

Modelos politómicos con el paquete `ltm` de R

José Ant. López Pina

Índice

1	Introducción	2
2	Lectura de la base de datos	3
3	Análisis clásico de ítems	5
4	Modelo de Crédito parcial generalizado	8
4.1	Estimación de parámetros	8
4.2	Gráficos	12
5	Modelo de respuesta graduada	15
5.1	Estimación de parámetros	15
5.2	Gráficos	18

Capítulo 1

Introducción

En este documento se presenta un ejemplo de cómo estimar parámetros con los modelos de respuesta al ítem empleando R o RStudio con el paquete `ltm`. No obstante, para invertir los ítems se necesita otro paquete de R (`psych`). Antes de realizar cualquier análisis, recuerde instalar ambos paquetes con la función `install.packages("ltm")` e `install.packages("psych")` y cargar las librerías correspondientes con las funciones `library(ltm)` y `library(psych)`.

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 los nombres de las variables en la primera fila y los ítems de la escala están separados (`sep`) por comas. La lectura del archivo se asigna al objeto `rse0` que contiene la matriz de datos que se desea analizar. Para estimar los parámetros de localización y los umbrales de las categorías, se deben eliminar de la matriz de datos las columnas que no forman parte del análisis, asignando el resultado a un nuevo objeto `rse1 <- rse0[, -c(1,12)]`. En este caso, 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 `ltm` realice las estimaciones apropiadamente. Para ello, debemos realizar la recodificación de los ítems invertidos con el paquete `psych`¹. Se asigna, entonces, a la función `clave` el orden directo (1) o inverso (-1) de los ítems del cuestionario. Debe haber tantos 1's y -1's como ítems haya en el cuestionario. Finalmente se crea un nuevo objeto (`rse`) donde se almacena la matriz de datos, una vez recodificados los ítems con la función `reverse.code()`. Estos pasos se resumen en las funciones siguientes:

```
library(psych)
setwd("c:/datos/")
rse0 <- read.table("rse.dat", header=T, sep=",")
rse1 <- rse0[, -c(1,12)]
clave <- c(1,-1,1,1,-1,1,1,-1,-1,-1)
rse <- reverse.code(clave, rse1)
```

Para comprobar que todo el proceso ha sido correcto se pueden obtener los seis primeros casos de cada objeto. Por ejemplo, con la función, en el objeto `rse1` aparecen los valores originales:

```
head(rse1)
```

```
Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10
```

¹Recuerde instalar el paquete `psych` antes de cargar la librería correspondiente.

```

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 la función:

```
head(rse)
```

```

      Q1 Q2- Q3 Q4 Q5- Q6 Q7 Q8- Q9- Q10-
[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 en este caso se ha añadido un signo - al lado del ítem invertido.

Antes de iniciar los análisis conviene evaluar la dimensionalidad de la matriz de datos con:

```
dim(rse)
```

```
[1] 500 10
```

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

```
summary(rse)
```

Q1		Q2-		Q3		Q4		Q5-	
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-		Q9-		Q10-	
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

La distribución de frecuencias en todos los ítems se obtiene con:

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

```

      Q1 Q2-  Q3  Q4 Q5-  Q6  Q7 Q8-  Q9-  Q10-
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

```

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:

```
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-	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-	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-	8	500	1.34	0.91	1	1.30	1.48	0	3	3	0.32	-0.68	0.04
Q9-	9	500	1.08	0.90	1	0.98	1.48	0	3	3	0.62	-0.31	0.04
Q10-	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-	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-	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-	0.91	0.91	0.91	0.52	9.7	0.0062	0.0074	0.51
Q9-	0.90	0.90	0.90	0.50	8.9	0.0067	0.0101	0.51
Q10-	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-	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-	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-	500	0.60	0.60	0.52	0.50	1.3	0.91
Q9-	500	0.71	0.70	0.67	0.63	1.1	0.90
Q10-	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-	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-	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-	0.17	0.45	0.25	0.13	0
Q9-	0.27	0.47	0.16	0.10	0
Q10-	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 este paquete con:

```
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 congénéricos (`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-      Q3      Q4      Q5-      Q6      Q7      Q8-  
0.4288371 0.4913827 0.5705609 0.2954630 0.5598519 0.6784633 0.6592813 0.3214192  
      Q9-      Q10-  
0.5666536 0.8000429
```

```
$status
```

```
[1] 0
```

```
$Call
```

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

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

Capítulo 4

Modelo de Crédito parcial generalizado

4.1 Estimación de parámetros

Este modelo fue diseñado especialmente para escalas de rendimiento con ítem politómicos donde se espera que los estudiantes muestren un conocimiento parcial, con la particularidad de que cada ítem puede tener un parámetro de discriminación diferente. Ahora procede estimar los parámetros con este modelo con:

```
library(ltm)
```

```
Cargando paquete requerido: MASS
```

```
Cargando paquete requerido: msm
```

```
Cargando paquete requerido: polycor
```

```
m.mcpgr <- gpcm(rse, IRT.param = T)  
summary(m.mcpgr)
```

Call:

```
gpcm(data = rse, IRT.param = T)
```

Model Summary:

log.Lik	AIC	BIC
-5021.547	10123.09	10291.68

Coefficients:

\$Q1

	value	std.err	z.value
Catgr.1	-1.752	0.143	-12.245
Catgr.2	-1.032	0.100	-10.364
Catgr.3	0.705	0.087	8.150
Dscrmn	1.885	0.180	10.458

\$`Q2-`

	value	std.err	z.value
Catgr.1	-2.100	0.192	-10.913

Catgr.2	-1.434	0.126	-11.396
Catgr.3	0.805	0.096	8.403
Dscrmn	1.635	0.158	10.339

\$Q3

	value	std.err	z.value
Catgr.1	-1.364	0.105	-13.040
Catgr.2	-0.374	0.078	-4.814
Catgr.3	1.131	0.096	11.737
Dscrmn	2.099	0.197	10.632

\$Q4

	value	std.err	z.value
Catgr.1	-2.338	0.227	-10.289
Catgr.2	-1.099	0.128	-8.605
Catgr.3	1.222	0.131	9.307
Dscrmn	1.221	0.123	9.934

\$`Q5-`

	value	std.err	z.value
Catgr.1	-1.286	0.113	-11.380
Catgr.2	-0.175	0.087	-2.008
Catgr.3	1.218	0.112	10.879
Dscrmn	1.660	0.157	10.558

\$Q6

	value	std.err	z.value
Catgr.1	-1.264	0.093	-13.601
Catgr.2	-0.058	0.071	-0.815
Catgr.3	1.397	0.101	13.776
Dscrmn	2.406	0.228	10.563

\$Q7

	value	std.err	z.value
Catgr.1	-1.025	0.081	-12.700
Catgr.2	0.043	0.069	0.624
Catgr.3	1.402	0.099	14.165
Dscrmn	2.650	0.257	10.331

\$`Q8-`

	value	std.err	z.value
Catgr.1	-1.732	0.202	-8.561
Catgr.2	0.812	0.171	4.760
Catgr.3	1.462	0.212	6.900
Dscrmn	0.799	0.085	9.363

\$`Q9-`

	value	std.err	z.value
Catgr.1	-0.877	0.111	-7.914
Catgr.2	1.129	0.140	8.049
Catgr.3	1.443	0.169	8.543

```
Dscrmn 1.249 0.124 10.050
```

```
$`Q10-`
```

```
      value std.err z.value
Catgr.1 -0.789 0.096 -8.175
Catgr.2 0.497 0.103 4.824
Catgr.3 1.038 0.120 8.682
Dscrmn 1.560 0.151 10.317
```

Integration:

```
method: Gauss-Hermite
quadrature points: 21
```

Optimization:

```
Convergence: 0
max(|grad|): 0.022
optimizer: nlminb
```

Un resumen de los resultados se puede obtener marcando directamente el objeto:

```
m.mcpg
```

Call:

```
gpcm(data = rse, IRT.param = T)
```

Coefficients:

```
      Catgr.1 Catgr.2 Catgr.3 Dscrmn
Q1      -1.752 -1.032  0.705  1.885
Q2-     -2.100 -1.434  0.805  1.635
Q3      -1.364 -0.374  1.131  2.099
Q4      -2.338 -1.099  1.222  1.221
Q5-     -1.286 -0.175  1.218  1.660
Q6      -1.264 -0.058  1.397  2.406
Q7      -1.025  0.043  1.402  2.650
Q8-     -1.732  0.812  1.462  0.799
Q9-     -0.877  1.129  1.443  1.249
Q10-    -0.789  0.497  1.038  1.560
```

Log.Lik: -5021.547

O también se puede obtener la misma información con:

```
coef(m.mcpg)
```

```
      Catgr.1 Catgr.2 Catgr.3 Dscrmn
Q1      -1.752 -1.032  0.705  1.885
Q2-     -2.100 -1.434  0.805  1.635
Q3      -1.364 -0.374  1.131  2.099
Q4      -2.338 -1.099  1.222  1.221
Q5-     -1.286 -0.175  1.218  1.660
Q6      -1.264 -0.058  1.397  2.406
Q7      -1.025  0.043  1.402  2.650
```

```

Q8-  -1.732  0.812  1.462  0.799
Q9-  -0.877  1.129  1.443  1.249
Q10- -0.789  0.497  1.038  1.560

```

Para cada ítem aparecen tres umbrales (`Catgr.1`, `Catgr.2`, `Catgr3`) supuesto que los ítems tienen cuatro categorías y un parámetro de discriminación (`Dscrmn`). En general, los ítems presentan parámetros de discriminación relativamente elevados, siendo los ítems Q7 y Q8 los más discriminativos. Un examen de los umbrales de las categorías revelan que todos están ordenados. La función `m.mcpG$convergence` permite evaluar la convergencia del modelo. Si el resultado es 0 entonces el modelo ha encontrado la convergencia. Ahora procede comprobar si el modelo explica los resultados obtenidos en los ítems de esta escala. Para ello, se utiliza:

```
GoF.gpcm(m.mcpG, B = 99)
```

Parametric Bootstrap Approximation to Pearson chi-squared Goodness-of-Fit Measure

Call:

```
gpcm(data = rse, IRT.param = T)
```

Tobs: 1.51974e+13

data-sets: 100

p-value: 0.01

Este paquete no produce estadísticos de ajuste individuales para cada ítem en la escala, sino un estadístico global, basado en una aproximación de remuestreo, que indica si el modelo se ajusta a las respuestas a los ítems. Dado que el valor de p (`p-value: 0.01`) es menor que 0.05, se rechaza la hipótesis nula de que el modelo explica los respuestas a los ítems en esta escala. Una opción es probar modelos alternativos restringiendo los parámetros de discriminación. Así, podemos estimar el modelo bajo la restricción de que todos los ítems siguen el modelo de Rasch ($\alpha_j = 1$) o bajo la restricción de que todos los ítems tienen el mismo parámetro de discriminación (modelo logístico de 1-p). Para ello creamos dos objetos diferentes que contemplen ambos análisis. Los objetos para estos análisis se especifican como:

```

m.mcpG.rasch <- gpcm(rse, constraint="rasch")
m.mcpG.1p <- gpcm(rse, constraint="1PL")

```

La función `anova()` permite comparar ambos modelos con el modelo de crédito parcial generalizado. Así, la comparación con el modelo de Rasch será:

```
anova(m.mcpG.rasch, m.mcpG)
```

Likelihood Ratio Table

	AIC	BIC	log.Lik	LRT	df	p.value
m.mcpG.rasch	10361.49	10487.93	-5150.75		30	
m.mcpG	10123.09	10291.68	-5021.55	258.4	40	<0.001

y la comparación con el modelo logístico de 1-p será:

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

Likelihood Ratio Table

	AIC	BIC	log.Lik	LRT	df	p.value
m.mcpg.1p	10229.30	10359.95	-5083.65		31	
m.mcpg	10123.09	10291.68	-5021.55	124.2	40	<0.001

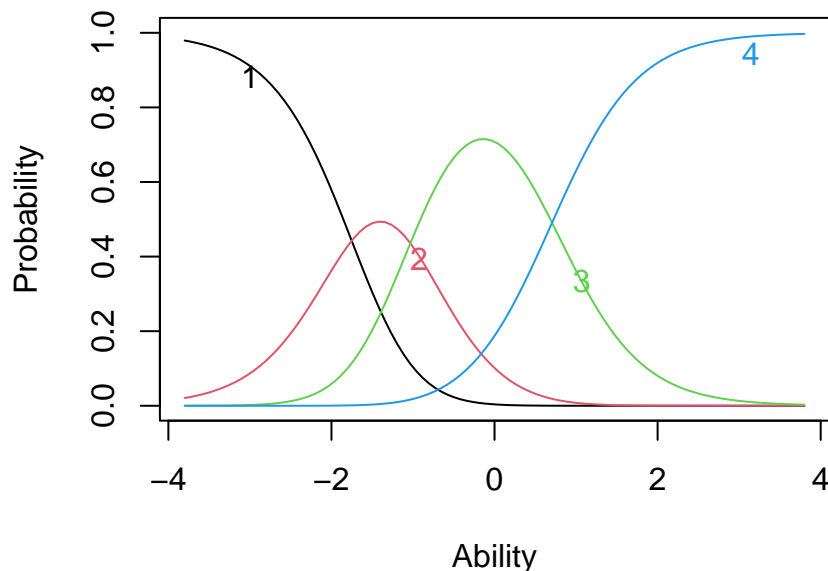
La hipótesis nula en ambos casos es que el modelo de crédito parcial o el modelo de crédito parcial generalizado con todas las pendientes iguales explican mejor las respuestas a los ítems. La hipótesis alternativa es que el modelo de crédito parcial generalizado es una mejor explicación de las respuestas a los ítems, la cual se cumple en ambos casos.

4.2 Gráficos

La Función de Respuesta de las Categorías (FRCs) para el ítem 1 en el modelo de crédito parcial generalizado se puede obtener con:

```
plot(m.mcpg, items=1)
```

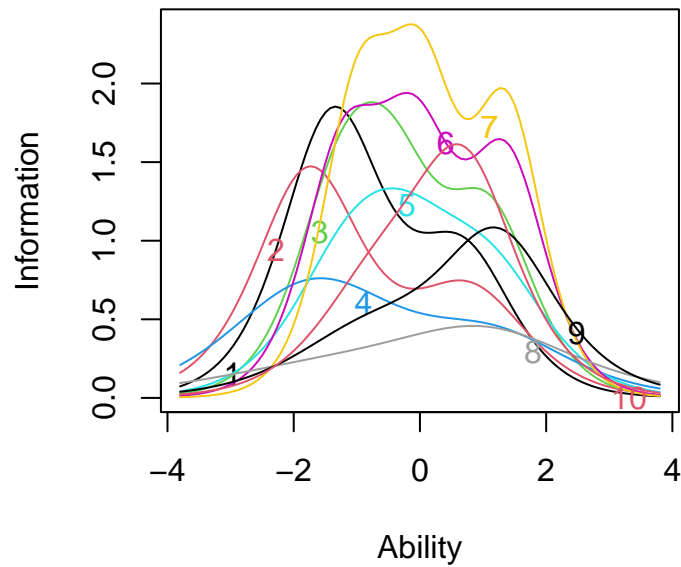
Item Response Category Characteristic Curves – Item



Si se desean graficar varios ítems, solo es necesario especificar en la función los que se desean graficar tal como `plot(m.mcpg, items=1:5, legend=F)` que producirá las FRCs de los cinco primeros ítems. Para graficar la función de información de todos los ítems se emplea:

```
plot(m.mcpg, type="IIC", legend=F)
```

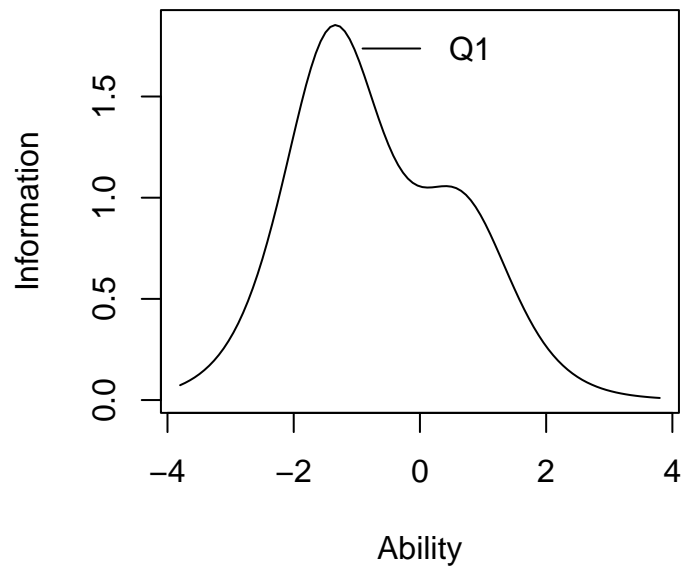
Item Information Curves



y para graficar la FII de un ítem concreto entonces la función tendrá la forma:

```
plot(m.mcpg,type="IIC",item=1,legend=T)
```

Item Information Curves



Por último, la estimación de los parámetros de habilidad para cada caso se realiza con:

```
fs <- factor.scores(m.mcpg, method="EAP")  
head(fs$score.dat, 5)
```

Q1	Q2-	Q3	Q4	Q5-	Q6	Q7	Q8-	Q9-	Q10-	Obs	Exp	z1	se.z1
----	-----	----	----	-----	----	----	-----	-----	------	-----	-----	----	-------

1	1	1	1	1	1	1	1	1	1	1	1	1.060672e+00	-2.705040	0.5013190
2	1	1	1	1	1	1	1	2	1	1	1	5.262648e-01	-2.518955	0.4641020
3	1	1	1	2	1	1	1	1	4	4	1	5.149286e-09	-1.398848	0.2399516
4	1	1	1	2	1	2	1	1	1	1	1	7.845728e-02	-2.062540	0.3507542
5	1	1	2	2	1	1	1	1	1	1	1	1.245354e-01	-2.100896	0.3562068

Capítulo 5

Modelo de respuesta graduada

5.1 Estimación de parámetros

Para implementar el modelo de respuesta graduada en `ltm` debemos emplear:

```
m.mrg <- grm(rse, IRT.param=T, Hessian=TRUE)
m.mrg$convergence
```

```
[1] 0
```

Las estimaciones de los parámetros se pueden obtener con:

```
coef(m.mrg)
```

	Extrmt1	Extrmt2	Extrmt3	Dscrmn
Q1	-1.897	-0.897	0.731	2.381
Q2-	-2.407	-1.286	0.790	2.065
Q3	-1.423	-0.306	1.163	2.556
Q4	-2.529	-0.920	1.204	1.631
Q5-	-1.349	-0.116	1.265	2.185
Q6	-1.289	-0.026	1.437	2.752
Q7	-1.047	0.100	1.444	3.018
Q8-	-1.609	0.491	1.928	1.234
Q9-	-0.820	0.896	1.834	1.868
Q10-	-0.818	0.426	1.290	2.237

Los errores típicos de las estimaciones de los parámetros se pueden obtener con:

```
summary(m.mrg)
```

Call:

```
grm(data = rse, IRT.param = T, Hessian = TRUE)
```

Model Summary:

log.Lik	AIC	BIC
-4954.842	9989.685	10158.27

Coefficients:

\$Q1

value	std.err	z.vals
-------	---------	--------

Extrmt1	-1.897	0.128	-14.792
Extrmt2	-0.897	0.155	-5.788
Extrmt3	0.731	0.145	5.047
Dscrmn	2.381	0.189	12.571

\$`Q2-`

	value	std.err	z.vals
Extrmt1	-2.407	0.173	-13.893
Extrmt2	-1.286	0.204	-6.304
Extrmt3	0.790	0.158	4.989
Dscrmn	2.065	0.170	12.168

\$Q3

	value	std.err	z.vals
Extrmt1	-1.423	0.099	-14.406
Extrmt2	-0.306	0.100	-3.066
Extrmt3	1.163	0.512	2.270
Dscrmn	2.556	0.196	13.057

\$Q4

	value	std.err	z.vals
Extrmt1	-2.529	0.202	-12.500
Extrmt2	-0.920	0.187	-4.928
Extrmt3	1.204	0.256	4.696
Dscrmn	1.631	0.139	11.726

\$`Q5-`

	value	std.err	z.vals
Extrmt1	-1.349	0.102	-13.192
Extrmt2	-0.116	0.084	-1.369
Extrmt3	1.265	0.502	2.522
Dscrmn	2.185	0.170	12.862

\$Q6

	value	std.err	z.vals
Extrmt1	-1.289	0.091	-14.170
Extrmt2	-0.026	0.082	-0.321
Extrmt3	1.437	1.420	1.012
Dscrmn	2.752	0.218	12.603

\$Q7

	value	std.err	z.vals
Extrmt1	-1.047	0.079	-13.275
Extrmt2	0.100	0.067	1.490
Extrmt3	1.444	2.059	0.701
Dscrmn	3.018	0.242	12.468

\$`Q8-`

	value	std.err	z.vals
Extrmt1	-1.609	0.158	-10.191
Extrmt2	0.491	0.086	5.688

```
Extrmt3  1.928  0.737  2.615
Dscrmn   1.234  0.113 10.910
```

\$`Q9-`

```
      value std.err z.vals
Extrmt1 -0.820  0.089 -9.202
Extrmt2  0.896  0.155  5.767
Extrmt3  1.834  1.725  1.063
Dscrmn   1.868  0.150 12.425
```

\$`Q10-`

```
      value std.err z.vals
Extrmt1 -0.818  0.081 -10.156
Extrmt2  0.426  0.069  6.185
Extrmt3  1.290  0.732  1.763
Dscrmn   2.237  0.170 13.160
```

Integration:

method: Gauss-Hermite
quadrature points: 21

Optimization:

Convergence: 0
max(|grad|): 0.013
quasi-Newton: BFGS

Dado que el paquete `ltm` no proporciona estadísticos de ajuste de los ítems en el modelo de respuesta graduada, una opción es comparar a través de la función `anova()` los resultados con un modelo restringido en que los parámetros de discriminación sean iguales para todos los ítems. Para estimar el modelo restringido empleamos:

```
m.mrg.res <- grm(rse, constrained=T, IRT.param=T)
```

cuyo resultado es:

```
coef(m.mrg.res)
```

```
      Extrmt1 Extrmt2 Extrmt3 Dscrmn
Q1      -1.992  -0.937   0.769  2.106
Q2-     -2.377  -1.274   0.783  2.106
Q3      -1.535  -0.326   1.250  2.106
Q4      -2.211  -0.817   1.065  2.106
Q5-     -1.366  -0.114   1.287  2.106
Q6      -1.421  -0.022   1.583  2.106
Q7      -1.185   0.123   1.635  2.106
Q8-     -1.164   0.373   1.421  2.106
Q9-     -0.774   0.856   1.743  2.106
Q10-    -0.840   0.437   1.328  2.106
```

Para comparar ambos modelos empleamos:

```
anova(m.mrg.res, m.mrg)
```

Likelihood Ratio Table

	AIC	BIC	log.Lik	LRT df	p.value
m.mrg.res	10075.63	10206.28	-5006.82		
m.mrg	9989.68	10158.27	-4954.84	103.95 9	<0.001

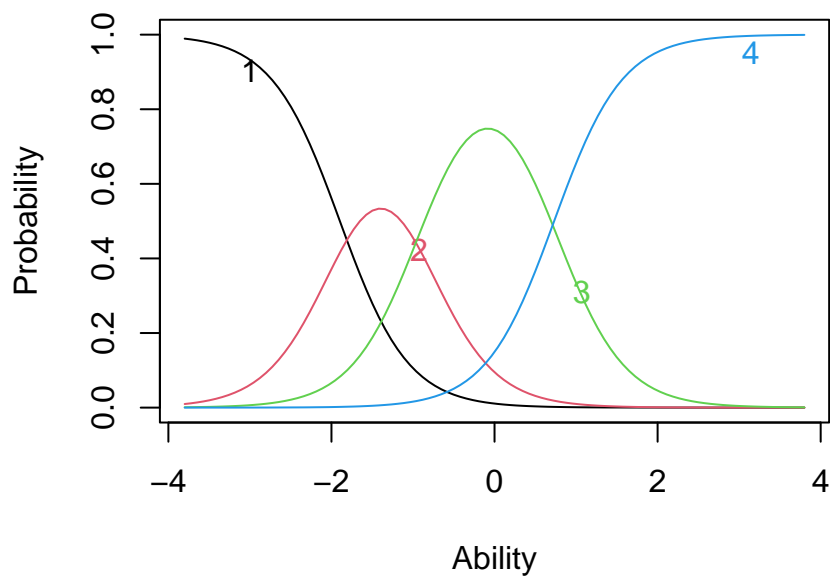
que arroja diferencias significativas entre ambas soluciones ($p < 0.001$), indicando que el modelo con diferentes parámetros de discriminación de los ítems se ajusta mejor a las respuestas que el modelo con parámetros de discriminación iguales para todos los ítems.

5.2 Gráficos

La FRCs en este modelo se determina con:

```
plot(m.mrg, items=1)
```

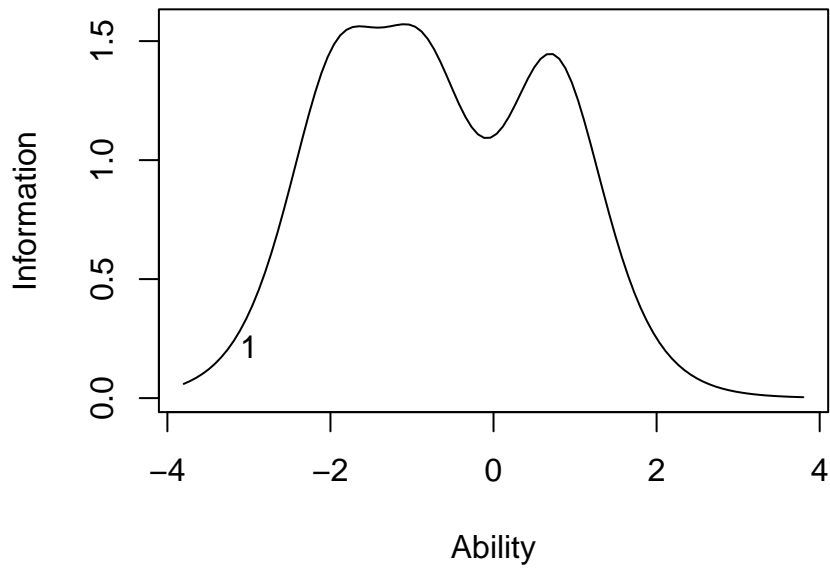
Item Response Category Characteristic Curves – Item



y la función de información se determina con:

```
plot(m.mrg, items=1, type="IIC")
```

Item Information Curves



La función de información para todos los ítems se obtiene con:

```
plot(m.mrg, type="IIC")
```

Item Information Curves

