Package 'bwsTools'

October 12, 2022

Type Package

Title Tools for Case 1 Best-Worst Scaling (MaxDiff) Designs

Version 1.2.0

Description Tools to design best-worst scaling designs (i.e., balanced incomplete block designs) and to analyze data from these designs, using aggregate and individual methods such as: difference scores, Louviere, Lings, Islam, Gudergan, & Flynn (2013) <doi:10.1016/j.ijresmar.2012.10.002>; analytical estimation, Lipovetsky & Conklin (2014) <doi:10.1016/j.jocm.2014.02.001>; empirical Bayes, Lipovetsky & Conklin (2015) <doi:10.1142/S1793536915500028>; Elo, Hollis (2018) <doi:10.3758/s13428-017-0898-2>; and network-based measures.

License MIT + file LICENSE

URL https://github.com/markhwhiteii/bwsTools

BugReports https://github.com/markhwhiteii/bwsTools/issues

Depends R (>= 3.5.0)

Imports crossdes, dplyr, igraph, magrittr, rlang, tidyr

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

VignetteBuilder knitr

Suggests testthat, knitr, rmarkdown, ggplot2

NeedsCompilation no

Author Mark White [aut, cre]

Maintainer Mark White <markhwhiteii@gmail.com>

Repository CRAN

Date/Publication 2020-08-26 22:40:03 UTC

R topics documented:

---

ae_mnl   *Analytical Estimation of a Multinomial Logit Model for Best-Worst Scaling*

Description

This uses Equations 7, 10, 12, 13, and 18 from Lipovetsky & Conklin (2014) to take vectors of total times shown to participants, total times selected as best, and total times selected as worst. It uses their closed-form solution to calculate utility coefficients—as well as their standard errors and confidence intervals—and choice probabilities.

Usage

```
ae_mnl(data, totals, bests, worsts, z = 1.96)
```

Arguments

data	A data.frame where each row represents an item, and three columns represent total times shown to participants, total times selected as bests, and total time selected as worsts.
totals	A string that is the name of the column for totals in the data.
bests	A string that is the name of the column for bests in the data.
worsts	A string that is the name of the column for worsts in the data.
z	A z-value to calculate the confidence intervals. Defaults to 1.96, a 95% CI.

Value

A data.frame containing the utility coefficients (with standard error and confidence intervals) and choice probabilities for each item (row) in the data.

References

Lipovetsky, S., & Conklin, M. (2014). Best-worst scaling in analytical closed-form solution. The Journal of Choice Modelling, 10, 60-68. doi: 10.1016/j.jocm.2014.02.001

Examples

```
# Replicate Table 6 from Lipovetsky & Conklin (2014)
d <- data.frame(
  totals = c(7145, 7144, 7144, 7144, 7145, 7145, 7144, 7146, 8166,
             7145, 7144, 7144, 7145, 7144, 7145, 7144, 7146),
  bests = c(1733, 968, 5218, 2704, 2307, 692, 1816, 689, 2483, 1422,
            362, 2589, 4158, 825, 829, 859, 966),
  worsts = c(1324, 2139, 113, 1010, 772, 3986, 1438, 2397, 1041, 1538,
  4597, 966, 305, 2875, 2256, 2259, 1604)
)
results <- ae_mnl(d, "totals", "bests", "worsts")
(d <- cbind(d, results))
```

---

agg   *Example Data for Non-BIBD Aggregate-Level Best-Worst Scaling*

Description

A tibble that shows the format a tibble should take when it is submitted to an aggregate scoring function that is NOT ae_mnl(), which assumes a balanced incomplete block design (BIBD).

Usage

```
agg
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 400 rows and 4 columns.

---

bibds   *Balanced Incomplete Block Designs for use in Best-Worst Scaling*

Description

A data.frame of 32 different balanced incomplete block designs (BIBD) that can be used to set up best-worst scaling studies. The columns are: design, which is an identification number that can is fed to get_bibd; t, which is the number of items (or "treatments") one will be using in the study; k, which is how many of the items the respondent sees per trial; r, which is how many times the participant sees each item across all of the trials; b, which is how many trials (or "blocks") there are in the design; and lambda, which indicates how many times each pair of options co-occur throughout the design. This is taken from Table 11.3 in Cochran & Cox (1957), including only the designs where t and b are both less than or equal to 20 (as any larger would put cognitive strain on a respondent).

Usage

```
bibds
```

Format

An object of class `data.frame` with 32 rows and 6 columns.

References

Cochran, W. G., & Cox, G. M. (1957). Balanced and partially balanced incomplete block designs. In Experimental designs (2nd ed., pp. 439–482). New York: John Wiley & Sons, Inc.

---

diffscoring   *Difference Method to Calculate Individual Best-Worst Scores*

Description

Calculate best-worst scores for each respondent-item combination. This is simply taking the total number of times a respondent selected the item as "best" and subtracting from that the number of times a respondent selected the item as "worst" (Louviere et al., 2013).

Usage

```
diffscoring(data, id, block, item, choice, std = FALSE, wide = FALSE)
```

Arguments

data	A data.frame of the type described in details.
id	A string of the name of the id column.
block	A string of the name of the block column.
item	A string of the name of the item column.
choice	A string of the name of the choice column.
std	Logical of whether or not one wants to standardize the data to a -1 to +1 range.
wide	Logical of whether or not one wants the data returned in long (each row is an item-respondent combination and all best-worst scores are in the same column) format (FALSE) or in wide format (where each row is a respondent, and the best-worst scores for the items are in their own columns). See the 'indiv' data as an example.

Details

This function requires data to be in a specified format. Each row must represent a respondent-block-label combination. That is, it indicates the person, the block (or trial), the item that was judged, and a column indicating whether it was chosen as best (+1), worst (-1), or wasn't selected as either (0).

Value

A data.frame containing the id and item columns as well as a "bws" column that indicates the best worst score. If 'wide = TRUE', then each item has its own column and the bws is filled-in those columns.

References

Louviere, J., Lings, I., Islam, T., Gudergan, S., & Flynn (2013). An introduction to the application of (case 1) best-worst scaling in marketing research. International Journal of Research in Marketing, 30(3), 292-303. doi: 10.1016/j.ijresmar.2012.10.002

Examples

```
data(indiv)
head(indiv)
diffscoring(indiv, "id", "block", "label", "value")
diffscoring(indiv, "id", "block", "label", "value", TRUE, TRUE)
```

---

elo   *Elo Method to Calculate Aggregate Best-Worst Scores*

Description

Calculate aggregate best-worst scores using Elo scoring. This specific application comes from Hollis (2018). It makes each individual/block pairwise comparisons and updates Elo scores based on who won and lost those comparisons. No ties are considered, which occurs between all of the items that have not been selected as either best or worst. Hollis (2018) also recommends adding two "dummy items": one that defeats every other item, and one that loses to every other item. This is employed here. The default K is 30, per Hollis (2018). Since Elo is temporal in nature, Hollis also recommends running various iterations, each with a different randomization of the order of matchups. The default is the 100 used by Hollis. These are averaged together to calculate individual Elo best-worst scores. Elo scores are all initialized at 1000.

Usage

```
elo(data, id, block, item, choice, K = 30, iter = 100)
```

Arguments

data	A data.frame of the type described in details.
id	A string of the name of the id column.
block	A string of the name of the block column.
item	A string of the name of the item column.
choice	A string of the name of the choice column.
K	The Elo K-factor. The default is 30, per Hollis (2018).
iter	Number of different randomizations of the "matchup" order to iterate through. The default is Hollis's (2018, 2019) recommendation.

Details

This function requires data to be in a specified format. Each row must represent a respondent-block-label combination. That is, it indicates the person, the block (or trial), the item that was judged, and a column indicating whether it was chosen as best (+1), worst (-1), or wasn't selected as either (0).

Value

A data.frame containing the item column as well as an "elo" column that indicates the Elo score.

References

Hollis, G. (2018). Scoring best-worst data in unbalanced many-item designs, with applications to crowdsourcing semantic judgments. Behavior Research Methods, 50(2), 711-729. doi: 10.3758/s13428-017-0898-2

Hollis, G. (2019). The role of number of items per trial in best-worst scaling experiments. Behavior Research Methods. doi: 10.3758/s13428-019-01270-w

Examples

```
data(agg)
head(agg)
# run more than 1 iter; just doing 1 here for speed
elo(agg, "pid", "trial", "character", "decision", iter = 1)
```

---

eloscoring   *Elo Method to Calculate Individual Best-Worst Scores*

Description

Calculate individual best-worst scores using Elo scoring. This specific application comes from Hollis (2018). It makes each block pairwise comparisons and updates Elo scores based on who won and lost those comparisons. No ties are considered, which occurs between all of the items that have not been selected as either best or worst. Hollis (2018) also recommends adding two "dummy items": one that defeats every other item, and one that loses to every other item. This is employed here. The default K is 30, per Hollis (2018). Since Elo is temporal in nature, Hollis also recommends running various iterations, each with a different randomization of the order of matchups. The default is the 100 used by Hollis. These are averaged together to calculate individual Elo best-worst scores. Elo scores are all initialized at 1000.

Usage

```
eloscoring(data, id, block, item, choice, K = 30, iter = 100, wide = FALSE)
```

Arguments

data	A data.frame of the type described in details.
id	A string of the name of the id column.
block	A string of the name of the block column.
item	A string of the name of the item column.
choice	A string of the name of the choice column.
K	The Elo K-factor. The default is 30, per Hollis (2018).
iter	Number of different randomizations of the "matchup" order to iterate through. The default is Hollis's (2018, 2019) recommendation.
wide	Logical of whether or not one wants the data returned in long (each row is an item-respondent combination and all best-worst scores are in the same column) format (FALSE) or in wide format (where each row is a respondent, and the best-worst scores for the items are in their own columns). See the 'indiv' data as an example.

Details

This function requires data to be in a specified format. Each row must represent a respondent-block-label combination. That is, it indicates the person, the block (or trial), the item that was judged, and a column indicating whether it was chosen as best (+1), worst (-1), or wasn't selected as either (0).

Value

A data.frame containing the id and item columns as well as a "elo" column that indicates the Elo score. If 'wide = TRUE', then each item has its own column and the Elo score is filled-in those columns.

References

Hollis, G. (2018). Scoring best-worst data in unbalanced many-item designs, with applications to crowdsourcing semantic judgments. Behavior Research Methods, 50(2), 711-729. doi: 10.3758/s13428-017-0898-2

Hollis, G. (2019). The role of number of items per trial in best-worst scaling experiments. Behavior Research Methods. doi: 10.3758/s13428-019-01270-w

Examples

```
data(indiv)
head(indiv)
# run more than 1 iter; just doing 1 here for speed
eloscoring(indiv, "id", "block", "label", "value", iter = 1)
```

---

e_bayescoring   *Empirical Bayes Method to Calculate Individual Best-Worst Scores*

Description

Individual utilities from empirical bayes estimations. Instead of doing the computationally-expensive hierarchical Bayesian multinomial logistic regression model, Lipovetsky & Conklin (2015) show an empirical Bayes way to calculate this analytically. This function calculates choice probabilities shown using Equation 10 in Lipovetsky & Conklin (2015) and transforms them to be on a linear regression coefficient scale. Default values for the E and alpha parameters are those performing best in their empirical example.

Usage

```
e_bayescoring(data, id, block, item, choice, E = 0.1, alpha = 1, wide = FALSE)
```

Arguments

data	A data.frame of the type described in details.
id	A string of the name of the id column.
block	A string of the name of the block column.
item	A string of the name of the item column.
choice	A string of the name of the choice column.
E	Value of precision shown in Equation 8 of Lipovetsky & Conklin (2015). If the naive estimate for a choice probability is 0, it is replaced with E; If the naive estimate for the choice probability is 1, i is replaced with 1 - E.
alpha	The mixing parameter shown in Equation 10 of Lipovetsky & Conklin (2015). This shapes how much the naive individual estimate and how much of the aggregate estimate influences the resulting estimate.
wide	Logical of whether or not one wants the data returned in long (each row is an item-respondent combination and all best-worst scores are in the same column) format (FALSE) or in wide format (where each row is a respondent, and the best-worst scores for the items are in their own columns). See the 'indiv' data as an example.

Details

This function requires data to be in a specified format. Each row must represent a respondent-block-label combination. That is, it indicates the person, the block (or trial), the item that was judged, and a column indicating whether it was chosen as best (+1), worst (-1), or wasn't selected as either (0).

Value

A data.frame containing the id and item columns as well as a "b_ebayes" column that indicates the utility coefficient. If 'wide = TRUE', then each item has its own column and the coefficient is filled-in those columns.

References

Lipovetsky, S., & Conklin, M. (2015). MaxDiff priority estimations with and without HB-MNL. Advances in Adaptive Data Analysis, 7(1). doi: 10.1142/S1793536915500028

Examples

```
data(indiv)
head(indiv)
e_bayescoring(indiv, "id", "block", "label", "value")
```

---

indiv	*Example Data for Individual-Level Best-Worst Scaling*

Description

A tibble that shows the format a tibble should take when it is submitted to a individual 'scoring' function.

Usage

```
indiv
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 520 rows and 4 columns.

---

make_bibd	*Make Balanced Incomplete Block Designs from bibds Designs*

Description

This function generates a balanced incomplete block design. It takes one argument, the design number from the bibds data.frame object. See bibds.

Usage

```
make_bibd(design, seed = 1839)
```

Arguments

design
: Integer from 1 to 32. Corresponds to the characteristics from the bibds data.frame object.

seed
: Integer to set seed for reproducibility, such that the same design will be returned on different occasions. Defaults to 1839, so that the function will, by default, yield reproducible designs.

Value

A tibble. The first column indicates the block, and the rest of the columns indicate which item is in each block.

---

prscoring *Page Rank Method to Calculate Individual Best-Worst Scores*

Description

Calculate best-worst scores for each respondent-item combination. This uses the page rank method. It works virtually the same as the walkscoring method; the only difference is that there is a parameter for randomly "teleporting" to a random node. This can help prevent random walks from getting "stuck" in certain areas. Page rank scores using are calculated and synthesized in the same way as the method used in walkscoring(). See White (2019).

Usage

```
prscoring(data, id, block, item, choice, ..., wide = FALSE)
```

Arguments

data	A data.frame of the type described in details.
id	A string of the name of the id column.
block	A string of the name of the block column.
item	A string of the name of the item column.
choice	A string of the name of the choice column.
...	Additional arguments to igraph::page_rank()
wide	Logical of whether or not one wants the data returned in long (each row is an item-respondent combination and all best-worst scores are in the same column) format (FALSE) or in wide format (where each row is a respondent, and the best-worst scores for the items are in their own columns). See the 'indiv' data as an example.

Details

This function requires data to be in a specified format. Each row must represent a respondent-block-label combination. That is, it indicates the person, the block (or trial), the item that was judged, and a column indicating whether it was chosen as best (+1), worst (-1), or wasn't selected as either (0).

Value

A data.frame containing the id and item columns as well as a "walk" column that indicates the best worst score. If 'wide = TRUE', then each item has its own column and the walkscore is filled-in those columns.

References

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems, 30(1), 107-117.

Gleich, D. F. (2014). PageRank beyond the Web. arxiv.org/abs/1407.5107

White, M. H., II. (2019). bwsTools: An R package for case 1 best-worst scaling. Retrieved from https://osf.io/xftvq/

Examples

```
## Not run:
data(indiv)
head(indiv)
prscoring(indiv, "id", "block", "label", "value")

## End(Not run)
```

---

vdata	*Example Data Used in Vignettes*

Description

A tibble that shows the format a tibble should take when it is submitted to a individual 'scoring' function.

Usage

```
vdata
```

Format

An object of class spec_tbl_df (inherits from tbl_df, tbl, data.frame) with 18200 rows and 4 columns.

---

walkscoring	*Walkscoring Method to Calculate Individual Best-Worst Scores*

Description

Calculate best-worst scores for each respondent-item combination. This uses the walkscoring method described in White (2019).

Usage

```
walkscoring(data, id, block, item, choice, walks = 10000, wide = FALSE)
```

Arguments

data	A data.frame of the type described in details.
id	A string of the name of the id column.
block	A string of the name of the block column.
item	A string of the name of the item column.
choice	A string of the name of the choice column.
walks	Integer indicating how many random walks to simulate.
wide	Logical of whether or not one wants the data returned in long (each row is an item-respondent combination and all best-worst scores are in the same column) format (FALSE) or in wide format (where each row is a respondent, and the best-worst scores for the items are in their own columns). See the 'indiv' data as an example.

Details

This function requires data to be in a specified format. Each row must represent a respondent-block-label combination. That is, it indicates the person, the block (or trial), the item that was judged, and a column indicating whether it was chosen as best (+1), worst (-1), or wasn't selected as either (0).

Value

A data.frame containing the id and item columns as well as a "walk" column that indicates the best worst score. If 'wide = TRUE', then each item has its own column and the walkscore is filled-in those columns.

References

White, M. H., II. (2019). bwsTools: An R package for case 1 best-worst scaling. Retrieved from https://osf.io/xftvq/

Examples

```
## Not run:
data(indiv)
head(indiv)
# use more than 100 walks; only using 100 here for speed
walkscoring(indiv, "id", "block", "label", "value", 100)

## End(Not run)
```

Index