

R. 74778

015 32300



CUADERNOS DE LA FUNDACION

Nº 48

**MATEMÁTICA ACTUARIAL NO VIDA
CON MAPLEV**

Autores: José Luis Vilar Zanón

Yolanda García Cid

Junio, 1999

LISTA DE CUADERNOS DE LA FUNDACION MAPFRE **ESTUDIOS EDITADOS:**

1. Filosofía Empresarial
2. Resultados de la Encuesta sobre "Altos Profesionales de Seguros" (A.P.S.)
3. Dirección y Gestión de la Seguridad
4. Los Seguros en una Europa cambiante: 1990-1995 (No disponible)
5. La Distribución Comercial del Seguro: Sus Estrategias y Riesgos
6. Elementos de Dirección Estratégica de la Empresa
7. Los Seguros de Responsabilidad Civil y su Obligatoriedad de Aseguramiento
8. La Implantación de un Sistema de Controlling Estratégico en la Empresa
9. Técnicas de Trabajo Intelectual
10. Desarrollo Directivo: Una Inversión Estratégica
11. El Concepto de Seguridad en la Ciencia y la Ciencia de la Seguridad
12. Los Seguros de Salud y la Sanidad Privada
13. Calidad Total y Seguridad
14. El Reaseguro de Exceso de Pérdidas
15. El Coste de los Riesgos en la Empresa Española 1991
16. La Legislación Española de Seguros y su Adaptación a la Normativa Comunitaria

Número Especial: Informe sobre el Mercado de Seguros 1993

17. Medio Ambiente Seguro: Desarrollo Futuro
18. El Seguro de Crédito a la Exportación en los países de la OCDE (Evaluación de los resultados de los aseguradores públicos)
19. Una Teoría de la Educación
20. El Reaseguro en los Procesos de Integración Económica

Número Especial: Informe sobre el Mercado de Seguros 1994

21. La Nueva Regulación de las Provisiones Técnicas en la Directiva de Cuentas de la C.E.E. Provisiones Técnicas de Seguros de Vida en las Directivas Comunitarias
22. Rentabilidad y Productividad de Entidades Aseguradoras
23. Análisis de la Demanda de Seguro Sanitario Privado
24. El Seguro: Expresión de Solidaridad desde la Perspectiva del Derecho
25. El Reaseguro Financiero
26. El Coste de los Riesgos en la Empresa Española 1993
27. La Calidad Total como Factor para elevar la Cuota de Mercado en Empresas de Seguros
28. La Naturaleza Jurídica del Seguro de Responsabilidad Civil
29. Ruina y Seguro de Responsabilidad Civil Decenal

Número Especial: Informe sobre el Mercado de Seguros 1995

30. El Tiempo del Directivo
31. Tipos Estratégicos, Orientación al Mercado y Resultados Económicos: Análisis Empírico del Sector Asegurador Español
32. Decisiones Racionales en Reaseguro

33. La función del Derecho en la Economía
34. El Coste de los Riesgos en la Empresa Española 1995
35. El Control de Riesgos en Fraudes Informáticos
36. Cláusulas Limitativas de los Derechos de los Asegurados y Cláusulas Delimitadoras del Riesgo Cubierto. Las Cláusulas de Limitación Temporal de la Cobertura en el Seguro de Responsabilidad Civil

Número Especial: Informe sobre el Mercado de Seguros 1996

37. La Responsabilidad Civil por Accidente de Circulación. Puntual Comparación de los Derechos Francés y Español
38. Legislación y Estadísticas del Mercado de Seguros en la Comunidad Iberoamericana
39. Perspectiva Histórica de los Documentos Estadístico-Contables del Órgano de Control: Aspectos Jurídicos, Formalización y Explotación
40. Resultados de la Encuesta sobre la Organización y Gestión de la Seguridad en la Empresa (1996)
41. De Maastricht a Amsterdam: Un paso más en la integración europea

Número Especial: Informe sobre el Mercado de Seguros 1996

42. La Responsabilidad Civil por contaminación del entorno y su aseguramiento
43. Resultados de la Encuesta sobre Disponibilidad de Instalaciones de Protección contra Incendios en la Empresa 1997”
44. Resultados de la Encuesta sobre Implantación en la Empresa de la Ley de Prevención de Riesgos Laborales
45. Los Impuestos en una Economía Global

46. Evolución y Predicción de las Tablas de Mortalidad Dinámicas para la Población Española
47. El Fraude en el Seguro del Automóvil: Cómo detectarlo
48. Matemática Actuarial no Vida con MapleV

Copyright: F.M.E.

Prohibida la reproducción total o parcial de este trabajo sin el permiso escrito del autor o de la FUNDACION MAPFRE ESTUDIOS.

PRESENTACIÓN

MapleV es un asistente de cálculo simbólico y, al mismo tiempo, un lenguaje de programación de alto nivel que permite ser utilizado como una herramienta práctica y potente, capaz de suministrar visualizaciones y ejemplificar las aplicaciones prácticas en Matemática Actuarial No Vida.

Los profesores D^a Yolanda García Cid y D. José Luis Vilar Zanón contribuyen eficazmente con este trabajo a superar las dificultades que surgen en las enseñanzas de la Matemática Actuarial no Vida, tanto en su vertiente teórica, utilizando para ello ejemplos, visualizaciones y cálculos, como en la actividad práctica, permitiendo poner a prueba la aplicabilidad y validez de los modelos, técnicas y criterios ideados para resolver diversos problemas planteados en la práctica aseguradora.

Gran parte de estos problemas pueden encontrarse en áreas tales como la modelización, la ordenación de riesgos, la solvencia y el reaseguro, la tarificación a posteriori, el cálculo de provisiones y reservas, etc.

Con el trabajo realizado por la profesora García Cid y el profesor Vilar Zanón, que ahora se publica, se pone de manifiesto, de forma rigurosa y práctica, la gran potencia instrumental que el lenguaje de Programación de Alto Nivel MapleV puede poner a disposición de la enseñanza y la aplicación práctica de la Matemática Actuarial No Vida.

Antonio Guardiola Lozano
Director del Instituto de Ciencias del Seguro
de la Fundación MAPFRE Estudios

MATEMÁTICA ACTUARIAL NO VIDA CON MAPLEV

Autores: José Luis Vilar Zanón

Departamento de Economía Financiera y
Actuarial

Universidad Complutense de Madrid

Yolanda García Cid

Departamento de Métodos Cuantitativos

Universidad San Pablo CEU

Trabajo realizado dentro del proyecto de investigación financiado por la Dirección General de Enseñanza Superior, Ministerio de Educación y Cultura, Referencia nº PB 96-0099.

ÍNDICE

	<u>Página</u>
I. INTRODUCCIÓN	1
II. BREVE DESCRIPCIÓN DE MAPLEV	4
II.1. Trabajando con lápiz y papel	4
II.2. El entorno de edición	5
II.3. El asistente de cálculo simbólico	6
III. MAPLEV EN LA ENSEÑANZA DE LA MATEMÁTICA ACTUARIAL NO VIDA	10
III.1. Comentarios generales	10
III.2. Áreas de aplicación seleccionadas	12
III.2.1. Criterios seguidos en la selección	12
III.2.2. Ajuste de distribuciones del número de siniestros	14
III.2.3. Cálculo numérico de distribuciones compuestas	19
III.2.4. Aproximación Normal-Power	24
IV. BIBLIOGRAFÍA	27
 APÉNDICES:	
APÉNDICE 1: Procedimientos del paquete Matac3	28
APÉNDICE 2: Ajuste de la binomial negativa	32
APÉNDICE 3: Ajuste de la Poisson-Inversa-Gaussiana ..	37
APÉNDICE 4: Algoritmo recursivo de Panjer (distribución secundaria, discreta y aritmética)	43
APÉNDICE 5: La aproximación Normal-Power	51

I. Introducción .

La Matemática Actuarial No Vida está formada por un amplio conjunto de Teorías y técnicas cuyo fin es el de estudiar los fenómenos financiero-actuariales mediante modelos matemáticos, intentando dar respuesta a diversos problemas planteados en la práctica aseguradora. La diversidad y complejidad de las técnicas matemático estadísticas empleadas por la Matemática Actuarial No Vida, convierte en una tarea compleja su enseñanza, particularmente en su vertiente práctica.

Desde el punto de vista de la enseñanza las dificultades surgen a la hora de apoyar las explicaciones mediante visualizaciones y ejemplos: las primeras implican la necesidad de realizar representaciones gráficas, y los segundos la ejecución de cálculos complejos en un número muy grande. Como ejemplos de todo lo anterior podríamos citar la necesidad eventual de visualización de un conjunto de trayectorias de un proceso estocástico, la resolución de un problema de optimización, la realización de simulaciones, el ajuste de un modelo probabilístico partiendo de unos datos de siniestralidad, etc..

Dicha enseñanza también puede verse obstaculizada si es abordada desde un punto de vista exclusivamente teórico que esquive los aspectos prácticos y aplicados. La consecuencia sería entonces que no se podría mostrar hasta qué punto las técnicas y enfoques expuestos a lo largo del curso ayudan a comprender los fenómenos estudiados así como a resolver los problemas planteados en la práctica.

Una forma de dar respuesta a estas dificultades es la utilización de aplicaciones informáticas adecuadas a nuestras necesidades. A lo largo de los últimos años el desarrollo de los ordenadores personales ha permitido la aparición de aplicaciones científicas cada vez más completas, capaces de realizar tareas que antes solamente podían ser ejecutadas en grandes instalaciones mediante la aplicación de lenguajes de programación tales como Fortran, Pascal o C. Para nuestros fines, y dado el carácter variado de la Matemática utilizada en la Matemática Actuarial No Vida y de su singular campo de aplicación, necesitamos un lenguaje de programación que nos permita codificar algoritmos de carácter matemático muy variado y que al mismo tiempo sea de un manejo lo más sencillo posible. También debe ser capaz de suministrarnos visualizaciones de una forma rápida y sencilla.

Los lenguajes citados anteriormente le exigen al usuario poseer unos altos conocimientos de Programación si la meta es alcanzar los objetivos que hemos definido más arriba, esto es, la codificación de una gran variedad de algoritmos. Esta característica los hace poco prácticos de cara a su utilización en el aula como apoyo a la enseñanza teórico-práctica, ya que cualquiera de ellos requeriría un alto grado de entrenamiento por parte de los alumnos y profesores.

Nuestro interés se orienta por tanto hacia los lenguajes de programación de alto nivel, dentro de los denominados *Asistentes de Cálculo Simbólico*, cuyo espectacular desarrollo ha ido paralelo al de las máquinas. Estos lenguajes se denominan *de alto nivel* debido a la variedad de tareas que ya vienen programadas en forma de procedimientos, la variedad de estructuras

de datos y tipos, y la facilidad de programación. Dentro de esta categoría podemos citar por ejemplo las aplicaciones MapleV, Matemática o Derive.

A lo largo de la presente memoria mostraremos de qué manera el lenguaje MapleV nos permite adquirir de forma sencilla y rápida una enorme capacidad de cálculo y de visualización. Intentaremos mostrar mediante ejemplos que dicho lenguaje es suficientemente potente como para ser capaz de codificar muchos de los algoritmos contenidos en las técnicas aplicadas en Matemática Actuarial No Vida. También, comprobaremos que MapleV no solo es un lenguaje de programación, sino también una aplicación con unas capacidades gráficas y de presentación tales que la convierten en una herramienta cuya ayuda puede ser muy efectiva tanto en el campo de la enseñanza como en el del trabajo aplicado.

El contenido de la memoria es el siguiente. En el epígrafe 2 se da un breve descripción de las capacidades y limitaciones de MapleV. A continuación viene el epígrafe 3 donde se dan algunas ideas acerca de como se puede utilizar el programa como apoyo a la enseñanza. También se da una breve descripción de las técnicas y algoritmos pertenecientes a las áreas en las cuales se han elegido los ejemplos que figuran en los apéndices. Dichos apéndices son ejemplos de lo que se puede llegar a proponer como parte de la enseñanza práctica de la Matemática Actuarial No Vida. Sirven para mostrar el inmenso campo de enseñanza práctica que queda abierto una vez que se da entrada a esta herramienta informática. Dedicamos el primer apéndice a la presentación del paquete **matac3** que contiene todos los procedimientos que hemos programado y que son utilizados en el resto de

los apéndices. Todos ellos son hojas de trabajo MapleV, y han sido impresos directamente desde dicho programa para que el lector pueda apreciar la calidad de la presentación de forma directa.

II. Breve descripción de MapleV.

Las siguientes líneas solo son una breve presentación del asistente de cálculo simbólico MapleV. Para una introducción exhaustiva se recomienda la lectura de los manuales (guías del usuario y del programador, junto al diccionario de procedimientos, véase la bibliografía) que acompañan al programa.

II.1. Trabajando con lápiz y papel.

La idea general del programa MapleV consiste en emular lo que tradicionalmente se ha llamado la resolución de un problema "con lápiz y papel". Esto se intenta lograr mediante la conjunción de dos características: MapleV es un editor de texto y un asistente de cálculo simbólico. Mediante lo primero podemos conseguir una apariencia muy agradable en la presentación de las hojas de trabajo que básicamente están compuestas de texto, sentencias ejecutables, y resultados de las ejecuciones. Mediante lo segundo podemos proceder a la resolución de multitud de tareas matemáticas de naturaleza simbólica o numérica.

II.2.El entorno de edición.

Nos referiremos en primer lugar al editor de textos. Dentro del sistema operativo Windows 95, el lugar en donde se desarrolla nuestro trabajo se denomina *hoja de trabajo maple* (Maple worksheet). Esta hoja de trabajo hace las veces del papel que tenemos sobre una mesa y en el cual vamos a proceder a ejemplificar, resolver un problema, hacer una presentación, etc.

Se dispone de una amplia gama de posibilidades para definir el estilo de estas hojas. Así, podemos diferenciar los colores del texto, de las sentencias maple, y de los resultados de las ejecuciones de estas últimas. Además se pueden definir los estilos de cada uno de estos tres tipos de entrada: tipo de letra, tamaño, símbolos, negrita, cursiva, subrallado, centrado, etc.. En resumen, tenemos a nuestra disposición las capacidades comunes a todos los tratamientos de texto del tipo WYSIWYG (what you see is what you get). Además existen estilos (que también pueden ser redefinidos por el usuario) para cosas tales como título, autor, sección, subsección, mensajes de error, leyenda de las gráficas, etc. .Cada entrada o salida puede ser configurada a gusto del usuario. Por otro lado las hojas pueden ser divididas en secciones, subsecciones, etc. que son expandibles, con lo que se logran muy buenos efectos en las presentaciones. La posibilidad de crear hipervínculos posibilita la interconexión entre las distintas hojas, lo que puede resultar beneficioso a la hora de utilizar algún elemento ya explicado en una hoja anterior o de mandar al lector a alguna página de ayuda para aclarar la sintaxis de los comandos utilizados.

Las hojas maple se pueden salvar en varios formatos: ordinario, texto, Latex o HTML. El formato ordinario (extensión .mws) es el que conserva todas las características anteriormente descritas permitiéndonos recuperarlas mediante la simple recarga del fichero en el entorno de edición. El formato Latex (extensión .tex) traduce nuestra hoja al lenguaje Latex; este lenguaje es un compilador para el tratamiento de textos que se ha convertido en los últimos años en una herramienta común para la edición de textos con contenido matemático, tanto a nivel de publicaciones periódicas como de libros. El formato HTML nos permite salvar una hoja maple para su posterior inclusión en una página de internet. Finalmente el formato de texto resulta útil cuando un programa (es decir, un conjunto de sentencias escritas en MapleV) requiere la utilización de mucha memoria, viéndonos obligados a ejecutarlo en un entorno distinto del Windows, por ejemplo MS-DOS o UNIX.

II.3.El asistente de cálculo simbólico.

Las aplicaciones agrupadas bajo esta denominación intentan recrear las matemáticas tal y como las piensa el ser humano, con el objetivo principal de resolver problemas de naturaleza simbólica. Ejemplos de este tipo de problemas son los siguientes: MapleV es capaz de representar y operar sobre números racionales y también sobre irracionales algebraicos o trascendentes (por ejemplo una raíz cuadrada o el número π), manipular expresiones algebraicas (factorización, simplificación, desarrollo), despejar incógnitas de ecuaciones, derivar funciones, calcular primitivas, operar con deltas de Dirac, etc.. Es decir que tenemos la posibilidad de abordar

la resolución de una tarea matemática no exclusivamente desde el punto de vista del Cálculo Numérico, sino del habitual que adoptamos cuando empezamos a razonar y a trabajar delante de una hoja de papel en blanco.

Esto se lleva a cabo mediante la *librería* maple de procedimientos o comandos programados en origen, que nos permite abordar la resolución de casi cualquier problema contenido en la Matemática elemental y superior. Estos procedimientos pueden ser de dos tipos: aquellos cuya carga en la memoria ram es automática cada vez que se inicia el programa (comandos residentes en memoria), y aquellos que no residen en la memoria. Los primeros son procedimientos cuyo uso es muy común, como por ejemplo el cálculo de una integral definida, de una derivada o de la solución de un sistema de ecuaciones. Los segundos son mucho más numerosos y están reagrupados en *paquetes* de procedimientos. Cada paquete se identifica con una área matemática concreta: por ejemplo existen paquetes de álgebra lineal, ecuaciones diferenciales, ecuaciones en derivadas parciales, estadística, finanzas, gráficas, animación de gráficas, etc. La utilización de uno de estos procedimientos se hace mediante la carga previa en memoria de todo el paquete en el que está contenido o, si se prefiere no abusar de los recursos del ordenador, mediante una llamada individual al procedimiento. El conjunto de todos estos paquetes forma la librería maple de procedimientos ya programados.

Una gran ventaja de MapleV de cara a la consecución de los dos objetivos mencionados al principio de esta memoria, es la posibilidad de que el usuario defina sus propias librerías. Estas librerías se componen de paquetes de

procedimientos escritos por el usuario. Cada paquete será específico de un campo de aplicación en donde aquel desarrolle su actividad matemática. A modo de ejemplo, nosotros hemos reagrupado algunos procedimientos referidos a la Matemática Actuarial No Vida dentro del paquete **matac3**, que está a su vez incluido en una librería definida por nosotros. Los procedimientos de dicho paquete serán presentados en el primer apéndice.

Un asistente de matemáticas que no fuese capaz de obtener soluciones numéricas no sería de gran ayuda. Para la evaluación numérica MapleV cuenta con una serie de procedimientos que nos permiten pasar en un momento dado de la solución obtenida a través del cálculo simbólico a su evaluación numérica. Así por ejemplo, podemos realizar una serie de cálculos que resuelvan algún problema en modo simbólico, trabajando con números racionales e irracionales, hasta el momento en que lleguemos a una solución que será exacta por venir expresada simbólicamente. En ese momento podemos *evaluar numéricamente* dicha solución, es decir, calcular una aproximación numérica de la solución con un grado de precisión fijado por el usuario. De esta forma nos podemos ahorrar todos los errores de truncamiento y redondeo que se hubieran producido inevitablemente de haberse abordado la resolución del problema desde la perspectiva del Cálculo Numérico. Esta evaluación numérica se hace en gran parte utilizando el comando **evalf**, que es uno de los procedimientos residentes en la memoria.

Una característica muy interesante de MapleV es que se trata de un lenguaje de programación. Gracias a ello podemos escribir nuestros propios

procedimientos y construir paquetes referidos a áreas específicas de nuestra actividad. La capacidad de programación viene dada por la existencia de estructuras condicionales (**if then else**, **piecewise**) y repetitivas (**for**, **while**), junto a una gran variedad de estructuras de datos y tipos predefinidos.

La transición entre las dos etapas de cálculo simbólico y cálculo numérico es la clave de la programación en lenguaje MapleV. De la elección del momento en que nuestro código debe evaluar numéricamente una solución depende en gran medida la precisión de dicha solución y la rapidez con la que será alcanzada. A este respecto es importante señalar que MapleV opera con un formato software de coma flotante representado en base diez, distinto del formato hardware (como es sabido este último viene representado en base dos). Esto hace que los cálculos numéricos sean más lentos. Un ejemplo de situaciones susceptibles de plantear este problema son aquellas que requieren el manejo de arrays grandes: pensemos por ejemplo en problemas de valoración de contratos derivados, de análisis multivariante o más en general, de cualquier problema que implique el uso de grandes matrices. Conviene precisar que por *lentitud* nos referimos a tiempos de ejecución superiores a los dos o tres minutos, más allá de los cuales dicha ejecución puede no resultar rentable si queremos llevarla a cabo en clase como ejercicio práctico para los alumnos.

Como remedio a esta situación, MapleV cuenta con el comando **evalhf** para la evaluación numérica en formato hardware de coma flotante. Gracias a él podemos aumentar la rapidez de los cálculos numéricos, pero tenien-

do en cuenta que perderemos algo de la potencia simbólica del lenguaje ya que algunos comandos no permiten ser anidados dentro de **evalhf**. Por ejemplo, no podremos utilizar **evalhf** al calcular una integral definida, ya que el comando **int**, que calcula la integral de una función para unos límites de integración dados, no puede ser anidado dentro de aquel. Esta última observación se refiere a la versión 4 del lenguaje, que fue la utilizada para redactar las hojas de trabajo presentadas en esta memoria, si bien posteriormente estas fueron traducidas a la versión 5.

La versión 5 intenta remediar esta situación mediante la inclusión de una nueva estructura, **arrayhf**. Esta consiste en arrays de números en formato hardware de coma flotante, lo que supone un incremento en la velocidad de los cálculos numéricos. También posibilita la ejecución de procedimientos dentro de la aplicación Matlab. Este es un Asistente de Cálculo Numérico, con lo que se supone que esta capacidad de conexión debería resolver definitivamente las dificultades de tipo numérico ya mencionadas.

III. MapleV en la enseñanza de la Matemática Actuarial No Vida.

III.1. Comentarios generales.

La utilización de MapleV puede enfocarse en dos direcciones. Por una parte para apoyar las explicaciones teóricas mediante los ejemplos, visualizaciones y cálculos que estimemos oportunos. Por otra, para ser el instrumento que posibilite una actividad práctica a lo largo de la cual se pongan a prueba la aplicabilidad y validez de los modelos, técnicas, y criterios ideados

para resolver diversos problemas planteados en la práctica aseguradora. Encontramos una gran variedad de estos problemas en áreas tales como modelización, ordenación de riesgos, solvencia y reaseguro, tarificación a posteriori, cálculo de provisiones y reservas, etc..

El análisis y posible resolución de un problema práctico genera toda una actividad de cara a los alumnos que debe ser realizada por el profesor y que podría ser descompuesta en los siguientes pasos sucesivos:

1. Identificación de algún problema concreto que se desea analizar y que implique la codificación de algoritmos.
2. Planteamiento del problema a lo largo de la clase y exposición de las técnicas y algoritmos de diversa naturaleza que son aplicables en la resolución del problema.
3. Traducción de las herramientas del punto anterior a los oportunos procedimientos maples por parte del profesor.
4. Redacción de una hoja de trabajo que ejemplifica la resolución de un ejemplo, y a lo largo de la cual se utilizan los anteriores procedimientos.
5. Explicación en clase con el apoyo de dicha hoja.
6. Proposición a los alumnos de tareas similares en forma de ejercicios, para su posterior resolución con la ayuda del asistente de cálculo simbólico, utilizando los procedimientos ya programados.
7. Corrección de los ejercicios.

La aplicación del anterior esquema implica que los alumnos hayan adquirido previamente un manejo mínimo del asistente, tanto en su vertiente de entorno de edición como en la de lenguaje de programación. Este aprendizaje es una meta mucho más accesible y realista que la representada por la programación de los mismos procedimientos en alguno de los lenguajes referidos en la introducción.

III.2. Áreas de aplicación seleccionadas.

III.2.1. Criterios seguidos en la selección.

A continuación damos un rápido resumen de las hojas de trabajo presentadas en los apéndices. Nuestro objetivo es el de dar algunos ejemplos que aclaren la sencillez en el manejo y las capacidades de programación, de cálculo y de visualización suministradas por el programa, mostrando de esta forma la valiosa ayuda que en la enseñanza teórico-práctica puede suministrarnos este asistente.

No obstante, pensamos que las tres áreas seleccionadas ilustran bastante ciertas características ya mencionadas de la Matemática de los Seguros No Vida: el carácter singular del campo en donde se aplican las técnicas matemático estadísticas, la complejidad y alto número de cálculos numéricos, y la necesidad de visualización.

La primera de las anteriores características es particularmente cierta a la hora de modelizar el riesgo actuarial. Encontramos aquí las distribuciones de probabilidad de tipo ponderado o compuesto. Las primeras plantean problemas numéricos en todo el proceso de ajuste, particularmente a la hora de estimar sus parámetros mediante el Método de la Máxima Verosimilitud, mientras que para las segundas la dificultad, motivada por el cálculo de convoluciones, reside en el cálculo numérico de las probabilidades por ellas suministradas.

La peculiaridad de estos modelos probabilísticos, así como de los algoritmos que resuelven el cálculo numérico de distribuciones compuestas, hace que sea imposible encontrar una aplicación informática diseñada a la medida de nuestras necesidades. Por esta razón hemos creído interesante mostrar de qué forma es posible ejemplificar y practicar en clase dentro de estas dos áreas. Esto se lleva a cabo mediante tres hojas de trabajo dedicadas al ajuste de las distribuciones binomial negativa y Poisson-inversa gaussiana al número de siniestros, y al cálculo de probabilidades de distribuciones compuestas, ya sean éstas últimas aplicadas al número de siniestros (Distribución Neyman de tipo A) o al daño total (véanse los apéndices 2, 3, y 4).

Otro ejemplo de peculiaridad es la utilización de ciertas aproximaciones de la distribución del daño total, como por ejemplo la Normal-Power. Es aquí donde más patente se hace la necesidad de visualización que facilite la comprensión del comportamiento de dicha aproximación, o su enfoque como un cambio de variable en una distribución normal. También desde los

puntos de vista visual y calculístico, se puede estudiar la sensibilidad de la probabilidad de ruina respecto de ciertos parámetros de un modelo discreto de las reservas, previamente expuesto en clase. Estos son básicamente los contenidos de la última hoja de trabajo (véase el apéndice 5).

En los siguientes apartados presentamos con más detalle algunas características de tipo matemático de las anteriores hojas de trabajo.

III.2.2. Ajuste de distribuciones del número de siniestros.

Para el ajuste de una distribución al número de siniestros fijamos un intervalo de tiempo pasado (que supondremos unitario), y observando la totalidad de la cartera, anotamos el número de pólizas que han sufrido $0, 1, 2, \dots, n, \dots$ siniestros. La distribución será por tanto ajustada a la variable aleatoria *Número de siniestros producidos por una póliza cualquiera de la cartera durante el periodo de tiempo*. A continuación las etapas recorridas son las siguientes:

- Cálculo de la media y varianza muestrales.
- Estimación de los parámetros de la distribución.
- Cálculo de las frecuencias teóricas suministradas por el modelo estimado.
- Reagrupamiento de las clases teóricas y observadas.
- Cálculo del estadístico de Pearson.

- Determinación del p-valor. Rechazo o no rechazo de la hipótesis nula.

Las anteriores etapas se completan dos veces, según que la estimación de los parámetros sea por el método de los momentos o el de la máxima verosimilitud.

Estudiamos el ajuste de dos distribuciones de Poisson ponderadas respectivamente por una gamma y una inversa gaussiana.

Ajuste de la binomial negativa. En este caso el ajuste ha sido hecho sobre los siguientes datos tomados de *Panjer y Willmot (1992) p. 301* :

Número de siniestros	Número de pólizas
0	20592
1	2651
2	297
3	41
4	7
5	0
6	1
Total de pólizas:	23589

La función de cuantía de la binomial negativa es:

$$p_n = \frac{\Gamma(r+n)}{\Gamma(r)n!} \left(\frac{1}{1+\beta}\right)^r \left(\frac{\beta}{1+\beta}\right)^n, \quad n = 0, 1, 2, \dots, \quad r, \beta > 0.$$

La media y la varianza de la distribución son:

$$\mu = r\beta, \quad \sigma^2 = r\beta(1+\beta).$$

Llamando \bar{X} , S^2 , a la media y varianza muestrales, los estimadores por momentos de los parámetros son

$$\bar{r} = \frac{\bar{X}^2}{S^2 - \bar{X}}, \quad \bar{\beta} = \frac{S^2}{\bar{X}} - 1.$$

En cuanto a los estimadores de máxima verosimilitud (emv), se tiene lo siguiente. Llamemos n al número total de pólizas, N_k al número de pólizas que han sufrido k siniestros durante el periodo de observación. Entonces el emv \hat{r} es la raíz de la ecuación:

$$H(r) = n \log \left(1 + \frac{\bar{X}}{r} \right) - \sum_{k=1}^{\infty} N_k \left(\sum_{m=0}^{k-1} \frac{1}{m+r} \right) = 0$$

que será única siempre que se tenga $n(n-1)^{-1}S^2 > \bar{X}$ (véase *Panjer y Willmot (1992) p. 310*). Aproximaremos \hat{r} con el método de Newton-Raphson, admitiendo como raíz la k -ésima iteración de la expresión:

$$r_k = r_{k-1} - \frac{H(r_{k-1})}{H'(r_{k-1})}.$$

Como valor inicial o semilla tomaremos la estimación por momentos \bar{r} .

Una vez aproximado este valor, el emv de β se aproxima mediante

$$\hat{\beta} = \frac{\bar{X}}{\hat{r}}.$$

Digamos finalmente que para reagrupar las clases teóricas se ha seguido el criterio de que en todas ellas haya al menos cuatro pólizas.

Ajuste de la Poisson-inversa gaussiana. En este caso hemos realizado el ajuste a los siguientes datos tomados de *Panjer y Willmot (1992) p. 301* :

Número de siniestros	Número de pólizas
0	7840
1	1317
2	239
3	42
4	14
5	4
6	4
7	1
Total de pólizas:	9461

Esta distribución consiste en una Poisson ponderada por una inversa gaussiana. La función de densidad de la inversa gaussiana es:

$$f(\lambda) = g (2\pi h \lambda^3)^{-1/2} \exp \left\{ -\frac{(\lambda - g)^2}{2h\lambda} \right\}, \quad \lambda > 0, h, g > 0,$$

y llamando Φ a la función de distribución $N(0, 1)$, su función de distribución se expresa de la siguiente forma:

$$F(\lambda) = \Phi \left[(h\lambda)^{-1/2} (\lambda - g) \right] + e^{2gh^{-1}} \Phi \left[- (h\lambda)^{-1/2} (\lambda + g) \right].$$

Su media y varianza son $\mu, \mu\beta$, respectivamente.

Las cuantías de la Poisson inversa gaussiana vienen dadas por:

$$p_k = \int_0^{+\infty} \frac{\lambda^k}{k!} e^{-\lambda} f(\lambda) d\lambda, \quad \lambda > 0.$$

En el apéndice 3 estas cuantías son calculadas recursivamente de la siguiente forma (véase *Lemaire (1995) p. 36*):

$$p_0 = \exp \left\{ \frac{g}{h} \left[1 - (1 + 2h)^{1/2} \right] \right\}$$

$$p_1 = g p_0 (1 + 2h)^{-1/2}$$

$$(1 + 2h) k (k - 1) p_k = h (k - 1) (2k - 3) p_{k-1} + g^2 p_{k-2}, \quad k = 2, 3, \dots$$

La media y la varianza son:

$$\mu = g, \quad \sigma^2 = g(1 + h).$$

De aquí se deducen los siguientes estimadores por momentos:

$$\bar{g} = \bar{X}, \quad \bar{h} = \frac{S^2}{\bar{X}} - 1, \quad \text{siempre que sea } S^2 > \bar{X}.$$

En cuanto a los estimadores de máxima verosimilitud se tiene que $\hat{g} = \bar{X}$, mientras que \hat{h} es la raíz positiva de la ecuación (véase *Lemaire (1995) p. 37*)

$$\frac{1}{n} \sum_{k=0}^{\infty} N_k T_k(h) = \bar{X},$$

siendo las $T_k(h)$ funciones racionales que se forman recursivamente según la siguiente regla

$$T_0(h) = \bar{X} (1 + 2h)^{-1/2}$$

$$T_k(h) = \frac{1}{1+2h} \left[(2k - 1) h + \frac{\bar{X}^2}{T_{k-1}(h)} \right], \quad k = 1, 2, 3, \dots$$

Nuevamente utilizaremos el método de Newton-Raphson para aproximar el valor de \hat{h} . El reagrupamiento de las clases se lleva a cabo siguiendo el criterio expuesto en el apartado anterior.

III.2.3.Cálculo numérico de distribuciones compuestas.

Consideremos las variables aleatorias

- N v.a. discreta con soporte en el conjunto $\{0, 1, 2, 3, \dots\}$ y distribución $(p_n)_{n=0}^{\infty}$.
- $(X_i)_{i=1}^{\infty}$ sucesión de v.a. independientes idénticamente distribuidas con distribución V tal que $V(0) = 0$.
- Las v.a. N, X_i independientes.

Entonces la suma aleatoria de variables aleatorias

$$S = \sum_{i=0}^N X_i \quad , \quad (S = 0 \text{ cuando } N = 0)$$

es una *variable aleatoria compuesta*. Su distribución también se denomina *distribución compuesta*, y es

$$F(x) = P\{S \leq x\} = \sum_{i=0}^{\infty} p_n V^{*n}(x).$$

La v.a. N se denomina *primaria* y X es la *secundaria*. Sus distribuciones p_n, V se denominan de igual forma.

En cuanto a la clasificación de una distribución compuesta (tipos, discreto, absolutamente continuo o mixto) resulta que:

- Si V es discreta entonces F es discreta.
- Si V es absolutamente continua con densidad v , entonces F es mixta: posee un átomo en el origen $x = 0$ con cuantía $f(0) = p_0$, y el trozo diferenciable con densidad v corresponde al intervalo $x > 0$.

En ambos casos, llamando v ya sea a la función de cuantía o de densidad de la distribución secundaria, y f a la derivada (generalizada) de F se tiene:

$$f(x) = P\{S \leq x\} = \sum_{i=0}^{\infty} p_n v^{*n}(x).$$

Las distribuciones compuestas tienen una gran importancia en la Matemática Actuarial No Vida. Se utilizan como modelos probabilísticos en muchas situaciones.

Como distribución del número de siniestros, proporcionan una gran variedad de modelos probabilísticos que enriquece nuestro arsenal de cara al ajuste de distribuciones a dicha variable. Estos modelos suelen ser biparamétricos o triparamétricos. Cuando se elige una Poisson como distribución primaria, las distribuciones compuestas obtenidas tomando distintas secundarias de tipo discreto (por ejemplo Poisson, binomial, o pertenecien-

tas a la familia binomial negativa extendida), forman una familia tal que para una media y varianza dadas siempre se puede elegir un miembro con asimetría tan grande como se quiera.

Así mismo, es bien conocida su utilización como distribución del daño total una vez que se ha procedido al ajuste de las distribuciones primaria (número de siniestros N) y secundaria (cuantías individuales de los siniestros X).

Por otro lado, en teoría de la ruina tanto las *probabilidades de supervivencia* a tiempo finito como la probabilidad de supervivencia en tiempo continuo con horizonte infinito son distribuciones compuestas. En el primer caso, la propia distribución del daño total, y en el segundo una compuesta cuya distribución primaria es geométrica mientras que la secundaria depende de la distribución V de las cuantías individuales.

Desde el punto de vista numérico, el cálculo de probabilidades de distribuciones compuestas se ve dificultado por el hecho de implicar el cálculo de convoluciones de la distribución secundaria.

Históricamente fue abordado mediante simulación e integración Monte-Carlo, hasta que en 1980 fue presentado el hoy ampliamente difundido *algoritmo recursivo de Panjer*, que permite calcular dichas probabilidades sin necesidad de abordar las molestas convoluciones.

Dicho resultado posee muchas extensiones y generalizaciones, pero nosotros vamos a presentarlo y codificarlo en su forma más básica que puede ser consultada en *Panjer y Willmot (1992) p. 197, corolario 6.6.1.*

Las condiciones que deben cumplir las distribuciones primaria y secundaria para poder aplicar el algoritmo son las siguientes.

1. La distribución primaria debe pertenecer a la clase $C(a, b, 0)$, de manera que tiene que cumplir lo siguiente

$$\exists a, b \text{ constantes} / \frac{p_n}{p_{n-1}} = a + \frac{b}{n}, \quad n = 1, 2, 3, \dots$$

Se demuestra que las únicas distribuciones de $C(a, b, 0)$ son la Poisson, binomial, binomial negativa, geométrica.

2. La distribución secundaria tiene que ser de tipo discreto y *aritmética*. Esto último significa que su soporte debe estar incluido en el conjunto

$$\{x = i h; i = 0, 1, 2, \dots\}$$

siendo $h > 0$.

Siempre que las distribuciones primaria y secundaria verifiquen las anteriores condiciones, las cuantías de la compuesta se podrán obtener de la

siguiente forma: (véase *Panjer y Willmot (1992) p.193*)

$$f(x) = \sum_{y=1}^{\infty} \left(a + b \frac{y}{x}\right) v(y) f(x - y) \quad , x = 1, 2, 3, \dots$$

De manera que disponiendo de $f(0)$ podemos calcular $f(1)$, con estos dos valores calculamos $f(2)$ y así sucesivamente. Si queremos calcular $f(x)$ tendremos que calcular previamente desde $f(0)$ hasta $f(x - 1)$.

Además el algoritmo es *recursivo*: el procedimiento que lo lleve a cabo deberá llamarse a sí mismo tantas veces como sea necesario para calcular todos los valores previos al $f(x)$. Esta característica haría que el cálculo pudiera llegar a ser muy lento si no fuera por una posibilidad que nos da el lenguaje MapleV para remediar estos casos: dentro del procedimiento hemos declarado la *option remember* cuyo efecto es que cada vez que el procedimiento es llamado crea una tabla en donde almacena todos los valores intermedios que haya calculado para llegar hasta el valor pedido. Dicha tabla es almacenada en la memoria y a la siguiente llamada el procedimiento, antes de calcular un valor, lo busca en dicha tabla y si lo encuentra lo toma. El efecto es que la velocidad de cálculo se incrementa de forma notabilísima, pero tiene un posible inconveniente. La llamada del procedimiento funciona mediante la especificación de tres parámetros: la distribución primaria, la secundaria, y el x par el cual queremos calcular la probabilidad $f(x)$. La creación de la tabla es local a cada llamada de parámetros: significa esto que serán creadas tantas tablas como distribuciones primarias y/o secundarias utilizemos en una sesión, corriéndose entonces el riesgo de que la memoria del ordenador se vea desbordada.

El algoritmo presentado más arriba tiene la siguiente limitación: solo sirve para distribuciones secundarias tales que $V(0) = 0$. Significa esto, por ejemplo, que no podemos realizar todavía ningún cálculo en situaciones que incluyan casos como los de los siniestros con cuantía nula con probabilidad mayor que cero, o con distribuciones compuestas para el número de siniestros N cuya distribución secundaria tenga un átomo en el origen, tales como las de la clase $C(a, b, 0)$.

Es bien conocido que esto puede remediarse de la siguiente forma. En primer lugar se cambia la distribución primaria rectificando uno de sus parámetros, multiplicándolo por $(1 - v_0)$. En segundo lugar se cambia la distribución de la variable secundaria X por la de la variable aleatoria condicionada $X_+ = X | X > 0$. Se sabe que la nueva distribución suministra las mismas probabilidades que la original, y al estar dentro de las hipótesis del resultado ($V_+(0) = 0$) el problema queda resuelto. El proceso está explicado con todo detalle en *Panjer y Willmot (1992) p. 218*.

III.2.4. Aproximación Normal-Power.

La aproximación Normal-Power sirve para aproximar las probabilidades de la distribución del daño total cuando la información con la que contamos consiste únicamente en tres características de esta distribución: media, desviación típica, y asimetría (μ, σ, γ) . Gracias a ella se pueden hacer por ejemplo, consideraciones acerca de la solvencia del negocio sin necesidad de ajustar las distribuciones del número de siniestros y de las cuantías individuales. En general esta aproximación será válida para carteras con una

asimetría no demasiado grande: podemos pensar en un coeficiente menor que 1.5 (véase *Beard, Pentikäinen, Pesonen (1984)*).

Llamando $s = \frac{S-\mu}{\sigma}$ a la v.a. tipificada del daño total, esta aproximación calcula un número real y tal que $P\{s \leq s\} \simeq N(0, 1)(y)$. Nos suministra pues un punto en el eje de abscisas tal que el área situada debajo de la campana de Gauss sobre el intervalo $(-\infty, y]$ es aproximadamente igual a la probabilidad de la distribución compuesta del daño total. De hecho tenemos una de estas transformaciones por cada valor de la asimetría de la variable S . Hemos tomado la siguiente expresión de la NP de *Beard, Pentikäinen, Pesonen (1984)*

$$y = v_{\gamma}^{-1}(s) = \begin{cases} s - \frac{\gamma}{6}(s^2 - 1) + \frac{\gamma^2}{36}(4s^3 - 7s), & s < -\sqrt{\frac{7}{4}} \\ s - \frac{\gamma}{6}(s^2 - 1), & -\sqrt{\frac{7}{4}} < s < 1 \\ \sqrt{1 + \frac{9}{\gamma^2} + \frac{6s}{\gamma}} - \frac{3}{\gamma}, & s > 1 \end{cases}$$

Mediante representaciones gráficas de esta función se puede ver de qué forma los tres trozos se empalman con continuidad. De especial interés es el estudio de la variación de la función v_{γ}^{-1} respecto de la asimetría del riesgo: se puede comprobar que cuando $\gamma \rightarrow 0^+$ la sucesión de gráficas tiende hacia la diagonal, lo que representaría que la aproximación normal es un caso particular de la NP para asimetría nula. Todo esto corresponde al apartado de lo que hemos denominado visualizaciones.

En el apartado práctico, podemos introducir un modelo para las reservas en tiempo discreto, en donde cada periodo represente un ejercicio:

$$R_n = R_0 + \sum_{i=1}^n (i c - S_i), \quad n = 1, 2, 3, \dots$$

siendo R_n la reserva al final del periodo n , $c = (1 + \theta)\lambda E\{X\}$ el ingreso por primas recargadas por unidad de tiempo (θ recargo de seguridad, $E\{X\}$ cuantía media del siniestro, λ número medio de siniestros), S_i daño total producido en la cartera durante el periodo i .

El objetivo consistirá entonces en aproximar la probabilidad de ruina en un periodo

$$P\{R_1 < 0\} = P\{S_1 \geq R_0 + c\}$$

mediante la Normal-Power, así como en estudiar la sensibilidad de esta probabilidad respecto de algunas de las variables de decisión del modelo, tales como las reservas iniciales o el recargo de seguridad. Realizamos este estudio con la ayuda de representaciones gráficas y de animaciones de estas respecto de las dos anteriores variables.

Un actividad más ambiciosa consistiría en abordar el estudio de la probabilidad de ruina a lo largo de varios periodos, pero se vería dificultada por el hecho de tener que calcular integrales multidimensionales, una de las tareas más rudas en Análisis numérico. Este es el motivo de limitarnos a un solo periodo.

IV. Bibliografía.

Beard,R.E., Pentikäinen,T. , Pesonen, E. (1982). *Risk Theory. The Stochastic Basis of Insurance*. Chapman and Hall.

DeGroot,M.H. (1988). *Probabilidad y Estadística*. Addison-Wesley Iberoamericana.

Latorre Llorens,L. (1992). *Teoría del Riesgo y sus Aplicaciones a la Empresa Aseguradora*. Editorial Mapfre.

Lemaire,J. (1995). *Bonus-Malus System in Automobile Insurance*. Kluwer Academic Press.

Monagan,M.B., Geddes,K.O., Heal,K.M., Labahn,G., Vorkoetter,S.M. (1996). *Maple V Programming Guide*. Springer-Verlag

Nieto de Alba,U., Vegas,J. (1995). *Matemática Actuarial*. Editorial Mapfre.

Panjer,H.H., Willmot,G.E. (1992). *Insurance Risk Models*. Society of Actuaries.

Redfern,D. (1996). *The Maple Handbook*. Springer-Verlag.

Apéndice 1: Procedimientos del Paquete matac3.

En esta hoja presentamos los procedimientos que componen el paquete **matac3** incluido en una librería definida por nosotros.

Los procedimientos han sido programados de forma que contienen una sentencia (**description**) gracias a la cual devuelven un mensaje de ayuda en el cual podemos leer para qué sirven, cuales son sus parámetros, etc. .

La sentencia que devuelve dichos mensales de ayuda es la siguiente:

```
readlib('matac3')[NombreDelProcedimiento];
```

Dicha sentencia no necesita la carga previa en memoria del paquete.

Por otro lado , la **option copyright** oculta el código de los procedimientos. Nuestros procedimientos la incorporan, razón por la cual solo aparecen los mensajes incorporados dentro de **description**.

Para los parámetros de las distribuciones, hemos adoptado la notación de *Panjer y Willmot (1992)*.

Para estimar por momentos los parámetros de la binomial negativa tenemos el procedimiento **EstimaPorMomentosBiNeg**:

```
> readlib('matac3')[EstimaPorMomentosBiNeg];
```

```
proc(m::numeric, v::numeric)
```

```
description `Estima por momentos los dos parámetros  $r$  ,  $\beta$ , de la Binomial Negativa.  $m$  ,  $v$  son la media y varianza muestrales.`
```

```
...
```

```
end
```

El procedimiento **CuantiaBiNeg** calcula las probabilidades para el modelo estimado. Sus parámetros son **x** , **y** que corresponden a los parámetros β , r de la distribución.

```
> readlib('matac3')[CuantiaBiNeg];
```

```
proc(x::numeric, y::numeric)
```

```
description `Calcula las cuantías de una Binomial Negativa con ciertos parámetros.  $x$ ,  $y$  son los parámetros  $\beta$ ,  $r$  que habrán sido estimados previamente. El número de Cuantías a calcular viene dado por el número de componentes de la lista FrecObs`
```

```
...
```

```
end
```

El procedimiento **ReagrupaTeoricas** reagrupa las clases teoricas de forma que en ninguna de las que resulten haya menos de 4 asegurados. Nos devuelve una lista con las frecuencias teóricas de las clases reagrupadas. Sus parámetros son: **Tot** (n° total de asegurados), **X** (la lista que contiene las probabilidades de la distribución teórica).

```
> readlib('matac3')[ReagrupaTeoricas];
```

```
proc(Tot::integer, X::list)
```

```
description `Reagrupa las clases teóricas de forma que ninguna de ellas contenga menos de 4 asegurados. Tot es el número total de asegurados. X es la lista que contiene las cuantías d`
```

e la Binomial Negativa calculadas mediante el procedimiento *CuantiabiNeg*.

...
end

El procedimiento **ReagrupaObservadas** reagrupa las frecuencias observadas siguiendo el mismo patrón que las teóricas. Sus parámetros son: *l* (nº de clases teóricas), **X** (lista de las frecuencias observadas)

```
> readlib('matac3') [ReagrupaObservadas];
```

```
proc(l::posint, X::list)
```

description *Reagrupa las frecuencias observadas siguiendo el mismo patrón que las teóricas (véase el procedimiento ReagrupaTeóricas). l es el número de clases teóricas una vez reagrupadas. X es la lista de las frecuencias observadas.*

...
end

El operador **Pearson** ha sido definido con la estructura de operador flecha y calcula el valor del estadístico de Pearson para unas frecuencias observadas (parámetro **Obs**) y unas frecuencias teóricas (parámetro **Teo**). **Obs** y **Teo** deben ser listas que contengan a las anteriores frecuencias:

```
> readlib('matac3') [Pearson];
```

$$(Obs::list, Teo::list) \rightarrow \sum_{k=1}^{nops(Obs)} \frac{(Obs_k - Teo_k)^2}{Teo_k}$$

El procedimiento **NewtonRaphson** aproxima la raíz de la ecuación $H(h)=0$ (obsérvese que la indeterminada debe ser llamada *h*). Sus parámetros formales son **Ecuacion** (la función *H* previamente definida como una expresión), **Derivada** (Derivada de *H* calculada mediante el comando **diff(H,h)**), **Semilla** (estimación por momentos del parámetro que queremos estimar por máxima verosimilitud), **Iteraciones** (número de veces que queremos iterar el método). Para una exposición detallada del método de Newton-Raphson, consúltese *Panjer y Willmot (1992)*.

```
> readlib('matac3') [NewtonRaphson];
```

```
proc(Ecuacion, Derivada, Semilla::numeric, Iteraciones::posint)
```

description *Aproxima la raíz de una ecuación $H(h)=0$ cuya indeterminada es *h*. Sus parámetros son Ecuacion (la función *H*), Derivada (la derivada de *H*), Semilla (aproximación inicial de la raíz de la ecuación), Iteraciones (nº de veces que queremos iterar el método).*

...
end

El procedimiento **EstimaPorMomentosPoinGau** nos da las estimaciones de los parámetros μ , β de la distribución Poisson ponderada por una inversa gaussiana. Sus parámetros son **m** (media muestral), **v** (varianza muestral).

```
> readlib('matac3') [EstimaPorMomentosPoinGau] ;
proc(m::numeric, v::numeric)
description `Calcula las estimaciones por momentos de los parámetros  $\mu$ ,  $\beta$  de la distribución Poisson Inversa Gaussiana.  $m$ ,  $v$  son la media y varianza muestrales.`
...
end
```

El procedimiento **CuantiaPoisInvGaus** calcula las probabilidades para el modelo correspondiente a la estimación por momentos. Sus parámetros son $g(\mu)$, $h(\beta)$, k (la cuantía que deseamos calcular):

```
> readlib('matac3') [CuantiaPoinGau] ;
proc(g::numeric, h::numeric, k::nonnegint)
description `Calcula la cuantía  $k$ -ésima de la distribución Poisson Inversa Gaussiana para ciertos valores de los correspondientes parámetros  $\mu$ ,  $\beta$ .  $g$ ,  $h$  son las estimaciones de estos dos parámetros.`
...
end
```

El procedimiento *recursivo* **GeneraTk**, genera los $T_k(\beta)$ (véase J.Lemaire (1996) pag. 37) necesarios para formar la ecuación cuya raíz es el estimador de máxima verosimilitud buscado. Sus parámetros son **media** (la media muestral), **k** (el subíndice de T_k).

```
> readlib('matac3') [GeneraTk] ;
proc(media::numeric, k::nonnegint)
description `En la estimación por máxima verosimilitud del parámetro  $\beta$  de la distribución Poisson Inversa Gaussiana, el EMV de  $\beta$  es la raíz de cierta ecuación (véase J. Lemaire (1995) p.37). Para generar dicha ecuación es necesario emplear este procedimiento que no suministra los  $T_k(\beta)$ .  $media$  es la media muestral.`
...
end
```

Al procedimiento que calcula la cuantía de una distribución compuesta lo denominamos **Panjer**. Sus parámetros son:

Primaria: El nombre al cual le asociamos la distribución primaria definida como un operador flecha.

Secundaria: lo mismo para la distribución secundaria.

x: el valor en el cual queremos calcular la cuantía de la distribución compuesta.

El procedimiento calcula los parámetros a , b que caracterizan a la primaria como miembro de la clase $a, b, 0$.

```
> readlib('matac3') [Panjer] ;
proc(Primaria::procedure, Secundaria::procedure, x::nonnegint)
description `Panjer aplica el algoritmo de Panjer (véase Panjer & Willmot (1992, corolario \
```

6.6.1)) para el cálculo de una distribución compuesta tal que la primaria sea Poisson, Binomial, Binomial Negativa, y la secundaria sea discreta aritmética. Ambas distribuciones (aquí llamadas Primaria, Secundaria) deben haber sido definidas previamente como procedimientos (por ejemplo operadores flecha). x es el valor correspondiente a la cuantía que queremos calcular. Los parámetros a, b que determinan a la primaria son calculados en el interior del procedimiento.

...

end

Consideremos la variable aleatoria tipificada del **Daño Total producido en una subcartera**

a lo largo de un periodo: $s = \frac{S - \mu}{\sigma}$.

La función (que depende del coeficiente de asimetría de la distribución compuesta), que a cada s le hace corresponder una y normal(0,1) es introducida en forma de operador flecha y la llamamos **Vinv**. Sus parámetros son s (el valor tipificado de la compuesta para el cual queremos calcular la probabilidad), **gamma** (la asimetría de la compuesta). El operador ha sido definido mediante el comando **piecewise** que permite definir cómodamente las funciones a trozos.

> readlib('matac3') [Vinv];

$(s, \gamma) \rightarrow \text{piecewise} \left(s < -\frac{1}{4} \sqrt{7} \sqrt{4}, s - \frac{1}{6} \gamma (s^2 - 1) + \frac{1}{36} \gamma^2 (4s^3 - 7s), \right.$

$\left. -\frac{1}{4} \sqrt{7} \sqrt{4} < s \text{ and } s < 1, s - \frac{1}{6} \gamma (s^2 - 1), \sqrt{1 + 9 \frac{1}{\gamma^2} + 6 \frac{s}{\gamma} - 3 \frac{1}{\gamma}} \right)$

>

Apéndice 2: Ajuste de la Binomial Negativa.

- + 1-Estimación de los parámetros r, β por el método de los momentos.**
- + 2-Reagrupamiento de las clases.**
- + 3-Cálculo del estadístico de Pearson y del p-valor.**
- + 4-Emv de los parámetros r, β .**
- + 5-Ajuste del nuevo modelo.**
- + 6-Bibliografía.**

Apéndice 2: Ajuste de la Binomial Negativa.

1-Estimación de los parámetros r, β por el método de los momentos.

Introducimos las frecuencias observadas asignándolas a la *lista FrecObs*:

```
> FrecObs := [20592, 2651, 297, 41, 7, 0, 1];  
FrecObs := [20592, 2651, 297, 41, 7, 0, 1]
```

Calculamos el nº total de asegurados mediante el comando **sum**.

```
> CasosTot := sum(FrecObs[i], i=1..nops(FrecObs));  
CasosTot := 23589
```

A continuación se calcula la media muestral.

```
> MediaMuestral := 1/CasosTot  
*sum(FrecObs[k] * (k-1), k=1..nops(FrecObs));
```

$$MediaMuestral := \frac{378}{2621}$$

La aproximación de este número racional es:

```
> evalf(MediaMuestral);  
.1442197634
```

Cálculo de la varianza muestral:

```
> VarianzaMuestral := 1/CasosTot * sum(FrecObs[k] *  
(k-1-MediaMuestral)^2, k=1..nops(FrecObs));
```

$$VarianzaMuestral := \frac{1125680}{6869641}$$

La aproximación de este número racional es:

```
> evalf(VarianzaMuestral);  
.1638630024
```

Almacenamos la media y varianza muestrales en la *lista Muestral*:

```
> Muestral := [MediaMuestral, VarianzaMuestral];  
Muestral := [  $\frac{378}{2621}$ ,  $\frac{1125680}{6869641}$  ]
```

El *procedimiento* **matac3[EstimaPorMomentosBiNeg]** nos da las estimaciones de los parámetros μ, β de la distribución. Sus parámetros son **m** (media muestral), **v** (varianza muestral).

Asignamos las dos estimaciones a la *lista EstimadoresMomentos*

```
> EstimadorMomento :=  
matac3[EstimaPorMomentosBiNeg](Muestral[1], Muestral[2]);
```

$$EstimadorMomento := \left[\frac{67471}{495369}, \frac{71442}{67471} \right]$$

```
> beta := EstimadorMomento[1]; r := EstimadorMomento[2];
```

$$\beta := \frac{67471}{495369}$$

$$r := \frac{71442}{67471}$$

Si aproximamos estos dos números racionales:

```
> evalf(beta);evalf(r);
.1362035170
1.058854915
```

2-Reagrupamiento de las clases.

El *procedimiento* **matac3[CuantiaBiNeg]** calcula las probabilidades para el modelo estimado. Sus parámetros son $x(\beta)$, $y(r)$.

Las probabilidades del modelo se asignan a la *lista* **ProBiNeg**:

```
> ProBiNeg := matac3[CuantiaBiNeg](beta, r);
ProBiNeg := [.87353438526831418204, .11087883510827038197,
.013682850918018587495, .0016724240778103594637, .00020343334510196325374,
.000024673829134486784714, .339745335003898 10-5]
> sum(ProBiNeg[j], j=1..nops(ProBiNeg));
1.000000000
```

El *procedimiento* **matac3[ReagrupaTeoricas]** reagrupa las clases teóricas de forma que en ninguna de las que resulten haya menos de 4 asegurados. Nos devuelve una lista con las frecuencias teóricas de las clases reagrupadas. Sus parámetros son: **Tot** (nº total de asegurados), **X** (la lista *ProBiNeg* que contiene las probabilidades de la distribución). La *lista* **ClasesTeoricas** almacena las clases teóricas reagrupadas:

```
> ClasesTeoricas := matac3[ReagrupaTeoricas](CasosTot,
ProBiNeg);
```

```
ClasesTeoricas :=
```

```
[20605.80261, 2615.520841, 322.7647704, 39.45081158, 5.460962660]
```

El nº total de clases es almacenado en la *variable* **Num**

```
> Num := nops(ClasesTeoricas);
```

```
Num := 5
```

El *procedimiento* **matac3[ReagrupaObservadas]** reagrupa las frecuencias observadas siguiendo el patrón de las teóricas. Sus parámetros son: **I** (nº de clases teóricas), **X** (lista de las frecuencias observadas)

Asignamos las frecuencias observadas a la *lista* **N**:

```
> ClasesObservadas :=
matac3[ReagrupaObservadas](Num, FrecObs);
ClasesObservadas := [20592, 2651, 297, 41, 8]
```

3-Cálculo del estadístico de Pearson y del p-valor.

El *operador* **matac3[Pearson]** calcula el valor del estadístico de Pearson. Su contenido puede visualizarse:

```
> readlib('matac3')[Pearson];
```

$$(Obs::list, Teo::list) \rightarrow \sum_{k=1}^{nops(Obs)} \frac{(Obs_k - Teo_k)^2}{Teo_k}$$

Sus parámetros son **Obs** (lista que contiene las clases observadas), **Teo** (lista que contiene las clases teóricas). Lo calculamos y el resultado es almacenado en **P**.

```
> P := matac3[Pearson](ClasesObservadas, ClasesTeoricas);
```

```
P := 3.788536724
```

Finalmente se calcula el p-valor para saber si podemos rechazar la hipótesis nula. Lo asignamos a **PvalorMomentos**. Los grados de libertad de la χ^2 son (n° de clases)-(n° de param. estimados)-1 = **Num-3**.

```
> PvalorMomentos := 1-
```

```
stats[statevalf, cdf, chisquare[Num-3]](P);
```

```
PvalorMomentos := .1504283539
```

Al haber obtenido un p-valor del 15% no podemos rechazar la hipótesis nula y aceptamos el modelo. Quizás lo podamos mejorar mediante una estimación de máxima verosimilitud. Es lo que hacemos en la siguiente sección.

```
> :
```

4-Emv de los parámetros r, β .

Ahora vamos a obtener aproximaciones de los emv de r, β . en primer lugar expresamos la ecuación cuya solución es el emv de r (véase *Panjer y Willmot (1992) p.310*). Lo hacemos en dos pasos. La variable a despejar es **h**.

1° paso:

```
> Sumatorio := [seq(sum((m+h)^(-1), m=0..k-1), k=1..nops(FrecObs))]:
```

2° paso:

```
> H := CasosTot*log(1+MediaMuestral/h) - sum(FrecObs[k]*Sumatorio[k-1], k=2..nops(FrecObs));
```

$$H := 23589 \ln\left(1 + \frac{378}{2621} \frac{1}{h}\right) - 2997 \frac{1}{h} - 346 \frac{1}{1+h} - 49 \frac{1}{2+h} - 8 \frac{1}{3+h} - \frac{1}{4+h} - \frac{1}{5+h}$$

La derivada de la anterior función es:

```
> DerivadaH := diff(H, h);
```

$$DerivadaH := -3402 \frac{1}{h^2 \left(1 + \frac{378}{2621} \frac{1}{h}\right)} + 2997 \frac{1}{h^2} + 346 \frac{1}{(1+h)^2} + 49 \frac{1}{(2+h)^2}$$

$$+ 8 \frac{1}{(3+h)^2} + \frac{1}{(4+h)^2} + \frac{1}{(5+h)^2}$$

```
> :
```

El *procedimiento* **matac3[NewtonRaphson]** aproxima la raíz de la ec. $H(h)=0$. Sus parámetros son **Ecuación** (la función H), **Derivada** ($DerivadaH$), **Semilla** (r , estimación por momentos del param. r), **Iteraciones** (n° de veces que queremos aplicar el método). Entonces el emv de r es aproximado por:

```
> Emvr := matac3[NewtonRaphson](H, DerivadaH, r, 5);
```

$Emvr := 1.117895395$

El emv de β se deduce fácilmente calculando:

> $EmvBeta := MediaMuestral/Emvr;$

$EmvBeta := .1290100702$

5-Ajuste del nuevo modelo.

Llevamos a cabo las mismas tareas que en el apartado 3.

Calculamos las probabilidades del nuevo modelo y las asignamos a la *lista* **ProBiNeg2**.

> $ProBiNeg2 := matac3[CuantiaBiNeg](EmvBeta, Emvr);$

$ProBiNeg2 := [.87315099769276053937, .11153632160650245328,$
 $.013496359065181862960, .0016028124446128479387, .00018854882077201092018,$
 $.000022053169589876799748, .290720058040873 \cdot 10^{-5}]$

> $sum(ProBiNeg2[k], k=1..nops(ProBiNeg2));$

1.000000000

Reagrupamos las clases teóricas asignadolas a la *lista* **ClasesTeoricas2**.

> $ClasesTeoricas2 := matac3[ReagrupaTeoricas](CasosTot,$
 $ProBiNeg2);$

$ClasesTeoricas2 :=$

$[20596.75888, 2631.030290, 318.3656141, 37.80874277, 5.036468306]$

El nº de clases reagrupadas es asignado a la *variable* **Num2**.

> $Num2 := nops(ClasesTeoricas2);$

$Num2 := 5$

Reagrupamos las clases teóricas asignadolas a la *lista* **ClasesObservadas2**.

> $ClasesObservadas2 := matac3[ReagrupaObservadas](Num2,$
 $FrecObs);$

$ClasesObservadas2 := [20592, 2651, 297, 41, 8]$

Calculamos el estadístico de Pearson que almacenamos en **Pemv** :

> $Pemv :=$

$matac3[Pearson](ClasesObservadas2, ClasesTeoricas2);$

$Pemv := 3.599668379$

Y por último calculamos el p-valor correspondiente:

> $PvalorEmv := 1-$

$stats[statevalf, cdf, chisquare[Num2-3]](Pemv);$

$PvalorEmv := .1653262987$

> :

6-Bibliografía.

Lemaire, J., 1995: Bonus-Malus Systems in Automobile Insurance, Kluwer Academic Press, Netherlands.

-Capítulo 3, pag. 21-43.

Panjer, H.H., Willmot, G.E., 1992: Insurance Risk Models, Society of Actuaries, USA.

-Capítulo 9, pag. 291-338.

Apéndice 3: Ajuste de la Poisson-Inversa Gaussiana.

- + 1- Estimación de los parámetros μ, β por el método de los momentos.**
- + 2-Reagrupamiento de las clases.**
- + 3-Cálculo del estadístico de Pearson y del p-valor.**
- + 4-Emv de los parámetros μ, β .**
- + 5-Ajuste del nuevo modelo.**
- + 6-Bibliografía.**

Apéndice 3: Ajuste de la Poisson-Inversa Gaussiana.

1- Estimación de los parámetros μ, β por el método de los momentos.

Introducimos las frecuencias observadas asignándolas a la *lista FrecObs*:

```
> FrecObs := [7840, 1317, 239, 42, 14, 4, 4, 1];
```

```
FrecObs := [7840, 1317, 239, 42, 14, 4, 4, 1]
```

Calculamos el número total de asegurados.

```
> CasosTot := sum(FrecObs[j], j=1..nops(FrecObs));
```

```
CasosTot := 9461
```

A continuación se calcula la media muestral.

```
> MediaMuestral := 1/CasosTot *  
sum(FrecObs[k] * (k-1), k=1..nops(FrecObs));
```

```
MediaMuestral :=  $\frac{2028}{9461}$ 
```

La aproximación de este número racional es:

```
> evalf(MediaMuestral);
```

```
.2143536624
```

Cálculo de la varianza muestral:

```
> VarianzaMuestral := 1/CasosTot * sum(FrecObs[k] *  
(k-1-MediaMuestral)^2, k=1..nops(FrecObs));
```

```
VarianzaMuestral :=  $\frac{25859664}{89510521}$ 
```

La aproximación de este número racional es:

```
> evalf(VarianzaMuestral);
```

```
.2889008321
```

Almacenamos la media y varianza muestrales en la *lista Muestral*:

```
> Muestral := [MediaMuestral, VarianzaMuestral];
```

```
Muestral :=  $\left[ \frac{2028}{9461}, \frac{25859664}{89510521} \right]$ 
```

El *procedimiento matac3[EstimaPorMomentosPoinGau]* nos da las estimaciones por momentos de los parámetros μ, β de la distribución. Sus parámetros son **m** (media muestral), **v** (varianza muestral).

Asignamos las dos estimaciones a la *lista EstimadorMomento*

```
> EstimadorMomento :=
```

```
matac3[EstimaPorMomentosPoinGau](Muestral[1], Muestral[2]);
```

```
EstimadorMomento :=  $\left[ \frac{2028}{9461}, \frac{556063}{1598909} \right]$ 
```

```
> mu := EstimadorMomento[1]; beta := EstimadorMomento[2];
```

$$\mu := \frac{2028}{9461}$$

$$\beta := \frac{556063}{1598909}$$

Si aproximamos estos dos números racionales obtenemos:

```
> evalf(beta);evalf(mu);
.3477765151
.2143536624
> :
```

2-Reagrupamiento de las clases.

El *procedimiento* **matac3[CuantiaPoisInvGaus]** calcula las probabilidades de la distribución Poisson-Inversa Gaussiana. Sus parámetros son **g** (μ), **h** (β) (estimaciones de los dos parámetros), **k** (la cuantía que deseamos calcular):

Vamos a calcular la probabilidad de la última clase (corresponde a $k \geq 4$); La asignamos al nombre **ColaDerecha**:

```
> Total:=0: for i from 0 to nops(FrecObs)-2 do Total:=
  Total+matac3[CuantiaPoinGau](mu,beta,i):od:Total;
.9999312326
```

```
> ColaDerecha:= 1-Total;
ColaDerecha := .0000687674
```

```
> Total+ColaDerecha;
1.000000000
```

Las probabilidades de la distribución obtenida se asignan a la *lista* **ProbPoinGau**:

```
> ProbPoinGau :=
  [seq(matac3[CuantiaPoinGau](mu,beta,k),k=0..nops(FrecObs)
  -2), ColaDerecha];
```

```
ProbPoinGau := [.83008953931325670151, .13664701856563111669,
.025261125822474273328, .0057984960913350627649, .0015437144638818344063,
.00045114248589075588400, .00014019583290424559270, .0000687674]
```

Estas probabilidades deben sumar 1:

```
> sum(ProbPoinGau[k],k=1..nops(ProbPoinGau));
1.000000000
```

```
> :
```

El *procedimiento* **matac3[ReagrupaTeoricas]** reagrupa las clases teóricas de forma que en ninguna de las que resulten haya menos de 4 asegurados. Nos devuelve una lista con las frecuencias teóricas de las clases reagrupadas. Sus parámetros son: **Tot** (nº total de asegurados), **X** (la lista que contiene las probabilidades de la distribución, en este caso **ProbPoinGau**). La *lista* **ClasesTeoricas** almacena las clases reagrupadas:

```
> ClasesTeoricas := matac3[ReagrupaTeoricas](CasosTot,
  ProbPoinGau);
```

```
ClasesTeoricas :=
[7853.477131, 1292.817443, 238.9955114, 54.85957152, 14.60508254, 6.245260205]
```

El número total de clases teóricas es almacenado en la *variable Num*

```
> Num := nops (ClasesTeóricas) ;
```

```
Num := 6
```

El *procedimiento matac3[ReagrupaObservadas]* reagrupa las frecuencias observadas siguiendo el patrón de las teóricas. Sus parámetros son: **1** (nº de clases teóricas), **X** (lista de las frecuencias observadas). Asignamos las frecuencias observadas a la *lista N* :

```
> ClasesObservadas :=
```

```
matac3 [ReagrupaObservadas] (Num, FrecObs) ;
```

```
ClasesObservadas := [ 7840, 1317, 239, 42, 14, 9 ]
```



3-Cálculo del estadístico de Pearson y del p-valor.

El *operador matac3[Pearson]* calcula el valor del estadístico de Pearson. Veamos en qué consiste:

```
> readlib ('matac3') [Pearson] ;
```

$$(Obs::list, Teo::list) \rightarrow \sum_{k=1}^{nops(Obs)} \frac{(Obs_k - Teo_k)^2}{Teo_k}$$

Por tanto sus parámetros son las dos listas: la de las clases observadas y la de las clases teóricas. Realizamos el cálculo y lo asignamos a la *variable P*.

```
> P := matac3 [Pearson] (ClasesObservadas, ClasesTeóricas) ;
```

```
P := 4.730032515
```

Finalmente se calcula el p-valor para saber si podemos rechazar la hipótesis nula. Lo asignamos a la *variable PvalorMomentos*. Los grados de libertad de la χ^2 son $(n^\circ \text{ de clases}) - (n^\circ \text{ de param. estimados}) - 1 = \mathbf{Num} - 3$.

```
> PvalorMomentos := 1 -
```

```
stats [statevalf, cdf, chisquare [Num-3]] (P) ;
```

```
PvalorMomentos := .1926669786
```

```
> :
```



4-Emv de los parámetros μ , β .

Ahora vamos a obtener aproximaciones de los emv de los parámetros μ , β .

```
>
```

El *procedimiento recursivo matac3[GeneraTk]*, genera los $T_k(\beta)$ (véase *J.Lemaire* pag. 37) necesarios para formar la ecuación cuya raíz es el estimador máxima verosimilitud del parámetro β . Sus parámetros son **media** (la media muestral), **k** (el subíndice de T_k).

Podemos generar la ecuación cuya raíz positiva es el EMV de β , llamándola **H**, de la siguiente forma:

```
> H:=0: for i from 1 to nops (FrecObs) do H :=
```

```
H+FrecObs [i] *matac3 [GeneraTk] (MediaMuestral, i-1) ;od:
```

```
H:=1/CasosTot *H-MediaMuestral:
```

La derivada de H la llamamos **DerivadaH**,

```
> DerivadaH:= diff (H, h) :
```

El *procedimiento matac3[NewtonRaphson]* aproxima la raíz de la ecuación $H(h)=0$.

Sus parámetros son **Ecuación** (la ecuación H), **Derivada** (DerivadaH), **Semilla** (h, estimación por momentos del param. β), **Iteraciones** (nº de veces que queremos iterar el método). Almacenamos la aproximación del emv de β en **EmvBeta**.: (5 iteraciones)
 > **EmvBeta := matac3[NewtonRaphson](H, DerivadaH, beta, 5);**

EmvBeta := .3285884725

El emv de μ es la media muestral:

> **EmvMu := evalf(MediaMuestral);**

EmvMu := .2143536624



5-Ajuste del nuevo modelo.

Llevamos a cabo las mismas tareas que en el apartado 3.

Calculamos la probabilidad de la última clase ($k \geq 5$) y la asignamos a **ColaDerecha**.

> **Total:=0: for i from 0 to nops(FrecObs)-2 do Total:=
 Total+matac3[CuantiaPoinGau](EmvMu, EmvBeta, i):od:Total;**
.9999422996

> **ColaDerecha:= 1-Total;**

ColaDerecha := .0000577004

Las probabilidades del nuevo modelo son almacenadas en la lista **ProbPoinGau2**

> **ProbPoinGau2 :=
 [seq(matac3[CuantiaPoinGau](EmvMu, EmvBeta, k), k=0..nops(FrecObs)-2), ColaDerecha];**

*ProbPoinGau2 := [.82908856946492302653, .13805349138533028566,
 .025180571998245297245, .0056308094797466356519, .0014537911657891103363,
 .00041137110119619998924, .00012369488076565069826, .0000577004]*

Reagrupamos las clases teóricas asignadolas a la lista **ClasesTeoricas2**.

> **ClasesTeoricas2 := matac3[ReagrupaTeoricas](CasosTot,
 ProbPoinGau2);**

ClasesTeoricas2 :=

[7844.006956, 1306.124082, 238.2333917, 53.27308849, 13.75431822, 5.608162739]

El número de las clases reagrupadas es asignado a **Num2**.

> **Num2 := nops(ClasesTeoricas2);**

Num2 := 6

Reagrupamos las clases observadas siguiendo el mismo esquema que para las teóricas:

> **ClasesObservadas2 := matac3[ReagrupaObservadas](Num2,
 FrecObs);**

ClasesObservadas2 := [7840, 1317, 239, 42, 14, 9]

Calculamos el estadístico de Pearson (lo llamamos **Pemv**)

> **Pemv := matac3[Pearson](ClasesObservadas2,
 ClasesTeoricas2);**

Pemv := 4.536351926

Y finalmente calculamos el p-valor:

> **PvalorEmv := 1-
 stats[statevalf, cdf, chisquare][Num2-3](Pemv);**

> :



6-Bibliografía.

Lemaire, J., 1995: Bonus-Malus Systems in Automobile Insurance, Kluwer Academic Press, Netherlands.







-Capítulo 3, pag. 21-43.

Panjer, H.H., Willmot, G.E., 1992: Insurance Risk Models, Society of Actuaries, USA.

-Capítulo 9, pag. 291-338.

Apéndice 4:

Algoritmo Recursivo de Panjer (Distribución Secundaria Discreta y Aritmética).

-  **Algoritmo recursivo de Panjer.**
-  **Caso de una secundaria con cuantía nula en el origen.**
-  **Caso de una secundaria con cuantía > 0 en el origen.**
-  **Cálculo de las cuantías de una distribución Neyman Tipo A.**
-  **Cálculo de las cuantías de una distribución del daño total.**
-  **Bibliografía.**

Apéndice 4:

Algoritmo Recursivo de Panjer (Distribución Secundaria Discreta y Aritmética).

Algoritmo recursivo de Panjer.

A lo largo de esta hoja vamos a utilizar el procedimiento **Panjer** que aplica el algoritmo recursivo de Panjer para el cálculo de una distribución compuesta. Este procedimiento se encuentra dentro del paquete **matac3**.

Si queremos leer la descripción del procedimiento, solo tenemos que ejecutar lo siguiente:

```
> readlib('matac3')[Panjer];
```

```
proc(Primaria::procedure, Secundaria::procedure, x::nonnegint)
```

```
description `Panjer aplica el algoritmo de Panjer (véase Panjer & Willmot (1992, corolar\
io 6.6.1)) para el cálculo de una distribución compuesta tal que la primaria sea Poisson, \
Binomial, Binomial Negativa, y la secundaria sea discreta aritmética. Ambas distribucion\
es (aquí llamadas Primaria, Secundaria) deben haber sido definidas previamente como p\
rocedimientos (por ejemplo operadores flecha). x es el valor correspondiente a la cuantía \
que queremos calcular. Los parámetros a, b que determinan a la primaria son calculados \
en el interior del procedimiento.`
```

...

```
end
```

En definitiva, el procedimiento permite calcular probabilidades de distribuciones compuestas tales que:

- 1) La **primaria** es de la clase $C(a, b, \theta)$.
- 2) La **secundaria** es **discreta y aritmética tal que $v(0)=0$** .

Sus parámetros formales son:

Primaria: El nombre al cual le pasamos la distribución primaria definida como un operador.

Secundaria: lo mismo para la distribución secundaria.

x: el valor en el cual queremos calcular la cuantía de la distribución compuesta.

El procedimiento calcula los parámetros a, b que caracterizan a la primaria.

Todos los calculos se llevarán a cabo con 20 dígitos decimales significativos, que se presumen exactos:

```
> Digits:=20;
```

Digits := 20

En primer lugar definimos una secundaria cualquiera sin átomo en el origen:

```
> v:=x->piecewise(x=1,1/8,x=2,1/8,x=3,3/8,x=4,2/8,x=5,1/8);
```

$$v := x \rightarrow \text{piecewise}\left(x = 1, \frac{1}{8}, x = 2, \frac{1}{8}, x = 3, \frac{3}{8}, x = 4, \frac{1}{4}, x = 5, \frac{1}{8}\right)$$

```
> v(0),v(1),v(1/2),v(-2),v(2),v(3/2),v(1000);
```

$$0, \frac{1}{8}, 0, 0, \frac{1}{8}, 0, 0$$

A continuación definimos una distribución de Poisson $P(\lambda)$, que llamamos P :

```
> P:=(lambda,n)->lambda^n/n!*exp(-lambda);
```

$$P := (\lambda, n) \rightarrow \frac{\lambda^n e^{(-\lambda)}}{n!}$$

```
> :
```



Caso de una secundaria con cuantía nula en el origen.

Algunos calculos sencillos. Definimos la $P(0.2)$ como operador:

```
> P0.2:=n->P(0.2,n);
```

$$P02 := n \rightarrow P(.2, n)$$

```
> P0.2(1);
```

$$.16374615061559637173$$

Las cuantías de 0 a 10 de la compuesta cuya secundaria es la v definida en el apartado anterior son:

```
> seq(matac3[Panjer](P0.2,v,i),i=0..10);
```

```
.81873075307798185867, .020468268826949546467, .020724122187286415798,  
.061918645312858518639, .042733920835961450862, .023052441135763809477,  
.0038913433844018851570, .0036913260236388671071, .0027200330508321910088,  
.0012311497467323461716, .00046571525703795165198
```

Comprobamos que las cuantías suman la unidad:

```
> total:=0:for i from 0 to 53 do total:=total +  
matac3[Panjer](P0.2,v,i):od:total;
```

$$1.00000000000000000000$$

Ahora repetimos los calculos con una distribución primaria $P(0.7)$:

```
> P0.7:=n->P(0.7,n);
```

$$P07 := n \rightarrow P(.7, n)$$

La cuantía en $x=0$ de la distribución compuesta es:

```
> matac3[Panjer](P0.7,v,0);
```

$$.49658530379140951470$$

Que efectivamente coincide con la cuantía de la primaria en el origen, $P0.7(0)$.

```
> P0.7(0);
```

$$.49658530379140951470$$

Las 11 primeras cuantías de la compuesta son:

```
> seq(matac3[Panjer](P0.7,v,i),i=0..10);
```

.49658530379140951470, .043451214081748332536, .045352204697824822085,
 .13421106903703354099, .10037691202655581272, .063131339669861409227,
 .029922936356944387502, .029500342496513274485, .023314363117902160273,
 .013208190652369013527, .0077033266499144808176

De nuevo comprobamos que suman 1:

```
> total:=0: for i from 0 to 70 do
  total:=total+matac3[Panjer](P0.7,v,i):od:total;
.9999999999999999997
```

Seguidamente vamos a hacer los cálculos para una compuesta cuya primaria sea una BN($r=0.5$, $\beta=0.02$) que llamaremos **BN1**. Pero primero definimos una binomial negativa genérica llamada **BN**:

```
> BN:=(r,beta,n)->GAMMA(n+r)/(GAMMA(r)*n!)*(1+beta)^(-r)*(beta/(1+beta))^n;
```

$$BN := (r, \beta, n) \rightarrow \frac{\Gamma(n+r)(1+\beta)^{-r} \left(\frac{\beta}{1+\beta}\right)^n}{\Gamma(r)n!}$$

Sustituyendo los parámetros por los anteriores valores obtenemos la BN1:

```
> BN1:=n->BN(0.5,0.02,n);
BN1:=n->BN(.5,.02,n)
```

La cuantía en $x=0$ de la distribución compuesta es: (utilizamos la misma distribución secundaria v)

```
> matac3[Panjer](BN1,v,0);
.99014754297667430915
```

Valor que efectivamente coincide con $BN1(0)$, la cuantía en el origen de la primaria.

```
> BN1(0);
.99014754297667430915
```

Las cuantías de la compuesta desde 0 hasta 10 son

```
> seq(matac3[Panjer](BN1,v,i),i=0..10);
.99014754297667430915, .0012134161065890616534, .0012156466509026444873,
.0036447139642446643187, .0024424597006946565305, .0012357762590322561381,
.000033572236938617864137, .000031460438279854667046,
.000022648296326127378024, .93617715242638004840 10-5,
.26716888250764101421 10-5
```

Y efectivamente suman la unidad:

```
> total:=0: for i from 0 to 46 do total:=total+
  matac3[Panjer](BN1,v,i):od:total;
1.0000000000000000000
```

A continuación usaremos una geométrica de parámetro $\beta = \frac{1}{3}$ que denominaremos

Geo1:

```
> Geo:=(beta,n)->1/(1+beta)*(beta/(1+beta))^n;
```

$$Geo := (\beta, n) \rightarrow \frac{\left(\frac{\beta}{1+\beta}\right)^n}{1+\beta}$$

> **Geo1** := n->Geo(1/3, n) ;

$$Geo1 := n \rightarrow Geo\left(\frac{1}{3}, n\right)$$

Las cinco primeras cuantías de la compuesta son

> **seq(matac3[Panjer](Geo1, v, i), i=0..4) ;**

$\frac{3}{4}$, .02343750000000000000, .02416992187500000000, .071800231933593750000.

.052071332931518554688

y suman la unidad:

> **total:=0: for i from 0 to 80 do**

total:=total+matac3[Panjer](Geo1, v, i):od: total;

.999999999999999748123

> :



Caso de una secundaria con cuantía > 0 en el origen.

Las mismas hipótesis que en el apartado anterior salvo que esta vez tomamos una secundaria con $v(0) > 0$. En este caso podemos aplicar el mismo algoritmo pero rectificando el modelo de la siguiente forma:

1) El parámetro de la primaria $\theta = \lambda, \beta, q$ (correspondiente a una Poisson, Binomial negativa-incluyendo la geométrica-, o binomial), se convierte en $(1-v(0))*\theta$.

2) Se toma como secundaria la de la v.a. **condicionada a realizaciones estrictamente positivas: $X/X > 0$**

Consideremos pues una secundaria con un átomo en el origen

> **v:= x->piecewise(x=0, 1/4, x=1, 1/4, x=2, 1/4, x=3, 1/4) ;**

$$v := x \rightarrow \text{piecewise}\left(x = 0, \frac{1}{4}, x = 1, \frac{1}{4}, x = 2, \frac{1}{4}, x = 3, \frac{1}{4}\right)$$

Obsérvese que la anterior secundaria tiene un átomo 1/4 en x=0. Tomemos una P(0.3):

> **P0.3:= n->P(0.3, n) ;**

$$P03 := n \rightarrow P(.3, n)$$

Rectificamos el parámetro λ según lo dicho más arriba. A la primaria rectificada la llamamos **Prec**.

> **Prec:= n->P(0.3*3/4, n) ;**

$$Prec := n \rightarrow P(.22500000000000000000, n)$$

Podemos comparar cuantías de la primaria y la misma rectificada

> **P0.3(1) ; Prec(1) ;**

.22224546620451535982

.17966614922085983464

Hacemos la rectificación de la secundaria, obteniendo la distribución de la variable condicionada $X/X > 0$: la llamamos **vrec**.

> **vrec:= x->**


```
> seq(matac3[Panjer](Primrec, Secrec, i), i=0..10);
.90421968001858424420, .062863108629467966497, .024187270733036359175,
.0067140878193034136918, .0015770681413802357876, .00034541907211973877472,
.000073836381122621071471, .000015529242515326163549,
.31987936244417020019 10-5, .64391818263778561391 10-6,
.12691863099785108120 10-6
```

Es posible que necesitemos utilizar esta distribución frecuentemente. Para que el código que calcula sus cuantías sea más fácil de recordar y más corto de escribir, podemos empaquetar los anteriores cálculos dentro de un operador que a cada n le asocia la cuantía de la Neyman tipo A. A dicho operador lo llamamos **NeymanTA** :

```
> NeymanTA := n->matac3[Panjer](Primrec, Secrec, n);
```

$$\text{NeymanTA} := n \rightarrow \text{matac3}_{\text{Panjer}}(\text{Primrec}, \text{Secrec}, n)$$

Por ejemplo, para calcular las 6 primeras cuantías debemos ejecutar lo siguiente:

```
> seq(NeymanTA(n), n=0..5);
.90421968001858424420, .062863108629467966497, .024187270733036359175,
.0067140878193034136918, .0015770681413802357876, .00034541907211973877472
```

Comprobemos que las cuantías suman la unidad.

```
> total:=0: for i from 0 to 30 do
  total:=total+NeymanTA(i):od:total;
.99999999999999999999
```

```
> :
```

Cálculo de las cuantías de una distribución del daño total.

Finalmente, vamos a ver mediante un ejemplo como sería el proceso completo de cálculo de las cuantías de la distribución del daño total. Supondremos que la distribución del número de siniestros es la anterior Neyman Tipo A y que la distribución de las cuantías individuales de los siniestros es la siguiente f :

```
> f := x->piecewise(x=1, 1/3, x=2, 1/3, x=3, 1/3);
```

$$f := x \rightarrow \text{piecewise}\left(x = 1, \frac{1}{3}, x = 2, \frac{1}{3}, x = 3, \frac{1}{3}\right)$$

Las *cuantías de la distribución de la siniestralidad total* se calcularían entonces de la siguiente forma:

1º paso: calculo de la $P(0.7)$ compuesta con f :

Como la secundaria no acumula probabilidad en 0, no hay que rectificar las distribuciones. Por tanto, llamandola **Paso1** y expresándola como un operador que a cada n le asocia la cuantía en n :

```
> Paso1 := n->matac3[Panjer](P0.7, f, n);
```

$$\text{Paso1} := n \rightarrow \text{matac3}_{\text{Panjer}}(P0.7, f, n)$$

```
> seq(Paso1(n), n=0..10);
.49658530379140951470, .11586990421799555343, .12938805971009503466,
.14395762729602447555, .043770035129928403635, .033592975213105581406,
```

```
.021505786064040281708, .0073333947293345551809, .0044077801978213459988,
.0021671981294084411496, .00076960199663794530276
> total:=0:for i from 0 to 50 do total:=total+Pasol(i):
  od:total;
1.00000000000000000000
```

2º paso: cálculo de la distribución P(0.2) compuesta con Paso1:

La distribución Paso1 sí que tiene un átomo en el origen. Su valor es :

```
> atomo:=Pasol(0);
atomo := .49658530379140951470
```

Por tanto debemos rectificar el parámetro de la P(0.2) (multiplicándolo por 1-atomo). A la nueva primaria la llamamos **PoiRec**

```
> parametro:=0.2*(1-atomo);
parametro := .10068293924171809706
```

```
> PoiRec:= n->P(parametro,n);
PoiRec := n → P(parametro, n)
```

```
> PoiRec(1);
.091039495104476896703
```

También debemos modificar la secundaria **Pasol**, tomando la condicionada a valores de $n > 0$:

```
> PasolRec:= n->(1-atomo)^(-1)*Pasol(n);
PasolRec := n →  $\frac{Pasol(n)}{1 - atomo}$ 
```

y recordando que esta variable n solo se puede tomar > 0 y entera. Finalmente calculamos la distribución compuesta de la variable aleatoria daño total. La llamaremos **DanoTot**.

```
> DanoTot:= n->matac3[Panjer](PoiRec,PasolRec,n);
DanoTot := n → matac3Panjer(PoiRec, PasolRec, n)
```

```
> seq(DanoTot(n), n=0..10);
.90421968001858424420, .020954369543155988833, .023641844069048917630,
.026577988514175306193, .0088279033124332426760, .0069462699530659683251,
.0046300724038565572080, .0018254761207574797231, .0011607608180448074287,
.00062942302344305050225, .00027687533653167697270
```

Comprobemos que las cuantías de la compuesta suman la unidad:

```
> total:=0:for i from 0 to 70 do total:=total+DanoTot(i):
  od: total;
1.00000000000000000000
```

Bibliografía.


Panjer H.H., Willmot G.E., 1992: Insurance Risk Models, Society of Actuaries, USA.
- p. 197, Corolario 6.6.1, p. 218, epígrafe 6.13, p. 257. epígrafe 7.4.

Apéndice 5: La Aproximación Normal-Power.

La Aproximación Normal-Power.

Cálculos.

 Aproximación de las probabilidades de la distribución compuesta.

 Sensibilidad de la probabilidad de ruina.

Bibliografía.

Apéndice 5: La Aproximación Normal-Power.

La Aproximación Normal-Power.

Consideremos la variable aleatoria tipificada del *Daño Total en un periodo de tiempo fijo*:

$$s = \frac{S - \mu}{\sigma}$$

El cambio de variable o transformación Normal-Power (NP) depende del coeficiente de asimetría de la distribución compuesta, y a cada s le hace corresponder una y distribuida normal(0,1). Siguiendo la notación de **Beard, Pentikäinen** (véase la bibliografía) dicha transformación ha sido introducida en forma de operador llamado **Vinv** (inversa de V):

> `readlib('matac3') [Vinv];`

$$(s, \gamma) \rightarrow \text{piecewise} \left(s < -\frac{1}{4} \sqrt{7} \sqrt{4}, s - \frac{1}{6} \gamma (s^2 - 1) + \frac{1}{36} \gamma^2 (4s^3 - 7s), \right.$$

$$\left. -\frac{1}{4} \sqrt{7} \sqrt{4} < s \text{ and } s < 1, s - \frac{1}{6} \gamma (s^2 - 1), \sqrt{1 + 9 \frac{1}{\gamma^2} + 6 \frac{s}{\gamma} - 3 \frac{1}{\gamma}} \right)$$

Si deseáramos trabajar con el cambio correspondiente a $\gamma = .2$ particularizamos γ en dicho valor y almacenamos el nuevo operador en el nombre **Vinv1**:

> `Vinv1:=matac3[Vinv](s,0.2);`

`Vinv1 := {`

$$.9922222222 s - .03333333333 s^2 + .03333333333 + .004444444444 s^3, s < -\frac{1}{4} \sqrt{7} \sqrt{4}$$

$$s - .03333333333 s^2 + .03333333333, -\frac{1}{4} \sqrt{7} \sqrt{4} - s < 0 \text{ and } s - 1 < 0$$

$$\sqrt{226.0000000 + 30.00000000 s - 15.00000000}, \text{ otherwise}$$

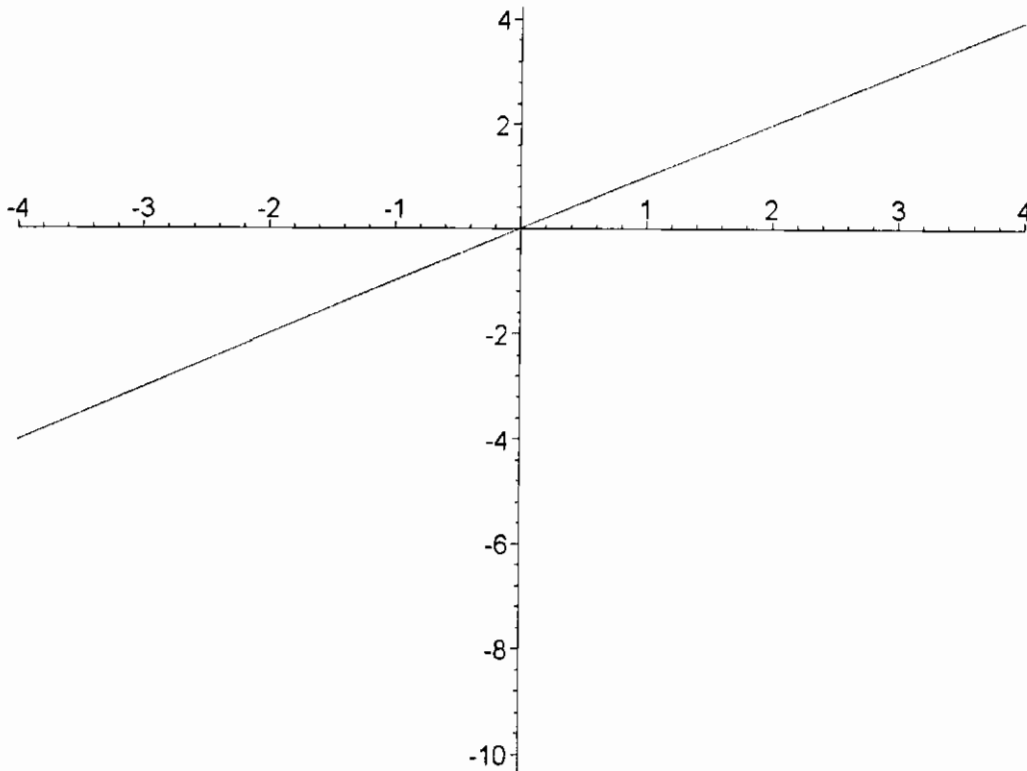
¿Como cambia **Vinv**(s, γ) a medida que la asimetría de la distribución compuesta aumenta?

Para contestar a esta pregunta realizamos una animación de la gráfica del operador **Vinv**.

Vamos a dibujar sucesivas gráficas de **Vinv** correspondientes a distintos valores de γ , para crear un efecto de dibujo animado. Para animar la gráfica solo tenemos que situar el puntero sobre ella, hacer clic una vez con el ratón, y aparecerán los mandos de animación :

> `plots[animate](matac3[Vinv], -4..4, 0.01..0.8, colour=blue, title = `ANIMACIÓN DE LA APROXIMACIÓN NP`);`

ANIMACION DE LA APROXIMACION NP



Si queremos una representación estática, podemos definir varias transformaciones para distintos coeficientes de asimetría y representarlas simultáneamente:

> `Vinv2:=matac3[Vinv](s,1);`

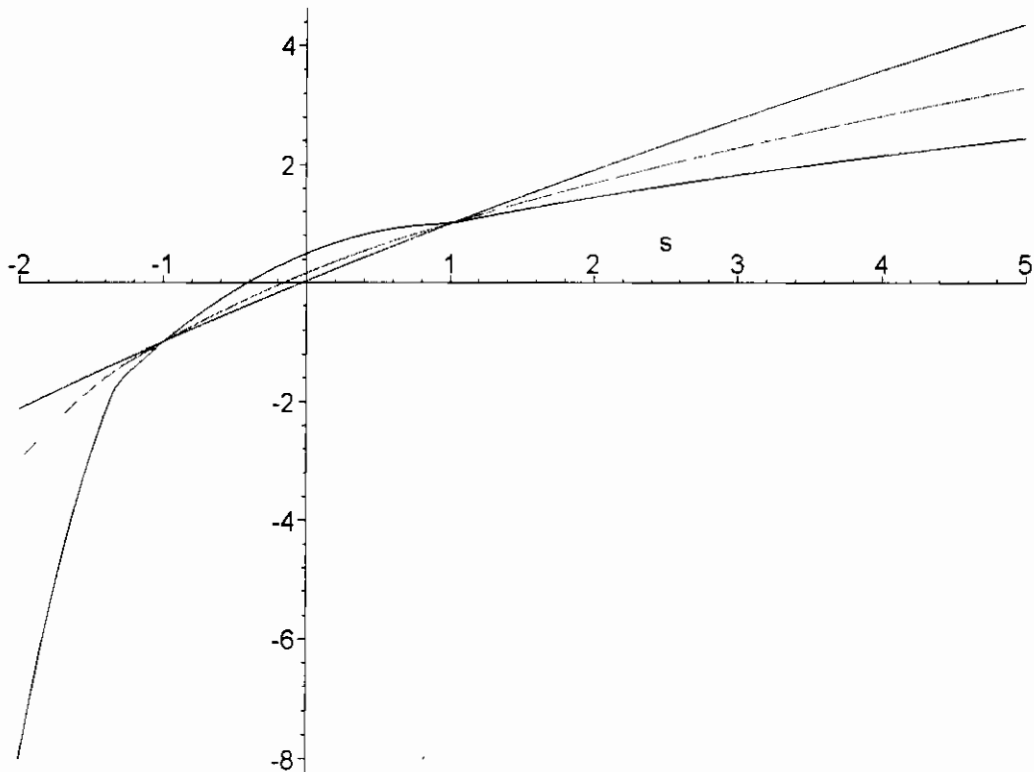
$$Vinv2 := \begin{cases} \frac{29}{36}s - \frac{1}{6}s^2 + \frac{1}{6} + \frac{1}{9}s^3 & s < -\frac{1}{4}\sqrt{7}\sqrt{4} \\ s - \frac{1}{6}s^2 + \frac{1}{6} & -\frac{1}{4}\sqrt{7}\sqrt{4} - s < 0 \text{ and } s - 1 < 0 \\ \sqrt{10 + 6s} - 3 & \text{otherwise} \end{cases}$$

> `Vinv3:=matac3[Vinv](s,3);`

$$Vinv3 := \begin{cases} -\frac{3}{4}s - \frac{1}{2}s^2 + \frac{1}{2} + s^3 & s < -\frac{1}{4}\sqrt{7}\sqrt{4} \\ s - \frac{1}{2}s^2 + \frac{1}{2} & -\frac{1}{4}\sqrt{7}\sqrt{4} - s < 0 \text{ and } s - 1 < 0 \\ \sqrt{2 + 2s} - 1 & \text{otherwise} \end{cases}$$

> `plot([Vinv1,Vinv2,Vinv3],s=-2..5,colour=[blue,red,black],title='APROXIMACIONES NP PARA ASIMETRÍAS = 0.2, 1, 3');`

APROXIMACIONES NP PARA ASIMETRIAS = 0.2, 1, 3

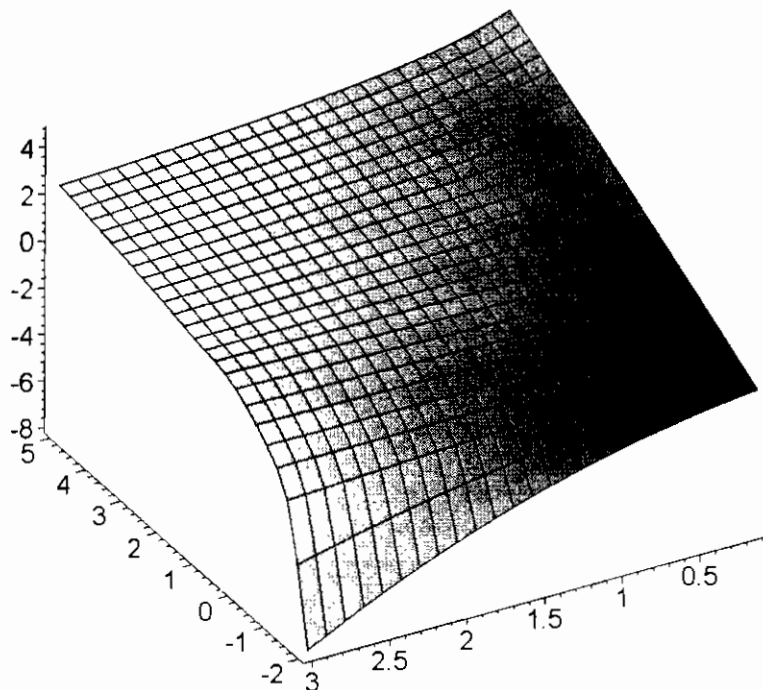


Una tercera posibilidad consiste en dibujar la gráfica de la función de dos variables (una superficie de R^3) **Vinv**.

Debemos definir con cuidado los rangos de las variables independientes ya que la función **Vinv** solo tiene sentido para valores de $\gamma > 0$: por ejemplo, s varía entre -2 y 5, la asimetría entre 0.1 y 3:

```
> plot3d(matac3[Vinv], -2..5, 0.1..3, title=`LA NP COMO FUNCIÓN DE 2 VARIABLES`);
```

LA NP COMO FUNCION DE 2 VARIABLES



> :

[-] Cálculos.

[-] Aproximación de las probabilidades de la distribución compuesta.

Supongamos que deseamos aproximar la probabilidad del suceso $\{S \leq 10\}$.

Supongamos también que la distribución de S tiene las siguientes características:

$$\mu = 7, \sigma = 5, \gamma = 1.5$$

La variable del daño total tipificada para S=10 es entonces :

$$> \text{DanoTip} := (10 - 7) / 5;$$

$$\text{DanoTip} := \frac{3}{5}$$

El valor correspondiente y distribuido $N(0,1)$ se obtiene así:

$$> \text{ValorNorm} := \text{matac3}[\text{Vinv}] (\text{DanoTip}, 1.5);$$

$$\text{ValorNorm} := .7600000000$$

Finalmente calculamos la probabilidad de la normal:

$$> \text{stats}[\text{statevalf}, \text{cdf}, \text{normald}] (\text{ValorNorm});$$

$$.7763727075$$

> :

[-] Sensibilidad de la probabilidad de ruina.

Llamemos R a las reservas, U a las reservas iniciales, θ al recargo de seguridad, μ a la cuantía media del siniestro, F a la distribución del daño total.

Recordemos que el modelo en tiempo discreto para el nivel de reservas a través de un periodo es

$$R = U + c - S, \text{ siendo } c = (1 + \theta)\mu$$

Se tiene entonces que $P\{R < 0\} = P\{S > U + c\} = 1 - F(U + c)$. Si queremos aproximar esta probabilidad, utilizaremos la aproximación NP de la siguiente forma. (tomaremos la media y la desviación típica del último ejemplo).

$$> \text{DanoTip} := (U + (1 + \theta) * 7 - 7) / 5;$$

$$\text{DanoTip} := \frac{1}{5} U + \frac{7}{5} \theta$$

El cambio **Vinv** depende ahora de tres parámetros: $x=U$ (reservas), $y=\theta$ (recargo de seguridad), γ (la asimetría). Al nuevo operador lo denominamos **Vruina**:

$$> \text{Vruina} :=$$

$$(x, y, \gamma) \rightarrow \text{evalf}(\text{matac3}[\text{Vinv}] (\text{subs}(U=x, \theta=y, \text{DanoTip}), \gamma));$$

$$\text{Vruina} := (x, y, \gamma) \rightarrow \text{evalf}(\text{matac3}_{\text{Vinv}}(\text{subs}(U=x, \theta=y, \text{DanoTip}), \gamma))$$

Por ejemplo, el valor distribuido $N(0,1)$ correspondiente a los datos $U=10, \theta=0.5, \gamma=1.5$ queda almacenado en **ValorNorm**

$$> \text{ValorNorm} := \text{Vruina}(10, 0.5, 1.5);$$

$$\text{ValorNorm} := 1.974921383$$

Y la probabilidad de la cola correspondiente es

$$> \text{ProbRuina} := 1 - \text{stats}[\text{statevalf}, \text{cdf}, \text{normald}] (\text{ValorNorm});$$

ProbRuina := .0241385352

Para abreviar el cálculo de la probabilidad de ruina aproximada, podemos juntar estos cálculos dentro de un procedimiento llamado **ProbRuina**: cuando le pasamos los parámetros $x=U$ (reservas), $y=\theta$ (recargo de seguridad), $z=\gamma$ (la asimetría), nos devuelve dicha probabilidad:

```
> ProbRuina:= proc(x,y,z)#x es la reserva inicial, y es
  el recargo de seguridad, z es la asimetría.
> local ValorNormal;
> ValorNormal:= Vruina(x,y,z);
> 1-stats[statevalf,cdf,normald](ValorNormal);
> end;
```

ProbRuina := **proc**(*x*, *y*, *z*)

local *ValorNormal*;

ValorNormal := *Vruina*(*x*, *y*, *z*);

1 - stats[*statevalf*, *cdf*, *normald*](*ValorNormal*)

end

Veamos su funcionamiento. Con reservas iniciales 10, recargo de seguridad 50%, y asimetría 1.5:

```
> ProbRuina(10,0.5,1.5);
```

.0241385352

Mismos datos salvo la asimetría que ahora vale 2.5:

```
> ProbRuina(10,0.5,2.5);
```

.0369980964

Ya estamos en condiciones de estudiar la sensibilidad de la probabilidad (aproximada) de ruina a los tres parámetros U , θ , γ . Recordemos que los dos primeros son variables de decisión, mientras que z es una cualidad del riesgo, su asimetría. Fijemos esta última en

$$\gamma = 1.5$$

Para este caso particular creamos el operador **Ruina1.5** que solo depende x , y , variables de decisión.

```
> Ruina1.5:=(x,y)->ProbRuina(x,y,1.5);
```

Ruina15 := (*x*, *y*) → ProbRuina(*x*, *y*, 1.5)

Por ejemplo, el caso ya calculado unas líneas más arriba:

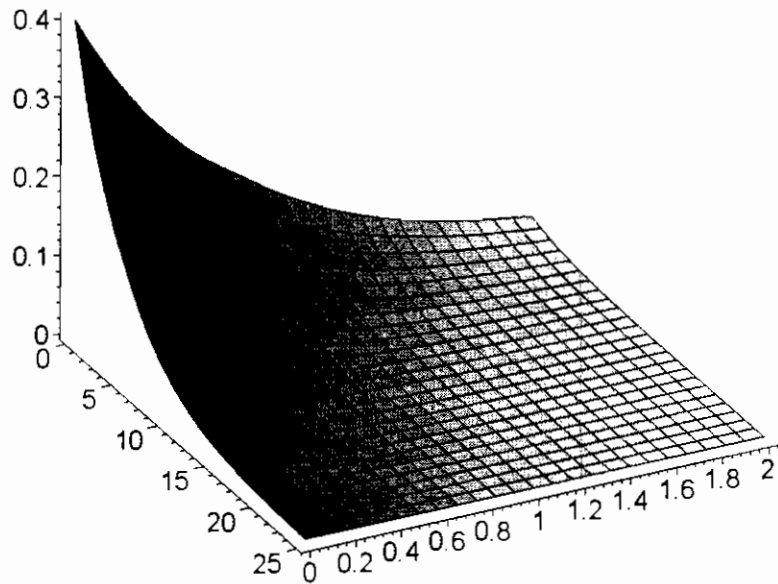
```
> Ruina1.5(10,0.5);
```

.0241385352

Podemos dibujar la gráfica de la función anterior para x con rango 0 a 25, y con rango 0 a 200%:

```
> plot3d(Ruina1.5,0..25,0..2,title=`GRÁFICA DE LA
  FUNCIÓN RUINA1.5`);
```

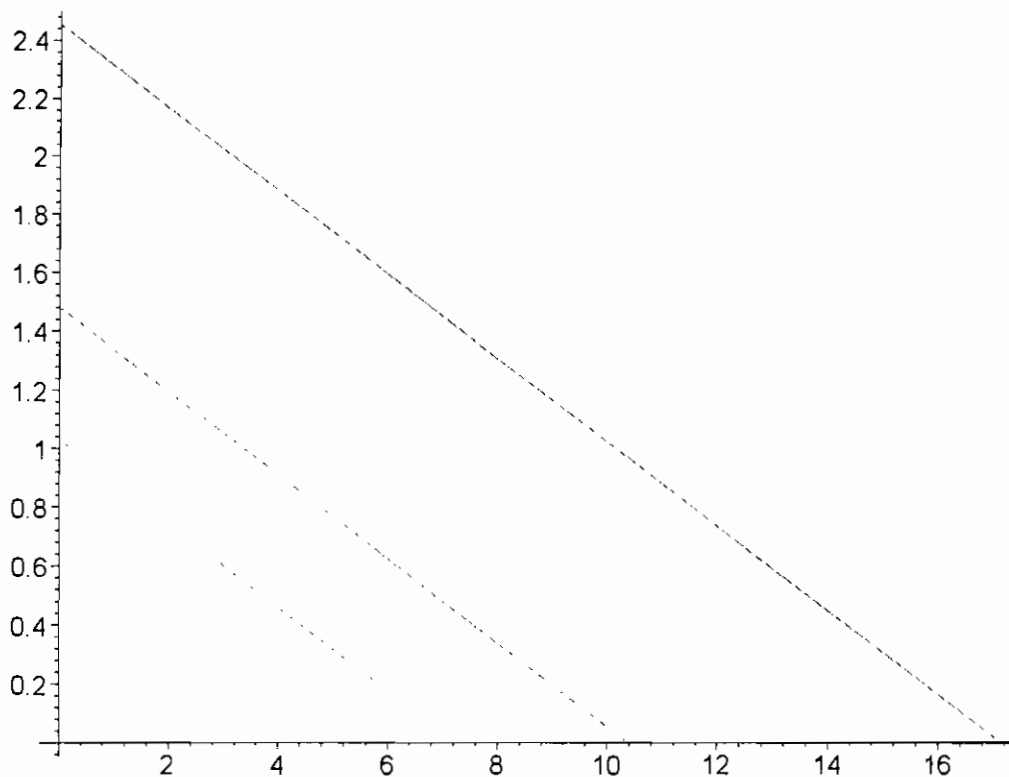

GRAFICA DE LA FUNCION RUINA1.5



Y también podemos dibujar las curvas de nivel que nos interesan. Por ejemplo para las probabilidades de ruina 0.01, 0.05, 0.1, 0.15 tenemos las siguientes combinaciones de reservas iniciales y recargos de seguridad (al parámetro **contours** le asignamos los niveles que deseamos dibujar; se los damos en forma de *lista*) (basta señalar con el ratón un punto de la curva de nivel y el programa nos devuelve sus coordenadas)

```
> plots[contourplot](Ruina1.5,0..25,0..3,  
  contours=[0.01,0.05,0.1,0.15],title=`CURVAS DE NIVEL  
  DE Ruina1.5`);
```

CURVAS DE NIVEL DE Ruina1.5



Si conocemos nuestras reservas iniciales podemos definir la probabilidad de ruina como función del recargo de seguridad y representarla gráficamente: fijemos pues $U=6$. Al nuevo operador que aproxima la probabilidad de ruina para U fija, $y (= \theta)$ variable lo llamamos **RuinaRes**:

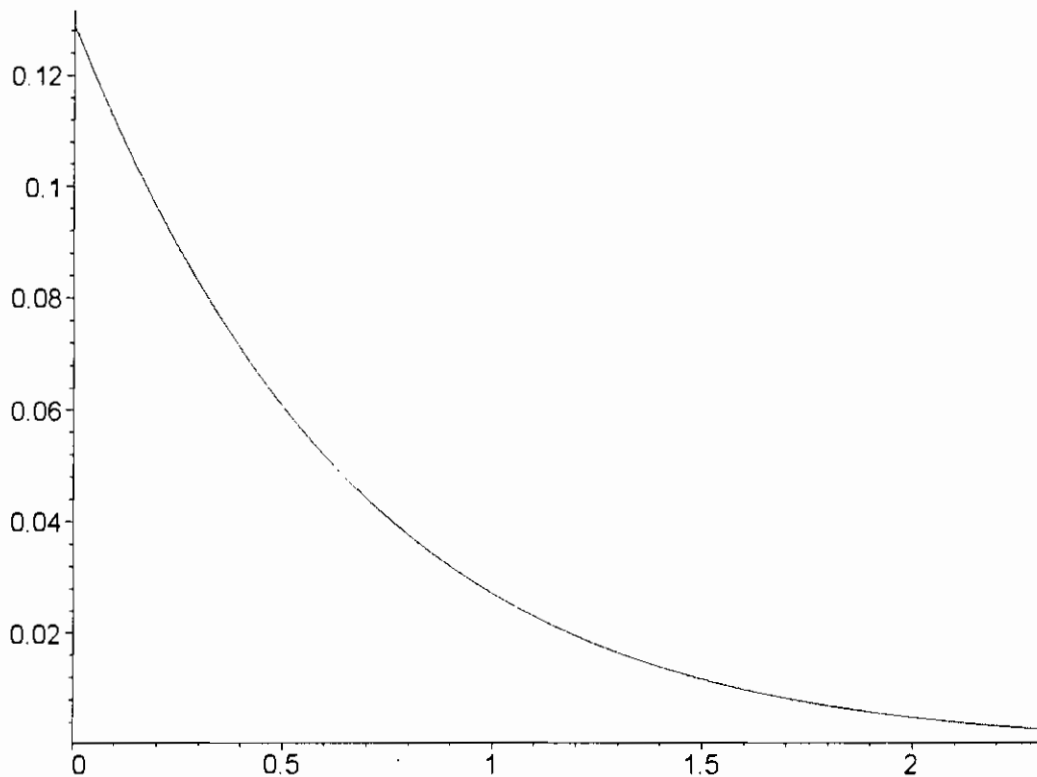
```
> RuinaRes := y -> ProbRuina(6, y, 1.5);
```

$$RuinaRes := y \rightarrow ProbRuina(6, y, 1.5)$$

Dibujamos su gráfica para y con rango de 0 a 230%:

```
> plot(RuinaRes, 0..2.3, colour=blue, title=`GRÁFICA DE
RuinaRes`);
```

GRAFICA DE RuinaRes



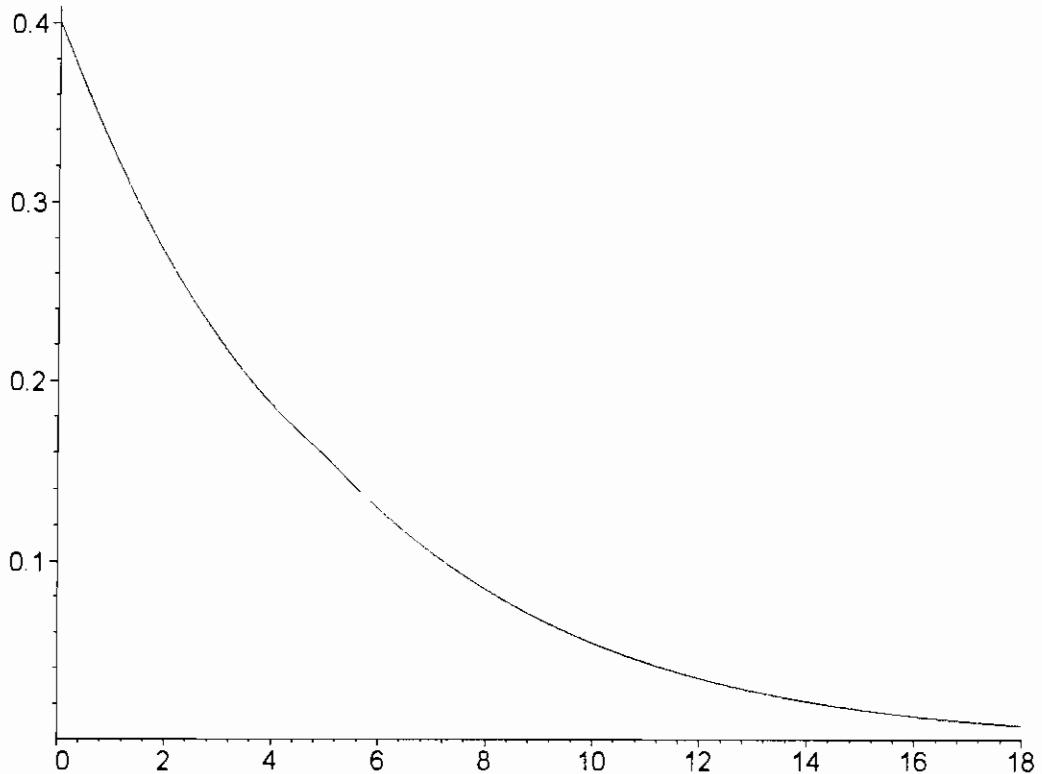
Si por diversos motivos, no podemos cobrar un recargo, entonces la pregunta es ¿cuantas reservas necesito para poder afrontar las indemnizaciones futuras con una alta probabilidad? El nuevo operador solo dependerá de $x (=U)$, ya que el recargo de seguridad queda fijado en 0. Lo llamamos **Ruina Rec**

```
> RuinaRec:=x->ProbRuina(x,0,1.5);
```

RuinaRec := x → ProbRuina(x, 0, 1.5)

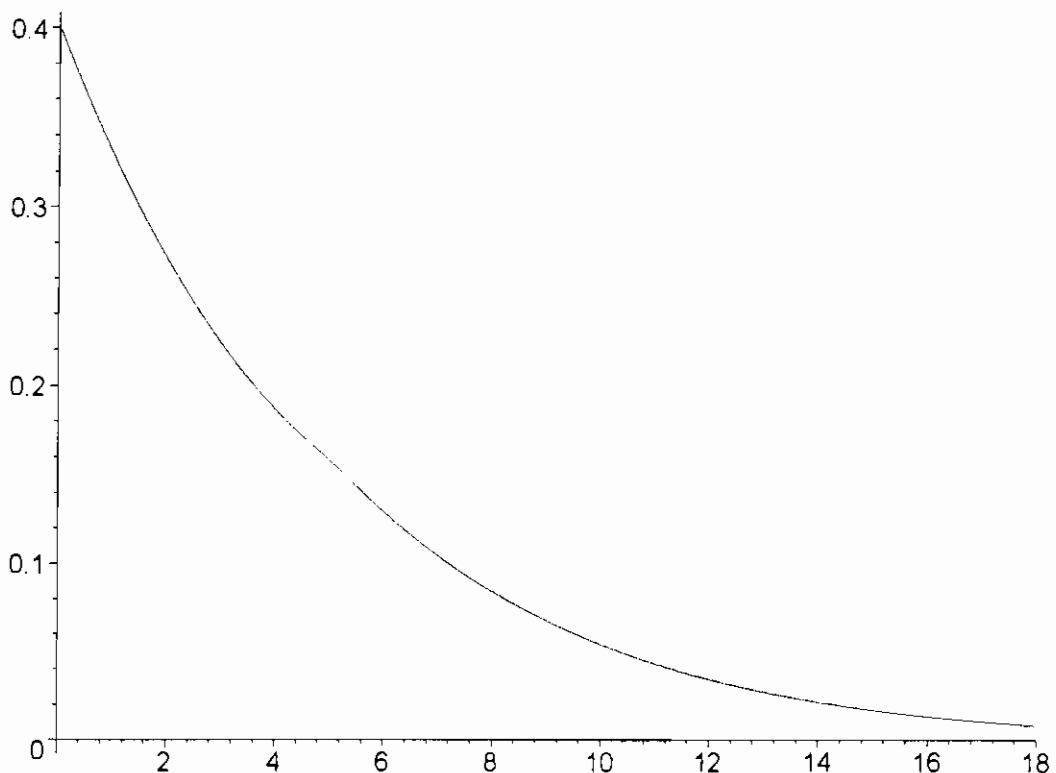
```
> plot(RuinaRec,0..18,colour=blue,title=`GRÁFICA DE
RuinaRec`);
```

GRAFICA DE RuinaRec



La variación de la probabilidad de ruina para distintos valores del recargo de seguridad desde 0 hasta 150% (dejando las reservas como variable independiente, y recordando que la asimetría del riesgo es 1.5):

```
> plots[animate](Ruina1.5,0..18,0..1.5,colour=blue,title  
  =`ANIMACIÓN DE Ruina1.5 RESPECTO DEL RECARGO`);  
  ANIMACION DE Ruina1.5 RESPECTO DEL RECARGO
```



A medida que el recargo crece, la probabilidad de ruina disminuye.



Bibliografía.

Beard, Pentikäinen, Pesonen (1984): Risk Theory. The Stochastic Basis of Insurance. Chapman and Hall.

Latorre Llorens, L. (1992): Teoría del Riesgo y sus Aplicaciones a la Empresa Aseguradora. Editorial Mapfre.