# Item Response Theory for Dichotomous Items

*Rachael Smyth and Andrew Johnson*

## Introduction

This lab discusses the use of Item Response Theory (or IRT) for dichotomous items. Item response theory focuses specifically on the items that make up tests. We will compare the items that make up a test and look at how well they measure the construct we're aiming to measure.

Item Response Theory uses responses to individual test items to estimate the following parameters of individual items:

- Item Difficulty
- Item Discrimination
- Guessing

The data that we will work through in this example is described (in detail) by:

Rizopoulos, D. (2006). ltm: An R package for latent variable modeling and item response theory analyses. *Journal of Statistical Software, 17*(5).

Rizopoulos uses the **LSAT** dataset that is included with the `ltm` package. This dataset is made up of the responses of 1000 individuals to 5 questions on the LSAT exam measuring one latent variable, *preparedness for law school*.

## Load Libraries

For this analysis, we need the following packages:

- `psych`
- `ltm`

```r
library(psych)
library(ltm)
```

## Loading the Data

The first thing we need to do is load our dataset.

```r
data(LSAT)
```

## Estimating the Parameters of the Rasch model

The simplest IRT model is the Rasch model, as it estimates only 1 parameter (difficulty) for each item. Within the Rasch model, however, we can either hold the discriminatory ability of each item constant at 1, or we can estimate a parameter that is different from 1 (the parameter still remains constant across all items). In order to determine whether or not our model-fitting is better when discriminatory ability is held constant at 1, or allowed to be a value other than 1, we need to evaluate the model parameters for each of these models.

## Model 1 (Rasch model with constrained discrimination)

First, we'll try fitting the Rasch model with discrimination set to 1.

```
fit1 <- rasch(LSAT, constraint = cbind(length(LSAT)+1,1))
```

The values presented in the *summary* of **fit1** show the difficulty of each item on the test.

```
summary(fit1)
```

```
##
## Call:
## rasch(data = LSAT, constraint = cbind(length(LSAT) + 1, 1))
##
## Model Summary:
##    log.Lik      AIC      BIC
##  -2473.054 4956.108 4980.646
##
## Coefficients:
##                  value std.err   z.vals
## Dffclt.Item 1 -2.8720  0.1287 -22.3066
## Dffclt.Item 2 -1.0630  0.0821 -12.9458
## Dffclt.Item 3 -0.2576  0.0766  -3.3635
## Dffclt.Item 4 -1.3881  0.0865 -16.0478
## Dffclt.Item 5 -2.2188  0.1048 -21.1660
## Dscrmn         1.0000      NA       NA
##
## Integration:
## method: Gauss-Hermite
## quadrature points: 21
##
## Optimization:
## Convergence: 0
## max(|grad|): 6.3e-05
## quasi-Newton: BFGS
```

The *coef* command produces an output that shows a summary of the item difficulty, item discrimination (1 for each item in this model), as well as the probability that the average individual will answer that item correctly.

```
coef(fit1,prob=TRUE)
```

```
##           Dffclt Dscrmn P(x=1|z=0)
## Item 1 -2.8719712      1  0.9464434
## Item 2 -1.0630294      1  0.7432690
## Item 3 -0.2576109      1  0.5640489
## Item 4 -1.3880588      1  0.8002822
## Item 5 -2.2187785      1  0.9019232
```

The final function that we will evaluate looks at the goodness of fit of this model to the data. If the p-value is significant, the results suggest that the model is not a good fit to the data. In our case, the p-value is not significant. As such, the model is a good fit to the data.

```
GoF.rasch(fit1,B=199)
```

```
##
## Bootstrap Goodness-of-Fit using Pearson chi-squared
##
## Call:
## rasch(data = LSAT, constraint = cbind(length(LSAT) + 1, 1))
##
## Tobs: 30.6
## # data-sets: 200
## p-value: 0.24
```

## Model 2 (Rasch model with unconstrained, but constant, discrimination)

In the next model that we will evaluate, all items are still required to have the same discrimination parameter, but it is unconstrained (i.e., it can be different from 1).

```
fit2 <- rasch(LSAT)
summary(fit2)
```

```
##
## Call:
## rasch(data = LSAT)
##
## Model Summary:
##    log.Lik      AIC      BIC
##  -2466.938 4945.875 4975.322
##
## Coefficients:
##                  value std.err   z.vals
## Dffclt.Item 1 -3.6153  0.3266 -11.0680
## Dffclt.Item 2 -1.3224  0.1422  -9.3009
## Dffclt.Item 3 -0.3176  0.0977  -3.2518
## Dffclt.Item 4 -1.7301  0.1691 -10.2290
## Dffclt.Item 5 -2.7802  0.2510 -11.0743
## Dscrmn         0.7551  0.0694  10.8757
##
## Integration:
## method: Gauss-Hermite
## quadrature points: 21
##
## Optimization:
## Convergence: 0
## max(|grad|): 2.9e-05
## quasi-Newton: BFGS
```

Note that the item difficulties are still listed under value, but as you can see at the bottom of the output, *Dscrmn* is not fixed at 1. The analysis suggests that the items have a discriminatory parameter of 0.7551, based on this model.

### Model 2 versus Model 1

We can test to see if *Model 2* has a significantly better fit than *Model 1* by evaluating the model characteristics within an ANOVA.

```
anova(fit1, fit2)
```

```
##
##  Likelihood Ratio Table
##          AIC     BIC  log.Lik   LRT df p.value
## fit1 4956.11 4980.65 -2473.05
## fit2 4945.88 4975.32 -2466.94 12.23  1  <0.001
```

The significant p-value in this chart tells us that *Model 2* is a better fit to the data than *Model 1*. This means that 0.7551 is a better estimate of the discriminatory ability than 1. If the p-value had not been significant, it would only mean that *Model 2* was not a better fit than *Model 1*. It would not mean that *Model 1* was a better fit than *Model 2* (i.e., the models could fit equally well).

## Estimating the Parameters of the Two Parameter Logistic Model (discriminatory ability)

We know that the Rasch model is a good fit to the data, and that this fit is better when discriminatory ability is allowed to take on a value other than 1. Our next step is to determine whether model fit improves when we allow this discriminatory ability to vary for each item. This is no longer a Rasch model - it is a two-parameter logistic model, and so we will need to apply a different function to this analysis (`ltm`).

```
fit3 <- ltm(LSAT ~ z1)
summary(fit3)
```

```
##
## Call:
## ltm(formula = LSAT ~ z1)
##
## Model Summary:
##    log.Lik      AIC      BIC
##  -2466.653 4953.307 5002.384
##
## Coefficients:
##                  value std.err  z.vals
## Dffclt.Item 1 -3.3597  0.8669 -3.8754
## Dffclt.Item 2 -1.3696  0.3073 -4.4565
## Dffclt.Item 3 -0.2799  0.0997 -2.8083
## Dffclt.Item 4 -1.8659  0.4341 -4.2982
## Dffclt.Item 5 -3.1236  0.8700 -3.5904
## Dscrmn.Item 1  0.8254  0.2581  3.1983
## Dscrmn.Item 2  0.7229  0.1867  3.8721
## Dscrmn.Item 3  0.8905  0.2326  3.8281
## Dscrmn.Item 4  0.6886  0.1852  3.7186
## Dscrmn.Item 5  0.6575  0.2100  3.1306
##
## Integration:
```

```
## method: Gauss-Hermite
## quadrature points: 21
##
## Optimization:
## Convergence: 0
## max(|grad|): 0.024
## quasi-Newton: BFGS
```

Note that the discriminatory ability is now different for each item (and ranges from 0.6575 to 0.8254). The values for the item difficulty parameter are also slightly different than they were in either *Model 1* or *Model 2*.

### Model 3 versus Model 2

We know that *Model 2* fits the data better than *Model 1*, and so we only need to test *Model 3* versus *Model 2*. We will do so in the same manner that we employed previously.

```
anova(fit2, fit3)
```

```
##
##  Likelihood Ratio Table
##          AIC     BIC  log.Lik  LRT df p.value
## fit2 4945.88 4975.32 -2466.94
## fit3 4953.31 5002.38 -2466.65 0.57  4   0.967
```

The p-value is not significant, which suggests that *Model 3* is not a significantly better fit than *Model 2*. *Model 2* is not necessarily better, but *Model 3* is not significantly better. This means that the fit of the model is not improved by estimating the discriminatory ability of each individual item.

## Estimating the Impact of Guessing

The final model we will attempt incorporates guessing. Guessing is typically estimated as a constant (i.e., it is constrained to be the same for each item), and this constant can either be added to the two-parameter logistic model that we fit earlier (wherein discriminatory ability was allowed to vary for each item), or it can be added to the Rasch model. We already know that the two-parameter model yielded results that were no better than the Rasch model, and so our most parsimonious option is to add our guessing constant to the Rasch model.

```
fit4 <- tpm(LSAT, type="rasch", max.guessing = 1)
summary(fit4)
```

```
##
## Call:
## tpm(data = LSAT, type = "rasch", max.guessing = 1)
##
## Model Summary:
##    log.Lik      AIC      BIC
##  -2466.731 4955.461 5009.447
##
## Coefficients:
##                  value std.err  z.vals
## Gussng.Item 1  0.0830  0.8652  0.0959
## Gussng.Item 2  0.1962  0.3545  0.5535
## Gussng.Item 3  0.0081  0.0817  0.0994
## Gussng.Item 4  0.2565  0.4776  0.5372
## Gussng.Item 5  0.4957  0.4839  1.0242
## Dffclt.Item 1 -3.1765  1.5090 -2.1050
## Dffclt.Item 2 -0.7723  1.0281 -0.7513
## Dffclt.Item 3 -0.2707  0.2375 -1.1395
## Dffclt.Item 4 -1.0332  1.4031 -0.7364
## Dffclt.Item 5 -1.4332  1.9055 -0.7521
## Dscrmn         0.8459  0.1851  4.5711
##
## Integration:
## method: Gauss-Hermite
## quadrature points: 21
##
## Optimization:
## Optimizer: optim (BFGS)
## Convergence: 0
## max(|grad|): 0.073
```

Now we see a difficulty parameter for each item, a discrimination parameter that remains constant for the model and a guessing parameter for each item.

## Model 4 versus Model 2

*Model 2* was better than *Model 1*, and *Model 3* was no better than *Model 2*. Thus, we only need to test to see if *Model 4* produces a better fit to the data than *Model 2*.

```r
anova(fit2, fit4)
```

```
##
##  Likelihood Ratio Table
##          AIC     BIC  log.Lik  LRT df p.value
## fit2 4945.88 4975.32 -2466.94
## fit4 4955.46 5009.45 -2466.73 0.41  5   0.995
```

The non-significant p-value in this comparison suggests that adding guessing to the model doesn't improve its fit to the data.

## Graphical Representations

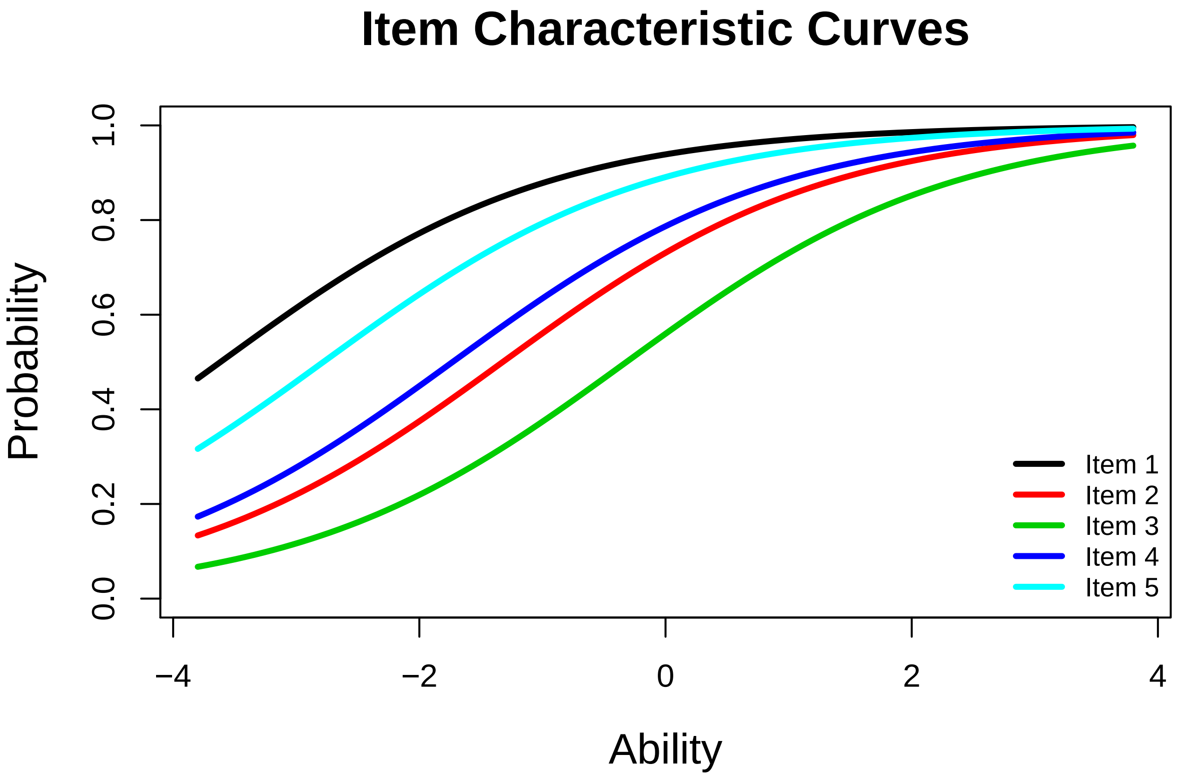
We can use a few different graphs to help us understand our data:

- Item Characteristic Curve
- Item Information Curve
- Test Information Curve

### Item Characteristic Curves

The Item Characteristic Curves provides us with more information about the underlying construct. They show us the probability of answering an item correctly at varying levels of ability. This tells us how well an item discriminates between respondents at various levels of the latent trait.

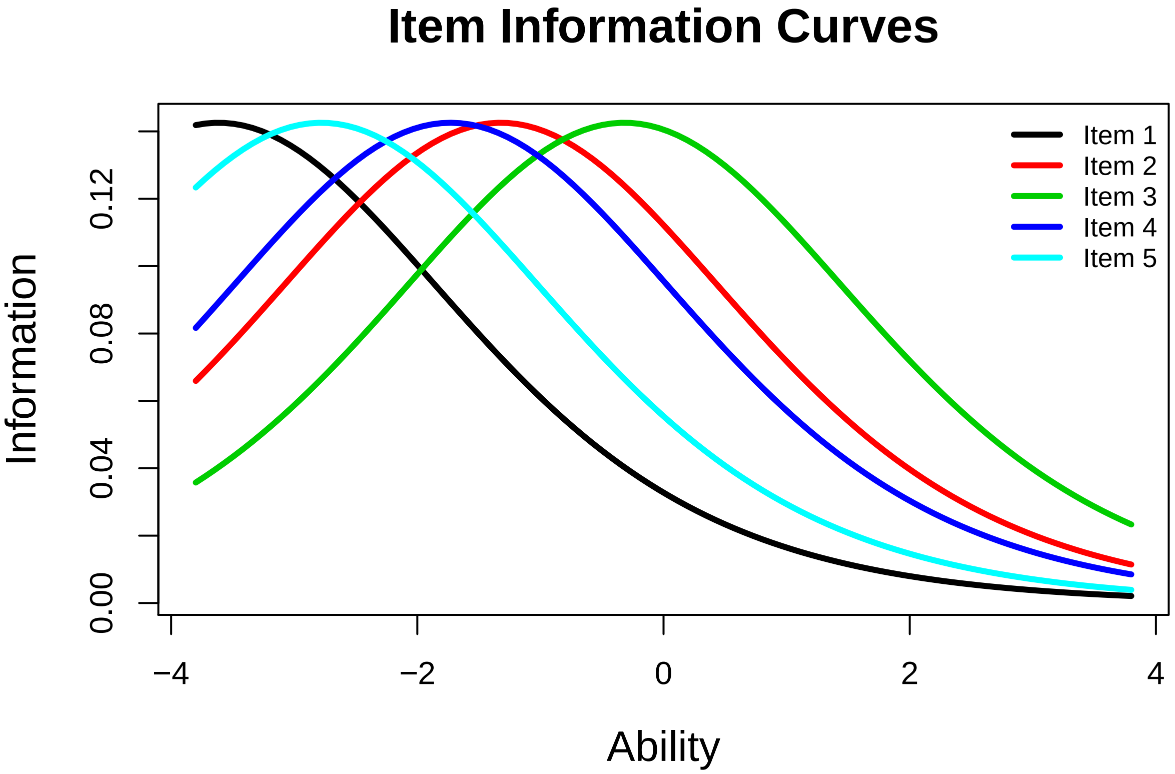
```
plot(fit2,
     legend = TRUE,
     cx = "bottomright",
     lwd = 3,
     cex.main = 1.5,
     cex.lab = 1.3,
     cex = 0.8)
```



### Item Information Curve

The Item Information Curve shows how well and precisely each item measures the latent trait at various levels of the attribute. Some items may provide more information at low levels of the attribute while others may provide more information at high levels of the attribute.

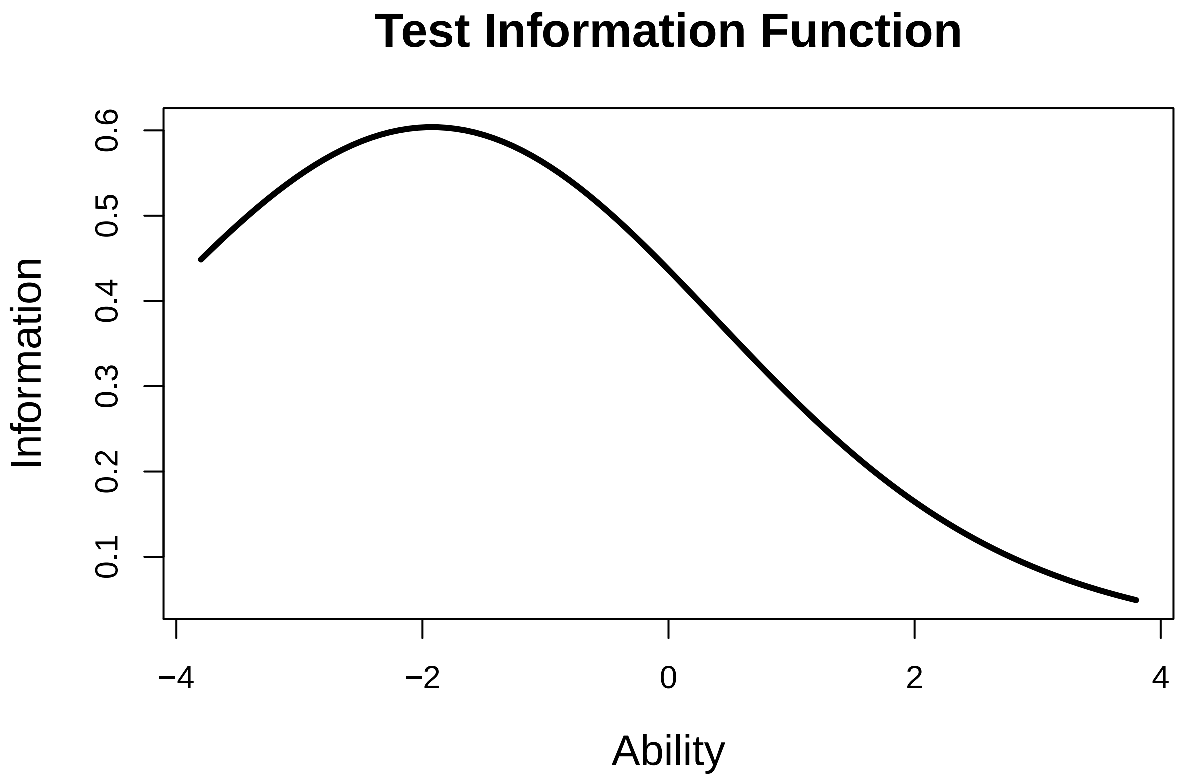
```r
plot(fit2,
     legend = TRUE,
     cx = "topright",
     cex = 0.8,
     type = "IIC",
     annot = FALSE,
     lwd = 3,
     cex.main = 1.5,
     cex.lab = 1.3)
```



### Test Information Function

The Test Information Function aggregates the Item Information Curves of all the items. It tells us how well the test measures the latent trait at various levels of the attribute. Ideally, we want this line to peak at the mean of the sample because that is where the highest number of individuals would be. It helps us estimate the ability level at which the test is most effective.

```r
plot(fit2,
     type="IIC",
     items = 0,
     lwd = 3,
     cex.main = 1.5,
     cex.lab = 1.3)
```


Test Information Function
Information
Ability