# Package ‘BoundIRT’

September 16, 2026

**Title** Fit Bounded Continuous Item Response Theory Models to Data

**Version** 0.8.0

**Description** Bounded continuous data are encountered in many areas of test application. Examples include visual analogue scales used in the measurement of personality, mood, depression, and quality of life; item response times from tests with item deadlines; confidence ratings; and pain intensity ratings. Using this package, item response theory (IRT) models suitable for bounded continuous item scores can be fitted to data within a Bayesian framework. The package draws on posterior sampling facilities provided by R-package 'rstan' (Stan Development Team, 2025)<https://mc-stan.org/>. Available models include the Beta IRT model by Noel and Dauvier (2007)<doi:10.1177/0146621605287691>, the continuous response model by Samejima (1973)<doi:10.1007/BF03372160>, the unbounded normal model by Mellenbergh (1994)<doi:10.1207/s15327906mbr2903_2>, and the Simplex IRT model by Flores et al. (2020)<doi:10.1007/978-3-030-43469-4_8>. All models can be fitted with or without zero-one inflation (Molenaar et al., 2022)<doi:10.3102/10769986221108455>. Model fit comparisons can be conducted using the Watanabe-Akaike information criterion (WAIC), leave-one-out cross-validation information criterion (LOOIC), and the fully marginalized likelihood (i.e., Bayes factors).

**License** GPL-3

**Encoding** UTF-8

**RoxygenNote** 7.3.3

**Biarch** true

**Depends** R (>= 3.5)

**Imports** methods, mvtnorm, MASS, statmod, bayesplot, Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), rstantools (>= 2.6.0)

**LinkingTo** BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

**SystemRequirements** GNU make

**LazyData** true

**LazyDataCompression** xz

**Suggests** loo, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**NeedsCompilation** yes

**Author** Dylan Molenaar [aut, cre]

**Maintainer** Dylan Molenaar <d.molenaar@uva.nl>

**Repository** CRAN

**Date/Publication** 2026-09-16 15:20:07 UTC

# Contents



---

`BoundIRT-package` *The 'BoundIRT' package.*

---

### Description

Bounded continuous data are encountered in many areas of test application. Examples include visual analogue scales used in the measurement of personality, mood, depression, and quality of life; item response times from tests with item deadlines; confidence ratings; and pain intensity ratings. Using this package, item response theory (IRT) models suitable for bounded continuous item scores can be fitted to data within a Bayesian framework. The package draws on posterior sampling facilities provided by R-package 'rstan' (Stan Development Team, 2025). Available models include the Beta IRT model by Noel and Dauvier (2007), the continuous response model by Samejima (1973), the unbounded normal model by Mellenbergh (1994), and the Simplex IRT model by Flores et al.

(2020). All models can be fitted with or without zero-one inflation (Molenaar et al., 2022). Model fit comparisons can be conducted using the Watanabe–Akaike information criterion (WAIC), the leave-one-out cross-validation information criterion (LOOIC), and the fully marginalized likelihood (i.e., Bayes factors).

### Author(s)

**Maintainer**: Dylan Molenaar <d.molenaar@uva.nl>

### References

Flores, S., Bazan, J.L., & Bolfarine, H. (2020). A hierarchical joint model for bounded response time and response accuracy. In M. Wiberg, D. Molenaar, J. González,U. Bockenholt, & K. S. Kim (Eds.), *Quantitative psychology: The 84th Annual Meeting of the Psychometric Society, Santiago de Chile, Chile* (pp. 95-109). Springer. doi:10.1007/9783030434694_8

Mellenbergh, G. J. (1994). A unidimensional latent trait model for continuous item responses. *Multivariate Behavioral Research*, **29**(3), 223-236. doi:10.1207/s15327906mbr2903_2

Molenaar, D., Cúri, M., & Bazán, J.L. (2022). Zero and One Inflated Item Response Theory Models for Bounded Continuous Data. *Journal of Educational and Behavioral Statistics*, **47**, 693-735. doi:10.3102/10769986221108455

Noel, Y., & Dauvier, B. (2007). A beta item response model for continuous bounded responses. *Applied Psychological Measurement*, **31**(1), 47-73. doi:10.1177/0146621605287691

Samejima, F. (1973). Homogeneous case of the continuous response model. *Psychometrika*, **38**(2), 203-219.doi:10.1007/BF02291114

Stan Development Team (2025). RStan: the R interface to Stan. R package version 2.32.7. https://mc-stan.org/

---

ACL *Adjectives Check List data*

---

### Description

This dataset contains the bounded continuous items responses of 244 subjects to 218 items from 22 subscales of the Adjectives Check List (ACL; Gough & Heilbrun, 1980) including the covariate `female` which is coded `0` and `1`

### Format

The item scores are in object `ACL` and have originally been scored on a 60 millimeter line segment on which the subjects had to indicate to what degree each adjective applies to them. These scores have been recoded into a [0,1] interval. The Abasement scale does not contain 1 scores but it does contain 0 scores. The item content (adjectives) are in the column names of the data matrix. The responses from items with an * in the item content are contra-indicative items and have been reversely coded.

The object `ACL_index` contains an index matrix with each i-th row corresponding to the i-th column in the ACL data matrix. In the index matrix it is indicated for each item which of the 22 ACL factors it is supposed to measure. The factor numbering is as follows:

1. Communality
2. Achievement
3. Dominance
4. Endurance
5. Order
6. Intraception
7. Nurturance
8. Affiliation
9. Exhibition
10. Autonomy
11. Aggression
12. Change
13. Succorance
14. Abasement
15. Deference
16. Personal Adjustment
17. Ideal Self
18. Critical Parent
19. Nurturant Parent
20. Adult
21. Free Child
22. Adapted Child

All subscales of the ACL have been analyzed using bounded continuous IRT models with zero-one inflation by Molenaar et al. (2022; see Table 10 for the model fit statistics of all ACL subscales and see Table 11 for the parameter estimates).

For ease of the examples in this package, the item responses of the Abasement subscale are added to a separate dataset accessible via `data(Abasement)`. In addition, a zero-inflated Beta IRT model has been pre-estimated on the Abasement scale. The `fitBIRT` output object containing the estimation results is accessible via `data(out_beta)`. Note that the results by Molenaar et al. (2022; Tables 10 and 11) are based on a model with both zero and one inflation. Therefore, the estimates in `out_beta` and the subsequent model fit statistics obtained using `bridgeBIRT` are slightly different.

`out_beta` has been obtained using this code:

```
set.seed(1310)
data(Abasement)
out_beta = fitBIRT(Abasement, model="beta")
```

## References

Gough, H. G., & Heilbrun, A. B. (1980). *The adjective check list, manual 1980 Edition*. Consulting Psychologists Press.

Molenaar, D., Cúri, M., & Bazán, J.L. (2022). Zero and One Inflated Item Response Theory Models for Bounded Continuous Data. *Journal of Educational and Behavioral Statistics*, **47**, 693-735. doi:10.3102/10769986221108455

### Examples

```
data(ACL)
# show the item responses for the items measuring the first ACL factor (Communality)
head(ACL[,ACL_index[,3]==1])
```

---

`bridgeBIRT` *Estimate fully marginalized log-likelihood using bridge sampling.*

---

### Description

This function uses bridge sampling as discussed by Gronau et al. (2017) to obtain the fully marginalized likelihood of an estimated bounded IRT model.

### Usage

```
bridgeBIRT(x, ...)
## S3 method for class 'BoundIRT'
bridgeBIRT(x,niter=10000,tol=1e-7,silent=FALSE,...)
```

### Arguments

| | |
|---|---|
| `x` | A `BoundIRT`-object. |
| `niter` | Number of iterations of the algorithm. |
| `tol` | Tolerance for the algorithm. Algorithm stops if the change in $\log(r_1)$ between subsequent iterations is smaller than this value, see **Details**. |
| `silent` | If TRUE no progress is displayed during estimation. |
| `...` | Currently unused. |

### Details

In bridge sampling, a so-called bridge function is used to estimate the fully marginalized likelihood by combining existing posterior samples with additional samples drawn from a proposal distribution. In `bridgeBIRT`, a multivariate normal proposal distribution is used. Using these proposal samples, the final expression for the marginal likelihood involves an iterative procedure (see Equation 4.1 from Meng & Wong, 1996) which stops if the change in marginal likelihood between iterations does not exceed the tolerance.

To improve numerical stability and avoid overflow in the `exp()` function, we follow the approach described in Appendix B of Gronau et al. (2017). Specifically, the median log-likelihood value, $l^\star$, is subtracted from the log-marginal likelihood, and only the remaining component, $r_1$, is estimated. After the iterative scheme has converged, these two components are recombined to obtain the final estimate of the marginal likelihood.

## Value

An object of class `bridgeBIRT`, a list containing:

| | |
|---|---|
| `log_ml` | The fully marginalized log-likelihood. |
| `r1` | The estimate of $r_1$. |
| `l_star` | The value of $l^\star$ used (median log-likelihood across posterior samples). |
| `r1_hist` | The history of $r_1$ estimates across the iterations of the algorithm. |

`summary.bridgeBIRT` and `plot.bridgeBIRT` methods are available for `bridgeBIRT` objects.

## Author(s)

Dylan Molenaar <d.molenaar@uva.nl>

## References

Gronau, Q. F., et al. (2017). A tutorial on bridge sampling. *Journal of mathematical psychology*, **81**, 80-97. doi:10.1016/j.jmp.2017.09.005

Meng, X. L., & Wong, W. H. (1996). Simulating ratios of normalizing constants via a simple identity: a theoretical exploration. *Statistica Sinica*, **6**, 831-860. https://www.jstor.org/stable/24306045

## See Also

`fitBIRT` to estimate various bounded continuous IRT models to data.

`plot.BoundIRT` to plot the item and test information curves and predicted densities for the different models.

`coef.BoundIRT` to extract item parameter estimates from a `BoundIRT` object.

`latregBIRT` to conduct a latent regression of the person parameters on some observed predictor variables.

`rBIRT` simulate data according to a (zero/one inflated) bounded IRT model.

## Examples

```
# read in the pre-estimated object containing the Abasement results
# for the Beta IRT model and estimate the marginal likelihood
data(out_beta)
bridgeBIRT(out_beta)
```

`coef.BoundIRT` *Obtain item parameter estimates*

### Description

Extracts a matrix of posterior means for the item parameters of a `BoundIRT`-object

### Usage

```
## S3 method for class 'BoundIRT'
coef(object,sd=FALSE, ...)
```

### Arguments

| | |
|---|---|
| `object` | A `BoundIRT`-object. |
| `sd` | A Boolean indicating if posterior standard deviations should be returned as well |
| `...` | Other arguments to be passed to `print()` |

### Value

A matrix containing the posterior mean (and standard deviations if sd=TRUE) of the item parameters from a bounded IRT model fit with the `fitBIRT()` function.

### Author(s)

Dylan Molenaar `<d.molenaar@uva.nl>`

### See Also

`fitBIRT` to estimate various bounded continuous IRT models to data.

`plot.BoundIRT` to plot the item and test information curves and predicted densities for the different models.

`coef.BoundIRT` to extract item parameter estimates from a `BoundIRT` object.

`bridgeBIRT` to use bridge sampling to calculate the marginal loglikelihood for a model in a `BoundIRT` object.

`latregBIRT` to conduct a latent regression of the person parameters on some observed predictor variables.

`rBIRT` simulate data according to a (zero and/or one inflated) bounded IRT model.

### Examples

```
# read in the pre-estimated object containing the Abasement results
data(out_beta)

coef(out_beta)
```

fitBIRT *Fit various IRT models for bounded continuous items to data*

## Description

This function fits various IRT models for bounded continuous items using MCMC sampling in R-package 'rstan' (Stan Development Team, 2025). Available models include the Beta IRT model by Noel and Dauvier (2007), the continuous response model by Samejima (1973), the unbounded normal model by Mellenbergh (1994), and the Simplex IRT model by Flores et al. (2020). All models can be fitted with or without a zero and/or one inflation (Molenaar et al., 2022).

## Usage

```
fitBIRT(z,
        model=c("beta","samejima","simplex","normal"),
        inflated=c("auto","zero","one","both","none"),
        iter=2000,
        warmup=NULL,
        nchains=1,
        prior=NULL,
        silent=FALSE,
        ...)
```

## Arguments

`z` A matrix of size `N` by `n` containing the bounded continuous item scores, where `N` is the number of subjects and `n` is the number of items. The items should be coded in a $(0, 1)$, a $[0, 1)$, a $(0, 1]$, or $[0, 1]$ interval.

`model` A character string indicating the model to be fit. The options are:

- **"beta"** (Default) The Beta IRT model by Noel and Dauvier (2007) with item specific discrimination parameter.
- **"samejima"** The IRT model by Samejima (1973).
- **"simplex"** The Simplex IRT model by Flores et al. (2020).
- **"normal"** The (unbounded) unidimensional normal linear factor model by Mellenbergh (1994)

`inflated` A character string indicating the kind of inflation to use. The options are:

- **"auto"** (Default) Zero and one inflation parameters are only added for items that actually contain zero and/or one scores .
- **"zero"** Zero inflation parameters are estimated for all items.
- **"one"** One inflation parameters are estimated for all items.
- **"both"** Zero and one inflation parameters are estimated for all items.
- **"none"** No zero-one inflation. The data shouldn't contain 0 or 1 scores.

`iter` Number of MCMC samples to draw, including warmup samples

`warmup` Number of warmup samples. Default is floor(iter/2)

| | |
|---|---|
| `nchains` | Number of MCMC chains to use. Default is 1. |
| `prior` | If `NULL`, the default priors are being used. Otherwise a list can be specified giving the prior standard deviation of (some of) the parameters. See **Details** |
| `silent` | If TRUE no details are displayed during estimation |
| `...` | Additional arguments to pass to `rstan::sampling()` |

## Details

Function `fitBIRT` fits the models discussed by Molenaar et al. (2022) including the facilities for zero and/or one inflation. If $\alpha_i$, $\beta_i$, and $\phi_i$ are respectively the discrimination, easiness, and dispersion parameters of item `i`, $\theta_p$ is the latent variable position of subject `p`, and $z_{pi}$ is the bounded continuous item score of subject `p` on item `i`, then the different models can be given as follows:

**Beta model** $z_{pi} \sim \text{Beta}(a_{pi}, b_{pi})$

$a_{pi} = \exp(.5(\alpha_i\theta_p + \beta_i + \log(\phi_i)))$
$b_{pi} = \exp(.5(-\alpha_i\theta_p - \beta_i + \log(\phi_i)))$

**Samejima model** $z'_{pi} \sim \text{Normal}(\alpha_i\theta_p + \beta_i, \phi_i)$

$z'_{pi} = \text{logit}(z_{pi})$

**Simplex model** $z_{pi} \sim \text{Simplex}(1/(1 + \exp(-(\alpha_i\theta_p + \beta_i))), \phi_i)$

**Normal model** $z_{pi} \sim \text{Normal}(\alpha_i\theta_p + \beta_i, \phi_i)$

**Zero and/or one inflation** To these models, zero and/or one inflation can be added as follows:

$A = P(z_{pi} = 0|\theta_p) = 1/(1 + \exp(-(\gamma_{0i} - \alpha_i\theta_p)))$
$B = P(z_{pi} = 1|\theta_p) = 1 - 1/(1 + \exp(-(\gamma_{1i} - \alpha_i\theta_p)))$

where $\gamma_{0i}$ and $\gamma_{1i}$ model the amount of zero and one inflation respectively. The conventional models above are accommodated as follows:

$f(z_{pi}|\theta_p) = A$ for $z_{pi} = 0$
$f(z_{pi}|\theta_p) = B$ for $z_{pi} = 1$
$f(z_{pi}|\theta_p) = [(1 - B) - A] \times k(z_{pi}|\theta_p)$ for $z_{pi} \in (0, 1)$

where $k(.)$ is the distribution according to the conventional model above. See Molenaar et al. (2022) for details.

In estimating the models above, normal priors are specified for $\theta_p$, $\log(\alpha_i)$, $\beta_i$, $\log(\phi_i)$, $\gamma_{0i}$, and $\gamma_{1i}$. All priors have mean 0, and by default, a standard deviation of 1 for $\theta_p$ (for identification purposes) and 10 for the other parameters. Changing (one of) the defaults involves specifying a list with elements `sT`, `sA`, `sB`, `sP` and/or `sG`, which overrides the default. It is thus possible to stick to the defaults, but change the prior standard deviation for $\log\alpha_i$ and $\beta_i$ by using for instance `prior=list(sA=5,sB=5)`. Note that setting `sG` affects both the prior of $\gamma_{0i}$ and $\gamma_{1i}$.

## Value

An object of class `BoundIRT` with values:

| | |
|---|---|
| `output` | The posterior means and standard deviations of the item parameters. |
| `theta` | The posterior means and standard deviations of the person parameters. |
| `mcmc` | The `stanfit` object containing all sampling results. See `stanfit`. |

| | |
|---|---|
| `z` | The item score matrix provided in argument `z`, with missing values coded as `-999`. |
| `model` | The fitted model, i.e. (one of) the options passed to argument `model`. |
| `inflated` | The kind of zero-one inflation used. Differs from the `inflated` argument if `inflated="auto"` was used, in which case this is the kind of inflation that was detected in `z` (one of `"none"`, `"zero"`, `"one"`, or `"both"`). |
| `prior` | The prior standard deviations used in the list format discussed above. |

For the `BoundIRT` object, the following methods are available:

| | |
|---|---|
| `summary.BoundIRT` | which gives the default `rstan` summary overview of the parameters. |
| `plot.BoundIRT` | which provides various IRT plotting facilities, see `plot.BoundIRT` |
| `print.BoundIRT` | which gives the posterior means and standard deviations of the item parameters. |
| `coef.BoundIRT` | which returns the posterior means of the item parameters, see `coef.BoundIRT`. |
| `mcmc.BoundIRT` | which returns the underlying `stanfit` object (same as element `mcmc` above). |
| `loo.BoundIRT`, `waic.BoundIRT` | which calculate the marginal LOOIC and WAIC using `loglikBIRT` and R-package `loo`. |
| `bridgeBIRT.BoundIRT` | which uses bridge sampling to estimate the fully marginalized log-likelihood, see `bridgeBIRT`. |
| `repBIRT.BoundIRT` | which simulates replicated datasets for posterior predictive checking, see `repBIRT`. |

## Author(s)

Dylan Molenaar <d.molenaar@uva.nl>

## References

Flores, S., Bazan, J.L., & Bolfarine, H. (2020). A hierarchical joint model for bounded response time and response accuracy. In M. Wiberg, D. Molenaar, J. González, U. Bockenholt, & K. S. Kim (Eds.), *Quantitative psychology: The 84th Annual Meeting of the Psychometric Society, Santiago de Chile, Chile* (pp. 95-109). Springer. doi:10.1007/9783030434694_8

Mellenbergh, G. J. (1994). A unidimensional latent trait model for continuous item responses. *Multivariate Behavioral Research*, **29**(3), 223-236. doi:10.1207/s15327906mbr2903_2

Molenaar, D., Cúri, M., & Bazán, J.L. (2022). Zero and One Inflated Item Response Theory Models for Bounded Continuous Data. *Journal of Educational and Behavioral Statistics*, **47**, 693-735. doi:10.3102/10769986221108455

Noel, Y., & Dauvier, B. (2007). A beta item response model for continuous bounded responses. *Applied Psychological Measurement*, **31**(1), 47-73. doi:10.1177/0146621605287691

Samejima, F. (1973). Homogeneous case of the continuous response model. *Psychometrika*, **38**(2), 203-219.doi:10.1007/BF02291114

Stan Development Team (2025). RStan: the R interface to Stan. R package version 2.32.7. `https://mc-stan.org/`

### See Also

`plot.BoundIRT` to plot the item and test information curves and predicted densities for the different models.

`coef.BoundIRT` to extract item parameter estimates from a `BoundIRT` object.

`loglikBIRT` to calculate the log-likelihood for each person and each posterior sample. Needed to calculate WAIC and LOOIC for a given model.

`repBIRT` to simulate replicated datasets using (a subset of) the posterior parameter samples. For posterior predictive model checking.

`bridgeBIRT` to use bridge sampling to calculate the marginal loglikelihood for a model in a `BoundIRT` object.

`latregBIRT` to conduct a latent regression of the person parameters on some observed predictor variables.

`rBIRT` simulate data according to a (zero/one inflated) bounded IRT model.

### Examples

```
#load the Abasement data which contains zero inflation
data(Abasement)

# fit a zero inflated beta IRT model
# for illustrative purposes, we only use 20 iterations.
# In practice use many more!

res=fitBIRT(Abasement,model="beta",iter=20)
print(res)   # quick overview
coef(res)   # extract posterior item parameter means
summary(res,c("alpha","beta"))   # request the rstan summary
```

---

## `latregBIRT` *Multivariate latent regression with one or more BoundIRT latent variables and observed covariates*

---

### Description

This function conducts a factor score regression using latent and observed variables (if any). The latent variables need to be estimated with `BoundIRT` but they can differ in their measurement model, as long as they are estimated on subscales administered to the same sample of subjects.

### Usage

```
latregBIRT(formula,
            observed,
            latent,
            method=c("bayes", "freq"),
            nboot=1000)
```

## Arguments

`formula` A formula to specify the linear regression of the latent variable on the predictor variables in `X`. See **Examples**.

`observed` A matrix of predictor variables. The column names of the matrix should at least include the predictor names provided in `formula`. All other variables are ignored.

`latent` A list of `BoundIRT` objects of which the element names correspond to the latent variables in `formula`. The objects need to be estimated on the same sample of subjects using the same number of MCMC samples.

`method` A character string indicating the method to use for parameter estimation. Default uses a Bayesian bootstrap to approximate the posterior regression parameter distribution. Alternative is a frequentist approach which also gives the analytical estimates (which are identical to the Bayesian ones due to the uniform priors used) and a frequentist bootstrap to produce standard errors.

`nboot` The number of bootstrap samples to obtain the standard errors. See **Details**.

## Details

`latregBIRT` fits a multivariate regression model to the posterior latent variable means from one or more bounded continuous IRT models:

$$\boldsymbol{y} = \boldsymbol{B}\boldsymbol{x} + \epsilon$$

where $\boldsymbol{y}$ and $\boldsymbol{x}$ are vectors of respectively outcome and predictor variables, $\boldsymbol{B}$ is a matrix of regression parameters, and $\epsilon$ is a vector of residuals. Both $\boldsymbol{y}$ and $\boldsymbol{x}$ may contain one or more latent variables estimated on subsets of the item level data using function `fitBIRT`. The resulting `BoundIRT`-objects need to be collected in a list and provided to `latregBIRT` using argument `latent`. Using the posterior latent variable means and standard deviations, the regression model above is fit correcting for the posterior uncertainty in the latent variables. That is, as the latent variable posteriors are misspecified with respect to the predictor variables, they are only unbiased if the number of items approaches infinity. As applications of (bounded continuous) IRT models involve a finite number of items, the posterior covariance matrix of the latent variables and the predictor variables is adjusted for the nonzero variance in the latent variable posteriors. The correction is based on the two-step approach by Croon (2002) in which the factor scores from a linear factor model are estimated in step 1 and used in a regression model in step 2. As such, Croon's method is a statistically valid alternative to fitting the factor model and latent regression model simultaneously. Vermunt (2025) discusses how Croon's method can be generalized to models with heterogeneous measurement error like IRT models. `latregBIRT` implements this idea by correcting the covariance matrix of the latent and observed variables using the formula from Vermunt (2025) and estimating the posterior distribution using Bayesian bootstrapping (or the sampling distribution in the case of `method="freq"`).

## Value

An object of class `latregBIRT` with values:

`output` A matrix containing the posterior regression slope means and standard deviations (`method="bayes"`), or the point estimates and standard errors (`method="freq"`).

`samples` A matrix containing the bootstrap samples.

Percentiles of the (bootstrap) regression slope distribution can be obtained via `summary.latregBIRT`, which takes its own `prob` argument (default `c(.025,.25,.50,.75,.975)`).

`plot.latregBIRT` and `coef.latregBIRT` methods are also available for `latregBIRT` objects.

## Author(s)

Dylan Molenaar <d.molenaar@uva.nl>

## References

Croon, M. (2002). Using predicted latent scores in general latent structure models. In G. Marcoulides and I. Moustaki (Eds.), *Latent Variable and Latent Structure Models* (pp. 195-224). Lawrence Erlbaum.

Vermunt J.K. (2025). Stepwise estimation of latent variable models: An overview of approaches. *Statistical Modelling*, **25**(6), 530-551. doi:10.1177/1471082X251355693

## See Also

`fitBIRT` to estimate various bounded continuous IRT models to data.

`plot.BoundIRT` to plot the item and test information curves and predicted densities for the different models.

`coef.BoundIRT` to extract item parameter estimates from a `BoundIRT` object.

`loglikBIRT` to calculate the log-likelihood for each person and each posterior sample. Needed to calculate WAIC and LOOIC for a given model.

`bridgeBIRT` to use bridge sampling to calculate the marginal loglikelihood for a model in a `BoundIRT` object.

`rBIRT` simulate data according to a (zero/one inflated) bounded IRT model.

## Examples

```
#######################################
# EXAMPLE 1: Multiple regression     #
#######################################

# open pre-estimated Beta IRT model on Abasement data (see ?Abasement)
 data(out_beta)

# simulate a latent regression with 2 covariates

EAP=out_beta$theta[,1]
N=length(EAP)
age=.4*EAP+sqrt(1-.4^2)*rnorm(N)
ses=.3*EAP+sqrt(1-.3^2)*rnorm(N)

# conduct latent regression
predictors=cbind(age,ses)
latregBIRT(Abasement~age+ses,predictors,list(Abasement=out_beta))
```

```
######################################
# EXAMPLE 2: Multivariate regression #
######################################

  # the following shows a more sophisticated latent regression model
  data(ACL)                               # see ?ACL for details about the data
  female = ACL[,1]
  Comm_items = ACL[,ACL_index[,3]==1]     # ACL factor: Communality
  Achiev_items = ACL[,ACL_index[,3]==2]   # ACL factor: Achievement
  Dom_items = ACL[,ACL_index[,3]==3]      # ACL factor: Dominance

  # fit measurement models to each of the subscales
  # (note, you can use different models for the different latent variables)
  # please also note that for illustrative purposes, only 20 mcmc samples are drawn.
  # in practice use (many) more

  set.seed(1234)
  Comm_res = fitBIRT(Comm_items, "beta",iter=20)
  Achiev_res = fitBIRT(Achiev_items, "samejima",iter=20)
  Dom_res = fitBIRT(Dom_items, "simplex",iter=20)

  observed=cbind(female)
  latent=list(Communality = Comm_res,
              Achievement = Achiev_res,
              Dominance = Dom_res)

  #conduct latent regression
  latreg_res = latregBIRT(Communality + Achievement ~ Dominance + female,
                          observed, latent)
  print(latreg_res)
  # cbind(Communality,Achievement) will also work
```

---

`loglikBIRT` *Calculate the log-likelihood values of each person for each posterior parameter sample*

---

## Description

This function can be used to calculate the log-likelihood values of each person for each posterior parameter sample in a `BoundIRT`-object.

## Usage

```
loglikBIRT(object, marginal=TRUE,nquad=15, silent=FALSE)
```

### Arguments

| | |
|---|---|
| object | A BoundIRT-object |
| marginal | If TRUE the marginal loglikelihood is calculated for each subject (default), otherwise the conditional likelihood is calculated for each subject-item combination. See **Details**. |
| nquad | If marginal=TRUE this argument determines the number of quadrature points used to approximate the integral in the marginal likelihood. |
| silent | If TRUE no details are displayed about the progress |

### Details

For latent variable models like the ones in BoundIRT, Merkle et al., (2019) recommend the use of the likelihood function that is marginalized over the person parameters (i.e., not over the item parameters) in calculating fit statistics like the WAIC and DIC. Using marginal=TRUE will output this marginal likelihood value for each person and each sample of the posterior. If the interest is in calculating the conventional fit measures (e.g., conditional WAIC and DIC), marginal=FALSE should be used to obtain the likelihood values for each person conditional on the person and item parameters of each posterior sample. Note that for the WAIC and LOOIC, direct functions loo and waic are available. The (marginal or conditional) log-likelihood values from loglikBIRT can also be used as a building block to compute other statistics, like DIC, which are not directly implemented in this package.

### Value

The function returns a matrix of dimensions (nsamples) by (N) containing the (marginal) log-likelihood values for each subject and posterior sample.

### Author(s)

Dylan Molenaar <d.molenaar@uva.nl>

### References

Gabry J, Simpson D, Vehtari A, Betancourt M, Gelman A (2019). Visualization in Bayesian workflow. *Journal of the Royal Statistical Society, A*, **182**, 389-402. doi:10.1111/rssa.12378

Merkle EC, Furr D, Rabe-Hesketh S. (2019).Bayesian Comparison of Latent Variable Models: Conditional Versus Marginal Likelihoods. *Psychometrika*, **84**(3), 802-829. doi:10.1007/s11336-019096790

### See Also

fitBIRT to estimate various bounded continuous IRT models to data.

plot.BoundIRT to plot the item and test information curves and predicted densities for the different models.

coef.BoundIRT to extract item parameter estimates from a BoundIRT object.

loo and waic to directly calculate the marginal LOOIC and WAIC for a BoundIRT object.

repBIRT to simulate replicated datasets using (a subset of) the posterior parameter samples. For posterior predictive model checking.

bridgeBIRT to use bridge sampling to calculate the marginal loglikelihood for a model in a BoundIRT object.

latregBIRT to conduct a latent regression of the person parameters on some observed predictor variables.

## Examples

```
#read in the prefitted results from the Abasement data (see ?Abasement and ?out_beta)
data(out_beta)

# Calculate marginal log-likelihood

  logL=loglikBIRT(out_beta)
```

---

loo *Calculate WAIC and LOOIC for a BoundIRT object*

---

## Description

These functions calculate the (marginal) WAIC and LOOIC for a fitted BoundIRT object.

## Usage

```
## S3 method for class 'BoundIRT'
loo(x, marginal=TRUE, nquad=15, silent=FALSE, ...)

## S3 method for class 'BoundIRT'
waic(x, marginal=TRUE, nquad=15, silent=FALSE, ...)
```

## Arguments

| | |
|---|---|
| x | A BoundIRT-object. |
| marginal | If TRUE (default) the marginal log-likelihood is used, otherwise the conditional log-likelihood is used. See loglikBIRT. |
| nquad | If marginal=TRUE this argument determines the number of quadrature points used to approximate the integral in the marginal likelihood. See loglikBIRT. |
| silent | If TRUE no details are displayed about the progress of loglikBIRT. |
| ... | Other arguments passed to loo or waic from R-package 'loo'. |

## Details

loo and waic first call loglikBIRT to obtain the (marginal) log-likelihood values for each person and each posterior sample, and then pass these on to functions loo and waic from R-package loo to calculate the LOOIC and WAIC, respectively.

## Value

An object of class loo (for loo) or of class waic (for waic), as returned by R-package loo. See loo and waic.

## Author(s)

Dylan Molenaar <d.molenaar@uva.nl>

## References

Vehtari, A., Gelman, A., & Gabry, J. (2017). Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing*, **27**, 1413-1432. doi:10.1007/s11222016-96964

## See Also

loglikBIRT to calculate the (marginal or conditional) log-likelihood values directly.

fitBIRT to estimate various bounded continuous IRT models to data.

bridgeBIRT to use bridge sampling to calculate the fully marginalized loglikelihood for a model in a BoundIRT object.

## Examples

```
#read in the prefitted results from the Abasement data (see ?Abasement and ?out_beta)
data(out_beta)


  if (requireNamespace("loo", quietly = TRUE)) {
    loo(out_beta)    # marginal LOOIC
    waic(out_beta)   # marginal WAIC
  }
```

---

plot.BoundIRT *Plotting function for* BoundIRT*-objects.*

---

## Description

This function can be used to produce item and test information curves, and the model implied densities of a fitted bounded continuous IRT model.

## Usage

```
## S3 method for class 'BoundIRT'
plot(x,
     what=c("icc","iic","tic","density"),
     items=NULL,
     separate=FALSE,
     range=c(-3,3),
     by=.001,
     ymax=NULL,
     xlab=NULL,
     ylab=NULL,
     main=NULL,
     col=NULL,
     lwd_prob=2,
     ...)
```

## Arguments

| | |
|---|---|
| `x` | A `BoundIRT`-object to produce the plots for. |
| `what` | A character string indicating what plot to produce. The options are:<br>**"icc"** (Default) Item characteristic curves.<br>**"iic"** Item information curves.<br>**"tic"** Test information curves.<br>**"density"** The model implied density for the observed data. |
| `items` | A numeric vector indicating which items to consider for the plots. If NULL all items are plotted (default). If `items` is provided and `what="tic"`, the test information is based only on these items. |
| `separate` | If `separate=TRUE` the different items are depicted in separate plots |
| `range` | Range of the latent variable on the x-axis. Argument will be neglected for `what="density"`. |
| `by` | increment for the x-axis of the plot. |
| `ymax` | Specify the maximum value for the y-axis. For multiple items, this can be an array of size length(items), or a single value for all items. |
| `xlab` | Specify an alternative label for the x-axis. |
| `ylab` | Specify an alternative label for the y-axis. |
| `main` | Specify an alternative title for plot. |
| `col` | The color code to be used for plotting different items. Can be an array and will be recycled if length(items)!=length(col). Argument can be any valid R color specification. Default is 1:6. |
| `lwd_prob` | For `what="density"` and a model with zero and/or one inflation, the probabilities of zero and one scores are depicted using vertical lines. The width of these lines can be set using this argument. Default is 2. Width of the other lines can be set using `...` below. |
| `...` | Other arguments to be passed to `matplot` |

.

## Details

Plotting the model implied densities is relatively slow due to the numerical integration involved in marginalizing over the latent variable. In the presence of zero and/or one inflation, model implied densities include the probability of zero and/or one scores by a vertical line. As this vertical line represents a probability and the rest of the plot is about density, the zero/one probabilities may be masked if density values are large even if the zero/one inflation is quite substantial. See **Examples** for how to extract the zero/one probabilities in the case these are hard to see from the plot.

## Value

Invisibly, a list with components:

| | |
|---|---|
| `x` | The x-axis values used (the latent variable grid, or the $[0, 1]$ item-score grid for `what="density"`). |
| `out` | The corresponding y-axis values: a matrix with one column per item in `items` (one row per value of `x`), except for `what="tic"` where a single vector summed across items is returned. |

## Author(s)

Dylan Molenaar `<d.molenaar@uva.nl>`

## See Also

`fitBIRT` to estimate various bounded continuous IRT models to data.

`coef.BoundIRT` to extract item parameter estimates from a `BoundIRT` object.

`loglikBIRT` to calculate the log-likelihood for each person and each posterior sample. Needed to calculate WAIC and LOOIC for a given model.

`repBIRT` to simulate replicated datasets using (a subset of) the posterior parameter samples. For posterior predictive model checking.

`bridgeBIRT` to use bridge sampling to calculate the marginal loglikelihood for a model in a `BoundIRT` object.

`latregBIRT` to conduct a latent regression of the person parameters on some observed predictor variables.

`rBIRT` simulate data according to a (zero/one inflated) bounded IRT model.

## Examples

```
#load results for zero inflated beta IRT model fit to the Abasement data
# see ?Abasement and ?out_beta
data(out_beta)

# plots for item 1
oldpar=par(mfrow=c(2,2))
plot(out_beta,what="icc",items=1)
plot(out_beta,what="iic",items=1)
plot(out_beta,what="tic",items=1)
den_plot=plot(out_beta,what="density",items=1)
```

```
par(oldpar)

#extract zero probabilities (there are no ones in the Abasement data)
den_plot$out[1]       # zero probabilities
```

---

`print.BoundIRT` *Print, summarize, and extract sampling results from a BoundIRT object*

---

### Description

`print` gives the posterior means and standard deviations of the item parameters of a `BoundIRT`-object. `summary` gives the default `rstan` summary overview of all sampled parameters. `mcmc` extracts the underlying `stanfit` object.

### Usage

```
## S3 method for class 'BoundIRT'
print(x, digits=3, ...)
## S3 method for class 'BoundIRT'
summary(object, ...)
## S3 method for class 'BoundIRT'
mcmc(x, ...)
```

### Arguments

| | |
|---|---|
| `x` | A `BoundIRT`-object. |
| `object` | A `BoundIRT`-object. |
| `digits` | Number of decimal places the values are rounded to. |
| `...` | Other arguments. For `summary.BoundIRT` these are passed to `rstan::summary`; for `print.BoundIRT` they are passed to `print`. |

### Value

`print` returns its argument invisibly. `summary` returns an object of class `summary.BoundIRT`, the result of `rstan::summary` applied to the `stanfit` object. `mcmc` returns that `stanfit` object directly.

### Author(s)

Dylan Molenaar <d.molenaar@uva.nl>

### See Also

`fitBIRT` to estimate various bounded continuous IRT models to data.

`coef.BoundIRT` to extract item parameter estimates from a `BoundIRT` object.

`plot.BoundIRT` to plot the item and test information curves and predicted densities for the different models.

## Examples

```
data(out_beta)

out_beta          # print.BoundIRT
summary(out_beta) # summary.BoundIRT
mcmc(out_beta)    # the underlying stanfit object
```

---

print.bridgeBIRT *Print, summarize, and plot a bridgeBIRT object*

---

## Description

`print` gives the estimated (log-)marginal likelihood of a `bridgeBIRT`-object. `summary` gives an overview of the bridge sampling results. `plot` plots the estimated marginal log-likelihood against the iterations of the bridge sampling algorithm, to inspect convergence.

## Usage

```
## S3 method for class 'bridgeBIRT'
print(x, digits=3, ...)
## S3 method for class 'bridgeBIRT'
summary(object, ...)
## S3 method for class 'bridgeBIRT'
plot(x, xlab="iteration", ylab="marginal log-likelihood", type="l", ...)
```

## Arguments

| | |
|---|---|
| `x` | A `bridgeBIRT`-object. |
| `object` | A `bridgeBIRT`-object. |
| `digits` | Number of decimal places the values are rounded to. |
| `xlab, ylab, type` | Standard plotting arguments, see `plot.default`. |
| `...` | Other arguments passed to `print` (for `print.bridgeBIRT`) or `plot.default` (for `plot.bridgeBIRT`). |

## Value

`print` returns its argument invisibly. `summary` returns an object of class `summary.bridgeBIRT`, a list with elements `log_ml` (the estimated log-marginal likelihood), `l_star`, `r1`, and `n_iter` (the number of algorithm iterations used). `plot` is called for its side effect of producing a plot.

## Author(s)

Dylan Molenaar <d.molenaar@uva.nl>

### See Also

bridgeBIRT to use bridge sampling to calculate the marginal loglikelihood for a model in a BoundIRT object.

### Examples

```
data(out_beta)
res=bridgeBIRT(out_beta)

res            # print.bridgeBIRT
summary(res)   # summary.bridgeBIRT
plot(res)      # plot.bridgeBIRT
```

---

print.latregBIRT *Print, summarize, plot, and extract coefficients from a latregBIRT object*

---

### Description

print gives the (bootstrap) regression slope estimates of a latregBIRT-object. summary gives the estimates plus percentiles for one or more regression slopes. plot gives histograms and QQ-plots of the (bootstrap) sampling distribution for one or more regression slopes. coef extracts the point estimates of the regression slopes.

### Usage

```
## S3 method for class 'latregBIRT'
print(x, digits=3, ...)
## S3 method for class 'latregBIRT'
summary(object, pars=NULL, prob=c(.025,.25,.50,.75,.975), ...)
## S3 method for class 'latregBIRT'
plot(x, pars=NULL, ask=TRUE, ...)
## S3 method for class 'latregBIRT'
coef(object, ...)
```

### Arguments

| | |
|---|---|
| x | A latregBIRT-object. |
| object | A latregBIRT-object. |
| digits | Number of decimal places the values are rounded to. |
| pars | A formula, or list of formulas, indicating which regression slope(s) to summarize or plot (see **Examples** in latregBIRT). If NULL (default), all regression slopes are used. |
| prob | A numeric vector of strictly increasing probabilities to compute percentiles of the regression slope(s). |
| ask | If TRUE (default), the user is prompted before each new plotting page. |
| ... | Other arguments. For print.latregBIRT, passed to print. |

### Value

`print` returns its argument invisibly. `summary` returns a matrix of class `summary.latregBIRT` with the (bootstrap) mean/estimate, standard deviation/error, and percentiles for the requested regression slope(s). `plot` is called for its side effect of producing plots. `coef` returns a matrix with the point estimates of the regression slopes.

### Author(s)

Dylan Molenaar <d.molenaar@uva.nl>

### See Also

`latregBIRT` to conduct a latent regression of the person parameters on some observed predictor variables.

### Examples

```
data(out_beta)
EAP=out_beta$theta[,1]
N=length(EAP)
age=.4*EAP+sqrt(1-.4^2)*rnorm(N)
ses=.3*EAP+sqrt(1-.3^2)*rnorm(N)
predictors=cbind(age,ses)
res=latregBIRT(Abasement~age+ses,predictors,list(Abasement=out_beta))

res              # print.latregBIRT
summary(res)     # summary.latregBIRT
coef(res)        # coef.latregBIRT

  plot(res)      # plot.latregBIRT
```

---

## `print.repBIRT` *Print, summarize, and plot a repBIRT object*

---

### Description

`print` gives a short overview of the replicated datasets in a `repBIRT`-object. `summary` gives the mean, standard deviation, and percentiles of the replications, per item. `plot` produces a posterior predictive density overlay plot (using `bayesplot::ppc_dens_overlay`) for one or more items.

### Usage

```
## S3 method for class 'repBIRT'
print(x, digits=3, ...)
## S3 method for class 'repBIRT'
summary(object, items=NULL, probs=c(.025,.25,.50,.75,.975), ...)
## S3 method for class 'repBIRT'
plot(x, items=1, ...)
```

### Arguments

| | |
|---|---|
| `x` | A `repBIRT`-object. |
| `object` | A `repBIRT`-object. |
| `digits` | Number of decimal places the values are rounded to. Currently has no visible effect, as `print.repBIRT` only displays integer counts, but is included for consistency with the other `print` methods in this package. |
| `items` | A numeric vector indicating which items to summarize or plot. Default is all items for `summary`, and item `1` for `plot`. |
| `probs` | A numeric vector of strictly increasing probabilities to compute percentiles of the replications. |
| `...` | For `plot.repBIRT`, other arguments passed to `bayesplot::ppc_dens_overlay`. For `print.repBIRT`, passed to `print`. |

### Value

`print` returns its argument invisibly. `summary` returns a matrix of class `summary.repBIRT` with the mean, sd, and percentiles of the replications for the requested items. `plot` produces a plot per requested item and returns the corresponding `ggplot` object(s) invisibly.

### Author(s)

Dylan Molenaar <d.molenaar@uva.nl>

### See Also

`repBIRT` to simulate replicated datasets using (a subset of) the posterior parameter samples.

### Examples

```
data(out_beta)
z_rep=repBIRT(out_beta,nrep=10)

z_rep             # print.repBIRT
summary(z_rep)    # summary.repBIRT
plot(z_rep,items=1) # plot.repBIRT
```

---

## rBIRT — *Simulate data according to various IRT models for bounded continuous items*

---

### Description

This function simulates data according to various IRT models for bounded continuous items. Available models include the Beta IRT model by Noel and Dauvier (2007), the continuous response model by Samejima (1973), the unbounded normal model by Mellenbergh (1994), and the Simplex IRT model by Flores et al. (2020). All models can be simulated with or without zero and/or one inflation (Molenaar et al., 2022).

## Usage

```
rBIRT(N,nit,alpha,beta,phi,theta,
      gamma0=NULL,
      gamma1=NULL,
      model=c("beta","samejima","simplex","normal"))
```

## Arguments

| | |
|---|---|
| `N` | Number of subjects |
| `nit` | Number of items |
| `alpha` | A nit-dimensional array of true discrimination parameter values |
| `beta` | A nit-dimensional array of true easiness parameter values |
| `phi` | A nit-dimensional array of true dispersion parameter values |
| `theta` | A N-dimensional array of true person parameter values |
| `gamma0` | Optional: A nit-dimensional array of true zero inflation parameters |
| `gamma1` | Optional: A nit-dimensional array of true one inflation parameters |
| `model` | A character string indicating what model to use to simulate the data. The options are:<br>**"beta"** (Default) The Beta IRT model by Noel and Dauvier (2007) with item specific discrimination parameter.<br>**"samejima"** The IRT model by Samejima (1973).<br>**"simplex"** The Simplex IRT model by Flores et al. (2020).<br>**"normal"** The (unbounded) unidimensional normal linear factor model by Mellenbergh (1994) |

## Details

If arguments gamma0 and gamma1 are not provided, data is simulated according to the conventional models. Zero and one inflation can be introduced separately by only providing gamma0 or gamma1, or zero and one inflation can both be introduced by specifying both gamma0 and gamma1. If the interest is in simulating some -but not all-, items with zero and/or one inflation, the uninflated items can be coded as -Inf in gamma0 and Inf in gamma1. See fitBIRT for the exact parameterization of the models.

## Value

A matrix of size `N` by `nit` containing the simulated item responses.

## Author(s)

Dylan Molenaar `<d.molenaar@uva.nl>`

## References

Flores, S., Bazan, J.L., & Bolfarine, H. (2020). A hierarchical joint model for bounded response time and response accuracy. In M. Wiberg, D. Molenaar, J. González, U. Bockenholt, & K. S. Kim (Eds.), *Quantitative psychology: The 84th Annual Meeting of the Psychometric Society, Santiago de Chile, Chile* (pp. 95-109). Springer. doi:10.1007/9783030434694_8

Mellenbergh, G. J. (1994). A unidimensional latent trait model for continuous item responses. *Multivariate Behavioral Research*, **29**(3), 223-236. doi:10.1207/s15327906mbr2903_2

Molenaar, D., Cúri, M., & Bazán, J.L. (2022). Zero and One Inflated Item Response Theory Models for Bounded Continuous Data. *Journal of Educational and Behavioral Statistics*, **47**, 693-735. doi:10.3102/10769986221108455

Noel, Y., & Dauvier, B. (2007). A beta item response model for continuous bounded responses. *Applied Psychological Measurement*, **31**(1), 47-73. doi:10.1177/0146621605287691

Samejima, F. (1973). Homogeneous case of the continuous response model. *Psychometrika*, **38**(2), 203-219. doi:10.1007/BF02291114

Stan Development Team (2025). RStan: the R interface to Stan. R package version 2.32.7. `https://mc-stan.org/`

## See Also

`fitBIRT` to estimate various bounded continuous IRT models to data.

`plot.BoundIRT` to plot the item and test information curves and predicted densities for the different models.

`coef.BoundIRT` to extract item parameter estimates from a `BoundIRT` object.

`loglikBIRT` to calculate the log-likelihood for each person and each posterior sample. Needed to calculate WAIC and LOOIC for a given model.

`repBIRT` to simulate replicated datasets using (a subset of) the posterior parameter samples. For posterior predictive model checking.

`bridgeBIRT` to use bridge sampling to calculate the marginal loglikelihood for a model in a `BoundIRT` object.

`latregBIRT` to conduct a latent regression of the person parameters on some observed predictor variables.

## Examples

```
# data sim with 100 subjects and 10 items according to
# beta IRT model with zero and one inflation

nit=10
N=100
beta=runif(nit,.5,1)
alpha=runif(nit,.5,.7)
gamma0=runif(nit,-3,-1.5)
gamma1=runif(nit,1.5,3)

theta=rnorm(N)
```

```
phi=seq(1.5,2.5,length=nit)
z=rBIRT(N,nit,alpha,beta,phi,theta,gamma0,gamma1,model="beta")
```

---

repBIRT — *Simulate replicated datasets using the posterior parameter samples from a BoundIRT-object*

---

## Description

To facilitate posterior predictive model checking, this function can be used to simulate replicated datasets for each of the samples from the posterior parameter distribution in a `BoundIRT`-object.

## Usage

```
repBIRT(x, ...)
## S3 method for class 'BoundIRT'
repBIRT(x, nrep=1000, silent=FALSE, ...)
```

## Arguments

| | |
|---|---|
| `x` | A `BoundIRT`-object |
| `nrep` | Number of replicated datasets to produce. See **Details**. |
| `silent` | If TRUE no details are displayed about the progress |
| `...` | Currently unused. |

## Details

`repBIRT` is a wrapper around `rBIRT` extracting the posterior parameter samples, and simulating data for a subset of the samples. That is, for many posterior predictive purposes, not all posterior parameter samples are needed. Therefore, only the last `nrep` samples will be used. Setting `nrep` to be equal to the total number of posterior samples will use all of them.

## Value

An object of class `repBIRT`, which is a list containing:

| | |
|---|---|
| `z_rep` | An array of size `nrep` by `N` by `nit` containing the replicated datasets. |
| `z` | The `N` by `nit` matrix of observed item scores used to fit the model (missing values coded as `NA`). |
| `N` | The number of persons. |
| `nit` | The number of items. |
| `model` | The fitted model. |
| `inflated` | The type of zero-one inflation used. |

`print.repBIRT`, `summary.repBIRT` and `plot.repBIRT` methods are available for `repBIRT` objects: `print.repBIRT` gives a short overview of the replications, `summary.repBIRT` gives the mean, sd and percentiles of the per-item replication averages, and `plot.repBIRT` produces a `bayesplot::ppc_dens_overlay` density overlay plot for the requested item(s).

## Author(s)

Dylan Molenaar <d.molenaar@uva.nl>

## References

Gabry J, Simpson D, Vehtari A, Betancourt M, Gelman A (2019). Visualization in Bayesian workflow. *Journal of the Royal Statistical Society, A*, **182**, 389-402. doi:10.1111/rssa.12378

## See Also

`fitBIRT` to estimate various bounded continuous IRT models to data.

`plot.BoundIRT` to plot the item and test information curves and predicted densities for the different models.

`coef.BoundIRT` to extract item parameter estimates from a `BoundIRT` object.

`loglikBIRT` to calculate the log-likelihood for each person and each posterior sample. Needed to calculate WAIC and LOOIC for a given model.

`bridgeBIRT` to use bridge sampling to calculate the marginal loglikelihood for a model in a `BoundIRT` object.

`latregBIRT` to conduct a latent regression of the person parameters on some observed predictor variables.

## Examples

```
#load the Abasement data and fitting results for the Beta IRT model
data(Abasement)
data(out_beta)

#generate 10 replicated datasets
z_rep=repBIRT(out_beta,nrep=10)
z_rep

#mean, sd and percentiles of the replications, per item
summary(z_rep)

#these can be used for e.g., posterior predictive checks
plot(z_rep,items=1)      #density overlay for item 1
plot(z_rep,items=1:2)    #density overlay for items 1 and 2
```

# Index