.BIB(trt = 7, b = 7, k = 4)
isGYD(bibd.ricebws1)
bibd.ricebws1

# Store rice characteristics used in the survey to items.ricebws1:
items.ricebws1 <- c(
  "Place_of_origin",
  "Variety",
  "Price",
  "Taste",
  "Safety",
  "Washfree_rice",
  "Milling_date")

# Convert the BIBD into the BWS questions:
bws.questionnaire(bibd.ricebws1, design.type = 2,
  item.names = items.ricebws1)

# Load the dataset ricebws1 containing the responses to
# the BWS questions:
data("ricebws1", package = "support.BWS")
dim(ricebws1)
names(ricebws1)

# Create the dataset for the analysis:
```

```
data.ricebws1 <- bws.dataset(
  respondent.dataset = ricebws1,
  response.type = 1,
  choice.sets = bibd.ricebws1,
  design.type = 2,
  item.names = items.ricebws1)

# Calculate BW scores:
count.ricebws1 <- bws.count(data = data.ricebws1)
count.ricebws1
```

Index