Modelos politómicos con el paquete `mirt` de R

José Ant. López Pina

Índice

Capítulo 1

Introducción

En este documento se presenta un ejemplo de cómo estimar parámetros en los modelos politómicos con el paquete `mirt` y se realiza un análisis clásico de ítems con el paquete `psych`. Antes de realizar cualquier análisis, debe instalar los paquetes correspondientes con `install.packages("mirt")` e `install.packages("psych")` y cargar la librería cuando proceda con `library(psych)` o `library(mirt)`.

Capítulo 2

Lectura de la base de datos

El primer paso consiste en abrir una carpeta donde se encontrarán los datos de los tests que se desean analizar. Por ejemplo, en el disco principal `(C:)` abrimos una carpeta `C:datos` para alojar los archivos de datos. Para ello, utilizaremos la función `setwd("C:/datos")`. Esta carpeta permanecerá activa en tanto no se cambie de denominación o se borre del disco duro. A continuación, se asigna la matriz de datos `rse.dat` a un objeto. El objeto puede tener cualquier nombre, por ejemplo, `rse0`. Para ello utilizamos la función siguiente: `rse0 <- read.table("rse.dat", header=T, sep=",")`.

La función `read.table` permite leer el archivo `rse.dat` que contiene, en la primera fila, los nombres de las variables y los ítems de la escala separados (`sep`) por comas. El archivo se asigna al objeto `rse0` que contendrá la matriz de datos que se desea analizar.

Dado que posteriormente se realizará un estudio del sesgo de los ítems (DIF), el archivo contiene una columna denominada como `Genero` que debe definirse en `R`, y que más adelante se utilizará en el estudio del DIF. No obstante, para estimar los parámetros de dificultad con los modelos politómicos TRI se debe eliminar de la matriz de datos, asignando el resultado a un nuevo objeto `rse1 <- rse0[, -c(1,12)]`, donde se han eliminado las columnas que contienen el número de identificación y el género

La base de datos contiene las respuestas de `500` personas a `10` ítems en una escala de autoestima. Sin embargo, dado que algunos ítems (`Q2`, `Q5`, `Q8`, `Q9` y `Q10`) están invertidos, el primer paso consistirá en reordenar las categorías para que el paquete `mirt` realice las estimaciones de parámetros apropiadamente. Para ello, se deben recodificar los ítems con la función `reverse.score()` incluida en este mismo paquete. En esta función se deben especificar los ítems que serán invertidos en un vector columna junto al objeto que tiene los datos, tal como aparece a continuación:

```r
library(mirt)
```

```
Cargando paquete requerido: stats4
```

```
Cargando paquete requerido: lattice
```

```r
setwd("c:/datos/")
rse0 <- read.table("rse.dat", header=T, sep=",")
rse1 <- rse0[, -c(1,12)]
rse <- reverse.score(rse1, c('Q2', 'Q5', 'Q8', 'Q9', 'Q10'))
```

Para comprobar que todo el proceso ha sido correcto se pueden obtener los seis primeros casos de cada objeto. Por ejemplo, con:

```r
head(rse1)
```

```
  Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10
1  2  1  1  2  3  1  1  2  2   2
2  3  3  3  3  0  3  3  0  0   0
3  3  1  2  2  1  1  1  2  2   2
4  3  0  2  2  1  2  3  0  1   1
5  0  1  1  3  2  1  2  0  1   2
6  3  0  3  3  2  2  2  2  0   0
```

aparecen los valores originales y con:

```r
head(rse)
```

```
  Q1 Q2.RS Q3 Q4 Q5.RS Q6 Q7 Q8.RS Q9.RS Q10.RS
1  2     2  1  2     0  1  1     1     1      1
2  3     0  3  3     3  3  3     3     3      3
3  3     2  2  2     2  1  1     1     1      1
4  3     3  2  2     2  2  3     3     2      2
5  0     2  1  3     1  1  2     3     2      1
6  3     3  3  3     1  2  2     1     3      3
```

aparecen los valores recodificados de los ítems `Q2`, `Q5`, `Q8`, `Q9` y `Q10`. Nótese que los ítems invertidos aparecen con la etiqueta `QX.RS`, donde `X` equivale al número del ítem invertido. Antes de iniciar los análisis conviene evaluar la dimensionalidad de la matriz de datos con:

```r
dim(rse)
```

```
[1] 500  10
```

También se puede examinar la distribución de respuestas en las cuatro categorías de respuesta para los 10 ítems con:

```r
apply(rse, 2, table)
```

```
   Q1 Q2.RS  Q3  Q4 Q5.RS  Q6  Q7 Q8.RS Q9.RS Q10.RS
0  32    19  61  23    74  71  93    85   136    130
1  85    66 141 107   159 178 174   225   234    184
2 243   279 211 266   185 191 176   124    82    106
3 140   136  87 104    82  60  57    66    48     80
```

y se pueden obtener los estadísticos básicos de los ítems con:

```r
summary(rse)
```

```
       Q1             Q2.RS             Q3              Q4             Q5.RS     
 Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.00  
 1st Qu.:2.000   1st Qu.:2.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.00  
 Median :2.000   Median :2.000   Median :2.000   Median :2.000   Median :2.00  
 Mean   :1.982   Mean   :2.064   Mean   :1.648   Mean   :1.902   Mean   :1.55  
 3rd Qu.:3.000   3rd Qu.:3.000   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:2.00  
 Max.   :3.000   Max.   :3.000   Max.   :3.000   Max.   :3.000   Max.   :3.00  
       Q6             Q7             Q8.RS           Q9.RS           Q10.RS     
 Min.   :0.00   Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000  
 1st Qu.:1.00   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:0.000   1st Qu.:0.000  
 Median :2.00   Median :1.000   Median :1.000   Median :1.000   Median :1.000  
 Mean   :1.48   Mean   :1.394   Mean   :1.342   Mean   :1.084   Mean   :1.272  
```

```
 3rd Qu.:2.00   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:2.000
 Max.   :3.00   Max.   :3.000   Max.   :3.000   Max.   :3.000   Max.   :3.000
```

Es conveniente asegurarse en esta salida que los valores mínimo (`Min.`) y máximo (`Max.`) de cada ítem se encuentran entre los valores mínimo y máximo especificados por la escala.

Capítulo 3

Análisis clásico de ítems

Previo a la aplicación de los modelos politómicos, se puede realizar un análisis descriptivo básico de todos los ítems. Para ello, empleamos el paquete `psych` con:

```
detach(package:mirt)
library(psych)
describe(rse)
```

```
       vars   n mean   sd median trimmed  mad min max range  skew kurtosis   se
Q1        1 500 1.98 0.84      2    2.06 1.48   0   3     3 -0.61    -0.11 0.04
Q2.RS     2 500 2.06 0.74      2    2.13 0.00   0   3     3 -0.66     0.49 0.03
Q3        3 500 1.65 0.91      2    1.69 1.48   0   3     3 -0.23    -0.72 0.04
Q4        4 500 1.90 0.77      2    1.94 0.00   0   3     3 -0.43    -0.07 0.03
Q5.RS     5 500 1.55 0.93      2    1.56 1.48   0   3     3 -0.09    -0.87 0.04
Q6        6 500 1.48 0.88      2    1.48 1.48   0   3     3 -0.04    -0.72 0.04
Q7        7 500 1.39 0.92      1    1.37 1.48   0   3     3  0.03    -0.85 0.04
Q8.RS     8 500 1.34 0.91      1    1.30 1.48   0   3     3  0.32    -0.68 0.04
Q9.RS     9 500 1.08 0.90      1    0.98 1.48   0   3     3  0.62    -0.31 0.04
Q10.RS   10 500 1.27 1.02      1    1.22 1.48   0   3     3  0.34    -1.00 0.05
```

Ahora conviene examinar las medias y los índices de discriminación de los ítems de esta escala con:

```
alpha(rse)
```

```

Reliability analysis
Call: alpha(x = rse)

  raw_alpha std.alpha G6(smc) average_r S/N    ase mean   sd median_r
      0.91      0.91    0.91      0.49 9.7 0.0062  1.6 0.65     0.49

    95% confidence boundaries
         lower alpha upper
Feldt     0.89  0.91  0.92
Duhachek  0.89  0.91  0.92

 Reliability if an item is dropped:
       raw_alpha std.alpha G6(smc) average_r S/N alpha se  var.r med.r
Q1          0.90      0.90    0.90      0.49 8.5   0.0069 0.0105  0.48
```

```
Q2.RS       0.90      0.90     0.90      0.50 8.9   0.0067 0.0100  0.49
Q3          0.89      0.89     0.90      0.48 8.4   0.0071 0.0105  0.48
Q4          0.90      0.90     0.91      0.51 9.2   0.0065 0.0108  0.51
Q5.RS       0.89      0.89     0.90      0.49 8.5   0.0070 0.0112  0.48
Q6          0.89      0.89     0.89      0.48 8.3   0.0072 0.0098  0.48
Q7          0.89      0.89     0.89      0.47 8.1   0.0073 0.0091  0.48
Q8.RS       0.91      0.91     0.91      0.52 9.7   0.0062 0.0074  0.51
Q9.RS       0.90      0.90     0.90      0.50 8.9   0.0067 0.0101  0.51
Q10.RS      0.89      0.89     0.89      0.48 8.4   0.0071 0.0105  0.48

 Item statistics
         n raw.r std.r r.cor r.drop mean   sd
Q1     500  0.75  0.76  0.73   0.68  2.0 0.84
Q2.RS  500  0.69  0.71  0.67   0.62  2.1 0.74
Q3     500  0.79  0.78  0.76   0.73  1.6 0.91
Q4     500  0.65  0.67  0.61   0.58  1.9 0.77
Q5.RS  500  0.76  0.76  0.73   0.69  1.6 0.93
Q6     500  0.80  0.80  0.78   0.74  1.5 0.88
Q7     500  0.82  0.82  0.80   0.76  1.4 0.92
Q8.RS  500  0.60  0.60  0.52   0.50  1.3 0.91
Q9.RS  500  0.71  0.70  0.67   0.63  1.1 0.90
Q10.RS 500  0.79  0.77  0.75   0.71  1.3 1.02

Non missing response frequency for each item
          0    1    2    3 miss
Q1     0.06 0.17 0.49 0.28    0
Q2.RS  0.04 0.13 0.56 0.27    0
Q3     0.12 0.28 0.42 0.17    0
Q4     0.05 0.21 0.53 0.21    0
Q5.RS  0.15 0.32 0.37 0.16    0
Q6     0.14 0.36 0.38 0.12    0
Q7     0.19 0.35 0.35 0.11    0
Q8.RS  0.17 0.45 0.25 0.13    0
Q9.RS  0.27 0.47 0.16 0.10    0
Q10.RS 0.26 0.37 0.21 0.16    0
```

que produce una pléyade de estadísticos para todos los ítems. En este caso, la media de cada ítem se encuentra en la columna `mean` y el índice de discriminación en la columna `r.drop`.

Finalmente, se pueden obtener diferentes coeficientes de fiabilidad en esta escala utilizando, de nuevo, el paquete `psych` con:

```r
splitHalf(rse)
```

```
Split half reliabilities
Call: splitHalf(r = rse)

Maximum split half reliability (lambda 4) =  0.94
Guttman lambda 6                          =  0.91
Average split half reliability            =  0.91
Guttman lambda 3 (alpha)                  =  0.91
Guttman lambda 2                          =  0.91
Minimum split half reliability  (beta)    =  0.86
```

```
Average interitem r =  0.49  with median =  0.49
```

que produce los coeficientes de consistencia interna alfa (`alpha`), omega(`omega.tot`), una medida de mitades τ-equivalentes (`tau`), una medida de tests congenéricos (`con`) y una medida experimental de la unidimensionalidad de la escala (`Uni`), y los coeficientes `lambda_2`, `lambda_3` y `lambda_6` de Guttman. Una estimación del `mayor límite menor` de Guttman se obtiene con:

```
glb.algebraic(rse)
```

```
Loading required namespace: Rcsdp

$glb
[1] 0.9416232

$solution
       Q1     Q2.RS        Q3        Q4     Q5.RS        Q6        Q7     Q8.RS
0.4288371 0.4913827 0.5705609 0.2954630 0.5598519 0.6784633 0.6592813 0.3214192
    Q9.RS    Q10.RS
0.5666536 0.8000429

$status
[1] 0

$Call
glb.algebraic(Cov = rse)
```

```
detach(package:psych)
```

Capítulo 4

Modelo de Crédito Parcial

4.1 Estimación de parámetros

Este modelo fue diseñado para tests con ítem politómicos donde se espera que los examinados muestren un conocimiento parcial en los ítems, con la particularidad de que todos los ítems tienen el mismo parámetro de discriminación.

La estimación de parámetros en este modelo con el paquete `mirt` se realiza con la función del modelo de crédito parcial generalizado, restringiendo que todos los parámetros de discriminación sean iguales a `1`, tal como aparece a continuación:

```
library(mirt)
esp <- 'F=1-10
START = (1-10, a1, 1.0)
FIXED = (1-10, a1)
FREE = (GROUP, COV_11)'
m.mcp <- mirt(rse, model=esp, itemtype="gpcm", SE=T, verbose=F)
```

En este caso, se crea el objeto `esp` que contiene, en una sola dimensión, los `10` ítems de la escala de autoestima. Los argumentos `START` y `FIXED` fijan que los parámetros de discriminación sean todos iguales a `1` y el argumento `FREE` permite que la varianza de las estimaciones de los parámetros varíe libremente.

El argumento `model=esp` especifica que los parámetros de los ítems se estimarán para una sola dimensión interpretativa con la restricción de que todos los parámetros de discriminación serán iguales a `1`. El argumento `itemtype` especifica el nombre del modelo (`gpcm`). El argumento `SE=T` pide que se calculen los errores típicos de los parámetros. Finalmente, el argumento `verbose=F` evita que se reproduzcan los resultados de todas las iteraciones en el texto.

Para obtener los resultados de la convergencia en la estimación de parámetros y los estadísticos de ajuste global emplearemos:

```
m.mcp
```

```

Call:
mirt(data = rse, model = esp, itemtype = "gpcm", SE = T, verbose = F)

Full-information item factor analysis with 1 factor(s).
Converged within 1e-04 tolerance after 25 EM iterations.
mirt version: 1.44.0
```

```
M-step optimizer: nlminb
EM acceleration: Ramsay
Number of rectangular quadrature: 61
Latent density type: Gaussian

Information matrix estimated with method: Oakes
Second-order test: model is a possible local maximum
Condition number of information matrix =  138.0979

Log-likelihood = -5082.696
Estimated parameters: 31
AIC = 10227.39
BIC = 10358.04; SABIC = 10259.65
G2 (1048544) = 4201.1, p = 1
RMSEA = 0, CFI = NaN, TLI = NaN
```

y para comprobar la convergencia de la estimación de parámetros emplearemos:

```
extract.mirt(m.mcp, what="converged")
```

```
[1] TRUE
```

cuyo resultado (`TRUE`) indica que no se ha encontrado problemas en la aplicación del método de máxima verosimilitud marginal para estimar los parámetros de los ítems y de la habilidad.

Una tabla simplicada con las estimaciones de los parámetros de los ítems aparece con:

```
coef(m.mcp, simplify=T, IRTpars=T)
```

```
$items
       a     b1     b2    b3
Q1     1 -2.827 -1.726 1.195
Q2.RS  1 -3.334 -2.276 1.312
Q3     1 -2.240 -0.631 1.943
Q4     1 -3.408 -1.502 1.758
Q5.RS  1 -2.028 -0.249 1.966
Q6     1 -2.154 -0.079 2.473
Q7     1 -1.720  0.086 2.512
Q8.RS  1 -2.021  0.785 2.057
Q9.RS  1 -1.263  1.577 2.282
Q10.RS 1 -1.212  0.805 1.655

$means
F
0

$cov
      F
F 2.609
```

Para cada ítem aparece el parámetro de discriminación en la columna (`a`) y los tres parámetros de umbral en las columnas: `b1`, `b2`, `b3`, asociados a las 4 categorías de los ítems. Como se aprecia en la salida, todos los parámetros de discriminación son iguales a `1`. Además, se aprecia que los umbrales de las categorías revelan que todos están ordenados.

Los errores típicos de los parámetros se obtienen con:

```r
coef(m.mcp, IRTpars=T, printSE=T)
```

```
$Q1
     a     b1     b2    b3
par  1 -2.827 -1.726 1.195
SE  NA  0.248  0.159 0.145

$Q2.RS
     a     b1     b2    b3
par  1 -3.334 -2.276 1.312
SE  NA  0.303  0.171 0.144

$Q3
     a     b1     b2    b3
par  1 -2.240 -0.631 1.943
SE  NA  0.192  0.143 0.167

$Q4
     a     b1     b2    b3
par  1 -3.408 -1.502 1.758
SE  NA  0.270  0.150 0.155

$Q5.RS
     a     b1     b2    b3
par  1 -2.028 -0.249 1.966
SE  NA  0.179  0.142 0.173

$Q6
     a     b1     b2    b3
par  1 -2.154 -0.079 2.473
SE  NA  0.179  0.139 0.189

$Q7
     a     b1    b2    b3
par  1 -1.720 0.086 2.512
SE  NA  0.166 0.141 0.194

$Q8.RS
     a     b1    b2    b3
par  1 -2.021 0.785 2.057
SE  NA  0.165 0.145 0.193

$Q9.RS
     a     b1    b2    b3
par  1 -1.263 1.577 2.282
SE  NA  0.146 0.161 0.225

$Q10.RS
     a     b1    b2    b3
par  1 -1.212 0.805 1.655
```

```
SE   NA   0.152 0.153 0.187

$GroupPars
    MEAN_1 COV_11
par      0   2.609
SE      NA   0.220
```

y para obtener los intervalos de confianza de los parámetros se emplea:

```
coef(m.mcp, IRTpars=T)
```

```
$Q1
         a     b1     b2    b3
par      1 -2.827 -1.726 1.195
CI_2.5  NA -3.314 -2.038 0.911
CI_97.5 NA -2.341 -1.413 1.480

$Q2.RS
         a     b1     b2    b3
par      1 -3.334 -2.276 1.312
CI_2.5  NA -3.928 -2.610 1.030
CI_97.5 NA -2.740 -1.941 1.595

$Q3
         a     b1     b2    b3
par      1 -2.240 -0.631 1.943
CI_2.5  NA -2.616 -0.911 1.615
CI_97.5 NA -1.864 -0.351 2.271

$Q4
         a     b1     b2    b3
par      1 -3.408 -1.502 1.758
CI_2.5  NA -3.938 -1.796 1.453
CI_97.5 NA -2.879 -1.209 2.063

$Q5.RS
         a     b1     b2    b3
par      1 -2.028 -0.249 1.966
CI_2.5  NA -2.378 -0.527 1.628
CI_97.5 NA -1.677  0.029 2.305

$Q6
         a     b1     b2    b3
par      1 -2.154 -0.079 2.473
CI_2.5  NA -2.504 -0.352 2.102
CI_97.5 NA -1.804  0.194 2.844

$Q7
         a     b1     b2    b3
par      1 -1.720  0.086 2.512
CI_2.5  NA -2.046 -0.191 2.132
CI_97.5 NA -1.394  0.363 2.893
```

```
$Q8.RS
          a     b1    b2    b3
par       1 -2.021 0.785 2.057
CI_2.5   NA -2.345 0.500 1.678
CI_97.5  NA -1.697 1.071 2.437

$Q9.RS
          a     b1    b2    b3
par       1 -1.263 1.577 2.282
CI_2.5   NA -1.549 1.262 1.841
CI_97.5  NA -0.978 1.892 2.723

$Q10.RS
          a     b1    b2    b3
par       1 -1.212 0.805 1.655
CI_2.5   NA -1.509 0.504 1.288
CI_97.5  NA -0.915 1.106 2.022

$GroupPars
        MEAN_1 COV_11
par          0  2.609
CI_2.5      NA  2.178
CI_97.5     NA  3.040
```

En el paquete `mirt` los umbrales estimados son resultado de la combinación del parámetro de localización y los parámetros de paso. Si se desea conocer los parámetros de localización de los ítems y los parámetros de paso, se debe emplear la siguiente estructura:

```
umbral <- coef(m.mcp, simplify=T, IRTpars=T)$'items'
localpar <- apply(umbral[,-1],1, mean)
paso <- data.frame(umbral[,-1]-localpar)
names(paso) <- c("paso1", "paso2", "paso3")
cbind(localpar,paso)
```

```
          localpar     paso1      paso2    paso3
Q1     -1.11919170 -1.707967 -0.6064849 2.314452
Q2.RS  -1.43240911 -1.901549 -0.8432606 2.744810
Q3     -0.30934265 -1.930752 -0.3214548 2.252207
Q4     -1.05085685 -2.357295 -0.4515897 2.808885
Q5.RS  -0.10340452 -1.924261 -0.1455385 2.069799
Q6      0.08002243 -2.234004 -0.1590926 2.393097
Q7      0.29281538 -2.012717 -0.2068342 2.219551
Q8.RS   0.27390647 -2.294921  0.5115360 1.783385
Q9.RS   0.86519144 -2.128538  0.7117814 1.416756
Q10.RS  0.41607381 -1.628143  0.3888958 1.239248
```

4.2 Evaluación del ajuste

Se puede comprobar el ajuste de la matriz de respuestas al modelo de crédito parcial con:

```
M2(m.mcp, type="M2*")
```

```
            M2 df                p        RMSEA    RMSEA_5    RMSEA_95        SRMSR
```

```
stats 105.9721 24 2.828626e-12 0.08273268 0.0669441 0.09904383 0.09211488
            TLI       CFI
stats 0.891465 0.8958064
```

Parece que esta matriz de respuestas no sigue este modelo, ya que el estadístico `M2` ha resultado altamente significativo ($p < 0.05$). Procede ahora realizar una evaluación del ajuste de los ítems con:

```
itemfit(m.mcp)
```

```
     item   S_X2 df.S_X2 RMSEA.S_X2 p.S_X2
1      Q1 50.448      35      0.030  0.044
2   Q2.RS 39.346      32      0.021  0.174
3      Q3 64.205      35      0.041  0.002
4      Q4 47.119      35      0.026  0.083
5   Q5.RS 37.839      37      0.007  0.431
6      Q6 39.585      37      0.012  0.355
7      Q7 59.523      37      0.035  0.011
8   Q8.RS 92.159      36      0.056  0.000
9   Q9.RS 37.426      36      0.009  0.403
10 Q10.RS 39.868      37      0.012  0.344
```

En este caso, los ítems `Q1`, Q3,Q7yQ8‘ también han alcanzado la significación estadística ($p < 0.05$) por lo que no se puede afirmar que el modelo produzca una buena explicación de los patrones de respuestas correspondientes.

4.3 Parámetros de habilidad

El paquete `mirt` permite estimar los parámetros de habilidad con estimadores `EAP`, `MAP` y `WLE` junto con sus errores típicos. Por ejemplo, para estimar los parámetros de habilidad con la opción `EAP` emplearemos:

```
head(fscores(m.mcp, method="EAP", full.scores=T, full.scores.SE=T))
```

```
               F      SE_F
[1,] -0.83810828 0.4720208
[2,]  2.66352828 0.5668876
[3,]  0.05102518 0.4705244
[4,]  1.84080666 0.4929166
[5,]  0.05102518 0.4705244
[6,]  1.84080666 0.4929166
```

Los estadísticos de ajuste para detectar patrones aberrantes se pueden estimar con:

```
head(personfit(m.mcp, method="EAP"))
```

```
     outfit  z.outfit     infit   z.infit         Zh
1 0.3910400 -1.611042 0.4423978 -1.447217  1.3814019
2 5.1310211  5.025148 3.2870067  3.396173 -5.2717135
3 0.4279726 -1.380579 0.4045224 -1.491183  1.2981200
4 0.6165029 -1.115268 0.5844515 -1.173321  0.8289513
5 2.9064966  2.830240 2.6515826  2.590176 -3.3340338
6 1.4077979  1.122706 1.5310923  1.335648 -0.7498776
```

que produce los estadísticos de ajuste de medias de cuadrados (`infit` y `outfit`), sus correspondientes transformaciones t (`z.infit` y `z.outfit`) y el estadístico l_z (`zh`). Normalmente, el patrón de respuestas se ajusta al modelo si los estadísticos `infit` y `outfit` se encuentran en el intervalo $[0.5, 1.5]$ o el estadístico $l_z < 1.5$.

4.4 Independencia local de los ítems

El paquete `mirt` permite evaluar el principio de independencia local de los ítems con dos procedimientos: `LD` y `Q3`. En este caso, emplearemos el estadístico $Q3$ con:

```r
residuals(m.mcp, type="Q3")
```

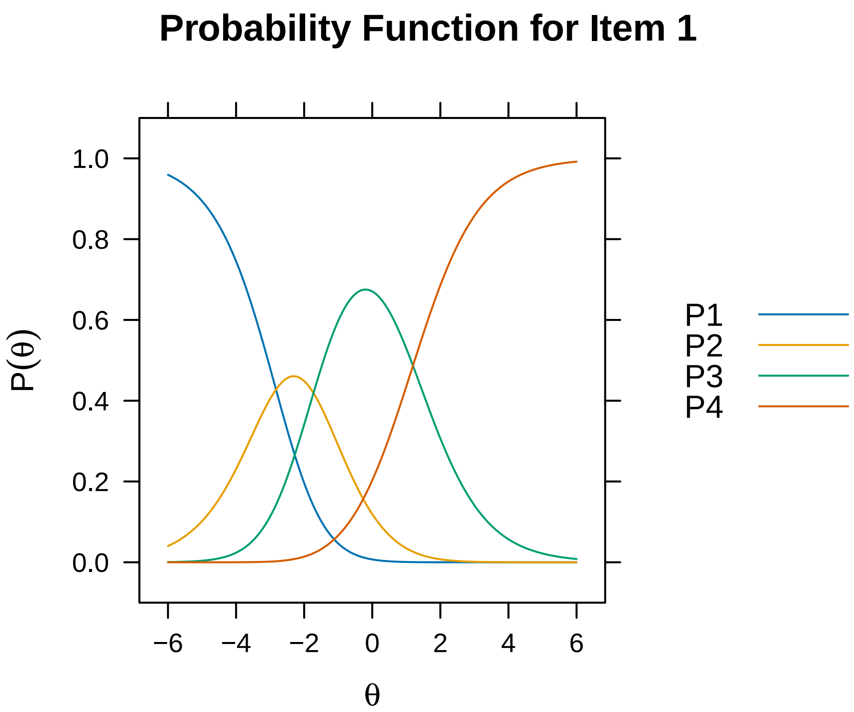
```
Q3 summary statistics:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -0.275  -0.189  -0.131  -0.100  -0.044   0.355

           Q1  Q2.RS     Q3     Q4  Q5.RS     Q6     Q7  Q8.RS  Q9.RS Q10.RS
Q1      1.000  0.250 -0.149 -0.044 -0.174 -0.043  0.013 -0.222 -0.214 -0.152
Q2.RS   0.250  1.000 -0.183  0.064 -0.038 -0.069 -0.092 -0.259 -0.206 -0.208
Q3     -0.149 -0.183  1.000 -0.117  0.103 -0.018 -0.004 -0.163 -0.142 -0.130
Q4     -0.044  0.064 -0.117  1.000 -0.112 -0.248 -0.093 -0.111 -0.134 -0.153
Q5.RS  -0.174 -0.038  0.103 -0.112  1.000 -0.110 -0.087 -0.174 -0.131 -0.191
Q6     -0.043 -0.069 -0.018 -0.248 -0.110  1.000  0.271 -0.010 -0.257 -0.219
Q7      0.013 -0.092 -0.004 -0.093 -0.087  0.271  1.000 -0.189 -0.275 -0.192
Q8.RS  -0.222 -0.259 -0.163 -0.111 -0.174 -0.010 -0.189  1.000 -0.100 -0.139
Q9.RS  -0.214 -0.206 -0.142 -0.134 -0.131 -0.257 -0.275 -0.100  1.000  0.355
Q10.RS -0.152 -0.208 -0.130 -0.153 -0.191 -0.219 -0.192 -0.139  0.355  1.000
```

En la tabla se aprecia que varios estadísticos `Q3` han obtenido valores por encima de `|0.20|`. Por tanto, no se puede asegurar que el principio de indepedencia local se cumpla.

4.5 Gráficos en el modelo de crédito parcial

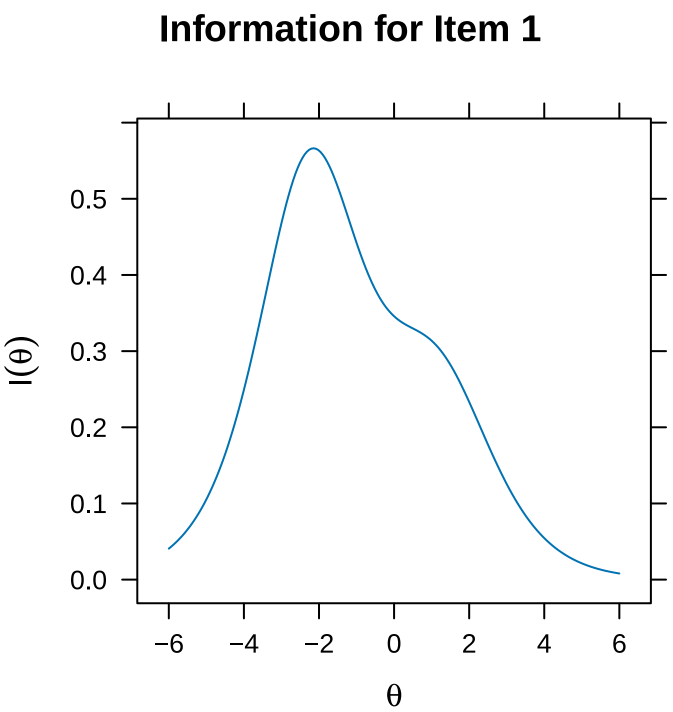
La Función de Respuesta de las Categorías de un ítem (e.g., ítem `1`) se puede representar gráficamente con:

```r
itemplot(m.mcp, 1)
```



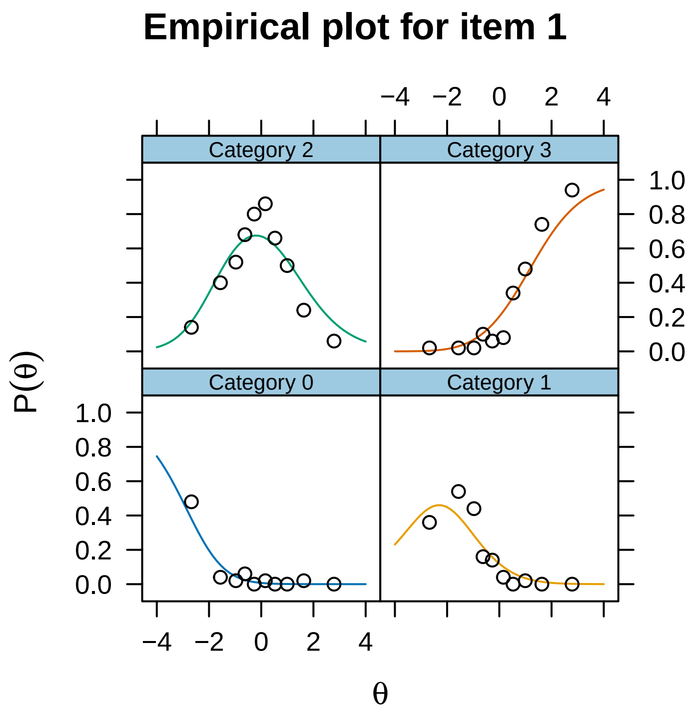
y la Función de Información de ese mismo ítem se obtiene con:

```
itemplot(m.mcp, item=1, type="info")
```



También es posible probar el ajuste de las FRCs empíricas (puntos) a las FRCs teóricas (líneas solidas) del modelo con:

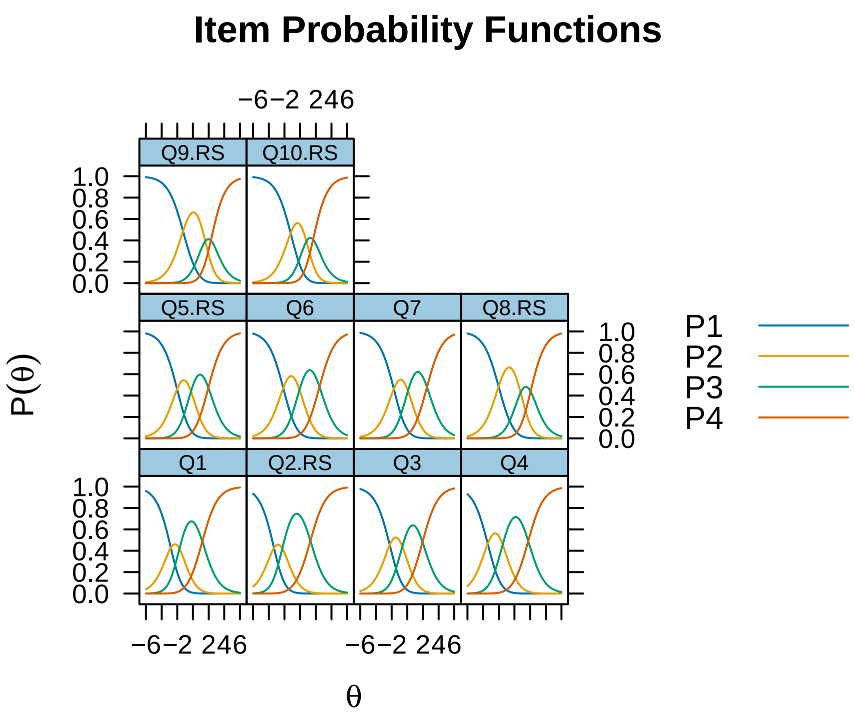
```r
itemfit(m.mcp, empirical.plot=1, empirical.CI=.95)
```



En principio, no parece que el modelo sea capaz de explicar adecuadamente el comportamiento de las personas en las categorías `0` y `3`.

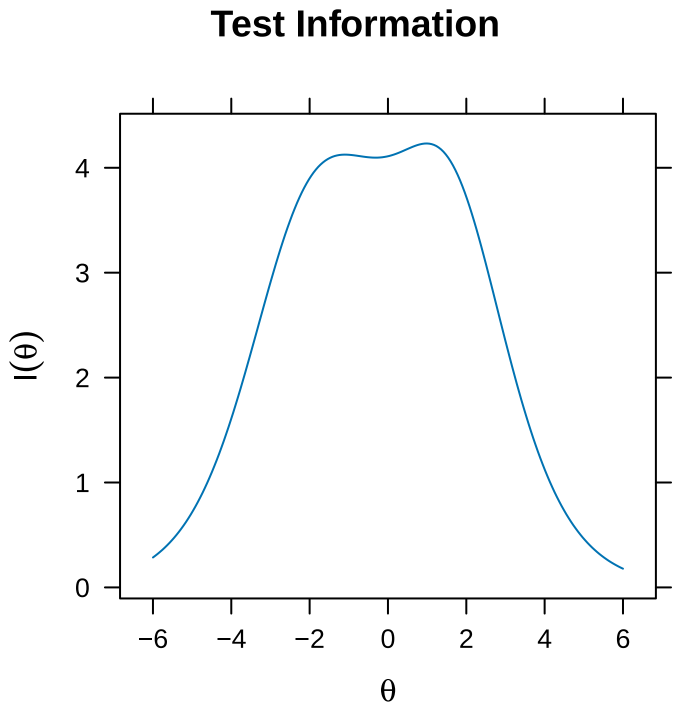
Las FRCs de todos los ítems se pueden obtener con:

```r
plot(m.mcp, type="trace")
```



y la FIT del test completo se puede obtener con:

```
plot(m.mcp, type="info")
```



Capítulo 5

Modelo de Escalas de Valoración

5.1 Estimación de parámetros

Para estimar los parámetros en el modelo de escalas de valoración especificaremos el tipo de modelo `rsm` en la función `mirt` correspondiente:

```
m.mev <- mirt(rse, model=1, itemtype="rsm", SE=T, verbose=F)
```

El argumento `model=1` especifica que los parámetros de los ítems se estimarán para una sola dimensión interpretativa; el argumento `itemtype` especifica el nombre del modelo (`rsm`); el argumento `SE=T` pide que se calculen los errores típicos de los parámetros, y finalmente el argumento `verbose=F` evita que se reproduzcan los resultados de todas las iteraciones en el texto.

Para obtener los resultados de la convergencia en la estimación de parámetros y los estadísticos de ajuste global emplearemos:

```
m.mev
```

```

Call:
mirt(data = rse, model = 1, itemtype = "rsm", SE = T, verbose = F)

Full-information item factor analysis with 1 factor(s).
Converged within 1e-04 tolerance after 23 EM iterations.
mirt version: 1.44.0
M-step optimizer: nlminb
EM acceleration: Ramsay
Number of rectangular quadrature: 61
Latent density type: Gaussian

Information matrix estimated with method: Oakes
Second-order test: model is a possible local maximum
Condition number of information matrix =  34.1323

Log-likelihood = -5179.333
Estimated parameters: 40
AIC = 10384.67
BIC = 10439.46; SABIC = 10398.19
G2 (1048562) = 4394.38, p = 1
```

```
RMSEA = 0, CFI = NaN, TLI = NaN
```

y para comprobar la convergencia de la estimación de parámetros emplearemos:

```r
extract.mirt(m.mev, what="converged")
```

```
[1] TRUE
```

cuyo resultado (`TRUE`) indica que no se ha encontrado problemas en la aplicación del método de máxima verosimilitud marginal.

Se obtiene una tabla simplicada con las estimaciones de los parámetros utilizando:

```r
coef(m.mev, simplify=T, IRTpars=T)
```

```
$items
       a1     b1     b2    b3      c
Q1      1 -3.246 -1.388 1.009  0.000
Q2.RS   1 -3.246 -1.388 1.009  0.226
Q3      1 -3.246 -1.388 1.009 -0.862
Q4      1 -3.246 -1.388 1.009 -0.214
Q5.RS   1 -3.246 -1.388 1.009 -1.105
Q6      1 -3.246 -1.388 1.009 -1.277
Q7      1 -3.246 -1.388 1.009 -1.488
Q8.RS   1 -3.246 -1.388 1.009 -1.616
Q9.RS   1 -3.246 -1.388 1.009 -2.261
Q10.RS  1 -3.246 -1.388 1.009 -1.789

$means
F1
 0

$cov
      F1
F1 2.584
```

Para cada ítem aparece el parámetro de discriminación en la columna (`a`) y los tres parámetros de umbral en las columnas: `b1`, `b2`, `b3`, asociados a las `4` categorías de los ítems. Como se aprecia en la salida, todos los parámetros de discriminación son iguales a `1` y los parámetros de umbral son iguales para todos los ítems. En este caso el parámetro `c` equivale a la facilidad del ítems, no a la dificultad. Además, no se obtiene con el promedio de las categorías de cada ítems, como ocurre en el modelo de crédito parcial.

Se obtiene la identificación del modelo situando el parámetro `c` del primer ítem a `0` aunque la varianza de las estimaciones varía libremente. No obstante, conviene advertir que la parametrización de este modelo en `mirt` no es equivalente a la que se obtiene en el modelo de crédito parcial.

Los errores típicos de los parámetros se obtienen con:

```r
coef(m.mev, IRTpars=T, printSE=T)
```

```
$Q1
    a1     b1     b2    b3  c
par  1 -3.246 -1.388 1.009  0
SE  NA  0.133  0.112 0.111 NA
```

```
$Q2.RS
    a1     b1     b2    b3     c
par  1 -3.246 -1.388 1.009 0.226
SE  NA  0.133  0.112 0.111 0.105

$Q3
    a1     b1     b2    b3      c
par  1 -3.246 -1.388 1.009 -0.862
SE  NA  0.133  0.112 0.111  0.104

$Q4
    a1     b1     b2    b3      c
par  1 -3.246 -1.388 1.009 -0.214
SE  NA  0.133  0.112 0.111  0.103

$Q5.RS
    a1     b1     b2    b3      c
par  1 -3.246 -1.388 1.009 -1.105
SE  NA  0.133  0.112 0.111  0.104

$Q6
    a1     b1     b2    b3      c
par  1 -3.246 -1.388 1.009 -1.277
SE  NA  0.133  0.112 0.111  0.105

$Q7
    a1     b1     b2    b3      c
par  1 -3.246 -1.388 1.009 -1.488
SE  NA  0.133  0.112 0.111  0.106

$Q8.RS
    a1     b1     b2    b3      c
par  1 -3.246 -1.388 1.009 -1.616
SE  NA  0.133  0.112 0.111  0.107

$Q9.RS
    a1     b1     b2    b3      c
par  1 -3.246 -1.388 1.009 -2.261
SE  NA  0.133  0.112 0.111  0.112

$Q10.RS
    a1     b1     b2    b3      c
par  1 -3.246 -1.388 1.009 -1.789
SE  NA  0.133  0.112 0.111  0.108

$GroupPars
    MEAN_1 COV_11
par      0  2.584
SE      NA  0.218
```

y para obtener los intervalos de confianza de los parámetros se emplea:

```r
coef(m.mev, IRTpars=T)
```

```
$Q1
        a1     b1     b2    b3  c
par      1 -3.246 -1.388 1.009  0
CI_2.5  NA -3.507 -1.608 0.792 NA
CI_97.5 NA -2.986 -1.169 1.226 NA

$Q2.RS
        a1     b1     b2    b3     c
par      1 -3.246 -1.388 1.009 0.226
CI_2.5  NA -3.507 -1.608 0.792 0.020
CI_97.5 NA -2.986 -1.169 1.226 0.432

$Q3
        a1     b1     b2    b3      c
par      1 -3.246 -1.388 1.009 -0.862
CI_2.5  NA -3.507 -1.608 0.792 -1.065
CI_97.5 NA -2.986 -1.169 1.226 -0.660

$Q4
        a1     b1     b2    b3      c
par      1 -3.246 -1.388 1.009 -0.214
CI_2.5  NA -3.507 -1.608 0.792 -0.417
CI_97.5 NA -2.986 -1.169 1.226 -0.011

$Q5.RS
        a1     b1     b2    b3      c
par      1 -3.246 -1.388 1.009 -1.105
CI_2.5  NA -3.507 -1.608 0.792 -1.309
CI_97.5 NA -2.986 -1.169 1.226 -0.901

$Q6
        a1     b1     b2    b3      c
par      1 -3.246 -1.388 1.009 -1.277
CI_2.5  NA -3.507 -1.608 0.792 -1.483
CI_97.5 NA -2.986 -1.169 1.226 -1.072

$Q7
        a1     b1     b2    b3      c
par      1 -3.246 -1.388 1.009 -1.488
CI_2.5  NA -3.507 -1.608 0.792 -1.696
CI_97.5 NA -2.986 -1.169 1.226 -1.280

$Q8.RS
        a1     b1     b2    b3      c
par      1 -3.246 -1.388 1.009 -1.616
CI_2.5  NA -3.507 -1.608 0.792 -1.826
CI_97.5 NA -2.986 -1.169 1.226 -1.407

$Q9.RS
```

```
        a1     b1     b2    b3      c
par      1 -3.246 -1.388 1.009 -2.261
CI_2.5  NA -3.507 -1.608 0.792 -2.482
CI_97.5 NA -2.986 -1.169 1.226 -2.041

$Q10.RS
        a1     b1     b2    b3      c
par      1 -3.246 -1.388 1.009 -1.789
CI_2.5  NA -3.507 -1.608 0.792 -2.001
CI_97.5 NA -2.986 -1.169 1.226 -1.577

$GroupPars
        MEAN_1 COV_11
par          0  2.584
CI_2.5      NA  2.157
CI_97.5     NA  3.011
```

5.2 Evaluación del ajuste

Se puede comprobar el ajuste de la matriz de respuestas al modelo de escalas de valoración con:

```
M2(m.mev, type="M2*")
```

```
            M2 df p     RMSEA   RMSEA_5  RMSEA_95      SRMSR       TLI
stats 398.7297 42 0 0.1304652 0.1188214 0.1421675 0.09703617 0.7300988
            CFI
stats 0.5465659
```

Parece que esta matriz de respuestas no sigue el modelo de crédito parcial generalizado, ya que el estadístico `M2` ha resultado altamente significativo ($p < 0.05$). Procede ahora realizar una evaluación del ajuste de los ítems con:

```
itemfit(m.mev)
```

```
     item    S_X2 df.S_X2 RMSEA.S_X2 p.S_X2
1      Q1  55.190      36      0.033  0.021
2   Q2.RS  68.049      31      0.049  0.000
3      Q3  66.781      34      0.044  0.001
4      Q4  86.875      36      0.053  0.000
5   Q5.RS  40.511      36      0.016  0.278
6      Q6  45.749      37      0.022  0.153
7      Q7  67.237      36      0.042  0.001
8   Q8.RS 120.478      36      0.069  0.000
9   Q9.RS  63.057      33      0.043  0.001
10 Q10.RS  88.908      34      0.057  0.000
```

En este caso, solo los ítems `Q5` y `Q6` parecen seguir el modelo ($p > 0.05$). Por tanto, no se puede afirmar que el modelo produzca una buena explicación de los patrones de respuestas correspondientes.

5.3 Parámetros de habilidad

Los parámetros de habilidad se pueden obtener con la opción `EAP`. Por tanto, emplearemos:

```
head(fscores(m.mev, method="EAP", full.scores=T, full.scores.SE=T))
```

```
               F1     SE_F1
[1,] -0.824715750 0.4535100
[2,]  2.759441223 0.5929226
[3,] -0.001525898 0.4567551
[4,]  1.847054111 0.5196833
[5,] -0.001525898 0.4567551
[6,]  1.847054111 0.5196833
```

y los estadísticos de ajuste para detectar patrones aberrantes se pueden estimar con:

```
head(personfit(m.mev, method="EAP"))
```

```
     outfit   z.outfit     infit    z.infit         Zh
1 0.4139812 -1.7275269 0.4266439 -1.6732741  1.3740198
2 7.9716258  6.0759956 3.9916570  4.1052420 -5.7831269
3 0.4183818 -1.6354600 0.4023400 -1.7085585  1.3368573
4 0.6880353 -0.6877767 0.6888012 -0.6684982  0.6863317
5 2.4256782  2.5645941 2.3512180  2.4719484 -3.3104800
6 1.6097614  1.3581549 1.7672009  1.5967960 -1.4954291
```

que produce los estadísticos de ajuste de medias de cuadrados (`infit` y `outfit`), sus correspondientes transformaciones t (`z.infit` y `z.outfit`) y el estadístico l_z (`zh`). Normalmente, el patrón de respuestas se ajusta al modelo si los estadísticos `infit` y `outfit` se encuentran en el intervalo $[0.5, 1.5]$ o el estadístico $l_z < 1.5$.

5.4 Independencia local de los ítems

El paquete `mirt` permite evaluar el principio de independencia local de los ítems con dos procedimientos: `LD` y `Q3`. En este caso, emplearemos el estadístico LD con:

```
residuals(m.mev, type="LD")
```

```
LD matrix (lower triangle) and standardized residual correlations (upper triangle)

Upper triangle summary:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -1.357  -0.331   0.140  -0.050   0.261   0.722

              Q1    Q2.RS      Q3      Q4   Q5.RS      Q6      Q7   Q8.RS
Q1                  0.303   0.233   0.235  -0.188   0.164   0.186  -0.610
Q2.RS    137.566            0.261   0.246   0.329   0.299   0.316  -0.397
Q3        81.281  102.202          -0.252   0.170   0.151   0.140  -0.274
Q4        82.725   90.650  95.288          -0.253  -0.217   0.225  -0.440
Q5.RS     53.117  162.072  43.238  96.011            0.091   0.161  -0.331
Q6        40.130  133.864  34.291  70.325  12.551            0.263  -0.214
Q7        51.976  149.367  29.197  75.958  38.751 103.984          -0.230
Q8.RS    558.453  235.842 112.324 290.221 164.322  68.552  79.245
Q9.RS   1147.599 2761.738 262.989 500.049 138.473 273.098 164.589 192.015
Q10.RS   181.698  781.083  78.996 251.546 121.111 122.199 114.084 186.174
```

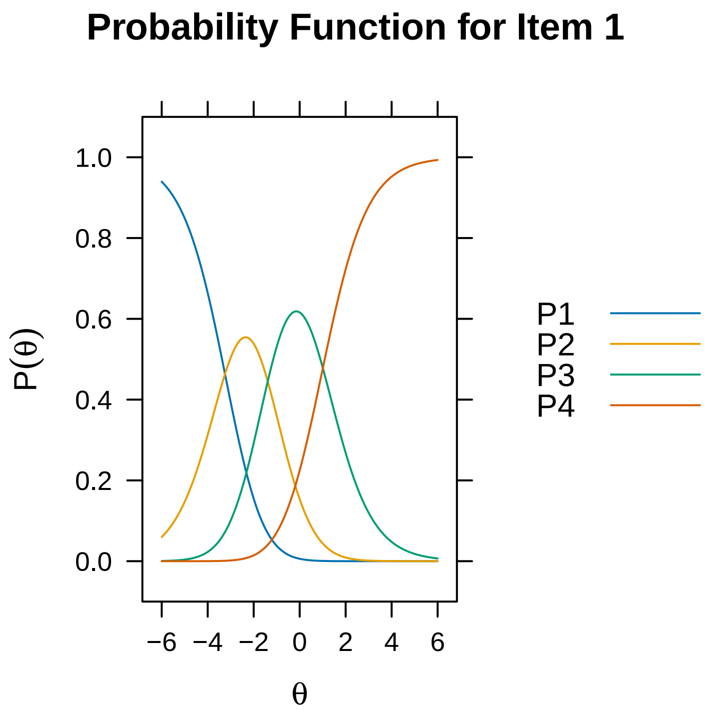
```
         Q9.RS Q10.RS
Q1      -0.875  0.348
Q2.RS   -1.357  0.722
Q3       0.419  0.229
Q4      -0.577 -0.410
Q5.RS   -0.304 -0.284
Q6      -0.427  0.285
Q7      -0.331  0.276
Q8.RS   -0.358 -0.352
Q9.RS           0.361
Q10.RS 195.093
```

Los elementos que aparecen en la diagonal superior de `LD` son equivalentes a la `V` de Cramer calculados a partir de los valores $LD - \chi^2$. Un valor absoluto de este estadístico elevado es un indicador de que no se cumple este principio.

5.5 Gráficos en el modelo de escalas de valoración

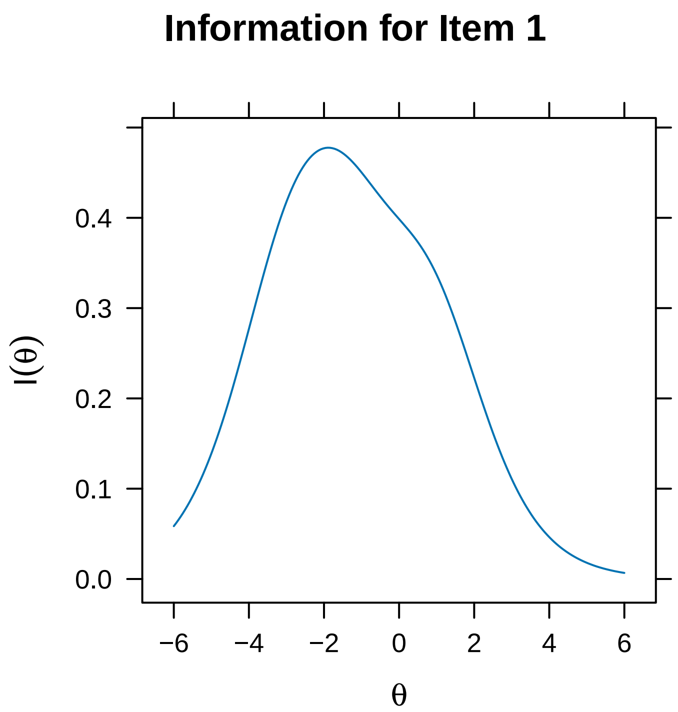
La Función de Respuesta de las Categorías de un ítem (e.g., ítem `1`) se puede representar gráficamente con:

```
itemplot(m.mev, 1)
```



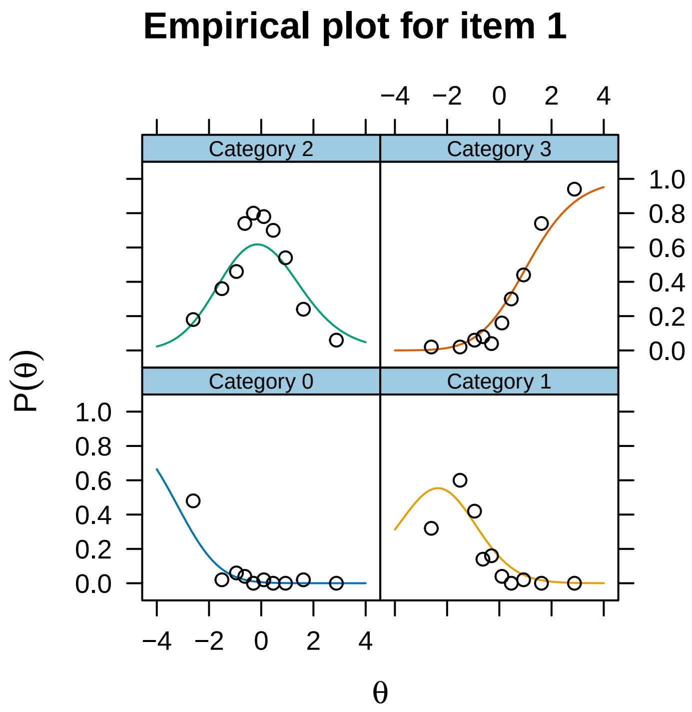
y la Función de Información de ese mismo ítem se obtiene con:

```
itemplot(m.mev, item=1, type="info")
```



También es posible probar el ajuste de las FRCs empíricas (puntos) a las FRCs teóricas (líneas solidas) del modelo con:

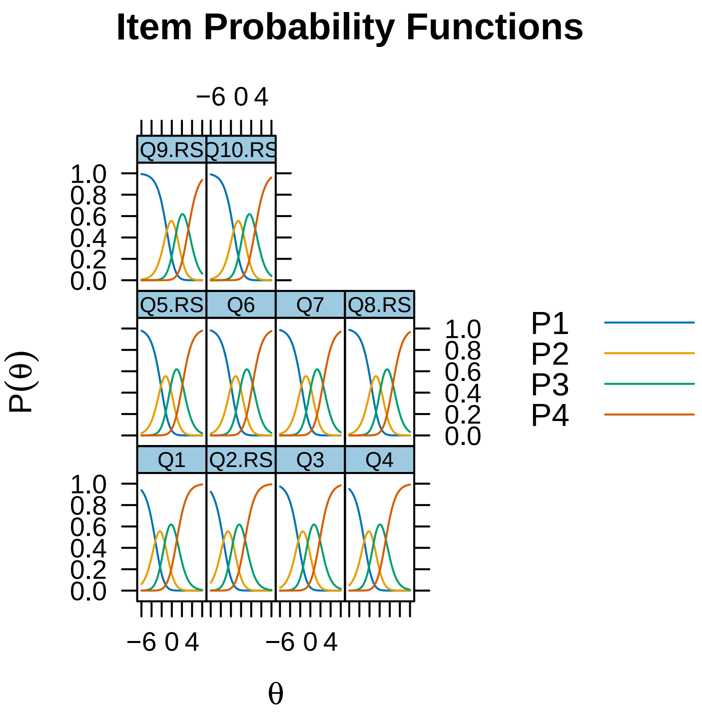
```
itemfit(m.mev, empirical.plot=1, empirical.CI=.95)
```



En principio, no parece que el modelo sea capaz de explicar adecuadamente el comportamiento de las personas en las categorías `2` y `3`.

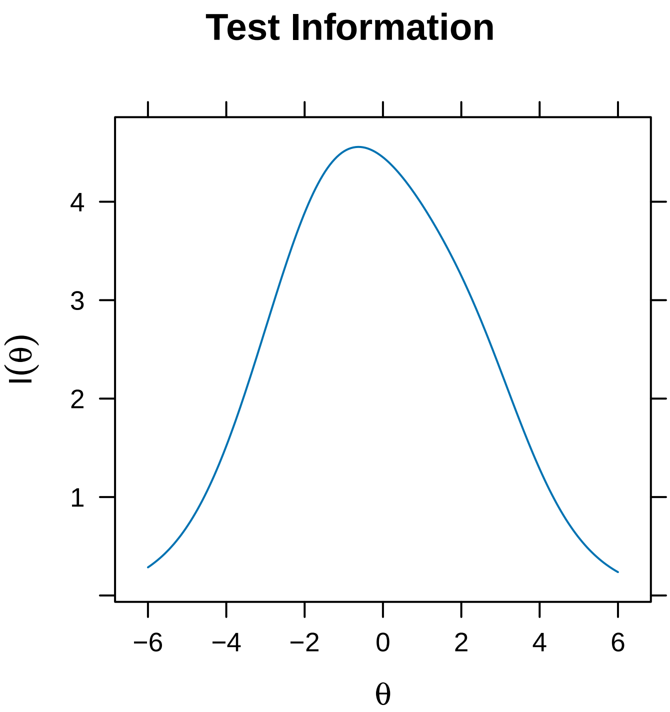
Las FRCs de todos los ítems se pueden obtener con:

```
plot(m.mev, type="trace")
```



y la FIT del test completo se puede obtener con:

```
plot(m.mev, type="info")
```



Capítulo 6

Modelo de Crédito Parcial Generalizado

6.1 Estimación de parámetros

Este modelo fue diseñado especialmente para tests con ítem politómicos donde se espera que los examinados muestren un conocimiento parcial, con la particularidad de que cada ítem puede tener un parámetro de discriminación diferente. El primer paso consiste en estimar los parámetros en este modelo con:

```
m.mcpg <- mirt(rse, model=1, itemtype="gpcm", SE=T, verbose=F)
```

El argumento `model=1` especifica que esperamos que la escala tenga una sola dimensión interpretativa; el argumento `itemtype` especifica el nombre del modelo (`gpcm`); el argumento `SE=T` pide que se calculen los errores típicos de los parámetros, y el argumento `verbose=F` evita que se reproduzcan los resultados de todas las iteraciones en el texto.

Para obtener los resultados de la convergencia en la estimación de parámetros y los estadísticos de ajuste global emplearemos:

```
m.mcpg
```

```

Call:
mirt(data = rse, model = 1, itemtype = "gpcm", SE = T, verbose = F)

Full-information item factor analysis with 1 factor(s).
Converged within 1e-04 tolerance after 163 EM iterations.
mirt version: 1.44.0
M-step optimizer: BFGS
EM acceleration: Ramsay
Number of rectangular quadrature: 61
Latent density type: Gaussian

Information matrix estimated with method: Oakes
Second-order test: model is a possible local maximum
Condition number of information matrix =  307.442

Log-likelihood = -5020.963
Estimated parameters: 40
```

```
AIC = 10121.93
BIC = 10290.51; SABIC = 10163.55
G2 (1048535) = 4077.64, p = 1
RMSEA = 0, CFI = NaN, TLI = NaN
```

y para comprobar la convergencia de la estimación de parámetros emplearemos:

```r
extract.mirt(m.mcpg, what="converged")
```

```
[1] TRUE
```

cuyo resultado (`TRUE`) indica que no se ha encontrado problemas en la aplicación del método de máxima verosimilitud marginal.

Para obtener una tabla simplicada con las estimaciones de los parámetros escribimos:

```r
coef(m.mcpg, simplify=T, IRTpars=T)
```

```
$items
          a     b1     b2    b3
Q1     1.935 -1.703 -0.989 0.708
Q2.RS  1.669 -2.047 -1.387 0.807
Q3     2.121 -1.322 -0.347 1.128
Q4     1.242 -2.280 -1.060 1.217
Q5.RS  1.698 -1.242 -0.152 1.210
Q6     2.445 -1.222 -0.038 1.387
Q7     2.689 -0.986  0.061 1.392
Q8.RS  0.817 -1.679  0.812 1.450
Q9.RS  1.284 -0.839  1.119 1.432
Q10.RS 1.605 -0.753  0.502 1.034

$means
F1
 0

$cov
   F1
F1  1
```

Para cada ítem aparece un parámetro de discriminación (`a`) y tres umbrales (`b1`, `b2`, `b3`) asociados a las `4` categorías de los ítems. En general, los resultados de este modelo indican que los parámetros de discriminación de estos ítems son relativamente elevados, siendo los ítems `Q6` y `Q7` los más discriminativos y los ítems `Q4` y `Q8` los menos discriminativos. Un examen de los umbrales de las categorías revelan que todos están ordenados.

Los errores típicos de los parámetros se obtienen con:

```r
coef(m.mcpg, IRTpars=T, printSE=T)
```

```
$Q1
        a     b1     b2    b3
par 1.935 -1.703 -0.989 0.708
SE  0.189  0.264  0.099 0.086

$Q2.RS
        a     b1     b2    b3
```

```
par 1.669 -2.047 -1.387 0.807
SE  0.165  0.341  0.126 0.095

$Q3
        a     b1     b2    b3
par 2.121 -1.322 -0.347 1.128
SE  0.204  0.164  0.078 0.097

$Q4
        a     b1     b2    b3
par 1.242 -2.280 -1.060 1.217
SE  0.126  0.319  0.127 0.130

$Q5.RS
        a     b1     b2    b3
par 1.698 -1.242 -0.152 1.210
SE  0.162  0.147  0.086 0.111

$Q6
        a     b1     b2    b3
par 2.445 -1.222 -0.038 1.387
SE  0.237  0.127  0.070 0.102

$Q7
        a     b1    b2    b3
par 2.689 -0.986 0.061 1.392
SE  0.266  0.107 0.068 0.100

$Q8.RS
        a     b1    b2    b3
par 0.817 -1.679 0.812 1.450
SE  0.089  0.170 0.168 0.209

$Q9.RS
        a     b1    b2    b3
par 1.284 -0.839 1.119 1.432
SE  0.130  0.122 0.138 0.166

$Q10.RS
        a     b1    b2    b3
par 1.605 -0.753 0.502 1.034
SE  0.160  0.095 0.100 0.118

$GroupPars
    MEAN_1 COV_11
par      0      1
SE      NA     NA
```

y para obtener los intervalos de confianza de los parámetros se emplea:

```
coef(m.mcpg, IRTpars=T)
```

```
$Q1
```

```
            a     b1     b2    b3
par     1.935 -1.703 -0.989 0.708
CI_2.5  1.565 -2.219 -1.184 0.539
CI_97.5 2.306 -1.186 -0.794 0.876

$Q2.RS
            a     b1     b2    b3
par     1.669 -2.047 -1.387 0.807
CI_2.5  1.346 -2.715 -1.634 0.620
CI_97.5 1.992 -1.378 -1.139 0.993

$Q3
            a     b1     b2    b3
par     2.121 -1.322 -0.347 1.128
CI_2.5  1.721 -1.644 -0.499 0.939
CI_97.5 2.520 -1.001 -0.195 1.317

$Q4
            a     b1     b2    b3
par     1.242 -2.280 -1.060 1.217
CI_2.5  0.995 -2.905 -1.310 0.962
CI_97.5 1.489 -1.654 -0.811 1.472

$Q5.RS
            a     b1     b2    b3
par     1.698 -1.242 -0.152 1.210
CI_2.5  1.380 -1.529 -0.320 0.992
CI_97.5 2.016 -0.955  0.016 1.428

$Q6
            a     b1     b2    b3
par     2.445 -1.222 -0.038 1.387
CI_2.5  1.981 -1.471 -0.175 1.186
CI_97.5 2.909 -0.973  0.100 1.588

$Q7
            a     b1     b2    b3
par     2.689 -0.986  0.061 1.392
CI_2.5  2.167 -1.195 -0.073 1.196
CI_97.5 3.210 -0.777  0.194 1.588

$Q8.RS
            a     b1    b2    b3
par     0.817 -1.679 0.812 1.450
CI_2.5  0.643 -2.012 0.484 1.041
CI_97.5 0.991 -1.345 1.141 1.859

$Q9.RS
            a     b1    b2    b3
par     1.284 -0.839 1.119 1.432
CI_2.5  1.029 -1.077 0.848 1.106
```

```
CI_97.5 1.539 -0.601 1.390 1.758

$Q10.RS
            a     b1    b2    b3
par     1.605 -0.753 0.502 1.034
CI_2.5  1.292 -0.939 0.306 0.803
CI_97.5 1.918 -0.567 0.699 1.265

$GroupPars
        MEAN_1 COV_11
par          0      1
CI_2.5      NA     NA
CI_97.5     NA     NA
```

6.2 Evaluación del ajuste

En primer lugar, se puede comprobar si la matriz de datos completa se ajusta al modelo de crédito parcial generalizado con:

```r
M2(m.mcpg, type="M2*")
```

```
            M2 df            p      RMSEA    RMSEA_5  RMSEA_95      SRMSR
stats 72.77438 15 1.425006e-09 0.08785607 0.06818307 0.1084525 0.05621697
            TLI       CFI
stats 0.8776063 0.9265638
```

Parece que esta matriz de respuestas no sigue el modelo de crédito parcial generalizado, ya que el estadístico `M2` ha resultado altamente significativo ($p < 0.05$). Procede ahora realizar una evaluación del ajuste de los ítems con:

```r
itemfit(m.mcpg)
```

```
     item   S_X2 df.S_X2 RMSEA.S_X2 p.S_X2
1      Q1 40.842      34      0.020  0.195
2   Q2.RS 37.842      32      0.019  0.220
3      Q3 48.063      32      0.032  0.034
4      Q4 42.265      37      0.017  0.254
5   Q5.RS 37.023      36      0.008  0.422
6      Q6 30.533      30      0.006  0.439
7      Q7 43.394      28      0.033  0.032
8   Q8.RS 77.720      49      0.034  0.006
9   Q9.RS 47.482      38      0.022  0.139
10 Q10.RS 40.480      37      0.014  0.319
```

En este caso, los ítems `Q3`, `Q7` y `Q8` también han alcanzado la significación estadística ($p < 0.05$) por lo que no se puede afirmar que el modelo produzca una buena explicación de los patrones de respuestas correspondientes.

6.3 Parámetros de habilidad

El paquete `mirt` permite estimar los parámetros de habilidad con estimadores `EAP`, `MAP` y `WLE` junto con sus errores típicos. Por ejemplo, para estimar los parámetros de habilidad con la opción `EAP` emplearemos:

```r
head(fscores(m.mcpg, method="EAP", full.scores=T, full.scores.SE=T))
```

```
              F1     SE_F1
[1,] -0.545489771 0.2715085
[2,]  1.634862309 0.3359461
[3,]  0.008907063 0.2745594
[4,]  1.118628804 0.2927173
[5,] -0.199128815 0.2730199
[6,]  1.138536763 0.2935637
```

Los estadísticos de ajuste para detectar patrones aberrantes se pueden estimar con:

```r
head(personfit(m.mcpg, method="EAP"))
```

```
     outfit   z.outfit     infit    z.infit         Zh
1 0.3809840 -1.6034342 0.4335905 -1.4108682  1.4085497
2 5.2514881  5.1902942 3.2021426  3.4003372 -5.6678033
3 0.5107831 -1.1319418 0.4186392 -1.4404322  1.1849050
4 0.6742579 -0.9402971 0.6706941 -0.8732235  0.6857259
5 2.7448466  2.6718869 2.6933587  2.6247593 -3.0833472
6 1.3158265  0.9423682 1.4440778  1.1705277 -0.4635185
```

que produce los estadísticos de ajuste de medias de cuadrados (`infit` y `outfit`), sus correspondientes transformaciones t (`z.infit` y `z.outfit`) y el estadístico l_z (`zh`). Normalmente, el patrón de respuestas se ajusta al modelo si los estadísticos `infit` y `outfit` se encuentran en el intervalo $[0.5, 1.5]$ o el estadístico $l_z < 1.5$.

6.4 Cumplimiento de supuestos

6.4.1 Unidimensionalidad

Este paquete, en realidad, aplica un análisis factorial no lineal para ofrecer una solución de acuerdo con el modelo de respuesta al ítem especificado. Por tanto, para comprobar si una solución bidimensional ofrece una mejor solución, solo tenemos que repetir la estimación de parámetros cambiando la opción `model=1` por `model=2` y crear un nuevo objeto para examinar los resultados con:

```r
m.mcpg2 <- mirt(rse, model=2, itemtype="gpcm", SE=T, verbose=F)
```

Si escribimos ahora:

```r
m.mcpg2
```

```

Call:
mirt(data = rse, model = 2, itemtype = "gpcm", SE = T, verbose = F)

Full-information item factor analysis with 2 factor(s).
Converged within 1e-04 tolerance after 118 EM iterations.
mirt version: 1.44.0
M-step optimizer: BFGS
EM acceleration: Ramsay
Number of rectangular quadrature: 31
Latent density type: Gaussian
```

```
Information matrix estimated with method: Oakes
Second-order test: model is a possible local maximum
Condition number of information matrix =  2364.595

Log-likelihood = -4939.037
Estimated parameters: 49
AIC = 9976.075
BIC = 10182.59; SABIC = 10027.06
G2 (1048526) = 3913.79, p = 1
RMSEA = 0, CFI = NaN, TLI = NaN
```

se obtienen los estadísticos de ajuste en la solución bidimensional. Todos los estadísticos (`AIC`, `BIC` y `SABIC`) han reducido sus valores en la solución bidimensional. Esto unido a que, en la solución unidimensional, el modelo no fue capaz de ajustar algunos ítems, quizás daría lugar a considerar la posibilidad de que esta escala de autoestima no fuera unidimensional.

6.4.2 Independencia local de los ítems

El paquete `mirt` permite evaluar el principio de independencia local de los ítems con dos procedimientos: `LD` y `Q3`. En este caso, emplearemos el estadístico $Q3$ con:

```
residuals(m.mcpg, type="Q3")
```

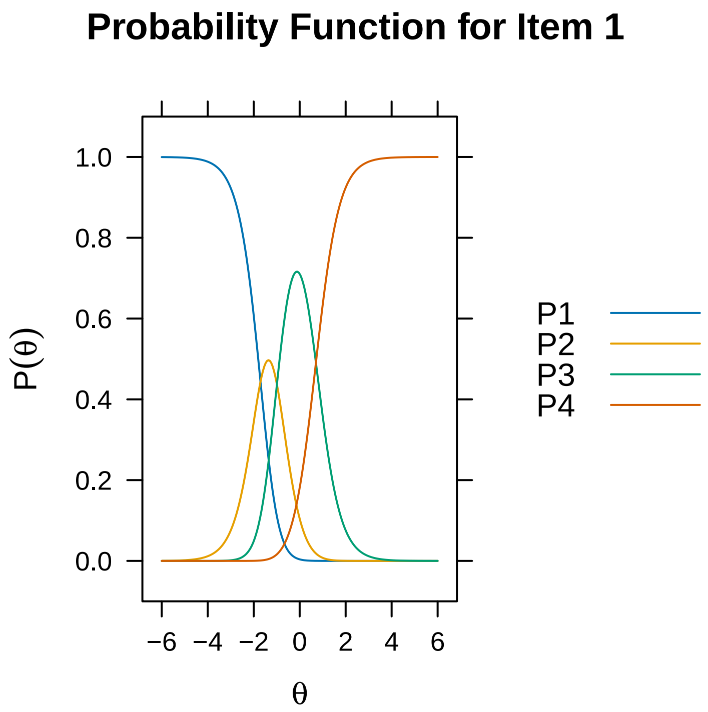
```
Q3 summary statistics:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -0.292  -0.180  -0.128  -0.099  -0.057   0.412

           Q1  Q2.RS     Q3     Q4  Q5.RS     Q6     Q7  Q8.RS  Q9.RS Q10.RS
Q1      1.000  0.229 -0.227 -0.017 -0.204 -0.151 -0.110 -0.132 -0.170 -0.142
Q2.RS   0.229  1.000 -0.227  0.098 -0.042 -0.140 -0.182 -0.172 -0.155 -0.180
Q3     -0.227 -0.227  1.000 -0.092  0.070 -0.146 -0.162 -0.071 -0.101 -0.128
Q4     -0.017  0.098 -0.092  1.000 -0.065 -0.253 -0.098 -0.030 -0.057 -0.080
Q5.RS  -0.204 -0.042  0.070 -0.065  1.000 -0.192 -0.195 -0.082 -0.069 -0.154
Q6     -0.151 -0.140 -0.146 -0.253 -0.192  1.000  0.112  0.071 -0.250 -0.259
Q7     -0.110 -0.182 -0.162 -0.098 -0.195  0.112  1.000 -0.128 -0.292 -0.260
Q8.RS  -0.132 -0.172 -0.071 -0.030 -0.082  0.071 -0.128  1.000  0.004 -0.024
Q9.RS  -0.170 -0.155 -0.101 -0.057 -0.069 -0.250 -0.292  0.004  1.000  0.412
Q10.RS -0.142 -0.180 -0.128 -0.080 -0.154 -0.259 -0.260 -0.024  0.412  1.000
```

En la tabla se aprecia que varios estadísticos `Q3` han obtenido valores por encima de `|0.20|`. Por tanto, no se puede asegurar que el principio de indepedencia local se cumpla.

6.5 Gráficos en el modelo de crédito parcial generalizado

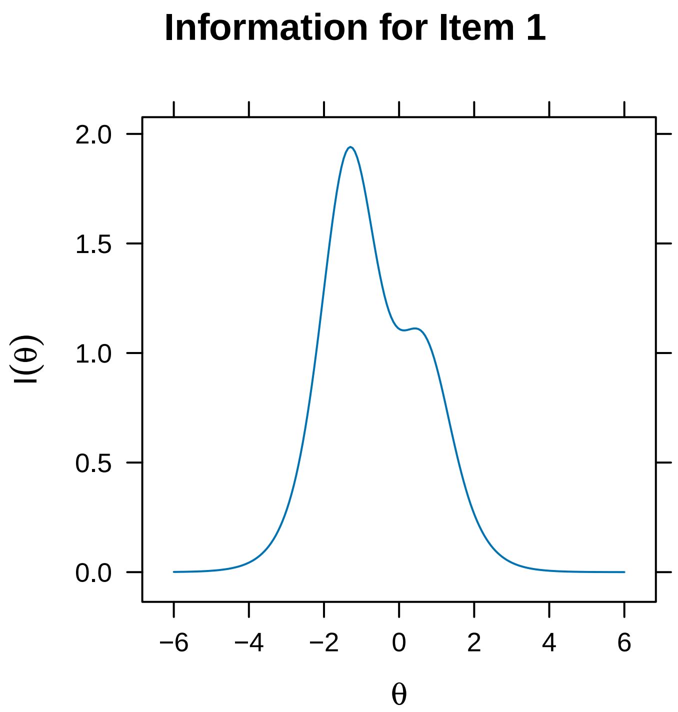
La Función de Respuesta de las Categorías de un ítem (e.g., ítem 1) se puede representar gráficamente con:

```r
itemplot(m.mcpg, 1)
```



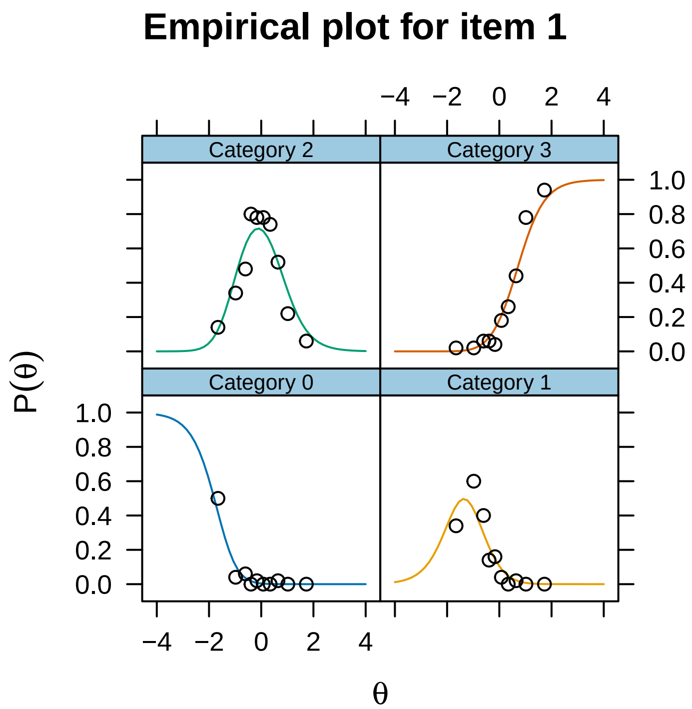
y la Función de Información de ese mismo ítem se obtiene con:

```r
itemplot(m.mcpg, item=1, type="info")
```



También es posible probar el ajuste de las FRCs empíricas (puntos) a las FRCs teóricas (líneas solidas) del modelo con:

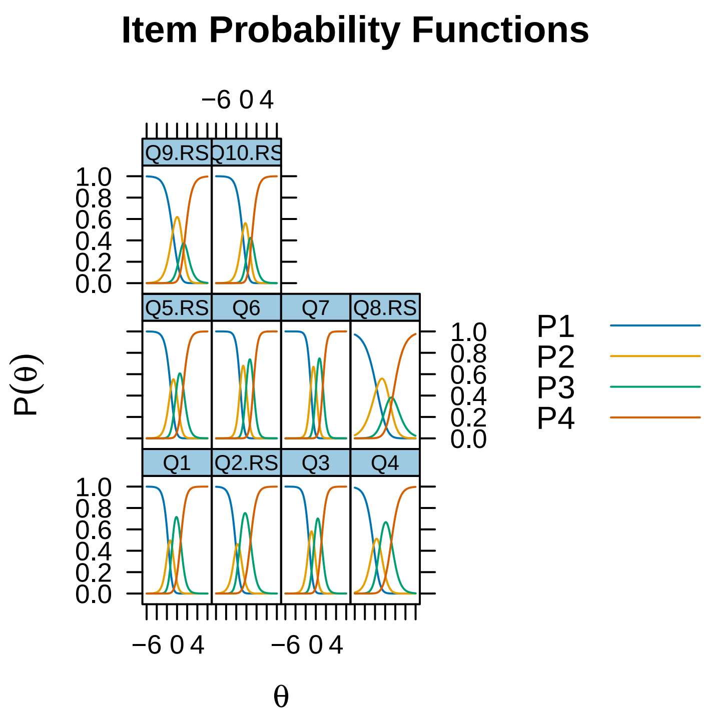
```r
itemfit(m.mcpg, empirical.plot=1, empirical.CI=.95)
```



En principio, parece que el modelo es capaz de explicar adecuadamente el comportamiento de las personas en estas categorías.

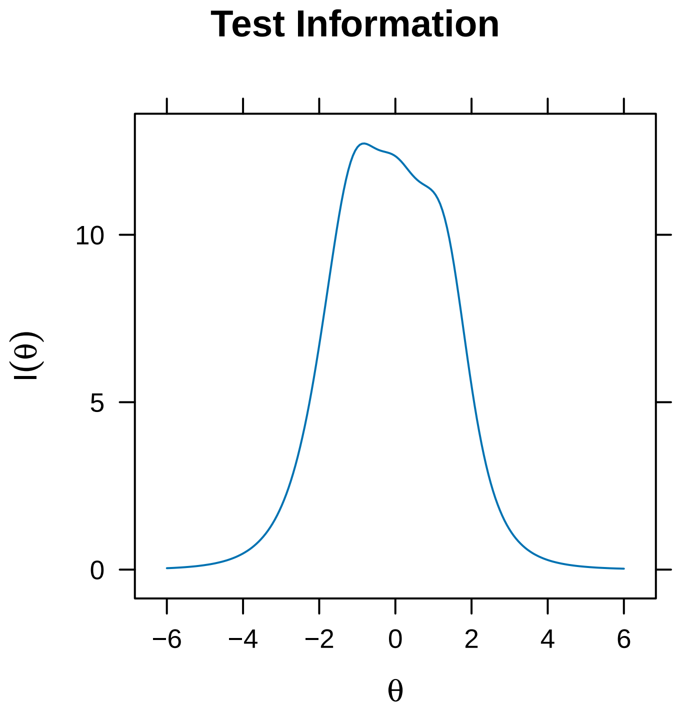
Las FRCs de todos lsdos ítems se pueden obtener con:

```r
plot(m.mcpg, type="trace")
```



y la FIT del test completo se puede obtener con:

```
plot(m.mcpg, type="info")
```



Capítulo 7

Modelo de Respuesta Graduada

7.1 Estimación de parámetros

Para estimar los parámetros en este modelo emplearemos:

```
m.mrg <- mirt(rse, model=1, itemtype="graded", SE=T, verbose=F)
```

y para comprobar si el algoritmo de estimación de parámetros ha encontrado la convergencia se emplea:

```
extract.mirt(m.mrg, what="converged")
```

```
[1] TRUE
```

que devuelve `TRUE`. El ajuste global del modelo se obtiene con:

```
m.mcpg
```

```

Call:
mirt(data = rse, model = 1, itemtype = "gpcm", SE = T, verbose = F)

Full-information item factor analysis with 1 factor(s).
Converged within 1e-04 tolerance after 163 EM iterations.
mirt version: 1.44.0
M-step optimizer: BFGS
EM acceleration: Ramsay
Number of rectangular quadrature: 61
Latent density type: Gaussian

Information matrix estimated with method: Oakes
Second-order test: model is a possible local maximum
Condition number of information matrix =  307.442

Log-likelihood = -5020.963
Estimated parameters: 40
AIC = 10121.93
BIC = 10290.51; SABIC = 10163.55
G2 (1048535) = 4077.64, p = 1
RMSEA = 0, CFI = NaN, TLI = NaN
```

y la tabla simplificada con las estimaciones de los parámetros en este modelo se obtiene con:

```
coef(m.mrg, simplify=T, IRTpars=T)
```

```
$items
            a     b1     b2    b3
Q1      2.476 -1.839 -0.869 0.703
Q2.RS   2.147 -2.335 -1.248 0.760
Q3      2.615 -1.388 -0.299 1.125
Q4      1.678 -2.463 -0.896 1.165
Q5.RS   2.250 -1.313 -0.113 1.222
Q6      2.828 -1.256 -0.027 1.388
Q7      3.090 -1.021  0.096 1.395
Q8.RS   1.271 -1.565  0.474 1.869
Q9.RS   1.941 -0.795  0.861 1.767
Q10.RS 2.339 -0.791  0.408 1.239

$means
F1
 0

$cov
   F1
F1  1
```

Para cada ítem aparece un parámetro de discriminación (`a`) y tres umbrales (`b1`, `b2`, `b3`) asociados a las 4 categorías de los ítems. En general, los ítems presentan parámetros de discriminación relativamente muy elevados. De nuevo, los ítems `Q6` y `Q7` son los más discriminativos. Además, todos los umbrales de las categorías están ordenados.

Para examinar los errores típicos de los parámetros empleamos:

```
coef(m.mrg, IRTpars=T, printSE=T)
```

```
$Q1
        a     b1     b2    b3
par 2.476 -1.839 -0.869 0.703
SE  0.204  0.128  0.080 0.077

$Q2.RS
        a     b1     b2    b3
par 2.147 -2.335 -1.248 0.760
SE  0.182  0.173  0.099 0.083

$Q3
        a     b1     b2    b3
par 2.615 -1.388 -0.299 1.125
SE  0.207  0.100  0.067 0.089

$Q4
        a     b1     b2    b3
par 1.678 -2.463 -0.896 1.165
SE  0.147  0.201  0.097 0.110
```

```
$Q5.RS
        a     b1     b2    b3
par 2.250 -1.313 -0.113 1.222
SE  0.177  0.101  0.069 0.098

$Q6
        a     b1     b2    b3
par 2.828 -1.256 -0.027 1.388
SE  0.230  0.091  0.064 0.097

$Q7
        a     b1    b2    b3
par 3.090 -1.021 0.096 1.395
SE  0.255  0.080 0.063 0.095

$Q8.RS
        a     b1    b2    b3
par 1.271 -1.565 0.474 1.869
SE  0.120  0.157 0.098 0.177

$Q9.RS
        a     b1    b2    b3
par 1.941 -0.795 0.861 1.767
SE  0.161  0.088 0.088 0.135

$Q10.RS
        a     b1    b2    b3
par 2.339 -0.791 0.408 1.239
SE  0.188  0.081 0.071 0.098

$GroupPars
    MEAN_1 COV_11
par      0      1
SE      NA     NA
```

y para examinar los intervalos de confianza escribimos:

```
coef(m.mrg, IRTpars=T)
```

```
$Q1
            a     b1     b2    b3
par     2.476 -1.839 -0.869 0.703
CI_2.5  2.076 -2.091 -1.025 0.552
CI_97.5 2.876 -1.588 -0.713 0.854

$Q2.RS
            a     b1     b2    b3
par     2.147 -2.335 -1.248 0.760
CI_2.5  1.791 -2.673 -1.442 0.596
CI_97.5 2.504 -1.996 -1.054 0.923

$Q3
            a     b1     b2    b3
```

```
par     2.615 -1.388 -0.299 1.125
CI_2.5  2.208 -1.583 -0.430 0.951
CI_97.5 3.021 -1.193 -0.169 1.298

$Q4
            a     b1     b2    b3
par     1.678 -2.463 -0.896 1.165
CI_2.5  1.391 -2.857 -1.085 0.949
CI_97.5 1.965 -2.068 -0.706 1.381

$Q5.RS
            a     b1     b2    b3
par     2.250 -1.313 -0.113 1.222
CI_2.5  1.903 -1.512 -0.248 1.029
CI_97.5 2.598 -1.115  0.021 1.414

$Q6
            a     b1     b2    b3
par     2.828 -1.256 -0.027 1.388
CI_2.5  2.377 -1.434 -0.152 1.198
CI_97.5 3.278 -1.078  0.099 1.578

$Q7
           a     b1     b2    b3
par     3.09 -1.021  0.096 1.395
CI_2.5  2.59 -1.178 -0.027 1.209
CI_97.5 3.59 -0.864  0.219 1.580

$Q8.RS
            a     b1    b2    b3
par     1.271 -1.565 0.474 1.869
CI_2.5  1.036 -1.872 0.283 1.522
CI_97.5 1.506 -1.257 0.666 2.216

$Q9.RS
            a     b1    b2    b3
par     1.941 -0.795 0.861 1.767
CI_2.5  1.625 -0.967 0.688 1.502
CI_97.5 2.257 -0.623 1.033 2.032

$Q10.RS
            a     b1    b2    b3
par     2.339 -0.791 0.408 1.239
CI_2.5  1.971 -0.949 0.269 1.048
CI_97.5 2.707 -0.633 0.547 1.431

$GroupPars
        MEAN_1 COV_11
par          0      1
CI_2.5      NA     NA
CI_97.5     NA     NA
```

7.2 Evaluación del ajuste

El estadístico M2 permite probar el ajuste del modelo a los ítems con:

```
M2(m.mrg, type="M2*")
```

```
            M2 df            p      RMSEA    RMSEA_5  RMSEA_95      SRMSR
stats 72.95865 15 1.320449e-09 0.08799606 0.06832637 0.1085886 0.06142995
           TLI       CFI
stats 0.877216 0.9263296
```

El estadístico M2 ha resultado significativo y el resto de estadísticos han obtenido valores por encima de los puntos de corte seleccionados. Por tanto, no se puede afirmar que este modelo produzca una buena explicación de las respuestas a los ítems en la escala de autoestima. Además, se puede evaluar el ajuste de los ítems con:

```
itemfit(m.mrg)
```

```
     item   S_X2 df.S_X2 RMSEA.S_X2 p.S_X2
1      Q1 37.822      33      0.017  0.258
2   Q2.RS 32.128      31      0.009  0.411
3      Q3 45.270      33      0.027  0.076
4      Q4 49.184      37      0.026  0.087
5   Q5.RS 34.720      36      0.000  0.529
6      Q6 31.522      30      0.010  0.390
7      Q7 44.325      28      0.034  0.026
8   Q8.RS 76.510      48      0.035  0.006
9   Q9.RS 41.199      39      0.011  0.375
10 Q10.RS 35.533      37      0.000  0.538
```

que evidencia que los ítems Q7 y Q8 no parecen seguir el modelo de respuesta graduada.

7.3 Parámetros de habilidad

El paquete mirt permite estimar los parámetros de habilidad con estimadores EAP, MAP y WLE, junto con sus errores típicos. En este caso, la estimación de los parámetros de habilidad se realizará con la opción MAP:

```
head(fscores(m.mrg, method="MAP", full.scores=T, full.scores.SE=T))
```

```
              F1     SE_F1
[1,] -0.50032216 0.2590253
[2,]  1.91431716 0.3579881
[3,]  0.02353596 0.2581993
[4,]  1.07454527 0.2556608
[5,] -0.10607403 0.2827575
[6,]  1.17416756 0.2807051
```

donde F1 es la estimación de la habilidad y SE_F1 es el error típico correspondiente. Los estadísticos de ajuste para detectar patrones aberrantes se pueden estimar con:

```
head(personfit(m.mrg, method="MAP"))
```

```
      outfit    z.outfit     infit   z.infit         Zh
1  0.4069704 -1.3442790 0.4576690 -1.192269  1.3028091
2 11.6293246  4.6547448 4.3062473  2.971755 -2.7957058
```

```
3   0.5415823 -0.9048396 0.4463820 -1.234794   1.0784352
4   0.6706328 -0.7619566 0.6745006 -0.748801   0.6805816
5   2.8035778  2.5286577 2.6713402  2.455925 -2.7627632
6   1.2897166  0.7757782 1.4591470  1.106846 -0.4799075
```

que produce los estadísticos de ajuste de medias de cuadrados (`infit` y `outfit`), sus correspondientes transformaciones t (`z.infit` y `z.outfit`) y el estadístico l_z (`zh`). Normalmente, el patrón de respuestas se ajusta al modelo si los estadísticos `infit` y `outfit` se encuentran en el intervalo $[0.5, 1.5]$ o el estadístico $l_z < 1.5$.

7.4 Cumplimiento de supuestos

7.4.1 Unidimensionalidad

Para probar la unidimensionalidad con este modelo debemos crear un nuevo objeto que contenga la solución bidimensional con:

```r
m.mrg2 <- mirt(rse, model=2, itemtype="graded", SE=T, verbose=F)
```

con vistas a la comparación con la solución unidimensional. Si escribimos:

```r
m.mrg2
```

```

Call:
mirt(data = rse, model = 2, itemtype = "graded", SE = T, verbose = F)

Full-information item factor analysis with 2 factor(s).
Converged within 1e-04 tolerance after 231 EM iterations.
mirt version: 1.44.0
M-step optimizer: BFGS
EM acceleration: Ramsay
Number of rectangular quadrature: 31
Latent density type: Gaussian

Information matrix estimated with method: Oakes
Second-order test: model is a possible local maximum
Condition number of information matrix =  286.6349

Log-likelihood = -4892.031
Estimated parameters: 49
AIC = 9882.062
BIC = 10088.58; SABIC = 9933.049
G2 (1048526) = 3819.77, p = 1
RMSEA = 0, CFI = NaN, TLI = NaN
```

se obtienen los estadísticos de ajuste en la solución bidimensional. Así, todos los estadísticos (`AIC`, `BIC` y `SABIC`) han reducido sus valores en la solución bidimensional. Esto unido a que en la solución unidimensional el modelo no fue capaz de ajustar algunos ítems, quizás daría lugar a considerar la posibilidad de que esta escala de autoestima no fuera unidimensional con este modelo.

7.4.2 Independencia local de los ítems

En este modelo emplearemos también el estadístico $Q3$ para evaluar la independencia local:

```r
residuals(m.mrg, type="Q3")
```

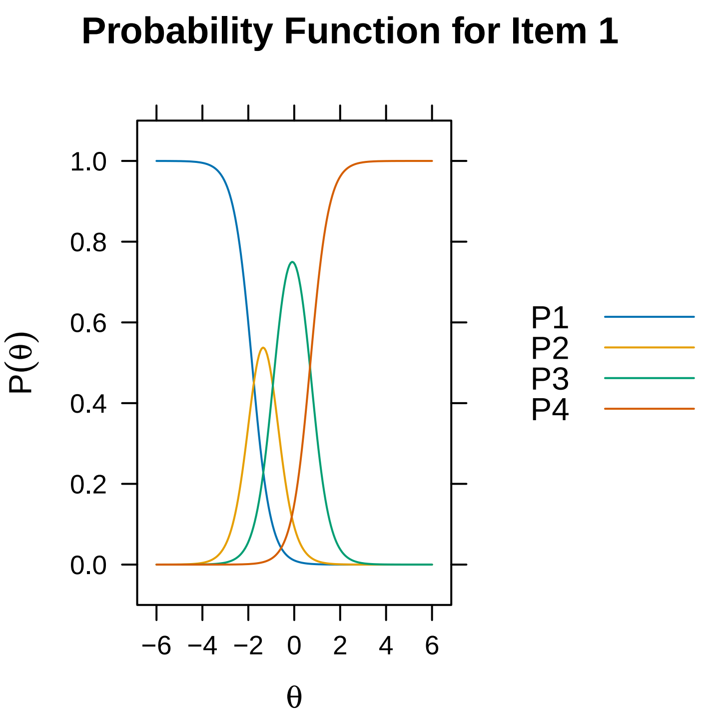
```
Q3 summary statistics:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -0.295  -0.178  -0.128  -0.098  -0.071   0.407

           Q1  Q2.RS     Q3     Q4  Q5.RS     Q6     Q7  Q8.RS  Q9.RS Q10.RS
Q1      1.000  0.226 -0.220 -0.031 -0.206 -0.153 -0.107 -0.141 -0.171 -0.138
Q2.RS   0.226  1.000 -0.223  0.089 -0.044 -0.142 -0.178 -0.182 -0.152 -0.175
Q3     -0.220 -0.223  1.000 -0.101  0.080 -0.128 -0.133 -0.071 -0.094 -0.108
Q4     -0.031  0.089 -0.101  1.000 -0.078 -0.269 -0.107 -0.048 -0.072 -0.090
Q5.RS  -0.206 -0.044  0.080 -0.078  1.000 -0.185 -0.181 -0.088 -0.073 -0.148
Q6     -0.153 -0.142 -0.128 -0.269 -0.185  1.000  0.128  0.068 -0.249 -0.243
Q7     -0.107 -0.178 -0.133 -0.107 -0.181  0.128  1.000 -0.130 -0.295 -0.243
Q8.RS  -0.141 -0.182 -0.071 -0.048 -0.088  0.068 -0.130  1.000  0.009 -0.020
Q9.RS  -0.171 -0.152 -0.094 -0.072 -0.073 -0.249 -0.295  0.009  1.000  0.407
Q10.RS -0.138 -0.175 -0.108 -0.090 -0.148 -0.243 -0.243 -0.020  0.407  1.000
```

En la tabla se aprecia, de nuevo, que varios estadísticos `Q3` han obtenido valores por encima de `|0.20|`. Por tanto, no se puede asegurar que el principio de indepedencia local se cumpla en este modelo.

7.5 Gráficos en el modelo de respuesta graduada

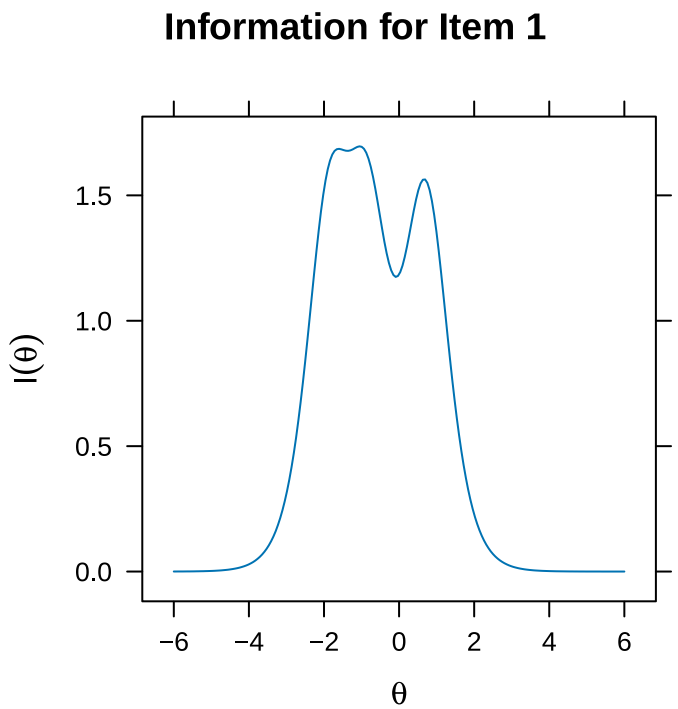
Para graficar las FRCs en este modelo emplearemos:

```r
itemplot(m.mrg, item=1)
```



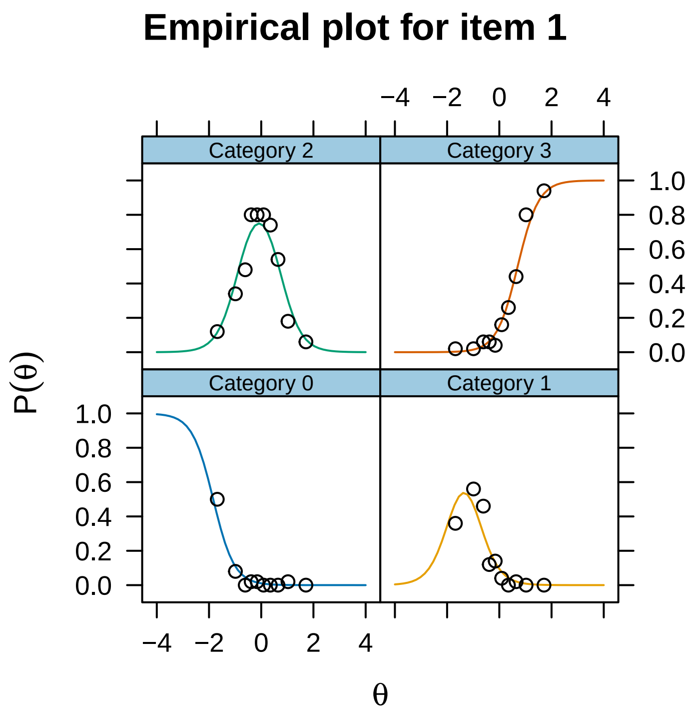
y para obtener la FII de ese mismo ítem emplearemos:

```r
itemplot(m.mrg, 1, type="info")
```



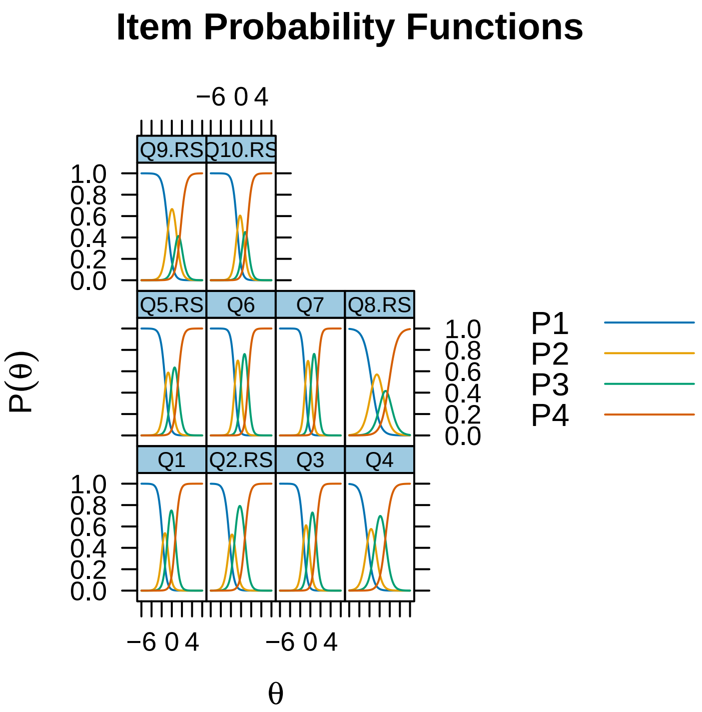
Finalmente, el ajuste de las FRCs empíricas a las FRCs teóricas se puede obtener con:

```r
itemfit(m.mrg, empirical.plot=1, empirical.CI=.95)
```



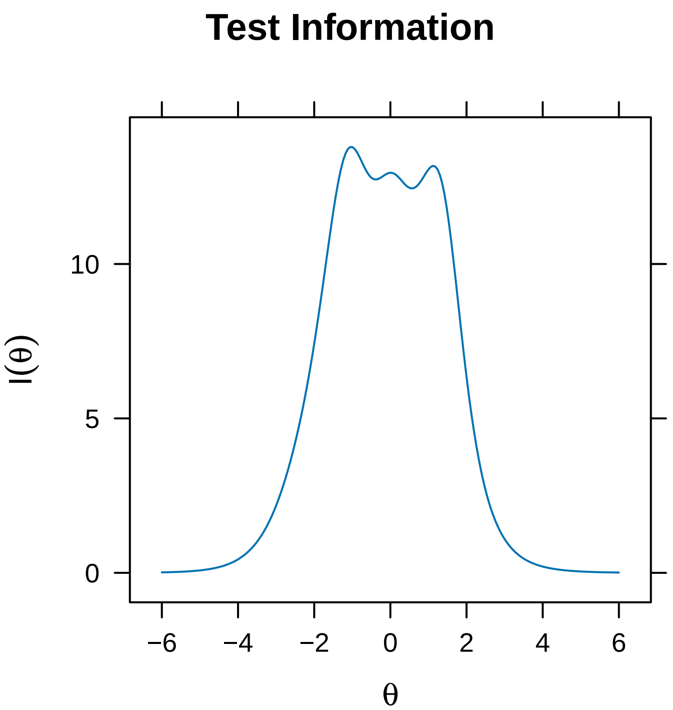
Las FRCs de todos los ítems se pueden obtener con:

```r
plot(m.mrg, type="trace")
```



y la FIT del test completo se puede obtener con:

```r
plot(m.mrg, type="info")
```



Capítulo 8

Comparación de modelos

Supuesto que algunos modelos están anidados, es posible comparar sus soluciones a través de la función siguiente:

```
anova(m.mev, m.mcp)
```

```
            AIC    SABIC       HQ      BIC    logLik      X2 df p
m.mev 10384.67 10398.19 10406.17 10439.46 -5179.333
m.mcp 10227.39 10259.65 10278.66 10358.05 -5082.696 193.275 18 0
```

En este caso, se compara el modelo de escalas de valoración vs. el modelo de crédito parcial que permite que los parámetros de umbral varíen de ítem a ítem. Parece que la restricción de que todos los parámetros de umbral sean iguales para todos los ítems no produce mejores estimaciones de los parámetros, dado que los estadísticos `AIC`, `SABIC` y `BIC` han resultado más bajos en el modelo de crédito parcial.

Una hipótesis más a investigar sería si la restricción de parámetros de discriminación iguales a `1` se puede mantener frente a la posibilidad de que varíen libremente tal como propone el modelo de crédito parcial generalizado. Para probar esta hipotésis, se emplea la función:

```
anova(m.mcp, m.mcpg)
```

```
             AIC    SABIC       HQ      BIC    logLik      X2 df p
m.mcp  10227.39 10259.65 10278.66 10358.05 -5082.696
m.mcpg 10121.93 10163.55 10188.08 10290.51 -5020.963 123.465  9 0
```

que ofrece como resultado que el modelo de crédito parcial generalizado ofrece un mejor resultado que el modelo de crédito parcial.

Generalmente, en la aplicación de estos modelos se detecta que el modelo que ofrece más posibilidades de variabilidad en los parámetros produce un mejor ajuste que los modelos más restrictivos, aunque emplear un modelo con más parámetros siempre produce una mayor complicación de las características individuales que afectan a las personas evaluadas.