Package 'bspam'

December 20, 2024

Type Package

Encoding UTF-8

Title Speed-Accuracy Psychometric Modeling for Binomial Count Outcome Data with R

Version 1.2.5

Author Akihito Kamata [aut, cre] (<https://orcid.org/0000-0001-9570-1464>),
Cornelis J. Potgieter [aut] (<https://orcid.org/0000-0002-1995-6817>),
Joseph F. T. Nese [aut] (<https://orcid.org/0000-0002-9878-7395>),
Yusuf Kara [aut] (<https://orcid.org/0000-0003-0691-0630>),
Sarunya Somsong [aut] (<https://orcid.org/0000-0002-4974-7346>),
Xin Qiao [aut] (<https://orcid.org/0000-0002-3248-7859>),
Kuo Wang [aut] (<https://orcid.org/0000-0002-4925-0591>)

Maintainer Akihito Kamata <akamata@gmail.com>

Description This is an R package that contains functions to fit the speed-accuracy psychometric model for count outcome data (Potgieter, Kamata & Kara, 2017; Kara, Kamata, Potgieter & Nese, 2020), where the accuracy is modeled by a binomial count latent variable model. For example, the use of this modeling technique allows model-based calibration and scoring for oral reading fluency (ORF) assessment data.

License GPL (>= 3)

Depends R (>= 3.5.0)

Imports tibble, rootSolve, tidyverse, tidyr, magrittr, doParallel, foreach, futile.logger, mvtnorm, nleqslv, tryCatchLog, parallel, miscTools, psych, rstan

Suggests rjags, testthat (>= 3.5.0)

URL https://github.com/kamataak/bspam

BugReports https://github.com/kamataak/bspam/issues

RoxygenNote 7.3.2

LazyData yes

Config/testthat/edition 3

NeedsCompilation no

R topics documented:

`bspam-package` *bspam : A package for fitting the speed-accuracy psychometric model for repeatedly measured count outcome data.*

Description

Purpose of this package

'bspam' is an R package that contains functions to fit the speed-accuracy psychometric model for repeatedly measured count outcome data (Potgieter, Kamata & Kara, 2017; Kara, Kamata, Potgieter & Nese, 2020), where the accuracy is modeled by a binomial count latent variable model. For example, the use of this modeling technique allows model-based calibration and scoring for oral reading fluency (ORF) assessment data.

Design philosophy

Write Design philosophy

Author(s)

Akihito Kamata <akamata@smu.edu>

Cornelis J. Potgieter <c.potgieter@tcu.edu>

Joseph F. T. Nese <jnese@uoregon.edu>

Yusuf Kara <ykara@mail.smu.edu>

Sarunya Somsong <sarunya.ss@hotmail.com>

Xin Qiao <xqiao@usf.edu>
Kuo Wang <wangkuo@nifty.com>

`bayes` *Bayes function when running mcem with bayes setting*

Description

Bayes function when running mcem with bayes setting

Usage

```
bayes(
  person.data = NA,
  person.id = "",
  task.id = "",
  max.counts = "",
  obs.counts = "",
  time = "",
  parallel = T,
  n.chains = NA,
  thin = 1,
  iter = NA,
  burn = NA
)
```

Arguments

`person.data`	- student reading data
`person.id`	The column name in the data that represents the unique individual identifier.
`task.id`	The column name in the data that represents the unique task identifier.
`max.counts`	The column name in the data that represents the number of words in a task.
`obs.counts`	The column name in the data that represents the words read correctly for each case.
`time`	The column name in the data that represents the time, in seconds, for each case.
`parallel`	parallel=T, #logical, run in parallel? "T" or "F"
`n.chains`	int., number of the chains
`thin`	int, thinning interval, a.k.a, period of saving samples
`iter`	int., number of the iterations after the burn-in period
`burn`	int., number of the burn-in iterations

Value

list

`bayes.wcpm` *Bayes function when running mcem with bayes setting*

Description

Bayes function when running mcem with bayes setting

Usage

```
bayes.wcpm(
  calib.data = NA,
  person.data = NA,
  person.id = NULL,
  task.id = NULL,
  occasion = NULL,
  group = NULL,
  max.counts = NULL,
  obs.counts = NULL,
  time = NULL,
  cases = NULL,
  external = NULL,
  type = NULL,
  parallel = T,
  n.chains = NA,
  iter = NA,
  burn = NA,
  thin = 1
)
```

Arguments

`calib.data`	- fit.model class object
`person.data`	- individual reading data
`person.id`	The column name in the data that represents the unique individual identifier.
`task.id`	The column name in the data that represents the unique task identifier.
`occasion`	The column name in the data that represents the unique occasion.
`group`	The column name in the data that represents the unique group.
`max.counts`	The column name in the data that represents the number of words in a task.
`obs.counts`	The column name in the data that represents the words read correctly for each case.
`time`	The column name in the data that represents the time, in seconds, for each case.
`cases`	- student id vectors, will directly use passage data if no calib.data provided
`external`	- if not NULL, will use not student read passages for estimating
`type`	- output type, "general" and "orf", default "general" only output tau & theta. "orf" will output wcpm
`parallel`	parallel=T, #logical, run in parallel? "T" or "F"
`n.chains`	int., number of the chains

iter	int., number of the iterations after the burn-in period
burn	int., number of the burn-in iteration
thin	int, thinning interval, a.k.a, period of saving samples

Value

list

`create_data_list` *Scoring.Passage function*

Description

Usage

```
create_data_list(Count, logT10, MaxN, a, b, alpha, beta, sigma, rho, C)
```

Arguments

a,	b: Model parameters related to the count data model (K-dim)
alpha,	beta: Model parameters related to the time data model (K-dim)
Count:	A vector with the number of words correct per passage It should be K-dimensional
logT10:	The log-scale reading time per 10 words per passage It should be K-dimensional
MaxN:	A vector of passage lengths It should be K-dimensional
sigma:	The latent standard deviation of the time latent component
rho:	The correlation between count and time latent components
C:	A vector of indicators whether a specific passage was censored (1) or fully observed (0) – K-dim

Details

Pre-processing data: Function prepares data for Stan model Enter a single student's data for preparation

Assume a student has read K >= 2 passages

`create_data_list_sentence` *Call by scoring.sentence.censoring function*

Description

Usage

```
create_data_list_sentence(
  Count,
  logT10,
  MaxN,
  Passage,
  a,
  b,
  alpha,
  beta,
  gamma1,
  gamma2,
  sigma,
  rho,
  rhoTestlet,
  C
)
```

Arguments

`a,`	b: Model parameters related to the count data model (K-dim)
`alpha,`	beta: Model parameters related to the time data model (K-dim)
`gamma1,`	gamma2: Hyperparameters related to the testlet model
`Count:`	A vector with the number of words correct per passage It should be K-dimensional
`logT10:`	The log-scale reading time per 10 words per passage It should be K-dimensional
`N:`	A vector of passage lengths It should be K-dimensional
`Passage:`	Task ids
`sigma:`	The latent standard deviation of the time latent component
`rho:`	The correlation between count and time latent components
`rhoTestlet:`	The testlet-based reliability
`C:`	A vector of indicators whether a specific sentence was censored (1) or fully observed (0) – K-dim

Details

will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

A copy of the GNU General Public License is available at http://www.gnu.org/licenses/

Pre-processing data: Function prepares data for Stan model Enter a single student's data for preparation

Assume a student has read K >= 2 passages

Value

list

`fit.model` *Estimate the model parameters*

Description

This is an interface function to estimate the model parameters based on the task-level (i.e., passage-level in ORF assessment context) accuracy and speed data by implementing the Monte Carlo EM algorithm described in Potgieter et al. (2017) or fully Bayesian method described in Kara et al. (2020).

Usage

```
fit.model(
  data = NA,
  person.id = "",
  task.id = "",
  sub.task.id = "",
  max.counts = "",
  obs.counts = "",
  time = "",
  k.in = 5,
  reps.in = 2,
  ests.in = NA,
  est = "mcem",
  se = "none",
  verbose = FALSE,
  testlet = FALSE
)
```

Arguments

`data`	A data frame or a data generated by the `prep` function. If this argument is used with prepared data, the next five arguments `person.id`, `task.id`,`max.counts`, `obs.counts`, `time` should be skipped.
`person.id`	Quoted variable name in `data` that indicates the unique person identifier.
`task.id`	Quoted variable name in `data` that indicates the unique task identifier. In the ORF assessment context, it is the passage identifier.

sub.task.id	Quoted variable name in data that indicates the unique sub task identifier. In the ORF assessment context, it is the sentence identifier.
max.counts	Quoted variable name in data that indicates the number of attempts in the task. In the ORF assessment context, it is the number of words in the passage.
obs.counts	Quoted variable name in data that indicates the number of successful attempts in each task. In the ORF assessment context, it is the number of words read correctly for the passage.
time	Quoted variable name in data that indicates the time in seconds took to complete the task. In the ORF context, it is the time took to complete reading the passage.
k.in	Numeric, indicating the number of imputations. Default is 5.
reps.in	Numeric, indicating the number of Monte-Carlo iterations. Default is 2.
ests.in	An optional list of numeric vectors, indicating initial values of the model parameters. If this argument is not given, mom function will be called to generate the initial values.
est	Quoted string, indicating the choice of the estimator. It has to be one of "mcem", "bayes". Default is "mcem".
se	Quoted string, indicating the choice of the standard errors. It has to be one of "none", "analytic", "bootstrap". Default is "none".
verbose	Boolean. If TRUE, the summary will be output. Default is FALSE.
testlet	Boolean. If TRUE, the fit.model.testlet will be run. Default is FALSE.

Details

Additional details...

Value

MCEM list, bayes list

Note

More & more additional note...

References

Potgieter, N., Kamata, A., & Kara, Y. (2017). An EM algorithm for estimating an oral reading speed and accuracy model. Manuscript submitted for publication.

Kara, Y., Kamata, A., Potgieter, C., & Nese, J. F. (2020). Estimating model-based oral reading fluency: A bayesian approach with a binomial-lognormal joint latent model. Educational and Psychological Measurement, 1–25.

See Also

scoring for scoring.

Examples

```
# example code
MCEM_run <- fit.model(data = passage2,
                      person.id = "id.student",
                      task.id = "id.passage",
                      max.counts = "numwords.pass",
                      obs.counts = "wrc",
                      time = "sec",
                      k.in = 5,
                      reps.in = 50,
                      est = "mcem")
```

`fit.model.testlet`

Description

Usage

```
fit.model.testlet(
  data = NULL,
  person.id = "",
  sub.task.id = "",
  obs.counts = "",
  time = "",
  task.id = "",
  max.counts = ""
)
```

Arguments

`data`	A data frame. It has the information of student, passage, sentence, obs.counts and time.
`person.id`	each student's id.
`sub.task.id`	each sentence's id.

obs.counts	The column name in the data that represents the words read correctly for each sentence
time	The column name in the data that represents the reading time for the sentence.
task.id	The column name in the data that represents the unique passage identifier.
max.counts	The column name in the data that represents the number of words in a sentence.

Details

Additional details...

Value

list

Note

Additional note...

Examples

```
# example code
fit.model.testlet <- function(data=NULL, person.id="", sub.task.id="",obs.counts="", time="", task.id="", max
```

---

get.cases *Returns cases (person and occasion) applied in [fit.model] function.*

Description

Returns cases (person and occasion) applied in [fit.model] function.

Usage

```
get.cases(data)
```

Arguments

data	= person response data

Value

cases vector

getBootstrapSE *Get bootstrap SE case is a single stu_season_id*

Description

Added MAP function 07/14/2021 Modified a bug of MLE 07/23/2021 Modified EAP 10/28/2021

Usage

```
getBootstrapSE(
  object,
  person.data,
  case = NA,
  perfect.cases,
  est = "map",
  kappa = 1,
  bootstrap = 100,
  external = NULL
)
```

Arguments

object	- mcem class object
person.data	- individual response data
case	- case number
perfect.cases	- perfect accurate case
est	- SE type.(MLE, EAP, and MAP.) default MAP
kappa	- Default kappa = 1, better be 5
bootstrap	- K number of bootstrap, default is 100
external	- if not NULL, will use unread task for estimating

Value

SE dataset

passage.calib.bayes *Task calibration example output object by Bayes*

Description

This is an example output object from running the `fit.model.byaes` function by using the Bayesian estimator option. It is a result of calibrating an oral reading fluency data set from Nese and Kamata (2014-2018) with xxxx students on 150 passages.

Usage

```
passage.calib.bayes
```

Format

two lists: $pass.param with 150 rows and $hyper.param with 4 variables

$pass.param

a parameter controlling binomial success probabilities

b parameter controlling binomial success probabilities

alpha parameter controlling reading times

beta parameter controlling reading times

se_a standard error of a

se_b standard error of b

se_alpha standard error of alpha

se_beta standard error of beta

passage.id passage ID

nwords.p the total

$hyper.param

vartau variance of latent reading ability tau

rho correlation between two latent variables

se_vartau standard error of vartar

se_rho standard error of rho

Source

https://jnese.github.io/core-blog/

References

Nese, J. F. T. & Kamata, A. (2014-2018). Measuring Oral Reading Fluency: Computerized Oral Reading Evaluation (Project No. R305A140203) [Grant]. Institute of Education Sciences, U.S. Department of Education. https://ies.ed.gov/funding/grantsearch/details.asp?ID=1492

---

passage.calib.mcem *Task calibration example output object by MCEM*

Description

This is an example output object from running the `fit.model` function by using the MCEM estimator option. It is a result of calibrating an oral reading fluency data set from Nese and Kamata (2014-2018) with xxxx students on 150 passages.

Usage

```
passage.calib.mcem
```

Format

A list of two elements: `$pass.param` is a dataframe with 150 rows and 10 variables, and `$hyper.param` is a dataframe with 1 row and 4 variables

`$pass.param`

`a` parameter controlling binomial success probabilities

`b` parameter controlling binomial success probabilities

`alpha` parameter controlling reading times

`beta` parameter controlling reading times

`se_a` standard error of a

`se_b` standard error of b

`se_alpha` standard error of alpha

`se_beta` standard error of beta

`passage.id` passage ID

`nwords.p` the total number of words in the passage

`$hyper.param`

`vartau` variance of latent reading ability tau

`rho` correlation between two latent variables

`se_vartau` standard error of vartar

`se_rho` standard error of rho

Source

https://jnese.github.io/core-blog/ Nese, J. F. T. & Kamata, A. (2014-2018). Measuring Oral Reading Fluency: Computerized Oral Reading Evaluation (Project No. R305A140203) [Grant]. Institute of Education Sciences, U.S. Department of Education. https://ies.ed.gov/funding/grantsearch/details.asp?ID=1492

`passage2` *Passage-level Oral Reading Fluency assessment data set*

Description

This is an example data set. It is a passage-level Oral Reading Fluency assessment data set for 85 students who were assigned to read 2 to 12 passages among the same 12 passages. The data is a small subset of the data collected by Nese and Kamata (2014-2018).

Usage

```
\code{data(passage2)}
```

Format

A data frame with 847 rows and 7 variables.

`id.student` unique student identifier

`occasion` identifier for longitudinal assessment occasions; here a triannual assessment administered in the fall, winter, and spring of a school year

`grade` student grade level

`id.passage` unique passage identifier

`numwords.pass` total number of words in the passage

`wrc` words read correct

`sec` seconds to read the passage

Source

https://jnese.github.io/core-blog/

References

Nese, J. F. T. & Kamata, A. (2014-2018). Measuring Oral Reading Fluency: Computerized Oral Reading Evaluation (Project No. R305A140203) [Grant]. Institute of Education Sciences, U.S. Department of Education. https://ies.ed.gov/funding/grantsearch/details.asp?ID=1492

---

plot.person *Plot function to show graph of scoring class*

Description

Usage

```
## S3 method for class 'person'
plot(object, person = NULL, parameter, show.se = T, sort = F)
```

Arguments

`object`	scoring object
`person`	person ids for plotting
`parameter`	model parameter for plotting, a,b,alpha,beta
`show.se`	standard error bar flag, TRUE for showing or FALSE for no showing
`sort`	sorting flag, TRUE or FALSE

Details

plot.task *Plot function to show graph of fit.model class*

Description

Copyright (C) 2021-2024 The ORF Project Team

Usage

```
## S3 method for class 'task'
plot(object, task = NULL, parameter, sort = F)
```

Arguments

object	fit.model object
task	task ids for plotting
parameter	model parameter for plotting, a,b,alpha,beta
sort	sorting flag, TRUE or FALSE

Details

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. A copy of the GNU General Public License is available at http://www.gnu.org/licenses/

prep *This file includes utilities of bspam package.*

Description

Copyright (C) 2021-2023 The ORF Project Team

Usage

```
prep(
  data = data,
  person.id = "",
  task.id = "",
  occasion = "",
  group = "",
  max.counts = "",
  obs.counts = "",
  time = "",
  sentence_level = FALSE
)
```

Arguments

data	= student response data
person.id	Quoted variable name in person.data that indicates the unique person identifier.
task.id	Quoted variable name in person.data that indicates the unique task identifier. In the ORF assessment context, it is the passage identifier.
occasion	The column name in the data that represents the unique occasion.
group	The column name in the data that represents the unique group.
max.counts	Quoted variable name in person.data that indicates the number of attempts in the task. In the ORF assessment context, it is the number of words in the passage.
obs.counts	Quoted variable name in person.data that indicates the number of successful attempts in each task. In the ORF assessment context, it is the number of words read correctly for the passage.
time	Quoted variable name in person.data that indicates the time in seconds took to complete the task. In the ORF context, it is the time took to complete reading the passage.
sentence_level	flag for sentence or passage level data, default is FALSE

Details

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. A copy of the GNU General Public License is available at http://www.gnu.org/licenses/

prep function prepares input data for fit.model function

Value

data list (data.long: data frame, data.wide: list of Y, logT10, N, I)

---

run.mcem *Base function for fitting the speed-accuracy model by MCEM*

Description

This is a base function for fitting the speed-accuracy model to estimate the model parameters by the Monte Carlo EM algorithm described in Potgieter et al. (2017). This function is used by the fit.model function, which is recommended for the users to use.

Usage

```
run.mcem(Y, logT10, N, I, k.in = 5, reps.in = 2, ests.in = NA, verbose = FALSE)
```

Arguments

Y	Numeric data matrix of accuracy scores in the size of n x I, where n is the number of tasks and I is the number of persons. Missing data are allowed.
logT10	Numeric data matrix of time in the size of n x I matrix, where n and I are defined above. The time data should be in the scale of log10(times). Missing data are allowed.
N	Numeric vector of passage lengths in the length of n.
I	Numeric, indicating the number of tasks.
k.in	Numeric, indicating the number of imputations. Default is 5.
reps.in	Numeric, indicating the number of Monte-Carlo iterations. Default is 2.
ests.in	An optional list of numeric vectors, indicating initial values of the model parameters. If this argument is not given, mom function will be called to generate the initial values.
verbose	Boolean. If TRUE, the summary will be output. Default is FALSE.

Details

If the user is desired to use this function, note that the response data file needs to be in the wide format, which can be reshaped from a long-format response data by the prep.wide function.

Value

mcem list

Note

Update Memo: 10/24/2024 For test 04/29/2021 Modified the mcem function based on Nelis's updated. 10/28/2021 Modified the mcem output

References

Potgieter, N., Kamata, A., & Kara, Y. (2017). An EM algorithm for estimating an oral reading speed and accuracy model. Manuscript submitted for publication.

See Also

prep.wide fit.model.

---

run.scoring *Base function for scoring by likelihood-based approaches*

Description

This is a base function for estimating factor scores by likelihood-based approaches described in Potgieter et al. (2024). This function is used by the scoring function, which is recommended for the users to use.

Usage

```
run.scoring(
  object,
  person.data,
  task.data,
  cases,
  perfect.cases,
  zero.cases,
  est = "map",
  lo = -12,
  hi = 12,
  q = 100,
  kappa = 1,
  external = NULL,
  type = NULL
)
```

Arguments

object	A class object. Output from calibration phase by fit.model function.
person.data	A data frame. A long-format response data object.
task.data	A data frame? Estimated task parameter values?
cases	A vector of individual id for which scoring is desired. If no information is is specified, it will estimate scores for all cases in the person.data.
perfect.cases	A list? A list of perfect cases.
est	Quoted string, indicating the choice of the estimator. It has to be one of code/"mle", "map", "eap", "bayes". Default is "map".
lo	Numeric, indicating the lower bound of the quadratures. Default is -12.
hi	Numeric, indicating the upper bound of the quadratures. Default is 12.
q	Numeric, indicating the number of quadratures. Default is 100.
kappa	Numeric, indicating ?? Default is 1.
external	An optional vector of task ID's in strings. If NULL (default), the wcpm scores are derived with the tasks the individuals were assigned to. If not NULL, wcpm scores are derived with the tasks provided in the vector, rather than the tasks the individuals were assigned.
type	Quoted string, indication of the choice of output. If "general" (default), wcpm scores are not reported. If "orf", wcpm scores will be reported.

Details

Update Memo: 04/29/2021 Modified the wcpm function based on Nelis's updated. 06/01/2021 Modified based on Nelis's MAP function. 06/01/2021 Modified based on Sarunya's BiEAP function. 06/20/2021 Modified based on Nelis's updated for MLE and EAP 06/21/2021 Modified a bug of MAP function. 07/12/2021 Modified est.eqs function based on Nelis's code. 07/13/2021 Added Map function for bootstrap. 07/30/2021 Modified wcpm function based on Sarunya's update

Value

wcpm list

References

Qiao, X, Potgieter, N., & Kamata, A. (2023). Likelihood Estimation of Model-based Oral Reading Fluency. Manuscript submitted for publication.

`scoring` *Estimate factor scores with task-level model*

Description

This is an interface function to estimated factor scores based on the task-level (i.e., passage-level in ORF assessment context) accuracy and speed data. It implements likelihood-based approaches (MLE, MAP, or EAP) described in Qiao et al. (under review) or fully Bayesian method described in Kara et al. (2020).

Usage

```
scoring(
  calib.data = NA,
  data = NA,
  person.id = "",
  task.id = "",
  sub.task.id = "",
  occasion = "",
  group = "",
  max.counts = "",
  obs.counts = "",
  time = "",
  cens = "",
  cases = NULL,
  est = "map",
  se = "analytic",
  failsafe = 0,
  bootstrap = 100,
  external = NULL,
  type = "general",
  censoring = FALSE,
  testlet = FALSE
)
```

Arguments

`calib.data`	A class object. Output from calibration phase by `fit.model` function
`data`	A data frame. It is necessary when with the next five arguments `person.id`, `task.id`,`max. counts`, `obs. counts`, `time`.
`person.id`	Quoted variable name in `data` that indicates the unique individual identifier.
`task.id`	Quoted variable name in `data` that represents the unique task identifier. In the ORF assessment context, it is the passage identifier.
`sub.task.id`	Quoted variable name in `data` that indicates the unique sub task identifier. In the ORF assessment context, it is the sentence identifier.

occasion	The column name in the data that represents the unique occasion.
group	The column name in the data that represents the unique group.
max.counts	Quoted variable name in data that represents the number of attempts in the task. In the ORF assessment context, it is the number of words in the passage.
obs.counts	The column name in the data that represents the words read correctly for each case.
time	The column name in the data that represents the time, in seconds, for each case.
cens	The column name in the data that represents the censoring indicators whether a specific task or sub task was censored (1) or fully observed (0). This column is necessary when censoring argument is TRUE.
cases	A vector of individual id for which scoring is desired. If no information is is specified, it will estimate scores for all cases in the data.
est	Quoted string, indicating the choice of the estimator. It has to be one of code/"mle", "map", "eap", "bayes". Default is "map".
se	Quoted string, indication the choice of the standard errors. It has to be one of code/"analytic", "bootstrap". Default is "analytic".
failsafe	Numeric, indicating the number of retries for bootstrap, which can be set between 0 and 50. Default is 0.
external	An optional vector of task ID's in strings. If NULL (default), the wcpm scores are derived with the tasks the individuals were assigned to. If not NULL, wcpm scores are derived with the tasks provided in the vector, rather than the tasks the individuals were assigned.
type	Quoted string, indication of the choice of output. If "general" (default), wcpm scores are not reported. If "orf", wcpm scores will be reported.
censoring	Boolean. If TRUE, interface will call task or sub task censoring. Default is FALSE.
testlet	Boolean. If TRUE, runs with sub task level, otherwise, with task level. This argument is necessary when censoring is TRUE. Default is FALSE.
bootstrp	Numeric, indicating the number of bootstrap iterations. Default is 100.

Details

Additional details...

Value

scoring list or Bootstrap dataset or censoring list

Note

More additional note...

References

Qiao, X, Potgieter, N., & Kamata, A. (2023). Likelihood Estimation of Model-based Oral Reading Fluency. Manuscript submitted for publication.

Kara, Y., Kamata, A., Potgieter, C., & Nese, J. F. (2020). Estimating model-based oral reading fluency: A bayesian approach with a binomial-lognormal joint latent model. Educational and Psychological Measurement, 1–25.

See Also

fit.model for model parameter estimation.

Examples

```
# example code
WCPM_all <- scoring(calib.data=MCEM_run,
                    data = passage2,
                    person.id = "id.student",
                    occasion = "occasion",
                    group = "grade",
                    task.id = "id.passage",
                    max.counts = "numwords.pass",
                    obs.counts = "wrc",
                    time = "sec",
                    est = "map",
                    se = "analytic",
                    type="general")
```

---

scoring.passage.censoring

scoring.passage function

Description

Usage

```
scoring.passage.censoring(
  Count = NULL,
  logT10 = NULL,
  N = NULL,
  a = NULL,
  b = NULL,
  alpha = NULL,
  beta = NULL,
  sigma = NULL,
  rho = NULL,
  C = NULL
)
```

Arguments

a,	b: Model parameters related to the count data model (K-dim)
alpha,	beta: Model parameters related to the time data model (K-dim)
Count:	A vector with the number of words correct per passage It should be K-dimensional
logT10:	The log-scale reading time per 10 words per passage It should be K-dimensional
N:	A vector of passage lengths It should be K-dimensional

sigma:	The latent standard deviation of the time latent component
rho:	The correlation between count and time latent components
C:	A vector of indicators whether a specific passage was censored (1) or fully observed (0) – K-dim

Details

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

A copy of the GNU General Public License is available at http://www.gnu.org/licenses/

Pre-processing data: Function prepares data for Stan model Enter a single student's data for preparation

Assume a student has read K >= 2 passages

Value

list

---

scoring.sentence.censoring

scoring.sentence.censoring function

Description

Copyright (C) 2021-2024 The ORF Project Team

Usage

```
scoring.sentence.censoring(
  Count = NULL,
  logT10 = NULL,
  N = NULL,
  Passage = NULL,
  a = NULL,
  b = NULL,
  alpha = NULL,
  beta = NULL,
  gamma1 = NULL,
  gamma2 = NULL,
  sigma = NULL,
  rho = NULL,
  rhoTestlet = NULL,
  C = NULL
)
```

Arguments

a,	b: Model parameters related to the count data model (K-dim)
alpha,	beta: Model parameters related to the time data model (K-dim)
gamma1,	gamma2: Hyperparameters related to the testlet model
Count:	A vector with the number of words correct per passage It should be K-dimensional
logT10:	The log-scale reading time per 10 words per passage It should be K-dimensional
N:	A vector of passage lengths It should be K-dimensional
Passage:	Task ids
sigma:	The latent standard deviation of the time latent component
rho:	The correlation between count and time latent components
rhoTestlet:	The testlet-based reliability
C:	A vector of indicators whether a specific sentence was censored (1) or fully observed (0) – K-dim

Details

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

A copy of the GNU General Public License is available at http://www.gnu.org/licenses/

Pre-processing data: Function prepares data for Stan model Enter a single student's data for preparation

Assume a student has read K >= 2 passages

Value

list

---

sentence.cens.high *Sentence-level data set with high censoring*

Description

This is an example data set. It is a sentence-level Oral Reading Fluency assessment data set with high proportion of censoring data for 58 students who were assigned to read 4 passages with a total of 23 sentences. The data is a small subset of the data collected by Nese and Kamata (2014-2018).

Usage

```
sentence.cens.high
```

Format

1334 rows and 9 variables:

`id.student` unique student identifier

`grade` student grade level

`id.passage` unique passage identifier

`ind.passage` passage index

`id.sentence` unique sentence sequence

`numwords.sent` the total number of words in the sentence

`wrc` the number of words read correct

`sec` time to read the sentence in seconds

`cens` whether a specific passage was censored (0) or fully observed (1)

Source

https://jnese.github.io/core-blog/

References

Nese, J. F. T. & Kamata, A. (2014-2018). Measuring Oral Reading Fluency: Computerized Oral Reading Evaluation (Project No. R305A140203) [Grant]. Institute of Education Sciences, U.S. Department of Education. https://ies.ed.gov/funding/grantsearch/details.asp?ID=1492

---

sentence.cens.low *Sentence-level data set with high censoring*

Description

This is an example data set. It is a sentence-level Oral Reading Fluency assessment data set with low proportion of censoring data for 58 students who were assigned to read 4 passages with a total of 23 sentences. The data is a small subset of the data collected by Nese and Kamata (2014-2018).

Usage

```
sentence.cens.low
```

Format

1334 rows and 9 variables:

`id.student` unique student identifier

`grade` student grade level

`id.passage` unique passage identifier

`ind.passage` passage index

`id.sentence` unique sentence sequence

`numwords.sent` the total number of words in the sentence

`wrc` the number of words read correct

`sec` time to read the sentence in seconds

`cens` whether a specific passage was censored (0) or fully observed (1)

Source

https://jnese.github.io/core-blog/

References

Nese, J. F. T. & Kamata, A. (2014-2018). Measuring Oral Reading Fluency: Computerized Oral Reading Evaluation (Project No. R305A140203) [Grant]. Institute of Education Sciences, U.S. Department of Education. https://ies.ed.gov/funding/grantsearch/details.asp?ID=1492

---

sentence.level.data *Sentence-level Oral Reading Fluency assessment data set*

Description

This is an example data set. It is a sentence-level Oral Reading Fluency assessment data set for 58 students who were assigned to read 4 passages with a total of 23 sentences. The data is a small subset of the data collected by Nese and Kamata (2014-2018).

Usage

```
sentence.level.data
```

Format

1334 rows and 8 variables:

`id.student` unique student identifier

`grade` student grade level

`id.passage` unique passage identifier

`ind.passage` passage index

`id.sentence` unique sentence sequence

`numwords.sent` the total number of words in the sentence

`wrc` the number of words read correct

`sec` time to read the sentence in seconds

Source

https://jnese.github.io/core-blog/

References

Nese, J. F. T. & Kamata, A. (2014-2018). Measuring Oral Reading Fluency: Computerized Oral Reading Evaluation (Project No. R305A140203) [Grant]. Institute of Education Sciences, U.S. Department of Education. https://ies.ed.gov/funding/grantsearch/details.asp?ID=1492

summary.bootstrap *summary the information of bootstrap class*

Description

Copyright (C) 2021-2023 The ORF Project Team

Usage

```
## S3 method for class 'bootstrap'
summary(
  object,
  digits = 4,
  geterror = FALSE,
  verbose = TRUE,
  factor.scores = FALSE
)
```

Arguments

object	= bootstrap object
digits	= print out numeric with specific digits
geterror,	summary error case, default FALSE
verbose	show summary on screen, default TRUE
factor.scores	- theta and tau output flag, default is FALSE

Details

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. A copy of the GNU General Public License is available at http://www.gnu.org/licenses/

Value

table

summary.fit.model *summary the information of fit.model class*

Description

Copyright (C) 2021-2023 The ORF Project Team

Usage

```
## S3 method for class 'fit.model'
summary(object, digits = 4, ...)
```

Arguments

object	= object
digits	= print out numeric with specific digits
...	= parameter

Details

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. A copy of the GNU General Public License is available at http://www.gnu.org/licenses/

Value

printing information

---

summary.scoring *summary the information of wcpm class*

Description

Copyright (C) 2021-2023 The ORF Project Team

Usage

```
## S3 method for class 'scoring'
summary(
  object,
  digits = 4,
  verbose = TRUE,
  factor.scores = TRUE,
  show = "short"
)
```

Arguments

object	= object
digits	= print out numeric with specific digits
verbose	- boolean, if TRUE, shows the summary, default is TRUE
factor.scores	- theta and tau output flag, default is TRUE # before was FALSE
show	- output flag, "long" and "short", default "short" only output estimate result. "long" will output estimate result and data.

Details

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. A copy of the GNU General Public License is available at http://www.gnu.org/licenses/

Value

scoring dataset with task information and estimated score

`testlet_scoring_multi_obs_multi_cens`

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. A copy of the GNU General Public License is available at http://www.gnu.org/licenses/

Description

Bayesian Sampling Models for Passage-Level Observations

Usage

```
testlet_scoring_multi_obs_multi_cens
```

Format

An object of class `character` of length 1.

Details

This code defines Bayesian sampling models for estimating latent variables based on passage-level observations for a specific student. The models are formulated differently depending on the number of censored (incomplete) and fully observed passages. The goal is to estimate latent variables such as accuracy (theta_acc) and speed (theta_spd) for the student's reading performance.

testlet_scoring_multi_obs_multi_cens_sentence

Description

Bayesian Sampling Models for Sentence-Level Observations

Usage

```
testlet_scoring_multi_obs_multi_cens_sentence
```

Format

An object of class `character` of length 1.

Details

This code defines Bayesian sampling models for estimating latent variables based on sentence-level observations for a specific student. The models are formulated differently depending on the number of censored (incomplete) and fully observed passages. The goal is to estimate latent variables such as accuracy (theta_acc) and speed (theta_spd) for the student's reading performance.

Index