Package ‘optiscale’

February 28, 2020

Type Package

Title Optimal Scaling

Version 1.2

Date 2020-02-26

Author William G. Jacoby

Maintainer William G. Jacoby <wm.g.jacoby@gmail.com>

Description Optimal scaling of a data vector is obtained through a least-squares transformation subject to appropriate measurement constraints. Further information about this methodology is provided by Young (1981)<doi:10.1007/BF02293796> and Jacoby (1999)< doi:10.2307/2991794>.

License GPL-2

Depends lattice

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2020-02-28 07:20:05 UTC

R topics documented:

`optiscale-package` *Optimal Scaling of a Data Vector*

Description

This package provides tools to perform an optimal scaling analysis on a data vector. The main result of the optimal scaling is a vector of scores which are a least-squares approximation to a vector of quantitative values, subject to measurement constraints based upon a vector of qualitative data values. See Young (1981) for details.

Details

Package:	optiscale
Type:	Package
Version:	1.2
Date:	2020-02-26
License:	GPL-2
LazyLoad:	yes

The function that performs the optimal scaling is `opscale()`. It produces an object of class "opscale". Generic methods are defined for `print`, `summary`, and `plot` (graphing optimally-scaled values versus original data values).

Author(s)

William G. Jacoby

Maintainer: William G. Jacoby <<`Jacoby@msu.edu`>>

References

Young, Forrest W. (1981) "Quantitative Analysis of Qualitative Data." *Psychometrika* 46: 357-388.

See Also

`opscale`, `plot.opscale`, `print.opscale`, `summary.opscale`

Examples

```
###   x1 is vector of qualitative data
###   x2 is vector of quantitative values
          x1 <- c(1,1,1,1,2,2,2,3,3,3,3,3,3)
          x2 <- c(3,2,2,2,1,2,3,4,5,2,6,6,4)
###   Optimal scaling, specifying that x1
###   is ordinal-discrete
   op.scaled <- opscale(x.qual=x1, x.quant=x2,
                level=2, process=1)
```

```
print(op.scaled)
summary(op.scaled)
```

---

elec92 *Public Opinion During the 1992 U.S. Presidential Election*

Description

This data set contains several variables from the Center for Political Studies 1992 National Election Study. Observations with missing values on any of the variables have been deleted.

Usage

```
data(elec92)
```

Format

A data frame with 1653 observations on the following 7 variables.

- `caseid` NES case identification number
- `bush` Respondents feeling thermometer rating of George H. W. Bush
- `ideol` Respondents ideological self-placement, seven-point scale ranging from 1=extremely liberal to 7=extremely conservative
- `econ4yr` Respondents judgment whether national economy has gotten better or worse over preceding four years, five-point scale ranging from 1=much better to 5=much worse
- `party` Respondents party identification, seven-point scale ranging from 0=strong Democrat to 6=strong Republican
- `choice` Difference in respondents feeling thermometer ratings of Bush and Clinton
- `clinton` Respondent"s feeling thermometer rating of Bill Clinton

Source

The full data set from which these observations and variables were extracted is available on the Study Page for the American National Election Studies 1992 Time Series Study, at https://electionstudies.org/data-center/1992-time-series-study/.

References

Jacoby, William G. (1999) "Levels of Measurement and Political Research: An Optimistic View." *American Journal of Political Science* 43: 271-301.

Examples

```
library(optiscale)
data(elec92)
summary(lm(choice ~ party + ideol + econ4yr, data = elec92))
```

Methods *S3 methods for opscale*

Description

Plot, print, shepard, stress, and summary methods for objects of class opscale

Usage

```
## S3 method for class 'opscale'
plot(x, ...)
## S3 method for class 'opscale'
print(x, ...)
## S3 method for class 'opscale'
summary(object, ...)
```

Arguments

object	Object of class opscale
x	Object of class opscale
...	Ignored

Details

Method print returns a listing of the data. summary describes the optimal scale transformation. plot calls os.plot and returns an object of class trellis that graphs optimally-scaled values against the original (qualitative) data values.

See Also

os.plot

Examples

```
  ###   x1 is vector of qualitative data
  ###   x2 is vector of quantitative values
            x1 <- c(1,1,1,1,2,2,2,3,3,3,3,3,3)
            x2 <- c(3,2,2,2,1,2,3,4,5,2,6,6,4)
  ###   Optimal scaling, specifying that x1
  ###   is ordinal-discrete
     op.scaled <- opscale(x.qual=x1, x.quant=x2,
                   level=2, process=1)
     print(op.scaled)
     summary(op.scaled)
     plot(op.scaled)
```

`opscale` *Optimal scaling of a data vector*

Description

This function produces an object of class "opscale", containing a vector that is a least-squares approximation to a vector of quantitative values, subject to measurement constraints based upon a vector of qualitative data values.

Usage

```
opscale(x.qual, x.quant = seq(1:length(x.qual)), level = 1,
   process = 1, na.impute = FALSE,
   na.assign = TRUE, rescale = TRUE)
```

Arguments

`x.qual`	A vector of data values, assumed to be qualitative.
`x.quant`	A vector of quantitative values.
`level`	Measurement level of `x.qual`. 1=nominal, 2=ordinal.
`process`	Measurement process of `x.qual`. 1=discrete, 2=continuous.
`na.impute`	If `TRUE`, then assign `x.quant` values to optimally scaled vector for missing entries in `x.qual`. of `FALSE` then assign `NA` to entries in optimally scaled vector corresponding to missing entries in `x.qual`.
`na.assign`	If `TRUE`, then if `x.quant` is missing, assign mean of optimally scaled values for the corresponding `x.qual` value to the optimally scaled vector. If `FALSE`, then leave optimally scaled value missing, even if `x.qual` value is present.
`rescale`	If `TRUE` then rescale optimally scaled vector to same mean and standard deviation as `x.qual`.

Details

The `opscale()` function operationalizes a measurement theory proposed by Young (1981) in order to facilitate an analysis strategy called “Alternating Least Squares, Optimal Scaling”. The optimal scaling transformation produces a vector (say, `OS`) that is a least-squares approximation to `x.quant`, subject to measurement constraints based upon `x.qual`. If `x.qual` is nominal level, then the values in `OS` are the conditional means of `x.quant`, within distinct categories of `x.qual`. If `x.qual` is ordinal level, then the values in `OS` are the conditional means of `x.quant`, adjusted to be weakly monotonic to the values in `x.qual`, using Kruskals (1964b) monotonic transformation. If `x.qual` is discrete, then all data objects sharing a common value in `x.qual` must be assigned the same value in `OS`. If `x.qual` is continuous, then data objects sharing a common value in `x.qual` can fall within a closed interval of values in `OS`. The continuous-discrete measurement process distinction corresponds to Kruskals (1964a) primary and secondary approaches to ties.

Value

The `opscale()` function returns an object of class "opscale" containing a list with the following components:

`qual`	The qualitative data vector, `x.qual`
`quant`	The vector of quantitative values, `x.quant`
`os`	The vector of optimally scaled values
`varname`	The name of the qualitative data vector, `x.qual`
`measlev`	The measurement level of the qualitative data vector specified in the `level` argument to `opscale`
`measproc`	The measurement process of the qualitative data vector specified in the `process` argument to `opscale`
`rescale`	Value of the `rescale` argument to `opscale`
`os.raw.mean`	Mean of optimally scaled values before rescaling
`os.raw.sd`	Standard deviation of optimally scaled values before rescaling

References

Kruskal, Joseph B. (1964a) “Multidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis.” *Psychometrika* 29: 1-27.

Kruskal, Joseph B. (1964b) “Nonmetric Multidimensional Scaling: A Numerical Method.” *Psychometrika* 29: 115-129.

Young, Forrest W. (1981) “Quantitative Analysis of Qualitative Data.” *Psychometrika* 46: 357-388.

See Also

`plot.opscale`, `print.opscale`, `summary.opscale`

Examples

```
###   x1 is vector of qualitative data
###   x2 is vector of quantitative values
          x1 <- c(1,1,1,1,2,2,2,3,3,3,3,3,3)
          x2 <- c(3,2,2,2,1,2,3,4,5,2,6,6,4)
###   Optimal scaling, specifying that x1
###   is ordinal-discrete
   op.scaled <- opscale(x.qual=x1, x.quant=x2,
                level=2, process=1)
   print(op.scaled)
   summary(op.scaled)
```

os.plot — *Graph of optimal scaling transformation*

Description

Line and point plot showing optimally-scaled values on the vertical axis, original data values (assumed to be qualitative) on the horizontal axis.

Usage

```
os.plot(x.qual, os.data,
   main.title = "Plot of optimal transformation")
```

Arguments

`x.qual`	Vector of data values, assumed to be qualitative.
`os.data`	Vector of optimally-scaled data values.
`main.title`	Main title for plot.

Value

Object of class `trellis`.

See Also

`plot.opscale`,

Examples

```
###   x1 is vector of qualitative data
###   x2 is vector of quantitative values
          x1 <- c(1,1,1,1,2,2,2,3,3,3,3,3,3)
          x2 <- c(3,2,2,2,1,2,3,4,5,2,6,6,4)
###   Optimal scaling, specifying that x1
###   is ordinal-discrete
   op.scaled <- opscale(x.qual=x1, x.quant=x2,
                level=2, process=1)
###   Plot of optimal scaling transformation
   os.plot(op.scaled$qual, op.scaled$os)
```

shepard *Shepard diagram for opscale*

Description

Graph showing data (assumed quantitative) on vertical axis, optimally-scaled data on horizontal axis.

Usage

```
shepard(x, ...)

shep.plot(x.quant, os.data, main.title = "Shepard Diagram")
```

Arguments

`x`	An object of class `opscale`
`x.quant`	Data vector, assumed to be quantitative.
`os.data`	Optimally-scaled data.
`main.title`	Main title for graph.
`...`	Ignored

Value

`shepard()` and `shep.plot()` both produce an object of class `trellis`

Warning

If using `shep.plot()`, the Shepard diagram should be created using "raw" optimally scaled values. That is, the OS values should NOT be rescaled to the mean and standard deviation of the original qualitative data.

Examples

```
###   x1 is vector of qualitative data
###   x2 is vector of quantitative values
          x1 <- c(1,1,1,1,2,2,2,3,3,3,3,3,3)
          x2 <- c(3,2,2,2,1,2,3,4,5,2,6,6,4)
###   Optimal scaling, specifying that x1
###   is ordinal-discrete, optimally scaled
###   values are not rescaled
   op.scaled <- opscale(x.qual=x1, x.quant=x2,
                level=2, process=1,
                rescale=FALSE)
###   Create Shepard diagram
   shepard(op.scaled)
###   Same results are produced by:
   shep.plot(op.scaled$quant, op.scaled$os)
```

stress *Stress coefficients for* opscale

Description

Calculates stress coefficients summarizing lack of fit between two vectors.

Usage

```
stress(x, ...)

calc.stress(quant, os, rescale = FALSE,
   os.raw.mean = mean(os, na.rm = TRUE),
   os.raw.sd = sd(os, na.rm = TRUE))
```

Arguments

x	Object of class opscale
quant	Data vector.
os	Vector of optimally-scaled data
rescale	If TRUE, the optimally-scaled data have been rescaled to the mean and standard deviation of the original qualitative data vector that was used in the optimal scaling transformation.
os.raw.mean	User-specified mean for optimally-scaled data, defaults to mean of os. Only needed if rescale = TRUE.
os.raw.sd	User-specified standard deviation for optimally-scaled data, defaults to standard deviation of os. Only needed if rescale = TRUE.
...	Ignored

Value

stress() and calc.stress() both produce a vector with three elements:

stress1	Kruskals Stress 1 coefficient
stress2	Kruskals Stress 2 coefficient
raw.stress	Sum of squared residuals between quant and os

Warning

If using calc.stress(), the stress coefficients must be created using "raw" optimally scaled values. That is, the OS values should NOT be rescaled to the mean and standard deviation of the original qualitative data.

Examples

```
###   x1 is vector of qualitative data
###   x2 is vector of quantitative values
          x1 <- c(1,1,1,1,2,2,2,3,3,3,3,3,3)
          x2 <- c(3,2,2,2,1,2,3,4,5,2,6,6,4)
###   Optimal scaling, specifying that x1
###   is ordinal-discrete, optimally scaled
###   values are not rescaled
   op.scaled <- opscale(x.qual=x1, x.quant=x2,
                level=2, process=1,
                rescale=FALSE)
###  Calculate stress coefficients
  stress(op.scaled)
###   Same results can be obtained with:
  calc.stress(op.scaled$quant, op.scaled$os)
```

Index