

UNA FUNCIÓN EN R PARA EL MÉTODO DELPHI DE NUBE. APLICACIONES AL PRONÓSTICO EDUCACIONAL

*A FUNCTION IN R FOR CLOUD DELPHI METHOD. APPLICATIONS TO THE EDUCATIONAL
FORECAST*

Dr.C. Miguel Cruz Ramírez
mcruzr@uho.edu.cu
<http://orcid.org/0000-0002-1697-1624>
Universidad de Holguín, Cuba

Tipo de contribución: Artículo de investigación científica
Recibido: 28-03-2020
Aceptado para su publicación: 14-06-2020

Resumen: El método Delphi tiene su origen a mediados del siglo XX, en la corporación Rand de los Estados Unidos. Inicialmente, sus aplicaciones se enmarcaban en problemas de pronóstico, mientras que el procesamiento de la información se constreñía al control de indicadores estadísticos elementales. Con el paso del tiempo, el método Delphi diversificó sus aplicaciones y perfeccionó sus mecanismos para el procesamiento de la información. El presente trabajo muestra una automatización del Delphi de Nube, el cual tiene como objetivo integrar los conceptos borrosos del Delphi Fuzzy y la incertidumbre/aleatoriedad presente en los procesos estocásticos. El procedimiento consiste en una función programada sobre el lenguaje estadístico R, de código abierto. Después de describir las características del Delphi de Nube, se presenta la nueva función CloudDelphi y se ejemplifica su utilidad y flexibilidad, durante la implementación de un pronóstico sobre los resultados académicos de un centro escolar. Finalmente, se valoran otras aplicaciones en escenarios del contexto educacional, donde es necesario realizar estimaciones y pronósticos con base en el conocimiento experto.

Palabras clave: Delphi de nube; modelo de nube; investigación educacional; toma de decisiones; R

Abstract: Delphi method emerged in the middle of the XXth century, in the Rand Corporation of the United States. Initially, its applications were framed in forecasting problems, while the information processing was limited to the control of elementary statistical indicators. Over time, the Delphi method diversified its applications and perfected its mechanisms for information processing. This paper presents an automation of the Cloud Delphi, which aims to integrate the slurred concepts of the Delphi Fuzzy and the uncertainty/randomness present in stochastic processes. The procedure consists of a function programmed in the statistical language R, open source. After describing the characteristics of the Cloud Delphi, the new CloudDelphi function is presented and its usefulness and flexibility are exemplified, during the implementation of a forecast on the academic results of a school. Finally, other applications are valued in educational context scenarios, where it is necessary to make estimates and forecasts, based on expert knowledge.

Keywords: Cloud Delphi method; Cloud model; educational research; decision making; R

1. INTRODUCCIÓN

El método Delphi tiene su origen en estudios clasificados por la corporación Rand de los Estados Unidos, durante los años 50 del siglo pasado. Los primeros estadios de desarrollo definieron sus características fundamentales y mostraron una elevada utilidad en el ámbito del pronóstico. Por tal motivo, comúnmente se asocia este método con el campo de la prospectiva (Cruz y Rúa, 2018). Su base epistémica fundamental reside en el supuesto de que el pensamiento colectivo posee mayor valor y objetividad que el individual, fundamentalmente en el abordaje de problemas nuevos y situaciones inéditas, donde no existen datos previos ni series cronológicas, así que recursos tales como los modelos estadísticos no resultan satisfactorios (Kaplan, Skogstad, & Girshick, 1950).

Son cuatro los rasgos definitorios fundamentales que demarcan un estudio Delphi: el anonimato de un conjunto de expertos, la iteración en el procesamiento de datos (apoyado fundamentalmente en encuestas estructuradas), la retroalimentación controlada (*feedback*) y la respuesta estadística de grupo (Cruz, 2009; Landeta, 1999). Durante el desarrollo de un estudio Delphi, dos conjuntos de participantes asumen roles específicos: el grupo de expertos que emiten sus criterios acerca de una problemática bien definida, y el grupo coordinador, encargado de sintetizar las respuestas y devolver el parecer colectivo a cada uno de los expertos en rondas subsiguientes.

Con el paso del tiempo, el método Delphi experimentó numerosas adaptaciones y surgieron decenas de variantes tales como el mini-Delphi, el e-Delphi, el Delphi Fuzzy (Delphi Borroso), y el Cloud Delphi (Delphi de Nube). Cada modificación, o bien aprovechó las nuevas posibilidades que fueron surgiendo con las tecnologías de la información y las comunicaciones, o bien intentó corregir deficiencias identificadas en investigaciones científicas (Hasson y Keeney, 2011). Los dos últimos ejemplos responden a problemas profundos, relacionados con la capacidad del método para captar con suficiente objetividad la evaluación, estimación, o la percepción colectiva acerca de problemas poco precisos.

En Cuba, la primera aplicación del método Delphi, de la cual se ha tenido referencia documentada, corresponde a un estudio prospectivo, desarrollado por especialistas del Instituto Nacional de Investigaciones Económicas (INIE). El estudio llevó como título *Utilización del método Delphi en la pronosticación: Una experiencia inicial*, y estaba relacionado con la generalización de una política

nacional de informática (Oñate, Ramos, y Díaz, 1988). Se trataba de una construcción colectiva de futuro, la cual constituyó el preámbulo de un proceso gradual y acelerado de implementación de la informática en todo el país (Álvarez, Duarte, y Madruga, 2016). Por ejemplo, como parte del perfeccionamiento del sistema educacional, pocos años después surgió la carrera de Licenciatura en Educación, con especialidad en Matemática y Computación (Plan de Estudios C, implementado a partir de 1991).

Entre aciertos y desaciertos, el método Delphi ha encontrado numerosas aplicaciones en el campo de las investigaciones educacionales (Avella, 2016; López-Gómez, 2018). Al nivel nacional, la mayor difusión del método Delphi en dicho ámbito, tuvo su origen en trabajos pioneros desarrollados en el Instituto Central de Ciencias Pedagógicas. Las necesidades prácticas en la evaluación de resultados científicos, relacionados con problemáticas educacionales, conllevó a un uso generalizado de este método, como caso particular del método de criterio de expertos. Por ejemplo, en un estudio reciente donde se procesó una amplia muestra de más de mil tesis doctorales defendidas en Cuba, de los campos de las Ciencias Pedagógicas y de las Ciencias de la Educación, se observó la utilización del método Delphi en relación directa con los siguientes métodos empíricos (Díaz, Cruz, Pérez, y Ortiz, 2020):

- Criterio de expertos,
- Consulta a expertos,
- Criterio de especialistas,
- Criterio de jueces,
- Criterio de usuarios,
- Juicio de expertos,
- Valoración de expertos, y
- Validación por expertos.

La panorámica actual expresa una tendencia marcada hacia el uso del Delphi en la evaluación y valoración de resultados científicos, mientras que el Delphi de pronóstico prácticamente carece de visibilidad. La también llamada “forma clásica” o “convencional”, encontró aplicaciones muy tempranamente, cuando Helmer (1966), señaló algunas posibilidades de aplicación en la planeación y en la innovación educacional. El pronóstico constituye un aspecto importante que requiere mayor atención desde las investigaciones educacionales, de modo que no deberían soslayarse sus

potencialidades para el diseño investigativo y la toma de decisiones.

El presente trabajo se centra en el Delphi de Nube, donde se imbrica, de manera sistémica, los mecanismos borrosos del Delphi Fuzzy y la incertidumbre/aleatoriedad presente en los procesos estocásticos (Yang, Zeng, y Zhang, 2012). Ante la necesidad de automatizar este tipo de Delphi, se adopta el objetivo de diseñar una función computacional, útil, relativamente sencilla y de uso práctico, para las investigaciones educacionales.

A continuación, después de describir las principales características del Delphi de Nube, se presenta la nueva función denominada *CloudDelphi* y se ejemplifica su utilidad y flexibilidad. Finalmente, se valoran otras aplicaciones en escenarios del contexto educacional, donde es necesario realizar evaluaciones, trazar políticas, efectuar estimaciones y pronósticos, con base en el conocimiento experto.

1.1. Enfoques no deterministas del método Delphi

La revisión bibliográfica sobre el desarrollo actual del método Delphi, muestra una tendencia hacia la búsqueda de alternativas que expresen de manera adecuada la imprecisión en las respuestas de los expertos. Esencialmente, la principal dificultad del Delphi convencional consiste en sus escasas posibilidades para captar la ambigüedad y la incertidumbre, por un camino indeterminista. Como norma, a los expertos no se les solicita indicar un valor numérico concreto, sino seleccionar etiquetas lingüísticas proveídas en una investigación. Estas últimas no tienen por qué percibirse de modo simétrico y/o equidistante por los expertos, incluso podrían tener diferentes significados semánticos lo cual puede disminuir la fiabilidad y validez de un estudio.

Para una implementación más efectiva del método Delphi, algunas investigaciones contemporáneas del campo de la educación han seleccionado el enfoque difuso (Chang, Liang, Shu, y Tsai, 2015; Hsu, Luo, y Ho, 2017; Tu, Tai, y Chen, 2016). A partir de la necesidad de perfeccionar el enfoque difuso clásico, también han surgido alternativas que tratan de paliar la falencia de lo borroso como aleatorio, el determinismo imperante en la selección de funciones de pertenencia, o bien la necesidad de flexibilizar el supuesto lógico del tercero excluido. Con ello, existen otros caminos viables, tales como el enfoque difuso intuicionista, el de números difusos pitagóricos, los conjuntos difusos de tipo-2, y los conjuntos difusos no estacionarios. El Delphi basado en dichas teorías constituye un campo fértil para nuevas ampliaciones teóricas (cf. Lixëndroui y

Lixëndroui, 2013; Roy y Garai, 2012).

El enfoque difuso no es ajeno a problemáticas epistémicas, como el contraste en las propias respuestas del panel de experto que suelen ser de naturaleza *crisp* (rígida), o bien la tensión e incertidumbre asociadas al dilema de localizar el valor más probable dentro de un intervalo, lo cual puede estar mediado por sucesos estocásticos. Como una versión alternativa, donde se trata de combinar lo borroso con la incertidumbre/aleatoriedad, se desarrolló el Modelo de Nube (Li, Liu, y Gan, 2009).

El citado modelo se precisa rigurosamente. En efecto, dado un concepto cualitativo T , definido sobre un universo de discurso $U = \{u\}$, sea $x \in U$ una instancia aleatoria del concepto T y $\mu_T(x) \in [0,1]$ el grado de certidumbre de x a lo largo de T , el cual corresponde a un número aleatorio con una tendencia estable. Entonces, la distribución de x en el universo U puede definirse como una nube y x constituye “una gota” de la nube. Se cumple que para todo $x \in U$, la aplicación $\mu_T(x)$ es no inyectiva, por su propia naturaleza. O sea, el grado de certeza de x a lo largo del concepto T es una distribución de probabilidad, en lugar de constituir un valor fijo (Li, Liu, y Gan, 2009: 360).

El Modelo de Nube se apoya en tres conceptos fundamentales: la expectativa (Ex), consistente en la esperanza matemática de que las gotas de la nube pertenezcan a un concepto del universo; la entropía (En), que expresa la medida borrosa de un concepto cualitativo, como ámbito de cobertura del universo que puede ser admitido por el concepto; y la hiperentropía (He), que es el grado de incertidumbre de la entropía, y refleja la dispersión de las gotas de la nube. Por tanto, la amplitud de la nube explica el grado de borrosidad, mientras que el espesor de la nube explica el grado de aleatoriedad (Xu y Xu, 2018: 3).

Algunos estudios han observado que dicho modelo no supera las extensiones difusas. Por ejemplo, Huang y Yang (2016) observaron que el Modelo de Nube, el de conjuntos difusos de tipo-2, y el de conjuntos difusos no estacionarios, se diferencian por sus motivaciones, fundamentos, expresión, metodología, representación de la incertidumbre, y resultados. Al respecto, estos autores concluyen que “[...] dichos modelos son complementarios para describir la incertidumbre del mundo objetivo, desde puntos de vista diferentes” (Huang y Yang, 2016: 225).

Si $x \in U$ es la representación aleatoria de un concepto cualitativo \tilde{A} en el universo de discurso U , la cual

satisface que $x \sim N(Ex, (En')^2)$, donde $En' \sim N(En, He^2)$, y $\mu_A(x) = \exp\{-(x-Ex)^2/[2(En')^2]\}$, entonces la distribución de x en U se denomina "nube normal". Así, con base en dicho modelo, Yang, Zeng y Zhang (2012) proponen el Delphi de Nube. Para ello se apoyan en las encuestas de valor de intervalo (*interval-valued surveys*). En efecto, la estimación de un valor, por medio de un intervalo, satisface lo siguiente:

- El valor medio del intervalo puede interpretarse como la expectativa Ex .
- La amplitud del intervalo puede reflejar una medida de la entropía En .
- La variación entre intervalos producidos por dos iteraciones sucesivas, puede reflejar el grado de incertidumbre medido por la hiper-entropía He .

Desde su etapa inicial, el Modelo de Nube encontró un terreno fértil en el campo de la inteligencia artificial, con códigos desarrollados principalmente en VC++ y MATLAB, incluso para generalizaciones bidimensionales (Li y Yi, 2008: 107-151). Para el caso del Delphi de Nube, es necesario emplear un algoritmo sintético, el cual produce cierta especie de nube explicativa del comportamiento global $\tilde{A}_s(Ex_s, En_s, He_s)$, a partir de n nubes $\tilde{A}_i(Ex_i, En_i, He_i)$, $1 \leq i \leq n$, correspondientes a cada uno de los expertos (Yang, Zeng, y Zhang, 2012).

De forma resumida, el algoritmo propuesto por el académico Li Deyi (Li, Liu, y Gan, 2009: 363; Li y Yi, 2008: 119), para la obtención de una nube normal compuesta por N gotas, es el siguiente:

Algoritmo 1. *Normal Cloud Generator NCG*(Ex, En, He, N).

Input: Tres parámetros Ex, En, He , y el número de gotas N de la nube.

Output: N gotas de la nube con su grado de incertidumbre.

Pasos:

1. Se genera un número aleatorio En'_i normalmente distribuido, cuya esperanza es En y su varianza He^2 .
 2. Se genera un número aleatorio x_i normalmente distribuido, cuya esperanza es Ex y su varianza $En_i'^2$.
 3. Se calcula el grado de incertidumbre $y_i = \mu_i(x) = \exp\{-(x_i - Ex)^2 / [2(En'_i)^2]\}$.
 4. Se obtiene la i -ésima gota $drop(x_i, y_i)$ de la nube.
 5. Se repite el ciclo 1-4, hasta generar las N gotas.
-

Los aportes de cada experto no necesariamente son

equiparables, de manera que puede ser necesaria la introducción de un vector normalizado, que exprese los pesos individuales. Para la estimación del peso, es posible tomar en consideración ciertos indicadores relacionados con las respuestas emitidas por los expertos, como la relación entre consistencia y capacidad de discriminación ante situaciones diferentes (Weiss y Shanteau, 2003), o bien consistencia y estabilidad en el juicio (Yang, Zeng, y Zhang, 2012). El presente trabajo adopta esta última variante, pero no desecha la posibilidad de implementar la primera.

Utilizando expresiones desarrolladas para la suma y producto de nubes (Li y Yi, 2008) es posible controlar, en cada ronda, el comportamiento de las respuestas emitidas por los expertos. Por este camino, Yang, Zeng, y Zhang, (2012) introducen un algoritmo para el cálculo de la nube promedio ponderada \tilde{A}_{wa} , basada en un vector peso normalizado. Cada medición individual se determina de modo inversamente proporcional a una suma de tres componentes: el valor modular normalizado de la diferencia entre las expectativas individual y colectiva, la borrosidad del juicio, y la incertidumbre. En el presente trabajo se adopta esta medida, pero no se descarta la posibilidad de desarrollar otros índices similares.

1.2. Estructura del Delphi de Nube

Siguiendo la descripción de sus autores, el Delphi de Nube consta de cinco etapas (Yang, Zeng, y Zhang, 2012), que pueden estructurarse a modo de algoritmo. Ello se describe seguidamente:

Algoritmo 2. *Normal Cloud Delphi NCD*(Ex, En, He, N).

Input: El número N gotas de la nube, así como i segmentos $[I_i^{(k)}, u_i^{(k)}] \subset \mathfrak{R}$, que constituyen las evaluaciones del panel de i expertos ($1 \leq i \leq n$) en la ronda k , de cierto aspecto consignado en una encuesta.

Output: La nube sintética $\tilde{A}_{wa}^{(k)}$ basada en la ponderación w_i de cada experto, y sendos índices de estabilidad de cada experto i después de la ronda k : $\Delta En_i^{(k)}$ y $Unc_i^{(k)}$.

Pasos:

1. Se inicia el contador de iteración ($k = 1$).
 2. Se obtiene una nube $NCG_i^{(k)}(Ex_i^{(k)}, En_i^{(k)}, He_i^{(k)}, N)$ para la respuesta k -ésima del experto i -ésimo, donde $Ex_i^{(k)} = (u_i^{(k)} + I_i^{(k)})/2$, $En_i^{(k)} = (u_i^{(k)} - I_i^{(k)})/6$. En el caso de $He_i^{(k)}$, si $k = 1$ entonces $He_i^{(1)} = En_i^{(1)}/K$ (donde K es una constante predefinida), y si $k > 1$, entonces $He_i^{(k)} = [\max\{u_i^{(k)} - u_i^{(k-1)}, 0\} + \max\{I_i^{(k-1)} - I_i^{(k)}, 0\}]/6$.
-

3. Se calcula el vector peso, mediante los componentes $w_i^{(k)} = [|Ex_i^{(k)} - Ex_s^{(k)}| / Ex_s^{(k)} + En_i^{(k)} + He_i^{(k)}]^{-1}$, los cuales se normalizan por intermedio de $w_i^{(k)} = w_i^{(k)} / \sum_{1 \leq i \leq n} w_i^{(k)}$.
4. Se genera información retroalimentada (*feedback*) por medio de dos procedimientos de agregación: uno para determinar la nube sintética $\tilde{A}_s^{(k)} = NCG_s^{(k)}(Ex_s^{(k)}, En_s^{(k)}, He_s^{(k)})$, y otro para expresar cada nube promedio ponderada $\tilde{A}_{wa}^{(k)} = NCG_{wa}^{(k)}(Ex_{wa}^{(k)}, En_{wa}^{(k)}, He_{wa}^{(k)})$, donde $1 \leq i \leq n$.
5. Se calculan sendos índices para controlar la estabilidad del juicio experto: uno consiste en la diferencia normalizada entre entropías consecutivas (ΔEn), y el otro es el grado de incertidumbre normalizado (Unc). Se obtienen como sigue: $\Delta En_i^{(k)} = |En_i^{(k-1)} - En_i^{(k)}| / En_i^{(k-1)}$, con $\Delta En_i^{(1)} = En_i^{(1)}$, y $Unc_i^{(k)} = He_i^{(k)} / En_i^{(k)}$, $1 \leq i \leq n$.
6. Se establece la condición de parada $\Delta En_i^{(k)} \leq \delta$ (prefijado de antemano) y $Unc_i^{(k)} = 0$. En caso de que un experto no cumpla dichas condiciones, el experimentador adopta decisiones puntuales.
7. $k \leftarrow k+1$, dando paso a la siguiente ronda Delphi.

En el segundo paso, el valor de $Ex_i^{(k)}$ como semisuma es obvio, para efectuar el cálculo de cada expectativa. La ecuación de $En_i^{(k)}$ expresa la sexta parte de la longitud de cada intervalo, inspirada en la conocida regla “ 3σ ” (bajo una distribución normal, si σ expresa la desviación típica y μ la media, entonces el intervalo $[\mu - 3\sigma, \mu + 3\sigma] \subset \mathfrak{R}$ comprende alrededor del 99.74% de la distribución). Por su parte, la ecuación de $He_i^{(k)}$ intenta captar en qué medida el juicio del experto i permanece estable tras dos iteraciones sucesivas $k-1$ y k . Si la respuesta k -ésima produce un intervalo que es subconjunto del $(k-1)$ -ésimo, resulta que $He_i^{(k)} = 0$ para el experto i . En todos los restantes casos se obtiene un valor mayor que cero. Además, a mayor incertidumbre/aleatoriedad, mayor será la hiper-entropía.

Respecto a la condición de parada, definida en el sexto paso, ΔEn refleja el grado de estabilidad borrosa del juicio experto, tras dos iteraciones sucesivas, mientras que Unc expresa el grado de estabilidad dentro de una gama de incertidumbre. Los autores citados recomiendan que de existir un experto i , el cual no satisface dichas reglas, entonces la información $\tilde{A}_s^{(k)}$ y $\tilde{A}_{wa}^{(k)}$, generadas en el cuarto paso, junto a su opinión individual $\tilde{A}_i^{(k)}$, se representan gráficamente de forma conjunta y se envían a dicho experto para un re-análisis. Otra vía consiste en prefijar un valor k mínimo para adoptar una decisión, de manera que a partir de la ronda k puede detenerse el proceso, desechando las opiniones de los expertos que se consideren difusas

e inestables.

2. MATERIALES Y MÉTODOS

El presente estudio utiliza el entorno y lenguaje de programación estadística R (The R Foundation for Statistical Computing, 2019), el cual se distribuye bajo la licencia GNU GPL y resulta disponible para los sistemas operativos Windows, Macintosh, Unix y GNU/Linux. R está orientado a objeto y tiene la ventaja de poder integrarse con numerosas bases de datos y bibliotecas. En su mejora continua participa una amplia comunidad de desarrolladores, a escala internacional. Numerosos sistemas comerciales permiten un intercambio compatible de códigos con R, por ejemplo, MATLAB, SPSS, Mathematica, entre otros. Además, existen versiones de R disponibles para los sistemas operativos Windows, Macintosh, Unix y GNU/Linux.

Junto al soporte R, en el presente trabajo también se implementa el método analítico-sintético, con el objetivo de simplificar y humanizar el trabajo científico. Con ello se busca proveer una alternativa coherente y de utilidad práctica, de modo que pueda ser implementada eficazmente en el ámbito de las investigaciones educacionales.

3. RESULTADOS Y DISCUSIÓN

La función *CloudDelphi* para la implementación del Delphi de Nube, aparece descrita en un anexo al final del presente trabajo. En realidad, la naturaleza del Algoritmo 2 (*NCD*) impone características de pseudo-algoritmo, como resultado de la imprecisión que caracterizan la toma de decisiones en el sexto paso. Por este motivo, la programación sobre R de la función *CloudDelphi* culmina en el quinto paso, dejando los pasos subsiguientes al trabajo particularizado de cada investigador. Como elemento adicional, este código solo utiliza las bibliotecas predeterminadas de R, y no requiere la instalación de paquetes adicionales para mayor simplicidad.

Las dos primeras entradas corresponden a cadenas, contentivas de una secuencia de parejas de números reales, los cuales constituyen los valores mínimo y máximo de cada segmento, respectivamente. La primera entrada *CurrentIntervals* corresponde a los intervalos de la ronda k , mientras que *ForegoingIntervals* refiere los intervalos de la ronda $k-1$. En la tercera variable *ForegoingEntropy* se introducen los valores de entropía, calculados en la ronda precedente. Salvo la primera variable, las dos siguientes pueden ser vacías.

Las 16 variables subsiguientes constituyen opciones que definen aspectos tales como: la constante K , correspondiente al segundo paso donde $K=6$ suele

adoptarse como valor predeterminado; las cantidades N de gotas de cada nube individual o de la nube sintética; los extremos del intervalo de visualización; junto a otras posibilidades gráficas. Por ejemplo, es posible variar el color y la forma de representación de las gotas, así como incluir o no cuatro curvas importantes: *Upper*, *Lower*, *Expectation*, y *Uncertainty*. Las dos primeras representan cotas de la nube, con base en la regla “ 3σ ”. La tercera expresa el valor esperado y_i para cada x_i .

La cuarta curva *Uncertainty* permite visualizar una medida de la incertidumbre. Precisamente, una de las características del modelo de nube implica que la mayor incertidumbre no se localiza en los extremos de un intervalo de confianza, ni tampoco en su punto medio, sino en zonas intermedias entre los extremos y el valor central. En general, cuando un experto señala un segmento para indicar una región donde se localiza su estimación, cognitivamente ello implica que su mayor seguridad apunta hacia los extremos y hacia el valor central. La incertidumbre se incrementa al alejarse de estos valores (Yang, Zeng, y Zhang, 2012). Por ejemplo, estimar una evaluación en el intervalo $[20, 30]$ sugiere tres números con mayor nivel de seguridad: los valores limítrofes 20 y 30, junto al valor medio $(20 + 30)/2 = 25$.

Como parte de las prestaciones de R, los gráficos producidos por CloudDelphi pueden exportarse en varios formatos (Metafile, Postscript, PDF, PNG, BMP, TIFF, JPEG), mayormente compatibles con requerimientos definidos por numerosas revistas y actas de congresos. Tomando en consideración el posible empleo de estos gráficos en presentaciones electrónicas, así como la presencia frecuente de varias nubes, CloudDelphi construye los gráficos con una relación de aspecto de tipo “panorámico”, o sea, con razón 16:9. No obstante, este formato puede ser modificado por el usuario modificando la línea de comando `windows(width = 16, height = 9)`, o bien transformando manualmente el dispositivo gráfico de R.

3.1. Un ejemplo de aplicación

En su investigación doctoral, Rodríguez (2015) establece estándares para la evaluación de la calidad en la educación superior. En particular, implementa el Delphi de Nube para el análisis de la escala de evaluación del programa educativo “Ciclo Inicial Optativo Ciencia y Tecnología”. La autora se apoya en el enfoque germinal de Yang, Zeng, y Zhang (2012), sobre el cual destaca tres ventajas: el requerimiento de un panel de expertos relativamente pequeño, la inmediatez en la retroalimentación y

obtención de resultados finales, y la ponderación del juicio de expertos sobre la base de criterios diversos (Rodríguez, 2015: 166). Uno de los estándares evaluado por cuatro expertos consistió en los *Niveles parciales de logro*, como “[...] expresión del grado de cumplimiento de objetivos, cantidad de ingresos y egresos por generación, cantidad de créditos ofrecidos” (Rodríguez, 2015: 168).

Como resultado de aplicar el Delphi de Nube, tomando $x =$ Niveles parciales de logro y $\mu =$ grado de certeza, la Tabla 1 resume los intervalos obtenidos después de tres iteraciones.

Tabla 1. Evaluaciones emitidas por los expertos

Evaluable	Primera iteración	Segunda iteración	Tercera iteración
1	[55, 60]	[50, 55]	[50, 55]
2	[40, 45]	[30, 35]	[30, 35]
3	[25, 30]	[30, 35]	[30, 35]
4	[20, 25]	[25, 30]	[25, 30]

Fuente: resumido de Rodríguez (2015: 302-304).

Obviamente, no es necesario continuar el proceso más allá de la tercera iteración, ya que los expertos emiten las mismas respuestas testeadas en la segunda iteración, con lo cual se evidencia estabilidad plena en el juicio experto. Siguiendo una recomendación de la autora, donde se sugiere desarrollar un software que implemente el Delphi de Nube (Rodríguez, 2015: 250), a continuación, se ilustra el resultado de procesar las dos primeras iteraciones, por intermedio de la función *CloudDelphi*. En efecto, para la primera iteración es suficiente introducir las siguientes líneas de comando, después de correr dicha función en la consola de R:

```
> a = c(55,60, 40,45, 25,30, 20,25)
> b = c()
> c = c()
> CloudDelphi(a, b, c, 20, 60)
```

Nótese que la primera cadena de caracteres corresponde a la variable *CurrentIntervals*, y representa la secuencia de intervalos de la primera iteración. Las dos cadenas subsiguientes son vacías, pues las variables *ForegoingIntervals* y *ForegoingEntropy* no tienen valores asignados. Las opciones *Min = 20*, *Max = 60*, modifican los valores predeterminados de la función *CloudDelphi*, con el objetivo de visualizar un intervalo similar al de la obra de referencia.

Los resultados exportados sobre la consola de R se

resumen en la Tabla 2, y coinciden con los valores calculados en la fuente de referencia.

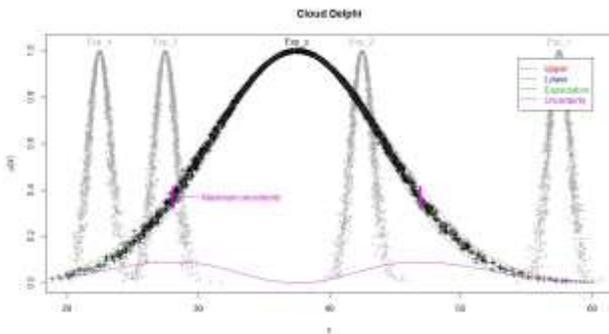
Tabla 2. Resultados de la primera iteración

Var.	Exp_1	Exp_2	Exp_3	Exp_4	Exp_s
Ex	57.50	42.50	27.50	22.50	37.50
En	.83	.83	.83	.83	6.67
He	.14	.14	.14	.14	.14
w	.21	.29	.26	.23	1.00
ΔEn	.83	.83	.83	.83	6.67
Unc	.17	.17	.17	.17	1.17

Fuente: elaboración propia con *CloudDelphi*.

Seguidamente, la Figura 1 facilita la visualización y comprensión sintética de los datos. Puede observarse que las opiniones de los cuatro expertos son relativamente discordantes, pero de escasa entropía. Ello se refleja en la nube promedio sintética Exp_s, la cual presenta escasa incertidumbre, pero es marcadamente difusa por su amplia elongación.

Figura 1. Visualización de la primera iteración



Fuente: elaboración propia con *CloudDelphi*.

Tras la segunda iteración ya es posible introducir parejas de intervalos, de modo que el cálculo de la hiper-entropía es más objetivo. Como fue descrito en el Algoritmo 2, este cálculo permite adoptar la entropía precedente como hiper-entropía para la variable *ForegoingEntropy* en la función. Así se muestra en los siguientes valores de entrada:

- > a = c(50,55, 30,35, 30,35, 25,30)
- > b = c(55,60, 40,45, 25,30, 20,25)
- > c = c(.83, .83, .83, .83, 6.67) # He en la Tabla 2
- > CloudDelphi(a, b, c, 20, 60)

Los nuevos resultados se condensan en la Tabla 3. Puede observarse que el experto No. 3 posee mayor peso relativo ($w = .29$). Junto a la nube producida por el segundo experto, Exp_3 se localiza más cercano a Exp_s y con igual entropía, pero con menor hiper-

entropía. Esto justifica su mayor peso.

Tabla 3. Resultados de la segunda iteración

Var.	Exp_1	Exp_2	Exp_3	Exp_4	Exp_s
Ex	52.50	32.50	32.50	27.50	36.25
En	.83	.83	.83	.83	5.00
He	.83	1.67	.83	.83	1.04
w	.24	.20	.29	.27	1.00
ΔEn	.00	.00	.00	.00	.25
Unc	1.00	2.00	1.00	1.00	2.00

Fuente: elaboración propia con *CloudDelphi*.

En esta iteración, la consola de R advierte sobre la existencia de al menos 50 problemas, todos similares. En efecto:

```
> warnings()
```

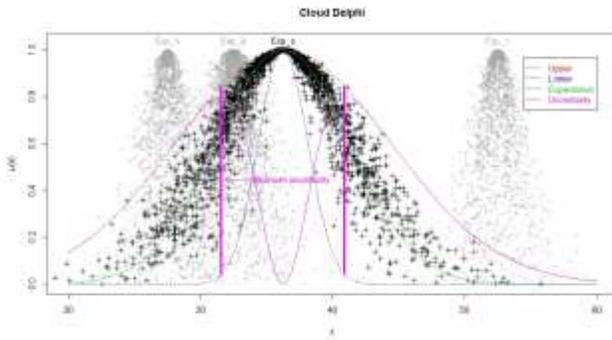
Warning messages:

```
1: In rnorm(1, mean = Ex, sd = Em_i) : NAs produced
```

Esto se debe a que los valores Em_i son resultado de la implementación previa de la función "rnorm", la cual genera un valor aleatorio normalmente distribuido, con media En y desviación típica He . El problema surge cuando este valor aleatorio es negativo, de modo que no expresa una desviación típica en el comando subsiguiente. Para valores grandes de N , dicho problema no es trascendente. Sin embargo, para valores pequeños de N , el usuario puede modificar el código utilizando el valor absoluto, o sea: $rnorm(1, mean = Ex, sd = abs(Em_i))$. Sin embargo, siguiendo la descripción original de Li, Liu, y Gan (2009) aquí se implementa este comando, aceptando la posibilidad de que aparezcan valores perdidos (Na) eventualmente para el valor predeterminado $N = 2000$.

La Figura 2 refleja los resultados de forma gráfica. Puede observarse que las nubes producidas por el segundo y tercer experto, prácticamente se confunden. El primer experto ofrece una opinión notablemente alejada del resto. Por este motivo, la nube promedio ponderada muestra valores difusos de la variable evaluada, extendidos mayormente por $Ex \pm 3En = 36.25 \pm 15$ ($En = 5.00$). Además de un comportamiento difuso elevado, también se observa una notable incertidumbre ($Unc = 2.00$), la cual se agudiza en valores cercanos a $x = 32$ y 41 . El valor esperado es precisamente el resultado de la variable *Expectation*, o sea: $Ex = 36.25$. Como se señaló previamente, ya que en la tercera iteración las respuestas se mantienen inamovibles, el proceso culmina exactamente aquí.

Figura 2. Visualización de la segunda iteración



Fuente: elaboración propia con *CloudDelphi*.

En todos los casos, los resultados coinciden con el estudio original. Las principales ventajas son tres: la posibilidad de englobar en un solo gráfico toda la información proveniente de cada ronda; la implementación sobre un paquete de código abierto, relacionado directamente con métodos estadísticos; y la flexibilidad de la función *CloudDelphi* para modificar elementos gráficos y de procesamiento de datos, conforme a los requerimientos de cada investigador o de normativas específicas de una revista, acta de congreso, o presentación electrónica.

3.2. Aplicaciones al pronóstico educacional

La utilidad del método Delphi ha sido constatada en investigaciones teóricas del campo de la educación (López-Gómez, 2018), lo cual se corrobora por estudios cuantitativos (Cruz y Rúa, 2018; Flostrand, Pitt, y Bridson, 2020). Sin embargo, la mayor difusión de sus aplicaciones constituye un fenómeno relativamente reciente y confinado a problemas muy puntuales. Desde una perspectiva epistemológica, Avella señala que el Delphi es “[...] apropiado excepcionalmente para un número de diligencias de investigación que implican una perspectiva post-positivista/constructivista y ofrece ventajas potenciales no disponibles en muchos de los diseños más cualitativos tradicionales, como el estudio de casos, la etnografía, o la teoría fundamentada” (Avella, 2016: 318).

En el ámbito del pronóstico, también son inagotables los problemas inherentes a las ciencias de la educación. Generalmente, cuando el pronóstico apunta hacia una variable unidimensional, las oportunidades de estudio se simplifican sustancialmente. Por el contrario, las miradas prospectivas de variables multidimensionales son mucho más complejas. El Delphi de Nube es aplicable en ambos contextos, siempre que las variables estén definidas con precisión. Para ello, existe una generalización natural inspirada en la

distribución normal multidimensional. Elementos teóricos definitorios del Delphi de Nube bidimensional, ya aparecen desarrollados por Li y Yi (2008: 133-138), con posibilidades directas de extensión a estudios Delphi.

A continuación, se presentan algunos problemas concretos, donde es posible implementar el Delphi de Nube, en espacios afines a las investigaciones educacionales. La selección mostrada tiene fines ilustrativos, no persigue exhaustividad ni el establecimiento de tipologías.

- Estudios prospectivos del sector de la educación, complementados por la vigilancia tecnológica (Escorsa et al., 2017; Oñate, Ramos, y Díaz, 1988).
- Identificación de áreas priorizadas y trazado de políticas educacionales (Kelleher et al., 2019; Maxey y Kezar, 2015).
- Gestión continua de la calidad en instituciones educativas, en carreras universitarias, y en programas académicos de posgrado, por intermedio de la identificación de indicadores y el establecimiento de normativas (Hernández y Fernández, 2018; Rodríguez, 2015 y 2019).
- Modelación estructural, basada en un conjunto de aspectos complejos que expresen relaciones consistentes y consensuadas (Linstone y Turoff, 2011).
- Estudios prospectivos sobre economía de la educación, relacionados con la gestión de los recursos humanos, la gestión de proyectos de investigación, la renovación tecnológica, entre otros aspectos (Cruz y Díaz, 2020).
- Establecimiento de puntos de corte en pruebas cognitivas, donde se requiere la fijación consensuada de aquellos valores que definen saltos cualitativos (Luzardo y Rodríguez, 2019).
- Identificación de competencias para los profesionales de la educación (Sever y Bostanci, 2020).

4. CONCLUSIONES

El resultado fundamental del presente trabajo consiste en la función *CloudDelphi*, la cual responde a la parte más estable del algoritmo típico de un Delphi de Nube. Este recurso ha sido desarrollado sobre el lenguaje R, al cual le asisten numerosas ventajas, como el hecho de contar con un elevado nivel de flexibilidad, poseer código abierto, junto a numerosas potencialidades para la investigación, por intermedio de recursos estadísticos.

La función *CloudDelphi* ha sido concebida para el contexto de las investigaciones educacionales, con tres elementos técnicos que facilitan su uso práctico: la diversidad de opciones que permiten modificar sus resultados, las facilidades gráficas adecuadas a normativas frecuentes en publicaciones científicas, y la presentación de valores tabulares que sintetizan las variables propias de un Delphi de Nube.

Numerosas son las oportunidades de mejora para la función *CloudDelphi*, relativas a la implementación de las dos últimas etapas en el Algoritmo 2. Especialmente, en lo concerniente al control automatizado de la estabilidad en el juicio experto. Sin embargo, existen aspectos que trascienden el marco de la programación de algoritmos, los cuales invitan a reflexionar más allá del marco del diseño y las prestaciones, e incluso del análisis de la eficiencia y complejidad computacional. Se trata de otros elementos, relacionados con la propia concepción del método Delphi de Nube.

Entre los elementos a profundizar en estudios futuros, se incluye el hecho de que la celebrada estabilidad en el juicio experto, puede ser el efecto negativo de una conducta inflexible y pertinaz. La expresión empleada para medir el peso de cada experto constituye una propuesta efectiva de sus autores, pero no agota las amplias posibilidades para efectuar dicha medida a partir de otros tipos de indicadores. En todo caso, es útil evaluar la viabilidad de implementar operadores OWA (*Ordered Weighted Averaging*), los cuales han resultado útiles en otros estudios, donde es necesario ponderar indicadores o bien fuentes de opinión experta (Yager, Kacprzyk, y Beliakov, 2011). Asimismo, también es promisorio desarrollar estudios basados en nubes normales multidimensionales, o bien comparativos entre los Delphi de Nube y otros enfoques no deterministas mencionados anteriormente. Todo ello provee innumerables problemas abiertos, los cuales constituyen temas pendientes para investigaciones ulteriores

5. REFERENCIAS BIBLIOGRÁFICAS

- Álvarez, P., Duarte, R. R., y Madruga, A. (2016). Cuba. En: J. L. Cordeiro (Ed.), *La Prospectiva en Iberoamérica: Pasado, Presente y Futuro* (pp. 135-155). Universidad del Valle: Instituto de Prospectiva, Información y Gestión del Conocimiento. Recuperado de: www4.pucsp.br/catedraignacysachs/downloads/libro-riber2016.pdf
- Avella, J. R. (2016). Delphi panels: research design, procedures, advantages, and challenges. *International Journal of Doctoral Studies*, 11, 305-321. Recuperado de: <http://www.informingscience.org/Publications/3561>
- Cruz, M. (2009). *El Método Delphi en las Investigaciones Educacionales*. La Habana: Academia.
- Cruz, M., y Rúa, J. A. (2018). Surgimiento y desarrollo del método Delphi: una perspectiva cuantitativa. *Biblios*, 71, 90-107, doi: [10.5195/biblios.2018.470](https://doi.org/10.5195/biblios.2018.470)
- Cruz, M., y Díaz, Y. (2020). Un Delphi automatizado para la búsqueda de consenso en la gestión de procesos universitarios. *EduSol*, 20(71), 67-81, Recuperado de: <http://edusol.cug.co.cu/index.php/EduSol/article/view/1171>
- Chang, C.-C., Liang, C., Shu, K.-M., and Tsai, C.-W. (2015). Key successful factors of knowledge management for university students using e-portfolios: Approach of Fuzzy Delphi and Fuzzy AHP. *Computer Applications in Engineering Education*, 23(5), 673-681. doi: [10.1002/cae.21640](https://doi.org/10.1002/cae.21640)
- Díaz, Y., Cruz, M., Pérez, M., C., y Ortiz, T. (2020). El método criterio de expertos en las investigaciones educacionales: visión desde una muestra de tesis doctorales. *Revista Cubana de Educación Superior*, 39(1). Recuperado de: <http://scielo.sld.cu/pdf/rces/v39n1/0257-4314-rces-39-01-e18.pdf>
- Escorsa, P., Escorsa, E., Montes, J. M., Ayarza, G., Arango, B., Chaur, J., Hernández, R., y Moreno, J. F. (2017). Ejercicio prospectivo del sector de la educación para Panamá al año 2040. *Espacios*, 38(43) Recuperado de: <http://www.revistaespacios.com/a17v38n43/a17v38n43p05.pdf>
- Flostrand, A., Pitt, L., y Bridson, S. (2020). The Delphi technique in forecasting – A 42-year bibliographic analysis (1975-2017). *Technological Forecasting & Social Change*, 150, 119773. doi: [10.1016/j.techfore.2019.119773](https://doi.org/10.1016/j.techfore.2019.119773)
- Hasson, F., y Keeney, S. (2011). Enhancing rigour in the Delphi technique research. *Technological Forecasting & Social Change*, 78(9), 1695-1704. doi: [10.1016/j.techfore.2011.04.005](https://doi.org/10.1016/j.techfore.2011.04.005)
- Helmer, O. (1966). *The Use of the Delphi Technique in Problems of Educational Innovations*. Rand Corporation Paper P-3499. Recuperado de:

- <http://www.rand.org/pubs/papers/2006/P3499.pdf>
- Hernández, G., y Fernández, J. (2018). La planificación estratégica e indicadores de calidad educativa. *Revista Nacional de Administración*, 9(1), 69-86. Recuperado de: <https://investiga.uned.ac.cr/revistas/index.php/rna/article/download/2103/2467>
- Hsu, W.-L., Luo H.-S., y Ho, S.-Y. (2017). Applying the fuzzy Delphi method to preservice training courses for tour guides in Taiwan. En: T.-H. Meen, S. D. Prior, and A. D. K.-T. Lam (Eds.), *Proceedings of the IEEE International Conference on Advanced Materials for Science and Engineering: Innovation, Science and Engineering, IEEE-ICAMSE 2016* (pp. 27-30). Institute of Electrical and Electronics Engineers Inc. doi: [10.1109/ICAMSE.2016.7840222](https://doi.org/10.1109/ICAMSE.2016.7840222)
- Huang, H.-C., y Yang, X. (2016). A comparative investigation of type-2 fuzzy sets, nonstationary fuzzy sets and cloud models. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 24(2), 213-227. doi: [10.1142/S0218488516500112](https://doi.org/10.1142/S0218488516500112)
- Kaplan, A., Skogstad, A. L., y Girshick, M. A. (1950). The prediction of social and technological events. *Public Opinion Quarterly*, 14(1), 93-110. doi: [10.1086/266153](https://doi.org/10.1086/266153)
- Kelleher, A. B., Sweeney, C., Foley, T., Hally, R. M., Meehan, E., Savage, E., Korn, B., y Cornally, N. (2019). An e-Delphi study to identify priority areas for education on advance care planning in COPD management. *Respiratory Care* (in press), doi: [10.4187/respcare.07072](https://doi.org/10.4187/respcare.07072)
- Landeta, J. (1999). *El Método Delphi: Una Técnica de Previsión para la Incertidumbre*. Barcelona: Ariel.
- Li, D., Liu, C., y Gan, W. (2009). A New Cognitive Model: Cloud Model. *International Journal of Intelligent Systems*, 24(3), 357-375. doi: [10.1002/int.20340](https://doi.org/10.1002/int.20340)
- Li, D., y Yi, D. (2008). *Artificial Intelligence with Uncertainty*. Boca Raton, FL: Chapman & Hall/CRC. doi: [10.1109/CIT.2004.1357163](https://doi.org/10.1109/CIT.2004.1357163)
- Linstone, H., y Turoff, M. (2011). Delphi: A brief look backward and forward. *Technological Forecasting & Social Change*, 78(9), 1712-1719. doi: [10.1016/j.techfore.2010.09.011](https://doi.org/10.1016/j.techfore.2010.09.011)
- Lixăndroi, D., y Lixăndroi, R. (2013). Using fuzzy models in managerial decision. En: A. R. Thomas, N. Al, y P. C. Bratianu (Eds.), *The Changing Business Landscape of Romania: Lessons for and from Transition Economies* (pp. 263-284). NY: Springer. doi: [10.1007/978-1-4614-6865-3_15](https://doi.org/10.1007/978-1-4614-6865-3_15)
- López-Gómez, E. (2018). El método Delphi en la investigación actual en educación: una revisión teórica y metodológica. *Educación XX1*, 21(1), 17-40. doi: [10.5944/educXX1.15536](https://doi.org/10.5944/educXX1.15536)
- Luzardo, M., y Rodríguez, P. (2019). Establecimiento de puntos de corte en pruebas cognitivas combinando Cloud Delphi y Teoría de Respuesta al Ítem. En: F. J. Murillo, y C. Martínez-Garrido (Eds.), *Actas del XIX Congreso Internacional de Investigación Educativa* (Vol. II, pp. 191-201). Madrid: Universidad Autónoma de Madrid. Recuperado de: https://aidipe2019.aidipe.org/files/2019/06/Actas_AIDIPE2019_Vol_II.pdf
- Maxey, D., y Kezar, A. (2015). Leveraging the Delphi technique to enrich knowledge and engage educational policy problems. *Educational Policy*, 30(7), 1-29. doi: [10.1177/0895904815586856](https://doi.org/10.1177/0895904815586856)
- Oñate, N., Ramos, L. y Díaz, A. (1988). Utilización del Método Delphi en la pronosticación: Una experiencia inicial. *Cuba: Economía Planificada*, 3(4), 9-48.
- Rodríguez, P. (2015). *Creación y Establecimiento de Estándares para la Evaluación de la Calidad de la Educación Superior: Un Modelo Adaptado a los Centros Universitarios Regionales de la UDELAR* (tesis doctoral). Madrid: Universidad Nacional de Educación a Distancia. Recuperado de: <https://dialnet.unirioja.es/servlet/tesis?codigo=71775>
- Rodríguez, P. (2019). Evaluación de un programa educativo universitario mediante el método Cloud Delphi. En: F. J. Murillo, y C. Martínez-Garrido (Eds.), *Actas del XIX Congreso Internacional de Investigación Educativa* (Vol. III, pp. 229-236). Madrid: Universidad Autónoma de Madrid. Recuperado de: https://aidipe2019.aidipe.org/files/2019/07/Actas_AIDIPE2019_Vol_III.pdf
- Roy, T. K., y Garai, A. (2012). Intuitionistic fuzzy Delphi method: more realistic and interactive

- forecasting tool. *Notes on Intuitionistic Fuzzy Sets*, 12(2), 37-50. Recuperado de: <http://ifigenia.org/images/0/06/NIFS-18-2-37-50.pdf>
- Sever, D., y Bostancı, K. T. (2020). The competencies of science teacher: A Delphi study. *European Journal of Education Studies*, 7(6), 82-112. doi: [10.5281/zenodo.3887330](https://doi.org/10.5281/zenodo.3887330)
- The R Foundation for Statistical Computing (2019). R (Version 3.6.2) [Computer software]. Recuperado de: <https://r-project.org>
- Tu, S.-C., Tai, D. W.-S., y Chen, J.-S. (2016). Construct digital literacy competences based on fusion fuzzy Delphi questionnaire. En: A. Hiramatsu, T. Matsuo, A. Kanzaki, and N. Komoda (Eds.), *Proceedings - 2016 5th IIAI International Congress on Advanced Applied Informatics, IIAI-AAI 2016* (pp. 281-284). Institute of Electrical and Electronics Engineers Inc. doi: [10.1109/IIAI-AAI.2016.70](https://doi.org/10.1109/IIAI-AAI.2016.70)
- Weiss, D. J., y Shanteau, J. (2003). Empirical assessment of expertise. *Human Factors*, 45(1), 104-114. doi: [10.1518/hfes.45.1.104.27233](https://doi.org/10.1518/hfes.45.1.104.27233)
- Xu, Q., y Xu, K. (2018). Assessment of air quality using a cloud model method. *Royal Society Open Science*, 5, 171580. doi: [10.1098/rsos.171580](https://doi.org/10.1098/rsos.171580)
- Yager, R., Kacprzyk, J., y Beliakov, G. (2011). *Recent Developments in the Ordered Weighted Averaging Operators: Theory and Practice*. Berlin: Springer-Verlag. doi: [10.1007/978-3-642-17910-5](https://doi.org/10.1007/978-3-642-17910-5)
- Yang, X.-J., Zeng, L., y Zhang, R. (2012). Cloud Delphi method. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 20(1), 77-97. doi: [10.1142/S0218488512500055](https://doi.org/10.1142/S0218488512500055)

6. ANEXOS

La función CloudDelphi sobre el lenguaje R

```
CloudDelphi = function(CurrentIntervals, ForegoingIntervals, ForegoingEntropy,
  Min = 10, Max = 80, K=6, Upper = TRUE, Lower = TRUE,
  Expectation = TRUE, Uncertainty = TRUE,
  DropExpertType = 1, DropSyntecticType = 3,
  DropExpertSize = .6, DropSyntecticSize = 1,
  CloudExpertIteration = 2000, CloudSyntecticIteration = 2000,
  CloudSyntecticColor = "black", CloudExpertColor = "grey66",
  DecimalPlaces = 2) {

# Debugging data
IntervalError = FALSE
if (length(CurrentIntervals)==0 || length(CurrentIntervals)/2!=round(length(CurrentIntervals)/2))
  IntervalError = TRUE && stop("There are some error in data", call. = FALSE) else
  Experts=length(CurrentIntervals)/2
for (i in 1:Experts) if (CurrentIntervals[2*i]-CurrentIntervals[2*i-1]<0) IntervalError = TRUE
if (length(ForegoingIntervals)!=0 && length(ForegoingIntervals)!=length(CurrentIntervals)) IntervalError = TRUE
if (length(ForegoingIntervals)==length(CurrentIntervals)) for (i in 1:Experts)
  if (ForegoingIntervals[2*i]-ForegoingIntervals[2*i-1]<0) IntervalError = TRUE
if (length(CurrentIntervals)!=2 && length(ForegoingEntropy)!=0 &&
  length(ForegoingEntropy)!=Experts && length(ForegoingEntropy)!=Experts+1) IntervalError = TRUE
if (length(CurrentIntervals)==2 && (length(ForegoingEntropy)>2 || length(ForegoingEntropy)>1)) IntervalError =
  TRUE

# Calculating expectation, entropy, hyper-entropy and weight
if (IntervalError) stop("There are some error in data", call. = FALSE) else {
  Ex_i=En_i=He_i=c(1:Experts)
  for (i in 1:Experts) {Ex_i[i]=(CurrentIntervals[2*i]+CurrentIntervals[2*i-1])/2
    En_i[i]=(CurrentIntervals[2*i]-CurrentIntervals[2*i-1])/6}
  if (length(ForegoingIntervals)==0) for (i in 1:Experts) He_i[i]=En_i[i]/K else
```

```

        for (i in 1:Experts) He_i[i]=(max(CurrentIntervals[2*i]-
        ForegoingIntervals[2*i],0)
        +max(ForegoingIntervals[2*i-1]-
        CurrentIntervals[2*i-1],0))/6}
Ex_s = mean(Ex_i)
En_s = (max(Ex_i+3*En_i)-min(Ex_i-3*En_i))/6
He_s = mean(He_i)
Weight = c(1:Experts)
for (i in 1:Experts) Weight[i] = 1/(abs((Ex_i[i]-Ex_s)/Ex_s)+En_i[i]+He_i[i])
SumWeight = sum(Weight)
for (i in 1:Experts) Weight[i] = Weight[i]/SumWeight

# Calculating the stability indexes of experts' judgments
DeltaEntropyIndex = UncertaintyIndex = c(0:Experts+1)
if (length(ForegoingEntropy)!=0) {for (i in 1:Experts) DeltaEntropyIndex[i] = abs(ForegoingEntropy[i] -
    En_i[i])/ForegoingEntropy[i]
    DeltaEntropyIndex[Experts+1] = abs(ForegoingEntropy[Experts+1] - En_s)/ForegoingEntropy[Experts+1]} else {
    DeltaEntropyIndex[1:Experts] = En_i
    DeltaEntropyIndex[Experts+1] = En_s}
UncertaintyIndex = He_i/En_i
UncertaintyIndex[Experts+1] = He_s/En_s

# Assembling data as matrix
Round = cbind(Ex_i, En_i, He_i, Weight, DeltaEntropyIndex[1:Experts], UncertaintyIndex[1:Experts])
Round = rbind(Round,c(Ex_s, En_s, He_s, 1, DeltaEntropyIndex[Experts+1], UncertaintyIndex[Experts+1]))
dimnames(Round) = list(c(), c("Ex","En","He","w", paste("Δ","En",sep=""), "Unc"))
rownames(Round) = rownames(Round, do.NULL=FALSE, prefix="Exp_")
rownames(Round)[Experts+1] = "Exp_s"
Round = t(Round)

# Preparing coordinate system
windows(width = 16, height = 9)
plot(0, 0, type="n", xaxt="s", yaxt="s", ylim=c(0,1.03), xlim=c(Min,Max),
     xlab="x", ylab="μ(x)", main="Cloud Delphi")

# Generating cloud's function
Cloud = function(Ex, En, He, Color, Type, Size, N,
                 CurveE=FALSE, CurveU=FALSE, CurveL=FALSE, CurveI=FALSE)
{P = array(0, dim=c(N,2))
  i = 1
  while (i <= N)
  {
    Em_i = rnorm(1, mean=En, sd=He)
    X_i = rnorm(1, mean = Ex, sd=Em_i)
    Mu_i = sqrt(2*pi)*Em_i*dnorm(X_i, mean=Ex, sd=Em_i)
    P[i,1] = X_i
    P[i,2] = Mu_i
    i = i+1
  }
  points(P, col=Color, pch=Type, cex = Size)
  D1 = En-3*He
  D2 = En+3*He

```

```

E = function(x) {sqrt(2*pi)*En*dnorm(x, Ex, En)}
U = function(x) {sqrt(2*pi)*D2*dnorm(x, Ex, D2)}
L = function(x) {sqrt(2*pi)*(D1)*dnorm(x, Ex, D1)}
I = function(x) {U(x)-L(x)}
if (CurveE) {curve(E(x), add=TRUE, col="green")}
if (CurveU) {curve(U(x), add=TRUE, col="red", lty=2, lwd=1)}
if (CurveL) {curve(L(x), add=TRUE, col="blue", lty=2, lwd=1)}
if (CurveI) {curve(I(x), add=TRUE, col="magenta")
  D3 = 2*D2*D1*sqrt((log(D2)-log(D1))/(D2^2-D1^2))
  X1 = Ex-D3
  X2 = Ex+D3
  segments(X1, L(X1), X1, U(X1), col=6, lty=1, lwd=4)
  segments(X2, L(X2), X2, U(X2), col=6, lty=1, lwd=4)
  text(X1+(Max-Min)*.13, (L(X1)+U(X1))/2, "Maximum uncertainty",
    col = "magenta")
  arrows(X1+(Max-Min)*.005, (L(X1)+U(X1))/2, X1+(Max-Min)*.05,
    (L(X1)+U(X1))/2, length = .1, angle=20, code=1, col="magenta")
}
par(new=TRUE)
if (CurveE | CurveU | CurveL | CurveI)
legend(.86*(Max-Min)+Min, .97, c(ifelse(CurveE, "Upper", ""),
  ifelse(CurveU, "Lower", ""),
  ifelse(CurveL, "Expectation", ""),
  ifelse(CurveI, "Uncertainty", "")),
  col = c(2,4,3,6), text.col = c(2,4,3,6),
  lty = c(2*CurveE, 2*CurveU, 1*CurveL, 1*CurveI),
  pch = NA, bg = "white")
}

# Plotting expert and syntectic cloud
if (Experts!=1) {for (i in 1:Experts) Cloud(Round[1,i], Round[2,i], Round[3,i],
  CloudExpertColor, DropExpertType, DropExpertSize,
  CloudExpertIteration)}
Cloud(Ex_s, En_s, He_s, CloudSyntecticColor, DropSyntecticType, DropSyntecticSize,
  CloudSyntecticIteration, Upper, Lower, Expectation, Uncertainty)
if (Experts!=1) {text(Round[1,Experts+1], 1.04, colnames(Round)[Experts+1],
  col=CloudSyntecticColor)
  for (i in 1:Experts) text(Round[1,i], 1.04, colnames(Round)[i],
    col=CloudExpertColor)
} else text(Round[1, Experts+1], 1.04, "Normal cloud generated",
  col=CloudSyntecticColor)

# Outputting matrix
round(Round, DecimalPlaces)
}

```