Package ‘kequate’

March 11, 2013

Type Package

Title The kernel method of test equating

Version 1.2.1

Date 2013-03-11

Author Bjorn Andersson, Kenny Branberg and Marie Wiberg

Maintainer Bjorn Andersson <bjorn.andersson@statistik.uu.se>

Description Implements the kernel method of test equating using the 
CB, EG, SG, NEAT CE/PSE and NEC designs, supporting gaussian,logistic and uniform ker-
nels and unsmoothed and pre-smoothed 
input data.

License GPL-3

LazyLoad yes

Depends R(>= 2.11.0), methods, stats, graphics

NeedsCompilation no

Repository CRAN

Date/Publication 2013-03-11 17:10:24

R topics documented:

kequate-package .................................................. 2
cdist ................................................................. 3
cdist-class .......................................................... 4
FTres ................................................................. 5
genseed .............................................................. 6
genseed-class ....................................................... 7
getEq ................................................................. 8
getEqirt ............................................................. 9
The kernel equating technique for equating test scores is implemented, supporting the Equivalent Groups (EG), Single Group (SG), Counterbalanced (CB), Non-Equivalent groups with Anchor Test Chain Equating (NEAT CE), Non-Equivalent groups with Anchor Test Post-Stratification Equating (NEAT PSE) and Non-Equivalent groups with Covariates (NEC) designs. Support for three types of kernels is provided: gaussian, logistic and uniform. Standard errors of equating and standard errors of the difference between two equating functions are provided for all designs and kernels. Also included are functions aiding the search for a proper log-linear pre-smoothing model and the ability to use Item Response Theory Observed Score Equating (IRT-OSE) in the Kernel Equating framework.
cdist

Author(s)
Bjorn Andersson, Kenny Branberg and Marie Wiberg
Maintainer: Bjorn Andersson <bjorn.andersson@statistik.uu.se>

References

cdist

Conditional Mean, Variance, Skewness and Kurtosis

Description
Calculates conditional means, variances, skewnesses and kurtoses for observed and estimated bivariate probability distributions of test scores.

Usage
`cdist(est, obs, xscores, ascores)`

Arguments

- `est`: Matrix of estimated bivariate score probabilities.
- `obs`: Matrix of observed bivariate score probabilities.
- `xscores`: Optional argument to specify the score vector for test X.
- `ascores`: Optional argument to specify the score vector for test A.

Value
An object of class `cdist` containing the following slots

- `est1`: Matrix of conditional means, variances, skewnesses and kurtoses of X given A for the estimated score distribution.
- `est2`: Matrix of conditional means, variances, skewnesses and kurtoses of A given X for the estimated score distribution.
- `obs1`: Matrix of conditional means, variances, skewnesses and kurtoses of X given A for the observed score distribution.
- `obs2`: Matrix of conditional means, variances, skewnesses and kurtoses of A given X for the observed score distribution.

Author(s)

`<bjorn.andersson@statistik.uu.se>`
`<kenny.branberg@stat.umu.se>`
`<marie.wiberg@stat.umu.se>`
References


See Also

kequate PREP

Examples

freqdata<-data.frame(X=c(1,2,2,1,2,2,2,3,1,2,1,4,2,1,1,3,3,3,3),
A=(c(0,2,1,1,0,3,1,2,2,0,2,0,3,1,1,2,2,1,2)))
Pdata<-kefreq(freqdata$X, 0:5, freqdata$A, 0:3)
Pglm<-glm(frequency~X+I(X^2)+A+I(A^2)+X:A, data=Pdata, family="poisson", x=TRUE)
Pobs<-matrix(Pdata$freq, nrow=6)/sum(Pglm$y)
Pest<-matrix(Pglm$fitted.values, nrow=6)/sum(Pglm$y)
cdP<-cdist(Pest, Pobs, 0:5, 0:3)
plot(cdP)

---

cdist-class  

Class "cdist"

Description

Stores conditional means, variances, skewnesses and kurtoses of observed and estimated bivariate distributions of test scores.

Objects from the Class

Objects can be created by calls of the form new("cdist", ...).

Slots

est1 Matrix of conditional means, variances, skewnesses and kurtoses of X given A for the estimated score distribution.
est2 Matrix of conditional means, variances, skewnesses and kurtoses of A given X for the estimated score distribution.
obs1 Matrix of conditional means, variances, skewnesses and kurtoses of X given A for the observed score distribution.
ob2 Matrix of conditional means, variances, skewnesses and kurtoses of A given X for the observed score distribution.

Methods

plot signature(x = "keout"): ...
**FTres**

Author(s)

<bjorn.andersson@statistik.uu.se>
<kenny.branberg@stat.umu.se>
<marie.wiberg@stat.umu.se>

See Also

cdist

Examples

```r
showClass("cdist")
```

<table>
<thead>
<tr>
<th>FTres</th>
<th>Freeman-Tukey Residuals</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Description

Calculates the Freeman-Tukey residuals for log-linear models of frequency data. If the frequencies are assumed to be Poisson distributed, then the Freeman-Tukey residuals are approximately normal distributed.

Usage

```r
FTres(obs, fit)
```

Arguments

- **obs**  
  A numeric vector containing the observed frequencies.
- **fit**  
  A numeric vector containing the estimated frequencies.

Details

For an observed frequency \( n_i \) and the estimated frequency \( m_i \), the Freeman-Tukey residual \( FT_i \) is defined as \( FT_i = \sqrt{n_i} + \sqrt{n_i + 1} - \sqrt{4m_i + 1} \).

Value

A numeric vector containing the Freeman-Tukey residuals.

Author(s)

<bjorn.andersson@statistik.uu.se>
<kenny.branberg@stat.umu.se>
<marie.wiberg@stat.umu.se>
References

See Also
glm

Examples

```r
#Example data:
P<-c(5, 20, 35, 25, 15)
x<-4
glm<-glm(P~I(x)+I(x^2), family="poisson", x=TRUE)
res<-FTres(glm$y, glm$fitted.values)
```

---

**genseed**

*Standard Error of the Difference Between Two Equating Functions*

Description
Given two equating functions with associated standard error vectors, the standard error of the difference between two equating functions (SEED) is calculated.

Usage

```r
genseed(in1, in2, linear = FALSE)
```

Arguments

- **in1**: S4 object of class keout containing information about the first equating function, as created by the function kequate.
- **in2**: S4 object of class keout containing information about the second equating function, as created by the function kequate.
- **linear**: A logical vector indicating if the two linear equatings are to be compared instead of the two equipercentile equatings. Default is FALSE.

Details

The standard error of the difference between two equating functions (SEED) can be used to compare equating functions and decide which is most suitable to use. For example, if the difference between a linear and an equipercentile equating function is small and within the boundaries of the estimated SEED the simpler linear equating function may be preferable. Any two equipercentile or linear equating functions can be compared using genseed, given that the log-linear model is identical between the two (and that the designs are comparable). As an example, it is impossible to compare an EG equating to an SG equating in this manner as the log-linear models are not the same. It is however useful to compare a chain equating to a post-stratification equating in the NEAT design, or to compare post-stratification equatings with different weights.
Value
A data frame containing the difference between the equated values of two equating functions and
the SEED of the two equatings for equating X to Y (eqYxD and SEEDYx).

Author(s)
<bjorn.andersson@statistik.uu.se>
<kenny.branberg@stat.umu.se>
<marie.wiberg@stat.umu.se>

References
Springer-Verlag New York.

See Also
kequate

Examples
P<-c(5, 2/zero.noslash, 35, 25, 15)
Q<-c(1/zero.noslash, 3/zero.noslash, 3/zero.noslash, 2/zero.noslash, 1/zero.noslash)
x<-/zero.noslash:4
glmx<-glm(P~I(x)+I(x^2), family="poisson", x=TRUE)
glmY<-glm(Q~I(x)+I(x^2), family="poisson", x=TRUE)
keEG<-kequate("EG", 0:4, 0:4, glmx, glmy)
keEGnew<-kequate("EG", 0:4, 0:4, glmx, glmy, hx=0.33, hy=0.33)
SEEDEG<-genseed(keEG, keEGnew)
plot(SEEDEG)

genseed-class

Description
Stores the difference between two equatings which use identical log-linear models, along with the
standard error of the difference of the two equatings.

Objects from the Class
Objects can be created by calls of the form new("genseed", ...).

Slots
out: Object of class "data.frame" containing the difference between the equated values of two
equating functions and the SEED of the two equatings for equating X to Y (eqYxD and SEEDYx).
Methods

plot signature(x = "genseed"): ...

Author(s)

<bjorn.andersson@statistik.uu.se>
<kenny.branberg@stat.umu.se>
<marie.wiberg@stat.umu.se>

See Also

kequate

tEq

Examples

showClass("genseed")

Description

Function to retrieve the equated values from an object of class keout created by kequate.

Usage

getEq(object)

Arguments

object An object of class keout as created by the function kequate.

Value

A numeric vector with the equated values.

Author(s)

<bjorn.andersson@statistik.uu.se>
<kenny.branberg@stat.umu.se>
<marie.wiberg@stat.umu.se>

See Also

kequate PREp
getEqirt

Examples

P<-c(5, 20, 35, 25, 15)
Q<-c(10, 30, 30, 20, 10)
x<-0:4
glmX<-glm(P~I(x)+I(x^2), family="poisson", x=TRUE)
glmY<-glm(Q~I(x)+I(x^2), family="poisson", x=TRUE)
keEG<-kequate("EG", 0:4, 0:4, glmX, glmY)
getEq(keEG)

getEqirt

Retrieve Equated Values From an IRT-Equating

Description

Function to retrieve the equated values of the IRT observed score equating from an object of class keout created by kequate.

Usage

getEqirt(object)

Arguments

object An object of class keout as created by the function kequate.

Value

A vector containing the equated values of the IRT-equating.

Author(s)

<bjorn.andersson@statistik.uu.se>
<kenny.branberg@stat.umu.se>
<marie.wiberg@stat.umu.se>

See Also

kequate

Examples

irtmatX<-matrix(0, 4, 3)
irtmatX[,1]<-c(0.1, 0.3, 0.2, 0.5)
irtmatX[,2]<-c(0.2, 0.5, 0.4, 0.7)
irtmatX[,3]<-c(0.3, 0.7, 0.6, 0.9)
irtmatY<-matrix(0, 4, 3)
irtmatY[,1]<-c(0.15, 0.2, 0.1, 0.4)
irtmatY[,2]<-c(0.2, 0.4, 0.3, 0.6)
irtmatY[,3]<-c(0.25, 0.6, 0.5, 0.8)
**getEqlin**

Retrieves Equated Values of the Linear Equating.

### Description
Function to retrieve the equated values of the linear equating from an object of class `keout` created by `kequate`.

### Usage
`getEqlin(object)`

### Arguments
- **object**: An object of class `keout` as created by the function `kequate`.

### Value
A numeric vector with the equated values of the linear equating.

### Author(s)
- Bjorn Andersson
- Kenny Branberg
- Marie Wiberg

### See Also
- `kequate`

### Examples
```r
tX<-c(12, 32, 54, 22, 12)
tY<-c(24, 36, 48, 12, 12)
x<-0:4

glmXirt<-glm(tX~xirt+I(xirt^2), family="poisson", x=TRUE)
glmYirt<-glm(tY~xirt+I(xirt^2), family="poisson", x=TRUE)
keEGirt<-kequate("EG", 0:4, 0:4, glmXirt, glmYirt, tX=tirtX, tY=tirtY)
getEqirt(keEGirt)
```

```r
P<-c(5, 2, 35, 25, 15)
Q<-c(1, 3, 3, 2, 1)
x<-0:4

glmP<-glm(P~I(x)+I(x^2), family="poisson", x=TRUE)
glmQ<-glm(Q~I(x)+I(x^2), family="poisson", x=TRUE)
keEG<-kequate("EG", 0:4, 0:4, glmP, glmQ)
getEqirt(keEG)
```
getEquating

Retrieve Information From an Equating

Description
Function to retrieve information from an object of class keout created by kequate.

Usage
getEquating(object)

Arguments
object An object of class keout as created by the function kequate.

Value
A data frame with the equated values, standard errors of equating, cumulative distribution functions for the equated tests, estimated score probabilities for the equated tests and other information about the equating.

Author(s)
<bjorn.andersson@statistik.uu.se>
<kenny.branberg@stat.umu.se>
<marie.wiberg@stat.umu.se>

See Also
kequate

Examples

```r
P<-c(5, 2/zero.noslash, 35, 25, 15)
Q<-c(1/zero.noslash, 3/zero.noslash, 3/zero.noslash, 2/zero.noslash, 1/zero.noslash)
x<-/zero.noslash:4
glmx<-glm(P~I(x)+I(x^2), family="poisson", x=TRUE)
glmy<-glm(Q~I(x)+I(x^2), family="poisson", x=TRUE)
keEG<-kequate("EG", /zero.noslash:4, /zero.noslash:4, glmx, glmy)
getEquating(keEG)
```
getH

Retrieve the Continuization Bandwidths

Description

Function to retrieve the bandwidths used in an equating from an object of class keout created by kequate.

Usage

getH(object)

Arguments

object

An object of class keout as created by the function kequate.

Value

A data frame containing the values of h used in the equating.

Author(s)

<bjorn.andersson@statistik.uu.se>
<kenny.branberg@stat.umu.se>
<marie.wiberg@stat.umu.se>

See Also

kequate

Examples

P<-c(5, 20, 35, 25, 15)
Q<-c(10, 30, 30, 20, 10)
x<-0:4
 glmx<-glm(P-I(x)+I(x^2), family="poisson", x=TRUE)
 glmy<-glm(Q-I(x)+I(x^2), family="poisson", x=TRUE)
 keEG<-kequate("EG", 0:4, 0:4, glmx, glmy)
 getH(keEG)
getPre

Retrieve the Percent Relative Error

Description

Function to retrieve the percent relative error (PRE) used in an equating from an object of class keout created by kequate.

Usage

getPre(object)

Arguments

object An object of class keout as created by the function kequate.

Value

A data frame containing the PRE of the equating.

Author(s)

<bjorn.andersson@statistik.uu.se>
<kenny.branberg@stat.umu.se>
<marie.wiberg@stat.umu.se>

See Also

kequate

Examples

P<-c(5, 2/zero.noslash, 35, 25, 15)
Q<-c(1/zero.noslash, 3/zero.noslash, 3/zero.noslash, 2/zero.noslash, 1/zero.noslash)
x<-/zero.noslash:4
glmx<-glm(P~I(x)+I(x^2), family="poisson", x=TRUE)
glmy<-glm(Q~I(x)+I(x^2), family="poisson", x=TRUE)
keEG<-kequate("EG", /zero.noslash:4, /zero.noslash:4, glmx, glmy)
getPre(keEG)
getScores  

Retrieve the Score Vectors of an Equating

Description

Function to retrieve the type of equating from an object of class keout created by kequate.

Usage

getScores(object)

Arguments

object       An object of class keout as created by the function kequate.

Value

A list containing the score vector(s) used in the equating.

Author(s)

<bjorn.andersson@statistik.uu.se>
<kenny.branberg@stat.umu.se>
<marie.wiberg@stat.umu.se>

See Also

kequate

Examples

P<-c(5, 2/0, 35, 25, 15)
Q<-c(1/0, 3/0, 3/0, 2/0, 1/0)
x<-0:4

glmx<-glm(P-I(x)+I(x^2), family="poisson", x=TRUE)
glmy<-glm(Q-I(x)+I(x^2), family="poisson", x=TRUE)
keEG<-kequate("EG", 0:4, 0:4, glmx, glmy)
getScores(keEG)
Description

Function to retrieve the standard errors of an equating from an object of class keout created by kequate.

Usage

gSee(object)

Arguments

object  
An object of class keout as created by the function kequate.

Value

A vector containing the standard errors of the equating.

Author(s)

<bjorn.andersson@statistik.uu.se>
<kenny.branberg@stat.umu.se>
<marie.wiberg@stat.umu.se>

See Also

kequate

Examples

P<-c(5, 20, 35, 25, 15)
Q<-c(10, 30, 30, 20, 10)
x<-0:4
glmx<-glm(P~I(x)+I(x^2), family="poisson", x=TRUE)
glmy<-glm(Q~I(x)+I(x^2), family="poisson", x=TRUE)
keEG<-kequate("EG", 0:4, 0:4, glmx, glmy)
gSee(keEG)
getSeed

Retrieve the Standard Errors of the Difference Between Two Equatings

Description

Function to retrieve the standard error of the difference between an equipercentile equating and a linear equating from an object of class keout created by kequate.

Usage

getSeed(object)

Arguments

object An object of class keout as created by the function kequate.

Value

An object of class genseed containing the difference between an equipercentile and a linear equating and the standard errors of the difference between an equipercentile equating and a linear equating.

Author(s)

<bjorn.andersson@statistik.uu.se>
<kenny.branberg@stat.umu.se>
<marie.wiberg@stat.umu.se>

See Also

kequate, genseed

Examples

```r
P<-c(5, 20, 35, 25, 15)
Q<-c(10, 30, 30, 20, 10)
x<-0:4
glmx<-glm(P-I(x)+I(x^2), family="poisson", x=TRUE)
glmy<-glm(Q-I(x)+I(x^2), family="poisson", x=TRUE)
keEG<-kequate("EG", 0:4, 0:4, glmx, glmy)
keEGseed<-getSeed(keEG)
plot(keEGseed)
```
getSeeirt

Retrieve Equated Values From an IRT-Equating

Description
Function to retrieve the standard errors of the IRT-equating from an object of class keout created by kequate.

Usage
getSeeirt(object)

Arguments
object
An object of class keout as created by the function kequate.

Value
A vector containing the standard errors of the IRT-equating.

Author(s)
<bjorn.andersson@statistik.uu.se>
<kenny.branberg@stat.umu.se>
<marie.wiberg@stat.umu.se>

See Also
kequate

Examples
irtmatX<-matrix(0, 4, 3)
irtmatX[,1]<-c(0.1, 0.3, 0.2, 0.5)
irtmatX[,2]<-c(0.2, 0.5, 0.4, 0.7)
irtmatX[,3]<-c(0.3, 0.7, 0.6, 0.9)
irtmatY<-matrix(0, 4, 3)
irtmatY[,1]<-c(0.15, 0.2, 0.1, 0.4)
irtmatY[,2]<-c(0.2, 0.4, 0.3, 0.6)
irtmatY[,3]<-c(0.25, 0.6, 0.5, 0.8)
irtX<-c(12, 32, 54, 22, 1/0)
irtY<-c(24, 36, 48, 12, 1/0)
xirt=0:4
glmXirt<-glm(irtX~xirt+I(xirt^2), family="poisson", x=TRUE)
glmYirt<-glm(irtY~xirt+I(xirt^2), family="poisson", x=TRUE)
keEGirt<-kequate("EG", 0:4, 0:4, glmXirt, glmYirt, irtx=irtmatX, irty=irtmatY)
getSeeirt(keEGirt)
**getSeelin**

**Retrieve Standard Errors From a Linear Equating**

**Description**

Function to retrieve the standard errors of a linear equating from an object of class keout created by kequate.

**Usage**

```r
getSeelin(object)
```

**Arguments**

- `object`  
  An object of class keout as created by the function kequate.

**Value**

A vector containing the standard errors of the linear equating.

**Author(s)**

- <bjorn.andersson@statistik.uu.se>
- <kenny.branberg@stat.umu.se>
- <marie.wiberg@stat.umu.se>

**See Also**

- `kequate`

**Examples**

```r
P<-c(5, 2, 35, 25, 15)
Q<-c(1, 3, 3, 2, 1)
x<-0:4

glmx<-glm(P~I(x)+I(x^2), family="poisson", x=TRUE)
glmy<-glm(Q~I(x)+I(x^2), family="poisson", x=TRUE)
keEG<-kequate("EG", 0:4, 0:4, glmx, glmy)
getSeelin(keEG)
```
**getDescription**

*Retrieve the Type of Equating*

**Description**

Function to retrieve the type of equating from an object of class keout created by kequate.

**Usage**

```r
getype(object)
```

**Arguments**

- `object` 
  An object of class keout as created by the function kequate.

**Value**

A character vector describing the type of equating conducted.

**Author(s)**

- bjorn.andersson@statistik.uu.se
- kenny.branberg@stat.umu.se
- marie.wiberg@stat.umu.se

**See Also**

- `kequate`

**Examples**

```r
P<-(5, 20, 35, 25, 15)
Q<-(10, 30, 30, 20, 10)
x<-0:4
glmx<-glm(P-I(x)+I(x^2), family="poisson", x=TRUE)
glm<-glm(Q-I(x)+I(x^2), family="poisson", x=TRUE)
keEG<-kequate("EG", 0:4, 0:4, glmx, glmy)
getype(keEG)
```
kefreq  Test Score Frequency Tabulation

Description
Tabulates the frequencies of score values/combinations of score values from data at the individual level. Can handle univariate or bivariate scores.

Usage
kefreq(in1, xscores, in2, ascores)

Arguments
in1 A numeric vector with the total test scores on test X for each individual.
xscores A numeric vector containing the possible score values on tests X and Y.
in2 A numeric vector with the total test scores on test Y or test A for each individual.
ascores A numeric vector containing the possible score values on test A.

Details
Test data often consists of data at the individual level, i.e. there is a data frame, matrix or vector containing the total test score for each test taker along with other possible information about each test taker such as the total anchor test score. In order to use such data in equating, the data needs to be converted into frequencies for each score value or combination of score values. The function kefreq handles both univariate and bivariate data, resulting in a data frame that is ordered correctly for usage with the kequate function to equate two tests. In the case of univariate data, only the first two arguments should be specified. In the case of equating using an SG design, only the first score value vector should be provided since the score values are the same for the two tests. All arguments should be used for a NEAT design.

Value
A data frame containing the score values and the frequencies associated with each score value/score value combination, sorted in the manner appropriate for usage with functions glm and kequate.

Author(s)
<bjorn.andersson@statistik.uu.se>
<kenny.branberg@stat.umu.se>
<marie.wiberg@stat.umu.se>

See Also
table, kequate
Examples

```r
freqdata<-data.frame(X=c(1,2,2,1,2,2,2,2,3,1,2,1,4,2,1,1,3,3,3,3),
                      A=(c(0,2,1,1,0,3,1,2,2,0,3,1,1,2,2,2,1,2)))
Pdata<-kefreq(freqdata$X, 0:5, freqdata$A, 0:3)
```

Description

Information relating to an equating between two tests X and Y.

Objects from the Class

Objects can be created by calls of the form `new("keout", ...)`.

Slots

- `Cr`: Object of class "matrix". The C-matrix from the log-linear model of test X on population P. (EG design only)
- `Cs`: Object of class "matrix". The C-matrix from the log-linear model of test Y on population Q. (EG design only)
- `Cp`: Object of class "matrix". The C-matrix from the log-linear model of tests X and Y or X and A on population P. (CB design: C-matrix from the log-linear model for the first group) (CB/SG/NEAT CE/NEAT PSE/NEC designs only)
- `Cq`: Object of class "matrix". The C-matrix from the log-linear model of tests X and Y or X and A on population Q. (CB design: C-matrix from the log-linear model for the second group) (CB/NEAT CE/NEAT PSE/NEC designs only)
- `SEEvect`: Object of class "SEEvect". Matrices containing the standard error vectors for the equatings. If `linear=TRUE`, then only the standard error vectors for the linear case are included.
- `Pest`: Object of class "matrix". The estimated probability matrix over population P.
- `Pobs`: Object of class "matrix". The observed probability matrix over population P.
- `Qest`: Object of class "matrix". The estimated probability matrix over population Q.
- `Qobs`: Object of class "matrix". The observed probability matrix over population Q.
- `scores`: Object of class "list". A list of data frames containing the score value vectors for the tests X and Y (all designs except NEAT CE) or the score value vectors for the tests X, Y and A (only for NEAT CE). Also included are the estimated score probabilities and the continuized cumulative distribution functions for the respective tests.
- `linear`: Object of class "logical". A logical vector. TRUE if `linear=TRUE` was specified, otherwise FALSE.
- `PRE`: Object of class "data.frame". A data frame containing the percent relative error (PRE) in the ten first moments between the equated scores and the reference distribution. (For chain equating, the PRE is calculated for the linking from X to A and the linking from A to Y.)
Object of class "data.frame". A data frame containing the continuization parameters used in the equating.

kernel: Object of class "character". A character vector denoting the kernel used.

type: Object of class "character". A character vector describing the design used.

equating: Object of class "data.frame". A data frame containing the equated values from X to Y and the associated standard errors (for either an equipercentile or a linear equating), as well as the SEED between the equipercentile and linear equating functions and the equated values and the associated standard errors in the linear case (if an equipercentile equating is conducted).

Methods

plot signature(x = "keout"): ...

summary signature(object = "keout"): ...

Author(s)

<bjorn.andersson@statistik.uu.se>
<kenny.branberg@stat.umu.se>
<marie.wiberg@stat.umu.se>

See Also

kequate

Examples

showClass("keout")

---

**kequate**

*Test Equating Using the Kernel Method*

**Description**

A function to conduct an equating between two parallel tests using kernel equating. Designs available are equivalent groups (EG), single group (SG), counterbalanced (CB), non-equivalent groups with anchor test using either chain equating (NEAT CE) or post-stratification equating (NEAT PSE) and non-equivalent groups using covariates (NEC).

**Usage**

kequate(design, ...)

**Arguments**

design A character vector indicating which design to use. Possible designs are: EG, SG, CB, NEAT_CE, NEAT_PSE or NEC.

... Further arguments which partly depend on the design chosen. (See section Details for further information.)
**Details**

Besides the above argument, additional arguments must be provided for the different equating designs.

**EG design:**
- ‘x’ A vector of possible score values on the test X to be equated, ordered from the lowest to the highest score.
- ‘y’ A vector of possible score values on the test Y to be equated, ordered from the lowest to the highest score.
- ‘r’, ‘s’ Numeric vectors containing the estimated or observed score probabilities for tests X and Y respectively. Alternatively objects of class ‘glm’.
- ‘DMP’, ‘DMQ’ The design matrices from the log-linear models for the estimated score probabilities for X and Y. Not needed if arguments r and s are of class ‘glm’.
- ‘N’, ‘M’ The sample sizes of the groups taking tests X and Y, respectively. Not needed if arguments r and s are of class ‘glm’.
- ‘hx’, ‘hy’ Optional arguments to specify the continuization parameters manually. (If linear=TRUE, then these arguments have no effect.)
- ‘hxlin’, ‘hylin’ Optional arguments to specify the continuization parameters manually in the linear case.
- ‘KPEN’ The constant used in deciding the optimal continuization parameter. Default is 0.
- ‘wpen’ An argument denoting at which point the derivatives in the second part of the penalty function should be evaluated. Default is 1/4.
- ‘linear’ Logical denoting if only a linear equating is to be performed. Default is FALSE.
- ‘irtx’, ‘irty’ Optional arguments to provide matrices of probabilities to answer correctly to the questions on the parallel tests X and Y, as estimated in an Item Response Theory (IRT) model.
- ‘smoothed’ A logical argument denoting if the data provided are pre-smoothed or not. Default is TRUE.
- ‘kernel’ A character vector indicating which kernel to use, either "gaussian", "logistic", "stdgaussian" or "uniform". Default is "gaussian".
- ‘slog’ The parameter used in defining the logistic kernel. Default is 1.
- ‘bunif’ The parameter used in defining the uniform kernel. Default is 0.5.
- ‘altopt’ Logical which sets the bandwidth parameter equal to a variant of Silverman’s rule of thumb. Default is FALSE.

**SG design:**
- ‘x’ A vector of possible score values on the test X to be equated, ordered from the lowest to the highest score.
- ‘y’ A vector of possible score values on the test Y to be equated, ordered from the lowest to the highest score.
- ‘P’ The estimated or observed probability matrix for scores on tests X and Y, where the columns denote scores on test Y and the rows denote scores on test X. Alternatively a vector of score probabilities or an object of class ‘glm’, where the entries are ordered first by the Y-scores and then by the X-scores.
- ‘DM’ The design matrix used in the log-linear model. Not needed if the argument P is of class ‘glm’.

---

**kequate** 23
‘N’ The sample size. Not needed if the argument P is of class ‘glm’.
‘hx’, ‘hy’ Optional arguments to specify the continuization parameter manually. (If linear=TRUE, then these arguments have no effect.)
‘hxlin’, ‘hylin’ Optional arguments to specify the continuization parameters manually in the linear case.
‘KPEN’ The constant used in deciding the optimal continuization parameter. Default is 0.
‘wpen’ An argument denoting at which point the derivatives in the second part of the penalty function should be evaluated. Default is 1/4.
‘linear’ Logical denoting if only a linear equating is to be performed. Default is FALSE.
‘irtx’, ‘irty’ Optional arguments to provide matrices of probabilities to answer correctly to the questions on the parallel tests X and Y, as estimated in an Item Response Theory (IRT) model.
‘smoothed’ A logical argument denoting if the data provided are pre-smoothed or not. Default is TRUE.
‘kernel’ A character vector indicating which kernel to use, either "gaussian", "logistic", "stdgaussian" or "uniform". Default is "gaussian".
‘slog’ The parameter used in defining the logistic kernel. Default is 1.
‘bunif’ The parameter used in defining the uniform kernel. Default is 0.5.

**CB design:**
‘x’ A vector of possible score values on the test X to be equated, ordered from the lowest to the highest score.
‘y’ A vector of possible score values on the test Y to be equated, ordered from the lowest to the highest score.
‘P12’, ‘P21’ The estimated or observed probability matrices for scores on first taking test X and then taking test Y, and first taking test Y and then taking test X respectively, where the rows denote scores on tests X and the columns denote scores on test Y. Alternatively numeric vectors or objects of class ‘glm’, where the entries are ordered first by the Y-scores and then by the X-scores.
‘DM12’, ‘DM21’ The design matrices from the log-linear models for the estimated score probabilities for the two test groups. Not needed if arguments P12 and P21 are of class ‘glm’.
‘N’, ‘M’ The sample sizes for the tests X and A and the tests Y and A, respectively. Not needed if arguments P12 and P21 are of class ‘glm’.
‘hx’, ‘hy’ Optional arguments to specify the continuization parameters manually. (If linear=TRUE, then these arguments have no effect)
‘hxlin’, ‘hylin’ Optional arguments to specify the continuization parameters manually in the linear case. (Applies both when linear=FALSE and when linear=TRUE.)
‘wcb’ The weighting of the two groups. Default is 0.5.
‘KPEN’ Optional argument to specify the constant used in deciding the optimal continuization parameter. Default is 0.
‘wpen’ An argument denoting at which point the derivatives in the second part of the penalty function should be evaluated. Default is 1/4.
‘linear’ Optional logical argument denoting if only a linear equating is to be performed. Default is FALSE.
‘irtx’, ‘irty’ Optional arguments to provide matrices of probabilities to answer correctly to the questions on the parallel tests X and Y, as estimated in an Item Response Theory (IRT) model.
‘smoothed’ A logical argument denoting if the data provided are pre-smoothed or not. Default is TRUE.
'kernel' A character vector indicating which kernel to use, either "gaussian", "logistic", "stdgaussian" or "uniform". Default is "gaussian".
'slog' The parameter used in defining the logistic kernel. Default is 1.
'bunif' The parameter used in defining the uniform kernel. Default is 0.5.
'alltopt' Logical which sets the bandwidth parameter equal to a variant of Silverman’s rule of thumb. Default is FALSE.

NEAT PSE or NEC design:
'x' A vector of possible score values on the test X to be equated, ordered from the lowest to the highest score.
'y' A vector of possible score values on the test Y to be equated, ordered from the lowest to the highest score.
'P', 'Q' The estimated or observed probability matrices for scores on tests X and A and tests Y and A respectively, where the rows denote scores ontests X or Y and the columns denote scores on test A. Alternatively numeric vectors or objects of class ‘glm’, where the entries are ordered first by the X-scores/Y-scores and then by the A-scores.
'DMP', 'DMQ' The design matrices from the log-linear models for the estimated score probabilities for X and A and Y and A. Not needed if arguments P and Q are of class ‘glm’.
'N', 'M' The sample sizes for the tests X and A and the tests Y and A, respectively. Not needed if arguments P and Q are of class ‘glm’.
'w' The weighting of the synthetic population. Default is 0.5.
'hx', 'hy' Optional arguments to specify the continuization parameters manually. (If linear=TRUE, then these arguments have no effect)
'hxlin', 'hylin' Optional arguments to specify the continuization parameters manually in the linear case. (Applies both when linear=FALSE and when linear=TRUE.)
'KPEN' Optional argument to specify the constant used in deciding the optimal continuization parameter. Default is 0.
'wpen' An argument denoting at which point the derivatives in the second part of the penalty function should be evaluated. Default is 1/4.
'linear' Optional logical argument denoting if only a linear equating is to be performed. Default is FALSE.
'irtx', 'irty' Optional arguments to provide matrices of probabilities to answer correctly to the questions on the parallel tests X and Y, as estimated in an Item Response Theory (IRT) model.
'smoothed' A logical argument denoting if the data provided are pre-smoothed or not. Default is TRUE.
'kernel' A character vector indicating which kernel to use, either "gaussian", "logistic", "stdgaussian" or "uniform". Default is "gaussian".
'slog' The parameter used in defining the logistic kernel. Default is 1.
'bunif' The parameter used in defining the uniform kernel. Default is 0.5.
'alltopt' Logical which sets the bandwidth parameter equal to a variant of Silverman’s rule of thumb. Default is FALSE.

NEAT CE design:
'x' A vector of possible score values on the test X to be equated, ordered from the lowest to the highest score.
‘y’ A vector of possible score values on the test Y to be equated, ordered from the lowest to the highest score.

‘a’ A vector containing the possible score values on the anchor test, ordered from the lowest score to the highest.

‘P’, ‘Q’ The estimated or observed probability matrices for scores on tests X and A and tests Y and A respectively, where the rows denote scores on test X or Y and the columns denote scores on test A. Alternatively numeric vectors or objects of class ‘glm’, where the entries are ordered first by the X-scores/Y-scores and then by the A-scores.

‘DMP’, ‘DMQ’ The design matrices from the log-linear models for the estimated score probabilities for X and A and Y and A, respectively. Not needed if arguments P and Q are of class ‘glm’.

‘N’, ‘M’ The sample sizes for the tests X and A and the tests Y and A, respectively. Not needed if arguments P and Q are of class ‘glm’.

‘hxP’, ‘hyQ’, ‘haP’, ‘haQ’ Optional arguments to specify the continuization parameters manually. (If linear=TRUE, then these arguments have no effect.)

‘hxPlin’, ‘hyQlin’, ‘haPlin’, ‘haQlin’ Optional arguments to specify the continuization parameters manually in the linear case. (Applies both when linear=FALSE and when linear=TRUE.)

‘KPEN’ Optional argument to specify the constant used in deciding the optimal continuization parameter. Default is 0.

‘wpen’ An argument denoting at which point the derivatives in the second part of the penalty function should be evaluated. Default is 1/4.

‘linear’ Optional logical argument denoting if only a linear equating is to be performed. Default is FALSE.

‘irtx’, ‘irty’ Optional arguments to provide matrices of probabilities to answer correctly to the questions on the parallel tests X and Y, as estimated in an Item Response Theory (IRT) model.

‘smoothed’ A logical argument denoting if the data provided are pre-smoothed or not. Default is TRUE.

‘kernel’ A character vector indicating which kernel to use, either "gaussian", "logistic", "stdgaussian" or "uniform". Default is "gaussian".

‘slog’ The parameter used in defining the logistic kernel. Default is 1.

‘bunif’ The parameter used in defining the uniform kernel. Default is 0.5.

‘altopt’ Logical which sets the bandwidth parameter equal to a variant of Silverman’s rule of thumb. Default is FALSE.

**Value**

Kequate returns an S4 object of class ‘keout’ which includes the following slots (accessed by using the get functions):

- **Cr** The C-matrix from the log-linear model of test X on population P. (EG design only)

- **Cs** The C-matrix from the log-linear model of test Y on population Q. (EG design only)

- **Cp** The C-matrix from the log-linear model of tests X and Y or X and A on population P. (SG/NEAT CE/NEAT PSE/NEC designs only)
The C-matrix from the log-linear model of tests X and Y or X and A on population Q. (NEAT CE/NEAT PSE/NEC designs only)

An object of class SEEvect consisting of matrices containing the standard error vectors for the equatings. If linear=TRUE, then only the standard error vectors for the linear case are included.

The estimated probability matrix over population P.

The observed probability matrix over population P.

The estimated probability matrix over population Q.

The observed probability matrix over population Q.

A list containing the score vectors for the tests to be equated and, in a NEAT CE design, the score vector of the anchor test. Also included are the estimated score probabilities and the continuized cumulative distribution functions for the respective tests.

A logical vector. TRUE if linear=TRUE was specified, otherwise FALSE.

A data frame containing the percent relative error in the ten first moments between the equated scores and the reference distribution. (For chain equating, the PRE is calculated for the linking from X to A and the linking from A to Y.)

A data frame containing the continuization parameters used in the equating.

A character vector denoting the kernel used.

A character vector describing the design used.

A data frame containing the equated values from X to Y and the associated standard errors (for either an equipercentile or a linear equating), as well as the SEED between the equipercentile and linear equating functions and the equated values and the associated standard errors in the linear case (if an equipercentile equating is conducted).

Author(s)

<bjorn.andersson@statistik.uu.se>
<kenny.branberg@stat.umu.se>
<marie.wiberg@stat.umu.se>

References


See Also

glm,kefreq
Examples

# EG toy example with different kernels
P<-(5, 20, 35, 25, 15)
Q<-(10, 30, 20, 10)
x<:0:4
glmP<-glm(P~I(x)+I(x^2), family="poisson", x=TRUE)
glmQ<-glm(Q~I(x)+I(x^2), family="poisson", x=TRUE)
keEG<-kequate("EG", 0:4, 0:4, glmP, glmQ)
keEGlog<-kequate("EG", 0:4, 0:4, glmP, glmQ, kernel="logistic", slog=sqrt(3)/pi)
keEGuni<-kequate("EG", 0:4, 0:4, glmP, glmQ, kernel="uniform", bunif=sqrt(3))
plot(keEG)

# NEAT example using simulated data
data(simeq)
freq1 <- kefreq(simeq$bivar1$X, 0:2, simeq$bivar1$A, 0:1)
freq2 <- kefreq(simeq$bivar2$Y, 0:2, simeq$bivar2$A, 0:1)
glm1<-glm(frequency~I(X)+I(X^2)+I(A)+I(A^2):I(X)+I(A):I(X^2)+I(A^2):I(X^2), family="poisson", data=freq1, x=TRUE)
glm2<-glm(frequency~I(X)+I(X^2)+I(A)+I(A^2):I(X)+I(A):I(X^2)+I(A^2):I(X^2), family="poisson", data=freq2, x=TRUE)
keNEATPSE <- kequate("NEAT_PSE", 0:2, 0:2, glm1, glm2)
keNEATCE <- kequate("NEAT_CE", 0:2, 0:2, 0:1, glm1, glm2)
summary(keNEATPSE)
summary(keNEATCE)

# IRT observed-score equating
keNEATCEirt <- kequate("NEAT_CE", 0:2, 0:2, 0:1, glm1, glm2, irtx=simeq$irtNEATx, irty=simeq$irtNEATy)
getEquating(keNEATCEirt)

plot-methods

Methods
signature(x = "cdist") Plots the conditional means and variances of an observed and an estimated bivariate test score distribution.
signature(x = "genseed") Plots the difference between two equating functions with +/- 2*SEED boundaries for both equating from X to Y and from Y to X.
signature(x = "keout") Plots the equated values from the equating against the score values of the equated test.
Description
Calculates the percent relative error (PRE) between an equated distribution and the reference distribution for the first ten moments.

Usage
PREp(eq, obs, r, s)

Arguments
- `eq`: A numeric vector containing the equated values from X to Y or Y to X.
- `obs`: The score vector of test Y or X.
- `r`: A vector of probabilities corresponding to the equated values.
- `s`: A vector of probabilities corresponding to the score values of test Y or X.

Details
If we equate test X to test Y, then we have equated values eYx with estimated probabilities r and estimated probabilities s for the score values on Y. To compare the moments between these two distributions, we can calculate the percent relative error (PRE) between them. If we denote the p:th moment of Y and eYx by $\mu(Y)$ and $\mu(eYx)$ respectively, the PRE for moment p is defined as

$$PRE(p) = 100\frac{\mu(eYx) - \mu(Y)}{\mu(Y)}.$$

Value
A numeric vector containing the percentage relative error for the first ten moments.

Author(s)
- bjorn.andersson@statistik.uu.se
- kenny.branberg@stat.umu.se
- marie.wiberg@stat.umu.se

References

See Also
- glm kequate
### Examples

```r
P <- c(5, 2, 35, 25, 15)
Q <- c(10, 30, 30, 20, 10)
x <- 4

glmX <- glm(P - 1(x) + I(x^2), family = "poisson", x = TRUE)
glmY <- glm(Q - 1(x) + I(x^2), family = "poisson", x = TRUE)
keEG <- kequate("EG", 0:4, glmX, glmY)
PREp(getEq(keEG), 0:4, glmX$fitted.values / 100, glmY$fitted.values / 100)
```

---

### simeq

*Simulated Test Data*

Contains bivariate test score data and IRT data to be used in equating.

**Usage**

```r
data(simeq)
```

**Format**

A list containing two data frames (bivar1 and bivar2) with 1000 observations each and two 5x20 matrices (irtNEATx and irtNEATy).

**Source**

The data was generated in R.

---

### summary-methods

~~ Methods for Function summary ~~

**Description**

~~ Methods for function summary ~~

**Methods**

```r
signature(object = "ANY")
signature(object = "keout")
signature(object = "sparseMatrix")
```
Index

*Topic classes
  cdist-class, 4
  genseed-class, 7
  keout-class, 21

*Topic datasets
  simeq, 30

*Topic methods
  plot-methods, 28
  summary-methods, 30

*Topic package
  kequate-package, 2


cdist, 3, 5
cdist-class, 4

FTres, 5

genseed, 6, 16
genseed-class, 7
getEq, 8
getEq,keout-method (getEq), 8
getEq-class (getEq), 8
getEqirt, 9
getEqirt,keout-method (getEqirt), 9
getEqirt-class (getEqirt), 9
getEqlin, 10
getEqlin,keout-method (getEqlin), 10
getEqlin-class (getEqlin), 10
getEquating, 11
getEquating,keout-method (getEquating), 11
getEquating-class (getEquating), 11
getH, 12
getH,keout-method (getH), 12
getH-class (getH), 12
getPre, 13
getPre,keout-method (getPre), 13
getPre-class (getPre), 13
getScores, 14
getScores,keout-method (getScores), 14
getScores-class (getScores), 14
getSee, 15
getSee,keout-method (getSee), 15
getSee-class (getSee), 15
getSeed, 16
getSeed,keout-method (getSeed), 16
getSeed-class (getSeed), 16
getSeeirt, 17
getSeeirt,keout-method (getSeeirt), 17
getSeeirt-class (getSeeirt), 17
getSeelin, 18
getSeelin,keout-method (getSeelin), 18
getSeelin-class (getSeelin), 18
type, 19
type,keout-method (type), 19
type-class (type), 19

glm, 6, 27, 29

kefreq, 20, 27
keout-class, 21
kequate, 4, 7–20, 22, 22, 29
kequate-package, 2
kequateCB (kequate), 22
kequateEG (kequate), 22
kequateNEAT_CE (kequate), 22
kequateNEAT_PSE (kequate), 22
kequateSG (kequate), 22

plot,cdist-method (plot-methods), 28
plot,genseed-method (plot-methods), 28
plot,keout-method (plot-methods), 28
plot-methods, 28
PREp, 4, 8, 29

simeq, 30
summary,ANY-method (summary-methods), 30
summary,keout-method (summary-methods), 30
summary,sparseMatrix-method (summary-methods), 30
summary-methods, 30

table, 20