Psychometrics

Part 1: Item Response Theory

SALVADOR CASTRO

Contents

1 Item Response Theory (IRT)

Item response theory (IRT) is the study of design, analysis, and scoring of tests, questionnaires, and other instruments measuring abilities or attitudes. IRT derives the maximum likelihood of each response as a function of the latent trait (i.e. ability) and some item parameterssuch as difficulty, discrimination and pseudo-chance. IRT is based on the idea that the probability of a correct response to an item is a mathematical function of person and item parameters. The person parameter is construed as a single latent trait. IRT is based on three assumptions:

A unidimensional trait denoted by (θ);

Local independence of items;

The response of a person to an item can be modeled by a mathematical item response function (IRF):

$$p(x_{(i,\,j)} = 1|\Theta_i,\ \alpha_j,\ \delta_j,\ \chi_j) = \chi_j + (1 - \chi_j)\frac{e^{\alpha_j(\Theta_i - \delta_j)}}{1 + e^{\alpha_j(\Theta_i - \delta_j)}}$$

where

- $p(x_{(i,\,j)} = 1|\Theta, : \delta_j)$ is person i's probability of the correct response to item j (i.e. $x_{(i,\,j)} = 1$),
- Θ_i is the person i's latent trait (ability), or person location,
- δ_j is the item j's (difficulty) location,
- α is the item j's discrimination (slope, or scale),
- χ is the pseudo-guessing parameter (lower bound, asymptote).

Preamble

```
# clear memory
rm(list = ls())
ls() # check memory
```

```
## character(0)
```

```r
# set the default working directory
setwd("/Users/salvadorcastro/Desktop/R Files/IRT/Part1")

# the number of digits to print
options(digits = 5)

# turn off warning messages
options(warn = -1)
```

1.1 Logistic Function and Item Characteristic Curves (ICC)

An item characteristic curve (ICC), (also known as Item Response Function (irf), is a sigmoid curve is a type of logistic function, which describes the relationship between a latent variable such as ability and the performance on a test or other measurement instruments. At the lowest levels of ability, the probability of correct response is near zero which increases with the levels of ability until the probability of correct response approaches 1.

"Item Characteristic Curves" from the Wolfram Demonstrations Project

Function: Given item (discrimination, difficulty and pseudo-guessing) and person location (ability) parameters, computes the probability of a correct response for the logistic function

```r
# The Logistic Function:
logistic <-
  function(alpha = NULL, delta, chi = NULL, theta = NULL){

    if (is.null(theta)) {
      # the latent trait continuum (theta)
      theta <- seq(from = -4, to = 4, by = 0.01)
    }

    if (is.null(alpha)) {
      # item discrimination
      alpha <- 1
    }

    if (is.null(chi)) {
```

```
      # guessing parameter
      chi <- 0
    }

    return(chi + ((1 - chi) * exp(alpha*(theta - delta))) /
             (1 + exp(alpha*(theta - delta))))
  } # end logistic

dump("logistic", file = "logistic.R")

# a person with average ability (theta = 0) has a 50% chance of
# a correct response for an item with a discrimination index
# of 1.0 (alpha = 1.0) and average difficulty (delta = 0)
logistic(alpha = 1, delta = 0, theta = 0) # should return 0.5
```

```
## [1] 0.5
```

```
# Alternative Logistic Function:
logistic2 <-
  function(alpha, delta, chi, theta=NULL){
    if (!is.null(theta)) {theta <- theta}
    else {theta <- seq(from = -4, to = 4, by = 0.01)}
    return((exp(alpha*(theta - delta)) + chi)/(exp(alpha*(theta - delta)) + 1))
  }
```

Function: Given item (discrimination, difficulty and pseudo-guessing) and person location (ability) parameters, computes expected score and the probability of the correct response according to the logistic model for the latent trait continium [-4, 4] (if theta is not provided), and plots optional irf and expected score graphs.

```
IRF <-
  function(parameter.matrix,
           person.theta = NULL,
           irf.plot = FALSE,
           trf.plot = FALSE,
           trace = FALSE){
```

```
source("logistic.R")
source("shadowtext.R")

parameter.matrix <- as.matrix(parameter.matrix)
# number of items
numitems <- nrow(parameter.matrix)

# the latent trait continuum (theta)
theta <- seq(from = -4, to = 4, by = 0.01)

if (is.null(person.theta)) {
  # the latent trait continuum (theta)
  person.theta <-  0
}

# number of examinees
npersons <- length(person.theta)

# probability of a correct resonse
person.p <- matrix(rep(NA, npersons*numitems),
                   npersons, numitems, byrow = TRUE)

# for expected score
expected.score <- rep(NA, npersons)

# for test response function
trf <- 0

# a matrix of zeros to hold probabilities
# for each person (rows) and item (columns)
probs.matrix <- score.matrix <- as.matrix(theta)

for (p in 1:npersons) { # p for rows
  for (i in 1:numitems) { # i for columns
    # compute the probability of a correct response
```

```
        probs <-  logistic(alpha = parameter.matrix[i, 1],
                           delta = parameter.matrix[i, 2],
                           chi = parameter.matrix[i, 3])

        # probability of a correct response for each person and item
        probs.matrix <- cbind(probs.matrix, probs)
        person.p[p, i] <- logistic(alpha = parameter.matrix[i, 1],
                                   delta = parameter.matrix[i, 2],
                                   chi = parameter.matrix[i, 3],
                                   theta = person.theta[p])
        # person.p[p, i] <- probs.matrix[which(probs.matrix[,1] == person.theta[p]), i+1]

        # test response function
        trf <- trf + probs
      }# end for i

      # compute the expected score
      score.matrix <- cbind(score.matrix, trf*100/(numitems*npersons))
      expected.score[p] <- score.matrix[which(score.matrix[ , 1] == person.theta[p]), p+1]
    }# end for p

    # Plots
    #----------------------------------------------------------------
    if (irf.plot) {
      # set color scheme
      mycolors <- palette(rainbow(2))
      mycolors <- palette(rainbow(2))
      if (numitems > 2) {
        mycolors <- palette(rainbow(numitems))
        mycolors <- palette(rainbow(numitems))
      }

      # start plot
      plot.new()

      ## add extra space to right margin of plot within frame
```

```
par(mar = c(3.5, 4, 2, 3.5))

# frame
box()

# plot tile
title(main = "Item Response Function")

for (i in 1:numitems) {
  for (p in 1:npersons) {
    # item response function
    par(new = TRUE)

    plot(probs.matrix[ , 1], probs.matrix[ , i+1],
         axes = FALSE,
         xlab = "",
         ylab = "",
         type = "l",
         ylim = c(0, 1),
         col = mycolors[i],
         lwd = 4)

    # axes
    if (i == 1) {
      # x-axis
      axis(1, pretty(range(theta), diff(range(theta))),
           las = 1,
           col = "black",
           col.axis = "black")

      # x label
      mtext(expression(paste("Ability (", Theta, ") / Item Difficulty (", delta, ")")
            side = 1,
            col = "black",
            line = 2.5)
```

```
          # left y-axis
          axis(2, pretty(c(0, 1), 10),
               las = 1,
               col = "black",
               col.axis = "black")

          # left y-axis label
          mtext(expression(paste("Probability of a correct response, p(", x[i, j], " = 1
                                 Theta[i], ", ", alpha[j],", ", delta[j], ", ", chi[j],")
                side = 2,
                col = "black",
                line = 2.5)
        }# end if i==1
      }# end for p
    }# end for i
  }# end if irf.plot

  # tracing lines
  if (trace) {
    for (i in 1:numitems) {
      for (p in 1:npersons) {
        # vertical segment
        segments(person.theta[p], -0.5,
                 person.theta[p], person.p[p, i],
                 lty = 2,
                 col = mycolors[i],
                 lwd = 1)

        # horizontal segment
        segments(person.theta[p], person.p[p, i],
                 min(theta)*1.1, person.p[p, i],
                 lty = 2,
                 col = mycolors[i],
                 lwd = 1)

        # probability of a correct response
```

```
          shadowtext(min(theta), person.p[p, i],
                     round(person.p[p, i], 3),
                     pos = 4,
                     cex = 1,
                     bg = mycolors[i])

          # coordinate point
          points(person.theta[p], person.p[p, i],
                 col = "white",
                 bg = mycolors[i],
                 pch = 21,
                 cex = .8)
        }# end if trace
      }# end for p
    }# end for i

    # trf plot
    if (trf.plot) {
      # test response function
      trf <- trf*100/numitems

      for (p in 1:npersons) {
        ## Allow a second plot on the same graph
        par(new = TRUE)

        plot(score.matrix[ , 1], score.matrix[ , p+1],
             col = "black",
             type = "l",
             lwd = 5,
             axes = FALSE,
             ylim = c(0, 100),
             xlab = "",
             ylab = "")

        ## right y-axis
        axis(4, pretty(c(0, 100), 10),
```

```
             col = "black",
             col.axis = "black",
             las = 1)

        ## right y-axis label
        mtext("Expected Score (%)",
              side = 4,
              col = "black",
              line = 2.5)

        # tracing lines for the trf
        # horizontal line
        segments(person.theta[p], expected.score[p],
                 max(theta)*1.1, expected.score[p],
                 lty = 3,
                 col = "black",
                 lwd = 1)
        # vertical line
        segments(person.theta[p], -5,
                 person.theta[p], expected.score[p],
                 lty = 3,
                 col = "black",
                 lwd = 1)
        # coordinates
        points(person.theta[p], expected.score[p],
               col = "white",
               bg = "black",
               pch = 21,
               cex = .7)

        shadowtext(max(theta), expected.score[p],
                   sprintf("%1.2f%%", round(expected.score[p], 3)),
                   pos = 2,
                   cex = 1,
                   bg = "black")
      }# end for p
```

```
    }# end if trf.plot

    return(list(probabilities = person.p,
                expected.score = expected.score))
  }# end IRF

dump("IRF", file = "IRF.R")
```

Plot item characteristic curve (ICC)

```
# item parameters
alpha <- 1 # Discrimination, scale, slope
delta <- 0 # Difficulty, item location
chi <- 0 # Pseudo-guessing, chance, asymptotic minimum
parameter.matrix <- cbind(alpha, delta, chi)

# the latent trait, ability (person location)
person.theta <- 0
```

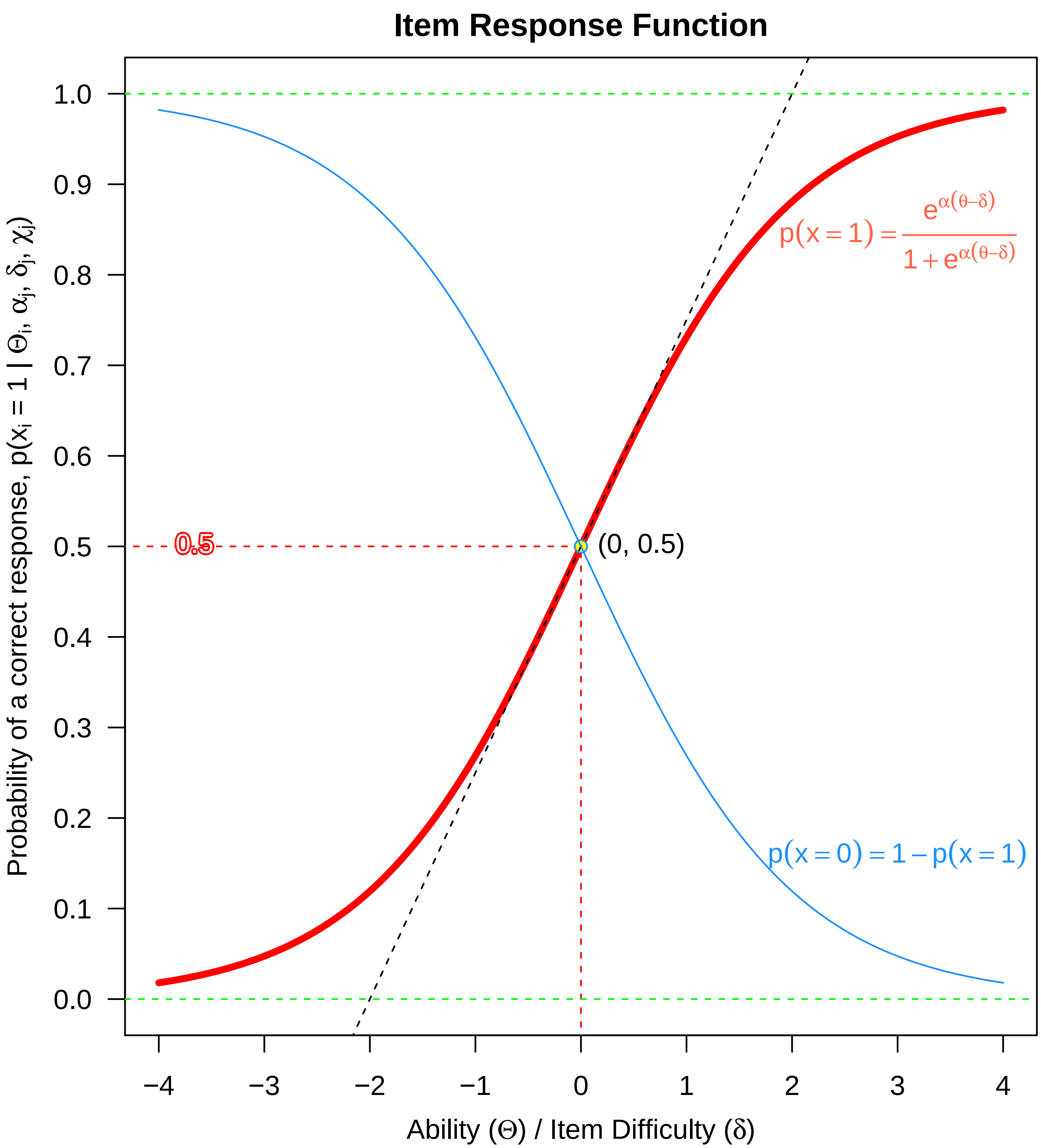
```
# graph the probability of correct response (red)
IRF(parameter.matrix = parameter.matrix,
    person.theta = person.theta,
    irf.plot = TRUE,
    trace = TRUE)
```

```
## $probabilities
##      [,1]
## [1,]  0.5
##
## $expected.score
## [1] 50
```

```
# display the irf equation on the graph
text(3, .85,
     expression(p(x == 1) == frac(e^{alpha(theta-delta)},
                                  1 + e^{alpha(theta-delta)})),
```

```
     col = "tomato")

# coordinates
points(delta, logistic(alpha, delta, chi, person.theta),
       col = "dodgerblue",
       bg = "yellow",
       pch = 21)

text(delta, logistic(alpha, delta, chi, person.theta),
     label = paste("(", person.theta, ", ",
                   logistic(alpha, delta, chi, person.theta), ")", sep = ""),
     col = "black",
     pos = 4)

# tangent line at delta = theta = 0, p = 0.5
# slope = discrimination = 0.25
abline(a = .5, b = .25, lty = 2)

# asymptotic minimum (pseudo-guessing) and maximum (carelessness)
abline(h = c(0, 1), col = "green", lty = 2)

# graph probability of incorrect response (blue)
q <- 1 - logistic(alpha, delta, chi)

# the latent trait continuum (theta)
theta <- seq(from = -4, to = 4, by = 0.01)

lines(theta, q,
      col = "dodgerblue",
      lwd = 1)

# display irf function on the graph
text(3, .16, expression(p(x == 0) == 1 - p(x == 1)), col = "dodgerblue")
```

Three item characteristic curves with the same difficulty ($\delta = 0.5$)**, but different dis-**



Item Response Function
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Probability of a correct response, $p(x_i = 1 \mid \Theta_i, \alpha_j, \delta_j, \chi_j)$
$p(x = 1) = \frac{e^{\alpha(\theta - \delta)}}{1 + e^{\alpha(\theta - \delta)}}$
0.5
(0, 0.5)
$p(x = 0) = 1 - p(x = 1)$
−4
−3
−2
−1
0
1
2
3
4
Ability (Θ) / Item Difficulty (δ)

Figure 1:

criminations (0.7, 1.0, 1.5)

```
# item parameters
alpha <- c(.7, 1.0, 1.5)
delta <- c(0, 0, 0)
chi <- c(0, 0, 0)

parameter.matrix <- cbind(alpha, delta, chi)

# the latent trait, ability (person location)
person.theta <- 0

IRF(parameter.matrix, person.theta,
    irf.plot = TRUE,
    trace = TRUE)
```

```
## $probabilities
##      [,1] [,2] [,3]
## [1,]  0.5  0.5  0.5
##
## $expected.score
## [1] 50
```

```
legend("bottomright",
       legend = sapply(alpha, function(x){as.expression(substitute(alpha == a, list(a = x)
       col = palette(rainbow(3)),
       bty = "n",
       text.col = palette(rainbow(3)),
       lwd = 4)
```

Three item characteristic curves with different difficulties (-1, 0, 1) but the same discrimination (1.0)

```
# item parameters
alpha <- c(1.0, 1.0, 1.0)
delta <- c(-1, 0, 1)
```

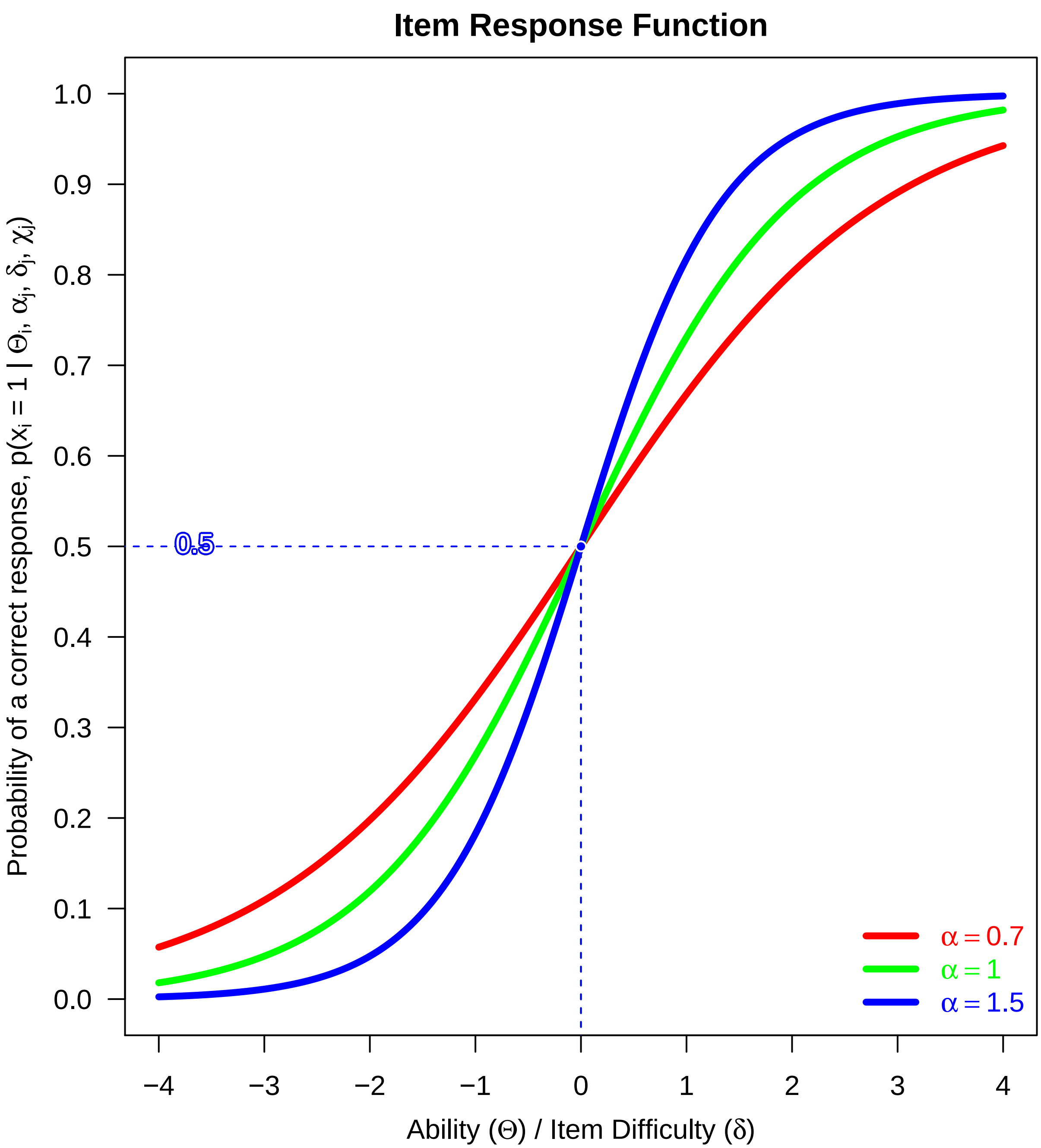

Item Response Function
Probability of a correct response, p(x_i = 1 | Θ_i, α_j, δ_j, χ_j)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.5
α = 0.7
α = 1
α = 1.5
−4
−3
−2
−1
0
1
2
3
4
Ability (Θ) / Item Difficulty (δ)

Figure 2:

```r
chi <- c(0, 0, 0)
parameter.matrix <- cbind(alpha, delta, chi)

# the latent trait, ability (person location)
person.theta <- 0

IRF(parameter.matrix, person.theta, irf.plot = TRUE, trace = TRUE)
```

```
## $probabilities
##          [,1] [,2]    [,3]
## [1,] 0.73106  0.5 0.26894
##
## $expected.score
## [1] 50
```

```r
# legend
legend("bottomright",
       legend = sapply(delta, function(x){as.expression(substitute(delta == a, list(a = x)
       col = palette(rainbow(3)),
       bty = "n",
       text.col = palette(rainbow(3)),
       lwd = 4,
       cex = 1)
```

Different persons' (with varying abilities: -2, -1, 0, 1, 2) probability of a correct response to a single item (parameters: discrimination = 1, difficulty = 0, guessing = 0)

```r
# item parameters
alpha <- 1 # Discrimination, scale, slope
delta <- 0 # Difficulty, item location
chi <- 0 # Pseudo-guessing, chance, asymptotic minimum
parameter.matrix <- cbind(alpha, delta, chi)

# person abilities
```

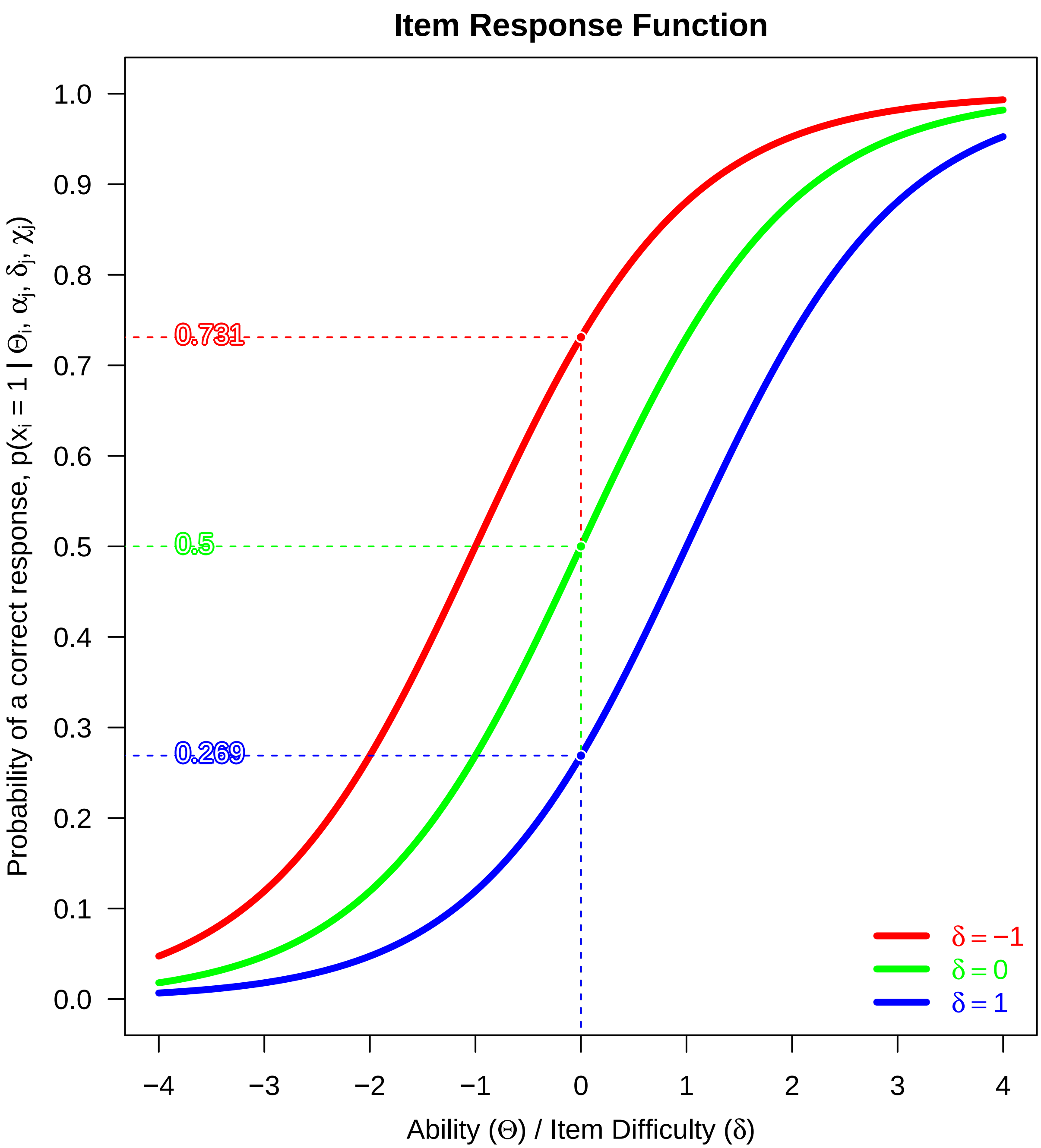

Item Response Function
Probability of a correct response, p(x_i = 1 | Θ_i, α_j, δ_j, χ_j)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.731
0.5
0.269
−4
−3
−2
−1
0
1
2
3
4
Ability (Θ) / Item Difficulty (δ)
δ = −1
δ = 0
δ = 1

Figure 3:

```r
person.theta <- -2:2

IRF(parameter.matrix,
    person.theta = person.theta,
    irf.plot = TRUE,
    trace = TRUE)
```

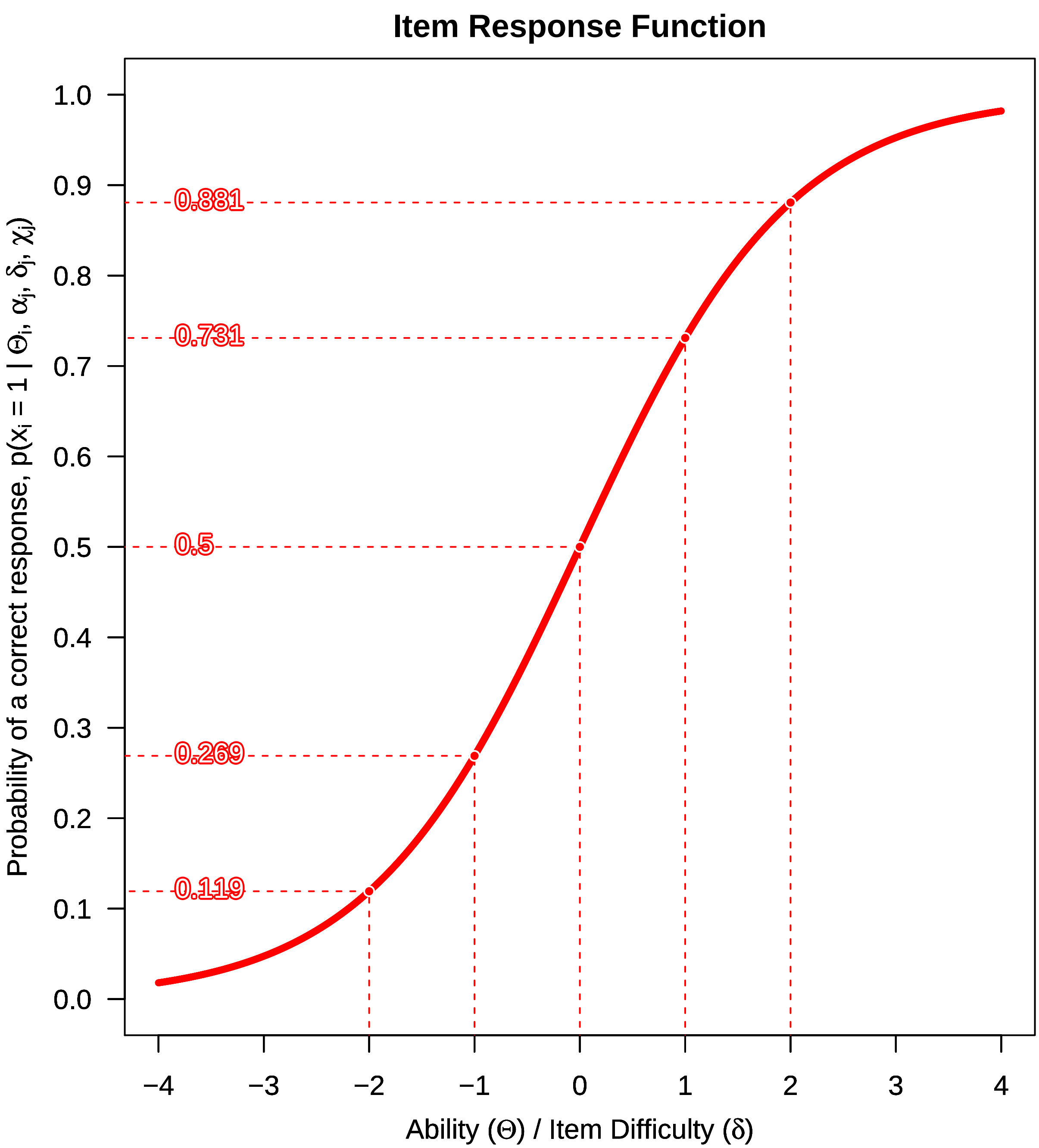


Figure 4:

```
## $probabilities
```

```
##          [,1]
## [1,] 0.11920
## [2,] 0.26894
## [3,] 0.50000
## [4,] 0.73106
## [5,] 0.88080
##
## $expected.score
## [1]  2.3841 10.7577 30.0000 58.4847 88.0797
```

1.1.1 Normal Ogive vs Logistic Curves (sigmoid) and the Scaling Constant, 1.702

Origin of the scaling constant d = 1.7, in item response theory (Camilli 1994):

The relationship between a latent ability (Θ) *and the probability of a correct response* (P) *on a test item is modelled by an item characteristic curve. This Demonstration plots the item characteristic curve of a single dichotomous item under two different models: the normal ogive model and the logistic model. The parameters a, b, and c represent item properties related to discrimination, difficulty, and guessing. The constant is used to scale the logistic curve. Notice that the two curves are nearly identical when* $D = 1.702$.

The scaling constant, D, used in item response theory minimizes the maximum difference between the normal and logistic distribution functions. The scaling constant, D, makes the logistic model's values similar to those of the normal ogive model (1 logit = 1.702 probits).

Demo: Comparing the Normal Ogive and Logistic Item Characteristic Curves

```
# Parameters
alpha <- 1.0 # Discrimination, scale, slope
delta <- 0.0 # Difficulty, item location
chi <- 0.0 # Pseudo-guessing, chance, asymptotic minimum
D <- 1.70174439 # scaling constant
# the latent trait continuum (theta), person location
theta <- seq(from = -4, to = 4, by = 0.1)
```

A matrix of theta and logistic theta values

```
logistic.theta <- cbind(theta, logistic(alpha, delta, chi, theta))
```

The logistic function with the scaling constant, D = 1.702

```
##  The 3PL Logistic Model with scaling:
logisticD <- function(alpha, delta, chi = NULL, theta = NULL){
  # both chi and theta are provided
  if (!is.null(chi) && !is.null(theta) ) {
    # chi is between 0 and 1
    if (!(chi > 1 || chi < 0)) {
      chi <- chi
      theta <- theta
    }
    # chi is not between 0 and 1
    else{
      return("chi must be between 0 and 1")
    }
  }
  # chi is provided but theta is NULL
  else if (!is.null(chi) && is.null(theta)) {
    if (!(chi > 1 || chi < 0)) {
      chi <- chi
      # the latent trait continuum (theta)
      theta <- seq(from = -4, to = 4, by = 0.1)
    }else{
      return("chi must be between 0 and 1")
    }
  }
  # chi is not provided but theta is provided
  else if (is.null(chi) && !is.null(theta)) {
    chi <- 0
    theta <- theta
  }
  # neither chi nor theta is provided
  else if (is.null(chi) && is.null(theta)) {
```

```r
    chi <- 0
    # the latent trait continuum (theta)
    theta <- seq(from = -4, to = 4, by = 0.1)
  }

  D <- 1.702
  return((exp(D*alpha*(theta - delta)) + chi) /
           (exp(D*alpha*(theta - delta)) + 1))
} # end logisticD

dump("logisticD", file = "logisticD.R")
```

A matrix of theta and the logistic theta values with scaling constant

```r
ld <- cbind(theta, logisticD(alpha, delta))
```

The Normal Ogive Probability Density Function

```r
# The probability density function (PDF) of standard normal distribution
normal.pdf <-
  function(x){
    return(exp(-(x^2/2))/sqrt(2*pi))
  }

no.pdf <- cbind(theta, normal.pdf(theta))
```

The Normal Ogive Cumulative Density Function

```r
# The cumulative distribution function (CDF) of the standard normal distribution
normal.cdf <-
  function(x){
    z <- seq(from = -4, to = x , by = 0.01)
    return(round((sum(normal.pdf(z))/100), 4))
  }
# The cumulative distribution function (CDF) of the standard normal distribution
# to hold CDF values of the latent trait continuum
no.cdf <- cbind(theta, rep(0, length(theta)))
```

```
# compute CDF for the latent trait continuum
for (i in 1:length(theta)) {
  no.cdf[i,2] <- normal.cdf(theta[i])
}
```

Check: For the normal distribution, the values less than one standard deviation away from the mean account for 68.27% of the set

```
sprintf("%.3f%%", 100*(normal.cdf(0)))
```

```
## [1] "50.200%"
```

```
sprintf("%.3f%%", 100*(normal.cdf(1) - normal.cdf(-1)))
```

```
## [1] "68.270%"
```

```
# two standard deviations from the mean account for 95.45%
sprintf("%.3f%%", 100*(normal.cdf(2) - normal.cdf(-2)))
```

```
## [1] "95.450%"
```

```
# three standard deviations account for 99.73%
sprintf("%.3f%%", 100*(normal.cdf(3) - normal.cdf(-3)))
```

```
## [1] "99.730%"
```

```
# four standard deviations account for 99.99%
sprintf("%.3f%%", 100*(normal.cdf(4) - normal.cdf(-4)))
```

```
## [1] "99.990%"
```

Plots

```r
# add extra space to right margin of plot within frame
par(mar = c(5, 4, 4, 0) + 0.1)

# The 3PL logistic curve
plot(logistic.theta,
     type = "l",
     lwd = 5,
     lty = 3,
     col = "tomato",
     axes = FALSE,
     ylim = c(0, 1),
     xlab = "",
     ylab = "",
     main = "Normal Ogive vs Logistic Curves\nwith different scaling constants")

# x-axis
axis(1, pretty(c(-4, 4), 8))

mtext(expression(paste("Ability, ", Theta)),
      side = 1,
      col = "black",
      line = 2.5)

# y-axis
axis(2, pretty(c(0, 1), 10),
     las = 1,
     col = "black",
     col.axis = "black")

mtext(expression(paste("Probability of a correct response, P(",
                       Theta,"|", alpha,",", delta, ",", chi,")")),
      side = 2,
      col = "black",
      line = 2.5)
box()
```

```
# Allow a second plot on the same graph
par(new = TRUE)

## plot logistic curve with scaling constant D
plot(ld,
     type = "l",
     col = "tomato",
     lwd = 5,
     axes = FALSE,
     ylim = c(0, 1),
     xlab = "",
     ylab = "")

## Allow a second plot on the same graph
par(new = TRUE)

# plot CDF
plot(no.cdf,
     type = "l",
     axes = FALSE,
     ylim = c(0, 1),
     lwd = 5,
     col = "dodgerblue",
     xlab = "",
     ylab = "")

# gridlines
grid(nx = NULL, ny = NULL, col = "lightgray", lty = "dotted", lwd = 1)

# legend
mycolors <- c("tomato", "tomato", "dodgerblue")
legend("topleft",
       c("Logistic Function without scaling, D=1.0",
         "Logistic with scaling constant, D=1.702",
         "Normal Ogive (CDF)"),
       col = mycolors,
```

```
       bty = "n",
       text.col = mycolors,
       lty = c(3, 1, 1),
       lwd = 5,
       cex = 1)
```

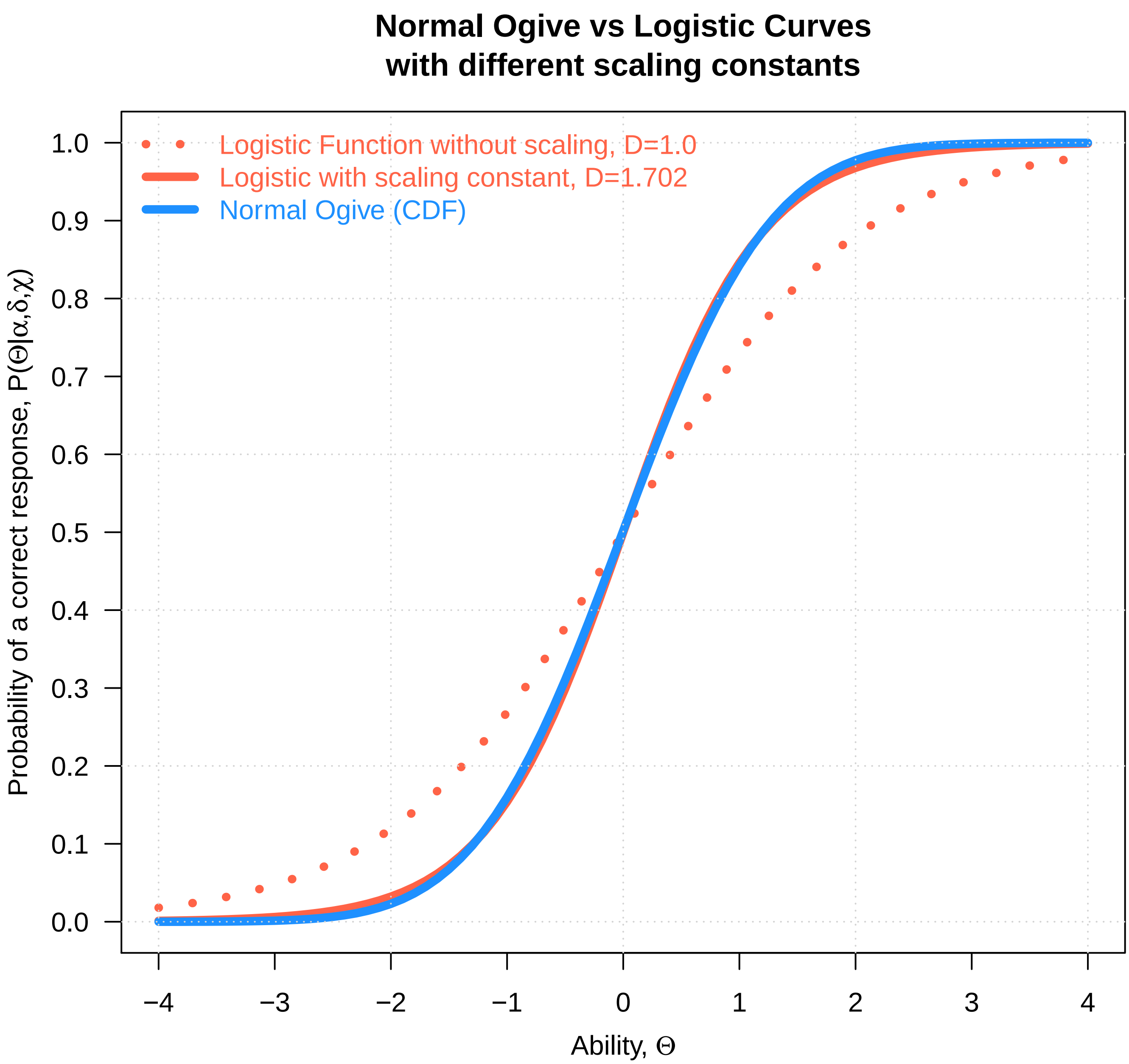


Figure 5:

1.1.2 Illustrations of Item Characteristic Curves with Various Parameters

Draw illustrations of Item Characteristic Curves (ICC) or Item Response Functions (IRF) for each of the following types of items. Label both horizontal and vertical axes and locate all item parameters on the graph.

- A slightly difficult Rasch item
- A moderately difficult 2-PL item with positive, but very low discrimination
- A 3-PL item that is slightly guessable, very discriminating, and very easy
- A difficult 2-PL item that discriminates negatively…perhaps a miscoded or miskeyed item

A slightly difficult Rasch item

The probability of a correct response to a Rasch item is

$$p(x_{(i,\,j)} = 1 \mid \Theta_i,\ \delta_j) = \frac{e^{(\Theta_i - \delta_j)}}{1 + e^{(\Theta_i - \delta_j)}}$$

where

- $p(x_{(i,\,j)} = 1 \mid \Theta_i,\ \delta_j)$ is the probability of the correct response (i.e. $x_{(i,\,j)} = 1$),
- Θ_i is the person i's location,
- δ_j is the item j's location, and when $\Theta = 0$ and $\delta = 0.5$, we have

$$p(x_{(i,\,j)} \mid 0, 0.5) = \frac{e^{(0-0.5)}}{1 + e^{(0-0.5)}}$$

simplifying,

$$p(x_{(i,\,j)} \mid 0, 0.5) = \frac{e^{-0.5}}{1 + e^{-0.5}}$$

Also the item information function for the Rasch model is

$$I_j(\Theta_i) = p_j(1 - p_j)$$

```r
# Parameters
alpha <- 1 # Discrimination, scale, slope
delta <- 0.5 # Difficulty, item location
chi <- 0 # Pseudo-guessing, chance, asymptotic minimum

parameter.matrix <- cbind(alpha, delta, chi)

# ability and difficulty
person.theta <- c(0, delta)

# plot
IRF(parameter.matrix,
    person.theta = person.theta,
    irf.plot = TRUE,
    trace = TRUE)
```

```
## $probabilities
##         [,1]
## [1,] 0.37754
## [2,] 0.50000
##
## $expected.score
## [1] 18.877 50.000
```

```r
# display irf function equation on the graph
text(3, .8, expression(p(theta) == frac(1, 1+ e^{-0.5})), col = "tomato")

# theta
text(0, 0,
     label = substitute(paste("ability: ", theta, " = ", a), list(a = person.theta)),
     srt = 90,
     col = "tomato",
     offset = -.5,
     pos = 4)

# slope at delta
```

```
y <- logistic(delta = delta, theta = delta)
points(delta+.2, y-.014,
       pch = 25,
       cex = 1.5,
       col = "dodgerblue",
       bg = "dodgerblue")

# delta
text(delta, 0,
     label = substitute(paste("item difficulty: ", delta, " = ", a), list(a = delta)),
     srt = 90,
     col = "tomato",
     offset = -.5,
     pos = 4)

# Discrimination, scale,
text(delta, y,
     label = substitute(paste("item discrimination (slope): ", alpha, " = ", a),
                        list(a = alpha)),
     srt = 64,
     pos = 3,
     col = "dodgerblue",
     cex = 1)

# gridlines
grid(nx = NULL, ny = NULL,
     col = "lightgray",
     lty = "dotted",
     lwd = 1)
```

A moderately difficult 2-PL item with positive, but very low discrimination

The probability of a correct response to a 2-PL dichotomous item is

$$p(x_{(i,j)} \mid \Theta_i, \alpha_j, \delta_j) = \frac{e^{\alpha_j(\Theta_i - \delta_j)}}{1 + e^{\alpha_j(\Theta_i - \delta_j)}}$$

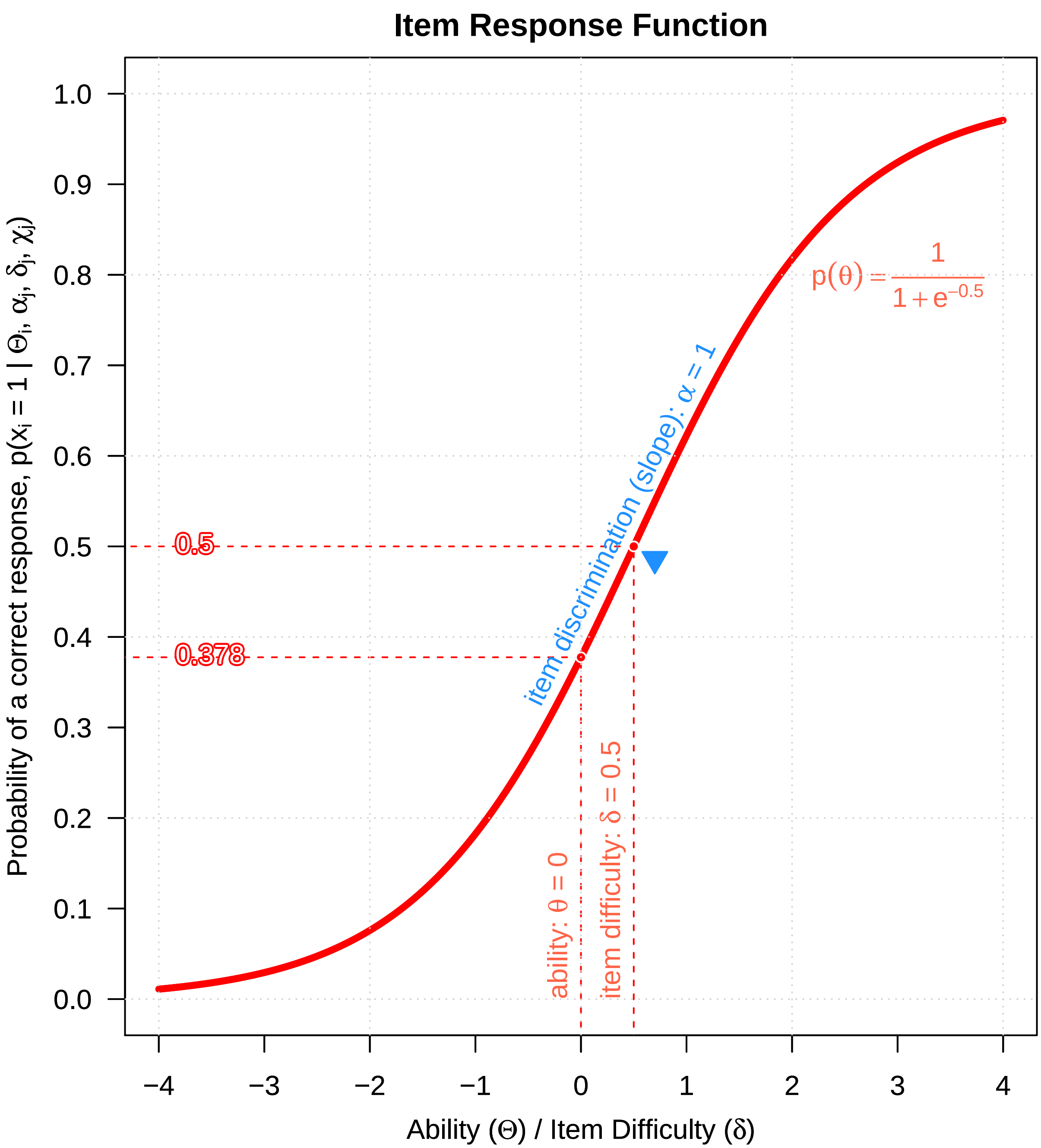

Item Response Function
Probability of a correct response, p(xi = 1 | Θi, αj, δj, χj)
Ability (Θ) / Item Difficulty (δ)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−4
−3
−2
−1
0
1
2
3
4
0.5
0.378
item discrimination (slope): α = 1
p(θ) = 1 / (1 + e^−0.5)
ability: θ = 0
item difficulty: δ = 0.5

Figure 6:

where

- $p(x_{(i,\,j)} = 1 \mid \Theta_i,\ \delta_j)$ is the probability of the correct response (i.e. $x_{(i,\,j)} = 1$),
- Θ_i is the person i's location (ability),
- δ_j is the item j's location (difficulty), and
- α_j is the item j's discrimination (slope, or scale)

when $\alpha = 0.3$, $\delta = 1$, we have

$$p(x_{(i,\,j)} \mid \Theta_i, 0.3, 1) = \frac{e^{0.3(\Theta_i - 1)}}{1 + e^{0.3(\Theta_i - 1)}}$$

Also the item information function for the 2PL model is

$$I_j(\Theta_i) = \alpha_j^2 p_j (1 - p_j)$$

therefore

$$I_j(\Theta_i) = 0.3^2 p_j (1 - p_j)$$

simplifying, we have

$$I_j(\Theta_i) = 0.09 p_j (1 - p_j)$$

```
# Parameters
alpha <- 0.3 # Discrimination, scale, slope
delta <- 1.0 # Difficulty, item location
chi <- 0.0 # Pseudo-guessing, chance, asymptotic minimum

parameter.matrix <- cbind(alpha, delta, chi)

# ability and difficulty
person.theta <- c(0, delta)

# Item response function plot by the logistic formula
IRF(parameter.matrix,
```

```
    person.theta = person.theta,
    irf.plot = TRUE,
    trace = TRUE)
```

```
## $probabilities
##          [,1]
## [1,] 0.42556
## [2,] 0.50000
##
## $expected.score
## [1] 21.278 50.000
```

```
# display irf function on the graph
text(2.5, .75,
     expression(p(theta) == frac(e^{0.3(theta - 1)}, 1 + e^{0.3(theta - 1)})),
     col = "tomato")

# theta
text(0, 0,
     label = substitute(paste("ability: ", theta, " = ", a),
                        list(a = person.theta)),
     srt = 90,
     col = "tomato",
     offset = -.5,
     pos = 4)

# delta
text(delta, 0,
     label = substitute(paste("item difficulty: ", delta, " = ", a),
                        list(a = delta)),
     srt = 90,
     col = "tomato",
     offset = -.5,
     pos = 4)

# Discrimination, scale,
```

```
text(delta, logistic(delta = delta, theta = delta),
     label = substitute(paste("item discrimination (slope): ", alpha, " = ", a),
                        list(a = alpha)),
     srt = 32,
     pos = 3,
     col = "dodgerblue",
     cex = 1)

# gridlines
grid(nx = NULL, ny = NULL,
     col = "lightgray",
     lty = "dotted",
     lwd = 1)
```

A 3-PL item that is slightly guessable, very discriminating, and very easy

The probability of a correct response to a dichotomous 3-PL item is

$$p(x_{(i,\,j)} = 1 \mid \Theta_i,\ \alpha_j,\ \delta_j,\ \chi_j) = \chi_j + (1-\chi_j)\frac{e^{\alpha_j(\Theta_i-\delta_j)}}{1+e^{\alpha_j(\Theta_i-\delta_j)}}$$

where

- $p(x_{(i,\,j)} = 1 \mid \Theta_i,\ \delta_j)$ is the probability of the correct response (i.e. $x_{(i,\,j)} = 1$),

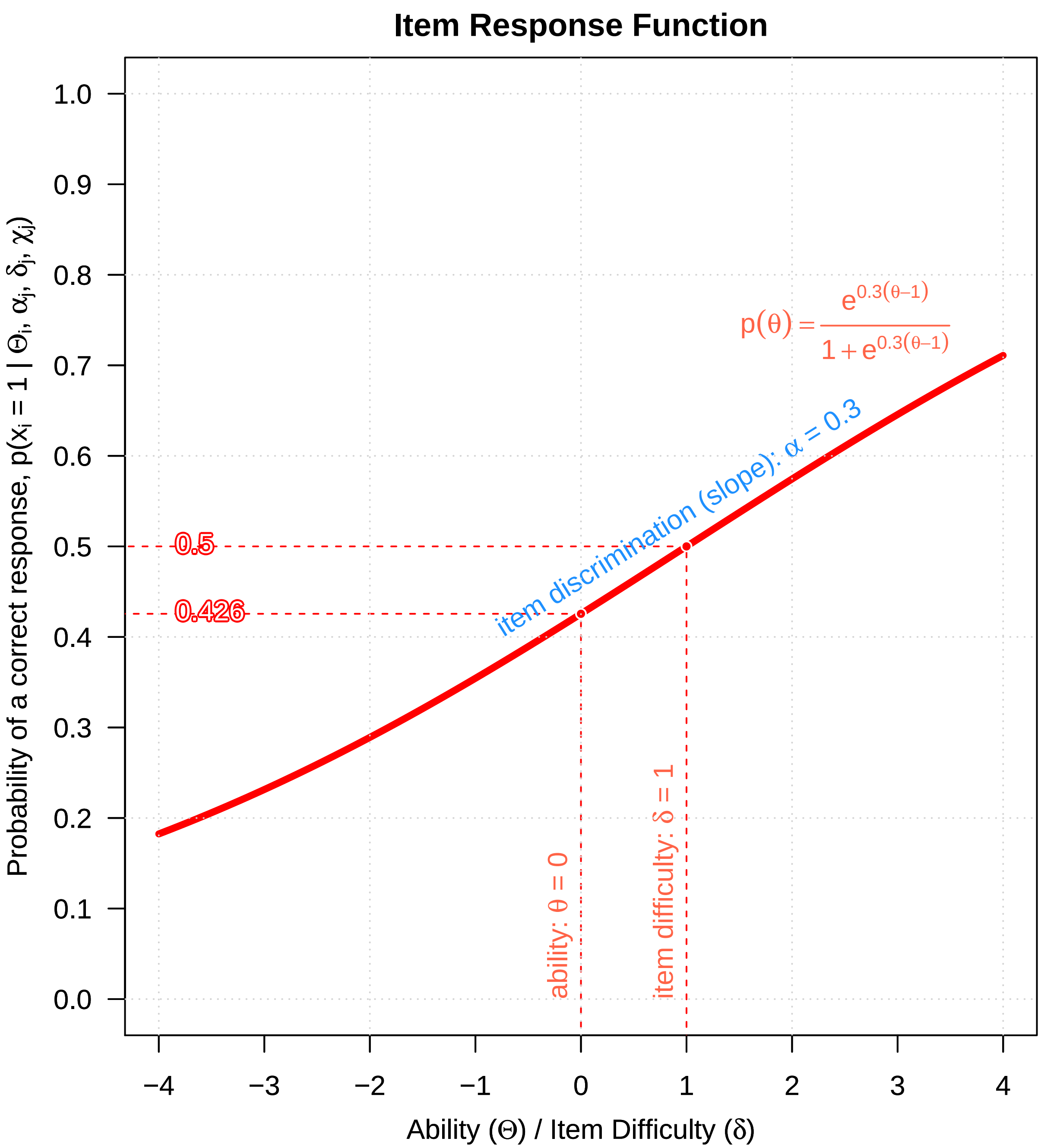
* Θ_i is the person i's location,

- δ_j is the item j's location,
- α is the item j's discrimination (slope, or scale), and
- χ is the pseudo-guessing parameter (lower bound, asymptote).

when $\alpha = 2$, $\delta =$ -1, $\chi = 0.15$, we have

$$p(x_{(i,\,j)} = 1 \mid \Theta_i,\ \delta_j) = 0.15 + (1-0.15)\frac{e^{2(\Theta_i+1)}}{1+e^{2(\Theta_i+1)}}$$

simplifying we get



Item Response Function
Probability of a correct response, p(x_i = 1 | Θ_i, α_j, δ_j, χ_j)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
p(θ) = e^{0.3(θ−1)} / (1 + e^{0.3(θ−1)})
item discrimination (slope): α = 0.3
0.5
0.426
ability: θ = 0
item difficulty: δ = 1
−4
−3
−2
−1
0
1
2
3
4
Ability (Θ) / Item Difficulty (δ)

Figure 7:

$$p(x_{(i,\,j)} = 1 \mid \Theta_i,\ \delta_j) = 0.15 + 0.85\frac{e^{2(\Theta_i+1)}}{1+e^{2(\Theta_i+1)}}$$

Also, the item information function for the 3PL model is

$$I_j(\Theta_i) = \alpha_j^2\left[\frac{(p_j-\chi_j)^2}{(1-\chi_j)^2}\right]\left[\frac{(1-p_j)}{p_j}\right]$$

thus

$$I_j(\Theta_i) = 2^2\left[\frac{(p_j-0.15)^2}{(1-0.15)^2}\right]\left[\frac{(1-p_j)}{p_j}\right]$$

simplifying, we have

$$I_j(\Theta_i) = 5.54\frac{(p_j-0.15)^2(1-p_j)}{p_j}$$

```
# Parameters
alpha <- 2.0 # Discrimination, scale, slope
delta <- -1.0 # Difficulty, item location
chi <- 0.15 # Pseudo-guessing, chance, asymptotic minimum

parameter.matrix <- cbind(alpha, delta, chi)

# ability and difficulty
person.theta <- c(0, delta)

# Item response function plot by the logistic formula
IRF(parameter.matrix,
    person.theta = person.theta,
    irf.plot = TRUE,
    trace = TRUE)
```

```
## $probabilities
##         [,1]
## [1,] 0.89868
## [2,] 0.57500
##
```

```
## $expected.score
## [1] 44.934 57.500
```

```
# display irf function equation on the graph
text(2.6, .9,
     expression(p(theta) == 0.15 + frac(0.85, 1 + e^{2(theta + 1)})),
     col = "tomato")

# theta
text(0, 0,
     label = substitute(paste("ability: ", theta, " = ", a),
                        list(a = person.theta)),
     srt = 90,
     col = "tomato",
     offset = -.5,
     pos = 4)

# delta
text(delta, 0,
     label = substitute(paste("item difficulty: ", delta, " = ", a),
                        list(a = delta)),
     srt = 90,
     col = "tomato",
     offset = -.5,
     pos = 4)

# Discrimination, scale
y <- logistic(alpha = alpha, delta = delta, chi = chi, theta = delta)
text(delta, y,
     label = substitute(paste("item discrimination (slope): ", alpha, " = ", a),
                        list(a = alpha)),
     srt = 74,
     pos = 3,
     col = "dodgerblue",
     cex = 1)

# Pseudo-guessing, chance, asymptotic minimum (c)
```

```
abline(h = chi, lty = 2, col = "green", lwd = 1)
text(2, chi,
     label = substitute(paste("pseudo-guessing: ", chi, " = ", a),
                        list(a = chi)),
     srt = 0,
     pos = 3,
     col = "green",
     cex = 1)

# gridlines
grid(nx = NULL, ny = NULL,
     col = "lightgray",
     lty = "dotted",
     lwd = 1)
```

A difficult 2-PL item that discriminates negatively, perhaps a miscoded or miskeyed item

```
# Parameters
alpha <- -1 # Discrimination, scale, slope
delta <- 2.0 # Difficulty, item location
chi <- 0 # Pseudo-guessing, chance, asymptotic minimum

parameter.matrix <- cbind(alpha, delta, chi)

# ability and difficulty
person.theta <- c(0, delta)

# Item response function plot by the logistic formula
IRF(parameter.matrix,
    person.theta = person.theta,
    irf.plot = TRUE,
    trace = TRUE)
```

```
## $probabilities
```

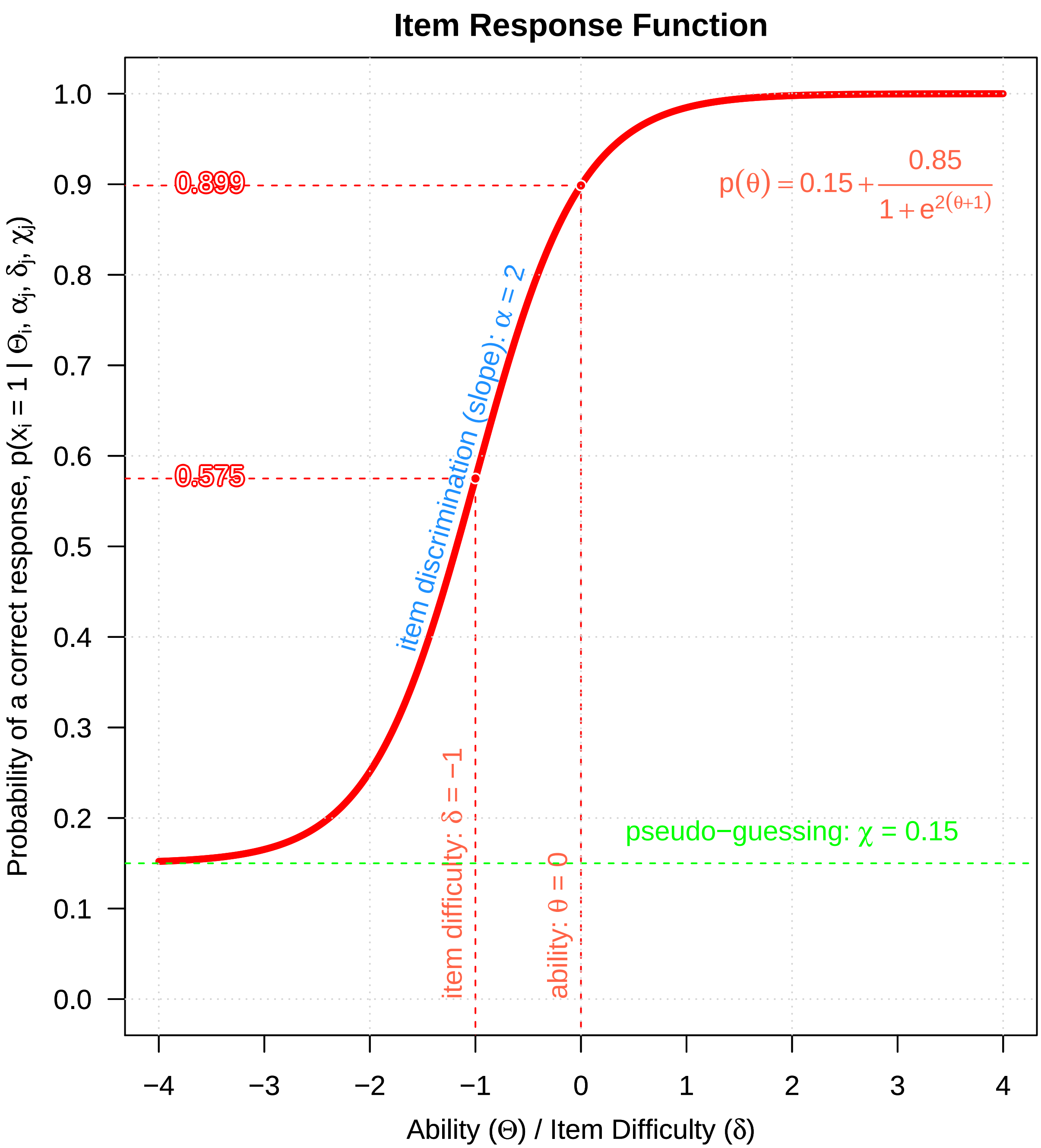

Item Response Function
Probability of a correct response, p(x_i = 1 | Θ_i, α_j, δ_j, χ_j)
Ability (Θ) / Item Difficulty (δ)
0.899
0.575
p(θ) = 0.15 + 0.85 / (1 + e^{2(θ+1)})
item discrimination (slope): α = 2
item difficulty: δ = −1
ability: θ = 0
pseudo−guessing: χ = 0.15
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−4
−3
−2
−1
0
1
2
3
4

Figure 8:

```
##        [,1]
## [1,] 0.8808
## [2,] 0.5000
##
## $expected.score
## [1] 44.04 50.00
```

```
# Discrimination, scale, slope
text(delta, (chi + 1)/2,
     label = substitute(paste("item discrimination (slope): ", alpha, " = ", a),
                        list(a = alpha)),
     srt = 296,
     pos = 3,
     col = "dodgerblue",
     cex = 1,
     offset = 1)

# theta
text(0, 0,
     label = substitute(paste("ability: ", theta, " = ", a),
                        list(a = person.theta)),
     srt = 90,
     col = "tomato",
     offset = -.5,
     pos = 4)



# slope at delta
y <- logistic(alpha = alpha, delta = delta, chi = chi, theta = delta)
points(delta-.2, y-.014, pch = 25, cex = 1.5, col = "dodgerblue", bg = "dodgerblue")

# delta
text(delta, 0,
     label = substitute(paste("item difficulty: ", delta, " = ", a),
                        list(a = delta)),
     srt = 90,
     col = "tomato",
```

```
     offset = -.5,
     pos = 4)

# gridlines
grid(nx = NULL, ny = NULL,
     col = "lightgray",
     lty = "dotted",
     lwd = 1)
```

1.1.3 Estimation of Item Parameters

```
mydata <- read.csv("MidTerm.csv", header = TRUE, na.strings = " ")

# Check
str(mydata)
```

```
## 'data.frame':    450 obs. of  15 variables:
##  $ V1 : int  0 0 0 1 0 0 1 0 0 0 ...
##  $ V2 : int  0 1 1 1 1 1 1 1 0 1 ...
##  $ V3 : int  1 1 1 1 1 0 1 0 0 1 ...
##  $ V4 : int  0 1 0 0 0 0 1 0 0 1 ...
##  $ V5 : int  1 0 0 0 0 1 0 1 1 0 ...
##  $ V6 : int  0 0 1 1 1 1 1 0 0 1 ...
##  $ V7 : int  0 1 1 1 1 1 1 0 0 1 ...
##  $ V8 : int  0 0 1 1 1 0 1 0 0 1 ...
##  $ V9 : int  0 0 1 1 1 0 1 0 0 1 ...
##  $ V10: int  1 1 1 1 1 1 1 0 1 1 ...
##  $ V11: int  0 1 1 1 1 1 1 1 1 1 ...
##  $ V12: int  0 1 0 0 0 0 1 0 0 1 ...
##  $ V13: int  0 0 1 1 0 0 1 0 0 1 ...
##  $ V14: int  0 0 0 1 0 0 1 0 0 1 ...
##  $ V15: int  1 1 1 1 1 0 1 0 1 1 ...
```

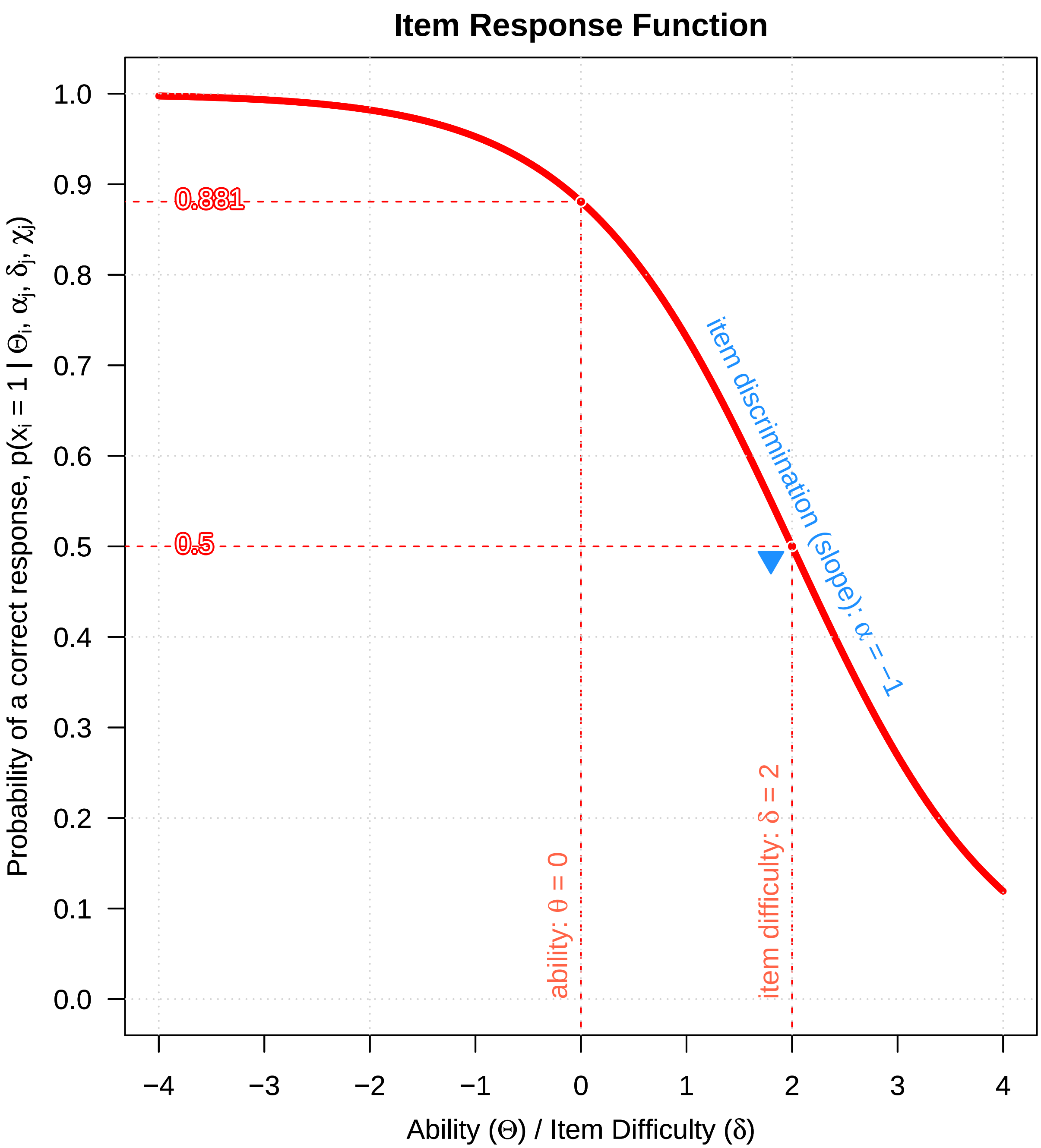

Item Response Function
Probability of a correct response, p(x_i = 1 | Θ_i, α_j, δ_j, χ_j)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.881
0.5
item discrimination (slope): α = −1
ability: θ = 0
item difficulty: δ = 2
−4
−3
−2
−1
0
1
2
3
4
Ability (Θ) / Item Difficulty (δ)

Figure 9:

```r
knitr::kable(head(mydata),
             caption = 'Random dichotomous test data')
```

Table 1: Random dichotomous test data

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15
0	0	1	0	1	0	0	0	0	1	0	0	0	0	1
0	1	1	1	0	0	1	0	0	1	1	1	0	0	1
0	1	1	0	0	1	1	1	1	1	1	0	1	0	1
1	1	1	0	0	1	1	1	1	1	1	0	1	1	1
0	1	1	0	0	1	1	1	1	1	1	0	0	0	1
0	1	0	0	1	1	1	0	0	1	1	0	0	0	0

Research Question: Which item was most discriminating and at what theta level would it function best?

```r
library(irtoys)
# Estimate item parameters (discrimination, difficulty, guessing)
est2PL <- irtoys::est(resp = mydata,
                      model = "2PL",
                      engine = "ltm")

# Estimate person parameters (ability)
theta2PL <- irtoys::eap(resp = mydata,
                        ip = est2PL$est,
                        qu = normal.qu())

knitr::kable(head(theta2PL),
             caption = 'Estmated person ability parameters')
```

Table 2: Estmated person ability parameters

est	sem	n
-1.27845	0.35607	15

est	sem	n
-0.22294	0.33872	15
0.63713	0.38915	15
1.48540	0.48689	15
0.47991	0.37758	15
-0.86393	0.34167	15

```
# Item Parameters
item.parameters <- data.frame(discrimination = est2PL$est[,1],
                              difficulty = est2PL$est[,2],
                              guessing = est2PL$est[,3])

# Sort Item Parameters according to Item Discrimination
item.parameters <- item.parameters[order(item.parameters$discrimination,
                                         decreasing = TRUE),]
knitr::kable(round(item.parameters, 3),
             caption = 'Item parameters')
```

Table 3: Item parameters

	discrimination	difficulty	guessing
V9	2.790	0.089	0
V8	2.514	-0.550	0
V6	2.473	-0.371	0
V14	2.369	0.919	0
V3	2.208	-1.173	0
V1	2.183	1.398	0
V11	2.171	-1.430	0
V15	2.168	-0.780	0
V10	2.164	-2.221	0
V2	1.903	-1.497	0
V7	1.297	-1.502	0
V13	1.072	0.679	0
V4	1.058	1.982	0
V12	0.980	1.584	0

	discrimination	difficulty	guessing
V5	-1.933	-0.049	0

Item 9 is the most discriminating ($\alpha_9 = 2.790$), and it would function best at $\Theta = \delta_9 = 0.089$

Research Question: Which items are the easiest and most difficult?

```
knitr::kable(round(item.parameters[order(item.parameters$difficulty,
                                          decreasing = TRUE),], 3),
           caption = 'Easiest items')
```

Table 4: Easiest items

	discrimination	difficulty	guessing
V4	1.058	1.982	0
V12	0.980	1.584	0
V1	2.183	1.398	0
V14	2.369	0.919	0
V13	1.072	0.679	0
V9	2.790	0.089	0
V5	-1.933	-0.049	0
V6	2.473	-0.371	0
V8	2.514	-0.550	0
V15	2.168	-0.780	0
V3	2.208	-1.173	0
V11	2.171	-1.430	0
V2	1.903	-1.497	0
V7	1.297	-1.502	0
V10	2.164	-2.221	0

```
knitr::kable(round(item.parameters[order(item.parameters$difficulty,
                                          decreasing = FALSE),], 3),
           caption = 'Most  difficult items')
```

Table 5: Most difficult items

	discrimination	difficulty	guessing
V10	2.164	-2.221	0
V7	1.297	-1.502	0
V2	1.903	-1.497	0
V11	2.171	-1.430	0
V3	2.208	-1.173	0
V15	2.168	-0.780	0
V8	2.514	-0.550	0
V6	2.473	-0.371	0
V5	-1.933	-0.049	0
V9	2.790	0.089	0
V13	1.072	0.679	0
V14	2.369	0.919	0
V1	2.183	1.398	0
V12	0.980	1.584	0
V4	1.058	1.982	0

Item 4 is the most difficult ($\delta_4 = 1.982$), and item 10 is the easiest ($\delta_{10} = -2.221$)

Research Question: Why do persons with scores of 14 all have the same theta value, but persons with scores of 13 have different theta values?

```
mydata$score <- rowSums(mydata[,1:15])

mydf <- data.frame(mydata, theta = theta2PL[,"est"])

knitr::kable(subset(mydf, subset = mydf$score == 13),
             caption = 'There are several different patterns all with a score of 13')
```

Table 6: There are several different patterns all with a score of 13

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	score	theta
10	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	13	1.45132
99	1	1	1	1	0	1	1	1	1	1	1	1	1	0	1	13	1.40873
152	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	13	0.87668
189	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	13	1.45132
190	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	13	1.73600
192	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	13	1.73600
214	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	13	1.73600
236	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	13	1.73600
334	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	13	1.73600
337	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	13	1.45132
341	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	13	1.73226
378	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	13	1.73600
414	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	13	1.45132
430	1	1	1	1	0	1	1	1	1	1	1	1	1	0	1	13	1.40873
447	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	13	1.73226

```
knitr::kable(subset(mydf, subset = mydf$score == 14),
          caption = 'There is a single scoring pattern (all answered only item 5 incorre
```

Table 7: There is a single scoring pattern (all answered only item 5 incorrectly) for a grade of 14

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	score	theta
7	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	14	2.0589
37	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	14	2.0589
113	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	14	2.0589
124	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	14	2.0589
135	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	14	2.0589
280	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	14	2.0589
284	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	14	2.0589
302	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	14	2.0589
331	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	14	2.0589

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	score	theta
342	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	14	2.0589

There are several different score patterns that all have a score of 13 while there is only one score pattern (they all missed item V5) for a score of 14. In the 1PL model, thetas are unique to scores; in the 2PL and 3PL models, thetas are unique to score patterns or response vectors.

1.2 Item (iif) and Test (tif) Information Functions

Item response theory extends the concept of reliability from a single index to a function called the information function. The IRT information function is the inverse of the conditional observed score standard error at any given test score. The statistical meaning of information is credited to Ronald Fisher who defined information as the reciprocal of the precision with which a parameter could be estimated. In other words, the higher the precision of an estimate of a parameter, the more known about the value of the parameter. Statistically, the precision with which a parameter is estimated is measured by the variability of the estimates around the value of the parameter. Hence, a measure of precision is the variance of the estimators, which is denoted by σ^2*. The amount of information, denoted by I, is given by the formula:*

$$I = \frac{1}{\sigma^2}$$

The test information at a given ability level is simply the sum of the item informations at that level,

$$I(\Theta) = \sum_{i=1}^{N} I_i(\Theta)$$

The item information function is for the 3PL model is

$$I(\Theta, \alpha, \delta, \chi) = \alpha^2 \frac{Q(\Theta)}{P(\Theta)} \Big[\frac{P(\Theta) - \chi}{1 - \chi}\Big]^2$$

1.2.1 The variance and the standard error of measurement of the ability estimate

The variance of the ability estimate, $\widehat{\Theta}$, is

$$\text{var}(\widehat{\Theta}) = \frac{1}{I(\widehat{\Theta})}$$

and the standard error of measurement is

$$\text{SEM}(\widehat{\Theta}) = \sqrt{\frac{1}{I(\widehat{\Theta})}}$$

Function: Given item (discrimination, difficulty and pseudo-guessing) and person location (ability) parameters, computes the item information function and SEM for the latent trait

```
information <-
  function(par.mat,
           person.theta,
           iif.plot = FALSE,
           tif.plot = FALSE,
           sem.plot = FALSE){

    source("logistic.R")

    par.mat <- as.matrix(par.mat)
    n <- nrow(par.mat)

    # Latent trait continium, person location (theta)
    theta <- seq(from = -4, to = 4, by = 0.01)
    k <- length(theta)

    # probability of correct/incorrect response and information matrices initiated
    p <-
      q <-
      I <-
      var.theta <-
      sem.theta <- matrix(data = NA, nrow = n, ncol = k, byrow = TRUE)

    # item and test information
    test.info <- rep(0, k)
```

```
# set color scheme
mycolors <- palette(rainbow(n+1))
mycolors <- palette(rainbow(n+1))

for (i in 1:n) {

  # probability of correct response
  p[i,] <- logistic(par.mat[i, 1], par.mat[i, 2], par.mat[i, 3], theta)

  # probability of incorrect response
  q[i,] <- 1 - p[i,]

  # item information
  I[i,] <- (par.mat[i, 1]^2) * (q[i,]/p[i,]) * ((p[i,] - par.mat[i, 3]) /
                                                    (1 - par.mat[i, 3]))^2

  # test information
  test.info <- test.info + I[i,]
}

# variance of theta
var.theta <- 1/test.info

# SEM of theta
sem.theta <- sqrt(var.theta)

if (iif.plot || tif.plot || sem.plot) {
  # start plot
  plot.new()

  ## add extra space to right margin of plot within frame
  par(mar = c(3.5, 4, 2, 3.5))

  # plot title
  title(main = "Information Function")
```

```
    # frame
    box()

    if (iif.plot && tif.plot) {
      left.ylim <- c(0, ceiling(max(test.info)))
      left.notches <- diff(left.ylim)
      left.ylab <- "Item Information"
      tif.side <- 4
    }
    else if (tif.plot && sem.plot) {
      left.ylim <- c(0, ceiling(max(test.info)))
      left.notches <- diff(left.ylim)
      right.ylim <- c(0, ceiling(max(sem.theta)))
      right.notches <- diff(right.ylim)
      left.ylab <- "Test Information"
      tif.side <- 2
    }
    else if (iif.plot && sem.plot) {
      left.ylim <- c(0, max(I))
      left.notches <- 5
      right.ylim <- c(0, ceiling(max(sem.theta)))
      right.notches <- diff(right.ylim)
      left.ylab <- "Item Information"
    }

    # x-axis
    x_axis <- function(){
      axis(1, pretty(range(theta), diff(range(theta))))

      # x-axis label
      mtext(expression(paste("Ability, ", Theta)),
            side = 1,
            col = "black",
            line = 2.5)
    }# end x_axis
  }# end if plot
```

```
# plot item information function (iif)
#----------------------------------------
if (iif.plot) {
  for (i in 1:n) {
    # Allow a second plot on the same graph
    par(new = TRUE)

    # iif plot
    plot(theta, I[i,],
         type = "l",
         lwd = 3,
         col = mycolors[i],
         axes = FALSE,
         ylim = left.ylim,
         xlab = "",
         ylab = "")

    if (i == 1) {
      # left y-axis
      axis(2, pretty(left.ylim, left.notches),
           col = mycolors[i],
           col.axis = mycolors[i],
           las = 1)

      mtext(left.ylab,
            side = 2,
            col = mycolors[i],
            line = 3)

      # x-axis
      x_axis()
    }# end if i==1
  }# end for i
}# end if iif.plot
```

```
# plot the SEM
#----------------------------------------
if (sem.plot) {
  # Allow a second plot on the same graph
  par(new = TRUE)

  plot(theta, sem.theta,
       type = "l",
       lwd = 3,
       col = mycolors[i+1],
       axes = FALSE,
       ylim = c(0, ceiling(max(sem.theta))),
       xlab = "",
       ylab = "")

  # x-axis
  x_axis()

  # right y-axis
  axis(4, pretty(c(0, ceiling(max(sem.theta))),
                 diff(c(0, ceiling(max(sem.theta))))),
       las = 1,
       col = mycolors[i+1],
       col.axis = mycolors[i+1])

  # right y-axis label
  mtext(expression(SEM(Theta)),
        side = 4,
        col = mycolors[i+1],
        line = 2)
}# end if sem.plot

# plot tif
#----------------------------------------
if (tif.plot) {
  ## Allow a second plot on the same graph
```

```
      par(new = TRUE)

      # test information function
      plot(theta, test.info,
           col = "black",
           type = "l",
           lwd = 5,
           axes = FALSE,
           ylim = c(0, ceiling(max(test.info))),
           xlab = "",
           ylab = "")

      # x-axis
      x_axis()

      # y-axis
      axis(tif.side, pretty(c(0, ceiling(max(test.info))),
                            diff(c(0, ceiling(max(test.info))))),
           las = 1,
           col = "black",
           col.axis = "black")

      # y-axis label
      mtext("Test Information",
            side = tif.side,
            col = "black",
            line = 2)
    }# end if tif.plot
  }# end information

dump("information", file = "information.R")

# item parameters
alpha <- 1 # Discrimination, scale, slope
delta <- 0 # Difficulty, item location
chi <- 0 # Pseudo-guessing, chance, asymptotic minimum
```

```
parameter.matrix <- cbind(alpha, delta, chi)

# the latent trait, ability (person location)
person.theta <- 0

# Plot item information function (iif)
information(parameter.matrix, person.theta, iif.plot = TRUE, sem.plot = TRUE)
# gridlines
grid(nx = NULL, ny = NULL, col = "lightgray", lty = "dotted", lwd = 1)
```

Note that the SEM function is quite low for abilities within the $\mu \pm 2\sigma$ range, and increases for both smaller and larger abilities.

1.3 Latent Trait (ability) or Person Location (Θ) Estimation

The Algorithm:

- Step 1: Calculate the probability of a correct response to each dichotomous item.
- Step 2: Determine the probabiliy of response pattern.
- Step 3: Perform the steps 1 and 2 for a range of ability values and determine which of the various values of ability has the highest likelihood of producing the given response pattern.

1.3.1 1-PL Model

```
# Parameter Matrix
# One row per item and three columns for each of the three parameters:
# a (or alpha) is discrimination
# b (delata) is difficulty
# c (chi) is pseudo-guessing

par.mat <- matrix(c(1.0, -2.5, 0,
                    1.0, -2.0, 0,
                    1.0, -1.5, 0,
```

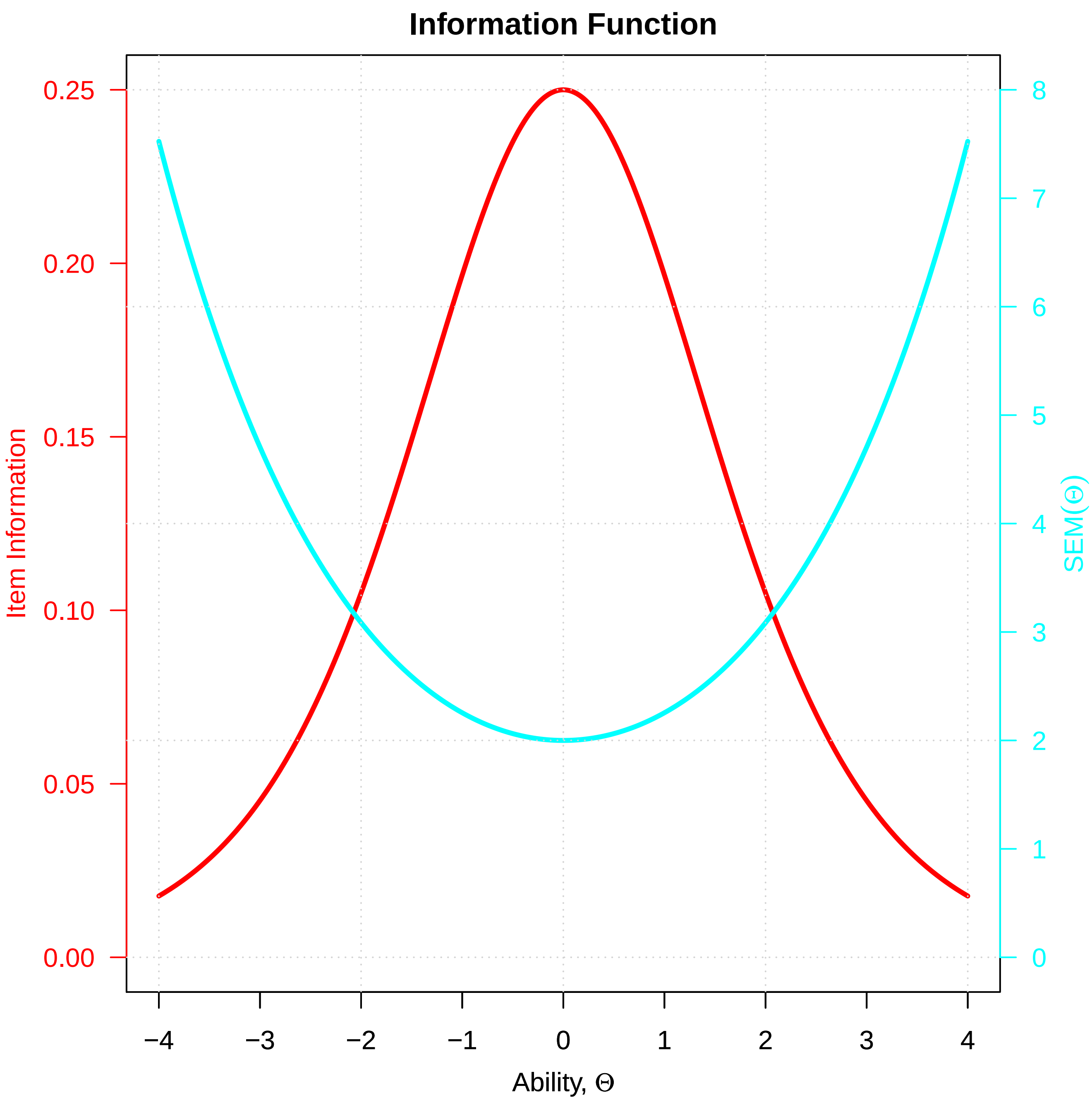

Information Function
Item Information
0.25
0.20
0.15
0.10
0.05
0.00
SEM(Θ)
8
7
6
5
4
3
2
1
0
−4
−3
−2
−1
0
1
2
3
4
Ability, Θ

Figure 10:

```
                        1.0, -1.0, 0,
                        1.0, -0.5, 0,
                        1.0,  0.0, 0,
                        1.0,  0.5, 0,
                        1.0,  1.0, 0,
                        1.0,  1.5, 0,
                        1.0,  2.0, 0,
                        1.0,  2.5, 0),
                       nrow = 11, ncol = 3,
                       byrow = TRUE)
dimnames(par.mat) <- list(rownames(par.mat, do.NULL = FALSE, prefix = "i"),
                          c("a", "b", "c"))

knitr::kable(t(par.mat),
             caption = '1PL item parameters matrix.')
```

Table 8: 1PL item parameters matrix.

	i1	i2	i3	i4	i5	i6	i7	i8	i9	i10	i11
a	1.0	1	1.0	1	1.0	1	1.0	1	1.0	1	1.0
b	-2.5	-2	-1.5	-1	-0.5	0	0.5	1	1.5	2	2.5
c	0.0	0	0.0	0	0.0	0	0.0	0	0.0	0	0.0

1.3.1.1 Probability of the Response Pattern and the Expected Score 1PL Determine the probabiliy of response pattern. Person location estimation depends on the conditional independence assumption, which states that for a given Θ the responses are independent of one another, so the probablity of any response pattern is the product of individual item probabilities (multiplication rule for independent events).

```
# A person with an average ability
person.theta <- 0
IRF(par.mat, person.theta,
    irf.plot = TRUE,
    trf.plot = TRUE,
    trace = TRUE)
```

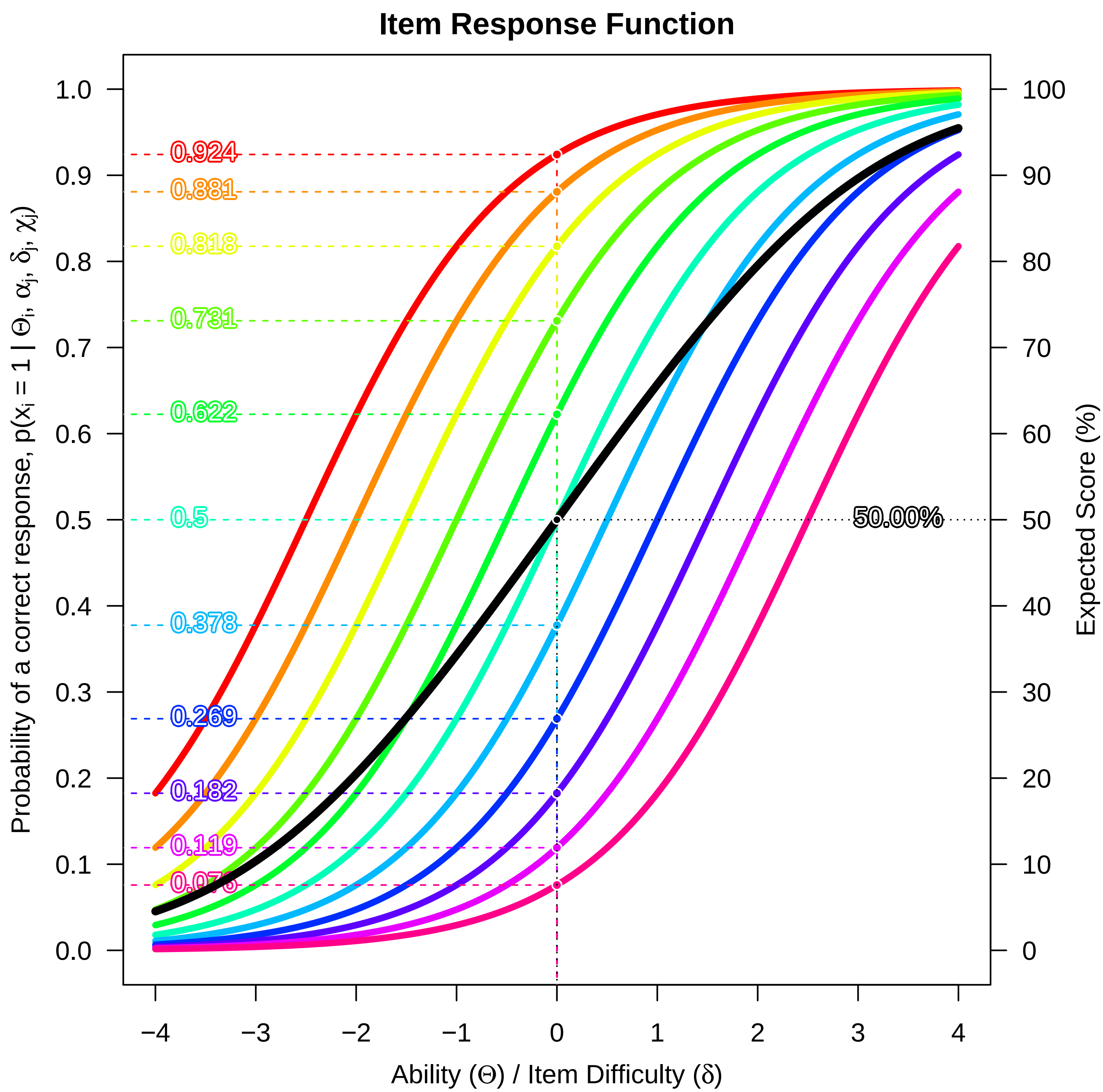

Item Response Function
0.924
0.881
0.818
0.731
0.622
0.5
0.378
0.269
0.182
0.119
50.00%
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
100
90
80
70
60
50
40
30
20
10
0
−4
−3
−2
−1
0
1
2
3
4
Probability of a correct response, p(x_i = 1 | Θ_i, α_j, δ_j, χ_j)
Expected Score (%)
Ability (Θ) / Item Difficulty (δ)

Figure 11:

```
## $probabilities
##         [,1]   [,2]    [,3]    [,4]    [,5] [,6]    [,7]    [,8]    [,9]
## [1,] 0.92414 0.8808 0.81757 0.73106 0.62246  0.5 0.37754 0.26894 0.18243
##       [,10]    [,11]
## [1,] 0.1192 0.075858
##
## $expected.score
## [1] 50
```

1.3.1.2 Item and Test Information Functions The maximum value of the item information function occurs at the point where the probabilities of a correct and of an incorrect response are both equal to 0.5. In other words, items in the 1PL model is most informative when the ability of the examinee is equal to the difficulty of items and it decreases as ability moves away from the item difficulty (i.e. when the item is eiter too easy or too diffucult for the examinee).

```r
information(par.mat, person.theta, iif.plot = TRUE, tif.plot = TRUE)
```

```r
information(par.mat, person.theta, tif.plot = TRUE, sem.plot = TRUE)
```

1.3.2 2-PL Model

```r
# Parameter Matrix
# One row per item and three columns for each of the three parameters:
# a (or alpha) is discrimination
# b (delata) is difficulty
# c (chi) is pseudo-guessing

par.mat <- matrix(c(1.1, -2.5, 0,
                    1.8, -2.0, 0,
                    0.9, -1.5, 0,
                    1.2, -1.0, 0,
```

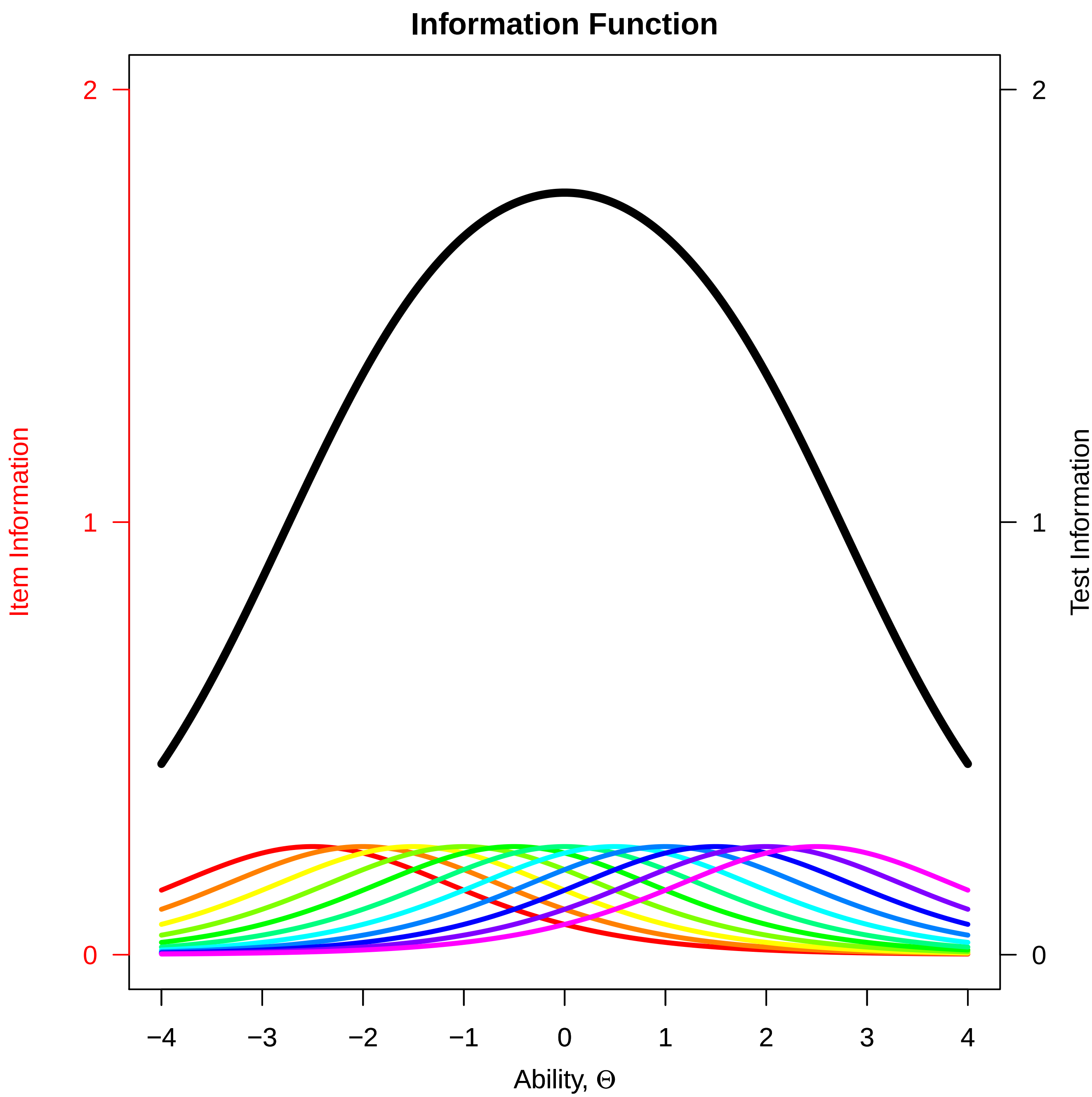

Information Function
Item Information
Test Information
0
1
2
0
1
2
−4
−3
−2
−1
0
1
2
3
4
Ability, Θ

Figure 12:

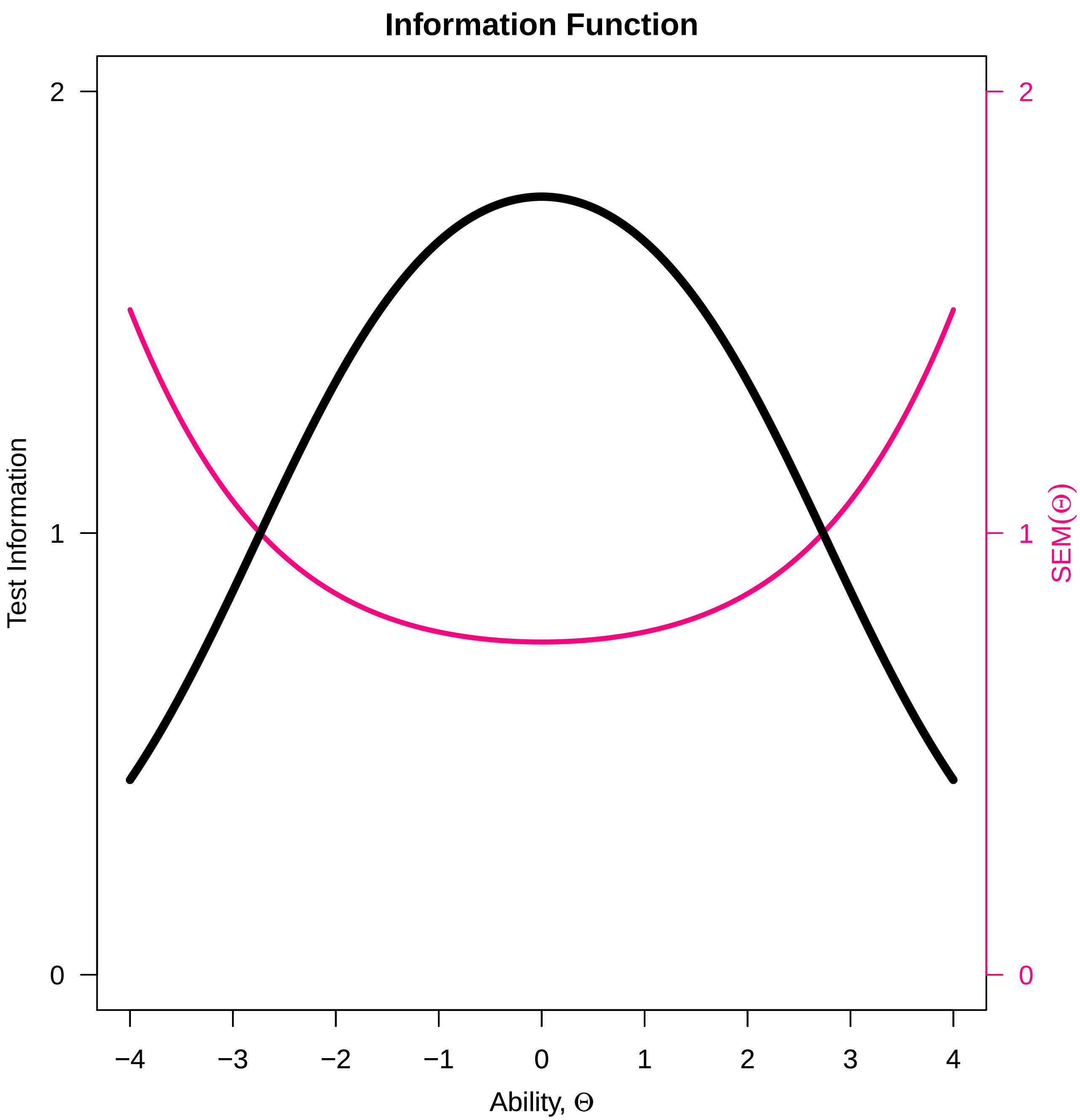

Information Function
Test Information
SEM(Θ)
Ability, Θ
0
1
2
−4
−3
−2
−1
0
1
2
3
4

Figure 13:

```
                    1.7, -0.5, 0,
                    1.0,  0.0, 0,
                    2.0,  0.5, 0,
                    1.3,  1.0, 0,
                    1.5,  1.5, 0,
                    0.6,  2.0, 0,
                    1.6,  2.5, 0),
                   nrow = 11, ncol = 3,
                   byrow = TRUE)
dimnames(par.mat) <- list(rownames(par.mat, do.NULL = FALSE, prefix = "I"),
                          c("a", "b", "c"))

knitr::kable(t(par.mat),
             caption = '2PL item parameters matrix.')
```

Table 9: 2PL item parameters matrix.

	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11
a	1.1	1.8	0.9	1.2	1.7	1	2.0	1.3	1.5	0.6	1.6
b	-2.5	-2.0	-1.5	-1.0	-0.5	0	0.5	1.0	1.5	2.0	2.5
c	0.0	0.0	0.0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0

```
# A person with an average ability
person.theta <- 0
IRF(par.mat, person.theta,
    irf.plot = TRUE,
    trf.plot = TRUE,
    trace = TRUE)
```

1.3.2.1 Probability of the Response Pattern and the Expected Score 2PL

```
## $probabilities
##           [,1]    [,2]     [,3]     [,4]     [,5] [,6]     [,7]     [,8]      [,9]
## [1,] 0.93991 0.9734 0.79413 0.76852 0.70057  0.5 0.26894 0.21417 0.095349
```

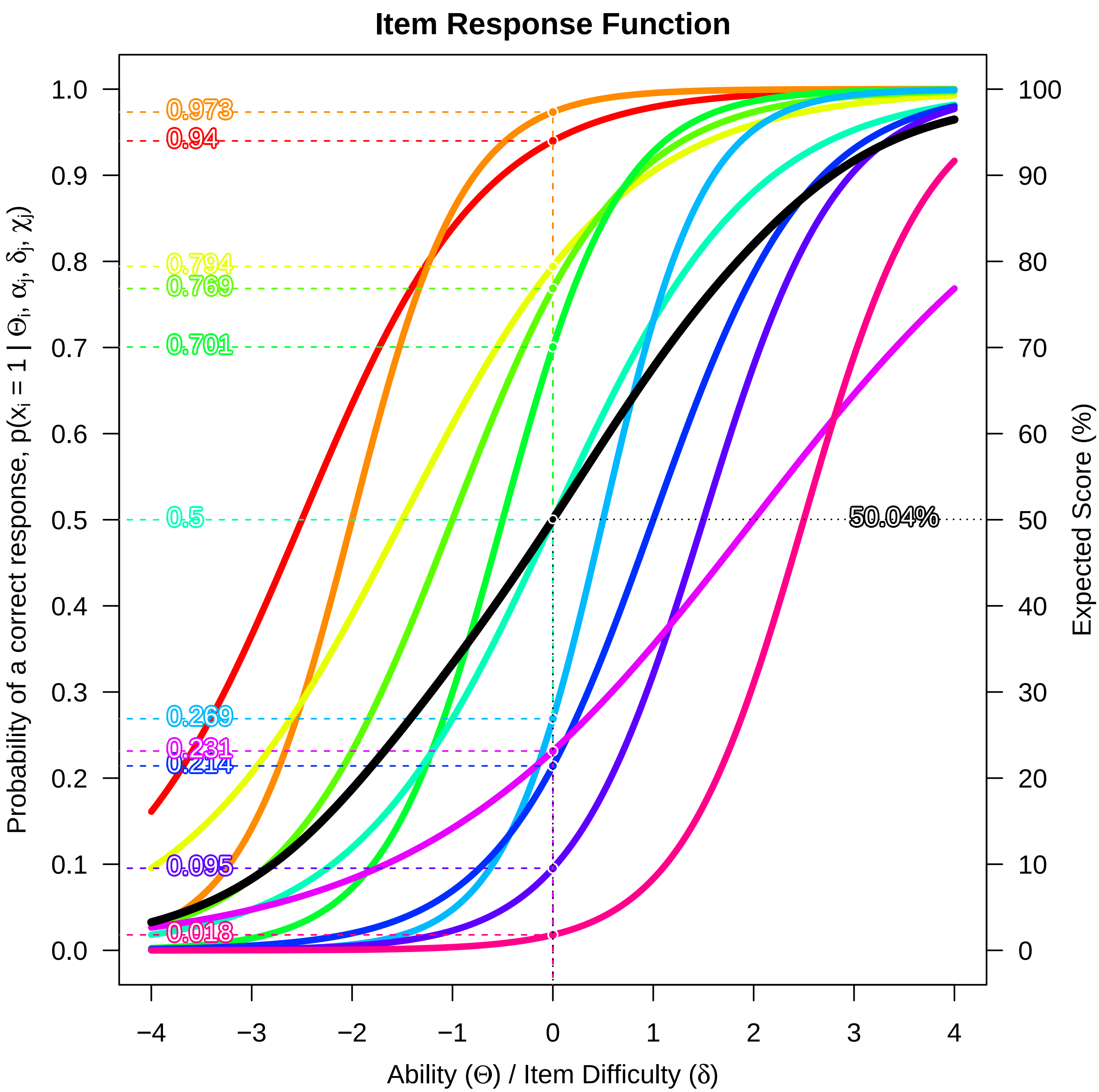

Item Response Function
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.973
0.94
0.794
0.769
0.701
0.5
0.269
0.231
0.214
0.095
0.018
50.04%
100
90
80
70
60
50
40
30
20
10
0
−4
−3
−2
−1
0
1
2
3
4
Ability (Θ) / Item Difficulty (δ)
Probability of a correct response, p(xi = 1 | Θi, αj, δj, χj)
Expected Score (%)

Figure 14:

```
##         [,10]    [,11]
## [1,] 0.23148 0.017986
##
## $expected.score
## [1] 50.041
```

```r
information(par.mat, person.theta, iif.plot = TRUE, tif.plot = TRUE)
```

```r
information(par.mat, person.theta, tif.plot = TRUE, sem.plot = TRUE)
```

1.3.2.2 Plot Item and Test Information Functions

1.3.3 3-PL Model

```r
# Parameter Matrix 3PL
# One row per item and three columns for each of the three parameters:
# a (or alpha) is discrimination
# b (delata) is difficulty
# c (chi) is pseudo-guessing

par.mat <- matrix(c(1.1, -2.5, 0.15,
                    1.8, -2.0, 0.10,
                    0.9, -1.5, 0.16,
                    1.2, -1.0, 0.09,
                    1.7, -0.5, 0.05,
                    1.0,  0.0, 0.20,
                    2.0,  0.5, 0.12,
                    1.3,  1.0, 0.02,
                    1.5,  1.5, 0.22,
                    0.6,  2.0, 0.25,
```

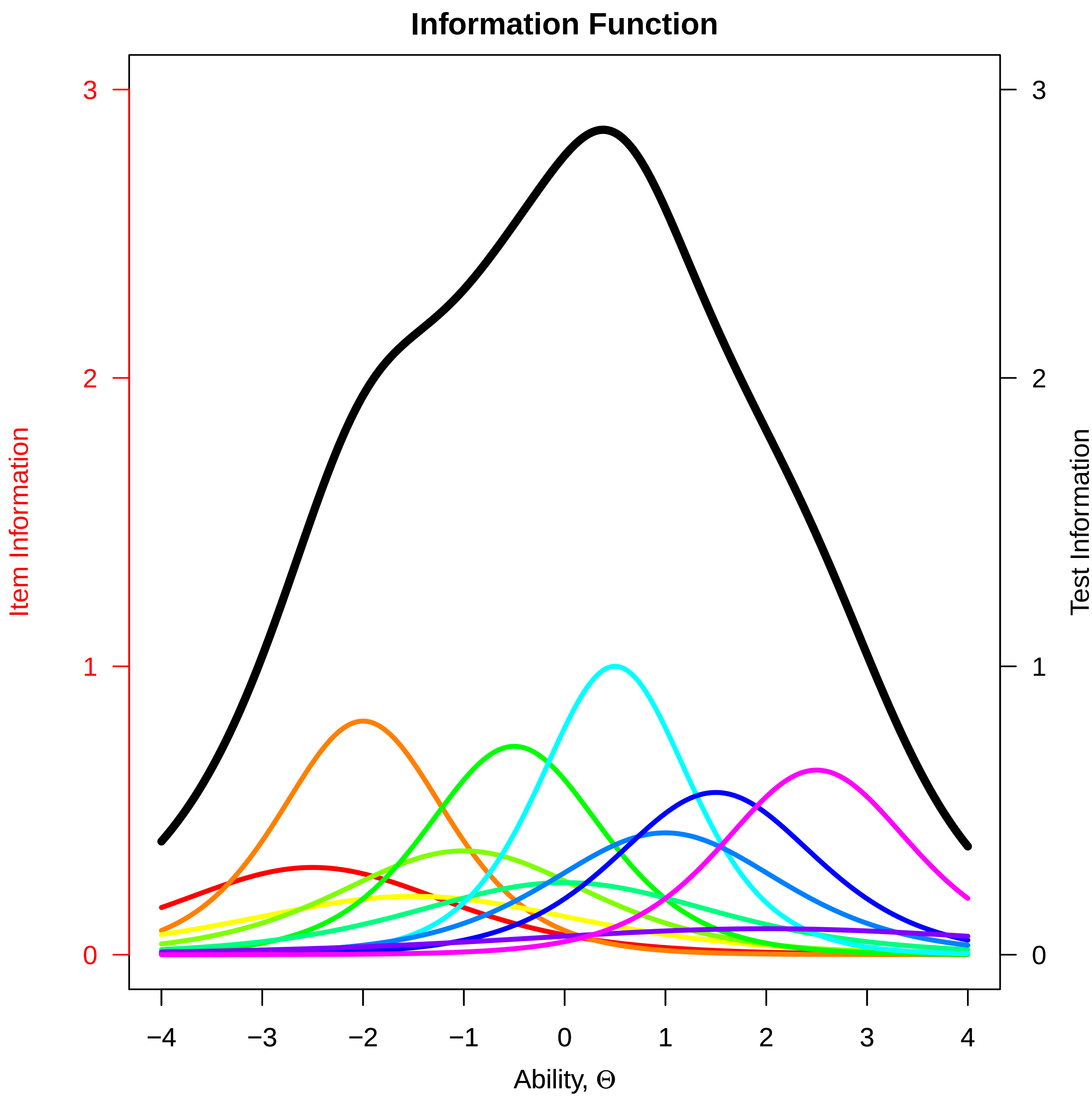

Information Function
Item Information
Test Information
Ability, Θ
3
2
1
0
−4
−3
−2
−1
0
1
2
3
4

Figure 15:

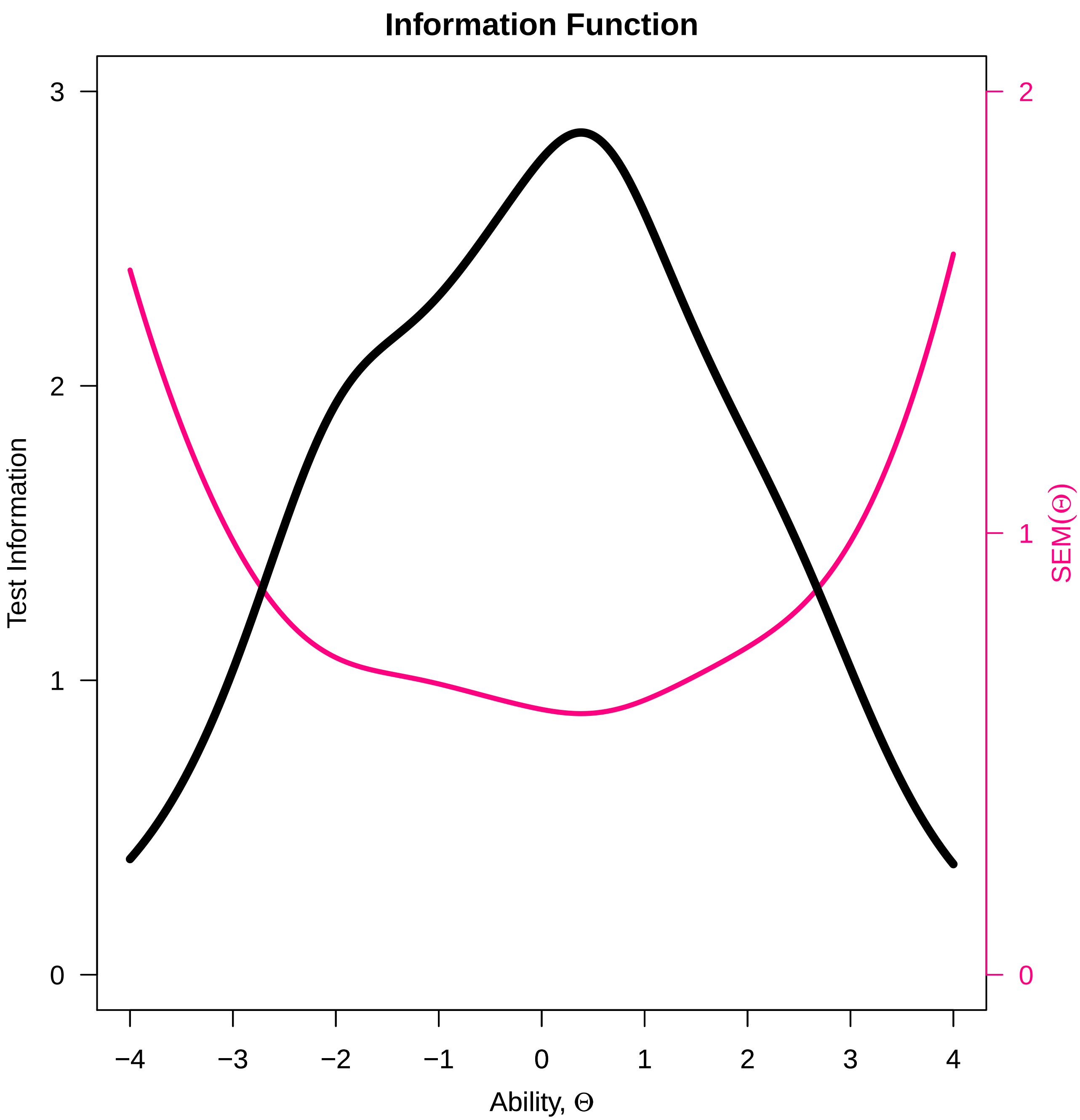

Information Function
Test Information
SEM(Θ)
Ability, Θ
3
2
1
0
2
1
0
−4
−3
−2
−1
0
1
2
3
4

Figure 16:

```
                     1.6,  2.5, 0.01),
                    nrow = 11, ncol = 3,
                    byrow = TRUE)
dimnames(par.mat) <- list(rownames(par.mat, do.NULL = FALSE, prefix = "I"),
                          c("a", "b", "c"))

knitr::kable(t(par.mat),
             caption = '3PL item parameters matrix.')
```

Table 10: 3PL item parameters matrix.

	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11
a	1.10	1.8	0.90	1.20	1.70	1.0	2.00	1.30	1.50	0.60	1.60
b	-2.50	-2.0	-1.50	-1.00	-0.50	0.0	0.50	1.00	1.50	2.00	2.50
c	0.15	0.1	0.16	0.09	0.05	0.2	0.12	0.02	0.22	0.25	0.01

```
# A person with an average ability
person.theta <- 0
IRF(par.mat, person.theta,
    irf.plot = TRUE,
    trf.plot = TRUE)
```

1.3.3.1 Probability of the Response Pattern and the Expected Score 3PL

```
## $probabilities
##         [,1]    [,2]    [,3]    [,4]    [,5] [,6]    [,7]    [,8]    [,9]
## [1,] 0.94893 0.97606 0.82707 0.78936 0.71554  0.6 0.35667 0.22988 0.29437
##        [,10]    [,11]
## [1,] 0.42361 0.027806
##
## $expected.score
## [1] 56.266
```

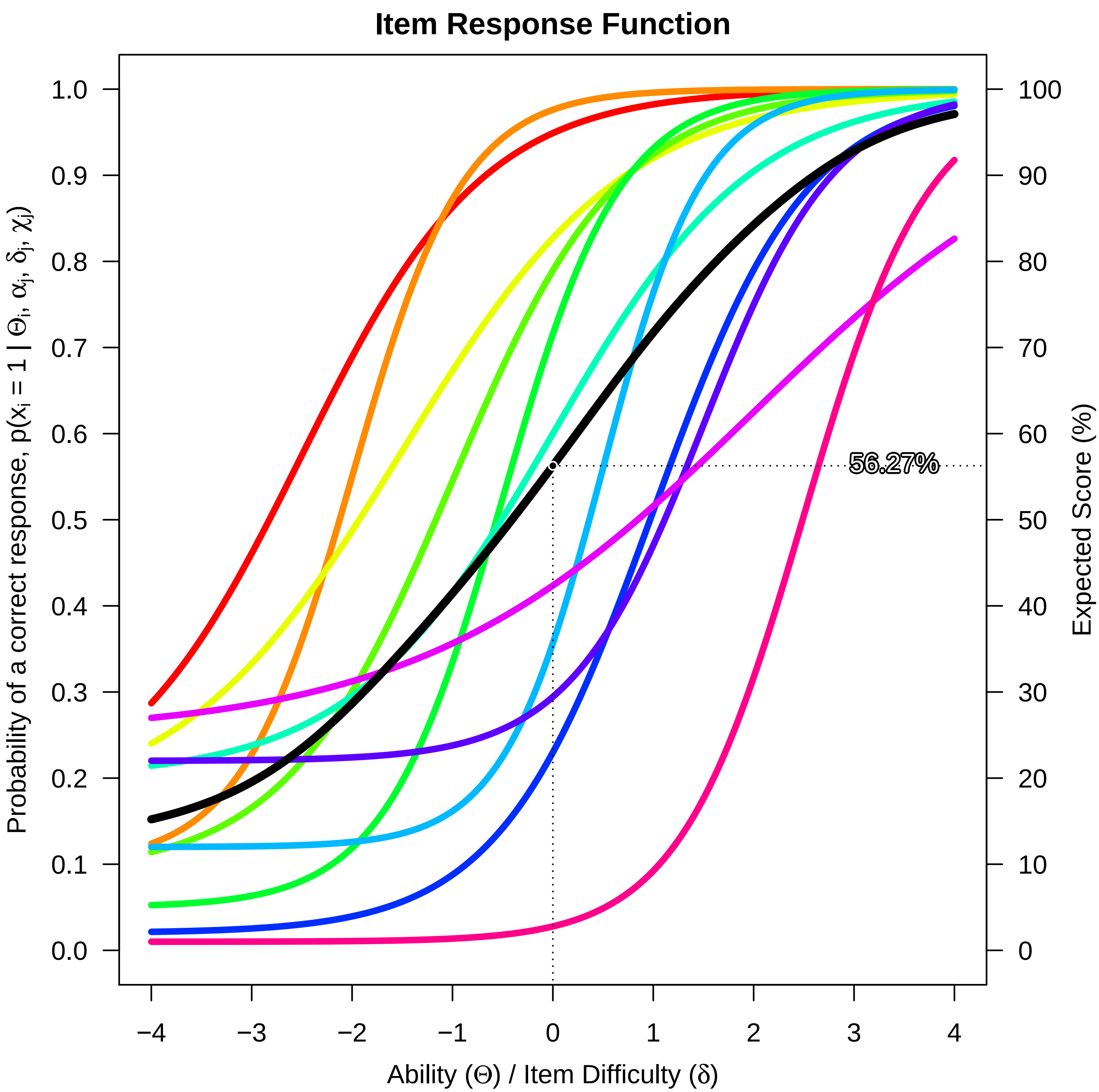

Item Response Function
Probability of a correct response, p(x_i = 1 | Θ_i, α_j, δ_j, χ_j)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Expected Score (%)
100
90
80
70
60
50
40
30
20
10
0
56.27%
−4
−3
−2
−1
0
1
2
3
4
Ability (Θ) / Item Difficulty (δ)

Figure 17:

```r
information(par.mat, person.theta, iif.plot = TRUE, tif.plot = TRUE)
```

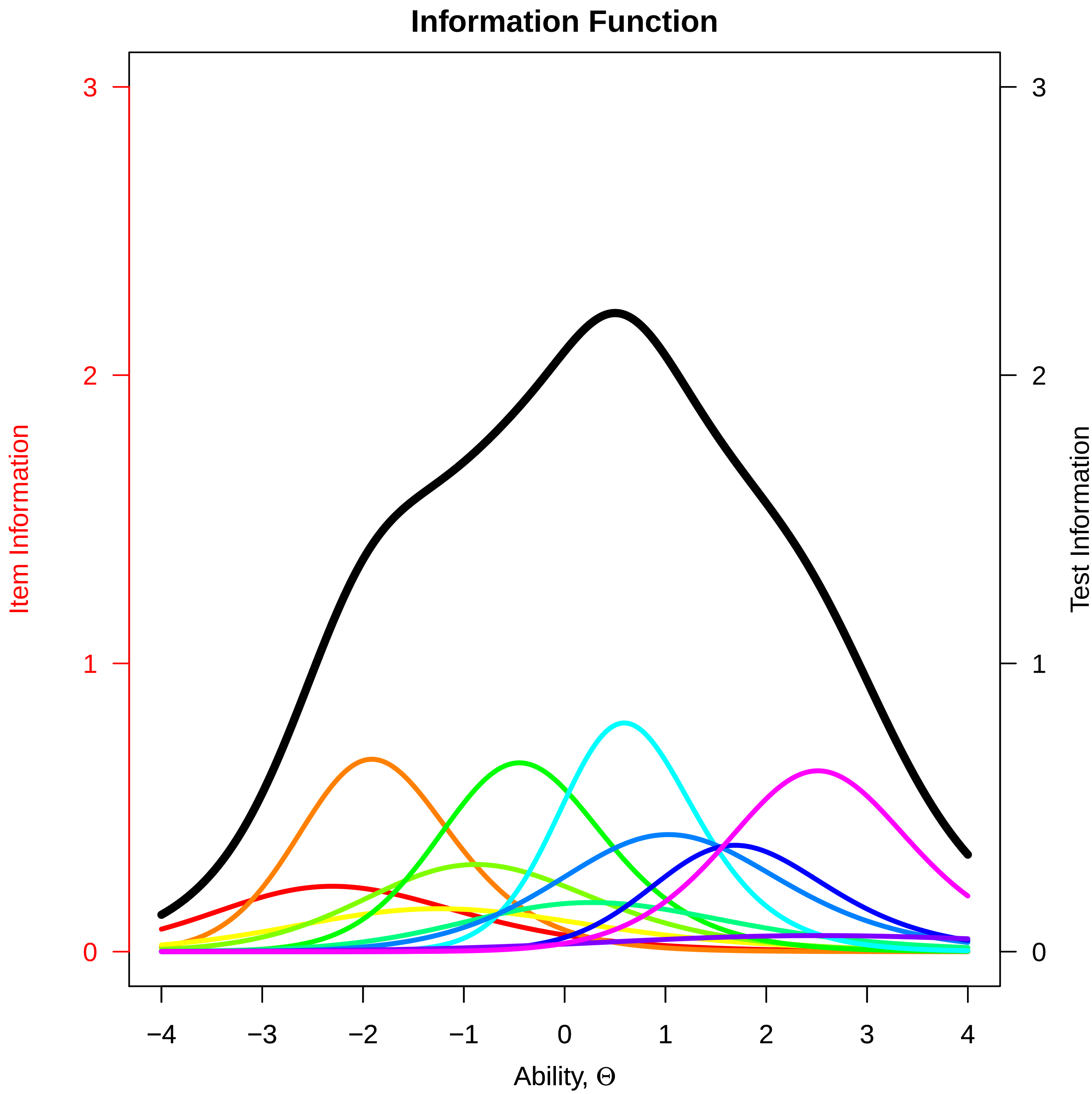


Figure 18:

```r
information(par.mat, person.theta, tif.plot = TRUE, sem.plot = TRUE)
```

1.3.3.2 Plot Item and Test Information Functions 3PL

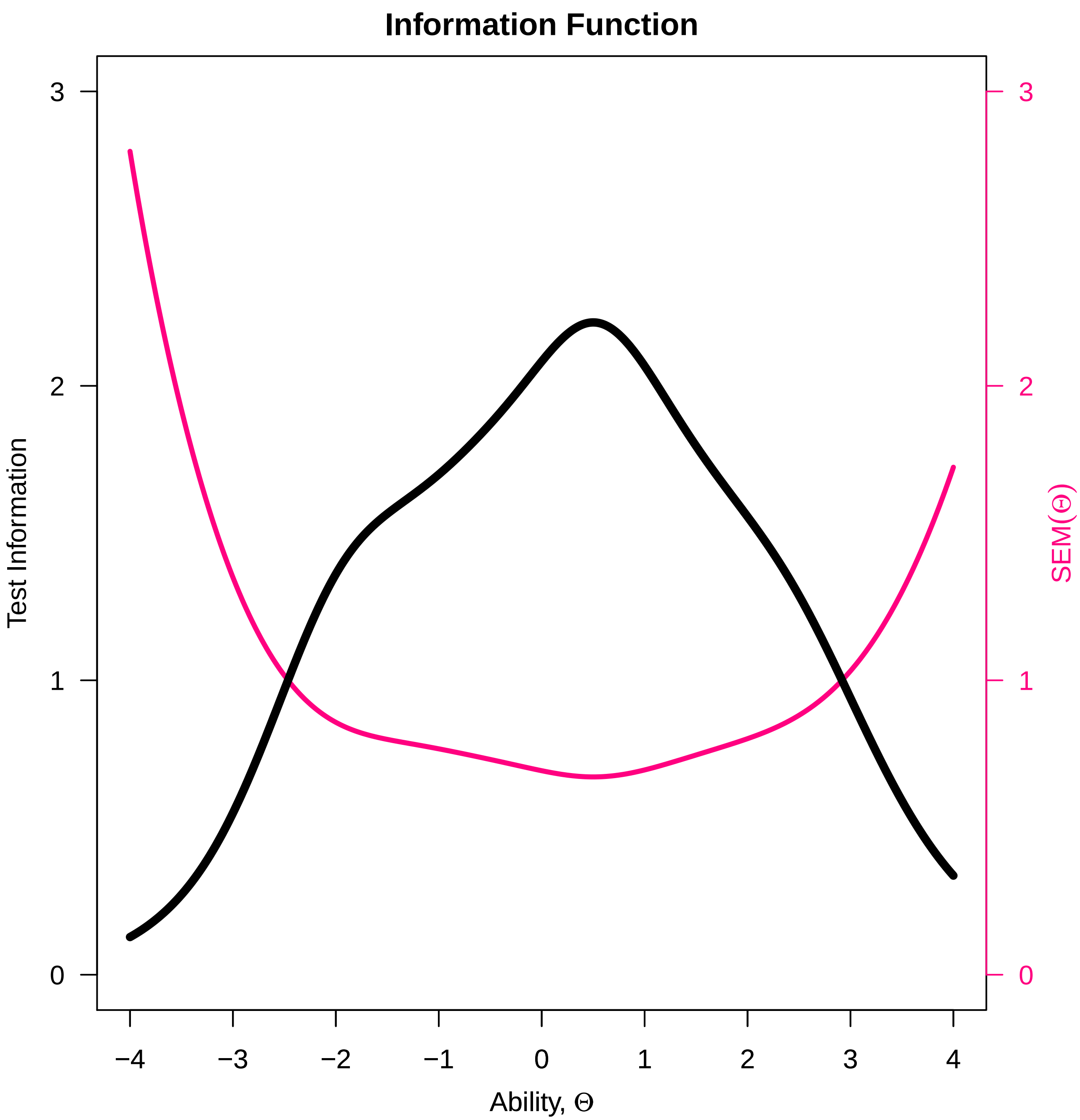

Information Function
Test Information
SEM(Θ)
Ability, Θ
0
1
2
3
−4
−3
−2
−1
0
1
2
3
4

Figure 19:

Plots by irtoys Package

```
# plot Item Response Function(irf)
plot(irtoys::irf(par.mat), co = NA, label = TRUE)
```

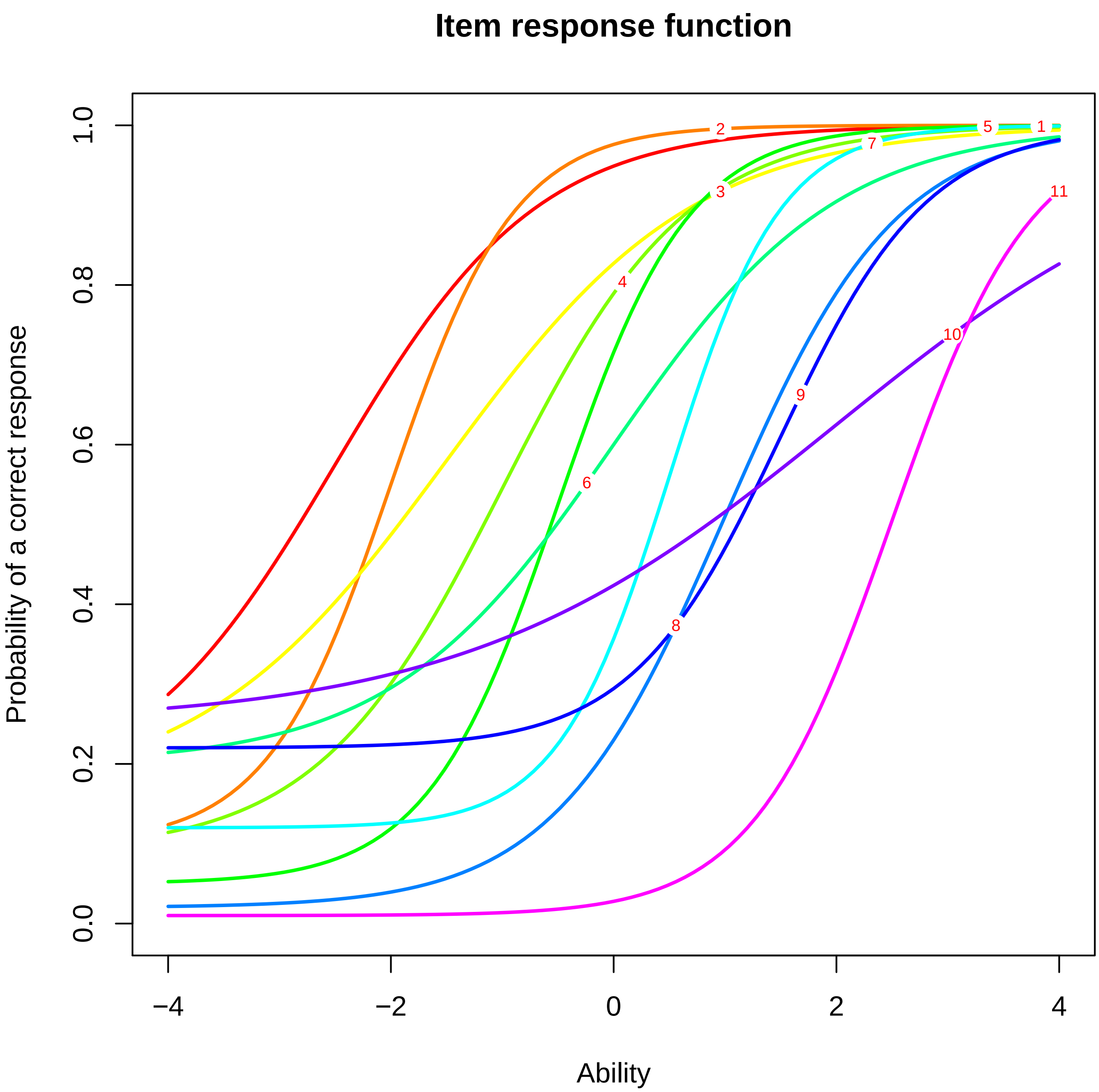


Figure 20:

```
# plot Test Response Function (trf)
plot(irtoys::trf(par.mat), co = NA)
```

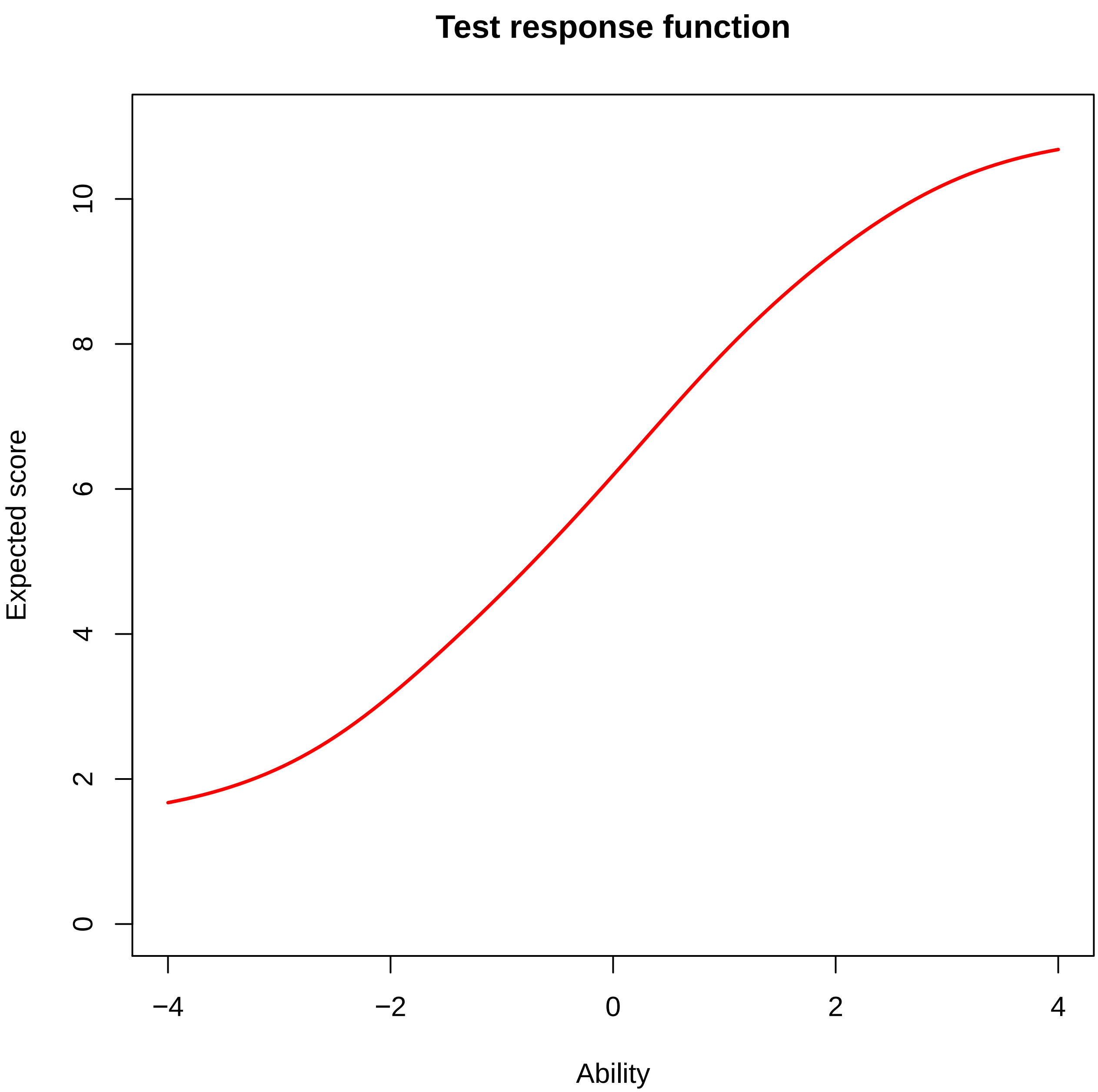

Test response function
Expected score
Ability
0
2
4
6
8
10
−4
−2
0
2
4

Figure 21:

```r
# plot Test Information Function(tif)
plot(irtoys::tif(par.mat))

# plot Item Information Function (iif)
plot(irtoys::iif(par.mat), co = NA, add = TRUE, label = TRUE)
```

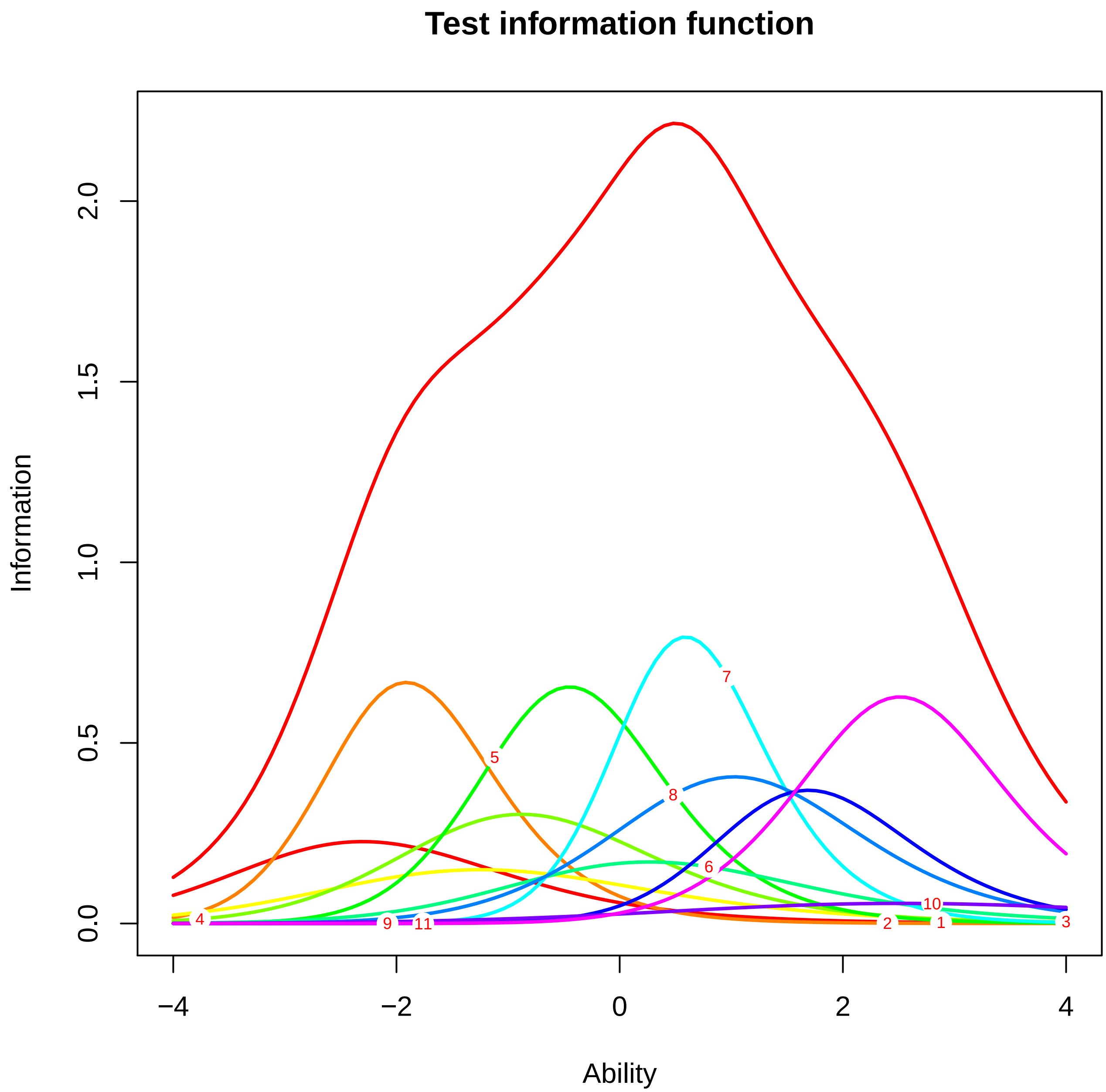


Figure 22:

1.3.4 Joint Maximum Likelihood Estimation

Assuming conditional independence and without loss of generalizability, the probability of a person's responses to dichotomous items is the product of the probability of the responses accross an instrument's items.

$$p(x_i|\Theta_j, \alpha_i, \delta_i) = \prod_{j=1}^{N} p_j^{x_j}(1 - p_j)^{1-x_j}$$

The joint likelihood function accross both persons and items is then the product of the product:

$$L = \prod_{i=1}^{N} \prod_{j=1}^{K} p_j(\Theta_i)^{x_{ij}}(1 - p_j(\Theta_i))^{1-x_{ij}}$$

where N is the number of items, and K is the number of persons.

Function: Given a response and item parameter matrix (α, δ, χ), computes the joint likelihood extimation for the latent trait continium $(-4, 4)$, then the maximum joint likelihood can be computed by using the built in R function, $max()$.

```
# given the row number, i, in the response matrix calculates
# the likelihood for that particular person
likelihood <-
  function(index = NULL, resp.mat, par.mat){

    # Latent trait continium, person location (theta)
    theta <- seq(from = -4, to = 4, by = 0.01)

    if (is.null(index)) {
      i <- 1:dim(resp.mat)[1]
    }
    else if (!is.null(index)) {
      i <- index
    }

    # initialize likelihood
    L <- rep(0, length(i))

    for (k in i) {
```

```
      if (resp.mat[k, 1] == 1) {
        L <- logistic(par.mat[1, 1], par.mat[1, 2], par.mat[1, 3], theta)
      }
      else if (resp.mat[k, 1] == 0) {
        L <- 1 - logistic(par.mat[1, 1], par.mat[1, 2], par.mat[1, 3], theta)
      }
      else{
        return("Response must be a 0/1 vector")
      }

      for (j in 2:ncol(resp.mat)) {
        if (resp.mat[k, j] == 1) {
          L <- L*logistic(par.mat[j, 1],
                          par.mat[j, 2],
                          par.mat[j, 3],
                          theta)
        }
        else if (resp.mat[k, j] == 0) {
          L <- L*(1 - logistic(par.mat[j, 1],
                               par.mat[j, 2],
                               par.mat[j, 3],
                               theta))
        }
        else {
          return("Response must be either 1 or 0")
        }
      }
    }
    return(L)
  } # end likelihood

dump("likelihood", file = "likelihood.R")
```

Ralph's Location Estimation

```r
# Response Matrix
resp.mat <- matrix(c(1, 1, 1, 1, 1, 1, 1, 1, 1,
                     1, 1, 1, 1, 1, 0, 0, 0, 0,
                     1, 0, 0, 0, 0, 0, 0, 0, 0),
                   nrow = 3,
                   ncol = 9,
                   byrow = TRUE)

dimnames(resp.mat) <- list(c("Ralph", "Suzy", "Alice"),
                          colnames(resp.mat, do.NULL = FALSE, prefix = "i"))

knitr::kable(resp.mat,
             caption = 'Person response matrix (1=correct response, 0=incorrect response.')
```

Table 11: Person response matrix (1=correct response, 0=incorrect response.

	i1	i2	i3	i4	i5	i6	i7	i8	i9
Ralph	1	1	1	1	1	1	1	1	1
Suzy	1	1	1	1	1	0	0	0	0
Alice	1	0	0	0	0	0	0	0	0

```r
# Latent trait continium, person location (theta)
theta <- seq(from = -4, to = 4, by = 0.01)

# Ralph's response vector is (1, 1, 1, 1, 1, 1, 1, 1, 1), so
# the Likelihood function for Ralph is p1*p2*p3*p4*p5*p6*p7*p8*p9
LthetaRalph <- likelihood(1, resp.mat, par.mat)

# Log-likelihood function for Ralph
LLthetaRalph <- log(LthetaRalph)

# posterior (normal distribution * likelihood)
posterior <- LthetaRalph*pnorm(theta, 0, 1)
```

Plot Log-likelihood function

```
LLplot <-
  function(i, LLtheta){

    # for arrows
    require(shape,
        quietly = TRUE,
        warn.conflicts = FALSE)

    require(graphics,
        quietly = TRUE,
        warn.conflicts = FALSE)

    #source("likelihood.R")

    # Latent trait continium, person location (theta)
    theta <- seq(from = -4, to = 4, by = 0.01)

    # Log-likelihood
    LLtheta <- log(likelihood(i, resp.mat, par.mat))

    data <- cbind(theta, LLtheta)

    plot(data,
         yaxs = "i",
         xaxs = "i",
         type = "l",
         lwd = 5,
         pch = 20,
         cex = .5,
         col = "darkgreen",
         xlim = c(min(theta), max(theta) + diff(range(theta)) * .1),
         ylim = c(min(LLtheta), max(LLtheta) + diff(range(LLtheta)) * .1),
         main = "Log-likelihood of Latent Trait (theta)")

    # tracing lines
    x0 <- subset(data, subset = data[,2] == max(data[,2]))[1]
```

```
y0 <- max(data[,2])

Arrows(x0 = x0,
       y0 = y0,
       x1 = x0,
       y1 = min(data[,2]),
       col = "black",
       arr.type = "triangle",
       arr.adj = 1,
       code = 2,
       lty = 3,
       lwd = 1)

segments(x0, y0,
         x1 = min(data[,1]),
         y1 = y0,
         col = "black",
         lty = 3,
         lwd = 1)

# maximum point
points(x0, y0,
       col = "white",
       bg = "darkgreen",
       pch = 21,
       cex = 1)

# coordinates of the maximum
text(x0, y0,
     pos = 3,
     cex = 1,
     col = "darkgreen",
     round(x0, 4)
)

# text maximum log-likelihood of theta
```

```
    text(x0, y0 -.07*diff(range(LLtheta)),
         "maximum log-likelihood of theta",
         pos = 2,
         offset = -.05,
         srt = 90,
         col = "darkgreen")
    points(x0, y0 -.035*diff(range(LLtheta)),
       pch = 24,
       cex = 1.5,
       col = "darkgreen",
       bg = "darkgreen")

    # gridlines
    grid(nx = NULL, ny = NULL, col = "lightgray", lty = "dotted", lwd = 1)
  } # end LLPlot
```

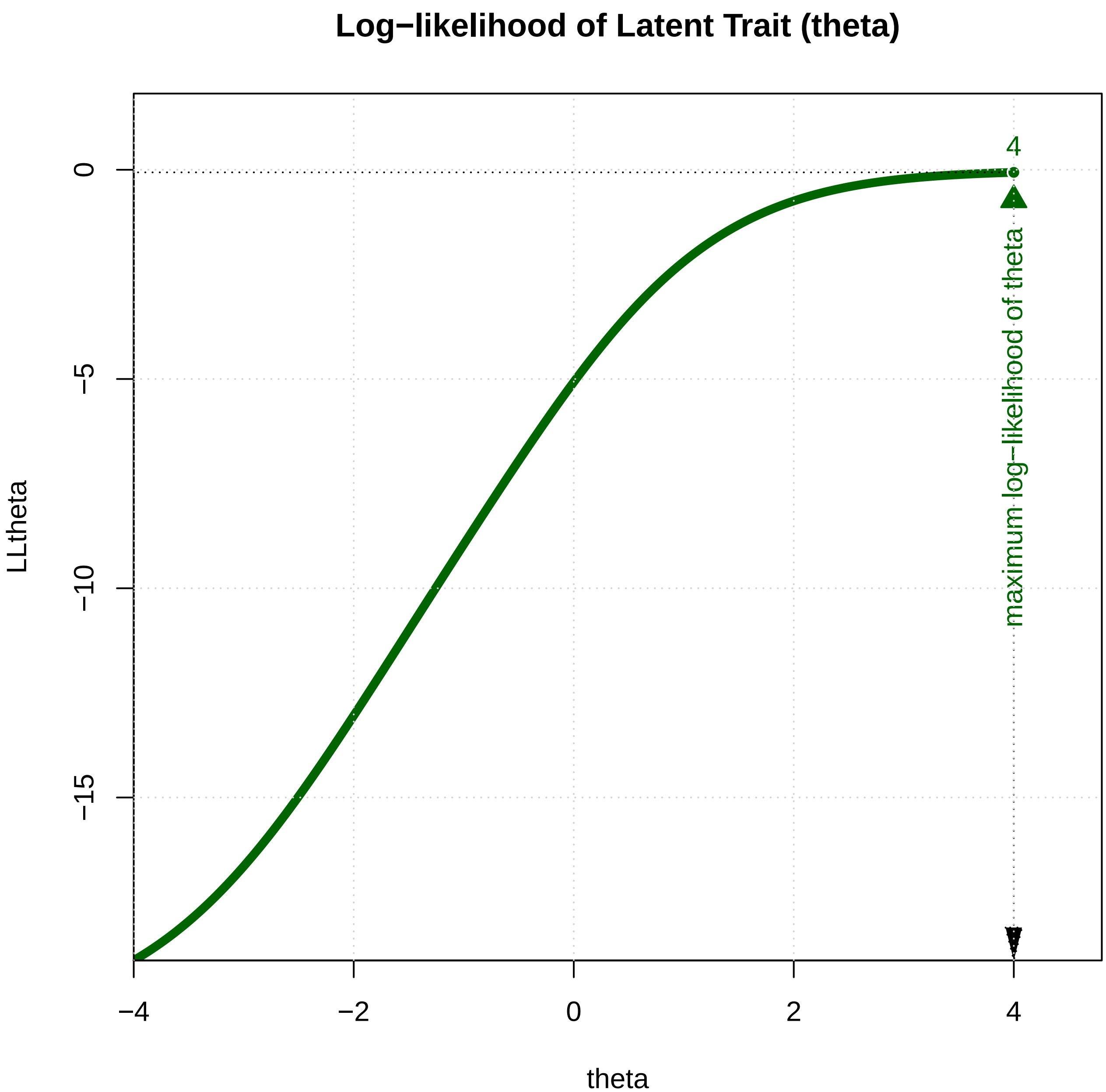
Plot Ralph's (1) Log-likelihood function

```
LLplot(1, LLthetaRalph)
```

1.3.5 Bayesian Strategy: Marginal Maximum Likelihood (MMLE) Person Location Estimation (Θ)

MLE cannot produce finite estimates of a person when he or she obtains either a zero or a perfect score. Additional information, which can come from previous experience or by making assumptions, can reduce the uncertainity about a person's location. For example, we can assume that the construct of interest is normally distributed in the population and the person is sampled from a normal population. The essence of this Bayesian strategy is that one has person location information in terms of a probability distribution, known as prior distribution, prior to obtaining any observational data. The result of integratingthe prior distribution with the observational data is a distribution referred to as the posterior distribution.

```
de Ayala, R. J. (2013) Theory and Practice of Item Response Theory Methodology in the Socia
```



Log−likelihood of Latent Trait (theta)
LLtheta
theta
0
−5
−10
−15
−4
−2
0
2
4
4
maximum log−likelihood of theta

Figure 23:

```
posterior.plot <-
  function(i, Ltheta){
    #source("shadowtext.R")

    # Latent trait continium, person location (theta)
    theta <- seq(from = -4, to = 4, by = 0.01)

    plot(theta, Ltheta,
         type = "l",
         lwd = 6,
         xlim = c(min(theta) - diff(range(theta))*.05, max(theta) +
                    diff(range(theta))*.05),
         ylim = c(0, max(max(Ltheta), max(dnorm(theta, 0, 1))) + .05),
         col = "dodgerblue",
         ylab = "Likelihood",
         xlab = expression(Theta))

    # normal distribution
    lines(theta, dnorm(theta, 0, 1),
          lwd = 6,
          col = "tomato")

    # posterior (normal distribution * likelihood)
    posterior <- Ltheta*dnorm(theta, 0, 1)
    lines(theta, posterior,
          lwd = 6,
          col = "green")

    # to mark the maximum likelihood on the graph with coordinates
    # local maximum coordinates
    x1 <- subset(data.frame(theta, Ltheta), subset = Ltheta == max(Ltheta))[1]$theta
    y1 <- max(Ltheta)

    # tracing lines
    segments(x1, min(Ltheta) - diff(range(Ltheta))*.2, x1, y1,
             col = "dodgerblue",
```

```
         lty = 2,
         lwd = 1)
segments(x1, y1, min(theta) - diff(range(theta))*.2, y1,
         col = "dodgerblue",
         lty = 2,
         lwd = 1)

# max point and it's coordinates
points(x1, y1, pch = 21, col = "white", bg = "dodgerblue", cex = .8)
shadowtext(x1, y1, paste("(", round(x1, 2), ", ",
                         round(y1, 2), ")", sep = ""),
           cex = 1,
           pos = 3,
           col = "dodgerblue")

# to mark the maximum posterior on the graph with coordinates
# local maximum coordinates
x2 <- subset(data.frame(theta, posterior),
             subset = posterior == max(posterior))[1]$theta
y2 <- max(posterior)

# max point and it's coordinates
points(x2, y2, pch = 21, col = "white", bg = "green", cex = .8)

shadowtext(x2, y2,
           paste("(" , round(x2, 2), ", ",  round(y2, 4), ")", sep = ""),
           cex = 1,
           pos = 3,
           col = "green")

# tracing lines
segments(x2, min(Ltheta) - diff(range(Ltheta))*.2, x2, y2,
         col = "green",
         lty = 2,
         lwd = 1)
segments(x2, y2, min(theta) - diff(range(theta))*.2, y2,
```

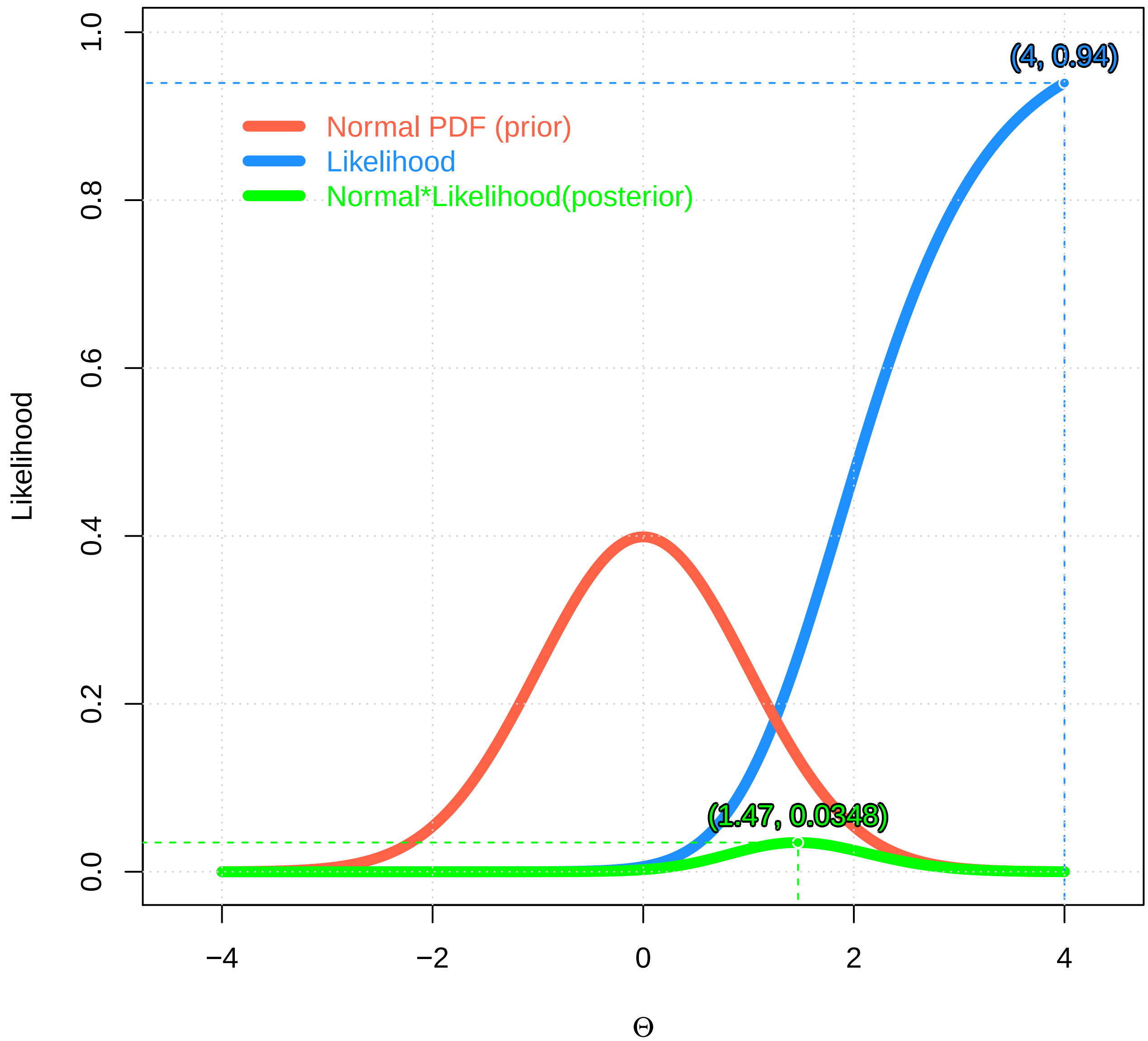
```r
             col = "green",
             lty = 2,
             lwd = 1)

    # gridlines
    grid(nx = NULL, ny = NULL, col = "lightgray",
         lty = "dotted",
         lwd = par("lwd"),
         equilogs = TRUE)

    # legend
    legend(-4, max(max(Ltheta), max(dnorm(theta, 0, 1))) - .01,
           c("Normal PDF (prior)", "Likelihood", "Normal*Likelihood(posterior)"),
           col = c("tomato", "dodgerblue", "green"),
           text.col = c("tomato", "dodgerblue", "green"),
           lty = 1,
           lwd = 6,
           bty = "n")

    return(posterior)
  } # end posterior.plot

dump("posterior.plot", file = "posterior.plot.R")
```

```r
posterior <- posterior.plot(1, LthetaRalph)
```

```r
# Ralph's location, theta (log-likelihood)
(thetaRalph1 <- subset(data.frame(theta, LLthetaRalph),
                       subset = LLthetaRalph == max(LLthetaRalph))[1]$theta)
```

```
## [1] 4
```

```r
# Ralph's location, theta (posterior, given normal prior)
(thetaRalph2 <- subset(data.frame(theta, posterior),
                       subset = posterior == max(posterior))[1]$theta)
```



(4, 0.94)
Normal PDF (prior)
Likelihood
Normal*Likelihood(posterior)
(1.47, 0.0348)
Likelihood
0.0
0.2
0.4
0.6
0.8
1.0
−4
−2
0
2
4
Θ

Figure 24:

```
## [1] 1.47
```

Ralph's Item Characteristic Curves (ICC)

```
IRF(par.mat, thetaRalph2, irf.plot = TRUE, trf.plot = TRUE)
```

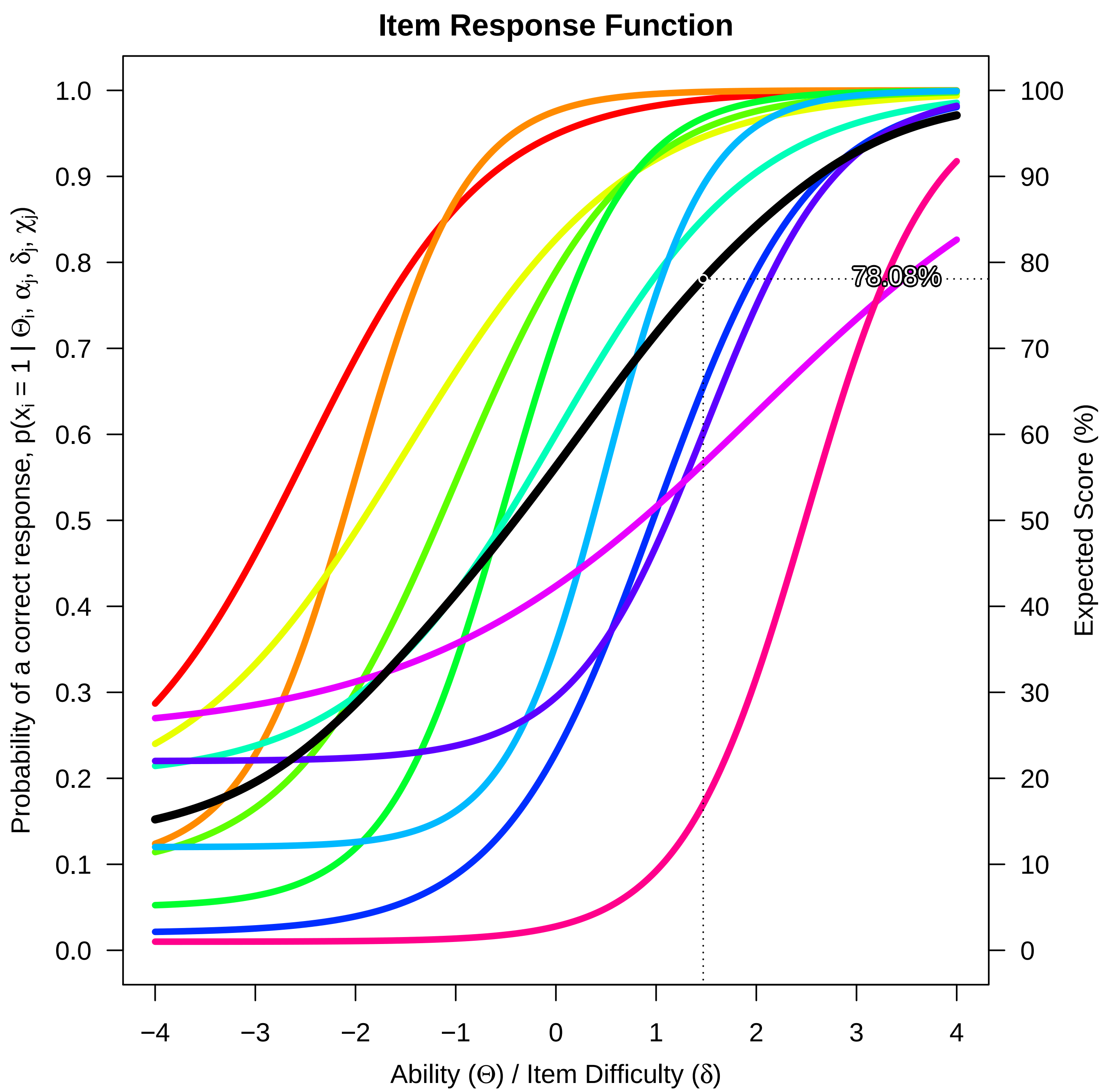


Figure 25:

```
## $probabilities
##         [,1]    [,2]    [,3]    [,4]    [,5]    [,6]    [,7]    [,8]
## [1,] 0.98935 0.99826 0.94575 0.95534 0.96777 0.85045 0.88943 0.65521
##         [,9]   [,10]   [,11]
## [1,] 0.60123 0.56587 0.16977
##
## $expected.score
## [1] 78.076
```

1.3.6 Percentile Rank

```
# Latent trait continium, person location (theta)
theta <- seq(from = -4, to = 4, by = 0.01)

# The probability density function (PDF) of standard normal distribution
normal.pdf <-
  function(x){
    return(exp(-(x^2/2))/sqrt(2*pi))
  }

no.pdf <- cbind(theta, normal.pdf(theta))

# The cumulative distribution function (CDF) of the standard normal distribution
normal.cdf <-
  function(x){
    z <- seq(from = -4, to = x , by = 0.01)
    return(round((sum(normal.pdf(z))/100), 4))
  }

# calculate CDF for the latent trait continuum
# The cumulative distribution function (CDF) of the standard normal distribution
# to hold CDF values of the latent trait continuum
no.cdf <- cbind(theta, rep(0.0, length(theta)))
for (i in 1:length(theta)) {
```

```
  no.cdf[i,2] <- normal.cdf(theta[i])
}
```

Plot CDF

```
plot(no.cdf,
     cex = .7,
     pch = 16,
     ylab = "Percentile Rank",
     col = adjustcolor("dodgerblue", alpha.f = 0.5))

lines(no.pdf,
      main = "Distribution of X1",
      col = adjustcolor("tomato", alpha.f = 0.5),
      lwd = 5)

person.theta <- 2

# shading
x <- c(seq(from = -4, to = person.theta, by = 0.01), person.theta, -4)
y <- c(normal.pdf(seq(from = -4, to = person.theta, by = 0.01)), 0, normal.pdf(-4))

polygon(x, y,
        border = adjustcolor("tomato", alpha.f = 0.3),
        col = adjustcolor("tomato", alpha.f = 0.3))

# tracing lines
x1 <- person.theta
y1 <- normal.cdf(x1)
segments(x1, -.5, x1, y1, col = "dodgerblue", lty = "dotted", lwd = 2)
segments(x1, y1, -4.5, y1, col = "dodgerblue", lty = "dotted", lwd = 2)
points(x1, y1, pch = 21, col = "tomato", bg = "yellow", cex = 1.1)

# text
shadowtext(-1, y1,
```

```
     paste("p = ", round(y1, 3)),
     col = "dodgerblue",
     bg = "black",
     pos = 3,
     cex = 1.2)

shadowtext(0, .1,
     paste("area = ", round(y1, 3)),
     col = "tomato",
     cex = 1.2)

shadowtext(-2, .4, "Standard Normal PDF", col = "tomato", pos = 3)
shadowtext(-.8, .9, "Standard Normal CDF", col = "dodgerblue")

# gridlines
grid(nx = NULL, ny = NULL,
     col = "lightgray",
     lty = "dotted",
     lwd = 1)
```

Suzy's Location Estimation

```
# Response Matrix
resp.mat <- matrix(c(1, 1, 1, 1, 1, 1, 1, 1, 1,
                     1, 1, 1, 1, 1, 0, 0, 0, 0,
                     1, 0, 0, 0, 0, 0, 0, 0, 0),
                   nrow = 3,
                   ncol = 9,
                   byrow = TRUE)

dimnames(resp.mat) <- list(c("Ralph", "Suzy", "Alice"),
                           colnames(resp.mat, do.NULL = FALSE, prefix = "i"))

knitr::kable(resp.mat, caption = 'Person response matrix.')
```

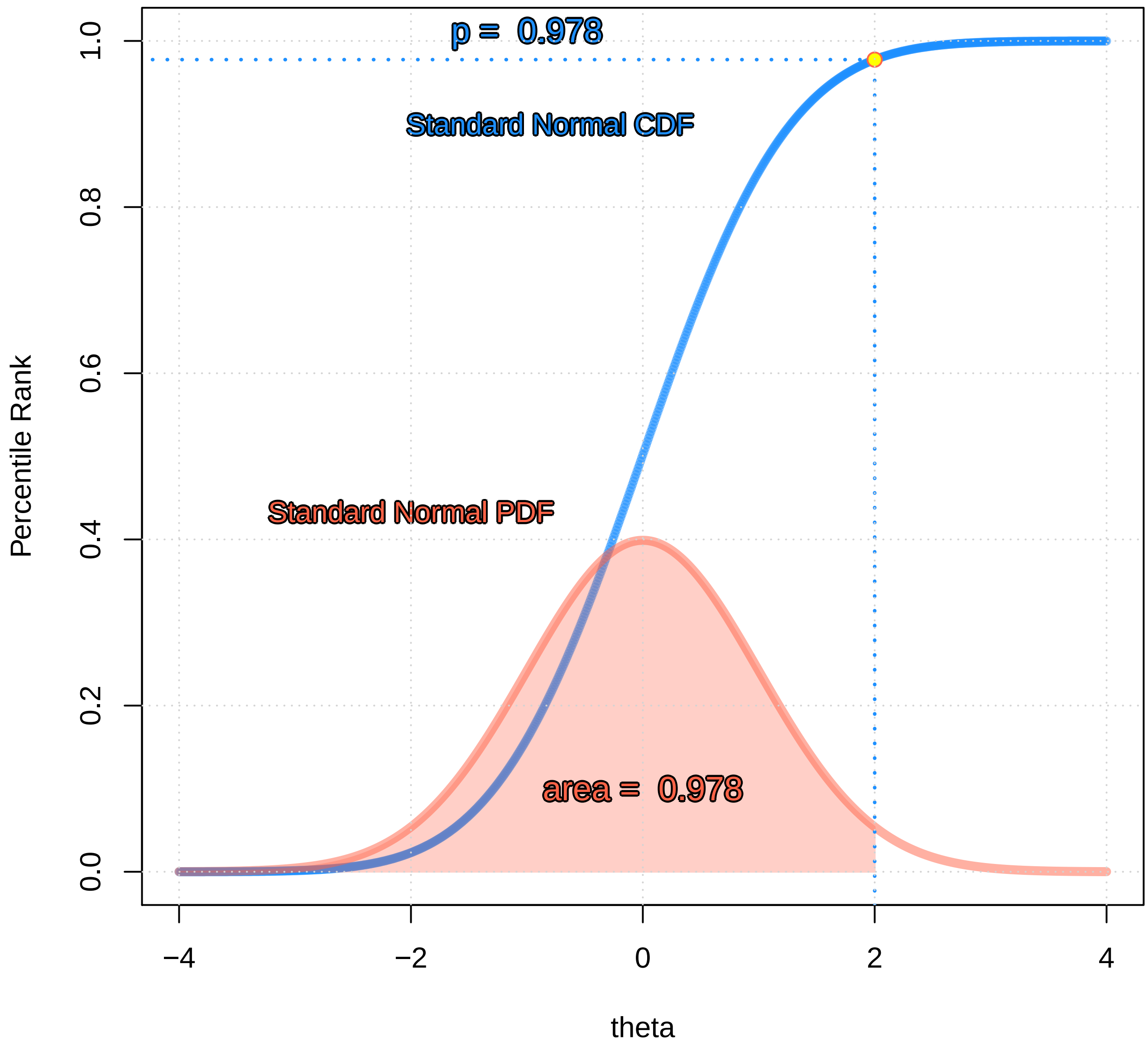

p = 0.978
Standard Normal CDF
Standard Normal PDF
area = 0.978
Percentile Rank
0.0
0.2
0.4
0.6
0.8
1.0
−4
−2
0
2
4
theta

Figure 26:

Table 12: Person response matrix.

	i1	i2	i3	i4	i5	i6	i7	i8	i9
Ralph	1	1	1	1	1	1	1	1	1
Suzy	1	1	1	1	1	0	0	0	0
Alice	1	0	0	0	0	0	0	0	0

```
# Suzy's response vector is (1, 1, 1, 1, 1, 0, 0, 0, 0), so
# the Likelihood function for Suzy is p1*p2*p3*p4*p5*q6*q7*q8*q9
LthetaSuzy <- likelihood(index = 2, resp.mat, par.mat)

# Log-likelihood function for Ralph
LLthetaSuzy <- log(LthetaSuzy)

# Plot Suzy's (2) Log-likelihood function
LLplot(2, LLthetaSuzy)
```

```
posterior <- posterior.plot(2, LthetaSuzy)
```

```
# Suzy's location, theta (log-likelihood)
(thetaSuzy1 <- subset(data.frame(theta,LLthetaSuzy),
                      subset = LLthetaSuzy == max(LLthetaSuzy))[1]$theta)
```

```
## [1] -0.2
```

```
# Suzy's location, theta (posterior, given normal prior)
(thetaSuzy2 <- subset(data.frame(theta, posterior),
                      subset = posterior == max(posterior))[1]$theta)
```

```
## [1] -0.14
```

```
# Item Characteristic Curves (ICC)
IRF(par.mat, thetaSuzy2, irf.plot = TRUE, trf.plot = TRUE)
```

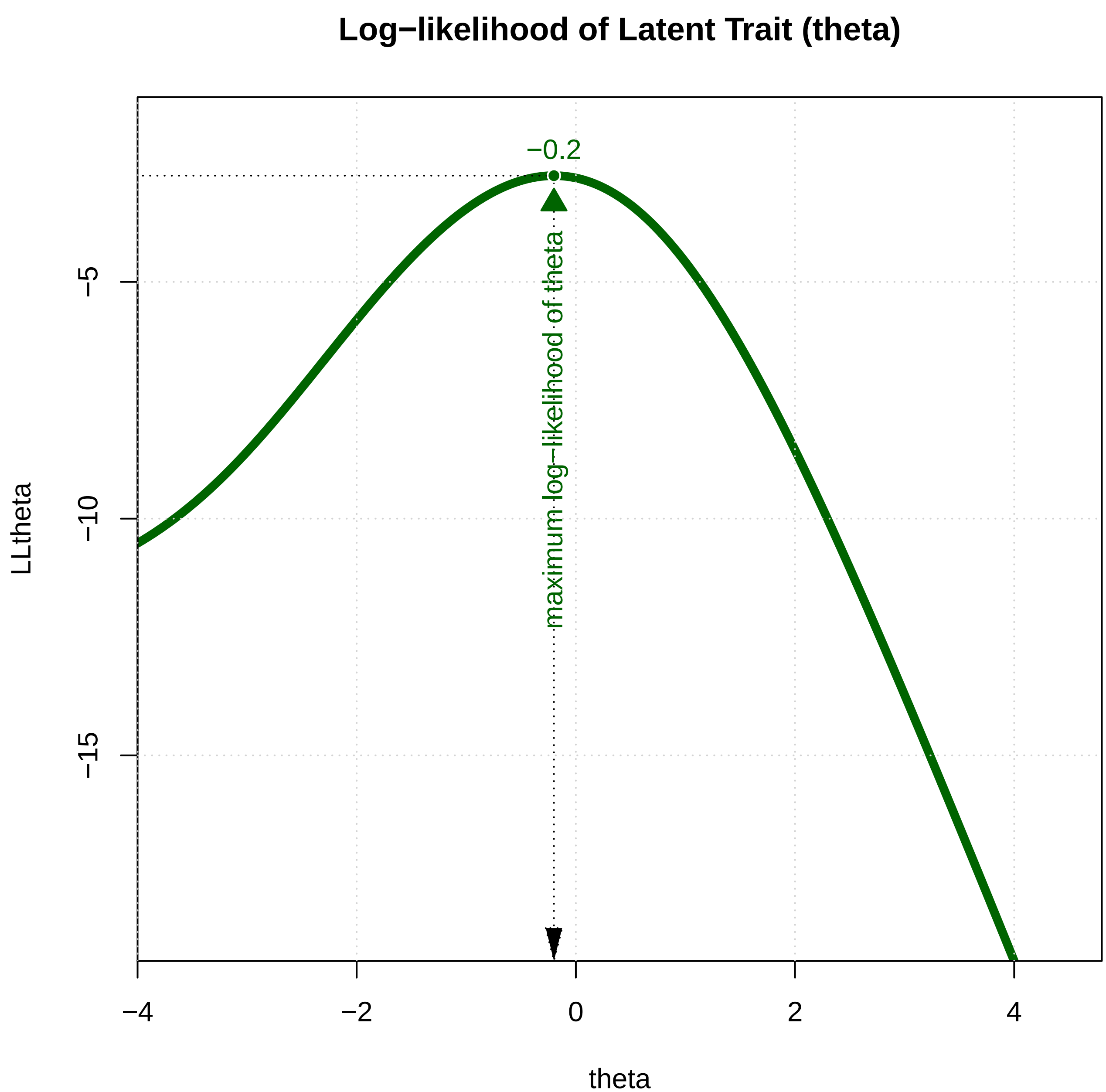

Log−likelihood of Latent Trait (theta)
−0.2
maximum log−likelihood of theta
LLtheta
−5
−10
−15
−4
−2
0
2
4
theta

Figure 27:

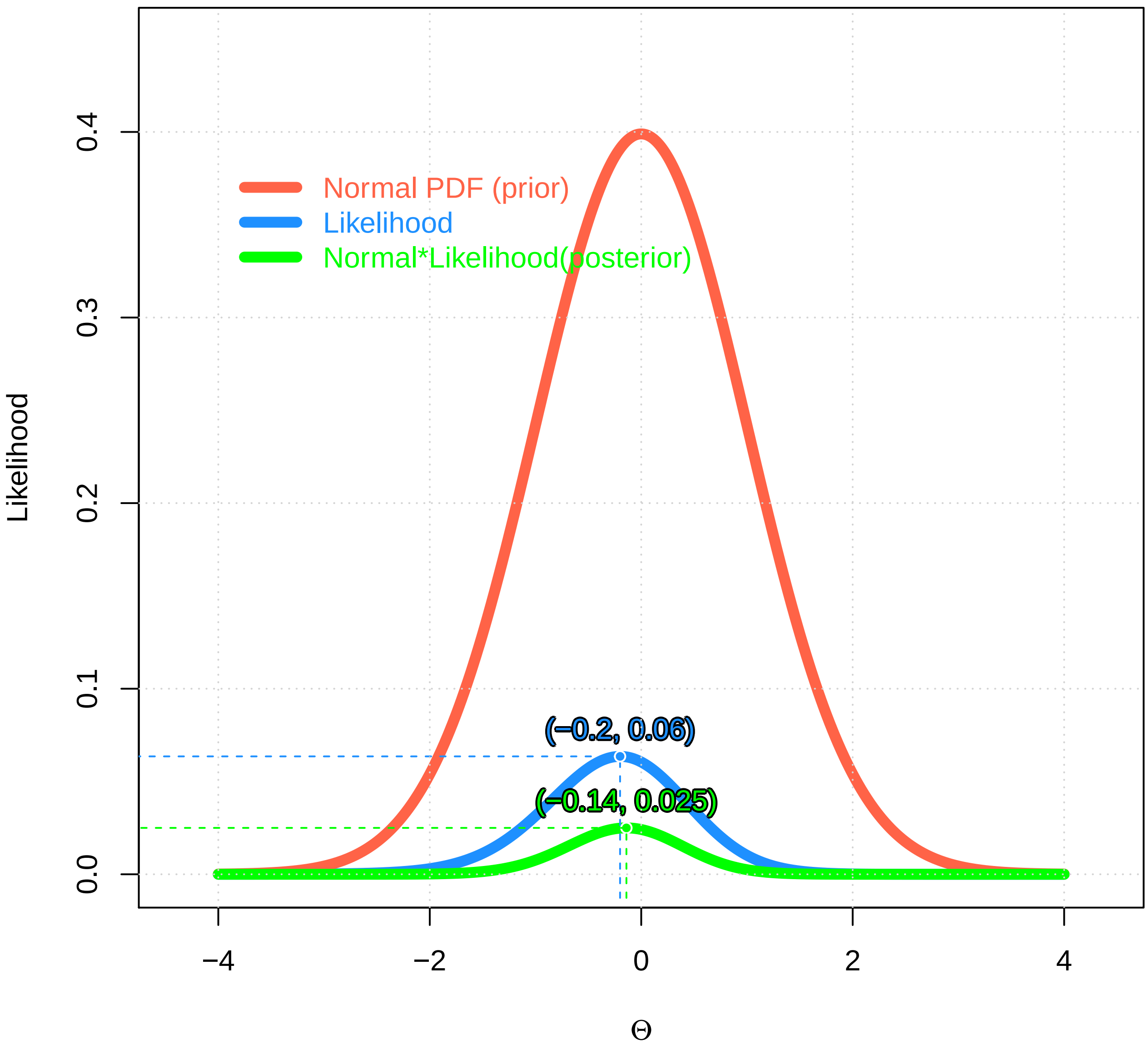

Normal PDF (prior)
Likelihood
Normal*Likelihood(posterior)
(−0.2, 0.06)
(−0.14, 0.025)
Likelihood
0.0
0.1
0.2
0.3
0.4
−4
−2
0
2
4
Θ

Figure 28:

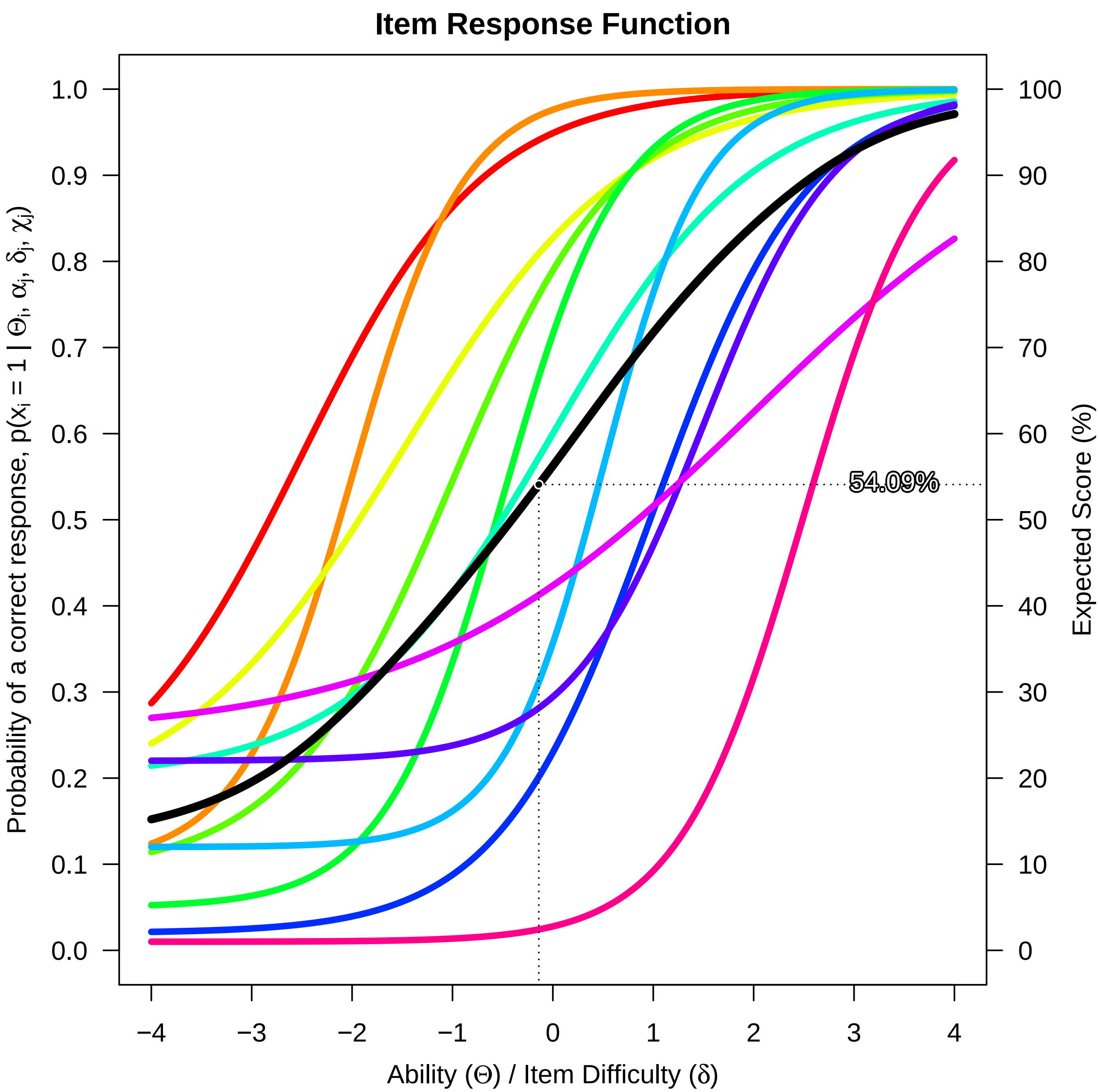

Item Response Function
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Probability of a correct response, p(xi = 1 | Θi, αj, δj, χj)
100
90
80
70
60
50
40
30
20
10
0
Expected Score (%)
54.09%
−4
−3
−2
−1
0
1
2
3
4
Ability (Θ) / Item Difficulty (δ)

Figure 29:

```
## $probabilities
##         [,1]    [,2]    [,3]    [,4]    [,5]    [,6]    [,7]    [,8]
## [1,] 0.94101 0.96944 0.80912 0.76095 0.66598 0.57205 0.31144 0.20142
##         [,9]   [,10]    [,11]
## [1,] 0.28139 0.41265 0.024284
##
## $expected.score
## [1] 54.089
```

```
# Suzy's percentile rank
sprintf("%1.2f%%", 100*normal.cdf(thetaSuzy2))
```

```
## [1] "44.63%"
```

The log likelihood function predicts that Suzy's location is at the far low-end of the latent trait continium ($\Theta = -4$). Where as the Bayesian strategy, which is by making use of the assumption that the latent trait Θ is normally distributed in the population, predict that Suzy's location is at ($\Theta = -2.4$).

Alice's Location Estimation

```
# Response Matrix
knitr::kable(resp.mat, caption = 'Person response matrix.')
```

Table 13: Person response matrix.

	i1	i2	i3	i4	i5	i6	i7	i8	i9
Ralph	1	1	1	1	1	1	1	1	1
Suzy	1	1	1	1	1	0	0	0	0
Alice	1	0	0	0	0	0	0	0	0

```
# Alice's response vector is (1, 0, 0, 0, 0, 0, 0, 0, 0), so
# the Likelihood function for Alice is p1*q2*q3*q4*q5*q6*q7*q8*q9
LthetaAlice <- likelihood(index = 3, resp.mat, par.mat)
```

```
# Log-likelihood function for Alice
LLthetaAlice <- log(LthetaAlice)

# Plot Alice's (3) Log-likelihood function
LLplot(3, LLthetaAlice)
```

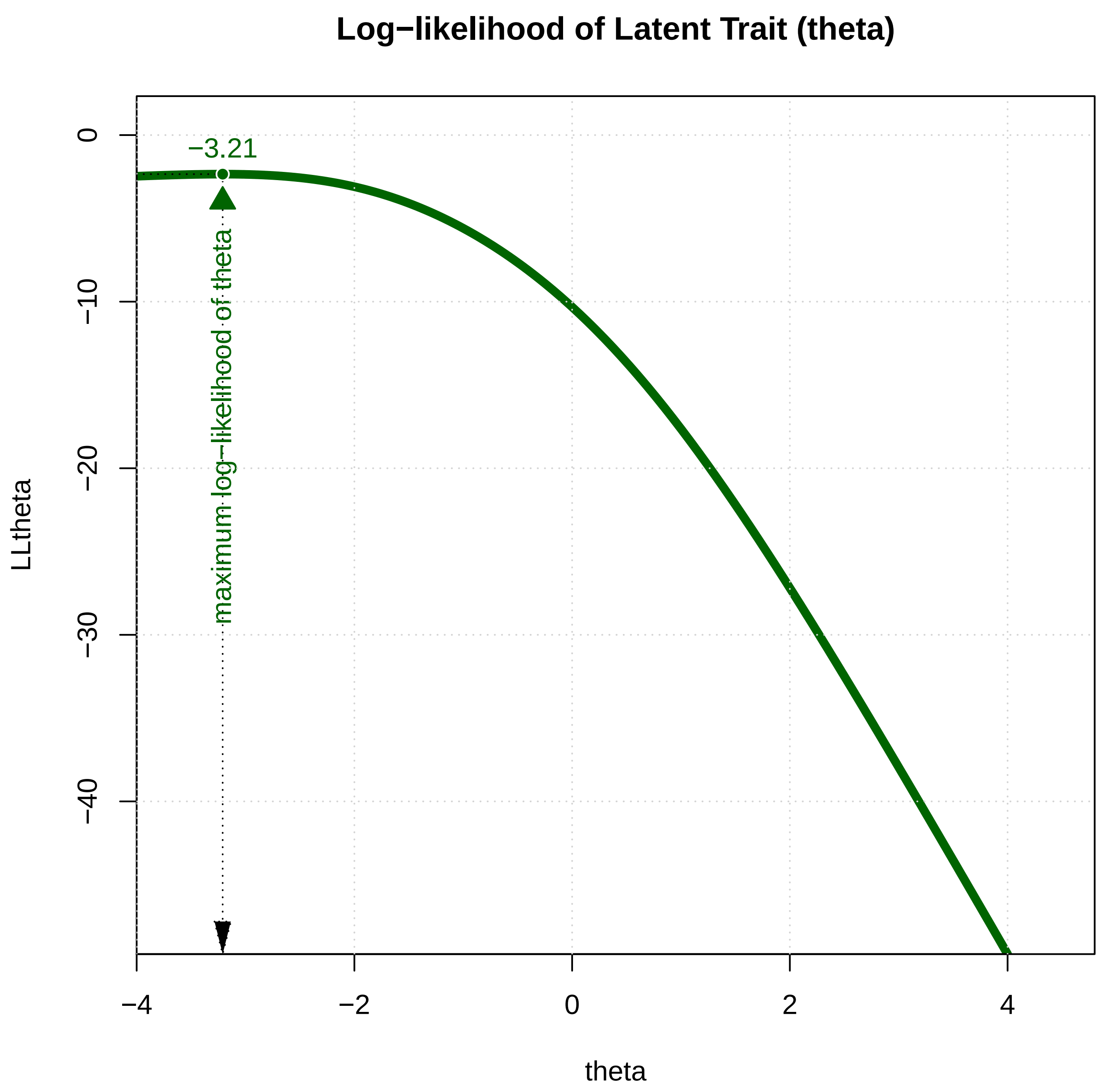


Figure 30:

```
posterior <- posterior.plot(3, LthetaAlice)
```

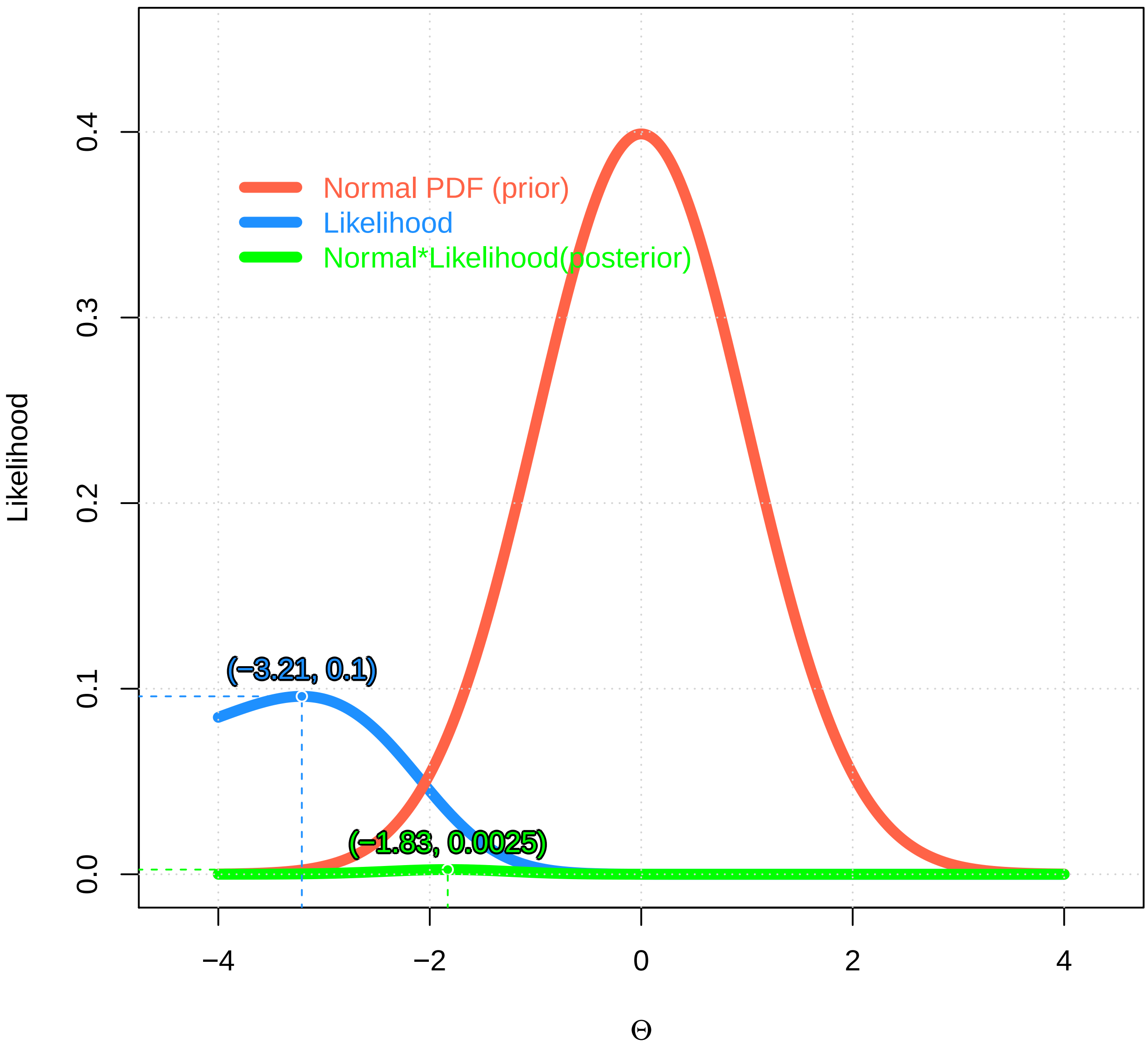

Normal PDF (prior)
Likelihood
Normal*Likelihood(posterior)
(−3.21, 0.1)
(−1.83, 0.0025)
Likelihood
0.0
0.1
0.2
0.3
0.4
−4
−2
0
2
4
Θ

Figure 31:

```r
# Alice's location, theta (log-likelihood)
(thetaAlice1 <- subset(data.frame(theta,LLthetaAlice),
                       subset = LLthetaAlice == max(LLthetaAlice))[1]$theta)
```

```
## [1] -3.21
```

```r
# Alice's location, theta (posterior, given normal prior)
(thetaAlice2 <- subset(data.frame(theta, posterior),
                       subset = posterior == max(posterior))[1]$theta)
```

```
## [1] -1.83
```

```r
# Item Characteristic Curves (ICC)
IRF(par.mat, thetaAlice2, irf.plot = TRUE, trf.plot = TRUE)
```

```
## $probabilities
##         [,1]    [,2]    [,3]    [,4]    [,5]    [,6]    [,7]     [,8]
## [1,] 0.72489 0.61832 0.51808 0.33545 0.13968 0.31059 0.12825 0.044134
##         [,9]   [,10]    [,11]
## [1,] 0.22525 0.31847 0.010969
##
## $expected.score
## [1] 30.674
```

```r
# Alice's percentile rank
sprintf("%1.2f%%", 100*normal.cdf(thetaAlice2))
```

```
## [1] "3.40%"
```

Graph the item information function for each of the items in the previous question.

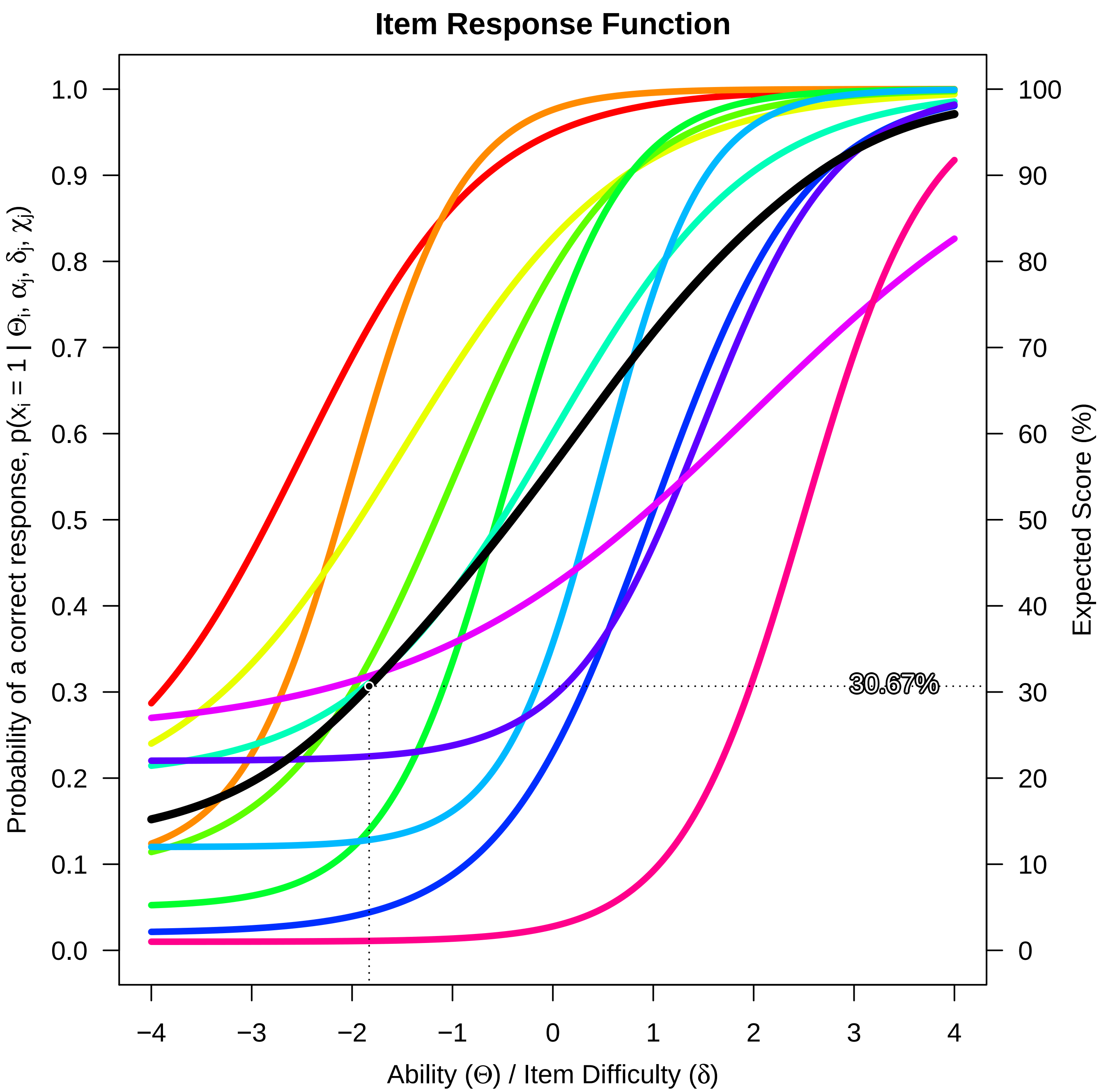

Item Response Function
Probability of a correct response, $p(x_i = 1 \mid \Theta_i, \alpha_j, \delta_j, \chi_j)$
Expected Score (%)
Ability (Θ) / Item Difficulty (δ)
30.67%
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
0
10
20
30
40
50
60
70
80
90
100
−4
−3
−2
−1
0
1
2
3
4

Figure 32:

```r
# Test information function plot
plot(irtoys::tif(par.mat),
    co = "black",
    main = "Test (black) and 9 Item information functions")

# Item information function plot
plot(irtoys::iif(par.mat),
     label = TRUE,
     add = TRUE,
     co = NA)

abline(v = par.mat[,"b"],
       lty = 2,
       col = palette())

text(par.mat[,"b"], .5,
     labels = par.mat[,"b"],
     srt = 90,
     pos = 3,
     col = palette())

# gridlines
grid(nx = NULL, ny = NULL,
     col = "lightgray",
     lty = "dotted",
     lwd = par("lwd"),
     equilogs = TRUE)
```

1.4 Checking IRT Model Assumptions

1.4.1 Assessing Conditional Independence, Q_3

The Theory and Practice of Item Response Theory (Ayala 2009):

Q_3 is the correlations between the residuals for a pair of items. The residual for an item is the

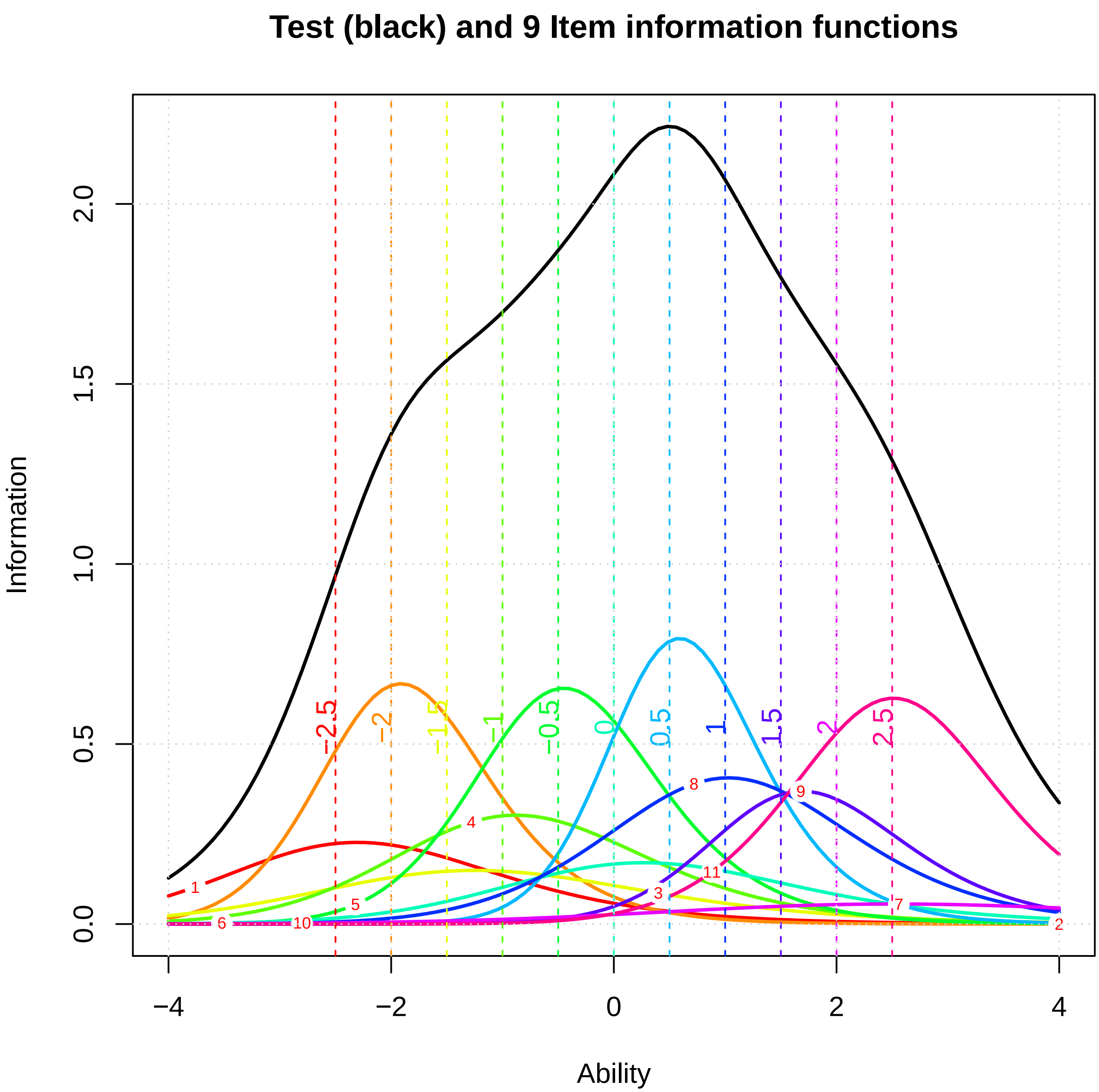

Test (black) and 9 Item information functions
Information
Ability
0.0
0.5
1.0
1.5
2.0
−4
−2
0
2
4
−2.5
−2
−1.5
−1
−0.5
0
0.5
1
1.5
2
2.5
1
2
3
4
5
6
7
8
9
10
11

Figure 33:

difference between an individual's observed reponse (0 or 1) and his or her expected response on that particular item.

```
# Import data
responses <- read.csv("MidTerm.csv", header = TRUE, na.strings = " ")

# Check
str(responses)
```

```
## 'data.frame':    450 obs. of  15 variables:
##  $ V1 : int  0 0 0 1 0 0 1 0 0 0 ...
##  $ V2 : int  0 1 1 1 1 1 1 1 0 1 ...
##  $ V3 : int  1 1 1 1 1 0 1 0 0 1 ...
##  $ V4 : int  0 1 0 0 0 0 1 0 0 1 ...
##  $ V5 : int  1 0 0 0 0 1 0 1 1 0 ...
##  $ V6 : int  0 0 1 1 1 1 1 0 0 1 ...
##  $ V7 : int  0 1 1 1 1 1 1 0 0 1 ...
##  $ V8 : int  0 0 1 1 1 0 1 0 0 1 ...
##  $ V9 : int  0 0 1 1 1 0 1 0 0 1 ...
##  $ V10: int  1 1 1 1 1 1 1 0 1 1 ...
##  $ V11: int  0 1 1 1 1 1 1 1 1 1 ...
##  $ V12: int  0 1 0 0 0 0 1 0 0 1 ...
##  $ V13: int  0 0 1 1 0 0 1 0 0 1 ...
##  $ V14: int  0 0 0 1 0 0 1 0 0 1 ...
##  $ V15: int  1 1 1 1 1 0 1 0 1 1 ...
```

If $|Q3|$ equals 1.0 then the two items are perfectly dependent. In contrast, a $Q3$ of 0.0 is a necessary, but not sufficient condition for independence because it may also mean that items in the pair exhibit a nonlinear relationship. A large $|Q3|$ value for an item pair means that items have similar parameters and share one or unique dimensions. Therefore, similarity of parameters for the item pairs is a necessary, but not sufficient condition for obtaining a large $Q3$. The square of $Q3$ may be interpreted as a measure of the amount of residual variance shared by an item pair. Therefore, item pairs with a large proportion of shared variability (e.g. 0.05) would indicate dependent items.

```
Q3 <-
  function(responses,
           model = NULL,
```

```
        parameters = NULL,
        thetas = NULL,
        probabilities = NULL){

  require(irtoys,
      quietly = TRUE,
      warn.conflicts = FALSE)

  source("logistic.R")

  responses <- as.matrix(responses)

  # the default model: 2PL
  if (is.null(model)) {
    model <- "2PL"
  }

  if (is.null(probabilities)) {
    if (is.null(parameters)) {
      # Estimate item parameters for IRT
      item.parameters <- irtoys::est(resp = responses,
                                     model = model,
                                     engine = "ltm")$est
    }

    if (is.null(thetas)) {
      # EAP estimation of ability
      thetas <- irtoys::eap(resp = responses,
                            ip = item.parameters,
                            qu = normal.qu())[,"est"]
    }

    # compute expected response
    pj <- logistic(alpha = item.parameters[,1],
                   delta = item.parameters[,2],
                   chi = item.parameters[,3],
```

```
                        theta = thetas)
    }

    # compute residuals
    dij <- responses - pj

    # compute the correlations between the residuals for a pair of items
    rq3 <- (cor(dij))^2

    flag <- TRUE
    for (i in 1:nrow(rq3)) {
      for (j in i:ncol(rq3)) {
        if (rq3[i,j] > 0.2 && rq3[i,j] < 1.0) {
          print(paste("items", i, "and", j, "rq3 = ", round(rq3[i,j], 2)))
          flag <- FALSE
        }
      }
    }
    if (flag) {print("There aren't any item-pairs with larger than 0.2 residual correlatio
  }# end Q3

dump("Q3", file = "Q3.R")
```

The list of item pairs with a large amount of shared residual variance ($Q3^2 > 0.2$) :.

```
Q3(responses)
```

```
## [1] "items 1 and 2 rq3 =  0.25"
## [1] "items 1 and 3 rq3 =  0.2"
## [1] "items 1 and 4 rq3 =  0.25"
## [1] "items 1 and 7 rq3 =  0.2"
## [1] "items 1 and 9 rq3 =  0.22"
## [1] "items 1 and 10 rq3 =  0.36"
## [1] "items 1 and 11 rq3 =  0.24"
## [1] "items 1 and 12 rq3 =  0.25"
## [1] "items 1 and 13 rq3 =  0.26"
```

```
## [1] "items 1 and 14 rq3 =  0.37"
## [1] "items 2 and 3 rq3 =  0.33"
## [1] "items 2 and 4 rq3 =  0.21"
## [1] "items 2 and 6 rq3 =  0.27"
## [1] "items 2 and 7 rq3 =  0.32"
## [1] "items 2 and 8 rq3 =  0.24"
## [1] "items 2 and 10 rq3 =  0.49"
## [1] "items 2 and 11 rq3 =  0.38"
## [1] "items 2 and 15 rq3 =  0.24"
## [1] "items 3 and 6 rq3 =  0.25"
## [1] "items 3 and 7 rq3 =  0.29"
## [1] "items 3 and 8 rq3 =  0.21"
## [1] "items 3 and 10 rq3 =  0.39"
## [1] "items 3 and 11 rq3 =  0.34"
## [1] "items 3 and 15 rq3 =  0.31"
## [1] "items 4 and 10 rq3 =  0.31"
## [1] "items 4 and 11 rq3 =  0.22"
## [1] "items 4 and 12 rq3 =  0.24"
## [1] "items 4 and 14 rq3 =  0.2"
## [1] "items 5 and 10 rq3 =  0.32"
## [1] "items 6 and 8 rq3 =  0.27"
## [1] "items 6 and 9 rq3 =  0.26"
## [1] "items 6 and 10 rq3 =  0.25"
## [1] "items 6 and 11 rq3 =  0.23"
## [1] "items 6 and 14 rq3 =  0.21"
## [1] "items 6 and 15 rq3 =  0.23"
## [1] "items 7 and 10 rq3 =  0.37"
## [1] "items 7 and 11 rq3 =  0.32"
## [1] "items 8 and 9 rq3 =  0.26"
## [1] "items 8 and 10 rq3 =  0.22"
## [1] "items 8 and 11 rq3 =  0.28"
## [1] "items 8 and 15 rq3 =  0.23"
## [1] "items 9 and 14 rq3 =  0.23"
## [1] "items 9 and 15 rq3 =  0.22"
## [1] "items 10 and 11 rq3 =  0.47"
## [1] "items 10 and 12 rq3 =  0.29"
```

```
## [1] "items 10 and 13 rq3 =  0.21"
## [1] "items 10 and 14 rq3 =  0.24"
## [1] "items 10 and 15 rq3 =  0.28"
## [1] "items 11 and 12 rq3 =  0.21"
## [1] "items 11 and 15 rq3 =  0.27"
## [1] "items 13 and 14 rq3 =  0.2"
```

1.4.2 Assessing the unidimensionality assumption

Unidimensionality assumption states that the observations on the manifest variables (items or questions) are solely a function of a single continious latent person variable (ability). For example, in a mathematics test it is assumed that there is a single latent mathematics proficiency variable that underlies the respondents' perfomance.

```
# Import data file
# A data frame of responses: persons (450) as rows, items (15) as columns,
# entries are either 0 or 1, no missing values
mytest <- read.csv("test.csv", header = TRUE, na.strings = " ")

# CONSTANTS
(npersons <- nrow(mytest))
```

```
## [1] 450
```

```
(nitems <- ncol(mytest))
```

```
## [1] 15
```

Polychoric correlation is a technique for estimating the correlation between two theorised normally distributed continuous latent variables, from two observed ordinal variables. Tetrachoric correlation is a special case of the polychoric correlation applicable when both observed variables are dichotomous. The tetrachoric correlation is the inferred Pearson Correlation from a two x two table with the assumption of bivariate normality.

```
mytest.tetra <- psych::tetrachoric(mytest,
                                   correct = TRUE,
                                   smooth = TRUE,
                                   global = TRUE)
```

```
tetmat <- mytest.tetra$rho
knitr::kable(round(tetmat, 1),
             caption = 'Tetrachoric correlations')
```

Table 14: Tetrachoric correlations

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15
V1	1.0	0.4	0.4	0.3	-0.6	0.6	0.3	0.6	0.7	0.2	0.4	0.4	0.5	0.7	0.5
V2	0.4	1.0	0.6	0.3	-0.5	0.7	0.5	0.6	0.6	0.7	0.6	0.2	0.4	0.5	0.5
V3	0.4	0.6	1.0	0.3	-0.6	0.6	0.6	0.6	0.7	0.6	0.6	0.3	0.4	0.7	0.7
V4	0.3	0.3	0.3	1.0	-0.3	0.4	0.3	0.5	0.4	0.1	0.4	0.4	0.2	0.4	0.3
V5	-0.6	-0.5	-0.6	-0.3	1.0	-0.6	-0.4	-0.6	-0.6	-0.6	-0.5	-0.5	-0.3	-0.6	-0.6
V6	0.6	0.7	0.6	0.4	-0.6	1.0	0.4	0.7	0.7	0.7	0.6	0.4	0.4	0.7	0.6
V7	0.3	0.5	0.6	0.3	-0.4	0.4	1.0	0.5	0.5	0.5	0.6	0.2	0.3	0.3	0.4
V8	0.6	0.6	0.6	0.5	-0.6	0.7	0.5	1.0	0.7	0.5	0.7	0.3	0.4	0.7	0.6
V9	0.7	0.6	0.7	0.4	-0.6	0.7	0.5	0.7	1.0	0.4	0.6	0.3	0.5	0.7	0.7
V10	0.2	0.7	0.6	0.1	-0.6	0.7	0.5	0.5	0.4	1.0	0.6	0.4	0.1	0.4	0.7
V11	0.4	0.6	0.6	0.4	-0.5	0.6	0.6	0.7	0.6	0.6	1.0	0.4	0.4	0.5	0.6
V12	0.4	0.2	0.3	0.4	-0.5	0.4	0.2	0.3	0.3	0.4	0.4	1.0	0.4	0.4	0.4
V13	0.5	0.4	0.4	0.2	-0.3	0.4	0.3	0.4	0.5	0.1	0.4	0.4	1.0	0.5	0.4
V14	0.7	0.5	0.7	0.4	-0.6	0.7	0.3	0.7	0.7	0.4	0.5	0.4	0.5	1.0	0.5
V15	0.5	0.5	0.7	0.3	-0.6	0.6	0.4	0.6	0.7	0.7	0.6	0.4	0.4	0.5	1.0

```
# Standard Deviations and Variances
tetfit <- stats::princomp(covmat = tetmat, cor = TRUE)
knitr::kable(cbind(sd = tetfit$sdev, var = (tetfit$sdev)^2),
             caption = 'Standard Deviations and Variances')
```

Table 15: Standard Deviations and Variances

	sd	var
Comp.1	2.82694	7.99157
Comp.2	1.17762	1.38680
Comp.3	0.94629	0.89547
Comp.4	0.91691	0.84072

	sd	var
Comp.5	0.87313	0.76236
Comp.6	0.73275	0.53693
Comp.7	0.72606	0.52716
Comp.8	0.71053	0.50485
Comp.9	0.60399	0.36480
Comp.10	0.55761	0.31093
Comp.11	0.52681	0.27753
Comp.12	0.50082	0.25082
Comp.13	0.46244	0.21386
Comp.14	0.33710	0.11364
Comp.15	0.15029	0.02259

1.4.3 Screeplot

Plots the variances against the number of the principal component.

```
screeplot(tetfit,
          type = "lines",
          col = "tomato",
          pch = 21,
          bg = "dodgerblue",
          lwd = 3,
          ylim = c(0, ceiling(max(tetfit$sdev*tetfit$sdev))))
```

According to the scree plot, it appears that the items are unidimensional.

1.4.4 A Parallel Analysis With Randomly Generated Polychoric Correlation Matrices

The function performs a parallel analysis using simulated polychoric correlation matrices. The eigenvalues (extracted following both FA and PCA methods) from each random generated polychoric correlation matrix and from the polychoric correlation matrix of real solutions from Polychorich vs Pearson correlations, FA vs PCA and PA vs MAP are presented. Random data sets are simulated assuming or a uniform or a multinomial distribution or via the bootstrap method of resampling (i.e., random permutations of cases).

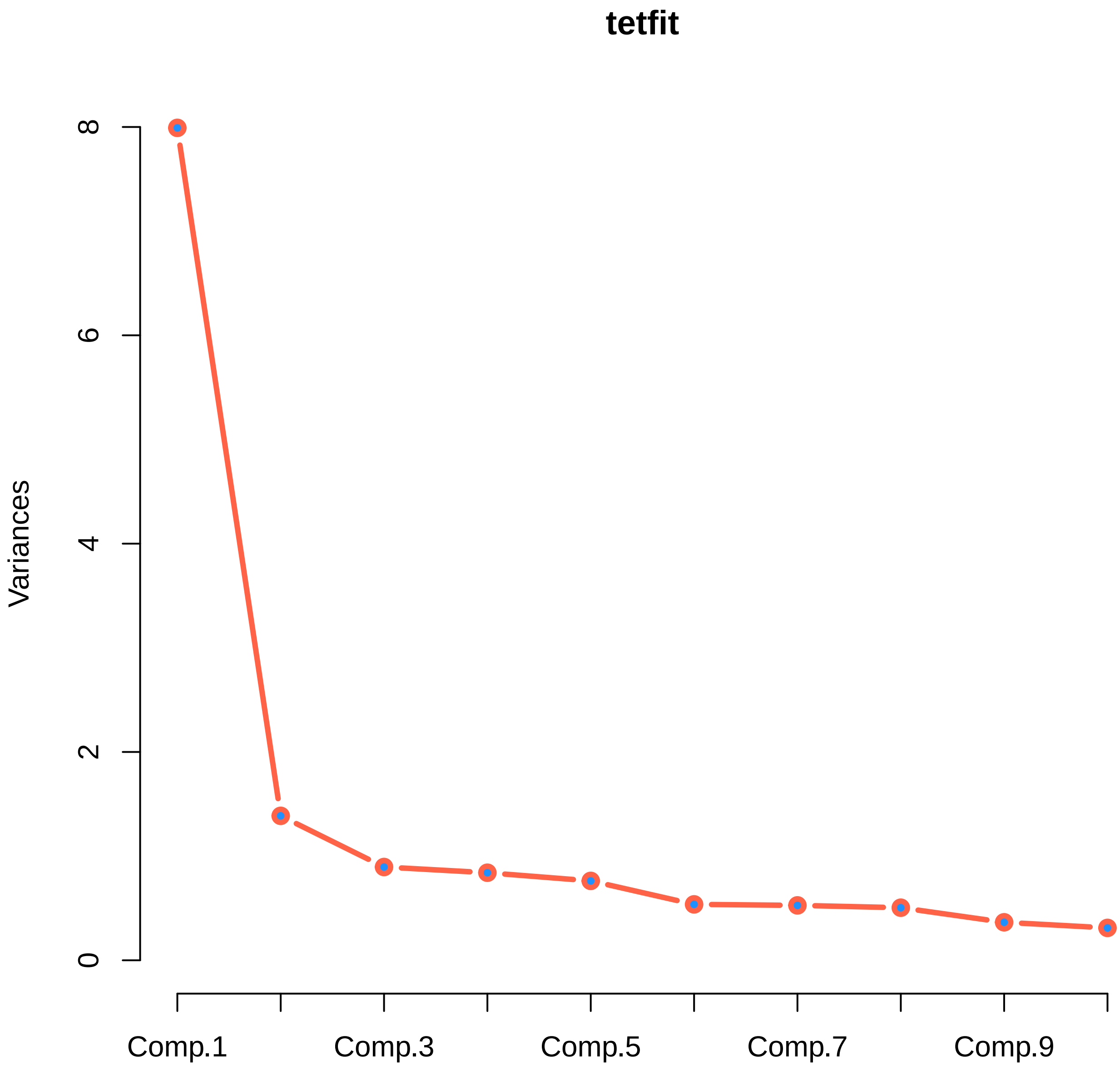


Figure 34:

The raw data.matrix should be numeric and none of the ordered category should be coded as 0 (zero). No automatic recode routine is provided within the function to deal with alphanumeric content of the ordered categories of manifest variables. So the user performs all these recodings before running the function.

```
require(random.polychor.pa, quietly = TRUE)
```

```
##
## Attaching package: 'psych'
##
## The following object is masked _by_ '.GlobalEnv':
##
##     logistic
##
## The following object is masked from 'package:eRm':
##
##     sim.rasch
##
## The following object is masked from 'package:irtoys':
##
##     sim
##
## The following object is masked from 'package:ltm':
##
##     factor.scores
##
## The following object is masked from 'package:polycor':
##
##     polyserial
##
##
## Attaching package: 'boot'
##
## The following object is masked from 'package:psych':
##
##     logit
##
```

```
## The following object is masked from 'package:msm':
##
##     cav
##
## The following object is masked from 'package:sm':
##
##     dogs
##
##
## Attaching package: 'lattice'
##
## The following object is masked from 'package:boot':
##
##     melanoma
##
##
## Attaching package: 'nFactors'
##
## The following object is masked from 'package:lattice':
##
##     parallel
```

```
# convert 0 to 2
head(mytest)
```

```
##   V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15
## 1  0  0  1  0  1  0  0  0  0   1   0   0   0   0   1
## 2  0  1  1  1  0  0  1  0  0   1   1   1   0   0   1
## 3  0  1  1  0  0  1  1  1  1   1   1   0   1   0   1
## 4  1  1  1  0  0  1  1  1  1   1   1   0   1   1   1
## 5  0  1  1  0  0  1  1  1  1   1   1   0   0   0   1
## 6  0  1  0  0  1  1  1  0  0   1   1   0   0   0   0
```

```
mydat <- ifelse(mytest == 0, 2, 1)
head(mydat)
```

```
##      V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15
```

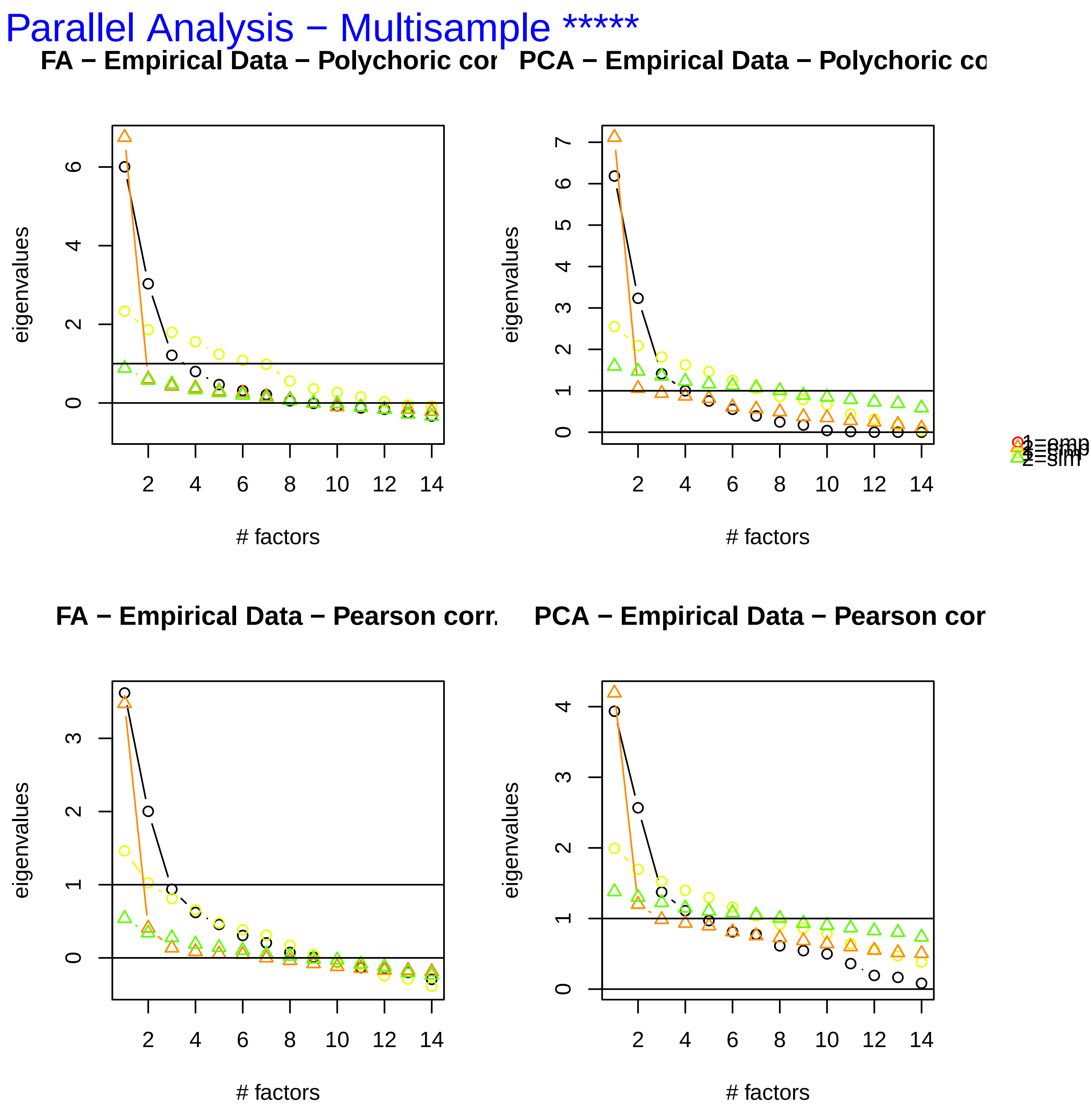
```
## [1,]  2  2  1  2  1  2  2  2  2   1   2   2   2   2   1
## [2,]  2  1  1  1  2  2  1  2  2   1   1   1   2   2   1
## [3,]  2  1  1  2  2  1  1  1  1   1   1   2   1   2   1
## [4,]  1  1  1  2  2  1  1  1  1   1   1   2   1   1   1
## [5,]  2  1  1  2  2  1  1  1  1   1   1   2   2   2   1
## [6,]  2  1  2  2  1  1  1  2  2   1   1   2   2   2   2
```

```
# Parallel Analysis With Randomly Generated Polychoric Correlation Matrices
random.polychor.pa(nvar = numitems,
                   n.ss = nsize,
                   nrep = 5,
                   nstep = "NULL",
                   data.matrix = mydat,
                   q.eigen = .99,
                   r.seed = "1959",
                   comparison = c("random-mg"),
                   distr = "uniform",
                   print.all = F)
```

```
## ***** RESULTS FOR PARALLEL ANALYSIS *****
## *** computation starts at: 18:24:00
## *** number of units (rows) in data.matrix: 450
## *** No missing values found
## *** Number of samples and size of samples to be compared:
##
##   1   2
##  62 388
##
## *** simulation method: RANDOM
## *** distribution: UNIFORM
## *** difficulty factor: FALSE
## *** number of variables (cols) in data.matrix: 14
## *** Groups of items with diffent number of categories found in your data.matrix:
##         Items Categories Min.Cat Max.Cat
## 1 GROUP    14          2       1       2
##
## The first simulation for FA took: 0.187 secs.
```

```
## The first simulation for PCA took: 1.048 secs.
## computation ended at: 18:24:02
## Elapsed Time: 0 min
## computation ended at: 18:24:04
## Elapsed Time: 0 min
##
##  Comparison between MULTI-GROUP RANDOM eigenvalues and EMPIRICAL eigenvalues
##
## ******* RESULTS for PARALLEL ANALYSIS:
##                                                                   sample.1
## # of factors (PCA) for Velicer MAP criterium (Pearson corr)...:         2
## # of factors (PCA) for Velicer MAP(4th power)(Polychoric corr):         2
## # of factors (PCA) for Velicer MAP criterium (Polychoric corr):         2
## # of factors (PCA) for Velicer MAP(4th power)(Pearson corr)...:         2
## # of factors (PCA) for PA method (Polychoric Corr.)..........:         2
## # of factors (PCA) for PA method (Pearson Corr.).............:         2
## # of factors for PA method (Polychoric Corr.)................:         2
## # of factors for PA method (Pearson Corr.)...................:         2
##                                                                   sample.2
## # of factors (PCA) for Velicer MAP criterium (Pearson corr)...:         1
## # of factors (PCA) for Velicer MAP(4th power)(Polychoric corr):         1
## # of factors (PCA) for Velicer MAP criterium (Polychoric corr):         1
## # of factors (PCA) for Velicer MAP(4th power)(Pearson corr)...:         1
## # of factors (PCA) for PA method (Polychoric Corr.)..........:         2
## # of factors (PCA) for PA method (Pearson Corr.).............:         2
## # of factors for PA method (Polychoric Corr.)................:         2
## # of factors for PA method (Pearson Corr.)...................:         2
```

With Pearson correlations and either PCA or PAF, it appears that there are two factors or dimensions using parallel analysis. However, with PCA, tetrachoric correlations, and Velicer's MAP procedure there is only one factor. As a result, dimensionality is not perfectly clear and the unidimensionality assumption may have been compromised.



Parallel Analysis − Multisample *****
FA − Empirical Data − Polychoric cor
PCA − Empirical Data − Polychoric co
FA − Empirical Data − Pearson corr.
PCA − Empirical Data − Pearson cor
eigenvalues
factors

Figure 35:

1.4.5 Horn's Parallel Analysis of Principal Components/Factors

Principal component analysis (PCA): PCA is the first phase of Exploratory factor analysis (EFA). Factor weights are computed in order to extract the maximum possible variance, with successive factoring continuing until there is no further meaningful variance left. The factor model must then be rotated for analysis.

```
require(paran, quietly = TRUE)
pca <- paran(mytest,
             centile = 95,
             cfa = FALSE,
             seed = 1804, # Go Bobcats!
             iterations = 500,
             graph = TRUE,
             color = TRUE,
             all = TRUE)
```

```
##
## Using eigendecomposition of correlation matrix.
## Computing: 10%  20%  30%  40%  50%  60%  70%  80%  90%  100%
##
##
## Results of Horn's Parallel Analysis for component retention
## 500 iterations, using the 95 centile estimate
##
## --------------------------------------------------
## Component   Adjusted    Unadjusted    Estimated
##             Eigenvalue  Eigenvalue    Bias
## --------------------------------------------------
## 1           4.275141    4.665540      0.390398
## 2           1.238006    1.536593      0.298587
## 3           0.703774    0.939297      0.235523
## 4           0.690833    0.873593      0.182759
## 5           0.712151    0.848790      0.136639
## 6           0.697517    0.797882      0.100364
## 7           0.674323    0.733146      0.058822
## 8           0.674797    0.694346      0.019548
```

```
## 9           0.684268    0.668929     -0.01533
## 10          0.661645    0.611294     -0.05035
## 11          0.656336    0.571213     -0.08512
## 12          0.676399    0.554864     -0.12153
## 13          0.683271    0.524573     -0.15869
## 14          0.692764    0.495777     -0.19698
## 15          0.726597    0.484155     -0.24244
## --------------------------------------------------
##
## Adjusted eigenvalues > 1 indicate dimensions to retain.
## (2 components retained)
```

Principal component analysis (PCA) suggests that there are two components (dimensions) in the data. That is the instrument is measuring two things at the same time which is a violation of IRT assumptions.

1.4.6 Principal Axis Factoring

Common factor analysis, also called principal factor analysis (PFA) or principal axis factoring (PAF), seeks the least number of factors which can account for the common variance (correlation) of a set of variables.

```
paf <- paran(mytest,
             centile = 95,
             cfa = TRUE,
             seed = 1804,
             iterations = 0,
             graph = TRUE,
             color = TRUE,
             all = TRUE)
```

```
##
## Using eigendecomposition of correlation matrix.
## Computing: 10%  20%  30%  40%  50%  60%  70%  80%  90%  100%
##
```

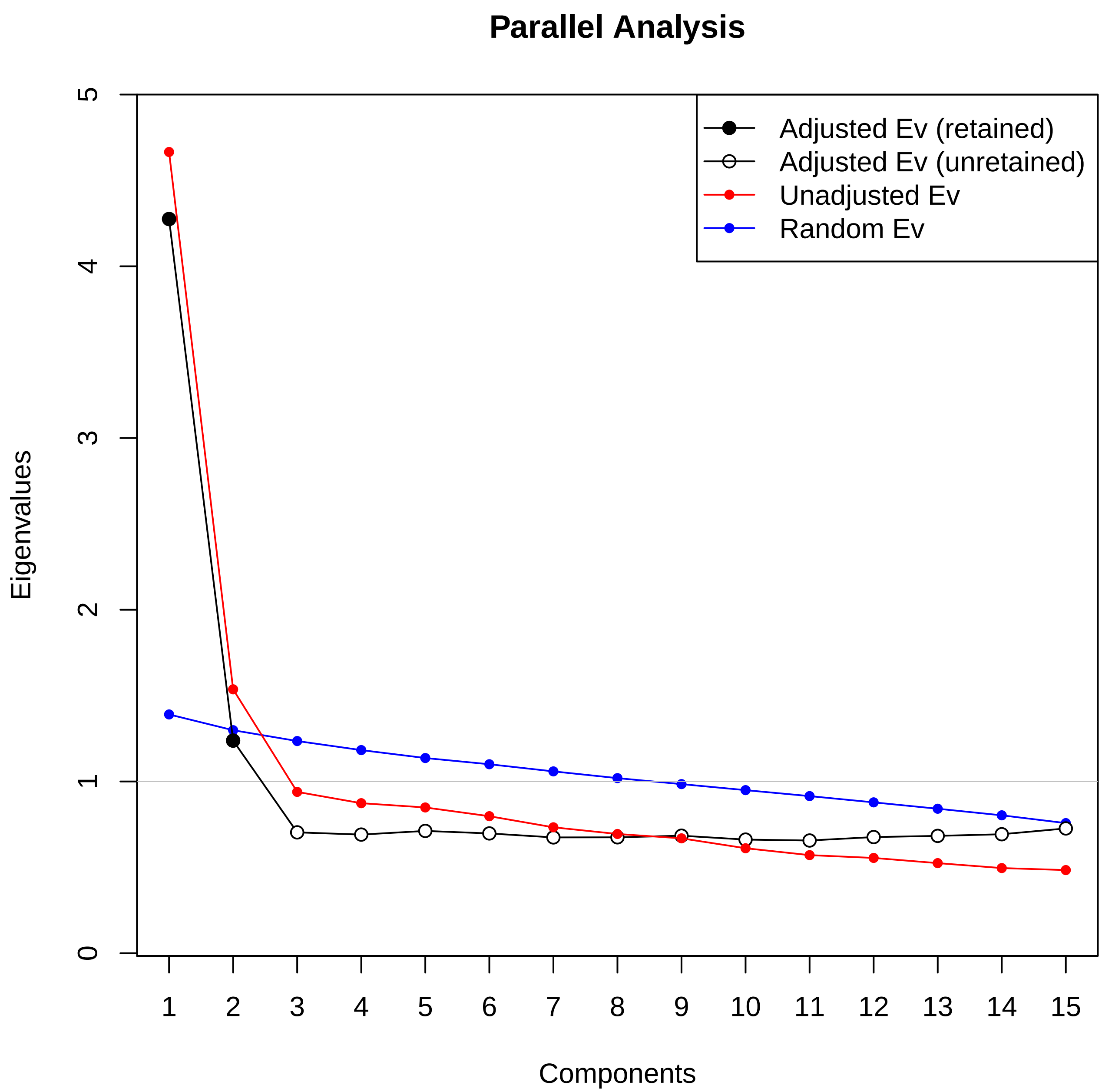

Parallel Analysis
Adjusted Ev (retained)
Adjusted Ev (unretained)
Unadjusted Ev
Random Ev
Eigenvalues
0
1
2
3
4
5
Components
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

Figure 36:

```
##
## Results of Horn's Parallel Analysis for factor retention
## 450 iterations, using the 95 centile estimate
##
## --------------------------------------------------
## Factor      Adjusted    Unadjusted    Estimated
##             Eigenvalue  Eigenvalue    Bias
## --------------------------------------------------
## No components passed.
## --------------------------------------------------
## 1            3.542369    3.976372      0.434003
## 2            0.443503    0.783075      0.339572
## 3           -0.152040    0.124109      0.276149
## 4           -0.111772    0.107293      0.219066
## 5           -0.089135    0.079919      0.169055
## 6           -0.102563    0.029252      0.131815
## 7           -0.080576    0.008756      0.089333
## 8           -0.060421   -0.01304      0.047379
## 9           -0.067095   -0.05810      0.008993
## 10          -0.083851   -0.10812     -0.02427
## 11          -0.064342   -0.12535     -0.06100
## 12          -0.051106   -0.14512     -0.09401
## 13          -0.033770   -0.16668     -0.13291
## 14          -0.019336   -0.18718     -0.16784
## 15           0.018223   -0.19224     -0.21046
## --------------------------------------------------
##
## Adjusted eigenvalues > 0 indicate dimensions to retain.
## (2 factors    retained)
```

In accordance with the results above, Horn's Parallel Analysis also shows two components in the latent person variable. As a result, there may be more than one latent variable that underlies the respondents' perfomance and the unidimensionality assumption has been comprimised because the observations on the manifest variables (items or questions) are not solely a function of a single continious latent person variable (ability).

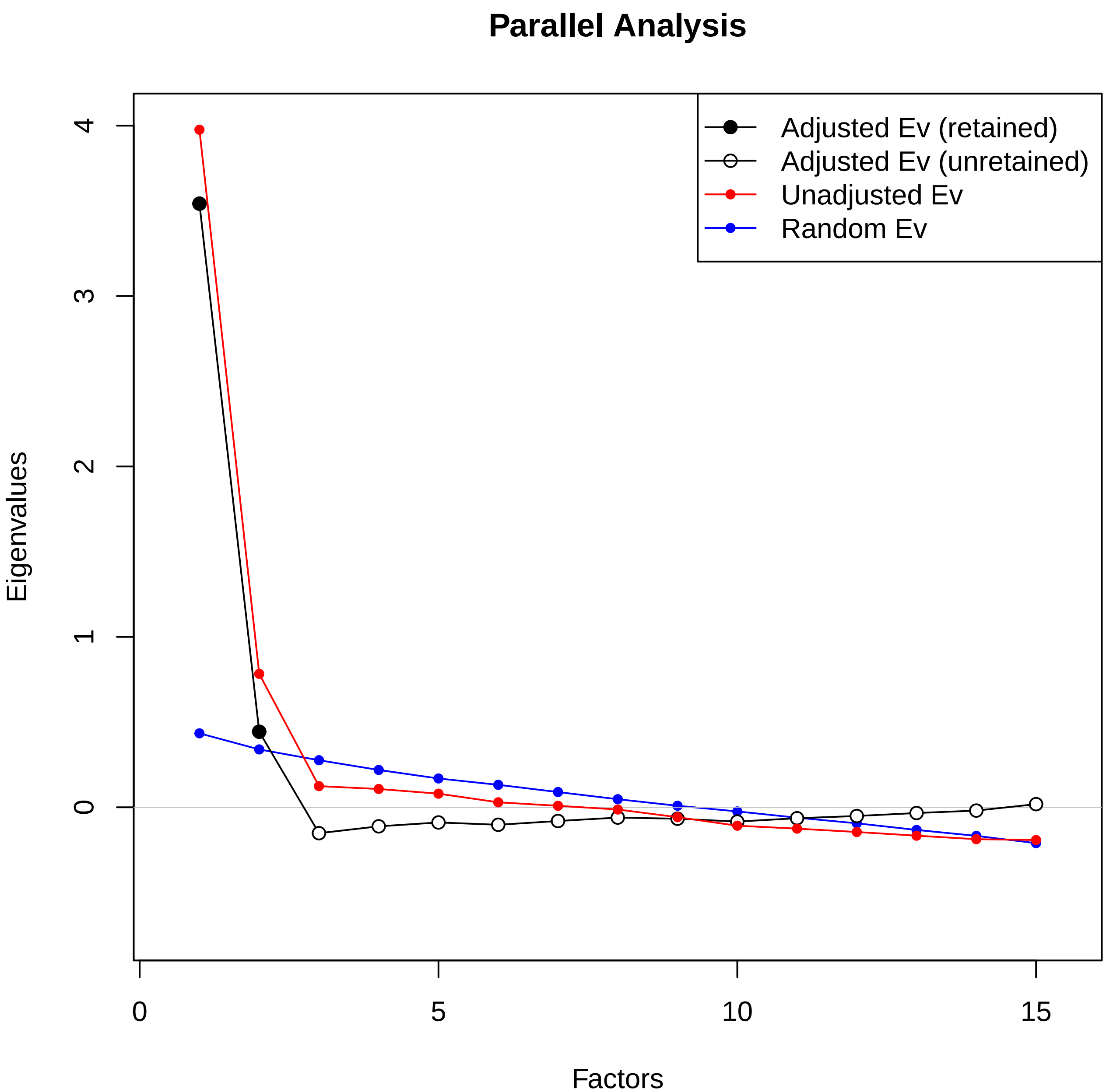

Parallel Analysis
Adjusted Ev (retained)
Adjusted Ev (unretained)
Unadjusted Ev
Random Ev
Eigenvalues
Factors
0
1
2
3
4
0
5
10
15

Figure 37:

```r
keys <- rep(1, times = nitems)
scores <- scoreItems(keys, mytest, totals = TRUE)
summary(scores)
```

```
## Call: scoreItems(keys = keys, items = mytest, totals = TRUE)
##
## Scale intercorrelations corrected for attenuation
##  raw correlations below the diagonal, (unstandardized) alpha on the diagonal
##  corrected correlations above the diagonal:
##      [,1]
## [1,] 0.74
```

```r
# Add scores to the data
mydf <- data.frame(mytest, scores$scores)
summary(mydf)
```

```
##        V1               V2               V3               V4
##  Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000
##  1st Qu.:0.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:0.000
##  Median :0.000   Median :1.000   Median :1.000   Median :0.000
##  Mean   :0.138   Mean   :0.867   Mean   :0.824   Mean   :0.149
##  3rd Qu.:0.000   3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:0.000
##  Max.   :1.000   Max.   :1.000   Max.   :1.000   Max.   :1.000
##        V5               V6               V7              V8
##  Min.   :0.000   Min.   :0.000   Min.   :0.00   Min.   :0.00
##  1st Qu.:0.000   1st Qu.:0.000   1st Qu.:1.00   1st Qu.:0.00
##  Median :0.000   Median :1.000   Median :1.00   Median :1.00
##  Mean   :0.482   Mean   :0.624   Mean   :0.82   Mean   :0.68
##  3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:1.00   3rd Qu.:1.00
##  Max.   :1.000   Max.   :1.000   Max.   :1.00   Max.   :1.00
##        V9              V10              V11              V12
##  Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000
##  1st Qu.:0.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:0.000
##  Median :0.000   Median :1.000   Median :1.000   Median :0.000
##  Mean   :0.473   Mean   :0.956   Mean   :0.869   Mean   :0.213
##  3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:0.000
```

```
## Max.   :1.000   Max.   :1.000   Max.   :1.000   Max.   :1.000
##      V13              V14              V15              S1
## Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   : 1.00
## 1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.000   1st Qu.: 7.00
## Median :0.000   Median :0.000   Median :1.000   Median : 9.00
## Mean   :0.358   Mean   :0.229   Mean   :0.733   Mean   : 8.42
## 3rd Qu.:1.000   3rd Qu.:0.000   3rd Qu.:1.000   3rd Qu.:10.00
## Max.   :1.000   Max.   :1.000   Max.   :1.000   Max.   :14.00
```

```
est1PL   <- est(resp = mytest,
                model = "1PL",
                engine = "ltm")

theta1PL <- eap(resp = mytest,
                ip = est1PL$est,
                qu = normal.qu())

mydf <- data.frame(mydf, thetahat = theta1PL[,1])
quintiles <- quantile(mydf$thetahat,
                      probs = seq(from = 1/3, to = 1.0, by = 1/3))
x <- cut(mydf$thetahat, c(-Inf, quintiles))
summary(x)
```

```
## (-Inf,-0.427] (-0.427,0.458]    (0.458,1.8]
##           152            158            140
```

```
mydf <- data.frame(mytest,
                   scores$scores,
                   theta = round(theta1PL[,1], 2),
                   group = as.numeric(x))
# by(mydata,mydf$group,cor)
knitr::kable(head(mydf),
             caption = 'Data frame with scores and ability quantiles \n (1=low, 2=medium, 3
```

Table 16: Data frame with scores and ability quantiles (1=low, 2=medium, 3=high).

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	S1	theta	group
0	0	1	0	1	0	0	0	0	1	0	0	0	0	1	4	-1.30	1
0	1	1	1	0	0	1	0	0	1	1	1	0	0	1	8	-0.14	2
0	1	1	0	0	1	1	1	1	1	1	0	1	0	1	10	0.46	2
1	1	1	0	0	1	1	1	1	1	1	0	1	1	1	12	1.09	3
0	1	1	0	0	1	1	1	1	1	1	0	0	0	1	9	0.16	2
0	1	0	0	1	1	1	0	0	1	1	0	0	0	0	6	-0.72	1

References

Ayala, R. J. de. 2009. *The Theory and Practice of Item Response Theory.* New York: Guilford Publications.

Camilli, G. 1994. "Origin of the Scaling Constant d = 1.7, in Item Response Theory." *Journal of Educational and Behavioral Statistics* 19 (3): 293–5.