# Package ‘support.BWS’

April 27, 2015

**Type** Package

**Title** Basic Functions for Supporting an Implementation of Best-Worst Scaling

**Version** 0.1-3

**Date** 2015-04-27

**Author** Hideo Aizaki

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

**Description** 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.

**License** GPL (>= 2)

**Suggests** DoE.base, crossdes, survival

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2015-04-27 11:04:02

## R topics documented:

support.BWS-package *Basic functions for supporting an implementation of 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 (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 $m$ columns, where $m$ 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 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. 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:

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

### Author(s)

Hideo Aizaki

### References

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.

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(1), 12–25.

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

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.count` *Calculating count-based BW scores*

---

### Description

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

### Usage

```
bws.count(data)

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

### Arguments

| | |
|---|---|
| `data` | A data frame containing the output from `bws.dataset()`. |
| `x` | An object of the S3 class “bws.count”. |
| `digits` | The number of significant digits. See the `format()` function. |
| `scientific` | Scores are encoded in scientific format. See the `format()` function. |
| `...` | Arguments passed to the `format()` function. |

### Details

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 ($W_{in}$) across 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 $sqrt.BW_i$.

### Value

The output from `bws.count()` 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. |

## Author(s)

Hideo Aizaki

## References

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

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(1), 12–25.

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

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.

## See Also

`bws.dataset`

## Examples

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

bws.dataset *Creating a data set suitable for BWS analysis on the basis of counting and modeling approaches*

### Description

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

### Usage

```
bws.dataset(respondent.dataset, response.type = 1,
            choice.sets, design.type = 1,
            item.names = NULL, row.renames = TRUE)
```

### Arguments

| | |
|---|---|
| `respondent.dataset` | A data frame containing a 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. |

### 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.

## Value

The function returns a data set containing 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. |
| 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 clogit() 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 clogit(). |

## Author(s)

Hideo Aizaki

## See Also

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

## Examples

```
## 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

## case 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 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


## case 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 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
# 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.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., `oa.design ()` in **DoE.base**; `find.BIB()` 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

## 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
```

# Index