

# Modelo de Rasch con el paquete **eRm** de R

José Ant. López Pina

# Índice

<b>1</b>	<b>Introducción</b>	<b>2</b>
<b>2</b>	<b>Gestión de los datos</b>	<b>3</b>
2.1	Lectura de la base de datos . . . . .	3
2.2	Corrección de la matriz de respuestas . . . . .	3
<b>3</b>	<b>Análisis de ítems con la TCT</b>	<b>6</b>
3.1	Estadísticos descriptivos de los ítems . . . . .	6
3.2	Análisis de ítems . . . . .	8
3.3	Análisis de distractores . . . . .	15
3.4	Análisis de la fiabilidad . . . . .	25
<b>4</b>	<b>Modelo de Rasch</b>	<b>27</b>
<b>5</b>	<b>Evaluación de la invarianza de parámetros</b>	<b>31</b>
5.1	Estadístico LR de Andersen . . . . .	31
5.2	Estadístico de Wald . . . . .	33
5.3	Eliminación de ítems con LR de Andersen y Wald . . . . .	33
5.4	Evaluación del ajuste con las medias de cuadrados . . . . .	35
<b>6</b>	<b>Parámetros de habilidad</b>	<b>37</b>
6.1	Estimación de parámetros . . . . .	37
6.2	Detección de patrones aberrantes . . . . .	38
<b>7</b>	<b>Índice de separación</b>	<b>40</b>
<b>8</b>	<b>Evaluación de los principios del modelo de Rasch</b>	<b>41</b>
8.1	Unidimensionalidad de la escala . . . . .	41
8.2	Independencia local de los ítems . . . . .	42
<b>9</b>	<b>Gráficos en el paquete eRm</b>	<b>45</b>
<b>10</b>	<b>Elaboración de un nuevo instrumento</b>	<b>52</b>
<b>11</b>	<b>Evaluación del DIF con los métodos de Andersen y Wald</b>	<b>54</b>

# Capítulo 1

## Introducción

En este documento se implementa el modelo de Rasch en el paquete `eRm`. Este paquete permite estimar parámetros con el método de máxima verosimilitud condicional en el modelo de Rasch. Sin embargo, `eRm` no permite estimar parámetros en los modelos de respuesta al ítem con más de un parámetro ni tampoco se puede emplear para estimar parámetros en modelos multidimensionales.

Para instalar el paquete escriba en R o RStudio: `install.packages("eRm")`. Si se instala 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 instalados ambos paquetes, no es preciso que se vuelvan a instalar cada vez que comienza 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(eRm)`.

Una función puede tener múltiples 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 el modelo de Rasch conviene realizar un estudio descriptivo previo de los ítems, así como un análisis del funcionamiento de los distractores de los ítems<sup>1</sup>. Para ello, emplearemos los paquetes `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("subscore")`, `install.packages("ShinnyItemAnalysis")` e `install.packages("CTT")`.

---

<sup>1</sup>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 de datos. 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. A continuación, se asigna la matriz de datos 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, para cada caso, separadas (`sep`) por comas. La lectura del archivo se asigna al objeto `rv0` que contiene la matriz de datos que se desea analizar. En este caso, el archivo contiene muchos datos ausentes indicados con `NA`, por lo que se pueden adoptar dos acciones: 1) Eliminar los casos que contengan datos ausentes o 2) imputar la respuesta 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)`.

Dado que posteriormente se realizará un estudio del sesgo de los ítems (DIF), el archivo contiene una columna denominada como `Genero`, que debe definirse en R, y que más adelante se utilizará en el estudio del DIF. No obstante, para estimar los parámetros de dificultad en el modelo de Rasch 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

Ahora en el objeto `rv2` disponemos de las respuestas no corregidas de los estudiantes, por lo que para aplicar el modelo de Rasch necesitamos la matriz de respuestas corregidas. Para ello utilizamos el paquete `ltm` que convertirá la matriz de respuestas original a una matriz de respuestas corregidas que contengan 1's y 0's. Para ello, instale el paquete `ltm` desde R con `install.packages("ltm")` y cargue la librería correspondiente con:

```
library(ltm)
```

Cargando paquete requerido: MASS

Cargando paquete requerido: msm

Cargando paquete requerido: polycor

El objeto `rv2` contiene las respuestas originales con las opciones de 1 a 4 de 500 niños a 18 ítems del test. Para obtener las respuestas correctas se crea un objeto (e.g., `clave`) que contenga las opciones correctas de cada ítem. Este objeto debe tener el mismo número de ítems hay en el objeto `rv2`. Finalmente se corrige la matriz de respuestas con la función `mult.choice(rv2, clave)`. La matriz de respuestas corregida se almacena ahora en el objeto `rv`. La secuencia de los pasos indicados anteriormente se detalla a continuación:

```
setwd("c:/datos/")
rv0 <- read.table("rv.dat", header=T, sep=",")
rv1 <- na.omit(rv0)
genero <- as.character(rv1$Genero)
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 <- mult.choice(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 una vez que se han eliminado los casos con omisiones NA. Si escribimos a continuación:

```
head(rv)
```

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

```
5 1
6 0
```

tenemos la matriz de respuestas corregida. La función:

```
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:

```
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:

```
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

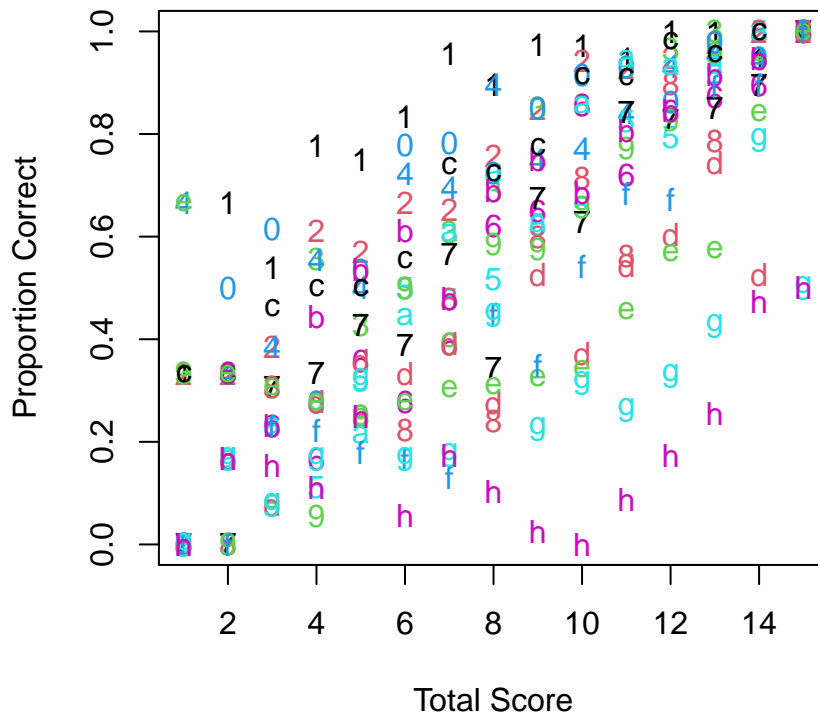
Pairwise Associations:

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:

```
library(CTT)
```

Adjuntando el paquete: 'CTT'



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			162	0.40298507	-0.09178694	0.01955168	0.2818182	0.54807692
2			34	0.08457711	-0.32898455	-0.15890411	0.2000000	0.03846154
3	*		61	0.15174129	0.02153160	0.18792030	0.1545455	0.03846154
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:ltm':

```
factor.scores
```

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 comprender 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
```

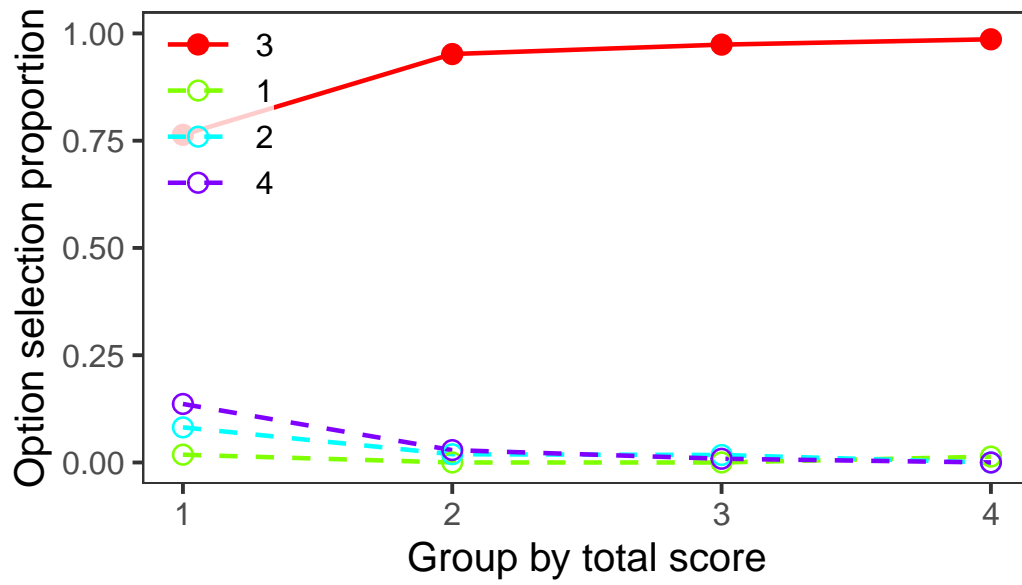
```
This is ShinyItemAnalysis version 1.5.5
- to run the interactive {shiny} app, call `run_app()`
- to learn more, visit `ShinyItemAnalysis.org`
```

```
plotDistractorAnalysis(rv2,clave,num.groups=4,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
```

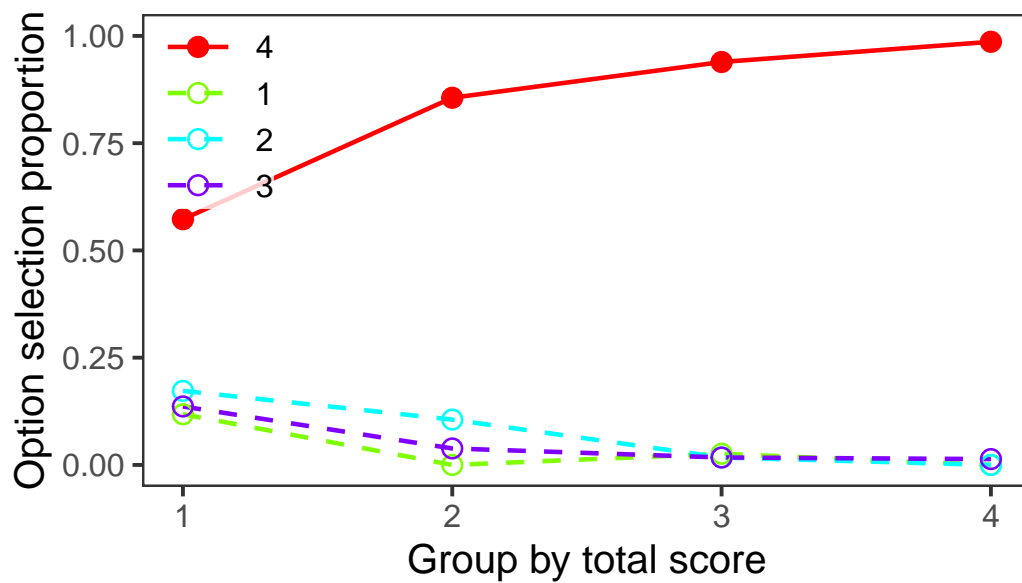
## RV1



```
plotDistractorAnalysis(rv2,clave,num.groups=4,item="RV2")
```

\$RV2

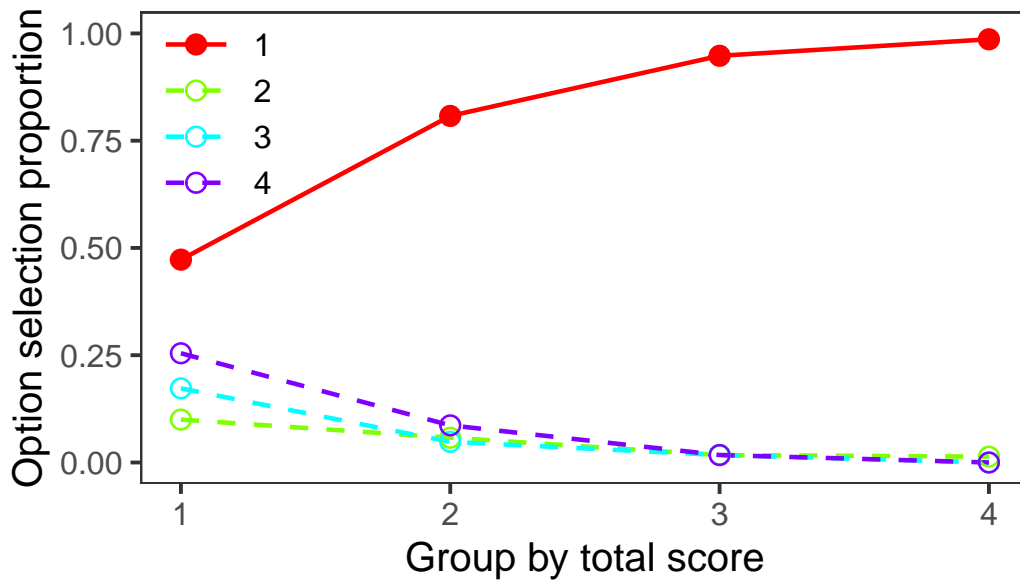
## RV2



```
plotDistractorAnalysis(rv2,clave,num.groups=4,item="RV3")
```

\$RV3

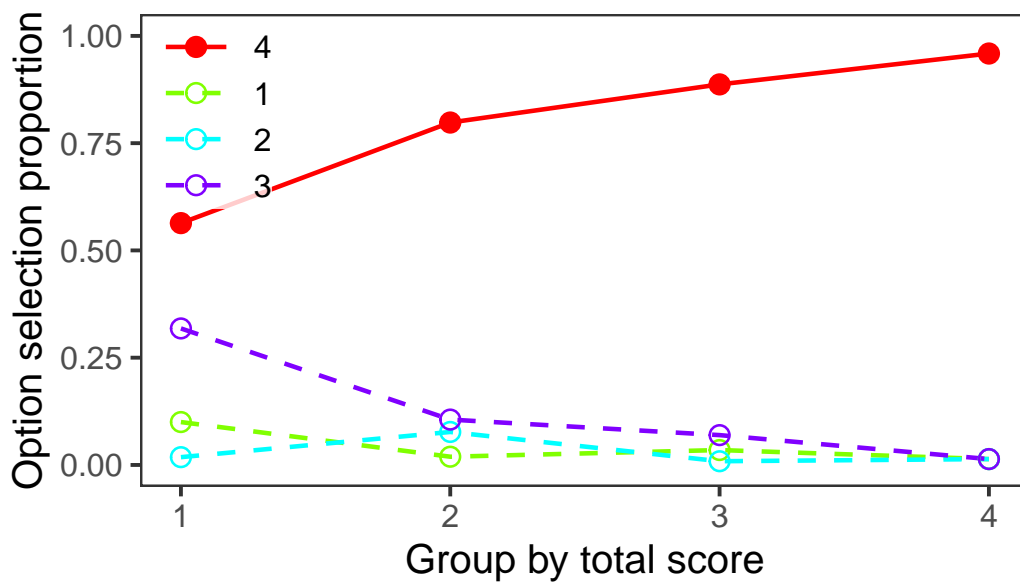
### RV3



```
plotDistractorAnalysis(rv2,clave,num.groups=4,item="RV4")
```

\$RV4

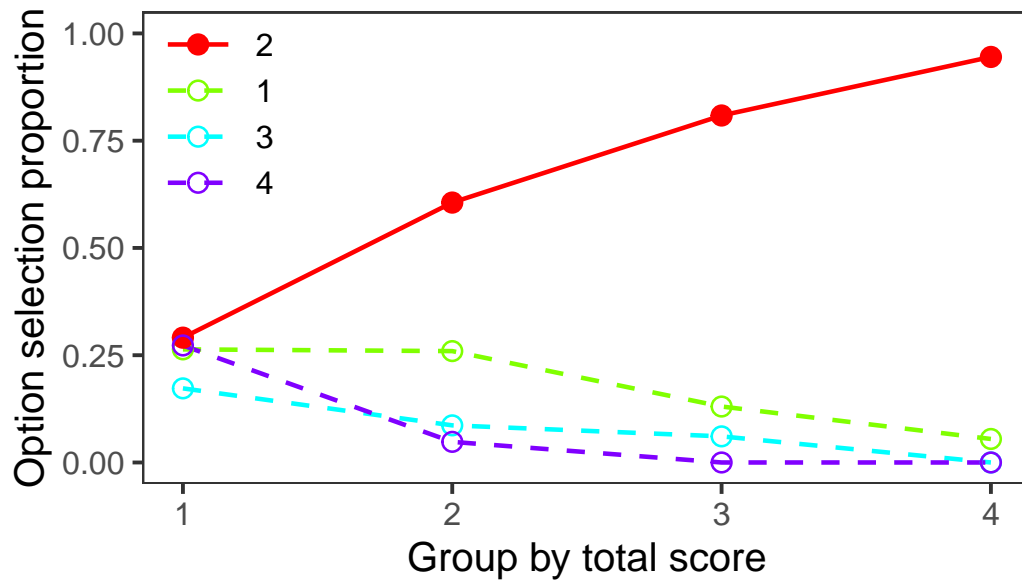
### RV4



```
plotDistractorAnalysis(rv2,clave,num.groups=4,item="RV5")
```

\$RV5

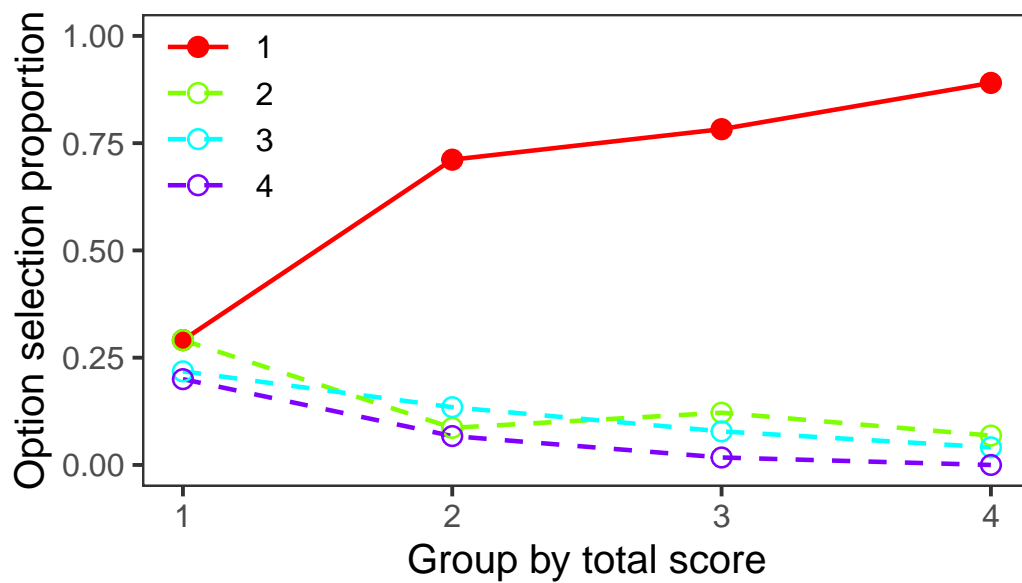
## RV5



```
plotDistractorAnalysis(rv2,clave,num.groups=4,item="RV6")
```

\$RV6

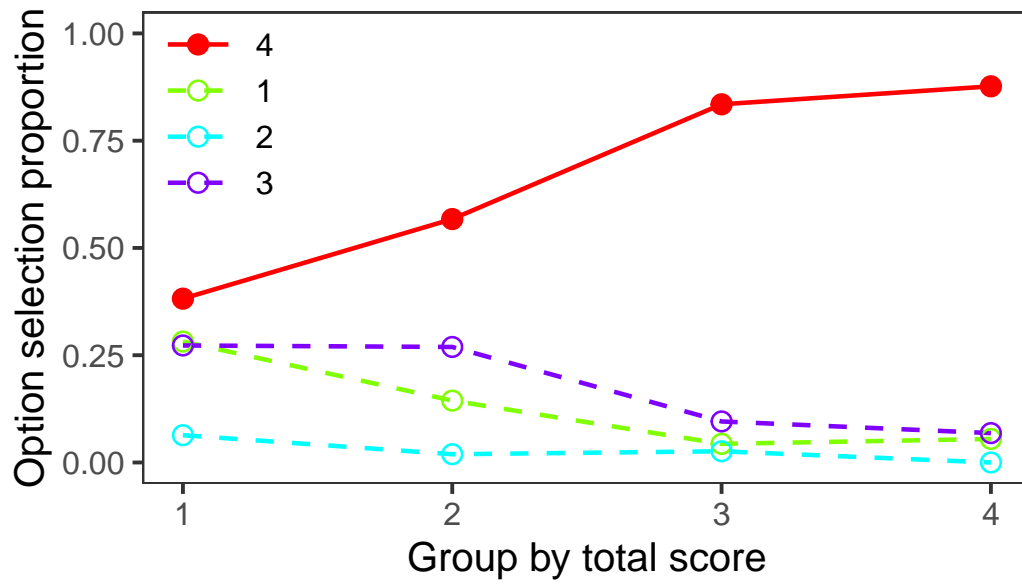
## RV6



```
plotDistractorAnalysis(rv2,clave,num.groups=4,item="RV7")
```

\$RV7

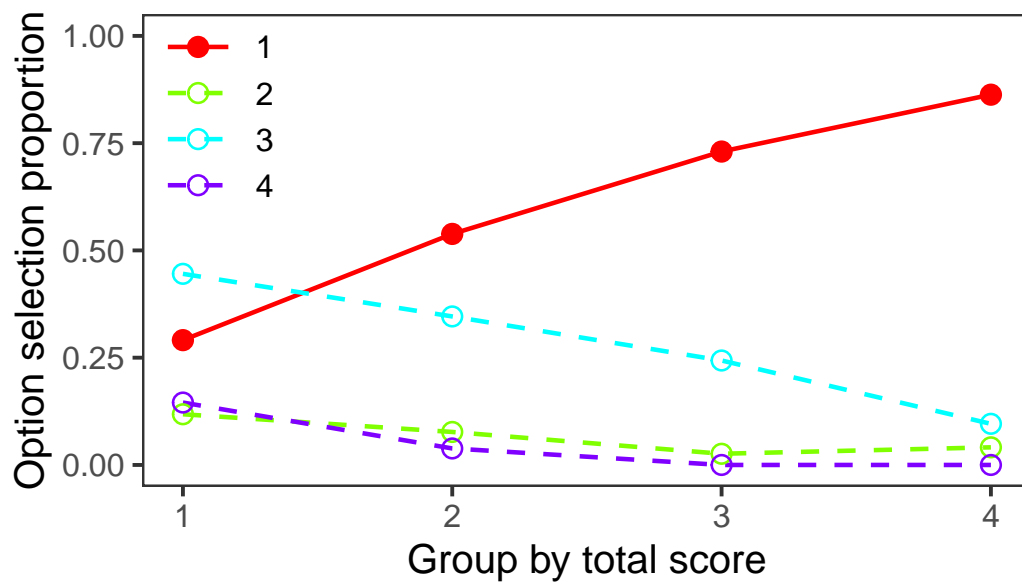
## RV7



```
plotDistractorAnalysis(rv2,clave,num.groups=4,item="RV8")
```

\$RV8

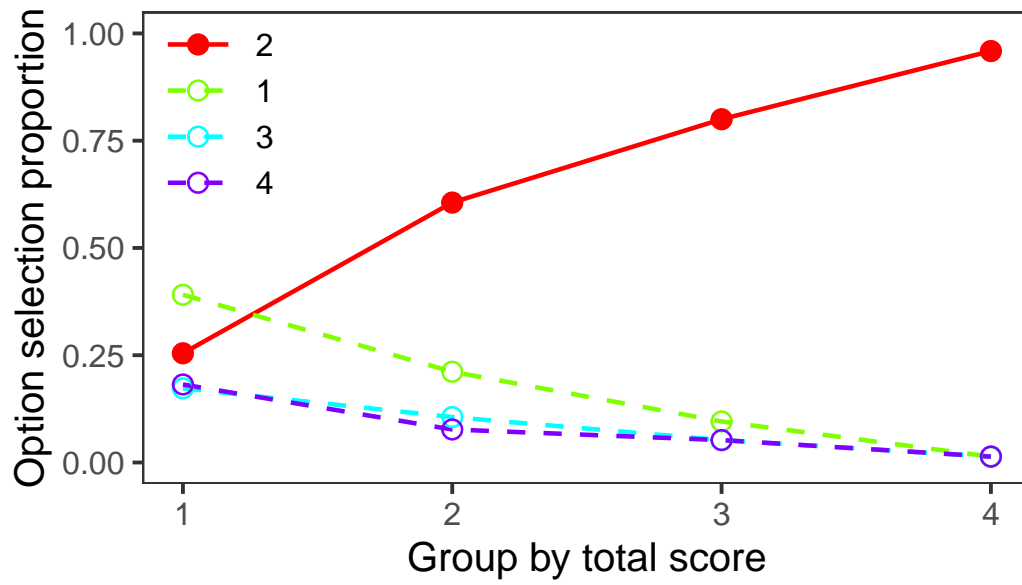
## RV8



```
plotDistractorAnalysis(rv2,clave,num.groups=4,item="RV9")
```

\$RV9

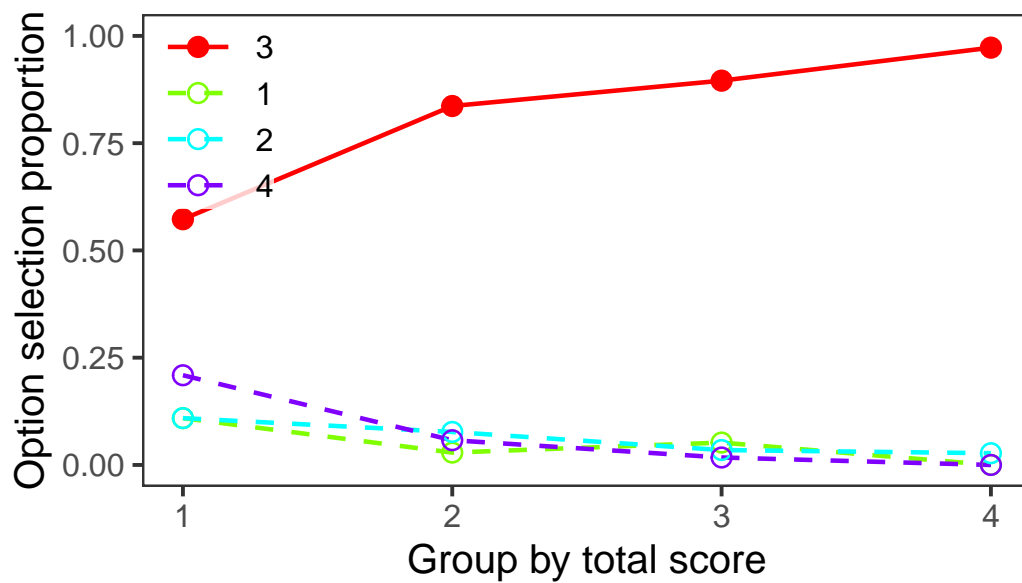
## RV9



```
plotDistractorAnalysis(rv2,clave,num.groups=4,item="RV10")
```

\$RV10

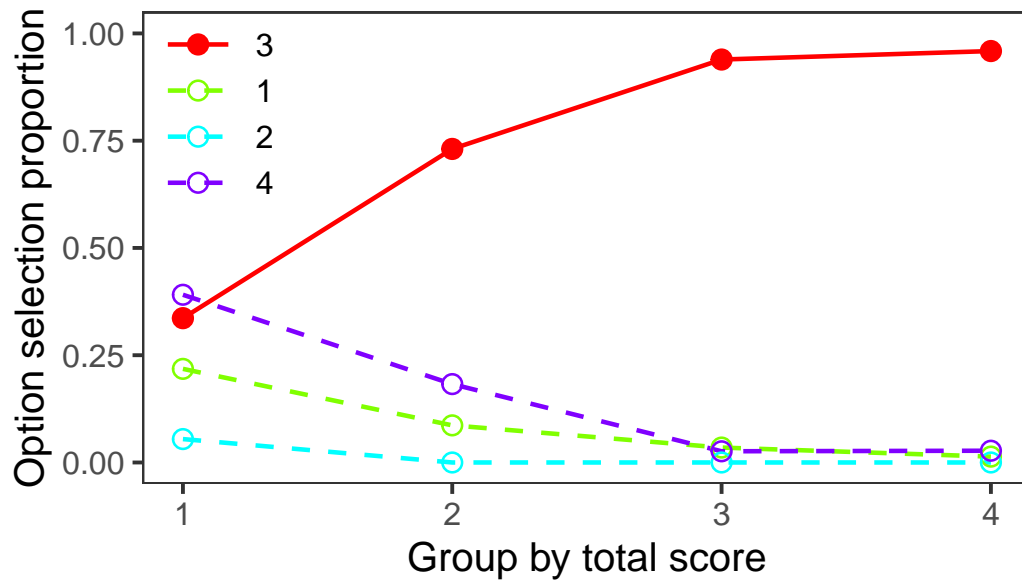
## RV10



```
plotDistractorAnalysis(rv2,clave,num.groups=4,item="RV11")
```

\$RV11

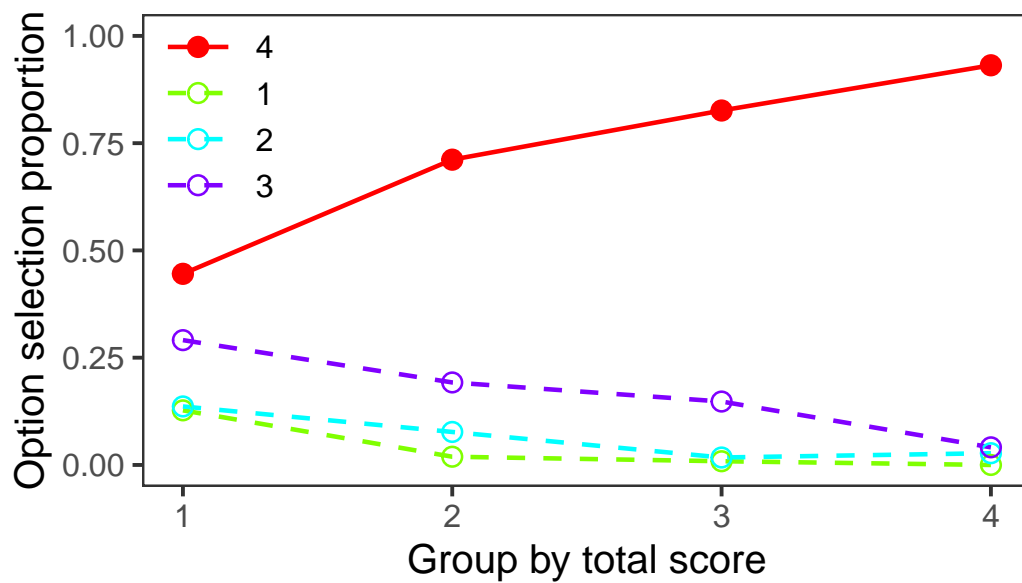
## RV11



```
plotDistractorAnalysis(rv2,clave,num.groups=4,item="RV12")
```

\$RV12

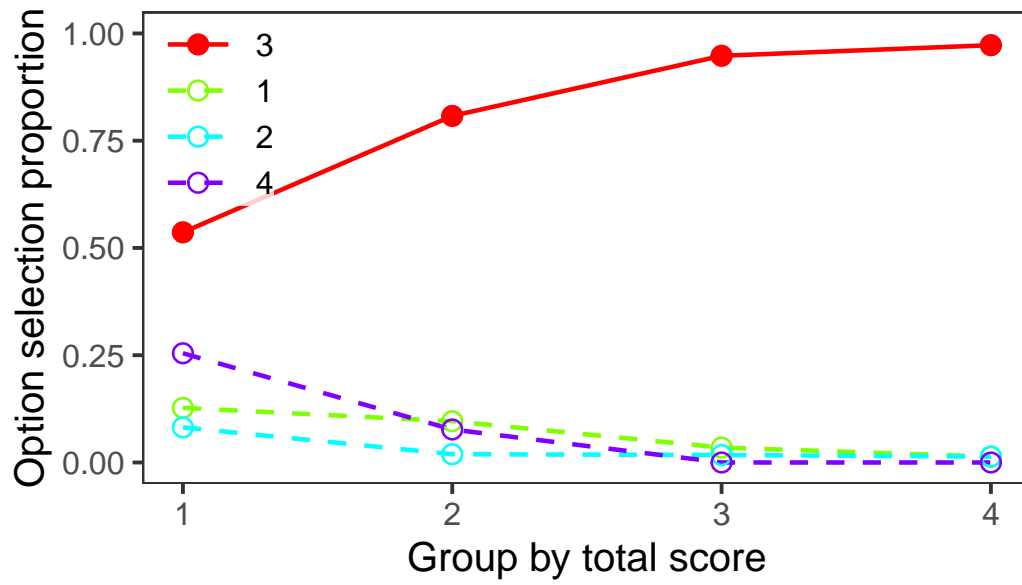
## RV12



```
plotDistractorAnalysis(rv2,clave,num.groups=4,item="RV13")
```

\$RV13

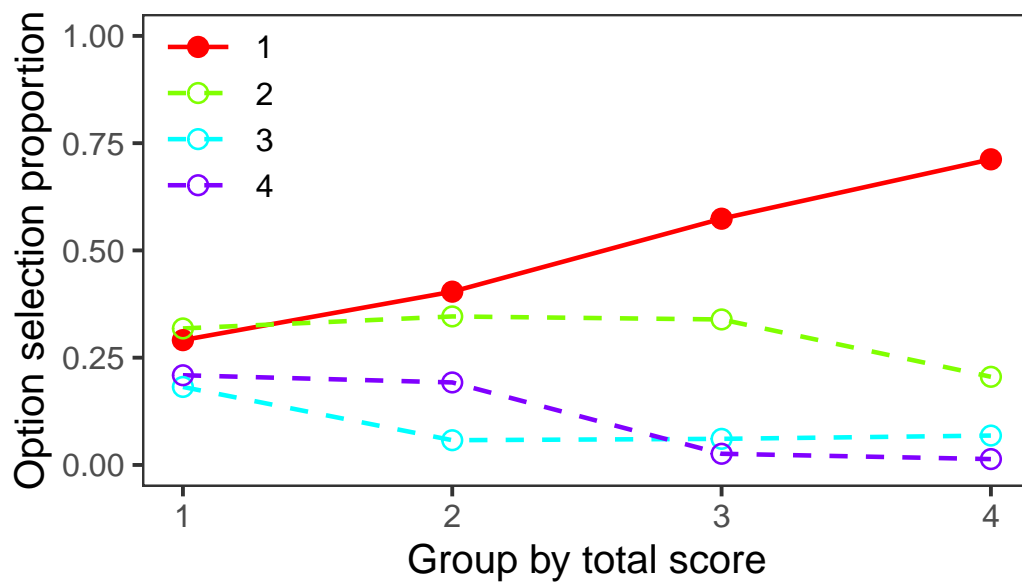
## RV13



```
plotDistractorAnalysis(rv2,clave,num.groups=4,item="RV14")
```

\$RV14

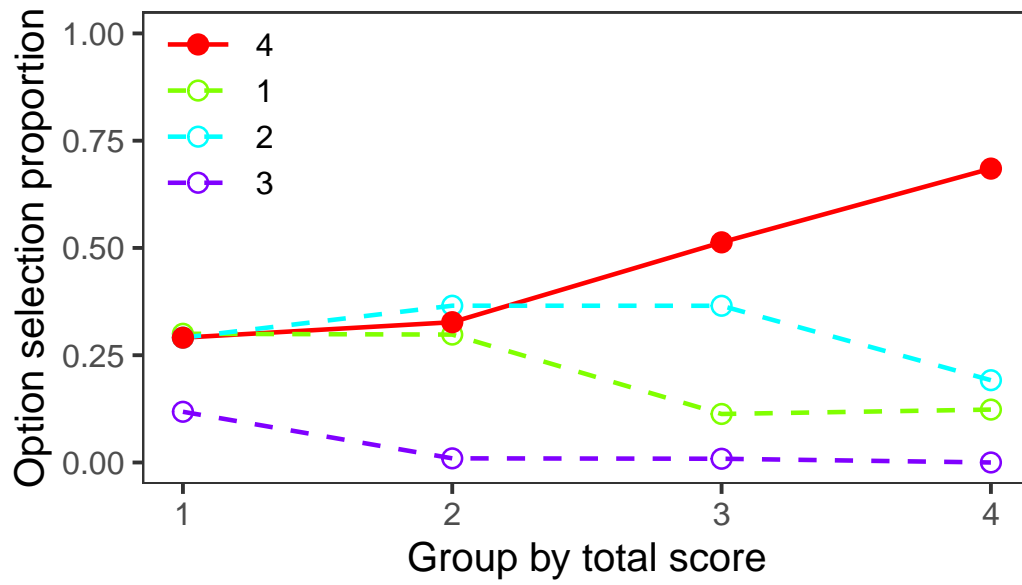
## RV14



```
plotDistractorAnalysis(rv2,clave,num.groups=4,item="RV15")
```

\$RV15

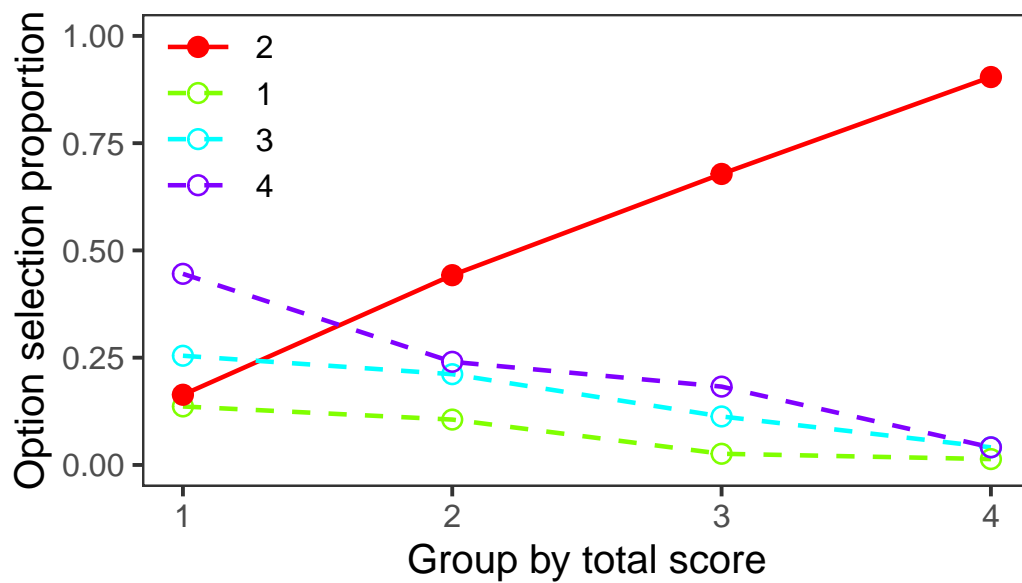
## RV15



```
plotDistractorAnalysis(rv2,clave,num.groups=4,item="RV16")
```

\$RV16

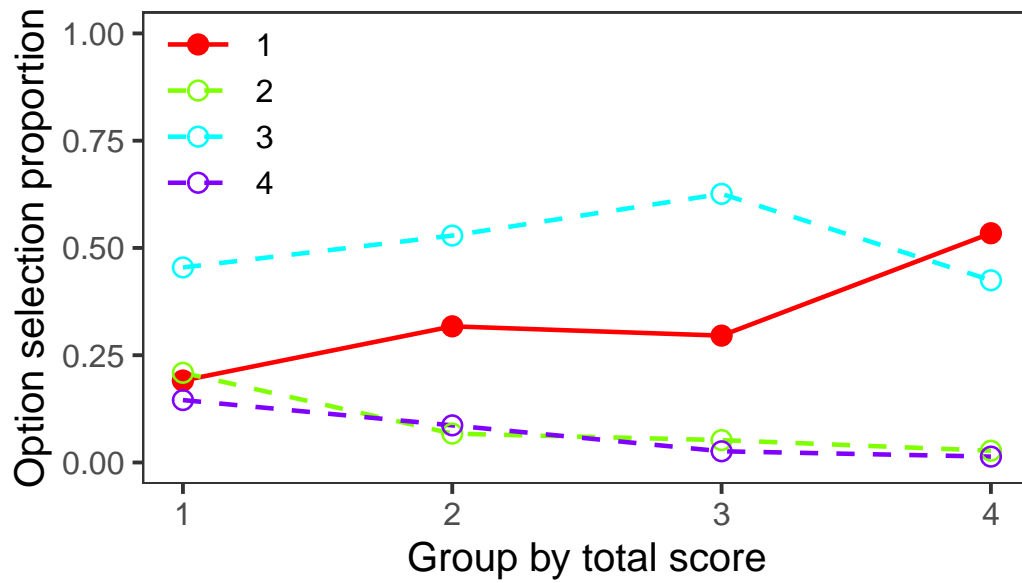
## RV16



```
plotDistractorAnalysis(rv2,clave,num.groups=4,item="RV17")
```

\$RV17

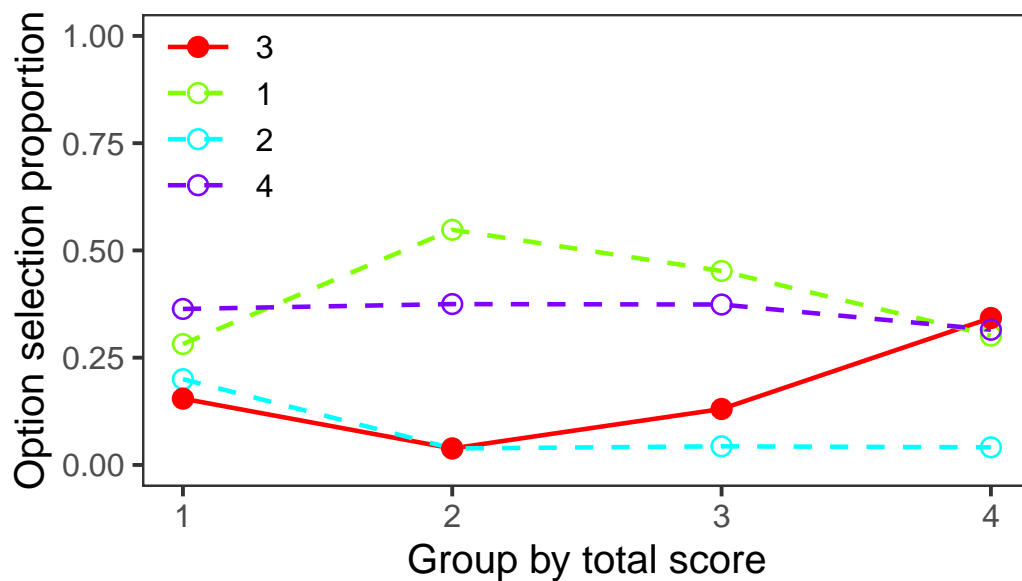
## RV17



```
plotDistractorAnalysis(rv2,clave,num.groups=4,item="RV18")
```

\$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 adecuadamente, 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:ltm':

```
factor.scores
```

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		

```
splitHalf(rv)
```

Split half reliabilities

```
Call: splitHalf(r = rv)
```

Maximum split half reliability (lambda 4) = 0.79

```

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  $\tau$ -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(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)
```

Todos los coeficientes de consistencia interna afirman que las puntuaciones de esta escala presentan una fiabilidad aceptable.

Para trabajar con el modelo de Rasch debemos eliminar las librerías activas hasta el momento y cargar la librería del paquete `eRm` con:

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

## Capítulo 4

# Modelo de Rasch

Para estimar los parámetros en el modelo de Rasch se utiliza la función `RM()` con la siguiente estructura: `m.rasch <- RM(rv, sum0=T)`. La opción `sum0=T` fija la media de los parámetros de dificultad a 0.

```
m.rasch <- RM(rv, sum0=T)
```

Una vez que se han estimado los parámetros, se puede comprobar la convergencia del proceso con:

```
m.rasch$convergence
```

```
[1] 1
```

que devuelve 1 o 0. Si devuelve 1, el algoritmo de estimación de parámetros ha encontrado la convergencia. Devuelve 0 en caso contrario. Para estimar los parámetros de dificultad escribiremos<sup>1</sup>:

```
round(cbind(-m.rasch$betapar,m.rasch$se.beta), 3)
```

	[,1]	[,2]
beta RV1	-1.955	0.176
beta RV2	-1.085	0.135
beta RV3	-0.812	0.127
beta RV4	-0.812	0.127
beta RV5	0.055	0.110
beta RV6	0.004	0.111
beta RV7	0.004	0.111
beta RV8	0.328	0.108
beta RV9	0.106	0.110
beta RV10	-0.935	0.131
beta RV11	-0.404	0.117
beta RV12	-0.333	0.116
beta RV13	-0.917	0.130
beta RV14	0.833	0.106
beta RV15	1.030	0.107
beta RV16	0.647	0.106
beta RV17	1.610	0.112
beta RV18	2.635	0.140

<sup>1</sup>La función `'round(,3)'` permite redondear las estimaciones hasta tres decimales.

En la primera columna [,1] aparece el parámetro de dificultad y en la segunda columna [,2] aparece el error típico correspondiente. Si se desea conocer el intervalo de confianza para cada ítem, entonces podemos escribir:

```
round(cbind(-confint(m.rasch)[,2], -confint(m.rasch)[,1]), 3)
```

		[,1]	[,2]
beta RV1		-2.300	-1.610
beta RV2		-1.350	-0.819
beta RV3		-1.061	-0.563
beta RV4		-1.061	-0.563
beta RV5		-0.161	0.272
beta RV6		-0.213	0.222
beta RV7		-0.213	0.222
beta RV8		0.116	0.539
beta RV9		-0.109	0.321
beta RV10		-1.191	-0.679
beta RV11		-0.635	-0.174
beta RV12		-0.560	-0.105
beta RV13		-1.172	-0.662
beta RV14		0.625	1.041
beta RV15		0.821	1.239
beta RV16		0.439	0.855
beta RV17		1.390	1.829
beta RV18		2.360	2.910

Un resumen de las dos funciones anteriores se puede obtener con:

```
summary(m.rasch)
```

Results of RM estimation:

Call: RM(X = rv, sum0 = T)

Conditional log-likelihood: -2921.675

Number of iterations: 16

Number of parameters: 17

Item (Category) Difficulty Parameters (eta): with 0.95 CI:

	Estimate	Std. Error	lower CI	upper CI
RV2	-1.085	0.135	-1.350	-0.819
RV3	-0.812	0.127	-1.061	-0.563
RV4	-0.812	0.127	-1.061	-0.563
RV5	0.055	0.110	-0.161	0.272
RV6	0.004	0.111	-0.213	0.222
RV7	0.004	0.111	-0.213	0.222
RV8	0.328	0.108	0.116	0.539
RV9	0.106	0.110	-0.109	0.321
RV10	-0.935	0.131	-1.191	-0.679
RV11	-0.404	0.117	-0.635	-0.174
RV12	-0.333	0.116	-0.560	-0.105
RV13	-0.917	0.130	-1.172	-0.662

RV14	0.833	0.106	0.625	1.041
RV15	1.030	0.107	0.821	1.239
RV16	0.647	0.106	0.439	0.855
RV17	1.610	0.112	1.390	1.829
RV18	2.635	0.140	2.360	2.910

Item easiness Parameters (beta) with 0.95 CI:

	Estimate	Std. Error	lower CI	upper CI
beta RV1	1.955	0.176	1.610	2.300
beta RV2	1.085	0.135	0.819	1.350
beta RV3	0.812	0.127	0.563	1.061
beta RV4	0.812	0.127	0.563	1.061
beta RV5	-0.055	0.110	-0.272	0.161
beta RV6	-0.004	0.111	-0.222	0.213
beta RV7	-0.004	0.111	-0.222	0.213
beta RV8	-0.328	0.108	-0.539	-0.116
beta RV9	-0.106	0.110	-0.321	0.109
beta RV10	0.935	0.131	0.679	1.191
beta RV11	0.404	0.117	0.174	0.635
beta RV12	0.333	0.116	0.105	0.560
beta RV13	0.917	0.130	0.662	1.172
beta RV14	-0.833	0.106	-1.041	-0.625
beta RV15	-1.030	0.107	-1.239	-0.821
beta RV16	-0.647	0.106	-0.855	-0.439
beta RV17	-1.610	0.112	-1.829	-1.390
beta RV18	-2.635	0.140	-2.910	-2.360

En esta tabla aparecen los parámetros de dificultad (*eta*) y facilidad (*beta*) conjuntamente, aunque en el modelo de Rasch, por regla general, siempre estaremos interesados en los parámetros de dificultad.

Nótese que en la tabla aparecen los parámetros de dificultad para todos los ítems, ya que por defecto `eRm` fija que la suma de los parámetros de dificultad debe ser 0. Esto se especificó en la función `RM(rv, sum0=T)`. Si la opción en esta función fuera `sum0=F`, entonces `eRm` fija el parámetro de dificultad del primer ítem a 0.

Los parámetros de facilidad (*beta*) son simplemente el complementario del parámetro de dificultad (*eta*). Para obtener un resumen de los parámetros de facilidad escribimos:

```
coef(m.rasch)
```

```

beta RV1    beta RV2    beta RV3    beta RV4    beta RV5    beta RV6
1.954983403 1.084783196 0.811891846 0.811891846 -0.055391569 -0.004187012
beta RV7    beta RV8    beta RV9    beta RV10   beta RV11   beta RV12
-0.004187001 -0.327610675 -0.106003074 0.934772852 0.404421905 0.332688256
beta RV13   beta RV14   beta RV15   beta RV16   beta RV17   beta RV18
0.916779073 -0.832810822 -1.029960955 -0.647105872 -1.609688757 -2.635266640
```

pero si escribimos:

```
-coef(m.rasch)
```

```

beta RV1    beta RV2    beta RV3    beta RV4    beta RV5    beta RV6
-1.954983403 -1.084783196 -0.811891846 -0.811891846 0.055391569 0.004187012
beta RV7    beta RV8    beta RV9    beta RV10   beta RV11   beta RV12
```

0.004187001	0.327610675	0.106003074	-0.934772852	-0.404421905	-0.332688256
beta RV13	beta RV14	beta RV15	beta RV16	beta RV17	beta RV18
-0.916779073	0.832810822	1.029960955	0.647105872	1.609688757	2.635266640

obtendremos de forma resumida los parámetros de dificultad de los 18 ítems.

## Capítulo 5

# Evaluación de la invarianza de parámetros

### 5.1 Estadístico LR de Andersen

Para probar la invarianza de parámetros en el modelo de Rasch se aplica el estadístico de razón de verosimilitud LR de Andersen. La función `LRtest` permite calcular este estadístico en dos subgrupos formados a partir de un criterio especificado. Por defecto, esta función permite dividir (`splitcr`) el grupo en dos subgrupos en función de la mediana (`median`), pero también se puede realizar la división en función de la media (`mean`)<sup>1</sup>. El resultado se puede almacenar en un objeto tal como,

```
lr <- LRtest(m.rasch, splitcr="median")
```

Si resumizamos el análisis con:

```
summary(lr)
```

```
Andersen LR-test:  
LR-value: 65.44  
Chi-square df: 17  
p-value: 0
```

```
Subject Subgroup: Raw Scores <= Median:  
Log-likelihood: -1866.795
```

```
Beta Parameters:  
      beta RV1  beta RV2  beta RV3  beta RV4  beta RV5  beta RV6  
Estimate 1.8854275 0.9724330 0.6151190 0.8109499 -0.2015107 0.01335468  
Std.Err. 0.1901849 0.1501838 0.1420737 0.1460491 0.1371383 0.13662991  
      beta RV7  beta RV8  beta RV9 beta RV10 beta RV11 beta RV12  
Estimate -0.08411975 -0.3399731 -0.2803626 0.9253136 0.1500975 0.3480267  
Std.Err. 0.13669780 0.1381725 0.1376581 0.1488953 0.1369834 0.1384142  
      beta RV13 beta RV14 beta RV15 beta RV16 beta RV17 beta RV18  
Estimate 0.7664221 -0.6268416 -0.8007352 -0.8458188 -1.0842151 -2.2235672
```

<sup>1</sup>Otra opción para `splitcr` puede ser una variable externa (e.g.: género) que permite evaluar el DIF globalmente con este estadístico.

Std.Err. 0.1450467 0.1421609 0.1458556 0.1469785 0.1540976 0.2202044

Subject Subgroup: Raw Scores > Median:  
 Log-likelihood: -1022.16

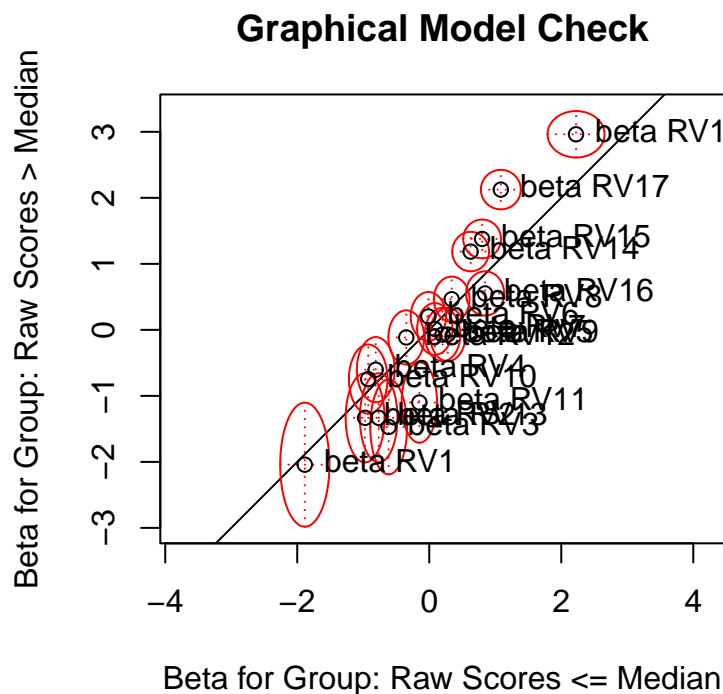
Beta Parameters:

	beta RV1	beta RV2	beta RV3	beta RV4	beta RV5	beta RV6
Estimate	2.0423970	1.3322264	1.4701113	0.6035543	0.07051262	-0.2034803
Std.Err.	0.4794083	0.3447765	0.3670495	0.2520788	0.20616783	0.1884481
	beta RV7	beta RV8	beta RV9	beta RV10	beta RV11	beta RV12
Estimate	-0.01355376	-0.4638627	0.07051262	0.7461865	1.1004164	0.1146519
Std.Err.	0.20033681	0.1748913	0.20616796	0.2672471	0.3109711	0.2093740
	beta RV13	beta RV14	beta RV15	beta RV16	beta RV17	beta RV18
Estimate	1.3322265	-1.185925	-1.3772301	-0.5515246	-2.124610	-2.9626092
Std.Err.	0.3447758	0.152341	0.1497934	0.1710053	0.152563	0.1795548

obtenemos el estadístico de Andersen y los parámetros beta para ambos grupo.

El estadístico de  $\chi^2$  (df:17) ha resultado significativo (p-value: 0), por lo que este test no sigue el modelo de Rasch. La salida de este estadístico produce, además, los parámetros de dificultad para los dos grupos (por encima y por debajo de la mediana), donde se aprecia que estos parámetros beta son claramente diferentes en algunos ítems en uno y otro grupo. Una visión de conjunto de los resultados en ambos grupos con este estadístico se puede obtener con `plotGOF()` que permite graficar los ítems en función del estadístico LR de Andersen:

```
plotGOF(lr, conf=list())
```



En este gráfico aparecen las estimaciones de los parámetros de los ítems para ambos grupos rodeados de una elipse en función de un intervalo confidencial. Si no existen diferencias entre los parámetros de ambos grupos, entonces los ítems (sus elipses) deben tocar la línea de referencia trazada en el gráfico. Si no la tocan, los ítems cuya elipse está por debajo de la

mediana son más difíciles para las personas que están en el subgrupo menos hábil y los ítems cuya elipse está por encima de la mediana son más difíciles para las personas que están en el subgrupo más hábil. En este caso se aprecia que según el estadístico LR de Andersen, no parecen seguir el modelo de Rasch los ítems RV3, RV11, RV14, RV15, RV17 y RV18. Los ítems RV3 y RV11 han resultado más difíciles para el grupo cuyas puntuaciones están por debajo de la mediana, mientras que los ítems RV14, RV15, RV17 y RV18 han resultado más difíciles para el grupo cuyas puntuaciones están por encima de la mediana.

## 5.2 Estadístico de Wald

El estadístico de Wald permite probar el ajuste de cada ítem al modelo de Rasch y se obtiene con:

```
Waldtest(m.rasch, splitcr="median")
```

Wald test on item level (z-values):

	z-statistic	p-value
beta RV1	-0.304	0.761
beta RV2	-0.957	0.339
beta RV3	-2.172	0.030
beta RV4	0.712	0.477
beta RV5	-1.099	0.272
beta RV6	0.932	0.352
beta RV7	-0.291	0.771
beta RV8	0.556	0.578
beta RV9	-1.415	0.157
beta RV10	0.586	0.558
beta RV11	-2.797	0.005
beta RV12	0.930	0.352
beta RV13	-1.513	0.130
beta RV14	2.683	0.007
beta RV15	2.757	0.006
beta RV16	-1.305	0.192
beta RV17	4.798	0.000
beta RV18	2.601	0.009

El estadístico de Wald indica que las probabilidades (p-value) de los ítems RV3, RV11, RV14, RV15, RV17 y RV18 han resultado significativas ( $p < .05$ ), por lo que puede que estos ítem no se comporten de acuerdo con lo que espera el modelo de Rasch. La decisión en función del estadístico de Wald coincide con la obtenida con el estadístico de razón de verosimilitud de Andersen.

## 5.3 Eliminación de ítems con LR de Andersen y Wald

El paquete eRm permite eliminar los ítems utilizando tres criterios: a) un estadístico basado en la distribución de  $\chi^2$ , b) el estadístico LR de Andersen y c) el estadístico de Wald.

Para estudiar la eliminación de ítems de acuerdo con el estadístico basado en  $\chi^2$  se emplea:

```
stepwiseIt(m.rasch, criterion = list("itemfit"), alpha=0.05)
```

Eliminated item - Step 1: RV18  
 Eliminated item - Step 2: RV15  
 Eliminated item - Step 3: RV17  
 Eliminated item - Step 4: RV14  
 Eliminated item - Step 5: RV12  
 Eliminated item - Step 6: RV4  
 Eliminated item - Step 7: RV6  
 Eliminated item - Step 8: RV8

Results for stepwise item elimination:  
 Number of steps: 8  
 Criterion: Itemfit

	Chisq	df	p-value
Step 1: RV18	679.898	401	0.000
Step 2: RV15	540.490	398	0.000
Step 3: RV17	535.151	397	0.000
Step 4: RV14	489.005	388	0.000
Step 5: RV12	449.863	372	0.003
Step 6: RV4	443.099	362	0.002
Step 7: RV6	415.721	357	0.017
Step 8: RV8	399.595	342	0.017

Si se utiliza el estadístico LR de Andersen, entonces se emplea:

```
stepwiseIt(m.rasch, criterion=list("LRtest"), alpha=0.05)
```

Eliminated item - Step 1: RV17  
 Eliminated item - Step 2: RV15  
 Eliminated item - Step 3: RV18  
 Eliminated item - Step 4: RV14

Results for stepwise item elimination:  
 Number of steps: 4  
 Criterion: Andersen's LR-test

	LR-value	Chisq	df	p-value
Step 1: RV17	65.440	17	0.000	
Step 2: RV15	39.269	16	0.001	
Step 3: RV18	46.808	15	0.000	
Step 4: RV14	29.939	14	0.008	
Step 5: None	16.810	13	0.208	

Finalmente, si se emplea el estadístico de Wald se debe utilizar:

```
stepwiseIt(m.rasch, criterion=list("Waldtest"), alpha=0.05)
```

Eliminated item - Step 1: RV17  
 Eliminated item - Step 2: RV15  
 Eliminated item - Step 3: RV18  
 Eliminated item - Step 4: RV14

Results for stepwise item elimination:  
 Number of steps: 4  
 Criterion: Waldtest

	z-statistic	p-value
Step 1: RV17	4.798	0.000
Step 2: RV15	2.858	0.004
Step 3: RV18	4.158	0.000
Step 4: RV14	3.895	0.000

En principio, parece que dada la tendencia a la significación del estadístico basado en  $\chi^2$  con muestras grandes y la mayor potencia de los estadísticos LR de Andersen y de Wald, los ítems RV14, RV15, RV17 y RV18 son claros candidatos a su eliminación de esta escala. No obstante, antes de tomar una decisión emplearemos también las medias de cuadrados ponderada y no ponderada para probar el ajuste de los ítems al modelo de Rasch.

## 5.4 Evaluación del ajuste con las medias de cuadrados

Las medias de cuadrados de residuales (INFIT/OUTFIT), también se pueden utilizar para examinar el ajuste de los ítems al modelo de Rasch. Para calcular estos estadísticos es preciso obtener previamente las estimaciones de los parámetros de habilidad definiendo un nuevo objeto al que le asignamos esas estimaciones:

```
hab <- person.parameter(m.rasch)
```

Una vez que se dispone del objeto donde se almacenan las estimaciones de los parámetros de habilidad se puede estudiar el ajuste de los ítems con las medias de cuadrados de residuales. Para ello, se utiliza:

```
itemfit(hab)
```

Itemfit Statistics:

	Chisq	df	p-value	Outfit MSQ	Infit MSQ	Outfit t	Infit t	Discrim
RV1	324.247	400	0.998	0.809	0.859	-0.774	-1.125	0.351
RV2	316.240	400	0.999	0.789	0.922	-1.496	-0.964	0.393
RV3	304.291	400	1.000	0.759	0.878	-2.053	-1.771	0.476
RV4	398.622	400	0.510	0.994	1.006	-0.005	0.110	0.311
RV5	336.624	400	0.991	0.839	0.896	-2.241	-2.098	0.493
RV6	383.273	400	0.718	0.956	0.953	-0.556	-0.898	0.424
RV7	396.137	400	0.545	0.988	0.994	-0.132	-0.103	0.375
RV8	387.156	400	0.668	0.965	0.972	-0.508	-0.586	0.395
RV9	333.896	400	0.993	0.833	0.880	-2.408	-2.492	0.526
RV10	379.835	400	0.759	0.947	0.945	-0.352	-0.721	0.369
RV11	296.076	400	1.000	0.738	0.820	-2.902	-3.179	0.573
RV12	397.808	400	0.522	0.992	1.008	-0.053	0.165	0.346
RV13	311.640	400	1.000	0.777	0.906	-1.761	-1.269	0.438
RV14	468.381	400	0.010	1.168	1.137	2.611	3.082	0.198
RV15	521.553	400	0.000	1.301	1.149	4.272	3.351	0.153
RV16	360.106	400	0.925	0.898	0.920	-1.709	-1.853	0.453
RV17	527.405	400	0.000	1.315	1.158	3.338	3.130	0.080
RV18	679.898	400	0.000	1.696	0.992	3.661	-0.065	0.019

Esta función produce una salida en la que aparece para cada ítem un estadístico de ajuste

basado en la distribución  $\chi^2$  y su correspondiente prueba de significación (**p-value**). Este valor resulta de multiplicar el tamaño muestral  $N$  por el numerador de la media de cuadrados **OUTFIT**. Además, en la tabla aparece el estadístico de media de cuadrados no ponderada (**Outfit MSQ**), el estadístico de media de cuadrados ponderada (**Infit MSQ**) y sus correspondientes transformaciones estandarizadas  $t$  de Wilson-Hifferty (**Outfit t** e **Infit t**), y finalmente el índice de discriminación de la teoría clásica de tests.

Generalmente, el estadístico de  $\chi^2$  lleva asociado un valor de  $p$  que indicará la significación del valor encontrado en cada ítem. Si  $p < 0.05$ , entonces el ítem no se ajusta el modelo de Rasch. En el caso de las medias de cuadrados, ambas siguen una distribución  $\chi^2$  aproximada con media igual a 1. En este caso, los ítems RV14, RV15, RV17 y RV18 han obtenido resultados significativos, por lo que procedería su eliminación.

Si se utilizan los estadísticos basados en medias de cuadrados de residuales, el intervalo para rechazar el ajuste del ítem en la media de cuadrados **INFIT** es  $1 \pm 2\sqrt{N}$ , donde  $N$  es el tamaño del grupo (400). En este caso, el intervalo resultante es  $[0.9-1.1]$ . Para la media de cuadrados **OUTFIT** el intervalo es  $1 \pm 6\sqrt{N}$  que resulta en los límites  $[0.7-1.3]$ . Atendiendo a la regla de que se rechazan sólo aquellos ítems cuya media de cuadrados sea superior al límite superior del intervalo, con la media de cuadrados **INFIT** se rechazarían los ítems RV14, RV15 y RV17, y con la media de cuadrados **OUTFIT** se rechazarían los ítems RV15, RV17 y RV18. Con respecto a las transformaciones  $t$ , el intervalo confidencial para considerar que el ítem ajusta el modelo de Rasch viene definido por  $\pm 2$ . En este caso, parece que los ítems RV14, RV15, RV17 y RV18 no siguen el modelo de Rasch.

## Capítulo 6

# Parámetros de habilidad

### 6.1 Estimación de parámetros

Los parámetros de habilidad para todas las puntuaciones en este test se pueden obtener simplemente pidiendo a R que despliegue el objeto `hab` que las contiene:

```
hab
```

Person Parameters:

Raw Score	Estimate	Std.Error
2	-2.44012595	0.7815549
3	-1.92279393	0.6681247
4	-1.52033799	0.6055499
5	-1.17809001	0.5672141
6	-0.87088904	0.5430850
7	-0.58448698	0.5285606
8	-0.30948023	0.5213579
9	-0.03867997	0.5204071
10	0.23423723	0.5254312
11	0.51574464	0.5368353
12	0.81346864	0.5558171
13	1.13765396	0.5847822
14	1.50391087	0.6283986
15	1.93952818	0.6965292
16	2.50226946	0.8147375
17	3.36728937	1.0867453
18	4.30532631	NA

En esta tabla aparecen las estimaciones de los parámetros de habilidad para cada puntuación empírica y su correspondiente error típico. Nótese que en el modelo de Rasch, la estimación de la habilidad es igual para una puntuación empírica concreta independientemente de los ítems que hayan sido contestados. Para obtener los parámetros de habilidad de cada persona se emplea la función<sup>1</sup>:

---

<sup>1</sup>Algunas funciones del paquete `eRm` producen salidas muy extensas, por lo que se han empleado las funciones `head(objeto)` o `summary(objeto)` para acotar la salida.

```
head(round(cbind(hab$thetapar$NAGroup1, hab$se.theta$NAGroup1), 3))
```

```
  [,1] [,2]
1 1.940 0.697
2 1.138 0.585
3 2.502 0.815
4 1.504 0.628
5 1.504 0.628
6 3.367 1.087
```

que produce un listado de las estimaciones de los parámetros de habilidad para las 400 personas del archivo `rv.dat`. Si deseamos conocer las probabilidades esperadas de acuerdo con el modelo de Rasch tenemos que escribir:

```
head(pmat(hab))
```

```
      RV1      RV2      RV3      RV4      RV5      RV6      RV7
1 0.9800527 0.9536604 0.9399935 0.9399935 0.8680855 0.8738394 0.8738394
2 0.9565880 0.9022464 0.8753971 0.8753971 0.7469219 0.7564781 0.7564781
3 0.9885387 0.9730657 0.9649114 0.9649114 0.9203328 0.9240073 0.9240073
4 0.9694953 0.9301304 0.9101774 0.9101774 0.8097704 0.8175333 0.8175333
5 0.9694953 0.9301304 0.9101774 0.9101774 0.8097704 0.8175333 0.8175333
6 0.9951421 0.9884799 0.9849199 0.9849199 0.9648347 0.9665313 0.9665313
      RV8      RV9      RV10     RV11     RV12     RV13     RV14
1 0.8336774 0.8621811 0.9465613 0.9124521 0.9065497 0.9456438 0.7515166
2 0.6921187 0.7372358 0.8881942 0.8237663 0.8131094 0.8863948 0.5756260
3 0.8979507 0.9165422 0.9688424 0.9481762 0.9445359 0.9682946 0.8415036
4 0.7642819 0.8018517 0.9197300 0.8708317 0.8625460 0.9183915 0.6617494
5 0.7642819 0.8018517 0.9197300 0.8708317 0.8625460 0.9183915 0.6617494
6 0.9543348 0.9630766 0.9866403 0.9775050 0.9758725 0.9864010 0.9265238
      RV15     RV16     RV17     RV18
1 0.7129116 0.7845569 0.5817203 0.3327577
2 0.5268973 0.6202355 0.3841347 0.1827819
3 0.8134080 0.8647322 0.7094225 0.4667996
4 0.6163182 0.7019927 0.4735802 0.2439110
5 0.6163182 0.7019927 0.4735802 0.2439110
6 0.9119217 0.9382072 0.8529089 0.6752490
```

## 6.2 Detección de patrones aberrantes

Los estadísticos de ajuste basados en  $\chi^2$  de medias de cuadrados de residuales y transformaciones Wilson-Hifferty también se pueden utilizar para evaluar si los patrones de respuesta de las personas se ajustan al modelo de Rasch. Para ello, se debe emplear:

```
pfit <- personfit(hab)
head(pfit$p.infitMSQ, 5)
```

```
      1      2      3      4      5
0.7350456 0.7775190 0.7366459 0.7245614 1.6515984
```

```
head(pfit$p.outfitMSQ, 5)
```

```
      1      2      3      4      5
0.6105044 0.6752679 0.4292343 0.5372264 2.0287802
```

En este caso, los criterios de ajuste de las medias de cuadrados de residuales no están claros, aunque en la literatura se suele utilizar el criterio de que las personas presentan patrones aberrantes si las medias de cuadrados se encuentran fuera del intervalo [0.5-1.5]. También se puede saber cuántas personas presentan patrones no esperados por el modelo de Rasch utilizando:

```
PersonMisfit(hab)
```

Percentage of Misfitting Persons: 1.7456 %

En este caso un 1.7544% de los examinados no presentan patrones de respuesta acordes a lo esperado por el modelo de Rasch. Se puede solicitar un resumen con:

```
summary(PersonMisfit(hab))
```

Percentage of Misfitting Persons:

1.7456 %

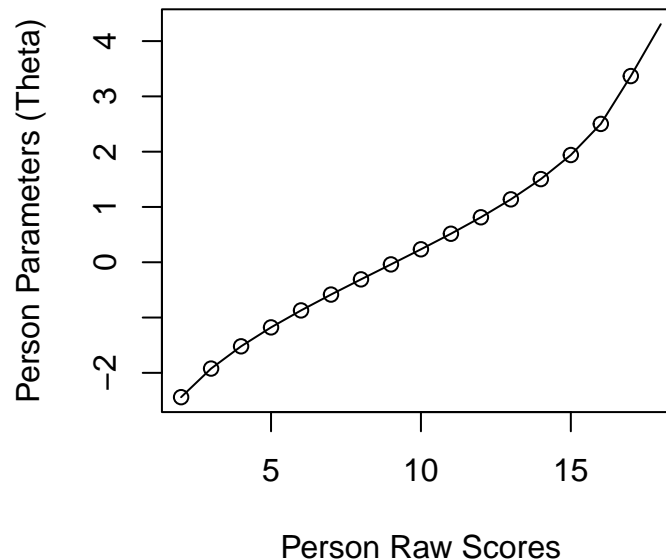
(i.e., 7 out of 401 persons have Chi-square based Z-values > 1.96)

entonces aparecerán los niños que contienen patrones aberrantes; en este caso 7 de 399 casos.

El diagrama de dispersión de los parámetros de habilidad vs. las puntuaciones empíricas se obtiene con<sup>2</sup>:

```
plot(hab)
```

### Plot of the Person Parameters



<sup>2</sup>También se puede emplear `plot(person.parameter(m.rasch))`.

## Capítulo 7

# Índice de separación

La fiabilidad de la separación de las personas se determina con la función `SepRel()`. Para obtener toda la información de esta función conviene enviar el resultado a un objeto y resumirla con `summary()`:

```
sep <- SepRel(hab)
summary(sep)
```

```
Separation Reliability: 0.6794
```

```
Observed Variance: 1.15 (Squared Standard Deviation)
```

```
Mean Square Measurement Error: 0.3686 (Model Error Variance)
```

En este caso, la fiabilidad de la separación de las personas no ha resultado muy elevada. Un valor mínimo aceptable es de 0.80. No obstante, este resultado significa que todos los niños están en un rango medio de habilidad, por otra parte esperado por un test construido con la teoría clásica de tests.

## Capítulo 8

# Evaluación de los principios del modelo de Rasch

### 8.1 Unidimensionalidad de la escala

En el paquete `eRm` se puede obtener una evaluación de la unidimensionalidad con el estadístico de Martin-Loëf empleando:

```
MLoef(m.rasch)
```

```
Martin-Loef-Test (split criterion: median)
LR-value: 63.972
Chi-square df: 79
p-value: 0.89
```

Supuesto que el valor de  $p=0.89$  no ha resultado significativo, se puede afirmar que la escala formada por los 18 ítems es unidimensional. No obstante, en la literatura sobre el modelo de Rasch se suele emplear también con frecuencia el Análisis de Componentes Principales de residuales para evaluar la unidimensionalidad de la escala.

El paquete `eRm` no dispone de una función para realizar este análisis, pero se puede utilizar la función `prcomp()` de R para realizar el análisis de componentes principales, aunque previamente habrá que determinar la matriz de residuales. Para ello, empleamos:

```
residual <- residuals(person.parameter(m.rasch))
summary(prcomp(residual, scale=T))
```

Importance of components:

	PC1	PC2	PC3	PC4	PC5	PC6	PC7
Standard deviation	1.24591	1.19159	1.14546	1.11882	1.08042	1.07591	1.05611
Proportion of Variance	0.08624	0.07888	0.07289	0.06954	0.06485	0.06431	0.06197
Cumulative Proportion	0.08624	0.16512	0.23801	0.30756	0.37241	0.43672	0.49868
	PC8	PC9	PC10	PC11	PC12	PC13	PC14
Standard deviation	1.0279	1.0147	1.0138	0.99256	0.96404	0.93448	0.92806
Proportion of Variance	0.0587	0.0572	0.0571	0.05473	0.05163	0.04851	0.04785
Cumulative Proportion	0.5574	0.6146	0.6717	0.72642	0.77805	0.82657	0.87442
	PC15	PC16	PC17	PC18			
Standard deviation	0.88728	0.86695	0.82610	0.19793			
Proportion of Variance	0.04374	0.04176	0.03791	0.00218			

Cumulative Proportion 0.91815 0.95991 0.99782 1.00000

R devuelve en realidad las desviaciones típicas de los componentes principales. Para obtener los eigenvalores se debe calcular elevar al cuadrado cada desviación típica. Así, el primer eigenvalor será  $1.24591^2 = 1.55229$ , que obviamente no llega al valor de 2.0 establecido como puntuación de corte, por lo que se puede afirmar que la escala es unidimensional.

## 8.2 Independencia local de los ítems

La independencia local se prueba a través de las correlaciones entre los residuales con los ítems seleccionados. Existe independencia local de los ítems si estas correlaciones no sobrepasan el valor de 0.20. Para ello, empleamos:

```
cor(residual)
```

	RV1	RV2	RV3	RV4	RV5
RV1	1.000000000	-0.107467039	-0.019295000	-0.056479224	-0.007903367
RV2	-0.107467039	1.000000000	-0.010259060	-0.047988823	-0.032068152
RV3	-0.019295000	-0.010259060	1.000000000	-0.053058396	0.013556499
RV4	-0.056479224	-0.047988823	-0.053058396	1.000000000	-0.107319103
RV5	-0.007903367	-0.032068152	0.013556499	-0.107319103	1.000000000
RV6	-0.034159885	-0.077288327	-0.043669723	0.039321318	-0.033865168
RV7	-0.100366161	-0.003282836	-0.045023766	-0.055679851	-0.074045653
RV8	-0.084164966	-0.049135652	-0.001983614	-0.076176407	-0.041722336
RV9	0.119261540	-0.139531682	-0.050516329	-0.076848499	-0.061635562
RV10	-0.018702426	0.035269307	-0.032378263	-0.076525041	-0.119546690
RV11	-0.075056543	-0.034944861	0.011111488	-0.080926186	0.010304579
RV12	-0.038951380	-0.066139167	-0.077981473	-0.005161951	-0.018520084
RV13	0.002790977	-0.046928811	-0.064097629	-0.008373846	-0.128742287
RV14	-0.091157071	-0.019396894	-0.096054130	-0.053314353	-0.069401603
RV15	0.047699019	-0.019285740	-0.111483204	-0.066238320	-0.063898452
RV16	-0.022365017	-0.068928520	0.038696106	-0.124170922	-0.094050349
RV17	-0.087549552	-0.043955333	-0.120393362	-0.050470728	-0.054374915
RV18	0.027545586	-0.082756980	-0.104331636	-0.012107500	-0.042863274
	RV6	RV7	RV8	RV9	RV10
RV1	-0.034159885	-0.100366161	-0.084164966	0.119261540	-0.018702426
RV2	-0.077288327	-0.003282836	-0.049135652	-0.139531682	0.035269307
RV3	-0.043669723	-0.045023766	-0.001983614	-0.050516329	-0.032378263
RV4	0.039321318	-0.055679851	-0.076176407	-0.076848499	-0.076525041
RV5	-0.033865168	-0.074045653	-0.041722336	-0.061635562	-0.119546690
RV6	1.000000000	-0.122569395	-0.044407150	-0.105524133	-0.039395474
RV7	-0.122569395	1.000000000	-0.095922620	0.046873011	0.011962181
RV8	-0.044407150	-0.095922620	1.000000000	-0.116653665	-0.093191986
RV9	-0.105524133	0.046873011	-0.116653665	1.000000000	-0.086406611
RV10	-0.039395474	0.011962181	-0.093191986	-0.086406611	1.000000000
RV11	-0.019283728	-0.145253606	-0.072333624	0.059147258	0.027933957
RV12	-0.081759778	-0.097319420	-0.042308484	-0.082725297	-0.102362689
RV13	-0.071251856	-0.104695841	0.012600279	-0.004373575	0.011506302
RV14	-0.032554721	-0.077231876	-0.013872589	-0.160796865	-0.091799510
RV15	-0.145105372	0.002693397	-0.154277342	-0.102307026	-0.025459358
RV16	-0.136700770	-0.020825401	-0.038384500	0.083996516	-0.004133227
RV17	-0.004253375	-0.110527578	-0.074585866	-0.141042756	-0.065294724

	RV11	RV12	RV13	RV14	RV15
RV18	-0.103403189	-0.064321357	-0.097482630	-0.087092529	-0.140893853
RV1	-0.07505654	-0.038951380	0.002790977	-0.09115707	0.047699019
RV2	-0.03494486	-0.066139167	-0.046928811	-0.01939689	-0.019285740
RV3	0.011111149	-0.077981473	-0.064097629	-0.09605413	-0.111483204
RV4	-0.08092619	-0.005161951	-0.008373846	-0.05331435	-0.066238320
RV5	0.01030458	-0.018520084	-0.128742287	-0.06940160	-0.063898452
RV6	-0.01928373	-0.081759778	-0.071251856	-0.03255472	-0.145105372
RV7	-0.14525361	-0.097319420	-0.104695841	-0.07723188	0.002693397
RV8	-0.07233362	-0.042308484	0.012600279	-0.01387259	-0.154277342
RV9	0.05914726	-0.082725297	-0.004373575	-0.16079686	-0.102307026
RV10	0.02793396	-0.102362689	0.011506302	-0.09179951	-0.025459358
RV11	1.00000000	-0.110502398	0.078377058	-0.12120482	-0.146778230
RV12	-0.11050240	1.000000000	0.017990510	-0.02182390	-0.082387672
RV13	0.07837706	0.017990510	1.000000000	-0.16529471	-0.044929420
RV14	-0.12120482	-0.021823895	-0.165294706	1.00000000	-0.081123391
RV15	-0.14677823	-0.082387672	-0.044929420	-0.08112339	1.000000000
RV16	0.04917505	-0.136006223	-0.060628380	-0.05272626	-0.202195503
RV17	-0.15864159	-0.069682781	-0.078393778	-0.03641552	-0.051470094
RV18	-0.05027606	0.011515374	-0.070926865	-0.06576174	0.113401917

	RV16	RV17	RV18
RV1	-0.022365017	-0.087549552	0.02754559
RV2	-0.068928520	-0.043955333	-0.08275698
RV3	0.038696106	-0.120393362	-0.10433164
RV4	-0.124170922	-0.050470728	-0.01210750
RV5	-0.094050349	-0.054374915	-0.04286327
RV6	-0.136700770	-0.004253375	-0.10340319
RV7	-0.020825401	-0.110527578	-0.06432136
RV8	-0.038384500	-0.074585866	-0.09748263
RV9	0.083996516	-0.141042756	-0.08709253
RV10	-0.004133227	-0.065294724	-0.14089385
RV11	0.049175047	-0.158641587	-0.05027606
RV12	-0.136006223	-0.069682781	0.01151537
RV13	-0.060628380	-0.078393778	-0.07092686
RV14	-0.052726256	-0.036415523	-0.06576174
RV15	-0.202195503	-0.051470094	0.11340192
RV16	1.000000000	-0.068314983	-0.05166478
RV17	-0.068314983	1.000000000	-0.02547750
RV18	-0.051664784	-0.025477497	1.00000000

Alternativamente, se puede calcular el estadístico Q3 de Yen disponible en el paquete `subscore`. Previamente debemos instalar el paquete `install.packages("subscore")` y cargar la librería correspondiente:

```
library(subscore)
```

```
Cargando paquete requerido: CTT
```

```
Adjuntando el paquete: 'CTT'
```

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

```
polyserial
```

Cargando paquete requerido: irtoys

Cargando paquete requerido: sm

Package 'sm', version 2.2-6.0: type help(sm) for summary information

Adjuntando el paquete: 'sm'

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

muscle

Cargando paquete requerido: sirt

- sirt 4.1-15 (2024-02-06 00:05:40)

```
summary(Yen.Q3(rv, IRT.model="1pl"))
```

There are 0 Q3 statistics exceed 0.2. There are 0 Q3 weighted statistics exceed 0.2.

	Length	Class	Mode
Q3	324	-none-	numeric
Q3.weighted	324	-none-	numeric

```
detach(package:subscore)
```

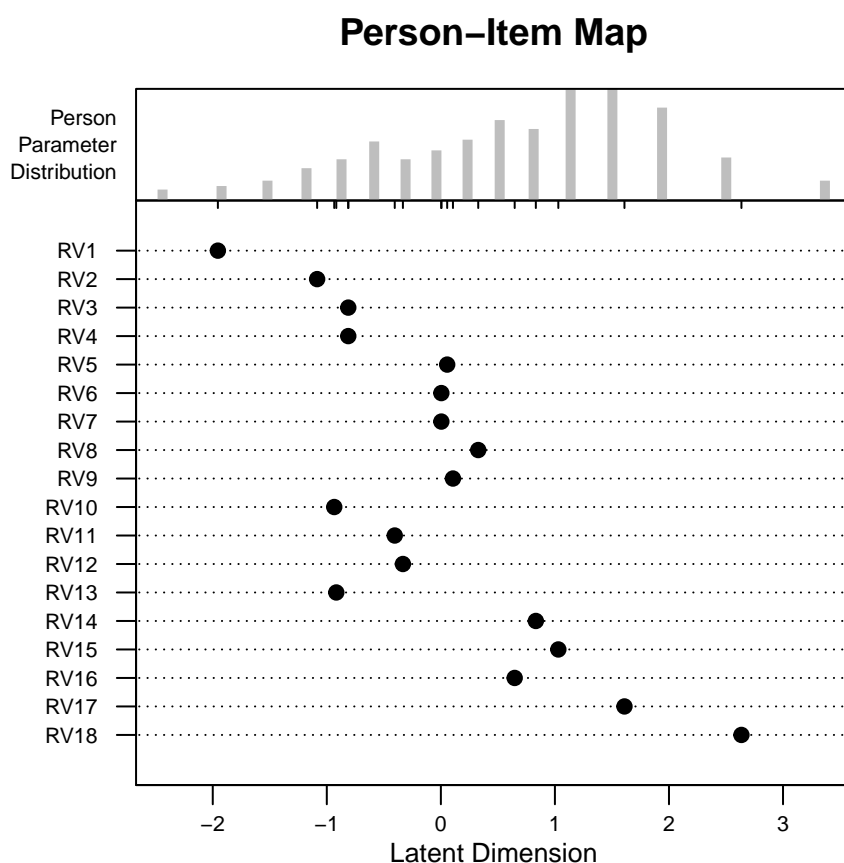
No se ha encontrado ningún estadístico Q3 que esté por encima de 0.2, por lo que se puede afirmar que los ítems son localmente independientes en esta escala.

## Capítulo 9

# Gráficos en el paquete eRm

Una vez que se han estimado los parámetros procede examinar gráficamente el ajuste de las respuestas a los ítems con el modelo de Rasch. Para ello, se debe construir el mapa de ítems-personas con:

```
plotPImap(m.rasch)
```

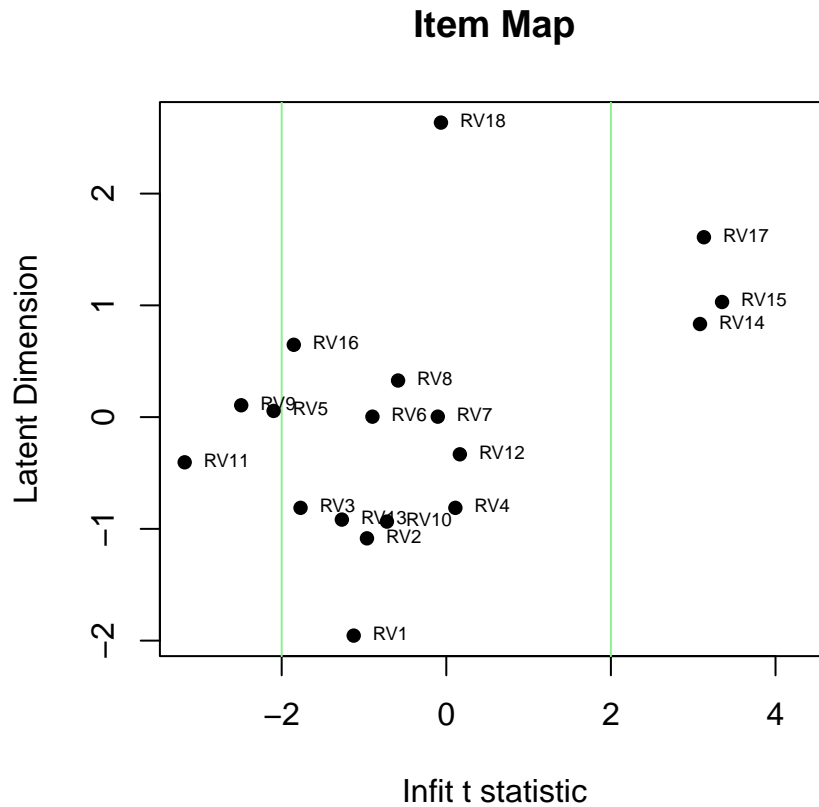


El mapa ítems-personas permite examinar la distribución de los parámetros de los ítems en función de la distribución de la habilidad. Si los ítems cubren todo el continuo de habilidad es un indicador que el modelo de Rasch es adecuado, en principio, para estimar la dificultad de los ítems en este grupo.

El paquete eRm también dispone de la función `plotPWmap(objeto)` que permite identificar

ítems o personas que no se ajustan al modelo, graficando los parámetros correspondientes contra su estadísticos t-infit. Por ejemplo:

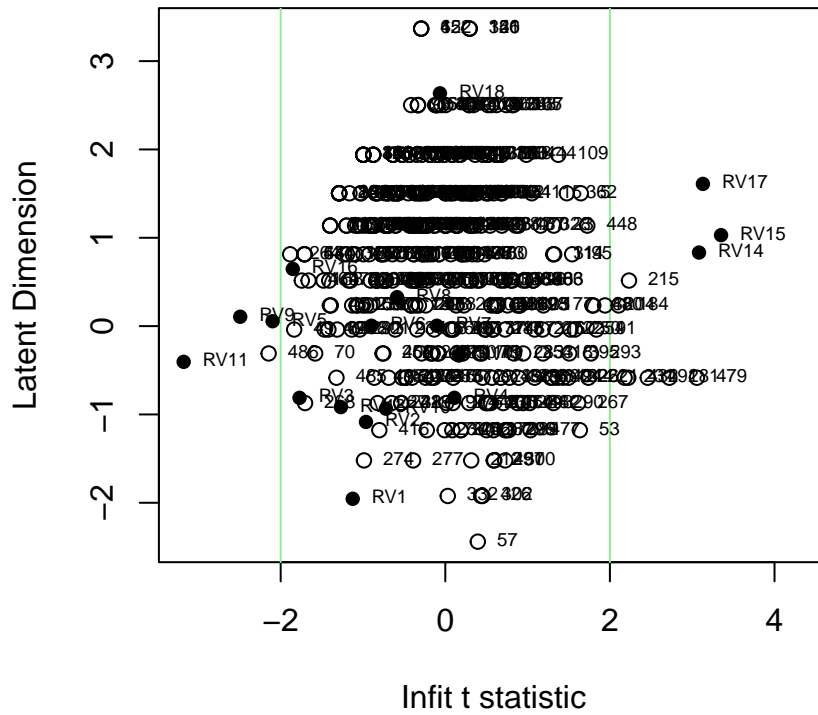
```
plotPWmap(m.rasch,imap=T)
```



En este caso, se aprecia los ítems que están fuera del intervalo  $[-2,+2]$  empleado por el estadístico t que sugieren que esos ítems no siguen el modelo de Rasch. El mismo gráfico se puede obtener para las personas con:

```
plotPWmap(m.rasch,pmap=T)
```

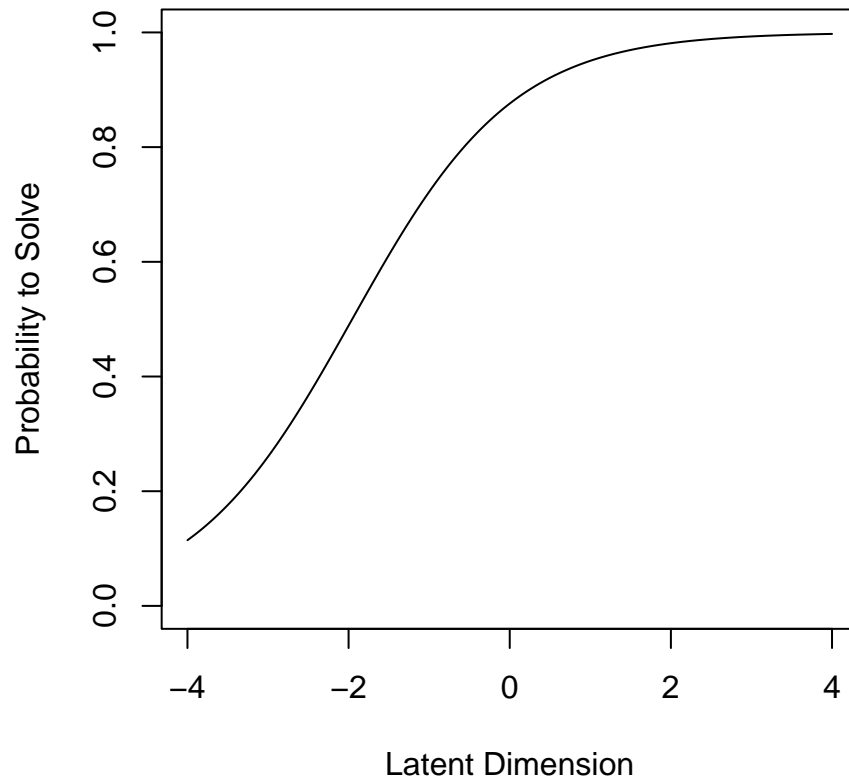
### Item/Person Map



No obstante, este gráfico es bastante más difícil de interpretar, sobre todo cuando el grupo es muy elevado, como en este caso. Ahora se puede representar gráficamente la FRI del primer ítem de este test con:

```
plotICC(m.rasch,item.subset=1)
```

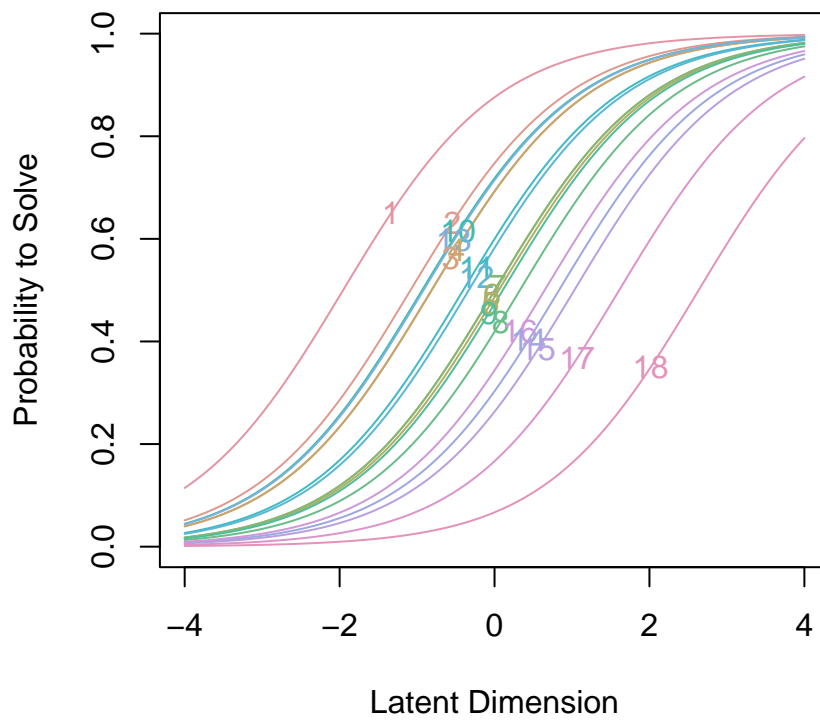
ICC plot for item RV1



Como se aprecia en la figura anterior, la FRI es monotonía creciente tal como pronostica el modelo de Rasch. Las FRIs de los 18 ítems también se pueden graficar conjuntamente con:

```
plotjointICC(m.rasch, legend=F)
```

## ICC plot

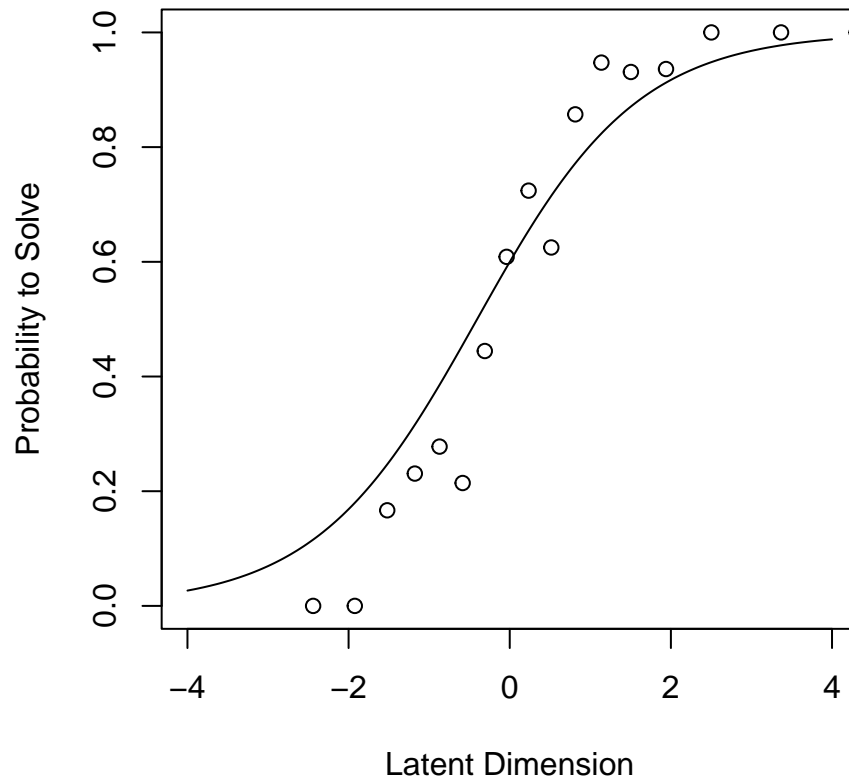


que como espera el modelo de Rasch, todas las FRI's son paralelas.

Para examinar el acuerdo de las frecuencias observadas y esperadas gráficamente se pueden obtener las FRI's empíricas junto a las teóricas según el modelo de Rasch con:

```
plotICC(m.rasch, item.subset=11, empICC=list("raw"))
```

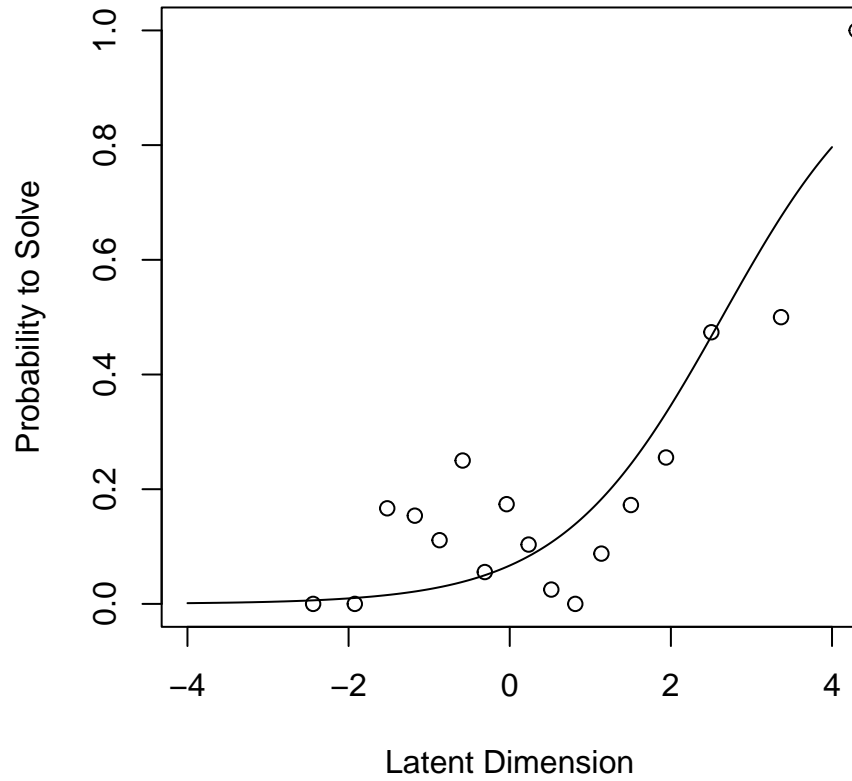
ICC plot for item RV11



En este caso, se observa que tanto el patrón empírico como el pronosticado por la función logística está de acuerdo con el modelo de Rasch, lo que es una buena evidencia de ajuste del ítem. Sin embargo, un patrón que muestra un fuerte desajuste entre las probabilidades empíricas y las pronosticadas por el modelo se puede observar en el ítem RV18.

```
plotICC(m.rasch, item.subset=18, empICC=list("raw"))
```

**ICC plot for item RV18**



En este gráfico se aprecia claramente que la FRI empírica es muy diferente de la pronosticada por el modelo de Rasch, mostrando un patrón aleatorio de respuesta e independiente del nivel de habilidad de los componentes del grupo.

## Capítulo 10

# Elaboración de un nuevo instrumento

Una vez que se han obtenido los estadísticos de ajuste parece que, considerando todos los estadísticos, los ítems RV14, RV15, RV17 y RV18 son los candidatos para su eliminación, por presentar estadísticos de ajuste que claramente indican un comportamiento peor de lo esperado por el modelo de Rasch. En este caso, procede su eliminación para comprobar si los ítems restantes siguen el modelo y repetir de nuevo la estimación de parámetros eliminados esos cuatro ítems. Previamente, se debe definir un nuevo objeto que contenga solo los ítems que se ajustaron al modelo de Rasch utilizando la siguiente estructura:

```
rv3 <- rv[, -c(14,15,17,18)]
```

En la nueva matriz de datos (rv3) ahora solo aparecen 14 ítems. El paso siguiente es aplicar el modelo de Rasch definiendo un nuevo objeto:

```
m.rasch1 <- RM(rv3)
```

y finalmente podemos examinar las estimaciones de los parámetros de los 14 ítems con:

```
summary(m.rasch1)
```

Results of RM estimation:

Call: RM(X = rv3)

Conditional log-likelihood: -1980.521

Number of iterations: 17

Number of parameters: 13

Item (Category) Difficulty Parameters (eta): with 0.95 CI:

	Estimate	Std. Error	lower CI	upper CI
RV2	-0.690	0.137	-0.958	-0.421
RV3	-0.403	0.129	-0.656	-0.149
RV4	-0.403	0.129	-0.656	-0.149
RV5	0.521	0.114	0.298	0.744
RV6	0.466	0.114	0.242	0.690
RV7	0.466	0.114	0.242	0.690
RV8	0.813	0.112	0.594	1.032

RV9	0.575	0.113	0.353	0.797
RV10	-0.532	0.133	-0.792	-0.272
RV11	0.029	0.120	-0.207	0.265
RV12	0.106	0.119	-0.128	0.339
RV13	-0.513	0.132	-0.772	-0.254
RV16	1.155	0.110	0.939	1.372

Item Easiness Parameters (beta) with 0.95 CI:

	Estimate	Std. Error	lower CI	upper CI
beta RV1	1.590	0.176	1.245	1.936
beta RV2	0.690	0.137	0.421	0.958
beta RV3	0.403	0.129	0.149	0.656
beta RV4	0.403	0.129	0.149	0.656
beta RV5	-0.521	0.114	-0.744	-0.298
beta RV6	-0.466	0.114	-0.690	-0.242
beta RV7	-0.466	0.114	-0.690	-0.242
beta RV8	-0.813	0.112	-1.032	-0.594
beta RV9	-0.575	0.113	-0.797	-0.353
beta RV10	0.532	0.133	0.272	0.792
beta RV11	-0.029	0.120	-0.265	0.207
beta RV12	-0.106	0.119	-0.339	0.128
beta RV13	0.513	0.132	0.254	0.772
beta RV16	-1.155	0.110	-1.372	-0.939

Para obtener los estadísticos de ajuste es preciso estimar los parámetros de las personas definiendo un nuevo objeto escribiendo:

```
hab1 <- person.parameter(m.rasch1)
itemfit(hab1)
```

Itemfit Statistics:

	Chisq	df	p-value	Outfit MSQ	Infit MSQ	Outfit t	Infit t	Discrim
RV1	316.339	371	0.982	0.850	0.897	-0.531	-0.831	0.336
RV2	346.929	371	0.810	0.933	0.986	-0.391	-0.142	0.354
RV3	286.330	371	1.000	0.770	0.904	-1.857	-1.356	0.479
RV4	436.498	371	0.011	1.173	1.077	1.289	1.065	0.279
RV5	348.393	371	0.795	0.937	0.979	-0.792	-0.359	0.432
RV6	420.957	371	0.037	1.132	1.032	1.582	0.581	0.381
RV7	411.572	371	0.072	1.106	1.078	1.294	1.375	0.326
RV8	409.301	371	0.083	1.100	1.056	1.376	1.062	0.337
RV9	320.191	371	0.973	0.861	0.922	-1.864	-1.447	0.503
RV10	375.908	371	0.419	1.011	0.987	0.121	-0.150	0.367
RV11	281.716	371	1.000	0.757	0.844	-2.557	-2.605	0.568
RV12	449.863	371	0.003	1.209	1.104	2.033	1.654	0.294
RV13	310.745	371	0.990	0.835	0.939	-1.195	-0.802	0.433
RV16	354.766	371	0.719	0.954	0.972	-0.642	-0.558	0.392

En este caso, el ítem RV12 ha obtenido una media de cuadrados INFIT por encima de 1.1, y un estadístico t Outfit por encima de +2, por lo que procede eliminar este ítem y repetir el análisis Rasch. Dejamos este proceso para el lector.

## Capítulo 11

# Evaluación del DIF con los métodos de Andersen y Wald

La evaluación del DIF se puede realizar en `eRm` a dos niveles. Por un lado, el estadístico LR de razón de verosimilitud de Andersen permite evaluar el DIF en el test completo y, por otro lado, el estadístico de Wald permite evaluar el DIF ítem a ítem. En este caso, la división de los grupos no se realizará en función de la mediana o de la media sino con la variable externa (`genero`) que contiene la definición del género de los niños.

Para evaluar el DIF globalmente con el estadístico de Andersen emplearemos:

```
LRtest(m.rasch, splitcr=genero)
```

```
Andersen LR-test:  
LR-value: 14.206  
Chi-square df: 17  
p-value: 0.652
```

El resultado del estadístico informa que no existe DIF ( $p=0.652$ ) en el test completo. No obstante, es conveniente examinar el DIF ítem a ítem con el estadístico de Wald:

```
Waldtest(m.rasch, splitcr=genero)
```

```
Wald test on item level (z-values):
```

	z-statistic	p-value
beta RV1	-0.184	0.854
beta RV2	0.248	0.804
beta RV3	-0.203	0.839
beta RV4	-0.474	0.635
beta RV5	0.890	0.374
beta RV6	-1.026	0.305
beta RV7	0.142	0.887
beta RV8	0.237	0.812
beta RV9	-0.446	0.655
beta RV10	-2.131	0.033
beta RV11	1.946	0.052
beta RV12	-1.074	0.283

beta RV13	-0.309	0.757
beta RV14	-0.064	0.949
beta RV15	0.433	0.665
beta RV16	-0.013	0.989
beta RV17	1.362	0.173
beta RV18	1.015	0.310

Parece que los ítems RV10 y RV11 presentan cierto DIF, por lo que deben examinarse con detenimiento por si es preciso su eliminación.