

# Package ‘catSurv’

March 27, 2017

**Title** Computerized Adaptive Testing

**Version** 0.0.0.9000

**Description** Provides methods of computerized adaptive testing for survey researchers. Includes functionality for data fit with the latent trait, Birnbaum’s three parameter, the graded response, or the generalized partial credit model. Additionally, includes several ability parameter estimation and item selection routines. During item selection, all calculations are done in compiled C++ code.

**Depends** R(>= 3.3.2)

**License** GPL-3

**Imports** ltm(>= 1.0.0),  
methods(>= 3.3.2),  
stats(>= 3.3.2),  
Rcpp(>= 0.12.9),

**LinkingTo** BH(>= 1.60.0.2),  
Rcpp(>= 0.12.9),  
RcppArmadillo,  
RcppGSL(>= 0.3.1)

**LazyData** true

**RoxygenNote** 5.0.1

**SystemRequirements** C++11

**Suggests** catIrt(>= 0.5.0),  
catR(>= 3.10),  
testthat(>= 1.0.2)

## R topics documented:

Cat-class . . . . .	2
checkStopRules . . . . .	4
d2LL . . . . .	5
dLL . . . . .	7
estimateSE . . . . .	8
estimateTheta . . . . .	10
expectedKL . . . . .	11
expectedObsInf . . . . .	12
expectedPV . . . . .	13
fisherInf . . . . .	14

fisherTestInfo . . . . .	16
getters . . . . .	17
gpcmCat . . . . .	19
grmCat . . . . .	20
likelihood . . . . .	22
likelihoodKL . . . . .	23
lookAhead . . . . .	24
ltmCat . . . . .	25
makeTree . . . . .	27
nfc . . . . .	28
npi . . . . .	29
obsInf . . . . .	31
polknowMT . . . . .	32
polknowTAPS . . . . .	36
posteriorKL . . . . .	37
prior . . . . .	39
probability . . . . .	40
selectItem . . . . .	42
setters . . . . .	44
storeAnswer . . . . .	46
tpmCat . . . . .	47

<b>Index</b>	<b>50</b>
--------------	-----------

---

Cat-class

*Computerized Adaptive Testing Survey (catSurv) Object*

---

## Description

Creates an object of class Cat. Cat objects are used in administering Computerized Adaptive Testing (CAT) Surveys. These objects contain several pieces of information relevant for CAT surveys, and are used as input in the main functions of the catSurv package.

## Details

Assume we have a survey battery with I questions. An object of the class Cat has the following slots:

- **guessing** A vector of length I of guessing parameters. Guessing parameters are only applicable for Cat objects fit with the "tpm" model, using the tpmCat function.
- **discrimination** A vector of length I of discrimination parameters.
- **difficulty** A vector or list of length I of difficulty parameters. For Cat objects of the "ltm" or "tpm" model, difficulty is a vector that contains a parameter for each item. For Cat objects of the "grm" or "gpcm" model, difficulty is a list that contains a vector for each item, and each vector contains a parameter for each response option.
- **answers** A vector of length I of answers to questions as given by the survey respondent. Unanswered questions have the value NA.
- **priorName** A character vector of length one giving the prior distribution to use for the ability parameter estimates. The options are "NORMAL" for the normal distribution, "STUDENT\_T" for the student's t distribution, and "UNIFORM" for the uniform distribution. The default value is "NORMAL".

- **priorParams** A numeric vector of length two of parameters for the distribution specified in the **priorName** slot. When **priorName** is set to "NORMAL", the first element of **priorParams** is the mean, and the second element is the standard deviation. When **priorName** is set to "STUDENT\_T", the first element of **priorParams** is  $\mu$ , a location parameter, and the second is degrees of freedom. When **priorName** is set to "UNIFORM", the elements of **priorParams** are lower and upper bounds, respectively. Note that the uniform distribution is only applicable for the "EAP" estimation method. The default values are 0, 1.
- **lowerBound** A numeric indicating the lower bound of the interval of the latent scale used in estimation. The default value is  $-5$ .
- **upperBound** A numeric indicating the upper bound of the interval of the latent scale used in estimation. The default value is 5.
- **model** A string indicating the model fit to the data. The options are "ltm" for the latent trait model, "tpm" for Birnbaum's three parameter model, "grm" for the graded response model, and "gpcm" for the generalized partial credit model.
- **estimation** A string indicating the approach to estimating ability parameters. The options are "EAP" for the expected a posteriori approach, "MAP" for the modal a posteriori approach, "MLE" for the maximum likelihood approach, and "WLE" for the weighted maximum likelihood approach. The default value is "EAP".
- **estimationDefault** A string indicating the approach to estimating ability parameters when the primary estimation choice indicated in the **estimation** slot is "MLE" or "WLE" and this estimation fails to converge. The options are "EAP" and "MAP". The default value is "MAP".
- **selection** A string indicating the approach for selecting the next item. The options are "EPV" for minimum expected posterior variance, "MEI" for maximum expected information, "MFI" for maximum Fisher information, "MPWI" for maximum posterior weighted information, "MLWI" for maximum likelihood weighted information, "KL" for the maximum expected Kullback-Leibler (KL) information, "LKL" maximum likelihood weighted KL information, "PKL" maximum posterior weighted KL information, "MFII" for ?????, and "RANDOM" where the next item is chosen randomly. The default value is "EPV".
- **z** A numeric used in calculating  $\delta$ .  $\delta$  is used in determining the bounds of integration for some **selectItem** methods. Default value is 0.9.
- **lengthThreshold** A numeric. The number of questions answered must be greater than or equal to this threshold to stop administering items. The default value is NA.
- **seThreshold** A numeric. The standard error estimate of the latent trait must be less than this threshold to stop administering items. The default value is NA.
- **infoThreshold** A numeric. The Fisher's information for all remaining items must be less than this threshold to stop administering items. The default value is NA.
- **gainThreshold** A numeric. The absolute value of the difference between the standard error of the latent trait estimate and the square root of the expected posterior variance for each item must be less than this threshold to stop administering items. The default value is NA.
- **lengthOverride** A numeric. The number of questions answered must be less than this override to continue administering items. The default value is NA.
- **gainOverride** A numeric. The absolute value of the difference between the standard error of the latent trait estimate and the square root of the expected posterior variance for each item must be less than this override to continue administering items. The default value is NA.

#### Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

**See Also**

[checkStopRules](#), [estimateTheta](#), [gpcmCat](#), [grmCat](#), [ltmCat](#), [selectItem](#), [tpmCat](#)

---

checkStopRules

*Check if Stop and/or Override Rules are Met*

---

**Description**

Evaluates the specified stopping and/or override rules to check if respondent should be asked further questions.

**Usage**

```
checkStopRules(catObj)
```

**Arguments**

catObj                      An object of class Cat

**Details**

The stopping rule thresholds are stored in the following Cat object slots: lengthThreshold, seThreshold, infoThreshold, and gainThreshold. The override thresholds are stored in the following Cat object slots: lengthOverride, gainOverride. A value of NA indicates the rule will not be used in evaluating if further questions should be administered. A user can specify any combination of stopping rules and/or overrides.

**Stopping Rules:**

lengthThreshold: Number of question's answered  $\geq a$

seThreshold:  $SE(\hat{\theta}) < a$

infoThreshold:  $FI < a \forall$  remaining items

gainThreshold:  $SE(\hat{\theta}) - \sqrt{EPV} < a \forall$  remaining items

**Overrides:**

lengthOverride: Number of question's answered  $< a$

gainOverride:  $|SE(\hat{\theta}) - \sqrt{EPV}| \geq a \forall$  remaining items

**Value**

The function checkStopRules returns a boolean. TRUE indicates all specified stopping rules are met and no specified overrides are met. No further items should be administered. FALSE indicates at least one specified stopping rule is not met, or if any specified override threshold is met. Additional items should be administered.

**Note**

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

**Author(s)**

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

**References**

Babcock, Ben, and David J. Weiss. 2009. "Termination Criteria in Computerized Adaptive Tests: Variable-Length CATs are not Biased." Proceedings of the 2009 GMAC Conference on Computerized Adaptive Testing. Vol. 14.

**See Also**

[Cat-class](#), [estimateSE](#), [expectedPV](#), [fisherInf](#)

**Examples**

```
## Not run:
## Create Cat object
data(npi)
ltm_cat <- ltmCat(npi)

## Store example answers
setAnswers(ltm_cat) <- c(1,0,1,0,1,0,0,0,1,1, rep(NA, 30))

## Stop administering items if standard error of ability
## estimate is low enough
setSeThreshold(ltm_cat) <- .5
checkStopRules(ltm_cat)

## Now stop if standard error is low enough, but only if respondent has
## answered 11 questions
setLengthOverride(ltm_cat) <- 11
checkStopRules(ltm_cat)

## When respondent has answered 11 questions and standard error
## of ability estimate is below .5, stop administering items
setAnswers(ltm_cat) <- c(1,0,1,0,1,0,0,0,1,1,0, rep(NA, 29))
checkStopRules(ltm_cat)

## End(Not run)
```

---

d2LL

---

*The Second Derivative of the Log-Likelihood*


---

**Description**

Calculates either the second derivative of the log-likelihood or the second derivative of the log-posterior evaluated at point  $\theta$ .

**Usage**

```
d2LL(catObj, theta, use_prior)
```

**Arguments**

<code>catObj</code>	An object of class <code>Cat</code>
<code>theta</code>	A numeric or an integer indicating the value for $\theta$
<code>use_prior</code>	A logical indicating whether to calculate based on the log-likelihood or log-posterior

**Details**

When the `usePrior` argument is `FALSE`, the function `d2LL` evaluates the second derivative of the log-likelihood at point  $\theta$ .

When the `usePrior` argument is `TRUE`, the function `d2LL` evaluates the second derivative of the log-posterior at point  $\theta$ .

The function `dLL2` is only available when using the normal prior distribution when the argument `use_prior` is `TRUE`.

**Value**

The function `d2LL` returns a numeric of the second derivative of the log-likelihood (or log-posterior) given a respondent's answer profile.

**Note**

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

**Author(s)**

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

**See Also**

[Cat-class](#), [dLL](#), [prior](#)

**Examples**

```
## Not run:
## Create Cat object, store example answers, and calculate
## second derivative of log-likelihood at theta = 1

## d2LL for Cat object of the ltm model
data(npi)
ltm_cat <- ltmCat(npi)
setAnswers(ltm_cat) <- c(1,0,1,0,1, rep(NA, 35))
d2LL(ltm_cat, theta = 1)

## d2LL for Cat object of the tpm model
data(polknow)
tpm_cat <- tpmCat(polknow)
setAnswers(tpm_cat) <- c(1,0,1,0, rep(NA, 35))
d2LL(tpm_cat, theta = 1)

## d2LL for Cat object of the grm model
```

```

data(nfc)
grm_cat <- grmCat(nfc)
setAnswers(grm_cat) <- c(1,3,4,5, rep(NA, 13))
d2LL(grm_cat, theta = 1)

## End(Not run)

```

dLL

*The First Derivative of the Log-Likelihood***Description**

Calculates either the first derivative of the log-likelihood or the first derivative of the log-posterior evaluated at point  $\theta$ .

**Usage**

```
dLL(catObj, theta, use_prior)
```

**Arguments**

catObj	An object of class Cat
theta	A numeric or an integer indicating the value for $\theta_j$
use_prior	A logical indicating whether to calculate based on the log-likelihood or log-posterior

**Details**

When the usePrior argument is FALSE, the function dLL evaluates the first derivative of the log-likelihood at point  $\theta$ .

When the usePrior argument is TRUE, the function dLL evaluates the first derivative of the log-posterior at point  $\theta$ .

The function dLL is only available when using the normal prior distribution when the use\_prior argument is TRUE.

**Value**

The function dLL returns a numeric of the derivative of the log-likelihood (or log-posterior) given a respondent's answer profile.

**Note**

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

**Author(s)**

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

**See Also**

[Cat-class](#), [prior](#)

**Examples**

```
## Not run:
## Create Cat object, store example answers, and calculate
## first derivative of log-likelihood at theta = 1

## dLL for Cat object of the ltm model
data(npi)
ltm_cat <- ltmCat(npi)
setAnswers(ltm_cat) <- c(1,0,1,0,1, rep(NA, 35))
dLL(ltm_cat, theta = 1)

## dLL for Cat object of the tpm model
data(polknow)
tpm_cat <- tpmCat(polknow)
setAnswers(tpm_cat) <- c(1,0,1,0, rep(NA, 35))
dLL(tpm_cat, theta = 1)

## dLL for Cat object of the grm model
data(nfc)
grm_cat <- grmCat(nfc)
setAnswers(grm_cat) <- c(1,3,4,5, rep(NA, 13))
dLL(grm_cat, theta = 1)

## End(Not run)
```

---

estimateSE

*Standard Error of Ability Parameter Estimate*

---

**Description**

Estimates the standard error for a respondent's ability parameter estimate,  $\theta$ .

**Usage**

```
estimateSE(catObj)
```

**Arguments**

catObj                      An object of class Cat

**Details**

The function estimateSE estimates the standard error of the ability estimate given the estimation approach of the Cat object, specified in estimation slot of Cat object.

The expected a posteriori approach is used when estimation slot is "EAP". This method involves integration. See **Note** for more information.

The modal a posteriori approach is used when estimation slot is "MAP". This method is only available using the normal prior distribution.



The maximum likelihood approach is used when estimation slot is "MLE". When the likelihood of the ability estimate is undefined, the MAP or EAP method will be used, determined by what is specified in the estimationDefault slot in Cat object.

The weighted maximum likelihood approach is used when estimation slot is "WLE". Estimating  $\theta$  requires root finding with the "Brent" method in the GNU Scientific Library (GSL) with initial search interval of  $[-5, 5]$ .

### Value

The function estimateSE returns a numeric for the standard error for  $\theta$ .

### Note

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

This function uses adaptive quadrature methods from the GNU Scientific Library (GSL) to approximate single-dimensional integrals with high accuracy. The bounds of integration are determined by the lowerBound and upperBound slots of the Cat object.

### Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

### See Also

[estimateTheta](#)

### Examples

```
## Not run:
## Create Cat object
data(npi)
ltm_cat <- ltmCat(npi)

## Store example answers
setAnswers(ltm_cat) <- c(1,0,1,0,1, rep(NA, 35))

## Set different estimation procedures and calculate
## ability estimate and its standard error
setEstimation(ltm_cat) <- "EAP"
estimateTheta(ltm_cat)
estimateSE(ltm_cat)

setEstimation(ltm_cat) <- "MAP"
estimateTheta(ltm_cat)
estimateSE(ltm_cat)

setEstimation(ltm_cat) <- "MLE"
estimateTheta(ltm_cat)
estimateSE(ltm_cat)

setEstimation(ltm_cat) <- "WLE"
estimateTheta(ltm_cat)
estimateSE(ltm_cat)
```

```
## End(Not run)
```

---

estimateTheta	<i>Estimate of the Respondent's Ability Parameter</i>
---------------	---

---

### Description

Estimates the expected value of the ability parameter  $\theta$ , conditioned on the observed answers, prior, and the item parameters.

### Usage

```
estimateTheta(catObj)
```

### Arguments

catObj            An object of class Cat

### Details

Estimation approach is specified in estimation slot of Cat object.

The expected a posteriori approach is used when estimation slot is "EAP". This method involves integration. See **Note** for more information.

The modal a posteriori approach is used when estimation slot is "MAP". This method is only available using the normal prior distribution.

The maximum likelihood approach is used when estimation slot is "MLE". When the likelihood is undefined, the MAP or EAP method will be used, determined by what is specified in the estimationDefault slot in Cat object.

The weighted maximum likelihood approach is used when estimation slot is "WLE". Estimating  $\theta$  requires root finding with the "Brent" method in the GNU Scientific Library (GSL) with initial search interval of  $[-5, 5]$ .

### Value

The function estimateTheta returns a numeric consisting of the expected value of the respondent's ability parameter.

### Note

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

This function uses adaptive quadrature methods from the GNU Scientific Library (GSL) to approximate single-dimensional integrals with high accuracy. The bounds of integration are determined by the lowerBound and upperBound slots of the Cat object.

### Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

**See Also**

[Cat-class](#), [estimateSE](#)

**Examples**

```
## Not run:
## Create Cat object
data(npi)
ltm_cat <- ltmCat(npi)

## Store example answers
setAnswers(ltm_cat) <- c(1,0,1,0,1, rep(NA, 35))

## Set different estimation procedures and estimate ability parameter
setEstimation(ltm_cat) <- "EAP"
estimateTheta(ltm_cat)

setEstimation(ltm_cat) <- "MAP"
estimateTheta(ltm_cat)

setEstimation(ltm_cat) <- "MLE"
estimateTheta(ltm_cat)

setEstimation(ltm_cat) <- "WLE"
estimateTheta(ltm_cat)

## End(Not run)
```

---

expectedKL

*Expected Kullback-Leibler Information*

---

**Description**

Calculates the expected Kullback-Leibler information for an individual question item.

**Usage**

```
expectedKL(catObj, item)
```

**Arguments**

catObj	An object of class Cat
item	An integer indicating the index of the question item

**Details**

The function `expectedKL` calculates the expected value of the Kullback-Leibler information for a specified item where the bounds of integration are  $\hat{\theta} \pm \delta$ , where  $\delta$  is  $z$  times the square root of the Fisher test information and  $z$  is specified in the `z` slot of the Cat object. See **Note** for more information on integration.

**Value**

The function `expectedKL` returns a numeric indicating the expected Kullback-Leibler information for the specified item, given the current answer profile and ability parameter estimate.

**Note**

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

This function uses adaptive quadrature methods from the GNU Scientific Library (GSL) to approximate single-dimensional integrals with high accuracy.

**Author(s)**

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

**See Also**

[likelihoodKL](#), [posteriorKL](#), [selectItem](#)

**Examples**

```
## Not run:
## Create Cat object
data(npi)
ltm_cat <- ltmCat(npi)

## Store example answers
setAnswers(ltm_cat) <- c(1,0,1,0,1, rep(NA, 35))

## Estimate EPV for different unasked items
expectedKL(ltm_cat, item = 10)
expectedKL(ltm_cat, item = 20)
expectedKL(ltm_cat, item = 30)

## End(Not run)
```

---

expectedObsInf

*Expected Observed Information*

---

**Description**

Calculates the expected information, which is the observed information attained from a specific response set times the probability of that profile occurring.

**Usage**

```
expectedObsInf(catObj, item)
```

**Arguments**

catObj	An object of class Cat
item	An integer indicating the index of the question item

**Value**

The function expectedObsInf returns a numeric value of the expected information.

**Note**

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

**Author(s)**

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

**See Also**

[estimateSE](#), [obsInf](#), [probability](#), [selectItem](#)

**Examples**

```
## Not run:
## Create Cat object
data(npi)
ltm_cat <- ltmCat(npi)

## Store example answers
setAnswers(ltm_cat) <- c(1,0,1,0,1, rep(NA, 35))

## Expected observed information for different items
expectedObsInf(ltm_cat, item = 10)
expectedObsInf(ltm_cat, item = 20)

## End(Not run)
```

---

expectedPV

*Expected Posterior Variance*

---

**Description**

Estimates the expected posterior variance for a respondent's estimated ability parameter for an item yet to be answered based on a respondent's ability parameter estimate from the already-answered items.

**Usage**

```
expectedPV(catObj, item)
```

**Arguments**

catObj	An object of class Cat
item	An integer indicating the index of the question item

**Value**

The function expectedPV returns a numeric value indicating a respondent's expected posterior variance for an unasked item.

**Note**

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

**Author(s)**

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

**See Also**

[estimateSE](#), [probability](#), [selectItem](#)

**Examples**

```
## Not run:
## Create Cat object
data(npi)
ltm_cat <- ltmCat(npi)

## Store example answers
setAnswers(ltm_cat) <- c(1,0,1,0,1, rep(NA, 35))

## Estimate EPV for different unasked items
expectedPV(ltm_cat, item = 10)
expectedPV(ltm_cat, item = 20)
expectedPV(ltm_cat, item = 30)

## End(Not run)
```

---

fisherInf

*Fisher's Information*


---

**Description**

Calculates the expected value of the observed information of the likelihood evaluated at the input value  $\theta$ .

**Usage**

```
fisherInf(catObj, theta, item)
```

**Arguments**

catObj	An object of class Cat
theta	A numeric or an integer indicating the potential value for $\theta$
item	An integer indicating the index of the question item

**Details**

For the dichotomous case, this Fisher's information is equivalent to the observed information.

**Value**

The function `fisherInf` returns a numeric of the expected value of the observed information of the likelihood evaluated at the input value  $\theta$ .

**Note**

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

**Author(s)**

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

**See Also**

[fisherTestInfo](#), [obsInf](#), [selectItem](#)

**Examples**

```
## Not run:
## Create Cat object
data(npi)
ltm_cat <- ltmCat(npi)

## Store example answers
setAnswers(ltm_cat) <- c(1,0,1,0,1, rep(NA, 35))

## Fisher's information for different items,
## at ability parameter of 1
fisherInf(ltm_cat, theta = 1, item = 10)
fisherInf(ltm_cat, theta = 1, item = 20)

## End(Not run)
```

---

fisherTestInfo	<i>Fisher's Test Information</i>
----------------	----------------------------------

---

**Description**

Calculates the total information gained for a respondent for all answered items, conditioned on *theta*.

**Usage**

```
fisherTestInfo(catObj)
```

**Arguments**

catObj	An object of class Cat
--------	------------------------

**Value**

The function `fisherTestInfo` returns a numeric indicating the total information gained for a respondent, given a specific answer set and the current estimate of *theta*.

**Note**

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

**Author(s)**

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

**See Also**

[fisherInf](#)

**Examples**

```
## Not run:
## Create Cat object
data(npi)
ltm_cat <- ltmCat(npi)

## Store example answers
setAnswers(ltm_cat) <- c(1,0,1,0,1, rep(NA, 35))

## Fisher's test information for answer profile
fisherTestInfo(ltm_cat)

## End(Not run)
```



**Description**

Getter methods to access slots of a Cat object.

**Usage**

```
## S4 method for signature 'Cat'  
getModel(catObj)
```

```
## S4 method for signature 'Cat'  
getGuessing(catObj)
```

```
## S4 method for signature 'Cat'  
getDiscrimination(catObj)
```

```
## S4 method for signature 'Cat'  
getDifficulty(catObj)
```

```
## S4 method for signature 'Cat'  
getAnswers(catObj)
```

```
## S4 method for signature 'Cat'  
getPriorName(catObj)
```

```
## S4 method for signature 'Cat'  
getPriorParams(catObj)
```

```
## S4 method for signature 'Cat'  
getLowerBound(catObj)
```

```
## S4 method for signature 'Cat'  
getUpperBound(catObj)
```

```
## S4 method for signature 'Cat'  
getEstimation(catObj)
```

```
## S4 method for signature 'Cat'  
getEstimationDefault(catObj)
```

```
## S4 method for signature 'Cat'  
getSelection(catObj)
```

```
## S4 method for signature 'Cat'  
getZ(catObj)
```

```
## S4 method for signature 'Cat'  
getLengthThreshold(catObj)
```

```
## S4 method for signature 'Cat'
getSeThreshold(catObj)

## S4 method for signature 'Cat'
getInfoThreshold(catObj)

## S4 method for signature 'Cat'
getGainThreshold(catObj)

## S4 method for signature 'Cat'
getLengthOverride(catObj)

## S4 method for signature 'Cat'
getGainOverride(catObj)
```

### Arguments

catObj            An object of class Cat

### Value

These functions return the respective slot from Cat object.

### Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson,  
Erin Rossiter, Min Hee Seo, Alex Weil

### See Also

[Cat-class](#), [setters](#)

### Examples

```
## Not run:
## Create Cat object
data(npi)
cat <- ltmCat(npi)

## Getting estimation slot before and after change
getEstimation(ltm_cat)
setEstimation(ltm_cat) <- "MAP"
getEstimation(ltm_cat)

## Getting discrimination slot before and after change
getDiscrimination(ltm_cat)
setDiscrimination(ltm_cat) <- rep(1, 40)
getDiscrimination(ltm_cat)

## End(Not run)
```

gpcmCat

*Computerized Adaptive Testing Generalized Partial Credit Model***Description**

This function fits the generalized partial credit model for ordinal polytomous data and populates the fitted values for discrimination and difficulty parameters to an object of class Cat.

**Usage**

```
## S4 method for signature 'data.frame'
gpcmCat(data, quadraturePoints = 21, ...)

## S4 method for signature 'gpcm'
gpcmCat(data, quadraturePoints = NULL, ...)
```

**Arguments**

data	A data frame of manifest variables or an object of class gpcm.
quadraturePoints	A numeric to be passed into the gpcm function indicating the number of Gauss-Hermite quadrature points. Only applicable when data is a data frame. Default value is 21.
...	arguments to be passed to methods. For more details about the arguments, see gpcm in the ltm package.

**Details**

The data argument of the function gpcmCat is either a data frame or an object of class gpcm from the ltm package. If it is a data frame each row represents a respondent and each column represents a question item. If it is an object of the class gpcm, it is output from the gpcm function in the ltm package.

The quadraturePoints argument of the function gpcmCat is used only when the data argument is a data frame. quadraturePoints is then passed to the gpcm function from the ltm package when fitting the generalized partial credit model to the data and is used when approximating the value of integrals.

**Value**

The function gpcmCat returns an object of class Cat with changes to the following slots:

- **difficulty** A list of difficulty parameters, where each element in the list corresponds to the difficulty parameters for an item.
- **discrimination** A vector consisting of of discrimination parameters for each item.
- **model** The string "gpcm", indicating this Cat object corresponds to a generalized partial credit model.

**Note**

In case the Hessian matrix at convergence is not positive definite try to use `start.val = "random"`.

**Author(s)**

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

**References**

Baker, Frank B. and Seock-Ho Kim. 2004. Item Response Theory: Parameter Estimation Techniques. New York: Marcel Dekker.

Muraki, Eiji. 1992. "A generalized partial credit model: Application of an EM algorithm." ETS Research Report Series 1992(1):1-30.

Rizopoulos, Dimitris. 2006. "ltm: An R Package for Latent Variable Modeling and Item Response Theory Analyses." Journal of Statistical Software 17(5):1-25.

**See Also**

[Cat-class](#), [grmCat](#)

---

grmCat

---

*Computerized Adaptive Testing Graded Response Model*


---

**Description**

This function fits the graded response model for ordinal polytomous data and populates the fitted values for discrimination and difficulty parameters to an object of class Cat.

**Usage**

```
## S4 method for signature 'data.frame'
grmCat(data, quadraturePoints = 21, ...)

## S4 method for signature 'grm'
grmCat(data, quadraturePoints = NULL, ...)
```

**Arguments**

<code>data</code>	A data frame of manifest variables or an object of class grm.
<code>quadraturePoints</code>	A numeric to be passed into the grm function indicating the number of Gauss-Hermite quadrature points. Only applicable when data is a data frame. Default value is 21.
<code>...</code>	arguments to be passed to methods. For more details about the arguments, see grm in the ltm package.

**Details**

The data argument of the function grmCat is either a data frame or an object of class grm from the ltm package. If it is a data frame each row represents a respondent and each column represents a question item. If it is an object of the class grm, it is output from the grm function in the ltm package.

The quadraturePoints argument of the function grmCat is used only when the data argument is a data frame. quadraturePoints is then passed to the grm function from the ltm package when fitting the graded response model to the data and is used when approximating the value of integrals.

## Value

The function `grmCat` returns an object of class `Cat` with changes to the following slots:

- `difficulty` A list of difficulty parameters, where each element in the list corresponds to the difficulty parameters for an item.
- `discrimination` A vector consisting of discrimination parameters for each item.
- `model` The string "grm", indicating this `Cat` object corresponds to a graded response model.

## Note

In case the Hessian matrix at convergence is not positive definite try to use `start.val = "random"`.

## Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

## References

- Baker, Frank B. and Seock-Ho Kim. 2004. Item Response Theory: Parameter Estimation Techniques. New York: Marcel Dekker.
- Samejima, Fumiko. 1969. "Estimation of Latent Ability Using a Response Pattern of Graded Scores." Psychometrika monograph supplement 34(4):100-114.
- Rizopoulos, Dimitris. 2006. "ltm: An R Package for Latent Variable Modeling and Item Response Theory Analyses." Journal of Statistical Software 17(5):1-25.

## See Also

[Cat-class](#), [gpcmCat](#)

## Examples

```
## Not run:
## Creating Cat object with raw data
data(nfc)
grm_cat1 <- grmCat(nfc, quadraturePoints = 100)

## Creating Cat object with fitted object of class grm
grm_fit <- grm(nfc, control = list(GHk = 100)) ## from ltm package
class(grm_fit)
grm_cat2 <- grmCat(grm_fit)

## Note the two Cat objects are identical
identical(grm_cat1, grm_cat2)

## Note the slots that have changed from default values
grm_cat1@model
grm_cat1@difficulty
grm_cat1@discrimination

## End(Not run)
```

---

likelihood	<i>Likelihood of the Specified Response Set</i>
------------	---

---

**Description**

Calculates the likelihood of a respondent, with ability parameter  $\theta$ , having offered the specific set of responses stored in the Cat objects answers slot. All calculations are conditional on the item-level parameters stored in the Cat object.

**Usage**

```
likelihood(catObj, theta)
```

**Arguments**

catObj	An object of class Cat
theta	A numeric or an integer indicating the value for $\theta_j$

**Value**

The function likelihood returns a numeric value of the likelihood of the respondent having offered the provided response profile.

**Note**

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

**Author(s)**

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

**See Also**

[Cat-class](#), [probability](#)

**Examples**

```
## Not run:
## Create Cat object, store example answers, and calculate
## likelihood at theta = 1

## Likelihood for Cat object of the ltm model
data(npi)
ltm_cat <- ltmCat(npi)
setAnswers(ltm_cat) <- c(1,0,1,0,1, rep(NA, 35))
likelihood(ltm_cat, theta = 1)

## Likelihood for Cat object of the tpm model
data(polknow)
tpm_cat <- tpmCat(polknow)
setAnswers(tpm_cat) <- c(1,0,1,0, rep(NA, 35))
```

```
likelihood(tpm_cat, theta = 1)

## Likelihood for Cat object of the grm model
data(nfc)
grm_cat <- grmCat(nfc)
setAnswers(grm_cat) <- c(1,3,4,5, rep(NA, 13))
likelihood(grm_cat, theta = 1)

## End(Not run)
```

---

likelihoodKL	<i>Expected Kullback-Leibler Information, Weighted by Likelihood</i>
--------------	--

---

## Description

Calculates the expected Kullback-Leibler information, weighted by likelihood, for a specified item.

## Usage

```
likelihoodKL(catObj, item)
```

## Arguments

catObj	An object of class Cat
item	An integer indicating the index of the question item

## Details

The function `likelihoodKL` calculates the expected Kullback-Leibler information for  $\hat{\theta}$ , weighting potential true values of  $\theta$ ,  $\theta_0$  by the likelihood of  $\theta_0$ .

This function involves integration. See **Note** for more information.

## Value

The function `likelihoodKL` returns a numeric indicating the expected Kullback-Leibler information weighted by the likelihood for the specified item, given the current answer profile and ability parameter estimate.

## Note

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

This function uses adaptive quadrature methods from the GNU Scientific Library (GSL) to approximate single-dimensional integrals with high accuracy. The bounds of integration are determined by the `lowerBound` and `upperBound` slots of the Cat object.

## Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

**See Also**

[expectedKL](#), [posteriorKL](#), [selectItem](#)

**Examples**

```
## Not run:
## Create Cat object
data(npi)
ltm_cat <- ltmCat(npi)

## Store example answers
setAnswers(ltm_cat) <- c(1,0,1,0,1, rep(NA, 35))

## Estimate EPV for different unasked items
likelihoodKL(ltm_cat, item = 10)
likelihoodKL(ltm_cat, item = 20)
likelihoodKL(ltm_cat, item = 30)

## End(Not run)
```

---

lookAhead

*Look Ahead to Select Next Item*


---

**Description**

Selects the next item that would be asked for all possible response options to the question the respondent is currently answering.

**Usage**

```
lookAhead(catObj, item)
```

**Arguments**

catObj	An object of class Cat
item	A numeric indicating the question item the respondent is currently answering.

**Value**

A function lookAhead returns a list of one element, estimates, a data.frame. The the first column of the data.frame is the possible response option to the question the respondent is currently answering and the second column is the next item that should be asked given each response.

**Note**

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

**Author(s)**

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil



**See Also**[selectItem](#)**Examples**

```
## Not run:
## Create Cat object
data(npi)
ltm_cat <- ltmCat(npi)

## Store example answers
setAnswers(ltm_cat) <- c(1,0,1,0,1, rep(NA, 35))

## What should be asked next if respondent is currently answering item 6
lookAhead(ltm_cat, 6)

## End(Not run)
```

ltmCat

*Computerized Adaptive Testing Latent Trait Model***Description**

This function fits the latent trait model for binary data and populates the fitted values for discrimination and difficulty parameters to an object of class Cat.

**Usage**

```
## S4 method for signature 'data.frame'
ltmCat(data, quadraturePoints = 21, ...)

## S4 method for signature 'ltm'
ltmCat(data, quadraturePoints = NULL, ...)
```

**Arguments**

<code>data</code>	A data frame of manifest variables or an object of class ltm.
<code>quadraturePoints</code>	A numeric to be passed into the ltm function indicating the number of Gauss-Hermite quadrature points. Only applicable when data is a data frame. Default value is 21.
<code>...</code>	arguments to be passed to methods. For more details about the arguments, see ltm in the ltm package.

**Details**

The data argument of the function ltmCat is either a data frame or an object of class ltm from the ltm package. If it is a data frame each row represents a respondent and each column represents a question item. If it is an object of the class ltm, it is output from the ltm function in the ltm package.

The `quadraturePoints` argument of the function `ltmCat` is used only when the data argument is a data frame. `quadraturePoints` is then passed to the `ltm` function from the `ltm` package when fitting the latent trait model to the data and is used when approximating the value of integrals.

### Value

The function `ltmCat` returns an object of class `Cat` with changes to the following slots:

- `difficulty` A vector consisting of difficulty parameters for each item.
- `discrimination` A vector consisting of discrimination parameters for each item.
- `model` The string "ltm", indicating this `Cat` object corresponds to a latent trait model.

### Note

In case the Hessian matrix at convergence is not positive definite try to use `start.val = "random"`.

### Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

### References

Baker, Frank B. and Seock-Ho Kim. 2004. Item Response Theory: Parameter Estimation Techniques. New York: Marcel Dekker.

Rizopoulos, Dimitris. 2006. "ltm: An R Package for Latent Variable Modeling and Item Response Theory Analyses." Journal of Statistical Software 17(5):1-25.

### See Also

[Cat-class](#), [tpmCat](#)

### Examples

```
## Not run:
## Creating Cat object with raw data
data(npi)
ltm_cat1 <- ltmCat(npi, quadraturePoints = 100)

## Creating Cat object with fitted object of class tpm
ltm_fit <- grm(npi, control = list(GHk = 100)) ## from ltm package
class(ltm_fit)
ltm_cat2 <- ltmCat(ltm_fit)

## Note the two Cat objects are identical
identical(ltm_cat1, ltm_cat2)

## Note the slots that have changed from default values
ltm_cat1@model
ltm_cat1@difficulty
ltm_cat1@discrimination

## End(Not run)
```

makeTree

*Make Tree of Possible Question Combinations***Description**

Pre-calculates a complete branching scheme of all possible questions-answer combinations and stores it as a list of lists or a flattened table of values.

**Usage**

```
makeTree(catObj, flat = FALSE)
```

**Arguments**

catObj	An object of class Cat
flat	A logical indicating whether to return tree as as a list of lists or a table

**Details**

The function takes a Cat object and generates a tree of all possible question-answer combinations, conditional on previous answers in the branching scheme and the current  $\theta$  estimates for the branch. The tree is stored as a list of lists, iteratively generated by filling in a possible answer, calculating the next question via selectItem, filling in a possible answer for that question, and so forth.

The length of each complete branching scheme within the tree is dictated by the lengthThreshold slot within the Cat object.

**Value**

The function makeTree returns either a list or a table. If the argument flat is FALSE, the default value, the function returns a list of lists.

If the argument flat is TRUE, the function takes the list of lists and configures it into a flattened table where the columns represent the battery items and the rows represent the possible answer profiles.

**Note**

This function is computationally expensive. If there are  $k$  response options and the researcher wants a complete branching scheme to include  $n$  items,  $k^{n-1}$  complete branching schemes will be calculated. Setting  $n$  is done via the lengthThreshold slot in the Cat object. See **Examples**.

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

**Author(s)**

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

**See Also**

[Cat-class](#), [checkStopRules](#), [selectItem](#)

## Examples

```
## Not run:
## Creating Cat object of ltm model
data(npi)
ltm_cat <- ltmCat(npi)

## Setting complete branches to include 3 items
setLengthThreshold(ltm_cat) <- 3

## Object returned is list of lists
ltm_list <- makeTree(ltm_cat)

## Object returned is table
ltm_table <- makeTree(ltm_cat, flat = TRUE)

## End(Not run)
```

---

nfc

*Need For Cognition*


---

## Description

Data of survey respondents' responses to 18 NFC questions, which is a reduced version of NFC. For each question, respondents could choose one of five response options: Agree strongly, agree somewhat, neither agree nor disagree, disagree somewhat, disagree strongly. Missingness is optional. Observations come from one of three administrations of the survey: The American Panel Survey, Amazon's Mechanical Turk in fall of 2014, or Amazon's Mechanical Turk in spring of 2015.

## Usage

```
data(nfc)
```

## Format

A data frame with 4043 observations on the following 18 variables.

NFC1 I really enjoy a task that involves coming up with new solutions to problems

NFC4 I would prefer a task that is intellectual, difficult, and important to one that is somewhat important but does not require much thought

NFC10 Learning new ways to think doesn't excite me very much

NFC12 I usually end up deliberating about issues even when they do not affect me personally

NFC15 The idea of relying on thought to make my way to the top appeals to me

NFC16 The notion of thinking abstractly is appealing to me

NFC19 I only think as hard as I have to

NFC21 I think tasks that require little thought once I've learned them

NFC22 I prefer to think about small, daily projects to long-term ones

- NFC23 I would rather do something that requires little thought than something that is sure to challenge my thinking abilities
- NFC24 I find satisfaction in deliberating hard and for long hours
- NFC29 I like to have the responsibility of handling a situation that requires a lot of thinkings
- NFC31 I feel relief rather than satisfaction after completing a task that required a lot of mental effort
- NFC32 Thinking is not my idea of fun
- NFC33 I try to anticipate and avoid situations where there is likely a chance I will have to think in depth about something
- NFC39 I prefer my life to be filled with puzzles that I must solve
- NFC40 I would prefer complex to simple problems
- NFC43 Its enough for me that something gets the job done; I dont care how or why it works

npi

*Narcissistic Personality Inventory*

### Description

Data from the test with the 40 paired items on one page. Respondents had to choose the one that fit them the best. Missingness is optional. Respondents' demographics were entered on the next page. In addition, they were also asked to affirm that their answers were accurate and suitable for research, those that did not (9%) are not included in this dataset.

### Usage

```
data(npi)
```

### Format

A data frame with 11,243 observations on the following 40 variables.

- Q1 0 = I have a natural talent for influencing people. 1 = I am not good at influencing people.
- Q2 0 = Modesty doesn't become me. 1 = I am essentially a modest person.
- Q3 0 = I would do almost anything on a dare. 1 = I tend to be a fairly cautious person.
- Q4 0 = When people compliment me I sometimes get embarrassed. 1 = I know that I am good because everybody keeps telling me so.
- Q5 0 = The thought of ruling the world frightens the hell out of me. 1 = If I ruled the world it would be a better place.
- Q6 0 = I can usually talk my way out of anything. 1 = I try to accept the consequences of my behavior.
- Q7 0 = I prefer to blend in with the crowd. 1 = I like to be the center of attention.
- Q8 0 = I will be a success. 1 = I am not too concerned about success.
- Q9 0 = I am no better or worse than most people. 1 = I think I am a special person.
- Q10 0 = I am not sure if I would make a good leader. 1 = I see myself as a good leader.
- Q11 0 = I am assertive. 1 = I wish I were more assertive.
- Q12 0 = I like to have authority over other people. 1 = I don't mind following orders.

- Q13 0 = I find it easy to manipulate people. 1 = I don't like it when I find myself manipulating people.
- Q14 0 = I insist upon getting the respect that is due me. 1 = I usually get the respect that I deserve.
- Q15 0 = I don't particularly like to show off my body. 1 = I like to show off my body.
- Q16 0 = I can read people like a book. 1 = People are sometimes hard to understand.
- Q17 0 = If I feel competent I am willing to take responsibility for making decisions. 1 = I like to take responsibility for making decisions.
- Q18 0 = I just want to be reasonably happy. 1 = I want to amount to something in the eyes of the world.
- Q19 0 = My body is nothing special. 1 = I like to look at my body.
- Q20 0 = I try not to be a show off. 1 = I will usually show off if I get the chance.
- Q21 0 = I always know what I am doing. 1 = Sometimes I am not sure of what I am doing.
- Q22 0 = I sometimes depend on people to get things done. 1 = I rarely depend on anyone else to get things done.
- Q23 0 = Sometimes I tell good stories. 1 = Everybody likes to hear my stories.
- Q24 0 = I expect a great deal from other people. 1 = I like to do things for other people.
- Q25 0 = I will never be satisfied until I get all that I deserve. 1 = I take my satisfactions as they come.
- Q26 0 = Compliments embarrass me. 1 = I like to be complimented.
- Q27 0 = I have a strong will to power. 1 = Power for its own sake doesn't interest me.
- Q28 0 = I don't care about new fads and fashions. 1 = I like to start new fads and fashions.
- Q29 0 = I like to look at myself in the mirror. 1 = I am not particularly interested in looking at myself in the mirror.
- Q30 0 = I really like to be the center of attention. 1 = It makes me uncomfortable to be the center of attention.
- Q31 0 = I can live my life in any way I want to. 1 = People can't always live their lives in terms of what they want.
- Q32 0 = Being an authority doesn't mean that much to me. 1 = People always seem to recognize my authority.
- Q33 0 = I would prefer to be a leader. 1 = It makes little difference to me whether I am a leader or not.
- Q34 0 = I am going to be a great person. 1 = I hope I am going to be successful.
- Q35 0 = People sometimes believe what I tell them. 1 = I can make anybody believe anything I want them to.
- Q36 0 = I am a born leader. 1 = Leadership is a quality that takes a long time to develop.
- Q37 0 = I wish somebody would someday write my biography. 1 = I don't like people to pry into my life for any reason.
- Q38 0 = I get upset when people don't notice how I look when I go out in public. 1 = I don't mind blending into the crowd when I go out in public.
- Q39 0 = I am more capable than other people. 1 = There is a lot that I can learn from other people.
- Q40 0 = I am much like everybody else. 1 = I am an extraordinary person.

---

obsInf	<i>Observed Information</i>
--------	-----------------------------

---

**Description**

Calculates the observed information of the likelihood of a respondent's ability  $\theta$  for a given item.

**Usage**

```
obsInf(catObj, theta, item)
```

**Arguments**

catObj	An object of class Cat
theta	A numeric or an integer indicating the value for $\theta$
item	An integer indicating the index of the question item

**Details**

The observed information is equivalent to the negative second derivative of the log-likelihood evaluated at  $\theta$ . This function should never be called when the respondent has answered no questions as the likelihood is not defined.

**Value**

The function obsInf returns a numeric value of the observed information of the likelihood, given  $\theta$ , for a given question item.

**Note**

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

**Author(s)**

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

**See Also**

[estimateTheta](#), [expectedObsInf](#)

**Examples**

```
## Not run:
## Create Cat object, store example answers, and calculate observed information
for an ability parameter of 1 for item 10

## observed information for Cat object of the ltm model
data(npi)
ltm_cat <- ltmCat(npi)
setAnswers(ltm_cat) <- c(1,0,1,0,1, rep(NA, 35))
```

```

obsInf(ltm_cat, theta = 1, item = 10)

## observed information for Cat object of the tpm model
data(polknow)
tpm_cat <- tpmCat(polknow)
setAnswers(tpm_cat) <- c(1,0,1,0, rep(NA, 35))
obsInf(tpm_cat, theta = 1, item = 10)

## observed information for Cat object of the grm model
data(nfc)
grm_cat <- grmCat(nfc)
setAnswers(grm_cat) <- c(1,3,4,5, rep(NA, 13))
obsInf(grm_cat, theta = 1, item = 10)

## End(Not run)

```

---

polknowMT

*MTurk Political Knowledge*

---

## Description

Data from Amazon Mechanical Turkers in 2012 responding to 64 political knowledge questions. Political knowledge questions had a varying number of response options, noted below. This dataset codes answers as either correct (a value of 0) or incorrect (a value of 1).

## Usage

```
data(polknowMT)
```

## Format

A data frame with 810 observations on the following 64 questions.

- Q1 How long is one term for the President of the United States?  
Eight years; Six years; Four years; Two years
- Q2 The FDA is part of the national government primarily responsible for regulating  
Food quality; The national parks; Electricity production and energy; Pollution and the environment
- Q3 Who is the Vice President of the United States?  
Leon Panetta; William Daley; Hillary Clinton; Joe Biden
- Q4 The federal debt is  
Much smaller than it was 20 years ago; The difference between imports and exports with foreign countries; The annual difference between spending and tax revenues; The accumulated borrowing of the federal government that has not been repaid
- Q5 How many times can an individual be elected President of the United States under current laws?  
Any number of terms; Three times; Twice; Once
- Q6 What do we call the first ten amendments to the Constitution?  
The Articles of Confederation; The inalienable right; The Bill of Rights; The Declaration of Independence



- Q7 Is the U.S. federal budget deficit, the amount by which the governments spending exceeds the amount of money it collects, now bigger, about the same, or smaller than it was during most of the 1990s?  
Smaller; About the same; Bigger
- Q8 Who signs bills to become laws?  
The President; The Vice President; The Chief Justice of the Supreme Court; The Secretary of State
- Q9 Which party is generally more supportive of creating a way for immigrants who are in the U.S. illegally to eventually become citizens?  
The Republican Party; The Democratic Party
- Q10 In what month do we vote for the President?  
November; October; February; January
- Q11 What are the two parts of the U.S. Congress?  
The Senate and the Supreme Court; The House of Lords and the House of Commons; The House of Representative and the Supreme Court; The Senate and House of Representatives
- Q12 Which party is generally more supportive of restricting access to abortion?  
The Republican Party; The Democratic Party
- Q13 Which of these political parties is considered most conservative?  
Green Party; Republican Party; Democratic Party
- Q14 The NRA is an organization that advocates for  
Clean elections; A cleaner environment; The rights of gun owners; Women's rights
- Q15 Compared to 30 years ago, is the difference in incomes between the top 20% of households and the bottom 20% of households now bigger, smaller, or the same?  
The difference is now the same as 30 years ago; The difference is now smaller than 30 years ago; The difference is now bigger than 30 years ago
- Q16 The EPA is part of the national government primarily responsible for regulating  
Food quality; The national parks; Electricity production and energy; Pollution and the environment
- Q17 Which party is generally more supportive of reducing the size of the defense budget?  
The Republican Party; The Democratic Party
- Q18 Which party is generally more supportive of increasing taxes on higher income people to reduce the federal budget deficit?  
The Republican Party; The Democratic Party
- Q19 Which party is generally more supportive of allowing drilling for oil in the Arctic National Wildlife Refuge?  
The Republican Party; The Democratic Party
- Q20 Who is the Commander in Chief of the military?  
The Attorney General; The President; The Secretary of Defense; The Vice President
- Q21 The First Amendment to the United States Constitution guarantees all of these rights EXCEPT  
Right to remain silent; Right to the free exercise of religion; Right to free speech; Right to peaceably assemble
- Q22 Roe v Wade is a case decided by the Supreme Court that relates to  
Executive power; Campaign finance; Birth control; Abortion
- Q23 Social Security is  
Funded by the personal income tax; Operated by state government; The responsibility of the Department of Defense; The benefit program for senior citizens

- Q24 What is Medicare?  
A private, non-profit organization that runs free health clinic; A private health insurance plan sold to individuals in all 50 states; A program run by state governments to provide health care to poor people; A program run by the U.S. federal government to pay for old people's health care
- Q25 How many senators are elected from each state?  
It depends on the population of the state; Four; Two; One
- Q26 How many votes are required in Congress to override a presidential veto  
A simple majority of both houses of Congress; A simple majority of one house of Congress; A two-thirds majority of both houses of Congress; A two-thirds majority of one house of Congress
- Q27 The Secretary of State  
Serves a two-year term; Serves the state governments; Is nominated by the president; Heads the armed services
- Q28 Near the end of an election campaign, a polls shows that an issue that no candidate has mentioned is of great concern to voters. What is most likely to happen?  
Some candidates will drop out of the race; Candidates will start talking about the issue; Newspapers will not report the results of the poll; The election will be postponed
- Q29 Liberals are generally said to  
Support pro-life policies; Oppose all tax increases; Support military spending; Support government programs to give government assistance the needy
- Q30 Which party is generally more supportive of reducing the size and scope of the federal government?  
The Republican Party; The Democratic Party
- Q31 The ability of a minority of senators to prevent a vote on a bill is known as  
Suspension of the rules; Enrollment; A veto; A filibuster
- Q32 conservatives are generally said to  
Support pro-choice policies; Support tax cuts; Oppose military spending; Support government programs to give government assistance to the needy
- Q33 Which of these regions of the country is generally considered to be most supportive of Republican candidates  
Midwest; South; West Coast; New England
- Q34 The presiding officer in the House of Representatives is  
The Majority Leader; The Sergeant at Arms; The Vice President of the United States; The Speaker
- Q35 Which of these countries is NOT a permanent member of the U.N. Security Council  
United Kingdom; France; India; China
- Q36 Which part has a majority of seats in the U.S. House of Representatives?  
Neither; Democrats; Republicans
- Q37 What state holds the first primary election in Presidential primaries?  
Florida; Nevada; South Carolina; New Hampshire
- Q38 Who is the Speaker of the House of Representatives?  
Mitt Romney; Eric Holder; Harry Reid; John Boehner
- Q39 Most cases are considered by the Supreme Court  
In even-numbered years; At the request of the Congress; Upon order of the president; With the approval of at least four justices

- Q40 How many Justices typically serve on the U.S. Supreme Court  
Eleven; Nine; Eight; Seven
- Q41 What job or political office is not held by Ben Bernanke?  
None of these; Minority Whip of the U.S. House; Chief Justice of the United States Supreme Court; Majority leader of the U.S. Senate, Chairman of the Federal Reserve
- Q42 Whose responsibility is it to nominate judges to the Federal Courts  
The state governors; The Supreme Court; Congress; President
- Q43 Who is the Chief Justice of the U.S. Supreme Court?  
Larry Thompson; Anthony Kennedy; David Cole; John Roberts
- Q44 The U.S. Senate  
Votes to confirm nominees to the U.S. Supreme court chosen by the House of Representatives; Plays no role in choosing the members of the U.S. Supreme Court; Chooses members of the U.S. Supreme Court; Votes to confirm nominees to the U.S. Supreme Court chosen by the President
- Q45 Which party has a majority of seats in the U.S. Senate  
Neither; Democrats; Republicans
- Q46 Which of the states listed below has the greatest number of electoral college votes in the U.S. Presidential Elections?  
Puerto Rico; Nevada; North Dakota; Washington, D.C.
- Q47 Citizens United v the FEC is a case decided by the Supreme Court that relates to  
Executive power; Campaign finance; Birth control; Abortion
- Q48 For how many years is a United States Senator elected that is, how many years are there in one full term of office for a U.S. Senator?  
None of these; Eight years; Six years; Four years; Two years
- Q49 Who is the Prime Minister of the United Kingdom?  
Richard Branson; Tony Hayward; Nick Clegg; David Cameron
- Q50 The president of Afghanistan is named  
Bashar al-Assad; Hosni Mubarak; Hamid Karzai; Nouri al-Maliki
- Q51 The House of Representatives has how many voting members?  
Four hundred and forty-one; Four hundred and thirty-five; Two hundred; One hundred
- Q52 The President of the Senate is  
The Majority Leader; The Sergeant at Arms; The Vice President of the United States; The senior senator of the majority party
- Q53 On which of the following federal programs is the most money spent each year?  
Medicare; Education; Subsidies to farmers; Aid to foreign countries
- Q54 What do all constitutional governments have?  
Separation of church and state; A bill of rights; A President as the head of government; Limits on political power
- Q55 One which of the following does the U.S. federal government spend the least money?  
Social Security; National defense; Medicare; Foreign aid
- Q56 The head of the Department of Justice is  
Kathleen Sebelius; Eric Holder; Timothy Geithner; Hillary Clinton
- Q57 The president may NOT  
Declare war; Pardon criminals without justification; Appoint federal officials when Congress is in recess; Refuse to sign legislation passed by Congress

- Q58 Which of these is NOT primarily the responsibility of the Federal government in Washington?  
Interstate commerce; Negotiating treaties with foreign countries; Education; National defense
- Q59 Who is the current president of Mexico?  
Vincente Fox; Hugo Chavez; Dilma Rousseff; Felipe Calderon
- Q60 Which of the following actions does the United States federal government commonly take to finance a budget deficit?  
Expanding public-works projects; Borrowing from the public; Imposing import quotas; Printing more money
- Q61 Common Cause is an organization that advocates for  
Women's rights; Clean elections; A cleaner environment; The right of gun owners
- Q62 The Byrd Rule is relevant  
During the confirmation of cabinet members; For national party conventions; During Congressional debates over non-budgetary policies; For the Reconciliation process
- Q63 The Majority Leader of the House of Representative is  
Nancy Pelosi; Kevin McCarthy; Eric Cantor; John Boehner
- Q64 On which of the following does the U.S. federal government spend the most money each year?  
Education; Medicare; Interest on the national debt; National defense

### Source

<https://dataverse.harvard.edu/dataset.xhtml?persistentId=hdl:1902.1/19381>

### References

Jacob M. Montgomery and Joshua Cutler. 2013. "Computerized Adaptive Testing for Public Opinion Surveys." *Political Analysis* (Spring 2013) 21 (2): 172-192.

---

polknowTAPS

*TAPS Political Knowledge*

---

### Description

Data of responses to a political knowledge battery from the May 2013 wave of The American Panel Survey (TAPS) out of Washington University in St. Louis.

### Usage

```
data(polknowTAPS)
```

### Format

A data frame with 1496 observations on the following 10 questions.

- Q1 Members of the U.S. Supreme Court serve  
1 = two-year terms. 2 = ten-year terms. 3 = life terms. 4 = terms determined by the president.  
5 = Don't know.
- Q2 Who is the Chief Justice of the U.S. Supreme Court?  
1 = John Roberts. 2 = Antonin Scalia. 3 = Mitt Romney. 4 = Hillary Clinton. 5 = Don't know.

- Q3 Social Security is  
 1 = the benefit program for senior citizens. 2 = the responsibility of the Department of Defense.  
 3 = operated by state governments. 4 = funded by the personal income tax. 5 = Don't know.
- Q4 On which of the following programs is the most money spent each year?  
 1 = aid to foreign countries. 2 = Medicare. 3 = subsidies to farmers. 4 = education. 5 = Don't know.
- Q5 Which party holds a majority of seats in the U.S. House of Representatives in Washington?  
 1 = Democrats. 2 = Republicans. 3 = Independents. 4 = Don't know.
- Q6 How many votes are required in Congress to override a presidential veto?  
 1 = a simple majority of one house of Congress. 2 = a simple majority of both houses of Congress. 3 = a two-thirds majority of one house of Congress. 4 = a two-thirds majority of both houses of Congress. 5 = Don't know.
- Q7 How long is one term for a member of the U.S. Senate?  
 1 = two years. 2 = four years. 3 = six years. 4 = eight years. 5 = Don't know.
- Q8 The ability of a minority of senators to prevent a vote on a bill is known as  
 1 = a veto. 2 = a filibuster. 3 = enrollment. 4 = suspension of the rules. 5 = Don't know.
- Q9 Who is the Vice President of the United States?  
 1 = Nancy Pelosi. 2 = John Boehner. 3 = Joseph Biden. 4 = Harry Reid. 5 = Don't know.
- Q10 A president may serve  
 1 = one term. 2 = two terms. 3 = three terms. 4 = any number of terms. 5 = Don't know.

## Details

TAPS is a monthly online survey of a national probability sample from a panel of about 2,000 adults in the United States. The panel was recruited in the fall of 2011. An address-based sampling (ABS) frame. That is, from a random selection of residential addresses, stratified using ancillary data on age and ethnic group, a panel of more than 2,000 was recruited. TAPS surveys are administered online; selected panelists who do not have a computer or online service are provided a computer and internet access by TAPS.

## Source

<http://taps.wustl.edu/data-archive>

---

posteriorKL

*Expected Kullback-Leibler Information, Weighted by the Prior*

---

## Description

Calculates the expected Kullback-Leibler information, weighted by likelihood and prior beliefs, for a specified item.

## Usage

```
posteriorKL(catObj, item)
```

**Arguments**

catObj	An object of class Cat
item	An integer indicating the index of the question item

**Details**

The function `posteriorKL` calculates the expected Kullback-Leibler information for  $\hat{\theta}$ , weighting potential true values of  $\theta$ ,  $\theta_0$  by the likelihood and posterior estimate of  $\theta_0$ .

This function involves integration. See **Note** for more information.

**Value**

The function `posteriorKL` returns a numeric indicating the expected Kullback-Leibler information weighted by the likelihood for the specified item, given the current answer profile and ability parameter estimate.

**Note**

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

This function uses adaptive quadrature methods from the GNU Scientific Library (GSL) to approximate single-dimensional integrals with high accuracy. The bounds of integration are determined by the `lowerBound` and `upperBound` slots of the Cat object.

**Author(s)**

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

**See Also**

[expectedKL](#), [likelihoodKL](#), [selectItem](#)

**Examples**

```
## Not run:
## Create Cat object
data(npi)
ltm_cat <- ltmCat(npi)

## Store example answers
setAnswers(ltm_cat) <- c(1,0,1,0,1, rep(NA, 35))

## Estimate EPV for different unasked items
posteriorKL(ltm_cat, item = 10)
posteriorKL(ltm_cat, item = 20)
posteriorKL(ltm_cat, item = 30)

## End(Not run)
```

prior

*Evaluate the Prior Density Distribution at Position x***Description**

Calculates the density at x of either the normal, Student's t, or uniform distribution.

**Usage**

```
prior(x, dist, params)
```

**Arguments**

x	A numeric value at which to evaluate the prior
dist	A string indicating the distribution (slot priorName of Cat object)
params	A length two numeric vector indicating the parameters of the distribution (slot priorParams of Cat object)

**Details**

The dist argument needs to be either "UNIFORM", "NORMAL", or "STUDENT\_T".

When dist is "NORMAL", the first element of params is the mean, the second element is the standard deviation.

When dist is "STUDENT\_T", the first element of params is the non-centrality parameters and the second is degrees of freedom.

When dist is "UNIFORM", the elements of params are the lower and upper bounds, of the interval, respectively. Note that the "UNIFORM" is only applicable for the expected a posteriori (EAP) estimation method.

**Value**

The function prior returns a numeric consisting of prior value,  $\pi(x)$ , given the value x.

**Note**

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

This function uses Boost C++ source libraries for the uniform and Student's t distributions and calls dnorm4 written in C which is identical to that of dnorm in R.

**Author(s)**

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

**See Also**

[Cat-class](#)

## Examples

```
## Not run:
## Create Cat object
data(npi)
cat <- ltmCat(npi)

## Prior calculation for different distributions
cat@priorName <- "NORMAL"
cat@priorParams <- c(0, 1) ## Parameters are mean and standard deviation
prior(x = 1, cat@priorName, cat@priorParams)

cat@priorName <- "STUDENT_T"
cat@priorParams <- c(1, 3) ## Parameters are non-centrality param and degrees of freedom
prior(x = 1, cat@priorName, cat@priorParams)

cat@priorName <- "UNIFORM"
cat@priorParams <- c(-1, 1) ## Parameters are lower bound and upper bound of interval
prior(x = 1, cat@priorName, cat@priorParams)

## End(Not run)
```

---

probability	<i>Probability of Responses to a Question Item or the Left-Cumulative Probability of Responses</i>
-------------	--

---

## Description

Calculates the probability of specific responses or the left-cumulative probability of responses to item conditioned on a respondent's ability ( $\theta$ ).

## Usage

```
probability(catObj, theta, item)
```

## Arguments

catObj	An object of class Cat
theta	A numeric or an integer indicating the value for $\theta_j$
item	An integer indicating the index of the question item

## Details

For the ltm model, the probability of non-zero response for respondent  $j$  on item  $i$  is

$$Pr(y_{ij} = 1|\theta_j) = \frac{\exp(a_i + b_i\theta_j)}{1 + \exp(a_i + b_i\theta_j)}$$

where  $\theta_j$  is respondent  $j$ 's position on the latent scale of interest,  $a_i$  is item  $i$ 's discrimination parameter, and  $b_i$  is item  $i$ 's difficulty parameter.

For the tpm model, the probability of non-zero response for respondent  $j$  on item  $i$  is



$$Pr(y_{ij} = 1|\theta_j) = c_i + (1 - c_i) \frac{\exp(a_i + b_i\theta_j)}{1 + \exp(a_i + b_i\theta_j)}$$

where  $\theta_j$  is respondent  $j$ 's position on the latent scale of interest,  $a_i$  is item  $i$ 's discrimination parameter,  $b_i$  is item  $i$ 's difficulty parameter, and  $c_i$  is item  $i$ 's guessing parameter.

For the grm model, the probability of a response in category  $k$  **or lower** for respondent  $j$  on item  $i$  is

$$Pr(y_{ij} < k|\theta_j) = \frac{\exp(\alpha_{ik} - \beta_i\theta_{ij})}{1 + \exp(\alpha_{ik} - \beta_i\theta_{ij})}$$

where  $\theta_j$  is respondent  $j$ 's position on the latent scale of interest,  $\alpha_{ik}$  the  $k$ -th element of item  $i$ 's difficulty parameter,  $\beta_i$  is discrimination parameter vector for item  $i$ . Notice the inequality on the left side and the absence of guessing parameters.

For the gpcm model, the probability of a response in category  $k$  for respondent  $j$  on item  $i$  is

$$Pr(y_{ij} = k|\theta_j) = \frac{\exp(\sum_{t=1}^k \alpha_i[\theta_j - (\beta_i - \tau_{it})])}{\sum_{r=1}^{K_i} \exp(\sum_{t=1}^r \alpha_i[\theta_j - (\beta_i - \tau_{it})])}$$

where  $\theta_j$  is respondent  $j$ 's position on the latent scale of interest,  $\alpha_i$  is the discrimination parameter for item  $i$ ,  $\beta_i$  is the difficulty parameter for item  $i$ , and  $\tau_{it}$  is the category  $t$  threshold parameter for item  $i$ , with  $k = 1, \dots, K_i$  response options for item  $i$ . For identification purposes  $\tau_{i0} = 0$  and  $\sum_{t=1}^1 \alpha_i[\theta_j - (\beta_i - \tau_{it})] = 0$ .

### Value

When the model slot of the catObj is "ltm", the function probability returns a numeric vector of length one representing the probability of observing a non-zero response.

When the model slot of the catObj is "tpm", the function probability returns a numeric vector of length one representing the probability of observing a non-zero response.

When the model slot of the catObj is "grm", the function probability returns a numeric vector of length  $k+1$ , where  $k$  is the number of possible responses. The first element will always be zero and the  $(k+1)$ th element will always be one. The middle elements are the cumulative probability of observing response  $k$  or lower.

When the model slot of the catObj is "gpcm", the function probability returns a numeric vector of length  $k$ , where  $k$  is the number of possible responses. Each number represents the probability of observing response  $k$ .

### Note

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

### Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

## References

- Baker, Frank B. and Seock-Ho Kim. 2004. Item Response Theory: Parameter Estimation Techniques. New York: Marcel Dekker.
- Choi, Seung W. and Richard J. Swartz. 2009. "Comparison of CAT Item Selection Criteria for Polytomous Items." Applied Psychological Measurement 33(6):419-440.
- Muraki, Eiji. 1992. "A generalized partial credit model: Application of an EM algorithm." ETS Research Report Series 1992(1):1-30.
- van der Linden, Wim J. 1998. "Bayesian Item Selection Criteria for Adaptive Testing." Psychometrika 63(2):201-216.

## See Also

[Cat-class](#)

## Examples

```
## Not run:
## Probability for Cat object of the ltm model
data(npi)
ltm_cat <- ltmCat(npi)
probability(ltm_cat, theta = 1, item = 1)

## Probability for Cat object of the tpm model
data(polknow)
tpm_cat <- tpmCat(polknow)
probability(tpm_cat, theta = 1, item = 1)

## Probability for Cat object of the grm model
data(nfc)
grm_cat <- grmCat(nfc)
probability(grm_cat, theta = 1, item = 1)

## End(Not run)
```

---

selectItem

*Select Next Item*

---

## Description

Selects the next item in the question set to be administered to respondent based on the specified selection method.

## Usage

```
selectItem(catObj)
```

## Arguments

catObj                      An object of class Cat

## Details

Selection approach is specified in the selection slot of the Cat object.

The minimum expected posterior variance criterion is used when the selection slot is "EPV". This method calls `expectedPV` for each unasked item.

The maximum Fisher's information criterion is used when the selection slot is "MFI". This method calls `fisherInf` for each unasked item.

The maximum likelihood weighted information criterion is used when the selection slot is "MLWI". This method involves integration. See **Note** for more information.

The maximum posterior weighted information criterion is used when the selection slot is "MPWI". This method involves integration. See **Note** for more information.

The maximum expected information criterion is used when the selection slot is "MEI". This method calls `expectedObsInf` for each unasked item.

The maximum Kullback-Leibler information criterion is used when the selection slot is "KL". This method calls `expectedKL` for each unasked item.

The maximum likelihood weighted Kullback-Leibler information criterion is used when the selection slot is "LKL". This method calls `likelihoodKL` for each unasked item.

The maximum posterior weighted Kullback-Leibler information criterion is used when the selection slot is "PKL". This method calls `posteriorKL` for each unasked item.

The "MFII" criterion is used when the selection slot is "MFII". This method involves integration. See **Note** for more information. The bounds of integration are  $\hat{\theta} \pm \delta$ , where  $\delta$  is  $z$  times the square root of the Fisher test information and  $z$  is specified in the `z` slot of the Cat object.

A random number generator is used when the selection slot is "RANDOM".

## Value

The function `selectItem` returns a list with two elements:

`estimates`: a `data.frame` with a row for each unasked question and three columns representing the item index number, the item name, and the item value (calculated by the specified selection method), and

`next_item`: a numeric representing the index of the item that should be asked next.

## Note

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

This function uses adaptive quadrature methods from the GNU Scientific Library (GSL) to approximate single-dimensional integrals with high accuracy. The bounds of integration are determined by the `lowerBound` and `upperBound` slots of the Cat object unless otherwise noted.

The "RANDOM" item selection criterion uses the package `RcppArmadillo` to randomly choose the next item among unasked questions. `RcppArmadillo` provides an exact reproduction of R's `sample` function that can be called from C++.

In the rare instance that item parameters are identical, it may be that that `selectItem` must choose between two items with the same value calculated by the selection criterion. In such an instance, `selectItem` will choose the item with the lower question index.

## Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

**See Also**

[estimateTheta](#), [expectedPV](#), [fisherInf](#)

**Examples**

```
## Not run:
## Create Cat object
data(npi)
ltm_cat <- ltmCat(npi)

## Store example answers
setAnswers(ltm_cat) <- c(1,0,1,0,1, rep(NA, 35))

## Set different selection criterion and choose next item
setSelection(ltm_cat) <- "EPV"
selectItem(ltm_cat)

setSelection(ltm_cat) <- "MFI"
selectItem(ltm_cat)

setSelection(ltm_cat) <- "MLWI"
selectItem(ltm_cat)

setSelection(ltm_cat) <- "MPWI"
selectItem(ltm_cat)

setSelection(ltm_cat) <- "MEI"
selectItem(ltm_cat)

setSelection(ltm_cat) <- "KL"
selectItem(ltm_cat)

setSelection(ltm_cat) <- "LKL"
selectItem(ltm_cat)

setSelection(ltm_cat) <- "PKL"
selectItem(ltm_cat)

setSelection(ltm_cat) <- "MFII"
selectItem(ltm_cat)

setSelection(ltm_cat) <- "RANDOM"
selectItem(ltm_cat)

## End(Not run)
```

---

setters

*Methods for Setting Value(s) to Cat Object Slots*

---

**Description**

Setter methods to control changes to the slots of a Cat object.

**Usage**

```
## S4 replacement method for signature 'Cat'
setGuessing(catObj) <- value

## S4 replacement method for signature 'Cat'
setDiscrimination(catObj) <- value

## S4 replacement method for signature 'Cat'
setDifficulty(catObj) <- value

## S4 replacement method for signature 'Cat'
setAnswers(catObj) <- value

## S4 replacement method for signature 'Cat'
setModel(catObj) <- value

## S4 replacement method for signature 'Cat'
setPriorName(catObj) <- value

## S4 replacement method for signature 'Cat'
setPriorParams(catObj) <- value

## S4 replacement method for signature 'Cat'
setLowerBound(catObj) <- value

## S4 replacement method for signature 'Cat'
setUpperBound(catObj) <- value

## S4 replacement method for signature 'Cat'
setEstimation(catObj) <- value

## S4 replacement method for signature 'Cat'
setEstimationDefault(catObj) <- value

## S4 replacement method for signature 'Cat'
setSelection(catObj) <- value

## S4 replacement method for signature 'Cat'
setZ(catObj) <- value

## S4 replacement method for signature 'Cat'
setLengthThreshold(catObj) <- value

## S4 replacement method for signature 'Cat'
setSeThreshold(catObj) <- value

## S4 replacement method for signature 'Cat'
setGainThreshold(catObj) <- value

## S4 replacement method for signature 'Cat'
setInfoThreshold(catObj) <- value
```

```
## S4 replacement method for signature 'Cat'
setLengthOverride(catObj) <- value
```

```
## S4 replacement method for signature 'Cat'
setGainOverride(catObj) <- value
```

### Arguments

catObj	An object of class Cat
value	The new value(s)

### Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

### See Also

[Cat-class](#), [getters](#)

### Examples

```
## Not run:
## Create Cat object
data(npi)
cat <- ltmCat(npi)

## Setting estimation slot
getEstimation(ltm_cat)
setEstimation(ltm_cat) <- "MAP"
getEstimation(ltm_cat)

## Setting discrimination slot
getDiscrimination(ltm_cat)
setDiscrimination(ltm_cat) <- rep(1, 40)
getDiscrimination(ltm_cat)

## End(Not run)
```

---

storeAnswer

*Update Answer to Single Item*

---

### Description

Stores answer to item  $k$  to the Cat object's answers slot.

### Usage

```
## S4 method for signature 'Cat'
storeAnswer(catObj, item, answer)
```

## Arguments

catObj	An object of class Cat
item	An integer indicating the index of the question item
answer	The answer to the item to be updated

## Details

The function `storeAnswer` updates the Cat object, but the updated object must be assigned to an object for the changes to be stored. See **Examples**.

## Value

The function `storeAnswer` returns an updated object of class Cat with the `answers` slot reflecting the newly stored answer to the indicated item. All previously stored answers remain the same, and all unanswered questions remain NA.

## Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

## Examples

```
## Not run:
## Create Cat object
data(npi)
ltm_cat <- ltmCat(npi)

## Printing current answers slot
ltm_cat@answers

## Storing answer of 0 to item 1
ltm_cat <- storeAnswer(ltm_cat, item = 1, answer = 0)

## Now object reflects answer to item 1
ltm_cat@answer

## End(Not run)
```

## Description

This function fits Birnbaum's three parameter model for binary data and populates the fitted values for discrimination, difficulty, and guessing parameters to an object of class Cat.

**Usage**

```
## S4 method for signature 'data.frame'
tpmCat(data, quadraturePoints = 21, ...)

## S4 method for signature 'tpm'
tpmCat(data, quadraturePoints = NULL, ...)
```

**Arguments**

<code>data</code>	A data frame of manifest variables or an object of class <code>tpm</code> .
<code>quadraturePoints</code>	A numeric to be passed into the <code>tpm</code> function indicating the number of Gauss-Hermite quadrature points. Only applicable when <code>data</code> is a data frame. Default value is 21.
<code>...</code>	arguments to be passed to methods. For more details about the arguments, see <code>tpm</code> in the <code>ltm</code> package.

**Details**

The `data` argument of the function `tpmCat` is either a data frame or an object of class `tpm` from the `ltm` package. If it is a data frame each row represents a respondent and each column represents a question item. If it is an object of the class `tpm`, it is output from the `tpm` function in the `ltm` package.

The `quadraturePoints` argument of the function `tpmCat` is used only when the `data` argument is a data frame. `quadraturePoints` is then passed to the `tpm` function from the `ltm` package when fitting Birnbaum's three parameter model to the data and is used when approximating the value of integrals.

**Value**

The function `tpmCat` returns an object of class `Cat` with changes to the following slots:

- `difficulty` A vector consisting of difficulty parameters for each item.
- `discrimination` A vector consisting of discrimination parameters for each item.
- `model` The string "tpm", indicating this `Cat` object corresponds to Birnbaum's three parameter model.

**Note**

In case the Hessian matrix at convergence is not positive definite try to use `start.val = "random"`.

**Author(s)**

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

**References**

Baker, Frank B. and Seock-Ho Kim. 2004. Item Response Theory: Parameter Estimation Techniques. New York: Marcel Dekker.



Birnbaum, Allan. 1968. Some Latent Trait Models and their Use in Inferring an Examinee's Ability. In F. M. Lord and M. R. Novick (Eds.), *Statistical Theories of Mental Test Scores*, 397-479. Reading, MA: Addison-Wesley.

Rizopoulos, Dimitris. 2006. "ltm: An R Package for Latent Variable Modeling and Item Response Theory Analyses." *Journal of Statistical Software* 17(5):1-25.

### See Also

[Cat-class](#), [ltmCat](#)

### Examples

```
## Not run:
## Creating Cat object with raw data
data(polknow)
tpm_cat1 <- tpmCat(polknow, quadraturePoints = 100)

## Creating Cat object with fitted object of class tpm
tpm_fit <- grm(polknow, control = list(GHk = 100)) ## from ltm package
class(tpm_fit)
tpm_cat2 <- tpmCat(tpm_fit)

## Note the two Cat objects are identical
identical(tpm_cat1, tpm_cat2)

## Note the slots that have changed from default values
tpm_cat1@model
tpm_cat1@difficulty
tpm_cat1@discrimination

## End(Not run)
```

# Index

## \*Topic **datasets**

nfc, [28](#)

npi, [29](#)

polknowMT, [32](#)

polknowTAPS, [36](#)

Cat-class, [2](#)

checkStopRules, [4](#), [4](#), [27](#)

d2LL, [5](#)

dLL, [6](#), [7](#)

estimateSE, [5](#), [8](#), [11](#), [13](#), [14](#)

estimateTheta, [4](#), [9](#), [10](#), [31](#), [44](#)

expectedKL, [11](#), [24](#), [38](#)

expectedObsInf, [12](#), [31](#)

expectedPV, [5](#), [13](#), [44](#)

fisherInf, [5](#), [14](#), [16](#), [44](#)

fisherTestInfo, [15](#), [16](#)

getAnswers (getters), [17](#)

getAnswers, Cat-method (getters), [17](#)

getDifficulty (getters), [17](#)

getDifficulty, Cat-method (getters), [17](#)

getDiscrimination (getters), [17](#)

getDiscrimination, Cat-method (getters),  
[17](#)

getEstimation (getters), [17](#)

getEstimation, Cat-method (getters), [17](#)

getEstimationDefault (getters), [17](#)

getEstimationDefault, Cat-method  
(getters), [17](#)

getGainOverride (getters), [17](#)

getGainOverride, Cat-method (getters), [17](#)

getGainThreshold (getters), [17](#)

getGainThreshold, Cat-method (getters),  
[17](#)

getGuessing (getters), [17](#)

getGuessing, Cat-method (getters), [17](#)

getInfoThreshold (getters), [17](#)

getInfoThreshold, Cat-method (getters),  
[17](#)

getLengthOverride (getters), [17](#)

getLengthOverride, Cat-method (getters),  
[17](#)

getLengthThreshold (getters), [17](#)

getLengthThreshold, Cat-method  
(getters), [17](#)

getLowerBound (getters), [17](#)

getLowerBound, Cat-method (getters), [17](#)

getModel (getters), [17](#)

getModel, Cat-method (getters), [17](#)

getPriorName (getters), [17](#)

getPriorName, Cat-method (getters), [17](#)

getPriorParams (getters), [17](#)

getPriorParams, Cat-method (getters), [17](#)

getSelection (getters), [17](#)

getSelection, Cat-method (getters), [17](#)

getSeThreshold (getters), [17](#)

getSeThreshold, Cat-method (getters), [17](#)

getters, [17](#), [46](#)

getUpperBound (getters), [17](#)

getUpperBound, Cat-method (getters), [17](#)

getZ (getters), [17](#)

getZ, Cat-method (getters), [17](#)

gpcm (gpcmCat), [19](#)

gpcmCat, [4](#), [19](#), [21](#)

gpcmCat, data.frame-method (gpcmCat), [19](#)

gpcmCat, gpcm-method (gpcmCat), [19](#)

grm (grmCat), [20](#)

grmCat, [4](#), [20](#), [20](#)

grmCat, data.frame-method (grmCat), [20](#)

grmCat, grm-method (grmCat), [20](#)

initialize, Cat-method (Cat-class), [2](#)

likelihood, [22](#)

likelihoodKL, [12](#), [23](#), [38](#)

lookAhead, [24](#)

ltm (ltmCat), [25](#)

ltmCat, [4](#), [25](#), [49](#)

ltmCat, data.frame-method (ltmCat), [25](#)

ltmCat, ltm-method (ltmCat), [25](#)

makeTree, [27](#)

nfc, [28](#)

- `np`, 29
- `obsInf`, 13, 15, 31
- `polknowMT`, 32
- `polknowTAPS`, 36
- `posteriorKL`, 12, 24, 37
- `prior`, 6, 8, 39
- `probability`, 13, 14, 22, 40
- `selectItem`, 4, 12–15, 24, 25, 27, 38, 42
- `setAnswers<-` (setters), 44
- `setAnswers<-`, Cat-method (setters), 44
- `setDifficulty<-` (setters), 44
- `setDifficulty<-`, Cat-method (setters), 44
- `setDiscrimination<-` (setters), 44
- `setDiscrimination<-`, Cat-method (setters), 44
- `setEstimation<-` (setters), 44
- `setEstimation<-`, Cat-method (setters), 44
- `setEstimationDefault<-` (setters), 44
- `setEstimationDefault<-`, Cat-method (setters), 44
- `setGainOverride<-` (setters), 44
- `setGainOverride<-`, Cat-method (setters), 44
- `setGainThreshold<-` (setters), 44
- `setGainThreshold<-`, Cat-method (setters), 44
- `setGuessing<-` (setters), 44
- `setGuessing<-`, Cat-method (setters), 44
- `setInfoThreshold<-` (setters), 44
- `setInfoThreshold<-`, Cat-method (setters), 44
- `setLengthOverride<-` (setters), 44
- `setLengthOverride<-`, Cat-method (setters), 44
- `setLengthThreshold<-` (setters), 44
- `setLengthThreshold<-`, Cat-method (setters), 44
- `setLowerBound<-` (setters), 44
- `setLowerBound<-`, Cat-method (setters), 44
- `setModel<-` (setters), 44
- `setModel<-`, Cat-method (setters), 44
- `setPriorName<-` (setters), 44
- `setPriorName<-`, Cat-method (setters), 44
- `setPriorParams<-` (setters), 44
- `setPriorParams<-`, Cat-method (setters), 44
- `setSelection<-` (setters), 44
- `setSelection<-`, Cat-method (setters), 44
- `setSeThreshold<-` (setters), 44
- `setters`, 18, 44
- `setUpperBound<-` (setters), 44
- `setUpperBound<-`, Cat-method (setters), 44
- `setZ<-` (setters), 44
- `setZ<-`, Cat-method (setters), 44
- `storeAnswer`, 46
- `storeAnswer`, Cat-method (storeAnswer), 46
- `tpm` (`tpmCat`), 47
- `tpmCat`, 4, 26, 47
- `tpmCat`, data.frame-method (`tpmCat`), 47
- `tpmCat`, `tpm`-method (`tpmCat`), 47