**Fisher Information: Calculated Expected and Observed Fisher Information for IRT**

FI calculates expected and/or observed Fisher information for various IRT models given a vector of ability values, a vector/matrix of item parameters, and an IRT model. It also calculates test information and expected/observed standard error of measurement.

The function FI returns item information, test information, and standard error of measurement for the binary response model ("brm") or the graded response model ("grm"). If the log likelihood is twice differentiable, expected Fisher information is the negative, expected, second derivative of the log likelihood with respect to the parameter. For the binary response model, expected item information simplifies to the following:

*I(θ\_i | a\_j, b\_j, c\_j) = (p\_{ij}^{\prime})^2/(p\_{ij}(1 - p\_{ij}))*

where *p\_{ij}^{\prime}* is the partial derivative of *p\_{ij}* with respect to *θ*, and *p\_{ij}* is the probability of response, as indicated in the help page for simIrt.

For the graded response model, expected item information simplifies to the following:

*I(θ\_i | a\_j, b\_{j1}, …, b\_{j(k - 1)}) = ∑\_k[(P\_{ijk}^{\prime})^2/P\_{ijk}]*

where *P\_{ijk}^{\prime}* is the partial derivative of *P\_{ijk}* with respect to *θ*, and *P\_{ijk}* is the probability of responding in category k as indicated in the help page for simIrt.

Binary Response Model

step 1

[set.seed](https://rdrr.io/r/base/Random.html)(888)

# generating random theta:

theta <- [rnorm](https://rdrr.io/r/stats/Normal.html)(20)

# generating an item bank under a 2-parameter binary response model:

b.params <- [cbind](https://rdrr.io/r/base/cbind.html)(a = [runif](https://rdrr.io/r/stats/Uniform.html)(100, .5, 1.5), b = [rnorm](https://rdrr.io/r/stats/Normal.html)(100, 0, 2), [c](https://rdrr.io/r/base/c.html) = 0)

# simulating responses using random theta:

b.mod <- [simIrt](https://rdrr.io/cran/catIrt/man/simIrt.html)(params = b.params, theta = theta, mod = "brm")

Binary response model simulation:

20 simulees, 100 items

# indicate class of params or extract it from simIrt object:

[class](https://rdrr.io/r/base/class.html)(b.params) <- "brm"

# calculating expected and observed information:

e.info <- [FI](https://rdrr.io/cran/catIrt/man/FI.html)(params = b.params, theta = theta, [type](https://rdrr.io/r/base/typeof.html) = "expected")

o.info <- [FI](https://rdrr.io/cran/catIrt/man/FI.html)(params = b.params, theta = theta, [type](https://rdrr.io/r/base/typeof.html) = "observed", resp = b.mod$resp)

# 2-parameter model, so expected informaion will be equal to observed information:

[all](https://rdrr.io/r/base/all.html)([signif](https://rdrr.io/r/base/Round.html)(e.info$item) == [signif](https://rdrr.io/r/base/Round.html)(o.info$item))

[1] TRUE

## step 2 ##

# generating an item bank under a 3-parameter binary response model:

b.params2 <- [cbind](https://rdrr.io/r/base/cbind.html)(a = [runif](https://rdrr.io/r/stats/Uniform.html)(100, .5, 1.5), b = [rnorm](https://rdrr.io/r/stats/Normal.html)(100, 0, 2), [c](https://rdrr.io/r/base/c.html) = .2)

# simulating responses using pre-specified thetas:

b.mod2 <- [simIrt](https://rdrr.io/cran/catIrt/man/simIrt.html)(params = b.params2, mod = "brm")

Binary response model simulation:

61 simulees, 100 items

# calculating expected and observed information:

# (if you don't indicate class, you can extract from simIrt object)

e.info2 <- [FI](https://rdrr.io/cran/catIrt/man/FI.html)(params = b.params2, theta = b.mod2$theta, [type](https://rdrr.io/r/base/typeof.html) = "expected")

o.info2 <- [FI](https://rdrr.io/cran/catIrt/man/FI.html)(params = b.params2, theta = b.mod2$theta, [type](https://rdrr.io/r/base/typeof.html) = "observed",

resp = b.mod2$resp)

# 3-parameter model, so expected information will not be equal to observed information:

[all](https://rdrr.io/r/base/all.html)([signif](https://rdrr.io/r/base/Round.html)(e.info2$item) == [signif](https://rdrr.io/r/base/Round.html)(o.info2$item))

[1] FALSE

## step 3 ##

# if theta is a scalar, item will be a vector and test will be a scalar:

e.info3 <- [FI](https://rdrr.io/cran/catIrt/man/FI.html)(params = b.params2, theta = 0, [type](https://rdrr.io/r/base/typeof.html) = "expected")

[dim](https://rdrr.io/r/base/dim.html)(e.info3$item) # no dimension because it's a vector

NULL

[length](https://rdrr.io/r/base/length.html)(e.info3$item) # of length equal to the number of items

[1] 100

# if params is a vector, item will be a matrix with one row:

e.info4 <- [FI](https://rdrr.io/cran/catIrt/man/FI.html)(params = [c](https://rdrr.io/r/base/c.html)(1, 2, 0), theta = [c](https://rdrr.io/r/base/c.html)(1, 2), [type](https://rdrr.io/r/base/typeof.html) = "expected")

[dim](https://rdrr.io/r/base/dim.html)(e.info4$item)

NULL

Graded Response Model

[set.seed](https://rdrr.io/r/base/Random.html)(999)

# generating random theta

theta <- [rnorm](https://rdrr.io/r/stats/Normal.html)(10)

# generating an item bank under a graded response model:

g.params <- [cbind](https://rdrr.io/r/base/cbind.html)(a = [runif](https://rdrr.io/r/stats/Uniform.html)(30, .5, 1.5), b1 = [rnorm](https://rdrr.io/r/stats/Normal.html)(30), b2 = [rnorm](https://rdrr.io/r/stats/Normal.html)(30),

b3 = [rnorm](https://rdrr.io/r/stats/Normal.html)(30), b4 = [rnorm](https://rdrr.io/r/stats/Normal.html)(30))

# sorting the parameters:

g.params <- [cbind](https://rdrr.io/r/base/cbind.html)(g.params[[](https://rdrr.io/r/base/Extract.html) , 1],

[t](https://rdrr.io/r/base/t.html)([apply](https://rdrr.io/r/base/apply.html)(g.params[[](https://rdrr.io/r/base/Extract.html) ,2:[dim](https://rdrr.io/r/base/dim.html)(g.params)[2]], MARGIN = 1,

FUN = [sort](https://rdrr.io/r/base/sort.html))))

# simulating responses using random theta:

g.mod <- [simIrt](https://rdrr.io/cran/catIrt/man/simIrt.html)(params = g.params, theta = theta, mod = "grm")

Graded response model simulation:

10 simulees, 30 items, 5 levels per item

# calculating expected and observed information:

[class](https://rdrr.io/r/base/class.html)(g.params) <- "grm" # always indicate model or extract from simulation.

e.info5 <- [FI](https://rdrr.io/cran/catIrt/man/FI.html)(params = g.params, theta = theta, [type](https://rdrr.io/r/base/typeof.html) = "expected")

o.info5 <- [FI](https://rdrr.io/cran/catIrt/man/FI.html)(params = g.params, theta = theta, [type](https://rdrr.io/r/base/typeof.html) = "observed", resp = g.mod$resp)

# grm, so expected informaion will not be equal to observed informaion:

[all](https://rdrr.io/r/base/all.html)([signif](https://rdrr.io/r/base/Round.html)(e.info5$item) == [signif](https://rdrr.io/r/base/Round.html)(o.info5$item))

[1] FALSE

# if theta is a vector and params is a vector, item will be a J x N matrix:

[dim](https://rdrr.io/r/base/dim.html)(e.info5$item)

[1] 10 30