Package 'kequate'

October 13, 2022

Encoding UTF-8

Type Package

Title The Kernel Method of Test Equating

Version 1.6.4

Date 2022-04-11

Description Implements the kernel method of test equating as defined in von Davier, A. A., Holland, P. W. and Thayer, D. T. (2004) <doi:10.1007/b97446> and Andersson, B. and Wiberg, M. (2017) <doi:10.1007/s11336-016-9528-7> using the CB, EG, SG, NEAT CE/PSE and NEC designs, supporting Gaussian, logistic and uniform kernels and unsmoothed and pre-smoothed input data.

License GPL-2 | GPL-3

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

Imports equateIRT, mirt, utils

LazyData true

NeedsCompilation no

Author Björn Andersson [aut, cre] (<https://orcid.org/0000-0002-9007-2440>),
Kenny Bränberg [aut],
Marie Wiberg [aut] (<https://orcid.org/0000-0001-5549-8262>)

Maintainer Björn Andersson <bjoern.h.andersson@gmail.com>

Repository CRAN

Date/Publication 2022-04-13 07:52:30 UTC

R topics documented:

kequate-package 2
cdist 3
cdist-class 5
FTres 6
genseed 7
genseed-class 8
getEq 9

getEqirt 10
getEqlin 11
getEquating 12
getH 13
getPre 14
getScores 15
getSee 16
getSeed 17
getSeeirt 18
getSeelin 19
getType 20
irtose 21
kefreq 24
keout-class 26
kequate 27
plot-methods 34
PREp 34
simeq 35
summary-methods 36

Index **37**

`kequate-package` *The Kernel Method of Test Equating*

Description

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.

Details

Package:	kequate
Type:	Package
Version:	1.6.4
Date:	2022-04-11
License:	GPL-3
LazyLoad:	yes
Depends:	R (>=2.11.0), methods, graphics, stats, ltm

Author(s)

Björn Andersson, Kenny Branberg and Marie Wiberg

Maintainer: Björn Andersson <bjoern.h.andersson@gmail.com>

References

Andersson, B. and Wiberg, M. (2017). Item Response Theory Observed-Score Kernel Equating. *Psychometrika*, **83**, 48–67. <doi:10.1007/s11336-016-9528-7>

Andersson, B., Branberg, K., and Wiberg, M. (2013). Performing the Kernel Method of Test Equating with the Package **kequate**. *Journal of Statistical Software*, **55(6)**, 1–25. <doi:10.18637/jss.v055.i06>

Branberg, K. and Wiberg, M. (2011). Observed Score Linear Equating with Covariates. *Journal of Educational Measurement*, **44(4)**, 419–440. <doi:10.1111/j.1745-3984.2011.00153.x>

von Davier, A.A., Holland, P.W., and Thayer, D.T. (2004). *The Kernel Method of Test Equating.* Springer-Verlag New York.

Holland, P.W. and Thayer, D. (1998). Univariate and Bivariate Loglinear Models for Discrete Test Score Distributions *ETS Technical Report No* **98-1**.

---

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

von Davier, A.A., Holland, P.W., Thayer, D.T. (2004). *The Kernel Method of Test Equating.* Springer-Verlag New York.

Holland, P.W., Thayer, D. (1998). Univariate and Bivariate Loglinear Models for Discrete Test Score Distributions *ETS Technical Report No* **98-1**.

See Also

`kequate` `PREp`

Examples

```
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(0,2,1,1,0,3,1,2,2,0,2,0,3,1,1,2,2,2,1,2)))
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.

obs2 Matrix of conditional means, variances, skewnesses and kurtoses of A given X for the observed score distribution.

Methods

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

Author(s)

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

See Also

cdist

Examples

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

FTres *Freeman-Tukey Residuals*

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

```
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

Andersson, B., Branberg, K., and Wiberg, M. (2013). Performing the Kernel Method of Test Equating with the Package **kequate**. *Journal of Statistical Software*, **55(6)**, 1–25. <doi:10.18637/jss.v055.i06>

Holland, P.W, Thayer, D. (1998). Univariate and Bivariate Loglinear Models for Discrete Test Score Distributions *ETS Technical Report No* **98-1**.

See Also

glm

Examples

```
#Example data:
P<-c(5, 20, 35, 25, 15)
x<-0:4
glmx<-glm(P~I(x)+I(x^2), family="poisson", x=TRUE)
res<-FTres(glmx$y, glmx$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

```
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

Andersson, B., Branberg, K., and Wiberg, M. (2013). Performing the Kernel Method of Test Equating with the Package **kequate**. *Journal of Statistical Software*, **55(6)**, 1–25. <doi:10.18637/jss.v055.i06>

von Davier, A.A., Holland, P.W., Thayer, D.T. (2004). *The Kernel Method of Test Equating.* Springer-Verlag New York.

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)
keEGnew<-kequate("EG", 0:4, 0:4, glmx, glmy, hx=0.33, hy=0.33)
SEEDEG<-genseed(keEG, keEGnew)
plot(SEEDEG)
```

genseed-class *Class "genseed"*

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`

Examples

```
showClass("genseed")
```

---

getEq *Retrieve Equated Values*

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`

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)
```

```
irtX<-c(12, 32, 54, 22, 10)
irtY<-c(24, 36, 48, 12, 10)
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)
getEqirt(keEGirt)
```

getEqlin *Retrieve 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 valuesof the linear 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)
getEqlin(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

```
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)
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, 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)
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, 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)
getScores(keEG)
```

getSee *Retrieve Standard Errors From an Equating*

Description

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

Usage

```
getSee(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)
getSee(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

```
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 Standard Errors of Equating 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, 10)
irtY<-c(24, 36, 48, 12, 10)
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

```
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

```
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)
getSeelin(keEG)
```

getType *Retrieve the Type of Equating*

Description

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

Usage

```
getType(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

```
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)
getType(keEG)
```

`irtose` *Item Response Theory Observed-Score Kernel Equating*

Description

A function to conduct an equating between two parallel tests using item response theory (IRT) observed-score kernel equating. Designs available are equivalent groups (EG) and non-equivalent groups with anchor test using chain equating (NEAT CE).

Usage

```
irtose(design="CE", P, Q, x, y, a=0, qpoints=seq(-6, 6, by=0.1), model="2pl", catsX=0,
catsY=0, catsA=0, see="analytical", replications=199, kernel="gaussian",
h=list(hx=0, hy=0, hxP=0, haP=0, hyQ=0, haQ=0),
hlin=list(hxlin=0, hylin=0, hxPlin=0, haPlin=0, hyQlin=0, haQlin=0), KPEN=0, wpen=0.5,
linear=FALSE, slog=1, bunif=1, altopt=FALSE, wS=0.5, eqcoef="mean-mean", robust=FALSE,
distribution = list("normal", par = data.frame(mu = 0, sigma = 1)), DS = FALSE,
CV = FALSE)
```

Arguments

`design` A character vector indicating which design to use. Possible designs are EG, CE and PSE.

`P` Object of class "matrix" or an object created by the R package ltm containing either the responses for each question in group P or the estimated IRT model in group P.

`Q` Object of class "matrix" or an object created by the R package ltm containing either the responses for each question in group Q or the estimated IRT model in group Q.

`x` Vector of possible score values of the main test X to be equated.

`y` Vector of possible score values of the main test Y to be equated.

`a` Vector of possible score values of the anchor test A. (Required for all NEAT designs.)

`qpoints` A numeric vector containing the quadrature points used in the equating. If not specified, the quadrature points from the IRT models will be used.

`model` A character vector indicating which IRT model to use. Available models for dichotomous data are 2pl and 3pl while available models for polytomous data are GPCM and GRM. Default is "2pl".

`catsX` The number of category answers for each item on test X. (GPCM and GRM only.)

`catsY` The number of category answers for each item on test Y. (GPCM and GRM only.)

`catsA` The number of category answers for each item on test A. (GPCM and GRM only.)

`see` A character vector indicating which standard errors of equating to use. Options are "analytical" and "bootstrap", with default "analytical". If using the double smoothing bandwidth selection method (DS = TRUE), the bootstrap is not supported.

`replications` The number of bootstrap replications if using the bootstrap standard error calculations. Default is 199.

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

`h` Optional argument to specify the continuization parameters manually as a list with suitable bandwidth parameters. In an EG design or a NEAT PSE design: hx and hy, in a NEAT CE design: hxP, haP, hyQ and haQ. (If linear=TRUE, then these arguments have no effect.)

`hlin` Optional argument to specify the linear continuization parameters manually as a list with suitable bandwidth parameters. In an EG design or a NEAT PSE design: hxlin and hylin, in a NEAT CE design: hxPlin, haPlin, hyQlin and haQlin.

`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.

`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.

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

`wS` The weighting of group P in a PSE equating. Default is 0.5.

`eqcoef` Character vector indicating which equating coefficients to use. Options are "mean-mean", "mean-gmean", "mean-sigma", "Stocking-Lord" and "Haebara". Default is "mean-mean". (Only for PSE. With polytomous IRT models, only "mean-mean" is available.)

`robust` Logical indicating whether the robust covariance matrix should be calculated for the IRT model. Warning: is slow for large models and large sample sizes. Default is FALSE. (Only for 2pl and GPCM models.)

`distribution` A list containing information regarding the latent distribution. Default is the standard normal distribution. (Currently not in use.)

`DS` Logical which enables bandwidth selection with the double smoothing method.

`CV` Logical which enables bandwidth selection with the cross-validation method.

Value

An S4 object of class 'keout' which includes the following slots (accessed by using the get functions):

`coveqYx` The asymptotic covariance matrix of the equating function.

`Pest` The estimated probability matrix over population P.

`Pobs` The observed probability matrix over population P.

`Qest` The estimated probability matrix over population Q.

`Qobs` The observed probability matrix over population Q.

`scores` 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.

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

`PRE` 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.)

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

`kernel` A character vector denoting the kernel used.

`type` A character vector describing the design used.

`equating` 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).

`see` A character vector denoting which standard errors of equating are used.

`replications` A numeric vector denoting the number of replications used when calculating the bootstrap standard errors of equating.

`irt` A list containing the objects from the package ltm and the adjusted asymptotic covariance matrices of the item parameters.

Author(s)

`<bjoern.h.andersson@gmail.com>`
`<kenny.branberg@stat.umu.se>`
`<marie.wiberg@stat.umu.se>`

References

Andersson, B. (2016). Asymptotic Standard Errors of Observed-Score Equating With Polytomous IRT Models. *Journal of Educational Measurement*, **53**, 459–477.

Andersson, B. (2018). Asymptotic Variance of Linking Coefficient Estimators for Polytomous IRT Models. *Applied Psychological Measurement*, **42**, 192–205.

Andersson, B. and Wiberg, M. (2017). Item Response Theory Observed-Score Kernel Equating. *Psychometrika*, **83**, 48–67.

von Davier, A.A., Holland, P.W., and Thayer, D.T. (2004). *The Kernel Method of Test Equating.* Springer-Verlag New York.

Ogasawara, H. (2003). Asymptotic standard errors of IRT observed-score equating methods. *Psychometrika*, **68**, 193-211.

See Also

kequate

Examples

```
## Not run:
#Generate data, non-equivalent groups with chain equating design
akX <- runif(15, 0.5, 2)
bkX <- rnorm(15)
akY <- runif(15, 0.5, 2)
bkY <- rnorm(15)
akA <- runif(15, 0.5, 2)
bkA <- rnorm(15)

dataP <- matrix(0, nrow = 1000, ncol = 30)
dataQ <- matrix(0, nrow = 1000, ncol = 30)

for(i in 1:1000){
ability <- rnorm(1)
dataP[i,1:15] <- (1 / (1 + exp(-akX * (ability - bkX)))) > runif(15)
dataP[i,16:30] <- (1 / (1 + exp(-akA * (ability - bkA)))) > runif(15)
}

for(i in 1:1000){
ability <- rnorm(1, mean = 0.5)
dataQ[i,1:15] <- (1 / (1 + exp(-akY * (ability - bkY)))) > runif(15)
dataQ[i,16:30] <- (1 / (1 + exp(-akA * (ability - bkA)))) > runif(15)
}
#Estimate the 2-PL IRT model with package ltm
sim2plP <- ltm(dataP ~ z1, IRT.param = TRUE)
sim2plQ <- ltm(dataQ ~ z1, IRT.param = TRUE)

#Equate the two tests
sim2plan <- irtose("CE", sim2plP, sim2plQ, 0:15, 0:15, 0:15)

## End(Not run)
```

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>`

References

Andersson, B., Branberg, K., and Wiberg, M. (2013). Performing the Kernel Method of Test Equating with the Package **kequate**. *Journal of Statistical Software*, **55(6)**, 1–25. <doi:10.18637/jss.v055.i06>

See Also

`table`, `kequate`

Examples

```
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(0,2,1,1,0,3,1,2,2,0,2,0,3,1,1,2,2,2,1,2)))
Pdata <- kefreq(freqdata$X, 0:5, freqdata$A, 0:3)
```

`keout-class` *Class "keout"*

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)

`coveqYx`: Object of class `"matrix"`. Matrix containing the covariance matrix of the equating function. (IRT-OSE 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.

`pdereqYx`: Object of class `"matrix"`. A matrix with the partial derivative vectors for the equating function.

`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.)

`h`**:** 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).

`irt`**:** Object of class `"list"`. A list containing the objects from the package ltm containing the IRT models specified, and the asymptotic covariance matrix of the item parameters under the regular IRT parametrization. (IRT-OSE only)

`see`**:** Object of class `"character"`. A character vector denoting which type of standard errors of equating that are used: analytical or bootstrap.

`replications`**:** Object of class `"numeric"`. The number of bootstrap replications if using the bootstrap standard error calculations.

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.

‘DS’ Logical which enables bandwidth selection with the double smoothing method.

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'.

'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.

'DS' Logical which enables bandwidth selection with the double smoothing method.

'CV' Logical which enables bandwidth selection with the cross-validation method.

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.

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

‘DS’ Logical which enables bandwidth selection with the double smoothing method.

‘CV’ Logical which enables bandwidth selection with the cross-validation method.

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 on tests 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.

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

‘DS’ Logical which enables bandwidth selection with the double smoothing method.

‘CV’ Logical which enables bandwidth selection with the cross-validation method.

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.

‘DS’ Logical which enables bandwidth selection with the double smoothing method.

‘CV’ Logical which enables bandwidth selection with the cross-validation method.

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)

`Cq` 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)

`SEEvect` 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.

`Pest` The estimated probability matrix over population P.

`Pobs` The observed probability matrix over population P.

`Qest` The estimated probability matrix over population Q.

`Qobs` The observed probability matrix over population Q.

`scores` 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.

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

`PRE` 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.)

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

`kernel` A character vector denoting the kernel used.

`type` A character vector describing the design used.

`equating` 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

Andersson, B., Branberg, K., and Wiberg, M. (2013). Performing the Kernel Method of Test Equating with the Package **kequate**. *Journal of Statistical Software*, **55(6)**, 1–25. <doi:10.18637/jss.v055.i06>

von Davier, A.A., Holland, P.W., Thayer, D.T. (2004). *The Kernel Method of Test Equating.* Springer-Verlag New York.

See Also

glm,kefreq

Examples

```
#EG toy example with different kernels
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)
keEGlog<-kequate("EG", 0:4, 0:4, glmx, glmy, kernel="logistic", slog=sqrt(3)/pi)
keEGuni<-kequate("EG", 0:4, 0:4, glmx, glmy, kernel="uniform", bunif=sqrt(3))
plot(keEG)

## Not run:
#NEAT example using simulated data
data(simeq)
freq1 <- kefreq(simeq$bivar1$X, 0:20, simeq$bivar1$A, 0:10)
freq2 <- kefreq(simeq$bivar2$Y, 0:20, simeq$bivar2$A, 0:10)
glm1<-glm(frequency~I(X)+I(X^2)+I(X^3)+I(X^4)+I(X^5)+I(A)+I(A^2)+I(A^3)+I(A^4)+
I(A):I(X)+I(A):I(X^2)+I(A^2):I(X)+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(A^3)+I(A^4)+I(A):I(X)+I(A):I(X^2)+
I(A^2):I(X)+I(A^2):I(X^2), family="poisson", data=freq2, x=TRUE)
keNEATPSE <- kequate("NEAT_PSE", 0:20, 0:20, glm1, glm2)
keNEATCE <- kequate("NEAT_CE", 0:20, 0:20, 0:10, glm1, glm2)
summary(keNEATPSE)
summary(keNEATCE)

#IRT observed-score equating
keNEATCEirt <- kequate("NEAT_CE", 0:20, 0:20, 0:10, glm1, glm2, irtx=simeq$irtNEATx,
irty=simeq$irtNEATy)
getEquating(keNEATCEirt)

## End(Not run)
```

plot-methods ~~ *Methods for Function plot* ~~

Description

~~ Methods for function `plot` ~~

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.

PREp *Percent Relative Error*

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

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

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

Andersson, B., Branberg, K., Wiberg, M. (2013). Performing the Kernel Method of Test Equating with the Package **kequate**. *Journal of Statistical Software*, **55(6)**, 1–25. <doi: 10.18637/jss.v055.i06>

von Davier, A.A., Holland, P.W., Thayer, D.T. (2004). *The Kernel Method of Test Equating*. Springer-Verlag New York.

See Also

glm 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)
PREp(getEq(keEG), 0:4, glmx$fitted.values/100, glmy$fitted.values/100)
```

simeq *Simulated Test Data*

Description

Contains bivariate test score data (bivar1 and bivar2), IRT data (irtNEATx and irtNEATy), data from an equivalent groups (EG) design (FXEGglm and FYEGglm) and data from a non-equivalent groups with covariates (NEC) design (testdata1 and testdata2), for usage in equating.

Usage

```
data(simeq)
```

Format

A list containing the data frames bivar1 and bivar2 with 1000 observations each, two 5x20 matrices irtNEATx and irtNEATy, two glm objects FXEGglm and FYEGglm and two data frames testdata1 and testdata2.

Source

The EG data is taken from the log-linear models specified in von Davier (2004). The remaining data was generated in R.

References

Branberg, K. and Wiberg, M. (2011). Observed Score Linear Equating with Covariates. *Journal of Educational Measurement*, **44(4)**, 419–440. <doi:10.1111/j.1745-3984.2011.00153.x>

von Davier, A.A., Holland, P.W., Thayer, D.T. (2004). *The Kernel Method of Test Equating.* Springer-Verlag New York.

`summary-methods` *~~ Methods for Function summary ~~*

Description

~~ Methods for function `summary` ~~

Methods

`signature(object = "ANY")`

`signature(object = "keout")`

`signature(object = "sparseMatrix")`

Index

∗ **classes**
  cdist-class, 5
  genseed-class, 8
  keout-class, 26
∗ **datasets**
  simeq, 35
∗ **methods**
  plot-methods, 34
  summary-methods, 36
∗ **package**
  kequate-package, 2

cdist, 3, *5*
cdist-class, 5

FTres, 6

genseed, 7, *17*
genseed-class, 8
getEq, 9
getEq,keout-method *(getEq)*, 9
getEq-class *(getEq)*, 9
getEqirt, 10
getEqirt,keout-method *(getEqirt)*, 10
getEqirt-class *(getEqirt)*, 10
getEqlin, 11
getEqlin,keout-method *(getEqlin)*, 11
getEqlin-class *(getEqlin)*, 11
getEquating, 12
getEquating,keout-method *(getEquating)*, 12
getEquating-class *(getEquating)*, 12
getH, 13
getH,keout-method *(getH)*, 13
getH-class *(getH)*, 13
getPre, 14
getPre,keout-method *(getPre)*, 14
getPre-class *(getPre)*, 14
getScores, 15
getScores,keout-method *(getScores)*, 15
getScores-class *(getScores)*, 15
getSee, 16
getSee,keout-method *(getSee)*, 16
getSee-class *(getSee)*, 16
getSeed, 17
getSeed,keout-method *(getSeed)*, 17
getSeed-class *(getSeed)*, 17
getSeeirt, 18
getSeeirt,keout-method *(getSeeirt)*, 18
getSeeirt-class *(getSeeirt)*, 18
getSeelin, 19
getSeelin,keout-method *(getSeelin)*, 19
getSeelin-class *(getSeelin)*, 19
getType, 20
getType,keout-method *(getType)*, 20
getType-class *(getType)*, 20
glm, *6*, *33*, *35*

irtose, 21

kefreq, 24, *33*
keout-class, 26
kequate, *4*, *8–20*, *24*, *25*, *27*, 27, *35*
kequate-package, 2
kequateCB *(kequate)*, 27
kequateEG *(kequate)*, 27
kequateNEAT_CE *(kequate)*, 27
kequateNEAT_PSE *(kequate)*, 27
kequateSG *(kequate)*, 27

plot,ANY,ANY-method *(plot-methods)*, 34
plot,cdist,ANY-method *(plot-methods)*, 34
plot,genseed,ANY-method *(plot-methods)*, 34
plot,keout,ANY-method *(plot-methods)*, 34
plot-methods, 34
PREp, *4*, *9*, 34

simeq, 35
summary,ANY-method *(summary-methods)*, 36

summary,keout-method *(*summary-methods*)*, 36
summary,sparseMatrix-method *(*summary-methods*)*, 36
summary-methods, 36

table, *25*