Package: TestGardener (via r-universe)

September 16, 2024

Version 3.3.3

Date 2024-03-20

Type Package

Title Information Analysis for Test and Rating Scale Data

Author James Ramsay [aut, cre], Juan Li [ctb], Marie Wiberg [ctb], Joakim Wallmark [ctb], Spencer Graves [ctb]

Maintainer James Ramsay <james.ramsay@mcgill.ca>

Depends R (>= 3.5), fda, rgl, ggplot2, plotly

Description Develop, evaluate, and score multiple choice examinations, psychological scales, questionnaires, and similar types of data involving sequences of choices among one or more sets of answers. This version of the package should be considered as brand new. Almost all of the functions have been changed, including their argument list. See the file NEWS.Rd in the Inst folder for more information. Using the package does not require any formal statistical knowledge beyond what would be provided by a first course in statistics in a social science department. There the user would encounter the concept of probability and how it is used to model data and make decisions, and would become familiar with basic mathematical and statistical notation. Most of the output is in graphical form.

License GPL (>= 2)

Imports dplyr, ggpubr, stringr, tidyr, pracma, utf8, knitr, rmarkdown,

LazyData true

NeedsCompilation no

Repository https://jamesramsay5.r-universe.dev

RemoteUrl https://github.com/jamesramsay5/testgardener

RemoteRef HEAD

RemoteSha 321775349d843ec2a4ffbdd185d292640395915c

Contents

Analyze . . . 3
chcemat_simulate . . . 6
dataSimulation . . . 7
density_plot . . . 8
DFfun . . . 10
entropies . . . 11
Entropy_plot . . . 12
eval.surp . . . 14
Fcurve . . . 15
Ffun . . . 17
Ffuns_plot . . . 18
ICC . . . 19
ICC_plot . . . 20
index2info . . . 23
index_distn . . . 25
index_fun . . . 26
index_search . . . 28
make_dataList . . . 30
mu . . . 33
mu_plot . . . 34
Power_plot . . . 35
Quant_13B_problem_chcemat . . . 37
Quant_13B_problem_dataList . . . 37
Quant_13B_problem_infoList . . . 38
Quant_13B_problem_key . . . 39
Quant_13B_problem_parmList . . . 40
Sbinsmth . . . 41
Sbinsmth_nom . . . 43
Scope_plot . . . 44
scoreDensity . . . 45
scorePerformance . . . 46
Sensitivity_plot . . . 48
SimulateData . . . 50
smooth.ICC . . . 51
smooth.surp . . . 52
Spca . . . 54
Spca_plot . . . 55
TestGardener . . . 57
TestInfo_svd . . . 57
TG_analysis . . . 58
TG_density.fd . . . 61

Index **64**

Analyze *Analyze test or rating scale data defined in* dataList*.*

Description

The test or rating scale data have already been processed by function make_dataList or or other code to produce the list object dataList. The user defines a list vector ParameterList which stores results from a set of cycles of estimating surprisal curves followed by estimating optimal score index values for each examinee or respondent. These score index values are within the interval [0,100]. The number of analysis cycles is the length of the parmList list vector.

Usage

```
Analyze(index, indexQnt, dataList, NumDensBasis=7, ncycle=10, itdisp=FALSE,
        verbose=FALSE)
```

Arguments

index — A vector of N score index values for the examinees or respondents. These values are in the percent interval [0,100].

indexQnt — A vector of length 2*nbin + 1 where nbin is the number of bins containing score index values. The vector begins with the lower boundary 0 and ends with the upper boundary 100. In between it alternates between the bin center value and the boundary separating the next bin.

dataList — A list that contains the objects needed to analyse the test or rating scale with the following fields:

- **chce:** A matrix of response data with N rows and n columns where N is the number of examinees or respondents and n is the number of items. Entries in the matrices are the indices of the options chosen. Column i of chce is expected to contain only the integers 1,...,noption.
- **optList:** A list vector containing the numerical score values assigned to the options for this question.
- **key:** If the data are from a test of the multiple choices type where the right answer is scored 1 and the wrong answers 0, this is a numeric vector of length n containing the indices the right answers. Otherwise, it is NULL.
- **Sfd:** An fd object for the defining the surprisal curves.
- **noption:** A numeric vector of length n containing the numbers of options for each item.
- **nbin:** The number of bins for binning the data.
- **scrrng:** A vector of length 2 containing the limits of observed sum scores.
- **scrfine:** A fine mesh of test score values for plotting.
- **scrvec:** A vector of length N containing the examinee or respondent sum scores.
- **itemvec:** A vector of length n containing the question or item sum scores.
- **percntrnk:** A vector length N containing the sum score percentile ranks.

indexQnt: A numeric vector of length 2*nbin + 1 containing the bin boundaries alternating with the bin centers. These are initially defined as seq(0,100,len=2*nbin+1).

Sdim: The total dimension of the surprisal scores.

PcntMarkers: The marker percentages for plotting: 5, 25, 50, 75 and 95.

`NumDensBasis`	The number of basis functions for representing the score density.
`ncycle`	The number of cycles executed by function `Analyze()`.
`itdisp`	If TRUE, the progress of the iterations within each cycle for estimating index are reported.
`verbose`	If TRUE, the stages of analysis within each cycle for estimating index are reported.

Details

The cycling process is described in detail in the references, and displayed in R code in the vignette `SweSATQuantitativeAnalysis`.

Value

The list vector `parmList` where each member is a named list object containing the results of an analysis cycle. These results are:

`index:`	The optimal estimates of the score index values for the examinees/respondents. This is a vector of length `N`.
`indexQnt:`	A vector of length 2*nbin+1 containing bin boundaries alternating with bin edges.
`SfdList:`	A list vector containing results from the estimation of surprisal curves. The list vector is of length `n`, the number of questions or items in the test of rating scale. For details concerning these results, see function `Sbinsmth()`.
`meanF:`	For each person, the mean of the optimal fitting function values.
`binctr:`	A vector of length `nbin` containing the bin centers within the interval [0,100].
`bdry:`	A vector of length `nbin+1` containing the bin boundaries.
`freq:`	A vector of length `nbin` containing the number of score index values in the bins. An score index value is within a bin if it is less than or equal to the upper boundary and greater than the lower boundary. The first boundary also contains zero values.
`pdf_fd`	Functional probability curves
`logdensfd:`	A functional data object defining the estimate of the log of the probability density function for the distribution of the score index values.
`C:`	The normalizing value for probability density functions. A density value is computed by dividing the exponential of the log density value by this constant.
`denscdf:`	The values over a fine mesh of the cumulative probability distribution function. These values start at 0 and end with 1 and are increasing. Ties are often found at the upper boundary, so that using these values for interpolation purposes may require using the vector `unique(denscdf)`.

indcdf	Equally spaced index values to match the number in denscdf.
Qvec	Locations of the marker percents.
index	The positions of each test taker on the score index continuum.
Fval:	A vector of length N containing the values of the negative log likelihood fitting criterion.
DFval:	A vector of length N containing the values of the first derivative of the negative log likelihood fitting criterion.
D2Fval:	A vector of length N containing the values of the second derivative of the negative log likelihood fitting criterion.
active:	A vector of length N of the activity status of the values of index. If convergence was not achieved, the value is TRUE, otherwise FALSE.
infoSurp:	The length of the space curve defined by the surprisal curves.

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

See Also

make_dataList, TG_analysis, index_distn, index2info, index_fun, Sbinsmth

Examples

```
## Not run:
  #  Example 1:  Input choice data and key for the short version of the
  #  SweSAT quantitative multiple choice test with 24 items and 1000 examinees
  #  input the choice data as 1000 strings of length 24
  #  setup the input data list object
  dataList <- Quant_13B_problem_dataList
  #  define the initial examinee indices and bin locations
  index    <- dataList$percntrnk
  indexQnt <- dataList$indexQnt
  #  Set the number of cycles (default 10 but here 5)
  ncycle <- 5
  parmListvec <- Analyze(index, indexQnt, ncycle=ncycle, dataList,
                         verbose=TRUE)
  #  two column matrix containing the mean fit and arclength values
  #  for each cycle
  HALsave <- matrix(0,ncycle,2)
  for (icycle in 1:ncycle) {
    HALsave[icycle,1] <- parmListvec[[icycle]]$meanF
```

```
    HALsave[icycle,2] <- parmListvec[[icycle]]$infoSurp
  }
  #  plot the progress over the cycles of mean fit and arc length
  par(mfrow=c(2,1))
  plot(1:ncycle, HALsave[,1], type="b", lwd=2,
       xlab="Cycle Number",ylab="Mean H")
  plot(1:ncycle, HALsave[,2], type="b", lwd=2,
       xlab="Cycle Number", ylab="Arc Length")
## End(Not run)
```

`chcemat_simulate` *Simulate a test or scale data matrix.*

Description

Used in `dataSimulation`, this function sets up an N by n matrix of index values that specify the index of the option chosen by an examinee or respondent for a specific question.

Usage

```
chcemat_simulate(index.pop, SfdList)
```

Arguments

`index.pop` A vector containing population score index values at which data are to be simulated.

`SfdList` A numbered list object produced by a TestGardener analysis of a test. Its length is equal to the number of items in the test or questions in the scale. Each member of `SfdList` is a named list containing information computed during the analysis. These named lists contain these objects:

- **Sfd:** A functional data object containing the `M` surprisal curves f. or a question.
- **M:** The number of options.
- **Pbin:** A matrix containing proportions at each bin.
- **Sbin:** A matrix containing surprisal values at each bin.
- **Pmatfine:** A matrix of probabilities over a fine mesh.
- **Smatfine:** A matrix of surprisal values over a fine mesh.
- **DSmatfine:** A matrix of the values of the first derivative of surprisal curves over fine mesh.
- **D2Smatfine:** A matrix of the values of the second derivative of surprisal curves over fine mesh.

Details

For each question and each examinee a vector of random multinomial integer values is generated using the probability transforms of the surprisal curves and the examinee's score index value.

Value

An N by n matrix of integer index values.

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315. s

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

`dataSimulation` *Simulation Based Estimates of Error Variation of Score Index Estimates*

Description

Estimate sum score,s score index values index and test information values bias and mean squared errors using simulated data.

Usage

```
dataSimulation(dataList, parmList, nsample = 1000)
```

Arguments

`dataList`	The list object set up by function `make_dataList`.
`parmList`	The list object containing objects computed by function `Analyze`.
`nsample`	The number of simulated samples.

Value

A named list object containing objects produced from analyzing the simulations, one set for each simulation:

`sumscr:`	Sum score estimates
`index:`	Score index estimates
`mu:`	Expected sum score estimates
`info:`	Total arc length estimates
`index.pop:`	True or population score index values
`mu.pop:`	Expected sum score population values
`info.pop:`	Total test length population values
`n:`	Number of items
`nindex:`	Number of index values
`indfine:`	Fine mesh over score index range
`Qvec:`	Five marker percentages: 5, 25, 50, 75 and 95

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

See Also

`scorePerformance`

---

density_plot *Plot the probability density function for a set of test scores*

Description

Plots the probability density function of a set of score values that are not at the score boundaries as a smooth curve, and also plots the proportions of score values at both boundaries as points. The score values are typically either the values of the score index values index or the infoSurp or information score values.

Usage

```
density_plot(scrvec, scrrng, Qvec, xlabstr=NULL, titlestr=NULL,
                          scrnbasis=15, nfine=101)
```

Arguments

`scrvec`	A vector of N score values
`scrrng`	A vector of length 2 containing boundary values
`Qvec`	A vector of length 5 containing the score values corresponding to the marker percentages 5, 25, 50, 75 and 95.
`xlabstr`	Label for abscissa
`titlestr`	Label for plot
`scrnbasis`	The number of spline basis functions used for representing the smooth density function
`nfine`	Number of plotting points

Value

A plot of the density function and a list vector `densfine` containing:

- `densfine:` Density values over a mesh of equally-spaced values of length 101.
- `N_min:` The number of examinees estimated to have zero information.
- `N_max:` The number of examinees estimated to have full information.

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

See Also

`scoreDensity`

Examples

```
#  Example 1.  Display probability density curve for the
#  score index values for the short SweSAT multiple choice test with
#  24 items and 1000 examinees
index <- Quant_13B_problem_parmList$index
Qvec  <- Quant_13B_problem_parmList$Qvec
#  plot the density for the score indices within interval c(0,100)
oldpar <- par(no.readonly=TRUE)
on.exit(oldpar)
par(mfrow=c(2,1))
density_plot(index, c(0,100), Qvec, xlabstr="Score index",
             titlestr="SweSAT 13B Theta Density",
             scrnbasis=11, nfine=101)
#  arc length or information values
scopevec <- Quant_13B_problem_infoList$scopevec
Qinfovec <- Quant_13B_problem_infoList$Qinfovec
infoSurp <- Quant_13B_problem_infoList$infoSurp
#  plot the density for the score indices within interval c(0,infoSurp)
density_plot(scopevec, c(0,infoSurp), Qinfovec, xlabstr="Score index",
             titlestr="SweSAT 13B Theta Density",
             scrnbasis=11, nfine=101)
```

`DFfun` *Compute the first and second derivatives of the negative log likelihoods*

Description

DFfun computes the first and second derivatives of the negative log likelihoods for a set of examinees. Items can be either binary or multi-option. The analysis is within the closed interval [0,100].

Usage

```
DFfun(index, SfdList, chcemat)
```

Arguments

`index`	Initial values for score indices in [0,n]/[0,100]. Vector of size N.
`SfdList`	A numbered list object produced by a TestGardener analysis of a test. Its length is equal to the number of items in the test or questions in the scale. Each member of `SfdList` is a named list containing information computed during the analysis.
`chcemat`	An `N` by `n` matrix of responses. If `N` = 1, it can be a vector of length `n`.

Value

A named list for results `DF` and `D2F`:

`DF:`	First derivatives of the negative log likelihood values, vector of size N
`D2F:`	Second derivatives of the negative log likelihood values, vector of size N

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

See Also

`make_dataList`, `index_fun`, `Ffun`, `Ffuns_plot`

Examples

```
#  Example 1:
#  Compute the first and second derivative values of the objective function
#  for locating each examinee for the 24-item short form of the
#  SweSAT quantitative test on the percentile score index continuum.
#  Use only the first five examinees.
chcemat <- Quant_13B_problem_dataList$chcemat
SfdList <- Quant_13B_problem_parmList$SfdList
index   <- Quant_13B_problem_parmList$index
DFfunResult <- DFfun(index[1:5], SfdList, chcemat[1:5,])
DFval  <- DFfunResult$DF
D2Fval <- DFfunResult$D2F
```

`entropies` *Entropy measures of inter-item dependency*

Description

Entropy I_1 is a scalar measure of how much information is required to predict the outcome of a choice number 1 exactly, and consequently is a measure of item effectiveness suitable for multiple choice tests and rating scales. Joint entropy $J_{1,2}$ is a scalar measure of the cross-product of multinomial vectors 1 and 2. Mutual entropy $I_{1,2} = I_1 + I_2 - J_{1,2}$ is a measure of the co-dependency of items 1 and 2, and thus the analogue of the negative log of a squared correlation R^2. this function computes all four types of entropies for two specificed items.

Usage

```
entropies(index, m, n, chcemat, noption)
```

Arguments

`index`	A vector of length N containing score index values for each test taker.
`m`	The index of the first choice.
`n`	The index of the second choice.
`chcemat`	The data matrix containing the indices of choisen options for each test taker.
`noption`	A vector containing the number of options for all items.

Value

A named list object containing objects produced from analyzing the simulations, one set for each simulation:

`I_m:`	The entropy of item m.
`I_n:`	The entropy of item n.
`J_nm:`	The joint entropy of items m and n.
`I_nm:`	The mutual entropy of items m and n.

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

See Also

Entropy_plot

Examples

```
#  Load needed objects
chcemat <- Quant_13B_problem_dataList$chcemat
index   <- Quant_13B_problem_parmList$index
noption <- matrix(5,24,1)
#  compute mutual entropies for all pairs of the first 6 items
Mvec    <- 1:6
Mlen    <- length(Mvec)
Hmutual <- matrix(0,Mlen,Mlen)
for (i1 in 1:Mlen) {
  for (i2 in 1:i1) {
    Result <- entropies(index, Mvec[i1], Mvec[i2], chcemat, noption)
    Hmutual[i1,i2] = Result$Hmutual
    Hmutual[i2,i1] = Result$Hmutual
  }
}
print("Matrix of mutual entries (off-digagonal) and self-entropies (diagonal)")
print(round(Hmutual,3))
```

`Entropy_plot` *Plot item entropy curves for selected items or questions.*

Description

Item the value of the entropy curve at a point `theta` is the expected value of the surprisal curve values. Entropy is a measure of the randomness of the surprisal value, which is maximized when all the surprisal curves have the same value and has a minimum of zero if all but a single curve has probability zero. This is unattainable in the calculation, but can be arbitrarily close to this state.

Usage

```
Entropy_plot(scrfine, SfdList, Qvec, dataList, plotindex=1:n,
             plotrange=c(min(scrfine),max(scrfine)), height=1.0, value=0,
             ttlsz=NULL, axisttl=NULL, axistxt=NULL)
```

Arguments

`scrfine`	A vector of length nfine (usually 101) containing equally spaced points spanning the plotrange. Used for plotting.
`SfdList`	A numbered list object produced by a TestGardener analysis of a test. Its length is equal to the number of items in the test or questions in the scale. Each member of `SfdList` is a named list containing information computed during the analysis.
`Qvec`	The five marker percentile values.
`dataList`	A list vector containing objects essential to an analysis.
`plotindex`	A set of integers specifying the numbers of the items or questions to be displayed.
`plotrange`	A vector of length 2 containing the plot boundaries within or over the score index interval c(0,100).
`height`	A positive real number defining the upper limit on the ordinate for the plots.
`value`	Number required by ggplot2. Defaults to 0.
`ttlsz`	Title font size.
`axisttl`	Axis title font size.
`axistxt`	Axis text(tick label) font size.

Details

An entropy curve for each question indexed in the `index` argument. A request for a keystroke is made for each question. The answer to question strongly defines the optimal position of an estimated score index value where the curve is high value. Values of entropy curves typically range over [0,1].

Value

The plots of the entropy curves specified in `plotindex` are produced as a side effect. If `saveplot` is TRUE, the plots of item entropy curves specified in `plotindex` are bundled into a single postscript or .pdf file and the file name is defined by `paste(dataList$titlestr,i,'-entropy.pdf',sep="")`. The file is then output as a returned value.

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

See Also

`Sensitivity_plot`, `Power_plot`, `Ffuns_plot`, `ICC_plot`

Examples

```
#  Example 1.  Display the item entropy curves for the
#  short SweSAT multiple choice test with 24 items and 1000 examinees
#  plot the entropy curve for the first item
dataList <- Quant_13B_problem_dataList
SfdList  <- Quant_13B_problem_parmList$SfdList
Qvec     <- Quant_13B_problem_parmList$Qvec
scrfine  <- seq(0,100,len=101)
oldpar   <- par(no.readonly=TRUE)
Entropy_plot(scrfine, SfdList, Qvec, dataList, plotindex=1)
par(oldpar)
```

`eval.surp` *Values of a Functional Data Object Defining Surprisal Curves.*

Description

A surprisal vector of length `M` is minus the log to a positive integer base `M` of a set of `M` multinomial probabilities. Surprisal curves are functions of a one-dimensional index set, such that at any value of the index set the values of the curves are a surprisal vector. See Details below for further explanations.

Usage

```
eval.surp(evalarg, Sfdobj, Zmat, nderiv = 0)
```

Arguments

`evalarg`	a vector or matrix of argument values at which the functional data object is to be evaluated.
`Sfdobj`	a functional data object of dimension `M-1` to be evaluated.
`Zmat`	An `M` by `M-1` matrix satisfying `Zmat'Zmat = I}` and `\code{Zmat'1 = 0`.
`nderiv`	An integer defining a derivatve of `Sfdobj` in the set `c(0,1,2)`.

Details

A surprisal `M`-vector is information measured in `M`-bits. Since a multinomial probability vector must sum to one, it follows that the surprisal vector `S` must satisfy the constraint `log_M(sum(M^(-S)) = 0`. That is, surprisal vectors lie within a curved `M-1`-dimensional manifold.

Surprisal curves are defined by a set of unconstrained `M-1` B-spline functional data objects defined over an index set that are transformed into surprisal curves defined over the index set.

Let `C` be a `K` by `M-1` coefficient matrix defining the B-spline curves, where `K` is the number of B-spline basis functions.

Let a `M` by `M-1` matrix `Z` have orthonormal columns. Matrices satisfying these constraints are generated by function `zerobasis()`.

Let `N` by `K` matrix be a matrix of B-spline basis values evaluated at `N` evaluation points using function `eval.basis()`.

Let `N` by `M` matrix `X = B * C * t(Z)`.

Then the `N` by `M` matrix `S` of surprisal values is `S = -X + outer(log(rowSums(M^X))/log(M),rep(1,M))`.

Value

A `N` by `M` matrix `S` of surprisal values at points `evalarg`, or their first or second derivatives.

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

See Also

`smooth.surp`

Examples

```
#  see example in man/smooth.surp.Rd
```

`Fcurve` *Construct grid of 101 values of the fitting function*

Description

A fast grid of values of the fitting function or one of its first two derivatives is constructed for use in function `indexsearch`.

Usage

```
Fcurve(SfdList, chcevec, nderiv=0)
```

Arguments

`SfdList`	A list vector containing specifications of surprisal curves for each item.
`chcevec`	A N by n matrix containing indices of chosen items for each test taker.
`nderiv`	Integer 0, 1 or 2 to indicate which level of derivative to use.

Value

A vector of length 101 containing grid values of a derivative of the fitting function

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

See Also

index_search

Examples

```
#  Compute a grid of values of the objective function for locating each
#  examinee or respondent for the 24-item short form of the SweSAT
#  quantitative test on the percentile score index continuum [0,100].
chcemat <- Quant_13B_problem_dataList$chcemat
SfdList <- Quant_13B_problem_parmList$SfdList
index   <- Quant_13B_problem_parmList$index
n       <- ncol(chcemat)
#  Fitting function for the first examinee
j <- 1
chcevec   <- as.numeric(chcemat[j,])
Fcurve1   <- Fcurve(SfdList, chcevec, 0)
#  First derivative of the fitting function for the first examinee
DFcurve1  <- Fcurve(SfdList, chcevec, 1)
#  Second derivative of the fitting function for the first examinee
D2Fcurve1 <- Fcurve(SfdList, chcevec, 2)
oldpar  <- par(no.readonly=TRUE)
par(mfrow=c(3,1))
indfine <- seq(0,100,len=101)
plot(indfine, Fcurve1,   type="l", xlab="", ylab="Fitting curve",
     main="Examinee 1")
plot(indfine, DFcurve1,  type="l", xlab="", ylab="First derivative")
points(index[1], 0, pch="o")
abline(0,0,lty=2)
plot(indfine, D2Fcurve1, type="l",
     xlab="Score index", ylab="Second derivative")
abline(0,0,lty=2)
points(index[1], 0, pch="o")
par(oldpar)
```

Ffun *Compute the negative log likelihoods associated with a vector of score index values.*

Description

Ffun computes the negative log likelihoods for a set of examinees, each at a single value index.

Usage

```
Ffun(index, SfdList, chcemat)
```

Arguments

index	A vector of size N containing values for score indices in the interval [0,100].
SfdList	A numbered list object produced by a TestGardener analysis of a test. Its length is equal to the number of items in the test or questions in the scale. Each member of SfdList is a named list containing information computed during the analysis.
chcemat	An N by n matrix of responses or, for a single examinee, a vector of length n.

Value

A vector of length N of negative log likelihood values.

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

See Also

make_dataList, index_fun, Ffun, Ffuns_plot

Examples

```
#  Example 1:  Compute the values of the objective function for locating each
#  examinee or respondent for the 24-item short form of the SweSAT quantitative
#  test on the percentile score index continuum [0,100].
#  Use only the first five examinees
chcemat <- Quant_13B_problem_dataList$chcemat
SfdList <- Quant_13B_problem_parmList$SfdList
index   <- Quant_13B_problem_parmList$index
Fval    <- Ffun(index[1:5], SfdList, chcemat[1:5,])
```

`Ffuns_plot` *Plot a selection of fit criterion F functions and their first two derivatives.*

Description

These plots indicate whether an appropriate minimum of the fitting criterion was found. The value of index should be at the function minimum, the first derivative be close to zero there, and the second derivative should be positive. If these conditions are not met, it may be worthwhile to use function `indexfun` initialized with an approximate minimum value of score index index to re-estimate the value of index.

Usage

```
Ffuns_plot(evalarg, index, SfdList, chcemat, plotindex=1)
```

Arguments

`evalarg`	A vector containingg the sore index values to be evaluated.
`index`	The vector of of length `N` of score index values.
`SfdList`	The list vector of length `n` containing the estimated surprisal curves.
`chcemat`	The entire `N` by `n` matrix of choice indices.
`plotindex`	A subset of the integers `1:N`.

Details

The curves are displayed in three vertically organized panels along with values of index and the values and first two derivative values of the fit criterion. If more than one index value is used, a press of the Enter or Return key moves to the next index value.

Value

A list vector is returned which is of the length of argument `plotindex`. Each member of the vector is a `gg` or `ggplot` object for the associated `plotindex value`. Each plot can be displayed using the `print` command. The plots of item power are produced as a side value even if no output object is specified in the call to the function.

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

See Also

index_fun, Ffun, DFfun

Examples

```
#  Example 1.  Display fit criterion values and derivatives for the
#  short SweSAT multiple choice test with 24 items and 1000 examinees
chcemat   <- Quant_13B_problem_dataList$chcemat
index     <- Quant_13B_problem_parmList$index
SfdList   <- Quant_13B_problem_parmList$SfdList
plotindex <- 1:3
indfine   <- seq(0,100,len=101)
Ffuns_plot(indfine, index, SfdList, chcemat, plotindex)
```

ICC *Plotting probability and surprisal curves for an item*

Description

This is an S3 object that contains information essential plotting probability and surprisal curves for a single multiple choice or rating question. Bin probabilities and surprisal values can also be plotted.

Usage

```
ICC(x, M, Sfd, Zmat, Pbin, Sbin, Pmatfine, Smatfine, DSmatfine, D2Smatfine,
                  PStdErr, SStdErr, ItemArcLen, itemStr=NULL, optStr=NULL)
```

Arguments

x	An item number.
M	The number of options for this item, including an option for missing or illegal values if required.
Sfd	A functional surprisal curve object defined by K B-spline basis functions and a K by M-1 matrix of coefficients.
Zmat	An M by M-1 matrix satisfying the conditions t(Zmat) Zmat = I and columns sum to zero.
Pbin	A nbin by M matrix of probabilities that a given bin is chosen by a test taker.
Sbin	A nbin by M matrix of surprisal values for the probabilities in Pbin.
Pmatfine	A 101 by M matrix of probability curve values over equally-spaced score index values spanning the interval [0,100].
Smatfine	A 101 by M matrix of surprisal curve values corresponding to the probability values in Pmatfine.
DSmatfine	A 101 by M matrix of first derivative values with respect to score index values for the surprisal values.

D2Smatfine	A 101 by M matrix of second derivative values.
PStdErr	A 101 by M matrix of standard error estimates for the probability curve values.
SStdErr	A 101 by M matrix of standard error estimates for the surprisal curve values.
ItemArcLen	The scope or arc length of the item curve.
itemStr	A string that is the name of the item.
optStr	A character vector containing labels for the item options.

Details

The name ICC for this object is an acronym for the term "item characteristic curve" widely used in the psychometric commuunity.

Function ICC is set up after the initialization process in function `make_dataList()` has created the members of `dataList`. Within this list is object `SfdList`, which cintains a functional data object `Sfd` for each item. Both the intial coefficient matrices and the subsequent estimates of them are available from `Sfd$coefs`, and therefore are available in the ICC object. These coefficient matrices are `K` by `M-1` where `K` is the number of basis functions and `M` is the number of options for asn item.

Value

The values returned are simply those in the argument list. The S3 ICC object checks each of these and makes available the S3 commands or methods `str`, `print` and `plot` that apply the corresponding `ICC` versions of these opterations.

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

ICC_plot — *Plot probability and surprisal curves for test or scale items.*

Description

ICC_plot plots each item in argument `plotindex` in turn after function`Sbinsmth()` has used spline smoothing to estimate item and option characteristic curves.

Usage

```
ICC_plot(scrfine, SfdList, dataList, Qvec,
         binctr=NULL, data_point = FALSE, ci = FALSE,
         plotType="S", Srng=c(0,5), DSrng=c(-0.2, 0.2), plotindex=1:n,
         titlestr = NULL, itemscopevec = rep(0, length(plotindex)),
         plotTitle = TRUE,autoplot = FALSE, plotMissing = TRUE,
         plotrange=c(min(scrfine),max(scrfine)), shaderange = NULL,
         ttlsz = NULL, axisttl = NULL, axistxt = NULL,
         lgdlab = NULL, lgdpos = "bottom")
```

Arguments

`scrfine`	A vector of 101 plotting points.
`SfdList`	A numbered list object produced by a TestGardener analysis of a test. Its length is equal to the number of items in the test or questions in the scale. Each member of `SfdList` is a named list containing information computed during the analysis.
`dataList`	A list that contains the objects needed to analyse the test or rating scale.
`Qvec`	A vector of five marker percentile values. For plotting over information, this is replaced by Qinfovec returned as parmList$Qinfovec.
`binctr`	A vector of bin center values. If the plot is over arc length or information, `binctr` is modified before calling `Sbinsth_plot` by the command `binctrinfo = pracma::interp1(indfine, alfine, binctr)`, and argument `binctr` is replaced by `binctrinfo`.
`data_point`	A logical value indicating whether to plot the data points.
`ci`	A logical value indicating whether to plot the confidence limits.
`plotType`	Type(s) of plot, default as "P" for probability, can also be "S" for surprisal, "DS" for sensitivity, and any combination of the three
`Srng`	A vector of length 2 specifing the plotting range for surprisal values.
`DSrng`	A vector of length 2 specifing the plotting range for sensitivity values.
`plotindex`	A vector of indices of items to be plotted.
`titlestr`	plot title
`itemscopevec`	A numeric vector containing item scope values.
`plotTitle`	indicator of showing the plot title, default as TRUE
`autoplot`	indicator for plotting all items in a batch
`plotMissing`	Determine if plot the extra option for missing/spoiled responses.
`plotrange`	A vector of length 2 containing the plot boundaries of the score index interval.
`shaderange`	a list of length 2 vector(s); set if users want to gray out specific score range(s)
`ttlsz`	Title font size.
`axisttl`	Axis title font size.
`axistxt`	Axis text(tick label) font size.
`lgdlab`	Legend label font size.
`lgdpos`	legend position, could be set as "None" to remove the legend.

Value

A list vector is returned which is of the length of argument `plotindex`. Each member of the vector is a gg or `ggplot` object for the associated `plotindex value`. Each plot can be displayed using the `print` command. The plots of item power are produced as a side value even if no output object is specified in the call to the function.

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

See Also

`ICC`, `Sensitivity_plot`, `Power_plot`, `Entropy_plot`, `Sbinsmth`

Examples

```
#  Example 1.  Display the item surprisal curves for the
#  short SweSAT multiple choice test with 24 items and 1000 examinees
dataList <- Quant_13B_problem_dataList
SfdList  <- Quant_13B_problem_parmList$SfdList
Qvec     <- Quant_13B_problem_parmList$Qvec
binctr   <- Quant_13B_problem_parmList$binctr
infoSurpvec  <- Quant_13B_problem_infoList$infoSurpvec
Qinfovec     <- Quant_13B_problem_infoList$Qinfovec
bininfoctr   <- Quant_13B_problem_infoList$bininfoctr
titlestr <- "Quant_13B_problem"
#  plot the curves for the first question over the score index
oldpar  <- par(no.readonly=TRUE)
indfine <- seq(0,100,len=101)
ICC_plot(indfine, SfdList, dataList, Qvec, binctr,
         data_point = TRUE, plotType = c("S", "P"),
         Srng=c(0,4), plotindex=1)
#  plot the curves for the first question over test information
ICC_plot(infoSurpvec, SfdList, dataList, Qinfovec, bininfoctr,
          data_point = TRUE, plotType = c("S", "P"),
          Srng=c(0,4), plotindex=1)
par(oldpar)
```

`index2info` *Compute results using arc length or information as the abscissa.*

Description

The one-dimensional psychometric model defines a space curve within the vector space defined by the total collection of option surprisal curves. This curve is a valuable resource since positions along the curve are defined in bits and positions on the curve are subject to the same strict properties that apply to physical measurements.

Function `index2info` is required to convert objects defined over the score index continuum `c(0,100)` to the same objects over the arc length continuum `c(0,infoSurp)`, and also vice versa. Since the arc length or information continuum is along a space curve that is invariant under strictly monotone transformations of the score index `index`, and is also a metric, it is an ideal choice for the abscissa in all plots.

Usage

```
index2info(index, Qvec, SfdList, binctr, itemindex=1:n, plotrng=c(0,100),
           shortwrd)
```

Arguments

`index`	A vector of score index, test score, or arc length values, one for each examinee or respondent.
`Qvec`	A vector of locations of the five marker percentages.
`SfdList`	A numbered list object produced by a TestGardener analysis of a test. Its length is equal to the number of items in the test or questions in the scale. Each member of `SfdList` is a named list containing information computed during the analysis.
`binctr`	A vector of locations of the bin centers.
`itemindex`	A vector containing the indices of the items to be used.
`plotrng`	A vector of length 2 containing the starting score index and end score index values of the range to be plotted.
`shortwrd`	If TRUE only vectors infoSurp and infoSurpvec are returned in order to speed up the computation within cycles in function `Analyze()` where only these objects are required. The default is FALSE.

Value

A named list object containing these results of the analysis:

`infoSurp`	The length of the test information or scale curve.
`infoSurpvec`	Positions on the test information or scale curve corresponding to a fine mesh of score index values (typically 101 values between 0 and 100).
`infoSurpfd`	Functional data object representing the relation between the score index abscissa and the infoSurp or information ordinate.

scopevec	A vector of positions on the test information or scale curve corresponding to the input score index values in argument index.
Qvec_al	Values in arc length of the five marker percentages.
binctr_al	Values in arc length of the bin centers.
Sfd.info	A functional data object representing the relation between the infoSurp or information abscissa and the score index ordinate.
Sdim.index	The dimension of the overspace, which equal to sum of the number of options in the items specified in itemindex.

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

See Also

Analyze

Examples

```
#  Example 1.  Display the scope or information curve for the
#  short SweSAT multiple choice test with 24 items and 1000 examinees.
#  The scope curve is constructed using the complete analysis cycles.
#  Set up the required arguments using the converged parmList object.
indfine     <- seq(0,100,len=101)
index       <- Quant_13B_problem_parmList$index
Qvec        <- Quant_13B_problem_parmList$Qvec
SfdList     <- Quant_13B_problem_parmList$SfdList
binctr      <- Quant_13B_problem_parmList$binctr
#  Carry out the construction of the information results.
infoList    <- index2info(index, Qvec, SfdList, binctr)
# Plot the shape of the information curve
oldpar <- par(no.readonly=TRUE)
Scope_plot(infoList$infoSurp, infoList$infoSurpvec)
par(oldpar)
```

index_distn *Compute score density*

Description

Computes the cumulated density for distribution function, the probability density function, and the log probability density function as fd objects by spline smoothing of the score values `indexdens` using the basis object `logdensbasis`. The norming constant `C` is also output.

The score values may score index values `index`, expected test score values `mu`, or arc length locations on the test information or scale curve. The argument functional data object `logdensfd` should have a range that is appropriate for the score values being represented: For score indices, [0,100], for expected test scores, the range of observed or expected scores; and for test information curve locations in the interval [0,`infoSurp`].

Usage

```
index_distn(indexdens, logdensbasis,
              pvec=c(0.05, 0.25, 0.50, 0.75, 0.95), nfine = 101)
```

Arguments

`indexdens`	A vector of score index, test score, or arc length values. In the score index case, these are usually only the values in the interior of the interval [0,100].
`logdensbasis`	A functional basis object for representing the log density function. The argument may also be a functional data object (`fd`) or a functional basis object (`Sbasis`).
`pvec`	A vector length NL containing the marker percentages.
`nfine`	The number of values in a fine grid, default as 101.

Value

A named list containing:

`pdf_fd:`	An fd object for the probability density function values over the fine mesh.
`cdffine:`	A vector of cumulative probability values beginning with zero and ending with 1. It must not have ties.
`pdffine:`	A vector of probability values.
`logdensfd:`	A functional data object (`fd`) representing the log of the probability function for input `index`.
`C:`	The normalization constant for computing the probability density function with the command `densityfd = exp(logdensfd)/C`.
`denscdf:`	A set of unique values of the cumulative probability function defined over an equally spaced mesh of score index values of the same length as `denscdf`.
`indcdf:`	A vector of values within [0,100] corresponding to the values in `denscdf`.

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

See Also

index_fun, index2info, mu, scoreDensity

Examples

```
#  Example 1.  Display the item power curves for the
#  short SweSAT multiple choice test with 24 items and 1000 examinees
#  Assemble information for estimating index density
indfine   <- seq(0,100,len=101)
SfdList   <- Quant_13B_problem_parmList$SfdList
index     <- Quant_13B_problem_parmList$index
N         <- length(index)
#  Define the density for only interior index values
inside    <- index > 0 & index < 100
indexdens <- index[inside]
logdensbasis <- fda::create.bspline.basis(c(0,100), 15)
index_distnList <- index_distn(index[inside], logdensbasis)
denscdf         <- as.numeric(index_distnList$denscdf)
indcdf          <- as.numeric(index_distnList$indcdf)
# adjusted marker score index values are computed by interpolation
markers <- c(.05, .25, .50, .75, .95)
Qvec    <- pracma::interp1(denscdf, indcdf, markers)
result  <- density_plot(indexdens, c(0,100), Qvec)
```

---

index_fun   *Compute optimal scores*

Description

The percentile score index values are estimated for each person. The estimates minimize the negative log likelihoods, which are a type of surprisal. The main optimization method is a safe-guarded Newton-Raphson method.

For any iteration the method uses only those scores that are within the interior of the interval [0,100] or at a boundary with a first derivative that would take a step into the interior, and have second derivative values exceeding the value of argument `crit`. Consequently the number of values being optimized decrease on each iteration, and iterations cease when either all values meet the convergence criterion or are optimized on a boundary, or when the number of iterations reaches `itermax`.

At that point, if there are any interior scores still associated with either non-positive second derivatives or values that exceed `crit`, the minimizing value along a fine mesh is used.

If `itdisp` is positive, the number of values to be estimated are printed for each iteration.

Usage

```
index_fun(index, SfdList, chcemat, itermax = 20, crit = 0.001,
          itdisp = FALSE)
```

Arguments

`index`	A vector of size `N` containing initial values for score indices in the interval [0,100].
`SfdList`	A list vector of length equal to the number of questions. Each member contains eight results for the surprisal curves associated with a question.
`chcemat`	A matrix number of rows equal to the number of examinees or respondents, and number of columns equal to number of items. The values in the matrix are indices of choices made by each respondent to each question.
`itermax`	Maximum number of iterations for computing the optimal index values. Default is 20.
`crit`	Criterion for convergence of optimization. Default is 1e-8.
`itdisp`	If TRchcematE, results are displayed for each iteration.

Value

A named list with these members:

`index_out:`	A vector of optimized score index value.
`Fval:`	The negative log likelihood criterion.
`DFval:`	The first derivative of the negative likelihood.
`D2Fval:`	The second derivative of the negative likelihood.
`iter:`	The number iterations used.

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

See Also

`index_distn`, `Ffun`, `DFfun`, `index2info`, `scoreDensity`

Examples

```
#  Optimize the indices defining the data fits for the first five examinees
#  input the choice indices in the 1000 by 24 choice index matrix
chcemat   <- Quant_13B_problem_chcemat
#  First set up the list object for surprisal curves computed from
#  initial index estimates.
SfdList   <- Quant_13B_problem_dataList$SfdList
#  Their initial values are the percent rank values ranging over [0,100]
index_in  <- Quant_13B_problem_dataList$percntrnk[1:5]
#  set up choice indices for first five examinees
chcemat_in <- chcemat[1:5,]
#  optimize the initial indices
indexfunList <- index_fun(index_in, SfdList, chcemat_in)
#  optimal index values
index_out    <- indexfunList$index_out
#  The surprisal data fit values
Fval_out     <- indexfunList$Fval
#  The surprisal data fit first derivative values
DFval_out    <- indexfunList$DFval
#  The surprisal data fit second derivative values
D2Fval_out   <- indexfunList$D2Fval
#  The number of index values that have not reached the convergence criterion
active_out   <- indexfunList$active
```

---

index_search *Ensure that estimated score index is global*

Description

Multiple minima are found quite often in the data fitting function that is minimized using function `indexfun`, and in roughly 10 percent of the estimates there is a minimum that it lower than that detected. The function searches a mesh of 101 points for minima, computes the fitting function at the minima, and assigns the location of the global minimum as the replacement index if the location differs by more than 0.5 from the value identified by `index_fun`. The function values and their first two derivatives are also replaced.

Usage

```
index_search(SfdList, chcemat, index, Fval, DFval, D2Fval, indexind=1:N)
```

Arguments

`SfdList`	A list vector containing specifications of surprisal curves for each item.
`chcemat`	An `N` by `n` matrix containing indices of chosen items for each test taker.
`index`	A vector containing all the score index values.
`Fval`	A vector containing the `N` function values.
`DFval`	A vector containing the `N` first derivative values.
`D2Fval`	A vector containing the `N` second derivative values.
`indexind`	A vector containing indices of values to be processed.

Value

A named list object containing objects produced from analyzing the simulations, one set for each simulation:

- `index:` A vector containing all the score index values including those that are altered.
- `Fval:` A vector containing the N function values included those that are altered.
- `DFval:` A vector containing the N first derivative values included those that are altered.
- `D2Fval:` A vector containing the N second derivative values included those that are altered.
- `changeindex:` Indices of the index values that are altered

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

See Also

`index_fun`

Examples

```
#  Search for values of index that are not at the global minimum of the
#  fitting function and replace them as well as their function and
#  derivative values associated with the fine grid value nearest the
#  the global minimum.
chcemat <- Quant_13B_problem_chcemat
key     <- Quant_13B_problem_key
SfdList <- Quant_13B_problem_parmList$SfdList
index   <- Quant_13B_problem_parmList$index
Fval    <- Quant_13B_problem_parmList$Fval
DFval   <- Quant_13B_problem_parmList$DFval
D2Fval  <- Quant_13B_problem_parmList$D2Fval
Result  <- index_search(SfdList, chcemat, index, Fval, DFval, D2Fval)
changeindex <- Result$changeindex
print(paste("Number changed =",length(changeindex)))
change  <- index[changeindex] - Result$index[changeindex]
```

make_dataList *Make a list object containing information required for analysis of choice data.*

Description

The list object `dataList` contains 22 objects that supply all of the information required to analyze the data. Initial values of the score indices in object `theta` and the bin boundaries and centres in object `thetaQnt`. The returned named list object contains 22 named members, which are described in the value section below.

Usage

```
make_dataList(chcemat, scoreList, noption, sumscr_rng=NULL,
              titlestr=NULL, itemlabvec=NULL, optlabList=NULL,
              nbin=nbinDefault(N), NumBasis=7, jitterwrd=TRUE,
              PcntMarkers=c( 5, 25, 50, 75, 95), verbose=FALSE)
```

Arguments

`chcemat` An N by n matrix. Column `i` must contain the integers from 1 to `M_i`, where `M_i` is the number of options for item `i`. If missing or illegitimate responses exist for item `i`, the column must also contain an integer greater than `M_i` that is used to identify such responoses. Alternatively, the column use `NA` for this purpose. Because missing and illegible responses are normally rare, they are given a different and simpler estimation procedure for their surprisal values.

`scoreList` Either a list of length n, each containing a vector of length `M_i` that assigns numeric weights to the options for that item. In the special case of multiple choice items where the correct option has weight 1 and all others weight 0, a single integer can identify the correct answer. If all the items are of the multiple type, scoreList may be a numeric vector of length n containing the right answer indices. List object scoreList is mandatory because these weights define the person scores for the surprisal curve estimation process.

`noption` A numeric vector of length ncontaining the number of choices for each item. These should not count missing or illegal choices. Although this object might seem redundant, it is needed for checking the consistencies among other objects and as an aid for detecting missing and illegal choices.

`sumscr_rng` A numeric vector of length two containing the initial and final values for the interval over which test scores are to be plotted. Default is minimum and maximum sum score.

`titlestr` A title string for the data and their analyses. Default is NULL.

`itemlabvec` A character value containing labels for the items. Default is NULL and item position numbers are used.

`optlabList` A list vector of length `n`, each element `i` of which is a character vector of length `M_i`. Default is NULL, and option numbers are used.

`nbin` The number of bins for containing proportions of examinees choosing options. The default is computed by a function that uses the number of examinees.

`NumBasis` The number of spline basis functions used to represent surprisal curves. The default is computed by a function that uses the number of examinees.

`jitterwrd` A boolian constant: TRUE implies adding a small random value to each sum score value prior to computing percent rank values.

`PcntMarkers` Used in plots of curves to display marker or reference percentage points for abscissa values in plots.

`verbose` If TRUE details of calculations are displayed.

Details

The score range defined `scrrng` should contain all of the sum score values, but can go beyond their boundaries if desired. For example, it may be that no examinee gets a zero sum score, but for reporting and display purposes using zero as the lower limit seems desirable.

The number of bins is chosen so that a minimum of at least about 25 initial percentage ranks fall within a bin. For larger samples, the number per bin is also larger, making the proportions of choice more accurate. The number bins can be set by the user, or by a simple algorithm used to adjust the number of bins to the number `N` or examinees.

The number of spline basis functions used to represent a surprisal curve should be small for small sample sizes, but can be larger when larger samples are involved.

There must be at least two basis functions, corresponding to two straight lines. The norder of this simple spline would not exceed 1, corresponding to taking only a single derivative of the resulting spline. But this rule is bent here to allow higher higher derivatives, which will autmatically have values of zero, in order to allow these simple linear basis functions to be used. This permits direct comparisons of TestGardener models with the many classic item response models that use two or less parameters per item response curve.

Adding a small value to discrete values before computing ranks is considered a useful way of avoiding any biasses that might arise from the way the data are stored. The small values used leave the rounded jittered values fixed, but break up ties for sum scores.

It can be helpful to see in a plot where special marker percentages 5, 25, 50, 75 and 95 percent of the interval [0,100] are located. The median abscissa value is at 50 per cent for initial percent rank values, for example, but may not be located at the center of the interval after iterations of the analysis cycle.

Value

A named list with named members as follows:

`chcemat:` A matrix of response data with N rows and n columns where N is number of examinees or respondents and n is number of items. Entries in the matrices are the indices of the options chosen. Column i of chcemat is expected to contain only the integers `1,...,noption`.

`optList:` A list vector containing the numerical score values assigned to the options for this question.

key:	If the data are from a test of the multiple choices type where the right answer is scored 1 and the wrong answers 0, this is a numeric vector of length n containing the indices the right answers. Otherwise, it is NULL.
Sfd:	A fd object for the defining the surprisal curves.
noption:	A numeric vector of length n containing the numbers of options for each item.
nbin:	The number of bins for binning the data.
scrrng:	A vector of length 2 containing the limits of observed sum scores.
scrfine:	A fine mesh of test score values for plotting.
scrvec:	A vector of length N containing the examinee or respondent sum scores.
itemvec:	A vector of length n containing the question or item sum scores.
percntrnk:	A vector length N containing the sum score percentile ranks.
thetaQnt:	A numeric vector of length 2*nbin + 1 containing the bin boundaries alternating with the bin centers. These are initially defined as `seq(0,100,len=2*nbin+1)`.
Sdim:	The total dimension of the surprisal scores.
PcntMarkers:	The marker percentages for plotting: 5, 25, 50, 75 and 95.
grbg:	A logical vector of length number of questions. TRUE for an item indicates that a garbage option must be added to the score values, and FALSE indicates that there are no illegal or missing responses and the number of options is equal to number of score values.

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

See Also

TG_analysis, Analyze, index_distn, index2info, index_fun, Sbinsmth

Examples

```
#  Example 1:  Input choice data and key for the short version of the
#  SweSAT quantitative multiple choice test with 24 items and 1000 examinees
#  input the choice data as 1000 strings of length 24
#  set up index and key data
chcemat <- Quant_13B_problem_chcemat
key     <- Quant_13B_problem_key
# number of examinees and of items
N <- nrow(chcemat)
n <- ncol(chcemat)
```

```
# number of options per item and option weights
noption <- rep(0,n)
for (i in 1:n) noption[i]  <- 4
scoreList <- list() # option scores
for (item in 1:n){
  scorei <- rep(0,noption[item])
  scorei[Quant_13B_problem_key[item]] <- 1
  scoreList[[item]] <- scorei
}
# Use the input information to define the
# big three list object containing info about the input data
dataList <- make_dataList(chcemat, scoreList, noption)
```

`mu` *Compute the expected test score by substituting probability of choices for indicator variable 0-1 values. Binary items assumed coded as two choice items.*

Description

Compute the expected test score by substituting probability of choices for indicator variable 0-1 values. Binary items assumed coded as two choice items.

Usage

```
mu(index, SfdList, scoreList)
```

Arguments

`index`	Initial values for score indices in the interval [0,100]. A vector of size `N`.
`SfdList`	A numbered list object produced by a TestGardener analysis of a test. Its length is equal to the number of items in the test or questions in the scale. Each member of `SfdList` is a named list containing information computed during the analysis.
`scoreList`	A numbered list of length n. Each member contains the weights assigned to each option for that item or question.

Value

A vector of test score values.

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Siberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Siberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

See Also

scoreDensity

Examples

```
#  Example 1.  Compute expected sum score values for the
#  short SweSAT multiple choice test with 24 items and 1000 examinees
scoreList <- Quant_13B_problem_dataList$scoreList
SfdList   <- Quant_13B_problem_parmList$SfdList
index     <- Quant_13B_problem_parmList$index
muvec     <- mu(index, SfdList, scoreList)
par(c(1,1))
hist(muvec,11)
```

---

mu_plot *Plot expected test score as a function of score index*

Description

The expected score `mu(index)` is a function of the score index index. A diagonal dashed line is displayed to show the linear relationship to the score range interval.

Usage

```
mu_plot(mufine, scrrng, titlestr)
```

Arguments

`mufine`	A mesh of 101 equally spaced values of mu as a function of index.
`scrrng`	A vector of length 2 containing the score range.
`titlestr`	A string containing the title of the data.

Value

A gg or `ggplot` object defining the plot of the expected test score `mu` as a function of the score index `index`. This is displayed by the `print` command. The plot is automatically displayed as a side value even if no return object is specified in the calling statement.

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

See Also

`scoreDensity`, `mu`

`Power_plot` *Plot item power curves for selected items or questions.*

Description

Item surprisal power curves are the square root of the sum over options of the squared surprisal sensitivity curves.

Usage

```
Power_plot(scrfine, SfdList, Qvec, dataList, plotindex=1:n,
           plotrange=c(min(scrfine),max(scrfine)), height=0.5,
           value=0, ttlsz=NULL, axisttl=NULL, axistxt=NULL)
```

Arguments

`scrfine`	A vector of length nfine (usually 101) containing equally spaced points spanning the plotrange. Used for plotting.
`SfdList`	A numbered list object produced by a TestGardener analysis of a test. Its length is equal to the number of items in the test or questions in the scale. Each member of `SfdList` is a named list containing information computed during the analysis.
`Qvec`	The five marker percentile values.
`dataList`	A list vector containing objects essential to an analysis.
`plotindex`	A set of integers specifying the numbers of the items or questions to be displayed.
`plotrange`	A vector of length 2 containing the plot boundaries within or over the score index interval c(0,100).
`height`	A positive real number defining the upper limit on the ordinate for the plots.
`value`	Number required by ggplot2. Defaults to 0.
`ttlsz`	Title font size.
`axisttl`	Axis title font size.
`axistxt`	Axis text(tick label) font size.

Details

A surprisal power curve for each question indexed in the `index` argument. A request for a keystroke is made for each question. The answer to question strongly defines the optimal position of an estimated score index value where the curve is high value. Values of power curves typically range over [0,0.5].

Value

The plots of the power curves specified in `plotindex` are produced as a side effect. If `saveplot` is TRUE, the plots of item power curves specified in `plotindex` are bundled into a single postscript or .pdf file and the file name is defined by `paste(dataList$titlestr,i,'-power.pdf',sep="")`. The file is then outputas a returned value.

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

See Also

`Sensitivity_plot`, `Entropy_plot`, `Ffuns_plot`, `ICC_plot`

Examples

```
#  Example 1.  Display the item power curves for the
#  short SweSAT multiple choice test with 24 items and 1000 examinees
#  plot the power curve for the first item
dataList <- Quant_13B_problem_dataList
SfdList  <- Quant_13B_problem_parmList$SfdList
Qvec     <- Quant_13B_problem_parmList$Qvec
scrfine <- seq(0,100,len=101)
oldpar <- par(no.readonly=TRUE)
Power_plot(scrfine, SfdList, Qvec, dataList, plotindex=1)
par(oldpar)
```

Quant_13B_problem_chcemat
Test data for 24 math calculation questions from the SweSAT data.

Description

These data are for a randomly selected subset of 1000 examinees.

Usage

```
Quant_13B_problem_chcemat
```

Format

A matrix object with 1000 rows and 24 columns. The integers indicate which answer was chosen for each question by the examinee associated with the row.

Quant_13B_problem_dataList
List of objects essential for an analysis of the abbreviated SweSAT Quantitative multiple choice test.

Description

The data are for 1000 randomly selected examinees taking 24 math analysis multiple choice questions.

Usage

```
Quant_13B_problem_dataList
```

Format

A named list.

Details

A named list with 19 members:

chcemat: A matrix of response data with N rows and n columns where N is the number of examinees or respondents and n is the number of items. Entries in the matrices are the indices of the options chosen. Column i of chcemat is expected to contain only the integers `1,...,noption`.

key: If the data are from a test of the multiple choices type where the right answer is scored 1 and the wrong answers 0, this is a numeric vector of length n containing the indices the right answers. Otherwise, it is NULL.

titlestr: A string containing a title for the analysis.

N: The number of persons tested

n: The number of questons or items

noption: A numeric vector of length n containing the numbers of options for each item.

Sdim: The total dimension of the surprisal scores.

grbgvec: A vector of length indicating which option for each item contains missing or illegal choice values. If 0, there is no such option.

ScoreList: A list vector or length n with each object a numeric vector of weights assigned to each option for each item.

nbin: The number of bins for binning the data.

NumBasis: The number of spline basis functions.

Sbasis: An basisfd object for the defining the surprisal curves.

itemlabvec: A character vector with a title string for each item.

optlabList: A list vector of length n with a character vector of labels for each object within each item.

scrvec: A vector of length N containing the examinee or respondent sum scores.

itmvec: A vector of length n containing the item sum scores.

scrjit: A numeric vector of length N containing small jitters to each sum score to break up ties,

sumscr_rng: A vector of length 2 containing the limits of observed sum scores.

SfdList: A list vector containing essential objects for each item.

scrfine: A fine mesh of test score values for plotting.

indexQnt: A numeric vector of length 2*nbin + 1 containing the bin boundaries alternating with the bin centers. These are initially defined as `seq(0,100,len=2*nbin+1)`.

percntrnk: A vector length N containing the sum score percentile ranks.

PcntMarkers: The marker percentages for plotting: 5, 25, 50, 75 and 95.

`Quant_13B_problem_infoList`
Arclength or information parameter list for 24 items from the quantitative SweSAT subtest.

Description

The data are for 1000 examinees randomly selected from those who took the 2013 quantitative subtest of the SweSAT university entrance exam. The questions are only the 24 math analysis questions, and each question has four options. The analysis results are after 10 cycles of alternating between estimating surprisal curves and estimating percentile score index values. The objects in list object `Quant_13B_problem_infoList` are required for plotting results over the arc length or information domain rather the score index domain. This domain is preferred because such plots are invariant with respect to changes in the score index domain. It also has a metric structure so that differences are comparable no matter where they fall within the information domain.

Usage

```
Quant_13B_problem_infoList
```

Format

A named list containing eight objects.

Value

The object `Quant_13B_problem_parmList` is a named list with these members:

infoSurp: The total length of the information domain measured in M-bits, where `M` is the number of options for a question.

Sfd: The log derivative functional data object defining a strictly increasing set of arc length values corresponding to set of score index values.

infoSurpvec: A mesh of equally-spaced values of indefinite integrals of sum of norms of surprisal derivatives.

scopevec The N arc length values corresponding to the N estimated score index values assigned to N examinees.

Qinfovec: The arc length positions corresponding to the marker percentages 5, 25, 50, 75 and 95.

index: A vector of score index values resulting from using function `monfd` with equally spaced arc length values and `Sfd.info`.

Sdim: The dimension of the over space containing the surprisal pcurves.

`Quant_13B_problem_key` *Option information for the short form of the SweSAT Quantitative test.*

Description

A vector that contains the indices of the right answers among the options for the 24 questions

Usage

```
Quant_13B_problem_key
```

Quant_13B_problem_parmList

Parameter list for 24 items from the quantitative SweSAT subtest.

Description

The data are for 1000 examinees randomly selected from those who took the 2013 quantitative subtest of the SweSAT university entrance exam. The questions are only the 24 math analysis questions, and each question has four options. The analysis results are after 10 cycles of alternating between estimating surprisal curves and estimating percentile score index values.

Usage

```
Quant_13B_problem_parmList
```

Format

A named list.

Value

The object `Quant_13B_problem_parmList` is a named list with these members:

- `index:` A vector of length `N` of estimated values of the percentile rank score index.
- `indexQnt:` A vector of length 2*nbin + 1 containing bin boundaries alternating with bin centres.
- `SfdList:` A list vector of length equal to the number of questions. Each member contains eight results for the surprisal curves associated with a question.
- `logdensfd:` A functional data object representing the logarithm of the density of the percentile rank score index values.
- `C:` The norming constant: the density function is `exp(logdensfd)/C`.
- `densfine:` A fine mesh of probability density values of the percentile rank score index.
- `denscdf:` A fine mesh of cumulative probability distribution values used for interpolating values.
- `Qvec:` The score index values associated with the five marker percentages 5, 25, 50, 75 and 95.
- `binctr:` A vector of length nbin containing the centres of the bins.
- `bdry:` A vector of length nbin+1 containing the boundaries of the bins.
- `freq:` An nbin by M matrix of frequencies with which options are chosen.
- `Smax:` A maximum surprisal value used for plotting purposes.
- `Hval:` The value of the fitting criterion `H` for a single examinee or respondent.
- `DHval:` The value of the first derivative of the fitting criterion `H` for a single examinee or respondent.

`D2Hval:`	The value of the second derivative of the fitting criterion `H` for a single examinee or respondent.
`active:`	A logical vector of length N indicating which estimates of index are converged (FALSE) or not converged (TRUE).
`infoSurp:`	The length in bits of the test information curve.
`infofine:`	A mesh of 101 equally spaced positions along the test information curve.
`Qinfovec:`	The positions of the five marker percentages on the test information curve.
`scopevec:`	A vector of length N containing the positions of each examinee or respondent on the test information curve.

`Sbinsmth` *Estimate the option probability and surprisal curves.*

Description

The surprisal curves for each item are fit to the surprisal transforms of choice probabilities for each of a set of bins of current performance values `index`. The error sums of squares are minimized by the surprisal optimization `smooth.surp` in the `fda` package. The output is a list vector of length `n` containing the functional data objects defining the curves.

Usage

```
Sbinsmth(index, dataList, indexQnt=seq(0,100, len=2*nbin+1),
         wtvec=matrix(1,n,1), iterlim=20, conv=1e-4, dbglev=0)
```

Arguments

`index`	A vector of length N containing current values of score index percentile values.
`dataList`	A list that contains the objects needed to analyse the test or rating scale.
`indexQnt`	A vector of length 2*n+1 containing the sequence of bin boundary and bin centre values.
`wtvec`	A vector of length `n` of weights on observations. Defaults to all ones.
`iterlim`	The maximum number of iterations used in optimizing surprisal curves. Defaults to 20.
`conv`	Convergence tolerance. Defaults to 0.0001.
`dbglev`	Level of output within `Sbinsmth`. If 0, no output, if 1 the error sum of squares and slope on each iterations, and if 2 or higher, results for each line search iteration with function `lnsrch`.

Details

The function first bins the data in order to achieve rapid estimation of the option surprisal curves. The argument `indexQnt` contains the sequence of bin boundaries separated by the bin centers, so that it is of length `2*nbin + 1` where `nbin` is the number of bins. These bin values are distributed over the percentile interval [0,100] so that the lowest boundary is 0 and highest 100. Prior to the call to `Sbinsmth` these boundaries are computed so that the numbers of values of `index` falling in the bins are roughly equal. It is important that the number of bins be chosen so that the bins contain at least about 25 values.

After the values of `index` are binned, the proportions that the bins are chosen for each question and each option are computed. Proportions of zero are given NA values.

The positive proportions are then converted to surprisal values where surprisal = -log_M (proportion) where log_M is the logarithm with base M, the number of options associated with a question. Bins with zero proportions are assigned a surprisal that is appropriately large in the sense of being in the range of the larger surprisal values associated with small but positive proportions. This surprisal value is usually about 4.

The next step is to fit the surprisal values for each question by a functional data object that is smooth, passes as closely as possible to an option's surprisal values, and has values consistent with being a surprisal value. The function `smooth.surp()` is used for this purpose. The arc length of thme item information curve is also computed.

Finally the curves and other results for each question are saved in object `SfdList`, a list vector of length n, and the list vector is returned.

Value

The optimized numbered list object `SfdList` with length `n` that provides data on the probability and surprisal data and curves. The 12 objects for each item are as follows:

- `Sfd:` A surprisal functional data object that is used for plotting. It also contains the coefficient matrix and functional data basis that define the object.
- `M:` The number of options, including if needed a final option which is for the missing and illegitimate responses.
- `Pbin:` A `nbin` by `M` matrix of proportions of choice for each option.
- `Sbin:` A `nbin` by `M` matrix of surprisal values for each option..
- `indfine:` A fine mesh of 101 equally spaced score index values over the interval [0,1].
- `Pmatfine:` A 101 by `M` matrix of probability values at each of the fine mesh points `indfine`.
- `Smatfine:` A 101 by `M` matrix of surprisal values at each of the fine mesh points `indfine`.
- `DSmatfine:` A 101 by `M` matrix of surprisal first derivative values at each of the fine mesh points `indfine`.
- `D2Smatfine:` A 101 by `M` matrix of surprisal second derivative values at each of the fine mesh points `indfine`.
- `PSrsErr:` The standard error for probability over the fine mesh.
- `PSrsErr:` The standard error for surprisal over the fine mesh.
- `itemScope:` The length of the item info curve.

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

See Also

ICC_plot, Sbinsmth

Examples

```
#  Example 1.  Display the initial probability and surprisal curves for the
#  first item in the short SweSAT multiple choice test with 24 items and
#  1000 examinees.
#  Note: The scope is 0 at this point because it is computed later
#  in the analysis.
dataList <- Quant_13B_problem_dataList
index    <- dataList$percntrnk
#  Carry out the surprisal smoothing operation
SfdResult   <- Sbinsmth(index, dataList)
  ## Not run:
  #  Set up the list object for the estimated surprisal curves
  SfdList     <- SfdResult$SfdList
  #  The five marker percentage locations for (5, 25, 50, 75, 95)
  binctr      <- dataList$binctr
  Qvec        <- dataList$PcntMarkers
  #  plot the curves for the first question
  scrfine   <- seq(0,100,len=101)
  ICC_plot(scrfine, SfdList, dataList, Qvec, binctr,
            data_point = TRUE, plotType = c("S", "P"),
            Srng=c(0,3), plotindex=1)

## End(Not run)
```

Sbinsmth_nom *List vector containing numbers of options and boundaries.*

Description

Set up objects needed for analyses of nominal data.

Usage

```
Sbinsmth_nom(bdry_nom, SfdList_nom)
```

Arguments

`bdry_nom`	Vector of length two containing the initial and final values of the scofre index.
`SfdList_nom`	A list vector of length equal to number of items. Each object is a list object containing the containing number of options and the nominal parameter matrix estimated by the mirt package.

Details

Called twice.

`Scope_plot` *Plot the score index* `index` *as a function of arc length.*

Description

Arc length or scope is the distance along the space curved traced out as score index `index` increases from 0 to 100. It is measured in bits and is remains unchanged if the score index continuum is modified.

Usage

```
Scope_plot(infoSurp, infoSurpvec, titlestr=NULL)
```

Arguments

`infoSurp`	This is a positive real number indicating the total length of the space curve. It is expressed in terms of numbers of bits.
`infoSurpvec`	A vector of length 101 containing equally-spaced arc-length distances along the test information curve.
`titlestr`	A string for the title of the data.

Value

A `gg` or `ggplot` object defining the plot of infoSurp along the test information curve as a function of the score index `index`. This is displayed by the `print` command. The plot is automatically displayed as a side value even if no return object is specified in the calling statement.

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

See Also

index2info

Examples

```
#
#  Example 1.  Display the arc length curve for the
#  SweSAT multiple choice test with 24 items and 1000  examinees
#
infoSurpvec <- Quant_13B_problem_infoList$infoSurpvec
infoSurp    <- Quant_13B_problem_infoList$infoSurp
oldpar <- par(no.readonly=TRUE)
Scope_plot(infoSurp, infoSurpvec)
on.exit(oldpar)
```

---

scoreDensity *Compute and plot a score density histogram and and curve.*

Description

The tasks of function `index.density()` and plotting the density are combined. The score density is plotted both as a histogram and as a smooth curve. All the score types may be plotted: sum scores, expected test scores, percentile score index values, and locations on the test information or scale curve. The plot is output as a ggplot2 plot object, which is actually plotted using the `print` command.

Usage

```
scoreDensity(scrvec, scrrng=c(0,100), ndensbasis=15, ttlstr=NULL, pltmax=0)
```

Arguments

`scrvec`	A vector of strictly increasing bin boundary values, with the first at the lowest plotting value and the last at the upper boundary. The number of bins in the histogram is one less than the number of bndry values.
`scrrng`	A vector of length 2 containing lower and upper boundaries on scores, which defaults to c(0,100).
`ndensbasis`	The number of spline basis functions to be used to represent the smooth density curve.
`ttlstr`	A string object used as a title for the plot. Defaults to none.
`pltmax`	An upper limit on the vertical axis for plotting. Defaults to the maximum curve value.

Value

A ggplot2 plot object `dens.plot` that can be displayed using command `print(dens.plot)`.

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

See Also

index_fun, index2info, mu, index_distn

Examples

```
#  Example 1.  Display probability density curves for the
#  short SweSAT multiple choice test with 24 items and 1000 examinees
SfdList <- Quant_13B_problem_parmList$SfdList
index   <- Quant_13B_problem_parmList$index
Qvec    <- Quant_13B_problem_parmList$Qvec
#  plot the density for the score indices within interval c(0,100)
index_int <- index[0 < index & index < 100]
oldpar <- par(no.readonly=TRUE)
scoreDensity(index_int)
par(oldpar)
```

---

scorePerformance — *Calculate mean squared error and bias for a set of score index values from simulated data.*

Description

After the simulated data matrices have been analyzed, prepare the objects necessary for the performance plots produced by functions `RMSEbias1.plot` and `RMSEbias2.plot`.

Usage

```
scorePerformance(dataList, simList)
```

Arguments

`dataList` A list that contains the objects needed to analyse the test or rating scale with the following fields:

chcemat: A matrix of response data with N rows and n columns where N is the number of examinees or respondents and n is the number of items. Entries in the matrices are the indices of the options chosen. Column i of chcemat is expected to contain only the integers `1,...,noption`.

optList: A list vector containing the numerical score values assigned to the options for this question.

key: If the data are from a test of the multiple choices type where the right answer is scored 1 and the wrong answers 0, this is a numeric vector of length n containing the indices the right answers. Otherwise, it is NULL.

Sfd: An fd object for the defining the surprisal curves.

noption: A numeric vector of length n containing the numbers of options for each item.

nbin: The number of bins for binning the data.

scrrng: A vector of length 2 containing the limits of observed sum scores.

scrfine: A fine mesh of test score values for plotting.

scrvec: A vector of length N containing the examinee or respondent sum scores.

itemvec: A vector of length n containing the question or item sum scores.

percntrnk: A vector length N containing the sum score percentile ranks.

chcematQnt: A numeric vector of length 2*nbin + 1 containing the bin boundaries alternating with the bin centers. These are initially defined as `seq(0,100,len=2*nbin+1)`.

Sdim: The total dimension of the surprisal scores.

PcntMarkers: The marker percentages for plotting: 5, 25, 50, 75 and 95.

`simList` A named list containing these objects:

sumscr: A matrix with row dimension `nchcemat`, the number of population score index values and column dimension `nsample`, the number of simulated samples.

chcemat: An `nchcemat` by `nsample` of estimated score index values.

mu: An `nchcemat` by `nsample` of estimated expected score values.

al: An `nchcemat` by `nsample` of estimated test information curve values.

thepop: A vector of population score index values.

mupop: A vector of expected scores computed from the population score index values.

alpop: A vector of test information values computed from the population score index values.

n: The number of questions.

Qvec: The five marker percentile values.

Value

A named list containing these objects:

sumscr: A matrix with row dimension `nchcemat`, the number of population score index values and column dimension `nsample`, the number of simulated samples.

chcemat: An `nchcemat` by `nsample` matrix of estimated score index values.

mu: An `nchcemat` by `nsample` matrix of estimated expected score values.

al: An `nchcemat` by `nsample` matrix of estimated test information curve values.

chcepop: A vector of population score index values.

mupop: A vector of expected scores computed from the population score index values.

infopop: A vector of test information values computed from the population score index values.

n: The number of questions.

Qvec: The five marker percentile values.

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

See Also

dataSimulation

Sensitivity_plot *Plots all the sensitivity curves for selected items or questions.*

Description

A sensitivity curve for an option is the first derivative of the corresponding surprisal curve. Its values can be positive or negative, and the size of the departure from zero at any point on the curve is the amount information contributed by that curve to locating the value of an examinee or respondent on the score index continuum.

Usage

```
Sensitivity_plot(scrfine, SfdList, Qvec, dataList, plotindex=1:n,
                 plotrange=c(min(scrfine),max(scrfine)),
                 key=NULL, titlestr=NULL, saveplot=FALSE, width=c(-0.2,0.2),
                 ttlsz=NULL, axisttl=NULL, axistxt=NULL, lgdlab=NULL)
```

Arguments

`scrfine`	A vector of length nfine (usually 101) containing equally spaced points spanning the plotrange. Used for plotting.
`SfdList`	A numbered list object produced by a TestGardener analysis of a test. Its length is equal to the number of items in the test or questions in the scale. Each member of `SfdList` is a named list containing information computed during the analysis.
`Qvec`	The values of the five marker percentiles.
`dataList`	A list that contains the objects needed to analyse the test or rating scale.
`plotindex`	A set of integers specifying the numbers of the items or questions to be displayed.
`plotrange`	A vector of length 2 containing the plot boundaries within or over the score index interval c(0,100).

key	A integer vector of indices of right answers. If the data are rating scales, this can be NULL.
titlestr	A title string for plots.
saveplot	A logical value indicating whether the plot should be saved to a pdf file.
width	A vector of length 2 defining the lower and upper limits on the ordinate for the plots.
ttlsz	Title font size.
axisttl	Axis title font size.
axistxt	Axis text(tick label) font size.
lgdlab	Legend label font size.

Details

Sensitivity curves for each question indexed in the index argument. A request for a keystroke is made for each question.

Value

A list vector is returned which is of the length of argument plotindex. Each member of the vector is a gg or ggplot object for the associated plotindex value. Each plot can be displayed using the print command. The plots of item power are produced as a side value even if no output object is specified in the call to the function.

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

See Also

Power_plot, Entropy_plot, ICC_plot

Examples

```
#  Example 1.  Display the option sensitivity curves for the
#  short SweSAT multiple choice test with 24 items and 1000 examinees.
dataList <- Quant_13B_problem_dataList
SfdList  <- Quant_13B_problem_parmList$SfdList
Qvec     <- Quant_13B_problem_parmList$Qvec
scrfine  <- seq(0,100,len=101)
oldpar   <- par(no.readonly=TRUE)
Sensitivity_plot(scrfine, SfdList, Qvec, dataList, plotindex=1)
par(oldpar)
```

`SimulateData` *Simulate Choice Data from a Previous Analysis*

Description

Simulation of data using a previous analysis requires only an ICC vector and two objects computed by function `theta.distn` along with a specification of the number of simulated the simulated persons.

Usage

```
SimulateData(nsim, indfine, denscdf, SfdList)
```

Arguments

`nsim`	Number of persons having simulated choices.
`indfine`	The score index values within [0,100] that are associated with the cumulative probability values in `denscdf`.
`denscdf`	The cumulative probability values within [0,1]. The values have to be discrete, begin with 0 and end with 1.
`SfdList`	List vector of length n of list vectors for item objects.

Details

Arguments `indfine` and `denscdf` can be obtained from the original analysis, but also can be specified to describe a different distribution of score index values.

Value

An nsim by n matrix of integers including 1 and 2 that specify each person's option choice for each item.

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

See Also

`dataSimulation`, `chcemat_simulate`

Examples

```
# example code to be set up
```

---

smooth.ICC *Smooth binned probability and surprisal values to make an* ICC *object.*

Description

An N by n matrix of positive integer choice index values is transformed to an nbin by M matrix of probability values by iteravely minimizing the sum of squared errors for bin values.

Usage

```
smooth.ICC(x, item, index, dataList, indexQnt=seq(0,100, len=2*nbin+1),
                        wtvec=matrix(1,n,1), iterlim=20, conv=1e-4, dbglev=0)
```

Arguments

x	An ICC object
item	Index of item being set up.
index	A vector of length N containing score index values for each person.
dataList	A list object set up by function `make.dataList` containing objects set up prior to an analysis of the data.
indexQnt	A vector of length 2*nbin + 1 containing, in sequence, the lower boundary of a bin, its midpoint, and the upper boundary.
wtvec	A vector of length n containing wseights for items.
iterlim	An integer specifying the maximum number of optimizations.
conv	A convergence criterion a little larger than 0.
dbglev	One of integers 0 (no optimization information), 1 (one line per optimization) or 2 (complete optimization display).

Value

An S3 class ICC object for a single item.

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

Examples

```
# example code to be set up
```

---

smooth.surp *Fit data with surprisal smoothing.*

Description

Surprisal is -log(probability) where the logarithm is to the base being the dimension `M` of the multinomial observation vector. The surprisal curves for each question are estimated by fitting the surprisal values of binned data using curves whose values are within the `M-1`-dimensional surprisal subspace that is within the space of non-negative M-dimensional vectors.

Usage

```
smooth.surp(argvals, y, Bmat0, Sfd, Zmat, wtvec=NULL, conv=1e-4,
            iterlim=50, dbglev=0)
```

Arguments

`argvals`	Argument value array of length N, where N is the number of observed curve values for each curve. It is assumed that that these argument values are common to all observed curves. If this is not the case, you will need to run this function inside one or more loops, smoothing each curve separately.
`y`	A `nbin` by `M_i` matrix of surprisal values to be fit.
`Bmat0`	A `Snbasis` by `M_i - 1` matrix containing starting values for the iterative optimization of the least squares fit of the surprisal curves to the surprisal data.
`Sfd`	A functional data object used toinitialize the optimization process.
`Zmat`	An `M` by `M-1` matrix satisfying `Zmat'Zmat <- I}` and `\code{Zmat'1 <- 0`.
`wtvec`	A vector of weights to be used in the smoothing.
`conv`	A convergence criterion.
`iterlim`	the maximum number of iterations allowed in the minimization of error sum of squares.
`dbglev`	Either 0, 1, or 2. This controls the amount information printed out on each iteration, with 0 implying no output, 1 intermediate output level, and 2 full output. If either level 1 or 2 is specified, it can be helpful to turn off the output buffering feature of S-PLUS.

Value

A named list of class `surpFd` with these members:

`PENSSE`	The final value of the penalized fitting criterion.
`DPENSSE`	The final gradient of the penalized fitting criterion.

D2PENSSE	The final hessian of the fitting criterion.
SSE	The final value of the error sum of squares.
DSSE	The final gradient of the error sum of squares.
D2SSE	The final hessian of the error sum of squares.
DvecSmatDvecB	The final cross derivative DvecSmatDvecX times DvecXmatDvecB of the surprisal curve and the basis coordinates.

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

See Also

eval.surp, ICC_plot, Sbinsmth

Examples

```
oldpar <- par(no.readonly=TRUE)
#  evaluation points
x <- seq(-2,2,len=11)
#  evaluate a standard normal distribution function
p <- pnorm(x)
#  combine with 1-p
mnormp <- cbind(p,1-p)
M <- 2
#  convert to surprisal values
mnorms <- -log2(mnormp)
#  plot the surprisal values
matplot(x, mnorms, type="l", lty=c(1,1), col=c(1,1),
        ylab="Surprisal (2-bits)")
# add some log-normal error
mnormdata <- exp(log(mnorms) + rnorm(11)*0.1)
#  set up a b-spline basis object
nbasis <- 7
sbasis <- create.bspline.basis(c(-2,2),nbasis)
#  define an initial coefficient matrix
cmat <- matrix(0,7,1)
#  set up a fd object for suprisal smoothing
Sfd <- fd(cmat, sbasis)
Zmat <- matrix(c(1,-1),2,1)
#  smooth the noisy data
result <- smooth.surp(x, mnormdata, cmat, Sfd, Zmat)
#  plot the data and the fits of the two surprisal curves
```

```
xfine <- seq(-2,2,len=51)
sfine <- eval.surp(xfine, result$Sfd, Zmat)
matplot(xfine, sfine, type="l", lty=c(1,1), col=c(1,1))
points(x, mnormdata[,1])
points(x, mnormdata[,2])
#  convert the surprisal fit values to probabilities
pfine <- 2^(-sfine)
#  check that they sum to one
apply(pfine,1,sum)
par(oldpar)
```

Spca *Functional principal components analysis of information curve*

Description

A test or scale analysis produces a space curve that varies with in the space of possible option curves of dimension `Sdim`. Fortunately, it is usual that most of the shape variation in the curve is within only two or three dimensions, and these can be fixed by using functional principal components analysis.

Usage

```
Spca(SfdList, nharm=2, Sdim=NULL, rotate=TRUE)
```

Arguments

`SfdList`	A numbered list object produced by a TestGardener analysis of a test. Its length is equal to the number of items in the test or questions in the scale. Each member of `SfdList` is a named list containing information computed during the analysis.
`Sdim`	Interval over which curve is plotted. All if Sdim == NULL.
`nharm`	The number of principal components of the test information or scale curve to be used to display the curve. Must be either 2 or 3.
`rotate`	If true, rotate principal components of the test information or scale curve to be used to display the curve to VARIMAX orientation.

Value

A named list with these members:

`harmvarmxfd`	Functional data objects for the principal components of the curve shape.
`varpropvarmx`	Proportions of variance accounted for by the principal components

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

See Also

Spca_plot

Examples

```
#  Example 1.  Display the test information curve for the
#  short SweSAT multiple choice test with 24 items and 1000 examinees
#  plot a two-dimension version of manifold curve
Sdim     <- Quant_13B_problem_dataList$Sdim
SfdList  <- Quant_13B_problem_parmList$SfdList
index    <- Quant_13B_problem_parmList$index
infoSurp <- Quant_13B_problem_parmList$infoSurp
#      <- Quant_13B_problem_dataList$Sdim
oldpar <- par(no.readonly=TRUE)
on.exit(oldpar)
Results <- Spca(SfdList, nharm=2, rotate=FALSE)
varprop <- Results$varpropvarmx
print("Proportions of variance accounted for and their sum:")
print(round(100*c(varprop,sum(varprop)),1))
#  plot a three-dimension version of manifold curve
SfdList   <- Quant_13B_problem_parmList$SfdList
index     <- Quant_13B_problem_parmList$index
infoSurp  <- Quant_13B_problem_parmList$infoSurp
Results   <- Spca(SfdList, nharm=3, rotate=FALSE)
varprop   <- Results$varpropvarmx
print("Proportions of variance accounted for and their sum:")
print(round(100*c(varprop,sum(varprop)),1))
```

Spca_plot — *Plot the test information or scale curve in either two or three dimensions.*

Description

A test or scale analysis produces a space curve that varies with in the space of possible option curves of dimension Sdim. Fortunately, it is usual that most of the shape variation in the curve is within only two or three dimensions, and these can be fixed by using functional principal components analysis.

Usage

```
Spca_plot(harmvarmxfd, nharm=2, titlestr=NULL)
```

Arguments

harmvarmxfd	Functional data objects for the principal components of the curve shape.
nharm	Number of principal components.
titlestr	A string for the title of the plot. Defaults to NULL.

Value

Side effect is a two or three-dimensional plot of the principal component approximation of the information curve using the plotly package. Function `plot_ly` does not return a value, but does render the graphic.

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

See Also

Spca

Examples

```
#  Example 1.  Display the test information curve for the
#  short SweSAT multiple choice test with 24 items and 1000 examinees
#  plot a two-dimension version of manifold curve
SfdList    <- Quant_13B_problem_parmList$SfdList
index    <- Quant_13B_problem_parmList$index
arclength  <- Quant_13B_problem_parmList$arclength
Results    <- Spca(SfdList,  nharm=2, rotate=TRUE)
varprop    <- Results$varpropvarmx
titlestr   <- "SweSAT problem items"
oldpar <- par(no.readonly=TRUE)
on.exit(oldpar)
Spca_plot(Results$harmvarmxfd, nharm=2, titlestr)
print("Proportions of variance accounted for and their sum:")
print(round(100*c(varprop,sum(varprop)),1))
#  plot a three-dimension version of manifold curve
SfdList    <- Quant_13B_problem_parmList$SfdList
index    <- Quant_13B_problem_parmList$index
arclength  <- Quant_13B_problem_parmList$arclength
Results    <- Spca(SfdList,  nharm=3, rotate=TRUE)
varprop    <- Results$varpropvarmx
Spca_plot(Results$harmvarmxfd, nharm=3, titlestr)
print("Proportions of variance accounted for and their sum:")
```

```
print(round(100*c(varprop,sum(varprop)),1))
```

TestGardener *Analyses of Tests and Rating Scales using Information or Surprisal*

Description

TestGardener is designed to permit the analysis of choice data from multiple choice tests and rating scales using information as an alternative to the usual models based on probability of choice.

Probabability and information are related by the simple transformation "information = -log probability". Another term for information is "surprisal."

The advantage of information methodology, often used in the engineering and physical sciences, is that measurabe, and therefore is on what is called a "ratio scale" in the social sciences. That is, information or rurprisal has a lower limit of zero, is unbounded above, and can be added, subtracted and rescaled with a positive multiplier.

The disadvantage of probability as a basis for representing choice is that differences near its two boundaries are on very different scales than those near 0.5, and our visual and other sensory systems, which are adapted to mangitudes, have many problems in assessing the nonlinear probability continuum.

TestGardener uses highly adaptable and computationally efficient spline basis functions to represent item characteristic curves for both probability and surprisal. Splines bases permit as much flexibility as the task requires, and also can control the smoothness and the order of differentiation.

The higher variability revealed by information or surprisal curves reveals many more insights into choice behavior than the usual simple curve employed in standard probability-based item response theory.

The use of information as a measure also implies a measure of inter-item covariation called mutual entropy. Entropy a function whose value at any point is the average across surprisal curves produced by summing over curves for a given item of the product of probability and surprisal.

Graphical display is a large part of the TestGardener capacity, with extensive use of the ggplot2 and plotly packages.

TestInfo_svd *Image of the Test Tnformation Curve in 2 or 3 Dimensions*

Description

The test information curve is the trajectory of joint variation of all the surprisal curves within the ambient space of dimension the total number of curves. But usually a very high percent of the shape variation in the curve can be represented in either two or three dimensions using the singular value decomposition of a matrix of total curve values over a fine mesh. The resulting approximation is converted to a set of surprisal curve values.

Usage

```
TestInfo_svd(scrfine, SfdList, itemindex=1:n, nharm=2)
```

Arguments

`scrfine`	A fine mesh of values over which the image is plotted. This is usually either the score index theta or the test arc length.
`SfdList`	A list vector of length n, the number of test items. Each list in the vector contains values of the surprisal curves for that item.
`itemindex`	A vector of item indices to be used in the approximation.
`nharm`	The number of dimension in the approximation, usually either two or three.

Value

The approximation is returned as a surprisal functional data object, and so are the percentages of the total variation fit by each dimension in the approximation.

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

`TG_analysis` *Statistics for Multiple choice Tests, Rating Scales and Other Choice Data)*

Description

Given an choice ingter-valued index matrix and a vector of numbers of item options, the function cycles through a set of iterations involving surprisal curve estimation followed by test taker index values.

Usage

```
TG_analysis(chcemat, scoreList, noption, sumscr_rng=NULL,
            titlestr=NULL, itemlabvec=NULL, optlabList=NULL,
            nbin=nbinDefault(N), NumBasis=7, NumDensBasis=7,
            jitterwrd=TRUE, PcntMarkers=c( 5, 25, 50, 75, 95),
            ncycle=10, itdisp=FALSE, verbose=FALSE)
```

Arguments

`chcemat`	An N by n matrix. Column i must contain the integers from 1 to M_i, where M_i is the number of options for item i. If missing or illegitimate responses exist for item i, the column must also contain an integer greater than M_i that is used to identify such responoses. Alternatively, the column use NA for this purpose. Because missing and illegible responses are normally rare, they are given a different and simpler estimation procedure for their surprisal values. U is mandatory.
`scoreList`	Either a list of length n, each containing a vector of length M_i that assigns numeric weights to the options for that item. In the special case of multiple choice items where the correct option has weight 1 and all others weight 0, a single integer can identify the correct answer. If all the items are of the multiple type, scoreList may be a numeric vector of length n containing the right answer indices. List object scoreList is mandatory because these weights define the person scores for the surprisal curve estimation process.
`noption`	A numeric vector of length `n` containing the number of options for each item.
`sumscr_rng`	A vector of length 2 indicating the initial and final sum score values. Default is NULL the whole sum score is used.
`titlestr`	A title string for the data and their analyses. Default is NULL.
`itemlabvec`	A character value containing labels for the items. Default is NULL and item position numbers are used.
`optlabList`	A list vector of length n, each element i of which is a character vector of length M_i. Default is NULL, and option numbers are used.
`nbin`	The number of bins containing proportions of choices.
`NumBasis`	The number of spline basis functions to use for surprisal values. Defaults to 7.
`NumDensBasis`	The number of spline basis functions to use for score probability density function. Defaults to 7.
`jitterwrd`	A logical object indicating whether a small jittering perturbation should be used to break up ties. Defaults to TRUE.
`PcntMarkers`	A vector of percentages inside of [0,100] that appear in plots. Defaults to c(5, 25, 50, 75, 95). Extra displays are provided. Defaults to FALSE.
`ncycle`	The number of cycles in the analysis. Defaults to 10.
`itdisp`	Display results for function `theta_fun`.
`verbose`	Extra displays are provided. Defaults to FALSE.

Details

This function in package TestGardener processes at a mininum two objects: (1) A matrix `chcemat` that contains indices of choices made in a sequence of choice situations (its number columns `n`) by a set of persons making the choices (its number of rows `N`); and (2) A list vector `scoreList` of length `n` containing numerical weights or scores for each choice available with in each of `n` choice situations (referred to as `items`).

The function returns three large lists containing objects that can be used to assess: (1) the probability that a choice will be made, and (2) the quantity of information, called `surprisal`, that the choice made reveals about the performance or experience of the person making the choice.

Value

Four list objects, each containing objects that are required for various displays, tables and other results:

`parmList` A list object containing objects useful for displaying results that involve the score index cotninuum:

- SfdList: A list object of length n, each containing objects for an item for displaying that item's surprisal curves as defined by the score index values after the analysis. See the help page for function `Analyze` for a description of these objects.
- Qvec: A vector containing the positions on the score index continuum of the marker percentages defined in the arguments of function `make_dataList()`.
- binctr: A vector of length `nbin` containing the positions on the score index continuum of the bin centres.
- indexScore: A vector of length `N` containing the positions on the score index contiuum of each person.
- infoSurp: The length of the test or scale information continuum in M-bits.

`infoList` A list object containing objects useful for displaying results that involve the scale information cotninuum:

- infofine: A fine mesh of 101 values that is used to plot the scale information comntinuum.
- scopevec: A vector of length `N` containing the positions on the scale information contiuum of each person.
- Qinfovec: A vector containing the positions on the scale information continuum of the marker percentages defined in the arguments of function `make_dataList()`.
- infobinctr: A vector of length `nbin` containing the positions on the scale information continuum of the bin centres.

`HALsave` A `ncycle` by 2 matrix containing in the first column the data fit values and in the second column the infoSurp or arclength values.

`analysisListvec`
A list of length `ncycle` containing the `parmList` object for each cycle.

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

See Also

`make_dataList`, `Analyze`, `index_distn`, `index2info`, `index_fun`, `Sbinsmth`

Examples

```
## Not run:
  #  Example 1:  Input choice data and key for the short version of the
  #  SweSAT quantitative multiple choice test with 24 items and 1000 examinees
  #  input the choice indices in the 1000 by 24 choice index matrix
  chcemat <- Quant_13B_problem_chcemat
  #  set up the key data
  key     <- Quant_13B_problem_key
  # number of examinees and of items
  N <- nrow(chcemat)
  n <- ncol(chcemat)
  # number of options per item and option weights
  noption <- rep(0,n)
  for (i in 1:n) noption[i]  <- 4
  scoreList <- list() # option scores
  for (item in 1:n){
    scorei <- rep(0,noption[item])
    scorei[key[item]] <- 1
    scoreList[[item]] <- scorei
  }
  #  Analyze the data and return the big three list objects:
  #  dataList (input info), parmList (score index info) and
  #  infoList (information or scope info)
  TGresult <- TG_analysis(chcemat, scoreList, noption,
                          NumBasis=4, ncycle=10, verbose=TRUE)
## End(Not run)
```

`TG_density.fd` *Compute a Probability Density Function*

Description

Like the regular S-PLUS function `density`, this function computes a probability density function for a sample of values of a random variable. However, in this case the density function is defined by a functional parameter object `logdensfdPar` along with a normalizing constant `C`.

The density function $p(indexdens)$ has the form `p(indexdens) = C exp[W(indexdens)]` where function $W(indexdens)$ is defined by the functional data object `logdensfdPar`.

Usage

```
## S3 method for class 'fd'
TG_density(indexdens, logdensfd, conv=0.0001, iterlim=20,
            active=1:nbasis, dbglev=0)
```

Arguments

`indexdens` a set observations, which may be one of two forms:

1. a vector of observations $indexdens_i$

2. a two-column matrix, with the observations $indexdens_i$ in the first column, and frequencies f_i in the second.

The first option corresponds to all $f_i = 1$.

`logdensfd` a functional data object specifying the initial value, basis object, roughness penalty and smoothing parameter defining function $W(t).$

`conv` a positive constant defining the convergence criterion.

`iterlim` the maximum number of iterations allowed.

`active` a logical vector of length equal to the number of coefficients defining `Wfdobj`. If an entry is TRUE, the corresponding coefficient is estimated, and if FALSE, it is held at the value defining the argument `Wfdobj`. Normally the first coefficient is set to 0 and not estimated, since it is assumed that $W(0) = 0$.

`dbglev` either 0, 1, or 2. This controls the amount information printed out on each iteration, with 0 implying no output, 1 intermediate output level, and 2 full output. If levels 1 and 2 are used, it is helpful to turn off the output buffering option in S-PLUS.

Details

The goal of the function is provide a smooth density function estimate that approaches some target density by an amount that is controlled by the linear differential operator `Lfdobj` and the penalty parameter. For example, if the second derivative of $W(t)$ is penalized heavily, this will force the function to approach a straight line, which in turn will force the density function itself to be nearly normal or Gaussian. Similarly, to each textbook density function there corresponds a $W(t)$, and to each of these in turn their corresponds a linear differential operator that will, when apply to $W(t)$, produce zero as a result. To plot the density function or to evaluate it, evaluate `Wfdobj`, exponentiate the resulting vector, and then divide by the normalizing constant `C`.

Value

a named list of length 4 containing:

`Wfdobj` a functional data object defining function $W(indexdens)$ that that optimizes the fit to the data of the monotone function that it defines.

`C` the normalizing constant.

`Flist` a named list containing three results for the final converged solution: (1) **f**: the optimal function value being minimized, (2) **grad**: the gradient vector at the optimal solution, and (3) **norm**: the norm of the gradient vector at the optimal solution.

`iternum` the number of iterations.

`iterhist` a `iternum+1` by 5 matrix containing the iteration history.

Author(s)

Juan Li and James Ramsay

References

Ramsay, J. O., Li J. and Wiberg, M. (2020) Full information optimal scoring. Journal of Educational and Behavioral Statistics, 45, 297-315.

Ramsay, J. O., Li J. and Wiberg, M. (2020) Better rating scale scores with information-based psychometrics. Psych, 2, 347-360.

See Also

intensity.fd

Index

∗ **datasets**
Quant_13B_problem_chcemat, 37
Quant_13B_problem_dataList, 37
Quant_13B_problem_infoList, 38
Quant_13B_problem_key, 39
Quant_13B_problem_parmList, 40

Analyze, 3, *24*, *32*, *60*

chcemat_simulate, 6, *50*

dataSimulation, 7, *48*, *50*
density_plot, 8
DFfun, 10, *19*, *27*

entropies, 11
Entropy_plot, *12*, 12, *22*, *36*, *49*
eval.surp, 14, *53*

Fcurve, 15
Ffun, *10*, *17*, 17, *19*, *27*
Ffuns_plot, *10*, *13*, *17*, 18, *36*

ICC, 19, *22*
ICC_plot, *13*, 20, *36*, *43*, *49*, *53*
index2info, *5*, 23, *26*, *27*, *32*, *45*, *46*, *60*
index_distn, *5*, 25, *27*, *32*, *46*, *60*
index_fun, *5*, *10*, *17*, *19*, *26*, 26, *29*, *32*, *46*, *60*
index_search, *16*, 28
intensity.fd, *63*

make_dataList, *5*, *10*, *17*, 30, *60*
mu, *26*, 33, *35*, *46*
mu_plot, 34

Power_plot, *13*, *22*, 35, *49*

Quant_13B_problem_chcemat, 37
Quant_13B_problem_dataList, 37
Quant_13B_problem_infoList, 38
Quant_13B_problem_key, 39
Quant_13B_problem_parmList, 40

Sbinsmth, *5*, *22*, *32*, 41, *43*, *53*, *60*
Sbinsmth_nom, 43
Scope_plot, 44
scoreDensity, *9*, *26*, *27*, *34*, *35*, 45
scorePerformance, *8*, 46
Sensitivity_plot, *13*, *22*, *36*, 48
SimulateData, 50
smooth.ICC, 51
smooth.surp, *15*, 52
Spca, 54, *56*
Spca_plot, *55*, 55

TestGardener, 57
TestInfo_svd, 57
TG_analysis, *5*, *32*, 58
TG_density.fd, 61