

TRABAJO FINAL DE MÁSTER

Título: Aplicación con Shiny: Provisiones no vida para triángulos *run-off* no anuales.

Autoría: Azael Martínez Martínez

Tutoría: Eva Boj del Val y Maria Teresa Costa Cor

Curso académico: 2021-2022



UNIVERSITAT DE
BARCELONA

Facultat d'Economia
i Empresa

Màster
**de Ciències
Actuarials
i Financeres**

Facultad de Economía y Empresa

Universidad de Barcelona

Trabajo Final de Máster

Máster en Ciencias Actuariales y Financieras

**APLICACIÓN CON SHINY:
PROVISIONES NO VIDA
PARA TRIÁNGULOS *RUN-
OFF* NO ANUALES**

Autoría: Azael Martínez Martínez

Tutoría: Eva Boj del Val y Maria Teresa Costa Cor

El contenido de este documento es de exclusiva responsabilidad del autor, quien declara que no ha incurrido en plagio y que la totalidad de referencias a otros autores han sido expresadas en el texto.

RESUMEN

Título: Aplicación con Shiny: Provisiones no vida para triángulos *run-off* no anuales

Autor: Azael Martínez Martínez

Tutoras: Eva Boj del Val y Maria Teresa Costa Cor

La nueva normativa de Solvencia II, junto a la próxima adaptación a las normativas financieras de IFRS 17, hacen que el papel del actuario cada vez tome más relevancia en el sector asegurador y financiero. Con el fin de adaptar este contexto en el ámbito profesional, en este trabajo se propone la realización de un aplicativo con el lenguaje R, complementando con el paquete Shiny, para la estimación de reservas mediante triángulos *run-off* para el ramo de no vida.

El aplicativo se centra en aquellas situaciones dónde los triángulos de datos no son triangulares, adaptando varios de los métodos habitualmente utilizados en el ámbito actuarial a dicha problemática. Se muestra la utilidad del aplicativo a partir de casos prácticos.

Palabras Clave: Provisiones técnicas, No vida, Solvencia II, Chain-ladder, R, Shiny

ABSTRACT

Title: Shiny App: Non-life claims for non-annual run-off triangles

Author: Azael Martínez Martínez

Tutors: Eva Boj del Val and Maria Teresa Costa Cor

The new environment with the Solvency II laws as well of the incoming IFRS 17, make the actuary a key player not only in the insurance sector, but also on the financial one. With the aim to adapt this environment to the professional scope, this document proposes an App build with the R language and the Shiny package, to calculate non-life claims with run-off triangles.

The app gets specially focus on the situation in which the run-off reserving triangles are not squared, adapting it to a selection of a few commonly used method for the actuarial claim calculation. Finally, the document ends with a few case studies to prove the utilities of the application itself.

Keywords: Claims reserving, Non-life, Solvency II, Chain-ladder, R, Shiny.

ÍNDICE

CAPÍTULO 1: INTRODUCCIÓN	1
1.1 Objetivos.....	1
CAPÍTULO 2: PROVISIONES TÉCNICAS	2
2.1 Provisiones técnicas.....	2
2.2 Provisión de prestaciones pendientes de liquidación o pago.....	2
2.3 Provisión de siniestros pendientes de declaración.....	2
2.5 Provisión de gastos internos de liquidación de siniestros.....	3
2.6 Provisiones a nivel contable y de solvencia.....	4
CAPÍTULO 3: CÁLCULO PROVISIONES	6
3.1 Triángulos <i>run-off</i>	6
3.2 Triángulos <i>run-off</i> no anuales.....	9
3.3 Métodos de cálculo de provisiones.....	9
3.3.1 Chain-ladder.....	10
3.3.2 Variación Chain-ladder.....	12
3.3.3 Mínimos cuadrados de Vylder.....	14
3.3.4 Modelo lineal generalizado.....	16
CAPÍTULO 4: APLICACIÓN CON SHINY	25
4.1 Conceptos básicos.....	25
4.2 Inputs.....	26
4.2.1 Tipos de cálculo.....	27
4.2.2 Años triángulo.....	27
4.2.3 Mes inicial.....	28
4.2.4 Último mes.....	28
4.2.5 Frecuencia filas y columnas.....	30
4.2.6 Subida del triángulo.....	31
4.2.7 Tipo de actualización flujos.....	32
4.2.8 Método de cálculo.....	36
4.2.9 Variaciones GLM.....	37
4.3 Outputs.....	38
4.3.1 Resumen.....	38
4.3.2 Factores de desarrollo.....	38
4.3.3 Triángulos estimados y originales.....	39

4.3.4	Pagos futuros	40
4.3.5	Ultimate loss	41
4.3.6	Factores de desarrollo individuales	41
CAPÍTULO 5: CASO PRÁCTICO		42
5.1	Chain-ladder.....	42
5.2	Variación Chain-ladder	45
CONCLUSIONES		49
BIBLIOGRAFÍA		50
NORMATIVA.....		52

ÍNDICE ILUSTRACIONES

Ilustración 1. Interfaz aplicación Shiny.....	26
Ilustración 2. Selección cálculo aplicación Shiny	27
Ilustración 3. Selección años a trabajar. Aplicación Shiny	27
Ilustración 4. Selección mes inicial. Aplicación Shiny	28
Ilustración 5. Selección último mes con datos. Aplicación Shiny	28
Ilustración 6. Output Aplicación Shiny para triángulo trimestral	29
Ilustración 7. Frecuencias filas y columnas. Aplicación Shiny	30
Ilustración 8. Subida del triángulo. Aplicación Shiny.....	31
Ilustración 9. Representación actualización flujos.	33
Ilustración 10. Selección método cálculo provisiones. Aplicación Shiny.....	36
Ilustración 11. Selección desviación escalada GLM. Aplicación Shiny	37
Ilustración 12. Cuadro resumen. Aplicación Shiny.....	38
Ilustración 13. Desplegable con la tabla formada por los factores de desarrollo. Aplicación.....	39
Ilustración 14. Triángulo estimado acumulado. Aplicación Shiny	40
Ilustración 15. Gráfico de pagos futuros. Aplicación Shiny	41
Ilustración 16. Factores de desarrollo calculados por el aplicativo. Ejemplo práctico	43
Ilustración 17. Triángulo acumulado estimado. Ejemplo práctico. Aplicación Shiny	44
Ilustración 18. Flujos de pagos futuros actualizados. Aplicación Shiny	45
Ilustración 19. Factores de desarrollo individuales obtenidos por el método variante de Chain-ladder. Aplicación Shiny	46
Ilustración 20. Factores de desarrollo finales obtenidos por el método variante de Chain-ladder. Aplicación Shiny	46
Ilustración 21. Triángulo estimado. Variante Chain-ladder. Aplicación Shiny	47
Ilustración 22. Flujos de pagos futuros actualizados. Aplicación Shiny	48

ÍNDICE TABLAS

Tabla 1. Triángulo <i>run-off</i> acumulado.	6
Tabla 2. Triángulo <i>run-off</i> . Cuantías estimadas.	7
Tabla 3. Triángulo <i>run-off</i> . Parte estimada ejemplo 2.....	7
Tabla 4. Ultimate loss en un triángulo <i>run-off</i> acumulado.....	8
Tabla 5. Triángulo <i>run-off</i> acumulado con periodos de desarrollo trimestrales y periodos de origen anuales.....	9
Tabla 6. Chain-ladder, ejemplo triángulo <i>run-off</i>	11
Tabla 7. Cálculo factores desarrollo Chain-ladder.....	12
Tabla 8. Tabla factores de desarrollo Chain-ladder	12
Tabla 9. Cálculo factores de desarrollo triángulo con frecuencia trimestral.....	12
Tabla 10. Factores de desarrollo individuales, variación Chain-ladder	13
Tabla 11. Pesos w con variación $i+j+1$	14
Tabla 12. Cálculo estimaciones para el método de mínimos cuadrados de Vylder	16
Tabla 13. Cuadro resumen familia exponencial.....	19
Tabla 14. Desviaciones no escaladas familia exponencial.....	22
Tabla 15. Representación nombre filas. Aplicativo Shiny.....	28
Tabla 16. Ejemplo triángulo <i>run-off</i> trimestral en función del último mes disponible	29
Tabla 17. Ejemplo 2. triángulo <i>run-off</i> trimestral en función del último mes disponible	29
Tabla 18. Actualización flujos.	32
Tabla 19. Flujos por años naturales.....	33
Tabla 20. Tipos de interés anuales.	34
Tabla 21. Cambio tipo spot a tipo forward.....	34
Tabla 22. Flujos para periodos naturales no anuales.....	35
Tabla 23. Cálculo factor de descuento con tipos forward.	36
Tabla 24. Triángulo acumulado. Ejemplo práctico	42
Tabla 25. Factores de desarrollo obtenidos por el método Chain-ladder	43
Tabla 26. Triángulo acumulado estimado. Ejemplo práctico.....	44
Tabla 27. Flujos de pagos futuros actualizados.....	44
Tabla 28. Factores de desarrollo individuales obtenidos por el método variante Chain-ladder..	45
Tabla 29. Pesos método variante de Chain-ladder	46
Tabla 30. Factores de desarrollo finales obtenidos por el método variante de Chain-ladder.....	46
Tabla 31. Triángulo estimado. Variante Chain-ladder	47
Tabla 32. Curva de tipos sin riesgo ni VA. Abril 2022.....	48
Tabla 33. Flujos de pagos futuros actualizados y factor de descuento.....	48

CAPÍTULO 1: INTRODUCCIÓN

En octubre de 2014, el Parlamento Europeo publicó lo que sería el inicio de un nuevo paradigma en la función actuarial, la Directiva 2009/138/CE, relativa a las actividades de seguro y reaseguro y conocida habitualmente como la directiva de Solvencia II.

Dicha directiva, cuyo preludio cae en Solvencia I, se realizó con la idea de asegurar una mayor regulación en las entidades aseguradoras, con el objetivo, como su nombre indica, de hacer dichas entidades más solventes. Todo ello para evitar cualquier incertidumbre económica de unas entidades que mueven grandes cantidades de dinero y que son muy relevantes en los mercados globales.

Además, a efectos contables, también se debe añadir a Solvencia II lo que serán las nuevas directrices contables con la vigencia de la IFRS 17, que sustituirá a la normativa IFRS 4, que sigue vigente hoy en día, a la espera de la total aplicación de la nueva ley.

De igual manera que ocurre con Solvencia II, la nueva normativa IFRS 17 pretende otorgar a las entidades aseguradoras una mayor transparencia en sus cuentas.

Dichas normativas, además, incluyen grandes cambios en la forma de trabajar de las entidades aseguradoras, donde la conexión entre los diferentes departamentos de las entidades, así como la participación obligatoria de terceras como auditorías, tendrán un papel crucial en todo el proceso.

La complejidad y extensión de la normativa provocan que este trabajo apenas pueda centrarse en una pequeña parte de estas. De esta forma, este trabajo se centra en el ramo de no vida y el cálculo de las provisiones técnicas de dicho ramo, basado en triángulos. Todo ello, ajustado a la problemática que se deriva en dicho cálculo cuando los datos del triángulo no son de periodicidad anual.

Esta problemática se abordará bajo la creación de una aplicación programada en lenguaje R y complementado con diversos paquetes, entre los que destaca Shiny (*Chang et al., 2021*).

1.1 Objetivos

El objetivo de este trabajo es realizar un programa basado en Shiny y lenguaje R (*R Development Core Team, 2022*) que permita al usuario calcular con facilidad las provisiones técnicas del ramo de no vida.

El aplicativo se centra en aquellas situaciones dónde se trabaja con triángulos *run-off* basados en periodos de desarrollo y origen con recurrencia distinta a la anual.

CAPÍTULO 2: PROVISIONES TÉCNICAS

2.1 Provisiones técnicas

Tal y como especifica el artículo 69 de la Ley 20/2015, de 14 de julio, de ordenación, supervisión y solvencia de las entidades aseguradoras y reaseguradoras (LOSSEAR), las provisiones técnicas serán aquellas que *“reflejen todas las obligaciones derivadas de contratos de seguro y reaseguro”*.

Siguiendo con dicho artículo, estas se valorarán tal que: *“El valor de las provisiones técnicas se corresponderá con el importe actual que las entidades aseguradoras y reaseguradoras tendrían que pagar si transfirieran sus obligaciones de seguro y reaseguro de manera inmediata a otra entidad aseguradora o reaseguradora.”*

De esta manera, y como iremos viendo a lo largo de las siguientes páginas, será necesario hacer una estimación futura de las obligaciones derivadas de dichos contratos, con el objetivo de conocer la cantidad total requerida para hacer frente a las mismas.

2.2 Provisión de prestaciones pendientes de liquidación o pago

Tal y como especifica el artículo 40 del Reglamento de Ordenación y Supervisión de los Seguros Privados (*Real Decreto 2486/1998*, conocido habitualmente como ROSSP), se considerarán provisiones de prestaciones pendientes de liquidación o pago *“el importe de todos aquellos siniestros ocurridos y declarados antes del cierre del ejercicio”*

A estos, además, se les sumarán *“los gastos de carácter externo inherentes a la liquidación de siniestros y, en su caso, los intereses de demora y las penalizaciones legalmente establecidas en las que haya incurrido la entidad”*, así como las provisiones por obligaciones relativas a participaciones en beneficios y los extornos.

Tal y como establece el diccionario de seguros de Allianz (*ALLIANZ SEGUROS, Diccionario de seguros*), se consideran extornos aquellos importes que se deban devolver por *“eventualidades externas”*, las cuales se pueden solucionar ya sea con la rescisión del contrato o la devolución de primas.

Las normas relativas a la *“Provisión de participación en beneficios y para extornos”*, quedan recogidas en el artículo 38 del ROSSP.

2.3 Provisión de siniestros pendientes de declaración

En el artículo 41 del ROSSP se define la provisión de los siniestros pendientes de declaración, que serán aquellos que, aunque hayan ocurrido dentro del período, por ciertas circunstancias aún no han sido declarados.

La compañía debe computar los siniestros en el año de ocurrencia, independientemente del año en que se paguen o se declaren, siendo dichos siniestros incluibles dentro de las provisiones de ese mismo año.

Como la compañía desconoce esas cantidades tendrá que realizar una estimación para concretarlas.

En el citado artículo 41 se establece un cálculo para la obtención del número de siniestros pendientes, simbolizado como N , en función del número de siniestros pendientes de los últimos tres ejercicios, tal que:

$$N_t = \frac{N_{t-1} + N_{t-2} + N_{t-3}}{P_{t-1} + P_{t-2} + P_{t-3}} \times P_t$$

- Donde P simbolizará las primas devengadas en los diferentes ejercicios.

El mismo artículo posteriormente ofrece una fórmula muy similar, para el cálculo de los costes medios de los siniestros pendientes de declarar, simbolizados como C , quedando tal que:

$$C_t = \frac{C_{t-1} + C_{t-2} + C_{t-3}}{Q_{t-1} + Q_{t-2} + Q_{t-3}} \times Q_t$$

- Donde Q simbolizará el coste medio de aquellos siniestros que ya hayan sido declarados.

2.5 Provisión de gastos internos de liquidación de siniestros

Siguiendo con el ROSSP, y concretamente el artículo 42, también se provisionarán aquellos gastos administrativos derivados de dichos siniestros pendientes de declarar o citando dicho artículo: *“necesarios para la total finalización de los siniestros que han de incluirse en la provisión de prestaciones tanto del seguro directo como del reaseguro aceptado”*.

De igual manera que en el anterior apartado, en caso de no tener metodología propia para el cálculo de estos, la ley establece un método a partir de la relación entre los *“gastos internos imputables a las prestaciones”* y *“el importe de las prestaciones pagadas en el ejercicio que se cierra”* con algunas correcciones.

Dicha relación, que será un porcentaje, *“deberá multiplicarse, al menos, por el 50 por 100 del importe de la provisión para prestaciones pendientes de liquidación o pago más el 100 por 100 del importe de la provisión de siniestros pendientes de declaración.”*

2.6 Provisiones a nivel contable y de solvencia

Como ya hemos visto en los primeros puntos de este capítulo, muchas de las directrices relativas a la contabilización de las provisiones provienen del ROSSP. Esta normativa que data del 1998 está pendiente de ser actualizada.

Esta situación ha provocado que, a la hora de computar las provisiones técnicas, nos encontraremos con diferentes resultados, pues, en función de si estas se realizan a nivel contable o a nivel de Solvencia II, tendrán un cálculo distinto.

De esta forma y con el objetivo de poder entender la doble computación de las provisiones técnicas en la siguiente lista, tal y como se indica en *Boj et al. (2020)*, se pueden ver las diferentes provisiones que se computan en función del punto de vista utilizado.

Las provisiones a calcular efectos de solvencia son:

1. Provisión del seguro de vida.
2. Provisión de los seguros de enfermedad con técnicas similares a las de los seguros de vida.
3. Provisión para primas (en los seguros distintos al de vida).
4. Provisión para siniestros pendientes (en los seguros distintos al de vida):
 - Provisión para siniestros declarados pendientes de liquidación o pago.
 - Provisión para siniestros ocurridos per no declarados.
 - Provisión por gastos de liquidación de siniestros (ULAE).
5. Provisión para primas (en los seguros de enfermedad calculados con técnicas similares a los seguros distintos al de vida).
6. Provisión para siniestros pendientes (en los seguros de enfermedad calculados con técnicas similares a los de los seguros distintos al de vida):
 - Provisión para siniestros declarados pendientes de liquidación o pago.
 - Provisión para siniestros ocurridos per no declarados.
 - Provisión por gastos de liquidación de siniestros (ULAE).

Mientras que las provisiones a calcular a efectos contables serán las siguientes:

1. Provisión de primas no consumidas.
2. Provisión de riesgos en curso.
3. Provisión de seguros de vida.
4. Provisión de participación en beneficios y para extornos.
5. Provisión de prestaciones:
 - Provisión de prestaciones pendientes de liquidación y pago.
 - Provisión de prestaciones pendientes de declaración.
 - Provisión de gastos internos de liquidación de siniestros.
6. Provisión de estabilización.

7. Provisión del seguro de decesos.
8. Provisión del seguro de enfermedad.
9. Provisión de desviaciones en las operaciones de capitalización por sorteo.
10. Provisión de gestión de riesgos derivados de la internacionalización asegurados por cuenta del estado.

CAPÍTULO 3: CÁLCULO PROVISIONES

3.1 Triángulos *run-off*

Dentro del ámbito profesional, a la hora de realizar los cálculos para obtener las provisiones de los siniestros ocurridos, pero no declarados/reportados aún (conocidos mayormente como *IBNR*, siglas en inglés de: *Incurred But Not Reported*), se suele trabajar con los llamados *run-off triangles*.

Estos triángulos, siguiendo con la propuesta de *Weindorfer (2012)*, suelen representarse de la siguiente manera:

Años de origen i	Años de desarrollo $j = 0, 1, \dots, k$					
	0	1	2	\dots	$k-1$	k
0	$C_{0,0}$	$C_{0,1}$	$C_{0,2}$	\dots	$C_{0,k-1}$	$C_{0,k}$
1	$C_{1,0}$	$C_{1,1}$	$C_{1,2}$	\dots	$C_{1,k-1}$	
\dots	\dots	\dots	\dots	\dots		
$k-2$	$C_{k-2,0}$	$C_{k-2,1}$	$C_{k-2,2}$			
$k-1$	$C_{k-1,0}$	$C_{k-1,1}$				
k	$C_{k,0}$					

Tabla 1. Triángulo *run-off* acumulado. [Fuente: Elaboración propia]

Donde:

- i : Años de origen de ocurrencia del siniestro.
- j : Años transcurridos hasta el pago de un siniestro.
- $C_{i,j}$: Cuantías acumuladas pagadas de siniestros ocurridos en el año de origen i , hasta el año de desarrollo j (incluyendo este mismo).
- Si las cantidades estuviesen sin acumular, entonces, se hará uso de la simbología $c_{i,j}$, que será la cantidad pagada sólo en el año de origen i y el año de desarrollo j , sin contabilizar los pagos de los anteriores años de desarrollo.

De esta manera, si nos fijamos de nuevo en la representación gráfica del triángulo, podremos corroborar que:

$$\sum_{j=0}^2 c_{0,j} = C_{0,2}$$

Es decir, $C_{0,2}$ es la suma de todas aquellas cuantías no acumuladas de siniestros ocurridos en el año de origen 0 pagados hasta el año de desarrollo 2 .

De esta forma, las casillas marcadas en gris en la *Tabla 2* representarán las cuantías estimadas futuras a pagar para cada uno de los años de origen y , por ende, la suma de las cuantías estimadas no acumuladas en cada año de origen i . Estas serán las cantidades que

tendrán que provisionar las compañías para hacer frente a los pagos futuros de aquellos siniestros ocurridos en el año de origen i .

Años de origen i	Años de desarrollo $j = 0, 1, \dots, k$					
	0	1	2	...	$k-1$	k
0	$C_{0,0}$	$C_{0,1}$	$C_{0,2}$...	$C_{0,k-1}$	$C_{0,k}$
1	$C_{1,0}$	$C_{1,1}$	$C_{1,2}$	$C_{1,k-1}$	
...		
$k-2$	$C_{k-2,0}$	$C_{k-2,1}$	$C_{k-2,2}$			
$k-1$	$C_{k-1,0}$	$C_{k-1,1}$				
k	$C_{k,0}$					

Tabla 2 Triángulo run-off. Cuantías estimadas. [Fuente: Elaboración propia]

Para facilitar la comprensión del último párrafo, haremos un ejemplo gráfico. Si quisiéramos obtener la provisión que debe hacer la compañía para el año de origen $k-1$, tendríamos que considerar las siguientes celdas en gris:

Años de origen i	Años de desarrollo $j = 0, 1, \dots, k$					
	0	1	2	...	$k-1$	k
0	$C_{0,0}$	$C_{0,1}$	$C_{0,2}$...	$C_{0,k-1}$	$C_{0,k}$
1	$C_{1,0}$	$C_{1,1}$	$C_{1,2}$	$C_{1,k-1}$	
...		
$k-2$	$C_{k-2,0}$	$C_{k-2,1}$	$C_{k-2,2}$			
$k-1$	$C_{k-1,0}$	$C_{k-1,1}$	$C'_{k-1,2}$...	$C'_{k-1,k-1}$	$C'_{k-1,k}$
k	$C_{k,0}$					

Tabla 3. Triángulo run-off. Parte estimada ejemplo 2. [Fuente: Elaboración propia]

De esta manera, y al estar trabajando con cuantías acumuladas, la cantidad que la compañía tendrá que aprovisionar para los siniestros ocurridos en $k-1$, será igual a:

$$C'_{k-1,k} - C_{k-1,1}$$

Donde:

- $C'_{k-1,k}$ será la cantidad acumulada estimada (de ahora en adelante las estimaciones en los triángulos *run-off* aparecerán como C') hasta el último periodo de desarrollo k . Como veremos un poco más adelante dicha cantidad se suele llamar *ultimate loss (IRMI)*
- $C_{k-1,1}$ será la cantidad ya conocida y que la compañía ya ha pagado.

De forma equivalente, también se podría obtener la provisión para el año de origen $k-1$ como la suma de las cuantías no acumuladas:

$$\sum_{j=2}^k C_{k-1,j}$$

Como ya hemos mencionado anteriormente, a la última columna del triángulo completado, una vez este incluya las estimaciones de cuantías futuras y siempre que esté acumulado ($C_{i,j}$ y no $c_{i,j}$), se le conoce normalmente como *ultimate loss*, siendo esta la suma de las cantidades ya conocidas y las futuras estimadas (que compondrán la *IBNR*).

Años de origen i	Años de desarrollo $j = 0, 1, \dots, k$					
	0	1	2	\dots	$k-1$	k
0	$C_{0,0}$	$C_{0,1}$	$C_{0,2}$	\dots	$C_{0,k-1}$	$C_{0,k}$
1	$C_{1,0}$	$C_{1,1}$	$C_{1,2}$	\dots	$C_{1,k-1}$	$C'_{1,k}$
\dots	\dots	\dots	\dots	\dots		\dots
$k-2$	$C_{k-2,0}$	$C_{k-2,1}$	$C_{k-2,2}$			$C'_{k-2,k}$
$k-1$	$C_{k-1,0}$	$C_{k-1,1}$				$C'_{k-1,k}$
k	$C_{k,0}$					$C'_{k,k}$

Tabla 4. *Ultimate loss* en un triángulo run-off acumulado. [Fuente: Elaboración propia]

El *ultimate loss* no es más que la cuantía total que la compañía espera acabar pagando en el futuro de todos aquellos siniestros ocurridos en un año de origen i .

3.2 Triángulos *run-off* no anuales

Si bien en la teoría se trabaja con triángulos *run-off* de carácter anual, debido principalmente a su facilidad de comprensión, en los entornos profesionales es habitual que las compañías, en muchas ocasiones, con el objetivo de poder hacer juicios y análisis más concisos, trabajen con triángulos con periodos más pequeños.

A modo de ejemplo, supongamos que una entidad aseguradora decide hacer un análisis de siniestralidad basándose en periodos trimestrales, frecuentemente utilizados a efectos contables.

De esta manera, si trabajamos con periodos de desarrollo no-anuales, pero dejamos los periodos de origen i como anuales, tendremos un triángulo completo que no será cuadrado, o lo que es lo mismo, que no tendrá el mismo número de filas y columnas.

Si consideramos que dicha entidad dispone de toda la información del último año de origen, $i = 2$, nos encontraremos con una representación gráfica tal que:

Años de origen i	Trimestres de desarrollo $j = 0, 1, \dots, k$											
	0	1	2	3	4	5	6	7	8	9	10	11
0	$C(0;0)$	$C(0;1)$	$C(0;2)$	$C(0;3)$	$C(0;4)$	$C(0;5)$	$C(0;6)$	$C(0;7)$	$C(0;8)$	$C(0;9)$	$C(0;10)$	$C(0;11)$
1	$C(1;0)$	$C(1;1)$	$C(1;2)$	$C(1;3)$	$C(1;4)$	$C(1;5)$	$C(1;6)$	$C(1;7)$				
2	$C(2;0)$	$C(2;1)$	$C(2;2)$	$C(2;3)$								

Tabla 5. Triángulo *run-off* acumulado con periodos de desarrollo trimestrales y periodos de origen anuales. [Fuente: Elaboración propia]

Con este simple ejemplo, de apenas tres años, podemos ver cómo varía el triángulo completo final, dejando doce columnas para cubrir los doce trimestres que componen los tres años considerados. De esta manera, lo que desde un punto de vista práctico podría ser un triángulo fácil, deja una situación mucho más compleja desde un punto gráfico, pero también desde un enfoque matemático.

3.3 Métodos de cálculo de provisiones

Cuando hablamos de los métodos utilizados para el cálculo de provisiones técnicas es muy habitual utilizar la clasificación propuesta por Taylor (1986), diferenciando dos tipologías:

- Metodologías deterministas
- Metodologías estocásticas.

Si bien es cierto que en este trabajo se pondrá poco énfasis en la forma de clasificar los métodos en sí, es importante conocer las diferencias entre ambos.

Mientras que las metodologías deterministas no usan variables aleatorias, las metodologías estocásticas sí lo hacen.

Como es de esperar, los métodos deterministas, como podrían ser el modelo Chain-ladder, el modelo Bornhuetter-Ferguson o el modelo Link Ratio entre otros, suelen tener menos dificultad que los métodos estocásticos, pues la inclusión de una variable aleatoria hace

que el modelo se complique. Esto hace que en el ámbito profesional sea más habitual utilizar los métodos deterministas.

Con la implementación de la Directiva de Solvencia II, *Directiva 2009/138/CE*, mucho más restrictiva que las anteriores, son muchas las compañías que han visto con buenos ojos la inclusión de los métodos estocásticos, dándoles mucho más protagonismo del que habían tenido anteriormente.

Los métodos basados en el modelo lineal generalizado (GLM) o el doble Chain-ladder cada vez cobran más protagonismo en un entorno que, con la implementación de la mencionada normativa, parece estar cambiando en muchos puntos clave.

Todo y con ello, este trabajo, con el objetivo de no complicar en exceso los triángulos *run-off*, se ha centrado en la programación de algunos métodos deterministas, mucho más sencillos no sólo en la práctica sino también a modo explicativo, añadiendo el modelo lineal generalizado como único método estocástico.

De esta manera, los métodos finalmente desarrollados han sido los siguientes:

Métodos deterministas:

- Chain-ladder.
- Variación Chain-ladder con ponderación de pesos $i+j+1$.
- Mínimos cuadrados de Vylder.

Métodos estocásticos:

- Modelo lineal generalizado, GLM.

3.3.1 Chain-ladder

El primer método propuesto es Chain-ladder, uno de los métodos deterministas más utilizados en la actualidad en el ámbito profesional debido a su fácil aplicación.

El método Chain-ladder utiliza el triángulo *run-off* acumulado y se basa en unos factores de desarrollo, denominados estimadores, que nos permiten calcular el gasto futuro estimado, a partir de la información histórica de la compañía aseguradora.

Este método pierde eficiencia al avanzar en los periodos de desarrollo, pues los estimadores obtenidos se calcularán con un menor número de datos históricos, siendo la estimación cada vez más incierta. Todo y con eso, lo habitual es que las compañías, como ya hemos comentado anteriormente, acumulen la gran mayoría de pagos en los primeros años de desarrollo (no confundir con periodos).

En *Boj et al. (2020)*, podremos ver como en la teoría se suelen deducir las fórmulas del método Chain-ladder a partir de un triángulo anual, tanto en sus periodos de desarrollo j como en sus periodos de origen i .

De este modo, el estimador o factor de desarrollo se obtiene como:

$$\hat{m}_h = \frac{\sum_{i=0}^{k-h-1} C_{i,h+1}}{\sum_{i=0}^{k-h-1} C_{i,h}}.$$

Estos factores de desarrollo posteriormente nos permitirán obtener las cuantías futuras estimadas y así completar el triángulo *run-off*, ya que:

$$C'_{i,h+1} = \hat{m}_h \cdot C_{i,h}.$$

Mediante este procedimiento se puede calcular también el *ultimate loss* como:

$$C'_{i,j} = C_{i,k-1} \cdot \prod_{h=k-1}^{j-1} \hat{m}_h.$$

Asumiendo $C'_{i,j}$ como la cuantía acumulada en el último período de desarrollo j disponible para un período de origen i distinto al primero, pues este tendrá todos los datos disponibles.

Dicha formulación, si bien es totalmente válida, no tiene en cuenta aquellas situaciones donde el triángulo *run-off* no tenga el mismo número de filas y de columnas (mismo número de periodos de origen y desarrollo).

Para poder entender mejor el método y aprovechando la facilidad de comprensión que otorga el hecho de considerarlo todo anual, a continuación se proporcionan ejemplos gráficos. Si partimos de un triángulo anual de k años de origen i , y k años de desarrollo j , entonces:

Años de origen i	Años de desarrollo $j = 0, 1, \dots, k$					
	0	1	2	\dots	$k-1$	k
0	$C_{0,0}$	$C_{0,1}$	$C_{0,2}$	\dots	$C_{0,k-1}$	$C_{0,k}$
1	$C_{1,0}$	$C_{1,1}$	$C_{1,2}$	\dots	$C_{1,k-1}$	
\dots	\dots	\dots	\dots	\dots		
$k-2$	$C_{k-2,0}$	$C_{k-2,1}$	$C_{k-2,2}$			
$k-1$	$C_{k-1,0}$	$C_{k-1,0}$				
k	$C_{k,0}$					

Tabla 6 Chain-ladder, ejemplo triángulo run-off. [Fuente: Elaboración Propia]

Para calcular el factor de desarrollo \hat{m}_0 , que será aquel que nos permita obtener las estimaciones de aquellas cuantías que vayan del año de desarrollo 0 al 1 , tendremos la formula siguiente:

$$\hat{m}_0 = \frac{\sum_{i=0}^{k-1} C_{i,1}}{\sum_{i=0}^{k-1} C_{i,0}}.$$

Lo que sería el equivalente a coger las siguientes cuantías del triángulo *run-off*:

Años de origen i	Años de desarrollo $j = 0, 1, \dots, k$		Años de origen i	Años de desarrollo $j = 0, 1, \dots, k$	
	0	1		0	1
0	$C_{0,0}$	$C_{0,1}$	0	$C_{0,0}$	$C_{0,1}$
1	$C_{1,0}$	$C_{1,1}$	1	$C_{1,0}$	$C_{1,1}$
...
$k-2$	$C_{k-2,0}$	$C_{k-2,1}$	$k-2$	$C_{k-2,0}$	$C_{k-2,1}$
$k-1$	$C_{k-1,0}$	$C_{k-1,1}$	$k-1$	$C_{k-1,0}$	$C_{k-1,1}$
k	$C_{k,0}$		k	$C_{k,0}$	

Tabla 7 Cálculo factores desarrollo Chain-ladder [Fuente: Elaboración propia]

donde los segmentos subrayados corresponden, por orden, a

$$\sum_{i=0}^{k-1} C_{i,1}$$

y,

$$\sum_{i=0}^{k-1} C_{i,0}$$

Como para el año de desarrollo 1 apenas existen k datos ($k - 1 + 1$, pues contabilizamos el 0), debemos coger la misma cantidad en el periodo de desarrollo anterior, que, en el ejemplo, sería el año de desarrollo 0. De esta forma, para aquellos triángulos *run-off* que dispongan del mismo número de periodos de desarrollo y origen, siempre se irá cogiendo un dato menos cuando avancemos en los posteriores años de desarrollo \hat{m}_h , como ya hemos visto con anterioridad en este trabajo.

	Años de desarrollo $j = 0, 1, \dots, k$					
	0 a 1	1 a 2	2 a 3	...	$k-2$ a $k-1$	$k-1$ a k
Filas utilizadas	$k-1$	$k-2$	$k-3$		2	1

Tabla 8 Tabla factores de desarrollo Chain-ladder [Fuente: Elaboración propia]

De esta forma, si trabajásemos con un triángulo *run-off* que no siguiese dichas características, como sería un triángulo trimestral, tal como se ha propuesto anteriormente, tendríamos:

Años de origen i	Trimestres de desarrollo $j = 0, 1, \dots, k$									
	0	1	2	3	4	5	6	7	8	9
0	$C(0;0)$	$C(0;1)$	$C(0;2)$	$C(0;3)$	$C(0;4)$	$C(0;5)$	$C(0;6)$	$C(0;7)$	$C(0;8)$	$C(0;9)$
1	$C(1;0)$	$C(1;1)$	$C(1;2)$	$C(1;3)$	$C(1;4)$	$C(1;5)$				
2	$C(2;0)$	$C(2;1)$								

Tabla 9 Cálculo factores de desarrollo triángulo con frecuencia trimestral [Fuente: Elaboración propia]

Para calcular el primer factor de desarrollo utilizaríamos todas las filas, pues el triángulo dispone de datos reales para todos ellos.

3.3.2 Variación Chain-ladder

Después de la aparición del método determinista Chain-ladder, el cual es punto de partida de la gran mayoría de los métodos posteriores, incluidos lo estocásticos, empezaron a

aparecer variantes de dicha metodología, en las que el factor de desarrollo se obtenía de una forma distinta.

En el cálculo de los factores de desarrollo en el método Chain-ladder todas las cuantías tienen la misma ponderación, independientemente de que la cuantía acumulada $C_{i,j}$ sea de un periodo de origen i más cercano al actual o no.

Este problema hizo que se planteasen diferentes tipos de ponderaciones, donde se intentaba dar mayor peso a aquellos periodos de origen i más cercanos, asumiendo que el comportamiento futuro debería parecerse más al de estos que al de los periodos de origen i más antiguos.

Dentro de dichas ponderaciones, destacan dos metodologías que utilizan dos tipos de pesos w_{ij} :

- $i + j + 1$,
- $(i + j + 1)^2$.

En este trabajo, con el objetivo de poder cubrir una mayor variedad de métodos, y teniendo en cuenta que la teoría sería la misma (adaptando el peso correspondiente), nos hemos centrado en la primera de las ponderaciones propuestas: $i + j + 1$.

Como en este método se dan ponderaciones diferentes a cada uno de los años de origen i , es habitual calcular los factores de desarrollo individuales, también denominados empíricos.

Dichos factores empíricos, de ahora en adelante d_{ih} , se calculan, a partir del caso de triángulos *run-off* con el mismo número de periodos de origen y desarrollo, como:

$$d_{ih} = \frac{C_{i,h+1}}{C_{i,h}}$$

- siempre que $C_{i,h}$ y $C_{i,h+1}$ existan.

De esta forma, y volviendo al triángulo anual que hemos estado utilizando como ejemplo, el número de d_{ih} que se calcularían serían:

Años de origen i	Factores de desarrollo $h = 0, 1, \dots, k-1$				
	0	1	2	...	$k-1$
0	$d_{0,0} = \frac{C_{0,1}}{C_{0,0}}$	$\frac{C_{0,2}}{C_{0,1}}$	$\frac{C_{0,3}}{C_{0,2}}$...	$\frac{C_{0,k}}{C_{0,k-1}}$
1	$\frac{C_{1,1}}{C_{1,0}}$	$\frac{C_{1,2}}{C_{1,1}}$	$\frac{C_{1,3}}{C_{1,2}}$	
...	
$k-2$	$\frac{C_{k-2,1}}{C_{k-2,0}}$	$\frac{C_{k-2,2}}{C_{k-2,1}}$			
$k-1$	$\frac{C_{k-1,1}}{C_{k-1,0}}$				

Tabla 10 Factores de desarrollo individuales, variación Chain-ladder [Fuente: Elaboración propia]

Siendo los pesos w_{ih} iguales a:

Años de origen i	Factores de desarrollo $h = 0, 1, \dots, k-1$				
	0	1	2	\dots	$k-1$
0	1	2	3	\dots	k
1	2	3	4	\dots	
\dots	\dots	\dots	\dots	\dots	
$k-2$	$k-1$	k			
$k-1$	k				

Tabla 11 Pesos w con variación $i+j+1$ [Fuente: Elaboración propia]

De esta manera, el factor de desarrollo final se obtendrá ponderando cada d_{ih} respecto a su peso w_{ih} , que es el peso total de dicho periodo de desarrollo j :

$$\hat{d}_h = \frac{\sum_{i=0}^{h-h-1} w_{ih} \cdot d_{ih}}{\sum_{i=0}^{h-h-1} w_{ih}}.$$

Una vez se tiene el factor de desarrollo final, el proceso para obtener las estimaciones de los pagos futuros se realizará de igual forma que se hace en el método Chain-ladder:

$$C'_{i,h+1} = \hat{d}_h \cdot C_{i,h}.$$

En el caso del *ultimate loss*,

$$C'_{i,j} = C_{i,k-1} \cdot \prod_{h=k-1}^{j-1} \hat{d}_h.$$

De igual manera que ocurre en el método de Chain-ladder, muchas de las fórmulas no son aptas para aquellos triángulos *run-off* que no tengan el mismo número de periodos de origen respecto a sus periodos de desarrollo.

3.3.3 Mínimos cuadrados de Vylder

Continuando con los métodos deterministas, el método de mínimos cuadrados de Vylder se basa en una estimación por el método de mínimos cuadrados. Esta estimación de mínimos cuadrados se hace mediante dos variables:

- x_i , que es la cuantía total a pagar por aquellos siniestros ocurridos en el período de origen i , o lo que es lo mismo, el *ultimate loss*.

- p_j , que será la proporción de dicha cuantía total que se paga en el período de desarrollo j .

Sí multiplicamos ambas variables, obtendremos la cuantía no acumulada del período de origen i y período de desarrollo j :

$$c_{ij} = x_i \cdot p_j$$

con la condición de que:

$$\sum_{j=0}^k p_j = 1$$

- Siendo k el número total de periodos de desarrollo j .

De esta manera, la fórmula de los estimadores por mínimos cuadrados quedará tal que:

$$\text{Min} \sum_{\forall i, j \in D} (c_{ij} - x_i \cdot p_j)^2$$

siendo:

$$x_i = \frac{\sum_{\forall j \in J_i} c_{ij} \cdot p_j}{\sum_{\forall j \in J_i} p_j^2}$$

$$p_j = \frac{\sum_{\forall i \in I_j} c_{ij} \cdot x_i}{\sum_{\forall i \in I_j} x_i^2}.$$

Con el objetivo de poder interpretar mejor el método y siguiendo con la misma estructura asumida en los métodos anteriores, haremos un ejemplo teórico con el triángulo *run-off* anual que hemos ido utilizando.

De esta manera, una vez hemos obtenido los resultados de x_i y p_j podremos obtener las cuantías no acumuladas c_{ij} y completar un triángulo no acumulado tal que:

Años de origen i	Años de desarrollo $j = 0, 1, \dots, k-1, k$					
	0	1	2	\dots	$k-1$	k
0	$c_{0,0}$	$c_{0,1}$	$c_{0,2}$	\dots	$c_{0,k-1}$	$c_{0,k}$
1	$c_{1,0}$	$c_{1,1}$	$c_{1,2}$	\dots	$c_{1,k-1}$	$x_1 \cdot p_k$
\dots	\dots	\dots	\dots	\dots	\dots	\dots
$k-2$	$c_{k-2,0}$	$c_{k-2,1}$	$c_{k-2,2}$	\dots	$x_{k-2} \cdot p_{k-1}$	$x_{k-2} \cdot p_k$
$k-1$	$c_{k-1,0}$	$c_{k-1,1}$	$x_{k-1} \cdot p_2$	\dots	$x_{k-1} \cdot p_{k-1}$	$x_{k-1} \cdot p_k$
k	$c_{k,0}$	$x_k \cdot p_1$	$x_k \cdot p_2$	\dots	$x_k \cdot p_{k-1}$	$x_k \cdot p_k$

Tabla 12 Cálculo estimaciones para el método de mínimos cuadrados de Vylder [Fuente: Elaboración propia]

- El número de x_i estimadas será igual al total de periodos de origen i ,
- y el número de p_j estimados será igual al total de periodos de desarrollo j .

Y siempre recordando que las estimaciones se realizarán en aquella parte del triángulo *run-off* desconocida, pues, como se puede ver en el triángulo, las cuantías c_{ij} ya conocidas se mantienen iguales.

3.3.4 Modelo lineal generalizado

Con el objetivo de abordar no solo métodos deterministas, que, como ya hemos visto, son de fácil ejecución, en este trabajo también se ha querido incluir uno de los métodos estocásticos más utilizados en la actualidad, el modelo GLM.

El modelo GLM, de las siglas del término anglosajón, *generalized linear model*, o modelo lineal generalizado (a veces MLG, si se usa la terminología española), es un método que ha adquirido cada vez más presencia en las funciones actuariales debido a la inclusión de las directivas de Solvencia II.

Siguiendo con la propuesta de Boj y Costa (2014), podemos describir este modelo como una función lineal formada por P predictores o un vector de factores de riesgo F (F_1, F_2, \dots, F_p), que intenta explicar un vector dependiente Y_{nx1} formado por las variables independientes y_i :

$$Y_{nx1} = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{pmatrix}$$

donde cada variable independiente tiene una estructura tal que:

$$y_i = \beta_0 + \sum_{j=1}^P \beta_j f_{ij} + \varepsilon_i.$$

A primera vista, dicha estructura parece idéntica a la ya propuesta por los modelos clásicos de regresión lineal, pero estos, a diferencia de los generalizados, limitan la función del error (y por ende de la regresión), a la normal, asumiendo que el error sigue la siguiente distribución:

$$\varepsilon_i \sim N(0, \sigma^2).$$

Dejando, que y_i tuviese entonces una distribución tal que:

$$y_i \sim N(\mu_i, \sigma^2)$$

donde,

$$E[y_i] = \mu_i = \beta_0 + \sum_{j=1}^P \beta_j f_{ij}$$

y varianza constante,

$$Var[y_i] = \sigma^2.$$

El modelo lineal generalizado asume que dicha distribución Y no tiene por qué seguir una normal, pudiendo esta seguir cualquiera de las distribuciones derivadas de la familia exponencial propuestas por *McCullagh y Nelder (1989)*.

Dichas distribuciones siguen una función de distribución tal que:

$$f(y_i, \theta_i, \phi_i) = \exp \left\{ \frac{y_i \theta_i - b(\theta_i)}{a(\phi_i)} + c(y_i, \phi_i) \right\}$$

donde:

- ϕ_i es el parámetro de dispersión y,
- θ_i es el parámetro canónico.

Por su parte, $a(\cdot)$, $b(\cdot)$ y $c(\cdot)$ representan un conjunto de funciones especificadas. Debido a esta función de densidad, la esperanza y varianza de la distribución cambiará, siendo entonces:

$$E[y_i] = \mu_i = b'(\theta_i)$$

y

$$Var[y_i] = b''(\theta_i) a(\phi_i)$$

quedando la varianza como un conjunto de dos funciones:

- $b''(\theta_i)$, denominada función de varianza y dependiente de la esperanza de y_i : $b''(\theta_i) = V(\mu_i) = V(b'(\theta_i))$.
- $a(\phi_i)$, normalmente formulada como $\frac{\phi}{w_i}$, donde el parámetro de dispersión se mantiene constante y los pesos varían para cada una de las n observaciones i .

Asumiendo lo visto, podemos reescribir la varianza como:

$$Var[y_i] = \frac{\phi V(\mu_i)}{w_i}$$

- de esta manera, si nuestro parámetro de dispersión ϕ fuese 1, los recíprocos de los pesos quedarían establecidos como parámetros de escala no constantes:

$$\frac{1}{w_i} = \phi_i.$$

Además de esto, el GLM también nos proporciona una función de ligadura $g(\cdot)$, que hace de enlace entre el predictor F (que es igual que $E[y_i]$ del modelo clásico de regresión) y la respuesta. Esta función de ligadura, que debe ser monótona y diferencial, se formula tal que:

$$g(E[y_i]) = g(\mu_i) = \eta_i = \beta_0 + \sum_{j=1}^P \beta_j f_{ij}.$$

Debemos de tener en cuenta que algunas distribuciones de la familia exponencial nos permiten trabajar con funciones de ligadura "naturales", llamadas canónicas. La característica especial de estas funciones de ligadura recae en el hecho de que este parámetro coincide con el predictor lineal:

$$\theta(\mu_i) = \eta_i.$$

En estos casos especiales, también tenemos la opción de trabajar con una función de ligadura derivada de la familia paramétrica:

$$g(\mu_i) = \eta_i = \begin{cases} \mu_i^\lambda & \text{para } \lambda \neq 0 \\ \log(\mu_i) & \text{para } \lambda = 0 \end{cases}$$

Como ya hemos dicho anteriormente, y siguiendo con la propuesta de *Boj y Costa (2014)*, el modelo GLM permite trabajar con diferentes distribuciones que no sean la normal. Dentro de la ya mencionada familia exponencial, nos encontramos con el siguiente conjunto de distribuciones, formado por sus consiguientes enlaces canónicos θ_i , así como el cálculo de su parámetro de dispersión ϕ_i .

	Gaussiana $N(\mu, \sigma^2)$	Binomial $B(m, \pi)/m$	Poisson $P(\mu)$	Gamma $G(\mu, v)$	Gaussiana Inversa $GI(\mu, \sigma^2)$
Rango (y)	$(-\infty, +\infty)$	$0, 1, \dots, m/m$	$0, 1, 2, \dots, \infty$	$(-\infty, +\infty)$	$(-\infty, +\infty)$
w	1	1	1	1	1
ϕ	σ^2	$1/m$	1	v^{-1}	σ^2
$b(\theta)$	$\theta^2/2$	$\log(1 + e^\theta)$	$\exp(\theta)$	$-\log(-\theta)$	$-(-2\theta)^{\frac{1}{2}}$
$c(y; \theta)$	$-\frac{1}{2} \left(\frac{y^2}{\phi} + \log(2\phi\pi) \right)$	$\log \binom{m}{my}$	$-\log y!$	$v \log(vy) - \log(y) - \log \Gamma(v)$	$-\frac{1}{2} \left(\log(2\pi\phi y^3) + \frac{1}{\phi y} \right)$
$\mu(\theta) = E(Y; \theta)$	θ	$\frac{e^\theta}{(1 + e^\theta)}$	$\exp(\theta)$	$\frac{-1}{\theta}$	$(-2\theta)^{-\frac{1}{2}}$
$\theta(\mu)$	μ	$\log(\mu/(1 - \mu))$	$\log(\mu)$	$\frac{1}{\mu}$	$\frac{1}{\mu^2}$
$V(\mu)$	1	$\mu(1 - \mu)$	μ	μ^2	μ^3

Tabla 13 Cuadro resumen familia exponencial [Fuente: *Boj y Costa (2014)*]

Dentro de estas distribuciones, cabe destacar la capacidad de la distribución Poisson a la hora de realizar los cálculos, pues esta distribución no permite resultados negativos, siendo esta característica especialmente útil.

Estimación Parámetros

Una vez hemos hecho una introducción al modelo en sí, tenemos que hablar de la estimación de los parámetros, empezando por las ya mencionadas β_j .

De la misma forma que ocurre en los modelos de regresión clásicos, estos parámetros se obtienen mediante funciones de máxima verosimilitud, y, para ser más exactos, la maximización del logaritmo de dicha función.

De esta manera, dicha fórmula quedará:

$$l = \sum_{i=1}^n l_i = \sum_{i=1}^n \left\{ \frac{y_i \theta_i - b(\theta_i)}{a(\phi_i)} + c(y_i, \phi_i) \right\}.$$

Cabe puntualizar que dicho cálculo será válido cuando trabajemos con distribuciones de la familia exponencial (las utilizadas en este trabajo), pero, en caso de no hacerlo, tendríamos que trabajar con la función de cuasi verosimilitud.

En dicha situación, nos encontramos con que sólo tendremos conocimiento de los dos primeros momentos ya vistos:

$$E[y_i] = \mu_i$$

y

$$Var[y_i] = \frac{\phi V(\mu_i)}{w_i}.$$

Por otro lado, la fórmula para estimar el parámetro β_j , quedará:

$$q(y; \mu) = \sum_{i=1}^n q_i = \sum_{i=1}^n w_i \int_{y_i}^{u_i} \frac{y_i - s}{\phi V(s)} ds$$

la cual, una vez resuelta, se resume en:

$$\sum_{i=1}^n w_i \frac{y_i - \mu_i}{\phi V(\mu_i)} \frac{d\mu_i}{d\beta_j} = 0.$$

Desviación y residuos

Cuando hablamos de modelos GLM, y una vez hemos parametrizado el modelo con todos sus componentes (función de enlace, distribución del error y el predictor utilizado), tenemos también que tener en cuenta la desviación del error del modelo, pues la minimización de esta nos permitirá obtener el modelo que mejor se adapte a nuestra muestra de datos.

Dicho error queda explicado en lo que se llama la *desviación escalada*, la cual queda formulada como:

$$D^*(y; \hat{\mu}) .$$

Esta *desviación*, a su vez, está explicada por la división entre dos funciones de verosimilitud, una en un estado normal y otra saturada:

$$D^*(y; \hat{\mu}) = -2 \log \left(\frac{L(y; \hat{\mu})}{L(y; y)} \right)$$

siendo:

- $L(y; \hat{\mu})$ la función de verosimilitud del modelo normal y,
- $L(y; y)$ la función de verosimilitud del modelo saturado.

Dicha *desviación* se comporta de tal manera que, a mayor número de predictores del modelo, menor es esta. En muchas ocasiones, y para poder trabajar con las diferentes distribuciones de las familias exponenciales, trabajaremos con las llamadas *desviaciones no escaladas*, que no son más que una variación de la *desviación* ya vista:

$$D(y; \hat{\mu}) = \phi D^*(y; \hat{\mu}) .$$

Las diferentes *desviaciones no escaladas*, en función de la familia exponencial escogida, quedan reflejadas en la siguiente tabla:

Distribución	Desviaciones D
Gaussiana $N(\mu, \sigma^2)$	$\sum_{i=1}^n (y_i - \hat{\mu}_i)^2$
Binomial $B(m, \pi)/m$	$2 \sum_{i=1}^n \left\{ y_i \log \left(\frac{y_i}{\hat{\mu}_i} \right) + (m - y_i) \log \left(\frac{m - y_i}{m - \hat{\mu}_i} \right) \right\}$
Poisson $P(\mu)$	$2 \sum_{i=1}^n \left\{ y_i \log \left(\frac{y_i}{\hat{\mu}_i} \right) - (y_i - \hat{\mu}_i) \right\}$

Gamma $G(\mu, \nu)$	$2 \sum_{i=1}^n \left\{ -\log \left(\frac{y_i}{\hat{\mu}_i} \right) + \frac{(y_i - \hat{\mu}_i)}{\hat{\mu}_i} \right\}$
Gaussiana Inversa $GI(\mu, \sigma^2)$	$\frac{\sum_{i=1}^n (y_i - \hat{\mu}_i)^2}{\hat{\mu}_i^2 y_i}$

Tabla 14 Desvianzas no escaladas familia exponencial [Fuente: Boj y Costa (2014)]

De igual manera que hemos visto anteriormente, en caso de trabajar con distribuciones que no pertenezcan a dicha familia, tendremos que trabajar con las funciones de cuasi-verosimilitud, quedando la *desvianza no escalada*:

$$D(y; \hat{\mu}) = \sum_{i=1}^n d_i = \sum_{i=1}^n 2w_i \frac{y_i - s}{V(s)} d(s) = -2\phi q(y; \hat{\mu}).$$

Cabe destacar que para el cálculo de estas *desvianzas* será necesario poder conocer los dos primeros momentos.

Una vez sabemos cómo calcular la desviación, y como ya hacemos en los modelos de regresión lineal clásicos, será importante analizar adecuadamente el residuo, para poder saber si los elementos escogidos en el modelo son los correctos o si por ende, nuestro modelo se podría adaptar mejor a otro tipo de distribución.

En los modelos GLM, existen distintos tipos de residuos:

- Por un lado, tenemos los residuos de Pearson:

$$r_i^P = \frac{y_i - \hat{\mu}_i}{\sqrt{\frac{V(\hat{\mu}_i)}{w_i}}}$$

donde r_i^P suele ser un residuo poco recomendado para distribuciones distintas a la normal (con las cuales trabajamos nosotros), pues puede ser bastante sesgado en estas situaciones.

De los residuos de Pearson también aparece una variación, los residuos de Anscombe, que pretenden solucionar el problema mencionado, intentando normalizar la distribución y poder trabajar con los residuos de Pearson en aquellas situaciones donde este suele dar problemas.

- Por otro lado, tenemos los residuos de la *desvianza*:

$$r_i^D = \text{signo}(y_i - \hat{\mu}_i) \sqrt{d_i}$$

donde:

- d_i es la i -ésima componente de la *desviación no escalada* cuando trabajamos con funciones de cuasi-verosimilitud.

A todo ello, debemos tener en cuenta, que, para el cálculo del parámetro de dispersión ϕ , influyente en ambas *desviaciones*, si nos encontramos con un modelo que tenga $P+1$ coeficientes, podremos calcular su estimación como:

$$\hat{\phi}^D = \frac{1}{n-P-1} \sum_{i=1}^n (r_i^D)^2 = \frac{1}{n-P-1} \sum_{i=1}^n d_i$$

- siendo r_i^D el residuo de las desviaciones.

O por el contrario, podremos utilizar los residuos bajo el método de Pearson:

$$\hat{\phi}^P = \frac{1}{n-P-1} \sum_{i=1}^n (r_i^D)^2 = \frac{1}{n-P-1} \sum_{i=1}^n w_i \frac{(y_i - \hat{\mu}_i)^2}{V(\hat{\mu}_i)}.$$

Función de enlace “Identidad” y “Logarítmica”

Siguiendo con la teoría propuesta por *Boj y Costa (2014)*, y originaria de *McCullagh y Nelder (1989)*, y recordando la fórmula de la función de enlace anteriormente mencionada, diremos que el modelo será multiplicativo o aditivo en función de la tipología utilizada de esta función enlace.

De esta manera, si consideramos una función de enlace “Identidad”, estaremos entonces trabajando con un modelo aditivo y, por ende, la fórmula utilizada será la siguiente:

$$\mu_i = \eta_i = \beta_0 + \sum_{j=1}^P \beta_j f_{ij}.$$

Si deseamos trabajar con un modelo multiplicativo, tendremos que trabajar con el logaritmo de la función, quedando la fórmula como:

$$\log(\mu_i) = \eta_i = \beta_0 + \sum_{j=1}^P \beta_j f_{ij}.$$

Por lo que, si queremos obtener μ_i tendremos que trabajar con la multiplicación de exponenciales:

$$\mu_i = \exp\left(\beta_0 + \sum_{j=1}^P \beta_j f_{ij}\right) = e^{\beta_0} \cdot e^{\beta_1 f_{i1}} \cdot e^{\beta_2 f_{i2}} \cdot \dots \cdot e^{\beta_P f_{iP}} = \prod_{j=1}^P e^{\beta_j f_{ij}}.$$

En caso de trabajar con variables de carácter binario, solo introduciremos las exponenciales cuando el factor f_{ij} , sea verdadero e igual a 1.

Familias paramétricas en el error

De igual forma que hemos visto anteriormente, podremos trabajar con distribuciones pertenecientes a la familia paramétrica en el error del modelo:

$$V(\mu_i) = \mu_i^\zeta.$$

Estas distribuciones, también nos permiten obtener resultados obtenidos por las familias exponenciales en alguno de sus casos particulares:

- Si $\zeta = 1$, la estructura de error es Poisson
- Si $\zeta = 2$, la estructura de error es Gamma
- Si $\zeta = 3$, la estructura de error es Inversa Gaussiana.

CAPÍTULO 4: APLICACIÓN CON SHINY

La aplicación resultante de este trabajo puede encontrarse en la siguiente ubicación:

<https://azaelmm.com/shiny/calnovida/>

4.1 Conceptos básicos

Antes de empezar con la explicación de la aplicación en Shiny, así como de los métodos utilizados, es importante conocer las bases sobre Shiny. Como se explica en la página oficial (*Chang et al., 2021*), Shiny es un paquete de R que permite crear aplicaciones interactivas basadas en el lenguaje R.

Uno de los grandes problemas de R como lenguaje siempre ha sido su pobre capacidad a la hora de graficar y presentar los datos. Si bien es cierto que la capacidad de cálculo de R como lenguaje es muy potente, con el paquete de Shiny se fortalece el gran punto débil del lenguaje, haciendo también lo propio con el RStudio (*Rstudio team, 2020*), la herramienta más utilizada para trabajar con R.

Con Shiny, además, podremos incorporar otros lenguajes como pueden ser HTML, JavaScript o CSS, lenguajes utilizados principalmente en las páginas web y que nos permiten diseñar aplicaciones que muestren los *outputs* proporcionados por R, de una forma mucho más visual.

Una vez sabido esto, debemos conocer los conceptos básicos de dicho paquete. Shiny se compone principalmente de dos partes:

- La *ui*, que es la interfaz y la parte compuesta por todos aquellos objetos con los que se puede interactuar, los cuales a su vez permiten al usuario modificar el *output* de la aplicación.
- Por otro lado, tenemos el servidor (*server*), que es la parte compuesta por todas aquellas funciones que se ejecutan en segundo plano (no son visibles), pero que nos permiten obtener todos aquellos resultados que luego el usuario podrá ver en su aplicativo.

A estos dos componentes, además, debemos añadir un tercer y crucial componente: la función llamada *shinyApp*, que nos permite interrelacionar los dos componentes ya mencionados. De esta forma, todos aquellos *inputs* proporcionados por el usuario pueden ser convertidos en variables, que a su vez se ejecutarán en las funciones del servidor, para finalmente obtener un resultado visible en la propia aplicación.

De esta forma, conseguimos lo que pretende el paquete, una aplicación interactiva, dónde un usuario, que quizás no tenga conocimientos en el campo, pueda obtener aquellos *outputs* deseados.

En esta última reflexión recae uno de los objetivos de este trabajo, proponer un aplicativo lo suficientemente fácil como para que todos aquellos con pocos o ningún conocimiento

actuarial puedan obtener resultados. Todo ello, adaptando el aplicativo a los problemas expuestos anteriormente, trabajar con triángulos que dispongan de números distintos de filas y columnas.

4.2 Inputs

Dentro de la interfaz *ui*, Shiny (*Chang et al., 2021*), proporciona unas herramientas llamadas *control widgets* que nos permiten obtener información del usuario, con el objetivo de poder realizar la interacción deseada.

En la aplicación propuesta en este trabajo, los *inputs* se encuentran en un *sidebar panel*, lo que no es más que un panel localizado en la parte lateral (en nuestro caso izquierda) del aplicativo, dejando el *main panel*, o panel principal en castellano, como localización para mostrar los *outputs*.

Además, tenemos que sumar la inclusión del paquete de R *shinydashboards* (*Chang y Ribeiro, 2021*), que añade nuevos estilos a la interfaz de Shiny y que nos permite crear una aplicación muy visual.

De esta manera, la presentación del aplicativo quedará tal que:

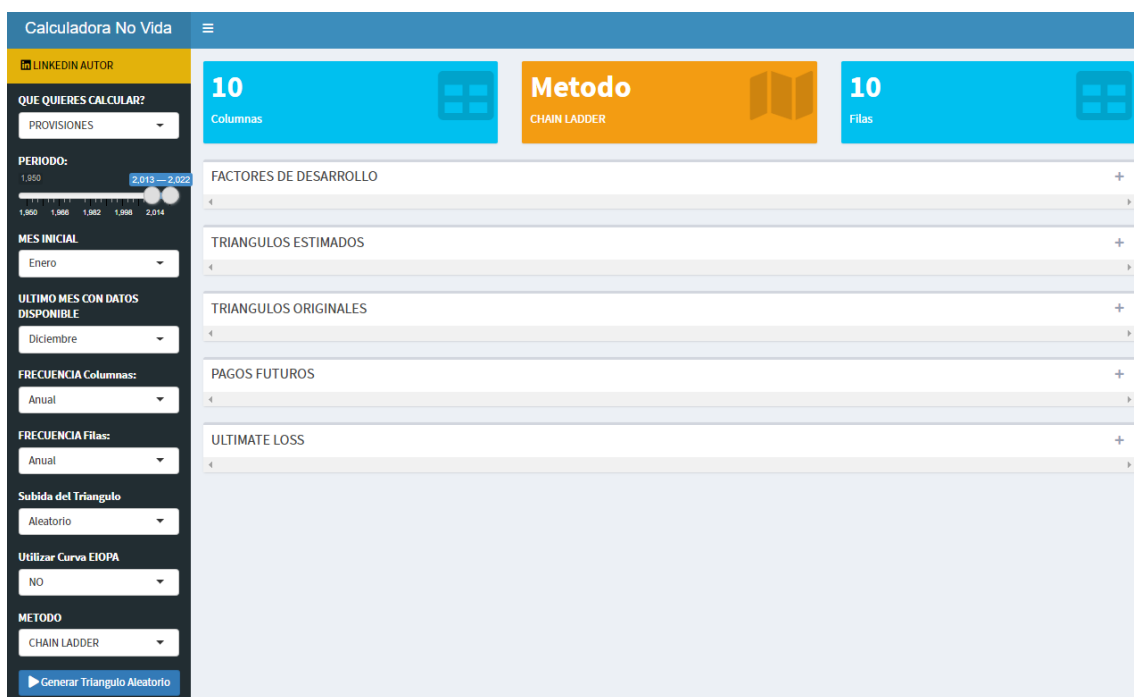


Ilustración 1. Interfaz aplicación Shiny [Fuente: Elaboración propia]

Siendo las opciones disponibles las siguientes:

- Tipo de cálculo
- Años triángulo
- Mes inicial
- Último mes
- Frecuencia filas y columnas
- Subida del triángulo

- Tipo de actualización flujos
- Método de cálculo
- Variaciones GLM.

Con el objetivo de conocer los tipos de *control widgets* disponibles en el paquete de Shiny (Chang et al., 2021), en las siguientes secciones haremos un repaso individualizado de cada una de las opciones, recordando además las problemáticas que pueden presentar algunas de ellas al trabajarse con triángulos *run-off* con diferente número de filas y columnas.

4.2.1 Tipos de cálculo

La primera opción que nos encontramos es un *selectInput* para seleccionar el tipo de cálculo que se desea realizar.

Dicho *widget* nos permite realizar un listado de opciones, para las cuales más adelante se pueden hacer paneles condicionales para que aparezcan funcionalidades específicas en función de los que se quiera calcular.

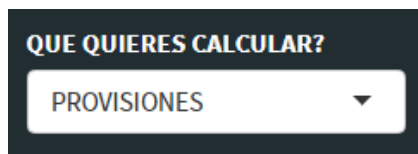


Ilustración 2. Selección cálculo aplicación Shiny [Fuente: Elaboración propia]

En nuestro caso, y debido a que se trata de un aplicativo pequeño, el listado solo tiene la opción de calcular “Provisiones”, aunque este desplegable está hecho para que cualquier persona, en caso de querer ampliar el aplicativo a otro tipo de cálculo actuarial, pueda hacerlo de una manera sencilla.

4.2.2 Años triángulo

El siguiente *input* disponible es un *sliderInput* que nos permite seleccionar el rango de años que posteriormente se utilizarán en los triángulos finales. Además, dicho *input* incorpora la función *Sys.Date()* como último año, por lo que, aunque pasen los años, el aplicativo siempre quedará actualizado.

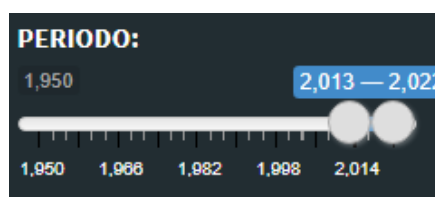


Ilustración 3. Selección años a trabajar. Aplicación Shiny [Fuente: Elaboración propia]

4.2.3 Mes inicial

Siguiendo con el listado, nos encontramos con otro *selectInput* que permite seleccionar el mes inicial de los periodos de origen i . Si bien es cierto que la mayoría de las compañías aseguradoras empezarán sus periodos de origen en enero, el mes que está puesto como predefinido, el aplicativo está programado para que aquellas compañías que quieran empezar sus periodos de origen i en otro momento puedan hacerlo.

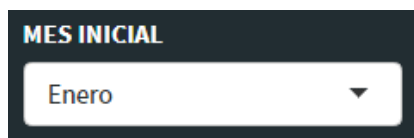


Ilustración 4. Selección mes inicial. Aplicación Shiny [Fuente: Elaboración propia]

De esta manera, los nombres de las filas quedarán predefinidos por ese *input* y por otro *input* que veremos más adelante, donde se especifica la periodicidad de las filas. En un ejemplo más sencillo, con años de origen i anuales, en función de si escogiéramos enero o julio, las diferencias serían las siguientes:

Mes inicial: enero	Años de desarrollo $j = 0, 1, \dots, k$		Mes inicial: julio	Años de desarrollo $j = 0, 1, \dots, k$	
Años de origen i	0	1	Años de origen i	0	1
01-13 (0)	$C_{0,0}$	$C_{0,1}$	07-13 (0)	$C_{0,0}$	$C_{0,1}$
01-14 (1)	$C_{1,0}$	$C_{1,1}$	07-14 (1)	$C_{1,0}$	$C_{1,1}$
...
01-20 ($k-2$)	$C_{k-2,0}$	$C_{k-2,1}$	07-20 ($k-2$)	$C_{k-2,0}$	$C_{k-2,1}$
01-21 ($k-1$)	$C_{k-1,0}$	$C_{k-1,1}$	07-21 ($k-1$)	$C_{k-1,0}$	$C_{k-1,1}$
01-22 (k)	$C_{k,0}$		07-22 (k)	$C_{k,0}$	

Tabla 15 Representación nombre filas. Aplicativo Shiny [Fuente: Elaboración propia]

4.2.4 Último mes

Seguimos con otro *selectInput*, que nos permite seleccionar el último mes en el cuál tenemos datos. Esta función es muy útil para aquellas entidades aseguradoras que, por ejemplo, en la mitad del año, quieran hacer un estudio preliminar de su siniestralidad en su ramo de no vida y que, por ende, solo dispongan de datos hasta junio de ese año.

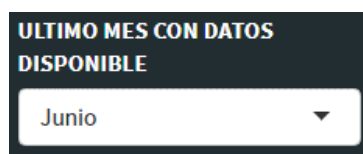


Ilustración 5 Selección último mes con datos. Aplicación Shiny [Fuente: Elaboración propia]

Al ser junio el último mes con datos disponibles de la compañía, y siguiendo con el triángulo de ejemplo utilizado en el punto 2.3, en caso de trabajar con trimestres, el triángulo con el que se debería trabajar sería:

Años de origen i	Trimestres de desarrollo $j = 0, 1, \dots, k$									
	0	1	2	3	4	5	6	7	8	9
0	$C(0;0)$	$C(0;1)$	$C(0;2)$	$C(0;3)$	$C(0;4)$	$C(0;5)$	$C(0;6)$	$C(0;7)$	$C(0;8)$	$C(0;9)$
1	$C(1;0)$	$C(1;1)$	$C(1;2)$	$C(1;3)$	$C(1;4)$	$C(1;5)$				
2	$C(2;0)$	$C(2;1)$								

Tabla 16 Ejemplo triángulo run-off trimestral en función del último mes disponible [Fuente: Elaboración propia]

y no este:

Años de origen i	Trimestres de desarrollo $j = 0, 1, \dots, k$											
	0	1	2	3	4	5	6	7	8	9	10	11
0	$C(0;0)$	$C(0;1)$	$C(0;2)$	$C(0;3)$	$C(0;4)$	$C(0;5)$	$C(0;6)$	$C(0;7)$	$C(0;8)$	$C(0;9)$	$C(0;10)$	$C(0;11)$
1	$C(1;0)$	$C(1;1)$	$C(1;2)$	$C(1;3)$	$C(1;4)$	$C(1;5)$	$C(1;6)$	$C(1;7)$				
2	$C(2;0)$	$C(2;1)$	$C(2;2)$	$C(2;3)$								

Tabla 17 Ejemplo 2. triángulo run-off trimestral en función del último mes disponible [Fuente: Elaboración propia]

Los últimos dos trimestres de cada año de origen i quedarán inhabilitados, pues no existirán datos de ese último semestre del año, ya que nuestro estudio se estará realizando seis meses después del último año de origen i .

Ilustración 6. Output Aplicación Shiny para triángulo trimestral [Fuente: Elaboración propia]

Todo y con esto, en este ejemplo se observa que, al estar trabajando con 2022 como último año y junio como último mes con datos disponibles, el código entenderá que la última fecha disponible será el 06/2022 (incluido), cosa que, a fecha de escritura de este trabajo, no existe. De esta forma debe tenerse en cuenta que el aplicativo, como cualquier otro, requerirá de la correcta introducción de datos por parte del usuario.

Más adelante entraremos más en detalle respecto a esta característica del aplicativo y como está programada, pues puede volverse aún más compleja.

4.2.5 Frecuencia filas y columnas

Siguiendo con el listado de los *inputs* del aplicativo, nos encontramos con otros dos *selectInput*, esta vez para seleccionar la frecuencia, tanto de los periodos de desarrollo j (frecuencia columnas), como de los periodos de origen i (frecuencia filas).

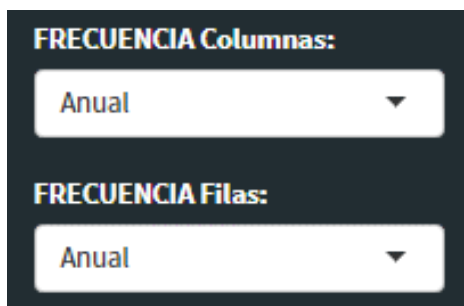


Ilustración 7 Frecuencias filas y columnas. Aplicación Shiny [Fuente: Elaboración propia]

Dichas opciones nos permitirán solucionar los problemas cuestionados en este trabajo, relativos a las posibilidades que nos abren los triángulos que no disponen del mismo número de filas y columnas.

Es importante destacar esa opción del aplicativo, pues la estructura del código depende mucho de estos parámetros, así como de las dos fechas comentadas con anterioridad.

Además, es importante recalcar la lógica que sigue el código respecto a estas variables, ya que son imprescindibles para que este sea capaz de saber qué parte del triángulo es conocida y cuál quedará como la estimación.

Para realizar este fragmento del código mencionado, será de especial relevancia el paquete de R *lubridate* (Spinu et al., 2021). Este paquete tiene la principal funcionalidad de permitirnos trabajar cómodamente con fechas, desde el formato de estas, hasta la capacidad de realizar operaciones con ellas, razón principal de su uso.

De esta manera, una vez tengamos la fecha inicial, esta servirá para determinar el número de las filas. Esta fecha inicial, juntamente con la fecha final, servirá para determinar el número de columnas.

Para mostrarlo de una manera gráfica, la forma de obtener el número de columnas del triángulo con las opciones especificadas constará de tres *inputs* protagonistas:

- Fecha inicial (que será el año del primer *input* visto + el mes seleccionado como inicial); si fuese, por ejemplo, julio y 2015, entonces la primera fecha y, por ende, el nombre del primer periodo de origen *i* (filas) sería 07-15.
- De igual forma con la fecha final, pero con el mes y años de los *inputs* relativos a esta cuestión.
- Y finalmente, la frecuencia de las columnas, si es anual, será igual a 1; si es semestral, será igual a 2; y así sucesivamente con el resto de las opciones.

Una vez tenemos estos datos, y mediante el paquete de R *lubridate* (Spinu et al., 2021), se ejecutará una función que calcula hasta qué periodos de desarrollo *j* tendremos información disponible, para cada uno de los periodos de origen *i* establecidos.

4.2.6 Subida del triángulo

Volviendo al listado, el siguiente *input*, de la misma tipología que los anteriores, nos permite escoger la opción en cuanto a la subida del triángulo. Con el objetivo de poder utilizar el aplicativo no sólo a nivel profesional, sino también a nivel teórico, las opciones disponibles son dos: un triángulo aleatorio y la opción de subir un archivo.

Si se opta por esta segunda opción, aparecerá un *fileInput*, que permitirá al usuario subir un archivo .csv con el triángulo *run-off* que desee trabajar.

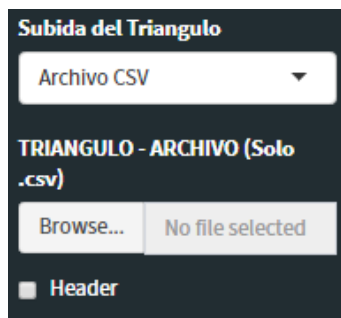


Ilustración 8 Subida del triángulo. Aplicación Shiny [Fuente: Elaboración propia]

El usuario tendrá que procurar que el formato del archivo .csv sea el adecuado. El aplicativo está reforzado con controles que darán error en caso de subir un archivo con el formato erróneo. El aplicativo sólo permite subir triángulos *run-off* no acumulados.

Por otro lado, también se debe tener en cuenta que tendremos que subir un archivo con un triángulo no acumulado que pueda trabajarse con las opciones escogidas anteriormente. De esta manera, dentro del *output* y en función de las opciones escogidas, el aplicativo nos irá indicando las filas y columnas del triángulo de datos (donde irá el triángulo *run-off*) a trabajar.

Así, queda en la responsabilidad del usuario que introduce el triángulo *run-off*, escoger las opciones adecuadas en el aplicativo para poder trabajar con el mismo.

4.2.7 Tipo de actualización flujos

Seguidamente, nos encontraremos con una opción que nos permite decidir entre tres tipologías de tipos de interés para la actualización de flujos:

- No utilizar interés (Ninguno)
- Tipo fijo
- Curva de tipos, *EIOPA (2022)*.

Esta opción está relacionada con los pagos futuros por año natural y, más concretamente, con el tipo de interés utilizado para la actualización de dichos flujos.

Pero ¿qué son los pagos futuros por año natural? Si bien es cierto, que en capítulos anteriores hemos indagado en la teoría relativa a los triángulos *run-off*, no hemos entrado al detalle en cuanto a este concepto.

Como ya hemos visto en múltiples ocasiones a lo largo de este trabajo, y volviendo al triángulo que hemos ido utilizando para la gran mayoría de ejemplos, normalmente nos centramos en los periodos, en este caso años, tanto de desarrollo j como de origen i , pero las diagonales también juegan un papel importante.

Años de origen i	Años de desarrollo $j = 0, 1, \dots, k$					
	0	1	2	\dots	$k-1$	k
0	$C_{0,0}$	$C_{0,1}$	$C_{0,2}$	\dots	$C_{0,k-1}$	$C_{0,k}$
1	$C_{1,0}$	$C_{1,1}$	$C_{1,2}$	\dots	$C_{1,k-1}$	
\dots	\dots	\dots	\dots	\dots		
$k-2$	$C_{k-2,0}$	$C_{k-2,1}$	$C_{k-2,2}$			
$k-1$	$C_{k-1,0}$	$C_{k-1,1}$				
k	$C_{k,0}$					

Tabla 18 Actualización flujos. [Fuente: Elaboración propia]

Si nos fijamos en el triángulo, veremos que para un siniestro ocurrido en el año de origen $k-1$ pero pagado en el año de desarrollo $j=2$, a efectos de año natural, dicho pago se ha realizado en $k-1+2$, o lo que es lo mismo, en el año $k+1$.

Si nos vamos al año de origen k , pero esta vez al año de desarrollo $j=1$, obtendremos exactamente el mismo año natural que antes.

Si seguimos con el mismo proceso, veremos como las diagonales nos permiten obtener los flujos de pagos de los años naturales (en el gráfico, simplificado como AN).

Años de origen i	Años de desarrollo $j = 0, 1, \dots, k$					
	0	1	2	\dots	$k-1$	k
0	$C_{0,0}$	$C_{0,1}$	$C_{0,2}$	\dots	$C_{0,k-1}$	$C_{0,k}$
1	$C_{1,0}$	$C_{1,1}$	$C_{1,2}$	\dots	$C_{1,k-1}$	AN1
\dots	\dots	\dots	\dots	\dots	AN1	AN2

$k-2$	$C_{k-2,0}$	$C_{k-2,1}$	$C_{k-2,2}$	AN1	AN2	AN3
$k-1$	$C_{k-1,0}$	$C_{k-1,1}$	AN1	AN2	AN3	AN4
k	$C_{k,0}$	AN1	AN2	AN3	AN4	AN5

Tabla 19 Flujos por años naturales. [Fuente: Elaboración propia]

De esta forma, y si juntamos los flujos en un gráfico de tiempo, tendremos:

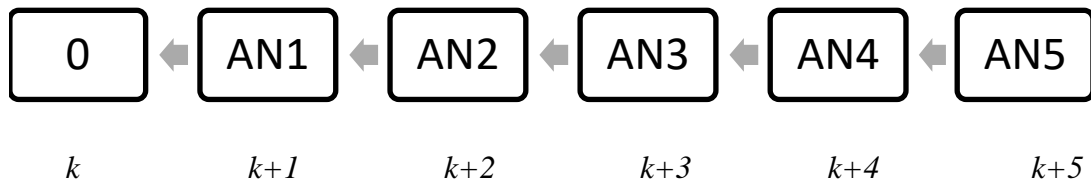


Ilustración 9 Representación actualización flujos. [Fuente: Elaboración propia]

donde cada flujo, al estar en un momento del tiempo distinto, podrá ser actualizado (en función de lo que decida el usuario en esta opción) al tipo de interés correspondiente.

Para poder actualizar dichos flujos a un tipo concreto, en el aplicativo aparecerá entonces un *Input*, donde podrá elegir entre establecer un tipo de interés fijo o bien, mediante la subida de un archivo .csv, utilizar la curva de tipos EIOPA.

Además, también aparecerá otra opción, la tipología de la actualización:

- Renta
- Flujo.

Actualización como renta

La primera opción, nos realiza la habitual actualización de flujos que en la gran mayoría de ocasiones se utiliza en el ámbito financiero.

Como hemos ido haciendo a lo largo de este trabajo, haremos un ejemplo para explicarlo.

Supongamos un tipo fijo del 2% y el mismo triángulo que acabamos de ver en la página anterior. La actualización sería de la siguiente manera:

$$(1 + 0,02)^{-i}$$

- donde i , serán los años transcurridos desde k . En $k+2$, serán $k+2-k$, o sea, 2 años.

Para obtener la suma de los pagos futuros, lo formularíamos de la siguiente manera:

$$\sum_{i=1}^5 AN_i \cdot (1 + 0,02)^{-i} .$$

En caso de disponer de una curva EIOPA y no de un tipo fijo lo formularíamos como:

$$\sum_{i=1}^5 AN_i \cdot (1 + EIOPA_i)^{-i}.$$

Como veremos más adelante, la obtención de $EIOPA_i$ tiene ciertas peculiaridades.

Actualización como flujo

La actualización de flujos difiere de la actualización de rentas, ya que se considera que los flujos de pagos serán continuos a lo largo del año en lugar de un pago al final de cada año como ocurre en el caso de las rentas. De esta manera, a partir de logaritmos primero se calcula el valor de cada flujo anual al inicio de ese año y después se actualiza dicho valor hasta el momento de cálculo de la provisión.

Siguiendo con el ejemplo utilizado para las rentas, cuando el tipo de interés sea un 2% anual, obtendríamos la siguiente fórmula:

$$\sum_{i=1}^5 AN_i \cdot \frac{1 - (1 + 0,02)^{-1}}{\log(1 + 0,02)} \cdot (1 + 0,02)^{-(i-1)}$$

En cambio, si disponemos de una curva EIOPA el proceso de cálculo sería distinto, ya que necesitaríamos disponer del tipo forward para cada uno de los años considerados para obtener el valor del flujo continuo al inicio de cada año y después utilizar el tipo de interés de la curva para actualizar dicho valor hasta el momento de cálculo de la provisión.

Los pasos a seguir serán:

- Primero tendremos que obtener la curva publicada por *EIOPA (2022)*, con los tipos sin riesgo para cada plazo considerado:

Años	Tipo interés anual
1	i_1
2	i_2
3	i_3
4	i_4

Tabla 20 Tipos de interés anuales. [Fuente: Elaboración propia]

- Después, siguiendo la propuesta de *Wai-Sum y Yiu-Kuen (2017)*, tendremos que convertir dicha curva, spot, en una de tipo forward. Para ello, utilizaremos las siguientes fórmulas:

Años	Tipo interés forward
1	$tf_1 = i_1$
2	$tf_2 = \frac{(1 + i_2)^2}{(1 + i_1)^1} - 1$
3	$tf_3 = \frac{(1 + i_3)^3}{(1 + i_2)^2} - 1$
4	$tf_4 = \frac{(1 + i_4)^4}{(1 + i_3)^3} - 1$

Tabla 21 Cambio tipo spot a tipo forward. [Fuente: Elaboración propia]

Finalmente, la fórmula a aplicar será:

$$\sum_{i=1}^5 AN_i \cdot \frac{1 - (1 + tf_i)^{-1}}{\log(1 + tf_i)} \cdot (1 + EIOPA_{i-1})^{-(i-1)}.$$

Curva EIOPA para triángulos no anuales

Como ya hemos visto a lo largo de las anteriores secciones, este trabajo se centra en la problemática derivada de la existencia de triángulos de datos con diferente número de filas y columnas.

En el caso de las actualizaciones de los pagos futuros, la variable que nos afecta es la frecuencia de las columnas, pues la disposición de la diagonal cambiará totalmente, además de que la curva EIOPA, que suele ser anual, no podrá ser aplicada como tal, sino que se tendrá que modificar.

Cuando trabajamos con un triángulo de datos que tiene el mismo número de filas y columnas, las cuantías por años naturales están dispuestas en las diagonales. En cambio, si la frecuencia de las columnas y de las filas es distinta, en las diagonales no tendremos cuantías que correspondan al mismo año natural o periodo.

Si volvemos al triángulo trimestral que hemos ido viendo a lo largo del trabajo, veremos como la diagonal es completamente diferente.

Años de origen i	Trimestres de desarrollo $j = 0, 1, \dots, k$											
	0	1	2	3	4	5	6	7	8	9	10	11
0	$C(0;0)$	$C(0;1)$	$C(0;2)$	$C(0;3)$	$C(0;4)$	$C(0;5)$	$C(0;6)$	$C(0;7)$	$C(0;8)$	$C(0;9)$	$C(0;10)$	$C(0;11)$
1	$C(1;0)$	$C(1;1)$	$C(1;2)$	$C(1;3)$	$C(1;4)$	$C(1;5)$	$C(1;6)$	$C(1;7)$	TN1	TN2	TN3	TN4
2	$C(2;0)$	$C(2;1)$	$C(2;2)$	$C(2;3)$	TN1	TN2	TN3	TN4	TN5	TN6	TN7	TN8

Tabla 22 Flujos para periodos naturales no anuales. [Fuente: Elaboración propia]

Además, pese a trabajar con un menor número de años, como los pagos son trimestrales, existe un mayor número de flujos futuros, siendo ocho los trimestres que cubren los dos años desconocidos (véase en la *Tabla 22*, el paso de AN , años naturales, a TN , trimestres naturales).

Una vez conocida la primera problemática, que en el aplicativo se soluciona mediante la búsqueda de la i estimación de cada fila para cada AN_i , pasamos a la siguiente cuestión, los tipos de interés no anuales.

Siguiendo con el ejemplo trimestral, por lo general la actualización se realiza siempre al último dato conocido, pero este, a diferencia del triángulo anual, no es el año anterior, sino el trimestre anterior.

Tanto la curva de tipos de EIOPA como un tipo de interés fijo, siempre serán tantos anuales. Lo que implica que, en caso de trabajar con triángulos que tengan frecuencias en sus periodos de desarrollo j distintos al anual, se tendrán que adaptar esos tipos de interés, calculando los tantos de interés equivalentes con la frecuencia elegida.

Así, si tenemos un tanto de interés fijo del 2% anual, pero trimestres de desarrollo j , se utilizará el tanto de interés trimestral equivalente, entonces:

$$(1 + ((1 + 0,02)^{\frac{1}{4}} - 1))^{-i}$$

En el caso de trabajar con la curva EIOPA, una vez calculados los tipos forward para cada año, los pasos a seguir serían:

- Pasar el tipo de interés forward a la recurrencia deseada, en nuestro caso trimestral

$$tft = (1 + tf)^{\frac{1}{4}} - 1$$

- Finalmente, tendremos que obtener los factores de descuento de los doce trimestres tal que:

Trimestres	Factor de descuento
1	$1/(1 + tft_1)$
2	$1/((1 + tft_1) \cdot (1 + tft_2))$
...	...
12	$1/((1 + tft_1) \cdot (1 + tft_2) \cdot \dots \cdot (1 + tft_{11}) \cdot (1 + tft_{12}))$

Tabla 23 Cálculo factor de descuento con tipos forward. [Fuente: Elaboración propia]

- Estos factores de descuento se podrán multiplicar por cada uno de los flujos de caja disponibles y obtener los flujos actualizados a la curva publicada por EIOPA (2022).

4.2.8 Método de cálculo

Retomando la lista se encuentra el *input* que nos permite seleccionar el método a utilizar para la resolución del triángulo propuesto y por ende las provisiones calculadas.

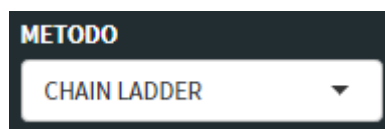


Ilustración 10 Selección método cálculo provisiones. Aplicación Shiny [Fuente: Elaboración propia]

Como recordatorio del capítulo 3.3, este trabajo se ha centrado en los siguientes métodos:

Métodos deterministas:

- Chain-ladder.
- Variación Chain-ladder con ponderación de pesos $i+j+1$.
- Mínimos cuadrados de Vyllder.

Métodos estocásticos:

- Modelo lineal generalizado, GLM.

Además de todo esto, en la opción del triángulo aleatorio, será necesario clicar un botón para que se ejecuten los cálculos y poder obtener el *output* con las opciones escogidas.

Esta última opción se realiza con el objetivo de evitar que, para cada cambio realizado en cualquiera de las opciones anteriormente comentadas, R ejecute de nuevo todo el cálculo cuando probablemente todas las opciones no se hayan seleccionado.

Es importante matizar que, si no se clica dicho botón, debido a los cambios en los triángulos pueden aparecer errores si se ha ejecutado el código con anterioridad, pues el aplicativo intentará adaptar los cálculos ya realizados para las nuevas opciones seleccionadas (algo que en la mayoría de las ocasiones provocará errores).

4.2.9 Variaciones GLM

Por otro lado, finalmente, en caso de escoger el modelo GLM como metodología de cálculo, el aplicativo nos permitirá seleccionar las *desviaciones no escaladas* (más detalles en el capítulo 3.3.4) y, por ende, la familia exponencial con la que queramos trabajar.



Ilustración 11 Selección desviación escalada GLM. Aplicación Shiny [Fuente: Elaboración propia]

En el caso de este trabajo, las distribuciones disponibles son:

- Poisson
- Gamma
- Gaussiana inversa.

Los cálculos relativo al modelo GLM se han realizado con el soporte de la función *glmReserve*, dentro del paquete *ChainLadder* (Gesmann et al., 2022).

4.3 Outputs

Una vez vistos tanto los conceptos como los *inputs*, también es importante conocer, de una manera resumida, los *outputs* que proporciona el aplicativo y que no hacen más que mostrar los resultados obtenidos por los cálculos de los métodos mencionados anteriormente.

De esta forma, y en función del método escogido, nos podremos encontrar con una combinación distinta (o su elusión) de los siguientes resultados:

- Breve resumen de las opciones escogidas
- Factores de desarrollo (Chain-ladder y variante Chain-ladder)
- Triángulos estimados
- Triángulos originales
- Pagos futuros
- Ultimate loss
- Factores de desarrollo individuales (variante Chain-ladder).

4.3.1 Resumen

Lo primero con lo que nos encontraremos en la sección del *output* del aplicativo son tres cajas con una breve información sobre el triángulo que se estimará. Si bien en el centro se encuentra el método escogido, la parte clave de este resumen recae en el número de columnas y filas del triángulo en función de las opciones escogidas por el usuario.

Esta opción es muy útil cuando el usuario pretende subir un triángulo y no sabe si las opciones escogidas son las adecuadas. Con este resumen, el usuario podrá intuir si realmente está escogiendo las opciones correctas para la estimación del triángulo deseado.



Ilustración 12 Cuadro resumen. Aplicación Shiny [Fuente: Elaboración propia]

Estas cajas están creadas a partir de un *widget* llamado *valueBoxOutput*, disponible dentro del paquete *shinydashboard* (Chang y Ribeiro, 2021), mencionado con anterioridad en la parte de los *inputs* y clave en el diseño del aplicativo.

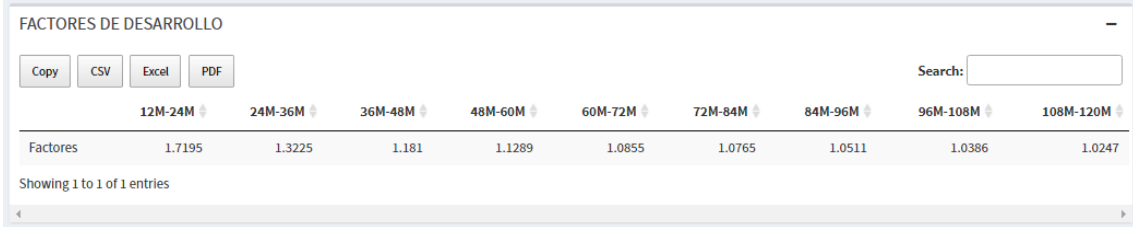
4.3.2 Factores de desarrollo

Una vez pasado el resumen ya mencionado, lo siguiente con lo que nos encontraremos es un primer desplegable del aplicativo. Con el objetivo de evitar un exceso de resultados, el aplicativo se ha hecho con cajas individuales desplegables para cada uno de los

resultados, por lo que el usuario podrá decidir si visualizarlas o no en función de los resultados buscados.

Estos desplegados son otro tipo de *outputs* disponibles en el paquete *shinydashboard* (Chang y Ribeiro, 2021) llamados *box*.

Dentro de esta caja encontraremos la tabla formada por los factores de desarrollo de la siguiente manera:



	12M-24M	24M-36M	36M-48M	48M-60M	60M-72M	72M-84M	84M-96M	96M-108M	108M-120M
Factores	1.7195	1.3225	1.181	1.1289	1.0855	1.0765	1.0511	1.0386	1.0247

Ilustración 13 Desplegable con la tabla formada por los factores de desarrollo. Aplicación Shiny [Fuente: Elaboración propia]

Esta tabla, igual que el resto de las que veremos a lo largo de esta sección, está creada a partir del paquete de R *DT* (Xie et al., 2022), que nos permite obtener tablas con un mejor diseño, así como dotar a las mismas de opciones adicionales, como son las que se pueden ver en la ilustración.

Además, en caso de desearse, el usuario podrá obtener la tabla resultante en un Excel o pdf, listo para poder ser presentado en otro tipo de plataforma o software.

4.3.3 Triángulos estimados y originales

De una forma similar a la comentada en el apartado anterior, en este caso nos encontramos con un desplegable en el que se encuentran los dos triángulos estimados: el acumulado y el no acumulado.

Dentro de este triángulo también se encuentra el ya explicado *ultimate loss*, que también forma parte del listado de *outputs* y que se encuentra en un gráfico de barras que veremos un poco más adelante.

TRIANGULOS ESTIMADOS

DESACUMULADO ACUMULADO

Copy CSV Excel PDF Search:

	12M	24M	36M	48M	60M	72M	84M	96M	108M	120M
01-13	811478	1218259	1651228	1991417	2271396	2485386	2595782	2704352	2782794	2851555
01-14	853329	1786623	2303751	2784284	3197600	3471818	3750603	3987182	4167322	4270294
01-15	876612	1633099	2103088	2463529	2673385	2810518	2943782	3073795	3192576	3271463
01-16	693767	1079292	1370778	1672150	1965032	2263745	2585020	2717235	2822238	2891974
01-17	1110658	1670590	2203314	2644711	2920074	3109815	3347663	3518885	3654866	3745175
01-18	667504	1236067	1798109	2085950	2373112	2576003	2773024	2914855	3027494	3102301
01-19	1069844	2019379	2598665	2925973	3303152	3585558	3859792	4057208	4213991	4318116
01-20	747809	1244370	1692099	1998350	2255952	2448826	2636120	2770949	2878027	2949141
01-21	796755	1228036	1624034	1917966	2165206	2350322	2530081	2659487	2762258	2830511
01-22	811478	1395314	1845253	2179223	2460141	2670473	2874718	3021751	3138521	3216072

Showing 1 to 10 of 10 entries

Ilustración 14 Triángulo estimado acumulado. Aplicación Shiny [Fuente: Elaboración propia]

Como se puede apreciar en la figura, en ambos triángulos se puede distinguir de una forma sencilla aquella información conocida (celdas naranjas), de aquella que está estimada y que, por ende, se ha obtenido mediante el método escogido.

El siguiente desplegable, como su nombre indica, es exactamente igual que este, pero con el triángulo formado exclusivamente por la información conocida.

Esta opción está disponible principalmente por si se desea utilizar el triángulo en algún otro lugar y que este sea fácilmente descargable.

4.3.4 Pagos futuros

El siguiente *output* que nos encontramos es un gráfico obtenido a partir de la combinación de los paquetes de R *apexcharter* (Perrier et al., 2022) y *ggplot2* (Wickham et al., 2021). El paquete de R *apexcharter* nos permite crear gráficos mucho más completos que los habituales de R, además de tener opciones adicionales.

De esta manera, en este nuevo desplegable nos encontramos con un gráfico de barras que contiene los pagos futuros ya actualizados. Estos pagos futuros se muestran por años naturales y corresponden al resultado de los cálculos explicados en el apartado 4.2.7.

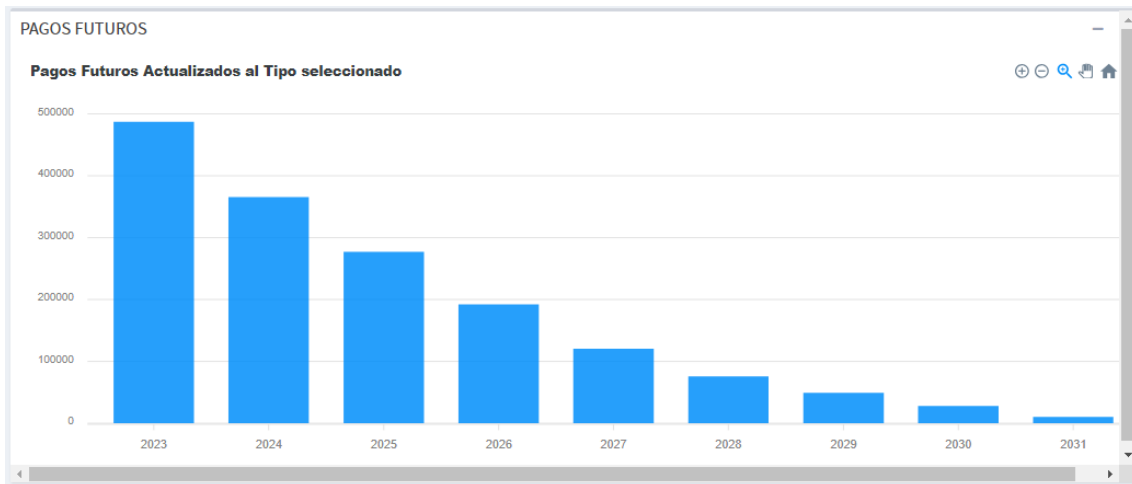


Ilustración 15 Gráfico de pagos futuros. Aplicación Shiny [Fuente: Elaboración propia]

Gracias a las opciones disponibles por el paquete mencionado, si desplazamos el ratón por encima de cada barra podremos ver los datos exactos para cada uno de ellos.

4.3.5 Ultimate loss

Continuando con los desplegados, el siguiente, de idénticas características al anterior, solo difiere de aquel respecto a los datos mostrados. Como ya hemos mencionado con anterioridad este gráfico solo muestra la última columna del triángulo de datos estimado, conocido como *ultimate loss*.

4.3.6 Factores de desarrollo individuales

Finalmente, y en el caso exclusivo de la variante de Chain-ladder, se encuentra una tabla con los factores de desarrollo individuales d_{ih} . Esto se debe a que el método de la variante de Chain-ladder (apartado 3.3.2) requiere de los factores de desarrollo individuales para su cálculo, por lo que se proporciona al usuario la información en caso de que éste la desee.

CAPÍTULO 5: CASO PRÁCTICO

En este último capítulo conciliaremos los *outputs* proporcionados por el aplicativo respecto a los resultados obtenido al hacer el mismo ejercicio de forma manual. De esta manera confirmaremos que el aplicativo está correctamente programado para realizar la gran mayoría de ejemplos.

La sentencia “*confirmaremos que el aplicativo está correctamente programado para realizar la gran mayoría de ejemplos*” se debe a la complejidad de comprobar todas las combinaciones de *inputs* posibles del aplicativo, lo que significaría que podríamos confirmar al 100% que el aplicativo no tiene fallo alguno.

Con el objetivo de evitar que el capítulo quede excesivamente extenso y para facilitar la comprensión, este capítulo se centrará en el desarrollo del mismo ejercicio para dos métodos de los mencionados en capítulos anteriores:

- Chain-ladder
- Variación Chain-ladder.

Como complemento a lo anteriormente mencionado, y con el objetivo de cubrir el mayor número de funcionalidades del aplicativo, el ejercicio se hará con uno de los triángulos generados por el mismo, pero para resolverlo se descargará éste y se resolverá mediante la subida manual del mismo (cubriendo así las dos opciones de subida de triángulo de datos disponibles en el aplicativo).

5.1 Chain-ladder

Siguiendo los pasos comentados en el párrafo anterior a este, el triángulo acumulado a trabajar será el siguiente:

Periodos de origen <i>i</i> (Trimestres)	Periodos de desarrollo <i>j</i> (Semestres)				
	6M	12M	18M	24M	30M
01-19	457.461	896.701	1.199.489	1.392.994	1.584.551
04-19	594.598	1.247.590	1.910.413	2.372.145	2.730.483
07-19	516.169	1.042.617	1.420.295	1.743.466	
10-19	645.712	1.059.848	1.329.331	1.507.033	
01-20	528.491	929.309	1.312.769		
04-20	417.964	860.769	1.312.516		
07-20	408.079	780.129			
10-20	564.949	848.427			
01-21	680.976				
04-21	457.461				

Tabla 24 Triángulo acumulado. Ejemplo práctico [Fuente: Elaboración propia]

Con el objetivo de hacer un ejemplo complejo para el aplicativo se ha optado por las siguientes opciones:

- Una recurrencia en los periodos de origen *i* trimestral

- Una recurrencia en los periodos de desarrollo j semestral
- Del 2019 hasta 2021
- Último mes disponible: Septiembre del 2021.

Como se puede observar en el triángulo, aunque el último mes existente es septiembre del 2021, la última fila proporcionada por el aplicativo acaba en abril del 2021. ¿A qué se debe eso? ¿Es un error?

No es ningún error. Si sumamos un trimestre a abril, nuestro siguiente periodo de origen empezaría en julio (07/21). Si a julio le intentamos sumar seis meses, lo que implicaría el primer periodo de desarrollo, este acabaría en diciembre del 2021, por lo que, al ser septiembre el último mes disponible, no tendríamos de información alguna.

El aplicativo es capaz de detectar esto y eliminar las filas sin datos.

Una vez sabido esto, y volviendo al método, el primer paso a seguir en el método Chain-ladder es el cálculo de los factores de desarrollo. Como ejemplo, el primer factor de desarrollo será la división entre las celdas en gris de la columna “12M”, respecto a las mismas en la columna “6M”.

Siguiendo el procedimiento explicado en el capítulo 3.3.1, obtendremos los siguientes factores:

	6M-12M	12M-18M	18M-24M	24M-30M
Factores de desarrollo	1,8545	1,4055	1,1973	1,1460

Tabla 25 Factores de desarrollo obtenidos por el método Chain-ladder [Fuente: Elaboración propia]

Los factores mostrados por el aplicativo son:

	6M-12M	12M-18M	18M-24M	24M-30M
Factores	1.8545	1.4055	1.1973	1.146

Ilustración 16 Factores de desarrollo calculados por el aplicativo. Ejemplo práctico [Fuente: Elaboración propia]

De esta manera, se corrobora que el aplicativo realiza el cálculo correcto.

Una vez obtenidos los factores de desarrollo sólo quedará utilizarlos para estimar la parte desconocida del triángulo. Si hacemos este cálculo, obtendremos el siguiente resultado:

Periodos de origen i (Trimestres)	Periodos de desarrollo j (Semestres)				
	6M	12M	18M	24M	30M
01-19	457.461	896.701	1.199.489	1.392.994	1.584.551
04-19	594.598	1.247.590	1.910.413	2.372.145	2.730.483
07-19	516.169	1.042.617	1.420.295	1.743.466	1.998.098
10-19	645.712	1.059.848	1.329.331	1.507.033	1.727.134

01-20	528.491	929.309	1.312.769	1.571.784	1.801.342
04-20	417.964	860.769	1.312.516	1.571.481	1.800.994
07-20	408.079	780.129	1.096.477	1.312.816	1.504.552
10-20	564.949	848.427	1.192.470	1.427.750	1.636.271
01-21	680.976	1.262.863	1.774.963	2.125.171	2.435.550
04-21	457.461	848.357	1.192.371	1.427.631	1.636.135

Tabla 26 Triángulo acumulado estimado. Ejemplo práctico [Fuente: Elaboración propia]

El aplicativo por su parte, estima el siguiente triángulo:

	6M	12M	18M	24M	30M
01-19	457461	896701	1199489	1392994	1584551
04-19	594598	1247590	1910413	2372145	2730483
07-19	516169	1042617	1420295	1743466	1998098
10-19	645712	1059848	1329331	1507033	1727134
01-20	528491	929309	1312769	1571784	1801342
04-20	417964	860769	1312516	1571481	1800994
07-20	408079	780129	1096477	1312816	1504552
10-20	564949	848427	1192470	1427750	1636271
01-21	680976	1262863	1774963	2125171	2435550
04-21	457461	848357	1192371	1427631	1636135

Ilustración 17 Triángulo acumulado estimado. Ejemplo práctico. Aplicación Shiny [Fuente: Elaboración propia]

De nuevo, se verifica la coincidencia entre el resultado obtenido manualmente. Finalmente, como parte final de este ejercicio práctico, realizaremos el mismo proceso para los pagos futuros actualizados, y lo haremos a un tipo fijo del 2%, bajo el modelo de renta.

Con el objetivo de comprender cómo serían los pagos por semestres naturales para el ejemplo utilizado, el pago estimado más cercano y, por ende, la primera cuantía a actualizar, sería el compuesto por los siguientes importes (recordar que debemos trabajar con el triángulo no acumulado):

$$CF1 = (1.262.863 - 680.976) + 316.348 + 259.015 + 254.632 = 1.411.881$$

Si realizamos este mismo proceso para los 8 pagos futuros y los actualizamos al 2% anual obtendremos los siguientes datos:

	1	2	3	4	5	6	7	8
CF	1.397.971	1.190.201	929.959	777.400	515.767	418.185	289.596	192.626

Tabla 27 Flujos de pagos futuros actualizados. [Fuente: Elaboración propia]

Es importante puntualizar que, al trabajar con semestres y al estar actualizando al último semestre con datos disponibles, cada importe no se actualizará al 2% anual, sino a su tanto equivalente semestral, siendo este el 0,995% semestral aproximadamente.

Por parte del *output*, el resultado obtenido es el siguiente:

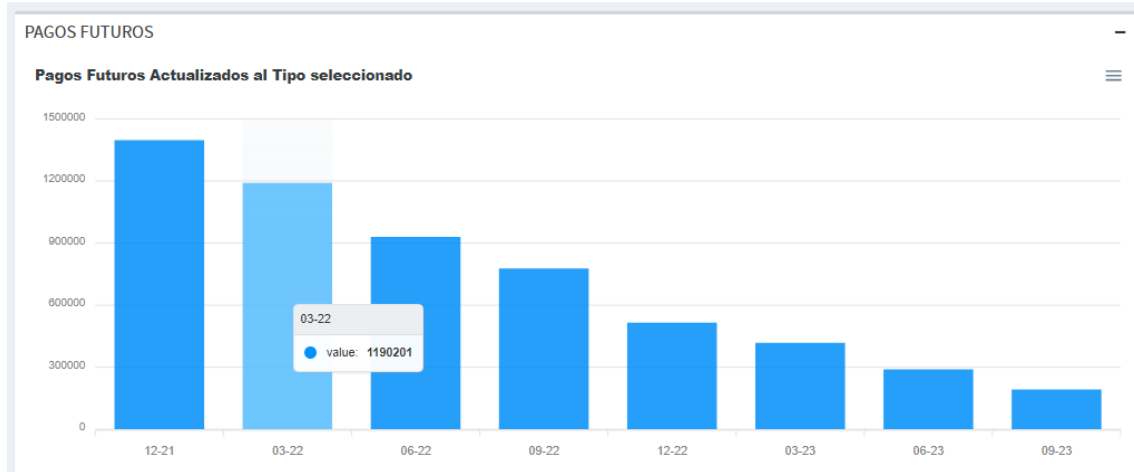


Ilustración 18 Flujos de pagos futuros actualizados. Aplicación Shiny [Fuente: Elaboración propia]

De esta manera y con este ejercicio sencillo, se confirma que el aplicativo, en un principio y para la gran mayoría de combinaciones posibles, está bien hecho.

5.2 Variación Chain-ladder

Como hemos comentado al principio de este capítulo, el caso práctico está compuesto por la resolución de un mismo ejercicio mediante los dos métodos relativos a Chain-ladder.

Por lo tanto, y volviendo al triángulo introducido anteriormente, debemos tener en cuenta como este segundo método se basa en los factores de desarrollo individuales.

De nuevo, con el triángulo acumulado, obtendremos los siguientes factores de desarrollo individuales:

Periodos de origen <i>i</i> (Trimestres)	Factores de desarrollo individuales			
	6M-12M	12M-18M	18M-24M	24M-30M
01-19	1,96017	1,33767	1,16132	1,13751
04-19	2,09821	1,53128	1,24169	1,15106
07-19	2,01991	1,36224	1,22754	
10-19	1,64136	1,25427	1,13368	
01-20	1,75842	1,41263		
04-20	2,05943	1,52482		
07-20	1,91171			
10-20	1,50178			
01-21				
04-21				

Tabla 28 Factores de desarrollo individuales obtenidos por el método variante de Chain-ladder [Fuente: Elaboración propia]

De igual manera que hemos hecho en la anterior resolución, los resultados obtenidos en el aplicativo son los siguientes:

FACTORES DE DESARROLLO INDIVIDUALES				
	6M-12M	12M-18M	18M-24M	24M-30M
01-19	1.9602	1.3377	1.1613	1.1375
04-19	2.0982	1.5313	1.2417	1.1511
07-19	2.0199	1.3622	1.2275	
10-19	1.6414	1.2543	1.1337	
01-20	1.7584	1.4126		
04-20	2.0594	1.5248		
07-20	1.9117			
10-20	1.5018			

Ilustración 19 Factores de desarrollo individuales obtenidos por el método variante de Chain-ladder. Aplicación Shiny [Fuente: Elaboración propia]

Como se puede apreciar, los resultados obtenidos vuelven a coincidir y se corrobora que, hasta ahora, el aplicativo soluciona correctamente el problema.

Siguiendo con el método, si volvemos a la teoría del apartado 3.3.2, podremos ver cómo esta variación se basa en la ponderación $i+j+1$. Por lo tanto, teniendo en cuenta el triángulo anterior, deberíamos trabajar con las siguientes ponderaciones:

Periodos de origen i (Trimestres)	Pesos variación Chain-ladder			
	6M-12M ($j=0$)	12M-18M ($j=1$)	18M-24M ($j=2$)	24M-30M ($j=3$)
01-19 ($i=0$)	1	2	3	4
04-19 ($i=1$)	2	3	4	5
07-19 ($i=2$)	3	4	5	
10-19 ($i=3$)	4	5	6	
01-20 ($i=4$)	5	6		
04-20 ($i=5$)	6	7		
07-20 ($i=6$)	7			
10-20 ($i=7$)	8			

Tabla 29 Pesos método variante de Chain-ladder [Fuente: Elaboración propia]

Una vez aplicamos dichas ponderaciones a cada uno de los factores de desarrollo individuales obtenemos los siguientes factores de desarrollo:

	6M-12M	12M-18M	18M-24M	24M-30M
Factores de desarrollo	1,8146	1,4126	1,1884	1,1450

Tabla 30 Factores de desarrollo finales obtenidos por el método variante de Chain-ladder [Fuente: Elaboración propia]

Por su parte, el aplicativo nos muestra los siguientes factores de desarrollo:

FACTORES DE DESARROLLO				
	6M-12M	12M-18M	18M-24M	24M-30M
Factores	1.8146	1.4126	1.1884	1.145

Showing 1 to 1 of 1 entries

Ilustración 20 Factores de desarrollo finales obtenidos por el método variante de Chain-ladder. Aplicación Shiny [Fuente: Elaboración propia]

De nuevo, se demuestra la coincidencia entre la resolución manual y el aplicativo.

Si aplicamos los factores de desarrollo obtenidos mediante esta variante del método Chain-ladder, obtendremos las siguientes estimaciones:

Periodos de origen <i>i</i> (Trimestres)	Periodos de desarrollo <i>j</i> (Semestres)				
	6M	12M	18M	24M	30M
01-19	457.461	896.701	1.199.489	1.392.994	1.584.551
04-19	594.598	1.247.590	1.910.413	2.372.145	2.730.483
07-19	516.169	1.042.617	1.420.295	1.743.466	1.996.339
10-19	645.712	1.059.848	1.329.331	1.507.033	1.725.613
01-20	528.491	929.309	1.312.769	1.560.043	1.786.312
04-20	417.964	860.769	1.312.516	1.559.743	1.785.968
07-20	408.079	780.129	1.101.975	1.309.544	1.499.480
10-20	564.949	848.427	1.198.450	1.424.190	1.630.755
01-21	680.976	1.235.719	1.745.521	2.074.309	2.375.167
04-21	457.461	830.122	1.172.593	1.393.464	1.595.572

Tabla 31 Triángulo estimado. Variante Chain-ladder [Fuente: Elaboración propia]

Por su parte, el aplicativo nos ha mostrado los siguientes resultados:

	6M	12M	18M	24M	30M
01-19	457461	896701	1199489	1392994	1584551
04-19	594598	1247590	1910413	2372145	2730483
07-19	516169	1042617	1420295	1743466	1996339
10-19	645712	1059848	1329331	1507033	1725613
01-20	528491	929309	1312769	1560043	1786312
04-20	417964	860769	1312516	1559743	1785968
07-20	408079	780129	1101975	1309544	1499480
10-20	564949	848427	1198450	1424190	1630755
01-21	680976	1235719	1745521	2074309	2375167
04-21	457461	830122	1172593	1393464	1595572

Ilustración 21 Triángulo estimado. Variante Chain-ladder. Aplicación Shiny [Fuente: Elaboración propia]

De nuevo, se vuelve a corroborar la validez de los resultados obtenidos en el aplicativo.

Finalmente, como parte final de este ejercicio práctico, realizaremos el mismo proceso para los pagos futuros actualizados, pero esta vez utilizaremos la curva de tipos europea publicada por EIOPA sin ajuste por volatilidad (*volatility adjustment, VA*) para abril de 2022, EIOPA (2022).

Los cuatro primeros años de la curva de tipos relativa a abril de 2022 está compuesta de la siguiente manera:

Años	Tipo de interés		
	Anual	Forward	Forward Semestral
1	0,161%	0,16%	0,08%
2	0,868%	1,58%	0,79%
3	1,132%	1,66%	0,83%

4 1,263% 1,66% 0,83%

Tabla 32 Curva de tipos sin riesgo ni VA. Abril 2022 [Fuente: EIOPA (2022)]

Estos cálculos se han realizado según lo establecido en el capítulo 4.2.7. Los flujos de caja actualizados a los tipos semestrales correspondientes quedarán tal que:

	1	2	3	4	5	6	7	8
CF	1.376.736	1.188.491	943.640	794.437	518.724	427.435	300.858	202.108
F dto.	0,9992	0,9984	0,9906	0,9829	0,9748	0,9668	0,9589	0,951
CF Actual	1.375.629	1.186.580	934.767	780.823	505.650	413.242	288.488	192.212

Tabla 33 Flujos de pagos futuros actualizados y factor de descuento. [Fuente: Elaboración propia]

Por parte del *output*, el resultado obtenido es el siguiente:

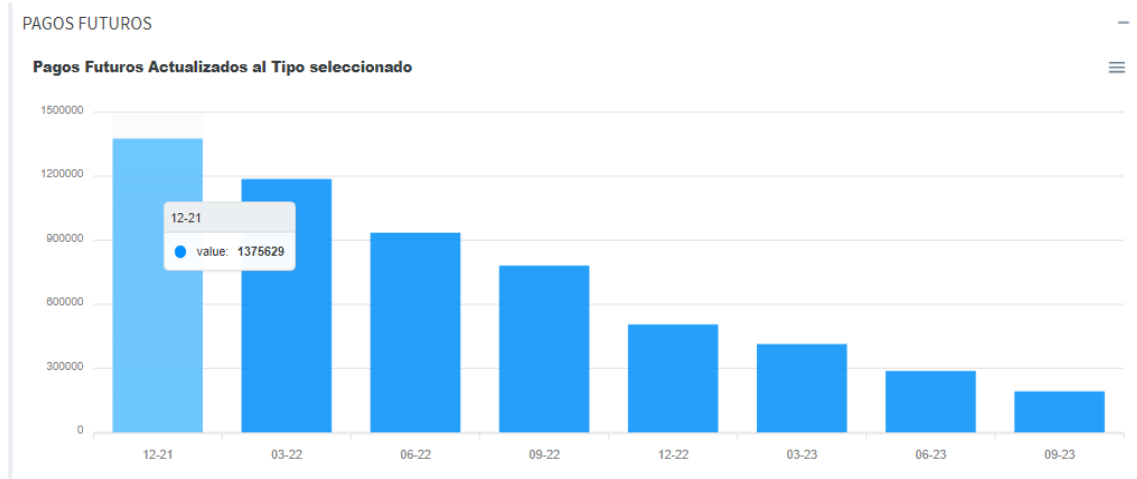


Ilustración 22 Flujos de pagos futuros actualizados. Aplicación Shiny [Fuente: Elaboración propia]

De esta manera y con este ejercicio sencillo, se confirma que el aplicativo, en un principio y para la gran mayoría de combinaciones posibles, está bien hecho.

Como conclusión, si bien es cierto que para una correcta comprobación se deberían ejecutar todas aquellas combinaciones de opciones posibles, así como la totalidad de los métodos disponibles, para este trabajo, y con el objetivo de no complicarlo en exceso, se han realizado un par de demostraciones como muestra. De manera adicional se han realizado múltiples comprobaciones con el objetivo de corroborar, con la mayor probabilidad posible, la eficacia del aplicativo.

CONCLUSIONES

Con la elaboración de este trabajo, y tras lo mencionado en la introducción, se ha conseguido el objetivo principal de éste: La creación de un aplicativo en R para el cálculo de reservas futuras en el ramo de no vida, que constituye una herramienta metodológica para los profesionales a la hora de trabajar con triángulos que no disponen de la misma recurrencia en sus períodos de origen i y sus períodos de desarrollo j .

En el transcurso del trabajo, tras realizar una pequeña introducción a las diferentes definiciones legales en cuanto a provisiones, se han clasificado los métodos de cálculo, para luego entrar en detalle en aquellos finalmente adaptados al aplicativo programado, diferenciando entre los de carácter determinista o estocástico.

Las verificaciones de los resultados obtenidos en la parte práctica nos permiten, con una cierta seguridad, confirmar las adaptaciones realizadas y la correcta realización de la aplicación final.

Con todo ello, no sólo se consigue una aplicación que permite realizar los cálculos mencionados anteriormente, sino que además se ha programado en un entorno sencillo y visual en Shiny para la obtención de resultados que podrían resultar más complejos, si se tuviesen que realizar manualmente. De igual manera, esta sencillez del aplicativo permite a cualquier profesional, sea actuario o no, obtener resultados sin requerir un gran conocimiento.

Todo y los grandes resultados obtenidos es importante entender que, debido a la complejidad de los lenguajes de programación, en el transcurso del trabajo no se han podido introducir todas aquellas funcionalidades deseadas. Las limitaciones en cuanto al tipo de archivo a subir, así como la imposibilidad de realizar cambios concretos en los triángulos, son puntos que se han dejado para futuras investigaciones.

De esta forma, se finaliza este trabajo de fin de máster, no como un punto final, sino como un punto y aparte hacia un aplicativo mucho más complejo y adaptable a todas las necesidades que cualquier usuario, sea o no actuario, pueda tener. Así como a ampliaciones sobre las magnitudes calculadas, especialmente para los métodos estocásticos, y sobre la incorporación de otras metodologías de cálculo de provisiones.

La aplicación resultante de este trabajo puede encontrarse en la siguiente ubicación:

<https://azaelmm.com/shiny/calnovida/>

BIBLIOGRAFÍA

ALLIANZ SEGUROS, *Diccionario de seguros*.

<https://www.allianz.es/descubre-allianz/mediadores/diccionario-de-seguros.html>

BOJ, E., CLARAMUNT, M. M. y COSTA, T., *provisio: Área de trabajo en lenguaje R para el cálculo de provisiones técnicas en seguros no de vida con métodos deterministas*. Depósito digital de la Universidad de Barcelona, colección de investigación-software, 2017.

<http://hdl.handle.net/2445/106653>

BOJ, E., CLARAMUNT, M. M. y COSTA, T., *Tarificación y provisiones (tercera edición)*. Depósito Digital de la Universidad de Barcelona, colección OMADO, 2020.

<http://hdl.handle.net/2445/149241>

BOJ, E., y COSTA, T., *Modelo lineal generalizado y cálculo de la provisión técnica*. Depósito digital de la Universidad de Barcelona, colección OMADO, 2014.

<http://hdl.handle.net/2445/49068>

CHANG, W., CHENG, J., ALLAIRE, JJ., SIEVERT, C., SCHLOERKE, B., XIE, Y., ALLEN, J., MCPHERSON, J., DIPERT, A., BORGES, B., Shiny, Paquete de R en Cran, versión 1.7.1, 2021.

<https://cran.r-project.org/package=shiny>

<https://shiny.rstudio.com/>

CHANG, W., RIBEIRO, B., Shinydashboard, Paquete de R en Cran, versión 0.7.2, 2021.

<https://cran.r-project.org/package=shinydashboard>

EIOPA., *Risk-free interest rate term structures*, EIOPA, European Insurance and Occupational Pensions Authority Westhafenplatz 1, 60327 Frankfurt am Main, Abril, 2022.

https://www.eiopa.europa.eu/tools-and-data/risk-free-interest-rate-term-structures_en

GESMANN, M., MURPHY, D., ZHANG, Y., CARRATO, A., CRUPI, G., DUTANG, CH., LACOUME, A., CHARPENTIER, A., WUTHRICH, M., CONCINA, F.,

DAL MORO, E., KRVAVYCH, Y., GOULET, V., DE VIRGILIS, M., ChainLadder, Paquete de R en Cran, versión 0.2.15, 2022.

<https://cran.r-project.org/package=ChainLadder>

IRMI, INTERNATIONAL RISK MANAGEMENT INSTITUTE, INC., *Insurance and Risk Management Terms*

<https://www.irmi.com/glossary>

MCCULLAGH, P. and NELDER, J., *Generalized linear models (2nd edition)*, Chapman and Hall, London, 1989.

PERRIER, V., MEYER, F., CHHIPA, J., BOSTOCK, M., Apexcharter, Paquete de R en Cran, versión 0.3.1, 2022.

<https://cran.r-project.org/package=apexcharter>

R DEVELOPMENT CORE TEAM., R Development Core Team, R: a language and environment for statistical computing. Vienna, Austria, 2022.

<https://www.r-project.org/>

RSTUDIO TEAM., *RStudio: Integrated Development for R*. RStudio, PBC, Boston, MA, 2020

<https://www.rstudio.com/>

SPINU, V., GROLEMUND, G., WICKHAM, H., VAUGHAN, D., LYTTLE, I., COSTIGAN, I., LAW, J., MITAROTONDA, D., LARMARANGE, J., BOISER, J., LEE, C., Lubridate, Paquete de R en Cran, versión 1.8.0, 2021.

<https://cran.r-project.org/package=lubridate>

TAYLOR, G.C., *Claim reserving in non-life insurance.*, North Holland, Insurance Series 1, 1986.

WEINDORFER, B., *A practical guide to the use of the Chain-ladder method for determining technical provisions for outstanding reported claims in non-life insurance.* University of Applied Sciences bfi Vienna, 2012.

WICKHAM, H., CHANG, W., HENRY, L., PEDERSEN, T. TAKAHASHI, K., WILKE, C., WOO, K., YUTANI, H., DUNNINGTON, D., Ggplot2, Paquete de R en Cran, versión 3.3.5, 2021.

<https://cran.r-project.org/package=ggplot2>

WAI-SUM, C., YIU-KUEN, T., *Financial Mathematics for Actuaries, 2nd Edition*, 2017

XIE, Y., CHENG, J., TAN, X., ALLAIRE, JJ., GIRLICH, M., FREEDMAN, G., RAUH, J., REAVIS, B., GERSEN, L., SZOPKA, B., PICKERING, A., HOLMES, W., DT, Paquete de R en Cran, versión 0.21, 2022.

<https://cran.r-project.org/package=DT>

NORMATIVA

Directiva 2009/138/CE del parlamento europeo y del consejo, de 25 de noviembre de 2009, sobre el seguro de vida, el acceso a la actividad de seguro y de reaseguro y su ejercicio (Solvencia II)

<http://data.europa.eu/eli/dir/2009/138/oj>

Ley 20/2015, de 14 de julio, de ordenación, supervisión y solvencia de las entidades aseguradoras y reaseguradoras.

<https://www.boe.es/eli/es/l/2015/07/14/20/con>

Real Decreto 2486/1998, de 20 de noviembre de 1998. Boletín Oficial del Estado, de 25 de noviembre de 1998. Reglamento de Ordenación y Supervisión de los Seguros Privados.

<https://www.boe.es/eli/es/rd/1998/11/20/2486/con>