

# Package ‘bspam’

December 8, 2023

**Type** Package

**Encoding** UTF-8

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

**Version** 1.2.2

**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

**Suggests** rjags, rstan, testthat (>= 3.5.0)

**URL** <https://github.com/kamataak/bspam>

**BugReports** <https://github.com/kamataak/bspam/issues>

**RoxygenNote** 7.2.3

**LazyData** yes

**Config/testthat/edition** 3

**NeedsCompilation** no

**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>>)

## R topics documented:

bspam-package . . . . .	2
bayes . . . . .	3
bayes.wcpm . . . . .	4

exclude_passages . . . . .	5
fit.model . . . . .	5
fit.model.testlet . . . . .	7
get.cases . . . . .	8
get.perfectcases . . . . .	9
getBootstrapSE . . . . .	9
get_errlist . . . . .	10
log.initiating . . . . .	10
MCEM . . . . .	11
passage.calib.bayes . . . . .	11
passage.calib.mcem . . . . .	12
passage2 . . . . .	13
plot.mcem . . . . .	13
prep . . . . .	14
preplong . . . . .	15
prepwide . . . . .	15
run.mcem . . . . .	16
run.scoring . . . . .	17
scoring . . . . .	18
sentence.data . . . . .	20
summary.bootstrap . . . . .	20
summary.fit.model . . . . .	21
summary.scoring . . . . .	22

## Index 23

---

bspam-package	<i>bspam : A package for fitting the speed-accuracy psychometric model for repeatedly measured count outcome data.</i>
---------------	--

---

## Description

### Purpose of this package

‘bspam’ 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.

### 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 mcm with bayes setting*

---

**Description**

Bayes function when running mcm 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 mcm with bayes setting***Description**

Bayes function when running mcm 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

---

exclude_passages	<i>To exclude error passages</i>
------------------	----------------------------------

---

**Description**

To exclude error passages

**Usage**

```
exclude_passages(passage)
```

**Arguments**

```
passage
```

**Value**

```
passage data set without error passages
```

---

fit.model	<i>Estimate the model parameters</i>
-----------	--------------------------------------

---

**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.data = NA,
  person.id = "",
  task.id = "",
  max.counts = "",
  obs.counts = "",
  time = "",
  k.in = 5,
  reps.in = 2,
```

```

    ests.in = NA,
    est = "mcm",
    se = "none",
    verbose = FALSE
  )

```

### Arguments

<code>data</code>	A data frame. A wide-format response data generated by the <code>prep</code> function. If this argument is used, the next 6 arguments <code>person.data</code> , <code>person.id</code> , <code>task.id</code> , <code>max.counts</code> , <code>obs.counts</code> , <code>time</code> should be skipped.
<code>person.data</code>	A data frame. A long-format response data object. If this this argument is used, the previous argument <code>data</code> should be skipped.
<code>person.id</code>	Quoted variable name in <code>person.data</code> that indicates the unique person identifier.
<code>task.id</code>	Quoted variable name in <code>person.data</code> that indicates the unique task identifier. In the ORF assessment context, it is the passage identifier.
<code>max.counts</code>	Quoted variable name in <code>person.data</code> that indicates the number of attempts in the task. In the ORF assessment context, it is the number of words in the passage.
<code>obs.counts</code>	Quoted variable name in <code>person.data</code> 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.
<code>time</code>	Quoted variable name in <code>person.data</code> 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.
<code>k.in</code>	Numeric, indicating the number of imputations. Default is 5.
<code>reps.in</code>	Numeric, indicating the number of Monte-Carlo iterations. Default is 2.
<code>ests.in</code>	An optional list of numeric vectors, indicating initial values of the model parameters. If this argument is not given, <code>mom</code> function will be called to generate the initial values.
<code>est</code>	Quoted string, indicating the choice of the estimator. It has to be one of "mcm", "bayes". Default is "mcm".
<code>se</code>	Quoted string, indicating the choice of the standard errors. It has to be one of "none", "analytic", "bootstrap". Default is "none".
<code>verbose</code>	Boolean. If TRUE, the summary will be output. 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(person.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 = "mcm")
```

---

fit.model.testlet

*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

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/>

## Usage

```
fit.model.testlet(
  data = NULL,
  id.student = "",
  id.sentence = "",
```

```
wrc = "",
sec = ""
)
```

Arguments

data	A data frame. It has the information of student, passage, sentence, wrcm and sec.
id.student	each student’s id.
id.sentence	each sentence’s id.
wrc	The column name in the data that represents the words read correctly for each sentence
sec	The column name in the data that represents the reading time for the sentence.

Details

Additional details...

Value

list

Note

Additional note...

Examples

```
# example code
fit.testlet <- fit.model.testlet(data=sentence_data, id.student="id.student", id.sentence="id.stntence", wrc
```

---

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



---

get.perfectcases	<i>Returns perfect cases (student and occasion) in which every word was read correctly.</i>
------------------	---

---

**Description**

Returns perfect cases (student and occasion) in which every word was read correctly.

**Usage**

```
get.perfectcases(data)
```

**Arguments**

data                      = person response data

**Value**

perfect accurate case vector

---

getBootstrapSE	<i>Get bootstrap SE case is a single stu_season_id</i>
----------------	--

---

**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	- mcm 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

---

get_errlist	<i>To get a string of error passages that have no at least two readers</i>
-------------	--

---

**Description**

To get a string of error passages that have no at least two readers

**Usage**

get\_errlist(passage)

**Arguments**

passage

**Value**

a string of error passages

---

log.initiating	<i>This file includes the log functions of bspam package.</i>
----------------	---

---

**Description**

Copyright (C) 2021-2023 The ORF Project Team

**Usage**

```
## S3 method for class 'initiating'
log()
```

**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/>

This is log.initiating function, which is used to log output.

---

MCEM

*Passage calibration data set*


---

**Description**

Passage calibration data set

**Usage**

MCEM

**Format**

An object of class `mce` of length 2.

---

`passage.calib.bayes`
*Passage calibration data with bayes*


---

**Description**

A data set calibrated with bayes.

**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 vartau

**se\_rho** standard error of rho

**Source**

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

---

passage.calib.mcem	<i>Passage calibration data with MCEM</i>
--------------------	---

---

**Description**

A data set calibrated with MCEM.

**Usage**

```
passage.calib.mcem
```

**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/>

---

passage2	<i>Passage-level student data set</i>
----------	---------------------------------------

---

**Description**

A data set consisted of reading accuracy and time data for 12 passages from 85 students.

**Usage**

```
passage2
```

**Format**

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/>

---

plot.mcem	<i>Plot function to show graph of mcem class</i>
-----------	--

---

**Description**

Copyright (C) 2021-2023 The ORF Project Team

**Usage**

```
## S3 method for class 'mcem'
plot(object, X)
```

**Arguments**

object	= mcem object
X	= dataset to plot
...	= 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/>

---

```
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 = ""
)
```

## Arguments

data                      = student response 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/>

prep function prepares input data for fit.model function

## Value

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

---

preplong	<i>Prepares data in a long format for [wcpm]</i>
----------	--

---

**Description**

Prepares data in a long format for [wcpm]

**Usage**

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

**Arguments**

data = person response data

**Value**

data frame

---

prepwide	<i>Prepares data in a wide format for [mcm].</i>
----------	--

---

**Description**

This function will return a list with 5 elements: Y: a matrix of words read correctly, where rows represent cases (student and occasion) and columns represent passages logt10: a [tibble::tibble()] of words read correctly N: the number of cases per passage I: the number of passages

**Usage**

```
prepwide(data, person.id, task.id, max.counts, obs.counts, time)
```

**Arguments**

data	A data frame.
time	The column name in the data that represents the time, in seconds, for each case.
studentid	The column name in the data that represents the unique student identifier.
passageid	The column name in the data that represents the unique passage identifier.
numwords.p	The column name in the data that represents the number of words in a passage.
wrc	The column name in the data that represents the words read correctly for each case.

---

run.mcem

*This file includes the Base functions of bspam package.*


---

## Description

Copyright (C) 2021-2023 The ORF Project Team

## Usage

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

## Arguments

Y	= n x I matrix of reading scores – missingness allowed
logT10	= n x I matrix of log10(reading times) – missingness allowed
N	= vector of passage lengths
I	= number of passages
k.in	= number of imputations, default is 5
reps.in	= number of Monte-Carlo iterations, default is 2
ests.in	= if not give, mom function will be called and get est.in output
verbose	- boolean, if shows the summary, 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/>

This is run.mcem function. Update Memo: 04/29/2021 Modified the mcem function based on Nelis's updated. 10/28/2021 Modified the mcem output

## Value

mcem list a,b = parameters controlling binomial success probabilities, each length I alpha,beta = parameters controlling reading times, each length I var\_tau = variance of latent reading ability tau rho = correlation between two latent variables



---

run.scoring	<i>This is run.scoring function.</i>
-------------	--------------------------------------

---

## Description

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

## Usage

```
run.scoring(
  object,
  person.data,
  task.data,
  cases,
  perfect.cases,
  est = "map",
  lo = -4,
  hi = 4,
  q = 100,
  kappa = 1,
  external = NULL,
  type = NULL
)
```

## Arguments

object	= fit.model object, if not be given will occur error and stop running
person.data	= individual response task data
task.data	= estimate parameters data
cases	= individual occasion id vector
perfect.cases	= perfect accurate case
est	= estimator, c("mle", "map", "eap", "bayes"), default "map"
lo	= default -4
hi	= default 4
q	= default 100
kappa	= default 1
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

## Value

wcpm list

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,
  person.data = NA,
  person.id = "",
  task.id = "",
  occasion = "",
  group = "",
  max.counts = "",
  obs.counts = "",
  time = "",
  cases = NULL,
  est = "map",
  se = "analytic",
  failsafe = 0,
  bootstrap = 100,
  external = NULL,
  type = "general"
)
```

**Arguments**

<code>calib.data</code>	A class object. Output from calibration phase by <code>fit.model</code> function
<code>person.data</code>	A data frame. A long-format response data object.
<code>person.id</code>	Quoted variable name in <code>person.data</code> that indicates the unique individual identifier.
<code>task.id</code>	Quoted variable name in <code>person.data</code> that represents the unique task identifier. In the ORF assessment context, it is the passage identifier.
<code>occasion</code>	The column name in the data that represents the unique occasion.
<code>group</code>	The column name in the data that represents the unique group.
<code>max.counts</code>	Quoted variable name in <code>person.data</code> that represents the number of attempts in the task. In the ORF assessment context, it is the number of words in the passage.
<code>obs.counts</code>	The column name in the data that represents the words read correctly for each case.
<code>time</code>	The column name in the data that represents the time, in seconds, for each case.
<code>cases</code>	- individual id vectors, will directly use task data if no <code>calib.data</code> provided

est	- estimator keyword / c("mle", "map", "eap", "bayes"), default is bayes
se	- standard error keyword / c("analytic", "bootstrap"), default is analytic
failsafe	- retry time for bootstrap / default 0, can set to 5 ~ 50
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
bootstrap	- set K number of bootstrap / default 100

## Details

Additional details...

## Value

scoring list or Bootstrap dataset

## Note

More & 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,
  person.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")
```

---

sentence.data	<i>Sentence-level student data set</i>
---------------	--

---

### Description

A data set consisted of reading time data for 4 passages and 23 sentences from 58 students.

### Usage

```
sentence.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** sentence identifier

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

**wrc** words read correct

**sec** seconds to read the sentence

### Source

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

---

summary.bootstrap	<i>summary the information of bootstrap class</i>
-------------------	---

---

### 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	<i>summary the information of fit.model class</i>
-------------------	---

---

**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

# Index

## \* datasets

- MCEM, [11](#)
- passage.calib.bayes, [11](#)
- passage.calib.mcem, [12](#)
- passage2, [13](#)
- sentence.data, [20](#)

- bayes, [3](#)
- bayes.wcpm, [4](#)
- bspam-package, [2](#)

- exclude\_passages, [5](#)

- fit.model, [5](#), [18](#), [19](#)
- fit.model.testlet, [7](#)

- get.cases, [8](#)
- get.perfectcases, [9](#)
- get\_errlist, [10](#)
- getBootstrapSE, [9](#)

- log.initiating, [10](#)

- MCEM, [11](#)

- passage.calib.bayes, [11](#)
- passage.calib.mcem, [12](#)
- passage2, [13](#)
- plot.mcem, [13](#)
- prep, [14](#)
- preplong, [15](#)
- prepwide, [15](#)

- run.mcem, [16](#)
- run.scoring, [17](#)

- scoring, [7](#), [18](#)
- sentence.data, [20](#)
- summary.bootstrap, [20](#)
- summary.fit.model, [21](#)
- summary.scoring, [22](#)