# Package ‘flexmet’

May 8, 2026

**Type** Package

**Title** Flexible Latent Trait Metrics using the Filtered Monotonic Polynomial Item Response Model

**Version** 1.1

**Description** Application of the filtered monotonic polynomial (FMP) item response model to flexibly fit item response models. The package includes tools that allow the item response model to be build on any monotonic transformation of the latent trait metric, as described by Feuerstahler (2019) <doi:10.1007/s11336-018-9642-9>.

**License** GPL-3

**Suggests** MonoPoly, knitr, rmarkdown, mirt

**Encoding** UTF-8

**RoxygenNote** 7.1.1

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Leah Feuerstahler [aut, cre]

**Maintainer** Leah Feuerstahler <lfeuerstahler@fordham.edu>

**Repository** CRAN

**Date/Publication** 2021-07-13 22:10:02 UTC

## Contents

---

`b2greek` *Find the Greek-Letter Parameterization corresponding to a b Vector of Item Parameters*

---

## Description

Convert the b vector of item parameters (polynomial coefficients) to the corresponding Greek-letter parameterization (used to ensure monotonicitiy).

## Usage

```
b2greek(bvec, ncat = 2, eps = 1e-08)
```

## Arguments

| | |
|---|---|
| `bvec` | b vector of item parameters (i.e., polynomial coefficients). |
| `ncat` | Number of response categories (first ncat - 1 elements of bvec are intercepts) |
| `eps` | Convergence tolerance. |

## Details

See greek2b for more information about the b (polynomial coefficient) and Greek-letter parameterizations of the FMP model.

## Value

A vector of item parameters in the Greek-letter parameterization.

## References

Liang, L., & Browne, M. W. (2015). A quasi-parametric method for fitting flexible item response functions. *Journal of Educational and Behavioral Statistics*, *40*, 5–34. doi: 10.3102/1076998614556816

## See Also

greek2b

## Examples

```
(bvec <- greek2b(xi = 0, omega = 1, alpha = c(.1, .1), tau = c(-2, -2)))
## 0.00000000  2.71828183 -0.54365637  0.29961860 -0.03950623  0.01148330

(b2greek(bvec))
##  0.0  1.0  0.1 -2.0  0.1 -2.0
```

---

fmp *Estimate FMP Item Parameters*

---

## Description

Estimate FMP item parameters for a single item using user-specified theta values (fixed-effects) using fmp_1, or estimate FMP item parameters for multiple items using fixed-effects or random-effects with fmp.

## Usage

```
fmp_1(
  dat,
  k,
  tsur,
  start_vals = NULL,
  method = "CG",
  priors = list(xi = c("none", NaN, NaN), omega = c("none", NaN, NaN), alpha =
    c("none", NaN, NaN), tau = c("none", NaN, NaN)),
  ...
)

fmp(
  dat,
  k,
  start_vals = NULL,
  em = TRUE,
  eps = 1e-04,
  n_quad = 49,
  method = "CG",
  max_em = 500,
  priors = list(xi = c("none", NaN, NaN), omega = c("none", NaN, NaN), alpha =
    c("none", NaN, NaN), tau = c("none", NaN, NaN)),
  ...
)
```

## Arguments

- `dat` Vector of item responses for N (# subjects) examinees. Binary data should be coded 0/1, and polytomous data should be coded 0, 1, 2, etc.
- `k` Vector of item complexities for each item, see details. If k < ncol(dat), k's will be recycled.
- `tsur` Vector of N (# subjects) surrogate theta values.
- `start_vals` Start values, For fmp_1, a vector of length 2k+2 in the following order:

  If k = 0: (xi_1, ..., x_C_i - 1, omega)

  If k = 1: (xi_1, ..., x_C_i - 1, omega, alpha1, tau1)

  If k = 2: (xi_1, ..., x_C_i - 1, omega, alpha1, tau1, alpha2, tau2)

  and so forth. For fmp, add start values for item 1, followed by those for item 2, and so forth. For further help, first fit the model without start values, then inspect the outputted parmat data frame.
- `method` Optimization method passed to optim.
- `priors` List of prior information used to estimate the item parameters. The list should have up to 4 elements named xi, omega, alpha, tau. Each list should be a vector of length 3: the name of the prior distribution ("norm" or "none"), the first parameter of the prior distribution, and the second parameter of the prior distribution. Currently, "norm" and 'none" are the only available prior distributions.
- `em` If "mirt", use the mirt (Chalmers, 2012) package to estimate item parameters. If TRUE, random-effects estimation is used via the EM algorithm. If FALSE, fixed effects estimation is used with theta surrogates.
- `eps` Covergence tolerance for the EM algorithm. The EM algorithm is said to converge is the maximum absolute difference between parameter estimates for successive iterations is less than eps. Ignored if em = FALSE.
- `n_quad` Number of quadrature points for EM integration. Ignored if em = FALSE
- `max_em` Maximum number of EM iterations (for em = TRUE only).
- `...` Additional arguments passed to optim (if em != "mirt") or mirt (if em == "mirt").

## Details

The FMP item response function for a single item $i$ with responses in categories $c = 0, ..., C_i - 1$ is specified using the composite function,

$$P(X_i = c|\theta) = exp(\sum_{v=0}^{c}(b_0 i_v + m_i(\theta)))/(\sum_{u=0}^{C_i-1} exp(\sum_{v=0}^{u}(b_{0i_v} + m_i(\theta))))$$

where $m(\theta)$ is an unbounded and monotonically increasing polynomial function of the latent trait $\theta$, excluding the intercept (s).

The item complexity parameter $k$ controls the degree of the polynomial:

$$m(\theta) = b_1\theta + b_2\theta^2 + ... + b_{2k+1}\theta^{2k+1},$$

where $2k + 1$ equals the order of the polynomial, $k$ is a nonnegative integer, and

$$b = (b1, ..., b(2k + 1))'$$

are item parameters that define the location and shape of the IRF. The vector $b$ is called the b-vector parameterization of the FMP Model. When $k = 0$, the FMP IRF equals either the slope-threshold parameterization of the two-parameter item response model (if maxncat = 2) or Muraki's (1992) generalized partial credit model (if maxncat > 2).

For $m(\theta)$ to be a monotonic function, the FMP IRF can also be expressed as a function of the vector

$$\gamma = (\xi, \omega, \alpha_1, \tau_1, \alpha_2, \tau_2, \cdots \alpha_k, \tau_k)'.$$

The $\gamma$ vector is called the Greek-letter parameterization of the FMP model. See Falk & Cai (2016a), Feuerstahler (2016), or Liang & Browne (2015) for details about the relationship between the b-vector and Greek-letter parameterizations.

## Value

| | |
|---|---|
| `bmat` | Matrix of estimated b-matrix parameters, each row corresponds to an item, and contains b0, b1, ...b(max(k)). |
| `parmat` | Data frame of parameter estimation information, including the Greek-letter parameterization, starting value, and parameter estimate. |
| `k` | Vector of item complexities chosen for each item. |
| `log_lik` | Model log likelihood. |
| `mod` | If em == "mirt", the mirt object. Otherwise, optimization information, including output from optim. |
| `AIC` | Model AIC. |
| `BIC` | Model BIC. |

## References


Chalmers, R. P. (2012). mirt: A multidimensional item response theory package for the R environment. *Journal of Statistical Software*, *48*, 1–29. doi: 10.18637/jss.v048.i06

Elphinstone, C. D. (1983). A target distribution model for nonparametric density estimation. *Communication in Statistics–Theory and Methods*, *12*, 161–198. doi: 10.1080/03610928308828450

Elphinstone, C. D. (1985). *A method of distribution and density estimation* (Unpublished dissertation). University of South Africa, Pretoria, South Africa.

Falk, C. F., & Cai, L. (2016a). Maximum marginal likelihood estimation of a monotonic polynomial generalized partial credit model with applications to multiple group analysis. *Psychometrika*, *81*, 434–460. doi: 10.1007/s1133601494287

Falk, C. F., & Cai, L. (2016b). Semiparametric item response functions in the context of guessing. *Journal of Educational Measurement*, *53*, 229–247. doi: 10.1111/jedm.12111

Feuerstahler, L. M. (2016). *Exploring alternate latent trait metrics with the filtered monotonic polynomial IRT model* (Unpublished dissertation). University of Minnesota, Minneapolis, MN. `http://hdl.handle.net/11299/182267`

Feuerstahler, L. M. (2019). Metric Transformations and the Filtered Monotonic Polynomial Item Response Model. *Psychometrika*, *84*, 105–123. doi: 10.1007/s1133601896429

Liang, L. (2007). *A semi-parametric approach to estimating item response functions* (Unpublished dissertation). The Ohio State University, Columbus, OH. Retrieved from https://etd.ohiolink.edu/

Liang, L., & Browne, M. W. (2015). A quasi-parametric method for fitting flexible item response functions. *Journal of Educational and Behavioral Statistics*, *40*, 5–34. doi: 10.3102/1076998614556816

Muraki, E. (1992). A generalized partial credit model: Application of an EM algorithm. *Applied Psychological Measurement*, *16*, 159–176. doi: 10.1177/014662169201600206

## Examples

```
set.seed(2345)
bmat <- sim_bmat(n_items = 5, k = 2, ncat = 4)$bmat

theta <- rnorm(50)
dat <- sim_data(bmat = bmat, theta = theta, maxncat = 4)

## fixed-effects estimation for item 1

tsur <- get_surrogates(dat)

# k = 0
fmp0_it_1 <- fmp_1(dat = dat[, 1], k = 0, tsur = tsur)

# k = 1
fmp1_it_1 <- fmp_1(dat = dat[, 1], k = 1, tsur = tsur)

## fixed-effects estimation for all items

fmp0_fixed <- fmp(dat = dat, k = 0, em = FALSE)

## random-effects estimation


fmp0_random <- fmp(dat = dat, k = 0, em = TRUE)


## random-effects estimation using mirt's estimation engine

fmp0_mirt <- fmp(dat = dat, k = 0, em = "mirt")
```

---

`get_surrogates` *Find Theta Surrogates*

---

## Description

Compute surrogate theta values as the set of normalized first principal component scores.

## Usage

```
get_surrogates(dat)
```

### Arguments

| | |
|---|---|
| `dat` | Matrix of binary item responses. |

### Details

Compute surrogate theta values as the normalized first principal component scores.

### Value

Vector of surrogate theta values.

### References

Liang, L., & Browne, M. W. (2015). A quasi-parametric method for fitting flexible item response functions. *Journal of Educational and Behavioral Statistics*, *40*, 5–34. doi: 10.3102/1076998614556816

### Examples

```
set.seed(2342)
bmat <- sim_bmat(n_items = 5, k = 2)$bmat

theta <- rnorm(50)
dat <- sim_data(bmat = bmat, theta = theta)

tsur <- get_surrogates(dat)
```

---

## greek2b — *Find the b Vector from a Greek-Letter Parameterization of Item Parameters.*

---

### Description

Convert the Greek-letter parameterization of item parameters (used to ensure monotonicitiy) to the b-vector parameterization (polynomial coefficients).

### Usage

```
greek2b(xi, omega, alpha = NULL, tau = NULL)
```

### Arguments

| | |
|---|---|
| `xi` | see details |
| `omega` | see details |
| `alpha` | see details, vector of length k, set to NULL if k = 0 |
| `tau` | see details, vector of length k, set to NULL if k = 0 |

## Details

For

$$m(\theta) = b_0 + b_1\theta + b_2\theta^2 + \cdots + b_{2k+1}\theta^{2k+1}$$

to be a monotonic function, a necessary and sufficient condition is that its first derivative,

$$p(\theta) = a_0 + a_1\theta + ... + a_{2k}\theta^{2k},$$

is nonnegative at all theta. Here, let

$$b_0 = \xi$$

be the constant of integration and

$$b_s = a_{s-1}/s$$

for $s = 1, 2, ..., 2k + 1$. Notice that $p(\theta)$ is a polynomial function of degree $2k$. A nonnegative polynomial of an even degree can be re-expressed as the product of k quadratic functions.

If $k \geq 1$:

$$p(\theta) = \exp\omega\Pi_{s=1}^{k}[1 - 2\alpha_s\theta + (\alpha_s^2 + \exp(\tau_s))\theta^2]$$

If $k = 0$:

$$p(\theta) = 0.$$

## Value

A vector of item parameters in the b parameterization.

## References

Liang, L., & Browne, M. W. (2015). A quasi-parametric method for fitting flexible item response functions. *Journal of Educational and Behavioral Statistics*, *40*, 5–34. doi: 10.3102/1076998614556816

## See Also

b2greek

## Examples

```
(bvec <- greek2b(xi = 0, omega = 1, alpha = .1, tau = -1))
## 0.0000000  2.7182818 -0.2718282  0.3423943

(b2greek(bvec))
##  0.0  1.0  0.1 -1.0
```

iif_fmp *FMP Item Information Function*

## Description

Find FMP item information for user-supplied item and person parameters.

## Usage

```
iif_fmp(theta, bmat, maxncat = 2, cvec = NULL, dvec = NULL)
```

## Arguments

| | |
|---|---|
| theta | Vector of latent trait parameters. |
| bmat | Items x parameters matrix of FMP item parameters (or a vector of FMP item parameters for a single item). |
| maxncat | Maximum number of response categories (the first maxncat - 1 columns of bmat are intercepts). |
| cvec | Optional vector of lower asymptote parameters. If cvec = NULL, then all lower asymptotes set to 0. |
| dvec | Optional vector of upper asymptote parameters. If dvec = NULL, then all upper asymptotes set to 1. |

## Value

Matrix of item information.

## Examples

```
# plot the IIF for a dichotomous item with k = 2

set.seed(2342)
bmat <- sim_bmat(n_items = 1, k = 2)$bmat

theta <- seq(-3, 3, by = .01)

information <- iif_fmp(theta = theta, bmat = bmat)

plot(theta, information, type = 'l')
```

`int_mat` *Numerical Integration Matrix*

## Description

Create a matrix for numerical integration.

## Usage

```
int_mat(
  distr = dnorm,
  args = list(mean = 0, sd = 1),
  lb = -4,
  ub = 4,
  npts = 10000
)
```

## Arguments

| | |
|---|---|
| `distr` | A density function with two user-specified parameters. Defaults to the normal distribution (dnorm), but any density function is permitted. |
| `args` | Named list of arguments to distr. |
| `lb` | Lower bound of range over which to numerically integrate. |
| `ub` | Upper bound of range over which to numerically integrate. |
| `npts` | Number of integration points. |

## Value

Matrix of two columns. Column 1 is a sequence of x-coordinates, and column 2 is a sequence of y-coordinates from a normalized distribution.

## See Also

rimse th_est_ml th_est_eap sl_link hb_link

@importFrom stats dnorm

## inv_poly *Polynomial Functions*

### Description

Evaluate a forward or inverse (monotonic) polynomial function.

### Usage

```
inv_poly(x, coefs, lb = -1000, ub = 1000)

fw_poly(y, coefs)
```

### Arguments

| | |
|---|---|
| x | Scalar polynomial function input. |
| coefs | Vector of coefficients that define a monotonic polynomial, see details. |
| lb | Lower bound of the search interval. |
| ub | Upper bound of the search interval. |
| y | Scalar polynomial function output. |

### Details

$$x = t_0 + t_1 y + t_2 y^2 + ...$$

Then, for coefs = $(t_0, t_1, t_2, ...)'$, this function finds the corresponding $y$ value (inv_poly) or $x$ value (fw_poly).

## irf_fmp *FMP Item Response Function*

### Description

Find FMP item response probabilities for user-supplied item and person parameters.

### Usage

```
irf_fmp(theta, bmat, maxncat = 2, returncat = NA, cvec = NULL, dvec = NULL)
```

### Arguments

| | |
|---|---|
| theta | Vector of latent trait parameters. |
| bmat | Items x parameters matrix of FMP item parameters (or a vector of FMP item parameters for a single item). |
| maxncat | Maximum number of response categories (the first maxncat - 1 columns of bmat are intercepts). |
| returncat | Response categories for which probabilities should be returned, 0,..., maxncat - 1. |
| cvec | Optional vector of lower asymptote parameters. If cvec = NULL, then all lower asymptotes set to 0. |
| dvec | Optional vector of upper asymptote parameters. If dvec = NULL, then all upper asymptotes set to 1. |

### Value

Matrix of item response probabilities.

### Examples

```
# plot the IRF for an item with 4 response categories and k = 2

set.seed(2342)
bmat <- sim_bmat(n_items = 1, ncat = 4, k = 2)$bmat

theta <- seq(-3, 3, by = .01)

probability <- irf_fmp(theta = theta, bmat = bmat,
                       maxncat = 4, returncat = 0:3)

plot(theta, probability[, , 1], type = 'l', ylab = "probability")
points(theta, probability[, , 2], type = 'l')
points(theta, probability[, , 3], type = 'l')
points(theta, probability[, , 4], type = 'l')
```

---

## linking *Linear and Nonlinear Item Parameter Linking*

---

### Description

Link two sets of FMP item parameters using linear or nonlinear transformations of the latent trait.

### Usage

```
sl_link(
  bmat1,
  bmat2,
  maxncat = 2,
  cvec1 = NULL,
  cvec2 = NULL,
  dvec1 = NULL,
  dvec2 = NULL,
  k_theta,
  int = int_mat(),
  ...
)

hb_link(
  bmat1,
  bmat2,
  maxncat = 2,
  cvec1 = NULL,
  cvec2 = NULL,
  dvec1 = NULL,
  dvec2 = NULL,
  k_theta,
  int = int_mat(),
  ...
)
```

### Arguments

| | |
|---|---|
| `bmat1` | FMP item parameters on an anchor test. |
| `bmat2` | FMP item parameters to be rescaled. |
| `maxncat` | Maximum number of response categories (the first maxncat - 1 columns of bmat1 and bmat2 are intercepts) |
| `cvec1` | Vector of lower asymptote parameters for the anchor test. |
| `cvec2` | Vector of lower asymptote parameters corresponding to the rescaled item parameters. |
| `dvec1` | Vector of upper asymptote parameters for the anchor test. |
| `dvec2` | Vector of upper asymptote parameters corresponding to the rescaled item parameters. |
| `k_theta` | Complexity of the latent trait transformation (k_theta = 0 is linear, k_theta > 0 is nonlinear). |
| `int` | Matrix with two columns, used for numerical integration. Column 1 is a grid of theta values, column 2 are normalized densities associated with the column 1 values. |
| `...` | Additional arguments passed to optim. |

## Details

The goal of item parameter linking is to find a metric transformation such that the fitted parameters for one test can be transformed to the same metric as those for the other test. In the Haebara approach, the overall sum of squared differences between the original and transformed individual item response functions is minimized. In the Stocking-Lord approach, the sum of squared differences between the original and transformed test response functions is minimized. See Feuerstahler (2016, 2019) for details on linking with the FMP model.

## Value

| | |
|---|---|
| `par` | (Greek-letter) parameters estimated by optim. |
| `value` | Value of the minimized criterion function. |
| `counts` | Number of function counts in optim. |
| `convergence` | Convergence criterion given by optim. |
| `message` | Message given by optim. |
| `tvec` | Vector of theta transformation coefficients $(t = t0, ...., t(2k_\theta + 1))$ |
| `bmat` | Transformed bmat2 item parameters. |

## References

Feuerstahler, L. M. (2016). *Exploring alternate latent trait metrics with the filtered monotonic polynomial IRT model* (Unpublished dissertation). University of Minnesota, Minneapolis, MN. http://hdl.handle.net/11299/182267

Feuerstahler, L. M. (2019). Metric Transformations and the Filtered Monotonic Polynomial Item Response Model. *Psychometrika*, *84*, 105–123. doi: 10.1007/s1133601896429

Haebara, T. (1980). Equating logistic ability scales by a weighted least squares method. *Japanese Psychological Research*, *22*, 144–149. doi: 10.4992/psycholres1954.22.144

Stocking, M. L., & Lord, F. M. (1983). Developing a common metric in item response theory. *Applied Psychological Measurement*, *7*, 201–210. doi: 10.1002/j.23338504.1982.tb01311.x

## Examples

```
set.seed(2342)
bmat <- sim_bmat(n_items = 10, k = 2)$bmat

theta1 <- rnorm(100)
theta2 <- rnorm(100, mean = -1)

dat1 <- sim_data(bmat = bmat, theta = theta1)
dat2 <- sim_data(bmat = bmat, theta = theta2)

# estimate each model with fixed-effects and k = 0
fmp0_1 <- fmp(dat = dat1, k = 0, em = FALSE)
fmp0_2 <- fmp(dat = dat2, k = 0, em = FALSE)

# Stocking-Lord linking
```

```
sl_res <- sl_link(bmat1 = fmp0_1$bmat[1:5, ],
                  bmat2 = fmp0_2$bmat[1:5, ],
                  k_theta = 0)


hb_res <- hb_link(bmat1 = fmp0_1$bmat[1:5, ],
                  bmat2 = fmp0_2$bmat[1:5, ],
                  k_theta = 0)
```

---

`rimse` *Root Integrated Mean Squared Difference Between FMP IRFs*

---

### Description

Compute the root integrated mean squared error (RIMSE) between two FMP IRFs.

### Usage

```
rimse(
  bvec1,
  bvec2,
  ncat = 2,
  c1 = NULL,
  d1 = NULL,
  c2 = NULL,
  d2 = NULL,
  int = int_mat()
)
```

### Arguments

| | |
|---|---|
| `bvec1` | Either a vector of FMP item parameters or a function corresponding to a non-FMP IRF. Functions should have exactly one argument, corresponding to the latent trait. |
| `bvec2` | Either a vector of FMP item parameters or a function corresponding to a non-FMP IRF. Functions should have exactly one argument, corresponding to the latent trait. |
| `ncat` | Number of response categories (first ncat - 1 elemnts of bvec1 and bvec2 are intercepts) |
| `c1` | Lower asymptote parameter for bvec1. Ignored if bvec1 is a function. |
| `d1` | Upper asymptote parameter for bvec1. Ignored if bvec1 is a function. |
| `c2` | Lower asymptote parameter for bvec2. Ignored if bvec2 is a function. |

d2 Upper asymptote parameter for bvec2. Ignored if bvec2 is a function.

int Matrix with two columns, used for numerical integration. Column 1 is a grid of theta values, column 2 are normalized densities associated with the column 1 values

## Value

Root integrated mean squared difference between two IRFs (dichotomous items) or expected item scores (polytomous items).

## References

Ramsay, J. O. (1991). Kernel smoothing approaches to nonparametric item characteristic curve estimation. *Psychometrika*, *56*, 611–630. doi: 10.1007/BF02294494

## Examples

```
set.seed(2342)
bmat <- sim_bmat(n_items = 2, k = 2, ncat = c(2, 5))$bmat

theta <- rnorm(500)
dat <- sim_data(bmat = bmat, theta = theta, maxncat = 5)

# k = 0
fmp0a <- fmp_1(dat = dat[, 1], k = 0, tsur = theta)
fmp0b <- fmp_1(dat = dat[, 2], k = 0, tsur = theta)


# k = 1
fmp1a <- fmp_1(dat = dat[, 1], k = 1, tsur = theta)
fmp1b <- fmp_1(dat = dat[, 2], k = 1, tsur = theta)


## compare estimated curves to the data-generating curve
rimse(fmp0a$bmat, bmat[1, -c(2:4)])
rimse(fmp0b$bmat, bmat[2, ], ncat = 5)


rimse(fmp1a$bmat, bmat[1, -c(2:4)])
rimse(fmp1b$bmat, bmat[2, ], ncat = 5)
```

---

sim_bmat *Randomly Generate FMP Parameters*

---

## Description

Generate monotonic polynomial coefficients for user-specified item complexities and prior distributions.

### Usage

```
sim_bmat(
  n_items,
  k,
  ncat = 2,
  xi_dist = list(runif, min = -1, max = 1),
  omega_dist = list(runif, min = -1, max = 1),
  alpha_dist = list(runif, min = -1, max = 0.5),
  tau_dist = list(runif, min = -3, max = 0)
)
```

### Arguments

| | |
|---|---|
| `n_items` | Number of items for which to simulate item parameters. |
| `k` | Either a scalar for the item complexity of all items or a vector of length n_items if different items have different item complexities. |
| `ncat` | Vector of length n_item giving the number of response categories for each item. If of length 1, all items will have the same number of response categories. |
| `xi_dist` | List of information about the distribution from which to randomly sample xi parameters. The first element should be a function that generates random deviates (e.g., runif or rnorm), and further elements should be named arguments to the function. |
| `omega_dist` | List of information about the distribution from which to randomly sample omega parameters. The first element should be a function that generates random deviates (e.g., runif or rnorm), and further elements should be named arguments to the function. |
| `alpha_dist` | List of information about the distribution from which to randomly sample alpha parameters. The first element should be a function that generates random deviates (e.g., runif or rnorm), and further elements should be named arguments to the function. Ignored if all k = 0. |
| `tau_dist` | List of information about the distribution from which to randomly sample tau parameters. The first element should be a function that generates random deviates (e.g., runif or rnorm), and further elements should be named arguments to the function. Ignored if all k = 0. |

### Details

Randomly generate FMP item parameters for a given k value.

### Value

| | |
|---|---|
| `bmat` | Item parameters in the b parameterization (polynomial coefficients). |
| `greekmat` | Item parameters in the Greek-letter parameterization |

### Examples

```
## generate FMP item parameters for 5 dichotomous items all with k = 2
set.seed(2342)
pars <- sim_bmat(n_items = 5, k = 2)
pars$bmat

## generate FMP item parameters for 5 items with varying k values and
##  varying numbers of response categories
set.seed(2432)
pars <- sim_bmat(n_items = 5, k = c(1, 2, 0, 0, 2), ncat = c(2, 3, 4, 5, 2))
pars$bmat
```

---

## sim_data *Simulate FMP Data*

---

### Description

Simulate data according to user-specified FMP item parameters and latent trait parameters.

### Usage

```
sim_data(bmat, theta, maxncat = 2, cvec = NULL, dvec = NULL)
```

### Arguments

| | |
|---|---|
| bmat | Matrix of FMP item parameters. |
| theta | Vector of latent trait values. |
| maxncat | Maximum number of response categories (the first maxncat - 1 columns of bmat are intercepts) |
| cvec | Optional vector of lower asymptote parameters. If cvec = NULL, then all lower asymptotes set to 0. |
| dvec | Optional vector of upper asymptote parameters. If dvec = NULL, then all upper asymptotes set to 1. |

### Value

Matrix of randomly generated binary item responses.

### Examples

```
## generate 5-category item responses for normally distributed theta
##   and 5 items with k = 2

set.seed(2342)
bmat <- sim_bmat(n_items = 5, k = 2, ncat = 5)$bmat
```

```
theta <- rnorm(50)
dat <- sim_data(bmat = bmat, theta = theta, maxncat = 5)
```

---

th_est_ml *Latent Trait Estimation*

---

## Description

Compute latent trait estimates using either maximum likelihood (ML) or expected a posteriori (EAP) trait estimation.

## Usage

```
th_est_ml(dat, bmat, maxncat = 2, cvec = NULL, dvec = NULL, lb = -4, ub = 4)

th_est_eap(
  dat,
  bmat,
  maxncat = 2,
  cvec = NULL,
  dvec = NULL,
  int = int_mat(npts = 33)
)
```

## Arguments

| | |
|---|---|
| dat | Data matrix of binary item responses with one column for each item. Alternatively, a vector of binary item responses for one person. |
| bmat | Matrix of FMP item parameters, one row for each item. |
| maxncat | Maximum number of response categories (the first maxncat - 1 columns of bmat are intercepts) |
| cvec | Vector of lower asymptote parameters, one element for each item. |
| dvec | Vector of upper asymptote parameters, one element for each item. |
| lb | Lower bound at which to truncate ML estimates. |
| ub | Upper bound at which to truncate ML estimates. |
| int | Matrix with two columns used for numerical integration in EAP. Column 1 contains the x coordinates and Column 2 contains the densities. |

## Value

Matrix with two columns: est and either sem or psd

| | |
|---|---|
| est | Latent trait estimate |
| sem | Standard error of measurement (mle estimates) |
| psd | Posterior standard deviation (eap estimates) |

## Examples

```
set.seed(3453)
bmat <- sim_bmat(n_items = 20, k = 0)$bmat

theta <- rnorm(10)
dat <- sim_data(bmat = bmat, theta = theta)

## mle estimates
mles <- th_est_ml(dat = dat, bmat = bmat)

## eap estimates
eaps <- th_est_eap(dat = dat, bmat = bmat)

cor(mles[,1], eaps[,1])
# 0.9967317
```

---

`transform_b` *Transform FMP Item Parameters*

---

## Description

Given FMP item parameters for a single item and the polynomial coefficients defining a latent trait transformation, find the transformed FMP item parameters.

## Usage

```
transform_b(bvec, tvec, ncat = 2)

inv_transform_b(bstarvec, tvec, ncat = 2)
```

## Arguments

| | |
|---|---|
| `bvec` | Vector of item parameters on the $\theta$ metric: (b0, b1, b2, b3, ...). |
| `tvec` | Vector of theta transformation polynomial coefficients: (t0, t1, t2, t3, ...) |
| `ncat` | Number of response categories (first ncat - 1 elements of bvec and bstarvec are intercepts) |
| `bstarvec` | Vector of item parameters on the $\theta^\star$ metric: (b*0, b*1, b*2, b*3, ...) |

## Details

Equivalent item response models can be written

$$P(\theta) = b_0 + b_1\theta + b_2\theta^2 + \cdots + b_{2k+1}\theta^{2k+1}$$

and

$$P(\theta^\star) = b_0^\star + b_1^\star\theta^\star + b_2^\star\theta^{\star 2} + \cdots + b_{2k^\star+1}^\star\theta^{2k^\star+1}$$

where

$$\theta = t_0 + t_1\theta^\star + t_2\theta^{\star 2} + \cdots + t_{2k_\theta+1}\theta^{\star 2k_\theta+1}$$

When using inv_transform_b, be aware that multiple tvec/bstarvec pairings will lead to the same bvec. Users are advised not to use the inv_transform_b function unless bstarvec has first been calculated by a call to transform_b.

## Value

Vector of transformed FMP item parameters.

## Examples

```
## example parameters from Table 7 of Reise & Waller (2003)
## goal: transform IRT model to sum score metric

a <- c(0.57, 0.68, 0.76, 0.72, 0.69, 0.57, 0.53, 0.64,
       0.45, 1.01, 1.05, 0.50, 0.58, 0.58, 0.60, 0.59,
       1.03, 0.52, 0.59, 0.99, 0.95, 0.39, 0.50)
b <- c(0.87, 1.02, 0.87, 0.81, 0.75, -0.22, 0.14, 0.56,
       1.69, 0.37, 0.68, 0.56, 1.70, 1.20, 1.04, 1.69,
       0.76, 1.51, 1.89, 1.77, 0.39, 0.08, 2.02)

## convert from difficulties and discriminations to FMP parameters

b1 <- 1.702 * a
b0 <- - 1.702 * a * b
bmat <- cbind(b0, b1)

## theta transformation vector (k_theta = 3)
##  see vignette for details about how to find tvec

tvec <- c(-3.80789e+00, 2.14164e+00, -6.47773e-01, 1.17182e-01,
          -1.20807e-02, 7.02295e-04, -2.13809e-05, 2.65177e-07)

## transform bmat
bstarmat <- t(apply(bmat, 1, transform_b, tvec = tvec))

## inspect transformed parameters
signif(head(bstarmat), 2)

## plot test response function
##  should be a straight line if transformation worked

curve(rowSums(irf_fmp(x, bmat = bstarmat)), xlim = c(0, 23),
      ylim = c(0, 23), xlab = expression(paste(theta,"*")),
      ylab = "Expected Sum Score")
abline(0, 1, col = 2)
```

# Index