

# Package ‘VersIRT’

September 22, 2019

**Type** Package

**Title** Estimate versatile types of IRT models and some useful tools.

**Version** 0.1.2

**Author** Susumu Shikano

**Maintainer** The package maintainer <susumu.shikano@uni-konstanz.de>

**Description** A package to estimate versatile types of item response theoretic (IRT) models.

**License** What license is it under?

**Depends** rjags, runjags, shapes, MASS

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.1

**NeedsCompilation** no

## R topics documented:

binIRT . . . . .	2
data.prep . . . . .	3
find.target . . . . .	3
icc . . . . .	4
identification . . . . .	5
information . . . . .	5
item.plot . . . . .	6
one.dim.plot.ci . . . . .	7
ord.irt.estimate . . . . .	7
rcIRTsq . . . . .	8
target.rotation . . . . .	9
theta.plot . . . . .	9
two.dim.plot.ci . . . . .	10
<b>Index</b>	<b>12</b>

---

binIRT

*This function runs MCMC to obtain posteriors.*


---

## Description

This function runs MCMC to obtain posteriors.

## Usage

```
binIRT(data = NULL, n.parameters = 2, n.chains = 3, n.burnin = 100,
       n.adapt = 100, n.sample = 100, n.thin = 1, n.dim = 1,
       parallel = TRUE)
```

## Arguments

data	A matrix with binary entries (0 or 1). Columns are items, rows are subjects.
n.parameters	Number of the parameters to be estimated. It can be 1, 2, or 3. Default is 2.
n.chains	Number of chains. Default is 3.
n.burnin	Number of iterations for burn-in. Default is 100.
n.adapt	Number of iterations for adaptive phase. Default is 100.
n.sample	Number of iterations to be collected from the posterior. Default is 100.
n.thin	Thinning interval. Default is 1.
n.dim	Number of dimension. Default is 1.
parallel	specifies whether to parallelize the computation. Default is TRUE.

## Value

An unidentified posterior.

## Examples

```
library(VersIRT)
data("example_binIRT")

cleaned.data <- data.prep(test.data)

raw.posterior <- binIRT(data=as.matrix(cleaned.data$data),
                       n.chains=3,n.adapt=500,n.burnin=500,n.sample=1000)

identified.posterior <- identification(raw.posterior)

item.plot(identified.posterior,param="alpha")
item.plot(identified.posterior,param="beta")

icc(identified.posterior,item="X_6alpha",ci.level=0.95,only.items=FALSE)
```

---

data.prep	<i>This function excludes items and/or subjects which are too homogeneous. This is only for binary data.</i>
-----------	--

---

## Description

This function excludes items and/or subjects which are too homogeneous. This is only for binary data.

## Usage

```
data.prep(data = NULL, threshold.item = 0.99,
          threshold.subject = 0.99, threshold.na = 0.99)
```

## Arguments

data	A matrix with binary entries (0 or 1). Columns are items, rows are subjects.
threshold.item	A value which determines whether to exclude individual items. If threshold.item=0.99 (default), the items with an entry's share being higher than 0.99 will be excluded.
threshold.subject	A value which determines whether to exclude individual subjects. If threshold.item=0.99 (default), the subjects with an entry's share being higher than 0.99 will be excluded.
threshold.na	A value determining whether to exclude individual subjects based on the number of missings. If threshold.na=0.99 (default), the subjects with an share of missing being higher than 0.99 will be excluded.

## Value

A list with a matrix after exclusion of items/subjects based on the thresholds (data) and two vectors (included.items and included.subjects) indicating which items and subjects were kept in the output matrix.

## Examples

```
library(VersIRT)
data("example_binIRT")

cleaned.data <- data.prep(test.data)
```

---

find.target	<i>This function find a target iteration.</i>
-------------	---

---

## Description

This function find a target iteration.

**Usage**

```
find.target(theta.array = NULL, n.sample.iter = NULL,
            graphic = FALSE)
```

**Arguments**

`theta.array`      The array of theta.

`n.sample.iter`    Number of randomly chosen iterations, among which the target iteration is searched.

**Value**

The target iteration of theta.

---

<code>icc</code>	<i>This function generates the item characteristic curve (ICC).</i>
------------------	---

---

**Description**

This function generates the item characteristic curve (ICC).

**Usage**

```
icc(identified.posterior = NULL, item = NULL, type = "ICC",
    dimension = 1, only.items = TRUE, ci.level = 0.95,
    xlab = "Latent Scale", ylab = "Response probability", main = NULL)
```

**Arguments**

`identified.posterior`      Object produced by identification(...)

`item`                      The name of the item whose ICC should be presented.

`type`                      The type of the curve. Default is "ICC". For the ordinal IRT, "IOCC" (Item Operation Characteristic Curves) is possible, as well.

`dimension`                determines which dimensions to be plotted. Default is 1.

`only.items`                IF FALSE, the distribution of theta estimates will be also plotted.

`ci.level`                 The width of credible interval. Default is 0.95.

`xlab`                      Default is "Latent Scale".

`ylab`                      Default is "Response probability".

`main`                      Default is the name of item.

**Value**

returns the item characteristic curve

---

identification	<i>This function identifies the parameters.</i>
----------------	---

---

### Description

This function identifies the parameters.

### Usage

```
identification(posterior = NULL, n.sample.iter = 10,
  targetvec = NULL, item.dir.identification = NULL)
```

### Arguments

posterior	A posterior object.
n.sample.iter	Number of randomly chosen iterations, among which the target iteration is searched.
targetvec	Target vector on which the theta will be rotated. Default is NULL. If specified with 'PCM', the two dimensional space will be rotated so that the variance of theta on the first dimension is maximized. This is only for a two dimensional model relevant.
item.dir.identification	determines the item number, based on which the latent dimension's direction is identified. The specified item is fixed on the positive side. Relevant only for the 1 dimensional model. Deafult is NULL.

### Value

An identified posterior object. This is a list of identified estimation results.

---

information	<i>This function computes information.</i>
-------------	--

---

### Description

This function computes information.

### Usage

```
information(identified.posterior = NULL)
```

### Arguments

identified.posterior	Object produced by identification(...)
item	The name of the item whose ICC should be presented.
only.items	IF FALSE, the distribution of theta estimates will be also plotted.
ci.level	The width of credible interval. Deafult is 0.95.
xlab	Default is "Latent Scale".
ylab	Default is "Response probability".
main	Default is the name of item.

**Value**

returns the item characteristic curve

---

item.plot	<i>This function generates item parameter graphics</i>
-----------	--

---

**Description**

This function generates item parameter graphics

**Usage**

```
item.plot(identified.posterior = NULL, item.colors = NULL,
  param = "both", which.dim = "both", item.sort = FALSE,
  item.names = NULL, legend.pos = "bottomright", no.legend = FALSE,
  xlim = NULL, ylim = NULL, line.type = NULL, line.width = 2,
  plot.density = 0.01, title = NULL, xlab = NULL, ylab = NULL,
  legend.cex = 1, cex = 1)
```

**Arguments**

identified.posterior	Object produced by identification(...).
item.colors	Colors for individual items. Default is NULL. If no color is given, it is replaced by a random color selection for these graphs.
param	specifies which item-specific parameters will be plotted. If param == "both", alpha and beta parameters are plotted. If param == "alpha" or param == "beta", only the corresponding parameter will be plotted.
which.dim	Only relevant for a two-dimensional model. If 'both' (default), both dimensions will be plotted. If 'dim1' or 'dim2', only the specified dimension will be plotted.
item.sort	If TRUE, the item will be sorted in order of the posterior median. #' This option is relevant only for the plot of a single parameter at one dimension. Default is FALSE.
item.names	Labels of individual items. If not specified, it takes the column (row) index of the original data matrix.
legend.pos	Position of the labels. Possible values are "topright", "topleft", "bottomright", and "bottomleft". The default is "bottomright".
no.legend	If TRUE, the labels are suppressed. Default is FALSE.
xlim	A vector determining the range of the x-axis. Default is NULL.
ylim	A vector determining the range of the y-axis. Default is NULL.
line.type	specifies the type of line. If NULL, all lines are solid (lty=1).
line.width	specifies the width of line. Default is 2.
plot.density	A vector of values specifying the densities to be plotted.
title	A title for the plot. Default is NULL, which suppresses the title.
xlab	Label for the first dimension.
ylab	Label for the second dimension.
legend.cex	cex-factor for the legend.
cex	cex-factor for the points.

**Value**

returns the selected figure. For type="item" there are some possible graphics depending on param.

---

one.dim.plot.ci	<i>This function generates a one-dimensional plot with credible intervals.</i>
-----------------	--

---

**Description**

This function generates a one-dimensional plot with credible intervals.

**Usage**

```
one.dim.plot.ci(points = NULL, ci = NULL, targetvec = NULL,
  xlim = NULL, ylim = NULL, xlab = NULL, ylab = NULL,
  item.names = NULL, plot.type = "p", pch = 19, item.col = "black",
  pos = 4, main = NULL, item.sort = FALSE)
```

**Arguments**

points	A vector with point estimates.
ci	A matrix with interval bounds. The matrix should have two columns.
xlim	The range of the x-axis.
ylim	The range of the y-axis.
xlab	The label of the x-axis.
ylab	The label of the y-axis.
item.names	The name of individual items
pch	The type of dot. Default is 19.
item.col	The color of items. A single value or a vector.
pos	The position of the item names.

**Value**

A one-dimensional plot.

---

ord.irt.estimate	<i>This function runs MCMC to obtain posteriors of an ordered IRT model.</i>
------------------	--

---

**Description**

This function runs MCMC to obtain posteriors of an ordered IRT model.

**Usage**

```
ord.irt.estimate(data = NULL, n.chains = 3, n.burnin = 100,
  n.adapt = 100, n.sample = 100, n.thin = 1, n.dim = 1,
  parallel = TRUE)
```

**Arguments**

<code>data</code>	A matrix with ordinal entries (1,2,3,...). Columns are items, rows are subjects.
<code>n.chains</code>	Number of chains
<code>n.burnin</code>	Number of iterations for burn in.
<code>n.adapt</code>	Number of iterations for adaptation phase.
<code>n.sample</code>	Number of posterior sample per chain to be stored.
<code>n.thin</code>	Thinning interval. Default is 1.
<code>n.dim</code>	Number of dimensions. Default is 1.
<code>parallel</code>	specifies whether to parallelize the computation. Default is TRUE.

**Value**

An unidentified posterior.

---

<code>rcIRTsq</code>	<i>This function runs MCMC to obtain posteriors for an IRT model for roll call data with options to fix status quo.</i>
----------------------	---

---

**Description**

This function runs MCMC to obtain posteriors for an IRT model for roll call data with options to fix status quo.

**Usage**

```
rcIRTsq(data = NULL, cluster.sq = NULL, n.chains = 3,
  n.burnin = 100, n.adapt = 100, n.sample = 100, n.thin = 1,
  n.dim = 1, parallel = TRUE)
```

**Arguments**

<code>data</code>	A matrix with binary entries (0 or 1). Columns are items, rows are subjects.
<code>cluster.sq</code>	A vector which clusters individual columns for each status quo estimation.
<code>n.chains</code>	Number of chains. Default is 3.
<code>n.burnin</code>	Number of iterations for burn-in. Default is 100.
<code>n.adapt</code>	Number of iterations for adaptive phase. Default is 100.
<code>n.sample</code>	Number of iterations to be collected from the posterior. Default is 100.
<code>n.thin</code>	Thinning interval. Default is 1.
<code>n.dim</code>	Number of dimension. Default is 1.
<code>parallel</code>	specifies whether to parallelize the computation. Default is TRUE.

**Value**

An unidentified posterior.



---

target.rotation	<i>This function conducts target rotations to the target. For internal use (the function 'identification').</i>
-----------------	---

---

### Description

This function conducts target rotations to the target. For internal use (the function 'identification').

### Usage

```
target.rotation(theta.array = NULL, targetvec = NULL,
  no.target = NULL, target = NULL)
```

### Arguments

theta.array	The array of theta.
targetvec	The vector to which the first dimension of the target will be rotated. If "PCM" the target, the target will be rotated so that theta's variance is maximum at the first dimension.
no.target	Number of randomly chosen iterations, among which the target iteration is searched.
target	The target iteration.

### Value

The target iteration of theta.

---

theta.plot	<i>This function plots theta estimates.</i>
------------	---

---

### Description

This function plots theta estimates.

### Usage

```
theta.plot(posterior = NULL, theta.colors = NULL,
  theta.labels = NULL, legend.pos = "bottomright", no.legend = FALSE,
  xlim = NULL, ylim = NULL, line.type = NULL, line.width = 2,
  plot.density = 0.01, title = NULL, xlab = NULL, ylab = NULL,
  legend.cex = 1, cex = 1)
```

**Arguments**

posterior	An identified posterior object.
theta.colors	Colors for individual subjects. Default is NULL. If no color is given, it is replaced by a random color selection for these graphs
theta.labels	Labels of individual subjects. If not specified, it takes the column (row) index of the original data matrix.
legend.pos	Position of the labels. Possible values are "topright", "topleft", "bottomright", and "bottomleft". The default is "bottomright".
no.legend	If TRUE, the labels are suppressed. Default is FALSE.
xlim	A vector determining the range of the x-axis. Default is NULL.
ylim	A vector determining the range of the y-axis. Default is NULL.
line.type	specifies the type of line. If NULL, all lines are solid (lty=1).
line.width	specifies the width of line. Default is 2.
plot.density	A vector of values specifying the densities to be plotted.
title	A title for the plot. Default is NULL, which suppresses the title.
xlab	Label for the first dimension.
ylab	Label for the second dimension.
legend.cex	cex-factor for the legend.
cex	cex-factor for the points.

**Value**

returns the figure.

---

two.dim.plot.ci	<i>This function generates a two-dimensional plot with credible intervals.</i>
-----------------	--

---

**Description**

This function generates a two-dimensional plot with credible intervals.

**Usage**

```
two.dim.plot.ci(points = NULL, ci = NULL, targetvec = NULL,
  xlim = NULL, ylim = NULL, xlab = NULL, ylab = NULL,
  item.names = NULL, plot.type = "p", pch = 19, item.col = "black",
  pos = NULL, main = NULL)
```

**Arguments**

points	A matrix with point estimates. The number of columns must be two.
ci	An array with interval bounds. The array should have the following dimensions: no of items, 2 for dimensions and 2 for both bounds.
xlim	The range of the x-axis.
ylim	The range of the y-axis.

<code>xlab</code>	The label of the x-axis.
<code>ylab</code>	The label of the y-axis.
<code>item.names</code>	The name of individual items
<code>pch</code>	The type of dot. Default is 19.
<code>item.col</code>	The color of items. A single value or a vector.
<code>pos</code>	The position of the item names.

**Value**

A two-dimensional plot.

# Index

binIRT, [2](#)

data.prep, [3](#)

find.target, [3](#)

icc, [4](#)

identification, [5](#)

information, [5](#)

item.plot, [6](#)

one.dim.plot.ci, [7](#)

ord.irt.estimate, [7](#)

rcIRTsq, [8](#)

target.rotation, [9](#)

theta.plot, [9](#)

two.dim.plot.ci, [10](#)