Package ‘support.BWS2’

March 22, 2019

Type Package

Title Basic Functions for Supporting an Implementation of Case 2 Best-Worst Scaling

Version 0.2-2

Date 2019-03-22

Author Hideo Aizaki

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

Description
Provides three basic functions that support an implementation of Case 2 (profile case) best-worst scaling. The first is to convert an orthogonal main-effect design into questions, the second is to create a dataset suitable for analysis, and the third is to calculate count-based scores.

License GPL (>= 2)

Suggests DoE.base, survival

NeedsCompilation no

Repository CRAN

Date/Publication 2019-03-22 06:00:03 UTC

R topics documented:

`support.BWS2-package` *Basic functions for supporting an implementation of Case 2 best–worst scaling*

Description

The package has three basic functions that support an implementation of Case 2 (profile case) best–worst scaling. The first is to convert an orthogonal main-effect design into questions, the second is to create a dataset suitable for analysis, and the third is to calculate count-based scores.

Details

The package is under development and thus may be changed substantially in the future.

1) Outline of Case 2 best–worst scaling

Case 2 (profile case) best–worst scaling (BWS) is a question-based survey method to elicit preferences for attribute levels (See Flynn 2010, Flynn et al. 2007 and 2008, Louviere et al. 2015, and Marley et al. 2008 for details of the subsection). A profile (choice set) has three or more attributes and each attribute has two or more levels. The profile is expressed as a combination of attribute levels. Numerous profiles are constructed using experimental designs. Attributes shown in each profile are fixed in all the profiles and a combination of attribute levels in each profile is changed according to the profiles. A profile selected from all the constructed profiles is presented to respondents, who are then asked to choose the best and worst attribute levels in the profile. This question is repeated until all profiles are evaluated. Analyzing the responses enables us to elicit preferences for the attribute levels.

A basic approach to constructing profiles is using an orthogonal main-effect design (OMED). Assume that the profiles have K attributes and each attribute has L_k levels. If all the attributes have the same number of levels, L, a L^K OMED is used to construct the profiles. Columns of the OMED correspond to attributes, while the rows to profiles. For example, profiles have four attributes and they have three levels: attribute A with levels A1, A2, and A3; attribute B with levels B1, B2, and B3; attribute C with levels C1, C2, and C3; and attribute D with levels D1, D2, and D3. A 3^4 OMED corresponding to the assumptions is as follows (see the section Examples of the function `bws2.dataset()` for code to generate the OMED):

1	3	2	3
3	1	2	2
3	3	3	1
2	3	1	2
2	2	2	1
1	1	1	1
1	2	3	2
3	2	1	3
2	1	3	3

Suppose that attributes A, B, C, and D are assigned to the first, second, third, and fourth column of the OMED, respectively, and the values 1, 2, and 3 used in the OMED correspond to the attribute-level values in each attribute: 1 = A1, 2 = A2, and 3 = A3 for attribute A; 1 = B1, 2 = B2, and 3 = B3 for attribute B; 1 = C1, 2 = C2, and 3 = C3 for attribute C; and 1 = D1, 2 = D2, and 3 = D3 for attribute D. Accordingly, the above-mentioned OMED can be transformed into the following:

A1	B3	C2	D3
A3	B1	C2	D2
A3	B3	C3	D1
A2	B3	C1	D2
A2	B2	C2	D1
A1	B1	C1	D1
A1	B2	C3	D2
A3	B2	C1	D3
A2	B1	C3	D3

The resultant OMED consists of nine rows: nine profiles, that is, nine Case 2 BWS questions, are constructed. For example, a profile corresponding to the first row of the OMED comprises A1, B3, C2, and D3. This means that respondents who face the question created from the first row of the OMED are asked to select their best and worst attribute levels from attribute levels A1, B3, C2, and D3, as follows:

Please select your best and worst attribute levels from the following four:

Best	Attribute	Worst
[_]	A1	[_]
[_]	B3	[_]
[_]	C2	[_]
[_]	D3	[_]

There are two approaches for analyzing responses to Case 2 BWS questions: a counting approach and modeling approach. The counting approach calculates scores on the basis of the number of times attribute level i is selected as the best (B_{in}: B score for attribute level i) and the worst (W_{in}: W score for attribute i) among all the questions for respondent n. A (disaggregated) best-minus-worst (BW) score and its standardized variant are defined as

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

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

where f_i is the frequency with which attribute level i appears across all questions.

The modeling approach uses discrete choice models to analyze responses. When using the modeling approach, a model type must be selected according to the assumption for respondents' choice behavior in Case 2 BWS questions and then a dataset must be formatted as per the selected model. There are three standard models: paired, marginal, and marginal sequential models. Although the three models commonly assume that the respondents derive utility for each attribute level shown in the profile, the assumption for how they select the best and worst attribute levels from the set of attribute levels in the profile differs among the three models.

The number of possible pairs in which attribute level i is selected as the best and attribute level j is selected as the worst ($i \neq j$) from K attribute levels is $K \times (K - 1)$. The paired model assumes that respondents select attribute level i as the best and attribute level j as the worst because the difference in utility between i and j represents the greatest utility difference among $K \times (K - 1)$ utility differences. Consider the example profile mentioned above. It contains four attribute levels: A1, B3, C2, and D3. The number of possible pairs is 12 $(= 4 \times (4 - 1))$. There are 12 possible pairs of the best and worst attribute levels (in each pair, the former is the best and the latter is the worst): (A1, B3), (A1, C2), (A1, D3), (B3, A1), (B3, C2), (B3, D3), (C2, A1), (C2, B3), (C2, D3), (D3, A1), (D3, B3), and (D3, C2). If a respondent selects A1 as the best attribute level and C2 as the worst, the paired model assumes that the respondent calculates 12 utility differences as per the 12 above-mentioned pairs and that the difference in utility between A1 and C2 is the maximum among the 12 utility differences.

The marginal model assumes that there are K possible best attribute levels and K possible worst attribute levels in a profile, that attribute level i is selected as the best from K possible best attribute levels in the profile, and that attribute level j is selected as the worst from K possible worst attribute levels. This is because the utility for attribute level i is the maximum among the utilities for K attribute levels and that for attribute level j is the minimum. Following the above example, the marginal model assumes that there are four possible best attribute levels and four possible worst attribute levels in the profile and interprets the respondent's choice behavior as follows: utility for A1 is the maximum among the four utilities for A1, B3, C2, and D3 and that for C2 is the minimum among the four.

The assumption of the marginal model that the worst attribute level is selected from K attribute levels would not be appropriate because the best attribute level in a profile must differ from the worst one in the profile. Thus, the marginal sequential model assumes that respondents select attribute level i as the best from K attribute levels in the profile and then attribute level j as the worst from the remaining $K - 1$ attribute levels. Following the above example, under the marginal sequential model assumption, there are four possible best attribute levels and three possible worst attribute levels in the profile. The model considers that the respondent selects A1 as the best from the four possible attribute levels because the utility for A1 is the highest among the utilities for A1, B3, C2, and D3, but selects C2 as the worst from three possible worst levels, B3, C2, and D3, because the utility for C2 is the least among the three.

The three models generally assume that the utility for attribute level i selected as the worst is the negative of the one selected as the best. Under these assumptions, and given the assumption for the stochastic component of utility, the probability of selecting attribute level i as the best and attribute level j as the worst can be expressed as a conditional logit model.

2) Role of the package and other packages needed to complete implementing Case 2 BWS

The package **support.BWS2** provides functions to convert an OMED into a series of Case 2 BWS questions, create a dataset for the analysis from the OMED and the responses to the questions, and calculate BWS scores. Other packages are needed to complete implementing Case 2 BWS with R: a package to construct OMEDs and another to analyze the responses on the basis of the modeling approach. For example, the `oa.design()` function in **DoE.base** (Groemping 2015) can construct OMEDs, while the functions `clogit()` in **survival** (Therneau 2016), `mlogit()` in **mlogit** (Croissant 2013), and `gmnl()` in **gmnl** (Sarrias and Daziano 2015) can fit the conditional logit model. The latter two functions are also used to fit advanced discrete choice models such as a mixed (random parameters) logit model. Refer to the task views about experimental designs (Groemping

2016) and econometrics (Zeileis 2017) on CRAN for details on packages for experimental designs and discrete choice models in R.

Acknowledgments

I would like to thank Professor Kazuo Sato for his kind support. This work was supported by JSPS KAKENHI Grant Numbers 25450341 and 16K07886.

Author(s)

Hideo Aizaki

References

Flynn TN (2010) Valuing citizen and patient preferences in health: recent developments in three types of best-worst scaling. *Expert Review of Pharmacoeconomics & Outcomes Research*, **10**(3), 259–267.

Flynn TN, Louviere JJ, Peters TJ, Coast J (2007) Best-Worst Scaling: What it can do for health care research and how to do it. *Journal of Health Economics*, **26**, 171–189.

Flynn TN, Louviere JJ, Peters TJ, Coast J (2008) Estimating preferences for a dermatology consultation using best-worst scaling: Comparison of various methods of analysis. *BMC Medical Research Methodology*, **8**(76). doi: 10.1186/1471-2288-8-76.

Croissant Y (2013) *mlogit: multinomial logit model*. R package version 0.2-4. `https://CRAN.R-project.org/package=mlogit`.

Groemping U (2015) *DoE.base: Full Factorials, Orthogonal Arrays and Base Utilities for DoE Packages*. R package version 0.27-1. `https://CRAN.R-project.org/package=DoE.base`.

Groemping U (2016) *CRAN Task View: Design of Experiments (DoE) & Analysis of Experimental Data*. `https://CRAN.R-project.org/view=ExperimentalDesign`.

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

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

Marley AAJ, Flynn TN, Louviere JJ (2008) Probabilistic models of set-dependent and attribute-level best-worst choice. *Journal of Mathematical Psychology*, **52**, 281–296.

Sarrias M, and Daziano R (2015) *gmnl: Multinomial Logit Models with Random Parameters*. R package version 1.1-1. `https://CRAN.R-project.org/package=gmnl`.

Therneau T (2015) *A Package for Survival Analysis in S*. Version 2.38, `https://CRAN.R-project.org/package=survival`.

Zeileis A (2017) *CRAN Task View: Econometrics*. `https://CRAN.R-project.org/view=Econometrics`.

`agritourism` *Potential tourists' valuation of agritourism*

Description

This dataset contains responses to Case 2 BWS questions. Respondents were asked to evaluate agritourism packages provided by dairy farms in Hokkaido, Japan.

Usage

```
data(agritourism)
```

Format

A data frame with 240 respondents on the following 21 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.

`gender` Respondents' gender: 1 = male; 2 = female.

`age` Respondents' age: 2 = 20s; 3 = 30s; 4 = 40s; 5 = 50s

See the section Examples for details.

Author(s)

Hideo Aizaki

See Also

support.BWS2-package, bws2.dataset, oa.design

Examples

```
## Not run:
# Agritourism refers to various activities offered by farms and ranches
# to visitors, such as hands-on farm work or outdoor recreation.
#
# In the Case 2 BWS questions, respondents were asked to evaluate
# agritourism packages provided by dairy farms (ranches) in Hokkaido, Japan.
# We assumed that the agritourism package consists of the following four
# types of activities, each with three activity items:
#  1. Hands-on ranch chores
#    (1) Milking a cow
#    (2) Feeding a cow
#    (3) Nursing a calf
#  2. Hands-on food processing
#    (1) Butter making
#    (2) Ice-cream making
#    (3) Creamy caramel making
#  3. Hands-on craft making
#    (1) Making a product from wool
#    (2) Making a product from wood
#    (3) Making a product from pressed flowers
#  4. Outdoor activities
#    (1) Horse riding
#    (2) Tractor riding
#    (3) Walking with cows
#
# As there are four activities and each activity has three items,
# a total of nine BWS questions were created using a three-level OMED
# with four columns. Each BWS question asked respondents to select
# the most and least interesting of the four activities shown
# in the question.
#
# In the following, we assume that the paired and marginal models with
# both attribute and attribute-level variables (Flynn et al. 2007; 2008)
# are fitted to the responses using the conditional logit model,
# with clogit() in the survival package.

# Load the package needed for the example:
require(survival)

options(digits = 4)

# The following OMED is generated using oa.design() in the DoE.base package:
# require(DoE.base)
```

```
# des <- data.matrix(
#    oa.design(nl = c(3,3,3,3), randomize = FALSE))
des <- cbind(
  c(1, 1, 1, 2, 2, 2, 3, 3, 3),
  c(1, 2, 3, 1, 2, 3, 1, 2, 3),
  c(1, 3, 2, 3, 2, 1, 2, 1, 3),
  c(1, 2, 3, 3, 1, 2, 2, 3, 1))

# The names of the attributes (activities) and attribute levels
# (activity items) were stored in the list attr.lev:
attr.lev <- list(
  chore = c("milking", "feeding", "nursing"),
  food = c("butter", "ice", "caramel"),
  craft = c("wool", "wood", "flower"),
  outdoor = c("horse", "tractor", "cow"))

# A series of Case 2 BWS questions were converted from the OMED using
# bws2.questionnaire():
bws2.questionnaire(choice.sets = des, attribute.levels = attr.lev,
  position = "left")

# The responses to the questions were stored in the dataset agritourism
# in the support.BWS2 package:
data(agritourism)
dim(agritourism)
colnames(agritourism)

# The names of the response variables used in the dataset agritourism
# were stored in the vector response.vars:
response.vars <- colnames(agritourism)[2:19]
response.vars

# The base level in each attribute was stored in the object base.lev
# in list format:
base.lev <- list(
  chore = c("nursing"),
  food = c("caramel"),
  craft = c("flower"),
  outdoor = c("cow"))

# The datasets for the paired model and the marginal model were created
# using bws2.dataset() and then stored in the objects pr.data1 and mr.data1,
# respectively:
pr.data1 <- bws2.dataset(
  data = agritourism,
  id = "id",
  response = response.vars,
  choice.sets = des,
  attribute.levels = attr.lev,
  reverse = TRUE,
  base.level = base.lev,
  model = "paired")
mr.data1 <- bws2.dataset(
```

```
  data = agritourism,
  id = "id",
  response = response.vars,
  choice.sets = des,
  attribute.levels = attr.lev,
  reverse = TRUE,
  base.level = base.lev,
  model = "marginal")
dim(pr.data1)
names(pr.data1)
dim(mr.data1)
names(mr.data1)

# The BWS scores were calculated using bws2.count() with the dataset for
# the marginal model:
scores <- bws2.count(mr.data1)
dim(scores)
names(scores)

# The scores for each level were aggregated among all respondents using
# sum() and bar plots of the scores were drawn using barplot():
sum(scores, "level")
barplot(scores, "bw", "level")

# If we only need aggregated B and W scores, these can be calculated from
# the dataset for a paired model as follows:
apply(pr.data1[pr.data1$RES == 1, c("BEST.LV", "WORST.LV")], 2, table)

# BW scores can be calculated according to groups of respondents.
# For example, the scores for male and those for female are given as follows:
sum(scores[agritourism$gender == 1, ], "level")
sum(scores[agritourism$gender == 2, ], "level")

# Bar plots for respondents in their 20s and those in their 50s can also be
# drawn using the following lines of code:
barplot(scores[agritourism$age == 2, ], "bw", "level")
barplot(scores[agritourism$age == 5, ], "bw", "level")

# We fitted the conditional logit model to the Case 2 BWS responses
# on the basis of the paired and marginal models with both attribute
# and attribute-level variables. The systematic component of the utility
# function for the example is
#    v = b1 chore + b2 food + b3 outdoor +
#        b4 milking + b5 feeding + b6 butter + b7 ice +
#        b8 wool + b9 wood + b10 horse + b11 tractor
# where chore, food, and outdoor are attribute variables (craft has been
# omitted); and milking, feeding, butter, ice, wool, wood, horse, and
# tractor are attribute-level variables (nursing has been omitted for chore,
# caramel has been omitted for food, flower has been omitted for craft,
# and cow has been omitted for outdoor); bs are coefficients to be estimated.
#
# The model formula for clogit(), corresponding to the systematic component
# mentioned above, is described as:
```

```
mf <- RES ~ chore + food + outdoor +
            milking + feeding + butter + ice +
            wool + wood + horse + tractor +
            strata(STR)

# We fitted the paired model using clogit() with the dataset pr.data1:
pr.out <- clogit(formula = mf, data = pr.data1)
pr.out

# The attribute-level variables are effect-coded ones, and thus the
# coefficient of the base level in each attribute can be calculated using:
b <- coef(pr.out)
(nursing <- -sum(b[4:5]))
names(nursing) <- "nursing"
(caramel <- -sum(b[6:7]))
names(caramel) <- "caramel"
(flower <- -sum(b[8:9]))
names(flower) <- "flower"
(cow <- -sum(b[10:11]))
names(cow) <- "cow"
craft <- 0
names(craft) <- "craft"
paired.model <- c(b[1:2], craft, b[3], b[4:5], nursing, b[6:7],
  caramel, b[8:9], flower, b[10:11], cow)
barplot(paired.model)

# The following code is for the marginal model:
mr.out <- clogit(formula = mf, data = mr.data1)
mr.out
b <- coef(mr.out)
(nursing <- -sum(b[4:5]))
names(nursing) <- "nursing"
(caramel <- -sum(b[6:7]))
names(caramel) <- "caramel"
(flower <- -sum(b[8:9]))
names(flower) <- "flower"
(cow <- -sum(b[10:11]))
names(cow) <- "cow"
marginal.model <- c(b[1:2], craft, b[3], b[4:5], nursing, b[6:7],
  caramel, b[8:9], flower, b[10:11], cow)
barplot(marginal.model)

# As mentioned in Flynn et al. (2008), the results from the paired model
# are similar to those from the marginal model: the correlation coefficient
# for the two results is calculated as follows:
cor(marginal.model, paired.model)
plot(marginal.model, paired.model,
  xlim = c(-0.5, 1), ylim = c(-0.5, 1))

## End(Not run)
```

bws2.count *Calculating count-based best–worst scaling scores*

Description

This function calculates best, worst, best-minus-worst, and standardized best-minus-worst scores for each respondent.

Usage

```
bws2.count(data, ...)

## S3 method for class 'bws2.count'
barplot(height, score = c("bw", "b", "w"),
  output = c("level", "attribute"), mfrow = NULL, ...)

## S3 method for class 'bws2.count'
sum(x, output = c("level", "attribute"), ...)
```

Arguments

`data`	A data frame containing the dataset generated from `bws2.dataset()`.
`x, height`	An object of the S3 class 'bws2.count'.
`output`	A character showing a type of BWS score calculated by this function: `"attribute"` is assigned to this argument when BWS scores for attributes are calculated or `"level"` is assigned when BWS scores for attribute levels are calculated.
`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, or `"bw"` is assigned when it is based on best-minus-worst scores.
`mfrow`	A two-element vector `c(nr, nc)`; bar plots will be drawn in an `nr`-by-`nc` array on the device by row.
`...`	Arguments passed to function(s) used internally.

Details

The `bws2.count()` function calculates disaggregated best (B), worst (W), best-minus-worst (BW), and standardized BW scores. For details on these scores, refer to the Details section on the help page of this package.

Output from this function is the object of S3 class 'bws2.count', which inherits from the S3 class 'data.frame'. The generic functions such as `barplot()` and `sum()` are available for the S3 class 'bws2.count'. The `barplot()` function draws the bar plots of B, W, or BW scores for each attribute when `output = "attribute"` or those for each attribute level when `output = "level"`. The `sum` function returns a data frame containing B, W, BW, and standardized BW scores for all respondents for each attribute when `output = "attribute"` or for each attribute level when `output = "level"`.

Value

The output from `bws2.count()`, which is the object of the S3 class 'bws2.count', is a data frame containing six types of variables: respondent's identification variable, B score variables, W score variables, BW score variables, standardized BW score variables, and respondent's characteristic variables. These scores are calculated by each respondent. The names of these score variables are `b.<name of attribute or attribute level>`, `w.<name of attribute or attribute level>`, `bw.<name of attribute or attribute level>`, and `sbw.<name of attribute or attribute level>`. Part `<name of attribute or attribute level>` for each score variable is set according to the argument `attribute.levels` in `bws2.dataset()` used to generate a dataset for `bws2.count()`.

The output has the following attributes:

`nquestions` A vector showing the number of questions.

`nrespondents` A vector showing the number of respondents.

`freq.levels` A variable showing the frequency of each attribute level in the choice sets.

`attribute.levels`
A list of attributes and their levels, which is the same as those assigned to argument `attribute.levels` in `bws2.dataset()` used to generate a dataset assigned to argument `data` of `bws2.count()`.

`vnames` A variable showing the names of each attribute level.

`b.names` A variable showing the names of B score by each attribute level.

`w.names` A variable showing the names of W score by each attribute level.

`bw.names` A variable showing the names of BW score by each attribute level.

`sbw.names` A variable showing the names of standardized BW score by each attribute level.

Author(s)

Hideo Aizaki

See Also

`support.BWS2-package`, `bws2.dataset`

Examples

```
## See examples in bws2.dataset()
```

---

bws2.dataset — *Creating a dataset suitable for Case 2 best–worst scaling analysis using counting and modeling approaches*

Description

This function creates a dataset used for `bws2.count()` in **support.BWS2** and functions for discrete choice models such as `clogit()` in **survival**.

Usage

```
bws2.dataset(data, id, response, choice.sets, attribute.levels,
  base.attribute = NULL, base.level = NULL,
  reverse = TRUE, model = "paired",
  attribute.variables = NULL, effect = NULL, delete.best = FALSE,
  type = c("paired", "marginal", "sequential"),
   ...)
```

Arguments

`data` A data frame containing a respondent dataset.

`id` A character showing the name of the respondent identification number variable used in the respondent dataset.

`response` A vector containing the names of response variables in the respondent dataset, showing the best and worst attribute levels selected in each Case 2 BWS question.

`choice.sets` A data frame or matrix containing an orthogonal main-effect design.

`attribute.levels` A list containing the names of the attributes and their levels.

`base.attribute` A character showing the base attribute: the argument is used when attribute variables are created as effect coded ones and `NULL` is assigned to the argument when attribute variables are created as dummy coded ones.

`base.level` A list containing the base level in each attribute: the argument is used when attribute level variables are created as effect coded ones and `NULL` is assigned to the argument when attribute level variables are created as dummy coded ones.

`reverse` A logical value denoted by `TRUE` when the signs of the attribute variables are reversed for the possible worst, or otherwise `FALSE`.

`model` A character showing a type of dataset created by this function: `"paired"` for a paired model, `"marginal"` for a marginal model, and `"sequential"` for a marginal sequential model.

`attribute.variables` A character showing a type of attribute variables, denoted by `"reverse"` when the attribute variables take the value of `1` for a possible best, `-1` for a possible worst, and `0` otherwise, or `"constant"` when the attribute variables are created as attribute-specific constants. The argument is deprecated. Please use the argument `reverse` instead.

`effect` A list containing the base level in each attribute: the argument is used when attribute level variables are created as effect coded ones and while `NULL` is assigned to the argument when attribute level variables are created as dummy coded ones. The argument is deprecated. Please use the argument `base.level` instead.

`delete.best` A logical value denoted by `TRUE` when deleting an attribute level 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.

type
: A character showing a type of dataset created by this function: "paired" for a paired model, "marginal" for a marginal model, and "sequential" for a marginal sequential model. The argument is deprecated. Please use the argument model instead.

...
: Optional arguments; currently not in use.

Details

The respondent dataset, in which each row corresponds to a respondent, must be organized by users and then assigned to the argument data. The dataset must include the respondent's identification number (id) variable in the first column and the response variables in the subsequent columns, each indicating which attribute levels are selected as the best and worst for each question. Other variables in the respondent dataset are treated as the respondents' characteristics such as gender and age. Respondents' characteristic variables are also stored in the resultant dataset created by the function bws2.dataset(). Although the names of the id and response variables are left to the discretion of the user, those of the id and response variables are assigned to the arguments id and response.

The response variables must be constructed such that the best attribute levels alternate with the worst by question. For example, when there are nine BWS questions, the variables are B1, W1, B2, W2, ..., B9, and W9. Here, Bi and Wi show the attribute levels selected as the best and worst in the i-th question. The row numbers of the attribute levels selected as the best and worst are stored in the response variables. For example, suppose that a respondent was asked to answer the following BWS question, which is the same as that shown on the help page of this package, and then selected A1 (attribute level in the first row) as the best and C2 (attribute level in the third row) as the worst.

Please select your best and worst attribute levels from the following four:

Best	Attribute	Worst
[_]	A1	[_]
[_]	B3	[_]
[_]	C2	[_]
[_]	D3	[_]

The response variables B1 and W1, corresponding to the respondent's answer to this question, take the value of 1 (= the attribute level in the first row) and 3 (= the attribute level in the third row).

The arguments choice.sets and attribute.levels are the same as those in bws2.questionnaire(). The order of questions in the respondent dataset has to be the same as that in choice.sets.

The arguments type, reverse, base.attribute, and base.level are set according to the model you will use: argument type is set as "paired" for the paired model, "marginal" for the marginal model, or "sequential" for the marginal sequential model; the argument reverse is set as "TRUE" for a model in which the signs of the attribute variables are reversed for the possible worst (Flynn et al. 2007 and 2008), or FALSE when not doing so (Hensher et al. 2015, Appendix 6B); the argument base.attribute is set as a character vector showing the base attribute for a marginal (sequential) model with effect-coded attribute variables; and the argument base.level is set as a list containing the base level in each attribute for a model with effect-coded level variables (Flynn et al. 2007 and

2008), while it is set as `NULL` for a model with dummy-coded attribute level variables (Hensher et al. 2015, Appendix 6B).

Note that the arguments `attribute.variables`, `effect`, `delete.best`, and `type` are deprecated and will be removed in the future.

Value

The function returns a dataset in data frame format for the paired model or one for the marginal (sequential) model. The dataset for the paired model contains the following variables and attribute and/or attribute-level variables explained above:

- `id` A respondent's identification number; the actual name and values of this variable is set according to the id variable in the respondent dataset.
- `Q` A serial number of BWS questions.
- `PAIR` A serial number for the possible pairs of the best and worst attribute levels for each question.
- `BEST` An attribute-level number treated as the best in the possible pairs of the best and worst attribute levels for each question.
- `WORST` An attribute-level number treated as the worst in the possible pairs of the best and worst attribute levels for each question.
- `BEST.AT` A character showing the attribute corresponding to the attribute level treated as the best in the possible pairs of the best and worst attribute levels for each question.
- `WORST.AT` A character showing the attribute corresponding to the attribute level treated as the worst in the possible pairs of the best and worst attribute levels for each question.
- `BEST.LV` A character showing the attribute level treated as the best in the possible pairs of the best and worst attribute levels for each question.
- `WORST.LV` A character showing the attribute level treated as the worst in the possible pairs of the best and worst attribute levels for each question.
- `RES.B` A row number in the profile corresponding to the attribute level selected as the best by respondents.
- `RES.W` A row number in the profile corresponding to the attribute level selected as the worst by respondents.
- `RES` Responses to BWS questions that takes the value of `1` if a possible pair of the best and worst attribute levels is selected by respondents and `0` otherwise: this variable is used as a dependent variable in the model formula of the function for discrete choice analysis (e.g., `clogit()` in the package **survival**).
- `STR` A stratification variable identifying each combination of respondent and question; the variable is also used in the model formula of `clogit()`.

The dataset 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 (attribute levels) for each question.

BW	A state variable that takes the value of 1 for the possible best attribute levels and -1 for the possible worst attribute levels.
ATT.cha	A character showing the attribute corresponding to the attribute level treated as the possible best or worst for each question.
ATT	An attribute number showing the attribute corresponding to the attribute level treated as the possible best or worst for each question.
LEV.cha	A character showing the attribute levels treated as the possible best or worst for each question.
LEV	An attribute level number showing the attribute level treated as the possible best or worst for each question.
RES	Responses to BWS questions that takes the value of 1 if the possible best or worst attribute level is selected by respondents and 0 otherwise.

The output has its attributes that consist of arguments assigned to this function (i.e., `id`, `response`, `choice.sets`, `attribute.levels`, `reverse`, `base.attribute`, `base.level`, `attribute.variables`, `effect`, `delete.best`, and `type`) and the following:

design.matrix	Design matrix.
lev.var.wo.ref	Names of attribute-level variables excluding base levels.
freq.levels	Frequency of attribute levels shown in all the questions.
respondent.characteristics	Names of variables corresponding to the respondents' characteristics: variables, except for the respondents' id and response variables, are considered the respondents' characteristics.

Author(s)

Hideo Aizaki

See Also

support.BWS2-package, oa.design, clogit

Examples

```
# Load package survival used for a conditional logit model analysis of
# the responses
require(survival)

# Set a three-level orthogonal main-effect design (OMED) with
# four columns
omed <- matrix(
  c(1,3,2,3,
    3,1,2,2,
    3,3,3,1,
    2,3,1,2,
    2,2,2,1,
    1,1,1,1,
    1,2,3,2,
```

```
    3,2,1,3,
    2,1,3,3),
  nrow = 9, ncol = 4, byrow = TRUE)
omed
## The OMED is generated by executing the following lines of code:
## require(DoE.base)
## set.seed(123)
## omed <- data.matrix(oa.design(nl = c(3, 3, 3, 3)))

# Set the names of the attributes and attribute levels
attr.lev <- list(
  A = c("A1","A2","A3"), B = c("B1","B2","B3"),
  C = c("C1","C2","C3"), D = c("D1","D2","D3"))

# Convert the OMED into Case 2 BWS questions using three formats:
## Attribute column is located on the left-hand side
bws2.questionnaire(omed, attribute.levels = attr.lev,
  position = "left")
## Attribute column is located in the center
bws2.questionnaire(omed, attribute.levels = attr.lev,
  position = "center")
## Attribute column is located on the right-hand side
bws2.questionnaire(omed, attribute.levels = attr.lev,
  position = "right")

# Set respondent dataset containing 20 respondents who answered
# nine BWS questions
resp.data <- data.frame(
  id = c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20),
  B1 = c(2,2,2,1,2,4,2,2,2,2,1,2,2,4,2,3,2,3,2,2),
  W1 = c(1,1,1,4,1,3,3,1,4,1,4,4,1,1,1,4,1,1,4,4),
  B2 = c(1,1,2,1,1,3,1,1,1,1,2,1,1,2,1,3,1,3,1,1),
  W2 = c(2,4,4,4,4,2,4,2,4,2,4,4,4,4,2,4,4,1,4,4),
  B3 = c(1,1,2,1,2,1,1,1,1,2,1,1,1,2,1,1,1,1,3,1),
  W3 = c(4,4,4,2,4,4,4,3,4,3,4,4,3,1,4,4,3,4,4,4),
  B4 = c(1,2,2,1,2,1,2,2,2,1,2,4,2,2,2,2,4,2,2,1,2),
  W4 = c(3,4,3,2,3,3,3,1,4,3,3,3,4,3,3,1,4,3,4,4),
  B5 = c(1,2,2,1,2,1,2,1,3,1,1,1,3,1,1,1,3,1,1,1),
  W5 = c(4,1,3,4,4,4,3,4,4,4,2,4,4,2,4,2,1,4,3,4),
  B6 = c(2,4,2,1,2,1,4,3,1,1,1,1,3,2,1,2,3,4,1,4),
  W6 = c(4,1,4,4,4,3,3,4,4,2,4,2,4,4,3,4,4,1,4,1),
  B7 = c(3,3,2,3,4,1,2,3,3,3,2,1,3,2,1,2,3,1,3,2),
  W7 = c(1,4,1,4,1,4,4,4,4,2,4,4,4,4,4,4,4,4,4,4),
  B8 = c(1,1,2,1,2,2,1,1,1,2,1,2,1,1,1,3,1,1,1,1),
  W8 = c(3,3,3,3,3,3,3,3,4,3,3,3,4,3,3,4,4,3,4,3),
  B9 = c(3,3,3,1,3,1,1,3,1,1,1,1,3,1,1,1,3,1,1,1),
  W9 = c(2,1,2,2,2,2,4,2,4,2,4,2,2,2,2,4,1,2,2,2))

# Create a dataset and conduct a conditional logit model analysis
## Set response variables
response.vars <- names(resp.data)[2:19]
## Set a base level in each attribute
base.lev <- list(
```

```
  A = c("A3"), B = c("B3"), C = c("C3"), D = c("D3"))
## Paired model with attribute and attribute-level variables
pr.data <- bws2.dataset(
  data = resp.data,
  id = "id",
  response = response.vars,
  choice.sets = omed,
  attribute.levels = attr.lev,
  reverse = TRUE,
  base.level = base.lev,
  model = "paired")
attributes(pr.data)$design.matrix
head(pr.data, 12)
### Attribute variable D is omitted from the model
pr <- clogit(RES ~ A + B + C +
  A1 + A2 + B1 + B2 + C1 + C2 + D1 + D2 + strata(STR),
  data = pr.data)
pr
### Calculate coefficients of base level variables
b.pr <- coef(pr)
-sum(b.pr[4:5]) # attribute level A3
-sum(b.pr[6:7]) # attribute level B3
-sum(b.pr[8:9]) # attribute level C3
-sum(b.pr[10:11]) # attribute level D3
## Marginal model with attribute and attribute-level variables
mr.data <- bws2.dataset(
  data = resp.data,
  id = "id",
  response = response.vars,
  choice.sets = omed,
  attribute.levels = attr.lev,
  reverse = TRUE,
  base.level = base.lev,
  model = "marginal")
attributes(mr.data)$design.matrix
head(mr.data, 8)
### Attribute variable D is omitted from the model
mr <- clogit(RES ~ A + B + C +
  A1 + A2 + B1 + B2 + C1 + C2 + D1 + D2 + strata(STR),
  data = mr.data)
mr
### Calculate coefficients of base level variables
b.mr <- coef(mr)
-sum(b.mr[4:5]) # attribute level A3
-sum(b.mr[6:7]) # attribute level B3
-sum(b.mr[8:9]) # attribute level C3
-sum(b.mr[10:11]) # attribute level D3

# Calculate BWS scores
bwscores <- bws2.count(mr.data)
sum(bwscores, "level")
barplot(bwscores, "bw", "level")
```

`bws2.questionnaire` *Converting an orthogonal main-effect design into Case 2 best–worst scaling questions*

Description

This function converts an orthogonal main-effect design into a series of Case 2 best–worst scaling questions.

Usage

```
bws2.questionnaire(choice.sets, attribute.levels = NULL,
  position = c("left", "center", "right"))
```

Arguments

`choice.sets` A data frame or matrix containing an orthogonal main-effect design.

`attribute.levels` A list containing the names of attributes and their levels.

`position` A character showing the position where attribute levels are shown in questions.

Details

The `bws2.questionnaire()` function converts an orthogonal main-effect design (OMED) into a series of Case 2 best–worst scaling (BWS) questions and then displays the resultant questions on an R console.

An OMED is assigned to the argument `choice.sets`, which may be generated by R functions such as `oa.design()` in **DoE.base** or manually copied from text books or websites related to experimental designs.

Attributes and their levels are assigned to the argument `attribute.levels` in list format. For example, suppose that profiles have four attributes, each of which has three levels: attribute A with levels A1, A2, and A3; attribute B with levels B1, B2, and B3; attribute C with levels C1, C2, and C3; and attribute D with levels D1, D2, and D3. In this case, the argument is set as follows:

```
attribute.levels = list(
A = c("A1", "A2", "A3"),
B = c("B1", "B2", "B3"),
C = c("C1", "C2", "C3"),
D = c("D1", "D2", "D3"))
```

The argument `position` is used to change the position of the attribute column in the resultant questions. When setting `position = "left"`, the attribute column is located on the left-hand side as follows:

```
Q1
Attribute Best Worst
A1        [ ]  [ ]
```

```
B1        [ ]  [ ]
C1        [ ]  [ ]
D1        [ ]  [ ]
```

When setting `position = "center"`, the attribute column is located in the center as follows:

```
Q1
Best Attribute Worst
[ ]  A1        [ ]
[ ]  B1        [ ]
[ ]  C1        [ ]
[ ]  D1        [ ]
```

When setting `position = "right"`, the attribute column is located on the right-hand side as follows:

```
Q1
Best Worst Attribute
[ ]  [ ]   A1
[ ]  [ ]   B1
[ ]  [ ]   C1
[ ]  [ ]   D1
```

Value

BWS questions converted from the design are returned.

Author(s)

Hideo Aizaki

See Also

`support.BWS2-package`, `bws2.dataset`, `oa.design`

Examples

```
## See examples in bws2.dataset()
```

Index