Package: irtpwr (via r-universe)

September 8, 2024

Title Power Analysis for IRT Models Using the Wald, LR, Score, and Gradient Statistics

Version 1.0.3.9000

Description Implementation of analytical and sampling-based power analyses for the Wald, likelihood ratio (LR), score, and gradient tests. Can be applied to item response theory (IRT) models that are fitted using marginal maximum likelihood estimation. The methods are described in our paper (Zimmer et al. (2022) <doi:10.1007/s11336-022-09883-5>).

License GPL (>= 3)

URL `https://github.com/flxzimmer/irtpwr`

BugReports `https://github.com/flxzimmer/irtpwr/issues`

Encoding UTF-8

RoxygenNote 7.2.3

Suggests testthat, knitr, rmarkdown

Imports mirt, Deriv, digest, spatstat.random, ggplot2, methods

VignetteBuilder knitr

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

RemoteUrl https://github.com/flxzimmer/irtpwr

RemoteRef HEAD

RemoteSha faa97fd860949c00a7234c7c269c386937a41ff0

Contents

calc.time 2
irtpwr 2
load.functions 4
pars.long 4
plot.irtpwrresult 5
print.summary.irtpwrresult 6
setup.hypothesis 7
summary.irtpwrresult 7

Index 9

calc.time *Calculate the computation time needed for the analytical method*

Description

Calculate the computation time needed for the analytical method

Usage

```
calc.time(hyp, n.items)
```

Arguments

hyp	Hypothesis object as created by the setup.hypothesis function
n.items	Number of items

Value

Numeric, estimated time in seconds

Examples

```
library(mirt)
dat <- expand.table(LSAT7)
mirtfit <- mirt(dat,1,verbose = FALSE)
hyp <- setup.hypothesis(type = '1PLvs2PL', altpars = mirtfit)
calc.time(hyp,n.items=7)
```

irtpwr *Perform Power Analysis*

Description

Perform analytical or sampling-based power analysis for the Wald, LR, score, or gradient statistic.

Usage

```
irtpwr(
  hyp,
  stat = c("Wald", "LR", "Score", "Gradient"),
  method = "analytical",
  sampling.npers = 10^5,
  approx.npers = 10^5,
  SE.type = "Oakes",
  sampling.mat = "ApproxFisher",
  power = NULL,
  N = NULL,
  alpha = NULL
)
```

Arguments

`hyp` Hypothesis Object created by the setup.hypothesis function

`stat` character vector containing the statistics to be calculated. Options are ’Wald’,’LR’,’Score’, and ’Gradient’. By default, all statistics are included

`method` character, indicating the method used. The options are ’analytical’(default) for the analytical power analysis method or ’sampling’ for the sampling-based method. The sampling-based method is generally recommended for higher numbers of items.

`sampling.npers` integer, sample size for the sampling-based approach. An artificial data set of this size is generated to fit a model and later estimate the noncentrality parameter from.

`approx.npers` integer, sample size for approximating the Fisher expected information matrix in the sampling-based approach. An artificial data set is calculated of this size to calculate the Fisher expected information matrix from. In contrast to the data created with the sampling.npers sample size, this sample is not used to fit a model.

`SE.type` Method for calculation of the observed information matrix used for calculating the statistics in the sampling-based approach (’Oakes’ by default). Another option is ’Fisher’.

`sampling.mat` Approach to calculate the information matrix used for calculating the statistics in the sampling-based approach. By default (’ApproxFisher’), an sampling-based approximation of the expected Fisher matrix is calculated using an observed information matrix of the type SE.type

`power` numeric, statistical power for which the necessary sample size is calculated

`N` integer, sample size for which the statistical power is calculated.

`alpha` numeric, alpha level

Value

function returns an object of class irtpwrresult

Examples

```
library(mirt)
dat <- expand.table(LSAT7)
mirtfit <- mirt(dat,1,verbose = FALSE)
hyp <- setup.hypothesis(type = '1PLvs2PL', altpars = mirtfit)
res <- irtpwr(hyp=hyp,alpha=.05,power =.8)
summary(res)
```

load.functions *Load IRT model functions and derivatives*

Description

This is a helper function used to generate custom hypotheses. See the 'adding_hypotheses' vignette.

Usage

```
load.functions(model, multi = FALSE)
```

Arguments

model	character, Desired Model (2PL,3PL, GPCM)
multi	logical, multidimensional model if TRUE (available for 2PL)

Value

nothing

Examples

```
funs = load.functions('2PL')
```

pars.long *Transform parameters to a longer format*

Description

This is a helper function used to generate custom hypotheses. See the 'adding_hypotheses' vignette.

Usage

```
pars.long(pars, itemtype, from.mirt = FALSE)
```

Arguments

`pars`	list of parameters. Can also be coefficients from a model fitted by mirt. In this case, the from.mirt argument has to be set to TRUE
`itemtype`	character, type of the item as string, e.g. '2PL'
`from.mirt`	logical, treat as coefficients from a model fitted by mirt if TRUE

Value

numeric vector

Examples

```
pars = list(a= c(1,1,1),d=c(0,0,0))
pars.long(pars,itemtype='2PL')
```

`plot.irtpwrresult` *Plot Power Curves*

Description

Plot the power curves using the resulting object of the irtpwr function.

Usage

```
## S3 method for class 'irtpwrresult'
plot(x, bounds = NULL, ...)
```

Arguments

`x`	Object of class irtpwrresult as created by the irtpwr function.
`bounds`	integer vector. the first entry is the lower bound of the x-axis in the plot (sample size). The second entry is the upper bound. By default, these values are chosen to cover a power range of .5 to .95.
`...`	additional arguments to be passed.

Value

A ggplot object

Examples

```
library(mirt)
dat <- expand.table(LSAT7)
mirtfit <- mirt(dat,1,verbose = FALSE)
hyp <- setup.hypothesis(type = '1PLvs2PL', altpars = mirtfit)
res <- irtpwr(hyp=hyp,alpha=.05,power =.8)
plot(res)
```

print.summary.irtpwrresult

Print Summary of the search result

Description

Print Summary of the search result

Usage

```
## S3 method for class 'summary.irtpwrresult'
print(x, ...)
```

Arguments

`x` Object of class irtpwrresult as created by the irtpwr function

`...` additional arguments to be passed

Value

An object of class summary.irtpwrresult

Examples

```
library(mirt)
dat <- expand.table(LSAT7)
mirtfit <- mirt(dat,1,verbose = FALSE)
hyp <- setup.hypothesis(type = '1PLvs2PL', altpars = mirtfit)
res <- irtpwr(hyp=hyp,alpha=.05,power =.8)
summary(res)
```

setup.hypothesis *Setup null and alternative hypothesis*

Description

Setup null and alternative hypothesis

Usage

```
setup.hypothesis(type, altpars = NULL, nullpars = NULL)
```

Arguments

type	preset, e.g. '1PLvs2PL'. Either a string for an existing preset or a list with the required functions for a custom preset. See the section extending the package for a tutorial on how to create a custom preset.
altpars	List of model parameters following the alternative hypothesis. The format depends on the preset. When parameters are derived from observed data by mirt, the coef_short function can be used to convert the parameters to the right format.
nullpars	List of model parameters following the null hypothesis. The format depends on the preset. Null parameters are not necessary for some hypothesis presets, e.g. 1PLvs2PL.

Value

a list specifying the hypothesis for usage in further functions, e.g. estimation of the noncentrality parameters.

Examples

```
library(mirt)
dat <- expand.table(LSAT7)
mirtfit <- mirt(dat,1,verbose = FALSE)
hyp <- setup.hypothesis(type = '1PLvs2PL', altpars = mirtfit)
```

summary.irtpwrresult *Summary of the Power Analysis*

Description

Output the resulting power or sample size for each statistic

Usage

```
## S3 method for class 'irtpwrresult'
summary(object, ..., power = NULL, N = NULL, alpha = NULL)
```

Arguments

`object`	Object of class irtpwrresult as created by the irtpwr function
`...`	additional arguments to be passed
`power`	numeric, statistical power for which the necessary sample size is calculated
`N`	integer, sample size for which the statistical power is calculated.
`alpha`	numeric, alpha level

Value

An object of class summary.irtpwrresult

Examples

```
library(mirt)
dat <- expand.table(LSAT7)
mirtfit <- mirt(dat,1,verbose = FALSE)
hyp <- setup.hypothesis(type = '1PLvs2PL', altpars = mirtfit)
res <- irtpwr(hyp=hyp,alpha=.05,power =.8)
summary(res)
```

Index

calc.time, 2

irtpwr, 2

load.functions, 4

pars.long, 4
plot.irtpwrresult, 5
print.summary.irtpwrresult, 6

setup.hypothesis, 7
summary.irtpwrresult, 7