

Package ‘cogirt’

May 8, 2026

Type Package

Title Cognitive Testing Using Item Response Theory

Version 1.0.0

Description Psychometrically analyze latent individual differences related to tasks, interventions, or maturational/aging effects in the context of experimental or longitudinal cognitive research using methods first described by Thomas et al. (2020) <doi:10.1177/0013164420919898>.

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Imports MASS, abind, mvtnorm, parallel, coda, numDeriv, grDevices, graphics, stats

Depends R (>= 2.10)

Suggests rmarkdown, knitr

VignetteBuilder knitr

NeedsCompilation no

Author Michael Thomas [aut, cre]

Maintainer Michael Thomas <michael.l.thomas@colostate.edu>

Repository CRAN

Date/Publication 2025-02-07 10:00:09 UTC

Contents

cog_cat	2
cog_cat_sim	3
cog_irt	5
cpt	7
dich_response_model	7
dich_response_sim	9
ex1	13

ex2	14
ex3	15
ex4	16
ex5	17
flanker	18
lrt	19
nback	19
plot.cog_cat_sim	20
plot.cog_irt	20
plt	21
sopt	21
sternberg	22
summary.cog_cat_sim	22
summary.cog_irt	23

Index	24
--------------	-----------

cog_cat	<i>Administer Cognitive Tests Using Computerized Adaptive Testing</i>
---------	---

Description

This function accepts an RDA file or a list containing selected objects and returns omega estimates, the standard error of omega, and the optimal next condition to administer for single-subject computerized adaptive testing. Adaptive testing is guided by D-optimality (see Segall, 2009).

Usage

```
cog_cat(rda = NULL, obj_fun = NULL, int_par = NULL)
```

Arguments

rda	An RDA file (or list) containing y, kappa, gamma, lambda, condition, omega_mu, omega_sigma2, zeta_mu, zeta_sigma2, nu_mu, and nu_sigma2. y should be a 1 by IJ row vector. All items not administered should have NA values in y. See package documentation for definitions and dimensions of these other objects.
obj_fun	A function that calculates predictions and log-likelihood values for the selected model (character).
int_par	Intentional parameters. That is, the parameters to optimize precision (scalar).

Value

A list with elements for omega parameter estimates (omega1), standard error of the estimates (se_omega), and the next condition to administer (next_condition).

References

Segall, D. O. (2009). Principles of Multidimensional Adaptive Testing. In W. J. van der Linden & C. A. W. Glas (Eds.), *Elements of Adaptive Testing* (pp. 57-75). https://doi.org/10.1007/978-0-387-85461-8_3

Examples

```
rda = ex5
rda$y[which(!rda$condition %in% c(3))] <- NA
cog_cat(rda = rda, obj_fun = dich_response_model, int_par = 1)
```

cog_cat_sim

Perform Simulated Computerized Adaptive Testing

Description

This function performs simulated adapting testing using the D-optimality criterion (Segall, 2009) which allows the user to focus on a subset of intentional abilities (or traits).

Usage

```
cog_cat_sim(
  data = NULL,
  model = NULL,
  guessing = NULL,
  contrast_codes = NULL,
  num_conditions = NULL,
  num_contrasts = NULL,
  constraints = NULL,
  key = NULL,
  omega = NULL,
  item_disc = NULL,
  item_int = NULL,
  conditions = NULL,
  int_par = NULL,
  start_conditions = NULL,
  max_conditions = Inf,
  omit_conditions = NULL,
  min_se = -Inf,
  link = "probit",
  verbose = TRUE
)
```

Arguments

data	A matrix of item responses (K by IJ). Rows should contain dichotomous responses (1 or 0) for the items indexed by each column.
model	An IRT model name. The options are "1p" for the one-parameter model, "2p" for the two-parameter model, "3p" for the three-parameter model, or "sdt" for a signal detection-weighted model.
guessing	Either a single numeric guessing value or a matrix of item guessing parameters (IJ by 1). This argument is only used when model = '3p'.
contrast_codes	Either a matrix of contrast codes (JM by MN) or the name in quotes of a R stats contrast function (i.e., "contr.helmert", "contr.poly", "contr.sum", "contr.treatment", or "contr.SAS"). If using the R stats contrast function items in the data matrix must be arranged by condition.
num_conditions	The total number of possible conditions (required if using the R stats contrast function or when constraints = TRUE).
num_contrasts	The number of contrasts, including intercept (required if using the R stats contrast function or when constraints = TRUE).
constraints	Either a logical (TRUE or FALSE) indicating that item parameters should be constrained to be equal over the J conditions, or a 1 by I vector of items that should be constrained to be equal across conditions.
key	An item key vector where 1 indicates a target and 2 indicates a distractor (IJ). Required when model = 'sdt'.
omega	A matrix of true omega parameters if known. These are estimated using the complete data if not supplied by the user.
item_disc	A matrix of item discrimination parameters if known. These are estimated using the complete data if not supplied by the user.
item_int	A matrix of item intercept parameters if known. These are estimated using the complete data if not supplied by the user.
conditions	A list of experimental conditions that the adaptive testing algorithm will choose from. The word "conditions" here refers to a single item or a group of items that should be administered together before the next iteration of adaptive testing. For cognitive experiments, multiple conditions can be assigned the same experimental level (e.g., memory load level).
int_par	The index of the intentional parameters, i.e., the column of the experimental effects matrix (omega) that should be optimized.
start_conditions	A vector of condition(s) completed prior to the onset of adaptive testing.
max_conditions	The maximum number of conditions to administer before terminating adaptive testing. If max_conditions is specified, min_se should not be. Note that this is the number of additional conditions to administer beyond the starting conditions.
omit_conditions	A vector of conditions to be omitted from the simulation.
min_se	The minimum standard error of estimate needed to terminate adaptive testing. If min_se is specified, max_conditions should not be.
link	The name ("logit" or "probit") of the link function to be used in the model.
verbose	Logical (TRUE or FALSE) indicating whether to print progress.

Value

A list with elements with the model used (model), true omega parameters (omega), various simulation parameters, final omega estimates (omega1) and information matrices (info1_omega), ongoing estimates of omega (ongoing_omega_est) and standard error of the estimates (ongoing_se_omega), and completed conditions (completed_conditions).

References

Segall, D. O. (2009). Principles of Multidimensional Adaptive Testing. In W. J. van der Linden & C. A. W. Glas (Eds.), *Elements of Adaptive Testing* (pp. 57-75). https://doi.org/10.1007/978-0-387-85461-8_3

Examples

```
sim_res <- cog_cat_sim(data = ex3$y, model = 'sdt', guessing = NULL,
                      contrast_codes = "contr.poly", num_conditions = 10,
                      num_contrasts = 2, constraints = NULL, key = ex3$key,
                      omega = ex3$omega, item_disc = ex3$lambda,
                      item_int = ex3$nu, conditions = ex3$condition,
                      int_par = c(1, 2), start_conditions = 3,
                      max_conditions = 3, link = "probit")

summary(sim_res)
plot(sim_res)
```

cog_irt

Fit Item Response Theory Models with Optional Contrast Effects

Description

This function estimates item response theory (IRT) model parameters. Users can optionally estimate person parameters that account for experimental or longitudinal contrast effects.

Usage

```
cog_irt(
  data = NULL,
  model = NULL,
  guessing = NULL,
  contrast_codes = NULL,
  num_conditions = NULL,
  num_contrasts = NULL,
  constraints = NULL,
  key = NULL,
  link = "probit",
  verbose = TRUE,
  ...
)
```

Arguments

data	A matrix of item responses (K by IJ). Rows should contain each subject's dichotomous responses (1 or 0) for the items indexed by each column.
model	An IRT model name. The options are "1p" for the one-parameter model, "2p" for the two parameter model, "3p" for the three-parameter model, or "sdt" for the signal detection-weighted model.
guessing	Either a single numeric guessing value or a matrix of item guessing parameters (IJ by 1). This argument is only used when model = '3p'.
contrast_codes	Either a matrix of contrast codes (JM by MN) or the name in quotes of a R stats contrast function (i.e., "contr.helmert", "contr.poly", "contr.sum", "contr.treatment", or "contr.SAS"). If using the R stats contrast function items in the data matrix must be arranged by condition.
num_conditions	The number of conditions (required if using the R stats contrast function or when constraints = TRUE).
num_contrasts	The number of contrasts including intercept (required if using the R stats contrast function or when constraints = TRUE).
constraints	Either a logical (TRUE or FALSE) indicating that item parameters should be constrained to be equal over the J conditions or a 1 by I vector of items that should be constrained to be equal across conditions.
key	An item key vector where 1 indicates target and 2 indicates distractor (IJ). Required when model = 'sdt'.
link	The name ("logit" or "probit") of the link function to be used in the model.
verbose	Logical (TRUE or FALSE) indicating whether to print progress.
...	Additional arguments.

Value

A list with elements for all parameters estimated (omega1, nu1, and/or lambda1), information values for all parameters estimated (info1_omega, info1_nu, and/or info1_lambda), the model log-likelihood value (log_lik), and the total number of estimated parameters (par) in the model.

Dimensions

I = Number of items per condition; J = Number of conditions or time points; K = Number of examinees; M Number of ability (or trait) dimensions; N Number of contrast effects (including intercept).

References

- Embretson S. E., & Reise S. P. (2000). *Item response theory for psychologists*. Mahwah, N.J.: L. Erlbaum Associates.
- Thomas, M. L., Brown, G. G., Patt, V. M., & Duffy, J. R. (2021). Latent variable modeling and adaptive testing for experimental cognitive psychopathology research. *Educational and Psychological Measurement*, 81(1), 155-181.

Examples

```
nback_fit_contr <- cog_irt(data = nback$y, model = "sdt",
                           contrast_codes = "contr.poly", key = nback$key,
                           num_conditions = length(unique(nback$condition)),
                           num_contrasts = 2)
plot(nback_fit_contr)
```

 cpt

CPT Data

Description

CPT task accuracy data collected from an online experiment. The condition vector indicates backward mask onset (50, 100, 150, or 200 ms). The key indicates whether items are targets (1) or distractors (2).

Usage

```
cpt
```

Format

A list with the following elements:

y Matrix of dichotomous responses.

key Item key vector where 1 indicates target and 2 indicates distractor (IJ)

condition Condition vector indicating distinct conditions or time points.

 dich_response_model

Dichotomous Response Model

Description

This function calculates predictions and log-likelihood values for a dichotomous response model framed using generalized latent variable modeling (GLVM; Skrondal & Rabe-Hesketh, 2004).

Usage

```
dich_response_model(
  y = NULL,
  omega = NULL,
  gamma = NULL,
  lambda = NULL,
  zeta = NULL,
  nu = NULL,
  kappa = NULL,
  link = NULL
)
```

Arguments

y	Item response matrix (K by IJ).
omega	Contrast effects matrix (K by MN).
gamma	Contrast codes matrix (JM by MN).
lambda	Item slope matrix (IJ by JM).
zeta	Specific effects matrix (K by JM).
nu	Item intercept matrix (IJ by 1).
kappa	Item guessing matrix (IJ by 1).
link	Choose between "logit" or "probit" link functions.

Value

p = response probability matrix (K by IJ); yhatstar = latent response variate matrix (K by IJ); log-likelihood = model log-likelihood (scalar).

Dimensions

I = Number of items per condition; J = Number of conditions; K = Number of examinees; M Number of ability (or trait) dimensions; N Number of contrasts (should include intercept).

References

Skrondal, A., & Rabe-Hesketh, S. (2004). *Generalized latent variable modeling: Multilevel, longitudinal, and structural equation models*. Boca Raton: Chapman & Hall/CRC.

dich_response_sim *Simulate Dichotomous Response Model*

Description

This function calculates the matrix of first partial derivatives, the matrix of second partial derivatives, and the information matrix for the posterior distribution with respect to theta (ability) based on the slope-intercept form of the item response theory model.

Usage

```
dich_response_sim(
  I = NULL,
  J = NULL,
  K = NULL,
  M = NULL,
  N = NULL,
  omega = NULL,
  omega_mu = NULL,
  omega_sigma2 = NULL,
  gamma = NULL,
  lambda = NULL,
  lambda_mu = NULL,
  lambda_sigma2 = NULL,
  nu = NULL,
  nu_mu = NULL,
  nu_sigma2 = NULL,
  zeta = NULL,
  zeta_mu = NULL,
  zeta_sigma2 = NULL,
  kappa = NULL,
  key = NULL,
  link = "probit"
)
```

Arguments

I	Number of items per condition.
J	Number of conditions.
K	Number of examinees
M	Number of ability (or trait) dimensions.
N	Number of contrasts (should include intercept).
omega	Contrast effects matrix (K by MN).
omega_mu	Vector of means for the examinee-level effects of the experimental manipulation (1 by MN).

omega_sigma2	Covariance matrix for the examinee-level effects of the experimental manipulation (MN by MN).
gamma	Contrast codes matrix (JM by MN).
lambda	Item slope matrix (IJ by JM).
lambda_mu	Vector of means for the item slope parameters (1 by JM)
lambda_sigma2	Covariance matrix for the item slope parameters (JM by JM)
nu	Item intercept matrix (K by IJ).
nu_mu	Mean of the item intercept parameters (scalar).
nu_sigma2	Variance of the item intercept parameters (scalar).
zeta	Specific effects matrix (K by JM).
zeta_mu	Vector of means for the condition-level effects nested within examinees (1 by JM).
zeta_sigma2	Covariance matrix for the condition-level effects nested within examinees (JM by JM).
kappa	kappa Item guessing matrix (IJ by 1). If kappa is not provided, parameter values are set to 0.
key	Option key where 1 indicates target and 2 indicates distractor.
link	Choose between logit or probit link functions.

Value

y = simulated response matrix; yhatstar = simulated latent response probability matrix; [simulation_parameters]

References

Skrondal, A., & Rabe-Hesketh, S. (2004). *Generalized latent variable modeling: Multilevel, longitudinal, and structural equation models*. Boca Raton: Chapman & Hall/CRC.

Thomas, M. L., Brown, G. G., Gur, R. C., Moore, T. M., Patt, V. M., Risbrough, V. B., & Baker, D. G. (2018). A signal detection-item response theory model for evaluating neuropsychological measures. *Journal of Clinical and Experimental Neuropsychology*, 40(8), 745-760.

Examples

```
# Example 1

I <- 100
J <- 1
K <- 250
M <- 1
N <- 1

omega_mu <- matrix(data = 0, nrow = 1, ncol = M * N)
omega_sigma2 <- diag(x = 1, nrow = M * N)
gamma <- diag(x = 1, nrow = J * M, ncol = M * N)
lambda_mu <- matrix(data = 1, nrow = 1, ncol = M)
lambda_sigma2 <- diag(x = 0.25, nrow = M)
```

```

zeta_mu <- matrix(data = rep(x = 0, times = M * J), nrow = 1, ncol = J * M)
zeta_sigma2 <- diag(x = 0, nrow = J * M, ncol = J * M)
nu_mu <- matrix(data = 0, nrow = 1, ncol = 1)
nu_sigma2 <- matrix(data = 1, nrow = 1, ncol = 1)
set.seed(624)
ex1 <- dich_response_sim(I = I, J = J, K = K, M = M, N = N,
  omega_mu = omega_mu, omega_sigma2 = omega_sigma2,
  gamma = gamma, lambda_mu = lambda_mu,
  lambda_sigma2 = lambda_sigma2, nu_mu = nu_mu,
  nu_sigma2 = nu_sigma2, zeta_mu = zeta_mu,
  zeta_sigma2 = zeta_sigma2)

```

Example 2

```

I <- 100
J <- 1
K <- 50
M <- 2
N <- 1
omega_mu <- matrix(data = c(3.50, 1.00), nrow = 1, ncol = M * N)
omega_sigma2 <- diag(x = c(0.90, 0.30), nrow = M * N)
gamma <- diag(x = 1, nrow = J * M, ncol = M * N)
key <- rbinom(n = I * J, size = 1, prob = .7) + 1
measure_weights <-
  matrix(data = c(0.5, -1.0, 0.5, 1.0), nrow = 2, ncol = M, byrow = TRUE)
lambda <- matrix(data = 0, nrow = I * J, ncol = J * M)
for(j in 1:J) {
  lambda[(1 + (j - 1) * I):(j * I), (1 + (j - 1) * M):(j * M)] <-
    measure_weights[key, ][(1 + (j - 1) * I):(j * I), ]
}
zeta_mu <- matrix(data = rep(x = 0, times = M * J), nrow = 1, ncol = J * M)
zeta_sigma2 <- diag(x = 0, nrow = J * M, ncol = J * M)
nu_mu <- matrix(data = 0, nrow = 1, ncol = 1)
nu_sigma2 <- matrix(data = .2, nrow = 1, ncol = 1)
set.seed(624)
ex2 <- dich_response_sim(I = I, J = J, K = K, M = M, N = N,
  omega_mu = omega_mu, omega_sigma2 = omega_sigma2,
  gamma = gamma, lambda = lambda, nu_mu = nu_mu,
  nu_sigma2 = nu_sigma2, zeta_mu = zeta_mu,
  zeta_sigma2 = zeta_sigma2, key = key)

```

Example 3

```

I <- 20
J <- 10
K <- 50
M <- 2
N <- 2
omega_mu <- matrix(data = c(2.50, -2.00, 0.50, 0.00), nrow = 1, ncol = M * N)
omega_sigma2 <- diag(x = c(0.90, 0.70, 0.30, 0.10), nrow = M * N)
contrast_codes <- cbind(1, contr.poly(n = J))[, 1:N]
gamma <- matrix(data = 0, nrow = J * M, ncol = M * N)
for(j in 1:J) {

```

```

    for(m in 1:M) {
      gamma[(m + M * (j - 1)), (((m - 1) * N + 1):((m - 1) * N + N))] <-
        contrast_codes[j, ]
    }
  }
  key <- rbinom(n = I * J, size = 1, prob = .7) + 1
  measure_weights <-
    matrix(data = c(0.5, -1.0, 0.5, 1.0), nrow = 2, ncol = M, byrow = TRUE)
  lambda <- matrix(data = 0, nrow = I * J, ncol = J * M)
  for(j in 1:J) {
    lambda[(1 + (j - 1) * I):(j * I), (1 + (j - 1) * M):(j * M)] <-
      measure_weights[key, ][(1 + (j - 1) * I):(j * I), ]
  }
  zeta_mu <- matrix(data = rep(x = 0, times = M * J), nrow = 1, ncol = J * M)
  zeta_sigma2 <- diag(x = 0.2, nrow = J * M, ncol = J * M)
  nu_mu <- matrix(data = c(0.00), nrow = 1, ncol = 1)
  nu_sigma2 <- matrix(data = c(0.20), nrow = 1, ncol = 1)
  set.seed(624)
  ex3 <- dich_response_sim(I = I, J = J, K = K, M = M, N = N,
    omega_mu = omega_mu, omega_sigma2 = omega_sigma2,
    gamma = gamma, lambda = lambda, nu_mu = nu_mu,
    nu_sigma2 = nu_sigma2, zeta_mu = zeta_mu,
    zeta_sigma2 = zeta_sigma2, key = key)

# Example 4

I <- 25
J <- 2
K <- 200
M <- 1
N <- 2
omega_mu <- matrix(data = c(1, -2), nrow = 1, ncol = M * N)
omega_sigma2 <- diag(x = c(1.00, 0.25), nrow = M * N)
contrast_codes <- cbind(1, contr.treatment(n = J))[, 1:N]
gamma <- matrix(data = 0, nrow = J * M, ncol = M * N)
for(j in 1:J) {
  for(m in 1:M) {
    gamma[(m + M * (j - 1)), (((m - 1) * N + 1):((m - 1) * N + N))] <-
      contrast_codes[j, ]
  }
}
lambda <- matrix(data = 0, nrow = I * J, ncol = J * M)
lam_vals <- rnorm(I, 1.5, .23)
for (j in 1:J) {
  lambda[(1 + (j - 1) * I):(j * I), (1 + (j - 1) * M):(j * M)] <- lam_vals
}
zeta_mu <- matrix(data = rep(x = 0, times = M * J), nrow = 1, ncol = J * M)
zeta_sigma2 <- diag(x = 0.2, nrow = J * M, ncol = J * M)
nu <- matrix(data = rnorm(n = I, mean = 0, sd = 2), nrow = I * J, ncol = 1)
set.seed(624)
ex4 <- dich_response_sim(I = I, J = J, K = K, M = M, N = N,
  omega_mu = omega_mu, omega_sigma2 = omega_sigma2,
  gamma = gamma, lambda = lambda, nu = nu,

```

```

zeta_mu = zeta_mu, zeta_sigma2 = zeta_sigma2)

# Example 5

I <- 20
J <- 10
K <- 1
M <- 2
N <- 2
omega_mu <- matrix(data = c(2.50, -2.00, 0.50, 0.00), nrow = 1, ncol = M * N)
omega_sigma2 <- diag(x = c(0.90, 0.70, 0.30, 0.10), nrow = M * N)
contrast_codes <- cbind(1, contr.poly(n = J))[, 1:N]
gamma <- matrix(data = 0, nrow = J * M, ncol = M * N)
for(j in 1:J) {
  for(m in 1:M) {
    gamma[(m + M * (j - 1)), ((m - 1) * N + 1):((m - 1) * N + N)] <-
      contrast_codes[j, ]
  }
}
key <- rbinom(n = I * J, size = 1, prob = .7) + 1
measure_weights <-
  matrix(data = c(0.5, -1.0, 0.5, 1.0), nrow = 2, ncol = M, byrow = TRUE)
lambda <- matrix(data = 0, nrow = I * J, ncol = J * M)
for(j in 1:J) {
  lambda[(1 + (j - 1) * I):(j * I), (1 + (j - 1) * M):(j * M)] <-
    measure_weights[key, ][(1 + (j - 1) * I):(j * I), ]
}
zeta_mu <- matrix(data = rep(x = 0, times = M * J), nrow = 1, ncol = J * M)
zeta_sigma2 <- diag(x = 0.2, nrow = J * M, ncol = J * M)
nu_mu <- matrix(data = c(0.00), nrow = 1, ncol = 1)
nu_sigma2 <- matrix(data = c(0.20), nrow = 1, ncol = 1)
set.seed(624)
ex5 <- dich_response_sim(I = I, J = J, K = K, M = M, N = N,
  omega_mu = omega_mu, omega_sigma2 = omega_sigma2,
  gamma = gamma, lambda = lambda, nu_mu = nu_mu,
  nu_sigma2 = nu_sigma2, zeta_mu = zeta_mu,
  zeta_sigma2 = zeta_sigma2, key = key)

```

ex1

Simulated Data for a Unidimensional Two-Parameter Item Response Model

Description

Data and parameters were simulated based on example 1 provided for the `sim_dich_response.R` function.

Usage

ex1

Format

A list with the following elements:

y Matrix of dichotomous responses.

ystar Matrix of latent response variates.

omega Subject-level effects of the experimental manipulation.

omega_mu Vector of means for the subject-level effects of the experimental manipulation (1 by K * M).

omega_sigma2 Covariance matrix for the subject-level effects of the experimental manipulation (K * M by K * M).

gamma Contrast codes matrix.

lambda Matrix of item slope parameters.

lambda_mu Vector of means for the item slope parameters (1 by JM).

lambda_sigma2 Covariance matrix for the item slope parameters (JM by JM).

nu Mean of the item intercept parameters (scalar).

nu_mu Mean of the item intercept parameters (scalar).

nu_sigma2 Variance of the item intercept parameters (scalar).

zeta Condition-level prediction errors.

zeta_mu Vector of means for the condition-level prediction errors (1 by J * M).

zeta_sigma2 Covariance matrix for the condition-level prediction errors (J * M by J * M).

kappa Item guessing matrix (K by IJ).

condition Condition vector indicating distinct conditions or time points.

key Item key vector where 1 indicates target and 2 indicates distractor (IJ) ...

 ex2

Simulated Data for a Signal Detection Weighted IRT Model

Description

Data and parameters were simulated based on example 2 provided for the `sim_dich_response.R` function.

Usage

ex2

Format

A list with the following elements:

y Matrix of dichotomous responses.

ystar Matrix of latent response variates.

omega Subject-level effects of the experimental manipulation.

omega_mu Vector of means for the subject-level effects of the experimental manipulation (1 by K * M).

omega_sigma2 Covariance matrix for the subject-level effects of the experimental manipulation (K * M by K * M).

gamma Contrast codes matrix.

lambda Matrix of item slope parameters.

lambda_mu Vector of means for the item slope parameters (1 by JM).

lambda_sigma2 Covariance matrix for the item slope parameters (JM by JM).

nu Mean of the item intercept parameters (scalar).

nu_mu Mean of the item intercept parameters (scalar).

nu_sigma2 Variance of the item intercept parameters (scalar).

zeta Condition-level prediction errors.

zeta_mu Vector of means for the condition-level prediction errors (1 by J * M).

zeta_sigma2 Covariance matrix for the condition-level prediction errors (J * M by J * M).

kappa Item guessing matrix (K by IJ).

condition Condition vector indiciting distinct conditions or time points.

key Item key vector where 1 indicates target and 2 indicates distractor (IJ) ...

 ex3

Simulated Data for a Signal Detection Weighted IRT Model with an Experimental Design

Description

Data and parameters were simulated based on example 3 provided for the `sim_dich_response.R` function.

Usage

ex3

Format

A list with the following elements:

y Matrix of dichotomous responses.

ystar Matrix of latent response variates.

omega Subject-level effects of the experimental manipulation.

omega_mu Vector of means for the subject-level effects of the experimental manipulation (1 by K * M).

omega_sigma2 Covariance matrix for the subject-level effects of the experimental manipulation (K * M by K * M).

gamma Contrast codes matrix.

lambda Matrix of item slope parameters.

lambda_mu Vector of means for the item slope parameters (1 by JM).

lambda_sigma2 Covariance matrix for the item slope parameters (JM by JM).

nu Mean of the item intercept parameters (scalar).

nu_mu Mean of the item intercept parameters (scalar).

nu_sigma2 Variance of the item intercept parameters (scalar).

zeta Condition-level prediction errors.

zeta_mu Vector of means for the condition-level prediction errors (1 by J * M).

zeta_sigma2 Covariance matrix for the condition-level prediction errors (J * M by J * M).

kappa Item guessing matrix (K by IJ).

condition Condition vector indicating distinct conditions or time points.

key Item key vector where 1 indicates target and 2 indicates distractor (IJ) ...

ex4

Simulated Data for a Unidimensional Two-Parameter Item Response Model with Two Measurement Occasions

Description

Data and parameters were simulated based on example 4 provided for the `sim_dich_response.R` function.

Usage

ex4

Format

A list with the following elements:

y Matrix of dichotomous responses.

ystar Matrix of latent response variates.

omega Subject-level effects of the experimental manipulation.

omega_mu Vector of means for the subject-level effects of the experimental manipulation (1 by K * M).

omega_sigma2 Covariance matrix for the subject-level effects of the experimental manipulation (K * M by K * M).

gamma Contrast codes matrix.

lambda Matrix of item slope parameters.

lambda_mu Vector of means for the item slope parameters (1 by JM).

lambda_sigma2 Covariance matrix for the item slope parameters (JM by JM).

nu Mean of the item intercept parameters (scalar).

nu_mu Mean of the item intercept parameters (scalar).

nu_sigma2 Variance of the item intercept parameters (scalar).

zeta Condition-level prediction errors.

zeta_mu Vector of means for the condition-level prediction errors (1 by J * M).

zeta_sigma2 Covariance matrix for the condition-level prediction errors (J * M by J * M).

kappa Item guessing matrix (K by IJ).

condition Condition vector indicating distinct conditions or time points.

key Item key vector where 1 indicates target and 2 indicates distractor (IJ) ...

 ex5

Simulated Single Subject Data for a Signal Detection Weighted IRT Model with an Experimental Design

Description

Data and parameters were simulated based on example 5 provided for the `sim_dich_response.R` function.

Usage

ex5

Format

A list with the following elements:

y Matrix of dichotomous responses.

ystar Matrix of latent response variates.

omega Subject-level effects of the experimental manipulation.

omega_mu Vector of means for the subject-level effects of the experimental manipulation (1 by K * M).

omega_sigma2 Covariance matrix for the subject-level effects of the experimental manipulation (K * M by K * M).

gamma Contrast codes matrix.

lambda Matrix of item slope parameters.

lambda_mu Vector of means for the item slope parameters (1 by JM).

lambda_sigma2 Covariance matrix for the item slope parameters (JM by JM).

nu Mean of the item intercept parameters (scalar).

nu_mu Mean of the item intercept parameters (scalar).

nu_sigma2 Variance of the item intercept parameters (scalar).

zeta Condition-level prediction errors.

zeta_mu Vector of means for the condition-level prediction errors (1 by J * M).

zeta_sigma2 Covariance matrix for the condition-level prediction errors (J * M by J * M).

kappa Item guessing matrix (K by IJ).

condition Condition vector indicating distinct conditions or time points.

key Item key vector where 1 indicates target and 2 indicates distractor (IJ) ...

flanker

Flanker Data

Description

Flanker task accuracy data collected from an online experiment. The condition vector indicates level of congruency ("congruent, incongruent_part, incongruent_all, neutral).

Usage

flanker

Format

A list with the following elements:

y Matrix of dichotomous responses.

condition Condition vector indicating distinct conditions or time points.

lrt *Method of anova for cogirt S3*

Description

This function compares fit of models produced by cogirt.

Usage

```
lrt(object, ...)
```

Arguments

object	An object of class 'cogirt'.
...	Additional arguments.

Value

An object of class "anova".

nback *N-Back Data*

Description

N-Back task accuracy data collected from an online experiment. The condition vector indicates working memory load level (1-back, 2-back, 3-back, or 4-back). The key indicates whether items are targets (1) or distractors (2).

Usage

```
nback
```

Format

A list with the following elements:

y Matrix of dichotomous responses.

key Item key vector where 1 indicates target and 2 indicates distractor (IJ)

condition Condition vector indiciting distinct conditions or time points.

plot.cog_cat_sim *Method of Plot for Simulated Adaptive Testing Using cogirt S3*

Description

This function produces plots for standard errors for cog_cat_sim results

Usage

```
## S3 method for class 'cog_cat_sim'  
plot(x, ...)
```

Arguments

x An object of class 'cog_cat_sim'.
... Additional arguments.

Value

This function returns a base R plot displayed in the graphics device. It does not return any value to the R environment.

plot.cog_irt *Method of Plot for cogirt S3*

Description

This function produces plots for parameter estimates produced for various cogirt models.

Usage

```
## S3 method for class 'cog_irt'  
plot(x, ...)
```

Arguments

x An x of class 'cog_irt'.
... Additional arguments.

Value

This function returns a base R plot displayed in the graphics device. It does not return any value to the R environment.

plt

PLT Data

Description

Probabilistic Learning Task (SOPT) accuracy data collected from an online experiment. The condition vector indicates feedback consistency (90 70 vector indicates which side was rewarded).

Usage

plt

Format

A list with the following elements:

y Matrix of dichotomous responses.

targ Item targ left vs. right vector (IJ)

fdbk Item fdbk left vs. right vector (IJ)

condition Condition vector indiciting distinct conditions or time points.

sopt

SOPT Data

Description

Self-Ordered Pointing Task (SOPT) accuracy data collected from an online experiment. The condition vector indicates working memory load level (3, 6, 9, or 12 items).

Usage

sopt

Format

A list with the following elements:

y Matrix of dichotomous responses.

condition Condition vector indiciting distinct conditions or time points.

 sternberg

Sternberg Data

Description

Sternberg task accuracy data collected from an online experiment. The condition vector indicates working memory load level (2, 4, 6, 8, 10, or 12 items). The key indicates whether items are targets (1) or distractors (2).

Usage

```
sternberg
```

Format

A list with the following elements:

y Matrix of dichotomous responses.

key Item key vector where 1 indicates target and 2 indicates distractor (IJ)

condition Condition vector indicating distinct conditions or time points.

 summary.cog_cat_sim

Method of Summary for cog_cat_sim S3

Description

This function provides summary statistics for simulated computerized adaptive testing.

Usage

```
## S3 method for class 'cog_cat_sim'
summary(object, ...)
```

Arguments

object An object of class 'cog_cat_sim'.

... Additional arguments.

Value

This function does not return a value to the R environment. Instead, it prints a comprehensive summary of the simulated computerized adaptive testing results to the console. The output includes model name and simulation settings as well as summary statistics for each parameter of interest. The function is intended for interactive use.

summary.cog_irt	<i>Method of Summary for cog_irt S3</i>
-----------------	---

Description

This function provides summary statistics for cogirt models.

Usage

```
## S3 method for class 'cog_irt'  
summary(object, ...)
```

Arguments

object	An object of class 'cog_irt'.
...	Additional arguments.

Value

This function does not return a value to the R environment. Instead, it prints a detailed summary of the specified IRT model to the console. The output includes the type of IRT model (e.g., One-Parameter, Two-Parameter, etc.), the number of subjects and items in the dataset, the log-likelihood of the model, and summary statistics (mean, standard deviation, median standard error, and reliability) for estimated parameters. The function is intended for interactive use to review the results of the fitted model.

Index

* datasets

- cpt, [7](#)
- ex1, [13](#)
- ex2, [14](#)
- ex3, [15](#)
- ex4, [16](#)
- ex5, [17](#)
- flanker, [18](#)
- nback, [19](#)
- plt, [21](#)
- sopt, [21](#)
- sternberg, [22](#)

- cog_cat, [2](#)
- cog_cat_sim, [3](#)
- cog_irt, [5](#)
- cpt, [7](#)

- dich_response_model, [7](#)
- dich_response_sim, [9](#)

- ex1, [13](#)
- ex2, [14](#)
- ex3, [15](#)
- ex4, [16](#)
- ex5, [17](#)

- flanker, [18](#)

- lrt, [19](#)

- nback, [19](#)

- plot.cog_cat_sim, [20](#)
- plot.cog_irt, [20](#)
- plt, [21](#)

- sopt, [21](#)
- sternberg, [22](#)
- summary.cog_cat_sim, [22](#)
- summary.cog_irt, [23](#)