Item Response Theory for Polytomous Items

Rachael Smyth

Introduction

This lab discusses the use of Item Response Theory (or IRT) for polytomous items. Item response theory focuses specifically on the items that make up tests. We will compare the items that make up a test and evaluate 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 for each item:

- Item Difficulty/Item Distance
- Item Discrimination
- Guessing/Chance

For more information on this highly complicated topic, you would be well-served by reading the excellent article by Dimitris Rizopolous, on the package that we will use in today's video on Item Response Theory.

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

Load Libraries

We'll need two packages for the analysis today:

- `ltm`
- `psych`

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

Loading the Data

We'll continue working with the `bfi` data that we used for *Item Analysis* and *Factor Analysis*.

```
data(bfi)
```

We also want to create a dataframe using only the personality items of the entire dataset.

```
bfi.items <- bfi[,1:25]
```

Estimating Parameters in Polytomous Items: The Graded Response Model

We are working with **personality** data, and so we'll look at the probability of endorsing a response as opposed to the correctness of a response in this example. The probability of endorsing a response will be a function of attribute and distance instead of one of ability and difficulty. In this example, we'll use attribute and distance because we're looking at **personality** items, but if you were looking at **ability** items, you would use the terms ability and difficulty.

The items in our dataset are all polytomous, or have more than 2 possible responses. Because we are using the `bfi` data, we do not have a correct and an incorrect response, but multiple responses ranging from *Very Inaccurate* to *Very Accurate*. We will use the **Graded Response Model** for our parameter estimation. Our data is ordinal level, but you can use **Graded Response Model** on data that is ordinal, interval or ratio.

Constrained versus Unconstrained Discrimination

There are two basic models that we can evaluate when setting up our IRT analysis: we can assume that all of the items are equally good at discriminating among respondents, or we can assume that each of the items have a different discrimination parameter. Thus, we are comparing between a "constrained" model and an "unconstrained" model - and, as we will see, it is possible to directly test the difference between these models, in terms of their fit to the data.

Model 1: Constrained Discrimination

The first model that we will evaluate is the simplest model, as it does not allow the discrimination parameter to vary between items. We might say that we are "constraining" the discrimination parameter within our analysis, and so this model is called a constrained model.

To indicate this in our R script, we use the code *constrained = TRUE*. We'll call this new model **fit1**.

```
fit1 <- grm(bfi.items, constrained = TRUE)
fit1
```

```
##
## Call:
## grm(data = bfi.items, constrained = TRUE)
##
## Coefficients:
##     Extrmt1  Extrmt2  Extrmt3  Extrmt4  Extrmt5  Dscrmn
## A1   -1.790    1.230    2.974    5.211    8.695   0.407
## A2  -10.149   -6.830   -5.116   -1.966    1.983   0.407
## A3   -8.544   -5.723   -4.048   -1.336    2.519   0.407
## A4   -7.593   -4.953   -3.684   -1.566    0.890   0.407
## A5   -9.597   -5.908   -3.874   -1.050    2.783   0.407
## C1   -9.056   -6.008   -3.778   -0.831    3.282   0.407
## C2   -8.583   -5.128   -3.176   -0.467    3.563   0.407
## C3   -8.700   -5.058   -3.163   -0.121    4.001   0.407
## C4   -2.435    0.594    2.500    5.341    9.365   0.407
## C5   -3.809   -1.210    0.069    2.482    5.428   0.407
## E1   -2.900   -0.288    1.191    3.160    5.852   0.407
## E2   -3.619   -0.708    0.538    3.031    5.745   0.407
```

```
## E3   -7.258   -4.246   -2.089    1.084    4.909   0.407
## E4   -7.387   -4.508   -2.911   -1.011    2.648   0.407
## E5   -8.405   -5.201   -3.263   -0.627    3.189   0.407
## N1   -3.055   -0.349    1.265    3.683    6.568   0.407
## N2   -5.180   -2.100   -0.478    2.301    5.481   0.407
## N3   -3.934   -0.994    0.374    2.823    5.846   0.407
## N4   -4.038   -0.971    0.530    3.118    5.882   0.407
## N5   -3.039   -0.315    1.122    3.434    5.966   0.407
## O1  -11.916   -7.654   -5.004   -1.691    1.832   0.407
## O2   -2.326    0.394    1.887    4.106    6.725   0.407
## O3   -8.958   -6.166   -3.753   -0.356    3.607   0.407
## O4   -9.783   -6.740   -5.048   -2.251    1.166   0.407
## O5   -2.541    0.828    3.060    5.640    9.089   0.407
##
## Log.Lik: -111365.9
```

When looking at the output for the model we've created called **fit1**, you can see a row for each personality item in the `bfi` scale. There are 6 columns in this case: 5 are the extremity parameters (because we have 6 possible responses) and one is the discrimination parameter. As you'll notice, the discrimination parameter (or *dscrmn*) is the same for each item. This is because we set this model with *constrained = TRUE*, assuming equal discrimination parameters across items. The extremity parameters show the latent trait score at which people have a 50/50 chance of selecting certain responses.

So, for example, let's look at row A1: The *Extrmt1* is -1.790 suggesting that those with a latent trait score of -1.790 have a 50/50 chance of selecting a 1 for that item. The *Extrmt2* is 1.230 and so those with a latent trait score of 1.230 have a 50/50 chance of selecting a 1 or a 2 for that item. *Extrmt3* is the latent trait score at which people have a 50/50 chance of selecting a 1, a 2 or a 3. *Extrmt4* is the latent trait score at which people have a 50/50 chance of selecting a 1, a 2, a 3 or a 4. Finally, *Extrmt5* is the latent trait score at which people have a 50/50 chance of selecting a 1, 2, 3, 4 or 5.

The problem with this analysis is that it assumes that all of the items on the measure are assessing the same underlying construct - and that this construct can be estimated by averaging across all of the items within the questionnaire. Thus, you are using "agreeableness" items to estimate "extraversion", "extraversion" items to estimate "neuroticism", and so on. A better practice might be to evaluate the items associated with each of the five scales, and check to see how well they predict the scale with which they are expected to be associated.

```
fit1.agree <- grm(bfi.items[,1:5], constrained = TRUE)
fit1.consc <- grm(bfi.items[,6:10], constrained = TRUE)
fit1.extra <- grm(bfi.items[,11:15], constrained = TRUE)
fit1.neuro <- grm(bfi.items[,16:20], constrained = TRUE)
fit1.open <- grm(bfi.items[,21:25], constrained = TRUE)
```

As you might expect, you will see different values within this output than you did when you evaluated the graded response model for the items across the entire dataset, rather than within specific scales. Consider the output for "agreeableness":

```
fit1.agree
```

```
##
## Call:
## grm(data = bfi.items[, 1:5], constrained = TRUE)
##
## Coefficients:
##     Extrmt1  Extrmt2  Extrmt3  Extrmt4  Extrmt5  Dscrmn
## A1   -0.869    0.605    1.456    2.553    4.263   0.849
## A2   -5.242   -3.584   -2.713   -1.056    1.070   0.849
## A3   -4.464   -3.040   -2.169   -0.714    1.369   0.849
## A4   -3.977   -2.633   -1.966   -0.833    0.495   0.849
## A5   -4.988   -3.138   -2.075   -0.555    1.514   0.849
##
## Log.Lik: -20748.94
```

Note that the discrimination parameter (*dscrmn*) is now equal to 0.849, rather than 0.407. And. . . for item A1, *Extrmt1* is -0.869, rather than -1.790. This is due to the fact that we are estimating a different latent variable with this model - when we use all of the items, we are estimating a general factor of personality, and when we use only the agreeableness items, we are estimating an agreeableness factor.

Model 2: Unconstrained Discrimination

Let's extend our model, and estimate discrimination separately for each item. This is done by using the *constrained = FALSE* specification, because this is a model in which the discrimination parameter is unconstrained.

```
fit2.agree <- grm(bfi.items[,1:5], constrained = FALSE)
fit2.consc <- grm(bfi.items[,6:10], constrained = FALSE)
fit2.extra <- grm(bfi.items[,11:15], constrained = FALSE)
fit2.neuro <- grm(bfi.items[,16:20], constrained = FALSE)
fit2.open <- grm(bfi.items[,21:25], constrained = FALSE)
fit2.agree
```

```
##
## Call:
## grm(data = bfi.items[, 1:5], constrained = FALSE)
##
## Coefficients:
##     Extrmt1  Extrmt2  Extrmt3  Extrmt4  Extrmt5  Dscrmn
## A1   -0.905    0.744    1.654    2.774    4.459   0.862
## A2    3.030    2.139    1.645    0.660   -0.650  -1.839
## A3    2.276    1.604    1.170    0.404   -0.730  -2.527
## A4    3.352    2.232    1.670    0.709   -0.414  -1.047
## A5    3.005    1.955    1.319    0.369   -0.948  -1.701
##
## Log.Lik: -19604.71
```

As you can see in the output for `fit2.agree` (the fit for the agreeableness items), the discrimination parameter (*dscrmn*) is different for each item. You can also see that the *Extrmt* values change in this model.

Which Model is Better?

To test the model fit statistics between Model 1 and Model 2, we can run an ANOVA that evaluates the difference between the two. We will need to do this separately for each of our five personality variables.

```
anova(fit1.agree,fit2.agree)
```

```
##
##  Likelihood Ratio Table
##                  AIC      BIC   log.Lik     LRT df p.value
## fit1.agree 41549.88 41704.25 -20748.94
## fit2.agree 39269.42 39447.54 -19604.71 2288.46  4  <0.001
```

```
anova(fit1.consc,fit2.consc)
```

```
##
##  Likelihood Ratio Table
##                  AIC      BIC   log.Lik     LRT df p.value
## fit1.consc 45843.55 45997.92 -22895.77
## fit2.consc 42059.20 42237.32 -20999.60 3792.35  4  <0.001
```

```
anova(fit1.extra,fit2.extra)
```

```
##
##  Likelihood Ratio Table
##                  AIC      BIC   log.Lik     LRT df p.value
## fit1.extra 47209.68 47364.05 -23578.84
## fit2.extra 43121.10 43299.22 -21530.55 4096.59  4  <0.001
```

```
anova(fit1.neuro,fit2.neuro)
```

```
##
##  Likelihood Ratio Table
##                  AIC      BIC   log.Lik    LRT df p.value
## fit1.neuro 43948.95 44103.32 -21948.47
## fit2.neuro 43503.81 43681.93 -21721.91 453.14  4  <0.001
```

```
anova(fit1.open,fit2.open)
```

```
##
##  Likelihood Ratio Table
##                 AIC      BIC   log.Lik     LRT df p.value
## fit1.open 42473.72 42628.09 -21210.86
## fit2.open 40767.36 40945.48 -20353.68 1714.36  4  <0.001
```

A significant *p-value* tells us that Model 2 is a better fit to the data than Model 1. And, as you can see from these outputs, the fit for Model 2 (when we didn't assume the same discrimination parameter for each item) is better for all five personality variables.

If the *p-value* had not been significant, we could not have said that Model 1 was a better fit to the data than Model 2, only that Model 2 was not a better fit to the data than Model 1.

Graphing the Results

We can now examine each of the items graphically. We have three good options for graphically evaluating our data:

1) We can create an Item Response Category Characteristic Curve for each item
2) We can create an Item Information Curve for each item
3) We can create a Test Information Curve for each of the scales within the measure

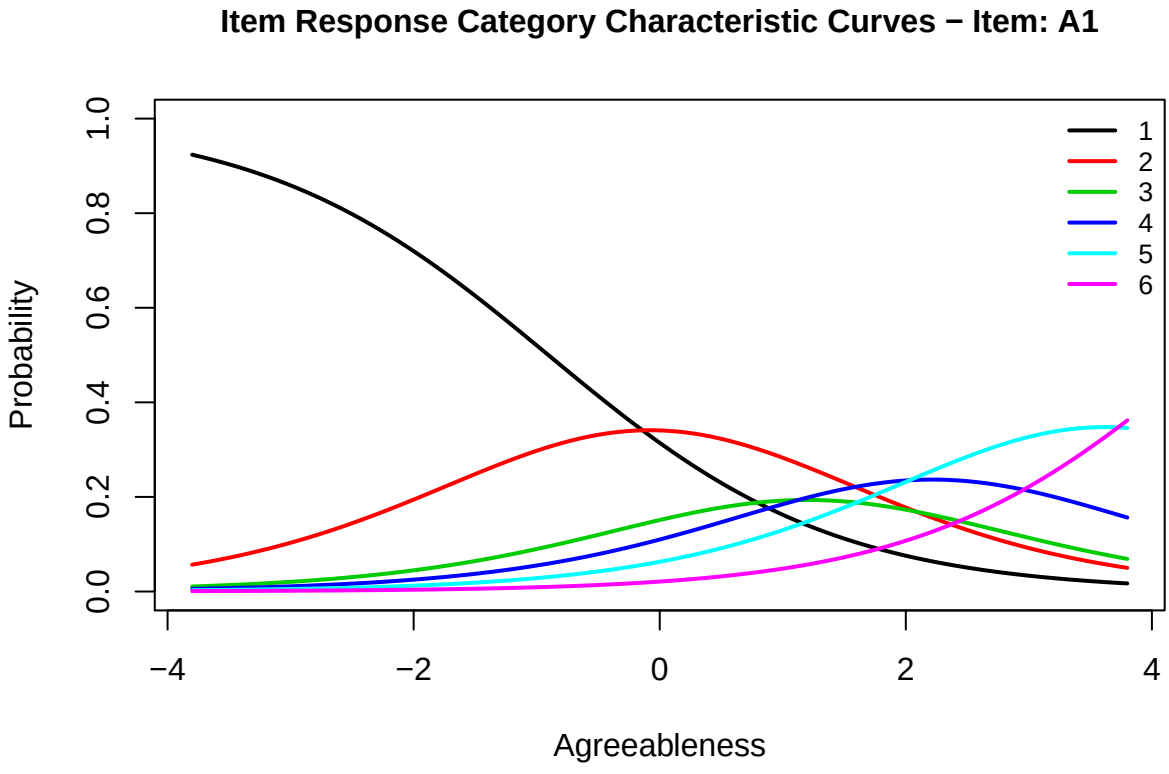
Using the Item Response Category Characteristic Curve

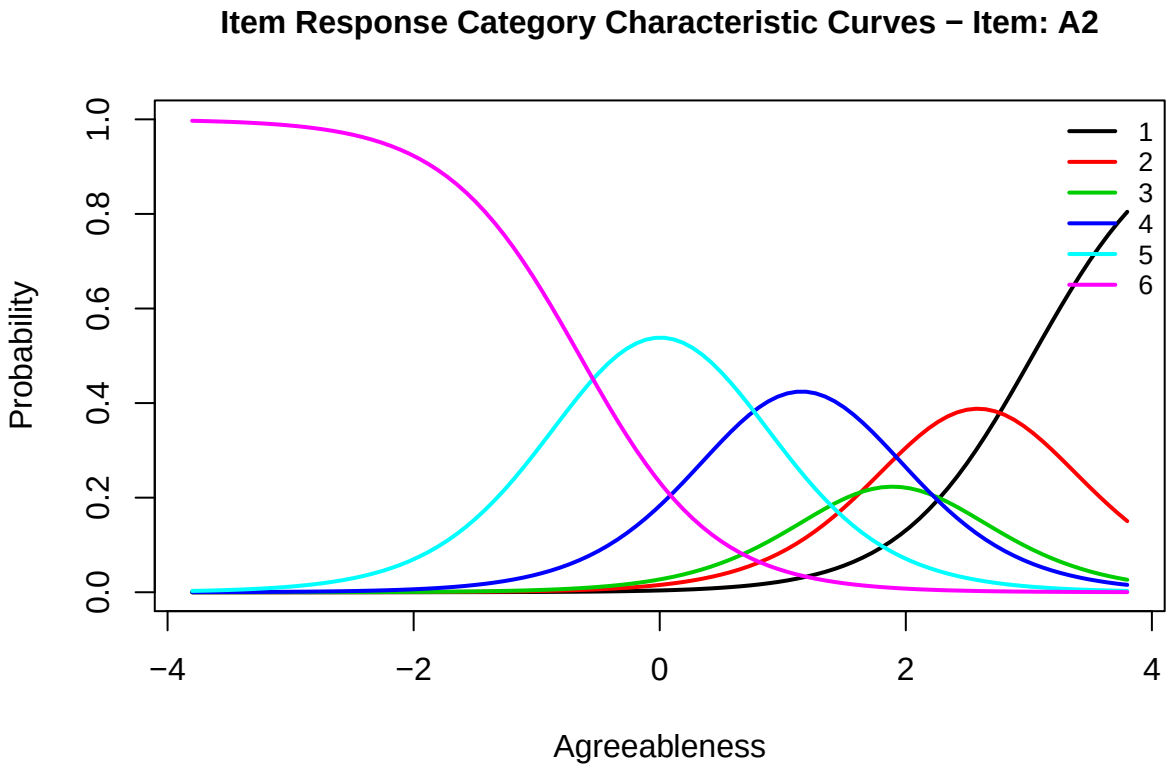
The Item Response Category Characteristic Curve shows the likelihood of respondents selecting a certain score on the scale (1-6) at various levels of the latent trait.

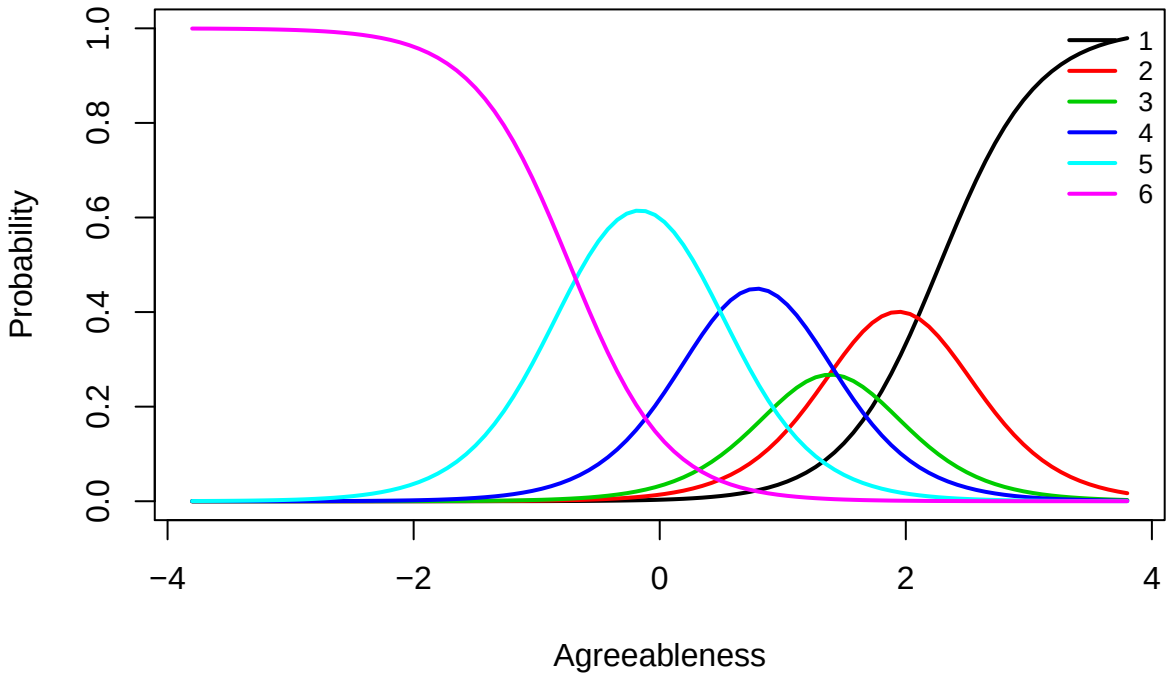
An item is better at discriminating between individuals when the curves are peaked and dispersed across all levels of the latent trait. For example, an item with high discrimination would have 6 peaks dispersed from low levels of the latent trait to high levels of the latent trait.

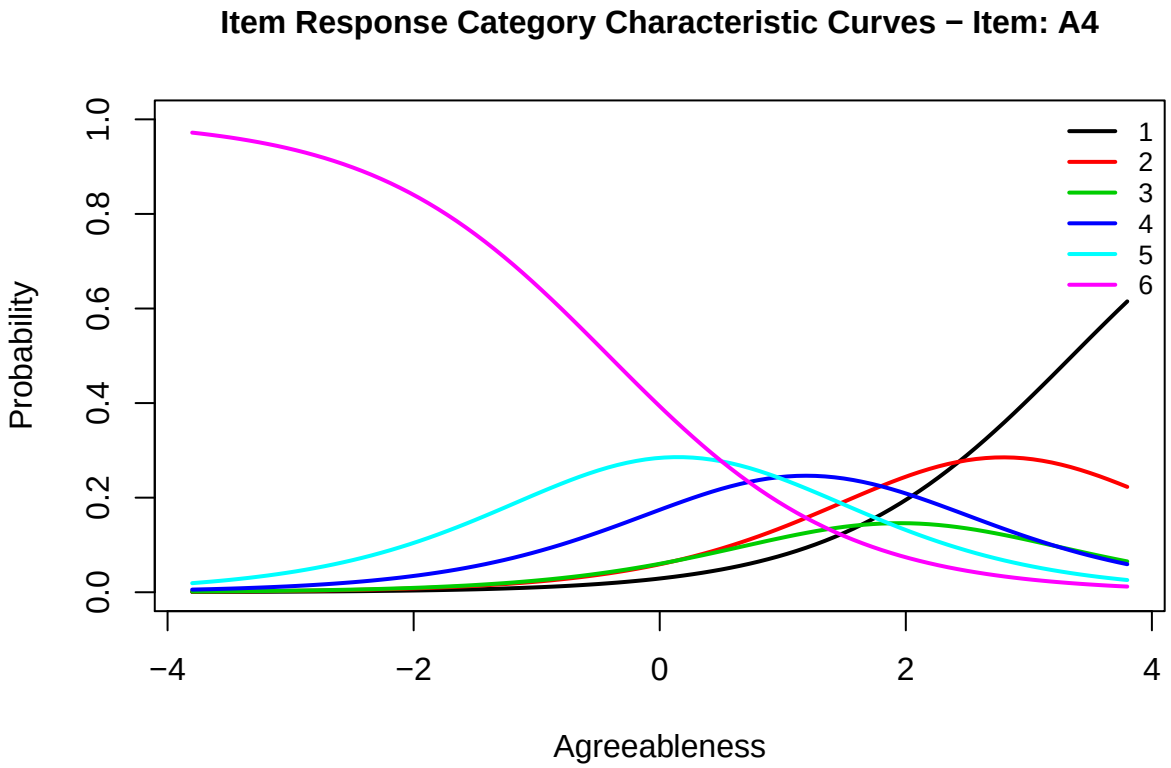
Normally, you would generate a graph for all 25 of the items, but for demonstration purposes, we will just look at the curves for the five items on the agreeableness scale.

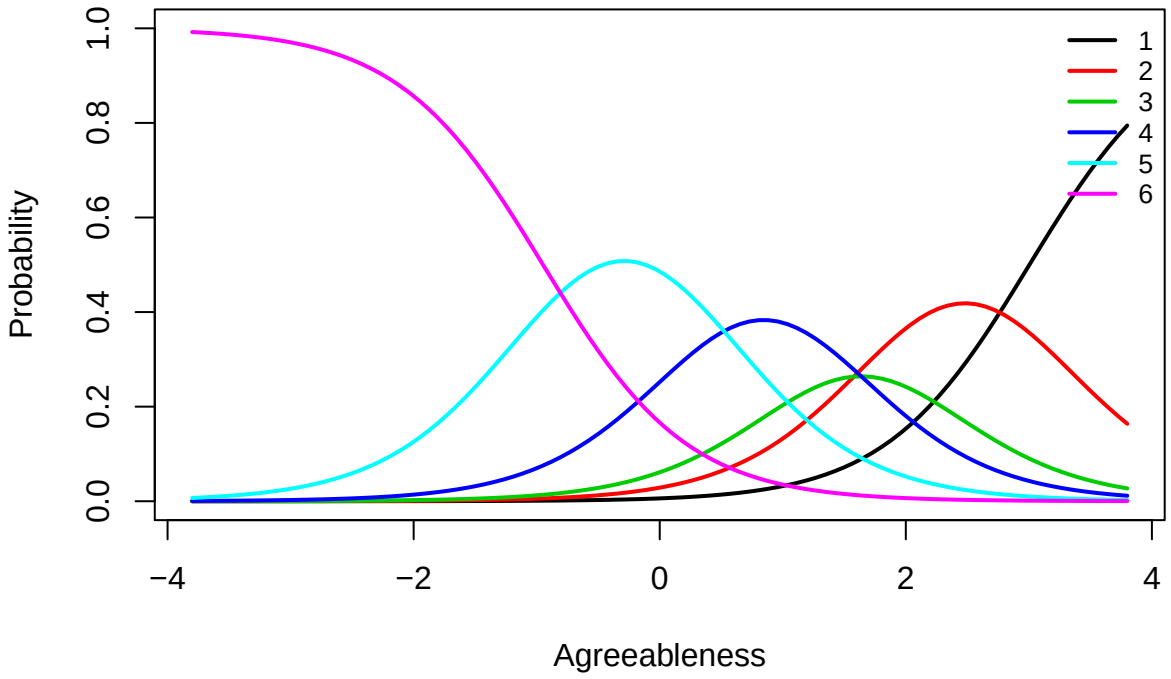
```
plot(fit2.agree, lwd = 2, cex = 0.8,
     legend = TRUE, cx = "topright",
     xlab = "Agreeableness", cex.main = 1,
     cex.lab = 1, cex.axis = 1)
```




Item Response Category Characteristic Curves − Item: A2
Probability
0.0
0.2
0.4
0.6
0.8
1.0
−4
−2
0
2
4
Agreeableness
1
2
3
4
5
6


Item Response Category Characteristic Curves − Item: A3
Probability
0.0
0.2
0.4
0.6
0.8
1.0
−4
−2
0
2
4
Agreeableness
1
2
3
4
5
6


Item Response Category Characteristic Curves − Item: A4
Probability
0.0
0.2
0.4
0.6
0.8
1.0
−4
−2
0
2
4
Agreeableness
1
2
3
4
5
6

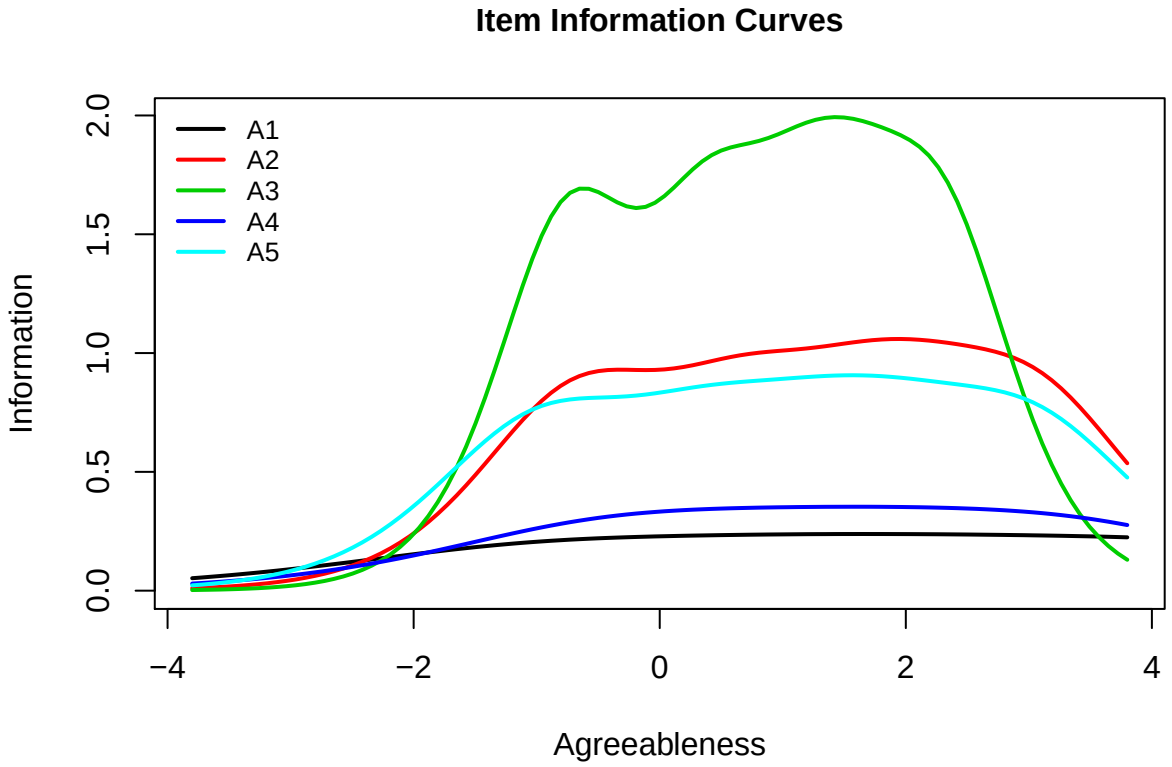

Item Response Category Characteristic Curves − Item: A5
Probability
0.0
0.2
0.4
0.6
0.8
1.0
−4
−2
0
2
4
Agreeableness
1
2
3
4
5
6

Using the Item Information Curve

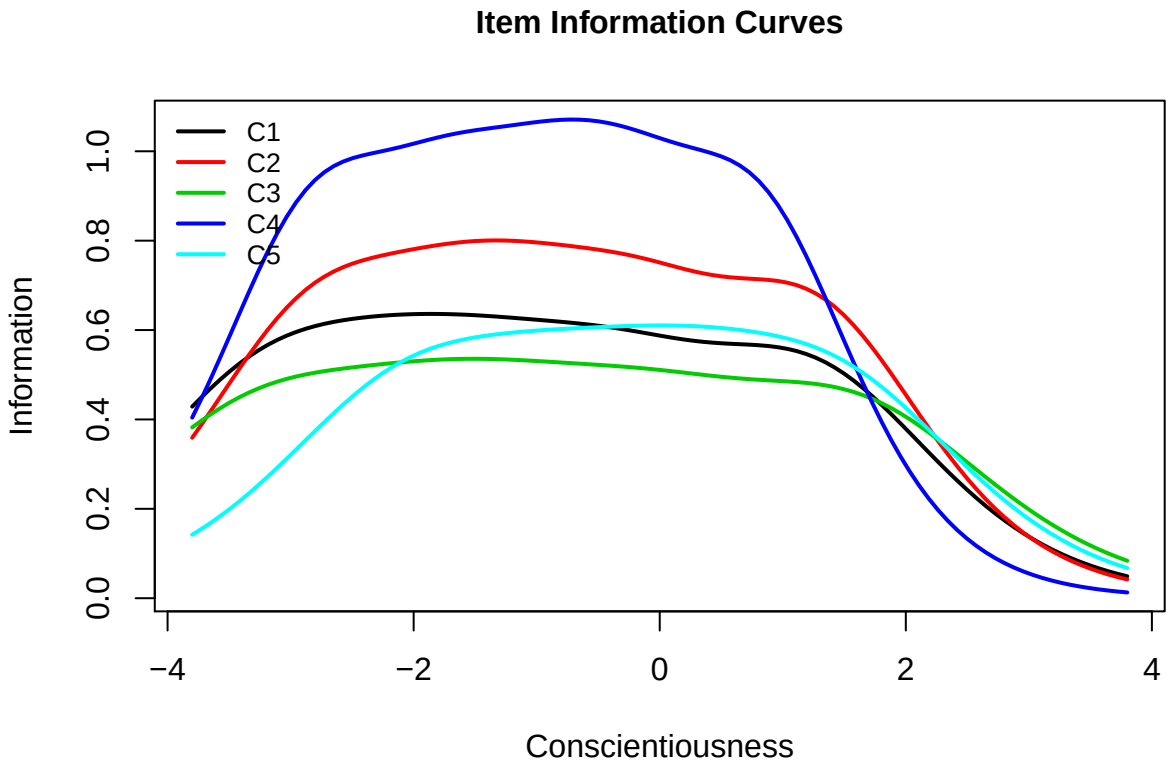
The Item Information Curve shows how well and precisely each item measures the latent trait at various levels of the attribute. Certain items may provide more information at low levels of the attribute, while others may provide more information at higher levels of the attribute.

Again, we need to evaluate our data separately for each of the scales, owing to the fact that we have five latent variables within our dataset.

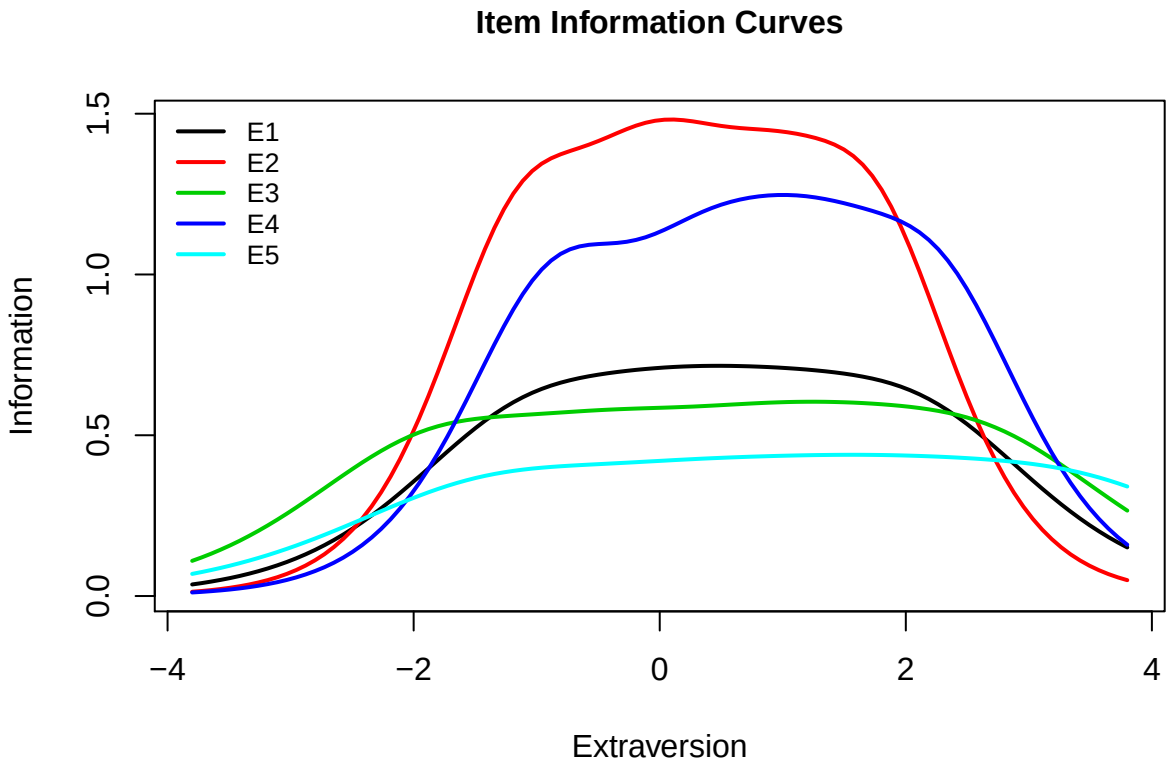
```
plot(fit2.agree, type = "IIC", lwd = 2,
     cex = 0.8, legend = TRUE,
     cx = "topleft", xlab = "Agreeableness",
     cex.main = 1, cex.lab = 1, cex.axis = 1)
```



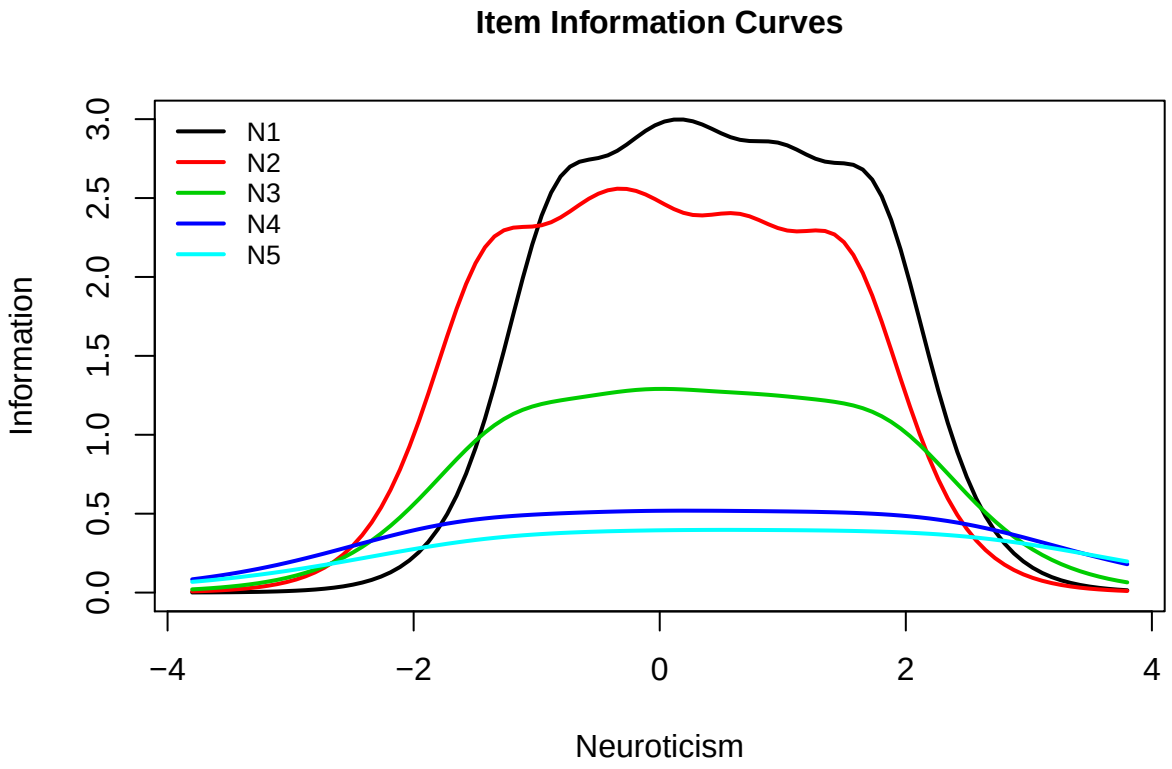
```
plot(fit2.consc, type = "IIC", lwd = 2,
     cex = 0.8, legend = TRUE,
     cx = "topleft", xlab = "Conscientiousness",
     cex.main = 1, cex.lab = 1, cex.axis = 1)
```



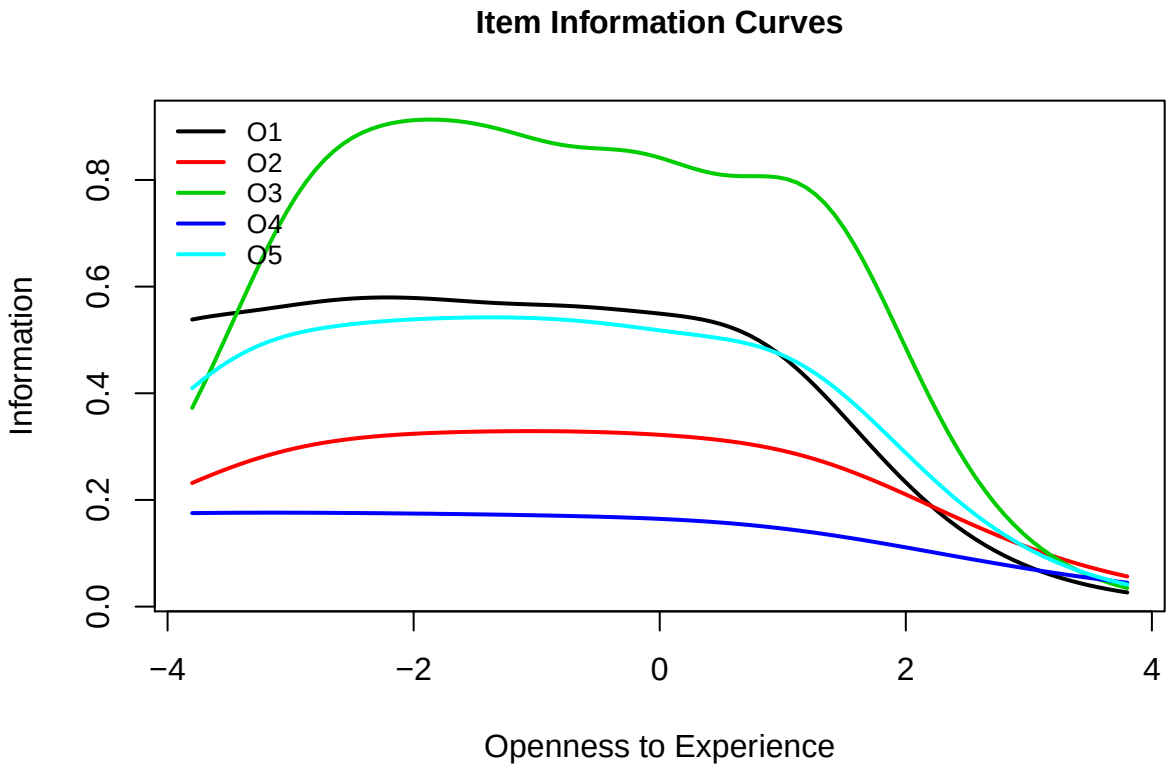
```r
plot(fit2.extra, type = "IIC", lwd = 2,
     cex = 0.8, legend = TRUE,
     cx = "topleft", xlab = "Extraversion",
     cex.main = 1, cex.lab = 1, cex.axis = 1)
```



```
plot(fit2.neuro, type = "IIC", lwd = 2,
     cex = 0.8, legend = TRUE,
     cx = "topleft", xlab = "Neuroticism",
     cex.main = 1, cex.lab = 1, cex.axis = 1)
```



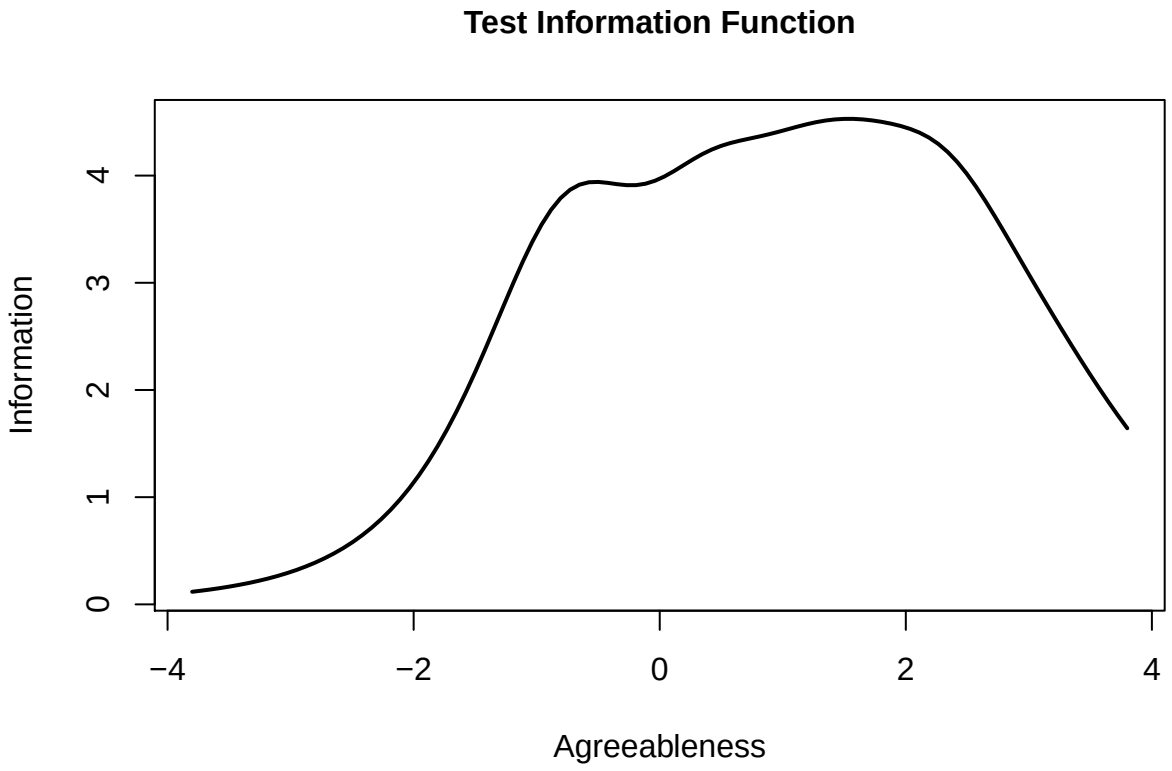
```
plot(fit2.open, type = "IIC", lwd = 2,
     cex = 0.8, legend = TRUE,
     cx = "topleft", xlab = "Openness to Experience",
     cex.main = 1, cex.lab = 1, cex.axis = 1)
```



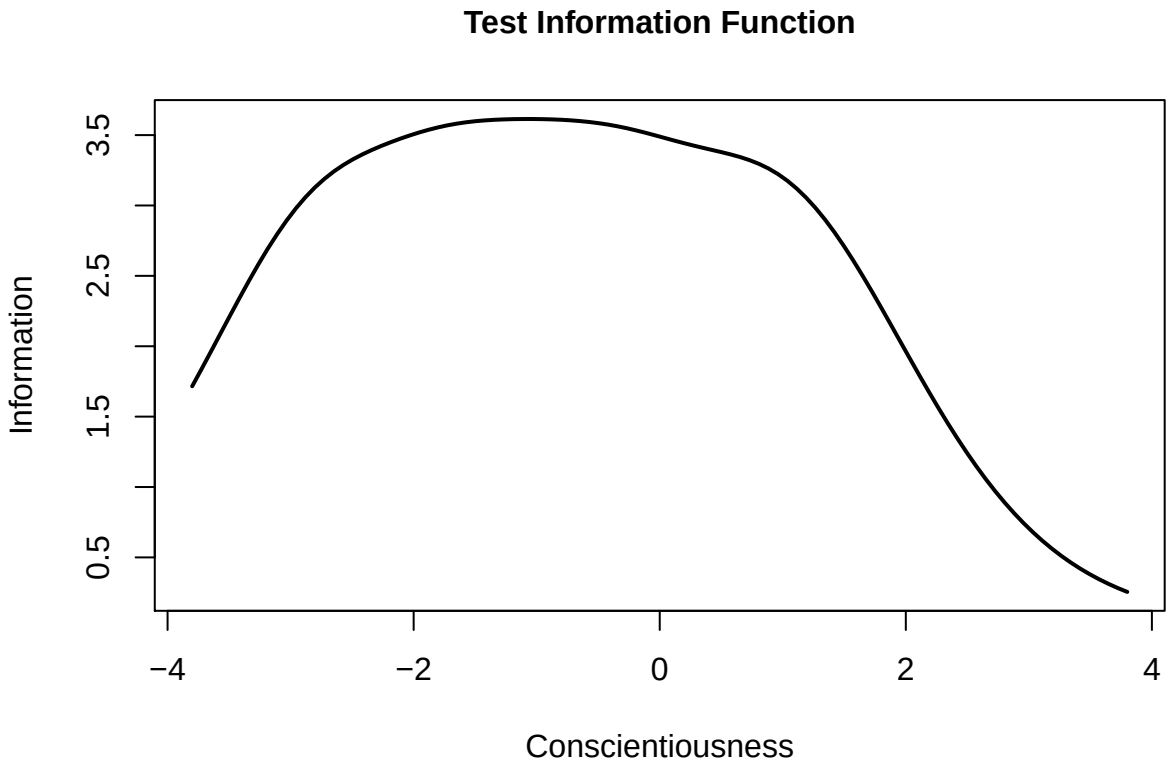
Using the Test Information Function

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

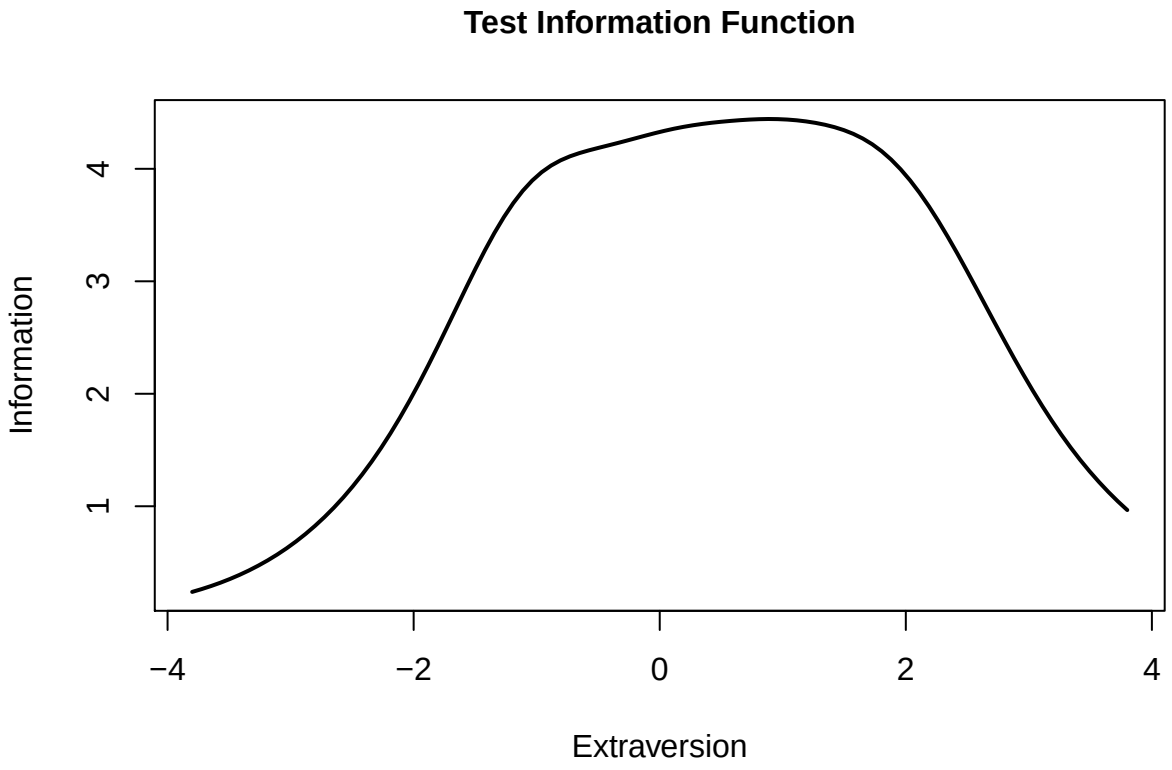
```
plot(fit2.agree, type = "IIC", items = 0,
     lwd = 2, xlab = "Agreeableness",
     cex.main = 1, cex.lab = 1, cex.axis = 1)
```



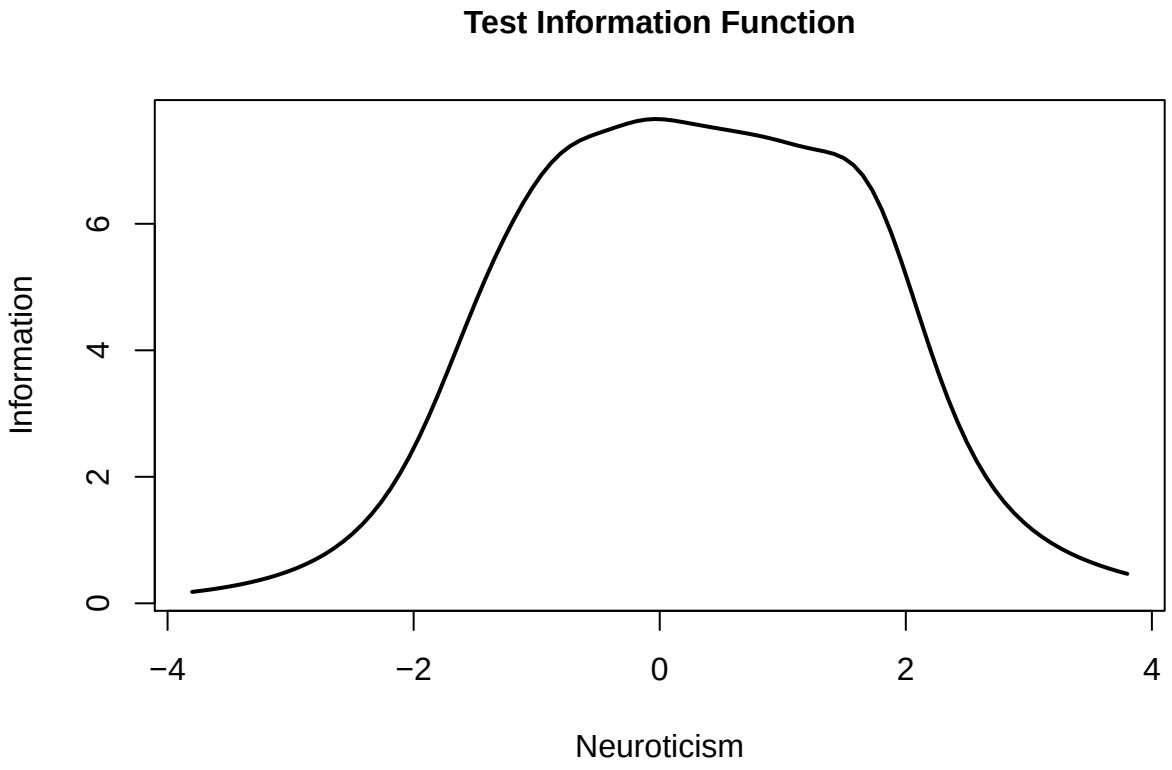
```r
plot(fit2.consc, type = "IIC", items = 0,
     lwd = 2, xlab = "Conscientiousness",
     cex.main = 1, cex.lab = 1, cex.axis = 1)
```



```
plot(fit2.extra, type = "IIC", items = 0,
     lwd = 2, xlab = "Extraversion",
     cex.main = 1, cex.lab = 1, cex.axis = 1)
```



```
plot(fit2.neuro, type = "IIC", items = 0,
     lwd = 2, xlab = "Neuroticism",
     cex.main = 1, cex.lab = 1, cex.axis = 1)
```



```
plot(fit2.open, type = "IIC", items = 0,
     lwd = 2, xlab = "Openness to Experience",
     cex.main = 1, cex.lab = 1, cex.axis = 1)
```