

---

MEDIDAS DE ACUERDO NOMINAL ENTRE OBSERVADORES:  
UNA IMPLEMENTACIÓN EN R

---



Autora: Ana Devaki Maldonado González  
Tutor: Pedro Femia Marzo

---

Máster en Estadística Aplicada  
Universidad de Granada  
Julio 2022

## Agradecimientos

Me gustaría agradecer a mi tutor, Pedro Femia, su dedicación y esfuerzo para que este trabajo salga adelante. He disfrutado mucho diseñando el paquete DeltaMAN, pero sin el material proporcionado por Pedro, no habría podido realizar la implementación con éxito. ¡Mil gracias!

Resumen . . . . .	iii
<b>1 Introducción</b>	<b>1</b>
<b>2 Índices de acuerdo corregidos por azar</b>	<b>3</b>
2.1 Coeficiente kappa de Cohen . . . . .	3
2.2 Modelo Delta . . . . .	7
2.2.1 Formulación del modelo . . . . .	8
2.2.2 Estimación de parámetros . . . . .	10
2.2.2.1 Existencia y unicidad de solución . . . . .	11
2.2.2.2 Bondad del ajuste . . . . .	12
2.2.3 Medidas de acuerdo corregidas por azar . . . . .	12
2.2.4 Estimación asintótica para el caso $K = 2$ . . . . .	15
2.3 Comparación entre $\kappa$ y $\Delta$ . . . . .	16
<b>3 El paquete DeltaMAN</b>	<b>18</b>
3.1 La función Kappa() . . . . .	18
3.2 La función Delta() . . . . .	20
3.2.1 Depuración de los datos de entrada: getM1() . . . . .	22
3.2.2 Obtención de la constante desconocida B: getB() . . . . .	26
3.2.3 Cálculo de los parámetros del modelo: get_parameters() . . . . .	27
3.2.4 Verificación del modelo: chisq.test.Delta() . . . . .	28

---

3.2.5	Cálculo de varianzas y covarianzas: <code>var.Delta()</code> y <code>cov.Delta()</code> . . . . .	29
3.2.6	Estimación de las medidas de acuerdo: <code>get_measures()</code> . . . . .	30
3.2.7	Visualización de resultados: <code>print()</code> y <code>summary()</code> . . . . .	32
3.2.8	Análisis asintótico: <code>asymptoticDelta()</code> . . . . .	33
3.3	La función <code>multiDelta()</code> . . . . .	35
3.4	La función <code>runDeltaShinyApp()</code> . . . . .	37
<b>4</b>	<b>Conclusiones</b>	<b>42</b>
	<b>Bibliografía</b>	<b>44</b>

Cuando dos observadores clasifican de forma independiente un objeto como perteneciente a una categoría, la clasificación puede ser coincidente bien porque se ha reconocido el objeto, o bien por azar. En este sentido, surge la necesidad de medir el grado de acuerdo no debido al azar entre los observadores. Uno de los índices de acuerdo más extendido es el coeficiente kappa de Cohen, que tiene la ventaja de ser muy sencillo de calcular. Sin embargo, su comportamiento es poco realista cuando las distribuciones marginales son muy asimétricas. El índice de acuerdo nominal Delta, formulado originalmente por Martín y Femia en 2004, surge como alternativa al índice kappa y ha sido utilizado en diversas disciplinas desde entonces.

El objetivo de este trabajo es implementar el cálculo de Delta en un paquete de R. Actualmente existen dos aplicaciones para realizar todos los cálculos relativos a Delta: una aplicación ejecutable solo en el sistema operativo Windows y una página web. El interés de crear el paquete de R es incorporar el cálculo de Delta como rutina del lenguaje estadístico de código abierto con mayor aceptación en la actualidad. El paquete, llamado DeltaMAN, ya se encuentra disponible en el CRAN, el repositorio central de R respaldado por The R Foundation. Además, se ha creado una aplicación de shiny que sirve como primera toma de contacto con la funcionalidad del paquete. Esta aplicación se encuentra dentro de DeltaMAN y además en el servidor shinyapps.io, por lo que se puede acceder a ella desde cualquier navegador y dispositivo.

## Abstract

When two raters independently classify an object as belonging to a particular class, the classification might be concordant due to either object recognition or pure chance. In that sense, it is necessary to measure the level of agreement not due to chance between the raters. One of the most widespread coefficients to measure inter-rater agreement is the so-called Cohen's kappa coefficient, which is very easy to calculate. However, kappa's performance is poor when the marginal distributions are very asymmetric. On the other hand, the Delta measurement of agreement for nominal data, developed by Martín and Femia in 2004, was proposed as an alternative to the kappa coefficient and has been applied to several disciplines ever since.

The goal of this work is to implement the Delta algorithm in an R package. Nowadays, there exists two applications that perform the Delta analysis: an executable application that can only be used on the Windows Operating System, and a website. The interest in creating an R package is to incorporate the Delta model as a routine of the most popular, open source language for statistical analysis. Our package, named DeltaMAN, has been submitted to CRAN repository and is already available. Moreover, a shiny app has been developed to provide a first contact to the package. This application is available within DeltaMAN and has also been deployed to shinyapps.io server, so it can be accessed from any browser and device.

Cuando dos observadores clasifican un grupo de objetos en dos o más categorías se puede estudiar la concordancia o acuerdo clasificatorio entre los dos observadores. Los datos de clasificación se pueden disponer en una tabla de doble entrada que exprese la frecuencia absoluta conjunta con que los dos observadores clasifican los objetos. Si llamamos  $\mathbf{R}$  al observador por filas y  $\mathbf{C}$  al observador por columnas, la tabla de clasificación para  $n$  objetos en  $K$  categorías es la mostrada en la [Tabla 1.1](#).

**Tabla 1.1** Frecuencias de clasificación ( $x_{ij}$ ) de  $n$  objetos en  $K$  categorías realizadas por dos observadores,  $\mathbf{R}$  y  $\mathbf{C}$ .

	$\mathbf{C}_1$	...	$\mathbf{C}_j$	...	$\mathbf{C}_K$	Total
$\mathbf{R}_1$	$x_{11}$	...	$x_{1j}$	...	$x_{1K}$	$r_1$
$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$
$\mathbf{R}_i$	$x_{i1}$	...	$x_{ij}$	...	$x_{iK}$	$r_i$
$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$
$\mathbf{R}_K$	$x_{K1}$	...	$x_{Kj}$	...	$x_{KK}$	$r_K$
Total	$c_1$	...	$c_j$	...	$c_K$	$\mathbf{n}$

El acuerdo observado entre ellos expresa el porcentaje de coincidencias en la clasificación en relación al total de objetos examinados, esto es,  $\frac{\sum x_{ii}}{n} \times 100$ . Sin embargo, parte del acuerdo observado puede ser debido al azar y no porque el criterio de ambos observadores coincida. Cuando el objeto es perfectamente reconocido por un observador, este será clasificado sin error. En cambio, cuando el observador no reconoce el objeto con claridad, este podría clasificarlo por azar. Como consecuencia, las clasificaciones de dos observadores que actúan

de forma independiente y al azar pueden ser coincidentes para un cierto número de objetos simplemente por azar. Por lo tanto, se hace necesario medir el grado de acuerdo no debido al azar al clasificar una serie de objetos.

El coeficiente kappa de Cohen [1] es una de las medidas de acuerdo corregido por azar más utilizadas debido a su facilidad de cálculo y a que siempre es válido, ya que no está basado en ningún modelo subyacente. Sin embargo, esta medida tiene un mal comportamiento cuando las frecuencias marginales están desbalanceadas.

El índice de acuerdo nominal Delta entre dos observadores fue propuesto en 2004 por Antonio Martín y Pedro Femia [2] con el objetivo de mejorar el clásico índice kappa de Cohen. El problema principal que plantea Delta es su cálculo, que no se puede hacer a mano, y que es válido siempre que se verifique el modelo asumido. En general, las estimaciones de Delta y kappa son similares, salvo cuando los marginales están muy desbalanceados, en cuyo caso Delta refleja la realidad mejor que kappa. Por otro lado, el índice Delta es solo válido para cualidades nominales, a diferencia de kappa que se puede generalizar fácilmente para tratar cualidades ordinales.

Actualmente existe una aplicación de libre descarga [3] que permite realizar los cálculos relativos a Delta, pero solo está soportada para el sistema operativo Windows. Afortunadamente, existe una alternativa en forma de aplicación web [4], que es accesible desde cualquier plataforma. Sin embargo, hoy en día la tendencia es publicar código abierto como parte de algún programa ampliamente distribuido y aceptado, a fin de ahorrar tiempo y esfuerzo en el tratamiento de datos. En la actualidad, R es el *software* estadístico de código abierto con mayor aceptación global. Por lo tanto, el **objetivo** de este trabajo es crear un paquete de R con la implementación del cálculo de Delta.

## Índices de acuerdo corregidos por azar

### 2.1 Coeficiente kappa de Cohen

Una de las principales ventajas del coeficiente kappa es la sencillez de sus cálculos. Suponiendo una matriz de clasificación como la mostrada en la [Tabla 1.1](#), el índice kappa consiste en calcular el cociente entre el exceso de acuerdos sobre el azar y el máximo exceso posible sobre el azar. Para ello, previamente se calcula el índice de acuerdo observado ( $I_o$ ), que refleja la proporción de acuerdos observados, y el índice de acuerdo esperado bajo el modelo de independencia ( $I_e$ ), que refleja la proporción de acuerdos esperados por azar:

$$I_o = \frac{\sum_{i=1}^K x_{ii}}{n},$$

$$I_e = \frac{\sum_{i=1}^K r_i c_i}{n^2}.$$

El exceso de acuerdos sobre el azar es  $I_o - I_e$ , mientras que el máximo exceso posible sobre el azar es  $1 - I_e$ . Así, pues, el coeficiente kappa se define como:

$$\hat{\kappa} = \frac{I_o - I_e}{1 - I_e}, \quad (2.1)$$

que expresa la proporción de acuerdo sobre el total de posibles acuerdos no debidos al azar.

El coeficiente  $\hat{\kappa}$  puede tomar valores entre  $-I_e/(1 - I_e)$  y  $+1$ :

- El valor máximo que puede alcanzar  $\hat{\kappa}$  es 1, que ocurre cuando la matriz de clasificación solo tiene valores positivos en la diagonal principal, es decir, todo son acuerdos,  $I_o = 1$ .

- Si la concordancia no es perfecta, sino parcial,  $I_o > I_e$  y el valor  $\hat{\kappa} \in (0, 1)$ .
- Si los acuerdos observados coinciden con los esperados por azar,  $I_o = I_e$ , la concordancia es nula,  $\hat{\kappa} = 0$ .
- Si el acuerdo esperado por azar supera al acuerdo observado,  $I_o < I_e$ , existe discordancia entre los observadores y el valor  $\hat{\kappa} \in (-I_e/(1 - I_e), 0)$ .
- Por último, si no hay valores positivos en la diagonal principal de la matriz de clasificación,  $I_o = 0$ , no hay ningún acuerdo, la discordancia es perfecta, y  $\hat{\kappa} = -I_e/(1 - I_e) < 0$ .

**Acuerdos parciales** En ocasiones resulta de interés estudiar el acuerdo entre los observadores en cada una de las categorías. Sin embargo, kappa es una medida de concordancia global que no permite estudiar acuerdos parciales directamente. Para hacerlo, es preciso colapsar la tabla de clasificación para cada categoría, obteniendo  $K$  tablas  $2 \times 2$  a partir de las cuales se puede analizar el acuerdo parcial en cada categoría.

### Ejemplo 2.1.

Dada la siguiente matriz de clasificaciones:

	A	B	C
A	15	4	3
B	5	21	4
C	0	1	25

podemos obtener tres tablas colapsadas, una por cada categoría (A, B y C):

A		B		C	
A	$\bar{A}$	B	$\bar{B}$	C	$\bar{C}$
A	15	7	B	21	9
$\bar{A}$	5	51	$\bar{B}$	5	43
			$\bar{C}$	7	45

El índice kappa global calculado es  $\hat{\kappa} = 0.671$ , mientras que los acuerdos parciales calculados a partir de las tablas colapsadas son:  $\hat{\kappa}_A = 0.609$ ,  $\hat{\kappa}_B = 0.611$  y  $\hat{\kappa}_C = 0.782$ .

**Kappa ponderado** Cuando existen más de dos categorías de clasificación ( $K > 2$ ) y la cualidad está definida en una escala ordinal, puede suceder que unos desacuerdos sean más importantes que otros. Es este caso, se puede asignar un peso  $w_{ij}$  a cada entrada  $x_{ij}$  de la matriz de clasificación con el objetivo de ponderar el desacuerdo ocurrido [5]. Por ejemplo, supongamos una cualidad ordinal que expresa el estado de gravedad de una enfermedad, cuyas categorías sean “incipiente, leve, avanzada y terminal”. A la hora de clasificar los distintos individuos, no es lo mismo que los dos observadores estén en desacuerdo en un

grado clasificatorio (p. ej., incipiente-leve) que en dos (p. ej., incipiente-avanzada) o tres (incipiente-terminal).

Aunque los pesos pueden ser asignados de forma subjetiva por el investigador, existen dos ponderaciones objetivas muy comunes basadas en las distancias entre las categorías: la lineal y la cuadrática. Se puede construir la matriz de pesos objetiva aplicando la siguiente fórmula:

$$w_{ij} = 1 - \left( \frac{|i - j|}{K - 1} \right)^r,$$

donde  $r$  tomará el valor 1 para aplicar la fórmula lineal, y 2 para aplicar la cuadrática. Si  $r = 0$ , obtendremos la matriz identidad de orden  $K$  y el coeficiente kappa calculado será no ponderado.

De esta forma, se construye una matriz de pesos cuya diagonal principal son unos ( $w_{ii} = 1$ ), que es simétrica ( $w_{ij} = w_{ji}$ ), cuyo desacuerdo máximo tiene un peso igual a 0 ( $w_{1K} = w_{K1} = 0$ ) y el resto de desacuerdos toman un valor entre 0 y 1 ( $0 \leq w_{ij} < 1$ , si  $i \neq j$ ).

Los índices de concordancia observado y esperado se calculan multiplicando el peso  $w_{ij}$  por cada elemento correspondiente de la matriz de frecuencias observadas y esperadas, respectivamente. Es decir,

$$I_o = \frac{\sum_{i=1}^K \sum_{j=1}^K w_{ij} x_{ij}}{n},$$

$$I_e = \frac{\sum_{i=1}^K \sum_{j=1}^K w_{ij} r_i c_j}{n^2}.$$

La expresión del coeficiente kappa ponderado,  $\hat{\kappa}_p$ , es igual que en la [Ecuación \(2.1\)](#), sustituyendo  $I_o$  e  $I_e$  por sus versiones ponderadas.

### Ejemplo 2.2.

Supongamos que dos oncólogos clasifican a 256 pacientes de cáncer en las categorías incipiente, leve, avanzado y terminal. La siguiente tabla muestra los diagnósticos de los dos oncólogos:

	Incipiente	Leve	Avanzado	Terminal
Incipiente	61	18	5	3
Leve	4	43	8	9
Avanzado	8	9	38	8
Terminal	2	5	7	28

Dado que las categorías son cualidades ordinales, se puede aplicar el coeficiente kappa ponderado, bien con la fórmula lineal o cuadrática. Las matrices de pesos calculadas mediante ambas fórmulas se muestran a continuación.

Fórmula lineal					Fórmula cuadrática				
	Incip.	Leve	Avanz.	Term.		Incip.	Leve	Avanz.	Term.
Incip.	1	$\frac{2}{3}$	$\frac{1}{3}$	0	Incip.	1	$\frac{8}{9}$	$\frac{5}{9}$	0
Leve	$\frac{2}{3}$	1	$\frac{2}{3}$	$\frac{1}{3}$	Leve	$\frac{8}{9}$	1	$\frac{8}{9}$	$\frac{5}{9}$
Avanz.	$\frac{1}{3}$	$\frac{2}{3}$	1	$\frac{2}{3}$	Avanz.	$\frac{5}{9}$	$\frac{8}{9}$	1	$\frac{8}{9}$
Term.	0	$\frac{1}{3}$	$\frac{2}{3}$	1	Term.	0	$\frac{5}{9}$	$\frac{8}{9}$	1

Aplicando la fórmula lineal, obtenemos:

$$I_o = 0.84, I_e = 0.597 \text{ y } \hat{\kappa}_p = 0.602.$$

Por otro lado, aplicando la fórmula cuadrática, obtenemos:

$$I_o = 0.91, I_e = 0.737 \text{ y } \hat{\kappa}_p = 0.658.$$

Por último, si no aplicamos ninguna ponderación, obtenemos:

$$I_o = 0.664, I_e = 0.259 \text{ y } \hat{\kappa} = 0.546.$$

**Inferencias sobre kappa** En muchas ocasiones estamos interesados no solo en dar una estimación puntual, sino un intervalo de confianza sobre el parámetro estimado. En este caso, será necesario estimar la varianza de kappa para realizar inferencias. Si  $n > 2 \times K^3$ , entonces  $\hat{\kappa}$  es aproximadamente Normal, de media  $\kappa$  y varianza  $V(\hat{\kappa})$  [6, 7].

$$V(\hat{\kappa}) = \frac{A - B}{n(1 - I_e)^2}, \quad (2.2)$$

donde

$$A = \sum_{i=1}^K \sum_{j=1}^K p_{ij} [w_{ij} - (\bar{w}_{i\cdot} + \bar{w}_{\cdot j}) (1 - \hat{\kappa})]^2,$$

$$B = [\hat{\kappa} - I_e (1 - \hat{\kappa})]^2,$$

siendo  $p_{ij}$  la proporción de objetos clasificados en la categoría  $i$ -ésima por el observador  $\mathbf{R}$  y en la categoría  $j$ -ésima por el observador  $\mathbf{C}$ , es decir,  $p_{ij} = x_{ij}/n$ ; y siendo  $\bar{w}_{i\cdot}$  y  $\bar{w}_{\cdot j}$  las medias ponderadas de los pesos por fila y por columna, respectivamente, definidas como

$$\bar{w}_{i\cdot} = \sum_{j=1}^K w_{ij} p_{\cdot j},$$

y

$$\bar{w}_{\cdot j} = \sum_{i=1}^K w_{ij} p_{i\cdot},$$

siendo

$$p_{i\cdot} = \sum_{j=1}^K p_{ij}$$

y

$$p_{\cdot j} = \sum_{i=1}^K p_{ij}$$

las proporciones marginales de filas y columnas, respectivamente.

Como se ha comentado previamente, el estimador muestral kappa sigue una distribución  $\mathcal{N}(\kappa, V(\kappa))$ , por lo que la [Ecuación \(2.2\)](#) puede utilizarse para construir intervalos de confianza o realizar contrastes de hipótesis sobre  $\kappa$ . El intervalo de confianza de dos colas al error  $\alpha$  es

$$\kappa \in \hat{\kappa} \pm z_{\alpha/2} \times \sqrt{\hat{V}(\hat{\kappa})}.$$

### Ejemplo 2.3.

Para el Ejemplo 2.2, el intervalo al 95% de confianza para  $\kappa$  no ponderado es

$$\hat{\kappa} = 0.546; \quad I_e = 0.259; \quad A = 0.403; \quad B = 0.184.$$

$$\hat{V}(\hat{\kappa}) = \frac{A - B}{n(1 - I_e)^2} = \frac{0.403 - 0.184}{256 \times (1 - 0.259)^2} = 0.00156.$$

$$\begin{aligned} \kappa \in \hat{\kappa} \pm z_{\alpha/2} \times \sqrt{\hat{V}(\hat{\kappa})} &= 0.546 \pm 1.96 \times 0.0395 \implies \\ \kappa &\in [0.469, 0.624] \end{aligned}$$

Al 95% de confianza,  $\kappa$  toma un valor entre 0.469 y 0.624, por lo que podemos decir que existe acuerdo parcial entre los dos observadores, ya que la concordancia nula,  $\kappa = 0$ , no está incluida en el intervalo.

## 2.2 Modelo Delta

El índice de acuerdo nominal Delta entre dos observadores [2, 8, 9] es más laborioso de calcular que  $\hat{\kappa}$ , pero presenta las ventajas de representar mejor la realidad cuando los marginales están muy desequilibrados y analizar más fácilmente los acuerdos parciales. Además, la interpretación de  $\hat{\Delta}$  es más sencilla, ya que expresa el porcentaje de respuestas concordantes no debidas al azar.

La formulación de Delta depende del tipo de muestreo realizado y del papel de los observadores. Al igual que kappa, el índice Delta se calcula a partir de una matriz de

clasificación como la mostrada en la [Tabla 1.1](#). Los datos de esta tabla pueden obtenerse de acuerdo a dos tipos de muestreo:

- Muestreo de *tipo I*. Se eligen  $n$  objetos al azar y se clasifican en  $K$  categorías por  $\mathbf{R}$  y  $\mathbf{C}$ . Como consecuencia, ningún marginal está fijado de antemano.
- Muestreo de *tipo II*. Se eligen  $r_i$  individuos al azar de entre los que son de tipo  $i$  según  $\mathbf{R}$  y se les clasifica en  $K$  categorías por  $\mathbf{C}$ . Como consecuencia, los marginales por filas están fijados de antemano.

Por otro lado, según el papel que juegan los observadores, se puede diferenciar entre:

- Uno de los observadores es un *estándar*, es decir, un observador proporciona las respuestas correctas. Por lo general, se supondrá que, de existir un estándar, este será el observador por filas ( $\mathbf{R}$ ). En este caso, el acuerdo de ambos será una medida de *conformidad*, que caracteriza la validez de  $\mathbf{C}$  frente al estándar  $\mathbf{R}$ .
- Ningún observador es un estándar. En este caso, el acuerdo entre los observadores es una medida de *consistencia*, que caracteriza la fiabilidad entre ambos.

En ambos tipos de muestreo,  $\mathbf{R}$  puede ser un estándar o no, aunque cuando el muestreo es de *tipo II*, lo habitual es que lo sea.

### 2.2.1 Formulación del modelo

Como se ha dicho anteriormente, la formulación de Delta depende del tipo de muestreo y del papel de los observadores.

**Muestreo tipo II** Comencemos por el caso en que el muestreo es de *tipo II* y el observador  $\mathbf{R}$  es un estándar que proporciona las clasificaciones correctas.

El modelo Delta asume que el observador  $\mathbf{C}$  reconoce un objeto de la categoría  $i$  (con  $i = 1, \dots, K$ ) con una intensidad  $\Delta_i$ . Si  $\mathbf{C}$  se enfrenta a un objeto de la categoría  $i$  y lo reconoce, lo clasifica correctamente, y si no (probabilidad  $1 - \Delta_i$ ), lo clasifica al azar. La probabilidad de clasificación al azar no es necesariamente uniforme, sino que  $\mathbf{C}$  puede elegir la categoría  $j$  con una probabilidad  $\pi_j$ . Por lo tanto, el modelo propuesto es

$$\begin{aligned} P(C_i|R_i) &= p_{ii} = \Delta_i + (1 - \Delta_i)\pi_i \\ P(C_j|R_i) &= p_{ij} = (1 - \Delta_i)\pi_j \quad (\text{para } i \neq j), \end{aligned} \tag{2.3}$$

donde  $0 \leq \pi_j \leq 1$  y  $\sum \pi_j = 1$ , por ser una probabilidad. Por otro lado,  $\Delta_i \leq 1$  ( $\forall i$ ), que puede llegar a ser negativo si una categoría se confunde con otra. En este modelo se

asume que cada fila de la matriz de clasificación procede de una distribución multinomial de tamaño  $r_i$  con parámetros  $(p_{i1}, p_{i2}, \dots, p_{iK})$ , con  $\sum_j p_{ij} = 1$ .

De este modelo, llamado *modelo II* se obtienen dos medidas de acuerdo: una parcial y otra global. Dado que  $\Delta_i$  refleja la intensidad con que  $\mathbf{C}$  reconoce la categoría  $i$ , una medida de conformidad global de  $\mathbf{C}$  respecto a  $\mathbf{R}$  vendrá dada por:

$$\Delta_{II} = \frac{\sum_{i=1}^K r_i \Delta_i}{n},$$

es decir, la conformidad global es la media ponderada de las intensidades  $\Delta_i$ , donde la ponderación viene dada por el total de objetos de la clase  $i$ . Por lo tanto, las conformidades parciales en cada clase son

$$\mathcal{A}_i = \frac{r_i \Delta_i}{n}.$$

**Muestreo tipo I** Suponiendo que no hay ningún marginal fijado de antemano, es decir, hay una sola muestra de tamaño  $n$ , el modelo de la [Ecuación \(2.3\)](#) se debe completar con:

$$P(R_i \cap C_j) = P(R_i) \times P(C_j | R_i) = q_i \times p_{ij} = q_{ij}, \quad (2.4)$$

donde  $q_i$  es la distribución marginal de filas, estimada como  $\hat{q}_i = r_i/n$ . Bajo este modelo, llamado *modelo I*, las  $x_{ij}$  de la tabla de clasificación son observaciones que proceden de una única distribución multinomial de parámetros  $q_{ij}$ . Las medidas de acuerdo global y parcial correspondientes a este modelo son, respectivamente:

$$\Delta_I = \sum_{i=1}^K q_i \Delta_i$$

y

$$\mathcal{A}_i = q_i \Delta_i.$$

Por lo general, se usará la notación simplificada  $\Delta$ , siempre que el tipo de muestreo esté suficientemente claro. El valor  $\Delta$  es la proporción de respuestas concordantes no atribuibles al azar. Estas concordancias se refieren a *conformidad* del observador  $\mathbf{C}$  respecto a  $\mathbf{R}$  si este último es un estándar, o a *consistencia* entre las respuestas de los dos observadores si ninguno es un estándar. De esta manera,  $\Delta_I = 0.70$  indica que  $\mathbf{C}$  y  $\mathbf{R}$  concuerdan (no debido al azar) en el 70% de los objetos que clasifican; mientras que  $\Delta_{II} = 0.70$  indica que  $\mathbf{C}$  reconoce (no debido al azar) el 70% de las clasificaciones realizadas por  $\mathbf{R}$ .

En ambos tipos de muestreo, la estimación del acuerdo global y acuerdos parciales es

$$\hat{\Delta} = \frac{\sum r_i \hat{\Delta}_i}{n}, \quad \hat{\mathcal{A}}_i = \frac{r_i \hat{\Delta}_i}{n},$$

es decir, no dependen del tipo de muestreo, y tampoco varían al intercambiar filas por columnas, por lo que son medidas válidas tanto de consistencia como de conformidad.

### 2.2.2 Estimación de parámetros

La estimación de  $\Delta$ , independientemente del tipo de muestreo, involucra la resolución de una ecuación no lineal que (en general) no tiene solución explícita. Es por ello que normalmente se usa un procedimiento numérico para su resolución.

La formulación dada en [2] implica determinar una constante desconocida  $B$ , que tiene que cumplir la restricción

$$B = n(1 - \Delta) \geq B_0 = \max_i \{ \sqrt{c_i - x_{ii}} + \sqrt{r_i - x_{ii}} \}^2.$$

Llamamos  $h$  a la categoría  $i$  que proporciona el valor  $B_0$ . La constante  $B$  es solución de la ecuación

$$y(B) = (K - 2)B + \sum s(i) \sqrt{\{B + (c_i - r_i)\}^2 - 4B(c_i - x_{ii})} = 0, \quad (2.5)$$

donde  $s(i)$  puede tomar los valores  $-1$  y  $+1$ .

La Ecuación (2.5) será aludida como  $y^-(B)$  si  $s(h) = -1$ , y como  $y^+(B)$  si  $s(h) = +1$ . El signo de las categorías distintas de  $h$  es siempre negativo, es decir,  $s(i) = -1, \forall i \neq h$ . En cambio, para la categoría  $h$ , el signo de  $s(h)$  dependerá de  $y^-(B_0)$ , de la siguiente manera

$$s(h) = \begin{cases} -1 & \text{si } y^-(B_0) \geq 0, \\ +1 & \text{si } y^-(B_0) < 0. \end{cases}$$

En el paquete presentado en el Capítulo 3 se utiliza el método de Newton-Raphson para resolver la Ecuación (2.5). Una vez obtenido el valor de  $B$ , se pueden calcular los parámetros del modelo de forma inmediata mediante las siguientes expresiones [2]:

- Probabilidad con la que se clasifica un objeto no reconocido en la categoría  $i$ :

$$\hat{\pi}_i = \frac{B + (c_i - r_i) + s(i) \sqrt{[B + (c_i - r_i)]^2 - 4B(c_i - x_{ii})}}{2B}. \quad (2.6)$$

- Intensidad con la que se reconoce un objeto de la categoría  $i$ :

$$\hat{\Delta}_i = \frac{x_{ii} - r_i \hat{\pi}_i}{r_i (1 - \hat{\pi}_i)}. \quad (2.7)$$

- Acuerdos parciales para la categoría  $i$  y acuerdo global:

$$\hat{A}_i = \frac{r_i \hat{\Delta}_i}{n}, \quad \hat{\Delta} = \frac{\sum r_i \hat{\Delta}_i}{n} = \sum \hat{A}_i. \quad (2.8)$$

### 2.2.2.1 Existencia y unicidad de solución

Puede ocurrir que el problema no tenga solución o que tenga infinitas soluciones. Ambos casos ocurren cuando existe una categoría  $h$  tal que  $n - \sum x_{ii} = c_h + r_h - 2x_{hh}$ , es decir, cuando todos los desacuerdos, si existen, están concentrados en la misma categoría o cuando no hay desacuerdos. El Ejemplo 2.4 muestra distintas situaciones para las que la Ecuación (2.5) tiene infinitas soluciones.

#### Ejemplo 2.4.

Matrices para las que la Ecuación (2.5) tiene infinitas soluciones.

	A			B			C		
	$c_1$	$c_2$	$c_3$	$c_1$	$c_2$	$c_3$	$c_1$	$c_2$	$c_3$
$r_1$	10	0	0	$r_1$	10	0	$r_1$	10	0
$r_2$	0	11	0	$r_2$	0	9	$r_2$	0	9
$r_3$	0	0	9	$r_3$	0	2	$r_3$	2	4

- Matriz A. No hay desacuerdos, es decir,  $x_{ii} = c_i = r_i \forall i$ . En este caso,  $\hat{\Delta}_i = +1 \forall i$  y los valores  $\hat{\pi}_i$  quedan indeterminados.
- Matriz B.  $x_{ii} = c_i = r_i$  en todas las categorías salvo dos.
- Matriz C. Todos los desacuerdos ocurren en una misma fila, es decir,  $x_{ii} = r_i$  en todas las categorías salvo en una.

En todos los casos, el problema se puede resolver incrementando todos los datos  $x_{ij}$  de la Tabla 1.1 en +0.5. A esta nueva tabla se le llamará  $M3$ . Este recurso, muy utilizado en estadística, garantiza que la Ecuación (2.5) tiene solución única siempre [2].

**Caso particular:  $K = 2$**  En el caso particular de matrices de orden 2, el modelo tiene demasiados parámetros a estimar y, como consecuencia, el problema tiene infinitas soluciones. Una manera de solucionar este problema es añadir una categoría ficticia con  $x_{33} = c_3 = r_3 = 1$ , es decir, garantizando que la categoría ficticia no se confunda con otra, e incrementar todos los datos en +0.5. Esta nueva tabla, a la que se llamará  $M2$ , se utiliza para estimar los parámetros de la manera habitual para tablas con  $K > 2$ , con la excepción de que los parámetros  $\Delta_i$  se estimarán solo para las categorías reales ( $i = 1, 2$ ). Por lo tanto, todas las medidas que se calculen a partir de  $\hat{\Delta}_i$  estarán referidas solo a las categorías originales, entendiendo además que el tamaño muestral es el original y no el incrementado, es decir,  $n = r_1 + r_2$ . Por consiguiente, la medida de acuerdo global se calculará como  $\hat{\Delta} = (r_1 \hat{\Delta}_1 + r_2 \hat{\Delta}_2) / (r_1 + r_2)$ . El Ejemplo 2.5 muestra una tabla  $2 \times 2$  y su correspondiente tabla ( $M2$ ) aumentada.

**Ejemplo 2.5. Tabla M2**

La Tabla (a) muestra la clasificación de 100 objetos en 2 categorías, según dos observadores independientes. La Tabla (b) muestra la Tabla (a) aumentada en la categoría ficticia 3.

(a) Tabla original	(b) Tabla M2																									
<table style="margin: auto; border-collapse: collapse;"> <thead> <tr> <th style="border: none;"></th> <th style="border: none; text-align: center;">c<sub>1</sub></th> <th style="border: none; text-align: center;">c<sub>2</sub></th> </tr> </thead> <tbody> <tr> <td style="border: none; text-align: center;">r<sub>1</sub></td> <td style="border: none; text-align: center;">50</td> <td style="border: none; text-align: center;">16</td> </tr> <tr> <td style="border: none; text-align: center;">r<sub>2</sub></td> <td style="border: none; text-align: center;">12</td> <td style="border: none; text-align: center;">31</td> </tr> </tbody> </table>		c <sub>1</sub>	c <sub>2</sub>	r <sub>1</sub>	50	16	r <sub>2</sub>	12	31	<table style="margin: auto; border-collapse: collapse;"> <thead> <tr> <th style="border: none;"></th> <th style="border: none; text-align: center;">c<sub>1</sub></th> <th style="border: none; text-align: center;">c<sub>2</sub></th> <th style="border: none; text-align: center;">c<sub>3</sub></th> </tr> </thead> <tbody> <tr> <td style="border: none; text-align: center;">r<sub>1</sub></td> <td style="border: none; text-align: center;">50.5</td> <td style="border: none; text-align: center;">16.5</td> <td style="border: none; text-align: center;">0.5</td> </tr> <tr> <td style="border: none; text-align: center;">r<sub>2</sub></td> <td style="border: none; text-align: center;">12.5</td> <td style="border: none; text-align: center;">31.5</td> <td style="border: none; text-align: center;">0.5</td> </tr> <tr> <td style="border: none; text-align: center;">r<sub>3</sub></td> <td style="border: none; text-align: center;">0.5</td> <td style="border: none; text-align: center;">0.5</td> <td style="border: none; text-align: center;">1.5</td> </tr> </tbody> </table>		c <sub>1</sub>	c <sub>2</sub>	c <sub>3</sub>	r <sub>1</sub>	50.5	16.5	0.5	r <sub>2</sub>	12.5	31.5	0.5	r <sub>3</sub>	0.5	0.5	1.5
	c <sub>1</sub>	c <sub>2</sub>																								
r <sub>1</sub>	50	16																								
r <sub>2</sub>	12	31																								
	c <sub>1</sub>	c <sub>2</sub>	c <sub>3</sub>																							
r <sub>1</sub>	50.5	16.5	0.5																							
r <sub>2</sub>	12.5	31.5	0.5																							
r <sub>3</sub>	0.5	0.5	1.5																							

**2.2.2.2 Bondad del ajuste**

La validez de  $\hat{\Delta}$  está sujeta a la adecuación del modelo subyacente para analizar un problema dado. El test apropiado para verificar la validez del modelo es el clásico test  $\chi^2$  de bondad de ajuste. Las frecuencias esperadas por el modelo Delta son

$$E_{ii} = x_{ii}; \text{ y } E_{ij} = r_i (1 - \hat{\Delta}_i) \hat{\pi}_j = \frac{(r_i - x_{ii})\hat{\pi}_j}{1 - \hat{\pi}_i}, \text{ para } i \neq j. \quad (2.9)$$

Las frecuencias esperadas y observadas se utilizan para determinar el estadístico  $\chi^2$ , mediante la expresión

$$\chi_{\text{exp}}^2 = \sum_{i \neq j} \frac{(x_{ij} - E_{ij})^2}{E_{ij}},$$

que, bajo la hipótesis nula de que el modelo es adecuado para analizar el problema, sigue una distribución  $\chi^2$  con  $(K - 1)(K - 2) - 1$  grados de libertad. Por lo tanto, un resultado significativo indica que el modelo Delta no es adecuado para analizar el problema. Este test será inválido si más del 20% de las frecuencias esperadas por el modelo son menores que 5 o alguna de ellas es menor que 1.

**2.2.3 Medidas de acuerdo corregidas por azar**

Además del nivel de acuerdo global ( $\hat{\Delta}$ ) y del nivel de acuerdo en cada categoría sobre el total de objetos clasificados ( $\hat{\mathcal{A}}_i$ ), se pueden obtener otros índices, también corregidos por azar, que informan de la concordancia entre los observadores.

- **Conformidad.** Expresa la capacidad del observador **C** de reconocer un objeto clasificado por el observador **R** como de categoría  $i$ . En otras palabras, de los objetos clasificados por **R** como  $i$ , porcentaje que **C** identifica como  $i$  (no debido al azar). Esta medida es útil cuando **R** es un estándar.

$$\hat{\mathcal{F}}_i = \hat{\Delta}_i.$$

- **Predictividad.** Expresa la credibilidad de que un objeto clasificado por **C** como de categoría  $i$  sea realmente de ese tipo según el observador **R**. En otras palabras, de los objetos clasificados por **C** como  $i$ , porcentaje que realmente son  $i$  (no debido al azar) según **R**. Esta medida es útil cuando **R** es un estándar y el muestreo es de tipo I.

$$\hat{\mathcal{P}}_i = \frac{r_i \hat{\Delta}_i}{c_i}.$$

- **Consistencia.** Expresa la fiabilidad del acuerdo en objetos de la categoría  $i$ . En otras palabras, de los objetos clasificados por **C** y/o **R** como  $i$ , porcentaje de acuerdos en esa categoría (no debido al azar). Es una medida útil cuando no hay un estándar y el muestreo es de tipo I.

$$\hat{\mathcal{S}}_i = \frac{2r_i \hat{\Delta}_i}{r_i + c_i}.$$

No todos los índices tienen sentido siempre, sino que depende del tipo de muestreo y de si hay o no un observador estándar. La [Tabla 2.1](#) recoge las medidas válidas en cada caso. El Ejemplo 2.6 muestra las medidas de concordancia calculadas para una tabla de orden 3.

**Tabla 2.1** Validez de las medidas de acuerdo en función del tipo de muestreo y la existencia o no de un observador estándar.

	No hay estándar	<b>R</b> es un estándar
Muestreo I	$\hat{\mathcal{S}}_i, \hat{\mathcal{A}}_i$ y $\hat{\Delta}$	$\hat{\mathcal{F}}_i, \hat{\mathcal{P}}_i, \hat{\mathcal{A}}_i$ y $\hat{\Delta}$
Muestreo II	$\hat{\mathcal{A}}_i$ y $\hat{\Delta}$	$\hat{\mathcal{F}}_i, \hat{\mathcal{A}}_i$ y $\hat{\Delta}$

### Ejemplo 2.6. Medidas de acuerdo

La Tabla (a) muestra la clasificación de 100 objetos en 3 categorías, según dos observadores independientes. La Tabla (b) muestra las medidas de acuerdo corregidas por azar para cada categoría de la Tabla (a), así como el índice Delta de acuerdo global.

(a)				(b)					
	A	B	C	Categoría	$\hat{\Delta}_i$	$\hat{\mathcal{A}}_i$	$\hat{\mathcal{F}}_i$	$\hat{\mathcal{P}}_i$	$\hat{\mathcal{S}}_i$
A	14	3	2	A	0.611	0.116	0.611	0.527	0.566
B	3	20	2	B	0.650	0.163	0.650	0.542	0.591
C	5	7	44	C	0.715	0.400	0.715	0.834	0.770
					$\hat{\Delta} = 0.679$				

Suponiendo un muestreo de tipo I y ningún estándar, la medida de acuerdo global,  $\hat{\Delta} = 0.679$ , indica que los observadores están de acuerdo en, aproximadamente, el 68% de sus respuestas, donde la mayoría de los acuerdos provienen del acuerdo parcial en la clase  $C$  ( $\hat{\mathcal{A}}_C = 40\%$ ). Por otro lado, ambos observadores están de acuerdo en, aproximadamente, el 57% de los objetos clasificados como  $A$  ( $\hat{\mathcal{S}}_A = 0.566$ ), el 59% de los clasificados como

$B$  ( $\hat{\mathcal{S}}_B = 0.591$ ) y el 77% de los clasificados como  $C$  ( $\hat{\mathcal{S}}_i = 0.770$ ).

Suponiendo un muestreo de tipo I y que el observador por filas es un estándar,  $\mathbf{C}$  reconoce y clasifica correctamente (no por azar) el 61% de los objetos de tipo  $A$  ( $\hat{\mathcal{F}}_A = 0.611$ ), el 65% de los objetos de tipo  $B$  ( $\hat{\mathcal{F}}_B = 0.650$ ) y el 72% de los objetos de tipo  $C$  ( $\hat{\mathcal{F}}_C = 0.715$ ). Por otro lado, de los objetos clasificados por  $\mathbf{C}$  como de clase  $A$ , el 53% son de ese tipo según el estándar  $\mathbf{R}$  ( $\hat{\mathcal{P}}_A = 0.527$ ), de los clasificados como  $B$ , el 54% lo son ( $\hat{\mathcal{P}}_B = 0.542$ ) y de los clasificados como  $C$ , el 83% lo son ( $\hat{\mathcal{P}}_C = 0.834$ ).

**Varianza estimada de las medidas de acuerdo** Las expresiones para estimar la varianza de estos índices dependen del tipo de muestreo asumido por el modelo. La [Tabla 2.2](#) muestra la estimación y varianza estimada de las medidas de concordancia corregidas por azar, que han sido obtenidas de [\[2, 8\]](#) e implementadas en el paquete DeltaMAN, descrito en el [Capítulo 3](#).

Para calcular la varianza de estos índices, previamente, es necesario obtener la matriz de covarianzas de los parámetros  $\Delta_i$ , cuya expresión viene dada por:

$$\widehat{Cov}(\hat{\Delta}_i, \hat{\Delta}_j) = V_{ij} = -\frac{(v_i E_i)(v_j E_j)}{E} + \delta_{ij} v_i \left\{ \frac{x_{ii}}{r_i^2} + v_i E_i \right\}, \quad (2.10)$$

donde

$$\delta_{ij} = \begin{cases} 0 & \text{si } i \neq j \\ 1 & \text{si } i = j \end{cases}; \quad v_i = \frac{1 - \hat{\Delta}_i}{1 - \hat{\pi}_i}; \quad E_i = \frac{\hat{\pi}_i}{B - r_i v_i}; \quad E = \sum E_i.$$

**Tabla 2.2** Estimación y varianza estimada de las medidas de concordancia corregidas por azar para tablas de orden  $K > 2$ .  $\hat{V}_{\mathbf{I}}$  y  $\hat{V}_{\mathbf{II}}$ : varianza bajo el muestreo de tipo I y de tipo II, respectivamente.

Validez	Medida	Estimación	$\hat{V}_{\mathbf{I}}$	$\hat{V}_{\mathbf{II}}$
Si $\mathbf{R}$ es un estándar	Conformidad	$\hat{\mathcal{F}}_i = \hat{\Delta}_i$	$V_{ii}$	$V_{ii}$
	Predictividad	$\hat{\mathcal{P}}_i = \frac{r_i \hat{\Delta}_i}{c_i}$	$\left(\frac{r_i}{c_i}\right)^2 \left\{ V_{ii} + \frac{c_i - r_i}{c_i r_i} \hat{\Delta}_i^2 \right\}$	—
Si no hay estándar	Consistencia	$\hat{\mathcal{S}}_i = \frac{2r_i \hat{\Delta}_i}{r_i + c_i}$	$\left(\frac{2r_i}{r_i + c_i}\right)^2 \left[ V_{ii} + \frac{\hat{\Delta}_i^2}{r_i + c_i} \left\{ \frac{c_i}{r_i} - 2 + \frac{2x_{ii}}{r_i + c_i} \right\} \right]$	—
Siempre	Acuerdo	$\hat{\mathcal{A}}_i = \frac{r_i \hat{\Delta}_i}{n}$	$\left(\frac{r_i}{n}\right)^2 \left\{ V_{ii} + \frac{n - r_i}{n r_i} \hat{\Delta}_i^2 \right\}$	$\left(\frac{r_i}{n}\right)^2 V_{ii}$
	Acuerdo global	$\hat{\Delta} = \sum \hat{\mathcal{A}}_i$	$\frac{n - E^{-1} - n \Delta^2}{n^2}$	$\frac{n - E^{-1} - \sum r_i \hat{\Delta}_i^2}{n^2}$

Las varianzas mostradas en la [Tabla 2.2](#) son válidas para tablas de orden mayor que 2. En el caso de  $K = 2$ , se puede estimar la varianza de las distintas medidas mediante estas expresiones, utilizando la tabla ampliada  $M2$ . Como salvedad, la varianza estimada del acuerdo global viene dada por:

$$\hat{V}_I(\hat{\Delta}) = \frac{\sum_i \sum_j r_i r_j V_{ij} + \sum_i r_i \hat{\Delta}_i^2 - n \hat{\Delta}^2}{n^2}; \quad \hat{V}_{II}(\hat{\Delta}) = \frac{\sum_i \sum_j r_i r_j V_{ij}}{n^2},$$

siendo  $\hat{V}_I(\hat{\Delta})$  la varianza de  $\hat{\Delta}$  suponiendo el muestreo de tipo I, y  $\hat{V}_{II}(\hat{\Delta})$  suponiendo el muestreo de tipo II, y extendiendo las sumas solo a las categorías originales ( $i = 1, 2$ ).

#### 2.2.4 Estimación asintótica para el caso $K = 2$

Como se ha comentado anteriormente, la [Ecuación \(2.5\)](#) tiene infinitas soluciones cuando  $K = 2$ . Para aplicar el modelo Delta a tablas de este tipo, es necesario aumentarla creando una categoría ficticia e incrementar todos los datos en  $c = 0.5$  (tabla  $M2$ ), a fin de obtener las medidas de concordancia en base a las categorías originales ( $i = 1, 2$ ).

Alternativamente, si en lugar de aumentar los datos en  $c = 0.5$ ,  $c \rightarrow 0$ , los estimadores de los parámetros convergen a [\[2\]](#)

$$\begin{aligned} \hat{\pi}_{1a} &= \frac{\sqrt{x_{21}}}{\sqrt{x_{12} + \sqrt{x_{21}}}}, & \hat{\pi}_{2a} &= \frac{\sqrt{x_{12}}}{\sqrt{x_{12} + \sqrt{x_{21}}}}, & \hat{\Delta}_a &= \frac{x_{11} + x_{22} - 2\sqrt{x_{12}x_{21}}}{n}, \\ \hat{\Delta}_{ia} &= \frac{x_{ii} - \sqrt{x_{12}x_{21}}}{r_i}. \end{aligned} \quad (2.11)$$

La solución obtenida es una aproximación de la primera, pero con la ventaja de que las expresiones para calcular los parámetros del modelo son explícitas, es decir, no es necesario resolver la [Ecuación \(2.5\)](#). Martín y Femia [\[10\]](#) demostraron que en general  $\hat{\Delta}_a$  es mayor que  $\hat{\Delta}$ . Estos autores redefinieron el estimador asintótico de Delta aumentando los datos en  $c = 1$ , con lo que el estimador para  $\Delta_i$  es

$$\hat{\Delta}_{ia(+1)} = \frac{x_{ii} + 1 - \sqrt{(x_{12} + 1)(x_{21} + 1)}}{r_i + 2},$$

y el estimador del acuerdo global es

$$\hat{\Delta}_{a(+1)} = \frac{x_{11} + x_{22} + 2 - 2\sqrt{(x_{12} + 1)(x_{21} + 1)}}{n + 4}.$$

Se puede demostrar que, con muestras grandes y grado de acuerdo no bajo,  $\hat{\Delta}_a > \hat{\Delta}_{a(+1)} > \hat{\Delta}$ , por lo que  $\hat{\Delta}_{a(+1)}$  es mejor aproximación de  $\hat{\Delta}$  que  $\hat{\Delta}_a$  [\[10\]](#).

A partir de la [Ecuación \(2.11\)](#), se pueden calcular las medidas de concordancia mediante las expresiones empleadas para  $K > 2$ , es decir, como se indica en la [Tabla 2.2](#). Las varianzas de estas medidas se obtienen mediante las expresiones indicadas en la [Tabla 2.3](#) [10], que se podrán calcular siempre y cuando no ocurra que  $x_{12} = x_{21} = 0$  o  $\exists r_i = 0$  o  $\exists c_i = 0$ .

**Tabla 2.3** Solución asintótica para la estimación de la varianza de las medidas de concordancia en tablas de orden  $K = 2$ .  $\hat{V}_{\text{I}}$  y  $\hat{V}_{\text{II}}$ : varianza bajo el muestreo de tipo I y de tipo II, respectivamente.

Validez	Medida	$\hat{V}_{\text{I}}$	$\hat{V}_{\text{II}}$
Si $\mathbf{R}$ es un estándar	$\hat{\mathcal{F}}_i$	$\frac{x_{ii}(1-\hat{\Delta}_i)+\frac{1}{4}(x_{12}+x_{21})}{r_i^2}$	$\frac{x_{ii}(1-\hat{\Delta}_i)+\frac{1}{4}\left(x_{12}+x_{21}-\frac{nx_{12}x_{21}}{r_1r_2}\right)}{r_i^2}$
	$\hat{\mathcal{P}}_i$	$\frac{x_{ii}(1-\hat{\mathcal{P}}_i)+\frac{1}{4}(x_{12}+x_{21})}{c_i^2}$	—
Si no hay estándar	$\hat{\mathcal{S}}_i$	$\frac{n(1-\hat{\Delta})}{(r_i+c_i)^2} \left\{ 2 - \frac{n(1-\hat{\Delta})(x_{12}+x_{21})}{(r_i+c_i)^2} \right\}$	—
Siempre	$\hat{\mathcal{A}}_i$	$\frac{x_{ii}+\frac{1}{4}(x_{12}+x_{21})-n\hat{\mathcal{A}}_i^2}{n^2}$	$\frac{x_{ii}(1-\hat{\Delta}_i)+\frac{1}{4}\left(x_{12}+x_{21}-\frac{nx_{12}x_{21}}{r_1r_2}\right)}{n_i^2}$
	$\hat{\Delta}$	$\frac{(1-\hat{\Delta})(1+\hat{\Delta})}{n}$	$\frac{1-\hat{\Delta}}{n} \left\{ \frac{x_{11}}{r_1} + \frac{x_{22}}{r_2} \right\}$

### Ejemplo 2.7. Comparación de estimadores asintóticos y no asintótico

Continuando con la tabla del [Ejemplo 2.5 \(a\)](#), las tres estimaciones del acuerdo global,  $\Delta$ , son:

$$\begin{array}{ccc}
 \text{(a)} & & \\
 \hline
 & c_1 & c_2 \\
 r_1 & 50 & 16 \\
 r_2 & 12 & 31 \\
 \hline
 \hat{\Delta} & = & 0.476 \\
 \hat{\Delta}_{a(+1)} & = & 0.471 \\
 \hat{\Delta}_a & = & 0.489
 \end{array}$$

Se observa que  $\hat{\Delta}_{a(+1)}$  se aproxima más a  $\hat{\Delta}$  que  $\hat{\Delta}_a$ .

## 2.3 Comparación entre $\kappa$ y $\Delta$

Como ya apuntaron Martín y Femia [2, 8], el coeficiente  $\kappa$  es muy sensible a las distribuciones marginales, no midiendo bien el acuerdo existente cuando los dos marginales están desequilibrados en igual sentido. Este problema se ilustra en el [Ejemplo 2.8](#).

**Ejemplo 2.8. Efecto de las distribuciones marginales**

(a)					(b)						
	$c_1$	$c_2$	$c_3$	Total		$c_1$	$c_2$	$c_3$	Total		
$r_1$	75	10	2	87	$\hat{\Delta} = 0.559$ $\hat{\kappa} = 0.024$	$r_1$	55	10	2	67	$\hat{\Delta} = 0.559$ $\hat{\kappa} = 0.522$
$r_2$	10	1	1	12		$r_2$	10	11	1	22	
$r_3$	0	1	0	1		$r_3$	0	1	10	11	
Total	85	12	3	100		Total	65	22	13	100	

En las Tablas (a) y (b) el valor de  $\hat{\kappa}$  es muy distinto a pesar de que en ambos casos hay el mismo número de acuerdos observados (76). El problema es que la Tabla (a) tiene las distribuciones marginales muy desequilibradas y en el mismo sentido, es decir, los totales de fila y de columna son  $> 80$ ,  $12$  y  $< 5$  en ambos casos. En la Tabla (b) los marginales están más equilibrados y los dos índices de concordancia arrojan resultados similares entre sí. Se puede observar que el valor  $\hat{\Delta}$  es idéntico para ambas tablas, ya que este índice no depende de las distribuciones marginales.

El Ejemplo 2.8 pone de manifiesto que el desequilibrio en las distribuciones marginales provoca que  $\hat{\kappa}$  proporcione una estimación muy deficiente. El Ejemplo 2.9 muestra el efecto de las distribuciones marginales sobre los acuerdos parciales.

**Ejemplo 2.9. Efecto de las distribuciones marginales sobre los acuerdos parciales**

Siguiendo con las Tablas del Ejemplo 2.8 y suponiendo un muestreo de tipo I sin observador estándar, los acuerdos parciales en cada categoría de la Tabla (a) son:

Clase	$\hat{\Delta}_i$	$\hat{\pi}_i$	$\hat{\mathcal{A}}_i$ (SE)	$\hat{\mathcal{S}}_i$ (SE)	$\hat{\kappa}_i$ (SE)
A	0.750	0.448	0.652 (0.1156)	0.759 (0.1293)	0.087 (0.1146)
B	-0.769	0.482	-0.092 (0.1054)	-0.769 (0.8402)	-0.042 (0.0869)
C	-0.075	0.070	-0.001 (9e-04)	-0.038 (0.0299)	-0.015 (0.0117)
$\hat{\Delta}$ (SE) = 0.559 (0.0805)					$\hat{\kappa}$ (SE) = 0.024 (0.0830)

La tabla muestra que la consistencia entre ambos observadores es importante en la clase A ( $\hat{\mathcal{S}}_A = 0.759$ ), mientras que la medida de acuerdo proporcionada por kappa para la clase A es muy baja ( $\hat{\kappa}_A = 0.087$ ) para la gran cantidad de acuerdos observados en esa clase.

Para la Tabla (b), los acuerdos parciales son:

Clase	$\hat{\Delta}_i$	$\hat{\pi}_i$	$\hat{\mathcal{A}}_i$ (SE)	$\hat{\mathcal{S}}_i$ (SE)	$\hat{\kappa}_i$ (SE)
A	0.675	0.448	0.452 (0.1165)	0.686 (0.1678)	0.510 (0.0907)
B	0.035	0.482	0.008 (0.1105)	0.035 (0.5020)	0.359 (0.1103)
C	0.902	0.070	0.099 (0.0300)	0.827 (0.0880)	0.811 (0.0915)
$\hat{\Delta}$ (SE) = 0.559 (0.0805)					$\hat{\kappa}$ (SE) = 0.522 (0.0845)

En este caso, las medidas de consistencia ( $\hat{\mathcal{S}}_i$ ) y de kappa parcial ( $\hat{\kappa}_i$ ) son más parecidas entre sí.

Todos los resultados mostrados se han obtenido mediante la implementación del modelo Delta en R, que se explica en el siguiente capítulo.

## El paquete DeltaMAN

El paquete DeltaMAN (por sus siglas en inglés, *Measurement of Agreement for Nominal data*), ha sido diseñado usando objetos de clase S3. Este paquete proporciona las funciones necesarias para calcular el índice de acuerdo nominal Delta, así como el resto de medidas de concordancia para cada categoría de la variable nominal. El paquete contiene una función principal, `Delta()` (o `delta()`), para calcular las medidas de acuerdo entre dos observadores. Además, se ha implementado una función (`multiDelta()`) para realizar análisis masivos de múltiples observadores contra un estándar (es decir, el observador que proporciona las clasificaciones correctas). El coeficiente kappa de Cohen, `Kappa()`, también ha sido implementado con el fin de comparar sus resultados con los de Delta. En el caso de kappa, la función está preparada para tratar datos nominales y ordinales, así como devolver acuerdos parciales. Finalmente, el paquete contiene una función preparada para ejecutar la aplicación de shiny asociada al paquete, llamada `runDeltaShinyApp()`. El paquete se encuentra actualmente en el repositorio CRAN y puede ser instalado y cargado de la forma habitual:

```
install.packages("DeltaMAN")
library(DeltaMAN)
```

### 3.1 La función Kappa()

Esta función calcula el coeficiente kappa de Cohen para datos nominales u ordinales. Los argumentos de entrada de la función son:

- `m` una matriz cuadrada de frecuencias absolutas, resultado de las clasificaciones realizadas por dos observadores.

- `r` un número entero (0, 1 o 2) utilizado para crear la matriz de pesos,

$$w_{ij} = 1 - \left( \frac{|i - j|}{K - 1} \right)^r$$

- Si `r = 0`, se calcula el coeficiente no ponderado (para variables nominales).
- Si `r = 1`, se calcula el coeficiente ponderado mediante la fórmula lineal.
- Si `r = 2`, se calcula el coeficiente ponderado mediante la fórmula cuadrática.
- `alternative` una cadena de texto especificando la hipótesis alternativa para construir el intervalo de confianza. Las opciones disponibles son las habituales: `"two.sided"`, `"greater"` y `"less"`.
- `conf.level` el nivel de confianza con el que construir el intervalo. Por defecto es 0.95.
- `partial` un valor lógico indicando si se desean obtener las medidas de acuerdo parciales. Estas medidas se calculan usando la fórmula no ponderada.

El resultado de esta función es una lista de tres elementos, o cuatro (si `partial = TRUE`):

- `k` el coeficiente kappa global calculado.
- `se` el error estándar del valor estimado de kappa.
- `conf.int` el intervalo de confianza construido para el parámetro kappa.
- `partial.kappa` un objeto de clase `data.frame` con tres columnas:
  - la categoría de la variable;
  - el acuerdo en cada categoría obtenido al “colapsar” la tabla;
  - el error estándar de la estimación para cada categoría.

Veamos un ejemplo con la siguiente matriz  $3 \times 3$ .

	A	B	C
A	25	5	3
B	8	21	4
C	3	3	25

```

# Crear el objeto matriz en R
m = matrix(c(25, 8, 3, 5, 21, 3, 3, 4, 25), ncol = 3,
           dimnames = list(c('A', 'B', 'C'),c('A', 'B', 'C')))
# Calcular coeficiente kappa no ponderado y acuerdos parciales
Kappa(m, r = 0, partial = TRUE)

## $k
## [1] 0.5978954
##
## $se
## [1] 0.06735388
##
## $conf.int
## [1] 0.4658842 0.7299066
##
## $partial.kappa
##   Class      kappa      SE
## 1     A 0.5730832 0.08682231
## 2     B 0.5268293 0.09218257
## 3     C 0.6944512 0.07831869

```

El resultado indica que, globalmente, los dos observadores están de acuerdo en un 59.79% del total de posibles acuerdos no debidos al azar. Los acuerdos parciales indican que el mayor grado de acuerdo no debido al azar se produce en la clase  $C$  ( $\hat{\kappa}_C = 69.45\%$ ).

## 3.2 La función Delta()

La función `Delta()`, también llamada `delta()`, calcula el coeficiente de concordancia Delta para variables nominales, así como las distintas medidas de acuerdo de cada categoría. Los argumentos de entrada de `Delta()`, o `delta()`, son:

- `data` puede ser tanto una matriz de clasificación bidimensional o los datos en bruto de la clasificación de dos observadores.
- `standard` un valor lógico indicando si existe un observador que proporciona las clasificaciones correctas (`TRUE`), o no (`FALSE`, valor por defecto). Si existe, se asume

que corresponde con las filas de la matriz de clasificación o con la primera columna del conjunto de datos en bruto.

- **fixedRows** un valor lógico indicando si la distribución marginal de las filas está fijada de antemano (muestreo de tipo II, **TRUE**) o no (muestreo de tipo I, **FALSE**). Su valor por defecto es **FALSE**.
- **rawdata** un valor lógico indicando si los datos de entrada (argumento **data**) son en bruto o una matriz de clasificación. Si no se especifica nada (su valor por defecto es **NULL**), la función **Delta()** se encarga de averiguar el tipo de entrada.
- **tol** valor numérico que indica el error máximo permitido al aproximar el valor de la constante desconocida B a su valor real. Su valor por defecto es de  $10^{-7}$ .
- **mxits** número entero que indica el número máximo de iteraciones que puede realizar el método numérico para aproximar el valor de la constante desconocida B a su valor real. Su valor por defecto es 100.

El resultado de esta función es un objeto de clase **"delta"**, que es una lista de nueve elementos, o diez (si el número de categorías es igual a dos):

- **Delta** lista de dos elementos (objeto de clase **"validMeasuresDelta"**) que contiene:
  - la estimación de los parámetros del modelo (**\$Estimates**): delta global ( $\hat{\Delta}$ ), delta parcial para cada categoría ( $\hat{\Delta}_i$ ) y la distribución de respuestas realizadas al azar por el observador correspondiente a las columnas de la matriz de clasificación ( $\hat{\pi}_i$ ); así como las medidas de acuerdo válidas para el modelo asumido (según muestreo tipo I y II y presencia o no de estándar), que pueden ser: acuerdo ( $\hat{A}_i$ ), conformidad ( $\hat{F}_i$ ), predictividad ( $\hat{P}_i$ ) y consistencia ( $\hat{S}_i$ ).
  - el error estándar (**\$SE**) de las estimaciones bajo el muestreo asumido.
- **Kappa** la estimación (**\$Estimate**) y error estándar (**\$SE**) del coeficiente kappa.
- **Data** los datos de entrada (**\$input**) y los analizados (**\$Analyzed**), que pueden o no ser los mismos.
- **GOF** el test  $\chi^2$  de bondad de ajuste del modelo (objeto de clase **"chisqTestDelta"**). Este elemento es una lista de 5 elementos: el estadístico del test (**\$statistic**); los grados de libertad de la distribución  $\chi^2$  (**\$df**); el p-valor calculado (**\$pval**); la tabla de frecuencias observadas (**\$Observed**); y la tabla de frecuencias esperadas por el modelo (**\$Expected**). La significación del test indica que el modelo Delta no es adecuado para analizar el problema.
- **fixedRows** un valor lógico que coincide con el argumento de entrada del mismo nombre. Será utilizado por el método **summary()** implementado para objetos de clase **"delta"**.

- `standard` un valor lógico que coincide con el argumento de entrada del mismo nombre. Será utilizado por el método `summary()` implementado para objetos de clase `"delta"`.
- `all.measures` lista de cinco elementos (objeto de clase `"measuresDelta"`) que contiene la estimación de los parámetros del modelo y todas las medidas de acuerdo, independientemente del tipo de muestreo o de la presencia de observador estándar.
- `problem.parameters` lista de tres elementos (objeto de clase `"defP"`) que contiene la estimación de la constante desconocida B (`$B`), la matriz analizada para aproximar el resultado (`$analyzedData`) y una lista de 10 elementos con información acerca de la estimación de B (`$info`).
- `cov` lista de tres elementos (objeto de clase `"cov.Delta"`) que contiene la matriz de covarianzas de las estimaciones parciales de delta (`$Cov_delta`), la matriz de covarianzas de la distribución de respuestas realizadas al azar (`$cov_Pi`) y la matriz de covarianzas mixta de los dos parámetros anteriores (`$cox_Mix`).
- `asymptoticDelta` lista de dos elementos que contienen el análisis asintótico realizado sobre matrices  $2 \times 2$ . El primer elemento (`$validMeasures`) contiene las estimaciones válidas para el modelo asumido, mientras que el segundo (`$allMeasures`) contiene todas las medidas independientemente del tipo de muestreo asumido o la presencia o no de observador estándar. A su vez, cada lista contiene otras dos listas: `$DA_A1` corresponde a la aproximación utilizando la matriz de clasificación original, y `$DA_A2` corresponde a la aproximación que utiliza los datos de la matriz original incrementados en una unidad.

Internamente, `Delta()` llama a diversas funciones auxiliares no exportadas (no visibles por el usuario), pero que siempre pueden ser utilizadas mediante el operador `:::`. Las siguientes subsecciones detallan las funciones auxiliares utilizadas por `Delta()`.

### 3.2.1 Depuración de los datos de entrada: `getM1()`

Los datos de entrada pueden ser de dos tipos:

- Tipo 1: una matriz cuadrada de orden  $K$ , como la mostrada en la [Tabla 1.1](#), donde la entrada  $x_{ij}$  es la frecuencia con que el observador por filas (**R**) clasifica el objeto como perteneciente a la clase  $i$  y el observador por columnas (**C**) lo clasifica como perteneciente a la clase  $j$ .
- Tipo 2: una tabla de datos en bruto, donde la primera columna corresponderá con el observador por filas (**R**) y la segunda con el observador por columnas (**C**). Cada entrada de la tabla,  $t_{R_i}$  y  $t_{C_i}$ , indica la clase (categoría) en la que los observadores **R** y **C** clasifican el objeto  $i$ -ésimo. A este tipo de entrada se le permite tener una columna extra, indicativa del índice del elemento clasificado.

Objeto	<b>R</b>	<b>C</b>
1	$t_{R_1}$	$t_{C_1}$
2	$t_{R_2}$	$t_{C_2}$
$\vdots$	$\vdots$	$\vdots$
$n$	$t_{R_n}$	$t_{C_n}$

La función principal para depurar los datos de entrada es `getM1()`, que tiene dos argumentos: `x`, los datos de entrada (tipo 1 o tipo 2); y `is.raw`, un valor lógico que indica si `x` son datos en bruto o no. Por defecto, `is.raw = NULL`, de manera que la función trata de averiguar el tipo de datos de entrada. El resultado de esta función es un objeto de clase "M1" y "table", esto es una tabla de frecuencias cuadrada y sin clases vacías (es decir, sin clases  $i$  que verifiquen  $r_i = c_i = 0$ ). Esta función utiliza dos funciones auxiliares, `table.ct()` y `checkTable()`, que garantizan este resultado.

El procesamiento de la entrada es el siguiente:

- Si `is.raw` toma un valor `TRUE` o `FALSE`, se comprueba la clase del objeto `x`, de manera que si es de tipo "matrix", se convierte a "data.frame" o "table", según si son datos en bruto (`is.raw = T`) o son frecuencias (`is.raw = F`), respectivamente.
  - Si `x` es de tipo 2 (datos en bruto), se le aplica la función `table.ct()`. Esta función comprueba el número de columnas de `x` y si hay 3, elimina una de ellas, asumiendo que la que tiene mayor número de valores distintos es la correspondiente al índice de las filas. Una vez eliminada la columna sobrante, si la hubiera, se guarda un mensaje de aviso de qué columna se ha eliminado, el cual aparecerá cuando se imprima por pantalla el resultado de la función `Delta()`, y se calcula la tabla de frecuencias con las otras 2 columnas, asegurándonos de que la matriz resultante es cuadrada y sin clases vacías. Por último, esta función añade cuatro atributos a la matriz depurada: `$classes`, un vector de nombres de las clases del objeto de entrada; `$valid_classes`, un vector de nombres de las clases válidas; `$empty_classes`, un vector de nombres de las clases vacías; y `$messages`, los avisos que se han ido guardando al depurar los datos de entrada.
  - Si `x` es de tipo 1 (matriz de frecuencias), se le aplica la función `checkTable()`, que devuelve siempre una matriz cuadrada y, en su caso, elimina las clases vacías. Esta función llama internamente a la función auxiliar anterior, `table.ct()`, de manera que el objeto resultante tiene los mismos atributos mencionados anteriormente.
- Si `is.raw = NULL`, entonces se intenta averiguar el tipo de entrada de datos.
  - Si el objeto `x` no es cuadrado o sus valores son no numéricos entonces se asume que son datos en bruto, se guarda un mensaje de aviso y se comprueba el número

de columnas que tiene. Si el número de columnas es inferior a 2 o superior a 3, la función arroja un error. Si hay 2 o 3 columnas, y `x` es de clase `"matrix"`, se convierte a `"data.frame"` y se le aplica la función `table.ct()`.

- Si el objeto `x` es cuadrado o de clase `"table"`, se asume que es una tabla de frecuencias, se guarda un mensaje de aviso y se aplica la función `checkTable()`.

El siguiente ejemplo muestra el uso de la función `getM1()`. Cuando no se usa el argumento `is.raw`, la función averigua el tipo de entrada de datos.

```
# Entrada de datos de tipo 1 (tabla de frecuencias)
m = matrix(c(25, 8, 3, 5, 21, 3, 3, 4, 25), ncol = 3)
m = DeltaMAN:::getM1(m); m

##      C
## R    A B C
## A 25 5 3
## B  8 21 4
## C  3 3 25

class(m)

## [1] "table" "M1"

attr(m, 'valid_classes')

## [1] "A" "B" "C"

attr(m, 'messages')

## [1] "Assuming 3x3 contingency table"

# Entrada de datos de tipo 2 (en bruto)
df = data.frame(ID = 1:80,
                R = sample(c('A', 'B', 'C'), 80, replace = T),
                C = sample(c('A', 'B', 'C'), 80, replace = T))
df = DeltaMAN:::getM1(df); df
```

```
##      C
## R    A  B  C
##   A  7 10  4
##   B  9 13 14
##   C  5 10  8

class(df)

## [1] "table" "M1"

attr(df, 'valid_classes')

## [1] "A" "B" "C"

attr(df, 'messages')

## [1] "Assuming raw data"
## [2] "Column ID is assumed to be an ID column and will be removed"
```

### Otras matrices: `getM2()`, `getM3()` y `getM4()`

Hasta ahora se ha definido la matriz depurada de clase "M1", que es cuadrada y sin clases vacías. A partir de esta matriz se pueden obtener las siguientes:

- Matriz de clase "M2", obtenida mediante la función `getM2()`, es una ampliación de la matriz "M1" de orden 2. Se construye añadiendo una clase extra, donde  $x_{33} = 1$ , e incrementando los datos en +0.5.
- Matriz de clase "M3", obtenida mediante la función `getM3()`, incrementa los datos en +0.5.
- Matriz de clase "M4", obtenida mediante la función `getM4()`, incrementa los datos en +1 y se construye solo en los casos en que la matriz "M1" es de orden 2.

### 3.2.2 Obtención de la constante desconocida B: getB()

La función `getB()` determina la constante desconocida  $B$ , que tiene que cumplir la restricción  $B = n(1 - \hat{\Delta}) \geq B_0 = \max_i \{\sqrt{c_i - x_{ii}} + \sqrt{r_i - x_{ii}}\}^2$ , y es solución de la [Ecuación \(2.5\)](#).

Esta función tiene tres argumentos de entrada: `M`, un objeto de clase "M1" (también "M2" o "M3"); `tol`, el error máximo tolerado para la aproximación; y `mxits`, el máximo número de iteraciones realizadas por el método numérico para aproximar  $B$ .

La función `getB()` hace uso de cinco funciones auxiliares para estimar el valor de  $B$ :

- `.getB0()`, calcula el valor  $B_0 = \max_i \{\sqrt{c_i - x_{ii}} + \sqrt{r_i - x_{ii}}\}^2$ . La función devuelve un vector de un elemento con el valor  $B_0$ , que contiene un atributo (`$h`) que indica la categoría que proporciona ese valor.
- `.getType()`, devuelve el tipo de problema para buscar la solución de  $B$  en  $y^+(B) = 0$  o  $y^-(B) = 0$ .
  - si  $y^-(B_0) > 0$ , la solución  $B$  se busca en  $y^-(B)$ .
  - si  $y^-(B_0) < 0$ , y
    - \* si  $(n - \sum x_{ii}) > (c_h + r_h - 2x_{hh})$ , la solución  $B$  se busca en  $y^+(B)$ .
    - \* si  $(n - \sum x_{ii}) = (c_h + r_h - 2x_{hh})$ ,  $B$  tiene infinitas soluciones. El problema se resuelve utilizando la tabla "M3".
  - si  $y^-(B_0) = 0$ , y
    - \* si  $(n - \sum x_{ii}) > (c_h + r_h - 2x_{hh})$ ,  $B = B_0$ .
    - \* si  $(n - \sum x_{ii}) = (c_h + r_h - 2x_{hh}) > 0$ ,  $B$  tiene infinitas soluciones. El problema se resuelve utilizando la tabla "M3".
    - \* si  $(n - \sum x_{ii}) = (c_h + r_h - 2x_{hh}) = 0$ ,  $B = 0$ .
- `.y()`, define la [Ecuación \(2.5\)](#) a resolver y devuelve el valor  $y(B)$ .
- `.dydb()`, aproxima la derivada de la función  $y(B)$  como  $y'(B) = \frac{y(B+0.01) - y(B)}{0.01}$ .
- `.newton_raphson()`, resuelve la ecuación  $y(B) = 0$  mediante el método de Newton-Raphson.

El resultado de la función `getB()` es una lista de clase "defP" que contiene el valor de  $B$  obtenido; la matriz de datos analizada, que puede ser distinta a la original si el problema tiene infinitas soluciones, en cuyo caso se aumenta la matriz original en +0.5; y una lista

con información sobre distintos valores utilizados para resolver la ecuación, como  $B_0$  o el tipo de problema.

Por ejemplo, utilizando la matriz  $\mathbf{m}$  definida en el apartado anterior, la solución de la ecuación  $y(B) = 0$  es 40.451. Se ha implementado un método `print()` para objetos de clase "`defP`", que da formato al resultado y permite devolverlo tanto en texto plano (`tex = FALSE`) como en formato L<sup>A</sup>T<sub>E</sub>X (`tex = TRUE`), así como mostrar un número concreto de dígitos decimales (`digits`).

```
B = DeltaMAN:::getB(m)
print(B, tex = TRUE, digits = 3)
```

B = 40.451 (6 iterations)

h	B0	PType	$Y^-$	$dY^-$	$Y^+$	$dY^+$	$n - s$	$c + r - 2x$	
1	2.000	39.596	1.000	7.876	-65.178	7.876	60.045	26.000	20.000

### 3.2.3 Cálculo de los parámetros del modelo: `get_parameters()`

Una vez obtenido el valor de la constante  $B$  se pueden calcular los parámetros del modelo Delta. La función `get_parameters()` calcula los parámetros  $\hat{\Delta}$ ,  $\hat{\Delta}_i$  y  $\hat{\pi}_i$  usando las expresiones indicadas en las Ecuaciones (2.6) a (2.8).

Esta función tiene dos argumentos de entrada:  $\mathbf{M}$ , un objeto de clase "`M1`" (también "`M2`" o "`M3`"); y `defP`, un objeto de clase "`defP`" obtenido con la función `getB()`. El resultado de la función `get_parameters()` es una lista de clase "`psetDelta`" que contiene la estimación de  $\hat{\Delta}$ ,  $\hat{\Delta}_i$  y  $\hat{\pi}_i$ .

Por ejemplo, utilizando la matriz  $\mathbf{m}$  y la solución de  $B$  obtenida en el apartado anterior, obtenemos los siguientes parámetros del modelo Delta:

```
DeltaMAN:::get_parameters(m, B)
```

```
## Class Delta Pi
## 1 A 0.590 0.409
## 2 B 0.415 0.378
## 3 C 0.754 0.213
## Overall Delta: 0.583
```

### 3.2.4 Verificación del modelo: `chisq.test.Delta()`

La validez de  $\hat{\Delta}$  está sujeta a la adecuación del modelo subyacente para analizar un problema dado. El test apropiado para verificar la validez del modelo es el clásico test  $\chi^2$  de bondad de ajuste. Las frecuencias esperadas por el modelo Delta se calculan según lo indicado en la [Ecuación \(2.9\)](#).

La función `chisq.test.Delta()` implementa el test para el modelo Delta. Esta función toma dos argumentos de entrada: `M`, un objeto de clase "M1" (también "M2" o "M3"); y `psetDelta`, un objeto de clase "psetDelta" obtenido con la función `get_parameters()`. Si el segundo argumento no se proporciona, los parámetros son calculados a partir de la matriz `M`.

El resultado de esta función es una lista de clase "`chisqTestDelta`", que contiene el valor del estadístico del test (`$statistic`); los grados de libertad de la distribución  $\chi^2$  (`$df`); el p-valor calculado (`$pval`); la tabla de frecuencias observadas (`$Observed`); y la tabla de frecuencias esperadas por el modelo (`$Expected`). La significación del test indica que el modelo Delta no es adecuado para analizar el problema.

Si el test es inválido, los elementos `$statistic`, `$df` y `$pval` devuelven el valor `NA` y el atributo `$validity` indica el motivo por el cual el test es inválido: o bien alguna de las frecuencias esperadas por el modelo es menor que 1, o más de un 20% de ellas es menor que 5.

Siguiendo con el ejemplo de la matriz `m`, el resultado del test de bondad de ajuste es:

```
DeltaMAN::chisq.test.Delta(m)
```

```
## Observed frequencies:
##      C
## R    A  B  C
## A 25  5  3
## B  8 21  4
## C  3  3 25
##
## Expected frequencies:
##      C
## R    A    B    C
## A 25.00  5.12  2.88
## B  7.88 21.00  4.12
## C  3.12  2.88 25.00
```

```
##
## Chi-square Goodness of fit:
## X-squared = 0.0211 , df = 1 , p-value = 0.884
```

A un nivel de significación del 5%, dado que el p-valor obtenido es mayor que  $\alpha$ , no se rechaza la hipótesis nula de que el modelo Delta es adecuado para analizar este problema.

### 3.2.5 Cálculo de varianzas y covarianzas: var.Delta() y cov.Delta()

Con el objetivo de calcular los errores estándar de las estimaciones, previamente es necesario calcular las matrices de covarianzas.

Las funciones `var.Delta()` y `cov.Delta()` toman tres argumentos de entrada: `M`, un objeto de clase "M1" (también "M2" o "M3"); `B`, el valor de la solución de la ecuación  $y(B) = 0$ , resultado de `getB()$B$root`; y `psetDelta`, un objeto de clase "psetDelta" obtenido con la función `get_parameters()`. Si el segundo y/o tercer argumento no se proporciona, estos se calculan a partir de la matriz `M`.

El resultado de `cov.Delta()` es una lista de clase "cov.Delta", que contiene la matriz de covarianzas de las estimaciones de los acuerdos parciales,  $\widehat{Cov}(\hat{\Delta}_i, \hat{\Delta}_j)$ , (`$Cov_delta`), calculada según la Ecuación (2.10); la matriz de covarianzas de la distribución de respuestas realizadas al azar,  $\widehat{Cov}(\hat{\pi}_i, \hat{\pi}_j) = -\frac{E_i E_j}{E} + \delta_{ij} E_i$ , (`$cov_Pi`); y la matriz de covarianzas mixta de los dos parámetros anteriores,  $\widehat{Cov}(\hat{\Delta}_i, \hat{\pi}_j) = \frac{v_i E_i E_j}{E} - \delta_{ij} v_i E_i$ , (`$cox_Mix`).

Siguiendo con el ejemplo de la matriz `m`, las matrices de covarianzas son:

```
print(DeltaMAN:::cov.Delta(m), tex = T, digits = 3)
```

$Cov(\hat{\Delta}_i, \hat{\Delta}_j)$				$Cov(\hat{\Delta}_i, \hat{\pi}_j)$				$Cov(\hat{\pi}_i, \hat{\pi}_j)$			
	A	B	C		A	B	C		A	B	C
A	0.023	-0.009	-0.000	A	-0.011	0.009	0.002	A	0.016	-0.013	-0.002
B	-0.009	0.033	-0.001	B	0.012	-0.016	0.004	B	-0.013	0.017	-0.004
C	-0.000	-0.001	0.009	C	0.001	0.001	-0.002	C	-0.002	-0.004	0.006

Por su parte, la función `var.Delta()` llama internamente a `cov.Delta()` y utiliza la matriz  $\widehat{Cov}(\hat{\Delta}_i, \hat{\Delta}_j)$  para calcular las varianzas de las estimaciones como se indica en la Tabla 2.2. El resultado de `var.Delta()` es una lista de clase "var.Delta", que contiene las varianzas estimadas de las medidas de acuerdo para el muestreo de tipo I (`$Sampling_I`) y II (`$Sampling_II`). Estos resultado se utilizan para calcular el error estándar de las estimaciones en la función `get_measure()`, explicada en la siguiente sección.

### 3.2.6 Estimación de las medidas de acuerdo: `get_measures()`

Esta función calcula la estimación y error estándar para el muestreo de tipo I y II (independientemente de las condiciones seleccionadas) de las distintas medidas de acuerdo corregidas por azar: acuerdo parcial ( $\hat{\mathcal{A}}_i$ ), conformidad ( $\hat{\mathcal{F}}_i$ ), predictividad ( $\hat{\mathcal{P}}_i$ ) y consistencia ( $\hat{\mathcal{S}}_i$ ); según las expresiones indicadas en la [Tabla 2.2](#).

La función toma tres argumentos de entrada: `M`, un objeto de clase "M1" (también "M2" o "M3"); `B`, el valor de la solución de la ecuación  $y(B) = 0$ , resultado de `getB()$B$root`; y `psetDelta`, un objeto de clase "psetDelta" obtenido con la función `get_parameters()`. Si el segundo y/o tercer argumento no se proporciona, estos se calculan a partir de `M`.

La función devuelve una lista de clase "measuresDelta" con cinco elementos:

- `$Global`, un objeto de clase "psetDelta" que contiene la estimación de  $\hat{\Delta}$ ,  $\hat{\Delta}_i$  y  $\hat{\pi}_i$ . A este objeto se le añaden en forma de atributos el error estándar de  $\Delta$  bajo los muestreo de tipo I y II (`attr(*$Global, "SE_Delta")`).
- `$Agreement`, una lista de tres elementos que contiene la estimación de los acuerdos parciales,  $\hat{\mathcal{A}}_i$ , y sus errores estándar para los muestreos de tipo I y II.
- `$Conformity`, una lista de tres elementos que contiene la estimación de la conformidad en cada categoría,  $\hat{\mathcal{F}}_i$ , y sus errores estándar para los muestreos de tipo I y II.
- `$Predictivity`, una lista de tres elementos que contiene la estimación de la predictividad en cada categoría,  $\hat{\mathcal{P}}_i$ , y sus errores estándar para los muestreos de tipo I y II.
- `$Consistency`, una lista de tres elementos que contiene la estimación de la consistencia en cada categoría,  $\hat{\mathcal{S}}_i$ , y sus errores estándar para los muestreos de tipo I y II.

Se ha implementado un método `print()` para objetos de clase "measuresDelta", que permite devolver el resultado en formato de texto plano (`tex = FALSE`) o L<sup>A</sup>T<sub>E</sub>X (`tex = TRUE`) y elegir el número de decimales a mostrar (`digits`). El resultado obtenido tiene forma tabular, mostrando las medidas estimadas y sus errores estándar en las columnas y las categorías de la variable nominal en las filas.

```
measures = DeltaMAN:::get_measures(m)
print(measures, tex = TRUE, digits = 3)
```

Class	Delta	Pi	Agreement		Conformity		Predictivity		Consistency	
A	0.590	0.409	0.201	(0.0593)* (0.0520)**	0.590	(0.1529)* (0.1529)**	0.541	(0.1428)*	0.564	(0.1433)*
B	0.415	0.378	0.141	(0.0653)* (0.0622)**	0.415	(0.1827)* (0.1827)**	0.472	(0.2056)*	0.442	(0.1909)*
C	0.754	0.213	0.241	(0.0466)* (0.0299)**	0.754	(0.0935)* (0.0935)**	0.730	(0.0935)*	0.742	(0.0834)*
Overall Delta:			0.583	(0.0728)* (0.0714)**						

\* Valid under type-I sampling (columns and rows totals are at random)

\*\* Valid under type-II sampling (rows totals are fixed beforehand)

### Medidas de acuerdo válidas: validMeasures()

Asociada a esta, la función `validMeasures` devuelve solo las medidas válidas bajo el modelo Delta asumido, de acuerdo con la [Tabla 2.1](#). Esta función toma tres argumentos de entrada: `measuresDelta`, un objeto de clase `"measuresDelta"` obtenido mediante la función `get_measures()`; `standard`, un valor lógico indicando si existe un observador que proporciona las clasificaciones correctas (`TRUE`), o no (`FALSE`); y `fixedRows`, un valor lógico indicando si el muestreo es de tipo I (`fixedRows = FALSE`) o de tipo II (`fixedRows = TRUE`).

El resultado de la función `validMeasures` es una lista de clase `validMeasuresDelta` que contiene dos elementos:

- `$Estimates` las estimaciones de los parámetros y medidas de acuerdo válidas para el modelo asumido.
- `$SE` el error estándar de las estimaciones válidas del modelo asumido.

Al igual que antes, se ha implementado un método `print()` para objetos de clase `"validMeasuresDelta"`, que incluye los argumentos `tex` y `digits`.

```
validM = DeltaMAN::validMeasures(measures, standard = T, fixedRows = F)
print(validM, tex = TRUE, digits = 3)
```

Class	Delta	Pi	Agreement		Conformity		Predictivity	
A	0.590	0.409	0.201	(0.0593)	0.590	(0.1529)	0.541	(0.1428)
B	0.415	0.378	0.141	(0.0653)	0.415	(0.1827)	0.472	(0.2056)
C	0.754	0.213	0.241	(0.0466)	0.754	(0.0935)	0.730	(0.0935)
Overall Delta:			0.583	(0.0728)				

### 3.2.7 Visualización de resultados: `print()` y `summary()`

Se ha implementado un método `print()` para objetos de clase `"delta"`, que únicamente devuelve los mensajes producidos durante la fase de depuración de datos y la estimación y error estándar de  $\hat{\Delta}$  para el tipo de muestreo asumido. Siguiendo con el ejemplo de la matriz `m`, el resultado de `delta()` dado un muestreo de tipo I y la existencia de un observador estándar por filas es:

```
D = delta(m, standard = TRUE, fixedRows = TRUE); D

## Assuming 3x3 contingency table
##
## Delta = 0.583, SE = 0.0728
```

Este método `print()`, así como los otros seis mencionados en los anteriores apartados son funciones no exportadas. Se ha creado un método `print()` genérico para objetos de clase `"deltaMAN"`, que es cualquier objeto creado por el paquete `DeltaMAN`. Es decir, los objetos de clase `"defP"`, `"psetDelta"`, `"chisqTestDelta"`, `"delta"`, etc., también son objetos de clase `"deltaMAN"`. Este método genérico llama internamente a los métodos `print()` específicos de cada clase. El motivo de crear el método genérico es tener una única función `print()` exportada, de manera que el método específico usado es transparente para el usuario final.

Además, se ha implementado un método `summary()` para objetos de clase `"delta"`, que elabora un informe básico (`fullReport = FALSE`), incluyendo la estimación y errores estándar de  $\hat{\Delta}$  y  $\hat{\kappa}$ , así como de las medidas de concordancia bajo las condiciones seleccionadas, o un informe completo (`fullReport = TRUE`), que incluye las matrices de covarianzas y todas las medidas de concordancia independientemente del tipo de muestreo. Para dar más versatilidad, `summary()` también incorpora un argumento `tex` que da formato al resultado y lo devuelve o bien en texto plano (`tex = FALSE`) o en formato  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  (`tex = TRUE`). Finalmente, es posible especificar el número de dígitos decimales (`digits`) con el que mostrar los resultados (por defecto, se muestran 4).

A continuación se muestra el informe básico construido a partir del objeto `D`.

```
summary(D, tex = TRUE, fullReport = FALSE, digits = 3)
```

#### SELECTED MODEL CONDITIONS

There is a gold standard classification by rows. Row marginal frequencies have not been set beforehand (type I sampling).

**STANDARD ANALYSIS****Input data**

Assuming 3x3 contingency table. Valid classes detected: 3

	A	B	C
A	25	5	3
B	8	21	4
C	3	3	25

**Summary: Goodness of fit, Kappa and Delta**

- $\chi^2 = 0.021$ ,  $df = 1$ ,  $p - value = 0.884$
- $\kappa (SE) = 0.598 (0.0674)$
- $\Delta (SE) = 0.583 (0.0728)$

**Parameters of the Model (Delta and Pi) and the Measures (SE) of****Concordance under the selected conditions**

Class	Delta	Pi	Agreement		Conformity		Predictivity	
A	0.590	0.409	0.201	(0.0593)	0.590	(0.1529)	0.541	(0.1428)
B	0.415	0.378	0.141	(0.0653)	0.415	(0.1827)	0.472	(0.2056)
C	0.754	0.213	0.241	(0.0466)	0.754	(0.0935)	0.730	(0.0935)
Overall Delta:			0.583	(0.0728)				

**3.2.8 Análisis asintótico: asymptoticDelta()**

En el caso de tablas  $2 \times 2$ , se puede realizar el análisis Delta estándar añadiendo una clase ficticia extra, es decir, usando una matriz de clase "M2". Por otro lado, se puede realizar un análisis asintótico utilizando la función `asymptoticDelta()`, que toma como único argumento una matriz de clase "M1" o "M4" y devuelve un objeto de clase "measuresDelta", es decir, de la misma clase que se obtiene mediante la función `get_measures()`. Esta función utiliza la Ecuación (2.11) para estimar los parámetros de la solución asintótica y las expresiones indicadas en la Tabla 2.3 para estimar la varianza de las medidas de concordancia.

Cuando una tabla es de dimensión  $K = 2$ , la función `Delta()` realizará el análisis asintótico con las matrices original ("M1") y original incrementada en una unidad ("M4"), así como el análisis estándar con la matriz "M2". Por lo tanto, el informe resultante será más extenso que en el caso de tablas de dimensión  $K > 2$ . Veamos un ejemplo dejando las opciones por defecto.

```
m2 = matrix(c(15, 5, 4, 21), ncol = 2)
D2 = delta(m2); D2
```

```
## Assuming 2x2 contingency table
##
## Delta = 0.563, SE = 0.117
```

```
summary(D2, tex = TRUE, fullReport = FALSE, digits = 3)
```

### SELECTED MODEL CONDITIONS

None of the raters is a gold standard. Row marginal frequencies have not been set beforehand (type I sampling).

### STANDARD ANALYSIS

#### Input data

Assuming 2x2 contingency table. Valid classes detected: 2

	A	B
A	15	4
B	5	21

#### Summary: Goodness of fit, Kappa and Delta

- $\chi^2 = 0.000$ ,  $df = 1$ ,  $p - value = 0.988$
- $\kappa (SE) = 0.593 (0.1210)$
- $\Delta (SE) = 0.563 (0.1174)$

#### Parameters of the Model (Delta and Pi) and the Measures (SE) of Concordance under the selected conditions

Class	Delta	Pi	Agreement		Consistency	
A	0.513	0.499	0.219	(0.1684)	0.501	(0.3740)
B	0.601	0.453	0.344	(0.1718)	0.612	(0.2928)
Overall Delta:			0.563	(0.1174)		

**ASYMPTOTIC ANALYSIS****ASYMPTOTIC SOLUTIONS. I- Based on the original data**

**Parameters of the Model (Delta and Pi) and the Measures (SE) of Concordance under the selected conditions**

Class	Delta	Pi	Agreement		Consistency	
A	0.554	0.528	0.234	(0.0855)	0.540	(0.1495)
B	0.636	0.472	0.367	(0.0921)	0.648	(0.1156)
Overall Delta:			0.601	(0.1191)		

**ASYMPTOTIC SOLUTIONS. II- Based on the original data incremented +1**

**Parameters of the Model (Delta and Pi) and the Measures (SE) of Concordance under the selected conditions**

Class	Delta	Pi	Agreement		Consistency	
A	0.501	0.523	0.215	(0.0829)	0.489	(0.1490)
B	0.590	0.477	0.337	(0.0894)	0.601	(0.1181)
Overall Delta:			0.552	(0.1191)		

### 3.3 La función multiDelta()

La función `multiDelta()` permite realizar un análisis masivo de varios observadores contra un estándar. Los argumentos de entrada son:

- `data` un objeto de clase `"data.frame"`, cuya primera columna es el observador estándar.
- `which.measure` una cadena de texto indicando qué medida o medidas de interés devolver. Las opciones posibles son: `"Delta"`, `"Agreement"`, `"Conformity"`, `"Predictivity"` y `"Consistency"`.
- `tol` valor numérico que indica el error máximo permitido al aproximar el valor de la constante desconocida B a su valor real. Su valor por defecto es de  $10^{-7}$ .
- `mxits` número entero que indica el número máximo de iteraciones que puede realizar el método numérico para aproximar el valor de la constante desconocida B a su valor real. Su valor por defecto es 100.

La función devuelve una lista de clase `"multiDelta"` (y también `"deltaMAN"`), que contiene tantos elementos como medidas han sido seleccionadas en el argumento `"which.measures"`. A su vez, cada elemento será una lista de tantos elementos como observadores no estándar tengamos, que contendrá la medida seleccionada y su error estándar para los tipos de muestreo que tengan sentido según la [Tabla 2.1](#). Por ejemplo, si `which.measures = c("Conformity", "Predictivity")`, cada sub-lista dentro de `$Conformity` tiene 3 elementos: la estimación de la conformidad para cada categoría (`$Measure`), el error estándar de la estimación bajo el muestreo de tipo I (`$SE_samplingI`) y el error estándar bajo el muestreo de tipo II (`$SE_samplingII`). En el caso de `$Predictivity`, cada sub-lista también contiene 3 elementos, pero el tercero (`$SE_samplingII`) tiene asignado el valor `NULL`.

El método `print()` implementado para objetos de clase `"multiDelta"` permite devolver el resultado en texto plano o formato  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  mediante el argumento `tex`, así como trasponer la matriz resultante en el caso de ser demasiado ancha, mediante el argumento `transpose`. Además, como en los demás métodos `print()` implementados, se puede especificar el número de dígitos decimales a mostrar mediante el argumento `digits`.

```
# Crear un data.frame con 8 observadores, el primero un estándar
dat = data.frame(replicate(8, sample(1:3, 120, replace = TRUE)))

# Calcular el modelo Delta de cada observador con el estándar
# y devolver las medidas de conformidad en cada categoría
mDelta = multiDelta(dat, which.measure = 'Conformity')
print(mDelta, tex = TRUE, transpose = TRUE)
```

## Conformity

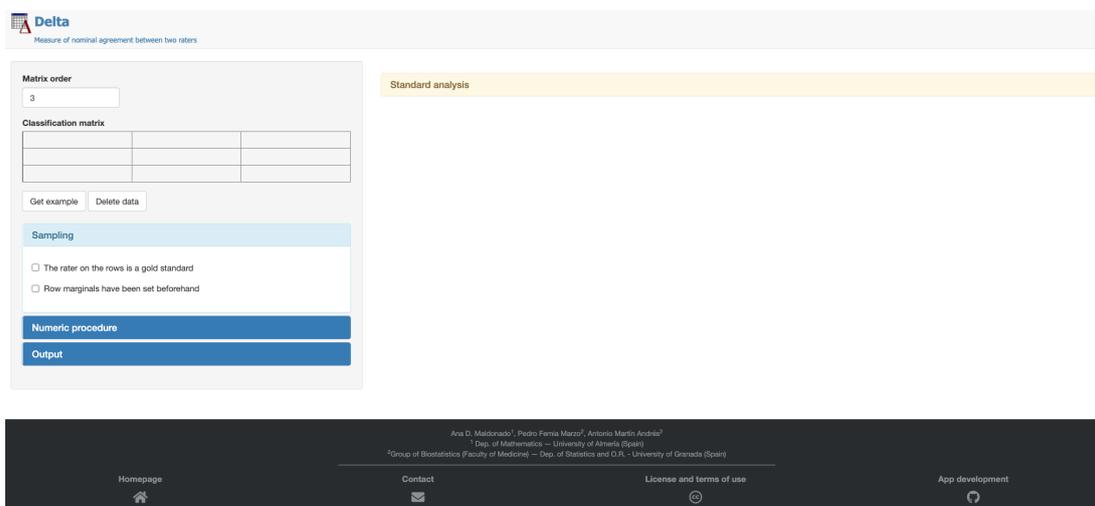
		1		2		3
X1.X2	0.0154	(0.13209)*	0.0327	(0.12863)*	-0.0785	(0.18384)*
		(0.13209)**		(0.12863)**		(0.18384)**
X1.X3	0.1486	(0.11768)*	0.0627	(0.15670)*	0.1467	(0.16528)*
		(0.11768)**		(0.15670)**		(0.16528)**
X1.X4	-0.1629	(0.16396)*	-0.0707	(0.12808)*	-0.1169	(0.13878)*
		(0.16396)**		(0.12808)**		(0.13878)**
X1.X5	-0.0661	(0.16305)*	0.1705	(0.12522)*	-0.1183	(0.16302)*
		(0.16305)**		(0.12522)**		(0.16302)**
X1.X6	0.3088	(0.10560)*	0.0118	(0.15950)*	-0.3153	(0.21946)*
		(0.10560)**		(0.15950)**		(0.21946)**
X1.X7	-0.2200	(0.16670)*	0.2527	(0.14397)*	0.1728	(0.13204)*
		(0.16670)**		(0.14397)**		(0.13204)**
X1.X8	-0.0120	(0.15433)*	-0.0022	(0.13610)*	-0.0323	(0.14506)*
		(0.15433)**		(0.13610)**		(0.14506)**

### 3.4 La función `runDeltaShinyApp()`

La función `runDeltaShinyApp()` permite ejecutar la aplicación de *shiny* para calcular el índice Delta entre dos observadores. Tan solo hay que llamarla del siguiente modo

```
runDeltaShinyApp()
```

y se abrirá una ventana como la mostrada en la [Figura 3.1](#). En la interfaz se distinguen dos zonas principales: un panel lateral, que sirve para introducir los datos y seleccionar distintas opciones; y un panel principal, en el que se muestran los resultados.



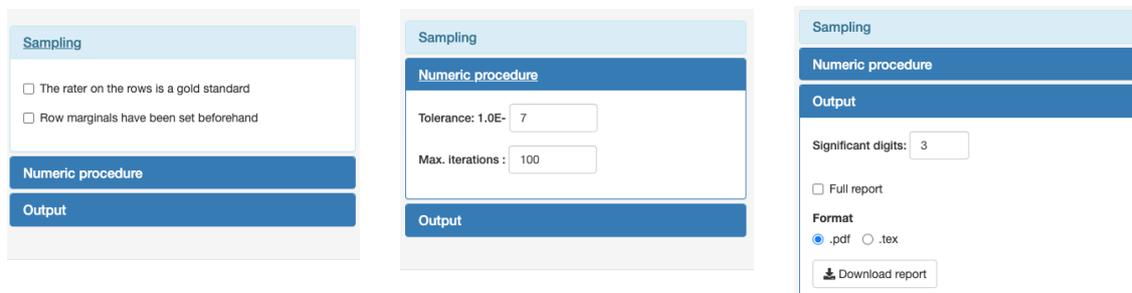
**Figura 3.1** Vista inicial de la aplicación Delta shiny.

**Panel lateral** Actualmente, la aplicación solo admite la entrada de datos de tipo 1. En el panel lateral se puede especificar el orden de la matriz (**Matrix order**) e introducir los datos en forma de tabla de frecuencias bidimensional (**Classification matrix**). El botón **Get example** proporciona una matriz de clasificación generada aleatoriamente, mientras que el botón **Delete data** elimina los datos introducidos.

Este panel contiene tres menús colapsables que, al pinchar sobre el nombre, se abren o cierran ([Figura 3.2](#)).

- En el menú **Sampling** ([Figura 3.2a](#)) se puede especificar el tipo de muestreo y si el observador de las filas es un estándar clicando sobre la caja correspondiente. El resultado mostrado en el panel principal se actualizará automáticamente.
- En el menú **Numeric procedure** ([Figura 3.2b](#)) se pueden cambiar los criterios de parada (tolerancia y número de iteraciones) del método numérico utilizado para estimar la constante  $B$ .

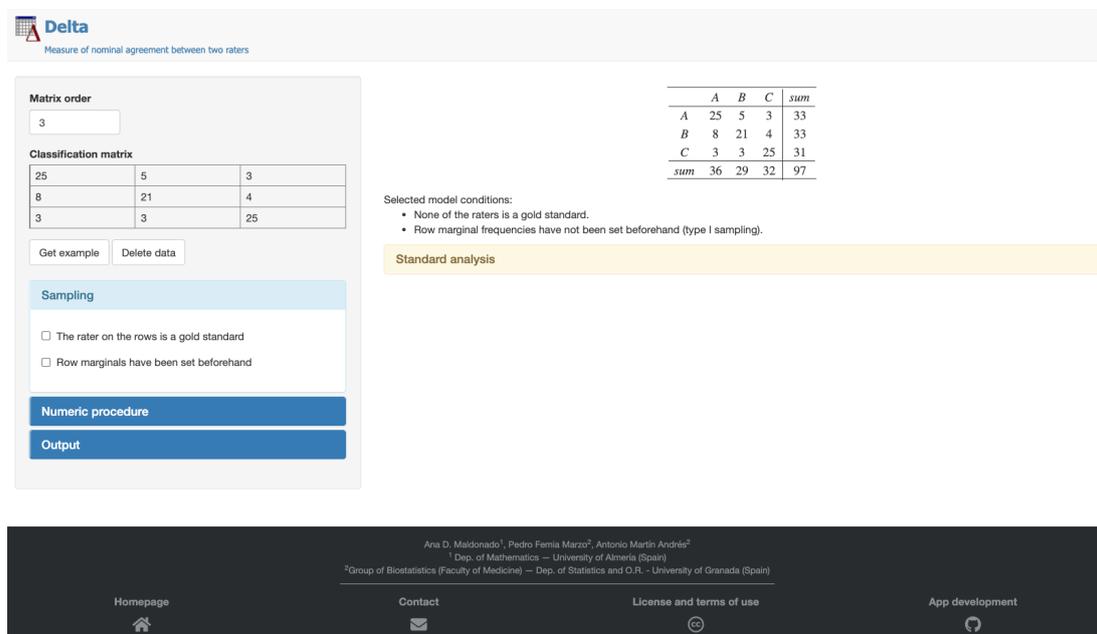
- En el menú **Output** (Figura 3.2c) se puede especificar el número de decimales a mostrar (**Significant digits**), así como si se desea exportar el informe completo (o básico), marcando (o no) la casilla **Full report**. La aplicación permite descargar dicho informe en formato **pdf** o **tex**, este último comprimido en un archivo **zip**.



(a) Opciones del modelo.      (b) Opciones numéricas.      (c) Exportar resultados.

**Figura 3.2** Selección de los supuestos del modelo Delta (a), de los criterios de parada del método numérico (b), y opciones para exportar los resultados (c).

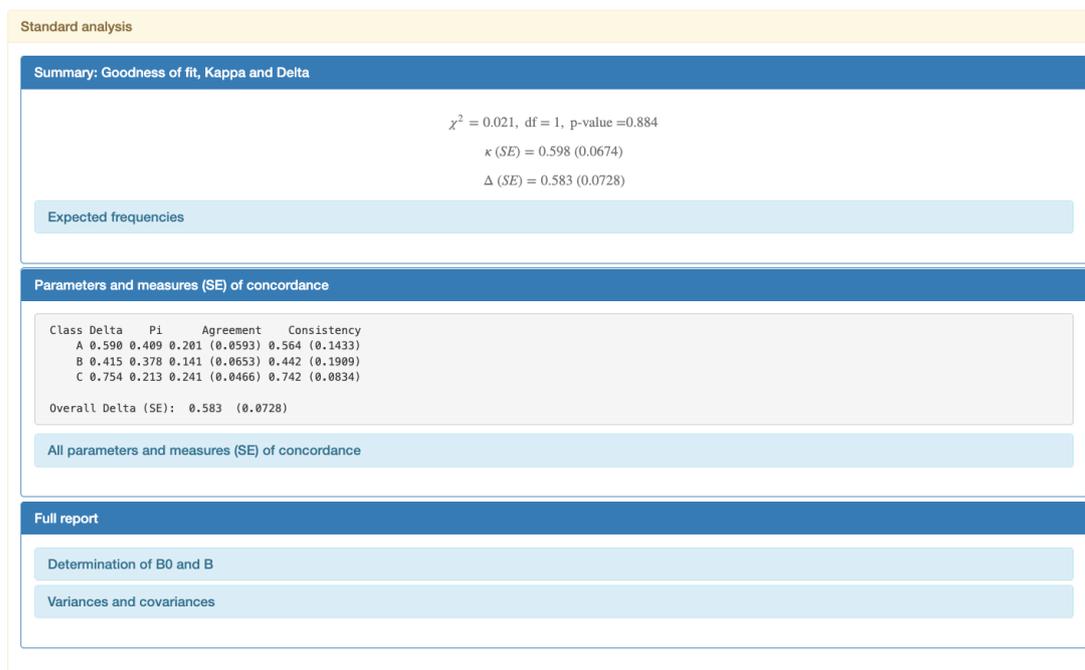
**Panel principal** Una vez introducida la matriz, pulsando en cualquier parte fuera del área donde se introducen los datos, se ejecuta internamente la función `Delta()` y se muestran los resultados en el panel principal, como muestra la Figura 3.3. Lo primero que se muestra es la matriz de clasificación introducida y las condiciones seleccionadas del modelo Delta (tipo de muestreo y presencia/ausencia de observador estándar).



**Figura 3.3** Resultado inicial mostrado por la aplicación Delta shiny.

A continuación, dentro del menú **Standard analysis** (Figura 3.4) aparecen los resultados en distintos menús colapsables, estructurados de la siguiente manera:

- **Summary: Goodness of fit, Kappa and Delta.** En este menú se muestran los índices  $\hat{\kappa}$  y  $\hat{\Delta}$  globales, así como el resultado del test de bondad de ajuste del modelo Delta.
  - **Expected frequencies.** En este menú se muestra la matriz de frecuencias esperadas por el modelo.
- **Parameters and measures (SE) of concordance.** En este menú se muestran las medidas de acuerdo válidas para las condiciones del modelo seleccionado, así como sus errores estándar.
  - **All parameters and measures (SE) of concordance.** En este menú se muestran todas las medidas de acuerdo y errores estándar, independientemente de las condiciones seleccionadas.
- **Full report.** Este menú engloba a los dos siguientes:
  - **Determination of B0 and B.** Este menú muestra el valor de  $B$  obtenido, así como información sobre distintos valores utilizados para resolver la [Ecuación \(2.5\)](#).
  - **Variances and covariances.** Este menú muestra las tres matrices de covarianzas calculadas, es decir,  $\widehat{Cov}(\hat{\Delta}_i, \hat{\Delta}_j)$ ,  $\widehat{Cov}(\hat{\pi}_i, \hat{\pi}_j)$  y  $\widehat{Cov}(\hat{\Delta}_i, \hat{\pi}_j)$ .



**Figura 3.4** Resultados del análisis estándar mostrados por Delta shiny.

En el caso de matrices  $2 \times 2$ , además del análisis estándar, el panel principal muestra las soluciones del análisis asintótico para los datos originales (ASYMPTOTIC SOLUTION I) y los incrementados en +1 (ASYMPTOTIC SOLUTION II), como muestra la Figura 3.5.

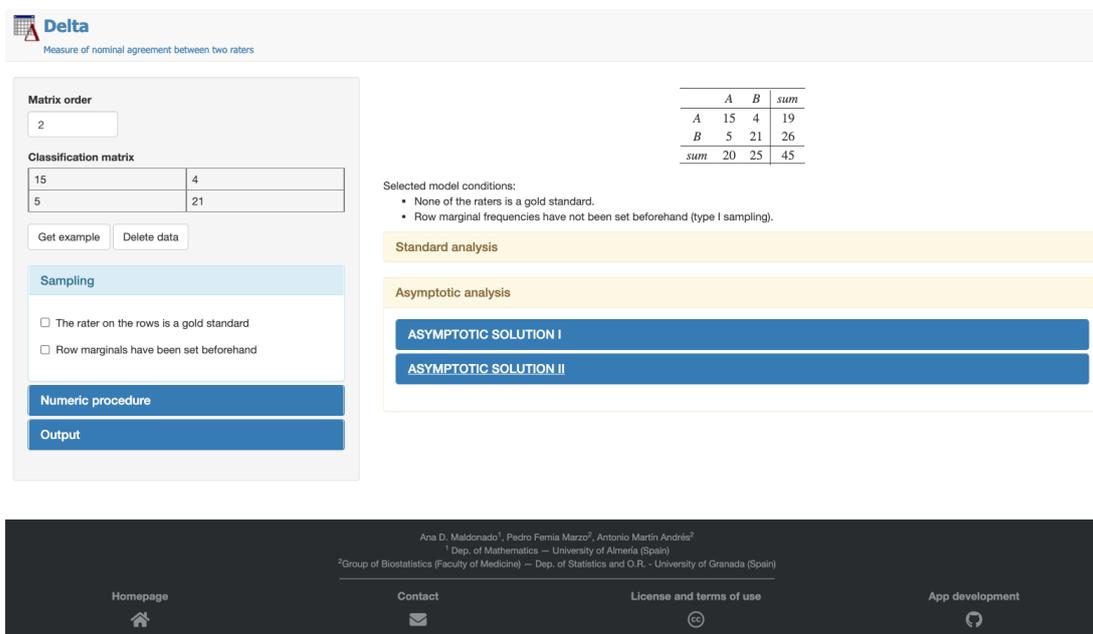


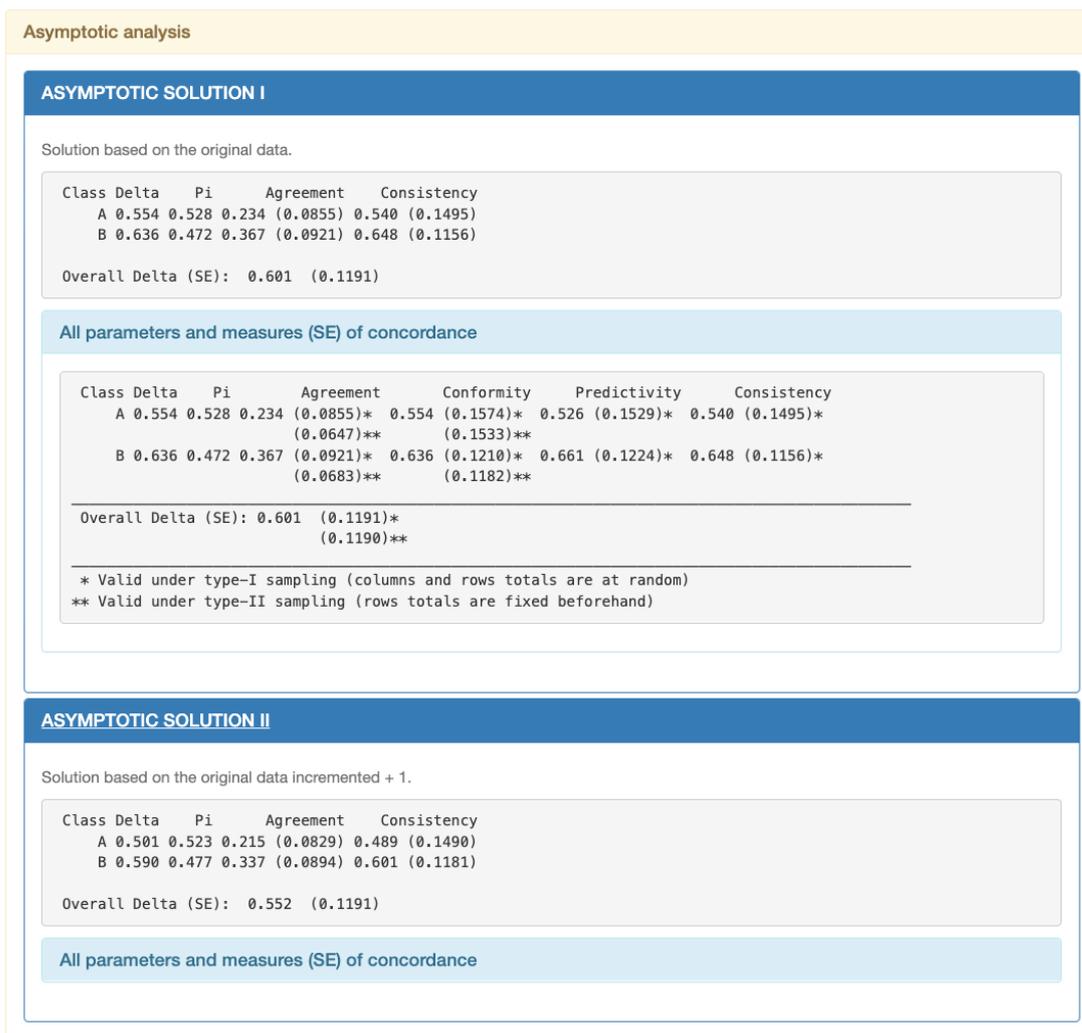
Figura 3.5 Vista de la aplicación Delta shiny con matrices de orden 2.

Dentro del menú para cada una de las dos soluciones asintóticas se muestran las medidas de acuerdo válidas para las condiciones del modelo seleccionado, así como otro menú colapsable que incluye todas las medidas de acuerdo (All parameters and measures (SE) of concordance), como muestra la Figura 3.6.

Por último, la aplicación muestra un pie de página con cuatro iconos:

- **Homepage** es un enlace a la web del grupo de Bioestadística de la Universidad de Granada.
- **Contact** abre el gestor de correo electrónico preparado para mandar un email a Pedro Femia y Ana Maldonado.
- **License and terms of use** abre una ventana con la licencia y términos de uso de la aplicación
- **App development** es un enlace a la página de GitHub de Ana Maldonado.

Alternativamente, se puede acceder a la aplicación Delta shiny a través de la siguiente dirección web: <https://admaldonado.shinyapps.io/deltaman/>.



**Figura 3.6** Resultados del análisis asintótico mostrados por Delta shiny.

El índice de acuerdo nominal Delta, formulado por Martín y Femia en 2004 [2], presenta una serie de ventajas con respecto al clásico índice kappa de Cohen. En primer lugar, kappa es muy sensible a las distribuciones marginales, lo cual hace que tenga mal comportamiento cuando los marginales están desequilibrados en igual sentido. En segundo lugar, la interpretación de kappa es menos intuitiva que la de Delta, ya que representa la “proporción de acuerdo sobre el total de posibles acuerdos no atribuibles al azar”, en lugar de simplemente la “proporción de acuerdos no debidos al azar”. Por último, Delta no solo proporciona una medida global, sino que permite evaluar el grado de acuerdo en cada categoría de forma directa, a diferencia de kappa, que precisa de colapsar la tabla de clasificación para obtener el grado de acuerdo en cada una.

No obstante, Delta presenta ciertas desventajas con respecto a kappa. El cálculo de Delta es más laborioso que el de kappa y, por lo general, no se puede hacer a mano. Sin embargo, una vez implementados ambos métodos en un programa de ordenador, su uso es igualmente sencillo. Además, el método numérico utilizado (Newton-Raphson) converge rápidamente a la solución, por lo que el cálculo de Delta también es muy rápido. Por otro lado, Delta está formulado para tratar únicamente cualidades nominales, a diferencia de kappa que puede ser generalizable para tratar cualidades ordinales.

En este trabajo se ha realizado una implementación del modelo Delta en R. El algoritmo implementado es consistente con los resultados devueltos por las aplicaciones desarrolladas con anterioridad [3, 4]. La principal ventaja de disponer de una implementación del modelo Delta en R es que puede ser utilizada en cualquier sistema operativo, a diferencia de [3], ya que R es un entorno multiplataforma. Además, R es uno de los lenguajes de programación más extendido actualmente en investigación científica, lo cual facilita la distribución y uso de la herramienta creada en este trabajo.

A nivel de usuario, el paquete DeltaMAN es muy sencillo de utilizar y está ampliamente documentado. Solo se han exportado (es decir, hecho visibles) las funciones que el usuario necesita utilizar para realizar el análisis de concordancia basado en Delta, dejando ocultas

todas las funciones auxiliares que son llamadas por la función principal. Esto permite que el paquete sea más intuitivo y fácil de usar para el usuario. No obstante, las funciones ocultas también pueden ser llamadas mediante el operador `:::`, ya que la filosofía de R es la del código abierto.

El método `summary` implementado para objetos de clase “delta” añade versatilidad y comodidad al poder obtener la salida de resultados directamente en código  $\LaTeX$ . Esta funcionalidad es aún más importante cuando se elaboran documentos mediante archivos de tipo Sweave (`.Rnw`). Este tipo de archivo proporciona un marco flexible para intercalar código de R y  $\LaTeX$ , combinando la potencialidad de ambos y ayudando a optimizar el tiempo invertido en generar informes.

La aplicación Delta shiny permite tener una primera toma de contacto con el paquete DeltaMAN de una forma aún más sencilla gracias a su interfaz gráfica. Además, tiene la ventaja de poder exportar los resultados del análisis en un archivo pdf o de  $\LaTeX$ , que puede ser editado por cualquier editor de texto. La aplicación se encuentra dentro del paquete DeltaMAN y, además, está alojada en el servidor shinyapps.io, por lo que se puede acceder a ella desde cualquier navegador y dispositivo, sin necesidad de tener instalado el paquete ni ser usuario de R.

## Bibliografía

- [1] J. Cohen, “A coefficient of agreement for nominal scales,” *Educational and psychological measurement*, vol. 20, no. 1, pp. 37–46, 1960.
- [2] A. Martín Andrés and P. Femia Marzo, “Delta: A new measure of agreement between two raters,” *British journal of mathematical and statistical psychology*, vol. 57, no. 1, pp. 1–19, 2004.
- [3] P. Femia-Marzo and A. Martín Andrés, “Delta (v. 4.1.4): A windows-based application to estimate delta, a measure of agreement between two raters on a nominal scale.” <http://www.ugr.es/local/bioest/Delta.exe>, 2016.
- [4] P. Femia-Marzo and A. Martín Andrés, “A web application to estimate delta, a measure of agreement between two raters on a nominal scale.” <https://wpd.ugr.es/~bioest/delta.php>, 2022.
- [5] J. Cohen, “Weighted kappa: nominal scale agreement provision for scaled disagreement or partial credit.,” *Psychological bulletin*, vol. 70, no. 4, p. 213, 1968.
- [6] J. L. Fleiss, J. Cohen, and B. S. Everitt, “Large sample standard errors of kappa and weighted kappa.,” *Psychological bulletin*, vol. 72, no. 5, p. 323, 1969.
- [7] A. Martín Andrés and J. D. Luna del Castillo, *Bioestadística+: Para las ciencias de la salud (5)*. Capitel Editores, 2004.
- [8] A. Martín Andrés and P. Femia Marzo, “Chance-corrected measures of reliability and validity in kk tables,” *Statistical methods in medical research*, vol. 14, no. 5, pp. 473–492, 2005.
- [9] P. Femia Marzo, A. Martín Andrés, and M. Álvarez Hernández, “El modelo delta para evaluar el grado de acuerdo entre dos observadores,” in *Modelización matemática de fenómenos del Medio Ambiente y la Salud*, Universidad Autónoma de Guerrero, 2012.
- [10] A. Martín Andrés and P. Femia Marzo, “Chance-corrected measures of reliability and validity in  $2 \times 2$  tables,” *Communications in Statistics—Theory and Methods*, vol. 37, no. 5, pp. 760–772, 2008.