Modelos dicotómicos con el paquete `mirt` de R

José Ant. López Pina

Índice

Capítulo 1

Introducción

En este documento se implementan los modelos de respuesta al ítem para ítems dicotómicos con el paquete `mirt`. Este paquete permite estimar parámetros con el método de máxima verosimilitud marginal tanto para modelos unidimensionales como multidimensionales.

Para instalar el paquete utilice `install.packages("mirt")` en `R` o `RStudio`. Si se instala el paquete por primera vez, `R` solicitará que se seleccione un `CRAN mirror`. Por defecto, elegir el `mirror` más cercano en España (Madrid o A Coruña). Una vez instalado el paquete `mirt`, no es preciso volver a instalarlo cada vez que comience `R` o `RStudio`. Es decir, el paquete instalado permanecerá activo siempre que no se elimine o no se actualice `R` a una nueva versión. Para trabajar con el paquete utilice la función: `library(mirt)`.

Las funciones pueden tener varios argumentos definidos por defecto. Si se desea conocer la estructura de la función completa con todos los argumentos, entonces debemos escribir en `R` `help(library)` o `?library`, y se abrirá una página web que indica la estructura de esta función.

Antes de comenzar con los modelos `TRI` conviene realizar un estudio descriptivo previo de los ítems, así como un análisis del funcionamiento de los distractores de los ítems[1]. Para ello, emplearemos los paquetes `ltm`, `psych`, `CTT` y `ShinnyItemAnalysis`.

Para realizar cualquier análisis conviene que instale todos los paquetes implicados en este documento con `install.packages("ltm")`, `install.package("psych")`, `install.packages(CTT)` y `install.packages(ShinnyItemAnalysis)`.

Una vez que haya instalado todos los paquetes debe cargar la librería de cada paquete cuando se le indique en el documento. En primer lugar, debemos cargar el paquete `mirt` con:

```
library(mirt)
```

```
Cargando paquete requerido: stats4
```

```
Cargando paquete requerido: lattice
```

[1]El análisis de distractores sólo se puede hacer si disponemos de las respuestas originales de los estudiantes.

Capítulo 2

Gestión de los datos

2.1 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 que contienen las respuestas a los ítems. Una vez que se haya creado la carpeta en el disco principal, `R` identificará la ruta donde se encuentra el archivo correspondiente con la función `setwd("C:/datos")`. Esta carpeta permanecerá activa en tanto no se cambie de denominación o se borre del disco principal. `R` admite estructuras más complejas como `setwd("C:/datos/ejemplos")`.

Una vez que se ha especificado la ruta, el paso siguiente consiste en asignar la matriz de datos `rv.dat` a un objeto. El objeto puede tener cualquier nombre, por ejemplo, `rv0`. Para ello, se utiliza la estructura siguiente: `rv0 <- read.table("rv.dat", header=T, sep=",")`.

La función `read.table` permite leer el archivo `rv.dat` que contiene los nombres de las variables en la primera fila. A partir de la segunda fila, aparecen las respuestas a los ítems separadas (`sep`) por comas. La lectura del archivo que contiene la matriz de datos que se desea analizar se asigna al objeto `rv0`. En este caso, el archivo contiene algunos datos ausentes indicados con `NA`, por lo que se pueden adoptar dos acciones: 1) Eliminar los casos que contengan datos ausentes o 2) imputar las respuestas con algún método de imputación. En este caso, hemos optado por eliminar los casos con datos ausentes, creando un nuevo objeto (`rv1`) que contenga sólo los casos que hayan contestado todos los ítems con `na.omit(rv0)`.

No obstante, para estimar los parámetros en los diferentes modelos de respuesta al ítem, se deben eliminar de la matriz de datos las columnas correspondientes al número de identificación y al género, asignando el resultado a un nuevo objeto: `rv2 <- rv1[, -c(1,20)]`.

2.2 Corrección de la matriz de respuestas

El archivo `rv.dat` contiene la matriz de respuestas originales en el grupo, pero es necesario corregirlas en función de la respuesta correcta en cada ítem, de manera que el objeto a analizar contenga las respuestas a los ítems en formato dicotómico (1/0). Para ello, emplearemos la función `key2binary()`.

El objeto `rv2` contiene las respuestas originales de los estudiantes con las opciones de `1` a `4`. Para obtener las respuestas correctas se crea un objeto (e.g., `clave`) que contenga las opciones correctas de cada ítem. Este objeto es un vector columna y debe tener el mismo número de elementos que ítems hay en el objeto `rv2`. Finalmente se corrige la matriz de respuestas con la función `key2binary(rv2, clave)`, que contiene el objeto con la matriz de respuestas `rv2` y el objeto con el vector columna `clave`. La matriz de respuestas corregida se almacena ahora en el objeto `rv`.

La secuencia de los pasos indicados anteriormente se detalla a continuación:

```
library(mirt)
setwd("c:/datos/")
rv0 <- read.table("rv.dat", header=T, sep=",")
rv1 <- na.omit(rv0)
rv2 <- rv1[, -c(1,20)]
clave <- c(3,4,1,4,2,1,4,1,2,3,3,4,3,1,4,2,1,3)
rv <- key2binary(rv2,clave)
```

Con la función:

```
head(rv2)
```

```
  RV1 RV2 RV3 RV4 RV5 RV6 RV7 RV8 RV9 RV10 RV11 RV12 RV13 RV14 RV15 RV16 RV17
1   3   4   1   4   2   1   3   1   2    3    3    4    3    1    4    2    4
2   3   4   1   4   2   1   4   3   2    3    3    3    3    1    2    2    3
3   3   4   1   4   2   1   4   1   2    3    3    4    3    3    4    2    1
4   3   4   1   4   2   1   4   3   2    3    3    4    3    1    4    4    3
5   3   4   1   4   2   1   3   1   2    2    3    4    1    1    4    1    1
6   3   4   1   4   2   1   4   1   2    3    3    4    3    1    4    2    1
  RV18
1    1
2    4
3    1
4    1
5    3
6    4
```

podemos comprobar que el objeto `rv2` contiene la matriz original de respuesta a los ítems eliminados los casos con omisiones `NA`. Y si escribimos a continuación:

```
head(rv)
```

```
     RV1 RV2 RV3 RV4 RV5 RV6 RV7 RV8 RV9 RV10 RV11 RV12 RV13 RV14 RV15 RV16
[1,]   1   1   1   1   1   1   0   1   1    1    1    1    1    1    1    1
[2,]   1   1   1   1   1   1   1   0   1    1    1    0    1    1    0    1
[3,]   1   1   1   1   1   1   1   1   1    1    1    1    1    0    1    1
[4,]   1   1   1   1   1   1   1   0   1    1    1    1    1    1    1    0
[5,]   1   1   1   1   1   1   0   1   1    0    1    1    0    1    1    0
[6,]   1   1   1   1   1   1   1   1   1    1    1    1    1    1    1    1
     RV17 RV18
[1,]    0    0
[2,]    0    0
[3,]    1    0
[4,]    0    0
[5,]    1    1
[6,]    1    0
```

tenemos la matriz de respuestas corregida. La función:

```r
names(rv1)
```

```
 [1] "NID"    "RV1"    "RV2"    "RV3"    "RV4"    "RV5"    "RV6"    "RV7"   
 [9] "RV8"    "RV9"    "RV10"   "RV11"   "RV12"   "RV13"   "RV14"   "RV15"  
[17] "RV16"   "RV17"   "RV18"   "Genero"
```

devuelve el nombre de las variables y la dimensionalidad final se obtiene con:

```r
dim(rv)
```

```
[1] 402  18
```

que determina `402` casos válidos para los `18` ítems de la escala.

Capítulo 3

Análisis de ítems con la TCT

3.1 Estadísticos descriptivos de los ítems

Se puede obtener un resumen de los estadísticos básicos de los ítems con:

```
library(ltm)
```

```
Cargando paquete requerido: MASS

Cargando paquete requerido: msm

Cargando paquete requerido: polycor


Adjuntando el paquete: 'ltm'

The following object is masked from 'package:mirt':

    Science
```

```
descript(rv)
```

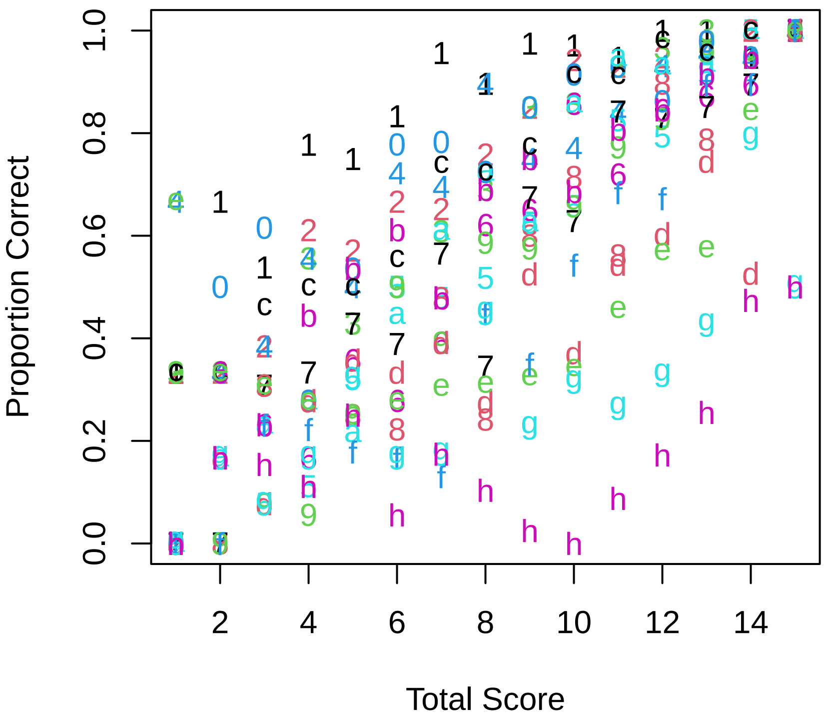
```

Descriptive statistics for the 'rv' data-set

Sample:
 18 items and 402 sample units; 0 missing values

Proportions for each level of response:
                     logit
RV1  0.0871 0.9129  2.3500
RV2  0.1741 0.8259  1.5566
RV3  0.2114 0.7886  1.3163
RV4  0.2114 0.7886  1.3163
RV5  0.3607 0.6393  0.5723
RV6  0.3507 0.6493  0.6158
RV7  0.3507 0.6493  0.6158
RV8  0.4154 0.5846  0.3416
RV9  0.3706 0.6294  0.5294
RV10 0.1940 0.8060  1.4240
RV11 0.2761 0.7239  0.9638
```

```
RV12 0.2886 0.7114  0.9024
RV13 0.1965 0.8035  1.4082
RV14 0.5224 0.4776 -0.0896
RV15 0.5647 0.4353 -0.2602
RV16 0.4826 0.5174  0.0697
RV17 0.6841 0.3159 -0.7726
RV18 0.8483 0.1517 -1.7210


Frequencies of total scores:
     0 1 2 3 4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
Freq 0 0 1 3 6 13 18 28 18 23 29 40 35 57 58 47 19  6  1


Point Biserial correlation with Total Score:
     Included Excluded
RV1    0.3706   0.2940
RV2    0.4384   0.3392
RV3    0.5066   0.4064
RV4    0.3762   0.2635
RV5    0.5363   0.4209
RV6    0.4806   0.3585
RV7    0.4460   0.3199
RV8    0.4679   0.3399
RV9    0.5513   0.4374
RV10   0.4152   0.3095
RV11   0.5829   0.4829
RV12   0.4147   0.2923
RV13   0.4687   0.3674
RV14   0.3185   0.1745
RV15   0.2679   0.1221
RV16   0.5081   0.3833
RV17   0.2127   0.0741
RV18   0.1295   0.0215


Cronbach's alpha:
                 value
All Items       0.7195
Excluding RV1   0.7100
Excluding RV2   0.7047
Excluding RV3   0.6981
Excluding RV4   0.7108
Excluding RV5   0.6948
Excluding RV6   0.7015
Excluding RV7   0.7055
Excluding RV8   0.7034
Excluding RV9   0.6930
Excluding RV10 0.7069
Excluding RV11 0.6894
Excluding RV12 0.7083
```

```
Excluding RV13 0.7019
Excluding RV14 0.7210
Excluding RV15 0.7263
Excluding RV16 0.6986
Excluding RV17 0.7295
Excluding RV18 0.7289


Pairwise Associations:
   Item i Item j p.value
1       6     15   1.000
2       8     18   1.000
3       2     18   1.000
4       1     17   1.000
5      13     18   1.000
6      17     18   1.000
7      14     15   0.987
8       7     18   0.976
9       8     15   0.968
10      3     17   0.926
```

que presenta la proporción de casos para cada opción de respuesta, las frecuencias correspondientes a cada puntuación empírica, el índice de discriminación de la teoría clásica de tests para cada ítem, el coeficiente alfa para el test completo y en función de si elimina cada uno de los ítems. El paquete `ltm` también permite obtener un gráfico de las puntuaciones empíricas vs. las proporciones de respuestas correctas con:

```
des <- descript(rv)
plot.descript(des)
```



3.2 Análisis de ítems

Un examen más pormenorizado del funcionamiento de cada ítem se puede obtener con el paquete CTT. Para ello, debemos cargar la librería correspondiente:

```
detach(package:ltm)
library(CTT)
```

```
Adjuntando el paquete: 'CTT'

The following object is masked from 'package:polycor':

    polyserial
```

```
distractorAnalysis(rv2,clave)
```

```
$RV1
  correct key   n        rspP       pBis      discrim      lower      mid50
1           1   3 0.007462687 -0.0627202 -0.004483188 0.01818182 0.00000000
2           2  13 0.032338308 -0.2633093 -0.081818182 0.08181818 0.01923077
3       *   3 367 0.912935323  0.2939629  0.222665006 0.76363636 0.95192308
4           4  19 0.047263682 -0.3560933 -0.136363636 0.13636364 0.02884615
        mid75      upper
1 0.000000000 0.01369863
2 0.017391304 0.00000000
3 0.973913043 0.98630137
```

```
4 0.008695652 0.00000000

$RV2
  correct key   n       rspP       pBis    discrim     lower      mid50
1           1  16 0.03980100 -0.2645027 -0.1181818 0.1181818 0.00000000
2           2  32 0.07960199 -0.3447234 -0.1727273 0.1727273 0.10576923
3           3  22 0.05472637 -0.2909262 -0.1226650 0.1363636 0.03846154
4       *   4 332 0.82587065  0.3392013  0.4135741 0.5727273 0.85576923
       mid75      upper
1 0.02608696 0.00000000
2 0.01739130 0.00000000
3 0.01739130 0.01369863
4 0.93913043 0.98630137

$RV3
  correct key   n       rspP       pBis     discrim     lower      mid50
1       *   1 317 0.78855721  0.4063998  0.51357410 0.4727273 0.80769231
2           2  20 0.04975124 -0.2052922 -0.08630137 0.1000000 0.05769231
3           3  26 0.06467662 -0.3397494 -0.17272727 0.1727273 0.04807692
4           4  39 0.09701493 -0.4406256 -0.25454545 0.2545455 0.08653846
      mid75      upper
1 0.9478261 0.98630137
2 0.0173913 0.01369863
3 0.0173913 0.00000000
4 0.0173913 0.00000000

$RV4
  correct key   n       rspP        pBis      discrim      lower      mid50
1           1  18 0.04477612 -0.21355504 -0.086301370 0.10000000 0.01923077
2           2  12 0.02985075 -0.07281419 -0.004483188 0.01818182 0.07692308
3           3  55 0.13681592 -0.43008434 -0.304483188 0.31818182 0.10576923
4       *   4 317 0.78855721  0.26354201  0.395267746 0.56363636 0.79807692
        mid75      upper
1 0.034782609 0.01369863
2 0.008695652 0.01369863
3 0.069565217 0.01369863
4 0.886956522 0.95890411

$RV5
  correct key   n       rspP       pBis    discrim     lower      mid50
1           1  75 0.18656716 -0.3205514 -0.2088418 0.2636364 0.25961538
2       *   2 257 0.63930348  0.4208861  0.6542964 0.2909091 0.60576923
3           3  35 0.08706468 -0.2847715 -0.1727273 0.1727273 0.08653846
4           4  35 0.08706468 -0.4799104 -0.2727273 0.2727273 0.04807692
       mid75      upper
1 0.13043478 0.05479452
2 0.80869565 0.94520548
3 0.06086957 0.00000000
4 0.00000000 0.00000000

$RV6
```

```
  correct key   n       rspP       pBis    discrim     lower      mid50
1       *   1 261 0.64925373  0.3584662  0.5995019 0.2909091 0.71153846
2           2  60 0.14925373 -0.3103976 -0.2224159 0.2909091 0.08653846
3           3  50 0.12437811 -0.3062565 -0.1770859 0.2181818 0.13461538
4           4  31 0.07711443 -0.3809728 -0.2000000 0.2000000 0.06730769
       mid75      upper
1 0.78260870 0.89041096
2 0.12173913 0.06849315
3 0.07826087 0.04109589
4 0.01739130 0.00000000

$RV7
  correct key   n       rspP       pBis     discrim      lower      mid50
1           1  55 0.13681592 -0.4059335 -0.22702366 0.28181818 0.14423077
2           2  12 0.02985075 -0.1989784 -0.06363636 0.06363636 0.01923077
3           3  74 0.18407960 -0.3094560 -0.20423412 0.27272727 0.26923077
4       *   4 261 0.64925373  0.3198739  0.49489415 0.38181818 0.56730769
       mid75      upper
1 0.04347826 0.05479452
2 0.02608696 0.00000000
3 0.09565217 0.06849315
4 0.83478261 0.87671233

$RV8
  correct key   n       rspP       pBis     discrim     lower      mid50
1       *   1 235 0.58457711  0.3398668  0.57210461 0.2909091 0.53846154
2           2  27 0.06716418 -0.2048755 -0.07708593 0.1181818 0.07692308
3           3 120 0.29850746 -0.3967173 -0.34956413 0.4454545 0.34615385
4           4  20 0.04975124 -0.3815480 -0.14545455 0.1454545 0.03846154
       mid75      upper
1 0.73043478 0.86301370
2 0.02608696 0.04109589
3 0.24347826 0.09589041
4 0.00000000 0.00000000

$RV9
  correct key   n       rspP       pBis    discrim     lower      mid50
1           1  77 0.19154229 -0.4884705 -0.3772105 0.3909091 0.21153846
2       *   2 253 0.62935323  0.4373676  0.7043587 0.2545455 0.60576923
3           3  37 0.09203980 -0.2515759 -0.1590286 0.1727273 0.10576923
4           4  35 0.08706468 -0.2974740 -0.1681196 0.1818182 0.07692308
       mid75      upper
1 0.09565217 0.01369863
2 0.80000000 0.95890411
3 0.05217391 0.01369863
4 0.05217391 0.01369863

$RV10
  correct key   n       rspP       pBis     discrim     lower      mid50
1           1  21 0.05223881 -0.2311673 -0.10909091 0.1090909 0.02884615
2           2  26 0.06467662 -0.2135805 -0.08169365 0.1090909 0.07692308
```

```
3        *   3 324 0.80597015  0.3094565  0.39987547 0.5727273 0.83653846
4            4  31 0.07711443 -0.4127259 -0.20909091 0.2090909 0.05769231
       mid75      upper
1 0.05217391 0.00000000
2 0.03478261 0.02739726
3 0.89565217 0.97260274
4 0.01739130 0.00000000

$RV11
  correct key   n       rspP       pBis     discrim      lower      mid50
1           1  38 0.09452736 -0.3613181 -0.20448319 0.21818182 0.08653846
2           2   6 0.01492537 -0.2293123 -0.05454545 0.05454545 0.00000000
3       *   3 291 0.72388060  0.4829083  0.62254047 0.33636364 0.73076923
4           4  67 0.16666667 -0.5000345 -0.36351183 0.39090909 0.18269231
       mid75      upper
1 0.03478261 0.01369863
2 0.00000000 0.00000000
3 0.93913043 0.95890411
4 0.02608696 0.02739726

$RV12
  correct key   n       rspP       pBis    discrim     lower      mid50
1           1  17 0.04228856 -0.2870424 -0.1272727 0.1272727 0.01923077
2           2  27 0.06716418 -0.2513142 -0.1089664 0.1363636 0.07692308
3           3  72 0.17910448 -0.3544841 -0.2498132 0.2909091 0.19230769
4       *   4 286 0.71144279  0.2923089  0.4860523 0.4454545 0.71153846
        mid75      upper
1 0.008695652 0.00000000
2 0.017391304 0.02739726
3 0.147826087 0.04109589
4 0.826086957 0.93150685

$RV13
  correct key   n       rspP       pBis     discrim      lower      mid50
1           1  29 0.07213930 -0.2447060 -0.11357410 0.12727273 0.09615385
2           2  14 0.03482587 -0.2315282 -0.06811955 0.08181818 0.01923077
3       *   3 323 0.80348259  0.3674204  0.43623910 0.53636364 0.80769231
4           4  36 0.08955224 -0.4525409 -0.25454545 0.25454545 0.07692308
       mid75      upper
1 0.03478261 0.01369863
2 0.01739130 0.01369863
3 0.94782609 0.97260274
4 0.00000000 0.00000000

$RV14
  correct key   n       rspP       pBis    discrim     lower      mid50
1       *   1 192 0.47761194  0.1744660  0.4214197 0.2909091 0.40384615
2           2 125 0.31094527 -0.1948545 -0.1127024 0.3181818 0.34615385
3           3  38 0.09452736 -0.2491577 -0.1133250 0.1818182 0.05769231
4           4  47 0.11691542 -0.3463419 -0.1953923 0.2090909 0.19230769
       mid75      upper
```

```
1 0.57391304 0.71232877
2 0.33913043 0.20547945
3 0.06086957 0.06849315
4 0.02608696 0.01369863

$RV15
  correct key   n       rspP       pBis     discrim     lower       mid50
1           1  86 0.21393035 -0.2826394 -0.17671233 0.3000000 0.298076923
2           2 126 0.31343284 -0.1770643 -0.09912827 0.2909091 0.365384615
3           3  15 0.03731343 -0.2945354 -0.11818182 0.1181818 0.009615385
4       *   4 175 0.43532338  0.1220627  0.39402242 0.2909091 0.326923077
        mid75     upper
1 0.113043478 0.1232877
2 0.365217391 0.1917808
3 0.008695652 0.0000000
4 0.513043478 0.6849315

$RV16
  correct key   n       rspP       pBis    discrim     lower     mid50
1           1  30 0.07462687 -0.2718952 -0.1226650 0.1363636 0.1057692
2       *   2 208 0.51741294  0.3833284  0.7404732 0.1636364 0.4423077
3           3  66 0.16417910 -0.2941050 -0.2134496 0.2545455 0.2115385
4           4  98 0.24378109 -0.4158916 -0.4043587 0.4454545 0.2403846
       mid75      upper
1 0.02608696 0.01369863
2 0.67826087 0.90410959
3 0.11304348 0.04109589
4 0.18260870 0.04109589

$RV17
  correct key   n       rspP        pBis     discrim     lower      mid50
1       *   1 127 0.31592040  0.07406213  0.34333748 0.1909091 0.31730769
2           2  38 0.09452736 -0.32012290 -0.18169365 0.2090909 0.06730769
3           3 208 0.51741294 -0.09830879 -0.02988792 0.4545455 0.52884615
4           4  29 0.07213930 -0.28376169 -0.13175592 0.1454545 0.08653846
       mid75      upper
1 0.29565217 0.53424658
2 0.05217391 0.02739726
3 0.62608696 0.42465753
4 0.02608696 0.01369863

$RV18
  correct key   n       rspP        pBis     discrim     lower      mid50
1           1 162 0.40298507 -0.09178694  0.01955168 0.2818182 0.54807692
2           2  34 0.08457711 -0.32898455 -0.15890411 0.2000000 0.03846154
3       *   3  61 0.15174129  0.02153160  0.18792030 0.1545455 0.03846154
4           4 145 0.36069652 -0.15006710 -0.04856787 0.3636364 0.37500000
       mid75      upper
1 0.45217391 0.30136986
2 0.04347826 0.04109589
3 0.13043478 0.34246575
```

```
4 0.37391304 0.31506849
```

Para cada ítem se presenta la frecuencia de respuestas en cada categoría (`n`) y se marca la categoría correcta con `*`; la proporción de respuestas en cada categoría (`rspP`); el índice de discriminación obtenido con la correlación biserial-puntual (`pBis`), el índice de discriminación (`D`) obtenido entre la diferencia de las proporciones del grupo superior (27%) menos la del grupo inferior (27%) y el porcentaje de respuestas obtenidas en diferentes grupos en función del cuartil.

Generalmente, un índice de discriminación válido actualmente se corresponde con la correlación biserial-puntual corregida. Se acepta que un ítem funciona bien cuando su índice de discriminación se encuentra en el intervalo (`0.30 - 0.70`), y su índice de dificultad se encuentre en el intervalo (`0.20 - 0.80`).

Para obtener los índices de dificultad y discriminación de los ítems conjuntamente empleamos el paquete **`psych`** con:

```
detach(package:CTT)
library(psych)
```

```

Adjuntando el paquete: 'psych'

The following object is masked from 'package:polycor':

    polyserial
```

y obtenemos estos estadísticos con:

```
alpha(rv)
```

```

Reliability analysis
Call: alpha(x = rv)

  raw_alpha std.alpha G6(smc) average_r S/N  ase mean   sd median_r
      0.72      0.72    0.73      0.13 2.6 0.02 0.63 0.18     0.13

    95% confidence boundaries
         lower alpha upper
Feldt     0.68  0.72  0.76
Duhachek  0.68  0.72  0.76

 Reliability if an item is dropped:
     raw_alpha std.alpha G6(smc) average_r S/N alpha se  var.r med.r
RV1       0.71      0.71    0.72      0.13 2.4    0.021 0.0081  0.13
RV2       0.70      0.71    0.71      0.12 2.4    0.021 0.0082  0.12
RV3       0.70      0.70    0.71      0.12 2.3    0.022 0.0075  0.12
RV4       0.71      0.71    0.72      0.13 2.5    0.021 0.0083  0.13
RV5       0.69      0.70    0.71      0.12 2.3    0.022 0.0077  0.12
RV6       0.70      0.70    0.71      0.12 2.4    0.021 0.0079  0.12
RV7       0.71      0.71    0.71      0.12 2.4    0.021 0.0079  0.13
RV8       0.70      0.71    0.71      0.12 2.4    0.021 0.0080  0.12
RV9       0.69      0.70    0.70      0.12 2.3    0.022 0.0070  0.12
RV10      0.71      0.71    0.72      0.12 2.4    0.021 0.0078  0.13
```

```
RV11      0.69      0.69    0.70      0.12 2.2    0.022 0.0066  0.12
RV12      0.71      0.71    0.72      0.13 2.4    0.021 0.0083  0.13
RV13      0.70      0.70    0.71      0.12 2.4    0.021 0.0077  0.12
RV14      0.72      0.72    0.73      0.13 2.6    0.020 0.0079  0.14
RV15      0.73      0.72    0.73      0.13 2.6    0.019 0.0077  0.14
RV16      0.70      0.70    0.71      0.12 2.3    0.022 0.0075  0.12
RV17      0.73      0.73    0.74      0.14 2.7    0.019 0.0073  0.14
RV18      0.73      0.73    0.74      0.14 2.8    0.019 0.0065  0.14

 Item statistics
       n raw.r std.r r.cor r.drop mean   sd
RV1  402  0.37  0.41 0.351  0.294 0.91 0.28
RV2  402  0.44  0.45 0.393  0.339 0.83 0.38
RV3  402  0.51  0.51 0.476  0.406 0.79 0.41
RV4  402  0.38  0.39 0.312  0.264 0.79 0.41
RV5  402  0.54  0.53 0.494  0.421 0.64 0.48
RV6  402  0.48  0.47 0.426  0.358 0.65 0.48
RV7  402  0.45  0.43 0.376  0.320 0.65 0.48
RV8  402  0.47  0.45 0.397  0.340 0.58 0.49
RV9  402  0.55  0.55 0.527  0.437 0.63 0.48
RV10 402  0.42  0.43 0.369  0.309 0.81 0.40
RV11 402  0.58  0.58 0.573  0.483 0.72 0.45
RV12 402  0.41  0.41 0.348  0.292 0.71 0.45
RV13 402  0.47  0.48 0.438  0.367 0.80 0.40
RV14 402  0.32  0.30 0.202  0.174 0.48 0.50
RV15 402  0.27  0.26 0.158  0.122 0.44 0.50
RV16 402  0.51  0.49 0.455  0.383 0.52 0.50
RV17 402  0.21  0.20 0.087  0.074 0.32 0.47
RV18 402  0.13  0.15 0.030  0.022 0.15 0.36

Non missing response frequency for each item
        0    1 miss
RV1  0.09 0.91    0
RV2  0.17 0.83    0
RV3  0.21 0.79    0
RV4  0.21 0.79    0
RV5  0.36 0.64    0
RV6  0.35 0.65    0
RV7  0.35 0.65    0
RV8  0.42 0.58    0
RV9  0.37 0.63    0
RV10 0.19 0.81    0
RV11 0.28 0.72    0
RV12 0.29 0.71    0
RV13 0.20 0.80    0
RV14 0.52 0.48    0
RV15 0.56 0.44    0
RV16 0.48 0.52    0
RV17 0.68 0.32    0
RV18 0.85 0.15    0
```

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

3.3 Análisis de distractores

Generalmente, un análisis de distractores ayuda a comprenden el comportamiento de los grupos a través de las distintas categorías de los ítems. A veces, los distractores no funcionan como se espera y puede ser necesario modificarlos. Para ello, cargamos la librería del paquete `ShinyItemAnalysis` y realizamos el análisis correspondiente con:

```
detach(package:psych)
library(ShinyItemAnalysis)
```

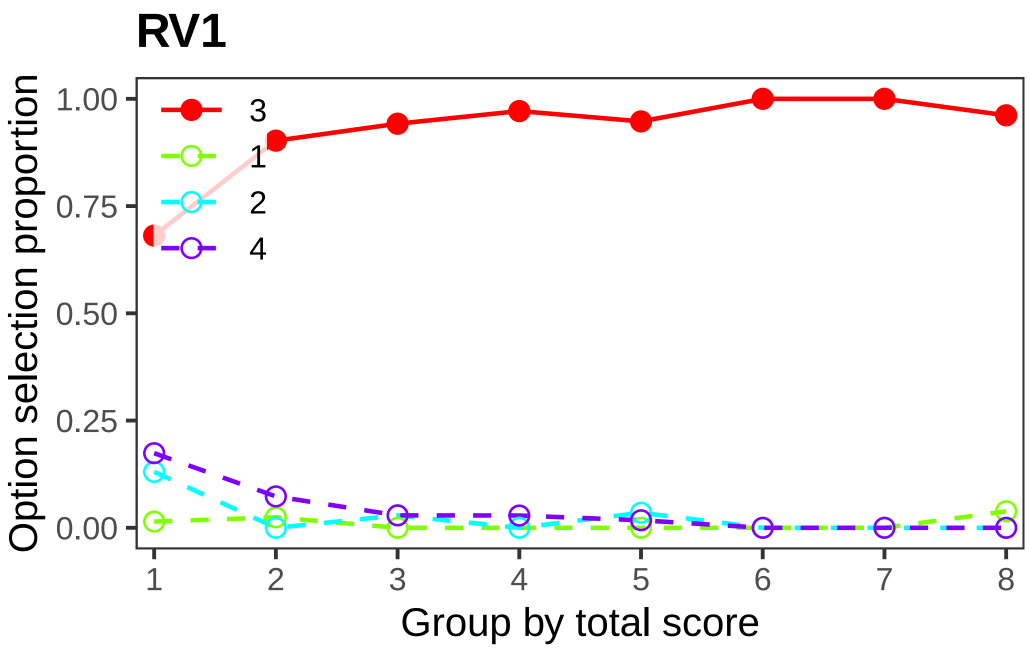
```
Warning: package 'ShinyItemAnalysis' was built under R version 4.5.3
```

```
This is ShinyItemAnalysis version 1.5.5
- to run the interactive {shiny} app, call `run_app()`
- to learn more, visit `ShinyItemAnalysis.org`
```

```
plotDistractorAnalysis(rv2,clave,num.groups=8,item="RV1")
```

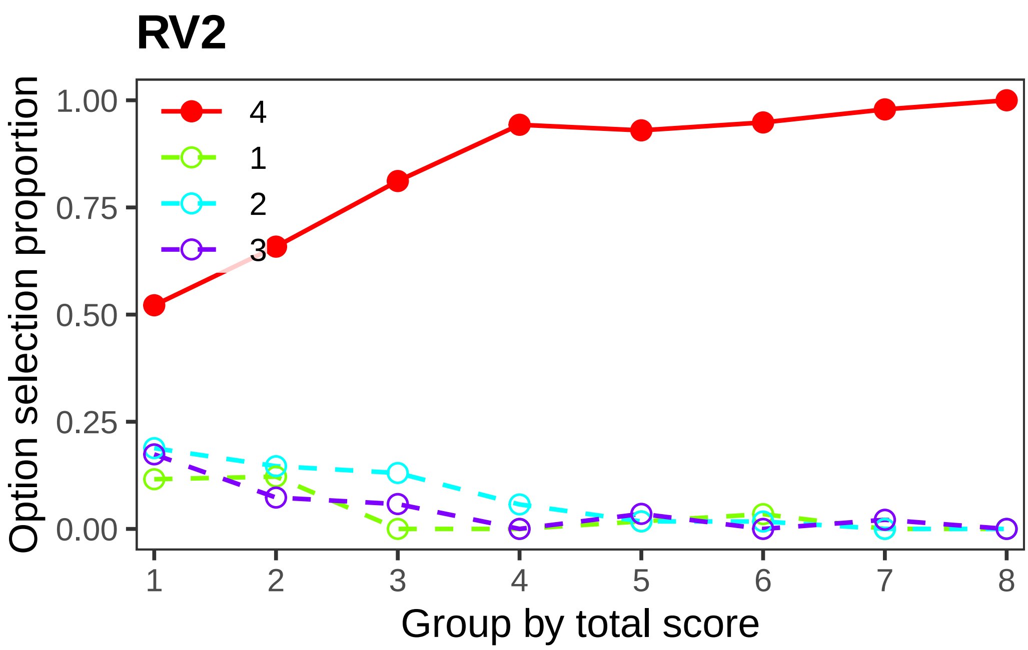
```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
i The deprecated feature was likely used in the ShinyItemAnalysis package.
  Please report the issue at
  <https://github.com/patriciamar/ShinyItemAnalysis/issues>.
```

```
$RV1
```



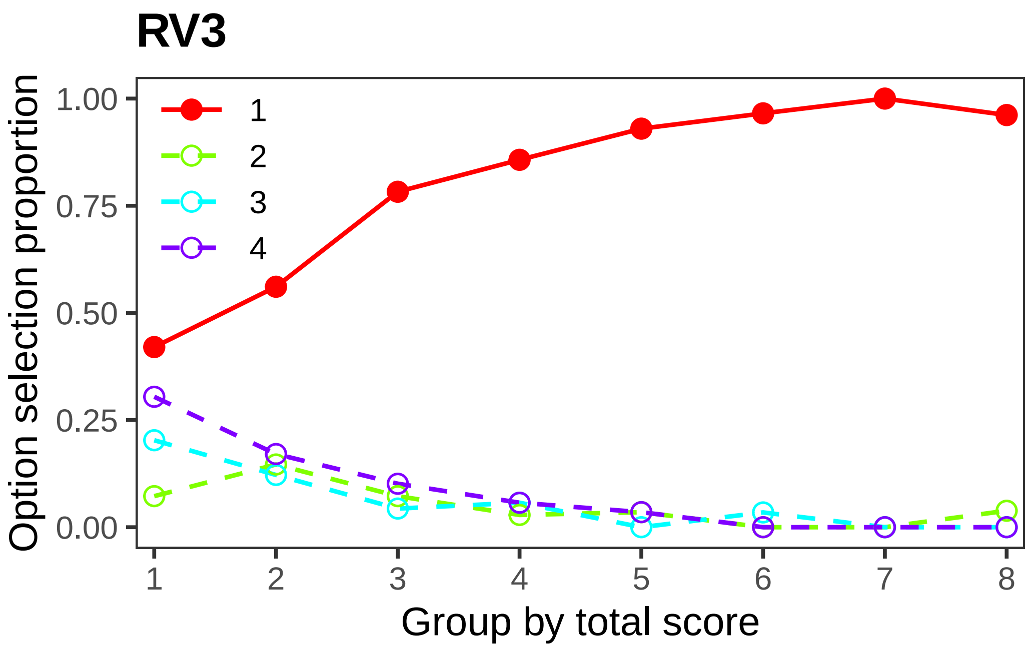

```
plotDistractorAnalysis(rv2,clave,num.groups=8,item="RV2")
```

```
$RV2
```



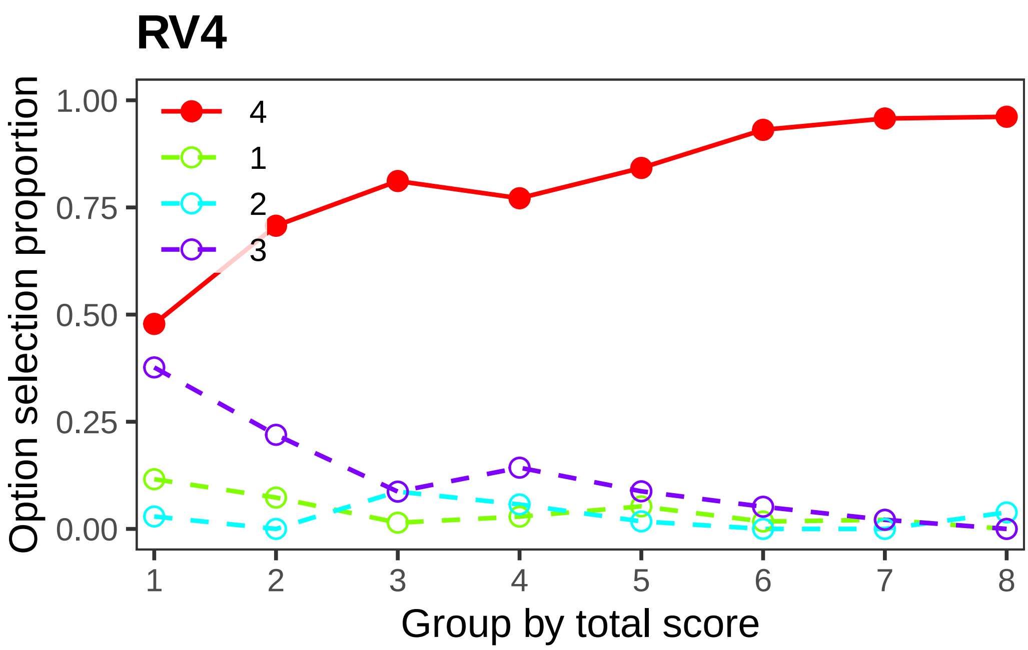

```
plotDistractorAnalysis(rv2,clave,num.groups=8,item="RV3")
```

```
$RV3
```



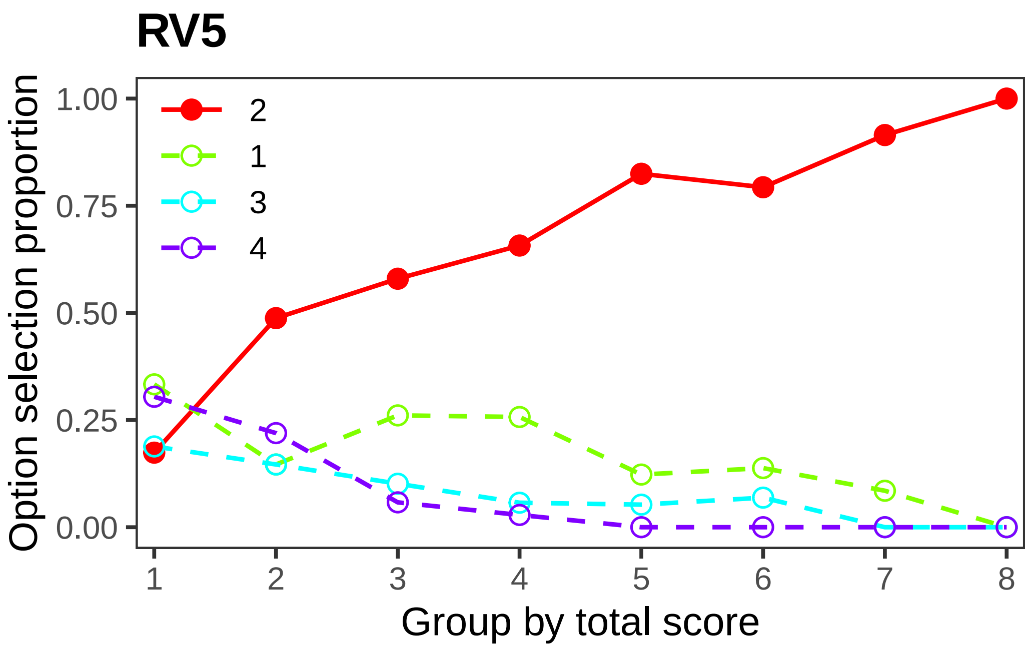

```
plotDistractorAnalysis(rv2,clave,num.groups=8,item="RV4")
```

```
$RV4
```



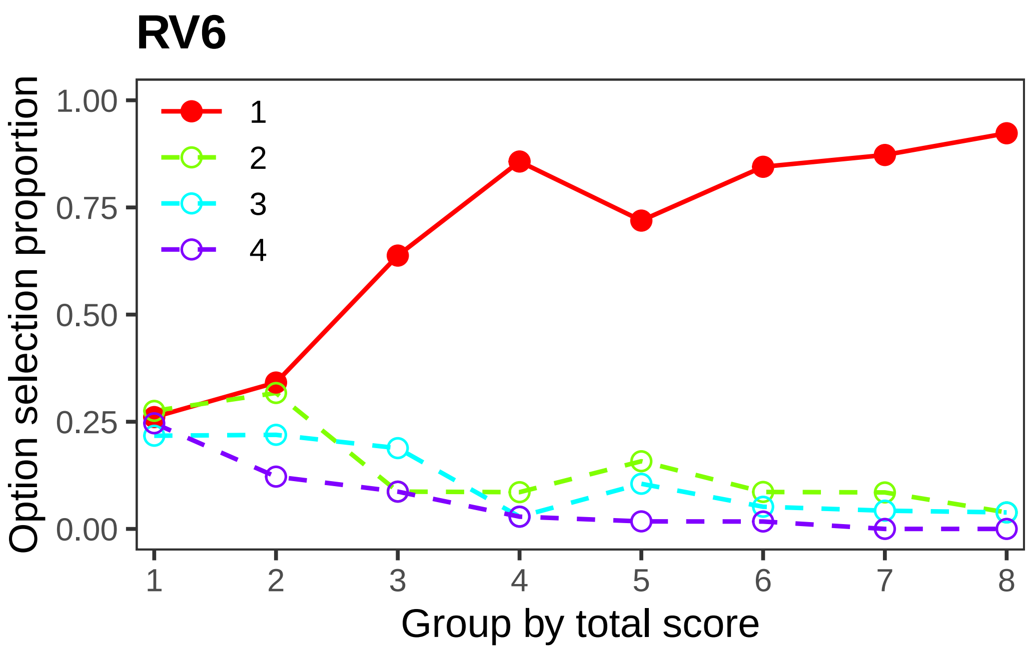

```
plotDistractorAnalysis(rv2,clave,num.groups=8,item="RV5")
```

```
$RV5
```



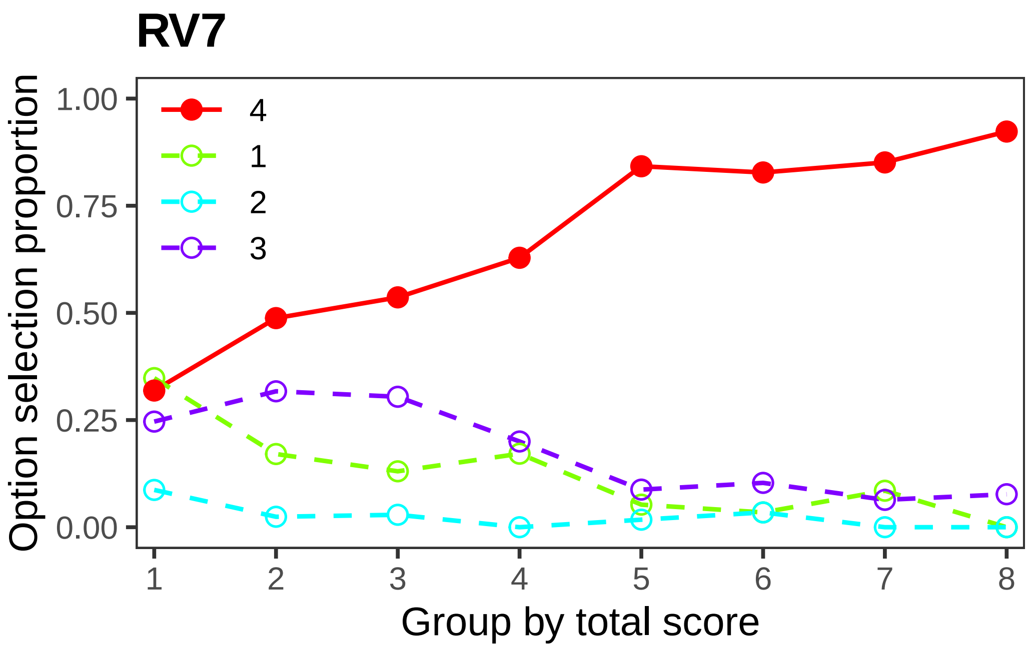

```
plotDistractorAnalysis(rv2,clave,num.groups=8,item="RV6")
```

```
$RV6
```



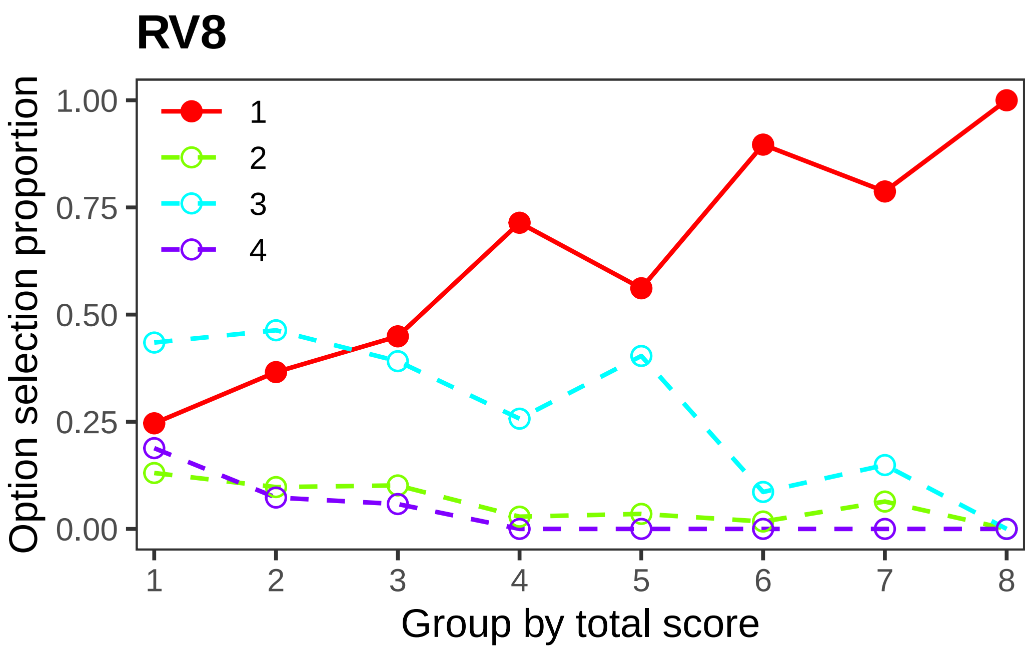

```
plotDistractorAnalysis(rv2,clave,num.groups=8,item="RV7")
```

```
$RV7
```



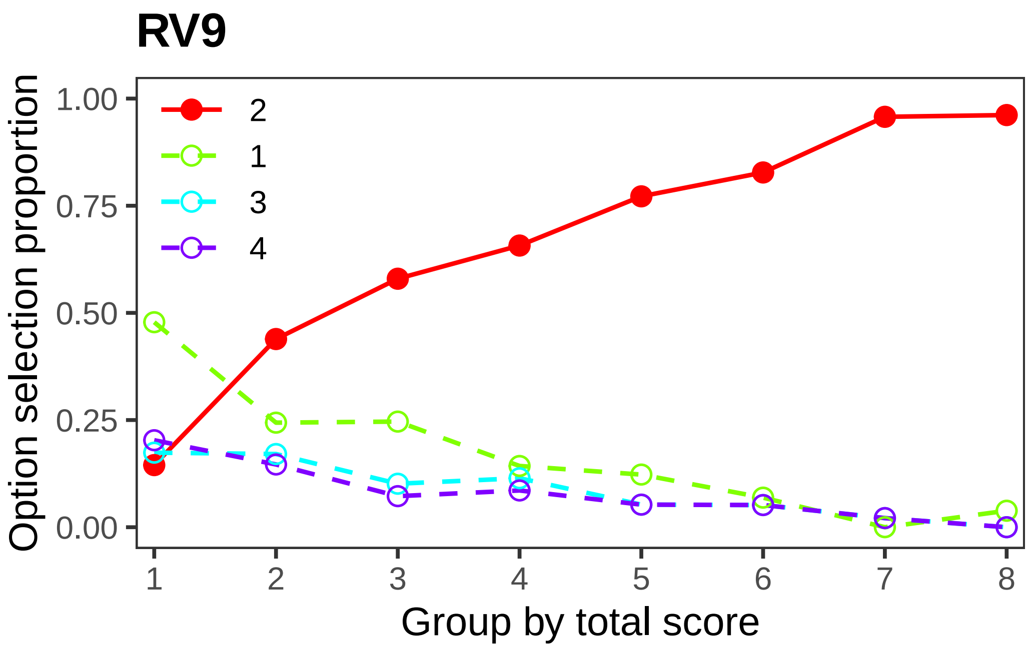

```
plotDistractorAnalysis(rv2,clave,num.groups=8,item="RV8")
```

```
$RV8
```



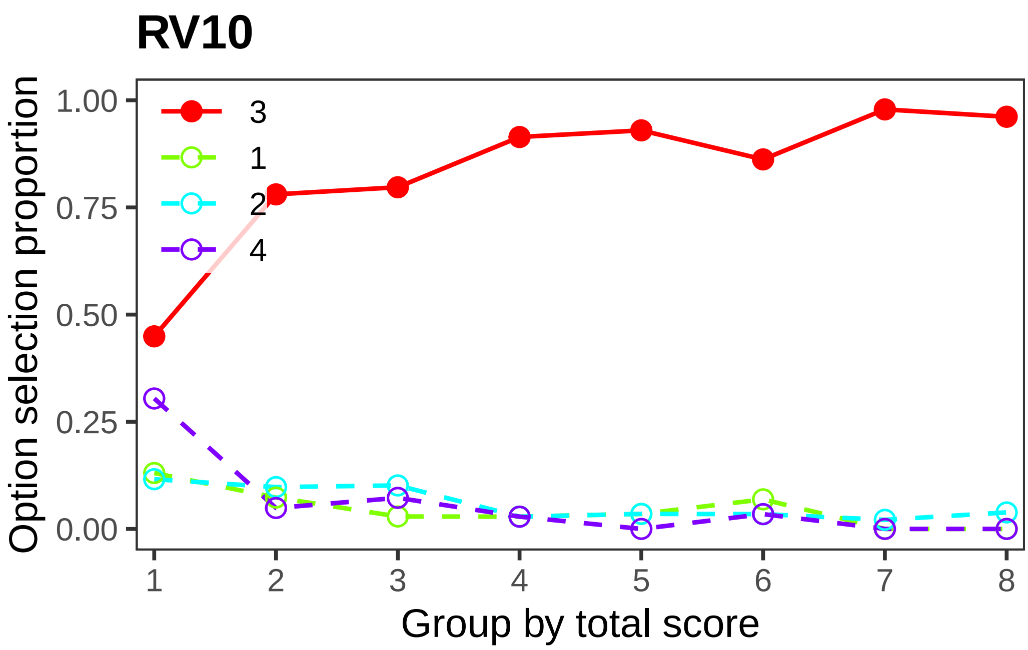

```
plotDistractorAnalysis(rv2,clave,num.groups=8,item="RV9")
```

$RV9



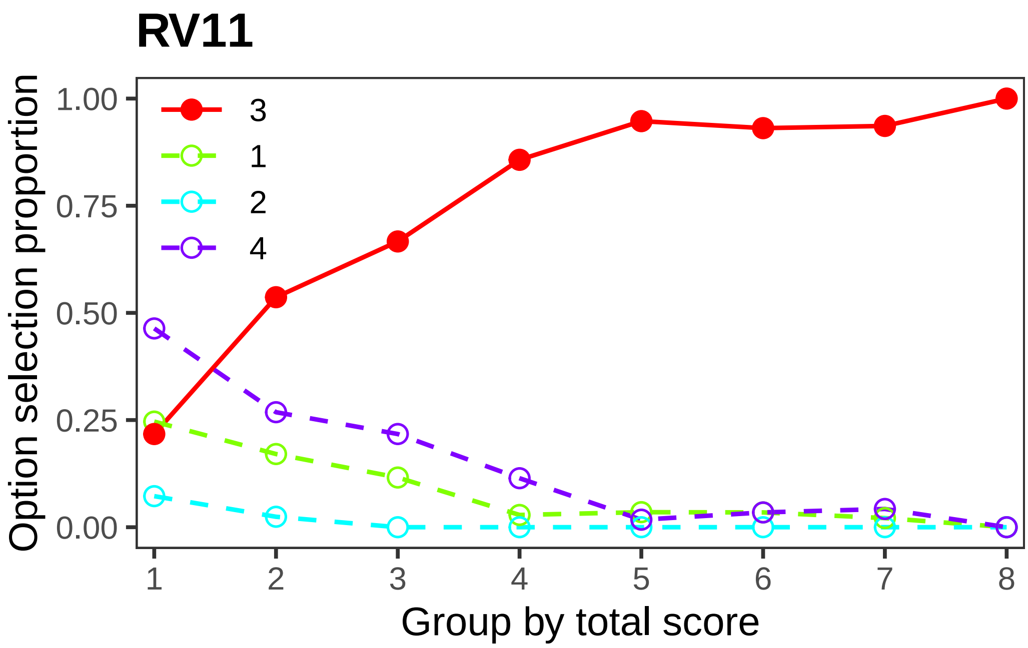

```
plotDistractorAnalysis(rv2,clave,num.groups=8,item="RV10")
```

$RV10



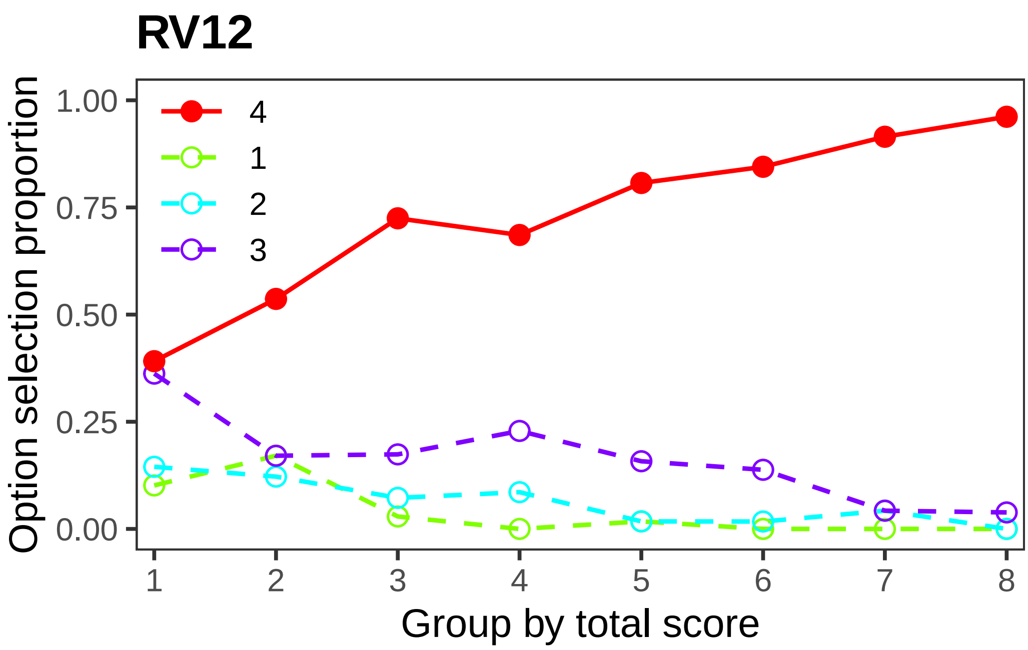

```
plotDistractorAnalysis(rv2,clave,num.groups=8,item="RV11")
```

```
$RV11
```



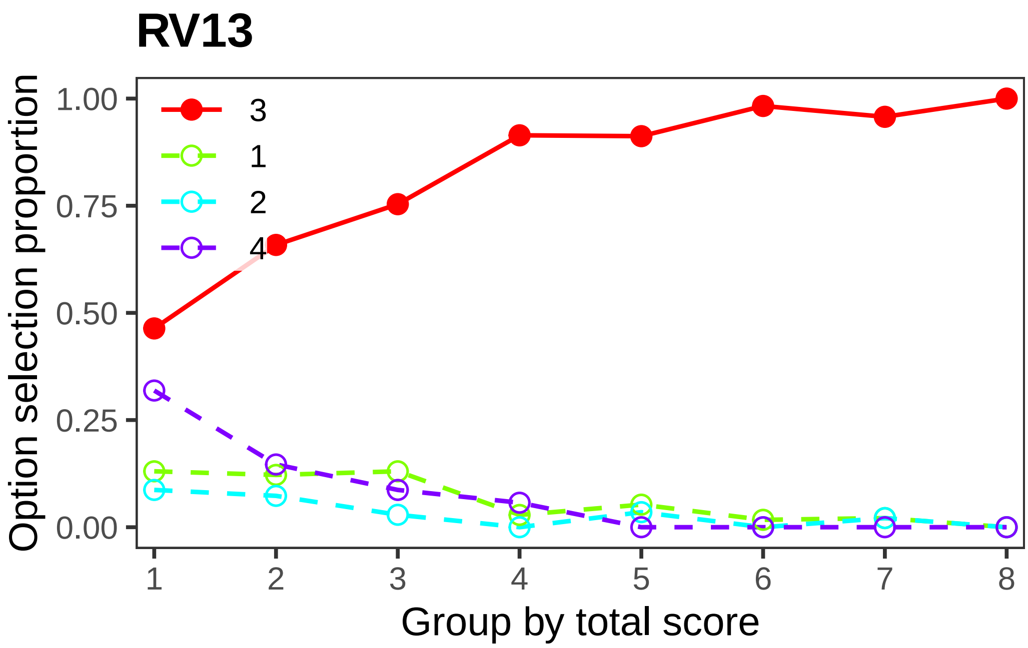

```
plotDistractorAnalysis(rv2,clave,num.groups=8,item="RV12")
```

```
$RV12
```



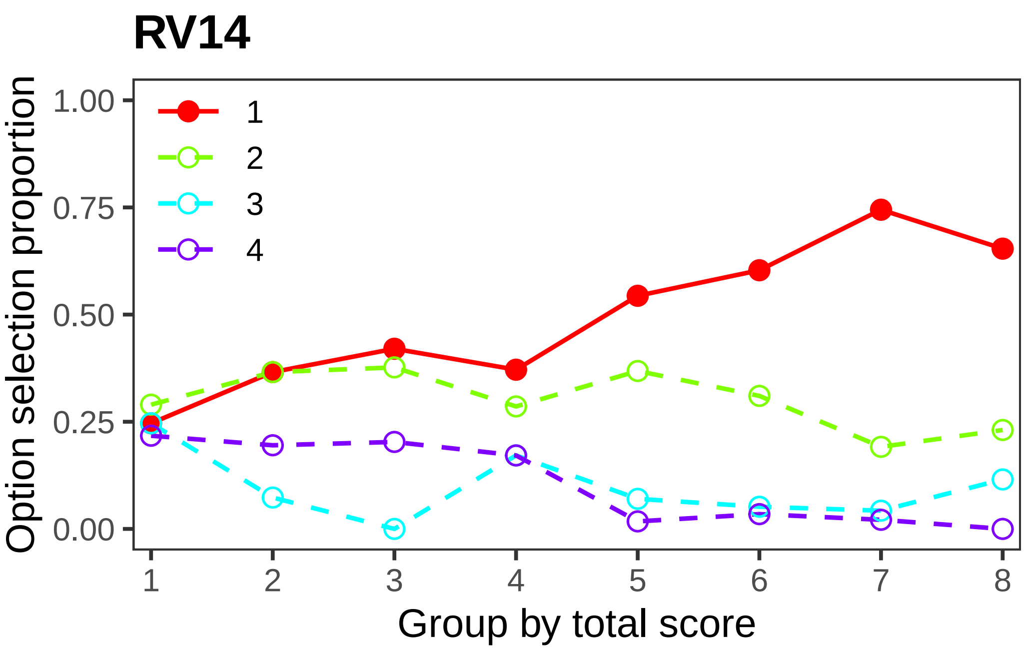

```
plotDistractorAnalysis(rv2,clave,num.groups=8,item="RV13")
```

```
$RV13
```



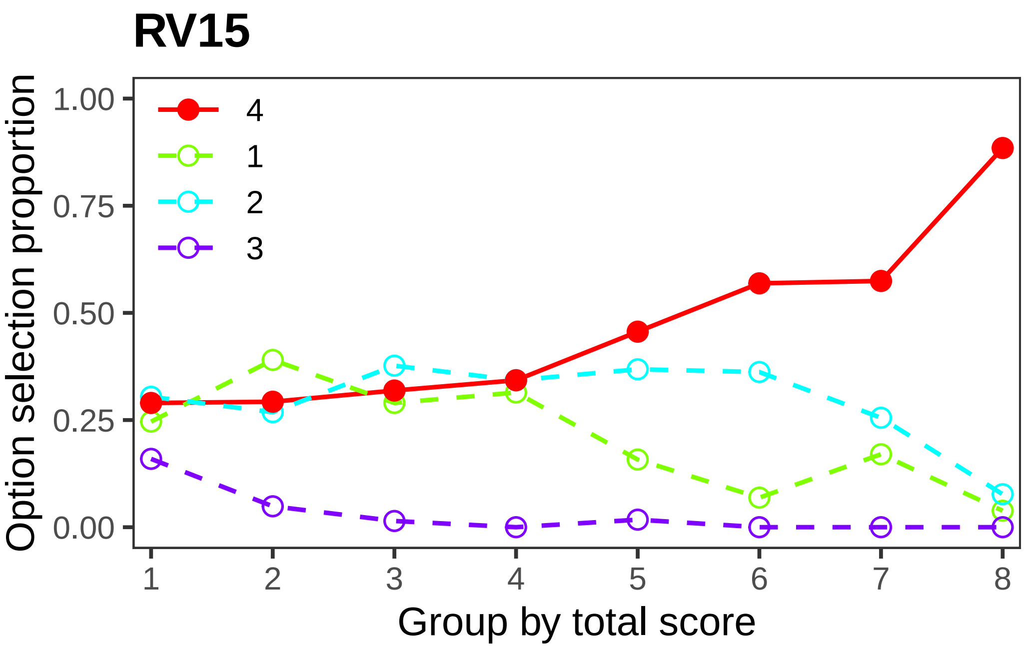

```
plotDistractorAnalysis(rv2,clave,num.groups=8,item="RV14")
```

```
$RV14
```



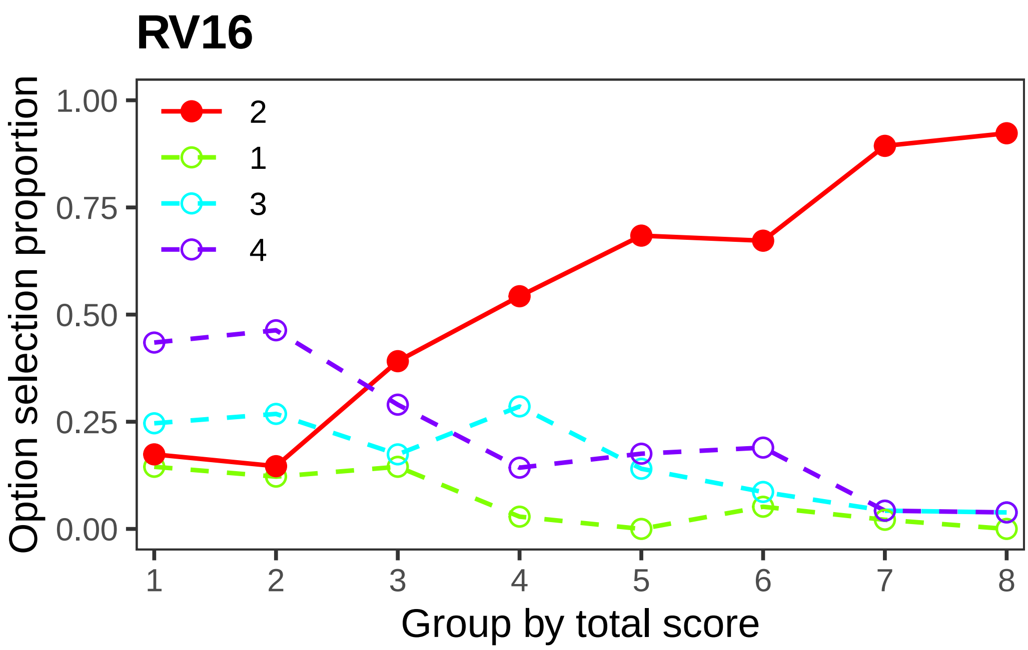

```
plotDistractorAnalysis(rv2,clave,num.groups=8,item="RV15")
```

$RV15



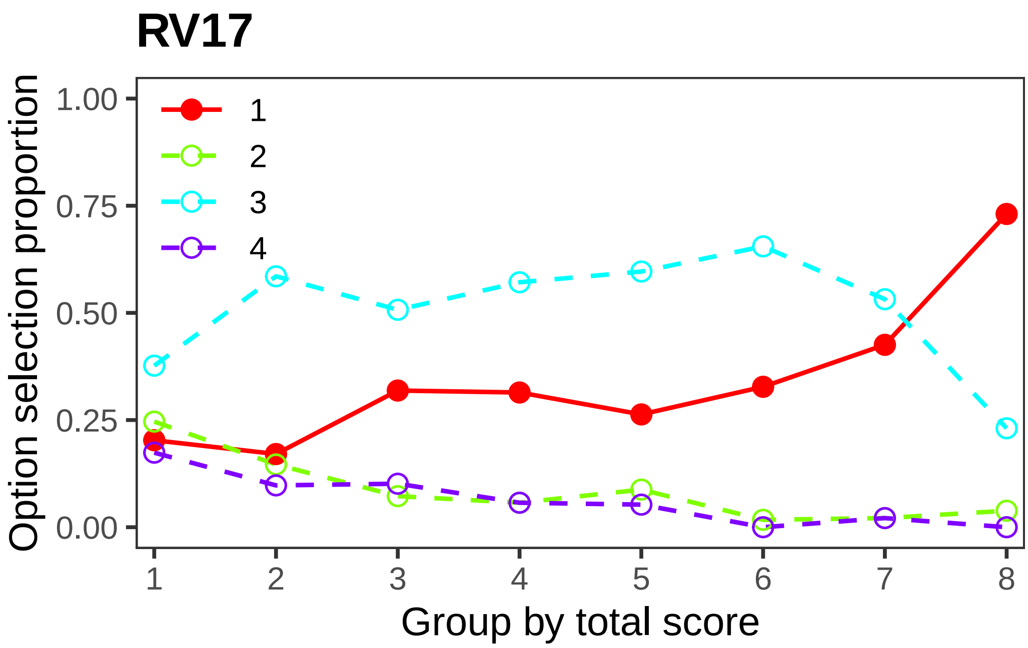

```
plotDistractorAnalysis(rv2,clave,num.groups=8,item="RV16")
```

$RV16



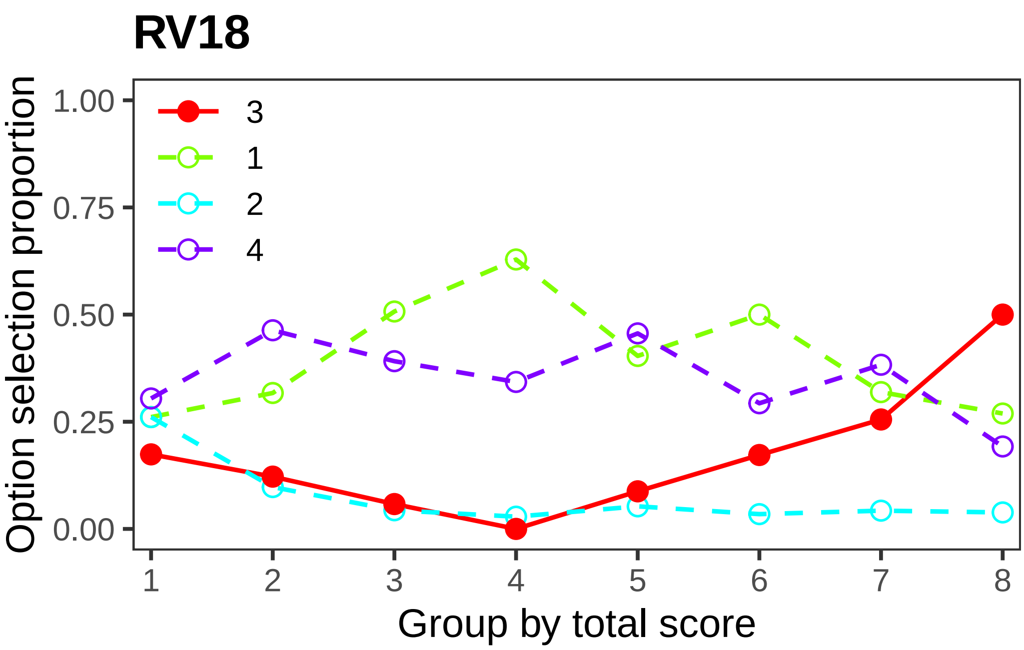

```
plotDistractorAnalysis(rv2,clave,num.groups=8,item="RV17")
```

$RV17




```
plotDistractorAnalysis(rv2,clave,num.groups=8,item="RV18")
```

$RV18



El análisis de distractores revela que las opciones incorrectas de algunos ítems no han funcionado adecuadamente. Por ejemplo, el distractor `2` en el ítem `RV14`, los distractores `1` y `2` en el ítem `RV15`, el distractor `3` en el ítem `RV17`, y `1` y `4` en el ítem `RV18` no han funcionado adecuamente, ya que se espera que la proporción de respuestas en estos distractores sean semejantes a la proporción de respuestas en el resto de distractores de cada ítem.

3.4 Análisis de la fiabilidad

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

```
detach(package:ShinyItemAnalysis)
library(psych)
```

```
Adjuntando el paquete: 'psych'

The following object is masked from 'package:polycor':

    polyserial
```

```
reliability(rv)
```

```
keys not specified, all items will be scored
```

```
Measures of reliability
reliability(keys = rv)
          omega_h alpha omega.tot  Uni  tau cong max.split min.split mean.r
All_items     0.5  0.72      0.74 0.63 0.67 0.93      0.79      0.63   0.13
          med.r n.items  CFI  ECV Beta  EVR  MAP
All_items  0.13      18 0.98 0.86 0.16 2.86 0.01
```

```r
splitHalf(rv)
```

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

Maximum split half reliability (lambda 4) =  0.78
Guttman lambda 6                          =  0.73
Average split half reliability            =  0.72
Guttman lambda 3 (alpha)                  =  0.72
Guttman lambda 2                          =  0.73
Minimum split half reliability  (beta)    =  0.63
Average interitem r =  0.13  with median =  0.13
```

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:

```r
glb.algebraic(rv)
```

```
Loading required namespace: Rcsdp

$glb
[1] 0.7985564


$solution
       RV1         RV2         RV3         RV4         RV5         RV6         RV7
```

```
0.04435465 0.04509502 0.06585761 0.04575419 0.09139744 0.09411299 0.08788800
       RV8        RV9       RV10       RV11       RV12       RV13       RV14
0.06655864 0.11582737 0.05755730 0.12638736 0.06372170 0.07073805 0.06305426
      RV15       RV16       RV17       RV18
0.06193214 0.12056819 0.04203266 0.05112091

$status
[1] 0

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

Una vez que se ha realizado el análisis con la `TCT` se puede comenzar a trabajar con los modelos de respuesta al ítem. Los resultados con estos modelos no tienen por qué coincidir con los de `TCT`, aunque, por regla general, las estimaciones de los parámetros de discriminación en el modelo logístico de 2-p están relacionadas con los índices de discriminación de la `TCT`.

Para trabajar con los modelos de respuesta al ítems debemos eliminar las librerías activas hasta el momento y cargar la librería del paquete `mirt` con:

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

Capítulo 4

Modelo de Rasch

4.1 Estimación de parámetros

Para estimar los parámetros en el modelo de Rasch debemos escribir:

```
m.rasch <- mirt(rv, model=1, itemtype="Rasch", SE=T, verbose=F)
```

En la función `mirt` aparece en primer lugar el objeto `rv` que contiene la matriz de datos que se analizará. El argumeto `model=1` especifica el número de dimensiones esperables. El modelo de medida se especifica con la opción `itemtype="Rasch"`; además, se solicita el cálculo de los errores típicos de los parámetros con `SE=T`. Finalmente, el argumento `verbose=F` inhibe la presentación de la secuencia de iteraciones en la estimación de parámetros. Si, a continuación, escribimos:

```
m.rasch
```

```
Call:
mirt(data = rv, model = 1, itemtype = "Rasch", SE = T, verbose = F)

Full-information item factor analysis with 1 factor(s).
Converged within 1e-04 tolerance after 14 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 =  8.141275

Log-likelihood = -3950.735
Estimated parameters: 19
AIC = 7939.469
BIC = 8015.402; SABIC = 7955.113
G2 (262124) = 3250.92, p = 1
RMSEA = 0, CFI = NaN, TLI = NaN
```

obtenemos los resultados del proceso de convergencia en la estimación de parámetros. Además, se obtienen los estadísticos AIC, BIC, SABIC, G2 y RMSEA que permiten comparar los resultados de este modelo con otros modelos de respuesta al ítem, con vistar a determinar qué modelo ajusta mejor los datos que se están analizando. Para evaluar si la estimación de parámetros ha encontrado la convergencia podemos escribir:

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

```
[1] TRUE
```

que indica que la estimación de parámetros ha encontrado la convergencia con el mensaje de TRUE.

Para obtener las estimaciones de los parámetros en este modelo debemos escribir:

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

```
$items
     a      b g u
RV1  1 -2.664 0 1
RV2  1 -1.793 0 1
RV3  1 -1.522 0 1
RV4  1 -1.522 0 1
RV5  1 -0.664 0 1
RV6  1 -0.715 0 1
RV7  1 -0.715 0 1
RV8  1 -0.394 0 1
RV9  1 -0.614 0 1
RV10 1 -1.644 0 1
RV11 1 -1.119 0 1
RV12 1 -1.048 0 1
RV13 1 -1.626 0 1
RV14 1  0.112 0 1
RV15 1  0.312 0 1
RV16 1 -0.075 0 1
RV17 1  0.906 0 1
RV18 1  1.971 0 1

$means
F1
 0

$cov
     F1
F1 0.79
```

Nótese que mirt genera una salida común para todos los modelos de respuesta al ítem. La primera columna presenta la pendiente que es proporcional al parámetro de discriminación (a); en el modelo de Rasch los valores de este parámetro deben ser iguales 1 en todos los ítems. En la segunda columna aparece el parámetro de dificultad (b). En la tercera columna aparece el párametro de pseudo-azar (g), que en el modelo de Rasch es igual a 0, y finalmente, en la cuarta columna aparece el parámetro (u) que se emplea para caracterizar los ítems en el modelo de cuatro parámetros (4-p), que no se trabaja en este documento. En esta salida, además, se obtiene que la media de los parámetros de habilidad se fija en $means = 0, mientras que la varianza vcov se estima libremente.

Para obtener los errores típicos de los parámetros podemos escribir:

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

```
$RV1
     a      b  g  u
par  1 -2.664  0  1
SE  NA  0.190 NA NA

$RV2
     a      b  g  u
par  1 -1.793  0  1
SE  NA  0.147 NA NA

$RV3
     a      b  g  u
par  1 -1.522  0  1
SE  NA  0.138 NA NA

$RV4
     a      b  g  u
par  1 -1.522  0  1
SE  NA  0.138 NA NA

$RV5
     a      b  g  u
par  1 -0.664  0  1
SE  NA  0.121 NA NA

$RV6
     a      b  g  u
par  1 -0.715  0  1
SE  NA  0.122 NA NA

$RV7
     a      b  g  u
par  1 -0.715  0  1
SE  NA  0.122 NA NA

$RV8
     a      b  g  u
par  1 -0.394  0  1
SE  NA  0.118 NA NA

$RV9
     a      b  g  u
par  1 -0.614  0  1
SE  NA  0.120 NA NA

$RV10
     a      b  g  u
par  1 -1.644  0  1
```

```
SE   NA   0.142 NA NA

$RV11
     a      b  g  u
par  1 -1.119  0  1
SE   NA  0.128 NA NA

$RV12
     a      b  g  u
par  1 -1.048  0  1
SE   NA  0.127 NA NA

$RV13
     a      b  g  u
par  1 -1.626  0  1
SE   NA  0.142 NA NA

$RV14
     a     b  g  u
par  1 0.112  0  1
SE   NA 0.117 NA NA

$RV15
     a     b  g  u
par  1 0.312  0  1
SE   NA 0.118 NA NA

$RV16
     a      b  g  u
par  1 -0.075  0  1
SE   NA  0.117 NA NA

$RV17
     a     b  g  u
par  1 0.906  0  1
SE   NA 0.124 NA NA

$RV18
     a     b  g  u
par  1 1.971  0  1
SE   NA 0.153 NA NA

$GroupPars
    MEAN_1 COV_11
par      0  0.790
SE      NA  0.085
```

Ahora, se pueden obtener los intervalos de confianza de los parámetros de dificultad con:

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

```
$RV1
         a      b  g  u
```

```
par      1 -2.664  0  1
CI_2.5  NA -3.037 NA NA
CI_97.5 NA -2.291 NA NA

$RV2
         a      b  g  u
par      1 -1.793  0  1
CI_2.5  NA -2.082 NA NA
CI_97.5 NA -1.505 NA NA

$RV3
         a      b  g  u
par      1 -1.522  0  1
CI_2.5  NA -1.793 NA NA
CI_97.5 NA -1.251 NA NA

$RV4
         a      b  g  u
par      1 -1.522  0  1
CI_2.5  NA -1.793 NA NA
CI_97.5 NA -1.251 NA NA

$RV5
         a      b  g  u
par      1 -0.664  0  1
CI_2.5  NA -0.901 NA NA
CI_97.5 NA -0.427 NA NA

$RV6
         a      b  g  u
par      1 -0.715  0  1
CI_2.5  NA -0.953 NA NA
CI_97.5 NA -0.477 NA NA

$RV7
         a      b  g  u
par      1 -0.715  0  1
CI_2.5  NA -0.953 NA NA
CI_97.5 NA -0.477 NA NA

$RV8
         a      b  g  u
par      1 -0.394  0  1
CI_2.5  NA -0.626 NA NA
CI_97.5 NA -0.162 NA NA

$RV9
         a      b  g  u
par      1 -0.614  0  1
CI_2.5  NA -0.850 NA NA
CI_97.5 NA -0.378 NA NA
```

```
$RV10
         a      b  g  u
par      1 -1.644  0  1
CI_2.5  NA -1.923 NA NA
CI_97.5 NA -1.366 NA NA

$RV11
         a      b  g  u
par      1 -1.119  0  1
CI_2.5  NA -1.370 NA NA
CI_97.5 NA -0.868 NA NA

$RV12
         a      b  g  u
par      1 -1.048  0  1
CI_2.5  NA -1.297 NA NA
CI_97.5 NA -0.799 NA NA

$RV13
         a      b  g  u
par      1 -1.626  0  1
CI_2.5  NA -1.904 NA NA
CI_97.5 NA -1.349 NA NA

$RV14
         a      b  g  u
par      1  0.112  0  1
CI_2.5  NA -0.117 NA NA
CI_97.5 NA  0.341 NA NA

$RV15
         a     b  g  u
par      1 0.312  0  1
CI_2.5  NA 0.081 NA NA
CI_97.5 NA 0.542 NA NA

$RV16
         a      b  g  u
par      1 -0.075  0  1
CI_2.5  NA -0.304 NA NA
CI_97.5 NA  0.154 NA NA

$RV17
         a     b  g  u
par      1 0.906  0  1
CI_2.5  NA 0.663 NA NA
CI_97.5 NA 1.148 NA NA

$RV18
         a     b  g  u
```

```
par      1 1.971  0  1
CI_2.5  NA 1.670 NA NA
CI_97.5 NA 2.271 NA NA

$GroupPars
        MEAN_1 COV_11
par          0  0.790
CI_2.5      NA  0.623
CI_97.5     NA  0.956
```

Para ordenar los ítems en función de los parámetros de dificultad en este modelo empleamos la secuencia siguiente:

```
coeficientes <- coef(m.rasch,simplify=T,IRTpars=T)
coeficientes_items <- coeficientes$items
b <- coeficientes_items[, "b"]
tabla <- data.frame(
Item=rownames(coeficientes$items),
Dificultad_b = b)
tabla_ordenada_b <- tabla[order(tabla$Dificultad_b), ]
print(tabla_ordenada_b)
```

```
     Item Dificultad_b
RV1   RV1  -2.66400978
RV2   RV2  -1.79342897
RV10 RV10  -1.64429228
RV13 RV13  -1.62641900
RV4   RV4  -1.52230897
RV3   RV3  -1.52230897
RV11 RV11  -1.11894913
RV12 RV12  -1.04805391
RV6   RV6  -0.71500509
RV7   RV7  -0.71500509
RV5   RV5  -0.66431244
RV9   RV9  -0.61417525
RV8   RV8  -0.39415863
RV16 RV16  -0.07500472
RV14 RV14   0.11187059
RV15 RV15   0.31151402
RV17 RV17   0.90580279
RV18 RV18   1.97069278
```

y la media y desviación típica de estos parámetros se puede obtener con:

```
mean(b)
```

```
[1] -0.7120862
```

```
sd(b)
```

```
[1] 1.093726
```

```
min(b)
```

```
[1] -2.66401
```

```r
max(b)
```

```
[1] 1.970693
```

4.2 Evaluación del ajuste

Para obtener una evaluación global del ajuste del modelo a los datos, el paquete `mirt` proporciona el estadístico `M2` con:

```r
M2(m.rasch)
```

```
            M2  df            p      RMSEA    RMSEA_5   RMSEA_95      SRMSR
stats 299.1176 152 1.149492e-11 0.04912904 0.0408233 0.05723945 0.08470245
           TLI       CFI
stats 0.884842 0.8855946
```

Este valor ha resultado significativo (`p < 0.05`), por lo que se rechaza que este conjunto de ítems sigan el modelo de Rasch. Para probar qué ítems pueden haber dado lugar a este resultado se emplea:

```r
itemfit(m.rasch)
```

```
   item   S_X2 df.S_X2 RMSEA.S_X2 p.S_X2
1   RV1  8.678      11      0.000  0.652
2   RV2  6.617      11      0.000  0.829
3   RV3 11.078      10      0.016  0.351
4   RV4  8.449      10      0.000  0.585
5   RV5 10.725      11      0.000  0.467
6   RV6  9.672      11      0.000  0.560
7   RV7 12.795      11      0.020  0.307
8   RV8 24.957      11      0.056  0.009
9   RV9 12.053      11      0.015  0.360
10 RV10 15.607      11      0.032  0.156
11 RV11 20.503      10      0.051  0.025
12 RV12  6.417      11      0.000  0.844
13 RV13  7.009      10      0.000  0.725
14 RV14 27.580      10      0.066  0.002
15 RV15 37.729      10      0.083  0.000
16 RV16 12.517      10      0.025  0.252
17 RV17 44.158      10      0.092  0.000
18 RV18 77.186       8      0.147  0.000
```

que obtiene el estadístico $S - \chi^2$ para cada ítem. Parece, entonces, que los items `RV8`, `RV11`, `RV14`, `RV15`, `RV17` y `RV18` no siguen el modelo puesto a prueba, dado que los estadísticos de ajuste han resultado significativos (`p < 0.05`).

4.3 Parámetros de habilidad

Para estimar los parámetros de habilidad [1] en este modelo con `mirt` debemos emplear:

[1] El paquete `mirt` estima los parámetros de habilidad para todos los miembros del grupo. Utilizando la función `head()` se reduce la salida impresa a los seis primeros casos.

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

```
            F1     SE_F1
[1,] 0.8285292 0.5193655
[2,] 0.3240616 0.4871257
[3,] 1.1092485 0.5408850
[4,] 0.5682133 0.5015918
[5,] 0.5682133 0.5015918
[6,] 1.4155446 0.5666450
```

que produce las estimaciones esperadas a posteriori `EAP` de los parámetros de habilidad de los 400 estudiantes junto con sus correspondientes errores típicos. Si se prefiere, se puede sustituir en `method` la opción `EAP` por el estimador máximo a posteriori (`MAP`) o por el estimador de máxima verosimilitud ponderado (`WLE`).

Para estudiar el ajuste de estos patrones de respuesta emplearemos:

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

```
     outfit   z.outfit     infit    z.infit         Zh
1 0.5080217 -0.8740364 0.6114359 -1.1161927  1.0178490
2 0.6507767 -0.8115817 0.7501690 -0.8621346  0.8751039
3 0.3429377 -1.1139670 0.5190814 -1.2716390  1.1471691
4 0.5013912 -1.1038598 0.6550283 -1.1137326  1.0940525
5 1.8067798  1.5049430 1.5568053  1.5840377 -1.7864230
6 0.1712698 -1.3957725 0.2720913 -2.0480484  1.4155328
```

que produce los estadísticos de ajuste de las 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 `Zh < 1.5`.

4.4 Evaluación de la unidimensionalidad

En primer lugar obtenemos los parámetros con dos dimensiones en `mirt` utilizando:

```r
m.rasch.2d <- mirt(rv,model=2,itemtype="2PL",verbose=F)
```

```
Warning: EM cycles terminated after 500 iterations.
```

Ahora se obtienen los estadísticos de ajuste del modelo unidimensional:

```r
M2(m.rasch)
```

```
            M2  df            p      RMSEA   RMSEA_5   RMSEA_95      SRMSR
stats 299.1176 152 1.149492e-11 0.04912904 0.0408233 0.05723945 0.08470245
           TLI       CFI
stats 0.884842 0.8855946
```

y se compara con los estadísticos de ajuste del modelo con dos dimensiones:

```r
M2(m.rasch.2d)
```

```
            M2  df         p RMSEA RMSEA_5    RMSEA_95      SRMSR   TLI CFI
stats 97.17313 118 0.9193575     0       0 0.009267968 0.03582227 1.021   1
```

En este caso,parece que una sola dimensión no es suficiente para explicar la variación de las respuestas en este test de razonamiento verbal, dado que los estadísticos de ajuste de la solución bidimensional han obtenido valores por debajo de los puntos de corte de los diferentes estadísticos.

4.5 Independencia local de los ítems

En este paquete se puede evaluar la independencia local de los ítems con los estadísticos `LD` y `Q3` de Yen. Para emplear el estadístico `Q3` escribimos:

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

```
Q3 summary statistics:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 -0.202  -0.080  -0.046  -0.041  -0.001   0.153 

        RV1    RV2    RV3    RV4    RV5    RV6    RV7    RV8    RV9   RV10
RV1   1.000 -0.120  0.028 -0.042  0.027 -0.017 -0.096 -0.058  0.153  0.016
RV2  -0.120  1.000  0.005 -0.028 -0.005 -0.057  0.024 -0.027 -0.108  0.045
RV3   0.028  0.005  1.000 -0.033  0.050 -0.011 -0.013  0.031 -0.019  0.003
RV4  -0.042 -0.028 -0.033  1.000 -0.101  0.075 -0.050 -0.080 -0.057 -0.043
RV5   0.027 -0.005  0.050 -0.101  1.000  0.000 -0.054 -0.022 -0.020 -0.089
RV6  -0.017 -0.057 -0.011  0.075  0.000  1.000 -0.105 -0.001 -0.085  0.012
RV7  -0.096  0.024 -0.013 -0.050 -0.054 -0.105  1.000 -0.065  0.085  0.025
RV8  -0.058 -0.027  0.031 -0.080 -0.022 -0.001 -0.065  1.000 -0.081 -0.078
RV9   0.153 -0.108 -0.019 -0.057 -0.020 -0.085  0.085 -0.081  1.000 -0.040
RV10  0.016  0.045  0.003 -0.043 -0.089  0.012  0.025 -0.078 -0.040  1.000
RV11 -0.046  0.015  0.046 -0.040  0.068  0.048 -0.120 -0.031  0.105  0.040
RV12 -0.014 -0.035 -0.066  0.004  0.003 -0.070 -0.068 -0.018 -0.078 -0.090
RV13  0.031 -0.034 -0.019 -0.009 -0.087 -0.034 -0.101  0.032  0.045  0.008
RV14 -0.075 -0.019 -0.069 -0.058 -0.055 -0.042 -0.078 -0.017 -0.141 -0.103
RV15  0.021 -0.021 -0.106 -0.081 -0.056 -0.153 -0.004 -0.152 -0.081 -0.035
RV16 -0.009 -0.036  0.066 -0.107 -0.066 -0.107  0.015 -0.030  0.130 -0.001
RV17 -0.086 -0.033 -0.125 -0.050 -0.060 -0.016 -0.134 -0.084 -0.143 -0.076
RV18  0.017 -0.071 -0.121 -0.032 -0.043 -0.104 -0.080 -0.080 -0.073 -0.155
       RV11   RV12   RV13   RV14   RV15   RV16   RV17   RV18
RV1  -0.046 -0.014  0.031 -0.075  0.021 -0.009 -0.086  0.017
RV2   0.015 -0.035 -0.034 -0.019 -0.021 -0.036 -0.033 -0.071
RV3   0.046 -0.066 -0.019 -0.069 -0.106  0.066 -0.125 -0.121
RV4  -0.040  0.004 -0.009 -0.058 -0.081 -0.107 -0.050 -0.032
RV5   0.068  0.003 -0.087 -0.055 -0.056 -0.066 -0.060 -0.043
RV6   0.048 -0.070 -0.034 -0.042 -0.153 -0.107 -0.016 -0.104
RV7  -0.120 -0.068 -0.101 -0.078 -0.004  0.015 -0.134 -0.080
RV8  -0.031 -0.018  0.032 -0.017 -0.152 -0.030 -0.084 -0.080
RV9   0.105 -0.078  0.045 -0.141 -0.081  0.130 -0.143 -0.073
RV10  0.040 -0.090  0.008 -0.103 -0.035 -0.001 -0.076 -0.155
RV11  1.000 -0.080  0.123 -0.107 -0.127  0.078 -0.151 -0.047
RV12 -0.080  1.000  0.035 -0.019 -0.094 -0.115 -0.080  0.003
RV13  0.123  0.035  1.000 -0.151 -0.053 -0.036 -0.079 -0.071
RV14 -0.107 -0.019 -0.151  1.000 -0.105 -0.050 -0.058 -0.097
RV15 -0.127 -0.094 -0.053 -0.105  1.000 -0.202 -0.049  0.058
RV16  0.078 -0.115 -0.036 -0.050 -0.202  1.000 -0.058 -0.080
```

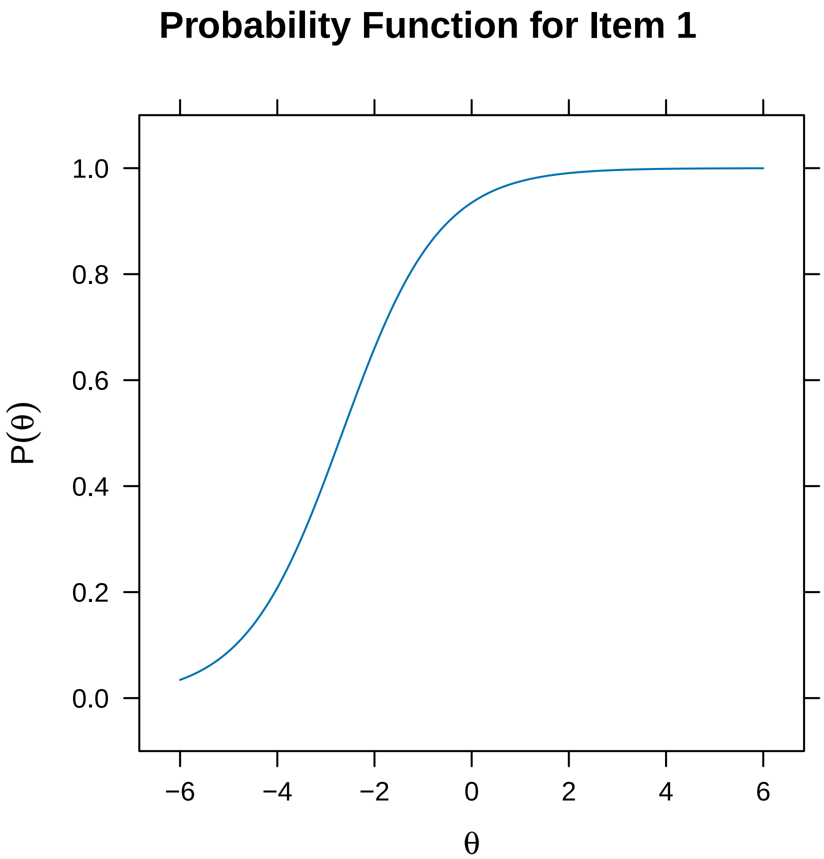
```
RV17 -0.151 -0.080 -0.079 -0.058 -0.049 -0.058  1.000 -0.046
RV18 -0.047  0.003 -0.071 -0.097  0.058 -0.080 -0.046  1.000
```

que, en principio, no detecta incumplimiento de este principio dado que prácticamente todos los valores de `Q3` están por debajo del punto de corte `0.2`.

4.6 Gráficos

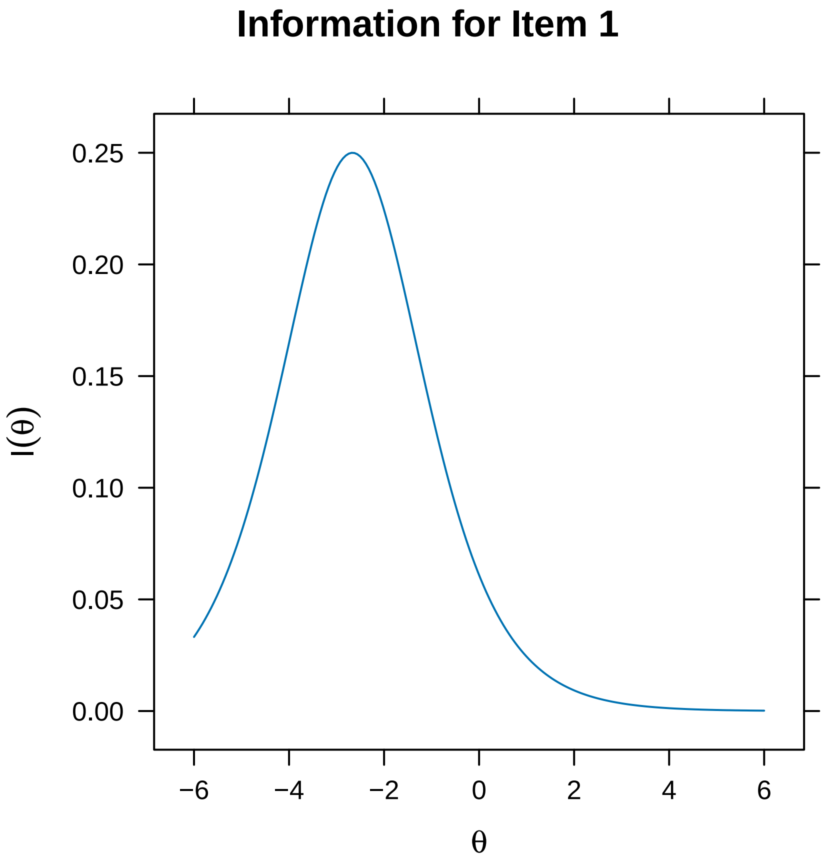
En este paquete se puede obtener la `FRI` de un ítem (e.g., ítem `1`) utilizando:

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



y la función de información de ese ítem (`FII`) con:

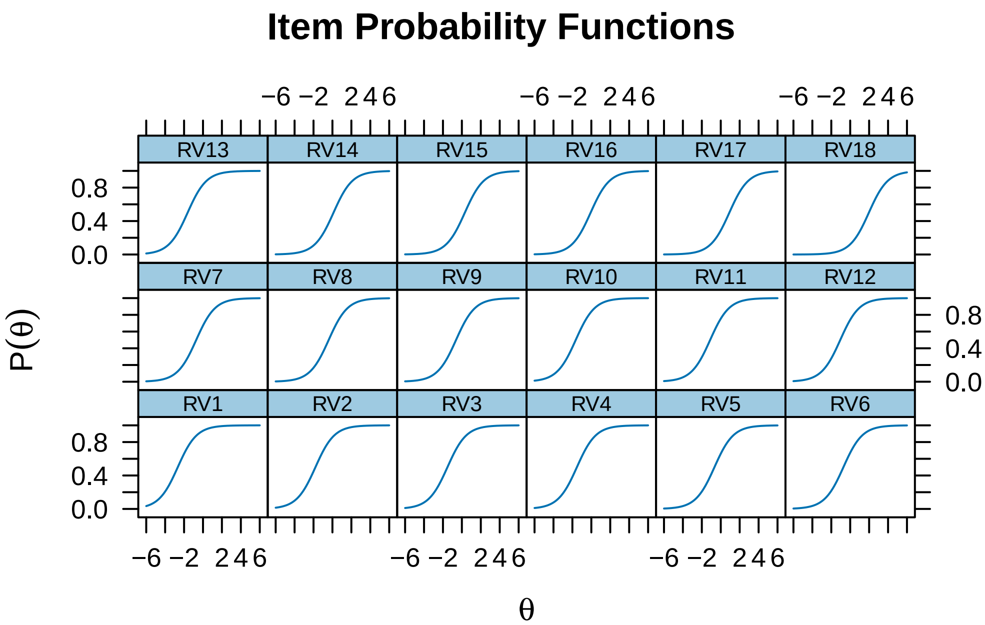
```
itemplot(m.rasch, item=1, type="info")
```



que como se aprecia ofrece la máxima información en el intervalo `[-2, +2]`.

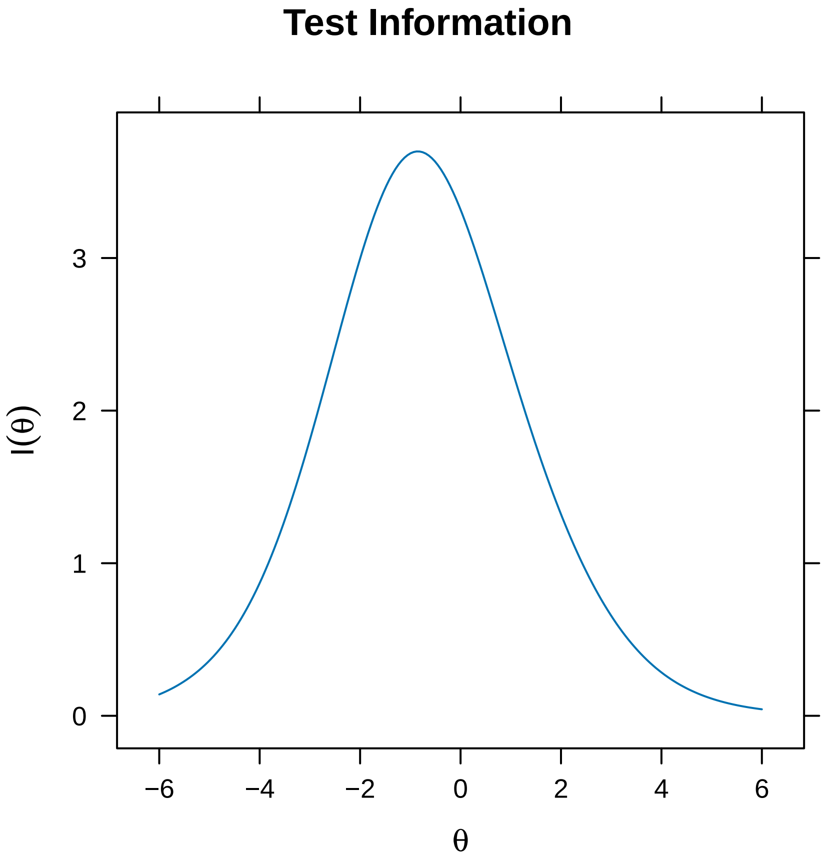
Las `FRIs` de todos los ítems se obtienen con:

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



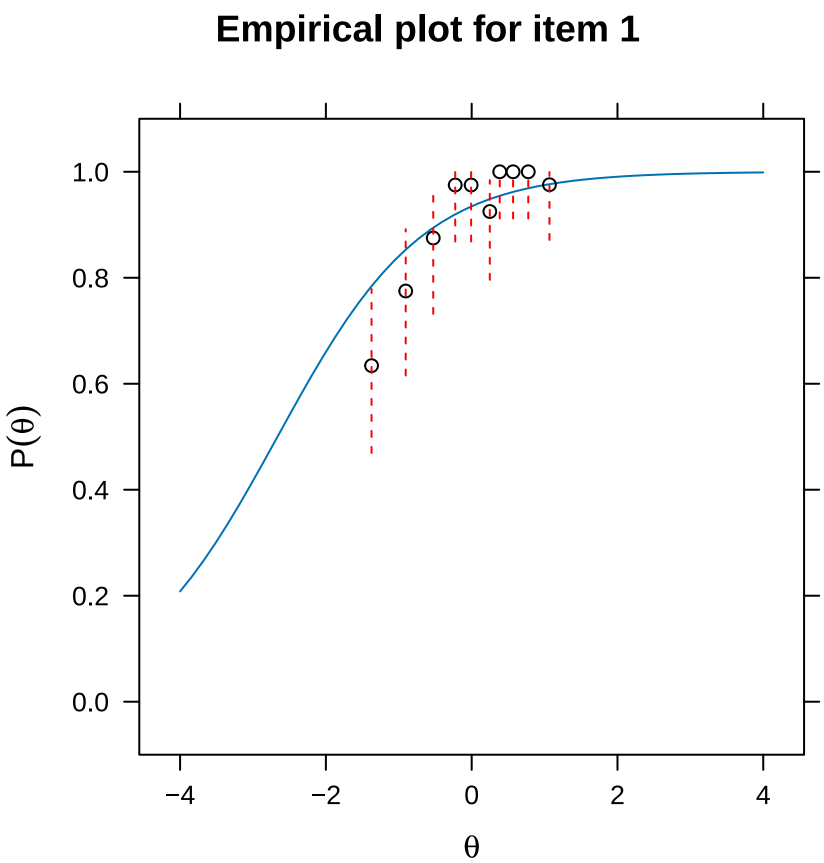
y la `FIT` para el test completo se obtiene con:

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



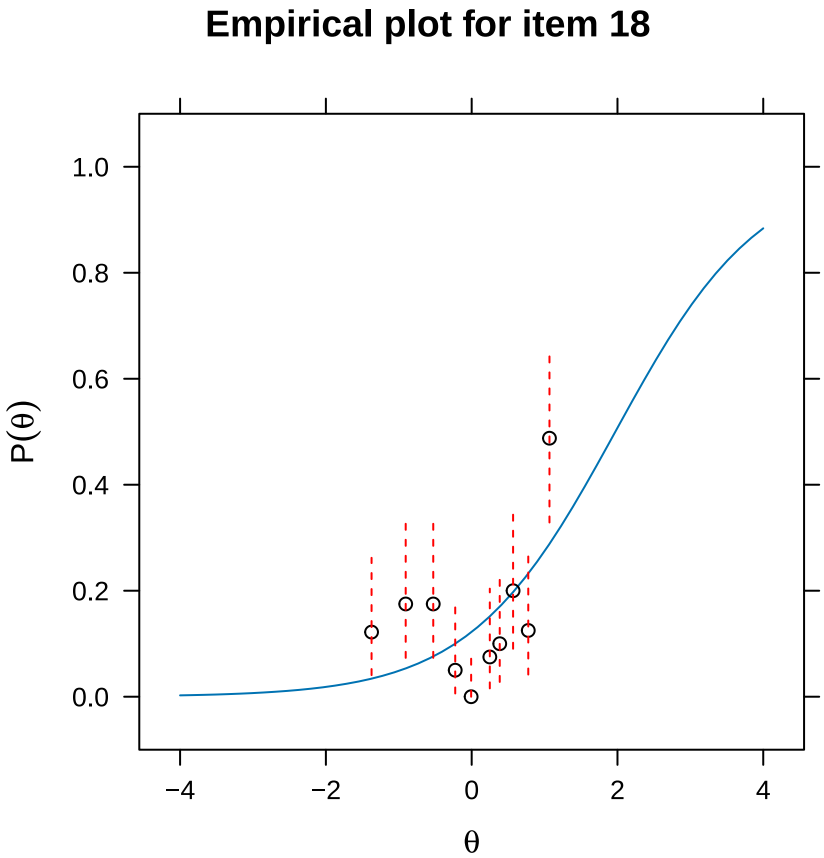
También es posible estudiar el ajuste de un ítem al modelo con:

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



Este gráfico produce una `FRI` teórica (línea azul) y una `FRI` empírica resultante de unir todos los puntos (blancos) en el gráfico. Además, la línea de puntos (roja) representa el intervalo de confianza al parámetro de dificultad de cada ítem. En principio, parece que la línea de puntos (blancos) ofrece una `FRI` muy cercana a la `FRI` pronosticada por el modelo. Sin embargo, en el ítem `RV18`, por ejemplo, el resultado es diferente, dado que se aprecia una línea de puntos paralela al eje de abscisas, de modo que no se puede asegurar que sigue este modelo.

```
itemfit(m.rasch, empirical.plot=18, empirical.CI=.95)
```


Empirical plot for item 18
1.0
0.8
0.6
0.4
0.2
0.0
P(θ)
−4
−2
0
2
4
θ

Capítulo 5

Modelo logístico de 1-p

5.1 Estimación de parámetros

La estimación de parámetros en el modelo logístico de `1-p` con el paquete `mirt` no se realiza cambiando directamente la función `itemtype`, sino que necesita algunos pasos previos, dado que en este paquete la función básica se define con el modelo logístico de `2-p`. Para especificar el modelo logístico de `1-p` en `mirt` debemos emplear los pasos siguientes:

```
esp <- 'F = 1-18
CONSTRAIN=(1-18,a1)'
m.1p <- mirt(rv, model=esp, itemtype="2PL", SE=T, verbose=F)
```

En primer lugar, se crea un objeto (e.g., `esp`) que contiene la definición de un factor unidimensional para los `18` ítems. A continuación, el argumento `CONSTRAIN` restringe a que todos los ítems tengan el mismo parámetro de discriminación. Ahora, en el argumento `model` se especifica el objeto con las restricciones (`esp`), y en el argumento `itemtype` se especifica el modelo logístico de `2-p` con el argumento 2PL. Si escribimos ahora:

```
m.1p
```

```

Call:
mirt(data = rv, model = esp, itemtype = "2PL", SE = T, verbose = F)

Full-information item factor analysis with 1 factor(s).
Converged within 1e-04 tolerance after 12 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 =  25.05298

Log-likelihood = -3950.735
Estimated parameters: 36
AIC = 7939.469
```

```
BIC = 8015.402; SABIC = 7955.113
G2 (262124) = 3250.92, p = 1
RMSEA = 0, CFI = NaN, TLI = NaN
```

tenemos los criterios de convergencia obtenidos en la estimación de parámetros y los estadísticos de ajuste global del modelo. Para examinar la convergencia del proceso de estimación escribimos:

```
extract.mirt(m.1p, what="converged")
```

```
[1] TRUE
```

Si el resultado es `TRUE`, la estimación de parámetros con este modelo ha convergido, para examinar las estimaciones de parámetros escribimos:

```
coef(m.1p, simplify=T, IRTpars=T)
```

```
$items
          a      b g u
RV1  0.889 -2.999 0 1
RV2  0.889 -2.019 0 1
RV3  0.889 -1.714 0 1
RV4  0.889 -1.714 0 1
RV5  0.889 -0.748 0 1
RV6  0.889 -0.805 0 1
RV7  0.889 -0.805 0 1
RV8  0.889 -0.444 0 1
RV9  0.889 -0.692 0 1
RV10 0.889 -1.851 0 1
RV11 0.889 -1.260 0 1
RV12 0.889 -1.180 0 1
RV13 0.889 -1.831 0 1
RV14 0.889  0.125 0 1
RV15 0.889  0.350 0 1
RV16 0.889 -0.085 0 1
RV17 0.889  1.018 0 1
RV18 0.889  2.217 0 1

$means
F
0

$cov
  F
F 1
```

En esta salida se aprecia ahora que todos los ítem tienen el mismo parámetro de discriminación, pero es diferente de `1`, tal como ocurre en el modelo de Rasch. Además, los ítems se diferencian en el parámetro de dificultad (`b`), aunque el parámetro de pseudo-azar (`c`) permanece en `0`, supuesto que este modelo no admite que los ítems se puedan acertar por azar. Para obtener los errores típicos debemos emplear:

```
coef(m.1p, IRTpars=T, printSE=T)
```

```
$RV1
```

```
        a      b  g  u
par 0.889 -2.999  0  1
SE  0.048  0.256 NA NA

$RV2
        a      b  g  u
par 0.889 -2.019  0  1
SE  0.048  0.190 NA NA

$RV3
        a      b  g  u
par 0.889 -1.714  0  1
SE  0.048  0.174 NA NA

$RV4
        a      b  g  u
par 0.889 -1.714  0  1
SE  0.048  0.174 NA NA

$RV5
        a      b  g  u
par 0.889 -0.748  0  1
SE  0.048  0.140 NA NA

$RV6
        a      b  g  u
par 0.889 -0.805  0  1
SE  0.048  0.142 NA NA

$RV7
        a      b  g  u
par 0.889 -0.805  0  1
SE  0.048  0.142 NA NA

$RV8
        a      b  g  u
par 0.889 -0.444  0  1
SE  0.048  0.135 NA NA

$RV9
        a      b  g  u
par 0.889 -0.692  0  1
SE  0.048  0.139 NA NA

$RV10
        a      b  g  u
par 0.889 -1.851  0  1
SE  0.048  0.181 NA NA

$RV11
        a      b  g  u
```

```
par 0.889 -1.260  0  1
SE  0.048  0.155 NA NA

$RV12
        a      b  g  u
par 0.889 -1.180  0  1
SE  0.048  0.152 NA NA

$RV13
        a      b  g  u
par 0.889 -1.831  0  1
SE  0.048  0.180 NA NA

$RV14
        a     b  g  u
par 0.889 0.125  0  1
SE  0.048 0.132 NA NA

$RV15
        a     b  g  u
par 0.889 0.350  0  1
SE  0.048 0.133 NA NA

$RV16
        a      b  g  u
par 0.889 -0.085  0  1
SE  0.048  0.132 NA NA

$RV17
        a     b  g  u
par 0.889 1.018  0  1
SE  0.048 0.147 NA NA

$RV18
        a     b  g  u
par 0.889 2.217  0  1
SE  0.048 0.201 NA NA

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

y para obtener los intervalos de confianza de los parámetros en este test emplearemos:

```
coef(m.1p, IRTpars=T)
```

```
$RV1
            a      b  g  u
par     0.889 -2.999  0  1
CI_2.5  0.795 -3.500 NA NA
CI_97.5 0.982 -2.498 NA NA
```

```
$RV2
            a      b  g  u
par     0.889 -2.019  0  1
CI_2.5  0.795 -2.391 NA NA
CI_97.5 0.982 -1.647 NA NA

$RV3
            a      b  g  u
par     0.889 -1.714  0  1
CI_2.5  0.795 -2.055 NA NA
CI_97.5 0.982 -1.373 NA NA

$RV4
            a      b  g  u
par     0.889 -1.714  0  1
CI_2.5  0.795 -2.055 NA NA
CI_97.5 0.982 -1.373 NA NA

$RV5
            a      b  g  u
par     0.889 -0.748  0  1
CI_2.5  0.795 -1.023 NA NA
CI_97.5 0.982 -0.474 NA NA

$RV6
            a      b  g  u
par     0.889 -0.805  0  1
CI_2.5  0.795 -1.083 NA NA
CI_97.5 0.982 -0.528 NA NA

$RV7
            a      b  g  u
par     0.889 -0.805  0  1
CI_2.5  0.795 -1.083 NA NA
CI_97.5 0.982 -0.528 NA NA

$RV8
            a      b  g  u
par     0.889 -0.444  0  1
CI_2.5  0.795 -0.709 NA NA
CI_97.5 0.982 -0.180 NA NA

$RV9
            a      b  g  u
par     0.889 -0.692  0  1
CI_2.5  0.795 -0.965 NA NA
CI_97.5 0.982 -0.420 NA NA

$RV10
            a      b  g  u
par     0.889 -1.851  0  1
```

```
CI_2.5  0.795 -2.206 NA NA
CI_97.5 0.982 -1.497 NA NA

$RV11
            a      b  g  u
par     0.889 -1.260  0  1
CI_2.5  0.795 -1.564 NA NA
CI_97.5 0.982 -0.956 NA NA

$RV12
            a      b  g  u
par     0.889 -1.180  0  1
CI_2.5  0.795 -1.479 NA NA
CI_97.5 0.982 -0.881 NA NA

$RV13
            a      b  g  u
par     0.889 -1.831  0  1
CI_2.5  0.795 -2.184 NA NA
CI_97.5 0.982 -1.479 NA NA

$RV14
            a      b  g  u
par     0.889  0.125  0  1
CI_2.5  0.795 -0.133 NA NA
CI_97.5 0.982  0.383 NA NA

$RV15
            a     b  g  u
par     0.889 0.350  0  1
CI_2.5  0.795 0.089 NA NA
CI_97.5 0.982 0.611 NA NA

$RV16
            a      b  g  u
par     0.889 -0.085  0  1
CI_2.5  0.795 -0.343 NA NA
CI_97.5 0.982  0.173 NA NA

$RV17
            a     b  g  u
par     0.889 1.018  0  1
CI_2.5  0.795 0.731 NA NA
CI_97.5 0.982 1.306 NA NA

$RV18
            a     b  g  u
par     0.889 2.217  0  1
CI_2.5  0.795 1.824 NA NA
CI_97.5 0.982 2.610 NA NA
```

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

5.2 Evaluación del ajuste

Para evaluar el ajuste global del modelo emplearemos el estadístico:

```r
M2(m.1p)
```

```
            M2  df            p      RMSEA    RMSEA_5   RMSEA_95     SRMSR
stats 299.2061 152 1.124068e-11 0.04914383 0.04083908 0.05725349 0.0847147
            TLI       CFI
stats 0.8847726 0.8855258
```

El valor del estadístico `M2` ha resultado altamente significativo (`p < 0.05`), y además el resto de estadísticos de ajuste no superan los puntos de corte establecidos, por lo que no parece que este modelo pueda explicar adecuadamente las respuestas en este test. No obstante, podemos examinar el ajuste de los ítems con:

```r
itemfit(m.1p)
```

```
   item   S_X2 df.S_X2 RMSEA.S_X2 p.S_X2
1   RV1  8.678      11      0.000  0.652
2   RV2  6.617      11      0.000  0.829
3   RV3 11.078      10      0.016  0.351
4   RV4  8.449      10      0.000  0.585
5   RV5 10.725      11      0.000  0.467
6   RV6  9.672      11      0.000  0.560
7   RV7 12.795      11      0.020  0.307
8   RV8 24.957      11      0.056  0.009
9   RV9 12.053      11      0.015  0.360
10 RV10 15.607      11      0.032  0.156
11 RV11 20.503      10      0.051  0.025
12 RV12  6.416      11      0.000  0.844
13 RV13  7.009      10      0.000  0.725
14 RV14 27.579      10      0.066  0.002
15 RV15 37.726      10      0.083  0.000
16 RV16 12.518      10      0.025  0.252
17 RV17 44.158      10      0.092  0.000
18 RV18 77.171       8      0.147  0.000
```

donde se aprecia un mal ajuste del modelo en los ítems `RV8`, `RV11`, `RV14`, `RV15`, `RV17` y `RV18`.

5.3 Parámetros de habilidad

Para estimar los parámetros de habilidad debemos emplear la función:

```r
head(fscores(m.1p, method="EAP", full.scores=T, full.scores.SE=T))
```

```
             F      SE_F
[1,] 0.9317593 0.5844849
```

```
[2,] 0.3640400 0.5482002
[3,] 1.2476780 0.6087035
[4,] 0.6388036 0.5644815
[5,] 0.6388036 0.5644815
[6,] 1.5923812 0.6376938
```

que produce las estimaciones esperadas a posteriori (`EAP`) de los parámetros de habilidad en este grupo. También se puede comprobar si se han producido patrones de respuesta aberrantes con:

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

```
     outfit   z.outfit     infit    z.infit         Zh
1 0.5080217 -0.8740364 0.6114359 -1.1161927  1.0178490
2 0.6507767 -0.8115817 0.7501690 -0.8621346  0.8751039
3 0.3429377 -1.1139670 0.5190814 -1.2716390  1.1471691
4 0.5013912 -1.1038598 0.6550283 -1.1137326  1.0940525
5 1.8067798  1.5049430 1.5568053  1.5840377 -1.7864230
6 0.1712698 -1.3957725 0.2720913 -2.0480484  1.4155328
```

que produce los estadísticos de ajuste `infit`, `outfit`, sus normalizaciones `z-infit`, `z-outfit` y el estadístico l_z (`Zh`). Recuerde que el intervalo para determinar si el patrón es aberrante se sitúa entre `[0.5 - 1.5]`.

5.4 Independencia local de los ítems

Para evaluar la independencia local debemos emplear:

```
residuals(m.1p, type="Q3")
```

```
Q3 summary statistics:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -0.202  -0.080  -0.046  -0.041  -0.001   0.153

        RV1    RV2    RV3    RV4    RV5    RV6    RV7    RV8    RV9   RV10
RV1   1.000 -0.120  0.028 -0.042  0.027 -0.017 -0.096 -0.058  0.153  0.016
RV2  -0.120  1.000  0.005 -0.028 -0.005 -0.057  0.024 -0.027 -0.108  0.045
RV3   0.028  0.005  1.000 -0.033  0.050 -0.011 -0.013  0.031 -0.019  0.003
RV4  -0.042 -0.028 -0.033  1.000 -0.101  0.075 -0.050 -0.080 -0.057 -0.043
RV5   0.027 -0.005  0.050 -0.101  1.000  0.000 -0.054 -0.022 -0.020 -0.089
RV6  -0.017 -0.057 -0.011  0.075  0.000  1.000 -0.105 -0.001 -0.085  0.012
RV7  -0.096  0.024 -0.013 -0.050 -0.054 -0.105  1.000 -0.065  0.085  0.025
RV8  -0.058 -0.027  0.031 -0.080 -0.022 -0.001 -0.065  1.000 -0.081 -0.078
RV9   0.153 -0.108 -0.019 -0.057 -0.020 -0.085  0.085 -0.081  1.000 -0.040
RV10  0.016  0.045  0.003 -0.043 -0.089  0.012  0.025 -0.078 -0.040  1.000
RV11 -0.046  0.015  0.046 -0.040  0.068  0.048 -0.120 -0.031  0.106  0.040
RV12 -0.014 -0.035 -0.066  0.004  0.003 -0.070 -0.068 -0.018 -0.078 -0.090
RV13  0.031 -0.034 -0.019 -0.009 -0.087 -0.034 -0.101  0.032  0.045  0.008
RV14 -0.075 -0.019 -0.069 -0.058 -0.055 -0.042 -0.078 -0.017 -0.141 -0.103
RV15  0.021 -0.021 -0.106 -0.081 -0.056 -0.153 -0.004 -0.152 -0.081 -0.035
RV16 -0.009 -0.036  0.066 -0.107 -0.066 -0.107  0.015 -0.030  0.130 -0.001
RV17 -0.086 -0.033 -0.125 -0.050 -0.060 -0.016 -0.134 -0.084 -0.143 -0.076
RV18  0.017 -0.071 -0.121 -0.032 -0.043 -0.104 -0.080 -0.080 -0.073 -0.155
       RV11   RV12   RV13   RV14   RV15   RV16   RV17   RV18
```

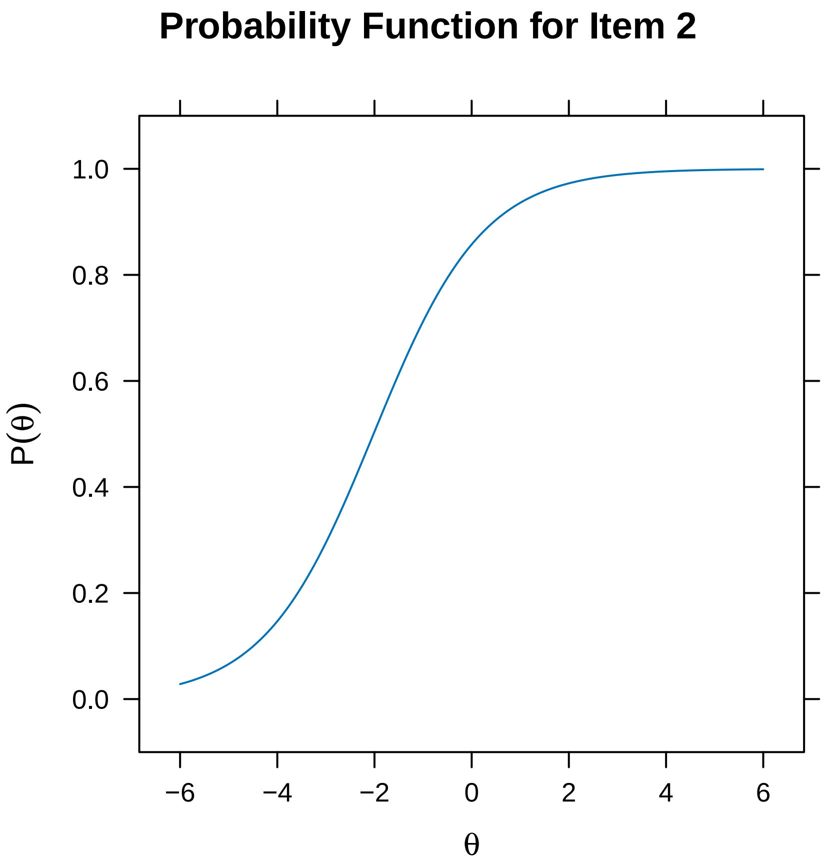
```
RV1   -0.046 -0.014  0.031 -0.075  0.021 -0.009 -0.086  0.017
RV2    0.015 -0.035 -0.034 -0.019 -0.021 -0.036 -0.033 -0.071
RV3    0.046 -0.066 -0.019 -0.069 -0.106  0.066 -0.125 -0.121
RV4   -0.040  0.004 -0.009 -0.058 -0.081 -0.107 -0.050 -0.032
RV5    0.068  0.003 -0.087 -0.055 -0.056 -0.066 -0.060 -0.043
RV6    0.048 -0.070 -0.034 -0.042 -0.153 -0.107 -0.016 -0.104
RV7   -0.120 -0.068 -0.101 -0.078 -0.004  0.015 -0.134 -0.080
RV8   -0.031 -0.018  0.032 -0.017 -0.152 -0.030 -0.084 -0.080
RV9    0.106 -0.078  0.045 -0.141 -0.081  0.130 -0.143 -0.073
RV10   0.040 -0.090  0.008 -0.103 -0.035 -0.001 -0.076 -0.155
RV11   1.000 -0.080  0.123 -0.107 -0.127  0.078 -0.151 -0.047
RV12  -0.080  1.000  0.035 -0.019 -0.094 -0.115 -0.080  0.003
RV13   0.123  0.035  1.000 -0.151 -0.053 -0.036 -0.079 -0.071
RV14  -0.107 -0.019 -0.151  1.000 -0.105 -0.050 -0.058 -0.097
RV15  -0.127 -0.094 -0.053 -0.105  1.000 -0.202 -0.049  0.058
RV16   0.078 -0.115 -0.036 -0.050 -0.202  1.000 -0.058 -0.080
RV17  -0.151 -0.080 -0.079 -0.058 -0.049 -0.058  1.000 -0.046
RV18  -0.047  0.003 -0.071 -0.097  0.058 -0.080 -0.046  1.000
```

que no detecta problemas de que se viole la indepedencia local de los ítems.

5.5 Gráficos

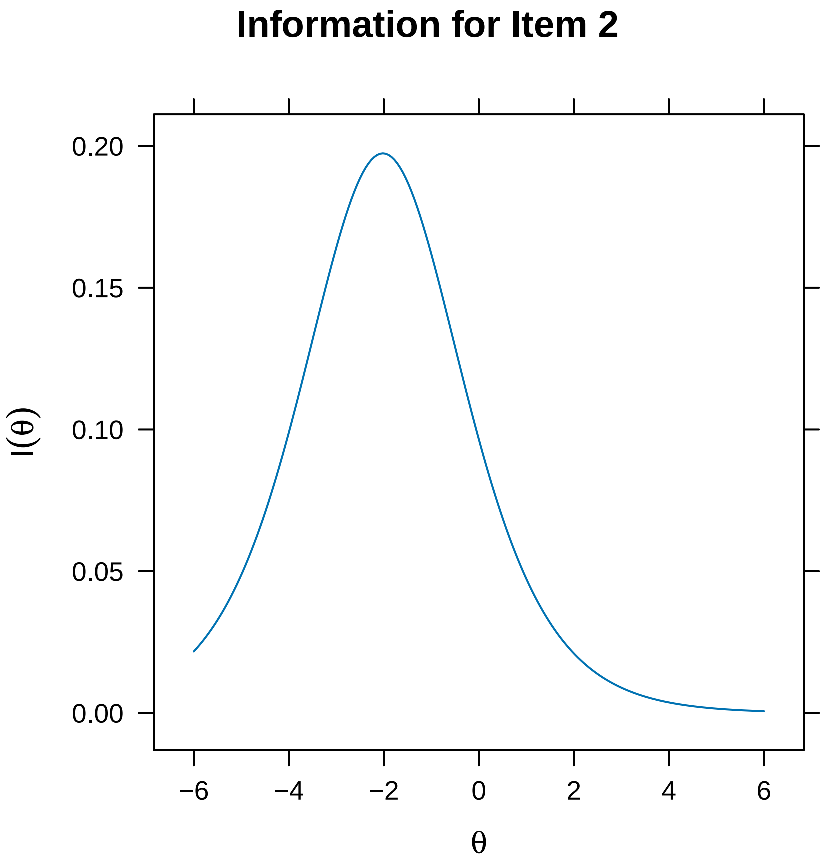
En este modelo, la `FRI` de un ítem (e.g., ítem 2) se obtiene con:

```
itemplot(m.1p, 2)
```



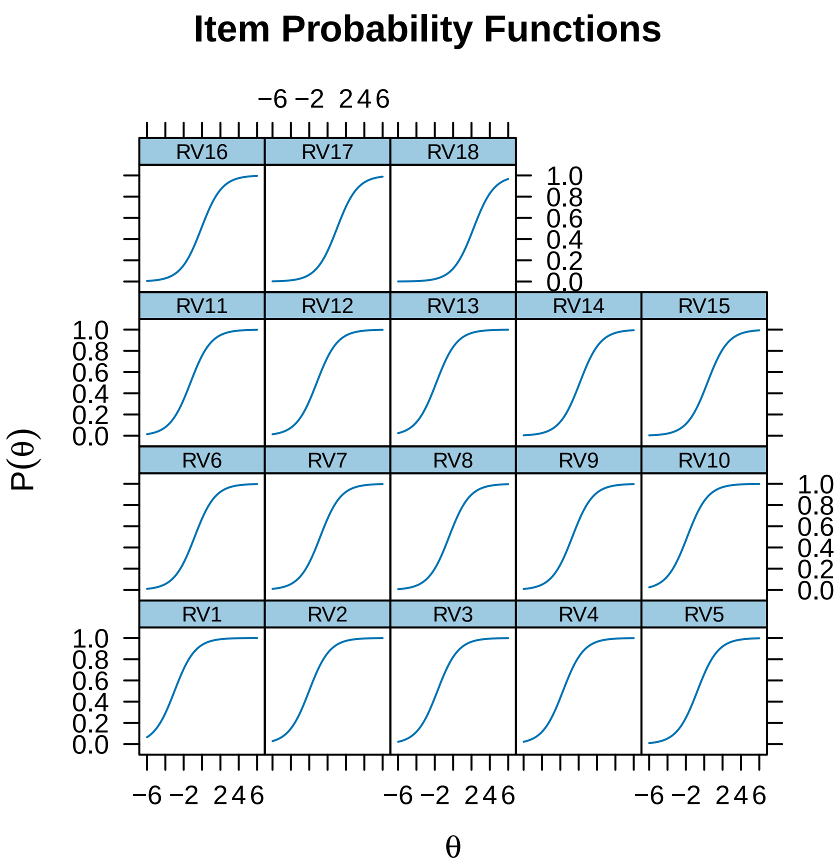
y la `FII` se obtiene con:

```
itemplot(m.1p, item=2, type="info")
```



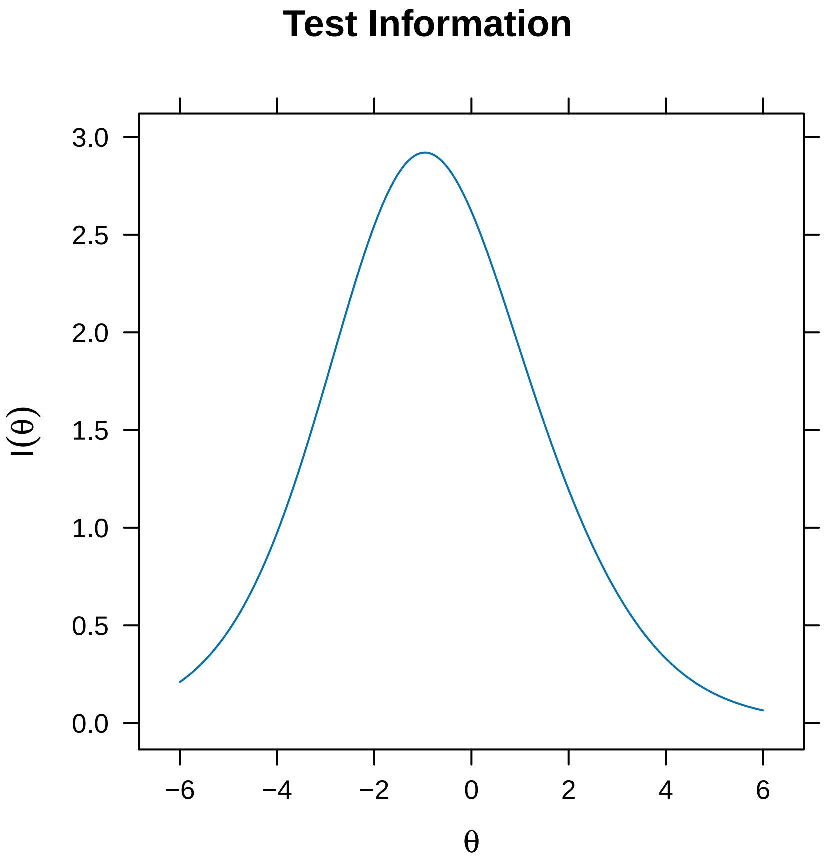
Las `FRIs` de todos los ítems se obtienen con:

```
plot(m.1p, type="trace")
```



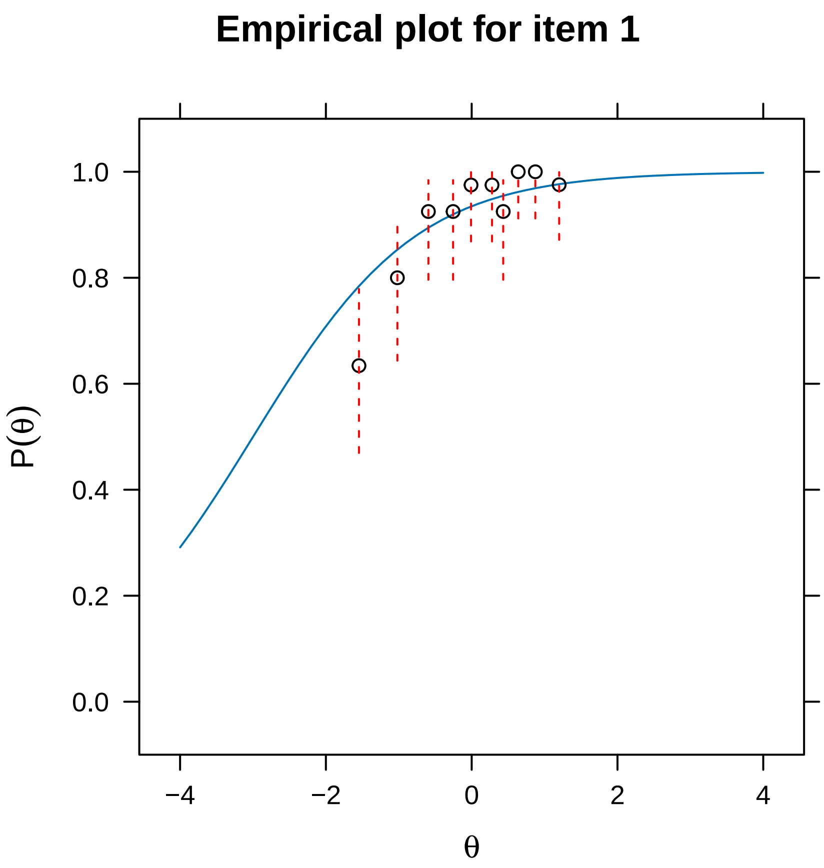
y la FIT para el test completo con:

```
plot(m.1p, type="info")
```



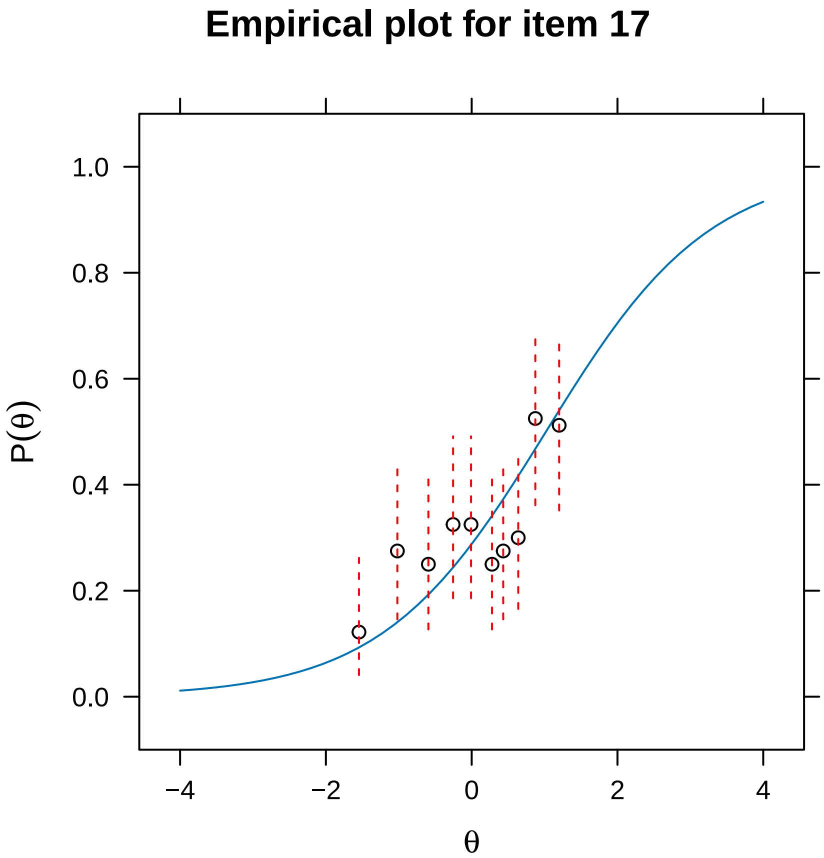
El paquete `mirt` también permite graficar la `FRI` empírica frente a la `FRI` teórica en este modelo con:

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



que en este caso muestra un buen ajuste entre ambas curvas. No ocurre igual en el ítem `RV17` como se aprecia en la siguiente figura:

```
itemfit(m.1p, empirical.plot=17, empirical.CI=.95)
```


Empirical plot for item 17
P(θ)
θ
1.0
0.8
0.6
0.4
0.2
0.0
−4
−2
0
2
4

Capítulo 6

Modelo logístico de 2-p

6.1 Estimación de parámetros

En esta sección se presenta la estimación de parámetros con el modelo logístico de `2-p`, los estadísticos de ajuste y los gráficos que lo diferencian de otros modelos. Esta presentación será más sucinta, dado que el resto de procesos se presentaron en los modelos anteriores.

Para estimar los parámetros en el modelo logístico de `2-p` emplearemos:

```
m.2p <- mirt(rv, model=1, itemtype="2PL", SE=T, verbose=F)
```

Para examinar las estimaciones escribiremos:

```
coef(m.2p, simplify=T, IRTpars=T)
```

```
$items
         a      b g u
RV1  1.331 -2.253 0 1
RV2  1.095 -1.737 0 1
RV3  1.503 -1.224 0 1
RV4  0.773 -1.910 0 1
RV5  1.231 -0.604 0 1
RV6  0.986 -0.752 0 1
RV7  0.879 -0.816 0 1
RV8  0.945 -0.430 0 1
RV9  1.473 -0.504 0 1
RV10 1.063 -1.627 0 1
RV11 1.871 -0.816 0 1
RV12 0.777 -1.312 0 1
RV13 1.319 -1.403 0 1
RV14 0.409  0.228 0 1
RV15 0.294  0.903 0 1
RV16 1.250 -0.071 0 1
RV17 0.171  4.539 0 1
RV18 0.111 15.553 0 1

$means
F1
 0
```

```
$cov
   F1
F1  1
```

En este caso, se aprecia que los ítems difieren ampliamente en función del parámetro de discriminación. Los ítems `RV3` y `RV11` son los más discriminativos, mientras que los ítems `RV17` y `RV18` son los menos discriminativos. Además, el ítem `RV18` ha obtenido un parámetro de dificultad muy extremo, lo que puede suponer un problema en la estimación de parámetros. No obstante, se puede examinar la convergencia con:

```
extract.mirt(m.2p, what="converged")
```

```
[1] TRUE
```

que devuelve `TRUE`.

A continuación se presentan los ítems ordenados en función del parámetro de dificultad:

```
coeficientes <- coef(m.2p,simplify=T,IRTpars=T)
coeficientes_items <- coeficientes$items
a <- coeficientes_items[, "a"]
b <- coeficientes_items[, "b"]
tabla <- data.frame(
Item=rownames(coeficientes$items),
Discriminacion_a = a,
Dificultad_b = b)
tabla_ordenada_b <- tabla[order(tabla$Dificultad_b), ]
print(tabla_ordenada_b)
```

```
     Item Discriminacion_a Dificultad_b
RV1   RV1        1.3308774  -2.25287893
RV4   RV4        0.7727555  -1.91032190
RV2   RV2        1.0950858  -1.73711359
RV10 RV10        1.0628400  -1.62724565
RV13 RV13        1.3191082  -1.40334912
RV12 RV12        0.7768873  -1.31179466
RV3   RV3        1.5025149  -1.22380872
RV7   RV7        0.8792498  -0.81646229
RV11 RV11        1.8710054  -0.81618536
RV6   RV6        0.9855897  -0.75172017
RV5   RV5        1.2310204  -0.60388494
RV9   RV9        1.4725667  -0.50419855
RV8   RV8        0.9447656  -0.43014397
RV16 RV16        1.2499060  -0.07075478
RV14 RV14        0.4093764   0.22767673
RV15 RV15        0.2940950   0.90312474
RV17 RV17        0.1713836   4.53902933
RV18 RV18        0.1109293  15.55263211
```

y en función del parámetro de discriminación:

```
tabla_ordenada_a <- tabla[order(tabla$Discriminacion_a), ]
print(tabla_ordenada_a)
```

```
     Item Discriminacion_a Dificultad_b
RV18 RV18        0.1109293  15.55263211
```

```
RV17 RV17        0.1713836   4.53902933
RV15 RV15        0.2940950   0.90312474
RV14 RV14        0.4093764   0.22767673
RV4   RV4        0.7727555  -1.91032190
RV12 RV12        0.7768873  -1.31179466
RV7   RV7        0.8792498  -0.81646229
RV8   RV8        0.9447656  -0.43014397
RV6   RV6        0.9855897  -0.75172017
RV10 RV10        1.0628400  -1.62724565
RV2   RV2        1.0950858  -1.73711359
RV5   RV5        1.2310204  -0.60388494
RV16 RV16        1.2499060  -0.07075478
RV13 RV13        1.3191082  -1.40334912
RV1   RV1        1.3308774  -2.25287893
RV9   RV9        1.4725667  -0.50419855
RV3   RV3        1.5025149  -1.22380872
RV11 RV11        1.8710054  -0.81618536
```

En este caso, los estadísticos básicos de estos parámetros se obtienen con:

```r
mean(a)
```

```
[1] 0.9711087
```

```r
sd(a)
```

```
[1] 0.4852554
```

```r
min(a)
```

```
[1] 0.1109293
```

```r
max(a)
```

```
[1] 1.871005
```

y para los parámetros de dificultad son:

```r
mean(b)
```

```
[1] 0.3201445
```

```r
sd(b)
```

```
[1] 4.085806
```

```r
min(b)
```

```
[1] -2.252879
```

```r
max(b)
```

```
[1] 15.55263
```

6.2 Estadísticos de ajuste

Para examinar el ajuste del modelo empleamos:

```r
M2(m.2p)
```

```
            M2  df         p RMSEA RMSEA_5   RMSEA_95     SRMSR     TLI CFI
stats 131.3573 135 0.5726145     0       0 0.02229789 0.0410899 1.00321   1
```

Dado que el valor de `p > 0.05` y el resto de estadísticos de ajuste cumplen los criterios especificados, parece que el modelo logístico de `2-p` podría ser adecuado para explicar las respuestas a los 18 ítems de este test. Si examinamos el ajuste de los ítems:

```r
itemfit(m.2p)
```

```
   item   S_X2 df.S_X2 RMSEA.S_X2 p.S_X2
1   RV1  5.937      10      0.000  0.821
2   RV2  4.989      11      0.000  0.932
3   RV3  5.592      10      0.000  0.848
4   RV4  8.190      11      0.000  0.696
5   RV5  8.079       9      0.000  0.526
6   RV6 11.163      10      0.017  0.345
7   RV7 13.732      11      0.025  0.248
8   RV8 24.463      11      0.055  0.011
9   RV9  7.586       9      0.000  0.576
10 RV10 15.504      10      0.037  0.115
11 RV11 10.669       9      0.022  0.299
12 RV12  6.040      10      0.000  0.812
13 RV13  4.695      10      0.000  0.911
14 RV14 16.747      11      0.036  0.116
15 RV15  7.171      11      0.000  0.785
16 RV16 11.121      10      0.017  0.348
17 RV17 18.804      12      0.038  0.093
18 RV18 23.433      11      0.053  0.015
```

se aprecia que el modelo logístico de `2-p` proporciona un buen ajuste de la mayor parte de los ítems, excepto los ítems `RV8` y `RV18`.

6.3 Independencia local de los ítems

Para evaluar la independencia local se puede emplear el estadístico `Q3` de Yen:

```r
residuals(m.2p, type="Q3")
```

```
Q3 summary statistics:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 -0.194  -0.073  -0.038  -0.039  -0.002   0.101 

        RV1    RV2    RV3    RV4    RV5    RV6    RV7    RV8    RV9   RV10
RV1   1.000 -0.146 -0.045 -0.037 -0.013 -0.038 -0.102 -0.074  0.101 -0.015
RV2  -0.146  1.000 -0.047 -0.010 -0.029 -0.061  0.033 -0.028 -0.173  0.032
RV3  -0.045 -0.047  1.000 -0.047 -0.034 -0.066 -0.041 -0.010 -0.161 -0.060
RV4  -0.037 -0.010 -0.047  1.000 -0.092  0.094 -0.025 -0.062 -0.072 -0.033
RV5  -0.013 -0.029 -0.034 -0.092  1.000 -0.025 -0.064 -0.044 -0.124 -0.126
RV6  -0.038 -0.061 -0.066  0.094 -0.025  1.000 -0.096 -0.002 -0.154 -0.003
RV7  -0.102  0.033 -0.041 -0.025 -0.064 -0.096  1.000 -0.057  0.051  0.023
RV8  -0.074 -0.028 -0.010 -0.062 -0.044 -0.002 -0.057  1.000 -0.144 -0.092
```

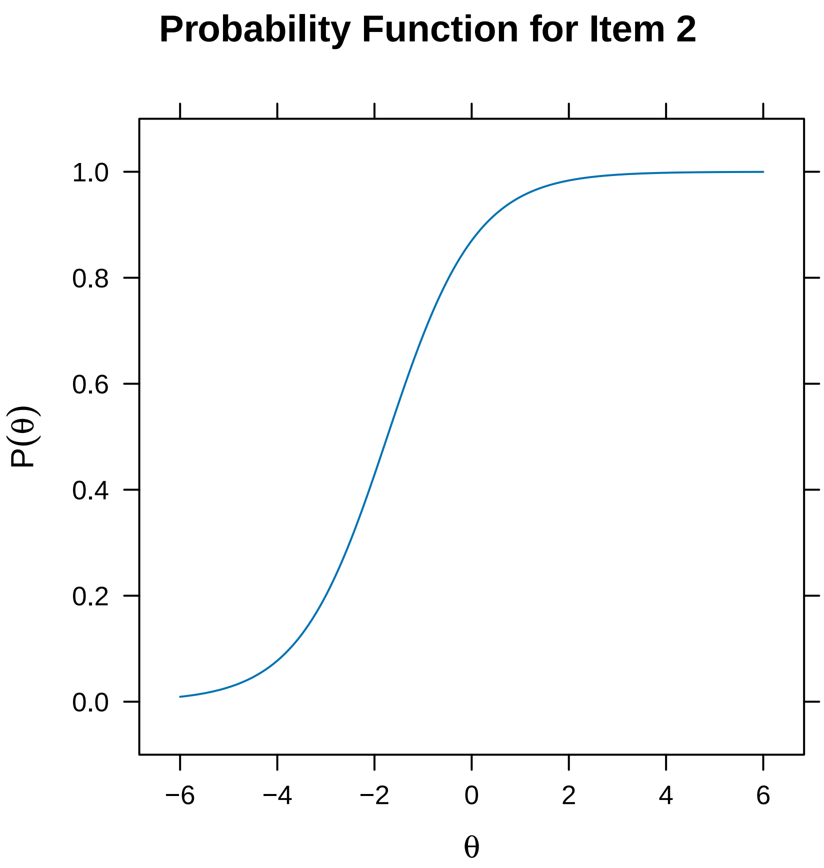
```
RV9   0.101 -0.173 -0.161 -0.072 -0.124 -0.154  0.051 -0.144  1.000 -0.111
RV10 -0.015  0.032 -0.060 -0.033 -0.126 -0.003  0.023 -0.092 -0.111  1.000
RV11 -0.167 -0.073 -0.145 -0.074 -0.066 -0.043 -0.194 -0.115 -0.087 -0.056
RV12 -0.009 -0.013 -0.078  0.036  0.016 -0.046 -0.040  0.003 -0.094 -0.078
RV13 -0.022 -0.070 -0.120 -0.012 -0.158 -0.072 -0.120  0.004 -0.057 -0.038
RV14 -0.041  0.034 -0.024 -0.007  0.007  0.018 -0.020  0.043 -0.093 -0.061
RV15  0.075  0.047 -0.030 -0.026  0.026 -0.077  0.061 -0.075 -0.004  0.020
RV16 -0.051 -0.073 -0.018 -0.115 -0.147 -0.155 -0.013 -0.073  0.021 -0.047
RV17 -0.040  0.037 -0.048  0.007  0.035  0.063 -0.059  0.002 -0.049 -0.019
RV18  0.060 -0.015 -0.050  0.007  0.032 -0.043 -0.025 -0.014  0.007 -0.109
       RV11   RV12   RV13   RV14   RV15   RV16   RV17   RV18
RV1  -0.167 -0.009 -0.022 -0.041  0.075 -0.051 -0.040  0.060
RV2  -0.073 -0.013 -0.070  0.034  0.047 -0.073  0.037 -0.015
RV3  -0.145 -0.078 -0.120 -0.024 -0.030 -0.018 -0.048 -0.050
RV4  -0.074  0.036 -0.012 -0.007 -0.026 -0.115  0.007  0.007
RV5  -0.066  0.016 -0.158  0.007  0.026 -0.147  0.035  0.032
RV6  -0.043 -0.046 -0.072  0.018 -0.077 -0.155  0.063 -0.043
RV7  -0.194 -0.040 -0.120 -0.020  0.061 -0.013 -0.059 -0.025
RV8  -0.115  0.003  0.004  0.043 -0.075 -0.073  0.002 -0.014
RV9  -0.087 -0.094 -0.057 -0.093 -0.004  0.021 -0.049  0.007
RV10 -0.056 -0.078 -0.038 -0.061  0.020 -0.047 -0.019 -0.109
RV11  1.000 -0.116 -0.012 -0.067 -0.048 -0.055 -0.060  0.049
RV12 -0.116  1.000  0.035  0.036 -0.030 -0.124 -0.014  0.047
RV13 -0.012  0.035  1.000 -0.106  0.016 -0.111 -0.007 -0.008
RV14 -0.067  0.036 -0.106  1.000 -0.044  0.003 -0.005 -0.055
RV15 -0.048 -0.030  0.016 -0.044  1.000 -0.132  0.004  0.098
RV16 -0.055 -0.124 -0.111  0.003 -0.132  1.000  0.039 -0.006
RV17 -0.060 -0.014 -0.007 -0.005  0.004  0.039  1.000 -0.008
RV18  0.049  0.047 -0.008 -0.055  0.098 -0.006 -0.008  1.000
```

que no detecta problemas de que se viole la indepedencia local de los ítems.

6.4 Gráficos

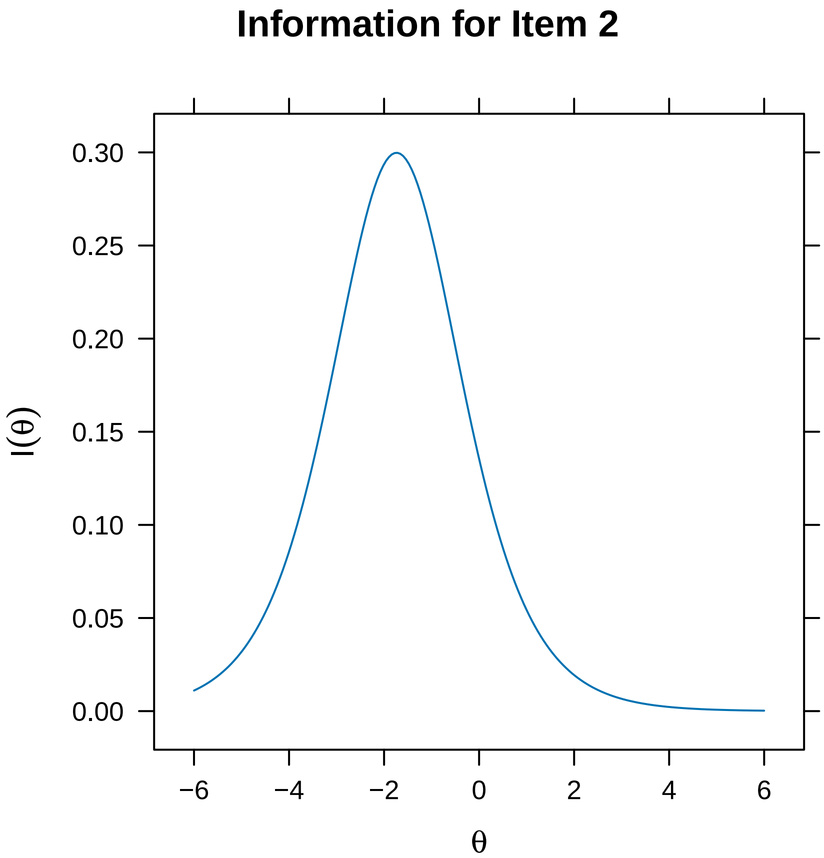
En este modelo, la `FRI` de un ítem (e.g., ítem 2) se obtiene con:

```r
itemplot(m.2p, 2)
```



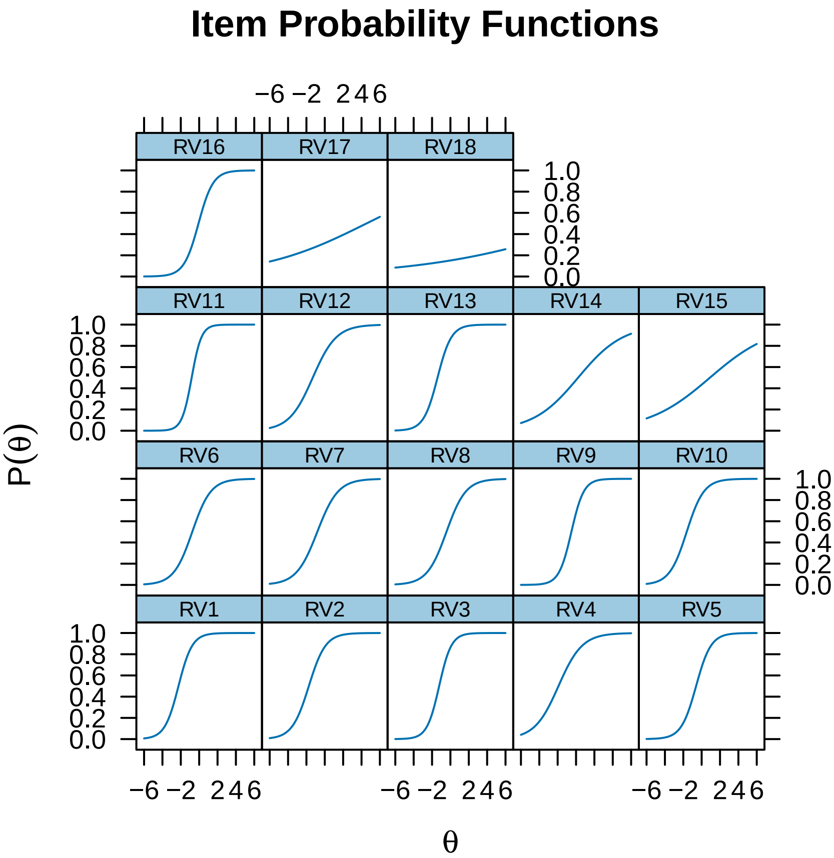
y la `FII` se obtiene con:

```
itemplot(m.2p, item=2, type="info")
```



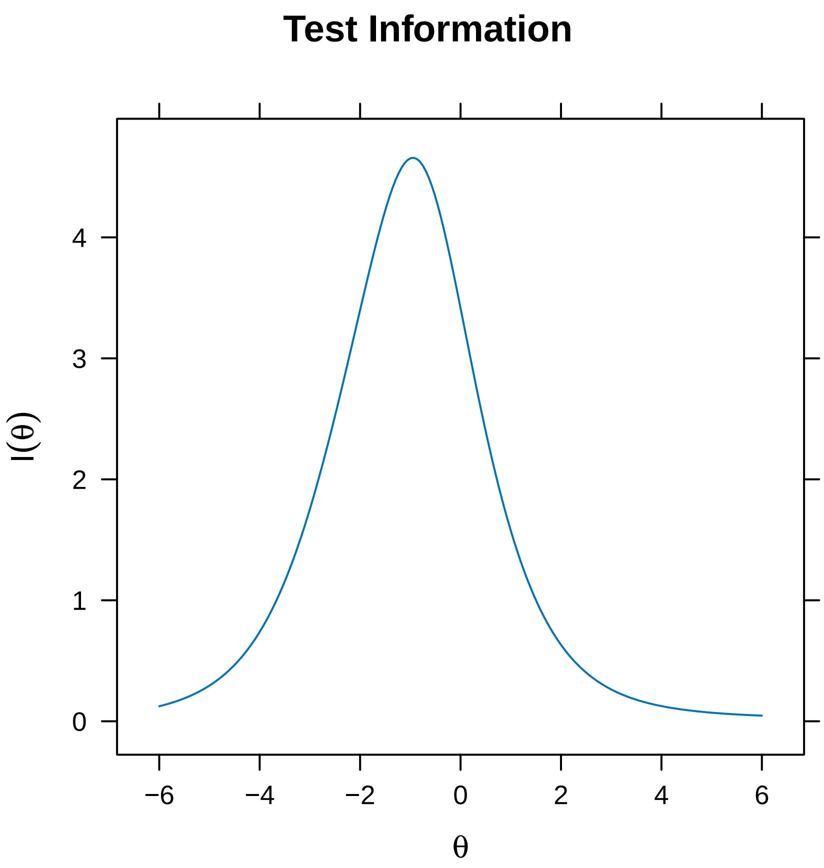
Las `FRIs` de todos los ítems se obtienen con:

```
plot(m.2p, type="trace")
```



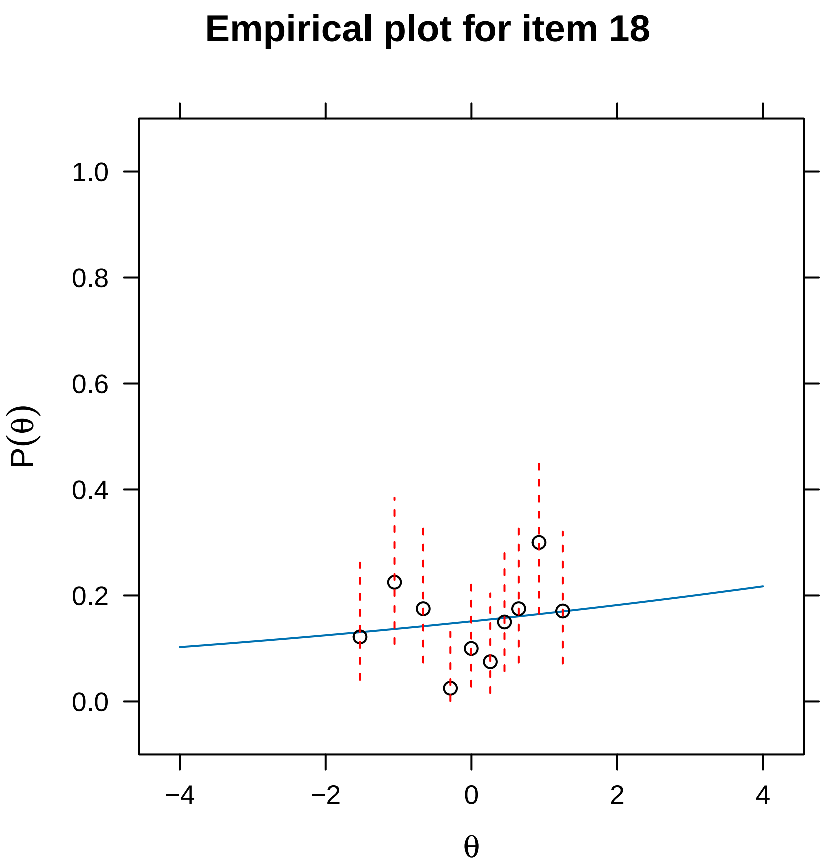
y la `FIT` para el test completo con:

```
plot(m.2p, type="info")
```



Para examinar el mal ajuste del modelo a las respuestas del ítem `RV18` se pueden graficar la curva empírica vs. la curva teórica con la función:

```
itemfit(m.2p, empirical.plot=18, empirical.CI=.95)
```



En el gráfico se aprecia que las `FRIs` son prácticamente paralelas al eje de abscisas, revelando que este ítem ha resultado muy difícil en el grupo con una muy baja capacidad discriminativa, dado que la probabilidad de acertarlo ha permanecido casi constante a través de todo el continuo de habilidad.

Capítulo 7

Modelo de 3-p

7.1 Estimacción de parámetros

Es posible que algunos de los ítems de este test hayan podido ser acertados por azar. Por tanto, se pueden estimar los parámetros de los ítems con el modelo de `3-p`. Dada la dificultad de la estimación del parámetro de pseudo-azar, el paquete `mirt` permite especificar una distribución previa para mejorar la estimación de este parámetro. En este caso, supuesto que los ítems del test tienen cuatro opciones, una estimación previa de este parámetro sería `1/4=0.25`, que se traslada al paquete con el argumento `guess`. No obstante, conviene advertir que el paquete `mirt` permite definir distribuciones previas de los parámetros de discriminación y pseudo-azar que, en ciertas condiciones, puede mejorar la estimación de parámetros.

La estimación de parámetros con este modelo se obtiene con:

```
m.3p <- mirt(rv, model=1, itemtype="3PL", guess= 0.25, SE=T, verbose=F)
```

y para examinar las estimaciones de los parámetros escribiremos:

```
coef(m.3p, simplify=T, IRTpars=T)
```

```
$items
          a      b     g u
RV1   1.423 -1.818 0.307 1
RV2   1.615 -0.713 0.437 1
RV3   3.086 -0.347 0.433 1
RV4   0.789 -1.845 0.017 1
RV5   1.229 -0.593 0.000 1
RV6   1.001 -0.734 0.001 1
RV7   0.903 -0.788 0.003 1
RV8   1.185 -0.025 0.155 1
RV9   1.472 -0.489 0.001 1
RV10  1.085 -1.579 0.015 1
RV11  2.427 -0.504 0.177 1
RV12  0.790 -1.285 0.003 1
RV13  2.350 -0.432 0.451 1
RV14  0.418  0.231 0.001 1
RV15  0.908  1.889 0.307 1
RV16  1.639  0.215 0.130 1
RV17  0.173  4.618 0.007 1
RV18  2.832  2.422 0.136 1
```

```
$means
F1
 0

$cov
   F1
F1  1
```

En este caso, se aprecia en la columna `g` que algunos ítems `RV2`, `RV3`, `RV13` y `RV15` han obtenido parámetros de pseudo-azar muy superiores a los esperados (`0.25`). Podemos obtener los errores típicos de los parámetros con:

```
coef(m.3p, IRTpars=T, printSE=T)
```

```
$RV1
        a      b     g  u
par 1.423 -1.818 0.307  1
SE  0.665  1.715 0.873 NA

$RV2
        a      b     g  u
par 1.615 -0.713 0.437  1
SE  0.690  0.684 0.235 NA

$RV3
        a      b     g  u
par 3.086 -0.347 0.433  1
SE  1.257  0.224 0.095 NA

$RV4
        a      b     g  u
par 0.789 -1.845 0.017  1
SE  0.195  0.859 0.326 NA

$RV5
        a      b     g  u
par 1.229 -0.593 0.000  1
SE  0.188  0.126 0.009 NA

$RV6
        a      b     g  u
par 1.001 -0.734 0.001  1
SE  0.166  0.156 0.011 NA

$RV7
        a      b     g  u
par 0.903 -0.788 0.003  1
SE  0.162  0.212 0.051 NA

$RV8
        a      b     g  u
par 1.185 -0.025 0.155  1
```

```
SE  0.409  0.422 0.159 NA

$RV9
        a      b     g  u
par 1.472 -0.489 0.001  1
SE  0.219  0.113 0.019 NA

$RV10
        a      b     g  u
par 1.085 -1.579 0.015  1
SE  0.242  0.601 0.296 NA

$RV11
        a      b     g  u
par 2.427 -0.504 0.177  1
SE  0.715  0.235 0.127 NA

$RV12
        a      b     g  u
par 0.790 -1.285 0.003  1
SE  0.156  0.293 0.057 NA

$RV13
        a      b     g  u
par 2.350 -0.432 0.451  1
SE  0.914  0.330 0.126 NA

$RV14
        a     b     g  u
par 0.418 0.231 0.001  1
SE  0.126 0.285 0.025 NA

$RV15
        a     b     g  u
par 0.908 1.889 0.307  1
SE  0.810 0.617 0.119 NA

$RV16
        a     b     g  u
par 1.639 0.215 0.130  1
SE  0.567 0.222 0.099 NA

$RV17
        a     b     g  u
par 0.173 4.618 0.007  1
SE  0.140 3.772 0.158 NA

$RV18
        a     b     g  u
par 2.832 2.422 0.136  1
SE  3.292 0.749 0.023 NA
```

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

y los intervalos de confianza correspondientes con:

```r
coef(m.3p, IRTpars=T)
```

```
$RV1
             a      b      g  u
par     1.423 -1.818  0.307  1
CI_2.5  0.120 -5.179 -1.405 NA
CI_97.5 2.725  1.542  2.019 NA

$RV2
             a      b      g  u
par     1.615 -0.713  0.437  1
CI_2.5  0.262 -2.054 -0.024 NA
CI_97.5 2.968  0.627  0.898 NA

$RV3
             a      b     g  u
par     3.086 -0.347 0.433  1
CI_2.5  0.623 -0.786 0.248 NA
CI_97.5 5.548  0.093 0.619 NA

$RV4
             a      b      g  u
par     0.789 -1.845  0.017  1
CI_2.5  0.407 -3.528 -0.622 NA
CI_97.5 1.171 -0.162  0.656 NA

$RV5
             a      b      g  u
par     1.229 -0.593  0.000  1
CI_2.5  0.860 -0.839 -0.017 NA
CI_97.5 1.598 -0.346  0.018 NA

$RV6
             a      b      g  u
par     1.001 -0.734  0.001  1
CI_2.5  0.676 -1.039 -0.021 NA
CI_97.5 1.327 -0.429  0.022 NA

$RV7
             a      b      g  u
par     0.903 -0.788  0.003  1
CI_2.5  0.586 -1.205 -0.098 NA
CI_97.5 1.220 -0.372  0.103 NA

$RV8
```

```
            a      b      g  u
par     1.185 -0.025  0.155  1
CI_2.5  0.385 -0.852 -0.157 NA
CI_97.5 1.986  0.802  0.467 NA

$RV9
            a      b      g  u
par     1.472 -0.489  0.001  1
CI_2.5  1.043 -0.710 -0.037 NA
CI_97.5 1.900 -0.268  0.039 NA

$RV10
            a      b      g  u
par     1.085 -1.579  0.015  1
CI_2.5  0.612 -2.758 -0.566 NA
CI_97.5 1.559 -0.401  0.595 NA

$RV11
            a      b      g  u
par     2.427 -0.504  0.177  1
CI_2.5  1.026 -0.965 -0.072 NA
CI_97.5 3.828 -0.043  0.426 NA

$RV12
            a      b      g  u
par     0.790 -1.285  0.003  1
CI_2.5  0.484 -1.860 -0.110 NA
CI_97.5 1.096 -0.710  0.115 NA

$RV13
            a      b     g  u
par     2.350 -0.432 0.451  1
CI_2.5  0.558 -1.080 0.204 NA
CI_97.5 4.142  0.215 0.698 NA

$RV14
            a      b      g  u
par     0.418  0.231  0.001  1
CI_2.5  0.171 -0.327 -0.047 NA
CI_97.5 0.666  0.789  0.049 NA

$RV15
             a     b     g  u
par      0.908 1.889 0.307  1
CI_2.5  -0.679 0.679 0.074 NA
CI_97.5  2.496 3.099 0.541 NA

$RV16
            a      b      g  u
par     1.639  0.215  0.130  1
CI_2.5  0.527 -0.220 -0.064 NA
```

```
CI_97.5 2.750  0.649  0.325 NA

$RV17
              a      b      g  u
par      0.173  4.618  0.007  1
CI_2.5  -0.101 -2.776 -0.303 NA
CI_97.5  0.448 12.011  0.317 NA

$RV18
              a     b     g  u
par      2.832 2.422 0.136  1
CI_2.5  -3.621 0.954 0.090 NA
CI_97.5  9.285 3.889 0.181 NA

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

7.2 Evaluación del ajuste

En cualquier caso, se puede examinar el ajuste global del modelo de `3-p` con:

```
M2(m.3p)
```

```
            M2  df         p RMSEA RMSEA_5   RMSEA_95      SRMSR      TLI CFI
stats 114.5006 117 0.5481115     0       0 0.02371488 0.04145487 1.002542   1
```

Este resultado muestra que el modelo de `3-p` también podría servir para explicar las respuestas del grupo en este test. Además, se puede estudiar el ajuste por ítems.

```
itemfit(m.3p)
```

```
   item   S_X2 df.S_X2 RMSEA.S_X2 p.S_X2
1   RV1  5.874       9      0.000  0.752
2   RV2  4.008      10      0.000  0.947
3   RV3  3.094       8      0.000  0.928
4   RV4  8.320      10      0.000  0.598
5   RV5  7.292       8      0.000  0.506
6   RV6 10.920       9      0.023  0.281
7   RV7 13.680      10      0.030  0.188
8   RV8 24.402      10      0.060  0.007
9   RV9  6.987       8      0.000  0.538
10 RV10 15.333       9      0.042  0.082
11 RV11 11.899       9      0.028  0.219
12 RV12  5.248       9      0.000  0.812
13 RV13  2.171       8      0.000  0.975
14 RV14 14.552      10      0.034  0.149
15 RV15  7.650      10      0.000  0.663
16 RV16  9.861       9      0.015  0.362
17 RV17 16.761      11      0.036  0.115
18 RV18 21.762      10      0.054  0.016
```

En este caso, parece que los ítems `RV8` y `RV18` no parecen seguir este modelo.

7.3 Independencia local de los ítems

Para evaluar la independencia local de los ítems se puede emplear el estadístico `LD`:

```r
residuals(m.3p, type="LD")
```

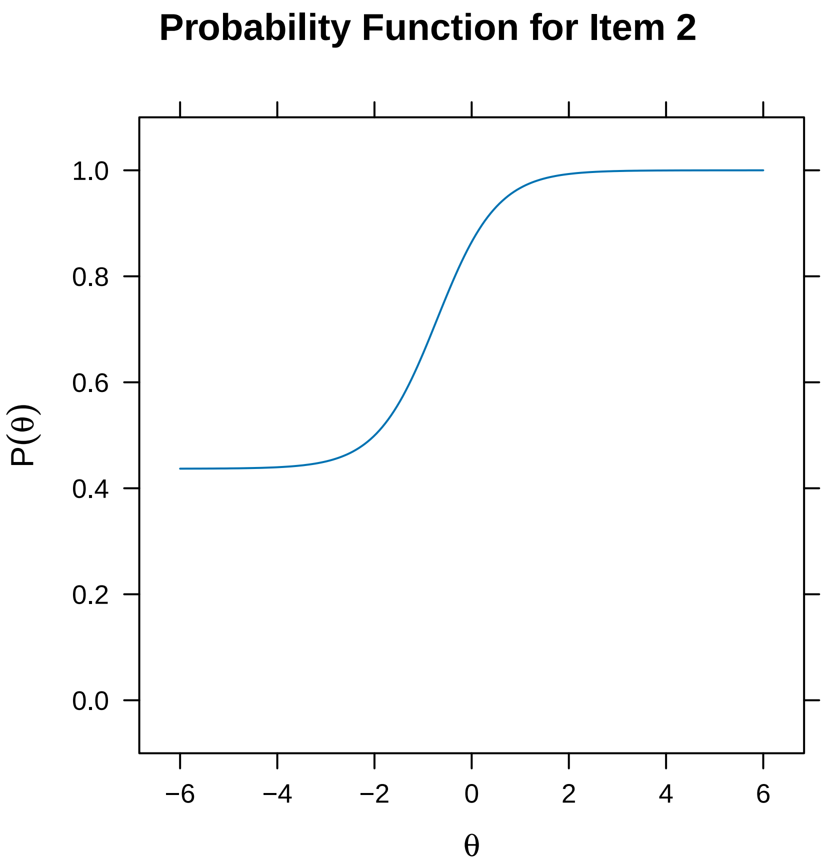
```
LD matrix (lower triangle) and standardized residual correlations (upper triangle)

Upper triangle summary:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 -0.115  -0.027   0.006   0.003   0.035   0.123 

       RV1    RV2   RV3    RV4    RV5    RV6    RV7    RV8    RV9   RV10   RV11
RV1        -0.068 0.031 -0.005  0.036  0.010 -0.053 -0.025  0.123  0.022 -0.043
RV2  1.860         0.018  0.019  0.026 -0.008  0.057  0.014 -0.073  0.056  0.015
RV3  0.395  0.129        0.015  0.046  0.018  0.020  0.044 -0.036  0.018 -0.015
RV4  0.010  0.138 0.091        -0.034  0.108  0.006 -0.021 -0.013  0.000  0.010
RV5  0.531  0.266 0.856  0.453         0.039 -0.005  0.022 -0.012 -0.047  0.043
RV6  0.037  0.029 0.129  4.663  0.602        -0.037  0.045 -0.044  0.039  0.049
RV7  1.129  1.319 0.160  0.012  0.008  0.544        -0.008  0.091  0.050 -0.072
RV8  0.253  0.082 0.783  0.178  0.203  0.816  0.026        -0.041 -0.035 -0.015
RV9  6.071  2.166 0.520  0.065  0.054  0.784  3.360  0.688        -0.028  0.041
RV10 0.190  1.276 0.135  0.000  0.888  0.596  1.003  0.488  0.309         0.029
RV11 0.727  0.091 0.085  0.041  0.746  0.963  2.067  0.089  0.675  0.330       
RV12 0.214  0.159 0.107  1.166  1.182  0.004  0.006  0.562  0.217  0.417  0.278
RV13 0.489  0.065 0.288  0.335  1.139  0.027  1.042  0.955  0.294  0.192  1.690
RV14 0.084  1.130 0.133  0.042  0.504  0.603  0.004  1.436  0.646  0.408  0.072
RV15 1.958  0.789 0.262  0.173  0.425  1.486  1.462  1.477  0.056  0.186  0.288
RV16 0.043  0.088 0.856  1.123  0.563  1.371  0.736  0.029  3.644  0.095  0.752
RV17 0.229  0.823 0.298  0.090  0.659  1.706  0.837  0.045  0.341  0.015  0.402
RV18 0.802  0.357 1.654  0.020  0.075  1.142  0.558  0.346  0.081  5.328  0.135
       RV12   RV13   RV14   RV15   RV16   RV17   RV18
RV1   0.023  0.035 -0.014  0.070  0.010 -0.024  0.045
RV2   0.020 -0.013  0.053  0.044 -0.015  0.045 -0.030
RV3  -0.016 -0.027  0.018 -0.026  0.046 -0.027 -0.064
RV4   0.054  0.029  0.010 -0.021 -0.053  0.015 -0.007
RV5   0.054 -0.053  0.035  0.032 -0.037  0.040  0.014
RV6  -0.003 -0.008  0.039 -0.061 -0.058  0.065 -0.053
RV7  -0.004 -0.051  0.003  0.060  0.043 -0.046 -0.037
RV8   0.037  0.049  0.060 -0.061  0.008  0.011 -0.029
RV9  -0.023  0.027 -0.040  0.012  0.095 -0.029 -0.014
RV10 -0.032  0.022 -0.032  0.022  0.015 -0.006 -0.115
RV11 -0.026  0.065 -0.013 -0.027  0.043 -0.032  0.018
RV12         0.067  0.051 -0.023 -0.054 -0.006  0.033
RV13  1.811        -0.062  0.018 -0.033  0.011 -0.027
RV14  1.036  1.567        -0.040  0.031 -0.001 -0.063
RV15  0.220  0.131  0.648        -0.100  0.005  0.080
RV16  1.168  0.442  0.389  3.988         0.041 -0.023
RV17  0.013  0.051  0.001  0.009  0.680        -0.013
RV18  0.440  0.288  1.583  2.598  0.221  0.071       
```

que no detecta problemas de violación de la indepedencia local de los ítems, dado que los valores de V de Cramer (triángulo superior de la matriz) son relativamente bajos.

7.4 Gráficos

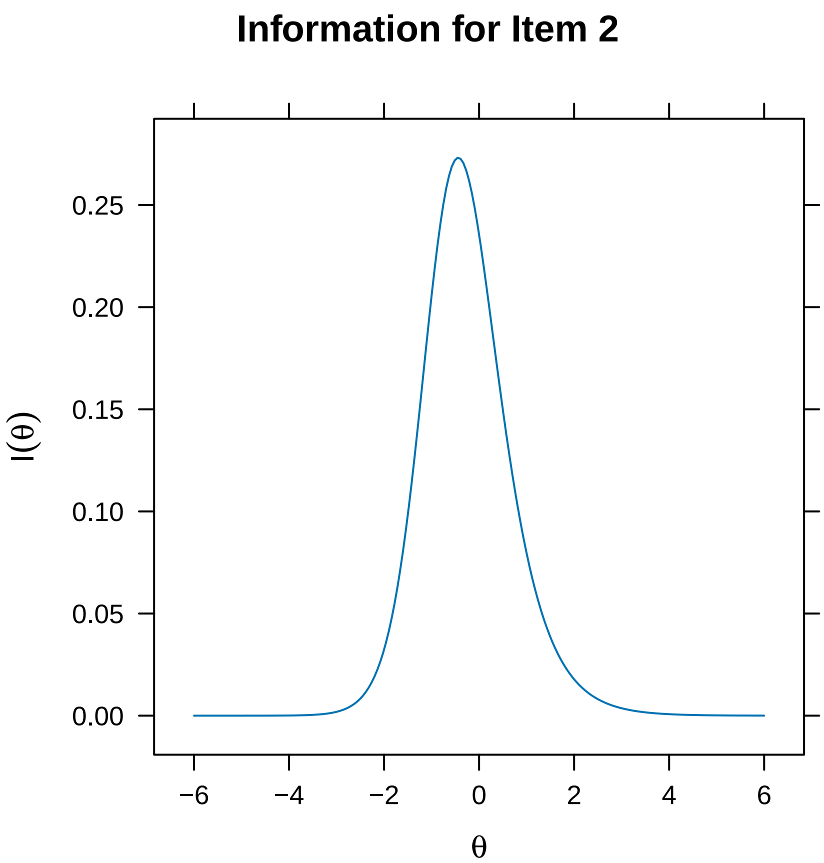
En este modelo, la `FRI` de un ítem (e.g., ítem `2`) se obtiene con:

```
itemplot(m.3p, 2)
```



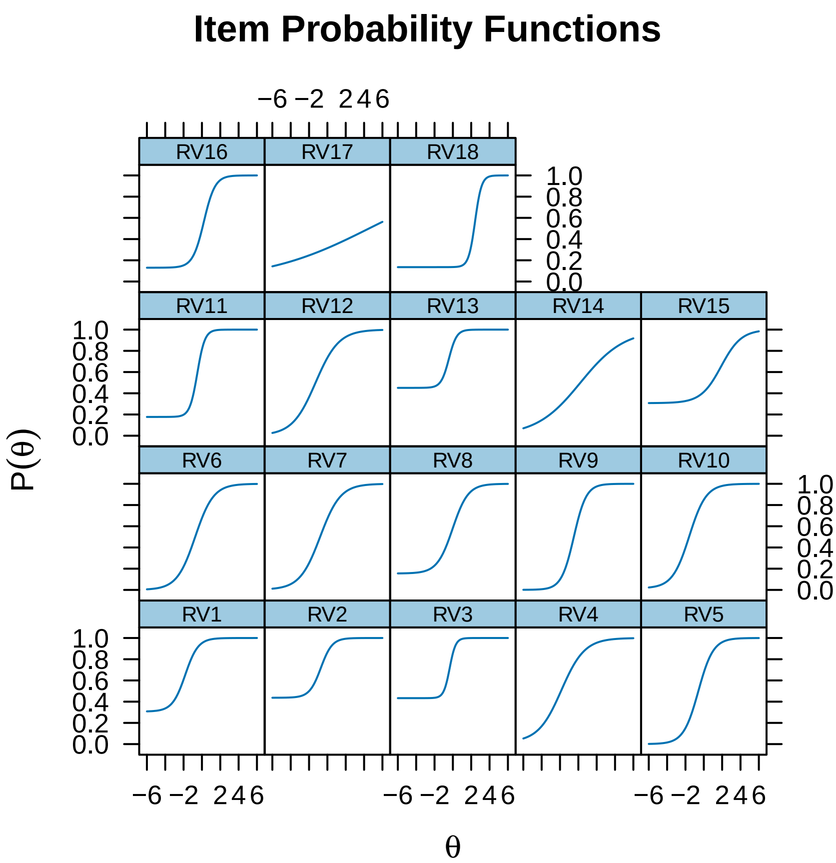
y la `FII` se obtiene con:

```
itemplot(m.3p, item=2, type="info")
```



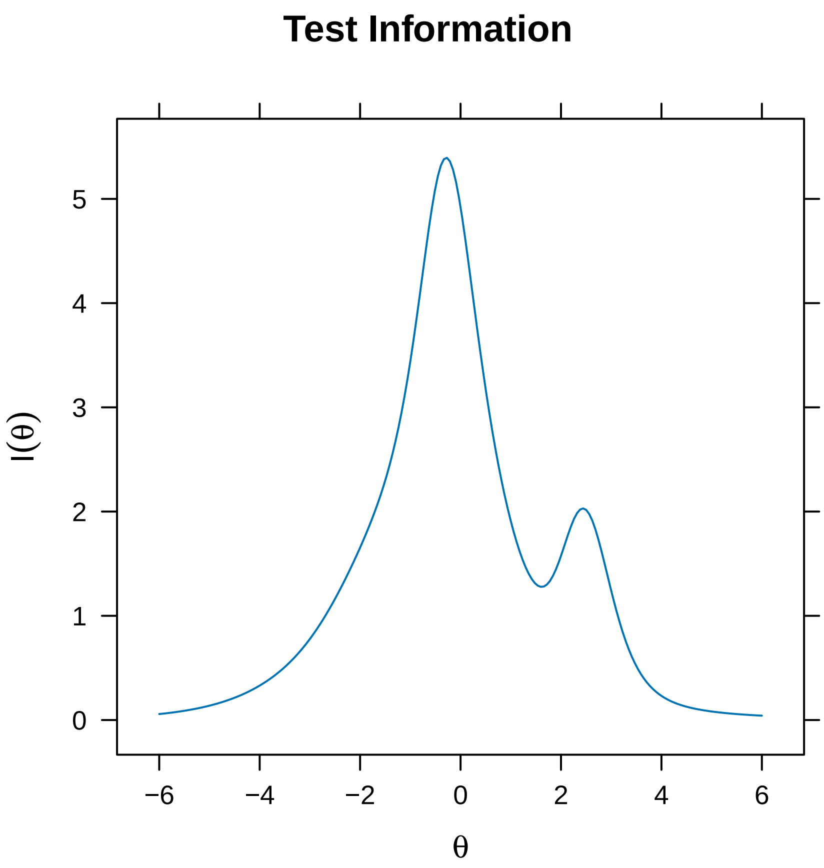
Las `FRIs` de todos los ítems se obtienen con:

```
plot(m.3p, type="trace")
```



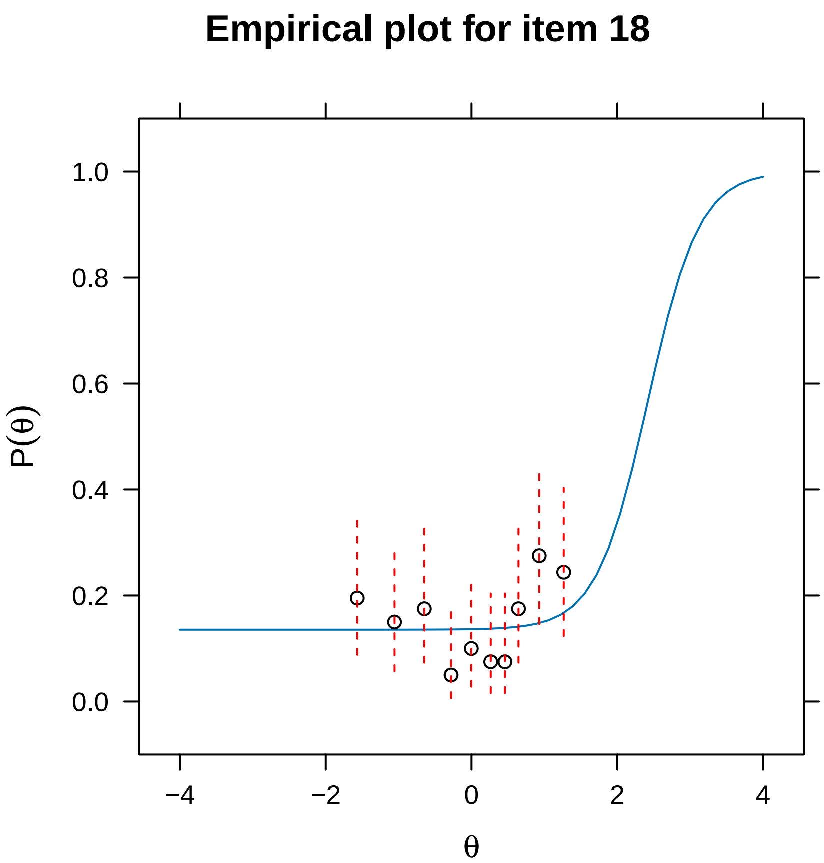
y la FIT para el test completo con:

```
plot(m.3p, type="info")
```



Para examinar la posible fuente de mal ajuste, se puede revisar el gráfico comparativo entre las `FRIs` con:

```
itemfit(m.3p, empirical.plot=18, empirical.CI=.95)
```



que replica el resultado en el modelo logístico de `2-p`, donde se aprecia que la probabilidad de acertar el ítem en el intervalo de habilidad `[-4, +2]` se mantiene prácticamente en la probabilidad de ser acertado por azar.

Capítulo 8

Comparación de modelos

Supuesto que los tres modelos están anidados, el paquete también permite determinar qué modelo se ajusta mejor a los datos de este test. En ese caso, se comparan los modelos dos a dos con la función `anova()`. Para compara el modelo logístico de `1-p` con el modelo logístico de `2-p` emplearemos:

```
anova(m.1p, m.2p)
```

```
          AIC    SABIC       HQ      BIC    logLik      X2 df p
m.1p 7939.469 7955.113 7969.534 8015.402 -3950.735
m.2p 7842.848 7872.489 7899.812 7986.720 -3885.424 130.621 17 0
```

La diferencia entre ambos modelos (χ^2) ha resultado altamente significativa (`p=0`) y los criterios `AIC`, `BIC` y `SABIC` han resultado más bajos en el modelo logístico de `2-p`, por lo que, en principio, este último modelo puede ser el más explicativo de las respuestas a los ítems en este test. A continuación, se comparan el modelo logístico de `2-p` con el modelo de `3-p` con:

```
anova(m.2p, m.3p)
```

```
          AIC    SABIC       HQ      BIC    logLik     X2 df     p
m.2p 7842.848 7872.489 7899.812 7986.720 -3885.424
m.3p 7864.227 7908.689 7949.674 8080.036 -3878.114 14.621 18 0.688
```

y el resultado evidencia que no se han encontrado diferencias significativas entre ambos modelos. Además, todos los estadísticos de ajuste han empeorado sus valores, por lo que se mantiene el modelo logístico de `2-p` como el apropiado para estos datos.