UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE INGENIERÍA ECONOMICA, ESTADISTICA Y CIENCIAS SOCIALES

TESIS

"PRIMA PURA SUJETO A RIESGOS DEPENDIENTES una propuesta metodológica con múltiples coberturas"

PARA OBTENER EL GRADO ACADÉMICO DE MAESTRO EN CIENCIAS EN CIENCIAS ACTUARIALES

ELABORADO POR:

JOSEPH GARCIA SANTIAGO
ORCID: 0009-0001-0829-7991

ASESOR:

Ph. D. JOSÉ JAVIER CERDA HERNÁNDEZ

ORCID: 0000-0002-9297-5694

LIMA – PERÚ

2023

Garcia Santiago, Joseph
jgarcias@uni.edu.pe
924932992

Dedicado a,
mi madre Elena Santiago Peña
por su constate amor y cariño,
siempre testigo de mis logros y
que a pesar de haber estado
lejos desde mi vida universitaria
siempre me apoyó incondicionalmente,

mi padre Flaviano Garcia Mendez
por creer siempre en mi capacidad,
por fortalecerme cuando creía
de no poder hacerlo, y sobretodo
por brindarme su cariño y apoyo,

A mi esposa Linda Quiñonez Huamanlazo
quién estuvo conmigo en los
momentos más díficiles que nos tocó
afrontar en estos últimos años.

mi hija Valentina
quién es mi motivo de superación
profesional.

AGRADECIMIENTOS

Agradezco a todos mis familiares, amigos y colegas del trabajo que de alguna manera me han brindado su apoyo y han sido testigos de este logro profesional. En particular:

Agradezco a mi asesor de tesis, Phd. José Cerda , por sus enseñazas, consejos y colaboración en lograr la presente tesis.

A mis revisores de tesis, Mg. Marco Ávila y Mg. Rafael Caparó, por su tiempo y disposición en revisar la tesis, además, de brindarme sugerencias de la misma.

Agradezco a Elizabeth Cáceres, Directora de Advisory de KPMG en Perú, por brindarme horas para desarrollar la presente tesis dentro de la jornada laboral, cuya finalidad ha sido y es, la de mantener un equipo de servicios profesionales altamente capacitado.

Agradezco a mis profesores de la Maestría en Ciencias Actuariales y el equipo de Posgrado de la Facultad de Ingeniería Económica, Estadística y Ciencias Sociales por haber llevado en adelante el desarrollo de esta maestría pionera en la formación profesionales Actuarios peruanos, además, de permitirme ser parte de la primera promoción de Actuarios que iniciamos esta maestría.

Índice general

RESUMEN VI

ABSTRACT VII

LISTA DE TABLAS VIII

LISTA DE FIGURAS IX

LISTA DE SÍMBOLOS Y SIGLAS XI

CAPÍTULO I: Planteamiento del problema 1

1.1 Descripción de la situación problemática . . . 1
1.2 Formulación del problema . . . 2
1.2.1 Problema general . . . 2
1.2.2 Problemas específicos . . . 2
1.3 Objetivos de la investigación . . . 2
1.3.1 Objetivo General . . . 2
1.3.2 Objetivos Específicos . . . 2
1.4 Justificación, alcances y limitaciones de la investigación . . . 3
1.4.1 Justificación . . . 3
1.4.2 Alcances . . . 3
1.4.3 Limitaciones . . . 3

CAPÍTULO II: Marco Teórico Conceptual 5

2.1 Antecedentes de la investigación . . . 5
2.2 Bases teóricas de la investigación . . . 7
2.2.1 Modelos Lineales Generalizados . . . 7
2.2.2 Cópulas . . . 19
2.2.3 Familias de Cópulas . . . 30
2.2.4 Modelos GLM basados en Cópulas . . . 35
2.2.5 Construcción de modelos multivariados altamente dependientes - 2 Alternativas . . . 42
2.3 Enfoque teórico asumido por el investigador . . . 49
2.3.1 Prima pura sujeto a efectos de dependencia . . . 49
2.4 Hipótesis . . . 53
2.4.1 Hipótesis general . . . 53

2.4.2 Hipótesis específicas . . . 53
2.5 Variables . . . 53
2.5.1 Robustez . . . 53
2.5.2 Asociación . . . 53
2.5.3 Desempeño . . . 54

CAPÍTULO III:METODOLOGÍA **55**
3.1 Tipo, nivel y diseño de la investigación . . . 55
3.2 Población, muestra y tamaño de muestra . . . 55
3.2.1 Población . . . 55
3.2.2 Muestra . . . 55
3.2.3 Tamaño de muestra . . . 55
3.3 Técnicas de análisis e instrumentos . . . 55
3.3.1 Técnicas de Análisis . . . 55
3.3.2 Instrumentos . . . 57
3.4 Operacionalización y Matriz de consistencia . . . 63
3.4.1 Cuadro de operacionalización de variables . . . 63
3.4.2 Matriz de consistencias . . . 64

CAPÍTULO IV: ANALISIS Y RESULTADOS **65**
4.1 Análisis descriptivo de los datos . . . 65
4.2 Análisis de los datos . . . 75
4.2.1 Modelos Marginales de la Frecuencia y de la Severidad . . . 75
4.2.2 Pruebas de indicios de Dependencia . . . 76
4.2.3 Modelos GLM Conjuntos de la Frecuencia y de la Severidad . . . 80
4.2.4 Agregación de la Prima Pura de Riesgo por Cobertura . . . 81
4.3 Interpretación y discusión de los resultados . . . 87

CONCLUSIONES **90**

RECOMENDACIONES **91**

REFERENCIAS BIBLIOGRÁFICAS **92**

ANEXOS **95**
ANEXO A: ANÁLISIS UNIVARIADO . . . 96
ANEXO B: MODELOS GLM . . . 112
ANEXO C: INDICIO DE DEPENDENCIA Y MODELOS GLM CONJUNTOS . 137

RESUMEN

En los últimos años, la industria de seguros en el Perú ha experimentado un notable crecimiento, catalizando el desarrollo de la profesión actuarial. En el ámbito del *pricing*, es común utilizar modelos predictivos. Aunque su implementación no está completamente extendida en las compañías de seguros, es necesario contar con modelos más eficientes que nos permitan estimar primas de manera adecuada. Esto significa cumplir con nuestras obligaciones frente a los asegurados por siniestros ocurridos, cubrir los gastos asociados y, finalmente, generar un margen de utilidad para la aseguradora.

Para la estimación de la prima de riesgo es usual analizar la distribución de la frecuencia y de la severidad sobre alguna linea de negocio. La modelación para una cartera de seguros normalmente incluyen covariables asociadas al asegurado, al bien asegurado, zonas geográficas e historial de siniestralidad, es decir, la modelación de la frecuencia y de la severidad es multivariada, con base a ello los modelos lineales generalizados son una herramienta de modelación por excelencia de la frecuencia y de la severidad en función de covariables. Sin embargo, en la práctica estos riesgos se suelen asumir independientes entre si, lo que nos conlleva a una sobre-estimación de la prima pura de riesgo.

En objetivo principal del presente trabajo, es incluir el efecto de la depedencia entre la frecuencia y la severidad en el cálculo de la prima pura. Para lograr esto utilizamos diversas familias de cópulas, cuyas propiedades teórica han demostrado un buen performance y una eficiencia en el modelamiento conjunto de riesgos con cierto grado de dependencia. Utilizamos cuatro familias de cópulas para construir un modelo conjunto de la frecuencia y de la severidad en presencia de covariables, y lo aplicamos a una linea de negocio expecífica de alta frecuencia, como lo son los seguros vehiculares. Además, los datos utilizados en el presente trabajo provienen de un repositorio estadísitico con datos reales de otro país, pero será suficiente para enteder y modelar la dependencia empírica observada entre la frecuencia y la severidad.

Finalmente, el principal aporte de este trabajo es mostrar una propuesta metodológica para el cálculo de la prima de riesgo, considerando múltiples coberturas en un entorno donde la frecuencia y la severidad son dependientes. Al final veremos que existe cierta sobre estimación de la prima de riesgo por limitarse al supuesto de independencia.

ABSTRACT

The insurance industry has boomed in recent years in Peru, and along with it the development of the actuarial profession. In the field of Pricing it is usual to use predictive models, although their application is not 100% implemented in insurance companies, it is necessary to have more efficient models that allow us to estimate sufficient premiums, that is to say, that allow us to meet our obligations in the face of claims, cover the expenses associated with the sale and finally generate a profit margin for the insurer. For the estimation of the risk premium it is usual to analyze the distribution of frequency and severity over some line of business. The modeling for an insurance portfolio normally includes covariates associated with the insured, the insured property, geographical areas and claims history, i.e., the modeling of frequency and severity is multivariate, based on which generalized linear models are a modeling tool par excellence for frequency and severity as a function of covariates. However, in practice these risks are usually assumed to be independent of each other, which leads to an overestimation of the pure risk premium.

The present work aims to include the effect of the dependence between frequency and severity, to achieve this we will use copulas functions, whose properties have demonstrated their efficiency in the joint modeling of risks with a certain degree of dependence. We will use four families of copulas to build a joint model of frequency and severity in the presence of covariates and applied on a high frequency line of business such as vehicle insurance, in addition, the information used comes from a statistical repository with real data from another country, but it will be enough to understand the relationship between frequency and severity.

Finally, the contribution of this work is to show a methodological proposal for the calculation of the risk premium, considering multiple coverages in an environment where frequency and severity are dependent. At the end we will see that there is some overestimation of the risk premium by limiting it to the assumption of independence.

LISTA DE TABLAS

Tabla N° 2.1 Distribuciones de la familia exponencial y sus parámetros 9
Tabla N° 2.2 Funciones enlaces comunes 11
Tabla N° 2.3 Devianza para distribuciones de la familia exponencial 17
Tabla N° 2.4 Datos del Cuarteto de Anscombe 25
Tabla N° 2.5 Relación entre el parámetro θ y τ de Kendall 36
Tabla N° 2.6 Primera derivada parcial 37
Tabla N° 2.7 Parámetros del modelo de distribución conjunta 38
Tabla N° 2.8 Trasformación no restricta de θ 40
Tabla N° 2.9 Expresiones de v por cada familia de cópulas 52
Tabla N° 3.1 Cuadro de operacionalización de variables 63
Tabla N° 3.2 Matriz de consistencia 64
Tabla N° 4.1 Variable: Tipo de póliza 66
Tabla N° 4.2 Variable: Antiguedad del vehículo 67
Tabla N° 4.3 Variable: Canal de venta 68
Tabla N° 4.4 Variable: Uso del vehículo 69
Tabla N° 4.5 Variable: Clase del vehículo 70
Tabla N° 4.6 Variable: Tipo de vehículo 70
Tabla N° 4.7 Variable: Marca del vehículo y por clase 71
Tabla N° 4.8 Variable: Zona demográfica 72
Tabla N° 4.9 Variable: Género del asegurado 73
Tabla N° 4.10 Variable: Tipo de Persona 73
Tabla N° 4.11 Variable: Edad del asegurado 74
Tabla N° 4.12 Modelo Marginal de Frecuencia y Severidad - Pérdidas Parciales 76
Tabla N° 4.13 Modelo Marginal de Frecuencia y Severidad - Pérdidas Totales . 77
Tabla N° 4.14 Modelo Marginal de Frecuencia y Severidad - Responsabilidad Civil 77
Tabla N° 4.15 Modelo Marginal de Frecuencia y Severidad - Asistencias 77
Tabla N° 4.16 Pérdida Parcial | Log-likelihood de los modelos de regresión . . . 80
Tabla N° 4.17 Pérdida Total | Log-likelihood de los modelos de regresión 80
Tabla N° 4.18 Responsabilidad Civil | Log-likelihood de los modelos de regresión 80
Tabla N° 4.19 Asistencias | Log-likelihood de los modelos de regresión 80
Tabla N° 4.20 Modelo de Regresión GLM Conjunto - Pérdidas Parciales 81
Tabla N° 4.21 Modelo de Regresión GLM Conjunto - Pérdidas Totales 82
Tabla N° 4.22 Modelo de Regresión GLM Conjunto - Responsabilidad Civil . . 82
Tabla N° 4.23 Modelo de Regresión GLM Conjunto - Asistencias 82
Tabla N° 4.24 Grados de dependencia bivariado - Tau de Kendall 85

LISTA DE FIGURAS

Figura N° 2.1 Superficie máxima Log-Verosimilitud 14
Figura N° 2.2 Distribución Chi-Cuadrado 18
Figura N° 2.3 Descripción de como se utiliza el Teorema de Sklar para calular la distribución conjunta de un vector aleatorio. 21
Figura N° 2.4 Función de densidad conjunta 22
Figura N° 2.5 Gráficos del Cuarteto de Anscombe 26
Figura N° 2.6 Chi-plot para variables independientes con $p = 0.95$ 29
Figura N° 2.7 Chi-plot para variables dependientes con $p = 0.95$ 29
Figura N° 2.8 Gráficos K-PLOT 30
Figura N° 2.9 Cópula Gaussiana $(\rho = 0.8)$ 31
Figura N° 2.10 Cópula Student$(\tau = 0.5, \nu = 4)$ 32
Figura N° 2.11 Cópula Clayton$(\theta = 0.5)$ 33
Figura N° 2.12 Cópula de Frank$(\theta = 0.5)$ 34
Figura N° 2.13 Cópula de Gumbel$(\theta = 0.5)$ 35
Figura N° 2.14 D-vine cópula: 5 variables, 4 árboles y 10 pair-cópulas 45
Figura N° 2.15 C-vine cópula: 5 variables, 4 árboles y 10 pair-cópulas 45
Figura N° 2.16 Ilustración del árbol $\tau = \{\emptyset, (1), (1,1), (1,2), (2)\}$ 46
Figura N° 2.17 Una ilustración de un modelo de agregación de 4 dimensiones . 48
Figura N° 2.18 Densidad conjunta de la Frecuencia y Severidad 51
Figura N° 2.19 Distribución de probabilidad de riesgos agregados 52
Figura N° 3.1 Medida de exposición de una póliza en un periodo de tiempo . . 56
Figura N° 3.2 Algoritmo para la construcción del Simple Quantile Plot 58
Figura N° 3.3 Ejemplo del *loss ratio chart* 59
Figura N° 3.4 Índice de Gini para un Modelo de Prima Pura 60
Figura N° 4.1 Frecuencia y Severidad por cobertura y periodo de análisis . . . 65
Figura N° 4.2 Matriz de correlación - Pérdida Parcial 75
Figura N° 4.3 Robustez de los modelos de GLM Marginales - Índice de GINI . 78
Figura N° 4.4 Indicio de dependencia entre la Frecuencia y la Severidad por Cobertura 78
Figura N° 4.5 Chi-Plots de la Frecuencia y Severidad por cobertura 79
Figura N° 4.6 K-Plots de la Frecuencia y Severidad por cobertura 79
Figura N° 4.7 Distribución de la Prima Pura de Riesgo 83
Figura N° 4.8 Ajuste a una distribución Gamma - Estimación de parámetros . 84
Figura N° 4.9 Formas de agregación de las primas por cobertura 84
Figura N° 4.10 Esquema de agregación de riesgos por cobertura 85
Figura N° 4.11 Distribución de la Prima de Riesgo Bivariada 86

Figura N° 4.12 Distribución de la Prima de Riesgo Multivariada Pérdida Parcial + Pérdida Total + Responsabilidad Civil + Asistencias 87
Figura N° 4.13 Prima Pura de Riesgo: Independencia vs. Dependencia 89
Figura N° .1 Agrupación de la variable Antiguedad del vehículo 96
Figura N° .2 Agrupación de la Marca Vehículo - Pérdida Parcial - Livianos . 97
Figura N° .3 Agrupación de la Marca Vehículo - Pérdida Total - Livianos . . 98
Figura N° .4 Agrupación de la Marca Vehículo - Responsabilidad Civil - Livianos 99
Figura N° .5 Agrupación de la Marca Vehículo - Asistencias - Livianos . . . 100
Figura N° .6 Agrupación de la Marca Vehículo - Pérdida Parcial - Pesados . . 101
Figura N° .7 Agrupación de la Marca Vehículo - Pérdida Total - Pesados . . . 102
Figura N° .8 Agrupación de la Marca Vehículo - Responsabilidad Civil - Pesados 103
Figura N° .9 Agrupación de la Marca Vehículo - Asistencias - Pesados 104
Figura N° .10 Agrupación de la Marca Vehículo - Pérdida Parcial - Vehículos Menores 105
Figura N° .11 Agrupación de la Marca Vehículo - Pérdida Total - Vehículos Menores 106
Figura N° .12 Agrupación de la Marca Vehículo - Responsabilidad Civil - Vehículos Menores 107
Figura N° .13 Agrupación de la Marca Vehículo - Asistencias - Vehículos Menores 108
Figura N° .14 Edad del asegurado - Sin Agrupar 109
Figura N° .15 Edad del asegurado - Agrupado en rangos 110
Figura N° .16 Rangos de Suma Asegurada del Vehículo 111
Figura N° .17 Matriz de correlación - Pérdida Total 112
Figura N° .18 Matriz de correlación - Responsabilidad Civil 113
Figura N° .19 Matriz de correlación - Asistencias 114

LISTA DE SÍMBOLOS Y SIGLAS

SÍMBOLOS

α : Vector de coeficientes del modelo de Severidad

β : Vector de coeficientes del modelo de Frecuencia

θ : Parámetro de de la función Cópula

λ : Vector esperado de la Frecuencia

μ : Vector esperado de la Severidad

ϕ : Parámetro de disepersión del modelo de Severidad

τ : Tau de Kendall - Parámetro de dependencia

SIGLAS

PPR	:	Prima Pura de Riesgo
PP	:	Cobertura Pérdida Parcial
PT	:	Cobertura Pérdida Total
RC	:	Cobertura Responsabilidad Civil
AS	:	Cobertura Asistencias
GLM	:	Generalized Linear Model

CAPÍTULO I: Planteamiento del problema

1.1 Descripción de la situación problemática

Los cambios normativos relacionados al sector asegurador han ido exigiendo a las compañías de seguros una medición más rigurosa y detallada de los riesgos a los que están expuestos para reducir su probabilidad de insolvencia. Dentro de estos requerimientos cuantitativos es fundamental obtener modelos que incluyan de una mejor forma los riesgos a los que se está expuesto, para obtener una mejor estimación de la reserva técnica, que es una provisión que debe tener la compañía de seguros para afrontar los pagos de los siniestros. En tal sentido, lo anterior se traduce en un cálculo de primas de riesgo más exactas utilizando modelos estadísticos y actuariales más sofisticados y reales.

Sin embargo, para el cálculo de la prima pura dentro de la teoría clásica de riesgo es común asumir por simplicidad que las distribuciones de la frecuencia y de la severidad son independientes. Como consecuencia, la prima pura de riesgo es el producto de la frecuencia promedio y la severidad promedio. Este supuesto utilizado por muchos practitioners en la industria de seguros, a menudo injustificado y rebatido por evidencia empírica (Czado, Kastenmeier, Brechmann, y Min, 2012; Lee, Park, y Ahn, 2018; Ohlsson y Johansson, 2010; P. Song, Li, y Yuan, 2009), limita el cálculo de la prima de riesgo generando una sobre estimación. Es por ello que encontrar modelos donde la dependencia entre la frecuencia y la severidad sean considerados pueden reducir aún más la varianza del estimador de la prima de riesgo, generando un estimador más robusto.

Otro problema abordado en la presente investigación es que cuando las primas de riesgo son calculadas por coberturas, es usual en la práctica agregarlas usando una suma simple de riesgos para obtener la prima de riesgo total. Esto genera nuevamente una sobre estimación de la cuantificación del riesgo, pudiendo generar una prima menos competitiva y atractiva para los clientes. Sin embargo, la evidencia empírica muestra que en el momento que ocurre un siniestro suelen activarse más de una cobertura (Anderson et al., 2007). Esto muestra que existe un nivel de dependencia que debe considerarse cuando las primas por cobertura son agregadas. En el presente trabajo también abordamos este problema que existe en la industria de seguros, incorporando cópulas para modelar la dependencia y estimar mejor es riesgo agregado.

Ante lo expuesto, se propone construir modelos más eficientes para la estimación de la prima pura de riesgo, considerando los efectos de la dependencia que existe entre riesgos que se quieren coberturar. Para la modelación conjunta de la frecuencia y severidad por cobertura proponemos utilizar modelos lineales generalizados basados en cópulas mixtas, y para la agregación de las primas de riesgo por cobertura construimos una cópula multivariada que logre agregarlos a un solo valor, es decir, llegar a la prima pura de riesgo total.

1.2 Formulación del problema

1.2.1 Problema general

- ¿Es posible construir una metodología de cálculo de la prima pura de riesgo sin suponer la independencia entre la frecuencia y severidad? ¿Es posible agregar diferentes riesgos mediante cópulas por cobertura de tal forma que nos permita obtener estimaciones confiables y más justas de la prima de riesgo para los asegurados frente al supuesto usual de independencia?

1.2.2 Problemas específicos

- ¿Es posible determinar un modelo lineal generalizado conjunto de la frecuencia y de la severidad para estimar la prima pura de riesgo de una determinada cobertura?
- ¿Cómo podemos medir el grado de dependencia entre la frecuencia y la severidad, e incorporarla en el modelo?
- ¿Es posible determinar una propuesta metodológica para el cálculo de la prima total mediante agregación de los riesgos asociados de cada cobertura mediante funciones cópulas?
- ¿Cómo podemos medir el grado de dependencia entre las primas por cobertura para el cálculo de la prima total?

1.3 Objetivos de la investigación

1.3.1 Objetivo General

- Proponer una propuesta metodológica del cálculo de la prima pura de riesgo sujeto a dependencia entre los riesgos de frecuencia y severidad, y agregados mediante cópulas por cobertura que nos permita estimaciones confiables y de acuerdo al riesgo especifico de cada asegurado frente supuesto usual de independencia.

1.3.2 Objetivos Específicos

- Determinar un modelo de regresión generalizado conjunto de la frecuencia y de la severidad para estimar la prima pura de riesgo de una determinada cobertura.
- Medir el grado de dependencia entre la frecuencia y la severidad.
- Determinar una propuesta metodológica de cálculo de la prima total mediante agregación de los riesgos asociados de cada cobertura mediante funciones cópulas.
- Medir el grado de dependencia entre las primas por cobertura para el cálculo de la prima total o prima agregada.

1.4 Justificación, alcances y limitaciones de la investigación

1.4.1 Justificación

La importancia de la presente investigación radica en que actualmente en la industria de seguros la mayoría de las empresas aseguradoras, y sobretodo en mercados emergentes como el nuestro, donde la profesión de actuario es cada vez más demandada, las técnicas suelen limitarse al uso de conceptos teóricos básicos y de uso muy práctico que no necesariamente son correctas o que asumen demasiados supuestos que no son realistas, generando un sesgo en las estimaciones de la prima de riesgo. El presente trabajo pretende reducir esta brecha al mostrar que se pueden aplicar conceptos teóricos más sofisticados de matemática actuarial, estadística y algoritmos de optimización para la construcción de modelos más eficientes para el cálculo de la prima de riesgo, desde el punto de vista estadístico. Además, la metodología descrita en el presente trabajo permite obtener una prima de acuerdo al riesgo específico de cada asegurado, y a la vez atractiva para los asegurados que adquieren un producto de seguro, puesto que al considerar la dependencia entre la severidad, la frecuencia y los riesgos coberturados el método para el cálculo de prima propuesto no usa la misma información que puede estar contenida en diferentes tipos de riesgos, generando una menor prima respecto al supuesto de independencia, como lo demuestra diversos trabajos en la literatura (Anderson et al., 2007; Czado et al., 2012; Garrido, Genest, y Schulz, 2016; Krämer, Brechmann, Silvestrini, y Czado, 2013; Lee, Park, y Ahn, 2019; P. Song et al., 2009; P. X.-K. Song, 2000).

1.4.2 Alcances

- Para el modelamiento de la dependencia utilizaremos solo las cópulas de Gauss, Clayton, Gumbel y Frank; siendo estas familias de cópulas las de mayor uso.
- Si bien pueden existir diversas coberturas en producto de seguros, el presente trabajo solo abarcará las coberturas de Pérdida Parcial, Pérdida Total, Responsabilidad Civil y Asistencias en seguros vehiculares.
- Se mostrará el efecto de la inclusión del efecto de la dependencia en la distribución de la prima pura de riesgo y de la pérdida total observada.

1.4.3 Limitaciones

- Nuestro análisis no se incluye algún esquema de recargos por gastos y márgenes de utilidad sobre la prima pura de riesgo total, que es objeto del presente trabajo. Los recargos obedecen a estrategias comeciales de acuerdo a sus canales de venta, y no son de naturaleza estocástica.
- La propuesta del presente trabajo de investigación es sólo aplicable a productos de seguros diferentes a vida.

- Los modelos predictivos del presente trabajo no se implementarán en sistemas informáticos empresariales de tal manera que nos permita observar en tiempo real su desempeño. Sin embargo, realizamos simulaciones mediante plantillas de hojas de cálculo para ver su aplicabilidad y desempeño.
- EL costo computacional de la implementación del algoritmo desarrollado en este trabajo es alto, y requiere de cierta expertise para su implementación.

CAPÍTULO II: Marco Teórico Conceptual

2.1 Antecedentes de la investigación

La Teoría de Riesgo Colectivo, introducida en (Lundberg, 1903), proporciona una marco científicamente aceptado por la academia y la industria para el planteamiento y resolución de los principales problemas y desafios que presentan las actividades de la industria aseguradora, donde la incertidumbre es una de las principales características que la definen (Kaas, Goovaerts, Dhaene, y Denuit, 2008). La siniestralidad es la variable aleatoria característica del negocio asegurador, cuyo análisis parte en entender sus componentes básicos: El número de siniestros y la cuantía de un siniestro. Bajo el supuesto de independencia de estas componentes o variables aleatorias, el valor esperado de la siniestralidad o más conocida como la prima pura de riesgo se obtiene como el producto de ambos (McNeil, Frey, y Embrechts, 2005; Ohlsson y Johansson, 2010; Sikov y Cerda-Hernández, 2021).

Sin embargo, dentro de las características de los riesgos asegurables por las compañías de seguros, es que estos deben ser homogéneos, con la finalidad de obtener una prima suficiente y competitiva que garantice la solvencia de la empresa. Es en ese sentido que los modelos lineales generalizados son herramientas muy utilizadas cuando se requiere incluir covariables o factores de riesgo que nos permitan segmentar la cartera total de una empresa de seguros en subsegmentos homogéneos, donde la cuantía de la prima pura de riesgo será proporcional a su grado de siniestralidad. El primer ejemplo aplicado a la industria de seguros donde se usan los GLM, específicamente a seguros de motores, fue presentado en (McCullagh y Nelder, 1989), y fue popularizado en la industria de seguros después de los pioneros trabajos de Renshaw (Renshaw, 1994) y Brockman y Wright (Brockman y Wright, 1992), donde se reconoce el potencial de los GLM como una herramienta de modelado integral para la frecuencia y la severidad en presencia de covariables. Además, estos modelos prestan especial atención a una diversa variedad de distribuciones disponibles, y a las técnicas de estimación de parámetros mediante conceptos de *cuasi-likelihood*.

Desde la introducción de los GLM para el análisis de carteras de seguros hecha por Renshaw, Brockman y Wright, los modelos lineales generalizados son y siguen siendo una de las técnicas estadísticas más utilizadas para la construcción de modelos multivariados para la frecuencia y para la severidad (Anderson et al., 2007; Czado et al., 2012; Lee et al., 2019). En la práctica, la evidencia empírica muestra que la frecuencia y la severidad son a menudo dependientes, y es por ello que existe la necesidad de incluir en los modelos la asociación potencial o dependencia existente (Czado et al., 2012; Lee et al., 2018; Ohlsson y Johansson, 2010; P. Song et al., 2009). Una forma de estudiar y capturar la dependencia entre variables aleatorias fue introducida en los trabajos pioneros de Sklar (Sklar, 1959), y específicamente aplicado a la industria de seguros en (Garrido et al., 2016), donde los autores consideran el número de siniestros como una covariable adicional en el modelamiento del costo medio, es

decir, lo que los autores proponen en dicho trabajo es construir una distribución condicional de la severidad respecto de la frecuencia, y que en términos prácticos, la prima pura de riesgo resulta como el producto de tres elementos: la media de la distribución marginal de la frecuencia, la media de la distribución modificada de la severidad y un factor de corrección por depedencia. En la misma dirección, (Frees, Lee, y Yang, 2016) construyen primero la distribución conjunta de la severidad y la frecuencia usando la distribución de la frecuencia y la distribución condicional de la severidad al centrarse en el uso de una cópula para modelar la dependencia entre estos.

Desde que se comprobó que el coeficiente de correlación lineal de Pearson no es un instrumento adecuado para medir relaciones de dependencia entre riesgos, el uso de las cópulas juega un papel importante para modelar dependencia, puesto que superan muchas de las limitaciones del coeficiente de correlación de Pearson. Las cópulas fueron presentadas y caracterizadas para variables continuas por Sklar (Sklar, 1959), quién resolvió algunos de los problemas formulados por M. Fréchet sobre la relación entre una función de distribución de probabilidad multidimensional y sus marginales de menor dimensión. Actualmente, la teoría de cópulas se ha convertido en una poderosa herramienta de modelado multivariado en muchos campos donde la dependencia entre varias variables aleatorias marginales, continuas o discretas, es de gran interés, y para los cuales la suposición de normalidad multivariada puede ser cuestionable, supuesto utilizado para usar el coeficiente de correlación de Pearson. Es por ello que existen investigaciones donde modelan la frecuencia y la severidad de forma conjunta mediante modelos lineales generalizados basados en cópula mixta aplicados a datos de Salud (de Leon y Wu, 2011) y a datos de automóviles (Czado et al., 2012; Frees et al., 2016). Otros trabajos interesantes, basados en las investigaciones anteriores, pero ampliando su aplicación sobre un mayor número de familias de cópulas se pueden encontra en (Czado et al., 2012) para estimar la pérdida total, y en (Kholifah, Lestari, y Devila, 2019) para estimar la prima pura de riesgo. Ambos trabajos proponen algoritmos de optimización en la estimación de los parámetros y procedimientos adecuados para la selección óptima de la cópula.

Finalmente, si bien en este trabajo nosotros utilizamos diferentes modelos ya desarrollados en la literatura de cópulas para realizar el modelamiento conjunto de la frecuencia y de la severidad para el cálculo de la prima pura de riesgo; en la presente tesis también proponemos una metodología para resolver un problema actual que se tiene en la industria de seguros en el Perú: calcular la prima pura de riesgo total de un seguro que ofrece varias coberturas simultaneamente. Este problema se observa empíricamente al activarse varias coberturas de forma simultanea ante un mismo siniestro. Sabemos que las distribuciones de la frecuencia y de la severidad se relacionan de manera distinta por cada cobertura, obteniendo una prima pura de riesgo por cada una de ellas, pero el problema es que actualmete la industria está tarifando un seguro con varias coberturas haciendo una suma simple de la primas obtenidas, sin incluir el grado de dependencia que pueda existir entre ellas. En este trabajo propone-

mos una metodología para tarifar este tipo de seguro incluyendo la dependencia entre ellas y construyendo una función conjunta multivariada (Aas, Czado, Frigessi, y Bakken, 2009) que logré agregar dichas primas incluyendo la dependencia entre ellas. De esta manera obtendremos primas precisas que garanticen la suficiencia de la compañía, con primas competitivas para el mercado de seguros.

2.2 Bases teóricas de la investigación

2.2.1 Modelos Lineales Generalizados

Como la teoría de Modelos Lineales Generalizados es una teoría bien estudiada actualmente en estadística, para el desarrollo de la base metodológica de este trabajo seguiremos De Jong, Heller, et al. (2008), Rao, Shalabh, Toutenburg, y Heumann (2010), Ohlsson y Johansson (2010).

Estos modelos son utilizados para calcular y cuantificar la relación entre la variable de respuesta y las variables explicativas. Estos modelos difieren de un modelo de regresión clásicos en dos importantes aspectos:

1. La distribución de la variable de respuesta es elegida de una familia de distribuciones, denominada *familia exponencial*. Por lo tanto, la distribución de la variable de respuesta no necesita ser Normal o cercana a ello, o puede ser explícitamente no Normal.

2. Una transformación de la media de la variable respuesta es linealmente relacionada a las variables explicativas.

La consecuencia de permitir que la variable respuesta sea parte de la familia exponencial es que la respuesta puede ser, y suele ser, heterocedástica. Así, la varianza variará con la media que, a su vez, puede variar con las variables explicativas. Esto contrasta con el supuesto homocedástico de los modelos de regresión con residuo normal. Los modelos lineales generalizados son ámpliamente utilizados en la industria de seguros, puesto que el supuesto de normalidad no aplica a ese tipo de datos. Por ejemplo, la distribución de las pérdidas generadas por los siniestros no sigue una distribución Normal (para más detalles ver Rao et al. (2010), Ohlsson y Johansson (2010)).

2.2.1.1 Modelos lineales generalizados

Los Modelos lineales generalizados (GLM por sus siglas en inglés) son una generalización de los clásicos modelos lineales de análisis de regresión y análisis de varianza. Los parámetros de estos modelos son estimados de acuerdo al principio de mínimos cuadrados y son optimizados de acuerdo a la teoría de dispersión mínima (Rao et al., 2010).

Según McCullagh y Nelder (1989), un modelo GLM consiste de tres componentes:

1. La *componente aleatoria*, que especifica la distribución de probabilidad de la variable de interés,

2. El *componente sistemático*, que especifica una función lineal de las variables explicativas,

3. La *función de enlace*, que describe una relación funcional entre el componente sistemático y la esperanza de la componente aleatoria.

Las tres componentes pueden ser especificadas de forma compacta de la siguiente manera:

La componente aleatoria consiste de una variable respuesta y, donde la distribución pertenece a la familia exponencial, es decir, la función de densidad de probabilidad de cada observación de la variable y tiene la forma:

$$f(y_i) = c(y_i, \phi) exp\left[\frac{y_i\theta - a(\theta)}{\phi}\right], \qquad g(\mu) = x^{'}\beta. \tag{2.1}$$

donde θ y ϕ son los parámetros. El parámetro θ es llamado parámetro canónico y ϕ es el parámetro de dispersión. Cualquier función de probabilidad que pueden ser escrito como en (2.1) se dicen que pertenecen a la familia exponencial. En términos de $a(\theta)$ podemos calcular la media y la varianza de y:

$$E(y) = \dot{a}(\theta), \quad Var(y) = \phi\ddot{a}(\theta),$$

donde $\dot{a}(\theta)$ y $\ddot{a}(\theta)$ son la primera y segunda derivada de $a(\theta)$ respecto de θ, respectivamente. En la Tabla 2.1 se muestra las diferentes formas de θ y $a(\theta)$ para algunas distribuciones que pertenecen a las distribuciones de la familia exponencial.

Aquí no mostramos la forma de la expresión $c(y, \phi)$ dado que no suele ser de interés en la mayoría de situaciones y aplicaciones. En el caso del parámetro de dispersión de (2.1) suele expresarse como ϕ/w, donde w es un ponderador. Esta forma es apropiada cuando tenemos datos agrupados. La segunda ecuación de (2.1) nos dice que una transformación de la media de la variable y, $g(\mu)$, es igual a una combinación lineal de las variables explicativas contenidas en el vector x, donde g es la función de enlace. Algunas observaciones sobre el modelo GLM son:

- La elección de la función $a(\theta)$ determina la distribución de la variable de respuesta.
- La elección de enlace $g(\mu)$ determina cómo la media está relacionada con las variables explicativas x. En el modelo de regresión lineal general, la relación entre la media de y y las variables explicativas es $\mu = x^{'}\beta$. En cambio, en los modelos GLM, esto se generaliza como $g(\mu) = x^{'}\beta$, donde g es una función monótona y diferenciable (tal como la función logaritmo o la función raiz cuadrada).

TABLA N° 2.1: Distribuciones de la familia exponencial y sus parámetros

Función de Distribución	θ	$a(\theta)$	ϕ	$E(y)$	$V(\mu)=\frac{Var(y)}{\phi}$
Binomial(n,π)	$\ln\frac{\pi}{1-\pi}$	$n\ln(1+e^{\theta})$	1	$n\pi$	$n\pi(1-\pi)$
Poisson(μ)	$\ln\mu$	e^{θ}	1	μ	μ
Normal(μ,σ^2)	μ	$\frac{1}{2}\theta^2$	σ^2	μ	1
Gamma(μ,ν)	$-\frac{1}{\mu}$	$-\ln(-\theta)$	$\frac{1}{\nu}$	μ	μ^2
Gamma Inversa(μ,σ^2)	$-\frac{1}{2\mu^2}$	$-\sqrt{-2\theta}$	σ^2	μ	μ^3
Binomial Negativa(μ,κ)	$\ln\frac{\kappa\mu}{1+\kappa\mu}$	$-\frac{1}{\kappa}\ln(1-\kappa e^{\theta})$	1	μ	$\mu(1+\kappa\mu)$

- De la ecuación (2.1) observamos que dado x, el parámetro μ es determinado a través de la relación $g(\mu)$. De igual forma dado μ, el parámetro θ es determinado a través de $a(\theta)=\mu$. Finalmente dado θ, y es determinada como un extracto de la densidad exponential espeficicada en $a(\theta)$.
- Las observaciones del vector y se asumen independientes.

2.2.1.2 Pasos en el modelado lineal generalizado

Dada una variable de respuesta y, la construcción de un GLM consta de los siguientes pasos:

1. Elija una distribución de respuesta $f(y)$ y, por lo tanto, elija $a(\theta)$ en (2.1). La distribución de la variable de respuesta se adapta a la situación dada.
2. Seleccione una funcion enlace $g(\mu)$. Esta elección a veces se simplifica eligiendo el llamado enlace "canónico" correspondiente a cada distribución de la variable de respuesta.
3. Elija las variables explicativas x en términos de las cuales se va a modelar $g(\mu)$. Se aplican consideraciones similares a las del modelo de regresión lineal ordinaria.
4. Recoja las observaciones $y_1,...,y_n$ sobre la respuesta y y los valores correspondientes $x_1,...,x_n$ sobre las variables explicativas x. Se supone que las observaciones sucesivas son independientes, es decir, la muestra se considerará como una muestra aleatoria simple de la población.

5. Ajuste el modelo estimando los parámetros contenidos en β y, si se desconoce, también estimadmos el parámetro ϕ. Para estimar los parámetros usamos el método de máxima verosimilitud.

6. Dada la estimación de β, genere predicciones (o valores ajustados) de y para diferentes configuraciones de x, y examine qué tan bien se ajusta el modelo examinando la desviación de los valores ajustados de los valores reales, así como otros diagnósticos del modelo. También se usará el valor estimado de β para ver si las variables explicativas dadas son o no importantes para determinar μ.

Los pasos adicionales en comparación con el modelo de regresión lineal ordinario son elegir la función $a(\theta)$ (lo que implica la distribución de la variable de respuesta) y la función enlace $g(\mu)$. La elección $a(\theta)$ está guiada por la naturaleza de la variable de respuesta. La elección de la función enlace viene sugerida por la forma funcional de la relación entre la respuesta y las variables explicativas.

En ejemplo reales, los pasos anteriores rara vez proceden en dicha secuencia, especialmente con datos de seguros. Los datos a menudo se recopilan antes de la especificación de un modelo. Por ejemplo, en la industria de los seguros, todas las pólizas de un tipo particular son colectadas y analizadas y modeladas posteriormente. Es probable que la exploración inicial de los datos sugiera diferentes modelos y diferentes distribuciones para la variable respuesta. Los ajustes iniciales a menudo van seguidos de refinamientos adicionales de ajustes, en los que algunas de las variables explicativas pueden descartarse o transformarse. Los datos pueden ser maquillados juiciosamente por la exclusión de varios casos, o la incorporación de diferentes efectos.

2.2.1.3 Función de enlace y enlace canónico

Las funciones de enlace $g(\mu)$ más utilizadas en las aplicaciones a datos reales son mostradas en la Tabla 2.2. Con la excepción de la función de enlace para el modelo logit, el resto de funciones de enlace son de la forma $g(\mu) = \mu^p$, siendo el caso logarítmico el límite de $(y^p - 1)/p$ cuando $p \to 0$.

Si $g(\mu) = \theta$, entonces g se denomina el enlace canónico correspondiente a $a(\theta)$. En este caso tenemos que θ es una combinación lineal de las variables explicativas, es decir, $\theta = x^{'}\beta$. Elegir el enlace canónico correspondiente a una distribución de respuesta f simplifica computacionalmente el proceso de estimación de los parámetros, aunque en la actualidad con la capacidad computacional moderna que se tiene esto ya no es una consideración primordial que se toma en cuenta al momento de usar el modelo GLM. En la Tabla 2.2 se muestran las distribuciones de respuesta para las cuales los enlaces comúnmente utilizados son canónicos. Las constantes en θ generalmente se omiten del enlace canónico.

TABLA N° 2.2: Funciones enlaces comunes

Función enlace	$g(\mu)$	Distribución
Identidad	μ	Normal
Logaritmo	$ln\mu$	Poisson
Potencia	μ^p	Gamma $(p=-1)$
Raiz cuadrada	$\sqrt{\mu}$	
Logit	$ln\frac{\mu}{1-\mu}$	Binomial

2.2.1.4 Variable *Offset*

Cuando la variable respuesta es una variable de conteo, como el número de siniestros en una cartera de seguros, o el número de muertes en un grupo de riesgo, requieren de un factor de corrección para el número de expuesto al riesgo n. Si μ es la media de la variable de conteo y, entonces la tasa de ocurrencia o frecuencia μ/n de nuestra variable de interés se expresa como:

$$g\left(\frac{\mu}{n}\right) = x^{'}\beta$$

Cuando g es la función logarítimica, esto se expresa como:

$$ln\left(\frac{\mu}{n}\right) = x^{'}\beta \Rightarrow \ln\mu = \ln(n) + x^{'}\beta$$

En la expresión de arriba n es llamado exposición y $\ln(n)$ es denominada la variable *offset*. El *offset* se puede interpretar como otra variable de x en la regresión, colocando un coeficiente β igual a la unidad.

Incluyendo el *offset*, y tiene un valor esperado directamente proporcional a la exposición:

$$\mu = ne^{x^{'}\beta}$$

Los *offsets* se utilizan para corregir el tamaño del grupo o los diferentes períodos de tiempo de observación o de exposición.

2.2.1.5 Estimación por Máxima Verosimilitud

La estimación por máxima verosimilitud (MLE por sus siglas en inglés) de los parámetros β y ϕ son obtenidos de la maximización de la función *log-likelihood* que se define de la siguiente manera:

$$l(\beta,\phi) = \sum_{i=1}^{n} \ln f(y_i;\beta,\phi) = \sum_{i=1}^{n}\left[\ln c(y_i,\phi) + \frac{y_i\theta_i - a(\theta_i)}{\phi}\right] \tag{2.2}$$

el cual asume que las observaciones de la variable respuesta y_i son independientes y distribuidas con una función de densidad que pertenece a la familia exponencial.

Para la estimación del parámetro β_j mediante MLE tenemos que derivarla función $l(\beta,\phi)$ respecto de β_j:

$$\frac{\partial l}{\partial \beta_j} = \sum_{i=1}^{n} \frac{\partial l}{\partial \theta_i}\frac{\partial \theta_i}{\partial \beta_j}$$

donde

$$\frac{\partial l}{\partial \theta_i} = \frac{y_i - a(\theta_i)}{\phi} = \frac{y_i - \mu_i}{\phi}, \quad \frac{\partial \theta_i}{\partial \beta_j} = \frac{\partial \theta_i}{\partial \eta_i}\frac{\partial \eta_i}{\partial \beta_j} = \frac{\partial \theta_i}{\partial \eta_i}x_{ij}.$$

Aquí $\eta_i = x_i^{'}\beta$ y x_{ij} es el componente i-ésimo de x_j. Establecer $\partial l/\partial \beta_j = 0$ produce la condiciones de primer orden para la maximización de la función de verosimilitud:

$$\sum_{i=1}^{n} \frac{\partial \theta_i}{\partial \eta_i} x_{ij}(y_i - \mu_i) = 0 \quad \Longleftrightarrow \quad X^{'}D(y-\mu) = 0 \tag{2.3}$$

Donde D es la matriz diagonal cuyos valores en la diagonal son $\partial\theta_i/\partial\eta_i$,

$$\left(\frac{\partial \theta_i}{\partial \eta_i}\right)^{-1} = \frac{\partial \eta_i}{\partial \theta_i} = \frac{\eta_i}{\partial \mu_i}\frac{\partial \mu_i}{\partial \theta_i} = \dot{g}(\mu_i)\ddot{a}(\theta_i) = \dot{g}(\mu_i)V(\mu_i)$$

Por lo tanto, D es una matriz diagonal con elementos en la diagonal dados por $(\dot{g}(\mu_i)V(\mu_i))^{-1}$. Las ecuaciones en (2.3) son denominadas a menudo ecuaciones de estimación para β. Note que β está implícito en estas ecuaciones, trabajando implícitamente a través de μ y D. Definimos las matrices diagonales G y W con entradas diagonal dadas por $\dot{g}(\mu_i)$ y $\left[(\dot{g}(\mu_i))^2 V(\mu_i)\right]^{-1}$, respectivamente. Entonces $D = WG$ y la ecuación (2.3) es equivalente a:

$$X^{'}WG(y-\mu) = 0 \tag{2.4}$$

Relación con los mínimos cuadrados ponderados: Utilizando una proximación mediante series de Taylor tenemos la siguiente aproximación

$$g(y_i) \approx g(\mu_i) + \dot{g}(\mu_i)(y_i - \mu_i) \quad \Rightarrow \quad g(y) \approx g(\mu) + G(y-\mu) \tag{2.5}$$

donde $g(y)$ es un vector con elementos $g(y_i)$. De forma similarmente tenemos la aproximación para $g(\mu)$. Reordenando y sustituyendo $\mu = X\beta$ obtenemos $G(y-\mu) \approx g(y) - X\beta$. Sustituyendo esta relación dentro de la ecuación (2.4) se obtiene la ecuación de estimación aproximada

$$X^{'}Wg(y) - X'WX\beta \approx 0 \quad \rightarrow \quad \hat{\beta} \approx \left(X^{'}WX\right)^{-1} X^{'}Wg(y) \tag{2.6}$$

De la relación de la izquierda en (2.5), $(\dot{g}(\mu_i))^2 V(\mu_i)$ es la aproximación de primer orden

de la varianza de $g(y_i)$, además del factor ϕ. Así, la resolución de la ecuación de estimación de (2.4) corresponde, aproximadamente, a la regresión ponderada de las respuestas transformadas $g(y_i)$ con pesos proporcionales a las varianzas.

Con un enlace Identidad $g(y_i) = y_i$, la ecuación (2.5) es exacta. Entonces,

$$\hat{\beta} = \left(X'WX\right)^{-1} X'Wy$$

donde W tiene diagonal con elementos $1/V(\mu_i)$. Por lo tanto, con una función enlace Identidad y $V(\mu_i)$ independiente de μ_i, la estimación máxima verosimil para β es el estimador mínimos cuadrados ponderados (para más detalles ver McCullagh y Nelder (1989) y Rao et al. (2010)).

Método de Newton-Raphson: Obtener soluciones explícitas o cerradas de las condiciones de primer orden (2.3) para maximizar la función de verosimilitud son usualmente díficiles, excepto para casos muy particulares como la Normal con enlace Identidad. Una forma de resolver este problema es recurrir a métodos numéricos. El método más utilizado para resolver las ecuaciones que aparecen del método de MV es el método de Newton, donde se supone que la primera y segunda derivada de la función a maximizar pueden evaluarse fácilmente en cada punto. Usando estas derivadas, se construye una aproximación cuadrática a la función de verosimilitud. El maximizador resultante del método de Newton se usa luego para derivar una nueva ecuación cuadrática que, a su vez, se maximiza. A menudo se encuentra que esta secuencia de maximizadores aproximados converge rápidamente al máximo buscado (Burden, Faires, y Burden, 2015).

Para simplificar la discusión supongamos que ϕ es conocido y denotemos por $l(\beta)$ el *log-likelihood* como una función del vector parámetro desconocido β. Si β contiene un único parámetro entonces la aproximación de la serie de Taylor en cualquier punto β es dado por

$$l(\beta+\delta) \approx l(\beta) + \dot{l}(\beta)\delta + \frac{\delta^2}{2}\ddot{l}(\beta)$$

Derivando respecto de δ e igualando a cero se obtiene

$$\dot{l} + \delta\ddot{l}(\beta) = 0 \quad \Rightarrow \quad \delta = -\left[\ddot{l}(\beta)\right]^{-1}\dot{l}(\beta)$$

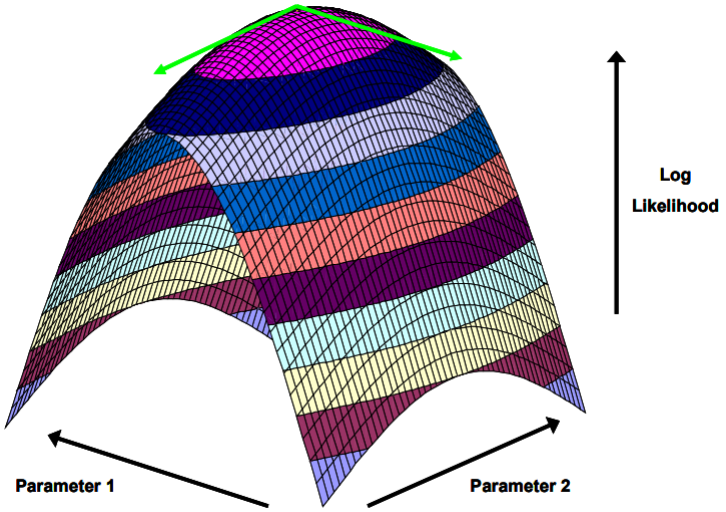
Con β dado y δ especificado, un punto más alto parece ser $\beta - \dot{l}(\beta)/\ddot{l}(\beta)$. Denotando $\beta^{(m)}$ como el valor de β en la iteración m, la ecuación de actualización es

$$\beta^{(m+1)} = \beta^{(m)} - \left[\ddot{l}(\beta^{(m)})\right]^{-1}\dot{l}(\beta^{(m)}) \tag{2.7}$$

La sucesión de esta ecuación converge a β. Además, como se mencionó anteriormente, esta sucesión converge rápidamente. El método puede ser adaptado cuando β es un vector. En este caso se realiza una aproximación cuadrática a la superficie de $l(\beta)$, y es esta supeficie

cuadrática la que se maximiza

FIGURA N° 2.1: Superficie máxima Log-Verosimilitud



Fuente: Imagen tomada de Anderson et al. (2007)

La ecuacion de actualización vista en (2.7) con la inversa interpretada como inversión de matriz, y donde $\dot{l}(\beta)$ es el vector de derivadas parciales $\partial l/\partial\beta_j$. El vector $\dot{l}(\beta)$ es llamado el *vector score*, y $\ddot{l}(\beta)$, la matriz de derivadas parciales cruzadas $\partial^2 l/(\partial\beta_j\partial\beta_k)$, la matriz *Hessiana*. La condición para un máximo es que la matriz *Hessiana* sea definida no positiva, es decir, $-(\ddot{\beta})$ no sea definida negativa. El procedimiento de evaluar repetidamente el *score* y la matriz de *Hessiana* para actualizar la estimación en (2.7), se denomina iteración de Newton-Raphson.

Fisher Scoring: Fisher sugirió reemplazar $\ddot{l}(\beta)$ en (2.7) por su esperado $E(\ddot{l}(\beta))$. También demostró que dado β, $E(\ddot{l}(\beta)) = -E(\dot{l}(\beta)\dot{l}'(\beta))$. Estos resultados son ciertos para todos los *log-likelihoods*, no solo para aquellas de la familia exponencial. La matriz $-E(\ddot{l}(\beta))$ es llamada la matriz de información de Fisher. Para los modelos GLM la matriz de información es

$$E[\dot{l}(\beta)\dot{l}'(\beta)] = \phi^{-2}X'DE[(y-\mu)(y-\mu)']DX = \phi^{-1}X'WX \tag{2.8}$$

Por lo tanto, sustituyendo $\ddot{l}(\beta)$ por su esperado en (2.7) y utilizando la ecuación (2.4) para $\dot{l}(\beta)$ obtenemos

$$\beta^{(m+1)} = \beta^{(m)} + \left[X'WX\right]^{-1}X'WG(y-\mu)$$

Esta ecuación suele reescribirse como

$$\beta^{(m+1)} = \left[X'WX\right]^{-1}X'W\left[X\beta^{(m)} + G(y-\mu)\right], \tag{2.9}$$

donde la expresión entre corchetes a la derecha se denomina ”variable dependiente local”. La ecuación (2.9) es similar a la regresión de mínimos cuadrados ponderados. La variable de-

pendiente local es calculado reemplazando μ por el estimado $\mu^{(m)}$ donde $g(\mu^{(m)}) = X\beta^{(m)}$. Calculando $\beta^{(m+1)}$, se requiere $V(\mu)$, la función varianza de la distribución y $\dot{g}(\mu)$, la derivada de la función enlace. El parámetro de dispersión ϕ no es requerido.

Para n grande, la inversa de la matriz de información de Fisher es aproximadamente la matriz de covarianza de $\hat{\beta}$. Además, los MLE son asintóticamente insesgados y, otra vez para muestras grandes, aproximadamente normal. Por lo tanto,

$$\hat{\beta} \approx N\left[\beta, \phi(X'WX)^{-1}\right].$$

Este resultado asintótico es la base para la prueba de significancia de Wald, que será discutido en la sección 2.2.1.8. En la práctica W es evaluado en $\hat{\beta}$.

Estimación de la dispersión: El parámetro de dispersión ϕ puede ser estimado por MLE o por el método de los momentos. MLE implica la solución iterativa de la ecuación $\partial l/\partial\phi = 0$, el cual es diferente para cada distribución de la variable de respuesta. La estimación de ϕ es generalmente considerada periférica al GLM.

2.2.1.6 Predicción e intervalos de confianza

Dada las variables explicativas x, el valor estimado de la media de y es $\hat{\mu}$, donde $g(\hat{\mu}) = x'\hat{\beta}$. Por ejemplo, con un enlace logarítmico:

$$\ln\hat{\mu} = x'\hat{\beta} \quad \Rightarrow \quad \hat{\mu} = e^{x'\hat{\beta}}.$$

Un intervalo de confianza alrededor del valor estimado es utilizado para indicar precisión. El cálculo del intervalo de confianza requiere de la distribución muestral de $\hat{\mu}$. La varianza del predictor lineal $x'\hat{\beta}$ es

$$Var(x'\hat{\beta}) = \phi x'(X'WX)^{-1}x$$

Por lo tanto, una aproximación del intervalo de confianza para la media, denotado por (μ_l, μ_u), viene dado por

$$g(\mu_l) = x'\hat{\beta} - z\sqrt{\phi x'(X'WX)^{-1}x} \quad , \quad g(\mu_u) = x'\hat{\beta} + z\sqrt{\phi x'(X'WX)^{-1}x}$$

donde z es un punto apropiado de la distribución normal estándar $N(0,1)$. El intervalo de confianza es exacto solo en el caso de un enlace identidad y una variable de respuesta Normal. En otros casos se trata solo de una aproximación, cuya precisión mejora al aumentar el tamaño de la muestra. El parámetro de dispersión ϕ se reemplaza por una estimación, lo que genera más errores de aproximación. El estimador $\hat{\mu}$ es insesgado cuando se utiliza el enlace identidad. Para otros enlaces resulta ser sesgado. Para ilustrar el problema, usamos

el enlace logarítmico y asumimos que $\hat{\beta}$ es aproximadamente normal. Entonces,

$$E(e^{x^{'}\hat{\beta}}) = exp\left[x^{'}\beta + \frac{1}{2}Var(x^{'}\hat{\beta})\right] \neq e^{x^{'}\beta} = \mu$$

El sesgo se incrementa con la varainza $Var(x^{'}\hat{\beta})$. Además, para muestras grandes se sabe que $Var(\hat{\beta})$ es pequeño y se espera que el sesgo sea insignificante (McCullagh y Nelder, 1989; Rao et al., 2010). El intervalo (μ_l, μ_u) es un intervalo de confianza para la media de y dado x. El ancho de este intervalo de predicción toma en cuenta la incertidumbre asociada tanto con $\hat{\mu}$ como con el resultado de la distribución de respuesta.

2.2.1.7 Evaluación de ajustes y Devianza

La bondad del ajuste de un modelo a los datos es una pregunta natural que surge con todos los modelos estadísticos. Los principios de las pruebas de significancia, la selección de modelos y las pruebas de diagnóstico, analizados en los modelos de regresión Normal, son los mismos para los GLM. Sin embargo, algunos detalles técnicos de los métodos difieren en algo.

Una forma de evaluar el ajuste de un modelo dado es compararlo con el modelo con el mejor ajuste posible. El mejor ajuste se obtendrá cuando haya tantos parámetros como observaciones: a esto se le llama modelo saturado. Un modelo saturado asegurará que haya flexibilidad completa en el ajuste de θ_i. Desde que

$$\frac{\partial l}{\partial \theta_i} = \frac{y_i - \mu_i}{\phi} = \frac{y_i - \dot{a}(\theta_i)}{\phi},$$

el MLE de θ_i bajo el modelo saturado es $\breve{\theta}_i$, donde $\dot{a}(\breve{\theta}_i) = y_i$. Por lo tanto, el valor ajustado es igual a la observación y el modelo saturado se ajusta perfectamente. El valor del $log-likelihood$ saturado es

$$\breve{l} \equiv \sum_{i=1}^{n}\left[\ln c(y_i, \phi) + \frac{y_i\breve{\theta}_i - a(\breve{\theta}_i)}{\phi}\right],$$

el cual es el posible máximo $log-likelihood$ para y dado la distribución de respuesta especificada por $a(\theta)$. Este valor es comparado a $\hat{l}$, el valor máximo del $log-likelihood$ basado en y y las variables explicativas dadas. La Devianza denotada como Δ, es definido como una medida de distancia entre el modelo saturado y el modelo ajustado:

$$\Delta \equiv (\breve{l} - \hat{l}).$$

- Cuando el modelo proporciona un buen ajuste, entonces se espera que $\hat{l}$ sea cercano a (pero no mayor que) $\breve{l}$. Un valor grande de la Devianza indica un modelo mal ajustado.
- El tamaño de Δ se evalúa en realación con la distribución χ^2_{n-p}. Esta es la distribu-

ción de muestreo aproximada de la desviación, suponiendo que el modelo ajustado es correcto y n es grande. El valor esperado de la Devianza es $n-p$. La regla de decisión para concluir si el modelo está mal ajustado es dividir Δ por sus grados de libertad $n-p$, y ver si se obtiene un valor mucho mayor que uno, de acuerdo al nivel crítico dado por el nivel significancia utilizado.

- Un cálculo directo muestra que para la familia exponencial

$$\Delta = 2\sum_{i=1}^{n}\left[\frac{y_i(\breve{\theta}_i - \hat{\theta}_i) - a(\breve{\theta}_i) + a(\hat{\theta}_i)}{\phi}\right] \tag{2.10}$$

ya que los términos que involucran $c(y_i, \phi)$ se cancelan, y donde $\breve{\theta}_i$ y $\hat{\theta}_i$ son tales que $\dot{a}(\breve{\theta}_i) = y_i$ y $g\left[\dot{a}(\hat{\theta}_i)\right] = x_i'\hat{\beta}$, respectivamente.

- Cuando ϕ es desconocido y reemplazado por su estimador, entonces la distribución para la Devianza se ve comprometida. Para el caso de la distribución de Poisson tenemos que $\phi = 1$, y la aproximación χ^2 es útil. En el caso de la distribución normal si se conoce σ^2, entonces la distribución χ^2 de la Devianza es exacta. Sin embargo, cuando se utiliza el estimador de σ^2, no podemos confiar en que la Devianza se distribuya como χ^2. *Hay que tener precaución en el uso de la Devianza como una medida general de bondad de ajuste*, puesto que su distribución aproximada a la χ^2_{n-p} depende de supuestos que con frecuencia no son sostenibles o son difíciles de verificar. Sin embargo, la Devianza es útil para probar la importancia de las variables explicativas en modelos anidados (Anderson et al., 2007; Garrido et al., 2016; Rao et al., 2010).

TABLA N° 2.3: Devianza para distribuciones de la familia exponencial

Distribución	Devianza Δ
Normal	$\frac{1}{\sigma^2}\sum_{i=1}^{n}(y_i - \hat{\mu}_i)^2$
Poisson	$2\sum_{i=1}^{n}\left[y_i \ln\left(\frac{y_i}{\hat{\mu}_i}\right) - (y_i - \hat{\mu}_i)\right]$
Binomial	$2\sum_{i=1}^{n} n_i\left[y_i \ln\left(\frac{y_i}{\hat{\mu}_i}\right) + (n_i - y_i)\ln\left(\frac{n_i - y_i}{1-\hat{\mu}_i}\right)\right]$
Gamma	$2\nu\sum_{i=1}^{n} -\left[\ln\left(\frac{y_i}{\hat{\mu}_i}\right) + \frac{y_i - \hat{\mu}_i}{\hat{\mu}_i}\right]$
Inversa Gausiana	$\frac{1}{\sigma^2}\sum_{i=1}^{n}\frac{(y_i - \hat{\mu}_i)^2}{\hat{\mu}_i^2 y_i}$
Binomial Negativa	$2\sum_{i=1}^{n}\left[y_i \ln\left(\frac{y_i}{\hat{\mu}_i}\right) - \left(y_i + \frac{1}{\kappa}\right)\ln\left(\frac{y_i + 1/\kappa}{\hat{\mu}_i + 1/\kappa}\right)\right]$

2.2.1.8 Prueba de significancia de las variables explicativas

Probar la siginificancia de las variables explicativas de modelos GLM es similar que en el Modelo Lineal General con residuos normales. Las hipótesis se escriben como $C\beta = r$, donde C es la matriz de hipótesis y r es un conjunto de valores dados.

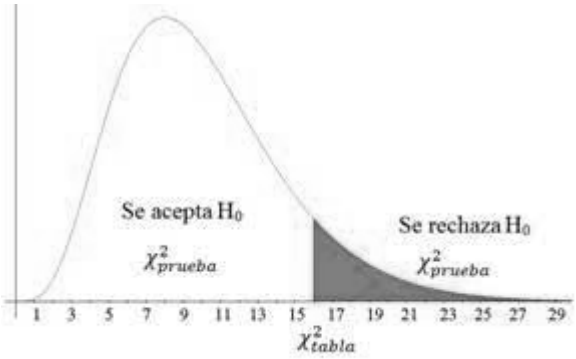
Existen diversos enfoques para probar hipótesis de la forma $C\beta = r$ y ver la significancia de las variables explicativas, pero todos ellos utiliza la $log-verosimilitud$, definido en la Sección 2.2.1.2 . Denotamos por $\hat{\beta}$ el MLE no restringido de β, y con $\tilde{\beta}$ el MLE de β cuando se maximiza sujeto a las restricciones $C\beta = r$. Además, escribimos $\hat{l}$, como el valor de l, en $\hat{\beta}$, y $\tilde{l}$, el valor de l en $\tilde{\beta}$. Obviamente $\hat{l} \geq \tilde{l}$. A continuación describimos los métodos más utilizados para ver significancia de las variables.

Prueba del ratio de verosimilitud: Aqui $\hat{l}$ es comparado con $\tilde{l}$. Un valor de $\hat{l}$ mucho más grande que $\tilde{l}$ es evidencia contra las restricciones. Ambos $\hat{\beta}$ y $\tilde{\beta}$ son requeridos. El ratio de verosimilitud es definido como $\lambda = \hat{L}/\tilde{L}$ donde $\hat{L}$ y $\tilde{L}$ son las verosimilitudes de los modelos restringidos y no restringidos, respectivamente. El estadístico de prueba de razón de verosimilitud es

$$2\ln\lambda = 2(\hat{l} - \tilde{l}) \tag{2.11}$$

Esto es siempre no-negativo, y tiene distribución χ^2_q si $C\beta = r$, donde q es el número de filas de C, es decir, el número de restricciones sobre β. Si $2\ln\lambda$ es pequeño (cercano a cero) entonces el modelo restringido es casi tan bueno como el modelo no restringido, el cual podría proporcionar que $C\beta = r$. La región de rechazo para la prueba es la cola superior de la distribución χ^2_q. La distribución χ^2_q del estadístico de razón de verosimilitud involucra el

FIGURA N° 2.2: Distribución Chi-Cuadrado



Fuente: Elaboración Propia

parámetro de dispersión ϕ, que a menudo se desconoce. En el caso de las distribuciones de Poisson y Binomial ϕ es conocido; en el caso de otras distribuciones, generalmente se debe estimar ϕ. La distribución χ^2_q sigue siendo apropiada en este caso, siempre que se utilice un estimador consistente para ϕ. El estadístico del ratio de verosimilitud puede ser expresado como la diferencia en las Devianzas de los modelos restringidos y no restrigidos, siempre

que se use la misma estimación para ϕ en ambas $log-likelihoods$. Tenga en cuenta que la aproximación de la distribución de la Devianza por una de χ^2 puede ser cuestionable, excepto en el caso de distribución de Poisson donde se conoce ϕ (Rao et al., 2010).

Prueba de *Wald*: Esta prueba mide qué tan lejos está $C\hat{\beta}$ de r. Si la diferencia $C\hat{\beta}-r$ es grande, esto proporciona evidencia contra las restricciones. Sobre la hipótesis nula H_0 : $C\beta = r$, entonces como $\hat{\beta} = N\left[\beta, \phi(X'WX)^{-1}\right]$ esto sigue que

$$C\hat{\beta}-r \approx N\left[0, \phi C(X'WX)^{-1}C'\right].$$

Esto nos lleva al famoso estadístico de Wald para probar $C\beta = r$:

$$(C\hat{\beta}-r)'\left[\phi C(X'WX)^{-1}C'\right]^{-1}(C\hat{\beta}-r) \approx \chi^2_q \tag{2.12}$$

Por lo tanto, el estadístico de Wald es el cuadrado de la distancia estadística de $C\hat{\beta}$ de r utilizando la matriz de covarianzas de $C\hat{\beta}$. En la práctica, W se reemplaza por una estimación y por lo tanto, la distribución de χ^2_q es aproximada (Rao et al., 2010).

2.2.2 Cópulas

2.2.2.1 Definicion

Actualmente, las Cópulas son una herramienta fundamental que permiten escribir la función de distribución conjunta de un vector aleatorio como la composición de una cópula y sus distribuciones marginales. En ese sentido, las cópulas se han convertido en una herramienta importante para modelas vectores aleatorios donde las marginales son dependendientes.

Definición 1. Una cópula es una función de distribución multivariada $C : [0,1]^n \to [0,1]$, cuyas distribuciones marginales tienen distribución uniforme $U_i \sim U(0,1), i = 1,...,n$. Las cópulas satisfacen las siguientes propiedades:

1. $C(u_1,...,u_d)$ es creciente en cada componente u_i.
2. $C(u_1,...,0,...,u_d) = 0$ para todo $u_i \in [0,1]$.
3. $C(1,...,1,u_i,1,...,1) = u_i$ para todo $i \in \{1,...,d\}$, $u_i \in [0,1]$.
4. Para todo $(a_1,...,a_d)$, $(b_1,...,b_d) \in [0,1]^d$ con $a_i \leq b_i$ tenemos que

$$\sum_{i_1=1}^{2}\cdots\sum_{i_d=1}^{2}(-1)^{i_1+\cdots+i_d}C(u_{1i_1},...,u_{di_d}) \geq 0$$

   donde $u_{j1} = a_j$ y $u_{j2} = b_j$ para todo $j \in \{1,...,d\}$.
5. Para cada $u_i \in [0,1]$, y para cada $j \in \{1,...,d\}$, la derivada parcial $\frac{\partial C}{\partial u_i}(u_1,...,u_d)$

existe es no negativa, es decir,

$$\frac{\partial C}{\partial u_i}(u_1, ..., u_d) \geq 0.$$

Si F_i es la distribución acumulada de una variable aleatoria X_i, entonces

$$C(F_1(x_1), \dots, F_d(x_d))$$

es una distribución d-dimensional para el vector aleatorio $X = (X_1, ..., X_d)$, con distribución marginal F_i, $i = 1, ..., d$. De igual forma, si H es una función de distribución acumulada d-dimensional con marginales $F_1, ..., F_d$, entonces existe una cópula d-dimensional C tal que, para todo $x = (x_1, ..., x_d)$ se tiene

$$H(x_1, ..., x_d) = C(F_1(x_1), ..., F_d(x_d)) \tag{2.13}$$

Si $F_1, \dots, F_d$ son continuas, entonces C es único (Sklar, 1959), en otro caso, existen varias posibles cópulas, pero todas ellas coinciden sobre la clausura de $Ran(F_1) \times \cdots \times Ran(F_d)$, donde $Ran(F)$ denota el rango de F (ver Genest y Neslehova (2007) para una demostración).

Teorema 1. Sea C una cópula. Entonces para todo $(u_1, ..., u_n), (v_1, ..., v_n) \in [0, 1]^n$ se tiene que:

$$|C(v_1, ..., v_n) - C(u_1, ..., u_n)| \leq \sum_{k=1}^{n} |v_k - u_k|.$$

La desigualdad anterior muestra que toda cópula C es uniformemente continua sobre $[0, 1]^n$.

Una demostración del Teorema 1 puede ser encontrada en (Nelsen, 2006).

El siguiente teorema nos muestra otra propiedad importante de las cópulas.

Teorema 2. Dada una cópula C, para todo $(u_1, ..., u_n) \in [0, 1]^n$, la derivada parcial $\partial C / \partial \mu_i$ existe para casi todo $u_i \in [0, 1]$ con $i \in 1, ..., n$. Además tenemos que

$$0 \leq \frac{\partial}{\partial u_i} C(u_1, ..., u_n) \leq 1.$$

La demostración de este teorema puede ser encontrada en (Nelsen, 2006).

2.2.2.2 Teorema de Sklar

En un importante trabajo (Sklar, 1959), Sklar demostró que es posible describir de forma completa la estructura de dependencia que existe en un vector aleatorio a través de las distribuciones marginales y una función link que fue denominada *cópula*. Este resultado demostró

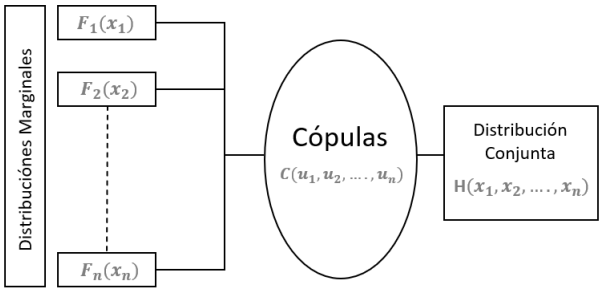
que toda distribución multivariada se puede generar a partir de una cópula y las distribuciones marginales, siendo una herramienta importante para diversas aplicaciones reales, donde se conoce las distribuciones marginales pero no se tiene información sobre la estructura de dependencia entre ellas. A continuación presentamos el teorema principal de la teoría de cópulas.

Teorema 3 (Teorema de SKLAR, Sklar (1959))**.** Sea H la distribución conjunta de las variables aleatorias $X_1, ..., X_n$ con marginales $F_1, ..., F_n$. Entonces existe una cópula C tal que

$$H(x_1, ..., x_n) = C(F_1(x_1), ..., F_n(x_n)), \quad \text{para todo} \quad x_1, ..., x_n \in \overline{\mathbb{R}}^1$$

Si $F_1, ..., F_n$ son continuas, entonces la cópula C es única. De otra forma, C está definida solamente en el $RangoF_1 \times .... \times RangoF_n$. Por otro lado, si C es una cópula y $F_1, ..., F_n$ son funciones de distribución, entonces la función H es una función de distribución conjunta con marginales $F_1, ..., F_n$.

FIGURA N° 2.3: Descripción de como se utiliza el Teorema de Sklar para calular la distribución conjunta de un vector aleatorio.



Fuente: Elaboración propia.

En la Figura 2.16, mostramos gráficamente una distribución bivariada. En las diversas aplicaciones a datos reales lo que generalmente se observa son las distribuciones marginales. Partiendo de la información muestral de las marginales se quiere construir la distribución conjunta. Las diversas familias de cópulas que existen, ayudan a encontrar una aproximación de la verdadera distribución conjunta.

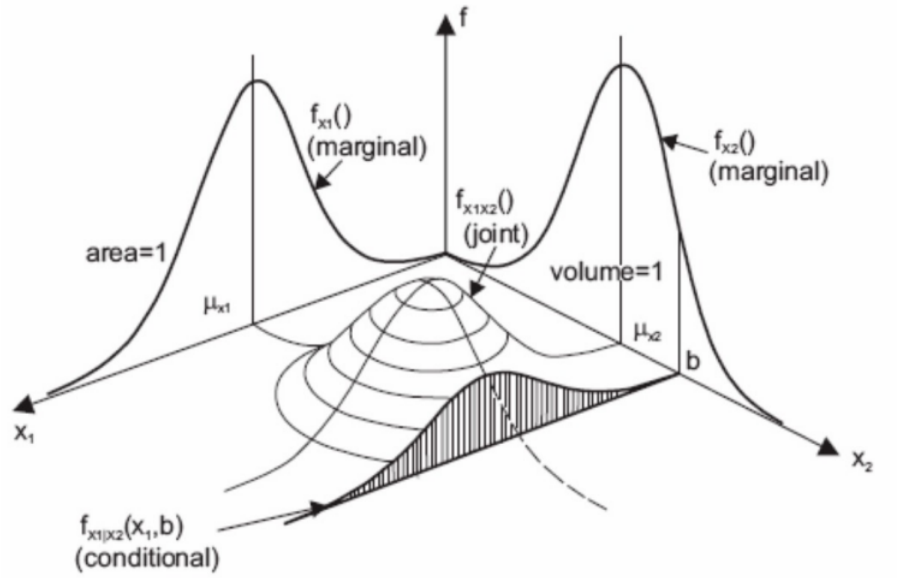
2.2.2.3 Cópulas y variables aleatorias

A continuación damos algunas definiciones necesarias para el desarrollo de este trabajo.

Definición 2. Dado un espacio de probabilidad $(\Omega, \mathfrak{F}, P)$, decimos que la función $X : (\Omega, \mathfrak{F}) \to (\mathbb{R}, \mathfrak{B})$ es una variable aleatoria, si X es una función medible, donde $\mathfrak{B}$ es el σ-álgebra de Borel.

[1] $\overline{\mathbb{R}}$ denota la recta real $[-\infty, \infty]$

FIGURA N° 2.4: Función de densidad conjunta



Fuente: Figura tomada de Uncertainty treatment in civil engineering numerical models (Matos et al., 2007)

Proposición: Decimos que las variables aleatorias $X_1, ..., X_n$ son independientes si, y sólo si, el producto de sus funciones de distribución $F_1(x_1), ..., F_n(x_n)$ es igual a su función de distribución acumulada conjunta $H(x_1, ..., x_n)$,

$$H(x_1, ..., x_n) = F_1(x_1) \times .... \times F_n(x_n), \quad \text{para todo} \quad x_1, ..., x_n \in \overline{\mathbb{R}}.$$

Teorema 4. Sean las variables aleatorias $X_1, ..., X_n$ con funciones de distribución continuas dadas por $F_1, ..., F_n$, y función de distribución conjunta H. Por Teorema de Sklar existe una única cópula C tal que

$$P(X_1 \leq x_1, ..., X_n \leq x_n) = H(x_1, ..., x_n) = C(F_1(x_1), ..., F_n(x_n)).$$

Entonces, $X_1, ..., X_n$ son independientes si, y sólo si,

$$C = \prod^{n} = u_1 \times .... \times u_n$$

Para una demostración del Teorema 4, ver por ejemplo (Nelsen, 2006).

Teorema 5. Sea $(X_1, ..., X_n)^T$ un vector aleatorio cuyas coordenadas son continuas, y sea C su respectiva cópula. Si $\alpha_1, ..., \alpha_n$ son funciones reales estrictamente crecientes con dominios dados por $RangoX_1$,..., $RangoX_n$, respectivamente. Entonces, el vector aleatorio $(\alpha_1(X_1), ..., \alpha_n(X_n))$ tiene cópula C. Es decir, C es invariante bajo transformaciones es-

trictamente crecientes de $X_1, ..., X_n$.

Para una demostración del Teorema 5, ver por ejemplo (Nelsen, 2006).
Esta propiedad será muy util en la aplicación que veremos más adelante.

2.2.2.4 Medidas de Dependencia

Las relaciones de dependencia entre variables aleatorias es uno de los temas más estudiados en probabilidad y estadística. La naturaleza de la dependencia puede tomar una variedad de formas y, a menos que se hagan algunos supuestos específicos sobre la dependencia, no se puede contemplar ningún modelo estadístico significativo. Por ese motivo, modelar la dependencia entre variables aleatorias es un problema complicado de abordar actualmente, desde el punto de vista estadístico y numérico.

Concordancia: Informalmente, un par de variables aleatorias son concordantes si los valores grandes de una variables tiende a estar asociado a valores grandes de la otra, y valores pequeños de uno con los valores pequeños del otro. Para ser más preciso, sean (x_i, y_i) y (x_j, y_j) dos observaciones de un vector aleatorio continuo (X, Y). Decimos que (x_i, y_i) y (x_j, y_j) son concordantes si $x_i < x_j$ y $y_i < y_j$, o si $x_i > x_j$ y $y_i > y_j$. Similarmente, decimos que (x_i, y_i) y (x_j, y_j) son discordantes si $x_i < x_j$ y $y_i > y_j$ o $x_i > x_j$ y $y_i < y_j$. Una formulación alternativa es dada por: (x_i, y_i) y (x_j, y_j) son concordantes si $(x_i - x_j)(y_i - y_j) > 0$ y son discordantes si $(x_i - x_j)(y_i - y_j) < 0$. *(ver Nelsen (2006) para más detalles y propiedades de esta definición)*

Teorema 6. Sea (X, Y) y $(\tilde{X}, \tilde{Y})$ dos vectores aleatorios independientes continuas con función de distribución conjunta H y $\tilde{H}$, respectivamente, y con marginales comunes F (de X y $\tilde{X}$) G (de Y y $\tilde{Y}$). Sea C y $\tilde{C}$ las cópulas de (X, Y) y $(\tilde{X}, \tilde{Y})$ respectivamente. Así tenemos que $H(x, y) = C(F(x), F(y))$ y $\tilde{H} = \tilde{C}(F(x), G(y))$. Sea Q la diferencia entre la probabilidad de concordancia y discordancia de (X, Y) y $(\tilde{X}, \tilde{Y})$, es decir,

$$Q = P\left[(X - \tilde{X})(Y - \tilde{Y}) > 0\right] - P\left[(X - \tilde{X})(Y - \tilde{Y}) < 0\right]$$

Entonces,

$$Q = Q(C, \tilde{C}) = 4 \iint_{[0,1]^2} \tilde{C}(u, v) dC(u, v) - 1$$

Para una demostración del Teorema 6, ver (Nelsen, 2006).

Definición 3 (Scarsini (1984))**.** Una medida de dependencia entre dos variables aleatorias continuas X e Y, denotada por κ, cuya cópula es C, es una medida de concordancia tal que:

1. κ está definida para todo par de variables aleatorias X, Y.
2. $-1 \leq \kappa_{X,Y} \leq 1, \quad \kappa_{X,X} = 1$ y $\kappa_{X,-X} = -1$

3. $\kappa_{X,Y} = \kappa_{Y,X}$
4. Si X e Y son independientes, entonces $\kappa_{X,Y} = 0$.
5. $\kappa_{-X,Y} = \kappa_{X,-Y} = -\kappa_{X,Y}$
6. Si C y $\tilde{C}$ son cópulas tales que $C \prec \tilde{C}$, entonces $\kappa_C \leq \kappa_{\tilde{C}}$
7. Si (X_n, Y_n) es una sucesión de vectores aleatorios continuos con cópulas C_n. Si C_n converge a C, tenemos que $\lim_{n\to\infty} \kappa_{C_n} = \kappa_C$

De la Definición 3 tenemos que si Y es una función no decreciente de X, entonces $\kappa_{X,Y} = \kappa_M = 1$ *(Comonoticidad)*, y si Y es una función monótona decreciente de X, entonces $\kappa_{X,Y} = \kappa_W = -1$ *(Contramonotonicidad)*. Si α y β son funciones estrictamente crecientes en el $Rango(X)$ y $Rango(Y)$, respectivamente, entonces $\kappa_{\alpha(X),\beta(Y)} = \kappa_{X,Y}$.

Correlación de Pearson ρ : Es una de las medidas de dependencia más utilizadas, el cual mide la relación lineal que existe entre variables aleatorias. Para dos variables aleatorias X e Y la correlación de Pearson $\rho(X,Y)$, se define por:

$$\rho(X,Y) = \frac{Cov(X,Y)}{\sqrt{Var(X)Var(Y)}}.$$

Por su fácil implementación, el coeficiente de correlación de Pearson es aceptado como una medida de dependencia de amplia aceptación. Su practicidad se debe a que sólo se necesita estimar los dos primeros momentos de los datos.

Otra ventaja que tiene la correlación de Pearson es su relación con la función de distribución normal multivariada, donde la correlación resume toda la relación de dependencia existente. Sin embargo, se debe tener cuidado cuando se utiliza la correlación de Pearson como medida de dependencia, puesto que no es adecuada para medir dependencia no lienal (*ver Embrechts (1999), McNeil et al. (2005)*). Otros inconvenientes de la correlación de Pearson son:

- Si las variables son normales y el coeficiente de correlación de Pearson es cero, entonces las variables son independientes. Para otras distribuciones esta propiedad no necesariamente es verdad.
- Únicamente se define para dos de variables aleatorias que tengan varianzas finitas. Esto es de vital importancia, cuando se analiza variables con colas pesadas, puesto que en estos casos la varianza puede no existir.
- La correlación de Pearson no es invariante bajo transformaciones estrictamente monótonas.

Una forma gráfica de entender que el coeficiente de correlación no es óptimo para medir la relación o asociación entre dos variables es mediante el cuarteto de Anscombe (Anscombe,

1973). Por ejemplo, consideremos los siguientes datos de la Tabla N° 2.4. Podemos observar que las variables X tienen la misma media al igual que las variables Y. Además, podemos observar que los coeficientes de correlación de cada grupo es el mismo.

TABLA N° 2.4: Datos del Cuarteto de Anscombe

	Grupo 1		Grupo 2		Grupo 3		Grupo 4	
	X	Y	X	Y	X	Y	X	Y
	10.00	8.04	10.00	9.14	10.00	7.46	8.00	6.58
	8.00	6.95	8.00	8.14	8.00	6.77	8.00	5.76
	13.00	7.58	13.00	8.74	13.00	12.74	8.00	7.71
	9.00	8.81	9.00	8.77	9.00	7.11	8.00	8.84
	11.00	8.33	11.00	9.26	11.00	7.81	8.00	8.47
	14.00	9.96	14.00	8.10	14.00	8.84	8.00	7.04
	6.00	7.24	6.00	6.13	6.00	6.08	8.00	5.25
	4.00	4.26	4.00	3.10	4.00	5.39	19.00	12.50
	12.00	10.84	12.00	9.13	12.00	8.15	8.00	5.56
	7.00	4.82	7.00	7.26	7.00	6.42	8.00	7.91
	5.00	5.68	5.00	4.74	5.00	5.73	8.00	6.89
Media	9.00	11.00	9.00	11.00	9.00	11.00	9.00	11.00
ρ	0.816		0.816		0.816		0.816	

En la Figura N° 2.5 mostramos la gráfica de dispersión de los 4 grupos de datos mostrados en la Tabla N° 2.4. El primer gráfico (arriba a la izquierda) muestra una típica relación lineal simple entre dos variables correlacionadas cumpliendo con la suposición de normalidad. En el segundo gráfico (arriba a la derecha) se observa que los datos no están relacionados de forma lineal, y el coeficiente de correlación de Pearson (que mide relación lineal) no sería apropiado. En la tercera gráfica (abajo a la izquierda) la distribución de los datos es lineal, pero con una recta de regresión diferente a las anteriores por causa del dato extremo observado, que influye lo suficiente como para alterar la recta de regresión y disminuir el coeficiente de correlación de 1 a 0.816. Por último, la cuarta gráfica (abajo a la derecha) es un ejemplo donde un valor atípico es suficiente para producir un coeficiente de correlación alto incluso cuando la relación entre las dos variables no es lineal.

El cuarteto muestra la importancia de visualizar gráficamente los conjunto datos antes de ajustar algún tipo de modelo.

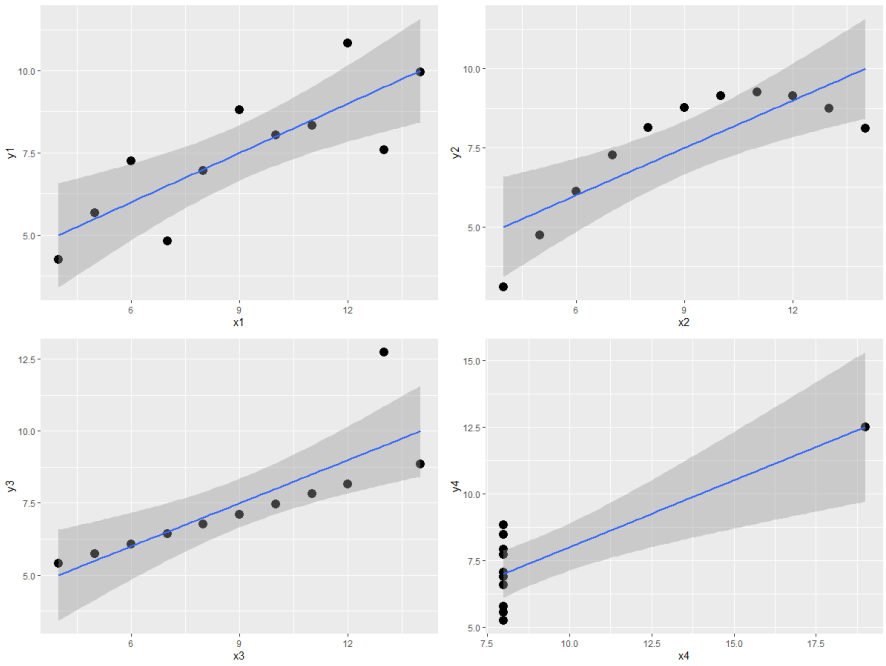
La Tau de Kedall τ: Sea $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$ una muestra aleatoria de n observaciones de un vector aleatorio continuo (X, Y). Un par de observaciones (x_i, y_i) y $(x_j.y_j)$ son concordantes o discordantes según la definición de Concordancia, dada anteriormente.

Existen $\binom{n}{2}$ diferentes pares (x_i, y_i) y $(x_j.y_j)$ de observaciones, y cada par es concordante o discordante. Sea N_c el número de pares concordantes y N_d el número de pares discordantes. Entonces la estimación del τ de Kendall para la muestra se define como:

$$\hat{\tau} = \frac{N_c - N_d}{N_c + N_d}$$

Equivalentemente el τ de Kendall se define como la probabilidad de concordancia menos

FIGURA N° 2.5: Gráficos del Cuarteto de Anscombe



la probabilidad de discordancia para un par de observaciones (x_i, y_i) y $(x_j.y_j)$ elegidas aleatoriamente de la muestra.

Siguiendo la definición anterior damos la definición del tau de Kendall para un vector aleatorio.

Definición 4. Definimos el tau de Kendall de vector aleatorio (X,Y) de la siguiente forma:

$$\tau(X,Y) = P\left[(X-\tilde{X})(Y-\tilde{Y}) > 0\right] - P\left[(X-\tilde{X})(Y-\tilde{Y}) < 0\right],$$

donde $(\tilde{X},\tilde{Y})$ es una copia independiente de (X,Y).

El siguiente teorema da una forma equivalente de calcular la tau de Kendall en función de la cópula de X e Y.

Teorema 7. Sea un vector aleatorio continuo (X,Y) con cópula C. Entonces la tau de Kendall es definido por

$$\tau_C = 4 \iint_{[0,1]^2} C(u,v)dC(u,v) - 1.$$

Observe que la tau de Kendall se puede escribir en función del valor esperado de la variable aleatoria $C(U,V)$, donde $U,V \sim U(0,1)$, es decir,

$$\tau_C = 4E[C(U,V)] - 1$$

.

2.2.2.5 Medidas gráficas para detectar dependencia bivariante

Chi-Plot: En Fisher y Switzer (2001), los autores construyen una medida gráfica como instrumento sencillo, rápido y útil para explorar extructuras de dependencia entre dos variables, dependiendo únicamente de los datos. La información que proporciona esta medida complementa a la obtenida de un diagrama de dispersión, en el cual puede resultar difícil de observar algún patrón de dependencia. Para su construcción procedemos de la siguiente manera: Sea $(X_1, Y_1), ..., (X_n, Y_n)$ una muestra aleatoria simple de H, función de distribución conjunta continua de la v.a. (X, Y). Denotamos por $I(A)$ la función indicadora del evento A.

Para cada observación x_i, y_i, seguimos los siguientes pasos:

- Calculamos H_i, F_i, G_i y S_i que son definidos por:

$$H_i = \frac{1}{n-1} \sum_{j \neq i} I(X_j \leq X_i, Y_j \leq Y_i)$$

$$F_i = \frac{1}{n-1} \sum_{j \neq i} I(X_j \leq X_i)$$

$$G_i = \frac{1}{n-1} \sum_{j \neq i} I(Y_j \leq Y_i)$$

$$S_i = sign\left\{ \left(F_i - \frac{1}{2}\right)\left(G_i - \frac{1}{2}\right)\right\}$$

- Calculamos λ_i y χ_i de la siguiente forma:

$$\lambda_i = 4 S_i max\left\{ \left(F_i - \frac{1}{2}\right)^2, \left(G_i - \frac{1}{2}\right)\right\}^2$$

$$\chi_i = \frac{H_i - F_i G_i}{\sqrt{F_i(1-F_i)G_i(1-G_i)}}$$

  El Chi-plot es un diagrama de los pares $(\lambda_i, \chi_i)_{i=1,...,n}$ donde λ_i es una medida de distancia de la observación (X_i, Y_i) al centro de los datos. Los valores de λ_i deben estar en el intervalo $[-1, 1]$.

  En caso de que los datos constituyan una muestra bivariente con marginales continuas independientes, los valores de λ_i se distribuyen uniformemente. Sin embargo, si X e Y son dependientes, los valores de λ_i se presentarán formando de clusters o agrupaciones. En particular, valores positivos de λ_i indican que X_i e Y_i son relativamente grandes (a la vez) o relativamente pequeñas (a la vez) respecto a sus medianas, mientras que λ_i negativos corresponden a X_i e Y_i situados en lados opuestos respecto a sus medianas.

Para cada muestra, χ_i se puede interpretar como el coeficiente de correlación asociado a $n-1$ pares (X_{ij}, Y_{ij}), definidos de la siguiente forma:

$$X_{ij} = \begin{cases} 1 & \text{Si} \quad X_j \leq X_i \\ 0 & \text{en otro caso} \end{cases}$$

$$Y_{ij} = \begin{cases} 1 & \text{Si} \quad Y_j \leq Y_i \\ 0 & \text{en otro caso} \end{cases}$$

para todo $j \neq i$. Así, tenemos que $-1 \leq \chi_i \leq 1$, para todo $i = 1, ..., n$. Además, la expresión $\sqrt{n\chi_i}$ es la raíz cuadrada del estadístico chi-cuadrado utilizado tradicionalmente para contrastar la independencia en la tabla de contingencia generada por lla muestra (X_i, Y_i).

- Graficar los pares de puntos $(\lambda_i, \chi_i) \in [-1,1] \times [-1,1]$

De acuerdo a Fisher y Switzer (2001), para evitar posibles desfases gráficos, se recomienda graficar únicamente los puntos que satisfacen

$$|\lambda| \leq 4\left(\frac{1}{n-1} - \frac{1}{2}\right)^2$$

Al momento de graficar la nube de puntos, se debe tener en cuenta la posibe pérdida de pares de puntos (entre dos y cuatro, en general) al dividir por cero, por lo que este método no es eficiente en muestras pequeñas.

Si queremos contrastar las hipótesis H_o: *Las variables son independientes* frente a H_1: *Las variables no son independientes*, los valores de χ_i que estén "muy lejos" de cero nos darán evidencia estadística para poder rechazar la hipótesis nula.

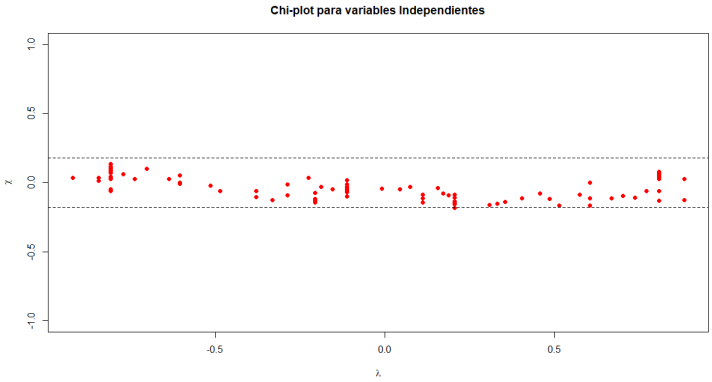
El término "muy lejos" está asociado con el nivel de significancia que se define a priori en la prueba de hipótesis. Fisher y Switzer (2001) proponen los siguientes límites de control $\pm c_p\sqrt{n}$, donde c_p es escogido aproximadamente como el $100p\,\%$[2] de los pares (λ_i, χ_i) que caen dentro de esas lineas.

Para ilustrar la técnica visual del estudio de la dependencia dada por la representación de Chi-plots, primero generamos una una muestra aleatorias simple de tamaño 100 de dos variables independientes, una extraida de una pariable $X \sim Poi(3)$ y otra $Y \sim N(0,1)$, y calculamos a partir de ellas los (λ_i, χ_i). De esta forma, se logran 100 pares de observaciones provenientes de dos variables independientes.

Ahora generamos una muestra aleatoria de dos variables que son dependientes, dando explícitamente una forma funcional a su dependencia. Para esto, escogemos $X \sim Poi(3)$ e

[2] p es equivalente al nivel de confianza. Se suele escoger $p = 0.9 \quad p = 0.95 \quad p = 0.99$

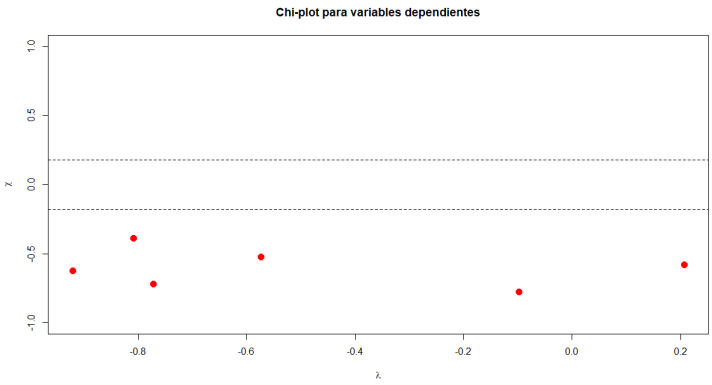
FIGURA N° 2.6: Chi-plot para variables independientes con $p = 0.95$



Fuente: Elaboración propia.

$Y = \frac{1}{1+X^2}$. Al tomar dos v.a. dependientes lo que obtendremos será una nube de puntos totalmente alejada de los márgenes de tolerancia dados por $\pm c_p\sqrt{n}$, puesto que las dos variables fueron escogidas dependientes. En la Figura 2.7 se puede observar claramente como la nube de puntos cae fuera de las bandas de tolerancia. Esto demuestraría que la muestra observada no es independiente, tal como fue tomada a priori.

FIGURA N° 2.7: Chi-plot para variables dependientes con $p = 0.95$



Fuente: Elaboración Propia

Para más detalles del método Chi-Plot, el lector puede revisar Fisher y Switzer (2001), que es el trabajo donde fue introducido este método, y de donde nos guiamos para desarrollar esta sección.

K-Plot: También conocida como Kendall-Plot. Esta es técnica gráfica que se utiliza para analizar la posible dependencia entre dos variables aleatorias fue propuesta por Genest y Boies en (Genest y Boies, 2003). Este método está inspirado en los famosos gráficos QQ-plots. A continuación pasamos a describir el método:

Primero representamos los pares $(W_{i:n}, H_{(i)})$ para todo $i \in 1, ..., n$ de modo que

$$H_{(1)} \leq H_{(2)} \leq ..... \leq H_{(n)}$$

es la muestra aleatoria ordenada, donde cada H_i es calculada como en el método Chi-plot.

Luego calculamos

$$W_{i:n} = n.\binom{n-1}{i-1}\int_0^1 w(K_0(w))^{i-1}.(1-K_0(w))^{n-i}dK_0w,$$

donde

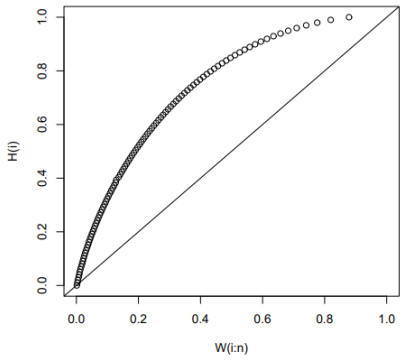
$$K_0(w) = P(U.V \leq w) = w - w.log(w) \quad w \in [0,1]$$

A medida que los pares $(W_{i:n}, H_{(i)})$ se vayan desviando con respecto a la diagonal, se puede ir asumiendo que existe una dependencia funcional entre las dos variables aleatorias. Cuando el desvio es mayor, eso implica que la dependencia también lo es.

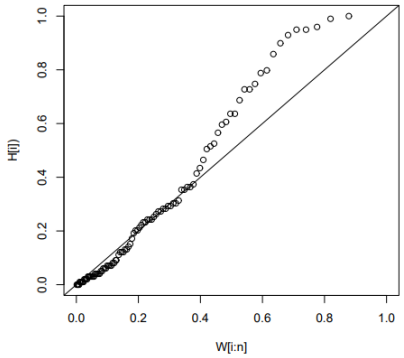
Para ilustrar este método consideremos una muestra de tamaño 100, de las variables X,Y, donde $Y = X^3$ y $X \sim N(0,1)$. En la Figura 2.8a observamos que la nube de puntos está por encima de la diagonal. Sin embargo, cuando las variables resultan ser independientes la gráfica se concentra en la diagonal. Por ejemplo, en la Figura 2.8b ilustramos un K-plot para dos variables independientes $X \sim N(0,1)$ e $Y \sim Po(3)$.

FIGURA N° 2.8: Gráficos K-PLOT

(a) Variables dependientes



(b) Variables independientes



2.2.3 Familias de Cópulas

2.2.3.1 Cópulas Elípticas

Su principal característica es que representan relaciones de dependencia simétricas sin importar si el analisis se realiza sobre la cola izquierda o derecha de las distribuciones implicadas. Las principales cópulas elípticas son:

Cópula Normal o Gaussiana: Sea $\mathbf{R}$ es la matriz de correlación de dimensión $d \times d$. Entonces la función de distribución acumulada de la cópula normal multivariante es:

$$C(u;\mathbf{R}) = \Phi_d(\Phi^{-1}(u_1), ..., \Phi^{-1}(u_d); \mathbf{R}), \quad u \in [0,1]^d$$

y la densidad de la cópula es

$$c(u;\mathbf{R}) = \frac{\phi_d(\Phi^{-1}(u_1), ..., \Phi^{-1}(u_d))}{\prod_{j=1}^{d} \phi(\Phi^{-1}(u_j))}, \quad u \in (0,1)^d,$$

cuando $\mathbf{R}$ es definida positiva.

Cuando $d = 2$, el parámetro de la función de la distribución acumulada, con $\rho \in [-1,1]$, nos conlleva a:

$$C(u,v,\rho) = \Phi_2(\Phi^{-1}(u), \Phi^{-1}(v); \rho), \quad 0 < u, v < 1.$$

La distribución condicional es:

$$C_{2|1}(v|u;\rho) = \Phi\left(\frac{\Phi^{-1}(v) - \rho\Phi^{-1}}{\sqrt{1-\rho}}\right).$$

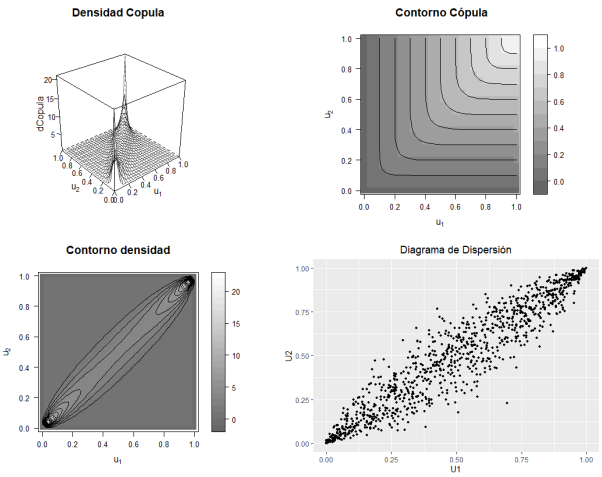
y la densidad de la cópula, como es descrito en (Joe, 2014), con $-1 < \rho < 1$ es:

$$c(u,v;\rho) = \frac{\phi_2(\Phi^{-1}(u), \Phi^{-1}(v); \rho)}{\phi(\Phi^{-1}(u))\phi(\Phi^{-1}(v))}$$

$$= (1-\rho^2)^{-1/2} \times \exp\left(-\frac{1}{2}(x^2+y^2-2\rho xy)(1-\rho^2)\right) \times \exp\left(\frac{1}{2}(x^2+y^2)\right)$$

Con $x = \Phi^{-1}(u), y = \Phi^{-1}(v)$ y $0 < u, v < 1$.

FIGURA N° 2.9: Cópula Gaussiana $(\rho = 0.8)$



Elaboración propia usando el software R.

Cópula de Student: Es la cópula asociada a la distribución t multivariante. Para el caso bivariado tenemos que

$$C(u,v;\rho,\nu) = t_{\rho,\nu}\left(t^{-1}(u), t^{-1}(v)\right)$$

Siendo t_ν la distribución de t-Student con ν grados de libertad y $t_{\rho,\nu}$ la distribución t- Student bivariada con ν grados de libertad y con matriz de correlación dada por ρ. La función de densidad de la cópula t bivariada se escribe como:

$$C(u,v;\rho,\nu) = \frac{\nu}{2\sqrt{1-\rho^2}} \cdot \frac{\Gamma\left(\frac{\nu}{2}\right)^2}{\Gamma\left(\frac{\nu+1}{2}\right)^2} \cdot \frac{\left(1+\frac{x^2+y^2-2\rho xy}{\nu(1-\rho^2)}\right)-\frac{\nu+2}{2}}{\left[\left(1+\frac{x^2}{\nu}\right)\left(1+\frac{y^2}{\nu}\right)\right]-\frac{\nu+1}{2}}$$

Con $x = t_\nu^{-1}(u)$ y $y = t_\nu^{-1}(v)$. Para obtener lo anterior utilizamos la expresión de la distribución t bivariada:

$$t_{\rho,\nu}(x,y) = \int_{-\infty}^{x}\int_{-\infty}^{y} \frac{1}{2\pi\sqrt{1-\rho^2}}\left(1+\frac{x^2+y^2-2\rho xy}{\nu(1-\rho^2)}\right)-\frac{\nu+2}{2}dxdy$$

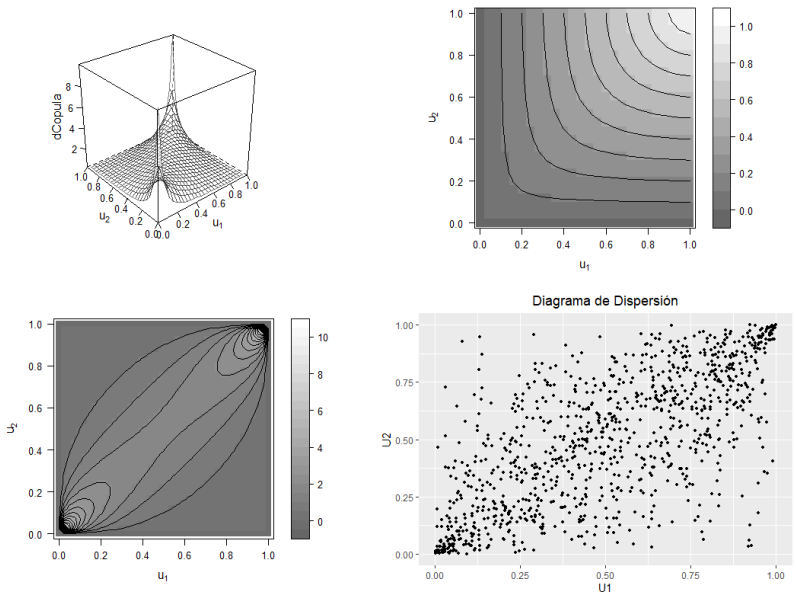
Luego utilizamos la siguiente propiedad de las cópulas

$$f(x,y) = C(F_1(x), F_2(y)).f_1(x).f_2(y)$$

con ello determinamos la expresión de la cópula t.

A menudo las cópulas elípticas son llamadas cópulas implícitas, puesto que no tienen una forma análítica, y solo se expresa en términos de las distribuciones bivariadas asociadas a los mismos.

FIGURA N° 2.10: Cópula Student($\tau = 0.5, \nu = 4$)



Elaboración propia usando el software R.

2.2.3.2 Cópulas Arquimedianas

Existe una gran diversidad de cópulas que pertencen a la familia arquimediana y gracias a esta variedad permiten modelar muchos tipos de estructuras de dependencia. Otra ventaja de esta familia de cópulas es que dependen solamente de un parámetro y son más fáciles de implementar numéricamente.

Definición 5. Sea $\varphi : [0,1] \to [0,\infty]$ una función continua, estrictamente decreciente y convexa tal que $\varphi(0) = \infty$ y $\varphi(1) = 0$. Una cópula arquimediana es una cópula que se puede representar de la siguiente forma:

$$C(u) = \varphi\left(\varphi^{-1}(u_1) + ... + \varphi^{-1}(u_d)\right), \quad u \in [0,1]^d,$$

La función φ recibe el nombre de **generador de la cópula**.

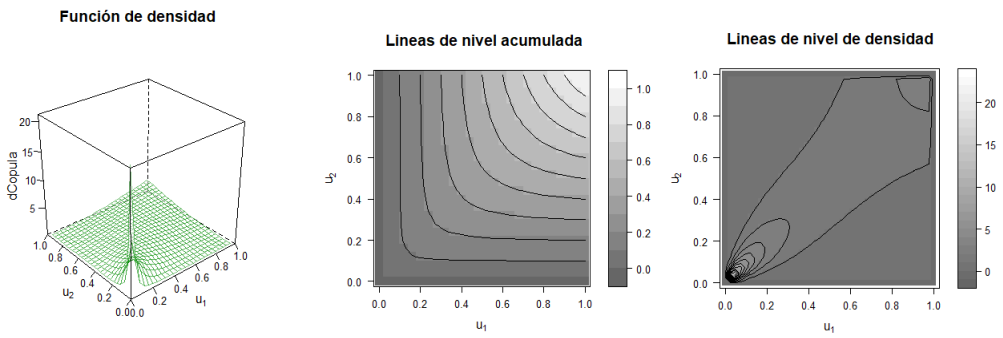
Cópula de Clayton: Sea $\varphi_\theta(t) = \frac{t^{-\theta}-1}{\theta}$ para $\theta > 0$, el generador que corresponde a la familia de Clayton. La inversa $\varphi_\theta^{-1}(t) = (1+\theta t)^{-\frac{1}{\theta}}$ es estrictamente monótona sobre $[0,\infty]$, puesto que para todo $t \in [0,\infty)$, tenemos que

$$(-1)^k \frac{d^k}{dt^k}\varphi^{-1}(t) = (-1)^{2k}\prod_{j=0}^{k}(1+\theta j)(1+\theta t)^{-\frac{1+k\theta}{\theta}} \geq 0, \quad k = 0,1,...$$

La cópula Clayton multivariada, con $n \geq 2$, está dada por

$$C_\theta(u_1,\cdots,u_n) := \left(u_1^{-\theta} + \cdots + u_n^{-\theta} - n + 1\right)^{-\frac{1}{\theta}}, \quad \theta > 0.$$

FIGURA N° 2.11: Cópula Clayton($\theta = 0.5$)



Elaboración propia usando el software R.

Cópula de Frank: Sea $\varphi_\theta(t) = -ln\dfrac{e^{-\theta t}-1}{e^{-\theta}-1}$, el generador de la familia de cópulas de Frank. Aunque los generadores de esta familia son estrictos, debemos restringir los valores de θ a

$(0,\infty)$. La inversa del generador es igual a

$$\varphi_\theta^{-1}(t) = -\frac{1}{\theta} ln[1-(1-e^{-\theta})e^{-t}].$$

Si probamos que $f(x) = -\dfrac{ln(1-x)}{\theta}$ es absolutamente monótona en $(0,1)$ y que $g(t) = (1-e^{-\theta})e^{-t}$ es completamente monótona en $[0,\infty)$, $\varphi_\theta^{-1} = f \circ g$ será completamente monótona en $[0,\infty)$. Entonces,

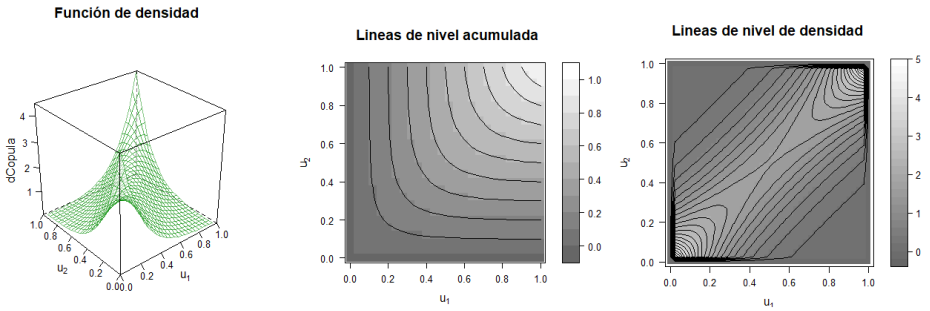
$$\frac{d^k}{dx^k} f(x) = \begin{cases} \frac{-ln(1-x)}{\theta}, & k=0 \\ \frac{(k-1)!}{\theta}(1-x)^{-k}, & k>0 \end{cases}$$

$$(-1)^k \frac{d^k}{dt^k} g(t) = (-1)^{2k}(1-e^{-\theta})e^{-t} \geq 0.$$

Entonces, para $\theta > 0$ y $n \geq 2$, la cópula de Frank multivariada está dada por

$$C_\theta(u_1,.....,u_n) := -\frac{1}{\theta} ln\left(1 + \frac{(e^{-\theta u_1}-1).....(e^{-\theta u_n}-1)}{(e^{-\theta}-1)^{n-1}}\right)$$

FIGURA N° 2.12: Cópula de Frank$(\theta = 0.5)$



Elaboración propia usando el software R.

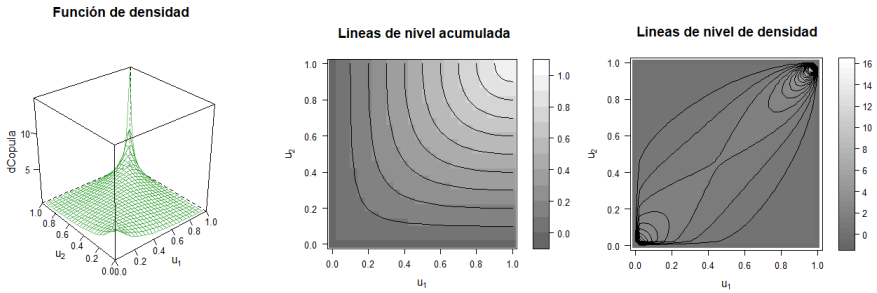
Cópula de Gumbel: Sea $\varphi_\theta = (-lnt)^\theta, \theta \geq 1$, el generador de la familia de Gumbel. Entonces $\varphi_\theta^{-1}(t) = exp\left(-t^{\frac{1}{\theta}}\right)$ es completamente monótona. En efecto, si g denota la derivada,

$$(-1)^k \frac{d^k}{dt^k} g(t) = (-1)^k \frac{\prod_{j=0}^{k}(1-\theta j)}{\theta^{k+1}} t^{\frac{1-k\theta}{\theta}} \geq 0.$$

Entonces, para $\theta \geq 1$ y $n \geq 2$, la cópula de Gumbel multivariada está dada por

$$C_\theta(u_1,.....,u_n) := exp\left(-\left[(-lnu_1)^\theta + ... + (-lnu_n)^\theta\right]^{\frac{1}{\theta}}\right)$$

FIGURA N° 2.13: Cópula de Gumbel($\theta = 0.5$)



Elaboración propia usando el software R.

2.2.4 Modelos GLM basados en Cópulas

2.2.4.1 Cópulas bivariadas para datos continuos-discretos

Una cópula bivariada $C : [0,1] \times [0,1] \to [0,1]$ es una función de distribución acumulada bivariante en $[0,1] \times [0,1]$ con marginales distribuidas uniformemente. La importancia de la cópula se sustenta mediante el Teorema de Sklar (Sklar, 1959). En el caso bivariado, establece que para cada función de distribución conjunta $F_{X,Y}$ de una variable aleatoria bivariada (X,Y) con funciones de distribución marginal F_X y F_Y, existe una cópula bivariada C tal que

$$F_{X,Y}(x,y) = C(F_X(x), F_Y(y))$$

Si X e Y son variables aleatorias continuas, la cópula C es única.

Si C es una cópula, por el Teorema de Sklar $C(F_X(x), F_Y(y))$ define una función de distribución bivariada con marginales dadas por F_X y F_Y. Esto nos permite modelar las distribuciones marginales y la dependencia conjunta por separado, puesto que podemos definir la cópula C independientemente de las distribuciones marginales.

Sabemos que las cópulas son invariantes bajo transformaciones monótonas de las distribuciones marginales. Por lo tanto, en lugar del coeficiente de correlación de Pearson, que mide asociaciones lineales, se utilizan medidas de asociación monótonas. Una opción muy común es la τ de Kendall,

$$\tau := 4 \int_{[0,1]^2} C(u,v) dC(u,v) - 1 \quad \in [-1,1]$$

En (Czado et al., 2012) se estudian modelos basados en cópula, donde se tiene una v.a. continua y una v.a. discreta. Sea X una variable aleatoria continua, e Y una variable aleatoria discreta. Su distribución conjunta es definida por una cópula parámetrica $C(.,.|\theta)$ que

depende del parámetro θ, es decir, la distribución conjunta está dada por

$$F_{X,Y|\theta}(x,y) = C(F_X(x), F_Y(y)|\theta)$$

Czado et al. (2012) usan principalmente cuatro familias de cópulas bivariadas parámetricas: Gauss, Clayton, Gumbel y Frank. Cada familia depende de un sólo parámetro θ. Este parámetro puede ser expresado en términos de la tau de Kendall como lo muestra la Tabla 2.5:

TABLA N° 2.5: Relación entre el parámetro θ y τ de Kendall

Familia	Cópula $C(u,v\|\theta)$	Rango de θ	Relación con τ
Gauss	$\Phi_2(\Phi^{-1}(u), \Phi^{-1}(v)\|\theta)$	$]-1,1[$	$\tau = \frac{2}{\pi} arcsin(\theta) \in \mathbb{R}$
Clayton	$(u^{-\theta} + v^{-\theta} - 1)^{-1/\theta}$	$]0,\infty[$	$\tau = \frac{\theta}{\theta+2} \in]0,\infty[$
Gumbel	$\exp\left(-\left((-\log u)^\theta + (-\log v)^\theta\right)^{1/\theta}\right)$	$[1,\infty[$	$\tau = \frac{\theta-1}{\theta} \in [0,\infty[$
Frank	$-\frac{1}{\theta}\log\left(1 + \frac{(e^{-\theta u}-1)(e^{-\theta v}-1)}{(e^{-\theta}-1)}\right)$	$\mathbb{R} - \{0\}$	$\tau = 1 - \frac{4}{\theta}[1 - D_1(\theta)] \in \mathbb{R} - \{0\}$

donde $D_k(x)$ es definido por

$$D_k(x) = \frac{k}{x^k} \int_0^x \frac{t^k}{e^t - 1} dt$$

denota la función Debye el cual está definida para $k \in \mathbb{N}$

En la Tabla 2.5 podemos observar que la cópula Clayton es sólo definida para valores positivos de la τ de Kendall, y que la cópula de Gumbel es sólo definida para valores no negativos de la τ de Kendall. Sin embargo, a través de una rotación, es posible extender estas familias de cópulas a valores negativos de τ.

Para el muestreo, la estimación y la predicción, necesitamos la función de densidad conjunta/masa de probabilidad de X e Y que se define por

$$f_{X,Y}(x,y) := \frac{\partial}{\partial x} P(X \leq x, Y = y) \tag{2.14}$$

Ahora derivamos las fórmulas para la densidad conjunta de X e Y en términos de la cópula $C(.,. \mid \theta)$, denotaremos lo siguiente:

$$C_1(u,v \mid \theta) := \frac{\partial}{\partial u} C(u,v \mid \theta) \tag{2.15}$$

para $u, v \in]0,1[$ la derivada parcial de la cópula con respecto a la primera variable. Debemos notar que esto es la densidad condicional de la variable aleatoria $V := F_Y(Y)$ dado

$U := F_X(X)$. En la Tabla 2.6 mostramos las derivadas parciales para las cópulas Clayton, Gumbel, Frank y Gauss. Para mayor detalle de las derivadas parciales de las funciones cópulas revisar (Joe, 2014; Schepsmeier y Stöber, 2014).

Teorema 8 (Función de densidad)**.** La función de densidad conjunta $f_{X,Y}$ de una variable aleatoria continua X y una variable aleatoria discreta Y puede escribirse en función de las marginales y la cópula C como,

$$f_{X,Y}(x,y|\theta) = f_X(x)\{C_1(F_X(x),F_Y(y)\mid\theta) - C_1(F_X(x),F_Y(y-1)\mid\theta)\}$$

(Para una demostración del Teorema 8, ver (Krämer et al., 2013).)

TABLA N° 2.6: Primera derivada parcial

Familia	$C_1(u,v\|\theta)$
Gauss	$\Phi\left(\frac{\Phi^{-1}(v)-\theta\Phi^{-1}(u)}{\sqrt{1-\theta^2}}\right)$
Clayton	$(u^{-\theta}+v^{-\theta}-1)^{-1/\theta-1}u^{-\theta-1}$
Gumbel	$u^{-1}\exp\left(-\left((-\log u)^{\theta}+(-\log v)^{\theta}\right)^{1/\theta}\right)$
Frank	$\frac{e^{\theta}(e^{\theta v}-1)}{e^{\theta(u+1)}+e^{\theta(v+1)}-e^{\theta}-e^{\theta(u+v)}}$

2.2.4.2 Distribuciones Marginales

Como una aplicación a la industria de seguros en Krämer et al. (2013) y Kholifah et al. (2019) los autores consideran el costo del siniestro X como variable aleatoria continua Gamma, y el número de siniestros Y como variable discreta. Para el desarrollo de este trabajo, utilizaremos las mismas distribuciones para la siniestralidad y para la frecuencia, propuestas en dichos trabajos. Es por ello que para el costo del siniestro utilizaremos el siguiente modelo:

$$f_X(x|\mu,\delta) = \frac{1}{x\Gamma\left(\frac{1}{\delta}\right)}\left(\frac{x}{\mu\delta}\right)^{\frac{1}{\delta}}\exp\left(-\frac{x}{\mu\delta}\right) \quad para \quad x>0$$

con media $\mu > 0$ y parámetro de dispersión dado por $\delta > 0$.

Para el número de siniestros Y, esta será modelada como una distribución Poisson con pa-

rámetro $\lambda > 0$,

$$f_Y(y|\lambda) = \frac{\lambda^y e^{-\lambda}}{y!} \quad para \quad y = 0, 1, 2, ...$$

TABLA N° 2.7: Parámetros del modelo de distribución conjunta

	Severidad	Frecuencia	Familia de cópulas
Distribución	Gamma	Poisson	Gauss, Clayton, Gumbel, Frank
Parámetros	$\mu > 0, \delta > 0$	$\lambda > 0$	$\theta \in \Theta$
Esperanza	$E(X) = \mu$	$E(Y) = \lambda$	–
Varianza	$Var(X) = \mu^2\delta$	$Var(Y) = \lambda$	–

2.2.4.3 Modelo de cópula conjunta para la frecuencia y severidad

Combinando las distribuciones marginales y el enfoque de cópula, obtenemos el siguiente modelo general.

Definición 6 (Modelo conjunto de la frecuencia y severidad)**.** El modelo Gamma-Poisson basado en cópula mixta para la frecuencia Y y la severidad X se define mediante la función de densidad conjunta

$$f_{X,Y}(x, y|\mu, \delta, \lambda, \theta) = f_X(x \mid \mu, \delta)\{C_1(F_X(x \mid \mu, \delta), F_Y(y \mid \lambda)|\theta) - C_1(F_X(x \mid \mu, \delta), F_Y(y-1 \mid \lambda) \mid \theta)\}$$

para $x > 0$ y $y = 0, 1, 2, ...$

El modelo depende de cuatro parámetros: los parámetros μ, δ (Gamma) y λ (Poisson) para las distribuciones marginales, y el parámetro de la cópula θ. La Tabla 2.7 muestra los parámetros y sus relaciones con la distribución conjunta.

Teorema 9 (Distribución condicional)**.** La distribución condicional de $Y|X = x$, es decir el número de siniestros dado un costo medio bajo el modelo basado en cópula de la Definicion 6, está dada por

$$P(Y = y|X = x, \mu, \delta, \lambda, \theta) = C_1(F_X(x|\mu, \delta), F_Y(y|\lambda)|\theta) - C_1(F_X(x|\mu, \delta), F_Y(y-1|\lambda)|\theta)$$

Para una demostración del Teorema 9, véase Krämer et al. (2013).

2.2.4.4 Estimación de la pérdida póliza: Prima de riesgo conjunta

Definición 7 (Pérdida póliza)**.** Para una póliza con costo medio X y número de siniestros Y, la pérdida póliza es definida como el producto de las cantidades mencionadas

$$L := X.Y$$

La pérdida póliza es una variable aleatoria continua y positiva, y esta depende de los parámetros mostrados en la Tabla 2.7.

Basado en la definición previa, Krämer et al. (2013) demuestran el siguiente teorema, que describe la distribución de la pérdida póliza en función de la cópula y las distribuciones marginales.

Teorema 10 (función de densidad de la pérdida póliza)**.** La distribución de la pérdida póliza L está dada por la siguiente función de densidad

$$f_L(l \mid \mu,\delta,\lambda,\theta) = \sum_{y=1}^{\infty}\left[C_1\left(F_X\left(\frac{l}{y} \mid \mu,\delta\right), F_Y(y|\lambda) \mid \theta\right) - C_1\left(F_X\left(\frac{l}{y} \mid \mu,\delta\right), F_Y(y-1 \mid \lambda)|\theta\right)\right].\frac{1}{y}f_X\left(\frac{l}{y} \mid \mu,\delta\right)$$

para $l > 0$.

Del teorema anterior podemos deducir la prima pura de riesgo (PPR) conjunta como el valor esperado de función de distribución de la pérdida póliza conjunta:

$$PPR = E(L) = \int_0^{\infty} l \times f_L(l|\mu,\delta,\lambda,\theta)dl$$

Esta expresión es la que utilizaremos en el presente trabajo de investigación para el cálculo de la prima pura de riesgo conjunta y que recoge el efecto de dependencia entre la frecuencia y la severidad.

2.2.4.5 Modelo de regresión para la frecuencia y severidad con cópulas

La metodología descrita en las secciones anteriores muestra como modelar la frecuencia y severidad de manera conjunta sin la presencia de covariables.

En esta sección describimos como incluir covariables para estimar los parámetros de la severidad y la frecuencia, en presencia de dependencia. Para incluir covariables en nuestro análisis utilizamos el enfoque desarrollado en (Czado et al., 2012). Para ello extendemos el modelo conjunto visto en la Definición 6, permitiendo que las distribuciones marginales del costo X y el número de siniestros Y dependan de un conjunto de covariables. **Más precisamente, aplicamos los modelos lineales generalizados para los problemas de regresión marginal y los combinamos con familias de cópulas bivariadas.**

Formulación del modelo: Sean $X_i \in \mathbb{R}_+$, $i = 1,2,3,...,n$ variables aleatorias independientes continuas e identicamente distribuidas con distribución Gamma, y sean $Y_i \in \mathbb{N}_{\geq 0}$, $i = 1,2,3,...,n$ variables aleatorias independientes discretas identicamente distribuidas con distribución de Poisson. Modelamos X_i en términos de un vector de covariables $r_i \in \mathbb{R}^p$ e Y_i en términos de un vector de covariables $s_i \in \mathbb{R}^q$. Los modelos de regresión marginales son especificados de la siguiente manera

$$X_i \sim Gamma(\mu_i, \delta) \quad \text{donde} \quad \ln(\mu_i) = r_i^\top \alpha$$
$$Y_i \sim Poisson(\lambda) \quad \text{donde} \quad \ln(\lambda_i) = ln(e_i) + s_i^\top \beta$$

el componente e_i denota a la exposición medida en años póliza.

Estimación de parámetro:

El objetivo es estimar el siguiente vector parámetro desconocido:

$$v := (\alpha^\top, \beta^\top, \theta, \delta)^\top \in \mathbb{R}^{p+q+2}$$

basados en n pares de observación (x_i, y_i). Utilizando el método de máxima verosimilitud visto en la subsección 2.2.1.5, se define la función $log-likelihood$ de parámetros como:

$$l(v \mid x, y) = \sum_{i=1}^{n} \ln\left(f_{X,Y}(x_i, y_i \mid v)\right)$$

siendo

$$x = (x_1, ..., x_n)^\top \in \mathbb{R}^n \quad \text{y} \quad y = (y_1, ..., y_n)^\top \in \mathbb{R}^n$$

La estimación de máxima verosimilitud está dada por

$$\hat{v} = \text{argmax}\, l(v \mid x, y)$$

En general, no existe una solución de forma cerrada para el problema de optimización. Por lo tanto, tenemos que maximizar el $log-likelihood$ recurriendo a métodos numéricos. En este trabajo usaremos el algoritmo de optimización BFGS (un método cuasi Newton) para maximizar el $log-likelihood$, que viene ya implementado en el software R. Como el parámetro de la cópula $\theta \in \Theta$ es en general restricta, como visto en la Tabla 2.5, transformamos θ via una función $g : \Theta \to \mathbb{R}$ tal que $g(\theta)$ sea no restricta. En la Tabla 2.8 mostramos las transformaciones que usaremos para las cópulas en este trabajo.

TABLA N° 2.8: Trasformación no restricta de θ

Familia	Rango de θ	$g(\theta)$
Gauss	$]-1,1[$	$\frac{1}{2} log\left(\frac{1+\theta}{1-\theta}\right)$
Clayton	$]0,\infty[$	$\log(\theta)$
Gumbel	$[1,\infty[$	$log(\theta - 0.9999)$
Frank	$\mathbb{R} - \{0\}$	$max(\theta, 0.0001.signo(\theta))$

Se procede a optimizar el $log-likelihood$ con respecto al vector de parámetros

$$(\alpha^\top, \beta^\top, g(\theta), \delta)^\top$$

Otra alternativa para estimar el vector de parámetros del modelo de regresión basado en cópulas, es aplicando el principio IFM (IFM por sus siglas en ingles, inference function for margins). El IFM es un procedimiento de estimación en dos pasos, propuesto por Joe (2014), un primer paso para estimar los parámetros de las marginales y el segundo paso para estimar el parámetro asociado a la cópula.

- Estimamos los modelos de regresión marginal de la frecuencia y de la severidad, mediante estimación máxima verosimil. Entonces calculamos por cada observación:

$$\hat{\mu} = \exp(R\hat{\alpha}) \in \mathbb{R}^n$$
$$\hat{\lambda} = \exp(S\hat{\alpha}) \odot e \in \mathbb{R}^n$$

  y una estimación de $\hat{\delta} \in \mathbb{R}$ como el parámetro de dispersión. En el modelo de la frecuencia , e representa la variable *offset* que es la exposción medida en años póliza, y $\odot$ denota una multiplicación por elementos de dos vectores. Los parámetros estimados son usados para transformar las observaciones x e y en valores que serán evaluados en la cópula posteriormente,

$$\begin{aligned} u_i &:= F_X\left(x_i|\hat{\mu_i}, \hat{\delta}\right) \in [0,1] \\ v_i &:= F_X\left(y_i|\hat{\lambda_i}\right) \in [0,1] \\ w_i &:= F_Y\left(y_i - 1|\hat{\lambda_i}\right) \in [0,1] \end{aligned}$$

  donde F_X y F_Y son la funciones de distribución Gamma y Poisson, respectivamente.

- Luego optimizamos el parámetro cópula θ maximizando el $log-likelihood$

$$\tilde{l}(\theta|u,v) := \sum_{i=1}^{n} \ln(C_1(u_i, v_i|\theta) - C_1(u_i, w_i|\theta))$$

La función $\tilde{l}$ puede ser maximizado numéricamente. En general, el tiempo de ejecución del método IFM es mucho menor comparado con el método MLE. Además, en Joe (2014), el autor demuestra que los estimadores que da método IFM son consistentes y asintóticamente normal.

2.2.4.6 Distribución asintótica de los parámetros de regresión

Para la construcción de los intervalos de confianza, se realiza mediante la Matriz de Información de Fisher que se define como

$$\mathcal{J}(v) := E\left[\frac{\partial l(v|x,y)}{\partial v} \cdot \left(\frac{\partial l(v|x,y)}{\partial v}\right)^{\top}\right] \in \mathbb{R}^{(p+q+2)\times(p+q+2)}$$

Bajo condiciones de regularidad se demuestra que

$$\sqrt{n}(v-\widehat{v}) \xrightarrow{D} \mathcal{N}_{p+q+2}(0, \mathcal{I}^{-1}(v))$$

donde, $\mathcal{N}_k$ denota una distribución normal multivariante de dimension k. Para la estimación de la información de Fisher, utilizamos el hecho de que

$$\mathcal{I}(v) = -E\left[\frac{\partial^2 l(v|x,y)}{\partial^2 v}\right]$$

y usamos la Matriz de Información de Fisher

$$\hat{\mathcal{I}}(v) := -\frac{\partial^2 l(v|x,y)}{\partial^2 v}$$

Lo anterior es la matriz Hessiana de la función $log-likelihood$. En nuestro caso, es factible calcular explícitamente las segundas derivadas parciales. Además, el algoritmo de optimización BFGS devuelve una aproximación de la matriz Hessiana que se obtiene mediante derivadas numéricas (Joe, 2014; Rao et al., 2010).

2.2.5 Construcción de modelos multivariados altamente dependientes - 2 Alternativas

El procedimiento dado por el método IFM es para un vector bivariado. Cuando se tiene dimensiones mayores el procedimiento es mucho más complejo. En esta sección presentamos dos alternativas para construir modelos multivariados con dependencia.

2.2.5.1 Alternativa 1: Construcción de Distribuciones Conjuntas Multivariantes

Considere el vector $X = (X_1, ..., X_n)$ de variables aleatorias con función de densidad conjunta $f(x_1, ..., x_n)$. Esta densidad puede ser factorizada como

$$f(x_1, ..., x_n) = f(x_n).f(x_{n-1}|x_n).f(x_{n-2}|x_{n-1}, x_n)...f(x_1|x_2, x_3, ..., x_n)$$

esta descomposición es única.

Si la distribución marginal de cada variable es $F_1(x_1), ..., F_n(x_n)$, y de acuerdo al Teorema de Sklar, la función de distribución conjunta puede ser escrito como:

$$F(x_1, ..., x_n) = C(F_1(x_1), ..., F_n(x_n))$$

para alguna cópula C de dimensión n. A su vez, la cópula C se puede expresar de la siguiente manera

$$C(u_1, ..., u_n) = F(F_1^{-1}(u_1), ..., F_n^{-1}(u_n))$$

donde $F_i^{-1}(u_i)$ es la función de distribución inversa de las marginales.

En Aas et al. (2009), los autores demuestran que la función de densidad conjunta f se puede escribir de la siguiente manera:

$$f(x_1,...,x_n) = c_{1...n}(F_1(x_1),...,F_n(x_n)) \times f_1(x_1) \times f_2(x_2) \times ... \times f_n(x_n)$$

donde $c_{1...n}(.)$ es la densidad de la función cópula n-variada.

Para una mejor comprensión de cómo podemos obtener la función de densidad conjunta multivariada, presentamos el caso para la función de densidad de 3 dimensiones. Esta tiene 6 maneras de poder descomponerse usando probabilidades condicionales:

$$\begin{aligned} f(x_1,x_2,x_3) = &\ f_{1|2,3}(x_1|x_2,x_3)f_{2|3}(x_2|x_3)f_3(x_3) \\ = &\ f_{1|2,3}(x_1|x_2,x_3)f_{3|2}(x_3|x_2)f_2(x_2) \\ = &\ f_{2|1,3}(x_2|x_1,x_3)f_{1|3}(x_1|x_3)f_1(x_1) \\ = &\ f_{2|1,3}(x_2|x_1,x_3)f_{3|1}(x_3|x_1)f_3(x_3) \\ = &\ f_{3|1,2}(x_3|x_1,x_2)f_{1|2}(x_1|x_2)f_1(x_1) \\ = &\ f_{3|1,2}(x_3|x_1,x_2)f_{2|1}(x_2|x_1)f_2(x_2) \end{aligned}$$

Si sólo nos enfocamos en una sola combinación, por ejemplo la primera

$$f_{1|2,3}(x_1|x_2,x_3)f_{2|3}(x_2|x_3)f_3(x_3)$$

y la escribimos en términos de cópulas bivariadas y densidades marginales:

$$f_{2|3}(x_2|x_3) = c_{2,3}(F_2(x_2),F_3(x_3))f_2(x_2)$$

$$f_{1|2,3}(x_1|x_2,x_3) = c_{1,2|3}(F_{1|3}(x_1|x_3),F_{2|3}(x_2|x_3))f_{1|3}(x_1|x_3)$$

el término $f_{1|3}(x_1|x_3)$ se puede descomponer aún más como

$$f_{1|3}(x_1|x_3) = c_{1,3}(F_1(x_1),F_3(x_3))f_1(x_1)$$

tomando en cuenta los resultados parciales obtenemos el siguiente resultado

$$\begin{aligned} f(x_1,x_2,x_3) &= f_1(x_1)f_2(x_2)f_3(x_3) \\ &\times c_{2,3}(F_2(x_2),F_3(x_3))c_{1,3}(F_1(x_1),F_3(x_3)) \\ &\times c_{1,2|3}(F_{1|3}(x_1|x_3),F_{2|3}(x_2|x_3)) \end{aligned}$$

Resulta que esta descomposición se puede visualizar muy bien en una estructura gráfica, como lo sugiere Aas et al. (2009).

Vine Cópulas: Los Vines cópulas son modelos gráficos flexibles para describir cópulas

multivariantes usando una cascada de cópulas bivariadas, llamadas *par-cópulas*.

Toda función de densidad $f(x_1,...,x_n)$ puede ser escrita de la siguiente manera condicional y con funciones cópulas:

$$f(x_i \mid v) = c_{x_i v_i|v_{-j}}\left(F(x_i \mid, F(\ v_i \mid v_{-j}))\right) F(x_i \mid v_{-j})$$

donde $v = \{x_{i+1},...,x_n\}$ es el conjunto condicional de las distribuciones marginales de x_i, la variable $v_i \in v$ y v_{-j} es el conjunto de variables en v después de extraer v_i con $i = 1,...,n-1$ y $c(u_1,u_2)$ es la cópula de densidad definida como $\dfrac{\partial c(u_1,u_2)}{\partial u_1 \partial u_2}$.

Cuando $f(x_i|v)$ es iterativamente descompuesta, ésta se convierte en el producto bivariado de densidad de cópulas y funciones marginales de x_i. si estas marginales también son descompuestas de forma iterativa, entonces $f(x_1,...,x_n)$ es el producto de densidad de cópulas bivariadas y sus marginales. Dos especiales $pair-copulas$ son utilizadas y cuyas densidades son las siguientes:

C-Vine Cópula

$$f(x_1,...,x_n) = \prod_{k=1}^{n} f_k(x_k) \prod_{j=1}^{n-1} \prod_{i=1}^{n-j} c_{j,j+1|1,...,j-1}\left(F_{j|1,..,j-1}(x_j|x_{1,..,j-1}), F_{j+i|1,..,j-1}(x_{j+i}|x_{1,..,j-1})\right)$$

D-Vine Cópula

$$f(x_1,...,x_n) = \prod_{k=1}^{n} f_k(x_k) \prod_{j=1}^{n-1} \prod_{i=1}^{n-j} c_{j,i+j|i+1,...,i+j-1}\left(F_{j|i+1,..,i+j-1}(x_j|x_{i+1,..,i+j-1}), F_{j+i|i+1,..,i+j-1}(x_{j+i}|x_{i+1,..,i+j-1})\right)$$

Como se indicó, estas estructuras Vine pueden ser más fácilmente entendidas por medio de gráficos utilizando árboles.

Por ejemplo, para el caso de una función de densidad conjunta de 5 variables $f(x_1,x_2,x_3,x_4,x_5)$ seguimos los $pair-copulas$ generados en el arbol del D-vine cópula para la construcción de la densidad conjunta:

Densidad conjunta utilizando D-vine cópula:

$$\begin{aligned} f(x_1,x_2,x_3,x_4,x_5) \quad &= f_1(x_1) \times f_2(x_2) \times f_3(x_3) \times f_4(x_4) \times f_5(x_5) \\ &\times c_{12}(F_1(x_1), F_2(x_2)) \times c_{23}(F_2(x_2), F_3(x_3)) \times c_{34}(F_3(x_3), F_4(x_4)) \times c_{45}(F_4(x_4), F_5(x_5)) \\ &\times c_{13|2}(F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2)) \times c_{24|3}(F_{2|4}(x_2|x_4), F_{2|3}(x_2|x_3)) \times c_{35|4}(F_{3|4}(x_3|x_4), F_{5|4}(x_5|x_4)) \\ &\times c_{14|23}(F_{1|23}(x_1|x_2,x_3), F_{4|23}(x_4|x_2,x_3)) \times c_{25|34}(F_{2|34}(x_2|x_3,x_4), F_{5|34}(x_5|x_3,x_4)) \\ &\times c_{15|234}(F_{1|234}(x_1|x_2,x_3,x_4), F_{5|234}(x_5|x_2,x_3,x_4)) \end{aligned}$$

FIGURA N° 2.14: D-vine cópula: 5 variables, 4 árboles y 10 pair-cópulas

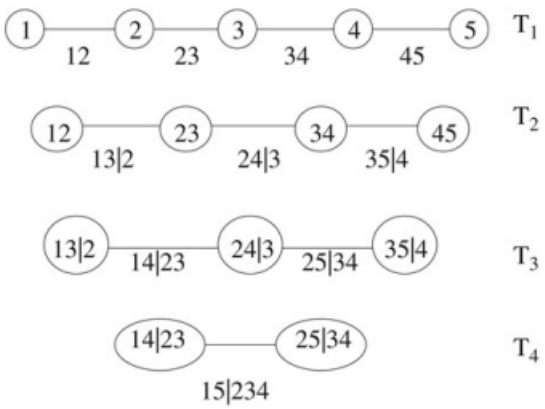
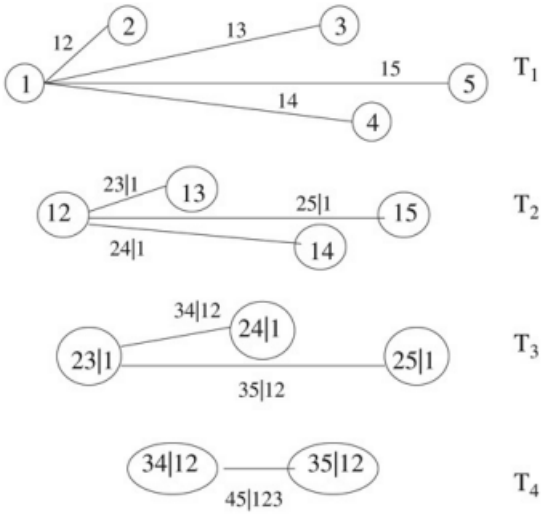


FIGURA N° 2.15: C-vine cópula: 5 variables, 4 árboles y 10 pair-cópulas



Densidad conjunta utilizando C-vine cópula:

$$
\begin{aligned}
f(x_1,x_2,x_3,x_4,x_5) \quad &= f_1(x_1)\times f_2(x_2)\times f_3(x_3)\times f_4(x_4)\times f_5(x_5)\\
&\times c_{12}(F_1(x_1),F_2(x_2))\times c_{13}(F_1(x_1),F_3(x_3))\times c_{15}(F_1(x_1),F_5(x_5))\times c_{14}(F_1(x_1),F_4(x_4))\\
&\times c_{23|1}(F_{2|1}(x_2|x_1),F_{3|1}(x_3|x_1))\times c_{25|1}(F_{2|1}(x_2|x_1),F_{5|1}(x_5|x_1))\times c_{24|1}(F_{2|1}(x_2|x_1),F_{4|1}(x_4|x_1))\\
&\times c_{34|12}(F_{3|12}(x_3|x_1,x_2),F_{4|12}(x_4|x_1,x_2))\times c_{35|12}(F_{3|12}(x_3|x_1,x_2),F_{5|12}(x_5|x_1,x_2))\\
&\times c_{45|123}(F_{4|123}(x_4|x_1,x_2,x_3),F_{5|123}(x_5|x_1,x_2,x_3))
\end{aligned}
$$

2.2.5.2 Alternativa 2: Agregación jerarquica de riesgos basado en cópula

Para fines de agregación de riesgos de alta dimensión, las clases de cópula más populares son demasiado restrictivas en términos de estructuras de dependencia alcanzables. Estas limitaciones se agravan a medida que aumentan las dimensiones. En Arbenz, Hummel, y

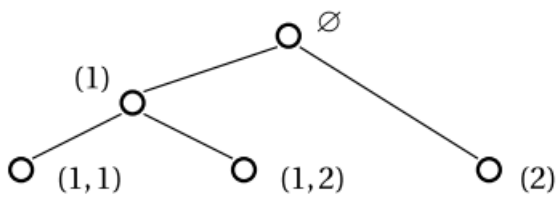
Mainik (2012) los autores plantean un método de agregación de riesgos jerárquico que es flexible en dimensiones altas. Nosotros seguiremos este trabajo para describir esta sección. Con este método basta con especificar una cópula de baja dimensión para cada paso de agregación en la jerarquía. Se pueden combinar cópulas y márgenes de cualquier tipo.

Definición 8. Un conjunto finito $\tau \subset \emptyset \cup \bigcup_{n=1}^{\infty} \mathbb{N}^n$ denota un árbol enraizado si:

1. La raiz $\emptyset$ esta contenida en τ.
2. para cada nodo $I = (i_1, ..., i_n) \in \tau$ el número de nodos hijos está dado por $N_I \in \mathbb{N}_0$, es decir, el nodo I tiene un hijo $(i_1, ..., i_n, k) \in \tau$ si y solo si $k \in \{n \in \mathbb{N} : 1 \leq n \leq N_I\}$,
3. cada nodo $(i_1, ..., i_d) \in \tau$ tiene un nodo pariente representado por $(i_1, ..., i_d) \in \tau$.

Ejemplo: Sea $\tau = \{\emptyset, (1), (1,1), (1,2), (2)\}$, como es ilustrado en la Figura N° 2.16, los nodos (1,1), (1,2) y (2) son nodos hojas. Los nodos $\emptyset$ y (1) son nodos ramas. Tenemos $N_\emptyset = N_1 = 2$ y $N_{1,1} = N_{1,2} = N_2 = 0$

FIGURA N° 2.16: Ilustración del árbol $\tau = \{\emptyset, (1), (1,1), (1,2), (2)\}$



Árboles de agregación por dependencia: Esta sección presenta un enfoque de agregación de riesgos basado en un árbol enraizado dado τ. Definimos sobre algún espacio de probabilidad $(\Omega, \mathfrak{A}, \mathbb{P})$, un vector aleatorio $(X_I)_{I \in \tau}$ que asigna a cada nodo $I \in \tau$ una variable aleatoria $X_I : \Omega \to \mathbb{R}$ tal que:

- Para cada nodo hoja $I \in \mathcal{L}(\tau)$, la variable X_I representan los riesgos cuyo agregado nos interesa,
- Los nodos ramas, $X_I, I \in \mathcal{B}(\tau)$, están dados por la agregación de los su hijos: $X_I = X_{I,1} + ... + X_{I,N_I}$

El enfoque de modelado consiste en definir distribuciones marginales $F_I, I \in \mathcal{L}(\tau)$ y las estructuras de dependencia para los pasos de agregación de modo que podamos calcular la distribución del agregado total $X_\emptyset$. Tenga en cuenta que por inducción

$$X_\emptyset = X_1 + ... + X_{N_\emptyset} = \sum_{I \in \mathcal{L}(\tau)} X_I$$

Con este enfoque podemos evitar modelar explícitamente la cópula de alta dimensión que describe la dependencia entre todos los nodos hoja $X_I, I \in \mathcal{L}(\tau)$. Para facilitar la notación,

eliminamos los corchetes de los vectores índice, así como el argumento τ siempre que el significado sea claro. Por ejemplo, $X_{(1,1)} = X_{1,1}, \mathcal{D}(I,\tau) = \mathcal{D}(I)$ y $\mathcal{LD}((1,1),\tau) = \mathcal{LD}(1,1)$

Sea la siguiente tripleta:

$$\left(\tau, (F_I)_{I\in\mathcal{L}(\tau)}, (C_I)_{I\in\mathcal{B}(\tau)}\right) \tag{2.16}$$

que consiste en:

- un árbol enraizado τ;
- Funciones de distribución $F_I : \mathbb{R} \to [0,1]$ para todo $I \in \mathcal{L}(\tau)$
- cópula $C_I : [0,1]^{\mathbb{N}_I} \to [0,1]$ para todo $I \in \mathcal{B}(\tau)$

Por ejemplo, en el ejemplo anterior podemos identificar a $\tau = \{\emptyset, (1), (1,1), (1,2), (2)\}$, tres distribuciones univariadas $F_{1,1}, F_{1,2}$ y F_2 y dos cópulas bivariadas C_1 y $C_\emptyset$.

Definición 9. Dada una tripleta como en (2.16), $(X_I)_{I\in\tau}$ es llamada ligeramente dependiente de los árboles si las siguientes condiciones son conocidas:

- Para cada nodo hoja $I \in \mathcal{L}$, la variable aleatoria X_I tiene una distirbución $F_I : \mathbb{P}[X_I \leq x] = F_I(x)$ para todo $x \in \mathbb{R}$.
- Para cada nodo rama $I \in \mathcal{B}, X_I$ es la suma de sus hijos, es decir, $X_I = \sum_{i=1}^{N_I} X_{I,i}$. La distribución de probabilidad marginal acumulada de X_I está denotada por $F_I : \mathbb{R} \to [0,1]$.
- Para cada nodo rama $I \in \mathcal{B}$, la estructura de dependencia de sus hijos $\mathcal{C}(I)$ está dada por la cópula C_I, es decir,

  $$\mathbb{P}\left[X_{I,1} \leq x_1, ..., X_{I,N_I} \leq x_{N_I}\right] = C_I\left(F_{I,1}(x_1), ..., F_{I,N_I}(x_{N_I})\right)$$

  , para todo $(x_1, ..., x_{N_I}) \in \mathbb{R}^{N_I}$

A continuación mostramos como se usa la agregación jerárquica de riesgos basado en cópula con un ejemplo explícito.

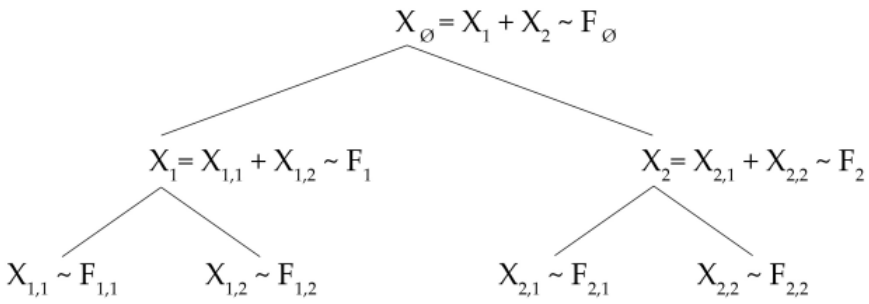
Ejemplo: Supongamos que se nos dan cuatro riesgos diferentes en un vector aleatorio $X = (X_{1,1}, X_{1,2}, X_{2,1}, X_{2,2})$. Aquí $X_{1,1}$ significa "Seguro de automóvil en Suiza" y $X_{1,2}$ significa "Seguro de automóvil en Italia". Además, $X_{2,1}$ significa "Terremoto Suiza" y $X_{2,2}$ significa "Terremoto Italia". En caso de que estemos interesados en el riesgo agregado $X_\emptyset := X_{1,1} + X_{1,2} + X_{2,1} + X_{2,2}$, podríamos intentar encontrar un modelo para la función de distribución conjunta F de X modelando primero las distribuciones marginales. $F_{1,1}, F_{1,2}, F_{2,1}$ y $F_{2,2}$ del riesgo individual e imponer un cópula C entre el riesgo individual. La distribución conjunta estaría entonces dada por

$$F(x_1, x_2, x_3, x_4) = C(F_{1,1}(x_1), F_{1,2}(x_2), F_{2,1}(x_3), F_{2,2}(x_4))$$

y la distribución de $X_\emptyset$ se puede calcular directamente a partir de F. Alternativamente, en un primer paso podríamos modelar las distribuciones de los riesgos parciales, $(X_{1,1}, X_{1,2})$ y $(X_{2,1}, X_{2,2})$, mediante combinando los riesgos $X_{1,1}$ y $X_{1,2}$ mediante una cópula bivariada C_1, mientras que $X_{2,1}$ y $X_{2,2}$ se combinan mediante una cópula bivariada C_2. Conociendo las distribuciones de los riesgos parciales, podemos calcular fácilmente la distribución de las sumas parciales $X_1 := X_{1,1} + X_{1,2}$ y $X_2 := X_{2,1} + X_{2,2}$. Las sumas parciales, X_1 y X_2, pueden luego combinarse nuevamente mediante un modelo de cópula bivariado adecuado $C_\emptyset$. Finalmente, esto nos permite calcular la distribución $F_\emptyset$ del agregado total $X_\emptyset := X_1 + X_2 = X_{1,1} + X_{1,2} + X_{2,1} + X_{2,2}$.

Este procedimiento se conoce como "agregación jerárquica de riesgos basada en cópulas" y se ilustra mejor mediante el llamado "árbol de agregación". El árbol de agregación representa gráficamente de qué manera están vinculados los riesgos individuales y qué estructura de dependencia se supone que existe entre ellos. La Figura N° 2.17 muestra el árbol de agregaciones correspondiente a la situación descrita anteriormente.

FIGURA N° 2.17: Una ilustración de un modelo de agregación de 4 dimensiones



La principal ventaja de la agregación jerárquica de riesgos es que no necesitamos especificar la cópula de todos los riesgos. Una vez más enfatizamos que es extremadamente improbable encontrar un modelo de cópula que describa adecuadamente la estructura de dependencia entre un gran número de riesgos. Las observaciones conjuntas entre todos los riesgos son demasiado raras y las estructuras de dependencia alcanzables de los modelos de cópula paramétricos comunes son demasiado limitadas. En lugar de ello, agregamos los riesgos jerárquicamente, lo que requiere únicamente especificar la dependencia conjunta entre las subcarteras agregadas en los diferentes pasos de agregación. Para estas subcarteras podemos utilizar cópulas de baja dimensión, que son bien conocidas por ser medidas de dependencia más realistas y ofrecer mayor flexibilidad. En particular, este modelo permite utilizar diferentes características de dependencia (dependencia de cola, asimetría radial, etc.) para cada paso de agregación.

El modelo puede verse como una herramienta de reducción de dimensiones: la complejidad y el número de parámetros se pueden ajustar a la complejidad necesaria y a la información disponible. Como consecuencia lógica, el método no especifica la distribución conjunta

completa de los riesgos individuales. En otras palabras, en general hay más de una distribución que satisface un modelo de árbol de agregación determinado. Llamaremos a esas distribuciones "ligeramente dependientes de los árboles".

2.3 Enfoque teórico asumido por el investigador

2.3.1 Prima pura sujeto a efectos de dependencia

En este trabajo se presentará una propuesta metodológica alternativa para el cálculo de la prima pura de riesgo, si bien se utilizará modelos lineales generalizados para modelar la frecuencia y la severidad en presencia de covariables, una herramienta de amplio uso en la industria aseguradora, mi propuesta se diferencia en el sentido que considero el grado de dependencia entre la frecuencia y la severidad. Además, las primas calculadas por cobertura serán agregadas considerando el grado de asociación que estas tienen, ya que en mi experiencia una cobertura activada suele estar acompañada de otras o tal vez ninguna al momento de la ocurrencia del siniestro. Agregarlas bajo una suma simple nos hace intuir rápidamente que existe una sobrestimación en las primas de riesgo calculadas.

2.3.1.1 Formulación metodológica

Para explicar de forma sencilla el modelo de prima de pura de riesgo (PPR) sujeto a dependencia, el modelo consta de los siguientes pasos:

1. Primero calculamos los modelos marginales de regresión generalizado de la frecuencia y de la severidad, esto por cada cobertura.
2. Las coberturas que se considerarán en este trabajo son las aplicadas en un seguro de automóviles:
   - PP: Pérdidas Parciales
   - PT: Pérdidas Totales
   - RC: Responsabilidad Civil
   - AS: Asistencias

   Considérese que estas coberturas hacen más de 95 % de la prima. En una venta real de estos seguros suelen ofrecerce muchas coberturas en productos de "Todo Riesgo".
3. Luego calculamos la densidad conjunta, en el caso de pérdidas parciales la densidad es $f_{PP}(X, N|\theta)$.
4. Con la funciones de densidad conjunta calculamos el valor esperado, que será la prima

pura de riesgo. En el caso de las pérdidas parciales tendremos

$$PPR_{PP} = \int_{l=0}^{\infty} l \times f_{PP}(X, N|\theta)$$

donde l representa a la pérdidas incurridas por cobertura de pérdidas parciales.

5. Una vez calculadas las primas puras de riesgo por coberturas procedemos a la agregación bivariada de las mismas mediante funciones cópulas. La prima pura de riesgo agregada de las coberturas Pérdida Parcial y Pérdida Total es

$$PPR_{PP+PT|\theta} = PPR_{PP} \oplus PPR_{PT}$$

donde el operador $\oplus$ es definido como una suma de riesgos con dependencia.

6. Por último, agregamos las primas bivariadas para obtener la prima total por las cuatro coberturas

$$PPR_{PP+PT+RC+AS|\theta} = (PPR_{PP} \oplus PPR_{PT}) \oplus (PPR_{RC} \oplus PPR_{AS})$$

En resumen el modelo propuesto tiene la siguiente forma esquemática, que va de derecha (modelos marginales) a izquierda (Prima Pura de riesgo conjunta total)

$$PPR_{PP+PT+RC+AS|\theta_7} \begin{cases} PPR_{PP+PT|\theta_5} \begin{cases} f_{PP}(X,N|\theta_1) \begin{cases} f_{PP}(N_{PP}) : \text{Modelo Marginal de la frecuencia} \\ f_{PP}(X_{PP}) : \text{Modelo Marginal de la Severidad} \end{cases} \\ f_{PT}(X,N|\theta_2) \begin{cases} f_{PT}(N_{PT}) : \text{Modelo Marginal de la frecuencia} \\ f_{PT}(X_{PT}) : \text{Modelo Marginal de la Severidad} \end{cases} \end{cases} \\ PPR_{RC+AS|\theta_6} \begin{cases} f_{RC}(X,N|\theta_3) \begin{cases} f_{RC}(N_{RC}) : \text{Modelo Marginal de la frecuencia} \\ f_{RC}(X_{RC}) : \text{Modelo Marginal de la Severidad} \end{cases} \\ f_{AS}(X,N|\theta_4) \begin{cases} f_{AS}(N_{AS}) : \text{Modelo Marginal de la frecuencia} \\ f_{AS}(X_{AS}) : \text{Modelo Marginal de la Severidad} \end{cases} \end{cases} \end{cases}$$

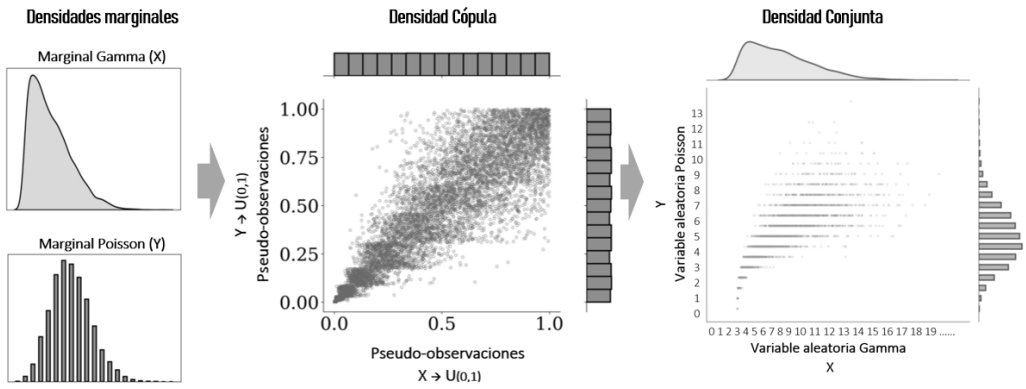
donde θ_i es el parámetro de dependencia de la cópula correspondiente.

Para las densidades conjuntas de la frecuencia y de la severidad el proceso es como el descrito en el punto 2.2.4.3, sin embargo en la Figura N° 2.18 podemos observar el proceso de modelación

Para la elección de la mejor función cópula, compararemos los $log-likelihoods$ de cada modelo, el máximo de ellos no dirá qué familia de cópula elegir.

Una vez definido la función de densidad conjunta, calculamos la prima pura de riesgo como

FIGURA N° 2.18: Densidad conjunta de la Frecuencia y Severidad



su valor esperado. Para ello utilizamos la fórmula definida en el Teorema 10.

Para realizar la suma de riesgos X e Y sujetos a dependencia, debemos conocer su distribución de probabilidad, en nuestro caso ambos son Gamma, dado que tanto X como Y representan al conjunto de primas de riesgos por coberturas. Con la data disponible (x_i, y_i) estimamos sus parámetros mediante máxima verosimilitud y procedemos a realizar los siguiente pasos:

- Generamos la primera variable aleatoria $\tilde{X}$ con distribución F_X, es decir:

$$u \sim U(0,1) \quad \rightarrow \quad \tilde{X} = F_X(u)$$

- Generamos la segunda variable aleatoria $\tilde{Y}$ con distribución F_Y, es decir:

$$v \sim U(0,1) \quad \rightarrow \quad \tilde{Y} = F_Y(v)$$

- Hallamos la primera derivada de la función cópula a partir de la densidad conjunta

$$P[X \leq \tilde{X}, Y \leq \tilde{Y}] = C(u,v)$$

  y la función condicional definida como

$$P[X \leq \tilde{X} | Y \leq \tilde{Y}] = \frac{\partial}{\partial u} C(u,v) = q(u,v)$$

- De $q(u,v)$ despejamos v para obtener $v = v(u,q)$, con ello podemos simular $u \sim U(0,1)$ y $q \sim U(0,1)$ para obtener los valores de v.

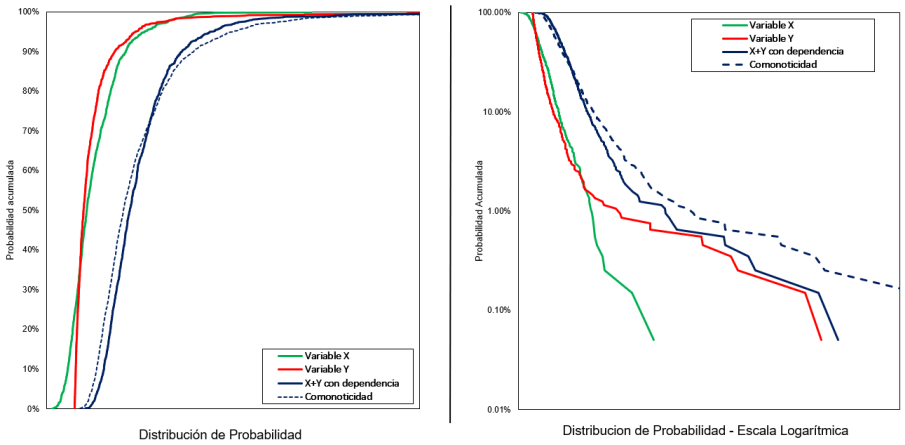
- Para las familias de cópulas que utilizaremos en este trabajo mostramos las expresiones de v para cada una:

TABLA N° 2.9: Expresiones de v por cada familia de cópulas

Familia	$C(u,v\|\theta)$	$q=\frac{\partial}{\partial u}C(u,v)$	v
Gauss	$\Phi_2(\Phi^{-1}(u),\Phi^{-1}(v)\|\theta)$	$\Phi\left(\frac{\Phi^{-1}(v)-\theta\Phi^{-1}(u)}{\sqrt{1-\theta^2}}\right)$	$\Phi\left(\theta\Phi^{-1}(u)+\Phi^{-1}(q)(\sqrt{1-\theta^2})\right)$
Clayton	$(u^{-\theta}+v^{-\theta}-1)^{-1/\theta}$	$(u^{-\theta}+v^{-\theta}-1)^{-1/\theta-1}u^{-\theta-1}$	$\left(1+u^{-\theta}(q^{\frac{-\theta}{\theta+1}}-1)\right)^{\frac{-1}{\theta}}$
Gumbel	$\exp\left(-((-\log u)^\theta+(-\log v)^\theta)^{1/\theta}\right)$	$u^{-1}\exp\left(-((-\log u)^\theta+(-\log v)^\theta)^{1/\theta}\right)$	$\exp\left(-((-\ln qu)^\theta-(-\ln u)^\theta)^{\frac{1}{\theta}}\right)$
Frank	$-\frac{1}{\theta}\log\left(1+\frac{(e^{-\theta u}-1)(e^{-\theta v}-1)}{(e^{-\theta}-1)}\right)$	$\frac{e^\theta(e^{\theta v}-1)}{e^{\theta(u+1)}+e^{\theta(v+1)}-e^\theta-e^{\theta(u+v)}}$	$\frac{1}{\theta}\ln\left(\frac{qe^\theta(e^{\theta u}-1)}{e^\theta qe^\theta(1-e^u)}\right)$

Una vez calculado u y v podemos obtener F_X y F_Y y F_{X+Y} y su correspondiente función de comonoticidad asociada a la suma de variables aleatorias. En el presente trabajo la prima total es la agregación bivariada de las primas por cobertura que se van calculando en cada paso. Las primas por cobertura se ajustan a una función de distribución de probabilidad Gamma, esto ya se dijo antes, cuyos parámetros son estimados mediante máxima verosimilitud.

FIGURA N° 2.19: Distribución de probabilidad de riesgos agregados



Fuente: Elaboración Propia usando el software R.

A modo de ejemplo, supongamos que tenemos dos primas $X=650$ y $Y=21$, siendo $X\sim Gamma(\alpha=0.98,\beta=798)$ y $Y\sim Gamma(\alpha=0.67,\beta=8.6)$, supongamos una cópula Clayton de parámetro $\theta=0.71$ para la agregación con dependencia de las primas de riesgo.

- Calculamos $u=F_X^{-1}(650)=0.57$ y $\hat{v}=F_Y^{-1}(21)=0.96$
- Con u y $\hat{v}$ podemos calcular $q=C_{clayton}(u=0.57,\hat{v}=0.96|\theta=0.71)=0.55$
- Por lo tanto, de acuerdo a la Tabla 2.9 obtenemos $v=0.61$.
- Por lo tanto, $650\oplus 21=F_X^{-1}(u)+F_Y^{-1}(v)=655$

2.4 Hipótesis

2.4.1 Hipótesis general

La propuesta metodológica del cálculo de la prima pura de riesgo sujeto a dependencia entre los riesgos de frecuencia y severidad, y agregados mediante funciones cópulas por cobertura nos permita estimaciones robustas y de acuerdo con el nivel de riesgo específico de los asegurados frente supuesto usual de independencia.

2.4.2 Hipótesis específicas

- Se puede construir un modelo de regresión conjunto de la frecuencia y de la severidad para estimar la prima pura de riesgo de una determinada cobertura.
- Se puede determinar el grado de dependencia entre la frecuencia y la severidad.
- La primas total se puede obtener mediante agregación de los riesgos asociados a cada cobertura mediante funciones cópulas.
- Se puede medir el grado de dependencia de las distribuciones de la primas de riesgo por cobertura.

2.5 Variables

2.5.1 Robustez

2.5.1.1 Definición conceptual

Se dice que una metodología es robusta respecto de las desviaciones de los supuestos del modelo, cuando el proceso continúa trabajando bien, aún cuando, en mayor o menor extensión, los supuestos no se mantienen.

2.5.1.2 Definición operacional

Para la medición de la robustez estadística del modelo GLM utilizaremos herramientas como *Simple Quantile Plots*, *Double Lift Charts*, *Loss Ratio Charts* y el Indice de Gini.

2.5.2 Asociación

2.5.2.1 Definición conceptual

La asociación estadística mide el grado de dependencia entre dos riesgos o variables, e implica una mayores aplicaciones que el coeficiente de correlación.

2.5.2.2 Definición operacional

El grado de asociación entre riesgos la mediremos mediante la τ de Kendall.

2.5.3 Desempeño

2.5.3.1 Definición conceptual

Se refiere a la comparación de los modelos candidatos, que se irán descartando mediante el uso de estadísticos que comparan el poder predictivo de los modelos frente a los datos observados.

2.5.3.2 Definición operacional

El desempeño de los modelos candidatos lo mediremos mediante el uso del $log-likelihood$ y la Devianza.

CAPÍTULO III: METODOLOGÍA

3.1 Tipo, nivel y diseño de la investigación

- Tipo de investigación es teórica.
- Nivel de investigación es explicativo.
- El diseño de la investigación va ser experimental.

3.2 Población, muestra y tamaño de muestra

3.2.1 Población

Sistema estadístico del seguros privado de seguros de automóviles de Brasil, cuya información pública está disponible está desde el año 2006. Los datos se obtuvieron de https://www2.susep.gov.br/menuestatistica/autoseg/principal.aspx

3.2.2 Muestra

Se seleccionó la información correspondiente de emisiones de pólizas de años 2015,2016, 2017, 2018 y 2019. Para la información de siniestros se tomaron los años 2015, 2016, 2017, 2018 y 2019. Sin embargo, para una buena estimación de la exposición, prima devengada[1] y siniestralidad se tomaron efectivamente los años 2016, 2017, 2018 y 2019.

3.2.3 Tamaño de muestra

La muestra seleccionada contiene *1,581,869* vehículos identificados por Número de póliza y número de certificado. Para los siniestros, se seleccionaron 4 coberturas: Pérdida Parcial *(N: 251,786 y M: 226,604,385 USD)*, Pérdida Total *(N: 6,684 y M:59,175,937 USD)*, Responsabilidad Civil *(N: 24,863 y M: 22,872,707 USD)* y Asistencias *(N: 14,630 y M: 638,490 USD)*. Donde: N es la cantidad de siniestros y M es el monto de siniestros incurridos en dólares.

3.3 Técnicas de análisis e instrumentos

3.3.1 Técnicas de Análisis

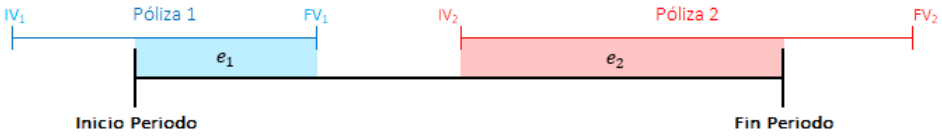
Para la estimación de la prima pura de riesgo debemos calcular de la mejor manera los siguientes indicadores actuariales que son vitales:

- **Expuestos:** Esta medida nos indica la proporción de tiempo en que un riesgo (Unidad

[1] No se pudo calcular de forma adecuada debido a la falta de información para separar los impuestos de la prima comercial

vehicular) estuvo expuesto respecto de un periodo determinado, como lo indica la Figura N° 3.1.

FIGURA N° 3.1: Medida de exposición de una póliza en un periodo de tiempo



Los periodos de tiempo que utilizaremos en nuestro estudio son los años indicados en la muestra. Una fórmula de cálculo de esta medida es la siguiente:

$$0 \leq e_i = \frac{Min(Fin_{periodo}, FV_i) - Max(Inicio_{periodo}, IV_i)}{Fin_{periodo} - Inicio_{periodo}} \leq 1$$

donde i es cada unidad vehicular con Inicio de Vigencia (IV) y Fin de Vigencia (FV) de la póliza.

- **Prima Devengada:** Es la prima que se gana en el periodo expuesto. Su cálculo por unidad vehicular es de la siguiente manera:

$$Prima_{DEV} = \frac{Min(Fin_{periodo}, FV_i) - Max(Inicio_{periodo}, IV_i)}{FV_i - IV_i} \times (\text{Prima Neta})$$

donde la Prima Neta es la prima cobrada por todo el periodo de su vigencia, sin IVA[2] y DE[3].

- **Siniestros Incurridos:** Esto monto inidica el valor del siniestro, a nivel de base de datos este monto es obtenido como:

$$\text{Incurrido} = \text{Reserva Pendiente} + \text{Pagos}$$

por cada siniestro y cobertura.

- **Siniestros Incurridos Últimos:** Como en la base de datos tenemos siniestros en estado Pagados y Pendientes, esto significa que existirá aún desarrollo de los siniestros. Para poder estimar el comportamiento último de los siniestros, calculamos los factores IBNR que nos permitirá estimar los siniestros últimos.

En virtud del modelo colectivo de riesgo, por unidad vehicular debemos calcular el monto total del siniestro S_i y la cantidad total del siniestro N, es decir, por póliza debemos calcular:

$$S_i = X_1 + ... + X_N,$$

[2]IVA: Impuesto a las Ventas
[3]DE: Derecho de Emisión

donde X_j $(1 \leq j \leq N)$ es cada siniestro de la unidad vehicular i.

Las medidas básicas definidas nos permitirá realizar el cálculo de la Frecuencia y de la Severidad en términos globales, por cada cobertura. Estos valores de la Frecuencia y de la Severidad nos ayudará en la validación de los interceptos en los modelos de regresión marginales.

De acuerdo a la formulación metodológica, por cobertura debemos construir 4 modelos de regresión conjunta, por cada familia de cópula considerada en este trabajo, y de acuerdo al criterio máximo $log-likelihood$ elegimos el mejor modelo.

El tratamiento de los datos se realizó utilizando gestores de base de datos como MS Access y SQL debido al alto volumen de los datos, y sobretodo por el alto performance necesarios en realizar las consultas y cruces respectivos. Las estimaciones máxima verosimiles de los parámetros de regresión y de las cópulas se realizaron en código R-Studio, utilizando los paquetes $Optim()$ y $CopulaRegresion()$.[4]

3.3.2 Instrumentos

Para poder comparar los distintos modelos de regresión se podrán utilizar los siguientes indicadores:

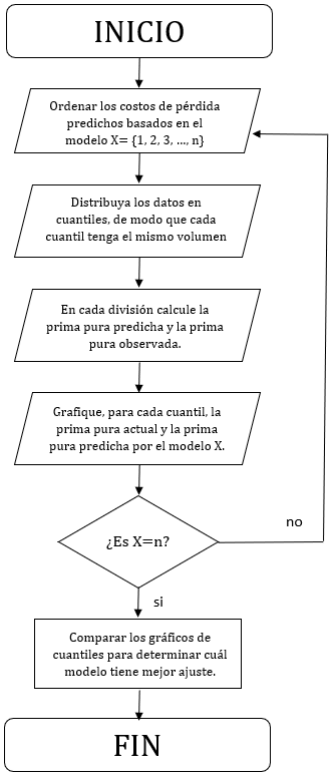
3.3.2.1 Indicadores de Robustez

- **Simple Quantile Plots:** Estos son una representación visual sencilla de la capacidad de un modelo para diferenciar con precisión entre los mejores y los peores riesgos. Suponga que hay dos modelos, el Modelo A y el Modelo B, los cuales producen una estimación del costo de la pérdida esperada para cada asegurado. Los gráficos de cuantiles simples se crean a través de los siguientes pasos mostrados en el algoritmo de la figura 3.2. Para determinar el mejor modelo, debemos considerar 3 criterios:

    - Precisión predictiva: Se refiere a qué tan bien es capaz cada modelo predecir la prima observada en cada cuantil.
    - Monoticidad: A medida que la prima pura predicha se incremente monotonicamente con los cuantiles, entonces la prima pura observada también se incrementará (y viceversa).
    - Distancia vertical entre el primer y último cuantil: El primer cuantil contiene los riesgos que el modelo cree tendrán una mejor experiencia, y el último cuantil contiene los riesgos que el modelo tendrán la peor experiencia. Una diferencia grande (llamada *lift*) entre la prima pura observada en los cuantiles con los costos de pérdida predichos más pequeños y más grandes indica que el modelo es capaz de distinguir al máximo los mejores y los peores riesgos.

[4] Paquete desarrollado en R por (Kholifah et al., 2019)

FIGURA N° 3.2: Algoritmo para la construcción del Simple Quantile Plot



- **Double lift Charts:** Este gráfico es similar *Simple Quantile Plots*, pero compara directamente dos modelos. Suponga que hay dos modelos, el Modelo A y el Modelo B, los cuales producen una estimación del costo de la pérdida esperada para cada asegurado. Este gráfico se crea a través de los siguientes pasos:

  - Calculamos por registro

$$\text{Sort ratio} = \frac{\text{Predicted cost from model A}}{\text{Predicted cost from model B}}$$

  - Ordenar los datos en función al Sort Ratio.
  - Dividir los datos en cuantiles, tales como quitiles o deciles.
  - Dentro de cada división, calcular la prima pura predicha promedio del modelo A, del modelo B y de los observados. Para cada una de esas cantidades, divida el promedio del cuantil por el promedio general.
  - Por cada cuantil, grafique los tres primas puras mencionadas anteriormente.

  En este gráfico, el modelo "ganador" es el que se acerca más a la prima pura real en cada cuantil.

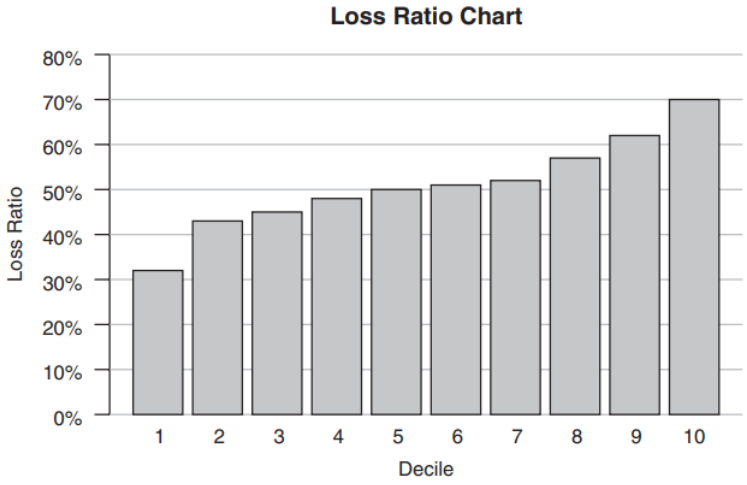
- **Loss Ratio Chart:** Este gráfico, en lugar de trazar la prima pura para cada categoría, se representa la tasa de siniestralidad. Los pasos para crear un gráfico de tasa de

pérdida son muy similares a los de crear un *Simple Quantile Plots*, con una diferencia importante:

- Ordenar los datos basados en las predicciones del modelo.
- Dividir los datos en cuantiles, de tal manera que cada cuantil tenga el mismo volumen de exposición.
- Dentro de cada división, calcular el *loss ratio* de los datos observados para los riesgos dentro la división.

Idealmente, el modelo es capaz de identificar deficiencias en el *scoring* actual al segmentar los riesgos en función del *loss ratio*. Por ejemplo, de acuerdo a la Figura 3.3.,

FIGURA N° 3.3: Ejemplo del *loss ratio chart*



podemos indicar que si un plan de scoring es perfecto, entonces todos los riesgos deberían tener el mismo *loss ratio*. El hecho de que este modelo sea capaz de segmentar los datos en divisiones de altos y bajos *loss ratios* es un fuerte indicador que el plan de scoring actual. La ventaje de este indicador es que es facil de entender y explicar. Tener en cuenta que el *loss ratio* es comunmente utilizado para medir la rentabilidad de la compañía de seguros.

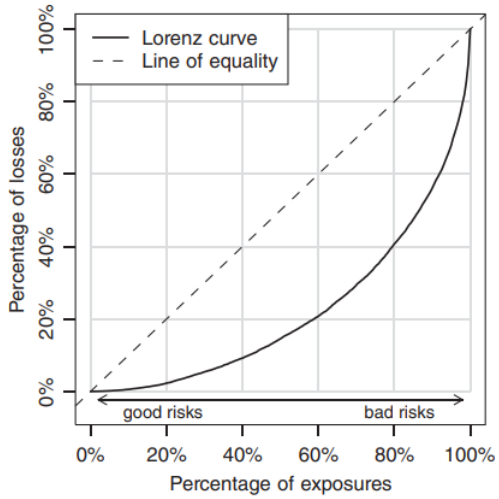
- **El Índice de GINI:** Llamado asi por el estadístico y sociólogo Corrado Gini, y es comúnmente usado en economía para cuantificar la desigualdad de ingresos en una nación. El índice de Gini puede también ser utilizado para medir el *lift* de un plan de calificación de seguros mediante la cuantificación de su capacidad para segmentar a la población en los mejores y peores riesgos. El índice de Gini para un modelo de predicción se calcula de la siguiente manera:
  - Ordenar los datos basados en el costo de pérdida predicho por el modelo. Los registros en la parte superior de los datos son entonces los riesgos en cual el

modelo cree son los mejores riesgos, y los registros al último de los datos son los riesgos el cual el modelo cree son los peores riesgos.

- En el eje x, grafique el porcentaje acumulado de los expuestos.
- En el eje y, grafique el porcentaje acumulado de las pérdidas.

El lugar geométrico de los puntos es la curva de Lorenz, y el índice de Gini es el doble del área entre la curva de Lorenz y la línea de igualdad.

FIGURA N° 3.4: Índice de Gini para un Modelo de Prima Pura



De la Figura N° 3.4, el ejemplo, identifica el 60% de los expuestos el cual contribuye solo el 20% del total de pérdidas. Su índice de GINI es del 56.1%. Note que este índice no cuantifica la rentabilidad de un particular plan de calificación de primas, pero sí cuantifica la capacidad del plan de calificación para diferenciar los mejores y los peores riesgos. Asumiendo que una aseguradora tiene precios y/o flexibilidad de suscripción, esto conducirá a una mayor rentabilidad.

3.3.2.2 Indicadores de Asociación

- **Tau de Kendall:** Dado (X_1, X_2) y (X_2, Y_2) independientes e idénticamentes distribuidas con cópula C,

$$\tau = Pr((X_1 - X_2)(Y_1 - Y_2) > 0) - Pr((X_1 - X_2)(Y_1 - Y_2) < 0)$$

Sin embargo, una forma de estimar τ de una muestra aleatoria de n pares $(x_i, y_i) \quad i = 1, 2, ..., n$ definimos la siguiente variable indicadora

$$A_{ij} = sgn(x_i - x_j)(y_i - y_j)$$

se puede notar que

$$\tau = E(A_{ij}) = (+1)Pr((x_i - x_j)(y_i - y_j) > 0) + (-1)Pr((x_i - x_j)(y_i - y_j) < 0)$$

De ello se deduce que un estimador insesgado del coeficiente de Kendall es la llamada τ de Kendall muestral:

$$\frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j>i} A_{ij}$$

- **Rho de Spearman:** Dado (X_1, X_2), (X_2, Y_2) y (X_3, Y_3) independientes e idénticamentes distribuidas con cópula C,

$$\rho = 3[Pr((X_1 - X_2)(Y_1 - Y_3) > 0) - Pr((X_1 - X_2)(Y_1 - Y_3) < 0)]$$

Sin embargo, una forma de estimar ρ de una muestra aleatoria de n pares $(x_i, y_i) \quad i = 1, 2, ..., n$ y recordando que ρ_s es la correlación de rango, uno puede cambiar a los rangos de las variables muestrales:

$$R_i = rank(X_i), \quad S_i = rank(Y_i)$$

donde la clasificación se tiene que hacer de orden ascendente. Al hacerlo se puede obtener la siguiente ρ muestral

$$\frac{\sum_{i=1}^{n}(R_i - \overline{R})(S_i - \overline{S})}{\sqrt{\sum_{i=1}^{n}(R_i - \overline{R})^2 \sum_{i=1}^{n}(S_i - \overline{S})^2}}$$

Teniendo en cuenta el hecho de que los rangos de n datos son los primeros n números enteros, la expresión anterior se simplifica en

$$1 - 6\frac{\sum_{i=1}^{n}(R_i - S_i)^2}{n(n^2 - 1)}$$

es la versión muestral de ρ y es un estimador insesgado.

3.3.2.3 Indicadores de Desempeño

- **Log-Likelihood:** Para cualquier conjunto dado de coeficientes, un GLM implica una media probabilística para cada registro. Eso, junto con el parámetro de dispersión y la forma de distribución elegida, implica una distribución de probabilidad completa. Por lo tanto, es posible calcular, para cualquier registro, la probabilidad (o densidad de probabilidad) que el GLM asignaría a la ocurrencia del resultado real que efectivamente ocurrió. Multiplicar esos valores en todos los registros produciría la probabilidad de que ocurran todos los resultados históricos; este valor se llama *log-likelihood*.

  Un GLM se ajusta encontrando el conjunto de parámetros para los cuales la probabili-

dad es la más alta. Esto es intuitivo; en ausencia de otra información, el mejor modelo es el que asigna la mayor probabilidad a los resultados históricos. Dado que la probabilidad suele ser un número extremadamente pequeño, el logaritmo de probabilidad, o *log-likelihood*, suele utilizarse en su lugar para que trabajar con él sea más manejable.

El *log-likelihood* por sí mismo puede ser difícil de interpretar. Por lo tanto, es útil relacionar el *log-likelihood* con sus límites superior e inferior hipotéticos alcanzables con los datos proporcionados.

En el extremo inferior de la escala se encuentra el *log-likelihood* del modelo nulo or **null model**, o un modelo hipotético sin predictores, solo una intersección. Tal modelo produciría la misma predicción para cada registro: la gran media. En el otro extremo se encuentra el modelo saturado, o un modelo hipotético con un número igual de predictores que registros en el conjunto de datos.

- **Devianza:** Esta medida para GLM se define como

$$Devianza = 1 \times (ll_{saturado} - ll_{modelo})$$

donde $ll_{saturado}$ es el *log-likelihood* del modelo saturado, y ll_{modelo} es el *log-likelihood* del modelo que está siendo evaluado. Esto puede expresarse más formalmente de la siguiente manera:

$$D = 2 \times \sum_{i=1}^{n} \ln f\left(y_i | \mu_i = y_i\right) - \ln f\left(y_i | \mu_i = \mu_i\right)$$

3.4 Operacionalización y Matriz de consistencia

3.4.1 Cuadro de operacionalización de variables

TABLA N° 3.1: Cuadro de operacionalización de variables

Variables	**Definición Conceptual**	**Dimensiones**	**Sub dimensiones**	**Indicadores**	**Tipo de variable**
Robustez	Se dice que una metodología es robusta respecto de las desviaciones de los supuestos del modelo, cuando el proceso continúa trabajando bien, aún cuando, en mayor o menor extensión, los supuestos no se mantienen.	Estabilidad	Precision	Quantile Plots	Razón
			Siniestralidad	Loss ratio Chart	Razón
			Desigualdad de riesgos	Indice de Gini	Razón
Asociación	La asociación estadística mide el grado de dependencia entre dos riesgos o variables.	Dependencia	Correlación por rangos	Tau de Kendall	Razón
Desempeño	Comparación de los modelos candidatos, que se descartando mediante el uso de estadísticos que comparan el poder predictivo de los modelos frente a los datos observados.	Probabilidad	Máxima Verosimilitud	Log-likelihood	Razón
			Cociente MVLE	Devianza	Razón

3.4.2 Matriz de consistencias

TABLA N° 3.2: Matriz de consistencia

Problemas	**Objetivos**	**Hipótesis**
General	**General**	**General**
Es posible determinar una propuesta metodológica del cálculo de la prima pura de riesgo sujeto a dependencia entre los riesgos de frecuencia y severidad, y agregados mediante funciones cópulas por cobertura que nos permita estimaciones robustas y de acuerdo con el nivel de riesgo específico de los asegurados frente supuesto usual de independencia	Determinar una propuesta metodológica del cálculo de la prima pura de riesgo sujeto a dependencia entre los riesgos de frecuencia y severidad, y agregados mediante funciones cópulas por cobertura que nos permita estimaciones robustas y de acuerdo con el nivel de riesgo específico de los asegurados frente supuesto usual de independencia.	La propuesta metodológica del cálculo de la prima pura de riesgo sujeto a dependencia entre los riesgos de frecuencia y severidad, y agregados mediante funciones cópulas por cobertura nos permita estimaciones robustas y de acuerdo con el nivel de riesgo específico de los asegurados frente supuesto usual de independencia.
Específicas	**Específicas**	**Específicas**
¿Es posible determinar un modelo de regresión conjunto de la frecuencia y de la severidad para estimar la prima pura de riesgo de una determinada cobertura?	Determinar un modelo de regresión conjunto de la frecuencia y de la severidad para estimar la prima pura de riesgo de una determinada cobertura.	Se puede construir un modelo de regresión conjunto de la frecuencia y de la severidad para estimar la prima pura de riesgo de una determinada cobertura.
¿Cómo podemos medir el grado de dependencia entre la frecuencia y la severidad?	Medir el grado de dependencia entre la frecuencia y la severidad.	Se puede determinar el grado de dependencia entre la frecuencia y la severidad.
¿Es posible determinar una propuesta metodológica para el cálculo de la prima total mediante agregación de los riesgos asociados de cada cobertura mediante funciones cópulas?	Determinar una propuesta metodológica de cálculo de la prima total mediante agregación de los riesgos asociados de cada cobertura mediante funciones cópulas.	La primas total se puede obtener mediante agregación de los riesgos asociados a cada cobertura mediante funciones cópulas.

CAPÍTULO IV: ANÁLISIS Y RESULTADOS

4.1 Análisis descriptivo de los datos

Para entender la información procederemos analizar la frecuencia y la severidad por cada una de las coberturas identificadaas en la base de datos. El periodo de análisis comprende desde el 01 de enero de 2016 hasta el 31 de diciembre de 2020, de acuerdo a los datos obetenidos de: https://www2.susep.gov.br/menuestatistica/autoseg/principal.aspx

FIGURA N° 4.1: Frecuencia y Severidad por cobertura y periodo de análisis

Podemos apreciar que para la Cartera Total la frecuencia está alrededor del 40 % en promedio y una severidad 1,000 USD; sin embargo, si lo separamos por cobertura, en el caso de las

Pérdidas Parciales la frecuencia es 30 % en promedio y una severidad 890 USD en promedio, para Péidadias Parciales la frecuencia es de 0.8 % en promedio y una severidad de 8,600 USD, para Responsabilidad Civil la frecuencia es de 3.1 % en promedio y una severidad de 1,100 USD y por últimos las Asistencias al Véhiculo tienen una frecuencia de 1.6 % en promedio y una severidad de 123 USD. Es posible notar un comportamiento distinto en el año 2020 por motivos de la pandemia del COVID-19, ya que siendo una estadística de datos reales es posibles este efecto.

Este análisis nos indica que debemos modelar de forma distinta tanto la frecuencia y la severidad por cobertura, además, descartar el año 2020 y realizar una predicción para los años futuros tomando los años desde el 2016 hasta el 2019, ya que por experiencia sabemos que los niveles prepapndemia se recuperan.

Acontinuación procedemos a describir las variables que se incluirán en los análisis y el modelamiento:

Tipo de póliza: Esta variable nos indica la forma de cómo se comercializa la póliza, que puede ser *Individual* o *Colectiva*. Una póliza individual generalmente solo cubre un vehículo, en cambio una póliza colectiva suele cubrir un colectivo de vehículos, generalmente de empresas y/o instituciones.

TABLA N° 4.1: Variable: Tipo de póliza

Cobertura	**Niveles**	Expuestos	Frecuencia	Severidad[1]	PMPY[2]
Pérdida Parciales	Individual	227,892	18 %	867	156
	Colectivo	817,699	26 %	906	234
Pérdida Totales	Individual	227,892	0.50 %	6,579	33
	Colectivo	817,699	0.68 %	9,324	63
Responsabilidad Civil	Individual	227,892	1.93 %	1,513	29
	Colectivo	817,699	2.50 %	793	20
Asistencias	Individual	227,892	0.63 %	139	0.9
	Colectivo	817,699	1.61 %	33	0.5

[1] En moneda USD

[2] Per Member Per Year: Monto incurrido de siniestro por cada unidad expuesta

Antiguedad del vehículo: Esta variable nos indica la edad del vehículo desde su fecha de fabricación hasta la fecha de emisión. En la base de datos se hallaron antiguedades de hasta 50 años, sin embargo, para su mejor análisis y rendimiento en el modelo estas son agrupadas, esta conversión se muestra en el anexo A. En la tabla 4.2 se muestran las antiguedades agrupadas.

TABLA N° 4.2: Variable: Antiguedad del vehículo

Cobertura	**Niveles**	Expuestos	Frecuencia	Severidad	PMPY
Pérdida Parciales	0	41,384	36%	859	305
	1	149,311	34%	896	308
	2	152,047	30%	940	279
	3	135,956	26%	970	251
	4	108,262	24%	926	222
	5	86,081	23%	945	219
	6	66,711	21%	871	185
	7	56,631	20%	840	168
	8	49,161	18%	856	157
	9	39,492	17%	814	138
	10	31,103	16%	782	123
	11-13	55,661	14%	801	115
	14-19	34,936	10%	682	68
	20+	38,854	5%	451	21
Pérdida Totales	0-1	190,695	1%	9443	80
	2	152,047	1%	8820	65
	3-5	330,299	1%	9636	62
	6	66,711	1%	9257	55
	7-16	254,595	1%	7336	37
	17+	51,243	0%	3309	9
Responsabilidad Civil	0-1	190,695	3%	736	19
	2-9	694,341	2%	950	23
	10-13	86,764	2%	997	22
	14+	73,790	2%	1085	20
Asistencias	0	41,384	1%	80	0.54
	1-2	301,358	1%	71	0.80
	3	135,956	1%	58	0.72
	4-5	194,343	1%	41	0.54
	6	66,711	2%	29	0.46
	7	56,631	2%	27	0.48
	8+	249,207	2%	24	0.46

Canales de venta: Esta variable nos indica el medio por el cual se realizó la venta de póliza. Se identificaron cinco canales principales: Banca-Digital, Instituciones, Agentes, Bróker, Exclusivos y otros. En la tabla 4.3 mostramos la frecuencia y la severidad por cada nivel.

TABLA N° 4.3: Variable: Canal de venta

Cobertura	**Niveles**	Expuestos	Frecuencia	Severidad	PMPY
Pérdida Parciales	Agentes	167,733	27%	903	242
	Banca-Digital	293,233	25%	893	226
	Bróker	290,420	22%	953	210
	Exclusivos	190,518	28%	845	232
	Instituciones	76,402	9%	900	83
	Otros	27,285	33%	879	293
Pérdida Totales	Agentes	167,733	1%	9317	72
	Banca-Digital	293,233	1%	9031	62
	Bróker	290,420	1%	9784	54
	Exclusivos	190,518	1%	7664	49
	Instituciones	76,402	0%	5620	25
	Otros	27,285	1%	9260	84
Responsabilidad Civil	Agentes	167,733	2%	790	19
	Banca-Digital	293,233	2%	675	14
	Bróker	290,420	2%	1058	26
	Exclusivos	190,518	3%	1042	33
	Instituciones	76,402	1%	1099	11
	Otros	27,285	3%	1055	36
Asistencias	Agentes	167,733	1%	44	0.61
	Banca-Digital	293,233	2%	27	0.40
	Bróker	290,420	1%	62	0.85
	Exclusivos	190,518	1%	42	0.59
	Instituciones	76,402	0%	47	0.13
	Otros	27,285	4%	46	1.74

Uso del vehículo: Esta variable nos indica el tipo de uso que se realiza con el vehículo. Se hallaron 20 usos en la base de datos, de las cuales lo reducimos en solo 7 usos. En la Tabla 4.4 mostramos la frecuencia y severidad por cada nivel.

TABLA N° 4.4: Variable: Uso del vehículo

Cobertura	**Niveles**	Expuestos	Frecuencia	Severidad	PMPY
Pérdida Parciales	Particular	621,227	2%	24	0.47
	Alquiler	31,250	1%	109	1.56
	Taxi	43,206	1%	31	0.35
	Comercial	96,993	1%	62	0.35
	Transporte Personal	40,394	0%	124	0.50
	Otros-Usos	51,518	0%	86	0.34
	Carga	161,002	0%	520	1.30
Pérdida Totales	Alquiler	31,250	34%	817	275
	Particular	621,227	29%	847	248
	Transporte Personal	40,394	23%	792	182
	Otros-Usos	51,518	22%	819	177
	Comercial	96,993	15%	799	119
	Carga	161,002	12%	736	88
	Taxi	43,206	11%	1948	223
Responsabilidad Civil	Alquiler	31,250	1%	7577	91
	Taxi	43,206	1%	11793	120
	Comercial	96,993	1%	4582	42
	Particular	621,227	1%	5898	39
	Transporte Personal	40,394	0%	11553	48
	Carga	161,002	0%	12131	40
	Otros-Usos	51,518	0%	23529	56
Asistencias	Particular	621,227	3%	599	17
	Carga	161,002	2%	1447	32
	Otros-Usos	51,518	2%	717	15
	Comercial	96,993	2%	957	15
	Transporte Personal	40,394	1%	1646	22
	Alquiler	31,250	1%	2392	31
	Taxi	43,206	1%	1894	23

Clase del vehículo: Esta variable nos indica si el vehículo es Liviano, Pesado o un Vehículo Menor. En la Tabla 4.5 mostramos la frecuencia y severidad por cada nivel.

TABLA N° 4.5: Variable: Clase del vehículo

Cobertura	**Niveles**	Expuestos	Frecuencia	Severidad	PMPY
Pérdida Parciales	Livianos	765,062	28%	824	231
	Veh. Menor	56,112	6%	499	30
	Pesados	207,983	14%	1547	222
Pérdida Totales	Livianos	765,062	28%	8223	2309
	Veh. Menor	56,112	6%	1798	106
	Pesados	207,983	14%	22369	3217
Responsabilidad Civil	Livianos	765,062	28%	651	183
	Veh. Menor	56,112	6%	965	57
	Pesados	207,983	14%	2133	307
Asistencias	Livianos	765,062	28%	29	8
	Veh. Menor	56,112	6%	36	2
	Pesados	207,983	14%	426	61

Tipo del vehículo: Esta variable nos muestra de una forma más desagregada la característica del vehículo. Se hallaron más de 20 tipos de vehículos los cuales fueron agrupadas de forma distinta por cobertura. En la Tabla 4.6 mostramos la frecuencia y severidad por cada nivel.

TABLA N° 4.6: Variable: Tipo de vehículo

Cobertura	**Niveles**	Expuestos	Frecuencia	Severidad	PMPY
Pérdida Parciales	Camioneta Rural	297,548	30%	22	7
	Automóvil	325,220	28%	27	8
	Camioneta Pick Up	116,260	23%	32	7
	Otros Livianos	26,033	22%	50	11
	Motocicletas	54,185	6%	47	3
	Otros Veh. Menores	1,926	0%	0	0
	Omnibus	22,649	29%	2	1
	Microbus	21,914	18%	21	4
	Camion	108,037	12%	9	1
	Remolcador y Semirremolque	55,384	12%	4	0
Pérdida Totales	Camioneta Pick Up	116,260	1%	50	0
	Automóvil	325,220	1%	98	1
	Camioneta Rural	297,548	1%	120	1
	Otros Livianos	26,033	1%	259	1
	Motocicletas	54,185	1%	22	0
	Otros Veh. Menores	1,926	0%	0	0
	Pesados	207,983	0%	44	0
Responsabilidad Civil	Camioneta Panel	21,780	3%	1989	60
	Camioneta Rural	297,548	3%	1496	43
	Camioneta STW	4,253	3%	0	0
	Automóvil	325,220	3%	1240	33
	Camioneta Pick Up	116,260	2%	845	13
	Motocicletas	54,185	1%	288	2
	Otros Veh. Menores	1,926	0%	0	0
	Omnibus	22,649	3%	612	16
	Remolcador y Semirremolque	55,384	2%	186	4
	Camion	108,037	2%	383	8
	Microbus	21,914	1%	1106	16
Asistencias	Automóvil	325,220	2%	1031	23
	Camioneta Rural	297,548	2%	1277	23
	Otros Livianos	142,294	1%	831	7
	Veh. Menores	56,112	0%	207	1
	Pesados	207,983	0%	296	1

Marca del Vehículo: Esta variable nos indica la marca del vehículo asegurado. En la base de datos se hallaron diverasas marcas, y por ser tan numerosos, lo que se hizo fue realizar un análisis pareto y centrarnos en las principales marcas, ademas, esto se hizo por clase de vehículo. En la base de datos no se tenía el dato del modelo asociado a la marca. En el anexo A mostraremos el detalle de las marcas utilizadas y modelo utilizado para su agrupación. En la Tabla 4.7 mostramos la frecuencia y severidad por cada nivel.

TABLA N° 4.7: Variable: Marca del vehículo y por clase

Cobertura	**Niveles**	Expuestos	Frecuencia	Severidad	PMPY
Pérdida Parciales	L1	44,293	41%	1042	424
	L2	72,561	35%	1029	355
	L3	541,482	28%	797	220
	L4	80,629	22%	619	139
	L5	26,097	16%	763	125
	P1	463	54%	343	185
	P2	839	40%	645	261
	P3	37,516	24%	1491	354
	P4	130,559	14%	1634	224
	P5	38,607	6%	1361	87
	M1	1,040	28%	1256	354
	M2	4,103	13%	1067	141
	M3	50,968	5%	285	14
Pérdida Totales	L1	1,679	2%	5076	82
	L2	6,327	1%	7496	83
	L3	271,591	1%	8663	77
	L4	485,465	1%	7889	45
	P1	2,738	1%	17958	164
	P2	13,839	1%	16549	90
	P3	76,503	0%	24093	92
	P4	114,904	0%	22538	50
	M1	1,450	6%	3303	189
	M2	15,410	2%	1318	28
	M3	39,252	1%	1885	17
Responsabilidad Civil	L1	26,132	4%	526	21
	L2	52,559	3%	509	17
	L3	278,741	3%	555	17
	L4	197,972	2%	695	17
	L5	209,659	2%	894	18
	P1	52,383	3%	2382	81
	P2	56,270	2%	2050	42
	P3	99,330	2%	1899	29
	M1	2,395	2%	654	13
	M2	6,647	1%	1640	20
	M3	15,839	1%	907	7
	M4	31,231	0%	655	2
Asistencias	L1	271,436	3%	16	0.40
	L2	281,532	2%	28	0.46
	L3	206,342	1%	64	0.76
	L4	5,753	1%	36	0.26
	P1	1,416	1%	100	0.63
	P2	8,646	0%	206	1.02
	P3	39,455	0%	165	0.65
	P4	119,041	0%	602	1.39
	P5	39,426	0%	281	0.09
	M1	1,040	2%	50	1.15
	M2	10,063	1%	46	0.34
	M3	45,009	0%	23	0.04

Zona demográfica: Esta variable nos indica la zona demográfica donde vive el asegurado. Lo ideal para esta variable sería registrar la zona de circulación del asegurado, pero normalmente no este campo no existe o está incompleto. Todas las zonas halladas se codifican con A para referirse a las provincias y con D a los disitritos de mayor densidad. En la Tabla 4.8 mostramos la frecuencia y severidad por cada nivel.

TABLA N° 4.8: Variable: Zona demográfica

Cobertura	**Niveles**	Expuestos	Frecuencia	Severidad	PMPY
Pérdida Parciales	A1	756	36%	701	253
	A2	474,830	28%	789	219
	A3	236,273	23%	925	213
	A4	222	16%	840	136
	A5	51,109	23%	743	174
	A6	10,301	21%	1028	221
	D1	81,738	22%	1111	248
	D2	159,887	18%	1200	219
	D3	10,607	14%	1320	189
	D4	19,869	11%	1668	187
Pérdida Totales	A1	6,655	1%	10749	132
	A2	83,219	1%	9813	94
	A3	622,206	1%	8157	48
	A4	51,109	1%	7385	43
	A5	10,301	1%	8252	75
	D1	73,410	1%	7358	63
	D2	73,280	1%	11081	75
	D3	101,730	1%	11856	62
	D4	23,680	0%	13090	46
Responsabilidad Civil	A1	587	5%	817	38
	A2	501,513	3%	691	22
	A3	209,612	2%	1367	29
	A4	51,477	3%	992	30
	A5	10,301	1%	1961	29
	D1	5,828	2%	404	10
	D2	61,350	1%	1533	20
	D3	64,764	1%	1670	17
	D4	112,674	1%	1334	11
	D5	27,485	1%	1492	8
Asistencias	A1	293,398	3%	15	0.39
	A2	112,298	2%	21	0.38
	A3	306,385	1%	76	0.79
	A4	51,109	1%	26	0.25
	A5	10,301	1%	72	0.50
	D1	5,828	2%	22	0.38
	D2	18,112	1%	103	0.98
	D3	12,527	1%	108	0.71
	D4	227,769	0%	277	0.84
	D5	7,865	0%	431	0.60

Género del asegurado: Esta variable nos indica el género del asegurado, que no necesariamente es el conductor del vehículo. Esta variable tendrá un nivel de ”No Aplica”, y está relacionado al tipo de persona jurídica, y en algunos casos por falta de información, pero

es de cantidad despreciable. En la Tabla 4.9 mostramos la frecuencia y severidad por cada nivel.

TABLA N° 4.9: Variable: Género del asegurado

Cobertura	**Niveles**	Expuestos	Frecuencia	Severidad	PMPY
Pérdida Parciales	Femenino	642,185	23%	932	214
	No Aplica	5,146	20%	962	190
	Masculino	398,259	26%	853	221
Pérdida Totales	Femenino	642,185	1%	10076	56
	No Aplica	5,146	1%	8257	51
	Masculino	398,259	1%	7433	57
Responsabilidad Civil	Femenino	642,185	2%	1023	24
	No Aplica	5,146	1%	930	7
	Masculino	398,259	2%	762	19
Asistencias	Femenino	642,185	1%	48	0.65
	No Aplica	5,146	1%	70	0.84
	Masculino	398,259	1%	37	0.55

Tipo de Persona: Esta variable nos indica si el asegurado es una Persona Natural o una Persona Jurídica (Empresas). En la Tabla 4.10 mostramos la frecuencia y severidad por cada nivel.

TABLA N° 4.10: Variable: Tipo de Persona

Cobertura	**Niveles**	Expuestos	Frecuencia	Severidad	PMPY
Pérdida Parciales	Empresa	550,779	20%	1028	208
	Personas	494,811	28%	798	226
Pérdida Totales	Empresa	550,779	1%	10467	60
	Personas	494,811	1%	7387	52
Responsabilidad Civil	Empresa	550,779	2%	1321	28
	Personas	494,811	3%	566	15
Asistencias	Empresa	550,779	1%	77	0.71
	Personas	494,811	2%	26	0.50

Edad del asegurado: Esta variable nos permitirá captar el comportamiento de la frecuencia de acuerdo a la edad del asegurado. En la Tabla 4.11 mostramos la frecuencia y severidad por cada nivel.

Suma Asegurada: Esta variable se refiere al valor del vehículo al momento de la suscripción. Para su análisis se dividió en rangos sugeridos, con la finalidad de obtener un comportamiento uniforme por cada cobertura. Estos rangos son mostrados en el Anexo A, y se puede observar como la frecuencia se comporta distinto por cada cobertura, por ejemplo, para las Pérdidas Parciales la relación directa, sin embargo, para el resto de coberturas la relación es inversa.

TABLA N° 4.11: Variable: Edad del asegurado

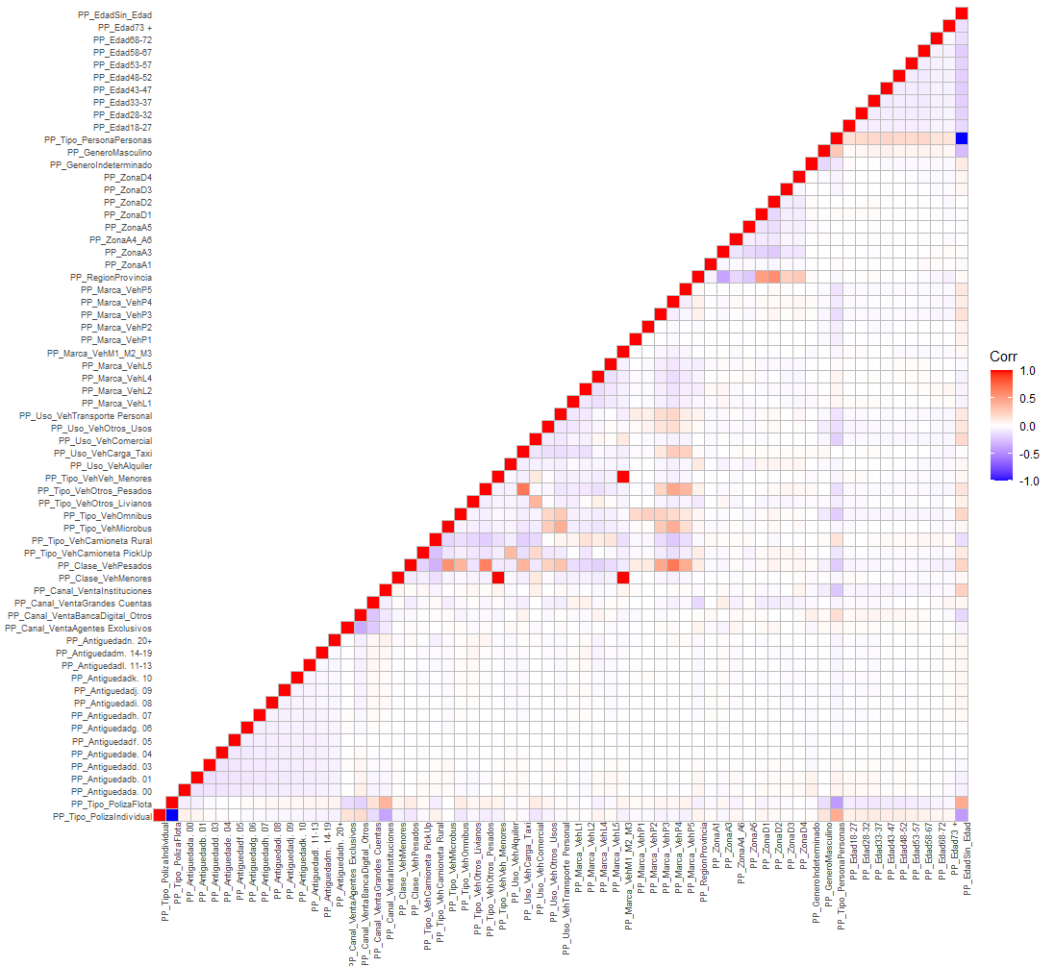
Cobertura	**Niveles**	Expuestos	Frecuencia	Severidad	PMPY
Pérdida Parciales	18-27	29,717	36%	838	304
	28-32	57,175	34%	797	269
	33-37	67,212	32%	804	261
	38-42	69,025	30%	805	240
	43-47	62,269	28%	800	227
	48-52	52,422	27%	799	213
	53-57	45,667	26%	767	200
	58-67	68,686	24%	778	184
	68-72	20,467	21%	818	169
	73 +	22,172	18%	764	135
	Sin Edad	550,779	20%	1028	208
Pérdida Totales	18-27	29,717	1%	6808	92
	28-32	57,175	1%	6972	75
	33-37	67,212	1%	7371	61
	38-42	69,025	1%	8392	61
	43-52	114,691	1%	7439	45
	53-57	45,667	1%	7451	40
	58-62	38,181	0%	8382	39
	63-77	62,800	0%	6828	30
	78 +	10,343	0%	4644	14
	Sin Edad	550,779	1%	10467	60
Responsabilidad Civil	18-27	29,717	3%	461	16
	28-32	57,175	3%	610	17
	33-37	67,212	3%	553	15
	38-42	69,025	3%	626	16
	43-52	114,691	3%	576	15
	53-57	45,667	3%	542	14
	58-62	38,181	3%	551	15
	63-67	30,505	2%	502	12
	68-72	20,467	3%	507	13
	73-77	11,828	3%	861	22
	78-82	6,239	3%	424	11
	83-87	2,669	2%	338	8
	88+	1,435	3%	445	12
	Sin Edad	550,779	2%	1321	28
Asistencias	18-27	29,717	2%	55	0.89
	28-32	57,175	2%	44	0.75
	33-37	67,212	2%	34	0.60
	38-42	69,025	2%	29	0.51
	43-47	62,269	2%	21	0.40
	48-52	52,422	2%	23	0.45
	53-57	45,667	2%	25	0.49
	58-67	68,686	2%	13	0.28
	68-72	20,467	2%	14	0.33
	73-82	18,067	3%	6	0.17
	83+	4,104	3%	9	0.32
	Sin Edad	550,779	1%	77	0.71

4.2 Análisis de los datos

4.2.1 Modelos Marginales de la Frecuencia y de la Severidad

Matriz de correlaciones De las variables analizadas procedemos a analizar la matriz de correlación para verficar posible multicolinealidad entre las variables, por ejemplo, para la cobertura Pérdida Parcial

FIGURA N° 4.2: Matriz de correlación - Pérdida Parcial



la matriz de correlación nos indica que que existen variables correlacionadas, de forma positiva y negativa. Esta información se tomará en cuenta para construir el modelo marginal y evitar problemas de multicolinealidad. Aunque, en la codificación se utilizó el paquete *Alias* del R para detectar los problemas de multicolinealidad, sin embargo, las conclusiones fueron similares. Esto se realizó para el resto de coberturas, cuyos resultados se muestran en el Anexo B.

Modelos Lineales Generalizados Marginales Habiendo análizado las posibles causas de multicolinealidad procedemos a construir los modelos marginales GLM de la Frecuencia y de la Severidad por cobertura. Los códigos en R se muestran en el Anexo B.

TABLA N° 4.12: Modelo Marginal de Frecuencia y Severidad - Pérdidas Parciales

var-level	Frecuencia (β)	Severidad (α)	var-level	Frecuencia (β)	Severidad (α)
(Intercept)	-1.4958	6.8387	Marca-VehL3	0.0000	0.0000
Antiguedada. 00	0.1930	-0.0258	Marca-VehL4	-0.1444	-0.1167
Antiguedadb. 01	0.1599	-0.0229	Marca-VehL5	-0.5007	0.1557
Antiguedadc. 02	0.0000	0.0000	Marca-VehM1-M2-M3	-0.6576	0.4385
Antiguedadd. 03	-0.1428	0.0377	Marca-VehP1	0.4069	-0.5971
Antiguedade. 04	-0.2258	-0.0109	Marca-VehP2	0.5295	0.0212
Antiguedadf. 05	-0.2515	0.0250	Marca-VehP3	0.4690	0.3193
Antiguedadg. 06	-0.3098	-0.0067	Marca-VehP4	0.0786	0.2637
Antiguedadh. 07	-0.3498	-0.0193	Marca-VehP5	-0.6642	0.2355
Antiguedadi. 08	-0.3780	0.0094	SAa.[1000-2000>	-1.2163	-0.7024
Antiguedadj. 09	-0.4083	-0.0028	SAb.[2000-3000>	-0.5825	-0.6936
Antiguedadk. 10	-0.4512	-0.0422	SAc.[3000-4000>	-0.4271	-0.5761
Antiguedadl. 11-13	-0.4880	0.0043	SAd.[4000-5000>	-0.3653	-0.4428
Antiguedadm. 14-19	-0.7549	-0.0800	SAe.[5000-6000>	-0.2691	-0.3932
Antiguedadn. 20+	-1.1901	-0.2489	SAf.[6000-7000>	-0.2375	-0.3658
Canal-VentaAgentes	0.0814	0.0102	SAg.[7000-8000>	-0.2229	-0.2953
Canal-VentaBancaDigital-Otros	-0.0701	-0.0047	SAh.[8000-9000>	-0.1767	-0.2781
Canal-VentaBroker	0.0000	0.0000	SAi.[9000-13000>	-0.1302	-0.2048
Canal-VentaExclusivos	0.0989	-0.0127	SAj.[13000-999999]	0.0000	0.0000
Canal-VentaInstituciones	-0.5083	0.0984	Tipo-PolizaFlota	0.1442	-0.1310
Edad18-27	0.1404	0.0531	Tipo-PolizaIndividual	0.0000	0.0000
Edad28-32	0.1090	-0.0190	Uso-VehAlquiler	0.0664	0.0340
Edad33-37	0.0717	-0.0123	Uso-VehCarga-Taxi	-0.8514	0.6021
Edad38-42	0.0000	0.0000	Uso-VehComercial	-0.3486	0.1093
Edad43-47	-0.0250	-0.0335	Uso-VehOtros-Usos	-0.2774	-0.1705
Edad48-52	-0.0749	-0.0187	Uso-VehParticular	0.0000	0.0000
Edad53-57	-0.0962	-0.0573	Uso-VehTransporte Personal	-0.2954	-0.2249
Edad58-67	-0.1921	-0.0366	ZonaA1	0.2952	-0.0801
Edad68-72	-0.2939	0.0246	ZonaA2	0.0000	0.0000
Edad73 +	-0.3475	0.0082	ZonaA3	-0.0485	0.0556
EdadSin-Edad	-0.0412	0.1208	ZonaA4-A6	-0.2146	0.1521
GeneroFemenino	0.0000	0.0000	ZonaA5	0.1168	-0.0924
GeneroIndeterminado	0.1069	0.0573	ZonaD1	-0.2794	0.2113
GeneroMasculino	0.0174	0.0290	ZonaD2	-0.3243	0.2174
Marca-VehL1	0.2538	0.2192	ZonaD3	-0.6388	0.2635
Marca-VehL2	0.1680	0.1789	ZonaD4	-0.5521	0.6066

Los modelos de regresión marginal construidos son la base de nuestro estudio, es por ello que debemos revisar la robustez de los modelos mediante el índice de GINI, estas se muestran en la figura Figura N° 4.3. Como se puede observar, el modelo GLM de la cobertura Pérdida Parcial tiene una capacidad de segmentar a lo malos y buenos riesgos en un 49.11 %, el modelo GLM de la cobertura Pérdida Total tiene una capcidad de 46.82 %, el modelo GLM de la cobertura Responsabilidad Civil tiene una capacidad de 53.28 % y el modelo GLM de la cobertura Asistencias tiene una capacidad de 64.26 %. De acuerdo a la literatura estándar, los valores de los índices obtenidos indican una buena capacidad de poder segmentar los datos.

4.2.2 Pruebas de indicios de Dependencia

En esta sección, mostraremos la existencia de cierta dependencia entre la frecuencia y la severidad, que usualmente suelen asumirse como independientes. Utilizaremos en primera instancia gráficos para revelar la dependencia.

TABLA N° 4.13: Modelo Marginal de Frecuencia y Severidad - Pérdidas Totales

var-level	Frecuencia (β)	Severidad (α)	var-level	Frecuencia (β)	Severidad (α)
(Intercept)	-6.0207	8.8907	Marca-VehM3	0.6250	-1.1767
Edad18-27	0.6982	-0.0300	Marca-VehP1-P2-P3-P4	-0.7029	0.6995
Edad28-32	0.4720	-0.0070	Tipo-PolizaFlota	-0.4449	-0.0767
Edad33-42	0.1939	0.0781	Tipo-PolizaIndividual	0.0000	0.0000
Edad43-52	0.0000	0.0000	Uso-VehAlquiler	0.5607	0.3007
Edad53-77	-0.2055	0.0419	Uso-VehAmbul-TranspIntProv-Carga-Tur-Esc	0.3141	0.2627
Edad78 +	-0.7924	-0.1815	Uso-VehComercial-Taxi-TransPers	0.1760	-0.1017
EdadSin-Edad	0.1872	0.2343	Uso-VehOtros-Usos	-0.7710	-0.8111
GeneroFemenino	0.0000	0.0000	Uso-VehParticular	0.0000	0.0000
GeneroMasculino	0.1818	0.0323	ZonaA1	0.7206	0.3403
Marca-VehL1	0.8985	-0.3399	ZonaA2-A5	0.4227	0.0544
Marca-VehL2	0.6156	0.0507	ZonaA3	0.0000	0.0000
Marca-VehL3	0.3973	0.1151	ZonaA4	0.1341	-0.1399
Marca-VehL4	0.0000	0.0000	ZonaD1	0.1883	0.1120
Marca-VehM1	2.1597	-0.5939	ZonaD2-D3	-0.1291	0.0208
Marca-VehM2	1.4703	-1.4691	ZonaD4	-0.5374	0.3940

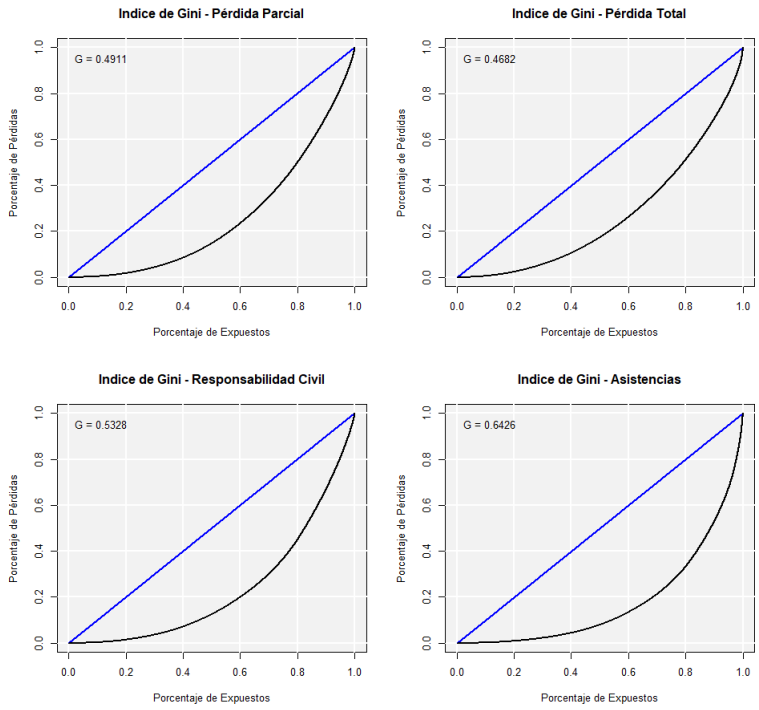
TABLA N° 4.14: Modelo Marginal de Frecuencia y Severidad - Responsabilidad Civil

var-level	Frecuencia (β)	Severidad (α)	var-level	Frecuencia (β)	Severidad (α)
(Intercept)	-4.5046	6.8322	Marca-VehP1-P2	0.3242	0.6791
Antiguedada. 0-1	0.1522	0.0148	Marca-VehP3	-0.1802	0.4541
Antiguedadb. 2-9	0.0000	0.0000	Uso-VehSerenazgo	0.8643	0.2877
Antiguedadc. 10+	-0.1068	-0.2256	Uso-VehTransInterProv-Urb	0.1963	-0.1287
Canal-VentaOtros	0.2161	-0.2133	Uso-VehParticular	0.0000	-2.2461
Canal-VentaExclusivos-Agentes	0.0435	0.0000	Uso-VehAlq-Amb-Carg-Com	-0.2263	0.0000
Canal-VentaBroker	0.0000	0.0213	Uso-VehTransporte Personal	-0.5190	0.2859
Canal-VentaBanca y Digital	-0.0923	0.2788	Uso-VehTurismo	-0.5698	0.0466
Canal-VentaInstituciones	-0.5771	-0.0148	Uso-VehEscolar	-0.5963	0.6057
Edad18-42	0.0740	0.0657	Uso-VehTaxi	-0.6426	0.4276
Edad43-52	0.0000	0.0000	Uso-VehInstrucción	-3.0233	0.0415
Edad53+	-0.0710	0.0962	ZonaA1	0.3943	0.6509
EdadSin-Edad	0.0108	0.2366	ZonaA2	0.0000	0.0000
Marca-VehL1	0.2404	-0.0516	ZonaA3	-0.2704	0.3174
Marca-VehL2	0.0879	-0.0326	ZonaA4	0.0804	0.1037
Marca-VehL3	0.0000	0.0000	ZonaA5	-0.6248	0.8208
Marca-VehL4	-0.0969	0.0594	ZonaD1	-0.3156	-0.2785
Marca-VehL5	-0.1261	0.1812	ZonaD2	-0.7704	0.4737
Marca-VehM1-M2	-0.5064	0.0299	ZonaD3	-0.9557	0.6803
Marca-VehM3	-1.2695	0.5495	ZonaD4	-1.1184	0.4885
Marca-VehM4	-1.6448	-0.0397	ZonaD5	-1.6368	0.5339

TABLA N° 4.15: Modelo Marginal de Frecuencia y Severidad - Asistencias

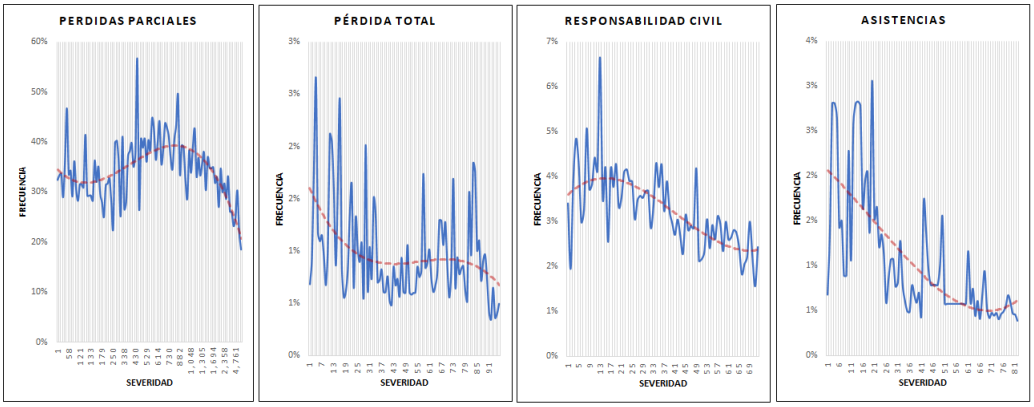
var-level	Frecuencia (β)	Severidad (α)	var-level	Frecuencia (β)	Severidad (α)
(Intercept)	-4.2874	4.6876	Marca-VehP3-P4	-0.6706	1.3200
Antiguedada. 0	-0.3732	0.6929	SAa.[1000-10000>	0.0000	0.0000
Antiguedadb. 1-2	0.0000	0.0000	SAb.[10000-18000>	-0.3577	0.2886
Antiguedadc. 3+	0.0508	-0.0947	SAc.[18000-99999]	-0.5868	0.5299
Canal-VentaAgentes-BancaDigital	0.1045	-0.4080	Uso-VehAlquiler-Escolar	0.6519	0.4431
Canal-VentaBroker	0.0000	0.0000	Uso-VehParticular	0.0000	0.0000
Canal-VentaExclusivos	-0.1341	-0.2050	Uso-VehSerenazgo	1.3647	0.9045
Canal-VentaInstituciones	-1.0202	0.0163	Uso-VehTaxi-Amb-Tur-Com	-0.1338	0.4297
Canal-VentaOtros	0.3212	-0.3016	Uso-VehTransPer-Carga-TransInt-TransUrb-Instr	-0.3862	0.1028
Marca-VehL1	0.0274	-0.2591	ZonaA1-A2	0.1095	-0.3183
Marca-VehL2	0.0000	0.0000	ZonaA3	0.0000	0.0000
Marca-VehL3-L4	-0.2537	0.0225	ZonaA4-A5	-0.1800	0.0915
Marca-VehM1-M2-M3	-1.0790	0.4842	ZonaD1-D4-D5	-0.8697	0.7099
Marca-VehP1	0.3766	-0.0327	ZonaD2-D3	0.2768	0.4620
Marca-VehP2-P5	-1.1444	2.0106			

FIGURA N° 4.3: Robustez de los modelos de GLM Marginales - Índice de GINI



En la Figura N° 4.4 se muestra un primer indicio de una forma de asociación entre la frecuencia y la severidad en valores brutos de los mismos.

FIGURA N° 4.4: Indicio de dependencia entre la Frecuencia y la Severidad por Cobertura



Es posible observar de la existencia de una relación u asociación no lineal entre la Frecuencia y la Severidad, por lo que para medir el grado de asociación será medido con la tau de kendal.

Otros gráficos que podemos utilizar para determinar la existencia de dependencia entre dos riesgos son el Chi-Plot y el K-Plot, para los cuales se definieron criterios de dependencia entre dos riesgos:

FIGURA N° 4.5: Chi-Plots de la Frecuencia y Severidad por cobertura

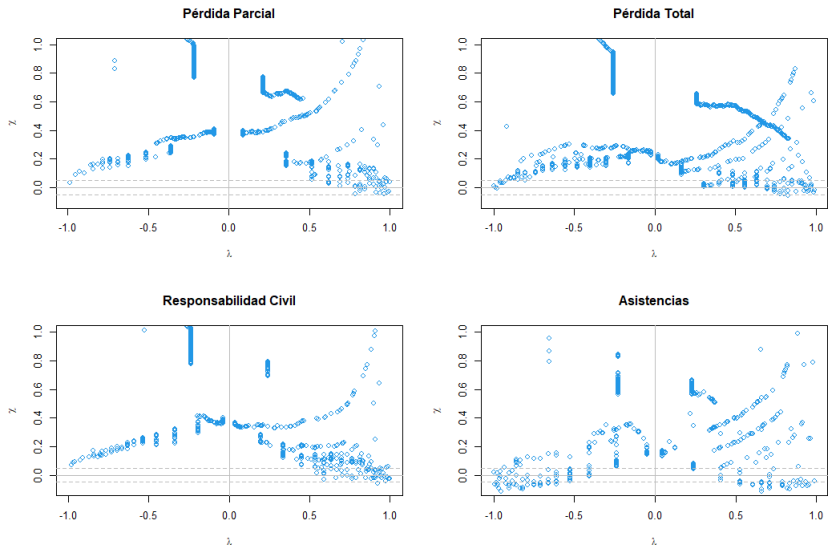
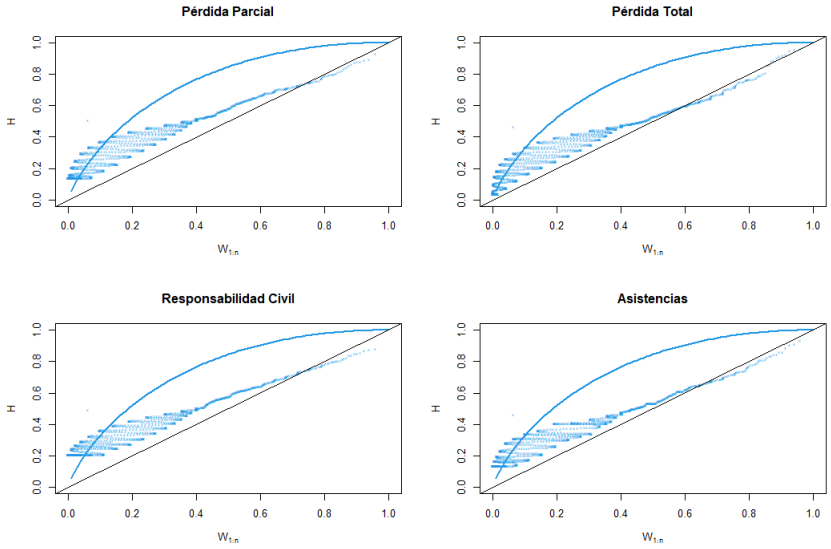


FIGURA N° 4.6: K-Plots de la Frecuencia y Severidad por cobertura



En la Figura N° 4.5 podemos observar que la mayoría de los puntos se encuentran fuera de los rangos al 95% de confianza. De acuerdo a (Genest y Boies, 2003) si tenemos puntos entrelazados, dentro y fuera de la región de confianza, tenemos una relación de dependencia compleja, lo que implica que utilizar relaciones del tipo lineal serán inservibles. En la Figura N° 4.6 el criterio era que si todos los puntos caían sobre la curva, entonces tendríamos un dependencia perfecta, y sobre la diagonal indicaría riesgos independientes, por lo que el mensaje de estos gráficos es similar, existe cierto nivel de dependencia entre los riesgos. El K-Plot nos da una información adicional, en el sentido que el grado de dependencia es menor en las colas de la distribución.

Con este apartado se concluye la existencia de cierto grado de dependencia entre la frecuencia y la severidad, en particular para el presente trabajo, el tipo de asociación es no lineal, que se hace complejo dado que varía a lo largo del cuerpo de la distribución.

4.2.3 Modelos GLM Conjuntos de la Frecuencia y de la Severidad

Para la construcción del modelo conjunto, utilizaremos como input los modelos GLM marginales de la Frecuencia y de la Severidad. Adicionalmente, calcularemos los valores iniciales de los parámetros para que estos sean optimizados mediante el método de *Log-likelihood Maximum Verosimil*. Entonces por cobertura tenemos:

TABLA N° 4.16: Pérdida Parcial | Log-likelihood de los modelos de regresión

Modelo	**Loglikelihood**	$\hat{\theta}$	τ	**PPR Base**
Cópula Gauss	-5,775,746	0.083700	0.053333	403
Cópula Clayton	-5,774,749	0.170904	0.078725	639
Cópula Gumbel	-5,772,251	1.010034	0.009934	509
Cópula Frank	-5,771,951	-0.285000	-0.031616	564

TABLA N° 4.17: Pérdida Total | Log-likelihood de los modelos de regresión

Modelo	**Loglikelihood**	$\hat{\theta}$	τ	**PPR Base**
Cópula Gauss	-188,829	0.08368066	0.05333513	92
Cópula Clayton	-196,067	10.48703	0.8398338	20
Cópula Gumbel	-188,835	1.00000	0.00000	95
Cópula Frank	-198,829	0.6069313	0.06692831	89

TABLA N° 4.18: Responsabilidad Civil | Log-likelihood de los modelos de regresión

Modelo	**Loglikelihood**	$\hat{\theta}$	τ	**PPR Base**
Cópula Gauss	-767,378	0.063070	0.040179	56
Cópula Clayton	-767,350	0.118702	0.056026	63
Cópula Gumbel	-786,157	14.216610	0.929660	7
Cópula Frank	-768,980	0.856516	0.094193	14

TABLA N° 4.19: Asistencias | Log-likelihood de los modelos de regresión

Modelo	**Loglikelihood**	$\hat{\theta}$	τ	**PPR Base**
Cópula Gauss	-288,939	0.07255326	0.04622946	7
Cópula Clayton	-289,311	0.02560843	0.01264234	6
Cópula Gumbel	-288,945	1.00012900	0.00012879	7
Cópula Frank	-288,942	0.36887960	0.04092350	7

En cada una de las tablas se muestran los valores de los *loglikelihood* por cobertura y por tipo de cópula utilizada. bajo el criterio de *loglikelihood* elegimos una familia de cópulas por cobertura. Para las Pérdidas Parciales el mayor *loglikelihood* es de -5,7751,951 por lo que elgimos la Cópula de Frank, para la cobertura Pérdida Total el mayor *loglikelihood* es de -188,829 por lo que elegimos la Cópula de Gauss, para la cobertura Responsabilidad Civil el mayor *loglikelihood* es de -767,350 por lo que elgimos la Cópula de Clayton y para

la cobertura de Asistencias el mayor *loglikelihood* es de -288,939 por lo que elegimos la Cópula de Gauss.

Los códigos en R que se utilizaron para calcular cada modelo de regresión se hallan en detalle en el Anexo C.

Por lo tanto, lo modelos de regresión conjunta de la Frecuencia y severidad por cada cobertura y con la familia de copula elegida son las siguientes:

TABLA N° 4.20: Modelo de Regresión GLM Conjunto - Pérdidas Parciales

Cópula FRANK | $\theta = 1.01$ | $Loglikelihood = -5,772,251$ | $\tau = 0.009934$					
var-level	**Frecuencia** (β)	**Severidad** (α)	**var-level**	**Frecuencia** (β)	**Severidad** (α)
(Intercept)	-1.5804	7.9157	Marca-VehL3	0.0000	0.0000
Antiguedada. 00	0.4223	-0.2577	Marca-VehL4	0.1943	-0.5419
Antiguedadb. 01	0.1898	0.2016	Marca-VehL5	0.3447	0.2673
Antiguedadc. 02	0.0000	0.0000	Marca-VehM1-M2-M3	-0.0339	0.2058
Antiguedadd. 03	-0.0424	0.1527	Marca-VehP1	0.4477	-0.6627
Antiguedade. 04	-0.0915	-0.0761	Marca-VehP2	0.5961	-0.1700
Antiguedadf. 05	-0.0821	-0.0566	Marca-VehP3	0.3735	0.5000
Antiguedadg. 06	-0.1114	-0.3866	Marca-VehP4	0.1105	0.0969
Antiguedadh. 07	-0.0047	-0.3385	Marca-VehP5	-0.0529	-0.0282
Antiguedadi. 08	-0.0267	-0.2672	SAa.[1000-2000>	-0.9033	-0.8563
Antiguedadj. 09	0.0031	-0.2118	SAb.[2000-3000>	-0.3307	-0.7701
Antiguedadk. 10	0.2576	-0.1106	SAc.[3000-4000>	-0.0984	-0.6664
Antiguedadl. 11-13	-0.0913	0.0072	SAd.[4000-5000>	0.0028	-0.5877
Antiguedadm. 14-19	-0.0318	-0.3697	SAe.[5000-6000>	0.3724	-0.7063
Antiguedadn. 20+	-0.5157	-0.4655	SAf.[6000-7000>	0.4289	-0.8169
Canal-VentaAgentes Exclusivos	0.1619	-0.1176	SAg.[7000-8000>	0.4228	-0.8143
Canal-VentaBancaDigital-Otros	-0.1518	0.2469	SAh.[8000-9000>	0.3204	-1.0248
Canal-VentaCorredores	0.0000	0.0000	SAi.[9000-13000>	0.0946	-0.5696
Canal-VentaGrandes Cuentas	0.0627	0.2721	SAj.[13000-999999]	0.0000	0.0000
Canal-VentaInstituciones	-0.6424	0.0105	Tipo-PolizaFlota	0.1055	0.1478
Edad18-27	0.2530	-0.1066	Tipo-PolizaIndividual	0.0000	0.0000
Edad28-32	0.1611	-0.0151	Uso-VehAlquiler	0.0173	-0.0529
Edad33-37	0.1174	0.0211	Uso-VehCarga-Taxi	-0.6184	0.3209
Edad38-42	0.0000	0.0000	Uso-VehComercial	-0.1140	-0.1903
Edad43-47	0.0664	-0.1118	Uso-VehOtros-Usos	0.5729	-0.4462
Edad48-52	0.0502	-0.2003	Uso-VehParticular	0.0000	0.0000
Edad53-57	0.0749	-0.2402	Uso-VehTransporte Personal	0.2981	-0.6437
Edad58-67	-0.0846	-0.1625	ZonaA1	0.4260	-0.0549
Edad68-72	0.1578	-0.3279	ZonaA2	0.0000	0.0000
Edad73 +	0.0937	-0.3081	ZonaA3	0.1675	-0.2783
EdadSin-Edad	-0.1640	0.3475	ZonaA4-A6	0.5853	-0.2157
GeneroFemenino	0.0000	0.0000	ZonaA5	0.4257	-0.5504
GeneroIndeterminado	0.3619	-0.0252	ZonaD1	-0.0066	-0.1416
GeneroMasculino	0.0583	0.0266	ZonaD2	-0.0783	-0.1510
Marca-VehL1	0.5418	-0.1716	ZonaD3	-0.0717	-0.0285
Marca-VehL2	0.3597	-0.1209	ZonaD4	-0.0424	0.7375

4.2.4 Agregación de la Prima Pura de Riesgo por Cobertura

A partir de esta subsección básicamente describiremos la propuesta metodológica para el cálculo de una única Prima Pura de Riesgo en función de los modelos conjuntos de Frecuencia y Severidad, incluyendo su grado de dependencia, calculados en los puntos anteriores.

TABLA N° 4.21: Modelo de Regresión GLM Conjunto - Pérdidas Totales

Cópula GAUSS | $\theta = 0.08368066$ | $Loglikelihood = -188,829$ | $\tau = 0.05333513$					
var-level	**Frecuencia** (β)	**Severidad** (α)	**var-level**	**Frecuencia** (β)	**Severidad** (α)
(Intercept)	-5.7924	10.2933	Marca-VehM3	0.7406	-1.8000
Edad18-27	0.7704	-0.2614	Marca-VehP1-P2-P3-P4	-0.7991	0.3155
Edad28-32	0.4645	-0.0641	Tipo-PolizaFlota	-0.4412	-0.2388
Edad33-42	0.1140	0.2567	Tipo-PolizaIndividual	0.0000	0.0000
Edad43-52	0.0000	0.0000	Uso-VehAlquiler	0.6932	-0.3058
Edad53-77	-0.2211	-0.0339	Uso-VehAmbul-TranspIntProv-Carga-Tur-Esc	0.6157	-0.0314
Edad78 +	-0.1549	-1.4905	Uso-VehComercial-Taxi-TransPers	0.3858	-0.6809
EdadSin-Edad	-0.0139	1.0199	Uso-VehOtros-Usos	0.8861	-1.8950
GeneroFemenino	0.0000	0.0000	Uso-VehParticular	0.0000	0.0000
GeneroMasculino	0.1247	0.1836	ZonaA1	1.8261	-1.1696
Marca-VehL1	2.9547	-1.7092	ZonaA2-A5	0.5650	-0.6182
Marca-VehL2	2.1613	-0.7824	ZonaA3	0.0000	0.0000
Marca-VehL3	0.4244	-0.0632	ZonaA4	0.7047	-1.2507
Marca-VehL4	0.0000	0.0000	ZonaD1	0.4840	-0.7387
Marca-VehM1	2.8090	-1.5017	ZonaD2-D3	0.0136	-0.5893
Marca-VehM2	1.7898	-2.0125	ZonaD4	0.4061	-1.1410

TABLA N° 4.22: Modelo de Regresión GLM Conjunto - Responsabilidad Civil

Cópula CLAYTON | $\theta = 0.1187016$ | $Loglikelihood = -767,350$ | $\tau = 0.05602563$					
var-level	**Frecuencia** (β)	**Severidad** (α)	**var-level**	**Frecuencia** (β)	**Severidad** (α)
(Intercept)	-4.2504	8.3939	Marca-VehP1-P2	0.1291	1.0761
Antiguedada. 0-1	0.3360	-0.5056	Marca-VehP3	-0.3123	0.3273
Antiguedadb. 2-9	0.0000	0.0000	Uso-VehSerenazgo	-0.0826	0.0804
Antiguedadc. 10+	0.0148	-0.7176	Uso-VehTransInterProv-Urb	1.7016	-1.4727
Canal-VentaOtros	-0.0591	-0.1849	Uso-VehParticular	3.1605	-2.5760
Canal-VentaGrandesCuentas-AgentesExc	0.0000	0.0000	Uso-VehAlq-Amb-Carg-Com	0.0000	0.0000
Canal-VentaCorredores	0.0302	0.1189	Uso-VehTransporte Personal	1.3405	-0.2897
Canal-VentaBanca y Digital	-0.2584	-0.0151	Uso-VehTurismo	-0.2753	-0.7953
Canal-VentaInstituciones	0.4135	-0.4016	Uso-VehEscolar	0.6063	0.2579
Edad18-42	-0.0519	0.3687	Uso-VehTaxi	0.4525	-0.5811
Edad43-52	0.0000	0.0000	Uso-VehInstrucción	1.2694	-1.4410
Edad53+	-0.1297	0.1406	ZonaA1	3.2422	-0.5422
EdadSin-Edad	-0.1315	0.4803	ZonaA2	0.0000	0.0000
Marca-VehL1	0.4185	-0.9047	ZonaA3	-0.1559	-0.4263
Marca-VehL2	0.2019	-0.7040	ZonaA4	0.3797	-0.8058
Marca-VehL3	0.0000	0.0000	ZonaA5	0.9758	-0.5989
Marca-VehL4	-0.0762	-0.2283	ZonaD1	1.4755	-1.5941
Marca-VehL5	-0.0950	-0.2114	ZonaD2	-0.2116	-0.7341
Marca-VehM1-M2	0.9316	-1.0202	ZonaD3	-0.4034	-0.5239
Marca-VehM3	-0.7572	-0.7411	ZonaD4	-0.6233	-0.6699
Marca-VehM4	-1.3062	-1.1169	ZonaD5	0.1940	-0.8339

TABLA N° 4.23: Modelo de Regresión GLM Conjunto - Asistencias

Cópula GAUSS | $\theta = 0.07255326$ | $Loglikelihood = -288,939$ | $\tau = 0.04622946$					
var-level	**Frecuencia** (β)	**Severidad** (α)	**var-level**	**Frecuencia** (β)	**Severidad** (α)
(Intercept)	-4.1665	6.1156	Marca-VehP3-P4	-1.7171	1.5925
Antiguedada. 0	0.3100	-0.2158	SAa.[1000-10000>	0.0000	0.0000
Antiguedadb. 1-2	0.0000	0.0000	SAb.[10000-18000>	-0.4015	0.0615
Antiguedadc. 3+	-0.0725	0.1895	SAc.[18000-99999]	-0.6796	0.3678
Canal-VentaAgentesExc-BancaDigital	0.0206	-0.2522	Uso-VehAlquiler-Escolar	0.9677	-0.0237
Canal-VentaCorredores	0.0000	0.0000	Uso-VehParticular	0.0000	0.0000
Canal-VentaGrandes Cuentas	-0.1012	-0.4350	Uso-VehSerenazgo	2.2115	0.6802
Canal-VentaInstituciones	-0.2116	-0.8539	Uso-VehTaxi-Amb-Tur-Com	0.3666	-0.4822
Canal-VentaOtros	0.7192	-0.2436	Uso-VehTransPer-Carga-TransInt-TransUrb-Instr	1.1631	-0.9644
Marca-VehL1	0.0646	-0.0913	ZonaA1-A2	0.1550	-0.1266
Marca-VehL2	0.0000	0.0000	ZonaA3	0.0000	0.0000
Marca-VehL3-L4	-0.1085	-0.2256	ZonaA4-A5	0.2712	-1.1097
Marca-VehM1-M2-M3	-0.4148	-0.6801	ZonaD1-D4-D5	-0.7206	0.4472
Marca-VehP1	0.7984	-0.2438	ZonaD2-D3	0.8993	-0.1378
Marca-VehP2-P5	-1.3206	1.4294			

Nuestro primer paso será calcular la Prima Pura de Riesgo por vehículo asegurado utilizando la fórmula que describimos en el teorema 10 de la subsección 2.2.4.3.

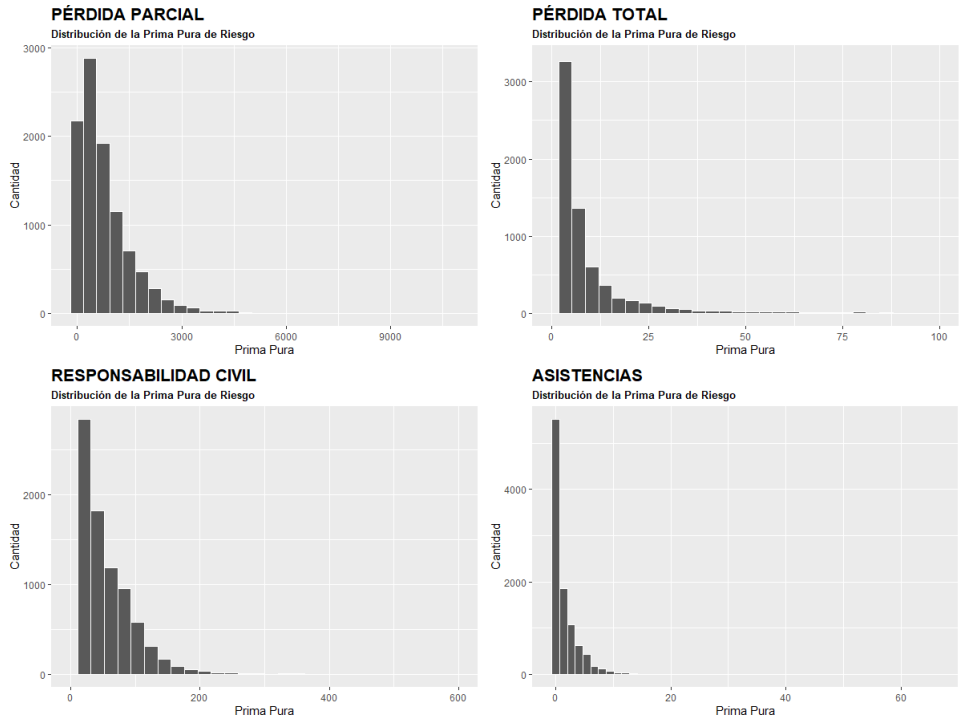
$$f_L(l|\mu,\delta,\lambda,\theta) = \sum_{y=1}^{\infty}\left[D_1\left(F_X\left(\frac{l}{y}|\mu,\delta\right),F_Y(y|\lambda)|\theta\right) - D_1\left(F_X\left(\frac{l}{y}|\mu,\delta\right),F_Y(y-1|\lambda)|\theta\right)\right].\frac{1}{y}f_X\left(\frac{l}{y}|\mu,\delta\right)$$

como se puede observar, la función de pérdida te pide las distribuciones marginales F_X y F_Y, además de los parámetros $\mu,\delta,\lambda,\theta$, para luego aplicar la siguiente integral que nos dará el valor esperado de la pérdida o la Prima Pura de Riesgo:

$$PPR = E(L) = \int_0^{\infty} l \times f_L(l|\mu,\delta,\lambda,\theta)dl$$

Debido a la alta capacidad computacional requerida, tomamos una muestra de 10,000 vehículos asegurados de nuestra base de datos para visualizar la distribución de la Prima Pura de Riesgo por cobertura, por lo que obtenemos:

FIGURA N° 4.7: Distribución de la Prima Pura de Riesgo



Dado que hemos tomado una muestra, necesitaremos ajustar cada distribución a una distribución de probabilidad conocida como la Gamma, para ello estimaremos los parámetros para cada una mediante Máxima Verosimilitud. Hasta este punto ya tenemos las primas pura de riesgo por cobertura, sin embargo, como hemos indicado en nuestras hipótesis, en la ocurrencia del siniestro suelen activarse más de una cobertura por lo que suele existir cierto grado de dependencia entre las pérdidas por cobertura, que son cubiertas por el pago de una sola prima. En ese sentido nuestro enfoque es el siguiente:

FIGURA N° 4.8: Ajuste a una distribución Gamma - Estimación de parámetros

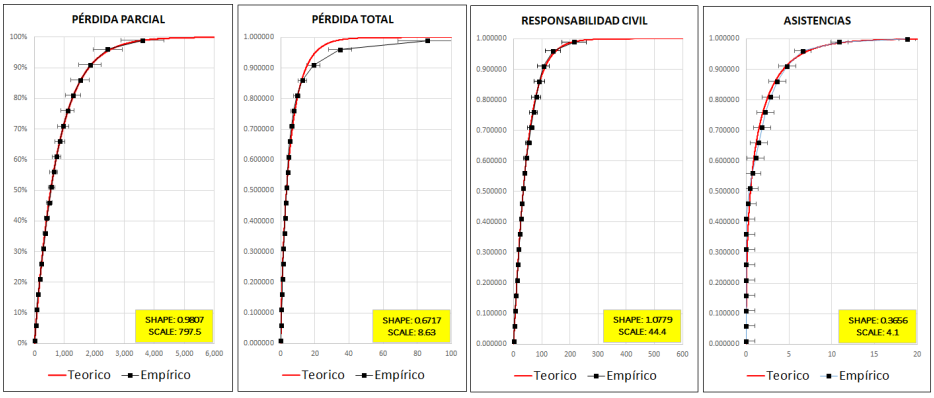
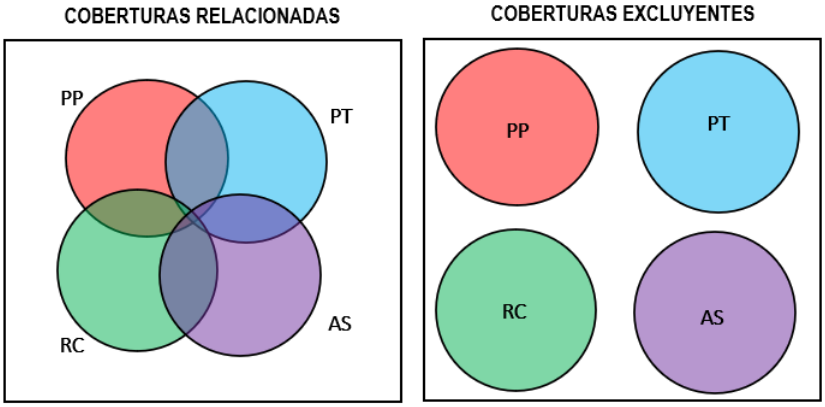


FIGURA N° 4.9: Formas de agregación de las primas por cobertura



El cuadro de la derecha de la Figura N° 4.9 indica que las coberturas son excluyentes entre sí, por lo que la prima final agregada será la suma simple de cada una. Este supuesto práctico es generalmente asumido en el ejercicio real de las aseguradoras.

Parte de nuestra propuesta metodológica es asumir cierto grado de relación entre las primas de cada cobertura, como el cuadro de la izquierda de la Figura N° 4.9, siendo no neceriamente la estructura relacional la que se muestra, es solo referencial. En ese sentido, la agregación de las primas de riesgo por cobertura no se limita a una suma simple, sino a una agregación de riesgos bivariados y dependientes.

El esquema que utlizaremos para agregar los riesgos por cobertura será el siguiente:

Como tenemos cuatro coberturas por agregar, existen seis formas de parejas bivariadas que se pueden formar, como debemos tomar solo una de ellas, nuestro criterio será elegir aquella pareja cuya grado de concordancia sea el mayor mediante el τ de Kendall. Los grados de asociación fueron los siguientes:

De acuerdo a los grados de asociación obtenidos iniciaremos la agrupación bivariada, bajo el criterio de mayor asociación, la primera pareja serán las coberturas de Pérdida Parcial y

FIGURA N° 4.10: Esquema de agregación de riesgos por cobertura

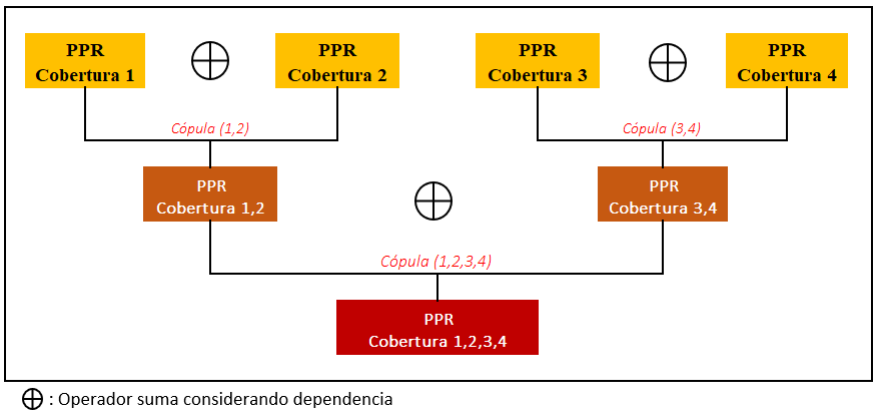


TABLA N° 4.24: Grados de dependencia bivariado - Tau de Kendall

N°	Pareja Bivariada	τ	Pareja Bivariada	τ
1	PP y PT	56.75%	RC y AS	17.76%
2	PP y RC	26.29%	PT y AS	17.71%
3	PP y AS	14.67%	PT y RC	32.03%

Pérdida Total, y la segunda pareja serán las coberturas de Responsabilidad Civil y Asistencia.

Entonces para agregar la prima pura de riesgo de la Pérdida Parcial y la Pérdida Total, necesitamos elegir una familia de cópulas que mejor refleje su asociación, bajo el criterio de mayor *loglikelihood* obtuvimos:

- PP y PT: elegimos la cópula de Clayton con parámetro $\theta = 0.714$ y $loglik = 1582$.
- RC y AS: elegimos la cópula de Clayton con parámetro $\theta = 0.504$ y $loglik = 424$

el detalle de cálculo y códigos en R utilizados se encuentran en el Anexo C.

El siguiente proceso es estimar la prima pura de riesgo bivariada de cada pareja de coberturas, este proceso se realizará mediante simulación montecarlo utilizando la propiedad descrita en la subsección 2.3.1.1., es decir:

- Simulamos q y u aleatoriamente entre 0 y 1.
- Calculamos v según lo indicado en la Tabla 2.9 de la subsección 2.3.1.1.

$$v = \left(1 + u^{-\theta}(q^{\frac{-\theta}{\theta+1}} - 1)\right)^{\frac{-1}{\theta}}$$

- Entonces dado que tenemos los vectores u y v procedemos a calcular la prima pura de riesgo agregada para la Pérdida Parcial y Pérdida Total de la siguiente forma:

$$PPR_{PP+PT} = Gamma^{-1}(u|shape = 0.098, scale = 797.5) + Gamma^{-1}(v|shape = 0.672, scale = 8.63)$$

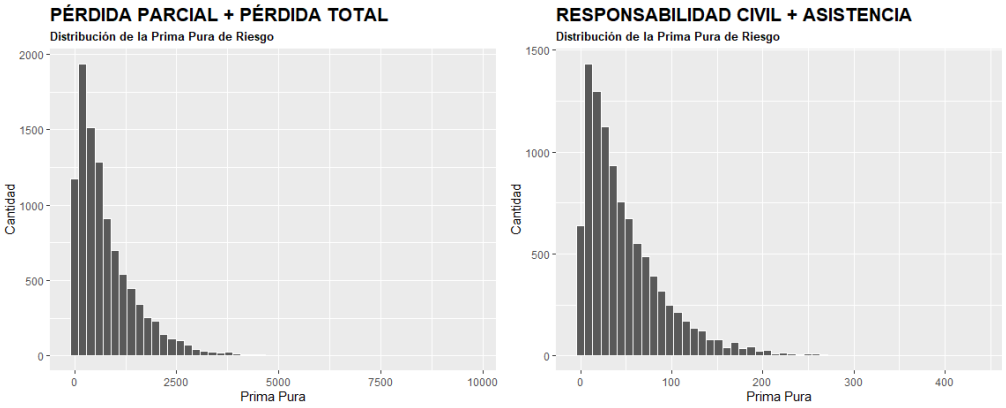
- Para las coberturas Responsabilidad Civil y Asistencias realizamos el mismo procedimiento, y como la cópula clayton fue la de mayor verosimilitud la forma de hallar v es la misma, entonces la prima pura de riesgo agregada es:

$$PPR_{RC+AS} = Gamma^{-1}(u|shape = 1.077, scale = 44.45) + Gamma^{-1}(v|shape = 0.366, scale = 4.13)$$

Observar que los parámetros de $shape$ y $scale$ son los estimados e indicados en la Figura N° 4.8.

Entonces la distribución de la prima pura de riesgo de cada par de coberturas se muestran en la siguiente figura:

FIGURA N° 4.11: Distribución de la Prima de Riesgo Bivariada



Finalmente, procederemos a agregar las primas bivariadas de cada distribución considerando la relación de dependencia que pueden tener, para ello calcularemos la familia cópula que mejor se ajuste de acuerdo al criterio del $loglink$, en este caso la familia cópula elegida fue la Gumbel con parámetro $\theta = 1.005$, $loglink = 1.0163$ y un grado de dependencia de $\tau = 0.53\,\%$. El cálculo se detalla en código R en el Anexo C.

Agregamos las primas de riesgo bajo dependencia siguiendo los siguientes pasos:

- Simulamos q y u aleatoriamente entre 0 y 1.
- Calculamos v según lo indicado en la Tabla 2.9 de la Subsección 2.3.1.1. para la familia Gumbel

$$v = \exp\left(-\left((-\ln qu)^{\theta} - (-\ln u)^{\theta}\right)^{\frac{1}{\theta}}\right)$$

- Entonces dado que tenemos los vectores u y v procedemos a calcular la prima pura de riesgo agregada de la siguiente manera:

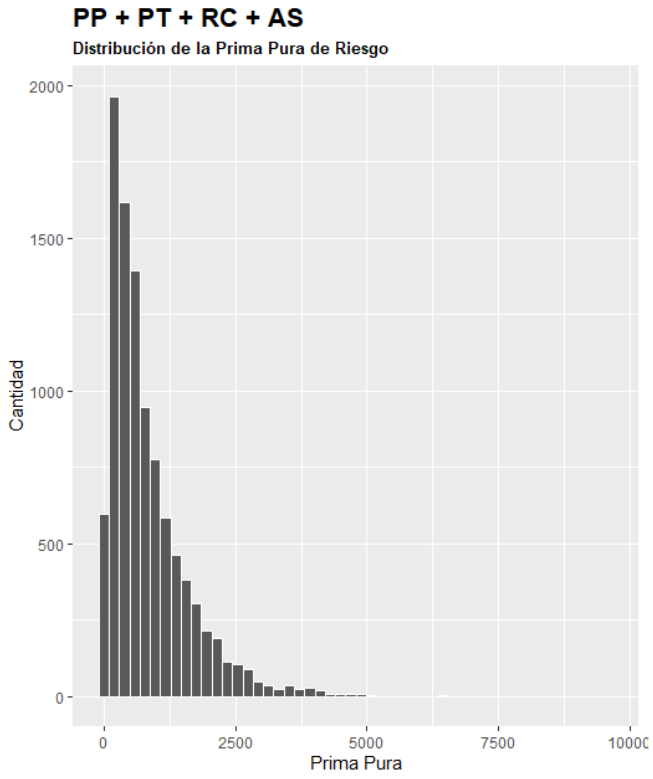
$$PPR_{PP+PT+RC+AS} = F^{-1}_{PP+PT}(u) + F^{-1}_{RC+AS}(v)$$

donde $F(PP+PT)$ y $F(RC+AS)$ son las distribuciones de probabilidad acumu-

ladas de la Figura N° 4.10.

La distribución de probabilidad de la prima pura de riesgo conjunta de las 4 coberturas consideradas en este trabajo es la siguiente:

FIGURA N° 4.12: Distribución de la Prima de Riesgo Multivariada Pérdida Parcial + Pérdida Total + Responsabilidad Civil + Asistencias



4.3 Interpretación y discusión de los resultados

El presente trabajo ha desarrollado una mixtura entre aspectos que se desarrollan en la práctica diaria del *pricing* de los departamentos actuariales, también se utilizaron tópicos de estudios realizados por los diversos autores mencionados en este trabajo y la propuesta metodológica de llegar a la prima de riesgo multicobertura considerando la dependencia entre riesgos.

Como hemos visto en el desarrollo de este trabajo, los modelos de regresión generalizados marginales de la frecuencia y de la severidad son la base del desarrollo del modelo conjunto, es por ello qué en términos de **Robustez**, estos modelos tienen coeficientes altamente siginificativos, como se muestran en el Anexo C, además, de acuerdo a los resultados del índice de GINI se demostró que los riesgos que se están analizando no todos son buenos o malos, sino que es posible segmentarlos. Por lo tanto, en este punto los resultados son acorde a la práctica actuarial en nuestra localidad.

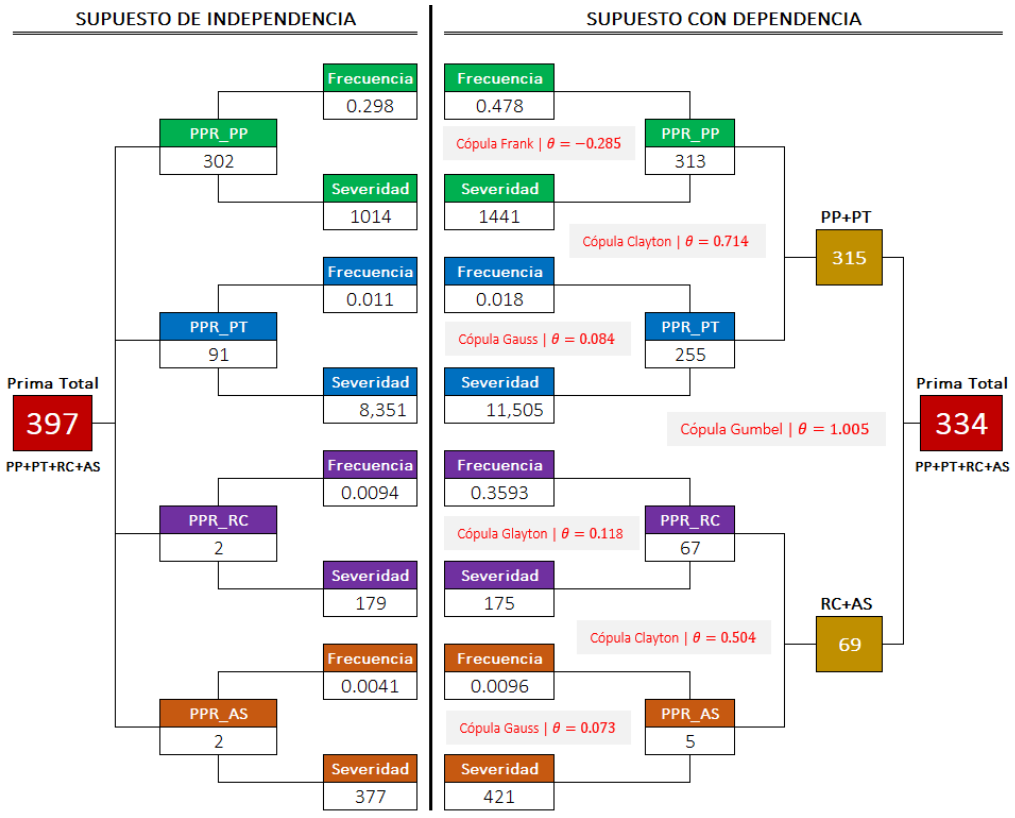
De los aspectos considerados, el más importante ha sido el de utilizar familias de cópulas, en particular las propiedades de la cópula-mixta, para la construcción de modelos conjuntos de probabilidad y poder medir la dependecia entre los riesgos. Los resultados obtenidos en la Tabla N° 4.20 son concordantes con los resultados obtenidos por Kholifah et al. (2019) y por Krämer et al. (2013) en sus respectivos trabajos. Un punto interesante fueron los resultados de los indicios de dependencia obtenidos por el Chi-Plot y el K-plot, ya que diversos trabajos muestran resultados bastante suavizados, sin embargo, del hecho de que los datos provienen de un comportamiento real, los gráficos Chi-Plot y K-Plot mostraron indicios de dependencia con ciertos patrones complejos de dependencia, para entender mejor esto sugiero revisar directamente el trabajo realizado por Fisher y Switzer (2001) ya que se explican diversas casuísticas en los resultados, y no de otros autores que toman su trabajo y se limitan a los casos simples. En este sentido, los resultados de **dependencia** han sido concordantes con la literatura revisada.

Referente a la propuesta metodologica de agregación de las primas de riesgo de cada cobertura, considerando la dependencia entre las mismas, parte del supuesto que la agregación usual y práctica que se realiza en la industria, ésta sobrestima la prima pura de riesgo, según el enfoque mostrado en la Figura N° 4.9. Para comprobar ello he realizado un cotizador que calcule la prima pura de riesgo de acuerdo a las características que pudiera tener un asegurado y su vehículo, y nos pemrite comparar las primas puras de riesgo bajo el supuesto de independencia y dependencia, considerado todos los parámetros estimados mostrados en la sección de análisis.

Una comparación se muestra en la Figura N° 4.13 de acuerdo a las siguientes variables:

Tipo Póliza	Canal	Edad	Sexo	Tipo vehículo	Marca Auto	Antiguedad	Valor Vehículo	Uso vehículo	Zona
Individual	Bróker	25	Femenino	Liviano	Toyota	0	20,000	Particular	A3

FIGURA N° 4.13: Prima Pura de Riesgo: Independencia vs. Dependencia



CONCLUSIONES

Se ha desarrollado una propuesta metodológica alternativa que incluyeron aspectos teóricos mostrados en las bases teóricas e implementados en la práctica mediante un cotizador. La aplicación sobre una base de datos con características reales demostraron que las hipótesis planteadas se cumplen. Finalmente luego de analizar los resultados obtenidos en el presente trabajo, se presentan las siguientes conclusiones:

1. Mediante las pruebas gráficas se pudo demostrar que existe un grado de asociación en la Frecuencia y la Severidad. Se observó que patrón dependencia no es lineal, al contrario, tiene patrones complejos de dependencia a lo largo del cuerpo de la distribución condicional.

2. La construcción de un modelo conjunto de la Frecuencia y de la Severidad fue posible utilizando propiedades de cópula mixta. Los coeficientes de regresión conjunta estimados son razonables, no soy muy distintos al modelo marginal estimado, se entiende que las diferencias halladas se explican por efecto de la dependencia considerado.

3. Se pudo medir el grado de dependencia bajo el concepto de concordancia entre variables aleatorias, principalmente utilizando la Tau de Kendall.

4. La propuesta metodológica de agregación bivariada de las primas puras de riesgo por cobertura, para llegar a la prima total, nos pudo demostrar la hipótesis de sobre estimación. Por la tanto, esta propuesta calcula una prima más realista acorde a las características de riesgo del asegurado.

RECOMENDACIONES

La metodología desarrollada en cierto sentido estima una prima pura de riesgo más compacta, sin embargo, en el transcurso se hallaron ciertas limitaciones, por lo que el objetivo de las siguientes recomendaciones es mejorar o complementar el trabajo realizado. Por lo tanto, las recomendaciones son las siguientes:

1. Una vez estimado los parámetros de las regresiones conjuntas, la simulación y/o predicción de valores ajustados al modelo suele requerir un alto performance a nivel computacional. Si bien la información disponible suele estar cercano al millón de datos, hubo en algunos puntos del trabajo donde se utilizó un muestra representativa para realizar algunas estimaciones, como por ejemplo el cálculo de los errores estándar de los coeficientes del modelo al momento de estimar su matriz Hessiana.

2. Un interesante complemento al trabajo es hallar la función de densidad conjunta para 4 riesgos (referente a la cantidad de coberturas) utilizando las propiedades de las Vine Cópulas, si bien fueron mencionadas en las bases teóricas, el presente trabajó se basó en otra alternativa también mostrada.

3. Una extensión al presente trabajo de tesis es calcular el indice de siniestralidad o *Loss Ratio* con el modelo multivariado conjunto estimado, y poder medir el impacto sobre la siniestralidad real, lo que se reflejaría en un aumento en el margen de utilidad. Esto no pudo ser posible, debido a que la información pública incluye datos de diferentes empresas de seguro y el valor de la prima comercial incluía aspectos de impuestos que no se pudieron separar en el proceso para poder estimar la prima devengada por periodo.

REFERENCIAS BIBLIOGRÁFICAS

Aas, K., Czado, C., Frigessi, A., y Bakken, H. (2009). Pair-copula constructions of multiple dependence. *Insurance: Mathematics and economics*, *44*(2), 182–198.

Anderson, D., Feldblum, S., Modlin, C., Schirmacher, D., Schirmacher, E., y Thandi, N. (2007). *A practitioner's guide to generalized linear models—a foundation for theory, interpretation and application.* Towers Watson.

Anscombe, F. J. (1973). Graphs in statistical analysis. *The american statistician*, *27*(1), 17–21.

Arbenz, P., Hummel, C., y Mainik, G. (2012). Copula based hierarchical risk aggregation through sample reordering. *Insurance: Mathematics and Economics*, *51*(1), 122–133.

Brockman, M. J., y Wright, T. S. (1992). Statistical motor rating: making efficient use of your data. *J. Inst. Actuar.*, *119*, 457–543.

Burden, R., Faires, D., y Burden, A. (2015). *Numerical analysis* (10a edición ed.). Cengage Learning.

Czado, C., Kastenmeier, R., Brechmann, E. C., y Min, A. (2012). A mixed copula model for insurance claims and claim sizes. *Scandinavian Actuarial Journal*, *4*, 278-305.

De Jong, P., Heller, G. Z., y cols. (2008). Generalized linear models for insurance data. *Cambridge Books*.

de Leon, A. R., y Wu, B. (2011). Copula-based regression models for a bivariate mixed discrete and continuous outcome. *Statistics in Medicine*, *30*(2), 175–185.

Embrechts, P. (1999). Correlation: pitfalls and alternatives. *Risk Magazine*, 69–71.

Fisher, N., y Switzer, P. (2001). Graphical assessment of dependence: Is a picture worth 100 tests? *The American Statistician*, *55*(3), 233–239.

Frees, E. W., Lee, G., y Yang, L. (2016). Multivariate frequency-severity regression models in insurance. *Risk*, *4*(4), 1–36.

Garrido, J., Genest, C., y Schulz, J. (2016). Generalized linear models for dependent frequency and severity of insurance claims. *Insurance: Mathematics and Economics*, *70*, 205–215.

Genest, C., y Boies, J.-C. (2003). Detecting dependence with kendall plots. *The American Statistician*, *57*(4), 275–284.

Genest, C., y Neslehova, J. (2007). A primer on copulas for count data. *The Astin Bulletin*, *37*, 475–515.

Joe, H. (2014). *Dependence modeling with copulas*. CRC press.

Kaas, R., Goovaerts, M., Dhaene, J., y Denuit, M. (2008). *Modern actuarial risk theory using r*. Springer Berlin, Heidelberg.

Kholifah, A. R., Lestari, D., y Devila, S. (2019). Premium calculation using marginal generalized linear model combined with copula. En (Vol. 2168). doi: 10.1063/1 .5132462

Krämer, N., Brechmann, E. C., Silvestrini, D., y Czado, C. (2013). Total loss estimation using copula-based regression models. *Insurance: Mathematics and Economics*, *53*(3), 829–839.

Lee, W., Park, S., y Ahn, J. Y. (2018). Investigating dependence between frequency and severity via simple generalized linear models. *Journal of the Korean Statistical Society*.

Lee, W., Park, S. C., y Ahn, J. Y. (2019). Investigating dependence between frequency and severity via simple generalized linear models. *Journal of the Korean Statistical Society*, *48*(1), 13–28.

Lundberg, F. (1903). *I. approximerad framställning af sannolikhetsfunktionen. ii. aterförsäkring af kollektivrisker*. Almqvist Wiksells.

Matos, J., y cols. (2007). Uncertainty treatment in civil engineering numerical models.

McCullagh, P., y Nelder, J. A. (1989). *Generalized linear models*. Chapman & Hall.

McNeil, A., Frey, R., y Embrechts, P. (2005). *Quantitative risk management: concepts, techniques and tools*. Princeton University Press.

Nelsen, R. B. (2006). *An introduction to copulas*. Springer.

Ohlsson, E., y Johansson, B. (2010). *Non-life insurance pricing with generalized linear models*. Springer.

Rao, C. R., Shalabh, Toutenburg, H., y Heumann, C. (2010). *Linear models and generalizations*. Springer Berlin, Heidelberg.

Renshaw, A. E. (1994). Modelling the claims process in the presence of covariates. *ASTIN Bulletin: the Journal of the IAA*, *24*(2), 265–285.

Scarsini, M. (1984). On measures of concordance. *Stochastica*, *8*(3), 201–218.

Schepsmeier, U., y Stöber, J. (2014). Derivatives and fisher information of bivariate copulas. *Statistical papers*, *55*(2), 525–542.

Sikov, A., y Cerda-Hernández, J. (2021). *Teoría de riesgo para seguros*. EDUNI.

Sklar, M. (1959). Fonctions de répartition à n dimensions et leurs marges. En *Annales de l'isup* (Vol. 8, pp. 229–231).

Song, P., Li, M., y Yuan, Y. (2009). Joint regression analysis of correlated data using gaussian copulas. *Biometrics*, *65*(1), 60–68.

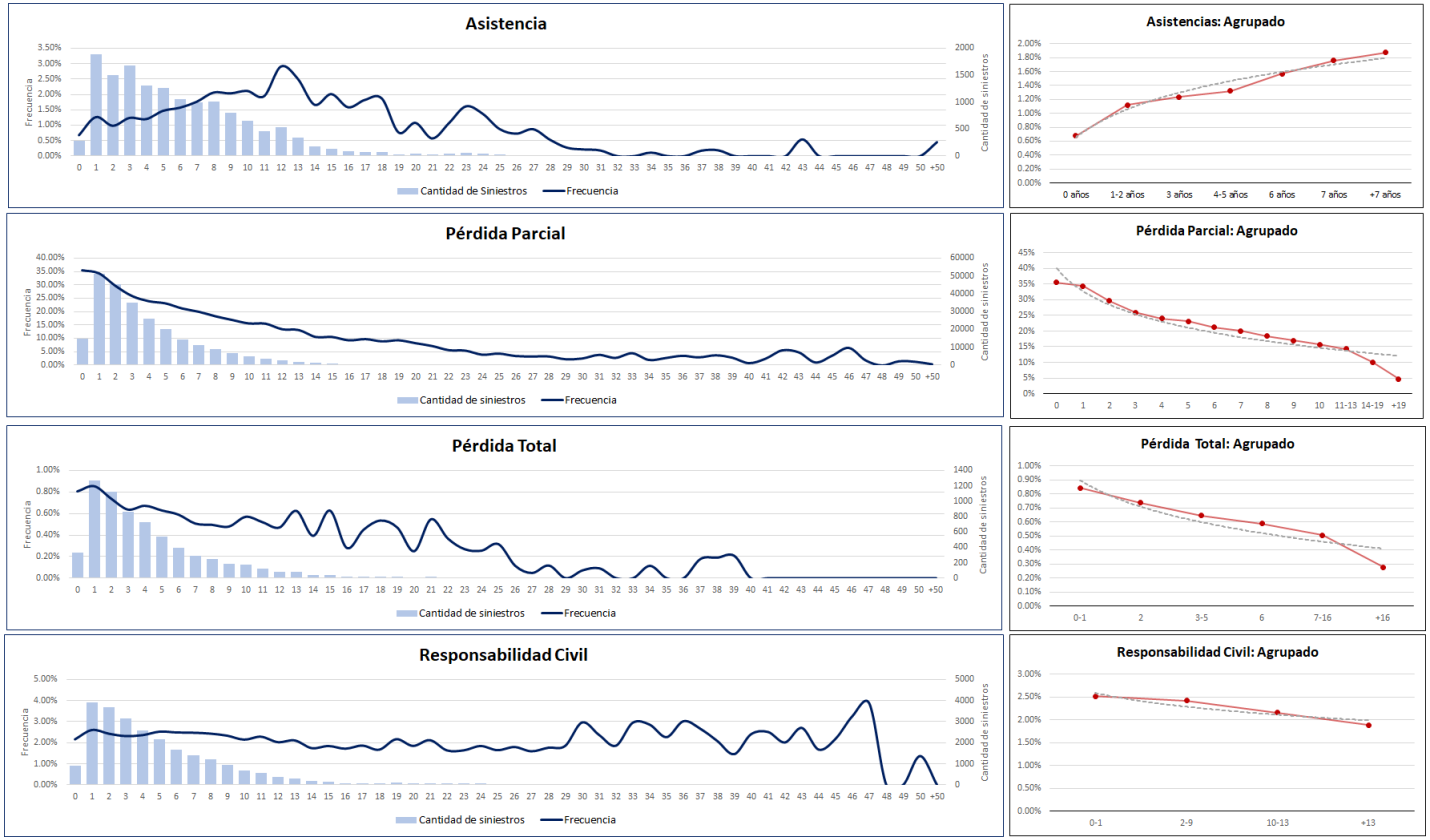
Song, P. X.-K. (2000). Multivariate dispersion models generated from gaussian copula. *Scandinavian Journal of Statistics*, *27*(2), 305–320.

ANEXOS

ANEXO A: ANÁLISIS UNIVARIADO

Variable: Antiguedad del Vehículo

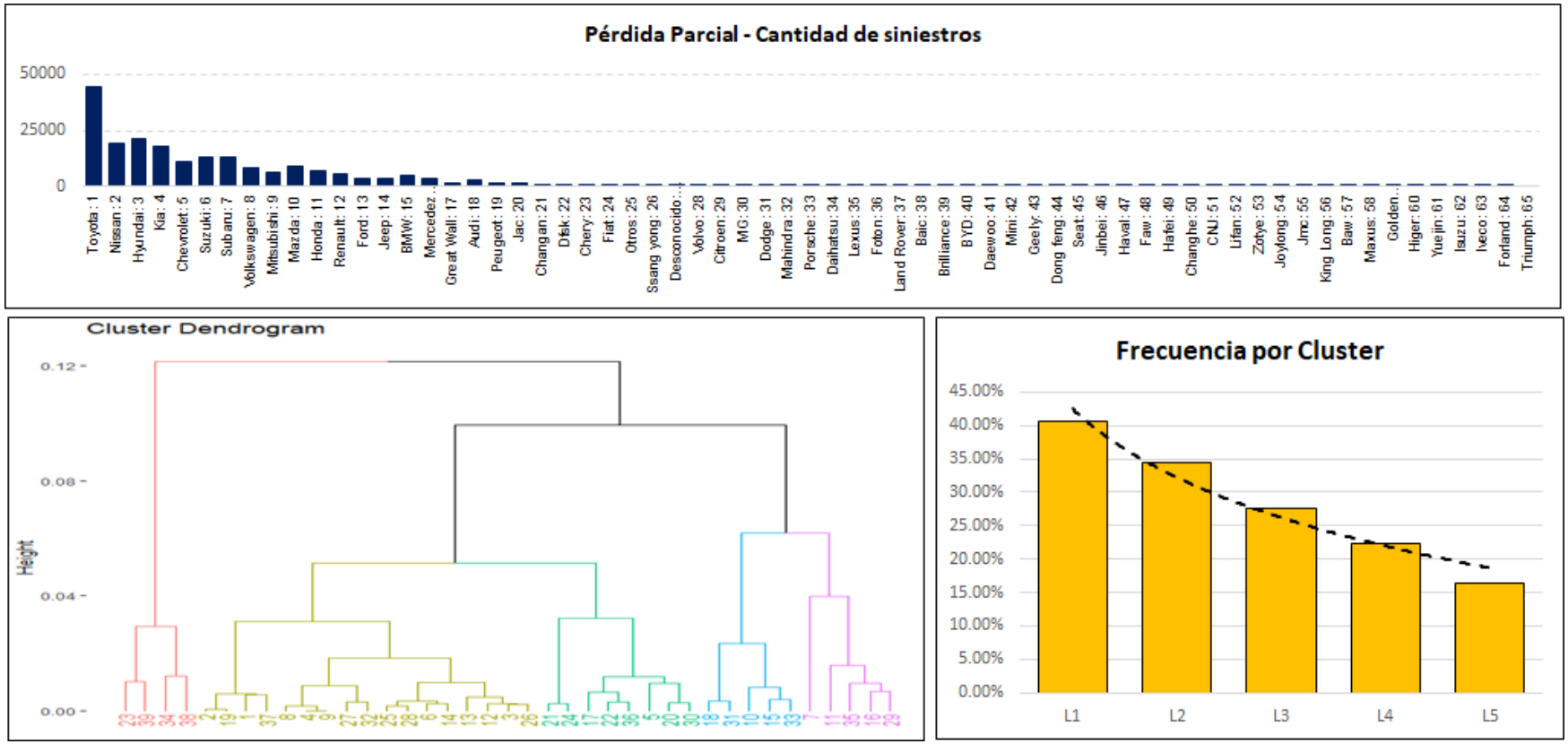
Figure A .1: Agrupación de la variable Antiguedad del vehículo



Variable: Marca de Vehículo

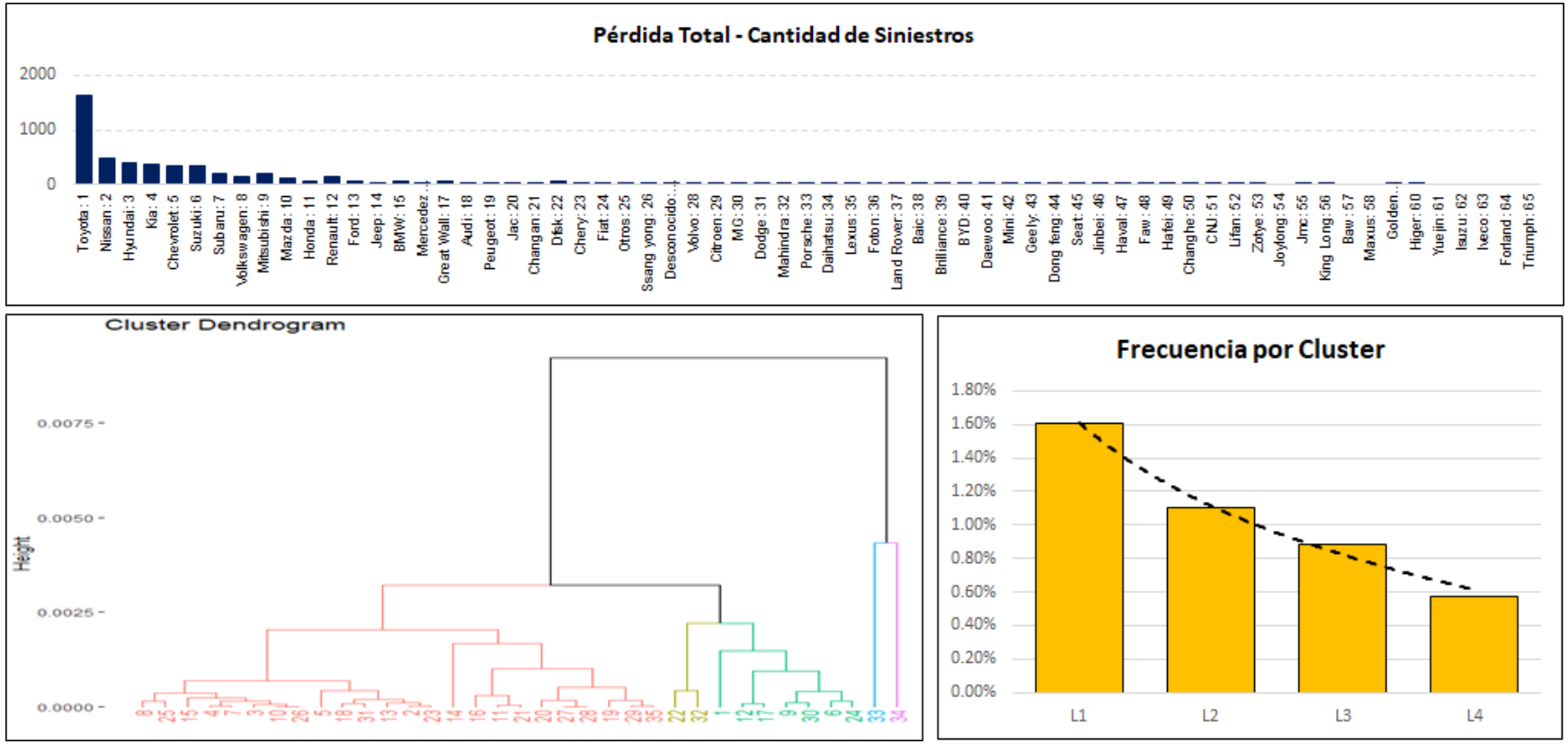
VEHÍCULOS LIVIANOS - PÉRDIDAS PARCIALES

Figure A .2: Agrupación de la Marca Vehículo - Pérdida Parcial - Livianos



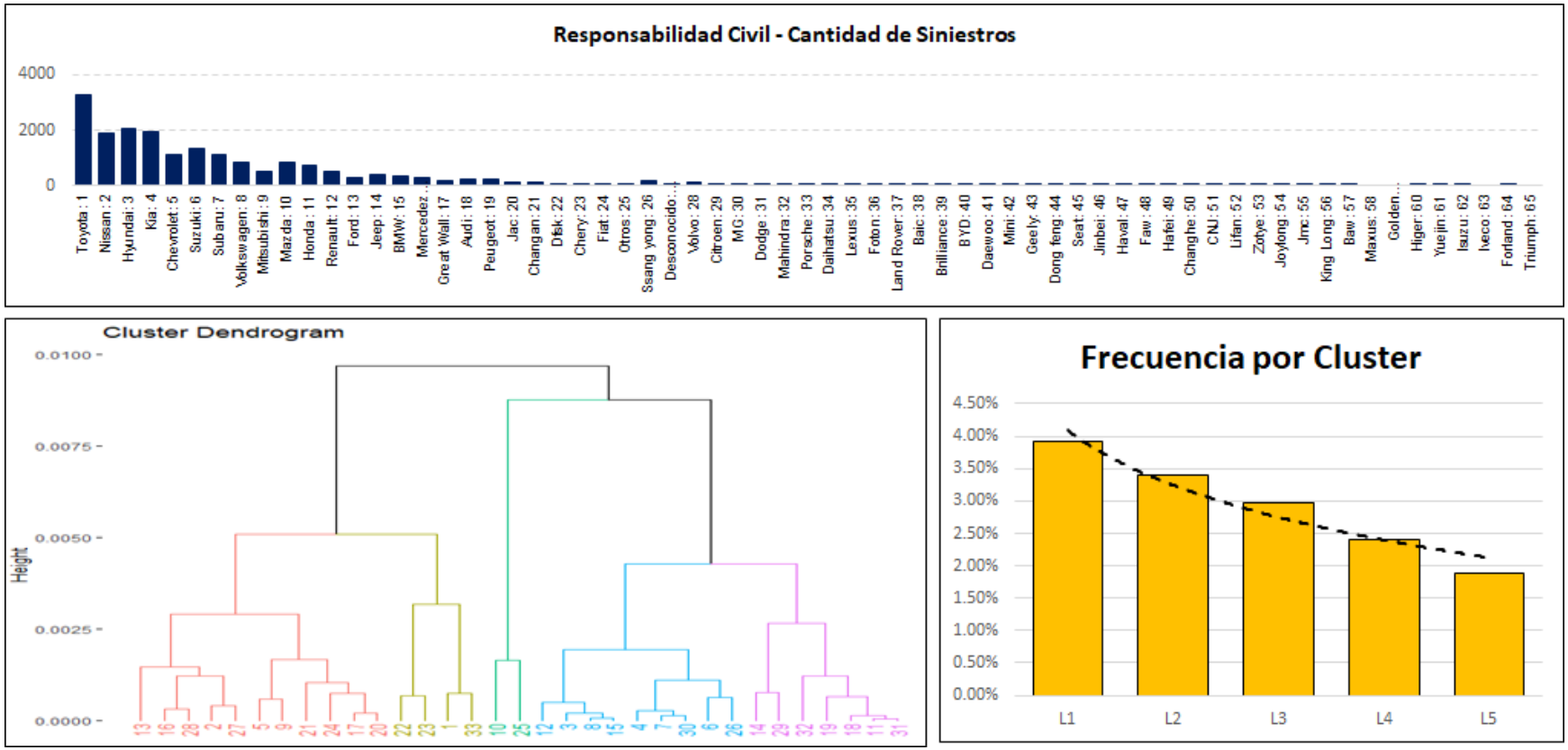
VEHÍCULOS LIVIANOS - PÉRDIDAS TOTALES

Figure A .3: Agrupación de la Marca Vehículo - Pérdida Total - Livianos



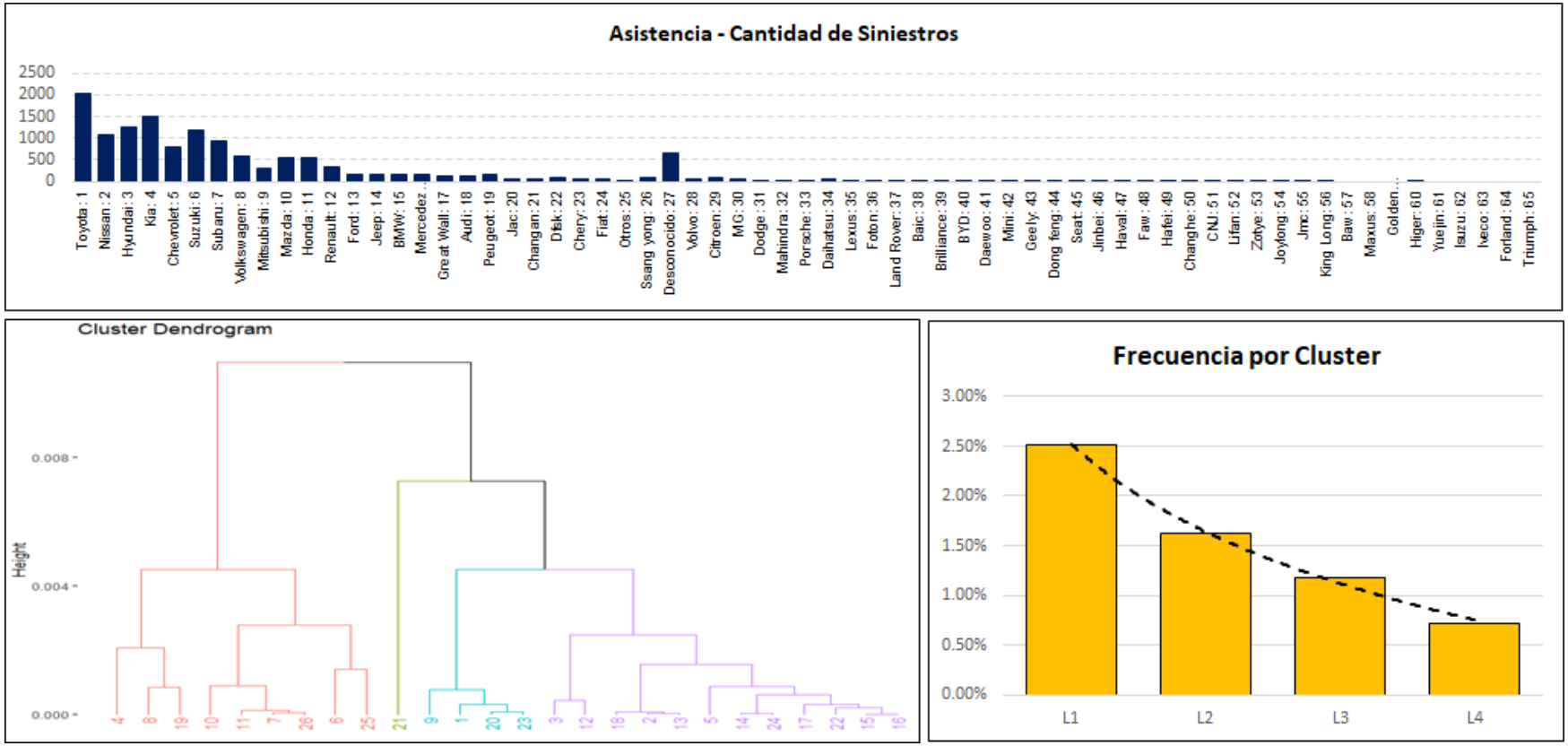
VEHÍCULOS LIVIANOS - RESPONSABILIDAD CIVIL

Figure A .4: Agrupación de la Marca Vehículo - Responsabilidad Civil - Livianos



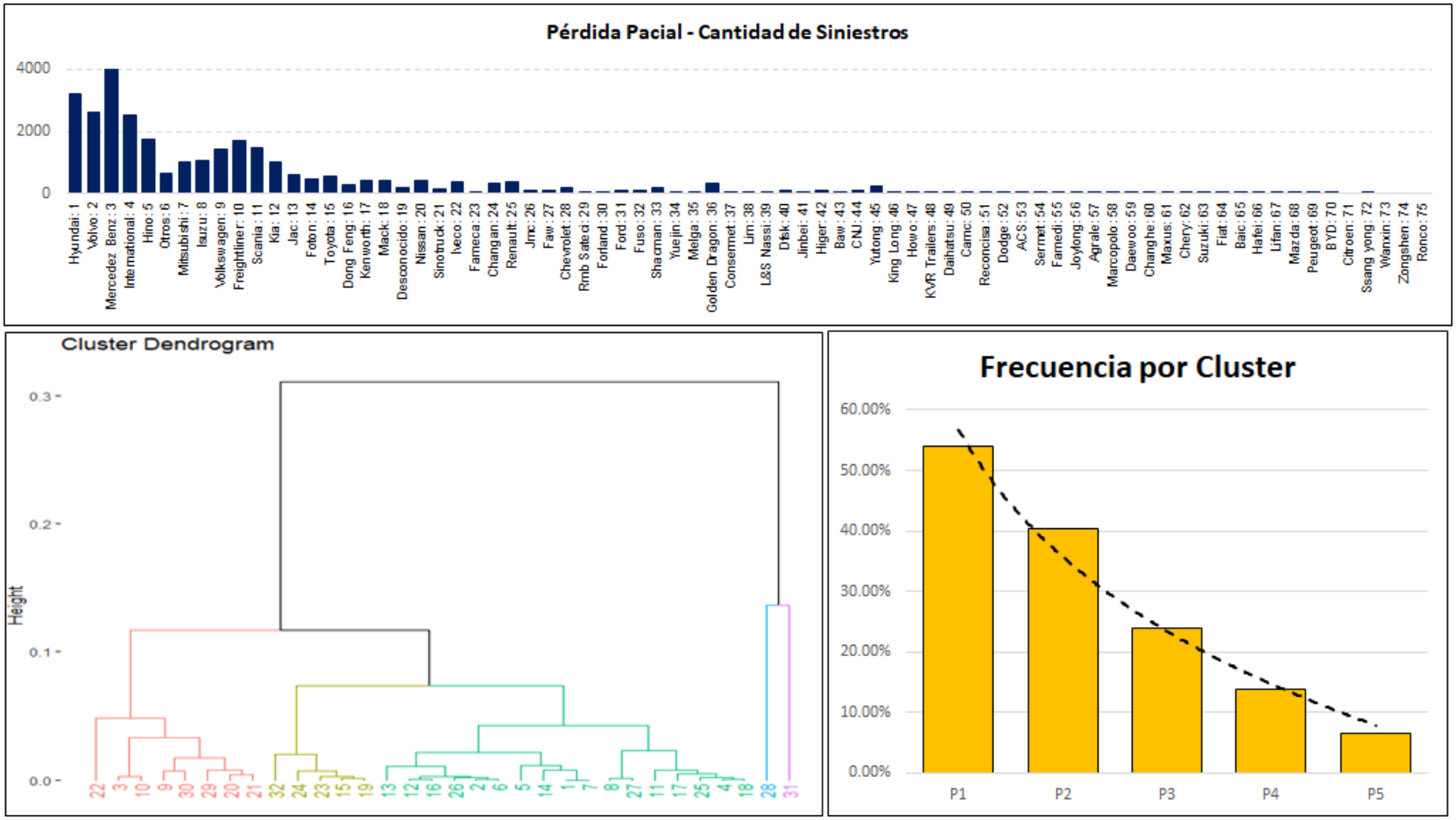
VEHÍCULOS LIVIANOS - ASISTENCIAS

Figure A .5: Agrupación de la Marca Vehículo - Asistencias - Livianos



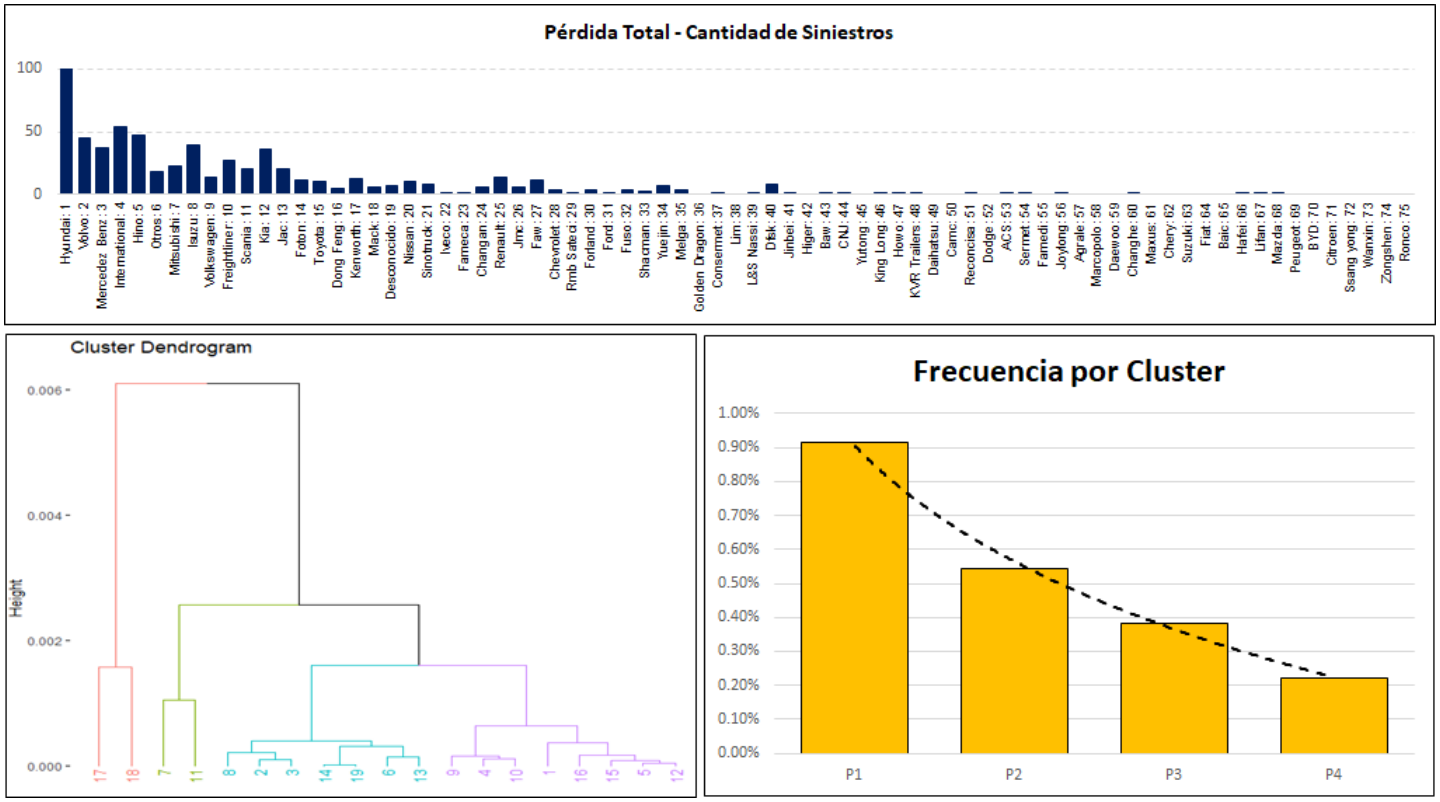
VEHÍCULOS PESADOS - PÉRDIDAS PARCIALES

Figure A .6: Agrupación de la Marca Vehículo - Pérdida Parcial - Pesados



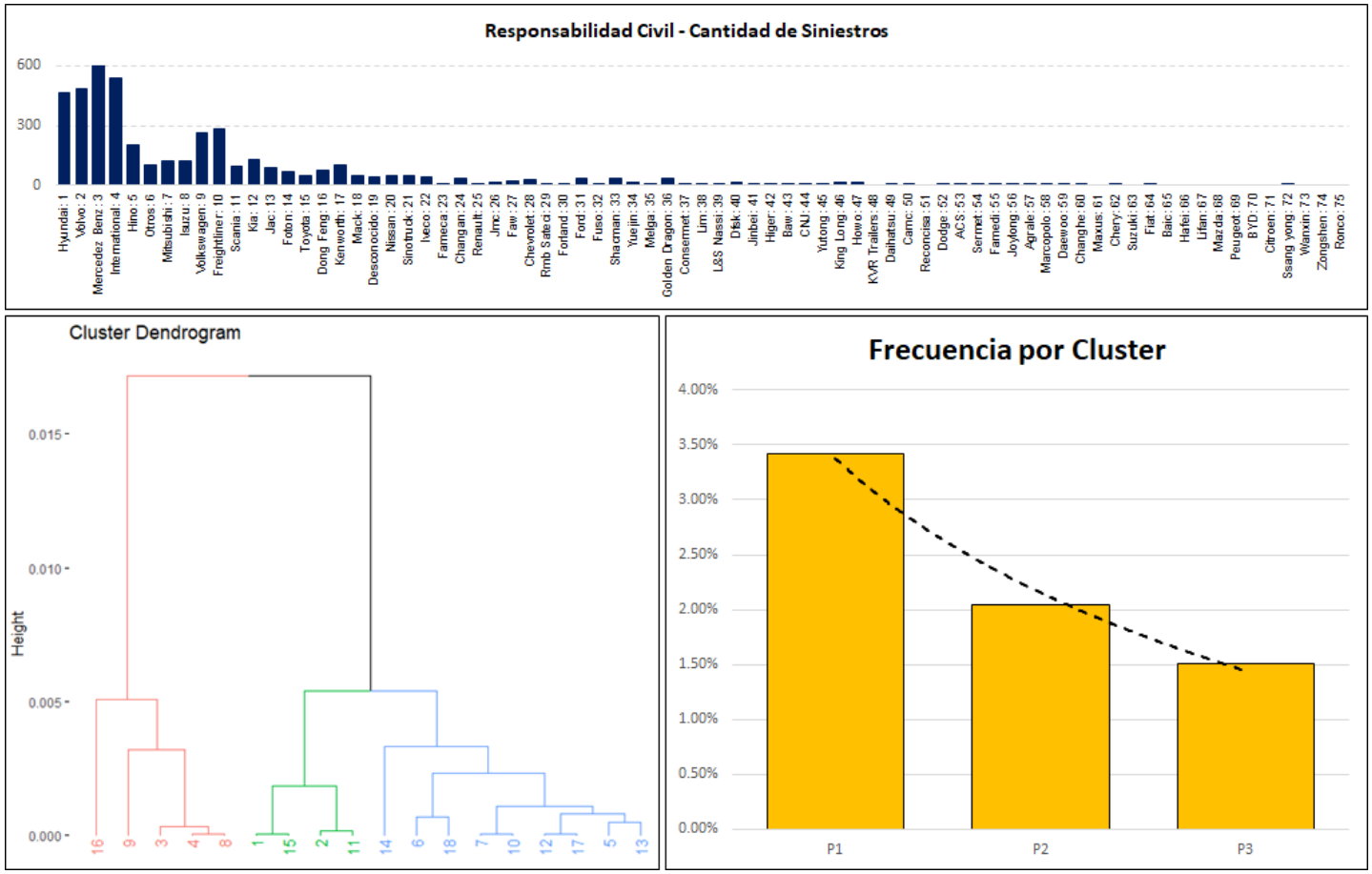
VEHÍCULOS PESADOS - PÉRDIDAS TOTALES

Figure A .7: Agrupación de la Marca Vehículo - Pérdida Total - Pesados



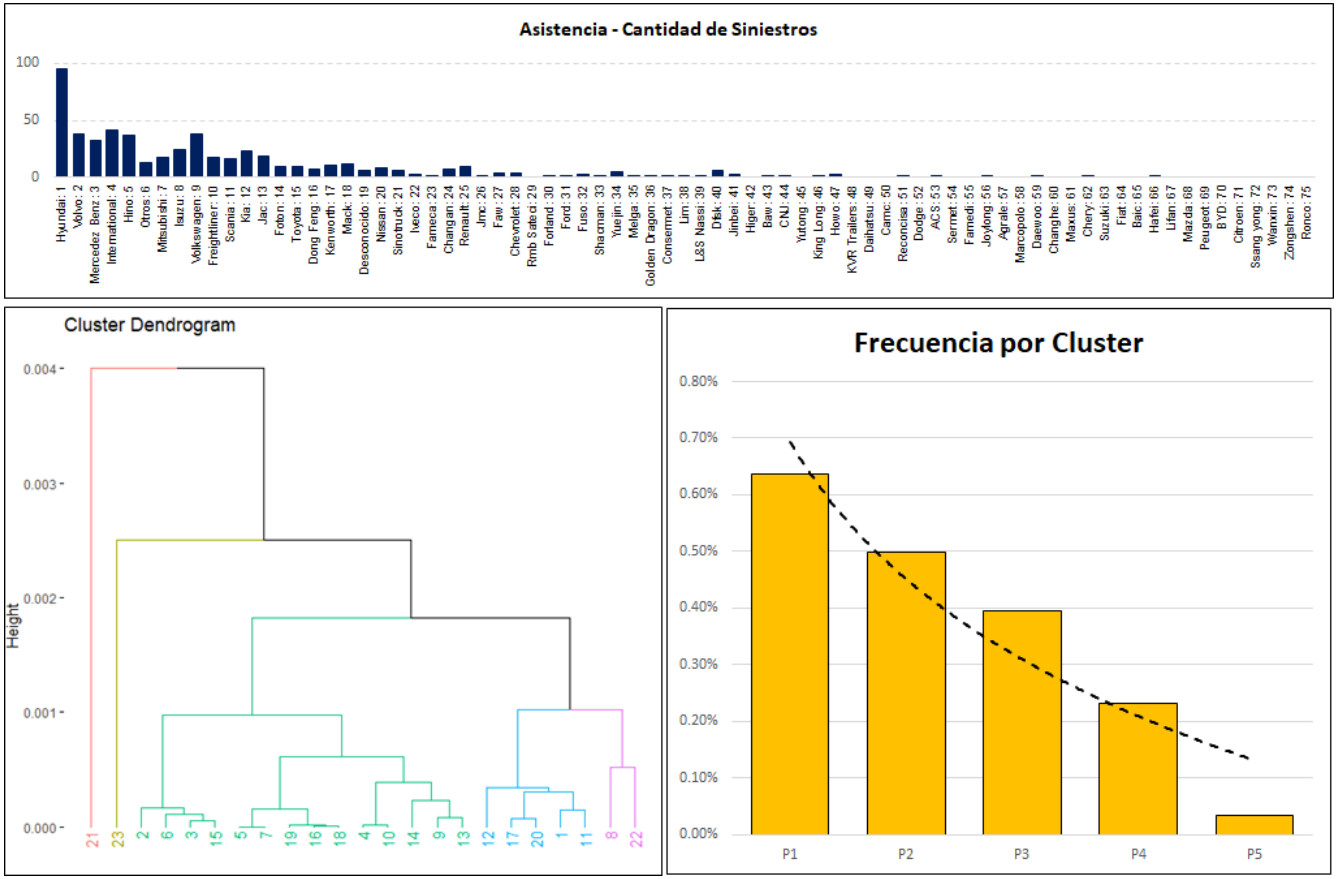
VEHÍCULOS PESADOS - RESPONSABILIDAD CIVIL

Figure A .8: Agrupación de la Marca Vehículo - Responsabilidad Civil - Pesados



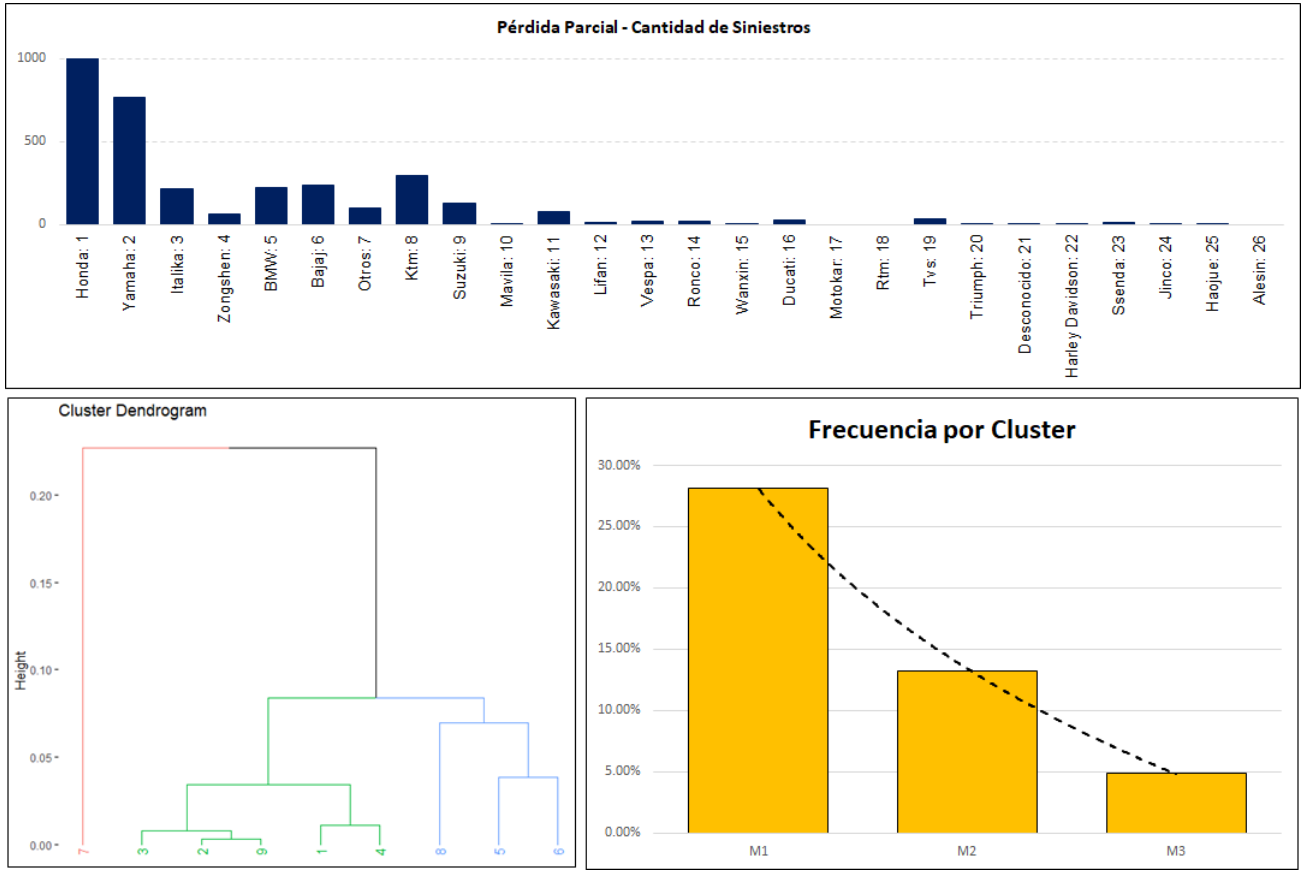
VEHÍCULOS PESADOS - ASISTENCIAS

Figure A .9: Agrupación de la Marca Vehículo - Asistencias - Pesados



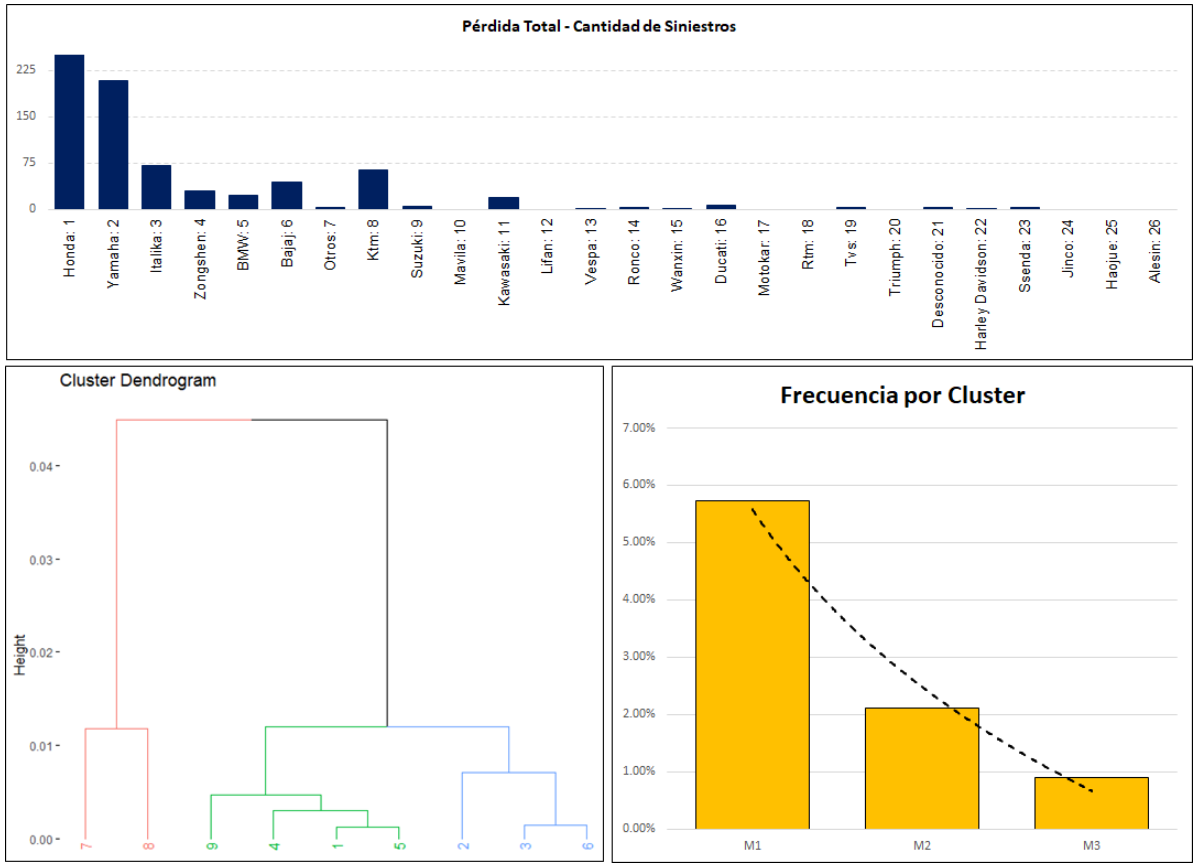
VEHÍCULOS MENORES - PÉRDIDAS PARCIALES

Figure A .10: Agrupación de la Marca Vehículo - Pérdida Parcial - Vehículos Menores



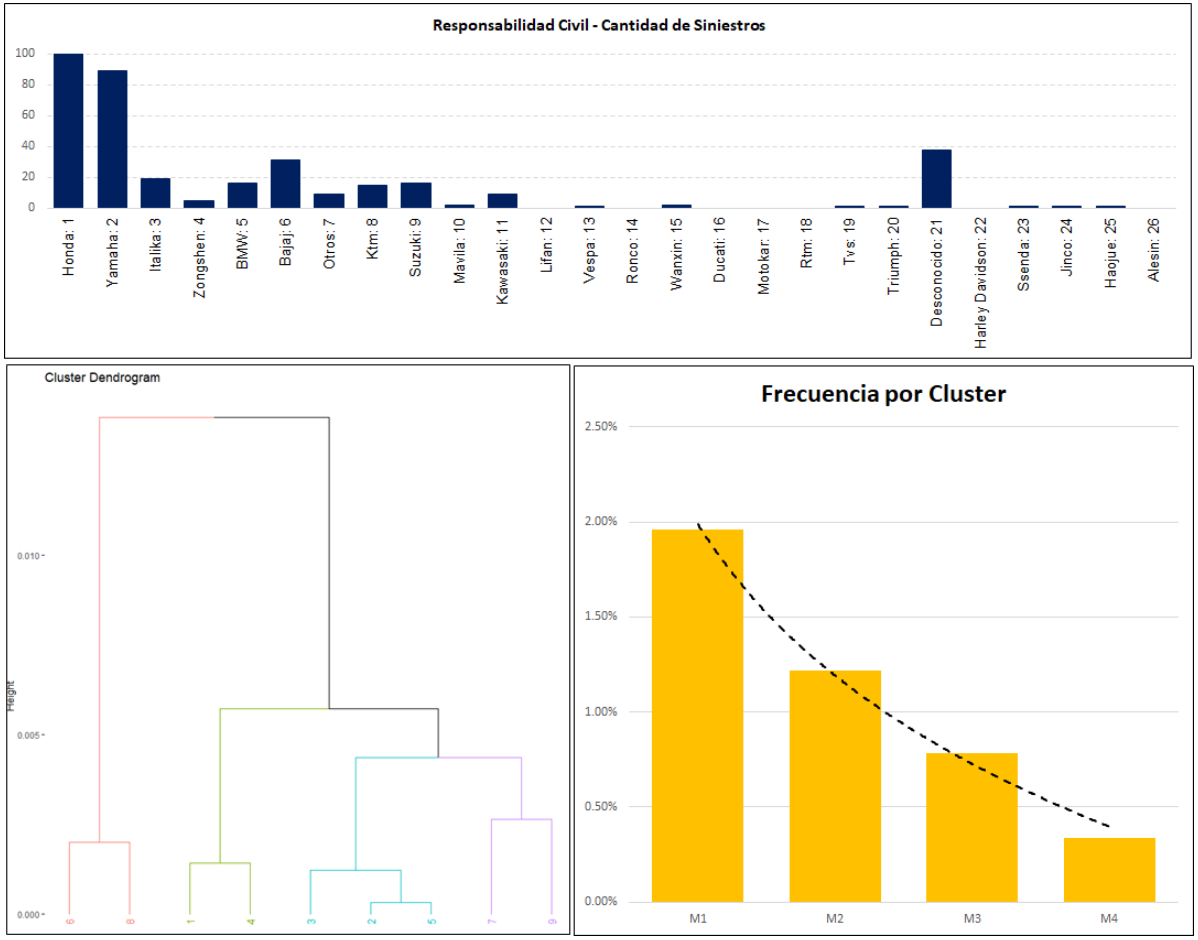
VEHÍCULOS MENORES - PÉRDIDAS TOTALES

Figure A .11: Agrupación de la Marca Vehículo - Pérdida Total - Vehículos Menores



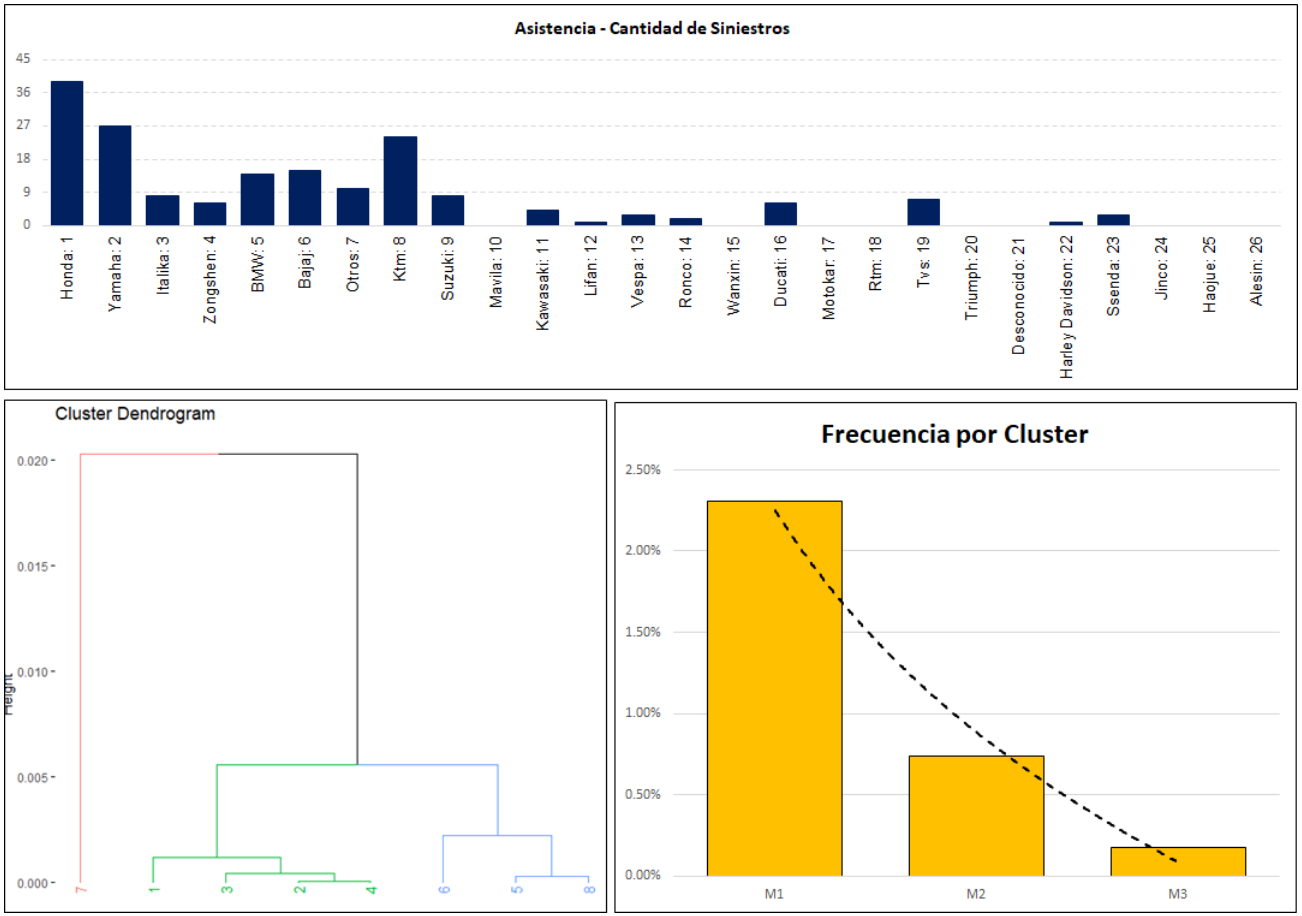
VEHÍCULOS MENORES - RESPONSABILIDAD CIVIL

Figure A .12: Agrupación de la Marca Vehículo - Responsabilidad Civil - Vehículos Menores



VEHÍCULOS MENORES - ASISTENCIAS

Figure A .13: Agrupación de la Marca Vehículo - Asistencias - Vehículos Menores



Variable: Edad del Asegurado

Figure A .14: Edad del asegurado - Sin Agrupar

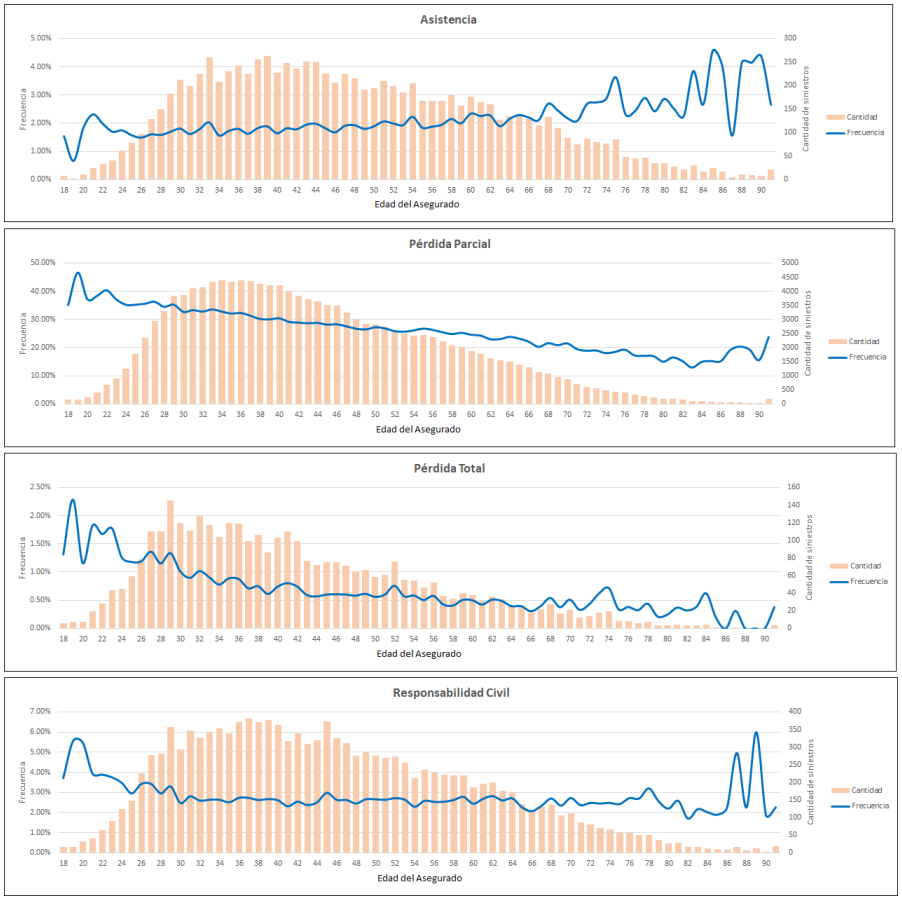
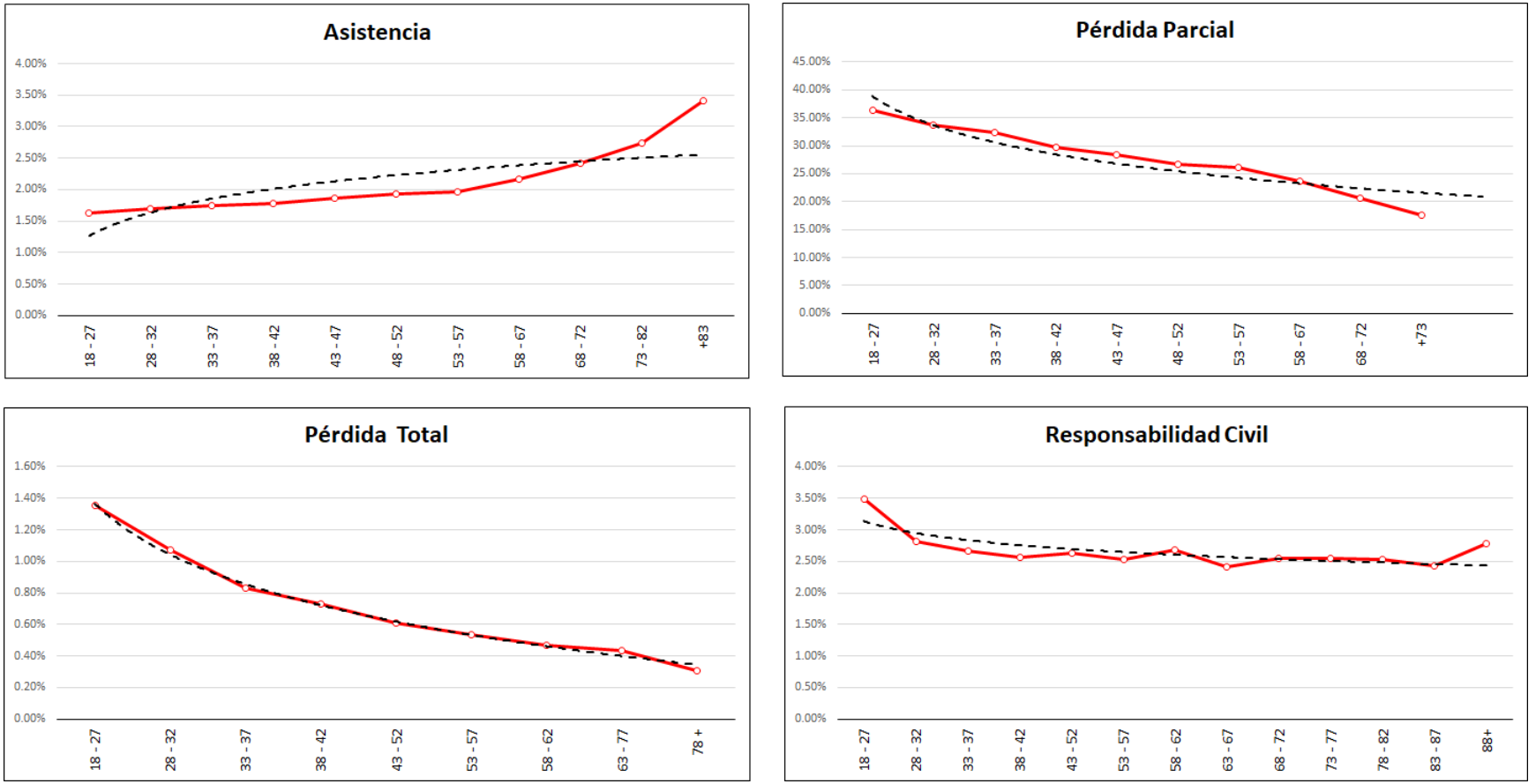
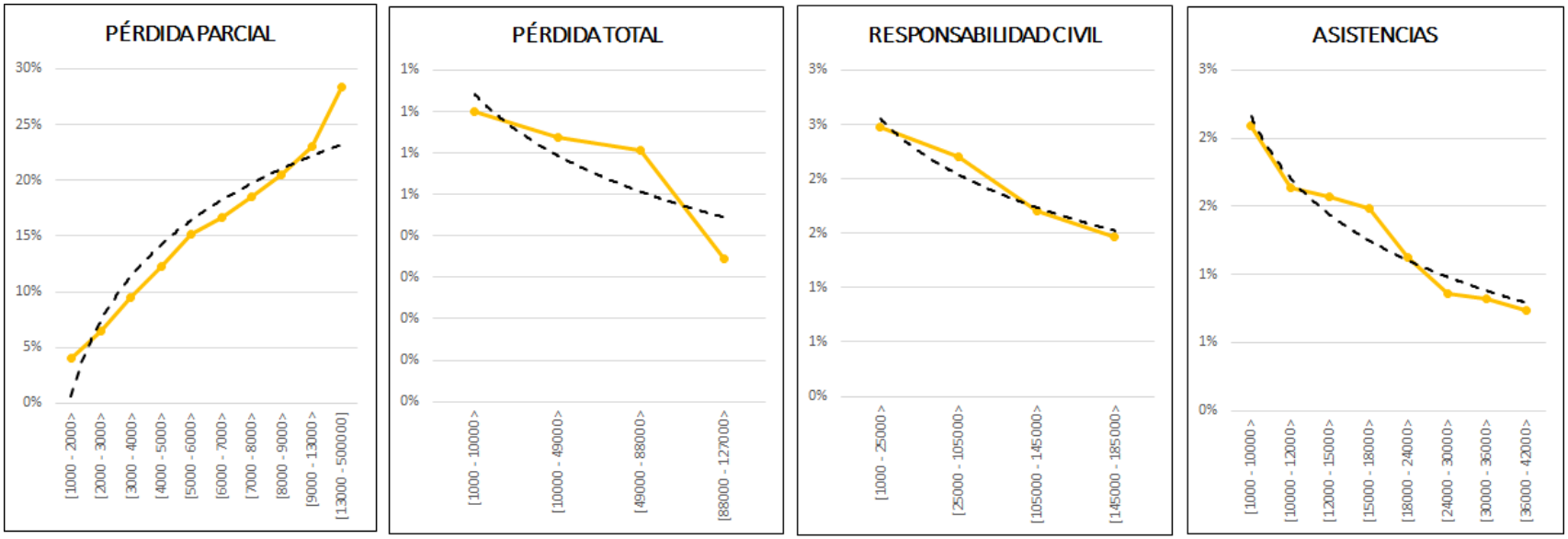


Figure A .15: Edad del asegurado - Agrupado en rangos



Variable: Suma Asegurada

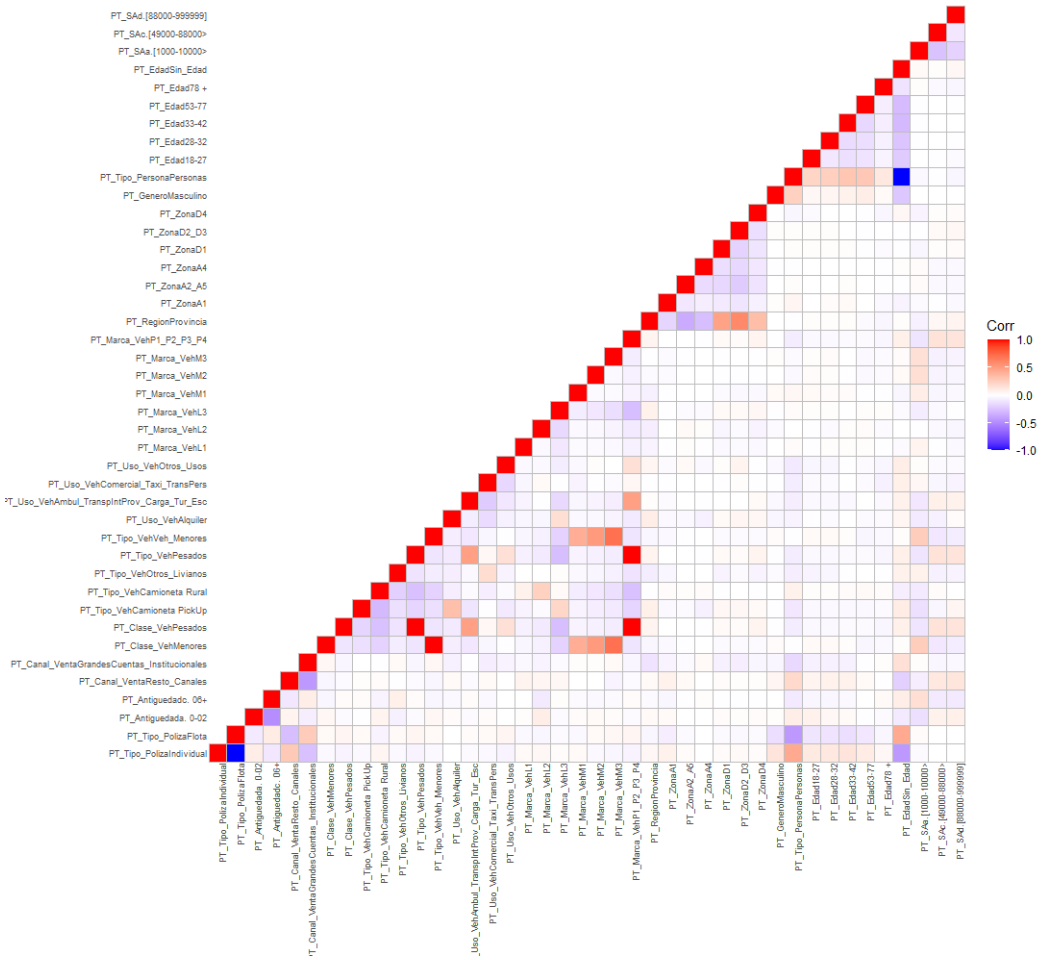
Figure A .16: Rangos de Suma Asegurada del Vehículo



ANEXO B: MODELOS GLM

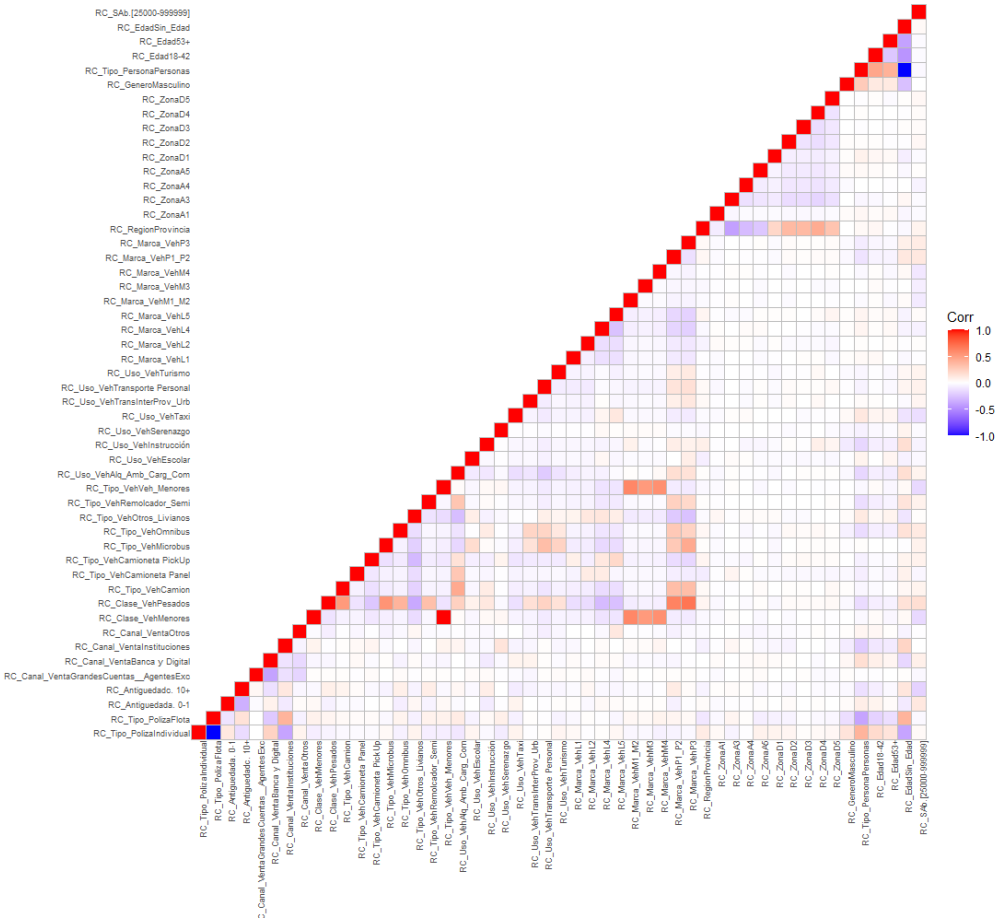
B.1. Matriz de Correlación entre las variables para detectar posible multicolinealidad

Figure B .17: Matriz de correlación - Pérdida Total



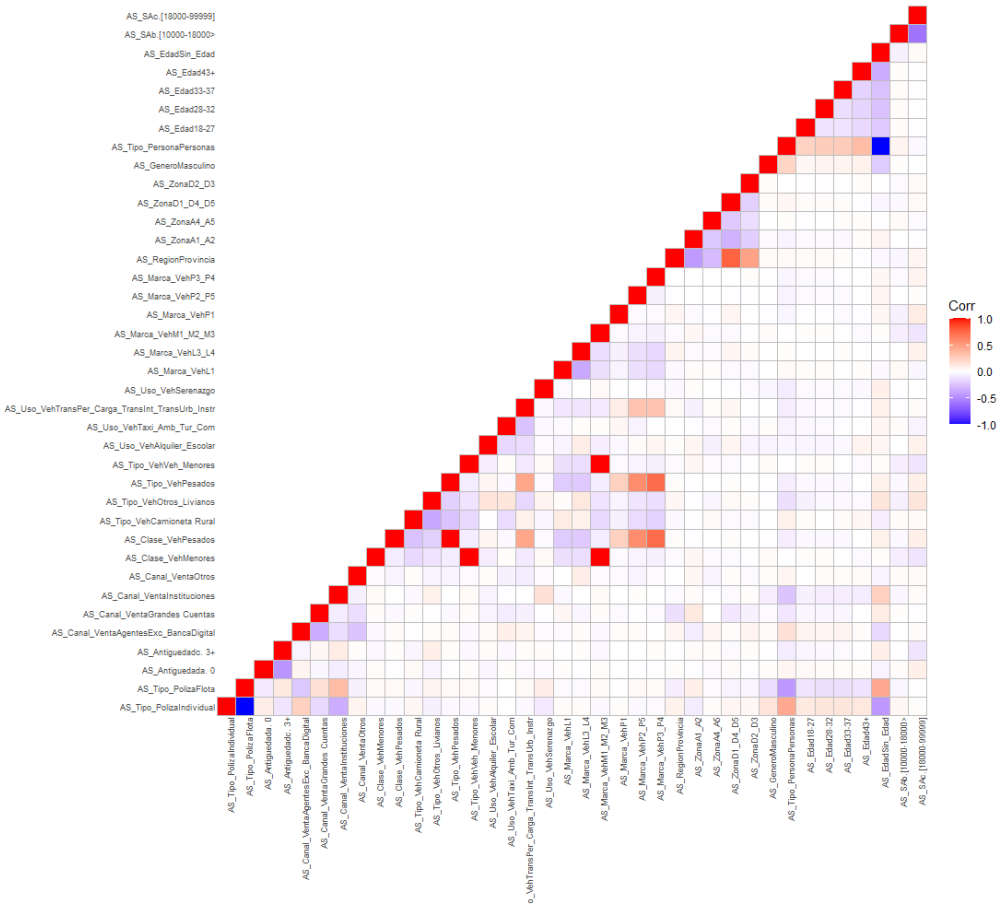

```
# MATRIZ DE CORRELACIÓN
library(ggplot2)
library(ggcorrplot)
library(dplyr)
load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Perdida_Total/glm.Frec.PT.RData")
load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Perdida_Total/Data_PT.RData")
data_PP <- data %>% group_by(PP_Tipo_Poliza,PP_Antiguedad,PP_Canal_Venta,PP_Clase_Veh,PP_Tipo_Veh,PP_Uso_Veh,PP_Marca_Veh,PP_Region,PP_Zona,PP_Genero,PP_Tipo_Persona,PP_Edad,PP_SA) %>%summarise(n=n())
data_PP <- data_PP[,1:13]
model.matrix(~0+.,data=data_PP) %>% cor(use="pairwise.complete.obs") %>%ggcorrplot(show.diag=TRUE, type="lower", lab=FALSE) +theme(axis.text.x=element_text(size=7, angle=90, vjust=1,hjust=1,margin=margin(-3,0,0,0)), axis.text.y=element_text(size=7,margin=margin(0,-3,0,0)),panel.grid.major=element_blank())
```

Figure B .18: Matriz de correlación - Responsabilidad Civil




```
# MATRIZ DE CORRELACIÓN
library(ggplot2)
library(ggcorrplot)
library(dplyr)
load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Responsabilidad_Civil/glm.Frec.RC.RData")
load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Responsabilidad_Civil/Data_RC.RData")
data_RC <- data %>% group_by(RC_Tipo_Poliza,RC_Antiguedad,RC_Canal_Venta,RC_Clase_Veh,RC_Tipo_Veh,RC_Uso_Veh,RC_Marca_Veh,RC_Region,RC_Zona,RC_Genero,RC_Tipo_Persona,RC_Edad,RC_SA) %>%
summarise(n=n())
data_RC <- data_RC[,1:13]
model.matrix(~0+.,data=data_RC) %>% cor(use="pairwise.complete.obs") %>% ggcorrplot(show.diag=TRUE, type="lower", lab=FALSE) + theme(axis.text.x=element_text(size=7, angle=90, vjust=1,hjust=1,margin=margin(-3,0,0,0)),axis.text.y=element_text(size=7,margin=margin(0,-3,0,0)),panel.grid.major=element_blank())
```

Figure B .19: Matriz de correlación - Asistencias




```r
# MATRIZ DE CORRELACIÓN
library(ggplot2)
library(ggcorrplot)
library(dplyr)
load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Asistencia/glm.Frec.AS.RData")
load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Asistencia/Data_AS.RData")
data_AS <- data %>%group_by(AS_Tipo_Poliza,AS_Antiguedad,AS_Canal_Venta,AS_Clase_Veh,AS_Tipo_Veh,AS_Uso_Veh,AS_Marca_Veh,AS_Region,AS_Zona,AS_Genero,AS_Tipo_Persona,AS_Edad,AS_SA) %>% summarise(n=n())
data_AS <- data_AS[,1:13]
model.matrix(~0+.,data=data_AS) %>% cor(use="pairwise.complete.obs") %>%ggcorrplot(show.diag=TRUE, type="lower", lab=FALSE) + theme(axis.text.x=element_text(size=7, angle=90, vjust=1,hjust=1,margin=margin(-3,0,0,0)),axis.text.y=element_text(size=7,margin=margin(0,-3,0,0)),panel.grid.major=element_blank())
```

B.2. Modelos Marginales de la Frecuencia y de la Severidad

Código R: Modelo Frecuencia - Pédida Parcial

```
################################################################################
##########                  MODELO GLM - FRECUENCIA
################################################################################

# PAQUETES Y LIBRERÍAS
options(repos = c(cran="http://cran.rstudio.com"))
library(RODBC)
library(tidyverse) # Incluye dplyr y tidyr para manejar los datos y ggplot para representarlos en grÃ¡ficos
library(plotly)    # Para graficos interactivos
library(lubridate) # manejar datas y horas
library(forcats)
library(rcartocolor)
library(readr)      # para leer y importar archivos
library(skimr)
library(gcmr)
db<-"C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/01_Frecuencia/03_Data/BD_GLM.accdb"
con <- odbcConnectAccess2007(db)
data <- sqlQuery(con,"select * from PP_Datos_Frecuencia")
data.NúmeroPóliza<-format(data.NúmeroPóliza,scientific = FALSE)
# Reducimos la data sin el año 2015 y 2020, de acuerdo a los análisis univariados realizados
data <- data %>% filter(Año!=2015 & Año!=2020)
data <- data %>% filter(PP_Clase_Veh!='Desconocido')
# Cambiamos el nombre de algunos niveles de algunas variables
data <- data %>% mutate(PP_Antiguedad = case_when(
  PP_Antiguedad == '0' ~ 'a. 00',PP_Antiguedad == '1' ~ 'b. 01',PP_Antiguedad == '2' ~ 'c. 02',
  PP_Antiguedad == '3' ~ 'd. 03',PP_Antiguedad == '4' ~ 'e. 04',PP_Antiguedad == '5' ~ 'f. 05',
  PP_Antiguedad == '6' ~ 'g. 06',PP_Antiguedad == '7' ~ 'h. 07',PP_Antiguedad == '8' ~ 'i. 08',
  PP_Antiguedad == '9' ~ 'j. 09',PP_Antiguedad == '10' ~ 'k. 10',PP_Antiguedad == '11-13' ~ 'l. 11-13',
  PP_Antiguedad == '14-19' ~ 'm. 14-19',PP_Antiguedad == '20+' ~ 'n. 20+'
))

data <- data %>% mutate(PP_SA = case_when(
  PP_SA == '[1000-2000>' ~ 'a.[1000-2000>',PP_SA == '[2000-3000>' ~ 'b.[2000-3000>',PP_SA == '[3000-4000]' ~ '
      c.[3000-4000>',PP_SA == '[4000-5000>' ~ 'd.[4000-5000>',PP_SA == '[5000-6000>' ~ 'e.[5000-6000>',PP_SA ==
       '[6000-7000>' ~ 'f.[6000-7000>',PP_SA == '[7000-8000>' ~ 'g.[7000-8000>',PP_SA == '[8000-9000>' ~ 'h
      .[8000-9000>',PP_SA == '[9000-13000>' ~ 'i.[9000-13000>',PP_SA == '[13000-999999]' ~ 'j.[13000-999999]'
))

# Convertimos a factor las variables

data.Año <- as.factor(data.Año)
data.PP_Tipo_Poliza <- as.factor(data.PP_Tipo_Poliza)
data.PP_Antiguedad <- as.factor(data.PP_Antiguedad)
data.PP_Canal_Venta <- as.factor(data.PP_Canal_Venta)
data.PP_Clase_Veh <- as.factor(data.PP_Clase_Veh)
data.PP_Tipo_Veh <- as.factor(data.PP_Tipo_Veh)
data.PP_Uso_Veh <- as.factor(data.PP_Uso_Veh)
data.PP_Marca_Veh <- as.factor(data.PP_Marca_Veh)
data.PP_Region <- as.factor(data.PP_Region)
data.PP_Zona <- as.factor(data.PP_Zona)
data.PP_Genero <- as.factor(data.PP_Genero)
data.PP_Tipo_Persona <- as.factor(data.PP_Tipo_Persona)
data.PP_Edad <- as.factor(data.PP_Edad)
data.PP_SA <- as.factor(data.PP_SA)

# Asignamos el intercepto para cada variable
data <- data %>% mutate(
  Año = fct_relevel(Año, "2022", after = 0),
  PP_Tipo_Poliza = fct_relevel(PP_Tipo_Poliza, "Individual", after = 0),
  PP_Antiguedad = fct_relevel(PP_Antiguedad, "c. 02", after = 0),
  PP_Canal_Venta = fct_relevel(PP_Canal_Venta, "Corredores", after = 0),
  PP_Clase_Veh = fct_relevel(PP_Clase_Veh, "Livianos", after = 0),
  PP_Tipo_Veh = fct_relevel(PP_Tipo_Veh, "Automovil", after = 0),
  PP_Uso_Veh = fct_relevel(PP_Uso_Veh, "Particular", after = 0) ,
  PP_Marca_Veh = fct_relevel(PP_Marca_Veh, "L3", after = 0),
  PP_Region = fct_relevel(PP_Region, "Lima", after = 0),
  PP_Zona = fct_relevel(PP_Zona, "A2", after = 0),
  PP_Genero = fct_relevel(PP_Genero, "Femenino", after = 0),
  PP_Tipo_Persona = fct_relevel(PP_Tipo_Persona, "Empresa", after = 0 ),
  PP_Edad = fct_relevel(PP_Edad, "38-42", after = 0),
  PP_SA = fct_relevel(PP_SA, "j.[13000-999999]", after = 0)
)
```

```r

# Verificación de frecuencia por factor

data %>% group_by(Año) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*100)
data %>% group_by(PP_Tipo_Poliza) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>%mutate(Freq=round(C/E,5)*100)
data %>% group_by(PP_Antiguedad) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>%mutate(Freq=round(C/E,5)*100)
data %>% group_by(PP_Canal_Venta) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>%mutate(Freq=round(C/E,5)*100)
data %>% group_by(PP_Clase_Veh) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>%mutate(Freq=round(C/E,5)*100)
data %>% group_by(PP_Tipo_Veh) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*100)
data %>% group_by(PP_Uso_Veh) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*100)
data %>% group_by(PP_Marca_Veh) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>%mutate(Freq=round(C/E,5)*100)
data %>% group_by(PP_Region) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*100)
data %>% group_by(PP_Zona) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*100)
data %>% group_by(PP_Genero) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*100)
data %>% group_by(PP_Tipo_Persona) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>%mutate(Freq=round(C/E,5)*100)
data %>% group_by(PP_Edad) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*100)
data %>% group_by(PP_SA) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*100)
# Agrupaciones motivadas por modelos previos
data <- data %>%
  mutate(
    PP_Canal_Venta=fct_collapse(PP_Canal_Venta,"BancaDigital_Otros"=c("Banca y Digital","Otros")),
    PP_Marca_Veh=fct_collapse(PP_Marca_Veh,"M1_M2_M3"=c("M1","M2","M3")),
    PP_Zona=fct_collapse(PP_Zona,"A4_A6"=c("A4","A6"))
  )
################################################################################################
# Agrupamos la data para modelar el GLM
  #data_model <- data %>% mutate(id=as.numeric(`Número Póliza`)*100+as.numeric(`Número Riesgo`)) %>% group_by(id,Año,PP_Tipo_Poliza,PP_Antiguedad,PP_Canal_Venta,PP_Uso_Veh,PP_Marca_Veh,PP_Zona,PP_Genero,PP_Edad,PP_SA)%>%summarise(Expuesto=sum(Expuesto),Prima_Devengada=sum(Prima_Devengada),Cantidad=sum(Frecuencia),Incurrido=sum(Incurrido))
  data_model <- data %>%group_by(Año,PP_Tipo_Poliza,PP_Antiguedad,PP_Canal_Venta,PP_Uso_Veh,PP_Marca_Veh,PP_Zona,PP_Genero,PP_Edad,PP_SA)%>%summarise(Expuesto=sum(Expuesto),Prima_Devengada=sum(Prima_Devengada),Cantidad=sum(Frecuencia),Incurrido=sum(Incurrido))
# Modelo GLM Pérdida Parcial
glm.Frec.PP <- glm(
  formula = Cantidad ~
    Año +
    PP_Tipo_Poliza +
    PP_Antiguedad +
    PP_Canal_Venta +
    #PP_Clase_Veh + (Se correlaciona con la MarcaVeh)
    #PP_Tipo_Veh +  (Se correlaciona con la MarcaVeh)
    PP_Uso_Veh +
    PP_Marca_Veh +
    #PP_Region + (Se correlaciona con la zona)
    PP_Zona +
    PP_Genero +
    #PP_Tipo_Persona + (Se correlaciona con la Edad)
    PP_Edad +
    PP_SA +
    offset(log(Expuesto)),
  family = poisson(link = "log"),
  data = data_model
)
# Desenmascarar posibles multicolinearidades entre las diferentes variables
temp <- alias(glm.Frec.PP)
temp2 <- as.data.frame(temp.Complete) # Si es vacio, no hay multicolinearidades
rm(temp,temp2)
# No se realizarán agrupaciones en los niveles de algunas variables por tener un buen modelo
# Incluimos las predicciones en la data
data <- data %>% mutate(Frecuencia.PRED = glm.Frec.PP.fitted.values)
data %>% summarise(sum(Frecuencia), sum(Frecuencia.PRED))
# Salvar un data frame con los coeficientes del modelo
coefs.Frecuencia.PP <- summary(glm.Frec.PP).coefficients %>%
  as_tibble() %>%
  mutate(var_level = glm.Frec.PP.coefficients %>% names()) %>%
```

```
  select(var_level, everything())
# Adicionar manualmente a la tabla los niveles atribuidos al intercept
temp <- tibble(
  var_level = c("Año2022","PP_Tipo_PolizaIndividual","PP_Antiguedadc. 02","PP_Canal_VentaCorredores","PP_Uso_
      VehParticular","PP_Marca_VehL3","PP_ZonaA2","PP_GeneroFemenino", "PP_Edad38-42","PP_SAj.[13000-999999]"
  ),
  Estimate = 0
)
f.round <- function(x, .digits = 4) round(x, digits = .digits)
coefs.Frecuencia.PP <- coefs.Frecuencia.PP %>%
  bind_rows(temp) %>% arrange(var_level) %>% mutate(`exp(Estimate)` = exp(Estimate)) %>% mutate_if(is.double,
      f.round)
# Esta función considera los datos de entrenamiento del GLM, agrega la exposición por variable #y la junta a
      la tabla de coeficientes.
incluir.exposicion <- function(.data, .tbl.coefs){
  names.vars <- names(.data %>% select(-Expuesto))
  n <- length(names.vars)
  list.temp <- vector(mode = "list", length = n)
  for(i in seq_along(names.vars)){
    varname <- sym(names.vars[[i]])
    list.temp[[i]] <- .data %>%
      group_by(!!varname) %>%
      summarize(Exposición = f.round(sum(Expuesto), .digits = 0)) %>%
      ungroup() %>%
      mutate(var_level = str_c(names.vars[[i]], !!varname)) %>%
      select(var_level, Exposición)
  }
  exp.by.var <- bind_rows(list.temp)
  return(.tbl.coefs %>% left_join(exp.by.var))
}
coefs.Frecuencia.PP <- data %>% select(-Frecuencia, -Frecuencia.PRED) %>%incluir.exposicion(coefs.Frecuencia.
      PP)
coefs.Frecuencia.PP %>% DT::datatable()
# Guardamos los coeficientes en excel
write.csv(coefs.Frecuencia.PP,"C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/01_
      Frecuencia/04_Coeficientes/coefs.Frecuencia.PP.csv",fileEncoding ="Latin1")
save(glm.Frec.PP, file = "C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_
      Marginales/Perdida_Parcial/glm.Frec.PP.RData")
#data<-data %>% group_by(`Número Póliza`,`Número Riesgo`,Año,PP_Tipo_Poliza,PP_Antiguedad,PP_Canal_Venta,PP_
      Uso_Veh,PP_Marca_Veh,PP_Zona,PP_Genero,PP_Edad,PP_SA) %>%summarise(Expuesto=sum(Expuesto),Prima_Devengada
      =sum(Prima_Devengada),Cantidad=sum(Frecuencia),Incurrido=sum(Incurrido))
save(data, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/
      Perdida_Parcial/Data_PP.RData")
save(data_model, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales
      /Perdida_Parcial/Data_PP_model.RData")
```

Código R: Modelo Severidad - Pédida Parcial

```
################################################################################
##########                  MODELO GLM - SEVERIDAD
################################################################################
# PAQUETES Y LIBRERÍAS
#options(repos = c(cran="http://cran.rstudio.com"))
#install.packages("RODBC")
#install.packages("skimr")
library(RODBC)
library(tidyverse) # Incluye dplyr y tidyr para manejar los datos y ggplot para representarlos en grÃ¡ficos
library(plotly)    # Para graficos interactivos
library(lubridate) # manejar datas y horas
library(forcats)
library(rcartocolor)
library(readr)      # para leer y importar archivos
library(skimr)
#db2<-"C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/02_CostoMedio/03_Data/BD_GLM_CM.
      accdb"
#con2 <- odbcConnectAccess2007(db2)
#data_CM <- sqlQuery(con2,"select * from PP_Datos_Severidad")
# Reducimos la data sin el año 2015 y 2020, de acuerdo a los análisis univariados realizados
#data_CM <- data_CM %>% filter(Año!=2015 & Año!=2020)
#data_CM <- data_CM %>% filter(PP_Clase_Veh!='Desconocido')
data_CM <- data_model %>% filter(Incurrido>0) # onitir todas las acciones y saltar al modelo GLM
#data_CM.`Número Póliza`<-format(data_CM.`Número Póliza`,scientific = FALSE)
# Cambiamos el nombre de algunos niveles de algunas variables
data_CM <- data_CM %>% mutate(PP_Antiguedad = case_when(
  PP_Antiguedad == '0' ~ 'a. 00',PP_Antiguedad == '1' ~ 'b. 01',PP_Antiguedad == '2' ~ 'c. 02',
```

```
  PP_Antiguedad == '3' ~ 'd. 03',PP_Antiguedad == '4' ~ 'e. 04',PP_Antiguedad == '5' ~ 'f. 05',
  PP_Antiguedad == '6' ~ 'g. 06',PP_Antiguedad == '7' ~ 'h. 07',PP_Antiguedad == '8' ~ 'i. 08',
  PP_Antiguedad == '9' ~ 'j. 09',PP_Antiguedad == '10' ~ 'k. 10',PP_Antiguedad == '11-13' ~ 'l. 11-13',
  PP_Antiguedad == '14-19' ~ 'm. 14-19',PP_Antiguedad == '20+' ~ 'n. 20+'
))

data_CM <- data_CM %>% mutate(PP_SA = case_when(
  PP_SA == '[1000-2000>' ~ 'a.[1000-2000>',PP_SA == '[2000-3000>' ~ 'b.[2000-3000>',PP_SA == '[3000-4000]' ~ '
      c.[3000-4000>',PP_SA == '[4000-5000>' ~ 'd.[4000-5000>',PP_SA == '[5000-6000>' ~ 'e.[5000-6000>',PP_SA ==
       '[6000-7000>' ~ 'f.[6000-7000>',PP_SA == '[7000-8000>' ~ 'g.[7000-8000>',PP_SA == '[8000-9000>' ~ 'h
      .[8000-9000>',PP_SA == '[9000-13000>' ~ 'i.[9000-13000>',PP_SA == '[13000-999999]' ~ 'j.[13000-999999]'
))

# Convertimos a factor las variables
data_CM.Año <- as.factor(data_CM.Año)
data_CM.PP_Tipo_Poliza <- as.factor(data_CM.PP_Tipo_Poliza)
data_CM.PP_Antiguedad <- as.factor(data_CM.PP_Antiguedad)
data_CM.PP_Canal_Venta <- as.factor(data_CM.PP_Canal_Venta)
#data_CM.PP_Clase_Veh <- as.factor(data_CM.PP_Clase_Veh)
#data_CM.PP_Tipo_Veh <- as.factor(data_CM.PP_Tipo_Veh)
data_CM.PP_Uso_Veh <- as.factor(data_CM.PP_Uso_Veh)
data_CM.PP_Marca_Veh <- as.factor(data_CM.PP_Marca_Veh)
#data_CM.PP_Region <- as.factor(data_CM.PP_Region)
data_CM.PP_Zona <- as.factor(data_CM.PP_Zona)
data_CM.PP_Genero <- as.factor(data_CM.PP_Genero)
data_CM.PP_Tipo_Persona <- as.factor(data_CM.PP_Tipo_Persona)
data_CM.PP_Edad <- as.factor(data_CM.PP_Edad)
data_CM.PP_SA <- as.factor(data_CM.PP_SA)

# Asignamos el intercepto para cada variable

data_CM <- data_CM %>% mutate(
  Año = fct_relevel(Año, "2022", after = 0),
  PP_Tipo_Poliza = fct_relevel(PP_Tipo_Poliza, "Individual", after = 0),
  PP_Antiguedad = fct_relevel(PP_Antiguedad, "c. 02", after = 0),
  PP_Canal_Venta = fct_relevel(PP_Canal_Venta, "Corredores", after = 0),
  PP_Clase_Veh = fct_relevel(PP_Clase_Veh, "Livianos", after = 0),
  PP_Tipo_Veh = fct_relevel(PP_Tipo_Veh, "Automovil", after = 0),
  PP_Uso_Veh = fct_relevel(PP_Uso_Veh, "Particular", after = 0) ,
  PP_Marca_Veh = fct_relevel(PP_Marca_Veh, "L3", after = 0),
  PP_Region = fct_relevel(PP_Region, "Lima", after = 0),
  PP_Zona = fct_relevel(PP_Zona, "A2", after = 0),
  PP_Genero = fct_relevel(PP_Genero, "Femenino", after = 0),
  PP_Tipo_Persona = fct_relevel(PP_Tipo_Persona, "Empresa", after = 0),
  PP_Edad = fct_relevel(PP_Edad, "38-42", after = 0),
  PP_SA = fct_relevel(PP_SA, "j.[13000-999999]", after = 0)
)

# Verificación de frecuencia por factor
data_CM %>% group_by(Año) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(PP_Tipo_Poliza) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(PP_Antiguedad) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(PP_Canal_Venta) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(PP_Clase_Veh) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(PP_Tipo_Veh) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(PP_Uso_Veh) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(PP_Marca_Veh) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(PP_Region) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(PP_Zona) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(PP_Genero) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(PP_Tipo_Persona) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(PP_Edad) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(PP_SA) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))

# Agrupaciones motivadas por modelos previos
data_CM <- data_CM %>%
  mutate(
    PP_Canal_Venta=fct_collapse(PP_Canal_Venta,"BancaDigital_Otros"=c("Banca y Digital","Otros")),
    PP_Marca_Veh=fct_collapse(PP_Marca_Veh,"M1_M2_M3"=c("M1","M2","M3") ),
    PP_Zona=fct_collapse(PP_Zona,"A4_A6"=c("A4","A6"))
  )

# Creamos las variable dependiente
data_CM <-data_CM %>% mutate(CM=Incurrido/Cantidad)
# Modelo GLM Pérdida Parcial
glm.Sev.PP <- glm(
```

```
  formula = CM ~
    Año +
    PP_Tipo_Poliza +
    PP_Antiguedad +
    PP_Canal_Venta +
    #PP_Clase_Veh + (Se correlaciona con la MarcaVeh)
    #PP_Tipo_Veh +  (Se correlaciona con la MarcaVeh)
    PP_Uso_Veh +
    PP_Marca_Veh +
    #PP_Region + (Se correlaciona con la zona)
    PP_Zona +
    PP_Genero +
    #PP_Tipo_Persona + (Se correlaciona con la Edad)
    PP_Edad +
    PP_SA,
    #offset(log(Expuesto)), Se coloca este offset en caso que la variable dependiente hubiera sido solo el
      Incurrido
  family = Gamma(link = "log"),
  data = data_CM
)

# Desenmascarar posibles multicolinearidades entre las diferentes variables
temp <- alias(glm.Sev.PP)
temp2 <- as.data.frame(temp.Complete) # Si es vacio, no hay multicolinearidades
rm(temp,temp2)
# No se realizarán agrupaciones en los niveles de algunas variables por tener un buen modelo
# Incluimos las predicciones en la data
data_CM <- data_CM %>% mutate(CM.PRED = glm.Sev.PP.fitted.values)
data_CM %>% summarise(sum(Incurrido), sum(CM.PRED))
# Salvar un data frame con los coeficientes del modelo
coefs.Severidad.PP <- summary(glm.Sev.PP).coefficients %>%as_tibble() %>%mutate(var_level =glm.Sev.PP.
    coefficients %>% names()) %>% select(var_level, everything())
# Adicionar manualmente a la tabla los niveles atribuidos al intercept
# (y arredondar los valores)
temp <- tibble(
  var_level = c("Año2022","PP_Tipo_PolizaIndividual","PP_Antiguedadc. 02","PP_Canal_VentaCorredores","PP_Uso_
    VehParticular","PP_Marca_VehL3","PP_ZonaA2","PP_GeneroFemenino","PP_Edad38-42","PP_SAj.[13000-999999]"),
  Estimate = 0
)

f.round <- function(x, .digits = 4) round(x, digits = .digits)
coefs.Severidad.PP <- coefs.Severidad.PP %>% bind_rows(temp) %>% arrange(var_level) %>% mutate(`exp(Estimate)` 
    = exp(Estimate)) %>% mutate_if(is.double, f.round)

# Esta función considera los datos de entrenamiento del GLM, agrega la exposición por variable y
# la junta a la tabla de coeficientes.
incluir.Cantidad <- function(.data, .tbl.coefs){
  names.vars <- names(.data %>% select(-Cantidad))
  n <- length(names.vars)
  list.temp <- vector(mode = "list", length = n)
  for(i in seq_along(names.vars)){
    varname <- sym(names.vars[[i]])
    list.temp[[i]] <- .data %>%
      group_by(!!varname) %>%
      summarize(Incurrido = f.round(sum(Cantidad), .digits = 0)) %>%
      ungroup() %>%
      mutate(var_level = str_c(names.vars[[i]], !!varname)) %>%
      select(var_level, Incurrido)
  }
  exp.by.var <- bind_rows(list.temp)
  return(.tbl.coefs %>% left_join(exp.by.var))
}
coefs.Severidad.PP <- data_CM %>% select(-CM.PRED) %>%incluir.Cantidad(coefs.Severidad.PP)
coefs.Severidad.PP %>% DT::datatable()
# Guardamos los coeficientes en excel
write.csv(coefs.Severidad.PP,"C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/02_
    CostoMedio/04_Coeficientes/coefs.Severidad.PP.csv",fileEncoding = "Latin1")
save(glm.Sev.PP, file = "C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_
    Marginales/Perdida_Parcial/glm.Sev.PP.RData")
save(data_CM, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/
    Perdida_Parcial/Data_PP_model_CM.RData")
```

Código R: Modelo Frecuencia - Pédida Total

```
################################################################################
```

```
##########                    MODELO GLM - FRECUENCIA
################################################################################

# PAQUETES Y LIBRERÍAS
#options(repos = c(cran="http://cran.rstudio.com"))
#install.packages("RODBC")
#install.packages("skimr")
library(RODBC)
library(tidyverse) # Incluye dplyr y tidyr para manejar los datos y ggplot para representarlos en grÃ¡ficos
library(plotly)    # Para graficos interactivos
library(lubridate) # manejar datas y horas
library(forcats)
library(rcartocolor)
library(readr)      # para leer y importar archivos
library(skimr)

db<-"C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/01_Frecuencia/03_Data/BD_GLM.accdb"
con <- odbcConnectAccess2007(db)
data <- sqlQuery(con,"select * from PT_Datos_Frecuencia")
str(data)
data.`Número Póliza`<-format(data.`Número Póliza`,scientific = FALSE)
# Reducimos la data sin el año 2015 y 2020, de acuerdo a los análisis univariados realizados
data <- data %>% filter(Año!=2015 & Año!=2020)
data <- data %>% filter(PT_Clase_Veh!='Desconocido')
# Cambiamos el nombre de algunos niveles de algunas variables
data <- data %>% mutate(PT_Antiguedad = case_when(PT_Antiguedad == '0-1' ~ 'a. 0-1',PT_Antiguedad == '2' ~ 'b. 02',PT_Antiguedad == '3-5' ~ 'c. 03-05',PT_Antiguedad == '6' ~ 'd. 06',PT_Antiguedad == '7-16' ~ 'e. 07-16',PT_Antiguedad == '17+' ~ 'f. 17+'
))
data <- data %>% mutate(PT_SA = case_when(PT_SA == '[1000-10000>' ~ 'a.[1000-10000>',PT_SA == '[10000-49000>' ~ 'b.[10000-49000>',PT_SA == '[49000-88000]' ~ 'c.[49000-88000>',PT_SA == '[88000-999999]' ~ 'd.[88000-999999]'
))
# Convertimos a factor las variables
data.Año <- as.factor(data.Año)
data.PT_Tipo_Poliza <- as.factor(data.PT_Tipo_Poliza)
data.PT_Antiguedad <- as.factor(data.PT_Antiguedad)
data.PT_Canal_Venta <- as.factor(data.PT_Canal_Venta)
data.PT_Clase_Veh <- as.factor(data.PT_Clase_Veh)
data.PT_Tipo_Veh <- as.factor(data.PT_Tipo_Veh)
data.PT_Uso_Veh <- as.factor(data.PT_Uso_Veh)
data.PT_Marca_Veh <- as.factor(data.PT_Marca_Veh)
data.PT_Region <- as.factor(data.PT_Region)
data.PT_Zona <- as.factor(data.PT_Zona)
data.PT_Genero <- as.factor(data.PT_Genero)
data.PT_Tipo_Persona <- as.factor(data.PT_Tipo_Persona)
data.PT_Edad <- as.factor(data.PT_Edad)
data.PT_SA <- as.factor(data.PT_SA)
# Asignamos el intercepto para cada variable
data <- data %>% mutate(
  Año = fct_relevel(Año, "2022", after = 0),
  PT_Tipo_Poliza = fct_relevel(PT_Tipo_Poliza, "Individual", after = 0),
  PT_Antiguedad = fct_relevel(PT_Antiguedad, "c. 03-05", after = 0),
  PT_Canal_Venta = fct_relevel(PT_Canal_Venta, "Corredores", after = 0),
  PT_Clase_Veh = fct_relevel(PT_Clase_Veh, "Livianos", after = 0),
  PT_Tipo_Veh = fct_relevel(PT_Tipo_Veh, "Automovil", after = 0),
  PT_Uso_Veh = fct_relevel(PT_Uso_Veh, "Particular", after = 0) ,
  PT_Marca_Veh = fct_relevel(PT_Marca_Veh, "L4", after = 0),
  PT_Region = fct_relevel(PT_Region, "Lima", after = 0),
  PT_Zona = fct_relevel(PT_Zona, "A3", after = 0
  ),PT_Genero = fct_relevel(PT_Genero, "Femenino", after = 0),
  PT_Tipo_Persona = fct_relevel(PT_Tipo_Persona, "Empresa", after = 0),
  PT_Edad = fct_relevel(PT_Edad, "43-52", after = 0),
  PT_SA = fct_relevel(PT_SA, "b.[10000-49000>", after = 0)
)
# Verificación de frecuencia por factor
data %>% group_by(Año) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*100)
data %>% group_by(PT_Tipo_Poliza) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>%mutate(Freq=round(C/E,5)*100)
data %>% group_by(PT_Antiguedad) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>%mutate(Freq=round(C/E,5)*100)
data %>% group_by(PT_Canal_Venta) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>%mutate(Freq=round(C/E,5)*100)
data %>% group_by(PT_Clase_Veh) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>%mutate(Freq=round(C/E,5)*100)
data %>% group_by(PT_Tipo_Veh) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*
```

```
      100)
data %>% group_by(PT_Uso_Veh) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*
      100)
data %>% group_by(PT_Marca_Veh) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>%mutate(Freq=round(C/E,5)*
      100)
data %>% group_by(PT_Region) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*
      100)
data %>% group_by(PT_Zona) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*100)
data %>% group_by(PT_Genero) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*
      100)
data %>% group_by(PT_Tipo_Persona) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>%mutate(Freq=round(C/E
      ,5)*100)
data %>% group_by(PT_Edad) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*100)
data %>% group_by(PT_SA) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*100)

# Agrupaciones motivadas por modelos previos
data <- data %>%
  mutate(
    PT_Antiguedad=fct_collapse(PT_Antiguedad,"a. 0-02"=c("a. 0-1","b. 02"),"c. 06+"=c("d. 06","e. 07-16","f.
      17+")),
  PT_Canal_Venta=fct_collapse(PT_Canal_Venta,"Resto_Canales"=c("Agentes Exclusivos","Banca y Digital","Otros")
      ,"GrandesCuentas_Institucionales"=c("Grandes Cuentas","Instituciones")),
  PT_Edad=fct_collapse(PT_Edad,"33-42"=c("33-37","38-42"),"53-77"=c("53-57","58-62","63-77")),
  PT_Genero=fct_collapse(PT_Genero,"Masculino"=c("Indeterminado","Masculino")),
  PT_Marca_Veh=fct_collapse(PT_Marca_Veh,"P1_P2_P3_P4"=c("P1","P2","P3","P4")),
  PT_Uso_Veh=fct_collapse(PT_Uso_Veh,"Ambul_TranspIntProv_Carga_Tur_Esc"=c("Ambulancia","Transporte
      Interprovincial","Carga","Turismo","Escolar"),"Comercial_Taxi_TransPers"=c("Comercial","Taxi","Transporte
      Personal"),"Otros_Usos"=c("Instrucción","Serenazgo","Transporte Urbano")),
    PT_Zona=fct_collapse(PT_Zona,"A2_A5"=c("A2","A5"),"D2_D3"=c("D2","D3"))
  )
################################################################################
# Agrupamos la data para modelar el GLM
#data_model <- data %>% mutate(id=as.numeric(`Número Póliza`)*100+as.numeric(`Número Riesgo`)) %>% group_by(id
      ,Año,PP_Tipo_Poliza,PP_Antiguedad,PP_Canal_Venta,PP_Uso_Veh,PP_Marca_Veh,PP_Zona,PP_Genero,PP_Edad,PP_SA)
      %>% summarise(Expuesto=sum(Expuesto),Prima_Devengada=sum(Prima_Devengada),Cantidad=sum(Frecuencia),
      Incurrido=sum(Incurrido))
data_model <- data %>% group_by(Año,
                                PT_Tipo_Poliza,
                                #PT_Antiguedad,
                                #PT_Canal_Venta,
                                PT_Uso_Veh,
                                PT_Marca_Veh,
                                PT_Zona,
                                PT_Genero,
                                PT_Edad
                                #PP_SA
                                )%>% summarise(Expuesto=sum(Expuesto),Prima_Devengada=sum(Prima_Devengada),
      Cantidad=sum(Frecuencia),Incurrido=sum(Incurrido))


# Modelo GLM Pérdida Parcial
glm.Frec.PT <- glm(
  formula = Cantidad ~
    Año +
    PT_Tipo_Poliza +
    #PT_Antiguedad + (No significativa)
    #PT_Canal_Venta + (No significativa)
    #PP_Clase_Veh + (Se correlaciona con la MarcaVeh)
    #PP_Tipo_Veh +  (Se correlaciona con la MarcaVeh)
    PT_Uso_Veh +
    PT_Marca_Veh +
    #PP_Region + (Se correlaciona con la zona)
    PT_Zona +
    PT_Genero +
    #PP_Tipo_Persona + (Se correlaciona con la Edad)
    PT_Edad +
    #PT_SA + (No significativa)
    offset(log(Expuesto)),
  family = poisson(link = "log"),
  data = data_model
)
# Desenmascarar posibles multicolinearidades entre las diferentes variables
temp <- alias(glm.Frec.PT)
temp2 <- as.data.frame(temp.Complete) # Si es vacio, no hay multicolinearidades
rm(temp,temp2)
# No se realizarán agrupaciones en los niveles de algunas variables por tener un buen modelo
```

```r
# Incluimos las predicciones en la data
data <- data %>% mutate(Frecuencia.PRED = glm.Frec.PT.fitted.values)
data %>% summarise(sum(Frecuencia), sum(Frecuencia.PRED))
# Salvar un data frame con los coeficientes del modelo
coefs.Frecuencia.PT <- summary(glm.Frec.PT).coefficients %>%
  as_tibble() %>%mutate(var_level = glm.Frec.PT.coefficients %>% names()) %>% select(var_level, everything())
# Adicionar manualmente a la tabla los niveles atribuidos al intercept
# (y arredondar los valores)
temp <- tibble(
  var_level = c("Año2022","PT_Tipo_PolizaIndividual",#"PT_Antiguedadc. 03-05","PT_Uso_VehParticular","PT_Marca_VehL4","PT_ZonaA3","PT_GeneroFemenino","PT_Edad43-52"),
  Estimate = 0
)

f.round <- function(x, .digits = 4) round(x, digits = .digits)
coefs.Frecuencia.PT <- coefs.Frecuencia.PT %>% bind_rows(temp) %>%
  arrange(var_level) %>% mutate(`exp(Estimate)` = exp(Estimate)) %>% mutate_if(is.double, f.round)
# Esta función considera los datos de entrenamiento del GLM, agrega la exposición por variable # la junta a la tabla de coeficientes.
incluir.exposicion <- function(.data, .tbl.coefs){
  names.vars <- names(.data %>% select(-Expuesto))
  n <- length(names.vars)
  list.temp <- vector(mode = "list", length = n)
  for(i in seq_along(names.vars)){
    varname <- sym(names.vars[[i]])
    list.temp[[i]] <- .data %>%
      group_by(!!varname) %>%
      summarize(Exposición = f.round(sum(Expuesto), .digits = 0)) %>%
      ungroup() %>%
      mutate(var_level = str_c(names.vars[[i]], !!varname)) %>%
      select(var_level, Exposición)
  }
  exp.by.var <- bind_rows(list.temp)
  return(.tbl.coefs %>% left_join(exp.by.var))
}
coefs.Frecuencia.PT <- data %>% select(-Frecuencia, -Frecuencia.PRED) %>%incluir.exposicion(coefs.Frecuencia.PT)
coefs.Frecuencia.PT %>% DT::datatable()
# Guardamos los coeficientes en excel
write.csv(coefs.Frecuencia.PT,"C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/01_Frecuencia/04_Coeficientes/coefs.Frecuencia.PT.csv",fileEncoding = "Latin1")
save(glm.Frec.PT, file = "C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Perdida_Total/glm.Frec.PT.RData")
save(data, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Perdida_Total/Data_PT.RData")
save(data_model, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Perdida_Total/Data_PT_model.RData")
```

Código R: Modelo Severidad - Pédida Total

```r
################################################################################
##########                    MODELO GLM - SEVERIDAD
################################################################################
# PAQUETES Y LIBRERÍAS
#options(repos = c(cran="http://cran.rstudio.com"))
#install.packages("RODBC")
#install.packages("skimr")
library(RODBC)
library(tidyverse) # Incluye dplyr y tidyr para manejar los datos y ggplot para representarlos en grÃ¡ficos
library(plotly)    # Para graficos interactivos
library(lubridate) # manejar datas y horas
library(forcats)
library(rcartocolor)
library(readr)      # para leer y importar archivos
library(skimr)
#db2<-"C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/02_CostoMedio/03_Data/BD_GLM_CM.accdb"
#con2 <- odbcConnectAccess2007(db2)
#data_CM <- sqlQuery(con2,"select * from PT_Datos_Severidad")
#str(data_CM)
# Reducimos la data sin el año 2015 y 2020, de acuerdo a los análisis univariados realizados
#data_CM <- data_CM %>% filter(Año!=2015 & Año!=2020)
#data_CM <- data_CM %>% filter(PT_Clase_Veh!='Desconocido')
#data_CM <- data_CM %>% filter(Incurrido>0)
#data_CM.`Número Póliza`<-format(data_CM.`Número Póliza`,scientific = FALSE)
```

```
data_CM <- data_model %>% filter(Incurrido>0) # onitir todas las acciones y saltar al modelo GLM
# Cambiamos el nombre de algunos niveles de algunas variables
data_CM <- data_CM %>% mutate(PT_Antiguedad = case_when(
  PT_Antiguedad == '0-1' ~ 'a. 0-1',PT_Antiguedad == '2' ~ 'b. 02',PT_Antiguedad == '3-5' ~ 'c. 03-05',PT_
      Antiguedad == '6' ~ 'd. 06',PT_Antiguedad == '7-16' ~ 'e. 07-16',PT_Antiguedad == '17+' ~ 'f. 17+'
))
data_CM <- data_CM %>% mutate(PT_SA = case_when(PT_SA == '[1000-10000>' ~ 'a.[1000-10000>',PT_SA == '
      [10000-49000>' ~ 'b.[10000-49000>',PT_SA == '[49000-88000]' ~ 'c.[49000-88000>',PT_SA == '[88000-999999]'
       ~ 'd.[88000-999999]'
))
# Convertimos a factor las variables
data_CM.Año <- as.factor(data_CM.Año)
data_CM.PT_Tipo_Poliza <- as.factor(data_CM.PT_Tipo_Poliza)
data_CM.PT_Antiguedad <- as.factor(data_CM.PT_Antiguedad)
data_CM.PT_Canal_Venta <- as.factor(data_CM.PT_Canal_Venta)
data_CM.PT_Clase_Veh <- as.factor(data_CM.PT_Clase_Veh)
data_CM.PT_Tipo_Veh <- as.factor(data_CM.PT_Tipo_Veh)
data_CM.PT_Uso_Veh <- as.factor(data_CM.PT_Uso_Veh)
data_CM.PT_Marca_Veh <- as.factor(data_CM.PT_Marca_Veh)
data_CM.PT_Region <- as.factor(data_CM.PT_Region)
data_CM.PT_Zona <- as.factor(data_CM.PT_Zona)
data_CM.PT_Genero <- as.factor(data_CM.PT_Genero)
data_CM.PT_Tipo_Persona <- as.factor(data_CM.PT_Tipo_Persona)
data_CM.PT_Edad <- as.factor(data_CM.PT_Edad)
data_CM.PT_SA <- as.factor(data_CM.PT_SA)
# Asignamos el intercepto para cada variable
data_CM <- data_CM %>% mutate(
  Año = fct_relevel(Año, "2022", after = 0),
  PT_Tipo_Poliza = fct_relevel(PT_Tipo_Poliza, "Individual", after = 0),
  PT_Antiguedad = fct_relevel( PT_Antiguedad, "c. 03-05", after = 0),
  PT_Canal_Venta = fct_relevel(    PT_Canal_Venta, "Corredores", after = 0),
  PT_Clase_Veh = fct_relevel(PT_Clase_Veh, "Livianos", after = 0),
  PT_Tipo_Veh = fct_relevel(PT_Tipo_Veh, "Automovil", after = 0),PT_Uso_Veh, "Particular", after = 0) ,
  PT_Marca_Veh = fct_relevel(PT_Marca_Veh, "L4", after = 0),
  PT_Region = fct_relevel(PT_Region, "Lima", after = 0),
  PT_Zona = fct_relevel(PT_Zona, "A3", after = 0),
  PT_Genero = fct_relevel(PT_Genero, "Femenino", after = 0),
  PT_Tipo_Persona = fct_relevel(PT_Tipo_Persona,"Empresa", after = 0),
  PT_Edad = fct_relevel(PT_Edad, "43-52", after = 0),
  PT_SA = fct_relevel(PT_SA, "b.[10000-49000>", after = 0)
)
# Verificación de frecuencia por factor
data_CM %>% group_by(Año) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(PT_Tipo_Poliza) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(PT_Antiguedad) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(PT_Canal_Venta) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(PT_Clase_Veh) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(PT_Tipo_Veh) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(PT_Uso_Veh) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(PT_Marca_Veh) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(PT_Region) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(PT_Zona) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(PT_Genero) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(PT_Tipo_Persona) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(PT_Edad) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(PT_SA) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
# Agrupaciones motivadas por modelos previos
data_CM <- data_CM %>%
  mutate(
    PT_Antiguedad=fct_collapse(PT_Antiguedad,"a. 0-02"=c("a. 0-1","b. 02"),"c. 06+"=c("d. 06","e. 07-16","f.
      17+")),
    PT_Canal_Venta=fct_collapse(PT_Canal_Venta,"Resto_Canales"=c("Agentes Exclusivos","Banca yDigital","Otros"
      ),"GrandesCuentas_Institucionales"=c("Grandes Cuentas","Instituciones")),
    PT_Edad=fct_collapse(PT_Edad,"33-42"=c("33-37","38-42"),"53-77"=c("53-57","58-62","63-77")),
    PT_Genero=fct_collapse(PT_Genero,"Masculino"=c("Indeterminado","Masculino")),
    PT_Marca_Veh=fct_collapse(PT_Marca_Veh,"P1_P2_P3_P4"=c("P1","P2","P3","P4")),
    PT_Uso_Veh=fct_collapse(PT_Uso_Veh,"Ambul_TranspIntProv_Carga_Tur_Esc"=c("Ambulancia","Transporte
      Interprovincial","Carga","Turismo","Escolar"),"Comercial_Taxi_TransPers"=c("Comercial","Taxi","Transporte
       Personal"),
      "Otros_Usos"=c("Instrucción","Serenazgo","Transporte Urbano")
    ),
    PT_Zona=fct_collapse(PT_Zona,"A2_A5"=c("A2","A5"),"D2_D3"=c("D2","D3"))
  )
# Creamos las variable dependiente
data_CM <-data_CM %>% mutate(CM=Incurrido/Cantidad)
# Modelo GLM Pérdida Parcial
```

```
glm.Sev.PT <- glm(
  formula = CM ~
    Año +
    PT_Tipo_Poliza +
    #PT_Antiguedad +
    #PT_Canal_Venta +
    #PT_Clase_Veh + (Se correlaciona con la MarcaVeh)
    #PT_Tipo_Veh +  (Se correlaciona con la MarcaVeh)
    PT_Uso_Veh +
    PT_Marca_Veh +
    #PT_Region + (Se correlaciona con la zona)
    PT_Zona +
    PT_Genero +
    #PT_Tipo_Persona + (Se correlaciona con la Edad)
    PT_Edad,
    #PP_SA,
  #offset(log(Expuesto)), Se coloca este offset en caso que la variable dependiente hubiera sido solo el
      Incurrido
  family = Gamma(link = "log"),
  data = data_CM
)
# Desenmascarar posibles multicolinearidades entre las diferentes variables
temp <- alias(glm.Sev.PT)
temp2 <- as.data.frame(temp.Complete) # Si es vacio, no hay multicolinearidades
rm(temp,temp2)
# No se realizarán agrupaciones en los niveles de algunas variables por tener un buen modelo
# Incluimos las predicciones en la data
data_CM <- data_CM %>% mutate(CM.PRED = glm.Sev.PT.fitted.values)data_CM %>% summarise(sum(Incurrido),sum(CM.
    PRED*Cantidad))
# Salvar un data frame con los coeficientes del modelo
coefs.Severidad.PT <- summary(glm.Sev.PT).coefficients %>as_tibble() %>%mutate(var_level = glm.Sev.PT.
    coefficients %>% names()) %>% select(var_level, everything())
# Adicionar manualmente a la tabla los niveles atribuidos al intercept
# (y arredondar los valores)
temp <- tibble(
  var_level = c("Año2022","PT_Tipo_PolizaIndividual",#"PT_Antiguedadc. 03-05","PT_Uso_VehParticular","PT_Marca
      _VehL4","PT_ZonaA3","PT_GeneroFemenino","PT_Edad43-52"
  ),
  Estimate = 0
)
f.round <- function(x, .digits = 4) round(x, digits = .digits)
coefs.Severidad.PT <- coefs.Severidad.PT %>%
  bind_rows(temp) %>%
  arrange(var_level) %>%
  mutate(`exp(Estimate)` = exp(Estimate)) %>%
  mutate_if(is.double, f.round)
# Esta función considera los datos de entrenamiento del GLM, agrega la exposición por variable y
# la junta a la tabla de coeficientes.
incluir.Cantidad <- function(.data, .tbl.coefs){
  names.vars <- names(.data %>% select(-Cantidad))
  n <- length(names.vars)
  list.temp <- vector(mode = "list", length = n)
  for(i in seq_along(names.vars)){
    varname <- sym(names.vars[[i]])
    list.temp[[i]] <- .data %>%
      group_by(!!varname) %>%
      summarize(Incurrido = f.round(sum(Cantidad), .digits = 0)) %>%
      ungroup() %>%
      mutate(var_level = str_c(names.vars[[i]], !!varname)) %>%
      select(var_level, Incurrido)
  }
  exp.by.var <- bind_rows(list.temp)
  return(.tbl.coefs %>% left_join(exp.by.var))
}
coefs.Severidad.PT <- data_CM %>%
  select(-CM.PRED) %>%
  incluir.Cantidad(coefs.Severidad.PT)
coefs.Severidad.PT %>% DT::datatable()
# Guardamos los coeficientes en excel
write.csv(coefs.Severidad.PT,"C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/02_
    CostoMedio/04_Coeficientes/coefs.Severidad.PT.csv",fileEncoding = "Latin1")
save(glm.Sev.PT, file = "C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_
    Marginales/Perdida_Total/glm.Sev.PT.RData")
save(data_CM, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/
    Perdida_Total/Data_PT_model_CM.RData")
```

Código R: Modelo Frecuencia - Responsabilidad Civil

```
################################################################################
##########                  MODELO GLM - FRECUENCIA
################################################################################
# PAQUETES Y LIBRERÍAS
#options(repos = c(cran="http://cran.rstudio.com"))
library(RODBC)
library(tidyverse) # Incluye dplyr y tidyr para manejar los datos y ggplot para representarlos en grÃ¡ficos
library(plotly)    # Para graficos interactivos
library(lubridate) # manejar datas y horas
library(forcats)
library(rcartocolor)
library(readr)      # para leer y importar archivos
library(skimr)
db<-"C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/01_Frecuencia/03_Data/BD_GLM.accdb"
con <- odbcConnectAccess2007(db)
data <- sqlQuery(con,"select * from RC_Datos_Frecuencia")
str(data)
data.`Número Póliza`<-format(data.`Número Póliza`,scientific = FALSE)
# Reducimos la data sin el año 2015 y 2020, de acuerdo a los análisis univariados realizados
data <- data %>% filter(Año!=2015 & Año!=2020)
data <- data %>% filter(RC_Clase_Veh!='Desconocido')
# Cambiamos el nombre de algunos niveles de algunas variables
data <- data %>% mutate(RC_Antiguedad = case_when(RC_Antiguedad == '0-1' ~ 'a. 0-1',RC_Antiguedad == '2-9' ~ '
    b. 2-9',RC_Antiguedad == '10-13' ~ 'c. 10-13',RC_Antiguedad == '14+' ~ 'd. 14+'
))
data <- data %>% mutate(RC_SA = case_when(RC_SA == '[1000-25000>' ~ 'a.[1000-25000>',RC_SA == '[25000-105000>'
     ~ 'b.[25000-105000>',RC_SA == '[105000-145000]' ~ 'c.[105000-145000>',RC_SA == '[145000-999999]' ~ 'd
    .[145000-999999]'
))

# Convertimos a factor las variables
data.Año <- as.factor(data.Año)
data.RC_Tipo_Poliza <- as.factor(data.RC_Tipo_Poliza)
data.RC_Antiguedad <- as.factor(data.RC_Antiguedad)
data.RC_Canal_Venta <- as.factor(data.RC_Canal_Venta)
data.RC_Clase_Veh <- as.factor(data.RC_Clase_Veh)
data.RC_Tipo_Veh <- as.factor(data.RC_Tipo_Veh)
data.RC_Uso_Veh <- as.factor(data.RC_Uso_Veh)
data.RC_Marca_Veh <- as.factor(data.RC_Marca_Veh)
data.RC_Region <- as.factor(data.RC_Region)
data.RC_Zona <- as.factor(data.RC_Zona)
data.RC_Genero <- as.factor(data.RC_Genero)
data.RC_Tipo_Persona <- as.factor(data.RC_Tipo_Persona)
data.RC_Edad <- as.factor(data.RC_Edad)
data.RC_SA <- as.factor(data.RC_SA)
# Asignamos el intercepto para cada variable
data <- data %>% mutate(
  Año = fct_relevel(Año, "2022", after = 0),
  RC_Tipo_Poliza = fct_relevel(RC_Tipo_Poliza, "Individual", after = 0),
  RC_Antiguedad = fct_relevel(RC_Antiguedad, "b. 2-9", after = 0),
  RC_Canal_Venta = fct_relevel(RC_Canal_Venta, "Corredores", after = 0),
  RC_Clase_Veh = fct_relevel(RC_Clase_Veh, "Livianos", after = 0),
  RC_Tipo_Veh = fct_relevel(RC_Tipo_Veh, "Automovil", after = 0),
  RC_Uso_Veh = fct_relevel(RC_Uso_Veh, "Particular", after = 0) ,
  RC_Marca_Veh = fct_relevel(RC_Marca_Veh, "L3", after = 0),
  RC_Region = fct_relevel(RC_Region, "Lima", after = 0),
  RC_Zona = fct_relevel(RC_Zona, "A2", after = 0),
  RC_Genero = fct_relevel(RC_Genero, "Femenino", after = 0),
  RC_Tipo_Persona = fct_relevel(RC_Tipo_Persona, "Empresa", after = 0),
  RC_Edad = fct_relevel(RC_Edad, "43-52", after = 0),
  RC_SA = fct_relevel(RC_SA, "a.[1000-25000>", after = 0)
)
# Verificación de frecuencia por factor
data %>% group_by(Año) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*100)
data %>% group_by(RC_Tipo_Poliza) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>%mutate(Freq=round(C/E
    ,5)*100)
data %>% group_by(RC_Antiguedad) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>%mutate(Freq=round(C/E,5)
    *100)
data %>% group_by(RC_Canal_Venta) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>%mutate(Freq=round(C/E
    ,5)*100)
data %>% group_by(RC_Clase_Veh) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>%mutate(Freq=round(C/E,5)*
    100)
data %>% group_by(RC_Tipo_Veh) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*
    100)
```

```
data %>% group_by(RC_Uso_Veh) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*100)
data %>% group_by(RC_Marca_Veh) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>%mutate(Freq=round(C/E,5)*100)
data %>% group_by(RC_Region) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*100)
data %>% group_by(RC_Zona) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*100)
data %>% group_by(RC_Genero) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*100)
data %>% group_by(RC_Tipo_Persona) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>%mutate(Freq=round(C/E,5)*100)
data %>% group_by(RC_Edad) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*100)
data %>% group_by(RC_SA) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*100)
# Agrupaciones motivadas por modelos previos
data <- data %>%
  mutate(
    RC_Antiguedad=fct_collapse(RC_Antiguedad,"c. 10+"=c("c. 10-13","d. 14+")),
    RC_Canal_Venta=fct_collapse(RC_Canal_Venta,"GrandesCuentas__AgentesExc"=c("Agentes Exclusivos","Grandes Cuentas")),
    RC_Edad=fct_collapse(RC_Edad,"18-42"=c("18-27","28-32","33-37","38-42"),"53+"=c("53-57","58-62","63-67","73-77","68-72","73-77","78-82","83-87","88+")),
    RC_Genero=fct_collapse(RC_Genero,"Masculino"=c("Indeterminado","Masculino")),
    RC_Marca_Veh=fct_collapse(RC_Marca_Veh,"M1_M2"=c("M1","M2"),"P1_P2"=c("P1","P2")),
    RC_SA=fct_collapse(RC_SA,"b.[25000-999999]"=c("b.[25000-105000>","c.[105000-145000>","d.[145000-999999]")),
    RC_Uso_Veh=fct_collapse(RC_Uso_Veh,"Alq_Amb_Carg_Com"=c("Alquiler","Ambulancia","Carga","Comercial"),
      "TransInterProv_Urb"=c("Transporte Interprovincial","Transporte Urbano")
    )
  )
##############################################################################################
# Agrupamos la data para modelar el GLM
#data_model <- data %>% mutate(id=as.numeric(`Número Póliza`)*100+as.numeric(`Número Riesgo`)) %>% group_by(id,Año,PP_Tipo_Poliza,PP_Antiguedad,PP_Canal_Venta,PP_Uso_Veh,PP_Marca_Veh,PP_Zona,PP_Genero,PP_Edad,PP_SA) %>%summarise(Expuesto=sum(Expuesto),Prima_Devengada=sum(Prima_Devengada),Cantidad=sum(Frecuencia),Incurrido=sum(Incurrido))
data_model <- data %>% group_by(Año,
                                #PT_Tipo_Poliza,
                                RC_Antiguedad,
                                RC_Canal_Venta,
                                RC_Uso_Veh,
                                RC_Marca_Veh,
                                RC_Zona,
                                #PT_Genero,
                                RC_Edad
                                #PP_SA
)%>% summarise(Expuesto=sum(Expuesto),Prima_Devengada=sum(Prima_Devengada),Cantidad=sum(Frecuencia),Incurrido=sum(Incurrido))
# Modelo GLM Pérdida Parcial
glm.Frec.RC <- glm(
  formula = Cantidad ~
    Año +
    #RC_Tipo_Poliza + (no siginificativo)
    RC_Antiguedad + #(No significativa)
    RC_Canal_Venta + #(No significativa)
    #RC_Clase_Veh + (Se correlaciona con la MarcaVeh)
    #RC_Tipo_Veh +  (Se correlaciona con la MarcaVeh)
    RC_Uso_Veh +
    RC_Marca_Veh +
    #RC_Region + (Se correlaciona con la zona)
    RC_Zona +
    #RC_Genero + (no siginifcativo)
    #RC_Tipo_Persona + (Se correlaciona con la Edad)
    RC_Edad +
    #RC_SA + #(No significativa)
    offset(log(Expuesto)),
  family = poisson(link = "log"),
  data = data_model
)

# Desenmascarar posibles multicolinearidades entre las diferentes variables
temp <- alias(glm.Frec.RC)
temp2 <- as.data.frame(temp.Complete) # Si es vacio, no hay multicolinearidades
rm(temp,temp2)
# No se realizarán agrupaciones en los niveles de algunas variables por tener un buen modelo
# Incluimos las predicciones en la data
data <- data %>% mutate(Frecuencia.PRED = glm.Frec.RC.fitted.values)
```

```r
data %>% summarise(sum(Frecuencia), sum(Frecuencia.PRED))
# Salvar un data frame con los coeficientes del modelo
coefs.Frecuencia.RC <- summary(glm.Frec.RC).coefficients %>%as_tibble() %>%mutate(var_level =glm.Frec.RC.coefficients %>% names()) %>% select(var_level, everything())
# Adicionar manualmente a la tabla los niveles atribuidos al intercept
# (y arredondar los valores)
temp <- tibble(
  var_level = c("Año2022","RC_Antiguedadb. 2-9","RC_Canal_VentaCorredores","RC_Uso_VehParticular","RC_Marca_VehL3","RC_ZonaA2","RC_Edad43-52"
  ),
  Estimate = 0
)
f.round <- function(x, .digits = 4) round(x, digits = .digits)
coefs.Frecuencia.RC <- coefs.Frecuencia.RC %>% bind_rows(temp) %>% arrange(var_level) %>% mutate(`exp(Estimate)` = exp(Estimate)) %>% mutate_if(is.double, f.round)
# Esta función considera los datos de entrenamiento del GLM, agrega la exposición por variable y
# la junta a la tabla de coeficientes.
incluir.exposicion <- function(.data, .tbl.coefs){
  names.vars <- names(.data %>% select(-Expuesto))
  n <- length(names.vars)
  list.temp <- vector(mode = "list", length = n)
  for(i in seq_along(names.vars)){
    varname <- sym(names.vars[[i]])
    list.temp[[i]] <- .data %>%
      group_by(!!varname) %>%
      summarize(Exposición = f.round(sum(Expuesto), .digits = 0)) %>%
      ungroup() %>%
      mutate(var_level = str_c(names.vars[[i]], !!varname)) %>%
      select(var_level, Exposición)
  }
  exp.by.var <- bind_rows(list.temp)
  return(.tbl.coefs %>% left_join(exp.by.var))

}
coefs.Frecuencia.RC <- data %>%
  select(-Frecuencia, -Frecuencia.PRED) %>%incluir.exposicion(coefs.Frecuencia.RC)
coefs.Frecuencia.RC %>% DT::datatable()
# Guardamos los coeficientes en excel
write.csv(coefs.Frecuencia.RC,"C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/01_Frecuencia/04_Coeficientes/coefs.Frecuencia.RC.csv",fileEncoding ="Latin1")
save(glm.Frec.RC, file = "C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Responsabilidad_Civil/glm.Frec.RC.RData")
save(data, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Responsabilidad_Civil/Data_RC.RData")
save(data_model, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Responsabilidad_Civil/Data_RC_model.RData")
```

Código R: Modelo severidad - Responsabilidad Civil

```r
################################################################################
##########                  MODELO GLM - SEVERIDAD
################################################################################
# PAQUETES Y LIBRERÍAS
#options(repos = c(cran="http://cran.rstudio.com"))
#install.packages("RODBC")
#install.packages("skimr")
library(RODBC)
library(tidyverse) # Incluye dplyr y tidyr para manejar los datos y ggplot para representarlos en grÃ¡ficos
library(plotly)    # Para graficos interactivos
library(lubridate) # manejar datas y horas
library(forcats)
library(rcartocolor)
library(readr)      # para leer y importar archivos
library(skimr)
#db2<-"C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/02_CostoMedio/03_Data/BD_GLM_CM.accdb"
#con2 <- odbcConnectAccess2007(db2)
#data_CM <- sqlQuery(con2,"select * from RC_Datos_Severidad")
#str(data_CM)
# Reducimos la data sin el año 2015 y 2020, de acuerdo a los análisis univariados realizados
#data_CM <- data_CM %>% filter(Año!=2015 & Año!=2020)
#data_CM <- data_CM %>% filter(RC_Clase_Veh!='Desconocido')
#data_CM <- data_CM %>% filter(Incurrido>0)
#data_CM.`Número Póliza`<-format(data_CM.`Número Póliza`,scientific = FALSE)
data_CM <- data_model %>% filter(Incurrido>0) # onitir todas las acciones y saltar al modelo GLM
```

```r
# Cambiamos el nombre de algunos niveles de algunas variables
data_CM <- data_CM %>% mutate(RC_Antiguedad = case_when(RC_Antiguedad == '0-1' ~ 'a. 0-1',RC_Antiguedad == '2-9' ~ 'b. 2-9',RC_Antiguedad == '10-13' ~ 'c. 10-13',RC_Antiguedad == '14+' ~ 'd. 14+'
))
data_CM <- data_CM %>% mutate(RC_SA = case_when(RC_SA == '[1000-25000>' ~ 'a.[1000-25000>',RC_SA == '[25000-105000>' ~ 'b.[25000-105000>',RC_SA == '[105000-145000]' ~ 'c.[105000-145000>',RC_SA == '[145000-999999]' ~ 'd.[145000-999999]'
))
# Convertimos a factor las variables
data_CM.Año <- as.factor(data_CM.Año)
data_CM.RC_Tipo_Poliza <- as.factor(data_CM.RC_Tipo_Poliza)
data_CM.RC_Antiguedad <- as.factor(data_CM.RC_Antiguedad)
data_CM.RC_Canal_Venta <- as.factor(data_CM.RC_Canal_Venta)
data_CM.RC_Clase_Veh <- as.factor(data_CM.RC_Clase_Veh)
data_CM.RC_Tipo_Veh <- as.factor(data_CM.RC_Tipo_Veh)
data_CM.RC_Uso_Veh <- as.factor(data_CM.RC_Uso_Veh)
data_CM.RC_Marca_Veh <- as.factor(data_CM.RC_Marca_Veh)
data_CM.RC_Region <- as.factor(data_CM.RC_Region)
data_CM.RC_Zona <- as.factor(data_CM.RC_Zona)
data_CM.RC_Genero <- as.factor(data_CM.RC_Genero)
data_CM.RC_Tipo_Persona <- as.factor(data_CM.RC_Tipo_Persona)
data_CM.RC_Edad <- as.factor(data_CM.RC_Edad)
data_CM.RC_SA <- as.factor(data_CM.RC_SA)
# Asignamos el intercepto para cada variable
data_CM <- data_CM %>% mutate(
  Año = fct_relevel(Año, "2022", after = 0),
  RC_Tipo_Poliza = fct_relevel(RC_Tipo_Poliza, "Individual", after = 0),
  RC_Antiguedad = fct_relevel(RC_Antiguedad, "b. 2-9", after = 0),
  RC_Canal_Venta = fct_relevel(RC_Canal_Venta, "Corredores", after = 0),
  RC_Clase_Veh = fct_relevel(RC_Clase_Veh, "Livianos", after = 0),
  RC_Tipo_Veh = fct_relevel(RC_Tipo_Veh, "Automovil", after = 0),
  RC_Uso_Veh = fct_relevel(RC_Uso_Veh, "Particular", after = 0) ,
  RC_Marca_Veh = fct_relevel(RC_Marca_Veh, "L3", after = 0),
  RC_Region = fct_relevel(RC_Region, "Lima", after = 0),
  RC_Zona = fct_relevel(RC_Zona, "A2", after = 0),
  RC_Genero = fct_relevel(RC_Genero, "Femenino", after = 0),
  RC_Tipo_Persona = fct_relevel(RC_Tipo_Persona, "Empresa", after = 0),
  RC_Edad = fct_relevel(RC_Edad, "43-52", after = 0),
  RC_SA = fct_relevel(RC_SA, "a.[1000-25000>", after = 0)
)
# Verificación de frecuencia por factor
data_CM %>% group_by(Año) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(RC_Tipo_Poliza) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(RC_Antiguedad) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(RC_Canal_Venta) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(RC_Clase_Veh) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(RC_Tipo_Veh) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(RC_Uso_Veh) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(RC_Marca_Veh) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(RC_Region) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(RC_Zona) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(RC_Genero) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(RC_Tipo_Persona) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(RC_Edad) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(RC_SA) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
# Agrupaciones motivadas por modelos previos
data_CM <- data_CM %>%
  mutate(
    RC_Antiguedad=fct_collapse(RC_Antiguedad,"c. 10+"=c("c. 10-13","d. 14+")),
    RC_Canal_Venta=fct_collapse(RC_Canal_Venta,"GrandesCuentas__AgentesExc"=c("Agentes Exclusivos","Grandes Cuentas")),
    RC_Edad=fct_collapse(RC_Edad,"18-42"=c("18-27","28-32","33-37","38-42"),"53+"=c("53-57","58-62","63-67","73-77","68-72","73-77","78-82","83-87","88+")),
    RC_Genero=fct_collapse(RC_Genero,"Masculino"=c("Indeterminado","Masculino")),
    RC_Marca_Veh=fct_collapse(RC_Marca_Veh,"M1_M2"=c("M1","M2"),"P1_P2"=c("P1","P2")),
    RC_SA=fct_collapse(RC_SA,"b.[25000-999999]"=c("b.[25000-105000>","c.[105000-145000>","d.[145000-999999]"),
    RC_Uso_Veh=fct_collapse(RC_Uso_Veh,"Alq_Amb_Carg_Com"=c("Alquiler","Ambulancia","Carga","Comercial"),"TransInterProv_Urb"=c("Transporte Interprovincial","Transporte Urbano"))
  )
# Creamos las variable dependiente
data_CM <-data_CM %>% mutate(CM=Incurrido/Cantidad)
# Modelo GLM Pérdida Parcial
glm.Sev.RC <- glm(
  formula = CM ~
    Año +
```

```
    #RC_Tipo_Poliza +
    RC_Antiguedad +
    RC_Canal_Venta +
    #RC_Clase_Veh + (Se correlaciona con la MarcaVeh)
    #RC_Tipo_Veh +  (Se correlaciona con la MarcaVeh)
    RC_Uso_Veh +
    RC_Marca_Veh +
    #RC_Region + (Se correlaciona con la zona)
    RC_Zona +
    #RC_Genero +
    #RC_Tipo_Persona + (Se correlaciona con la Edad)
    RC_Edad,
    #RC_SA,
  #offset(log(Expuesto)), Se coloca este offset en caso que la variable dependiente hubiera sido solo el
      Incurrido
  family = Gamma(link = "log"),
  data = data_CM
)
# Desenmascarar posibles multicolinearidades entre las diferentes variables
temp <- alias(glm.Sev.RC)
temp2 <- as.data.frame(temp.Complete) # Si es vacio, no hay multicolinearidades
rm(temp,temp2)
# No se realizarán agrupaciones en los niveles de algunas variables por tener un buen modelo
# Incluimos las predicciones en la data
data_CM <- data_CM %>% mutate(CM.PRED = glm.Sev.RC.fitted.values)
data_CM %>% summarise(sum(Incurrido), sum(CM.PRED*Cantidad))
# Salvar un data frame con los coeficientes del modelo
coefs.Severidad.RC <- summary(glm.Sev.RC).coefficients %>%
  as_tibble() %>%
  mutate(var_level = glm.Sev.RC.coefficients %>% names()) %>%
  select(var_level, everything())
# Adicionar manualmente a la tabla los niveles atribuidos al intercept
# (y arredondar los valores)
temp <- tibble(
  var_level = c("Año2022","RC_Antiguedadb. 2-9","RC_Canal_VentaCorredores","RC_Uso_VehParticular","RC_Marca_
      VehL3","RC_ZonaA2","RC_Edad43-52"),
  Estimate = 0
)
f.round <- function(x, .digits = 4) round(x, digits = .digits)
coefs.Severidad.RC <- coefs.Severidad.RC %>% bind_rows(temp) %>% arrange(var_level) %>% mutate(`exp(Estimate)` 
      = exp(Estimate)) %>% mutate_if(is.double, f.round)
# Esta función considera los datos de entrenamiento del GLM, agrega la exposición por variable y
# la junta a la tabla de coeficientes.
incluir.Cantidad <- function(.data, .tbl.coefs){
  names.vars <- names(.data %>% select(-Cantidad))
  n <- length(names.vars)
  list.temp <- vector(mode = "list", length = n)
  for(i in seq_along(names.vars)){
    varname <- sym(names.vars[[i]])
    list.temp[[i]] <- .data %>%
      group_by(!!varname) %>%
      summarize(Incurrido = f.round(sum(Cantidad), .digits = 0)) %>%
      ungroup() %>%
      mutate(var_level = str_c(names.vars[[i]], !!varname)) %>%
      select(var_level, Incurrido)
  }
  exp.by.var <- bind_rows(list.temp)
  return(.tbl.coefs %>% left_join(exp.by.var))
}
coefs.Severidad.RC <- data_CM %>%
  select(-CM.PRED) %>%
  incluir.Cantidad(coefs.Severidad.RC)
coefs.Severidad.RC %>% DT::datatable()
# Guardamos los coeficientes en excel
write.csv(coefs.Severidad.RC,"C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/02_
    CostoMedio/04_Coeficientes/coefs.Severidad.RC.csv",fileEncoding = "Latin1")
save(glm.Sev.RC, file = "C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_
    Marginales/Responsabilidad_Civil/glm.Sev.RC.RData")
save(data_CM, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/
    Responsabilidad_Civil/Data_RC_model_CM.RData")
```

Código R: Modelo Frecuencia - Asistencias

```
################################################################################
##########                    MODELO GLM - FRECUENCIA
```

```
################################################################################
# PAQUETES Y LIBRERÍAS
#options(repos = c(cran="http://cran.rstudio.com"))
#install.packages("RODBC")
#install.packages("skimr")
library(RODBC)
library(tidyverse) # Incluye dplyr y tidyr para manejar los datos y ggplot para representarlos en grÃ¡ficos
library(plotly)    # Para graficos interactivos
library(lubridate) # manejar datas y horas
library(forcats)
library(rcartocolor)
library(readr)      # para leer y importar archivos
library(skimr)
db<-"C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/01_Frecuencia/03_Data/BD_GLM.accdb"
con <- odbcConnectAccess2007(db)
data <- sqlQuery(con,"select * from AS_Datos_Frecuencia")
str(data)
data.`Número Póliza`<-format(data.`Número Póliza`,scientific = FALSE)
# Reducimos la data sin el año 2015 y 2020, de acuerdo a los análisis univariados realizados
data <- data %>% filter(Año!=2015 & Año!=2020)
data <- data %>% filter(AS_Clase_Veh!='Desconocido')
data <- data %>% filter(Frecuencia<100)
# Cambiamos el nombre de algunos niveles de algunas variables
data <- data %>% mutate(AS_Antiguedad = case_when(
  AS_Antiguedad == '0' ~ 'a. 0',
  AS_Antiguedad == '1-2' ~ 'b. 1-2',
  AS_Antiguedad == '3' ~ 'c. 3',
  AS_Antiguedad == '4-5' ~ 'd. 4-5',
  AS_Antiguedad == '6' ~ 'e. 6',
  AS_Antiguedad == '7' ~ 'f. 7',
  AS_Antiguedad == '8+' ~ 'g. 8+'
))
data <- data %>% mutate(AS_SA = case_when(
  AS_SA == '[1000-10000>' ~ 'a.[1000-10000>',
  AS_SA == '[10000-12000>' ~ 'b.[10000-12000>',
  AS_SA == '[12000-15000>' ~ 'c.[12000-15000>',
  AS_SA == '[15000-18000>' ~ 'd.[15000-18000>',
  AS_SA == '[18000-24000>' ~ 'e.[18000-24000>',
  AS_SA == '[24000-30000>' ~ 'f.[24000-30000>',
  AS_SA == '[30000-36000>' ~ 'g.[30000-36000>',
  AS_SA == '[36000-99999>' ~ 'h.[36000-999999]'
))
# Convertimos a factor las variables
data.Año <- as.factor(data.Año)
data.AS_Tipo_Poliza <- as.factor(data.AS_Tipo_Poliza)
data.AS_Antiguedad <- as.factor(data.AS_Antiguedad)
data.AS_Canal_Venta <- as.factor(data.AS_Canal_Venta)
data.AS_Clase_Veh <- as.factor(data.AS_Clase_Veh)
data.AS_Tipo_Veh <- as.factor(data.AS_Tipo_Veh)
data.AS_Uso_Veh <- as.factor(data.AS_Uso_Veh)
data.AS_Marca_Veh <- as.factor(data.AS_Marca_Veh)
data.AS_Region <- as.factor(data.AS_Region)
data.AS_Zona <- as.factor(data.AS_Zona)
data.AS_Genero <- as.factor(data.AS_Genero)
data.AS_Tipo_Persona <- as.factor(data.AS_Tipo_Persona)
data.AS_Edad <- as.factor(data.AS_Edad)
data.AS_SA <- as.factor(data.AS_SA)
# Asignamos el intercepto para cada variable
data <- data %>% mutate(
  Año = fct_relevel(Año, "2022", after = 0),
  AS_Tipo_Poliza = fct_relevel(AS_Tipo_Poliza, "Individual", after = 0),
  AS_Antiguedad = fct_relevel(AS_Antiguedad, "b. 1-2", after = 0),
  AS_Canal_Venta = fct_relevel(AS_Canal_Venta, "Corredores", after = 0),
  AS_Clase_Veh = fct_relevel(AS_Clase_Veh, "Livianos", after = 0),
  AS_Tipo_Veh = fct_relevel(AS_Tipo_Veh, "Automovil", after = 0),
  AS_Uso_Veh = fct_relevel(AS_Uso_Veh, "Particular", after = 0) ,
  AS_Marca_Veh = fct_relevel(AS_Marca_Veh, "L2", after = 0),
  AS_Region = fct_relevel(AS_Region, "Lima", after = 0),
  AS_Zona = fct_relevel(AS_Zona, "A3", after = 0),
  AS_Genero = fct_relevel(AS_Genero, "Femenino", after = 0),
  AS_Tipo_Persona = fct_relevel( AS_Tipo_Persona, "Empresa", after = 0),
  AS_Edad = fct_relevel(AS_Edad, "38-42", after = 0),
  AS_SA = fct_relevel(AS_SA, "a.[1000-10000>", after = 0)
)
# Verificación de frecuencia por factor
data %>% group_by(Año) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*100)
```

```
data %>% group_by(AS_Tipo_Poliza) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>%mutate(Freq=round(C/E,5)*100)
data %>% group_by(AS_Antiguedad) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>%mutate(Freq=round(C/E,5)*100)
data %>% group_by(AS_Canal_Venta) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>%mutate(Freq=round(C/E,5)*100)
data %>% group_by(AS_Clase_Veh) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>%mutate(Freq=round(C/E,5)*100)
data %>% group_by(AS_Tipo_Veh) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*100)
data %>% group_by(AS_Uso_Veh) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*100)
data %>% group_by(AS_Marca_Veh) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>%mutate(Freq=round(C/E,5)*100)
data %>% group_by(AS_Region) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*100)
data %>% group_by(AS_Zona) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*100)
data %>% group_by(AS_Genero) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*100)
data %>% group_by(AS_Tipo_Persona) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>%mutate(Freq=round(C/E,5)*100)
data %>% group_by(AS_Edad) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*100)
data %>% group_by(AS_SA) %>% summarise(E=sum(Expuesto), C=sum(Frecuencia)) %>% mutate(Freq=round(C/E,5)*100)
# Agrupaciones motivadas por modelos previos
data <- data %>%
  mutate(
    AS_Antiguedad=fct_collapse(AS_Antiguedad,"c. 3+"=c("c. 3","d. 4-5","e. 6","f. 7","g. 8+")
    ),
    AS_Canal_Venta=fct_collapse(AS_Canal_Venta,"AgentesExc_BancaDigital"=c("Agentes Exclusivos","Banca y Digital")
    ),
    AS_Edad=fct_collapse(AS_Edad,"43+"=c("43-47","48-52","53-57","58-67","68-72","73-82","83+")
    ),
    AS_Genero=fct_collapse(AS_Genero,"Masculino"=c("Indeterminado","Masculino")
    ),
    AS_Marca_Veh=fct_collapse(AS_Marca_Veh,"L3_L4"=c("L3","L4"),"M1_M2_M3"=c("M1","M2","M3"),"P3_P4"=c("P3","P4"),"P2_P5"=c("P2","P5")
    ),
    AS_SA=fct_collapse(
      AS_SA,"b.[10000-18000>"=c("b.[10000-12000>","c.[12000-15000>","d.[15000-18000>"),"c.[18000-99999]"=c("e.[18000-24000>","f.[24000-30000>","g.[30000-36000>","h.[36000-999999]")
    ),
    AS_Uso_Veh=fct_collapse(
      AS_Uso_Veh,"Alquiler_Escolar"=c("Alquiler","Escolar"),"Taxi_Amb_Tur_Com"=c("Taxi","Ambulancia","Turismo","Comercial"),"TransPer_Carga_TransInt_TransUrb_Instr"=c("Transporte Personal","Carga","Transporte Interprovincial","Transporte Urbano","Instrucción")
    ),
    AS_Zona=fct_collapse(AS_Zona,"A1_A2"=c("A1","A2"),"A4_A5"=c("A4","A5"),"D1_D4_D5"=c("D1","D4","D5"),"D2_D3"=c("D2","D3")
    )
  )
##############################################################################################
# Agrupamos la data para modelar el GLM
#data_model <- data %>% mutate(id=as.numeric(`Número Póliza`)*100+as.numeric(`Número Riesgo`)) %>% group_by(id,Año,PP_Tipo_Poliza,PP_Antiguedad,PP_Canal_Venta,PP_Uso_Veh,PP_Marca_Veh,PP_Zona,PP_Genero,PP_Edad,PP_SA) %>% summarise(Expuesto=sum(Expuesto),Prima_Devengada=sum(Prima_Devengada),Cantidad=sum(Frecuencia),Incurrido=sum(Incurrido))
data_model <- data %>% group_by(Año,
                                #PT_Tipo_Poliza,
                                AS_Antiguedad,
                                AS_Canal_Venta,
                                AS_Uso_Veh,
                                AS_Marca_Veh,
                                AS_Zona,
                                AS_SA
)%>% summarise(Expuesto=sum(Expuesto),Prima_Devengada=sum(Prima_Devengada),Cantidad=sum(Frecuencia),Incurrido=sum(Incurrido))
# Modelo GLM Asistencias
glm.Frec.AS <- glm(
  formula = Cantidad ~
    Año +
    #AS_Tipo_Poliza +
    AS_Antiguedad +
    AS_Canal_Venta +
    #AS_Clase_Veh + (Se correlaciona con la MarcaVeh)
    #AS_Tipo_Veh +  (Se correlaciona con la MarcaVeh)
```

```
    AS_Uso_Veh +
    AS_Marca_Veh +
    #AS_Region + (Se correlaciona con la zona)
    AS_Zona +
    #AS_Genero +
    #AS_Tipo_Persona + (Se correlaciona con la Edad)
    #AS_Edad +
    AS_SA +
    offset(log(Expuesto)),
  family = poisson(link = "log"),
  data = data_model
)
# Desenmascarar posibles multicolinearidades entre las diferentes variables
temp <- alias(glm.Frec.AS)
temp2 <- as.data.frame(temp.Complete) # Si es vacio, no hay multicolinearidades
rm(temp,temp2)
# No se realizarán agrupaciones en los niveles de algunas variables por tener un buen modelo
# Incluimos las predicciones en la data
data <- data %>% mutate(Frecuencia.PRED = glm.Frec.AS.fitted.values)
data %>% summarise(sum(Frecuencia), sum(Frecuencia.PRED))
# Salvar un data frame con los coeficientes del modelo
coefs.Frecuencia.AS <- summary(glm.Frec.AS).coefficients %>%
  as_tibble() %>%
  mutate(var_level = glm.Frec.AS.coefficients %>% names()) %>%
  select(var_level, everything())
# Adicionar manualmente a la tabla los niveles atribuidos al intercept
# (y arredondar los valores)
temp <- tibble(
  var_level = c("Año2022","AS_Antiguedadb. 1-2","AS_Canal_VentaCorredores","AS_Uso_VehParticular","AS_Marca_
      VehL2","AS_ZonaA3","AS_SAa.[1000-10000>"
  ),
  Estimate = 0
)
f.round <- function(x, .digits = 4) round(x, digits = .digits)
coefs.Frecuencia.AS <- coefs.Frecuencia.AS %>%
  bind_rows(temp) %>%
  arrange(var_level) %>%
  mutate(`exp(Estimate)` = exp(Estimate)) %>%
  mutate_if(is.double, f.round)
# Esta función considera los datos de entrenamiento del GLM, agrega la exposición por variable y
# la junta a la tabla de coeficientes.
incluir.exposicion <- function(.data, .tbl.coefs){
  names.vars <- names(.data %>% select(-Expuesto))
  n <- length(names.vars)
  list.temp <- vector(mode = "list", length = n)
  for(i in seq_along(names.vars)){
    varname <- sym(names.vars[[i]])
    list.temp[[i]] <- .data %>%
      group_by(!!varname) %>%
      summarize(Exposición = f.round(sum(Expuesto), .digits = 0)) %>%
      ungroup() %>%
      mutate(var_level = str_c(names.vars[[i]], !!varname)) %>%
      select(var_level, Exposición)
  }
  exp.by.var <- bind_rows(list.temp)
  return(.tbl.coefs %>% left_join(exp.by.var))
}
coefs.Frecuencia.AS <- data %>%
  select(-Frecuencia, -Frecuencia.PRED) %>%
  incluir.exposicion(coefs.Frecuencia.AS)
coefs.Frecuencia.AS %>% DT::datatable()
# Guardamos los coeficientes en excel
write.csv(coefs.Frecuencia.AS,"C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/01_
    Frecuencia/04_Coeficientes/coefs.Frecuencia.AS.csv",fileEncoding ="Latin1")
save(glm.Frec.AS, file = "C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_
    Marginales/Asistencia/glm.Frec.AS.RData")
save(data, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/
    Asistencia/Data_AS.RData")
save(data_model, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales
    /Asistencia/Data_AS_model.RData")
```

Código R: Modelo Severidad - Asistencias

```
################################################################################
##########                  MODELO GLM - SEVERIDAD
```

```
################################################################################
# PAQUETES Y LIBRERÍAS
#options(repos = c(cran="http://cran.rstudio.com"))
#install.packages("RODBC")
#install.packages("skimr")
library(RODBC)
library(tidyverse) # Incluye dplyr y tidyr para manejar los datos y ggplot para representarlos en grÃ¡ficos
library(plotly)    # Para graficos interactivos
library(lubridate) # manejar datas y horas
library(forcats)
library(rcartocolor)
library(readr)      # para leer y importar archivos
library(skimr)
db2<-"C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/02_CostoMedio/03_Data/BD_GLM_CM.accdb"
con2 <- odbcConnectAccess2007(db2)
#data_CM <- sqlQuery(con2,"select * from AS_Datos_Severidad")
#str(data_CM)
# Reducimos la data sin el año 2015 y 2020, de acuerdo a los análisis univariados realizados
#data_CM <- data_CM %>% filter(Año!=2015 & Año!=2020)
#data_CM <- data_CM %>% filter(AS_Clase_Veh!='Desconocido')
#data_CM <- data_CM %>% filter(Incurrido>0)
#data_CM.`Número Póliza`<-format(data_CM.`Número Póliza`,scientific = FALSE)
#data_CM <- data_CM %>% filter(Cantidad<100)
data_CM <- data_model %>% filter(Incurrido>0) # onitir todas las acciones y saltar al modelo GLM
# Cambiamos el nombre de algunos niveles de algunas variables
data_CM <- data_CM %>% mutate(AS_Antiguedad = case_when(
  AS_Antiguedad == '0' ~ 'a. 0',AS_Antiguedad == '1-2' ~ 'b. 1-2',AS_Antiguedad == '3' ~ 'c. 3',
  AS_Antiguedad == '4-5' ~ 'd. 4-5',AS_Antiguedad == '6' ~ 'e. 6',AS_Antiguedad == '7' ~ 'f. 7',AS_Antiguedad == '8+' ~ 'g. 8+'
))
data_CM <- data_CM %>% mutate(AS_SA = case_when(
  AS_SA == '[1000-10000>' ~ 'a.[1000-10000>',AS_SA == '[10000-12000>' ~ 'b.[10000-12000>',AS_SA == '[12000-15000>' ~ 'c.[12000-15000>',AS_SA == '[15000-18000>' ~ 'd.[15000-18000>',AS_SA == '[18000-24000>' ~ 'e.[18000-24000>',AS_SA == '[24000-30000>' ~ 'f.[24000-30000>',AS_SA == '[30000-36000>' ~ 'g.[30000-36000>',AS_SA == '[36000-99999>' ~ 'h.[36000-999999]'
))
# Convertimos a factor las variables
data_CM.Año <- as.factor(data_CM.Año)
data_CM.AS_Tipo_Poliza <- as.factor(data_CM.AS_Tipo_Poliza)
data_CM.AS_Antiguedad <- as.factor(data_CM.AS_Antiguedad)
data_CM.AS_Canal_Venta <- as.factor(data_CM.AS_Canal_Venta)
data_CM.AS_Clase_Veh <- as.factor(data_CM.AS_Clase_Veh)
data_CM.AS_Tipo_Veh <- as.factor(data_CM.AS_Tipo_Veh)
data_CM.AS_Uso_Veh <- as.factor(data_CM.AS_Uso_Veh)
data_CM.AS_Marca_Veh <- as.factor(data_CM.AS_Marca_Veh)
data_CM.AS_Region <- as.factor(data_CM.AS_Region)
data_CM.AS_Zona <- as.factor(data_CM.AS_Zona)
data_CM.AS_Genero <- as.factor(data_CM.AS_Genero)
data_CM.AS_Tipo_Persona <- as.factor(data_CM.AS_Tipo_Persona)
data_CM.AS_Edad <- as.factor(data_CM.AS_Edad)
data_CM.AS_SA <- as.factor(data_CM.AS_SA)
# Asignamos el intercepto para cada variable
data_CM <- data_CM %>% mutate(Año = fct_relevel(Año, "2022", after = 0),
  AS_Tipo_Poliza = fct_relevel(AS_Tipo_Poliza, "Individual", after = 0),
  AS_Antiguedad = fct_relevel(AS_Antiguedad, "b. 1-2", after = 0),
  AS_Canal_Venta = fct_relevel(AS_Canal_Venta, "Corredores", after = 0),
  AS_Clase_Veh = fct_relevel(AS_Clase_Veh, "Livianos", after = 0),
  AS_Tipo_Veh = fct_relevel(AS_Tipo_Veh, "Automovil", after = 0),
  AS_Uso_Veh = fct_relevel(AS_Uso_Veh, "Particular", after = 0) ,
  AS_Marca_Veh = fct_relevel(AS_Marca_Veh, "L2", after = 0),
  AS_Region = fct_relevel(AS_Region, "Lima", after = 0),
  AS_Zona = fct_relevel(AS_Zona, "A3", after = 0),
  AS_Genero = fct_relevel(AS_Genero, "Femenino", after = 0),
  AS_Tipo_Persona = fct_relevel(AS_Tipo_Persona, "Empresa", after = 0),
  AS_Edad = fct_relevel(AS_Edad, "38-42", after = 0
  ),AS_SA = fct_relevel(AS_SA, "a.[1000-10000>", after = 0)
)
# Verificación de frecuencia por factor
data_CM %>% group_by(Año) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(AS_Tipo_Poliza) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(AS_Antiguedad) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(AS_Canal_Venta) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(AS_Clase_Veh) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(AS_Tipo_Veh) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(AS_Uso_Veh) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
```

```
data_CM %>% group_by(AS_Marca_Veh) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(AS_Region) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(AS_Zona) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(AS_Genero) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(AS_Tipo_Persona) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(AS_Edad) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
data_CM %>% group_by(AS_SA) %>% summarise(M=sum(Incurrido), C=n()) %>% mutate(CM=round(M/C,5))
# Agrupaciones motivadas por modelos previos
data_CM <- data_CM %>%
  mutate(
    AS_Antiguedad=fct_collapse(AS_Antiguedad,"c. 3+"=c("c. 3","d. 4-5","e. 6","f. 7","g. 8+")),
    AS_Canal_Venta=fct_collapse(AS_Canal_Venta,"AgentesExc_BancaDigital"=c("Agentes Exclusivos","Banca y Digital")),
    AS_Edad=fct_collapse(AS_Edad,"43+"=c("43-47","48-52","53-57","58-67","68-72","73-82","83+")),
    AS_Genero=fct_collapse(AS_Genero,"Masculino"=c("Indeterminado","Masculino")),
    AS_Marca_Veh=fct_collapse(AS_Marca_Veh,"L3_L4"=c("L3","L4"),"M1_M2_M3"=c("M1","M2","M3"),"P3_P4"=c("P3","P4"),"P2_P5"=c("P2","P5")),
    AS_SA=fct_collapse(AS_SA,"b.[10000-18000>"=c("b.[10000-12000>","c.[12000-15000>","d.[15000-18000>"),"c.[18000-99999]"=c("e.[18000-24000>","f.[24000-30000>","g.[30000-36000>","h.[36000-999999]")),
    AS_Uso_Veh=fct_collapse(AS_Uso_Veh,"Alquiler_Escolar"=c("Alquiler","Escolar"),"Taxi_Amb_Tur_Com"=c("Taxi","Ambulancia","Turismo","Comercial"),"TransPer_Carga_TransInt_TransUrb_Instr"=c("TransportePersonal","Carga","Transporte Interprovincial","Transporte Urbano")),
    AS_Zona=fct_collapse(AS_Zona,"A1_A2"=c("A1","A2"),"A4_A5"=c("A4","A5"),"D1_D4_D5"=c("D1","D4","D5"),"D2_D3"=c("D2","D3"))
  )
# Creamos las variable dependiente
data_CM <-data_CM %>% mutate(CM=Incurrido/Cantidad)
# Modelo GLM Pérdida Parcial
glm.Sev.AS <- glm(
  formula = CM ~
    Año +
    #AS_Tipo_Poliza +
    AS_Antiguedad +
    AS_Canal_Venta +
    #AS_Clase_Veh + (Se correlaciona con la MarcaVeh)
    #AS_Tipo_Veh +  (Se correlaciona con la MarcaVeh)
    AS_Uso_Veh +
    AS_Marca_Veh +
    #AS_Region + (Se correlaciona con la zona)
    AS_Zona +
    #AS_Genero +
    #AS_Tipo_Persona + (Se correlaciona con la Edad)
    #AS_Edad,
    AS_SA,
  #offset(log(Expuesto)), Se coloca este offset en caso que la variable dependiente hubiera sido solo el Incurrido
  family = Gamma(link = "log"),
  data = data_CM
)

# Desenmascarar posibles multicolinearidades entre las diferentes variables
temp <- alias(glm.Sev.AS)
temp2 <- as.data.frame(temp.Complete) # Si es vacio, no hay multicolinearidades
rm(temp,temp2)
# No se realizarán agrupaciones en los niveles de algunas variables por tener un buen modelo
# Incluimos las predicciones en la data
data_CM <- data_CM %>% mutate(CM.PRED = glm.Sev.AS.fitted.values)
data_CM %>% summarise(sum(Incurrido), sum(CM.PRED*Cantidad))
# Salvar un data frame con los coeficientes del modelo
coefs.Severidad.AS <- summary(glm.Sev.AS).coefficients %>%
  as_tibble() %>%
  mutate(var_level = glm.Sev.AS.coefficients %>% names()) %>%
  select(var_level, everything())
# Adicionar manualmente a la tabla los niveles atribuidos al intercept
# (y arredondar los valores)
temp <- tibble(
  var_level = c("Año2022","AS_Antiguedadb. 1-2","AS_Canal_VentaCorredores","AS_Uso_VehParticular","AS_Marca_VehL2","AS_ZonaA3","AS_SAa.[1000-10000>"),
  Estimate = 0
)
f.round <- function(x, .digits = 4) round(x, digits = .digits)
coefs.Severidad.AS <- coefs.Severidad.AS %>%
  bind_rows(temp) %>% arrange(var_level) %>% mutate(`exp(Estimate)` = exp(Estimate)) %>% mutate_if(is.double, f.round)
# Esta función considera los datos de entrenamiento del GLM, agrega la exposición por variable y
# la junta a la tabla de coeficientes.
```

```r
incluir.Cantidad <- function(.data, .tbl.coefs){
  names.vars <- names(.data %>% select(-Cantidad))
  n <- length(names.vars)
  list.temp <- vector(mode = "list", length = n)
  for(i in seq_along(names.vars)){
    varname <- sym(names.vars[[i]])
    list.temp[[i]] <- .data %>%
      group_by(!!varname) %>%
      summarize(Incurrido = f.round(sum(Cantidad), .digits = 0)) %>%
      ungroup() %>%
      mutate(var_level = str_c(names.vars[[i]], !!varname)) %>%
      select(var_level, Incurrido)
  }
  exp.by.var <- bind_rows(list.temp)
  return(.tbl.coefs %>% left_join(exp.by.var))
}
coefs.Severidad.AS <- data_CM %>%
  select(-CM.PRED) %>%
  incluir.Cantidad(coefs.Severidad.AS)
coefs.Severidad.AS %>% DT::datatable()
# Guardamos los coeficientes en excel
write.csv(coefs.Severidad.AS,"C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/02_CostoMedio/04_Coeficientes/coefs.Severidad.AS.csv",fileEncoding = "Latin1")
save(glm.Sev.AS, file = "C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Asistencia/glm.Sev.AS.RData")
save(data_CM, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Asistencia/Data_AS_model_CM.RData")
```

Código R: Índice de GINI

```r
load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Perdida_Parcial/glm.Frec.PP.RData")
load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Perdida_Total/glm.Frec.PT.RData")
load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Responsabilidad_Civil/glm.Frec.RC.RData")
load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Asistencia/glm.Frec.AS.RData")

load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Perdida_Parcial/Data_PP.RData")
load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Perdida_Total/Data_PT.RData")
load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Responsabilidad_Civil/Data_RC.RData")
load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Asistencia/Data_AS.RData")

library(REAT)
library(ineq)

# PERDIDA PARCIAL
data <- data %>% mutate(FREC_PRED_PP=predict(glm.Frec.PP,newdata =data,type = "response"))
data_PP<-cbind(data.Expuesto,data.FREC_PRED_PP)
data_PP <- as.data.frame(data_PP)
data <- data %>% mutate(FREC_PRED_PT=predict(glm.Frec.PT,newdata data,type = "response"))
# PERDIDA TOTAL
data_PT<-cbind(data.Expuesto,data.FREC_PRED_PT)
data_PT <- as.data.frame(data_PT)
# RESPONSABILIDAD CIVIL
data <- data %>% mutate(FREC_PRED_RC=predict(glm.Frec.RC,newdata =data,type = "response"))
data_RC<-cbind(data.Expuesto,data.FREC_PRED_RC)
data_RC <- as.data.frame(data_RC)
# ASISTENCIA
data <- data %>% mutate(FREC_PRED_AS=predict(glm.Frec.AS,newdata =data,type = "response"))
data_AS<-cbind(data.Expuesto,data.FREC_PRED_AS)
data_AS <- as.data.frame(data_AS)
# GRAFICOS
par(mfrow=c(2,2))
lorenz(data_PP.V2 ,weighting = data_PP.V1,lcg = TRUE,lcx = "Porcentaje de Expuestos",lcy="Porcentaje de Pérdidas",lctitle="Indice de Gini - Pérdida Parcial",lsize=2)
lorenz(data_PT.V2 ,weighting = data_PT.V1,lcg = TRUE,lcx = "Porcentaje de Expuestos",lcy="Porcentaje de Pérdidas",lctitle="Indice de Gini - Pérdida Total",lsize=2)
lorenz(data_RC.V2 ,weighting = data_RC.V1,lcg = TRUE,lcx = "Porcentaje de Expuestos",lcy="Porcentaje de Pérdidas",lctitle="Indice de Gini - Responsabilidad Civil",lsize=2)
```

```
35 lorenz(data_AS.V2 ,weighting = data_AS.V1,lcg = TRUE,lcx = "Porcentaje de Expuestos",lcy="Porcentaje de
       Pérdidas",lctitle="Indice de Gini - Asistencias",lsize=2)
```

ANEXO C:INDICIO DE DEPENDENCIA Y MODELOS GLM CONJUNTOS

C.1. Indicios de Dependencia

```
data_PP <- data %>%
  group_by(Año,PP_Tipo_Poliza,PP_Antiguedad,PP_Canal_Venta,PP_Uso_Veh,PP_Marca_Veh,PP_Zona,PP_Genero,PP_Edad,
      PP_SA) %>%
  summarise(Exp=sum(Expuesto),Cantidad=sum(Frecuencia),Incurrido=sum(Incurrido))

data_PP <- data_PP %>% mutate(
  Frecuencia=Cantidad/Exp,
  Severidad=Incurrido/Cantidad
)

data_PP <- data_PP %>% filter(Cantidad>0 & Severidad>=0)
write.csv(data_PP,"C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/10_Overleaf_Tesis/Cuadros/Indicio_
    DEP.csv",fileEncoding = "Latin1")
#---------------------------------
data_PT <- data %>%
  group_by(Año,PT_Tipo_Poliza,PT_Uso_Veh,PT_Marca_Veh,PT_Zona,PT_Genero,PT_Edad) %>%
  summarise(Exp=sum(Expuesto),Cantidad=sum(Frecuencia),Incurrido=sum(Incurrido))

data_PT <- data_PT %>% mutate(
  Frecuencia=Cantidad/Exp,
  Severidad=Incurrido/Cantidad
)

data_PT <- data_PT %>% filter(Cantidad>0 & Severidad>=0)
write.csv(data_PT,"C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/10_Overleaf_Tesis/Cuadros/Indicio_
    DEP_PT.csv",fileEncoding = "Latin1")
#---------------------------------
data_RC <- data %>%
  group_by(Año,RC_Antiguedad,RC_Canal_Venta,RC_Uso_Veh,RC_Marca_Veh,RC_Zona,RC_Edad) %>%
  summarise(Exp=sum(Expuesto),Cantidad=sum(Frecuencia),Incurrido=sum(Incurrido))

data_RC <- data_RC %>% mutate(
  Frecuencia=Cantidad/Exp,
  Severidad=Incurrido/Cantidad
)

data_RC <- data_RC %>% filter(Cantidad>0 & Severidad>=0)
write.csv(data_RC,"C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/10_Overleaf_Tesis/Cuadros/Indicio_
    DEP_RC.csv",fileEncoding = "Latin1")
#---------------------------------
data_AS <- data %>%
  group_by(Año,AS_Antiguedad,AS_Canal_Venta,AS_Uso_Veh,AS_Marca_Veh,AS_Zona,AS_SA) %>%
  summarise(Exp=sum(Expuesto),Cantidad=sum(Frecuencia),Incurrido=sum(Incurrido))

data_AS <- data_AS %>% mutate(
  Frecuencia=Cantidad/Exp,
  Severidad=Incurrido/Cantidad
)

data_AS <- data_AS %>% filter(Cantidad>0 & Severidad>=0)
write.csv(data_AS,"C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/10_Overleaf_Tesis/Cuadros/Indicio_
    DEP_AS.csv",fileEncoding = "Latin1")
#---------------------------------
```

C.2. Gráficos CHI-PLOT y K-PLOT

```

library(MASS)
library(evd)
library(asbio)
library(copula)
library(VineCopula)
library(base)
load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Perdida_Parcial/
    glm.Frec.PP.RData")
load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Perdida_Total/
    glm.Frec.PT.RData")
load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Responsabilidad_
    Civil/glm.Frec.RC.RData")
load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Asistencia/glm.
    Frec.AS.RData")
```

```
12
13 load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Perdida_Parcial/
       Data_PP.RData")
14 load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Perdida_Total/
       Data_PT.RData")
15 load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Responsabilidad_
       Civil/Data_RC.RData")
16 load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Asistencia/Data_
       AS.RData")
17
18 FS_PP<- cbind(data_PP.Cantidad,data_PP.Severidad)
19 FS_PP<-pobs(FS_PP)
20 FS_PP <- as.data.frame(FS_PP)
21 rand_df_PP <- FS_PP[sample(nrow(FS_PP), size=1000), ]
22
23 FS_PT<- cbind(data_PT.Cantidad,data_PT.Severidad)
24 FS_PT<-pobs(FS_PT)
25 FS_PT <- as.data.frame(FS_PT)
26 rand_df_PT <- FS_PT[sample(nrow(FS_PT), size=1000), ]
27
28 FS_RC<- cbind(data_RC.Cantidad,data_RC.Severidad)
29 FS_RC<-pobs(FS_RC)
30 FS_RC <- as.data.frame(FS_RC)
31 rand_df_RC <- FS_RC[sample(nrow(FS_RC), size=1000), ]
32
33 FS_AS      <- cbind(data_AS.Cantidad,data_AS.Severidad)
34 FS_AS      <-pobs(FS_AS)
35 FS_AS      <- as.data.frame(FS_AS)
36 rand_df_AS <- FS_AS[sample(nrow(FS_AS), size=1000), ]
37
38 par(mfrow=c(2,2))
39 BiCopChiPlot(rand_df_PP.V1,rand_df_PP.V2, ylim=c(-0.1,1), main="Pérdida Parcial", col=4)
40 BiCopChiPlot(rand_df_PT.V1,rand_df_PT.V2, ylim=c(-0.1,1), main="Pérdida Total", col=4)
41 BiCopChiPlot(rand_df_RC.V1,rand_df_RC.V2, ylim=c(-0.1,1), main="Responsabilidad Civil", col=4)
42 BiCopChiPlot(rand_df_AS.V1,rand_df_AS.V2, ylim=c(-0.1,1), main="Asistencias", col=4)
43
44 BiCopKPlot(rand_df_PP.V1,rand_df_PP.V2, lwd=2, col=4, main="Pérdida Parcial")
45 BiCopKPlot(rand_df_PT.V1,rand_df_PT.V2, lwd=2, col=4, main="Pérdida Total")
46 BiCopKPlot(rand_df_RC.V1,rand_df_RC.V2, lwd=2, col=4, main="Responsabilidad Civil")
47 BiCopKPlot(rand_df_AS.V1,rand_df_AS.V2, lwd=2, col=4, main="Asistencias")
```

C.3. Modelo Conjunto - Pérdida Parcial

```
1  ##############################################################################
2  ##############################################################################
3  ####                                                                      ######
4  ####                  REGRESION BASADO EN COPULAS                       ######
5  ####                       PERDIDA PARCIAL                              ######
6  ####                                                                      ######
7  ##############################################################################
8  ##############################################################################
9  # Cargamos las librerías necesarias
10   #library(copula)
11   library(RODBC)
12   library(tidyverse)
13   library(MASS)
14   #library(GJRM)
15   #library(devtools)
16   library(CopulaRegression)
17   library(VineCopula)
18 library(optimx)
19   #install_url('http://cran.r-project.org/src/contrib/Archive/CopulaRegression/CopulaRegression_0.1-5.tar.gz')
20 ##############################################################################
21 # REGRESION GAUSSIANA
22 ##############################################################################
23
24 # Set Directory
25   setwd("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Perdida_
       Parcial")
26
27 # Carga los modelos GLM MARGINALES calculados en otos querys
28   load("glm.Frec.PP.RData")
29   load("glm.Sev.PP.RData")
30   load("Data_PP.RData")
31   load("Data_PP_model.RData")
```

```
load("Data_PP_model_CM.RData")

# Creamos las variables de respuestas y la matriz de datos de cada glm marginal
  # SEVERIDAD
  x <- data_CM.Incurrido/data_CM.Cantidad #(Costo Medio)
  var_x <-model.matrix(glm.Sev.PP)  #data_CM[,3:12]
  # FRECUENCIA
  y <- data_model.Cantidad
  Expuestos <- data_model.Expuesto
  var_y <- model.matrix(glm.Frec.PP)   #data[,3:12]

# Estimacion de parámetros iniciales utilizando las distribuciones marginales
  # SEVERIDAD
  sd.alpha <- sqrt(diag(vcov(glm.Sev.PP)))
  alpha_0  <- glm.Sev.PP.coefficients
  delta_0  <- summary(glm.Sev.PP).dispersion
  mu_0     <- exp(var_x%*%alpha_0)

  # FRECUENCIA
  beta_0   <- glm.Frec.PP.coefficients
  sd.beta  <- sqrt(diag(vcov(glm.Frec.PP)))
  lambda_0 <- exp(var_y%*%beta_0)*Expuestos

# REGRESION MEDIANTE COPULAS

  # COPULA GAUSSIANA
  family=1
  # Estimación inicial del parámetro theta para estimarlo mediante el algoritmo de MPLE
  theta_ini <- BiCopEst(rank(data_model.Incurrido-exp(var_y%*%alpha_0) )/(length(y)+1),rank(y*Expuestos-lambda_0)/(length(y)+1), family=family).par
  #theta_ini <- BiCopEst(rank(data_CM.Incurrido-exp(var_x%*%alpha_0))/(length(x)+1),rank(data_CM.Cantidad*data_CM.Expuesto-exp(var_x%*%beta_0)*data_CM.Expuesto)/(length(x)+1), family=family).par
  tau_ini    <- BiCopPar2Tau(par=theta_ini,family = family)

  # Estimación de las seudo-observaciones
  u<-pgam(x,mu_0,delta_0)
  v    <- ppois(data_CM.Cantidad,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)
  vv   <- ppois(data_CM.Cantidad-1,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)

  # Creamos la función primera derivada de la Función Copula
    Du_Gaus <- function(u,v,theta){
      u[u<=0]=0.001
      u[u>=1]=0.999
      v[v<=0]=0.001
      v[v>=1]=0.999
      out <- pnorm((qnorm(v)-theta*qnorm(u))/(sqrt(1-theta^2)))
    return(out)
    }

  # Función a maximizar
  f_aux <- function(para){
    D_u_0<-Du_Gaus(u,v,para)-Du_Gaus(u,vv,para)
    D_u_0[D_u_0<=0]=1
    D_u_0 <- log(D_u_0)
    out<-(-sum(D_u_0))
    return(out)
  }

  # Proceso de maximización
  #para_ini <- theta2z(theta_ini,family = family)
  para.ifm <- optim(theta_ini,f_aux,method = "BFGS").par
  #theta_0  <- z2theta(para.ifm,family = family)
  theta_0  <- para.ifm # Este será nuestro parámetro inicial estimado con el método IFM y que luego usaremos para la regresión conjunta
  tau      <-BiCopPar2Tau(par=theta_0,family=family) # tau de Khendal para estimar el grado de dependencia

  # Estimamos los coeficientes de regresión y el parámetro cópula mediante máxima verosimilitud

  # Creamos nuestra función a maximizar
  ######################################################################
    f_aux_reg <- function(para){

        p       <- ncol(var_x)
        q       <- ncol(var_x)
        alpha   <- para[1:p]
        beta    <- para[(p+1):(p+q)]
```

```
        # theta    <- z2theta(para[p+q+1],family)
        theta    <- para[p+q+1]
        delta    <- para[p+q+2]
        lambda   <- as.vector(exp(var_y%*%beta)*data_model.Expuesto)
        mu       <- as.vector(exp(var_y%*%alpha))
        #dummy    <- density_joint(x,y,mu,delta,lambda,theta,family=family,zt=FALSE)
        u2 <- pgam(data_model.Incurrido,mu,delta)
        v2<-ppois(y,lambda)
        vv2<-ppois(y-1,lambda)

        marginal.x <- dgam(data_model.Incurrido,mu,delta)
        marginal.x[marginal.x>=1]=0
        marginal.x[marginal.x<=0]=0
        marginal.x[is.na(marginal.x)]=0
        marginal.x[marginal.x==-Inf]=0
        marginal.x[marginal.x==Inf]=0
        par_der    <- Du_Gaus(u2,v2,theta)
        par_der1   <- Du_Gaus(u2,vv2,theta)
        dummy<- par_der-par_der1
        dummy[y==0]=par_der[y==0]
        out<-marginal.x*dummy
        out[out<=0]=1e-10
        ll       <- -sum(log(out))
        #if(negative==TRUE) ll <- (-ll)
        return(ll)
}

#########################################################################
para_0<-c(alpha_0,beta_0,theta_0,delta_0) # Parámetros iniciales
a<-now()
para_optim <- optim(para_0,f_aux_reg,method = "BFGS",hessian = FALSE) # Estimación Maxima verosmil de los
    parámetros (Toma hasta 3h sin matriz hessiana. 8h con Matriz hessiana)
b<-now()
b-a
#load("C:/Users/josephgarcial/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_Conjuntos/Perdida_
    Parcial/JointModel_Gaus.RData")
para_optim.value # Log_Likelihood
vector <- para_optim.par
p        <- ncol(var_y)
q        <- ncol(var_y)

# Parámetros optimizados
    alpha_gau<-vector[1:p]
    beta_gau<-vector[(p+1):(p+q)]
    theta_gau=z2theta(vector[p+q+1],family) #-0.09506928
    delta_gau<-exp(vector[p+q+2]) #4.271691
    tau_gau<-BiCopPar2Tau(par=theta_gau,family=family)#-0.06061453
    head(beta_0)
# Determinamos los errores estandar para el test de Wald
    hessian_gau   <- para_optim.hessian
    Hinv          <- ginv(hessian_gau)
    sd            <- sqrt(diag(Hinv))
    sd.alpha_gau  <-sd[1:p]
    sd.beta_gau   <- sd[(p+1):(p+q)]
    sd.theta_gau  <-sd[p+q+1]

# Ordenamos los coeficientes estimados en una tabla
    Reg_Cop_Gausiana <- data.frame(Estimate_Frec=round(beta_gau,4),Std.error.frec=0,z_value_frec=0,"Prob_
    frec"=0,Estimate_Sev=round(alpha_gau,4),Std.error.sev=0,z_value_sev=0,"Prob_Sev"=0)
    var_level<-row.names(Reg_Cop_Gausiana)
    Reg_Cop_Gausiana <- Reg_Cop_Gausiana %>% as.tibble() %>%  mutate(var_level=var_level) %>% relocate(var_
    level)

    # Agregamos los interceptos
    temp <- tibble(
      var_level = c(
        "Año2022",
        "PP_Tipo_PolizaIndividual",
        "PP_Antiguedadc. 02",
        "PP_Canal_VentaCorredores",
        "PP_Uso_VehParticular",
        "PP_Marca_VehL3",
        "PP_ZonaA2",
        "PP_GeneroFemenino",
        "PP_Edad38-42",
        "PP_SAj.[13000-999999]"
```

```
        ),
        Estimate_Frec = 0
        ,
        Estimate_Sev = 0
      )

      # coeficientes del modelo de regresion con dependencia
      Reg_Cop_Gausiana <- Reg_Cop_Gausiana %>% bind_rows(temp) %>% arrange(var_level) %>% mutate(`exp(Estimate
      _Frec)`=round(exp(Estimate_Frec),4),          `exp(Estimate_Sev)`=round(exp(Estimate_Sev),4))
      Reg_Cop_Gausiana <-Reg_Cop_Gausiana %>%  relocate(`exp(Estimate_Frec)`, .after=Prob_frec)

# Guardamos el modelo conjunto y los coeficientes
write.csv(Reg_Cop_Gausiana, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_
    GLM_Conjuntos/Perdida_Parcial/Coef_Gaus.csv", fileEncoding = "Latin1")
save(para_optim, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_
    Conjuntos/Perdida_Parcial/JointModel_Gaus.RData")

# Simulacion de la Pérdida Total
# ================================================================
    # CON DEPENDENCIA
    # ---------------
              lambda_loss_dep  <- exp(var_x%*%beta_gau)*data_CM.Expuesto
              mu_loss_dep      <- exp(var_x%*%alpha_gau)
              delta_loss_dep   <- delta_gau
              # Función de densidad fx
              f_loss <-
                function(loss,mu,d,lam,theta){
                  n<-length(loss)
                  if (length(lam)==1) lam  <- rep(lam,n)
                  if (length(mu)==1) mu    <- rep(mu,n)
                  out                      <- vector(length=n)

                  for(i in 1:n){
                    N  <-1:20
                    v  <-ppois(N,lam[i])
                    vv <-ppois(N-1,lam[i])
                    u  <-pgam(loss[i]/N,mu[i],d)

                    Der_cop <-Du_Gaus(u,v,theta)-Du_Gaus(u,vv,theta)
                    dummy   <-Der_cop*dgam(loss[i]/N,mu[i],d)/N
                    out[i]  <-sum(dummy)
                  }

                  out[loss<=0]=0
                  return(out)
                }

              # Función de densidad acumulada Fx
              F_loss <-
                function(loss,mu,d,lam,theta){
                  out<-vector(length=length(loss))

                  for(i in 1:length(loss)){
                    floss <- function(s){
                      f_loss(s,mu[i],d,lam[i],theta)
                    }
                    out[i] <- integrate(floss,0,loss[i]).value
                  }
                  return(out)
                }

              # Estimación del Total Loss (fuerte dedicacion computacional)

              #k <-length(lambda_loss_dep)
              k <- length(lambda_loss_dep) # Cantidad de numeros aleatorios (Debe ser del tamaño de lambda)
              m <- 10 # Cantidad de muestras aleatorias (Deben ser 10 mil)
              L <- vector(length=k)
              S <- vector(length=m)

              for(j in 1:m){
                r_uni<-runif(k)
                for(i in 1:k){
                  f_root <-function(s){
                    F_loss(s,mu_loss_dep[i],delta_loss_dep,lambda_loss_dep[i],theta_gau)-r_uni[i]
                  }
```

```
                    tryCatch(
                      error = function(cnd) mu_loss_dep[i]*lambda_loss_dep[i],
                      loss<-uniroot(f_root,lower = 0,upper = 500000)
                    )

                    #print(loss.root)
                    L[i]<-loss.root
                    perc<-paste(i/k*100,"%")
                    print(perc)
                  }
                  S[j]<-sum(L)
                  perc<-paste(j/m*100,"%")
                  print(perc)
                  #print(S[j])
                }
          S

      # CON INDEPENDENCIA
      # -----------------
                lambda_loss  <-predict(glm.Frec.PP, newdata=data_CM, type='response') # CAMBIAR LOS DATOS A UN
      MODELO DE RIESGO COLECTIVO
                mu_loss      <-predict(glm.Sev.PP, newdata=data_CM, type='response')
                delta_loss   <-delta_0

                M<-seq(1:10000)
                for(i in 1:10000){
                  muestra_cant<-rpois(length(lambda_loss),lambda_loss)
                  S<-vector(length=length(lambda_loss))
                  for(j in 1:length(muestra_cant)){
                    if(muestra_cant[j]==0){
                      S[j]=0
                    }else{
                      S[j]=sum(rgam(n =muestra_cant[j],mu_loss[j],delta_loss ))
                    }
                  }

                  M[i] <- sum(S)
                  porc<-paste((i/10000)*100,"%")
                  print(porc)
                  #print(M[i])
                }

                hist(M)
  # Simulacion de la Frecuencia con  Independencia
        #ADECUAR LOS DATOS DE SIMULACION A UN ESTRUCTURA DE MODELO COLECTIVO DE RIESGO
        # Frecuencia
        N<-seq(1:10000)
        for(i in 1:10000){
          N[i] <- sum(simulate(glm.Frec.PP,nsim=1, type="response" ).sim_1)/sum(data_model.Expuesto)
          porc<-paste((i/10000)*100,"%")
          print(porc)
        }
        hist(N)
#####################################################################
  # Limpiamos el ambiente de variables
    rm(list = ls())
    gc()

#########################################################################
# REGRESION CLAYTON
#########################################################################

    # Set Directory
    setwd("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Perdida_
      Parcial")

    # Carga los modelos GLM MARGINALES calculados en otos querys
    load("glm.Frec.PP.RData")
    load("glm.Sev.PP.RData")
    load("Data_PP.RData")
    load("Data_PP_model.RData")
    load("Data_PP_model_CM.RData")

    # Creamos las variables de respuestas y la matriz de datos de cada glm marginal
    # SEVERIDAD
    x <- data_CM.Incurrido/data_CM.Cantidad #(Costo Medio)
```

```
var_x <-model.matrix(glm.Sev.PP)  #data_CM[,3:12]
# FRECUENCIA
y <- data_model.Cantidad
Expuestos <- data_model.Expuesto
var_y <- model.matrix(glm.Frec.PP)   #data[,3:12]

# Estimacion de parámetros iniciales utilizando las distribuciones marginales
# SEVERIDAD
sd.alpha <- sqrt(diag(vcov(glm.Sev.PP)))
alpha_0  <- glm.Sev.PP.coefficients
delta_0  <- summary(glm.Sev.PP).dispersion
mu_0     <- exp(var_x%*%alpha_0)

# FRECUENCIA
beta_0   <- glm.Frec.PP.coefficients
sd.beta  <- sqrt(diag(vcov(glm.Frec.PP)))
lambda_0 <- exp(var_y%*%beta_0)*Expuestos

# REGRESION MEDIANTE COPULAS

# COPULA CLAYTON
family=3
# Estimación inicial del parámetro theta para estimarlo mediante el algoritmo de MPLE
theta_ini <- BiCopEst(rank(data_model.Incurrido-exp(var_y%*%alpha_0) )/(length(y)+1),rank(y*Expuestos-
  lambda_0)/(length(y)+1), family=family).par
#theta_ini <- BiCopEst(rank(data_CM.Incurrido-exp(var_x%*%alpha_0))/(length(x)+1),rank(data_CM.Cantidad*
  data_CM.Expuesto-exp(var_x%*%beta_0)*data_CM.Expuesto)/(length(x)+1), family=family).par
tau_ini   <- BiCopPar2Tau(par=theta_ini,family = family)

# Estimación de las seudo-observaciones
u<-pgam(x,mu_0,delta_0)
v   <- ppois(data_CM.Cantidad,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)
vv  <- ppois(data_CM.Cantidad-1,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)

# Creamos la función primera derivada de la Función Copula
Du_Clay <- function(u,v,theta){
  u[u<=0]=0.0001
  u[u>1]=1
  v[v<=0]=0.0001
  v[v>1]=1
  #out <- pnorm((qnorm(v)-theta*qnorm(u))/(sqrt(1-theta^2)))
  out <- (u^(-theta)+v^(-theta)-1)^((-1/theta)-1)*(u^(-theta-1))
  return(out)
}

# Función a maximizar
f_aux <- function(para){
  D_u_0<-Du_Clay(u,v,para)-Du_Clay(u,vv,para)
  D_u_0[D_u_0<=0]=1
  D_u_0 <- log(D_u_0)
  out<-(-sum(D_u_0))
  return(out)
}

# Proceso de maximización
#para_ini <- theta2z(theta_ini,family = family)
para.ifm <- optim(theta_ini,f_aux,method = "BFGS").par
#theta_0  <- z2theta(para.ifm,family = family)
theta_0  <- para.ifm # Este será nuestro parámetro inicial estimado con el método IFM y que luego usaremos
   para la regresión conjunta
tau      <-BiCopPar2Tau(par=theta_0,family=family) # tau de Khendal para estimar el grado de dependencia

# Estimamos los coeficientes de regresión y el parámetro cópula mediante máxima verosimilitud

# Creamos nuestra función a maximizar
################################################################
f_aux_reg <- function(para){
  p       <- ncol(var_x)
  q       <- ncol(var_x)
  alpha   <- para[1:p]
  beta    <- para[(p+1):(p+q)]
  # theta   <- z2theta(para[p+q+1],family)
  theta   <- z2theta(para[p+q+1],family = family)
  delta   <- para[p+q+2]
  lambda  <- as.vector(exp(var_y%*%beta)*data_model.Expuesto)
  mu      <- as.vector(exp(var_y%*%alpha))
```

```
397       #dummy    <- density_joint(x,y,mu,delta,lambda,theta,family=family,zt=FALSE)
398       u2 <- pgam(data_model.Incurrido,mu,delta)
399       v2<-ppois(y,lambda)
400       vv2<-ppois(y-1,lambda)
401
402       marginal.x <- dgam(data_model.Incurrido,mu,delta)
403       marginal.x[marginal.x>=1]=0
404       marginal.x[marginal.x<=0]=0
405       marginal.x[is.na(marginal.x)]=0
406       marginal.x[marginal.x==-Inf]=0
407       marginal.x[marginal.x==Inf]=0
408       par_der    <- Du_Clay(u2,v2,theta)
409       par_der1   <- Du_Clay(u2,vv2,theta)
410       dummy<- par_der-par_der1
411       dummy[y==0]=par_der[y==0]
412       out<-marginal.x*dummy
413       out[out<=0]=1e-10
414       ll        <- -sum(log(out))
415       #if(negative==TRUE) ll <- (-ll)
416       return(ll)
417     }
418     ######################################################################
419     para_0<-c(alpha_0,beta_0,theta2z(theta_0,family=family),delta_0) # Parámetros iniciales
420     a<-now()
421     para_optim <- optim(para_0,f_aux_reg,method = "BFGS",hessian = FALSE) # Estimación Maxima verosmil de los
          parámetros (Toma hasta 3h sin matriz hessiana. 8h con Matriz hessiana)
422     b<-now()
423     b-a
424     #load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_Conjuntos/Perdida_
          Parcial/JointModel_Gaus.RData")
425     para_optim.value # Log_Likelihood
426     vector <- para_optim.par
427     p         <- ncol(var_y)
428     q         <- ncol(var_y)
429     Log_Likelihood<-para_optim.value
430     # Parámetros optimizados
431     alpha_clay<-vector[1:p]
432     beta_clay<-vector[(p+1):(p+q)]
433     theta_clay=z2theta(vector[p+q+1],family) #-0.09506928
434     delta_clay<-exp(vector[p+q+2]) #4.271691
435     tau_clay<-BiCopPar2Tau(par=theta_clay,family=family)#-0.06061453
436     head(beta_clay)
437     # Determinamos los errores estandar para el test de Wald
438     hessian_gau    <- para_optim.hessian
439     Hinv           <- ginv(hessian_gau)
440     sd             <- sqrt(diag(Hinv))
441     sd.alpha_gau   <-sd[1:p]
442     sd.beta_gau    <- sd[(p+1):(p+q)]
443     sd.theta_gau   <-sd[p+q+1]
444
445     # Ordenamos los coeficientes estimados en una tabla
446     #Reg_Cop_Gausiana <- data.frame(Estimate_Frec=round(beta_gau,4),Std.error.frec=round(sd.beta_gau,4),z_
          value_frec=round(beta_gau/sd.beta_gau,4),"Prob_frec"=round(2*(1-pnorm(abs(beta_gau/sd.beta_gau))),4),
          Estimate_Sev=round(alpha_gau,4),Std.error.sev=round(sd.alpha_gau,4),z_value_sev=round(alpha_gau/sd.alpha_
          gau,4),"Prob_Sev"=round(2*(1-pnorm(abs(alpha_gau/sd.alpha_gau))),4))
447     Reg_Cop_Clayton <- data.frame(Estimate_Frec=round(beta_clay,4),Std.error.frec=0,z_value_frec=0,"Prob_frec"
          =0,Estimate_Sev=round(alpha_clay,4),Std.error.sev=0,z_value_sev=0,"Prob_Sev"=0)
448     var_level<-row.names(Reg_Cop_Clayton)
449     Reg_Cop_Clayton <- Reg_Cop_Clayton %>% as.tibble() %>%  mutate(var_level=var_level) %>% relocate(var_level
          )
450
451     # Agregamos los interceptos
452     temp <- tibble(
453       var_level = c(
454         "Año2022",
455         "PP_Tipo_PolizaIndividual",
456         "PP_Antiguedadc. 02",
457         "PP_Canal_VentaCorredores",
458         "PP_Uso_VehParticular",
459         "PP_Marca_VehL3",
460         "PP_ZonaA2",
461         "PP_GeneroFemenino",
462         "PP_Edad38-42",
463         "PP_SAj.[13000-999999]"
464       ),
465       Estimate_Frec = 0
```

```
    ,
    Estimate_Sev = 0
  )

  # coeficientes del modelo de regresion con dependencia
  Reg_Cop_Clayton <- Reg_Cop_Clayton %>% bind_rows(temp) %>% arrange(var_level) %>%mutate(`exp(Estimate_Frec
    )`=round(exp(Estimate_Frec),4),    `exp(Estimate_Sev)`=round(exp(Estimate_Sev),4))
  Reg_Cop_Clayton <-Reg_Cop_Clayton %>% relocate(`exp(Estimate_Frec)`, .after=Prob_frec)

  # Guardamos el modelo conjunto y los coeficientes
  write.csv(Reg_Cop_Clayton, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_
    GLM_Conjuntos/Perdida_Parcial/Coef_Clay.csv", fileEncoding = "Latin1")
  save(para_optim, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_
    Conjuntos/Perdida_Parcial/JointModel_Clay.RData")

  # Simulacion de la Pérdida Total
  # ================================================================

  # CON DEPENDENCIA
  # ---------------
  lambda_loss_dep  <- exp(var_x%*%beta_clay)*data_CM.Expuesto
  mu_loss_dep      <- exp(var_x%*%alpha_clay)
  delta_loss_dep   <- delta_clay
  # Función de densidad fx

  f_loss <-
    function(loss,mu,d,lam,theta){
      n<-length(loss)
      if (length(lam)==1) lam  <- rep(lam,n)
      if (length(mu)==1) mu    <- rep(mu,n)
      out                      <- vector(length=n)

      for(i in 1:n){
        N  <-1:20
        v  <-ppois(N,lam[i])
        vv <-ppois(N-1,lam[i])
        u  <-pgam(loss[i]/N,mu[i],d)

        Der_cop <-Du_Clay(u,v,theta)-Du_Clay(u,vv,theta)
        dummy   <-Der_cop*dgam(loss[i]/N,mu[i],d)/N
        out[i]  <-sum(dummy)
      }

      out[loss<=0]=0
      return(out)
    }

  # Función de densidad acumulada Fx
  F_loss <-
    function(loss,mu,d,lam,theta){
      out<-vector(length=length(loss))

      for(i in 1:length(loss)){
        floss <- function(s){
          f_loss(s,mu[i],d,lam[i],theta)
        }
        out[i] <- integrate(floss,0,loss[i]).value
      }
      return(out)
    }

  # Estimación del Total Loss (fuerte dedicacion computacional)

  #k <-length(lambda_loss_dep)
  k <- 100#length(lambda_loss_dep) # Cantidad de numeros aleatorios (Debe ser del tamaño de lambda)
  m <- 10 # Cantidad de muestras aleatorias (Deben ser 10 mil)
  L <- vector(length=k)
  S <- vector(length=m)

  for(j in 1:m){
    r_uni<-runif(k)
    for(i in 1:k){
      f_root <-function(s){
        F_loss(s,mu_loss_dep[i],delta_loss_dep,lambda_loss_dep[i],theta_clay)-r_uni[i]
      }
```

```
539           tryCatch(
540             error = function(cnd) loss<-mu_loss_dep[i]*lambda_loss_dep[i],
541             loss<-uniroot(f_root,lower = 0,upper = 500000).root
542           )
543
544           #print(loss.root)
545           L[i]<-loss
546           perc<-paste(i/k*100,"%")
547           print(perc)
548         }
549         S[j]<-sum(L)
550         perc<-paste(j/m*100,"%")
551         print(perc)
552         #print(S[j])
553       }
554       S
555
556       # CON INDEPENDENCIA
557       # -----------------
558       lambda_loss  <-predict(glm.Frec.PP, newdata=data_CM, type='response') # CAMBIAR LOS DATOS A UN MODELO DE
            RIESGO COLECTIVO
559       mu_loss      <-predict(glm.Sev.PP, newdata=data_CM, type='response')
560       delta_loss   <-delta_0
561
562       M<-seq(1:10000)
563       for(i in 1:10000){
564         muestra_cant<-rpois(length(lambda_loss),lambda_loss)
565         S<-vector(length=length(lambda_loss))
566         for(j in 1:length(muestra_cant)){
567           if(muestra_cant[j]==0){
568             S[j]=0
569           }else{
570             S[j]=sum(rgam(n =muestra_cant[j],mu_loss[j],delta_loss ))
571           }
572         }
573
574         M[i] <- sum(S)
575         porc<-paste((i/10000)*100,"%")
576         print(porc)
577         #print(M[i])
578       }
579
580       hist(M)
581       # Simulacion de la Frecuencia con  Independencia
582       #ADECUAR LOS DATOS DE SIMULACION A UN ESTRUCTURA DE MODELO COLECTIVO DE RIESGO
583       # Frecuencia
584       N<-seq(1:10000)
585       for(i in 1:10000){
586         N[i] <- sum(simulate(glm.Frec.PP,nsim=1, type="response" ).sim_1)/sum(data_model.Expuesto)
587         porc<-paste((i/10000)*100,"%")
588         print(porc)
589       }
590       hist(N)
591
592
593       ######################################################################
594       # Limpiamos el ambiente de variables
595       rm(list = ls())
596       gc()
597
598 ##########################################################################
599 # REGRESION GUMBEL
600 ##########################################################################
601
602       # Set Directory
603       setwd("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Perdida_
            Parcial")
604
605       # Carga los modelos GLM MARGINALES calculados en otos querys
606       load("glm.Frec.PP.RData")
607       load("glm.Sev.PP.RData")
608       load("Data_PP.RData")
609       load("Data_PP_model.RData")
610       load("Data_PP_model_CM.RData")
611
612       # Creamos las variables de respuestas y la matriz de datos de cada glm marginal
```

```
613     # SEVERIDAD
614     x <- data_CM.Incurrido/data_CM.Cantidad #(Costo Medio)
615     var_x <-model.matrix(glm.Sev.PP)  #data_CM[,3:12]
616     # FRECUENCIA
617     y <- data_model.Cantidad
618     Expuestos <- data_model.Expuesto
619     var_y <- model.matrix(glm.Frec.PP)   #data[,3:12]
620
621     # Estimacion de parámetros iniciales utilizando las distribuciones marginales
622     # SEVERIDAD
623     sd.alpha <- sqrt(diag(vcov(glm.Sev.PP)))
624     alpha_0  <- glm.Sev.PP.coefficients
625     delta_0  <- summary(glm.Sev.PP).dispersion
626     mu_0     <- exp(var_x%*%alpha_0)
627
628     # FRECUENCIA
629     beta_0   <- glm.Frec.PP.coefficients
630     sd.beta  <- sqrt(diag(vcov(glm.Frec.PP)))
631     lambda_0 <- exp(var_y%*%beta_0)*Expuestos
632
633     # REGRESION MEDIANTE COPULAS
634
635     # COPULA GUMBEL
636     family=4
637     # Estimación inicial del parámetro theta para estimarlo mediante el algoritmo de MPLE
638     theta_ini <- BiCopEst(rank(data_model.Incurrido-exp(var_y%*%alpha_0) )/(length(y)+1),rank(y*Expuestos-
          lambda_0)/(length(y)+1), family=family).par
639     #theta_ini <- BiCopEst(rank(data_CM.Incurrido-exp(var_x%*%alpha_0))/(length(x)+1),rank(data_CM.Cantidad*
          data_CM.Expuesto-exp(var_x%*%beta_0)*data_CM.Expuesto)/(length(x)+1), family=family).par
640     tau_ini   <- BiCopPar2Tau(par=theta_ini,family = family)
641
642     # Estimación de las seudo-observaciones
643     u<-pgam(x,mu_0,delta_0)
644     v   <- ppois(data_CM.Cantidad,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)
645     vv  <- ppois(data_CM.Cantidad-1,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)
646
647     # Creamos la función primera derivada de la Función Copula
648     Du_Gum <- function(u,v,theta){
649       u[u<=0]=0.0001
650       u[u>1]=1
651       v[v<=0]=0.0001
652       v[v>1]=1
653       #out <- pnorm((qnorm(v)-theta*qnorm(u))/(sqrt(1-theta^2)))
654       #out <- (u^(-theta)+v^(-theta)-1)^((-1/theta)-1)*(u^(-theta-1))
655       out <- (u^-1)*exp(-((-log(u))^theta+(-log(v))^theta)^(1/theta))
656       return(out)
657     }
658
659     # Función a maximizar
660     f_aux <- function(para){
661       D_u_0<-Du_Gum(u,v,para)-Du_Gum(u,vv,para)
662       D_u_0[D_u_0<=0]=1
663       D_u_0 <- log(D_u_0)
664       out<-(sum(D_u_0))
665       return(out)
666     }
667
668     # Proceso de maximización
669     #para_ini <- theta2z(theta_ini,family = family)
670     para.ifm <- optim(theta_ini,f_aux,method = "BFGS").par
671     #theta_0  <- z2theta(para.ifm,family = family)
672     theta_0  <- para.ifm # Este será nuestro parámetro inicial estimado con el método IFM y que luego usaremos
          para la regresión conjunta
673     tau      <-BiCopPar2Tau(par=theta_0,family=family) # tau de Khendal para estimar el grado de dependencia
674
675     # Estimamos los coeficientes de regresión y el parámetro cópula mediante máxima verosimilitud
676
677     # Creamos nuestra función a maximizar
678     ######################################################################
679     f_aux_reg <- function(para){
680
681       p        <- ncol(var_x)
682       q        <- ncol(var_x)
683       alpha    <- para[1:p]
684       beta     <- para[(p+1):(p+q)]
685       # theta    <- z2theta(para[p+q+1],family)
```

```
    theta    <- z2theta(para[p+q+1],family = family)
    delta    <- para[p+q+2]
    lambda   <- as.vector(exp(var_y%*%beta)*data_model.Expuesto)
    mu       <- as.vector(exp(var_y%*%alpha))
    #dummy    <- density_joint(x,y,mu,delta,lambda,theta,family=family,zt=FALSE)
    u2 <- pgam(data_model.Incurrido,mu,delta)
    v2<-ppois(y,lambda)
    vv2<-ppois(y-1,lambda)

    marginal.x <- dgam(data_model.Incurrido,mu,delta)
    marginal.x[marginal.x>=1]=0
    marginal.x[marginal.x<=0]=0
    marginal.x[is.na(marginal.x)]=0
    marginal.x[marginal.x==-Inf]=0
    marginal.x[marginal.x==Inf]=0

    par_der    <- Du_Clay(u2,v2,theta)
    par_der1   <- Du_Clay(u2,vv2,theta)

    dummy<- par_der-par_der1

    dummy[y==0]=par_der[y==0]

    out<-marginal.x*dummy

    out[out<=0]=1e-10
    ll       <- -sum(log(out))

    #if(negative==TRUE) ll <- (-ll)
    return(ll)
}
#################################################################
para_0<-c(alpha_0,beta_0,theta2z(theta_0,family=family),delta_0) # Parámetros iniciales
a<-now()
para_optim <- optim(para_0,f_aux_reg,method = "BFGS",hessian = FALSE) # Estimación Maxima verosmil de los
  parámetros (Toma hasta 3h sin matriz hessiana. 8h con Matriz hessiana)
b<-now()
b-a #3.485547 hours
#load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_Conjuntos/Perdida_
  Parcial/JointModel_Gaus.RData")
para_optim.value # Log_Likelihood
vector <- para_optim.par
p        <- ncol(var_y)
q        <- ncol(var_y)
Log_Likelihood<-para_optim.value
# Parámetros optimizados
alpha_gum<-vector[1:p]
beta_gum<-vector[(p+1):(p+q)]
theta_gum=z2theta(vector[p+q+1],family) #-0.09506928
delta_gum<-exp(vector[p+q+2]) #4.271691
tau_gum<-BiCopPar2Tau(par=theta_gum,family=family)#-0.06061453
head(beta_gum)
head(alpha_gum)

# Determinamos los errores estandar para el test de Wald
hessian_gum    <- para_optim.hessian
Hinv           <- ginv(hessian_gum)
sd             <- sqrt(diag(Hinv))
sd.alpha_gum   <-sd[1:p]
sd.beta_gum    <- sd[(p+1):(p+q)]
sd.theta_gum   <-sd[p+q+1]

# Ordenamos los coeficientes estimados en una tabla
#Reg_Cop_Gausiana <- data.frame(Estimate_Frec=round(beta_gau,4),Std.error.frec=round(sd.beta_gau,4),z_
  value_frec=round(beta_gau/sd.beta_gau,4),"Prob_frec"=round(2*(1-pnorm(abs(beta_gau/sd.beta_gau))),4),
  Estimate_Sev=round(alpha_gau,4),Std.error.sev=round(sd.alpha_gau,4),z_value_sev=round(alpha_gau/sd.alpha_
  gau,4),"Prob_Sev"=round(2*(1-pnorm(abs(alpha_gau/sd.alpha_gau))),4))
Reg_Cop_Gumbel <- data.frame(Estimate_Frec=round(beta_gum,4),Std.error.frec=0,z_value_frec=0,"Prob_frec"
  =0,Estimate_Sev=round(alpha_gum,4),Std.error.sev=0,z_value_sev=0,"Prob_Sev"=0)
var_level<-row.names(Reg_Cop_Gumbel)
Reg_Cop_Gumbel <- Reg_Cop_Gumbel %>% as.tibble() %>%  mutate(var_level=var_level) %>% relocate(var_level)

# Agregamos los interceptos
temp <- tibble(
  var_level = c(
    "Año2022",
```

```
        "PP_Tipo_PolizaIndividual",
        "PP_Antiguedadc. 02",
        "PP_Canal_VentaCorredores",
        "PP_Uso_VehParticular",
        "PP_Marca_VehL3",
        "PP_ZonaA2",
        "PP_GeneroFemenino",
        "PP_Edad38-42",
        "PP_SAj.[13000-999999]"
      ),
      Estimate_Frec = 0
      ,
      Estimate_Sev = 0
    )

    # coeficientes del modelo de regresion con dependencia
    Reg_Cop_Gumbel <- Reg_Cop_Gumbel %>% bind_rows(temp) %>% arrange(var_level) %>% mutate(`exp(Estimate_Frec)
      `=round(exp(Estimate_Frec),4),     `exp(Estimate_Sev)`=round(exp(Estimate_Sev),4))
    Reg_Cop_Gumbel <-Reg_Cop_Gumbel %>%  relocate(`exp(Estimate_Frec)`, .after=Prob_frec)
    # Guardamos el modelo conjunto y los coeficientes
    write.csv(Reg_Cop_Gumbel, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_
      GLM_Conjuntos/Perdida_Parcial/Coef_Gumb.csv", fileEncoding = "Latin1")
    save(para_optim, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_
      Conjuntos/Perdida_Parcial/JointModel_Gumb.RData")

    # Simulacion de la Pérdida Total
    # ================================================================

    # CON DEPENDENCIA
    # ---------------
    lambda_loss_dep  <- exp(var_x%*%beta_clay)*data_CM.Expuesto
    mu_loss_dep      <- exp(var_x%*%alpha_clay)
    delta_loss_dep   <- delta_clay
    # Función de densidad fx

    f_loss <-
      function(loss,mu,d,lam,theta){
        n<-length(loss)
        if (length(lam)==1) lam  <- rep(lam,n)
        if (length(mu)==1) mu    <- rep(mu,n)
        out                      <- vector(length=n)

        for(i in 1:n){
          N  <-1:20
          v  <-ppois(N,lam[i])
          vv <-ppois(N-1,lam[i])
          u  <-pgam(loss[i]/N,mu[i],d)

          Der_cop <-Du_Clay(u,v,theta)-Du_Clay(u,vv,theta)
          dummy   <-Der_cop*dgam(loss[i]/N,mu[i],d)/N
          out[i]  <-sum(dummy)
        }

        out[loss<=0]=0
        return(out)
      }

    # Función de densidad acumulada Fx
    F_loss <-
      function(loss,mu,d,lam,theta){
        out<-vector(length=length(loss))

        for(i in 1:length(loss)){
          floss <- function(s){
            f_loss(s,mu[i],d,lam[i],theta)
          }
          out[i] <- integrate(floss,0,loss[i]).value
        }
        return(out)
      }

    # Estimación del Total Loss (fuerte dedicacion computacional)

    #k <-length(lambda_loss_dep)
    k <- 100#length(lambda_loss_dep) # Cantidad de numeros aleatorios (Debe ser del tamaño de lambda)
    m <- 10 # Cantidad de muestras aleatorias (Deben ser 10 mil)
```

```
829     L <- vector(length=k)
830     S <- vector(length=m)
831
832     for(j in 1:m){
833       r_uni<-runif(k)
834       for(i in 1:k){
835         f_root <-function(s){
836           F_loss(s,mu_loss_dep[i],delta_loss_dep,lambda_loss_dep[i],theta_clay)-r_uni[i]
837         }
838
839         tryCatch(
840           error = function(cnd) loss<-mu_loss_dep[i]*lambda_loss_dep[i],
841           loss<-uniroot(f_root,lower = 0,upper = 500000).root
842         )
843
844         #print(loss.root)
845         L[i]<-loss
846         perc<-paste(i/k*100,"%")
847         print(perc)
848       }
849       S[j]<-sum(L)
850       perc<-paste(j/m*100,"%")
851       print(perc)
852       #print(S[j])
853     }
854     S
855
856     # CON INDEPENDENCIA
857     # -----------------
858     lambda_loss  <-predict(glm.Frec.PP, newdata=data_CM, type='response') # CAMBIAR LOS DATOS A UN MODELO DE
          RIESGO COLECTIVO
859     mu_loss      <-predict(glm.Sev.PP, newdata=data_CM, type='response')
860     delta_loss   <-delta_0
861
862     M<-seq(1:10000)
863     for(i in 1:10000){
864       muestra_cant<-rpois(length(lambda_loss),lambda_loss)
865       S<-vector(length=length(lambda_loss))
866       for(j in 1:length(muestra_cant)){
867         if(muestra_cant[j]==0){
868           S[j]=0
869         }else{
870           S[j]=sum(rgam(n =muestra_cant[j],mu_loss[j],delta_loss ))
871         }
872       }
873
874       M[i] <- sum(S)
875       porc<-paste((i/10000)*100,"%")
876       print(porc)
877       #print(M[i])
878     }
879
880     hist(M)
881
882     # Simulacion de la Frecuencia con  Independencia
883     #ADECUAR LOS DATOS DE SIMULACION A UN ESTRUCTURA DE MODELO COLECTIVO DE RIESGO
884     # Frecuencia
885     N<-seq(1:10000)
886     for(i in 1:10000){
887       N[i] <- sum(simulate(glm.Frec.PP,nsim=1, type="response" ).sim_1)/sum(data_model.Expuesto)
888       porc<-paste((i/10000)*100,"%")
889       print(porc)
890     }
891     hist(N)
892
893 ########################################################################
894     # Limpiamos el ambiente de variables
895     rm(list = ls())
896     gc()
897
898 ########################################################################
899 # REGRESION FRANK
900 ########################################################################
901
902     # Set Directory
903     setwd("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Perdida_
```

```
    Parcial")

# Carga los modelos GLM MARGINALES calculados en otos querys
load("glm.Frec.PP.RData")
load("glm.Sev.PP.RData")
load("Data_PP.RData")
load("Data_PP_model.RData")
load("Data_PP_model_CM.RData")

# Creamos las variables de respuestas y la matriz de datos de cada glm marginal
# SEVERIDAD
x <- data_CM.Incurrido/data_CM.Cantidad #(Costo Medio)
var_x <-model.matrix(glm.Sev.PP)  #data_CM[,3:12]
# FRECUENCIA
y <- data_model.Cantidad
Expuestos <- data_model.Expuesto
var_y <- model.matrix(glm.Frec.PP)   #data[,3:12]

# Estimacion de parámetros iniciales utilizando las distribuciones marginales
# SEVERIDAD
sd.alpha <- sqrt(diag(vcov(glm.Sev.PP)))
alpha_0  <- glm.Sev.PP.coefficients
delta_0  <- summary(glm.Sev.PP).dispersion
mu_0     <- exp(var_x%*%alpha_0)

# FRECUENCIA
beta_0   <- glm.Frec.PP.coefficients
sd.beta  <- sqrt(diag(vcov(glm.Frec.PP)))
lambda_0 <- exp(var_y%*%beta_0)*Expuestos

# REGRESION MEDIANTE COPULAS

# COPULA GUMBEL
family=5
# Estimación inicial del parámetro theta para estimarlo mediante el algoritmo de MPLE
theta_ini <- BiCopEst(rank(data_model.Incurrido-exp(var_y%*%alpha_0) )/(length(y)+1),rank(y*Expuestos-
  lambda_0)/(length(y)+1), family=family).par
#theta_ini <- BiCopEst(rank(data_CM.Incurrido-exp(var_x%*%alpha_0))/(length(x)+1),rank(data_CM.Cantidad*
  data_CM.Expuesto-exp(var_x%*%beta_0)*data_CM.Expuesto)/(length(x)+1), family=family).par
tau_ini   <- BiCopPar2Tau(par=theta_ini,family = family)

# Estimación de las seudo-observaciones
u<-pgam(x,mu_0,delta_0)
v   <- ppois(data_CM.Cantidad,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)
vv  <- ppois(data_CM.Cantidad-1,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)

# Creamos la función primera derivada de la Función Copula
Du_Frank <- function(u,v,theta){
  u[u<=0]=0.0001
  u[u>1]=1
  v[v<=0]=0.0001
  v[v>1]=1
  #out <- pnorm((qnorm(v)-theta*qnorm(u))/(sqrt(1-theta^2)))
  #out <- (u^(-theta)+v^(-theta)-1)^((-1/theta)-1)*(u^(-theta-1))
  #out <- (u^-1)*exp(-((-log(u))^theta+(-log(v))^theta)^(1/theta))
  out <- (exp(theta)*(exp(theta*v)-1))/(exp(theta*(u+1))+exp(theta*(v+1))-exp(theta)-exp(theta*(u+v)))
  return(out)
}

# Función a maximizar
f_aux <- function(para){
  D_u_0<-Du_Frank(u,v,para)-Du_Frank(u,vv,para)
  D_u_0[D_u_0<=0]=1
  D_u_0 <- log(D_u_0)
  out<-(-sum(D_u_0))
  return(out)
}

# Proceso de maximización
#para_ini <- theta2z(theta_ini,family = family)
para.ifm <- optim(theta_ini,f_aux,method = "BFGS").par
#theta_0  <- z2theta(para.ifm,family = family)
theta_0  <- para.ifm # Este será nuestro parámetro inicial estimado con el método IFM y que luego usaremos
   para la regresión conjunta
tau      <-BiCopPar2Tau(par=theta_0,family=family) # tau de Khendal para estimar el grado de dependencia
```

```
976  # Estimamos los coeficientes de regresión y el parámetro cópula mediante máxima verosimilitud
977
978  # Creamos nuestra función a maximizar
979  ######################################################################
980  f_aux_reg <- function(para){
981
982    p        <- ncol(var_x)
983    q        <- ncol(var_x)
984    alpha    <- para[1:p]
985    beta     <- para[(p+1):(p+q)]
986    # theta    <- z2theta(para[p+q+1],family)
987    theta    <- z2theta(para[p+q+1],family = family)
988    delta    <- para[p+q+2]
989    lambda   <- as.vector(exp(var_y%*%beta)*data_model.Expuesto)
990    mu       <- as.vector(exp(var_y%*%alpha))
991    #dummy    <- density_joint(x,y,mu,delta,lambda,theta,family=family,zt=FALSE)
992    u2 <- pgam(data_model.Incurrido,mu,delta)
993    v2<-ppois(y,lambda)
994    vv2<-ppois(y-1,lambda)
995
996    marginal.x <- dgam(data_model.Incurrido,mu,delta)
997    marginal.x[marginal.x>=1]=0
998    marginal.x[marginal.x<=0]=0
999    marginal.x[is.na(marginal.x)]=0
1000   marginal.x[marginal.x==-Inf]=0
1001   marginal.x[marginal.x==Inf]=0
1002
1003   par_der    <- Du_Frank(u2,v2,theta)
1004   par_der1   <- Du_Frank(u2,vv2,theta)
1005
1006   dummy<- par_der-par_der1
1007
1008   dummy[y==0]=par_der[y==0]
1009
1010   out<-marginal.x*dummy
1011
1012   out[out<=0]=1e-10
1013   ll       <- -sum(log(out))
1014
1015   #if(negative==TRUE) ll <- (-ll)
1016   return(ll)
1017 }
1018
1019 ######################################################################
1020 para_0<-c(alpha_0,beta_0,theta2z(theta_0,family=family),delta_0) # Parámetros iniciales
1021 a<-now()
1022 para_optim <- optim(para_0,f_aux_reg,method = "BFGS",hessian = FALSE) # Estimación Maxima verosmil de los
       parámetros (Toma hasta 3h sin matriz hessiana. 8h con Matriz hessiana)
1023 b<-now()
1024 b-a #2.82135 hours
1025 #load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_Conjuntos/Perdida_
       Parcial/JointModel_Gaus.RData")
1026 para_optim.value # Log_Likelihood
1027 vector <- para_optim.par
1028 p        <- ncol(var_y)
1029 q        <- ncol(var_y)
1030 Log_Likelihood<-para_optim.value
1031 # Parámetros optimizados
1032 alpha_frank<-vector[1:p]
1033 beta_frank<-vector[(p+1):(p+q)]
1034 theta_frank=z2theta(vector[p+q+1],family)
1035 delta_frank<-exp(vector[p+q+2]) #4.271691
1036 tau_frank<-BiCopPar2Tau(par=theta_frank,family=family)
1037 head(beta_frank)
1038 head(alpha_frank)
1039
1040 # Determinamos los errores estandar para el test de Wald
1041 hessian_frank    <- para_optim.hessian
1042 Hinv             <- ginv(hessian_frank)
1043 sd               <- sqrt(diag(Hinv))
1044 sd.alpha_frank   <-sd[1:p]
1045 sd.beta_frank    <- sd[(p+1):(p+q)]
1046 sd.theta_frank   <-sd[p+q+1]
1047
1048 # Ordenamos los coeficientes estimados en una tabla
1049 #Reg_Cop_Gausiana <- data.frame(Estimate_Frec=round(beta_gau,4),Std.error.frec=round(sd.beta_gau,4),z_
```

```
      value_frec=round(beta_gau/sd.beta_gau,4),"Prob_frec"=round(2*(1-pnorm(abs(beta_gau/sd.beta_gau))),4),
      Estimate_Sev=round(alpha_gau,4),Std.error.sev=round(sd.alpha_gau,4),z_value_sev=round(alpha_gau/sd.alpha_
      gau,4),"Prob_Sev"=round(2*(1-pnorm(abs(alpha_gau/sd.alpha_gau))),4))
Reg_Cop_Frank <- data.frame(Estimate_Frec=round(beta_frank,4),Std.error.frec=0,z_value_frec=0,"Prob_frec"
  =0,Estimate_Sev=round(alpha_frank,4),Std.error.sev=0,z_value_sev=0,"Prob_Sev"=0)
var_level<-row.names(Reg_Cop_Frank)
Reg_Cop_Frank <- Reg_Cop_Frank %>% as.tibble() %>%  mutate(var_level=var_level) %>% relocate(var_level)

# Agregamos los interceptos
temp <- tibble(
  var_level = c(
    "Año2022",
    "PP_Tipo_PolizaIndividual",
    "PP_Antiguedadc. 02",
    "PP_Canal_VentaCorredores",
    "PP_Uso_VehParticular",
    "PP_Marca_VehL3",
    "PP_ZonaA2",
    "PP_GeneroFemenino",
    "PP_Edad38-42",
    "PP_SAj.[13000-999999]"
  ),
  Estimate_Frec = 0
  ,
  Estimate_Sev = 0
)

# coeficientes del modelo de regresion con dependencia
Reg_Cop_Frank <- Reg_Cop_Frank %>% bind_rows(temp) %>% arrange(var_level) %>% mutate(`exp(Estimate_Frec)`=
  round(exp(Estimate_Frec),4),`exp(Estimate_Sev)`=round(exp(Estimate_Sev),4))
Reg_Cop_Frank <-Reg_Cop_Frank %>%  relocate(`exp(Estimate_Frec)`, .after=Prob_frec)

# Guardamos el modelo conjunto y los coeficientes
write.csv(Reg_Cop_Frank, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_
  GLM_Conjuntos/Perdida_Parcial/Coef_Frank.csv", fileEncoding = "Latin1")
save(para_optim, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_
  Conjuntos/Perdida_Parcial/JointModel_Frank.RData")

# Simulacion de la Pérdida Total
# ================================================================

# CON DEPENDENCIA
# ---------------
lambda_loss_dep  <- exp(var_x%*%beta_frank)*data_CM.Expuesto
mu_loss_dep      <- exp(var_x%*%alpha_frank)
delta_loss_dep   <- delta_frank


# Función de densidad fx

f_loss <-
  function(loss,mu,d,lam,theta){
    n<-length(loss)
    if (length(lam)==1) lam  <- rep(lam,n)
    if (length(mu)==1) mu    <- rep(mu,n)
    out                      <- vector(length=n)

    for(i in 1:n){
      N  <-1:20
      v  <-ppois(N,lam[i])
      vv <-ppois(N-1,lam[i])
      u  <-pgam(loss[i]/N,mu[i],d)

      Der_cop <-Du_Frank(u,v,theta)-Du_Frank(u,vv,theta) # Primera derivada de la función copula
      dummy   <-Der_cop*dgam(loss[i]/N,mu[i],d)/N # Teorela del Total Loss
      out[i]  <-sum(dummy)
    }

    out[loss<=0]=0
    return(out)
  }

# Función de densidad acumulada Fx
F_loss <-
  function(loss,mu,d,lam,theta){
    out<-vector(length=length(loss))
```

```
1119
1120          for(i in 1:length(loss)){
1121            floss <- function(s){
1122              f_loss(s,mu[i],d,lam[i],theta)
1123            }
1124            out[i] <- integrate(floss,0,loss[i]).value # Integral para el cálculo del a F(x) a partir de f(x)
1125          }
1126          return(out)
1127        }
1128
1129      # Estimación del Total Loss (fuerte dedicacion computacional)
1130
1131      #k <-length(lambda_loss_dep)
1132      k <- 100#length(lambda_loss_dep) # Cantidad de numeros aleatorios (Debe ser del tamaño de lambda)
1133      m <- 10 # Cantidad de muestras aleatorias (Deben ser 10 mil)
1134      L <- vector(length=k)
1135      S <- vector(length=m)
1136
1137      for(j in 1:m){
1138        r_uni<-runif(k)
1139        for(i in 1:k){
1140          f_root <-function(s){
1141            F_loss(s,mu_loss_dep[i],delta_loss_dep,lambda_loss_dep[i],theta_frank)-r_uni[i]
1142          }
1143
1144          tryCatch(
1145            error = function(cnd) loss<-mu_loss_dep[i]*lambda_loss_dep[i],
1146            loss<-uniroot(f_root,lower = 0,upper = 500000).root
1147          )
1148
1149          #print(loss.root)
1150          L[i]<-loss
1151          perc<-paste(i/k*100,"%")
1152          print(perc)
1153        }
1154        S[j]<-sum(L)
1155        perc<-paste(j/m*100,"%")
1156        print(perc)
1157        #print(S[j])
1158      }
1159      S
1160
1161      # CON INDEPENDENCIA
1162      # ------------------
1163      lambda_loss  <-predict(glm.Frec.PP, newdata=data_CM, type='response') # CAMBIAR LOS DATOS A UN MODELO DE
           RIESGO COLECTIVO
1164      mu_loss      <-predict(glm.Sev.PP, newdata=data_CM, type='response')
1165      delta_loss   <-delta_0
1166
1167      M<-seq(1:10000)
1168      for(i in 1:10000){
1169        muestra_cant<-rpois(length(lambda_loss),lambda_loss)
1170        S<-vector(length=length(lambda_loss))
1171        for(j in 1:length(muestra_cant)){
1172          if(muestra_cant[j]==0){
1173            S[j]=0
1174          }else{
1175            S[j]=sum(rgam(n =muestra_cant[j],mu_loss[j],delta_loss ))
1176          }
1177        }
1178
1179        M[i] <- sum(S)
1180        porc<-paste((i/10000)*100,"%")
1181        print(porc)
1182        #print(M[i])
1183      }
1184
1185      hist(M)
1186
1187      # Simulacion de la Frecuencia con  Independencia
1188      #ADECUAR LOS DATOS DE SIMULACION A UN ESTRUCTURA DE MODELO COLECTIVO DE RIESGO
1189      # Frecuencia
1190      N<-seq(1:10000)
1191      for(i in 1:10000){
1192        N[i] <- sum(simulate(glm.Frec.PP,nsim=1, type="response" ).sim_1)/sum(data_model.Expuesto)
1193        porc<-paste((i/10000)*100,"%")
```

```
      print(porc)
    }
    hist(N)
```

C.4. Modelo Conjunto - Pérdida Total

```
################################################################################
################################################################################
####                                                                        ######
####                    REGRESION BASADO EN COPULAS                         ######
####                          PERDIDA TOTAL                                 ######
####                                                                        ######
################################################################################
################################################################################

# Cargamos las librerías necesarias

  #library(copula)
  library(RODBC)
  library(tidyverse)
  library(MASS)
  #library(GJRM)
  #library(devtools)
  library(CopulaRegression)
  library(VineCopula)
  #library(optimx)
  #install_url('http://cran.r-project.org/src/contrib/Archive/CopulaRegression/CopulaRegression_0.1-5.tar.gz')

#########################################################################
# REGRESION GAUSSIANA
#########################################################################

# Set Directory
  setwd("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Perdida_Total
      ")

# Carga los modelos GLM MARGINALES calculados en otos querys
  load("glm.Frec.PT.RData")
  load("glm.Sev.PT.RData")
  load("Data_PT.RData")
  load("Data_PT_model.RData")
  load("Data_PT_model_CM.RData")

# Creamos las variables de respuestas y la matriz de datos de cada glm marginal
  # SEVERIDAD
  x <- data_CM.Incurrido/data_CM.Cantidad #(Costo Medio)
  var_x <-model.matrix(glm.Sev.PT)  #data_CM[,3:12]
  # FRECUENCIA
  y <- data_model.Cantidad
  Expuestos <- data_model.Expuesto
  var_y <- model.matrix(glm.Frec.PT)   #data[,3:12]

# Estimacion de parámetros iniciales utilizando las distribuciones marginales
  # SEVERIDAD
  sd.alpha <- sqrt(diag(vcov(glm.Sev.PT)))
  alpha_0  <- glm.Sev.PT.coefficients
  delta_0  <- summary(glm.Sev.PT).dispersion
  mu_0     <- exp(var_x%*%alpha_0)

  # FRECUENCIA
  beta_0   <- glm.Frec.PT.coefficients
  sd.beta  <- sqrt(diag(vcov(glm.Frec.PT)))
  lambda_0 <- exp(var_y%*%beta_0)*Expuestos

# REGRESION MEDIANTE COPULAS

  # COPULA GAUSSIANA
  family=1
  # Estimación inicial del parámetro theta para estimarlo mediante el algoritmo de MPLE
  theta_ini <- BiCopEst(rank(data_model.Incurrido-exp(var_y%*%alpha_0) )/(length(y)+1),rank(y*Expuestos-lambda
      _0)/(length(y)+1), family=family).par
  #theta_ini <- BiCopEst(rank(data_CM.Incurrido-exp(var_x%*%alpha_0))/(length(x)+1),rank(data_CM.Cantidad*data
      _CM.Expuesto-exp(var_x%*%beta_0)*data_CM.Expuesto)/(length(x)+1), family=family).par
  tau_ini    <- BiCopPar2Tau(par=theta_ini,family = family)

```

```
# Estimación de las seudo-observaciones
u<-pgam(x,mu_0,delta_0)
v    <- ppois(data_CM.Cantidad,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)
vv   <- ppois(data_CM.Cantidad-1,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)

# Creamos la función primera derivada de la Función Copula
  Du_Gaus <- function(u,v,theta){
    u[u<=0]=0.001
    u[u>=1]=0.999
    v[v<=0]=0.001
    v[v>=1]=0.999
    out <- pnorm((qnorm(v)-theta*qnorm(u))/(sqrt(1-theta^2)))
  return(out)
  }

# Función a maximizar
f_aux <- function(para){
  D_u_0<-Du_Gaus(u,v,para)-Du_Gaus(u,vv,para)
  D_u_0[D_u_0<=0]=1
  D_u_0 <- log(D_u_0)
  out<-(-sum(D_u_0))
  return(out)
}

# Proceso de maximización
#para_ini <- theta2z(theta_ini,family = family)
para.ifm <- optim(theta_ini,f_aux,method = "BFGS").par
#theta_0  <- z2theta(para.ifm,family = family)
theta_0  <- para.ifm # Este será nuestro parámetro inicial estimado con el método IFM y que luego usaremos
    para la regresión conjunta
tau      <-BiCopPar2Tau(par=theta_0,family=family) # tau de Khendal para estimar el grado de dependencia

# Estimamos los coeficientes de regresión y el parámetro cópula mediante máxima verosimilitud

# Creamos nuestra función a maximizar
#######################################################################
  f_aux_reg <- function(para){

      p        <- ncol(var_x)
      q        <- ncol(var_x)
      alpha    <- para[1:p]
      beta     <- para[(p+1):(p+q)]
      # theta    <- z2theta(para[p+q+1],family)
      theta    <- para[p+q+1]
      delta    <- para[p+q+2]
      lambda   <- as.vector(exp(var_y%*%beta)*data_model.Expuesto)
      mu       <- as.vector(exp(var_y%*%alpha))
      #dummy    <- density_joint(x,y,mu,delta,lambda,theta,family=family,zt=FALSE)
      u2 <- pgam(data_model.Incurrido,mu,delta)
      v2<-ppois(y,lambda)
      vv2<-ppois(y-1,lambda)

      marginal.x <- dgam(data_model.Incurrido,mu,delta)
      marginal.x[marginal.x>=1]=0
      marginal.x[marginal.x<=0]=0
      marginal.x[is.na(marginal.x)]=0
      marginal.x[marginal.x==-Inf]=0
      marginal.x[marginal.x==Inf]=0

      par_der    <- Du_Gaus(u2,v2,theta)
      par_der1   <- Du_Gaus(u2,vv2,theta)

      dummy<- par_der-par_der1

      dummy[y==0]=par_der[y==0]

      out<-marginal.x*dummy

      out[out<=0]=1e-10
      ll       <- -sum(log(out))

      #if(negative==TRUE) ll <- (-ll)
      return(ll)
}

#######################################################################
```

```
142
143 para_0<-c(alpha_0,beta_0,theta_0,delta_0) # Parámetros iniciales
144 a<-now()
145 para_optim <- optim(para_0,f_aux_reg,method = "BFGS",hessian = TRUE) # Demora 6min
146 b<-now()
147 b-a
148 #load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_Conjuntos/Perdida_
    Parcial/JointModel_Gaus.RData")
149 para_optim.value # Log_Likelihood
150 vector <- para_optim.par
151 p        <- ncol(var_y)
152 q        <- ncol(var_y)
153
154 # Parámetros optimizados
155     alpha_gau<-vector[1:p]
156     beta_gau<-vector[(p+1):(p+q)]
157     theta_gau=z2theta(vector[p+q+1],family) #-0.09506928
158     delta_gau<-exp(vector[p+q+2]) #4.271691
159     tau_gau<-BiCopPar2Tau(par=theta_gau,family=family)#-0.06061453
160     head(beta_gau)
161     head(alpha_gau)
162 # Determinamos los errores estandar para el test de Wald
163     hessian_gau   <- para_optim.hessian
164     Hinv          <- ginv(hessian_gau)
165     sd            <- sqrt(diag(Hinv))
166     sd.alpha_gau  <-sd[1:p]
167     sd.beta_gau   <- sd[(p+1):(p+q)]
168     sd.theta_gau  <-sd[p+q+1]
169
170 # Ordenamos los coeficientes estimados en una tabla
171     Reg_Cop_Gausiana <- data.frame(Estimate_Frec=round(beta_gau,4),Std.error.frec=round(sd.beta_gau,4),z_
    value_frec=round(beta_gau/sd.beta_gau,4),"Prob_frec"=round(2*(1-pnorm(abs(beta_gau/sd.beta_gau))),4),
    Estimate_Sev=round(alpha_gau,4),Std.error.sev=round(sd.alpha_gau,4),z_value_sev=round(alpha_gau/sd.alpha_
    gau,4),"Prob_Sev"=round(2*(1-pnorm(abs(alpha_gau/sd.alpha_gau))),4))
172
173     # Agregamos los interceptos
174     temp <- tibble(
175       var_level = c(
176         "Año2022",
177         "PT_Tipo_PolizaIndividual",
178         "PT_Uso_VehParticular",
179         "PT_Marca_VehL4",
180         "PT_ZonaA3",
181         "PT_GeneroFemenino",
182         "PT_Edad43-52"
183       ),
184       Estimate_Frec = 0
185       ,
186       Estimate_Sev = 0
187     )
188
189     # coeficientes del modelo de regresion con dependencia
190     Reg_Cop_Gausiana <- Reg_Cop_Gausiana %>% bind_rows(temp) %>% arrange(var_level) %>% mutate(`exp(Estimate
    _Frec)`=round(exp(Estimate_Frec),4),`exp(Estimate_Sev)`=round(exp(Estimate_Sev),4))
191     Reg_Cop_Gausiana <-Reg_Cop_Gausiana %>%  relocate(`exp(Estimate_Frec)`, .after=Prob_frec)
192
193 # Guardamos el modelo conjunto y los coeficientes
194 write.csv(Reg_Cop_Gausiana, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_
    GLM_Conjuntos/Perdida_Total/Coef_Gaus.csv", fileEncoding = "Latin1")
195 save(para_optim, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_
    Conjuntos/Perdida_Total/JointModel_Gaus.RData")
196
197 # Simulacion de la Pérdida Total
198 # ============================================================================
199
200     # CON DEPENDENCIA
201     # ---------------
202               lambda_loss_dep  <- exp(var_x%*%beta_gau)*data_CM.Expuesto
203               mu_loss_dep      <- exp(var_x%*%alpha_gau)
204               delta_loss_dep   <- delta_gau
205               # Función de densidad fx
206               f_loss <-
207                 function(loss,mu,d,lam,theta){
208                   n<-length(loss)
209                   if (length(lam)==1) lam   <- rep(lam,n)
210                   if (length(mu)==1) mu     <- rep(mu,n)
```

```
                    out                         <- vector(length=n)

                    for(i in 1:n){
                      N  <-1:20
                      v  <-ppois(N,lam[i])
                      vv <-ppois(N-1,lam[i])
                      u  <-pgam(loss[i]/N,mu[i],d)

                      Der_cop <-Du_Gaus(u,v,theta)-Du_Gaus(u,vv,theta)
                      dummy   <-Der_cop*dgam(loss[i]/N,mu[i],d)/N
                      out[i]  <-sum(dummy)
                    }

                    out[loss<=0]=0
                    return(out)
                  }

                # Función de densidad acumulada Fx
                F_loss <-
                  function(loss,mu,d,lam,theta){
                    out<-vector(length=length(loss))

                    for(i in 1:length(loss)){
                      floss <- function(s){
                        f_loss(s,mu[i],d,lam[i],theta)
                      }
                      out[i] <- integrate(floss,0,loss[i]).value
                    }
                    return(out)
                  }

                # Estimación del Total Loss (fuerte dedicacion computacional)

                #k <-length(lambda_loss_dep)
                k <- length(lambda_loss_dep) # Cantidad de numeros aleatorios (Debe ser del tamaño de lambda)
                m <- 100 # Cantidad de muestras aleatorias (Deben ser 10 mil)
                L <- vector(length=k)
                S <- vector(length=m)

                for(j in 1:m){
                  r_uni<-runif(k)
                  for(i in 1:k){
                    f_root <-function(s){
                      F_loss(s,mu_loss_dep[i],delta_loss_dep,lambda_loss_dep[i],theta_gau)-r_uni[i]
                    }

                    tryCatch(
                      error = function(cnd) mu_loss_dep[i]*lambda_loss_dep[i],
                      loss<-uniroot(f_root,lower = 0,upper = 500000)
                    )

                    #print(loss.root)
                    L[i]<-loss.root
                    perc<-paste(i/k*100,"%")
                    print(perc)
                  }
                  S[j]<-sum(L)
                  perc<-paste(j/m*100,"%")
                  print(perc)
                  #print(S[j])
                }
        head(S)

# CON INDEPENDENCIA
# -----------------
                lambda_loss  <-predict(glm.Frec.PT, newdata=data_CM, type='response') # CAMBIAR LOS DATOS A UN
 MODELO DE RIESGO COLECTIVO
                mu_loss      <-predict(glm.Sev.PT, newdata=data_CM, type='response')
                delta_loss   <-delta_0

                M<-seq(1:10000)
                for(i in 1:10000){
                  muestra_cant<-rpois(length(lambda_loss),lambda_loss)
                  S<-vector(length=length(lambda_loss))
                  for(j in 1:length(muestra_cant)){
                    if(muestra_cant[j]==0){
```

```
                    S[j]=0
                  }else{
                    S[j]=sum(rgam(n =muestra_cant[j],mu_loss[j],delta_loss ))
                  }
                }

                M[i] <- sum(S)
                porc<-paste((i/10000)*100,"%")
                print(porc)
                #print(M[i])
              }

              hist(M,breaks = 50)
              mean(M)
              sum(data.Incurrido)


# Simulacion de la Frecuencia con  Independencia
      #ADECUAR LOS DATOS DE SIMULACION A UN ESTRUCTURA DE MODELO COLECTIVO DE RIESGO
      # Frecuencia
      N<-seq(1:10000)
      for(i in 1:10000){
        N[i] <- sum(simulate(glm.Frec.PT,nsim=1, type="response" ).sim_1)/sum(data_model.Expuesto)
        porc<-paste((i/10000)*100,"%")
        print(porc)
      }
      hist(N,breaks = 50)
      mean(N)
      mean(data.Frecuencia)

#########################################################################
# Limpiamos el ambiente de variables
  rm(list = ls())
  gc()

#########################################################################
# REGRESION CLAYTON
#########################################################################

  # Set Directory
  setwd("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Perdida_
    Total")

  # Carga los modelos GLM MARGINALES calculados en otos querys
  load("glm.Frec.PT.RData")
  load("glm.Sev.PT.RData")
  load("Data_PT.RData")
  load("Data_PT_model.RData")
  load("Data_PT_model_CM.RData")

  # Creamos las variables de respuestas y la matriz de datos de cada glm marginal
  # SEVERIDAD
  x <- data_CM.Incurrido/data_CM.Cantidad #(Costo Medio)
  var_x <-model.matrix(glm.Sev.PT)  #data_CM[,3:12]
  # FRECUENCIA
  y <- data_model.Cantidad
  Expuestos <- data_model.Expuesto
  var_y <- model.matrix(glm.Frec.PT)   #data[,3:12]

  # Estimacion de parámetros iniciales utilizando las distribuciones marginales
  # SEVERIDAD
  sd.alpha <- sqrt(diag(vcov(glm.Sev.PT)))
  alpha_0  <- glm.Sev.PT.coefficients
  delta_0  <- summary(glm.Sev.PT).dispersion
  mu_0     <- exp(var_x%*%alpha_0)

  # FRECUENCIA
  beta_0   <- glm.Frec.PT.coefficients
  sd.beta  <- sqrt(diag(vcov(glm.Frec.PT)))
  lambda_0 <- exp(var_y%*%beta_0)*Expuestos

  # REGRESION MEDIANTE COPULAS

  # COPULA CLAYTON
  family=3
  # Estimación inicial del parámetro theta para estimarlo mediante el algoritmo de MPLE
```

```
361   theta_ini <- BiCopEst(rank(data_model.Incurrido-exp(var_y%*%alpha_0) )/(length(y)+1),rank(y*Expuestos-
        lambda_0)/(length(y)+1), family=family).par
362   #theta_ini <- BiCopEst(rank(data_CM.Incurrido-exp(var_x%*%alpha_0))/(length(x)+1),rank(data_CM.Cantidad*
        data_CM.Expuesto-exp(var_x%*%beta_0)*data_CM.Expuesto)/(length(x)+1), family=family).par
363   tau_ini    <- BiCopPar2Tau(par=theta_ini,family = family)
364
365   # Estimación de las seudo-observaciones
366   u<-pgam(x,mu_0,delta_0)
367   v    <- ppois(data_CM.Cantidad,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)
368   vv   <- ppois(data_CM.Cantidad-1,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)
369
370   # Creamos la función primera derivada de la Función Copula
371   Du_Clay <- function(u,v,theta){
372     u[u<=0]=0.0001
373     u[u>1]=1
374     v[v<=0]=0.0001
375     v[v>1]=1
376     #out <- pnorm((qnorm(v)-theta*qnorm(u))/(sqrt(1-theta^2)))
377     out <- (u^(-theta)+v^(-theta)-1)^((-1/theta)-1)*(u^(-theta-1))
378     return(out)
379   }
380
381   # Función a maximizar
382   f_aux <- function(para){
383     D_u_0<-Du_Clay(u,v,para)-Du_Clay(u,vv,para)
384     D_u_0[D_u_0<=0]=1
385     D_u_0 <- log(D_u_0)
386     out<-(sum(D_u_0))
387     return(out)
388   }
389
390   # Proceso de maximización
391   #para_ini <- theta2z(theta_ini,family = family)
392   para.ifm <- optim(theta_ini,f_aux,method = "BFGS").par
393   #theta_0  <- z2theta(para.ifm,family = family)
394   theta_0  <- para.ifm # Este será nuestro parámetro inicial estimado con el método IFM y que luego usaremos
        para la regresión conjunta
395   tau       <-BiCopPar2Tau(par=theta_0,family=family) # tau de Khendal para estimar el grado de dependencia
396
397   # Estimamos los coeficientes de regresión y el parámetro cópula mediante máxima verosimilitud
398
399   # Creamos nuestra función a maximizar
400   ######################################################################
401   f_aux_reg <- function(para){
402
403     p        <- ncol(var_x)
404     q        <- ncol(var_x)
405     alpha    <- para[1:p]
406     beta     <- para[(p+1):(p+q)]
407     # theta    <- z2theta(para[p+q+1],family)
408     theta    <- z2theta(para[p+q+1],family = family)
409     delta    <- para[p+q+2]
410     lambda   <- as.vector(exp(var_y%*%beta)*data_model.Expuesto)
411     mu       <- as.vector(exp(var_y%*%alpha))
412     #dummy    <- density_joint(x,y,mu,delta,lambda,theta,family=family,zt=FALSE)
413     u2 <- pgam(data_model.Incurrido,mu,delta)
414     v2<-ppois(y,lambda)
415     vv2<-ppois(y-1,lambda)
416
417     marginal.x <- dgam(data_model.Incurrido,mu,delta)
418     marginal.x[marginal.x>=1]=0
419     marginal.x[marginal.x<=0]=0
420     marginal.x[is.na(marginal.x)]=0
421     marginal.x[marginal.x==-Inf]=0
422     marginal.x[marginal.x==Inf]=0
423
424     par_der    <- Du_Clay(u2,v2,theta)
425     par_der1   <- Du_Clay(u2,vv2,theta)
426
427     dummy<- par_der-par_der1
428
429     dummy[y==0]=par_der[y==0]
430
431     out<-marginal.x*dummy
432
433     out[out<=0]=1e-10
```

```
    ll       <- -sum(log(out))

    #if(negative==TRUE) ll <- (-ll)
    return(ll)
}

#################################################################
para_0<-c(alpha_0,beta_0,theta2z(theta_0,family=family),delta_0) # Parámetros iniciales
a<-now()
para_optim <- optim(para_0,f_aux_reg,method = "BFGS",hessian = TRUE) # Estimación Maxima verosmil de los parámetros (Toma hasta 3h sin matriz hessiana. 8h con Matriz hessiana)
b<-now()
b-a
#load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_Conjuntos/Perdida_Parcial/JointModel_Gaus.RData")
para_optim.value # Log_Likelihood
vector <- para_optim.par
p        <- ncol(var_y)
q        <- ncol(var_y)
Log_Likelihood<-para_optim.value
# Parámetros optimizados
alpha_clay<-vector[1:p]
beta_clay<-vector[(p+1):(p+q)]
theta_clay=z2theta(vector[p+q+1],family) #-0.09506928
delta_clay<-exp(vector[p+q+2]) #4.271691
tau_clay<-BiCopPar2Tau(par=theta_clay,family=family)#-0.06061453
head(beta_0)
head(beta_clay)
head(alpha_0)
head(alpha_clay)
# Determinamos los errores estandar para el test de Wald
hessian_clay    <- para_optim.hessian
Hinv            <- ginv(hessian_clay)
sd              <- sqrt(diag(Hinv))
sd.alpha_clay   <-sd[1:p]
sd.beta_clay    <- sd[(p+1):(p+q)]
sd.theta_clay   <-sd[p+q+1]

# Ordenamos los coeficientes estimados en una tabla
Reg_Cop_Clayton <- data.frame(Estimate_Frec=round(beta_clay,4),Std.error.frec=round(sd.beta_clay,4),z_value_frec=round(beta_clay/sd.beta_clay,4),"Prob_frec"=round(2*(1-pnorm(abs(beta_clay/sd.beta_clay))),4),Estimate_Sev=round(alpha_clay,4),Std.error.sev=round(sd.alpha_clay,4),z_value_sev=round(alpha_clay/sd.alpha_clay,4),"Prob_Sev"=round(2*(1-pnorm(abs(alpha_clay/sd.alpha_clay))),4))

Reg_Cop_Clayton <- Reg_Cop_Clayton %>% as.tibble() %>%  mutate(var_level=var_level) %>% relocate(var_level)

# Agregamos los interceptos
temp <- tibble(
  var_level = c(
    "Año2022",
    "PT_Tipo_PolizaIndividual",
    "PT_Uso_VehParticular",
    "PT_Marca_VehL4",
    "PT_ZonaA3",
    "PT_GeneroFemenino",
    "PT_Edad43-52"
  ),
  Estimate_Frec = 0
  ,
  Estimate_Sev = 0
)

# coeficientes del modelo de regresion con dependencia
Reg_Cop_Clayton <- Reg_Cop_Clayton %>% bind_rows(temp) %>% arrange(var_level) %>% mutate(`exp(Estimate_Frec)`=round(exp(Estimate_Frec),4),`exp(Estimate_Sev)`=round(exp(Estimate_Sev),4))

Reg_Cop_Clayton <-Reg_Cop_Clayton %>%  relocate(`exp(Estimate_Frec)`, .after=Prob_frec)

# Guardamos el modelo conjunto y los coeficientes
write.csv(Reg_Cop_Clayton, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_Conjuntos/Perdida_Total/Coef_Clay.csv", fileEncoding = "Latin1")
save(para_optim, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_Conjuntos/Perdida_Total/JointModel_Clay.RData")

# Simulacion de la Pérdida Total
```

```
# ============================================================

# CON DEPENDENCIA
# ---------------
lambda_loss_dep   <- exp(var_x%*%beta_clay)*data_CM.Expuesto
mu_loss_dep       <- exp(var_x%*%alpha_clay)
delta_loss_dep    <- delta_clay
# Función de densidad fx
f_loss <-
  function(loss,mu,d,lam,theta){
    n<-length(loss)
    if (length(lam)==1) lam   <- rep(lam,n)
    if (length(mu)==1) mu     <- rep(mu,n)
    out                       <- vector(length=n)

    for(i in 1:n){
      N  <-1:20
      v  <-ppois(N,lam[i])
      vv <-ppois(N-1,lam[i])
      u  <-pgam(loss[i]/N,mu[i],d)

      Der_cop <-Du_Clay(u,v,theta)-Du_Clay(u,vv,theta)
      dummy   <-Der_cop*dgam(loss[i]/N,mu[i],d)/N
      out[i]  <-sum(dummy)
    }

    out[loss<=0]=0
    return(out)
  }

# Función de densidad acumulada Fx
F_loss <-
  function(loss,mu,d,lam,theta){
    out<-vector(length=length(loss))

    for(i in 1:length(loss)){
      floss <- function(s){
        f_loss(s,mu[i],d,lam[i],theta)
      }
      out[i] <- integrate(floss,0,loss[i]).value
    }
    return(out)
  }

# Estimación del Total Loss (fuerte dedicacion computacional)

#k <-length(lambda_loss_dep)
k <- 100#length(lambda_loss_dep) # Cantidad de numeros aleatorios (Debe ser del tamaño de lambda)
m <- 10 # Cantidad de muestras aleatorias (Deben ser 10 mil)
L <- vector(length=k)
S <- vector(length=m)

for(j in 1:m){
  r_uni<-runif(k)
  for(i in 1:k){
    f_root <-function(s){
      F_loss(s,mu_loss_dep[i],delta_loss_dep,lambda_loss_dep[i],theta_clay)-r_uni[i]
    }

    tryCatch(
      error = function(cnd) loss<-mu_loss_dep[i]*lambda_loss_dep[i],
      loss<-uniroot(f_root,lower = 0,upper = 500000).root
    )

    #print(loss.root)
    L[i]<-loss
    perc<-paste(i/k*100,"%")
    print(perc)
  }
  S[j]<-sum(L)
  perc<-paste(j/m*100,"%")
  print(perc)
  #print(S[j])
}
S
```

```
    # CON INDEPENDENCIA
    # -----------------
    lambda_loss  <-predict(glm.Frec.PP, newdata=data_CM, type='response') # CAMBIAR LOS DATOS A UN MODELO DE
      RIESGO COLECTIVO
    mu_loss      <-predict(glm.Sev.PP, newdata=data_CM, type='response')
    delta_loss   <-delta_0

    M<-seq(1:10000)
    for(i in 1:10000){
      muestra_cant<-rpois(length(lambda_loss),lambda_loss)
      S<-vector(length=length(lambda_loss))
      for(j in 1:length(muestra_cant)){
        if(muestra_cant[j]==0){
          S[j]=0
        }else{
          S[j]=sum(rgam(n =muestra_cant[j],mu_loss[j],delta_loss ))
        }
      }

      M[i] <- sum(S)
      porc<-paste((i/10000)*100,"%")
      print(porc)
      #print(M[i])
    }

    hist(M)

    # Simulacion de la Frecuencia con  Independencia
    #ADECUAR LOS DATOS DE SIMULACION A UN ESTRUCTURA DE MODELO COLECTIVO DE RIESGO
    # Frecuencia
    N<-seq(1:10000)
    for(i in 1:10000){
      N[i] <- sum(simulate(glm.Frec.PP,nsim=1, type="response" ).sim_1)/sum(data_model.Expuesto)
      porc<-paste((i/10000)*100,"%")
      print(porc)
    }
    hist(N)

    ######################################################################
    # Limpiamos el ambiente de variables
    rm(list = ls())
    gc()

######################################################################
# REGRESION GUMBEL
######################################################################

    # Set Directory
    setwd("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Perdida_
      Total")

    # Carga los modelos GLM MARGINALES calculados en otos querys
    load("glm.Frec.PT.RData")
    load("glm.Sev.PT.RData")
    load("Data_PT.RData")
    load("Data_PT_model.RData")
    load("Data_PT_model_CM.RData")

    # Creamos las variables de respuestas y la matriz de datos de cada glm marginal
    # SEVERIDAD
    x <- data_CM.Incurrido/data_CM.Cantidad #(Costo Medio)
    var_x <-model.matrix(glm.Sev.PT)  #data_CM[,3:12]
    # FRECUENCIA
    y <- data_model.Cantidad
    Expuestos <- data_model.Expuesto
    var_y <- model.matrix(glm.Frec.PT)   #data[,3:12]

    # Estimacion de parámetros iniciales utilizando las distribuciones marginales
    # SEVERIDAD
    sd.alpha <- sqrt(diag(vcov(glm.Sev.PT)))
    alpha_0  <- glm.Sev.PT.coefficients
    delta_0  <- summary(glm.Sev.PT).dispersion
    mu_0     <- exp(var_x%*%alpha_0)

    # FRECUENCIA
    beta_0   <- glm.Frec.PT.coefficients
```

```
651  sd.beta  <- sqrt(diag(vcov(glm.Frec.PT)))
652  lambda_0 <- exp(var_y%*%beta_0)*Expuestos
653
654  # REGRESION MEDIANTE COPULAS
655
656  # COPULA GUMBEL
657  family=4
658  # Estimación inicial del parámetro theta para estimarlo mediante el algoritmo de MPLE
659  theta_ini <- BiCopEst(rank(data_model.Incurrido-exp(var_y%*%alpha_0) )/(length(y)+1),rank(y*Expuestos-
       lambda_0)/(length(y)+1), family=family).par
660  #theta_ini <- BiCopEst(rank(data_CM.Incurrido-exp(var_x%*%alpha_0))/(length(x)+1),rank(data_CM.Cantidad*
       data_CM.Expuesto-exp(var_x%*%beta_0)*data_CM.Expuesto)/(length(x)+1), family=family).par
661  tau_ini    <- BiCopPar2Tau(par=theta_ini,family = family)
662
663  # Estimación de las seudo-observaciones
664  u<-pgam(x,mu_0,delta_0)
665  v    <- ppois(data_CM.Cantidad,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)
666  vv   <- ppois(data_CM.Cantidad-1,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)
667
668  # Creamos la función primera derivada de la Función Copula
669  Du_Gum <- function(u,v,theta){
670    u[u<=0]=0.0001
671    u[u>1]=1
672    v[v<=0]=0.0001
673    v[v>1]=1
674    #out <- pnorm((qnorm(v)-theta*qnorm(u))/(sqrt(1-theta^2)))
675    #out <- (u^(-theta)+v^(-theta)-1)^((-1/theta)-1)*(u^(-theta-1))
676    out <- (u^-1)*exp(-((-log(u))^theta+(-log(v))^theta)^(1/theta))
677    return(out)
678  }
679
680  # Función a maximizar
681  f_aux <- function(para){
682    D_u_0<-Du_Gum(u,v,para)-Du_Gum(u,vv,para)
683    D_u_0[D_u_0<=0]=1
684    D_u_0 <- log(D_u_0)
685    out<-(sum(D_u_0))
686    return(out)
687  }
688
689  # Proceso de maximización
690  #para_ini <- theta2z(theta_ini,family = family)
691  para.ifm <- optim(theta_ini,f_aux,method = "BFGS").par
692  #theta_0  <- z2theta(para.ifm,family = family)
693  theta_0  <- para.ifm # Este será nuestro parámetro inicial estimado con el método IFM y que luego usaremos
       para la regresión conjunta
694  tau      <-BiCopPar2Tau(par=theta_0,family=family) # tau de Khendal para estimar el grado de dependencia
695
696  # Estimamos los coeficientes de regresión y el parámetro cópula mediante máxima verosimilitud
697
698  # Creamos nuestra función a maximizar
699  ######################################################################
700  f_aux_reg <- function(para){
701
702    p        <- ncol(var_x)
703    q        <- ncol(var_x)
704    alpha    <- para[1:p]
705    beta     <- para[(p+1):(p+q)]
706    # theta    <- z2theta(para[p+q+1],family)
707    theta    <- z2theta(para[p+q+1],family = family)
708    delta    <- para[p+q+2]
709    lambda   <- as.vector(exp(var_y%*%beta)*data_model.Expuesto)
710    mu       <- as.vector(exp(var_y%*%alpha))
711    #dummy    <- density_joint(x,y,mu,delta,lambda,theta,family=family,zt=FALSE)
712    u2 <- pgam(data_model.Incurrido,mu,delta)
713    v2<-ppois(y,lambda)
714    vv2<-ppois(y-1,lambda)
715
716    marginal.x <- dgam(data_model.Incurrido,mu,delta)
717    marginal.x[marginal.x>=1]=0
718    marginal.x[marginal.x<=0]=0
719    marginal.x[is.na(marginal.x)]=0
720    marginal.x[marginal.x==-Inf]=0
721    marginal.x[marginal.x==Inf]=0
722
723    par_der    <- Du_Gum(u2,v2,theta)
```

```
    par_der1  <- Du_Gum(u2,vv2,theta)

    dummy<- par_der-par_der1

    dummy[y==0]=par_der[y==0]

    out<-marginal.x*dummy

    out[out<=0]=1e-10
    ll       <- -sum(log(out))

    #if(negative==TRUE) ll <- (-ll)
    return(ll)
}

#################################################################
para_0<-c(alpha_0,beta_0,theta2z(theta_0,family=family),delta_0) # Parámetros iniciales
a<-now()
para_optim <- optim(para_0,f_aux_reg,method = "BFGS",hessian = TRUE) # Estimación Maxima verosmil de los
  parámetros (Toma hasta 3h sin matriz hessiana. 8h con Matriz hessiana)
b<-now()
b-a #3.485547 hours
#load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_Conjuntos/Perdida_
  Parcial/JointModel_Gaus.RData")
para_optim.value # Log_Likelihood
vector <- para_optim.par
p        <- ncol(var_y)
q        <- ncol(var_y)
Log_Likelihood<-para_optim.value
# Parámetros optimizados
alpha_gum<-vector[1:p]
beta_gum<-vector[(p+1):(p+q)]
theta_gum=z2theta(vector[p+q+1],family) #-0.09506928
delta_gum<-exp(vector[p+q+2]) #4.271691
tau_gum<-BiCopPar2Tau(par=theta_gum,family=family)#-0.06061453
head(beta_0)
head(beta_gum)
head(alpha_0)
head(alpha_gum)

# Determinamos los errores estandar para el test de Wald
hessian_gum   <- para_optim.hessian
Hinv          <- ginv(hessian_gum)
sd            <- sqrt(diag(Hinv))
sd.alpha_gum  <-sd[1:p]
sd.beta_gum   <- sd[(p+1):(p+q)]
sd.theta_gum  <-sd[p+q+1]

# Ordenamos los coeficientes estimados en una tabla
Reg_Cop_Gumbel <- data.frame(Estimate_Frec=round(beta_gum,4),Std.error.frec=round(sd.beta_gum,4),z_value_
  frec=round(beta_gum/sd.beta_gum,4),"Prob_frec"=round(2*(1-pnorm(abs(beta_gum/sd.beta_gum))),4),Estimate_
  Sev=round(alpha_gum,4),Std.error.sev=round(sd.alpha_gum,4),z_value_sev=round(alpha_gum/sd.alpha_gum,4),"
  Prob_Sev"=round(2*(1-pnorm(abs(alpha_gum/sd.alpha_gum))),4))

Reg_Cop_Gumbel <- Reg_Cop_Gumbel %>% as.tibble() %>%  mutate(var_level=var_level) %>% relocate(var_level)

# Agregamos los interceptos

temp <- tibble(
  var_level = c(
    "Año2022",
    "PT_Tipo_PolizaIndividual",
    "PT_Uso_VehParticular",
    "PT_Marca_VehL4",
    "PT_ZonaA3",
    "PT_GeneroFemenino",
    "PT_Edad43-52"
  ),
  Estimate_Frec = 0
  ,
  Estimate_Sev = 0
)

# coeficientes del modelo de regresion con dependencia
Reg_Cop_Gumbel <- Reg_Cop_Gumbel %>% bind_rows(temp) %>% arrange(var_level) %>% mutate(`exp(Estimate_Frec)
  `=round(exp(Estimate_Frec),4),`exp(Estimate_Sev)`=round(exp(Estimate_Sev),4))
```

```
794
795     Reg_Cop_Gumbel <-Reg_Cop_Gumbel %>%  relocate(`exp(Estimate_Frec)`, .after=Prob_frec)
796
797     # Guardamos el modelo conjunto y los coeficientes
798     write.csv(Reg_Cop_Gumbel, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_
          GLM_Conjuntos/Perdida_Total/Coef_Gumb.csv", fileEncoding = "Latin1")
799     save(para_optim, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_
          Conjuntos/Perdida_Total/JointModel_Gumb.RData")
800
801     # Simulacion de la Pérdida Total
802     # ============================================================
803
804     # CON DEPENDENCIA
805     # ---------------
806     lambda_loss_dep  <- exp(var_x%*%beta_clay)*data_CM.Expuesto
807     mu_loss_dep      <- exp(var_x%*%alpha_clay)
808     delta_loss_dep   <- delta_clay
809     # Función de densidad fx
810
811     f_loss <-
812       function(loss,mu,d,lam,theta){
813         n<-length(loss)
814         if (length(lam)==1) lam  <- rep(lam,n)
815         if (length(mu)==1) mu    <- rep(mu,n)
816         out                      <- vector(length=n)
817
818         for(i in 1:n){
819           N  <-1:20
820           v  <-ppois(N,lam[i])
821           vv <-ppois(N-1,lam[i])
822           u  <-pgam(loss[i]/N,mu[i],d)
823
824           Der_cop <-Du_Clay(u,v,theta)-Du_Clay(u,vv,theta)
825           dummy   <-Der_cop*dgam(loss[i]/N,mu[i],d)/N
826           out[i]  <-sum(dummy)
827         }
828
829         out[loss<=0]=0
830         return(out)
831       }
832
833     # Función de densidad acumulada Fx
834     F_loss <-
835       function(loss,mu,d,lam,theta){
836         out<-vector(length=length(loss))
837
838         for(i in 1:length(loss)){
839           floss <- function(s){
840             f_loss(s,mu[i],d,lam[i],theta)
841           }
842           out[i] <- integrate(floss,0,loss[i]).value
843         }
844         return(out)
845       }
846
847     # Estimación del Total Loss (fuerte dedicacion computacional)
848
849     #k <-length(lambda_loss_dep)
850     k <- 100#length(lambda_loss_dep) # Cantidad de numeros aleatorios (Debe ser del tamaño de lambda)
851     m <- 10 # Cantidad de muestras aleatorias (Deben ser 10 mil)
852     L <- vector(length=k)
853     S <- vector(length=m)
854
855     for(j in 1:m){
856       r_uni<-runif(k)
857       for(i in 1:k){
858         f_root <-function(s){
859           F_loss(s,mu_loss_dep[i],delta_loss_dep,lambda_loss_dep[i],theta_clay)-r_uni[i]
860         }
861
862         tryCatch(
863           error = function(cnd) loss<-mu_loss_dep[i]*lambda_loss_dep[i],
864           loss<-uniroot(f_root,lower = 0,upper = 500000).root
865         )
866
867         #print(loss.root)
```

```
        L[i]<-loss
        perc<-paste(i/k*100,"%")
        print(perc)
      }
      S[j]<-sum(L)
      perc<-paste(j/m*100,"%")
      print(perc)
      #print(S[j])
    }
    S

    # CON INDEPENDENCIA
    # -----------------
    lambda_loss  <-predict(glm.Frec.PP, newdata=data_CM, type='response') # CAMBIAR LOS DATOS A UN MODELO DE
      RIESGO COLECTIVO
    mu_loss      <-predict(glm.Sev.PP, newdata=data_CM, type='response')
    delta_loss   <-delta_0

    M<-seq(1:10000)
    for(i in 1:10000){
      muestra_cant<-rpois(length(lambda_loss),lambda_loss)
      S<-vector(length=length(lambda_loss))
      for(j in 1:length(muestra_cant)){
        if(muestra_cant[j]==0){
          S[j]=0
        }else{
          S[j]=sum(rgam(n =muestra_cant[j],mu_loss[j],delta_loss ))
        }
      }

      M[i] <- sum(S)
      porc<-paste((i/10000)*100,"%")
      print(porc)
      #print(M[i])
    }

    hist(M)

    # Simulacion de la Frecuencia con  Independencia
    #ADECUAR LOS DATOS DE SIMULACION A UN ESTRUCTURA DE MODELO COLECTIVO DE RIESGO
    # Frecuencia
    N<-seq(1:10000)
    for(i in 1:10000){
      N[i] <- sum(simulate(glm.Frec.PP,nsim=1, type="response" ).sim_1)/sum(data_model.Expuesto)
      porc<-paste((i/10000)*100,"%")
      print(porc)
    }
    hist(N)


    ######################################################################
    # Limpiamos el ambiente de variables
    rm(list = ls())
    gc()

######################################################################
# REGRESION FRANK
######################################################################

    # Set Directory
    setwd("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Perdida_
      Total")

    # Carga los modelos GLM MARGINALES calculados en otos querys
    load("glm.Frec.PT.RData")
    load("glm.Sev.PT.RData")
    load("Data_PT.RData")
    load("Data_PT_model.RData")
    load("Data_PT_model_CM.RData")

    # Creamos las variables de respuestas y la matriz de datos de cada glm marginal
    # SEVERIDAD
    x <- data_CM.Incurrido/data_CM.Cantidad #(Costo Medio)
    var_x <-model.matrix(glm.Sev.PT)  #data_CM[,3:12]
    # FRECUENCIA
    y <- data_model.Cantidad
```

```
942     Expuestos <- data_model.Expuesto
943     var_y <- model.matrix(glm.Frec.PT)    #data[,3:12]
944
945     # Estimacion de parámetros iniciales utilizando las distribuciones marginales
946     # SEVERIDAD
947     sd.alpha <- sqrt(diag(vcov(glm.Sev.PT)))
948     alpha_0  <- glm.Sev.PT.coefficients
949     delta_0  <- summary(glm.Sev.PT).dispersion
950     mu_0     <- exp(var_x%*%alpha_0)
951
952     # FRECUENCIA
953     beta_0   <- glm.Frec.PT.coefficients
954     sd.beta  <- sqrt(diag(vcov(glm.Frec.PT)))
955     lambda_0 <- exp(var_y%*%beta_0)*Expuestos
956
957     # REGRESION MEDIANTE COPULAS
958
959     # COPULA GUMBEL
960     family=5
961     # Estimación inicial del parámetro theta para estimarlo mediante el algoritmo de MPLE
962     theta_ini <- BiCopEst(rank(data_model.Incurrido-exp(var_y%*%alpha_0) )/(length(y)+1),rank(y*Expuestos-
          lambda_0)/(length(y)+1), family=family).par
963     #theta_ini <- BiCopEst(rank(data_CM.Incurrido-exp(var_x%*%alpha_0))/(length(x)+1),rank(data_CM.Cantidad*
          data_CM.Expuesto-exp(var_x%*%beta_0)*data_CM.Expuesto)/(length(x)+1), family=family).par
964     tau_ini    <- BiCopPar2Tau(par=theta_ini,family = family)
965
966     # Estimación de las seudo-observaciones
967     u<-pgam(x,mu_0,delta_0)
968     v    <- ppois(data_CM.Cantidad,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)
969     vv   <- ppois(data_CM.Cantidad-1,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)
970
971     # Creamos la función primera derivada de la Función Copula
972     Du_Frank <- function(u,v,theta){
973       u[u<=0]=0.0001
974       u[u>1]=1
975       v[v<=0]=0.0001
976       v[v>1]=1
977       #out <- pnorm((qnorm(v)-theta*qnorm(u))/(sqrt(1-theta^2)))
978       #out <- (u^(-theta)+v^(-theta)-1)^((-1/theta)-1)*(u^(-theta-1))
979       #out <- (u^-1)*exp(-((-log(u))^theta+(-log(v))^theta)^(1/theta))
980       out <- (exp(theta)*(exp(theta*v)-1))/(exp(theta*(u+1))+exp(theta*(v+1))-exp(theta)-exp(theta*(u+v)))
981       return(out)
982     }
983
984     # Función a maximizar
985     f_aux <- function(para){
986       D_u_0<-Du_Frank(u,v,para)-Du_Frank(u,vv,para)
987       D_u_0[D_u_0<=0]=1
988       D_u_0 <- log(D_u_0)
989       out<-(-sum(D_u_0))
990       return(out)
991     }
992
993     # Proceso de maximización
994     #para_ini <- theta2z(theta_ini,family = family)
995     para.ifm <- optim(theta_ini,f_aux,method = "BFGS").par
996     #theta_0  <- z2theta(para.ifm,family = family)
997     theta_0  <- para.ifm # Este será nuestro parámetro inicial estimado con el método IFM y que luego usaremos
          para la regresión conjunta
998     tau      <-BiCopPar2Tau(par=theta_0,family=family) # tau de Khendal para estimar el grado de dependencia
999
1000    # Estimamos los coeficientes de regresión y el parámetro cópula mediante máxima verosimilitud
1001
1002    # Creamos nuestra función a maximizar
1003    ######################################################################
1004    f_aux_reg <- function(para){
1005
1006      p        <- ncol(var_x)
1007      q        <- ncol(var_x)
1008      alpha    <- para[1:p]
1009      beta     <- para[(p+1):(p+q)]
1010      # theta    <- z2theta(para[p+q+1],family)
1011      theta    <- z2theta(para[p+q+1],family = family)
1012      delta    <- para[p+q+2]
1013      lambda   <- as.vector(exp(var_y%*%beta)*data_model.Expuesto)
1014      mu       <- as.vector(exp(var_y%*%alpha))
```

```
    #dummy    <- density_joint(x,y,mu, delta ,lambda , theta , family=family , zt=FALSE)
    u2 <- pgam(data_model.Incurrido ,mu, delta)
    v2<-ppois(y,lambda)
    vv2<-ppois(y-1,lambda)

    marginal.x <- dgam(data_model.Incurrido ,mu, delta)
    marginal.x[marginal.x>=1]=0
    marginal.x[marginal.x<=0]=0
    marginal.x[is.na(marginal.x)]=0
    marginal.x[marginal.x==-Inf]=0
    marginal.x[marginal.x==Inf]=0

    par_der    <- Du_Frank(u2,v2,theta)
    par_der1   <- Du_Frank(u2,vv2,theta)

    dummy<- par_der-par_der1

    dummy[y==0]=par_der[y==0]

    out<-marginal.x*dummy

    out[out<=0]=1e-10
    ll       <- -sum(log(out))

    #if(negative==TRUE) ll <- (-ll)
    return(ll)
  }

  ################################################################
  para_0<-c(alpha_0,beta_0,theta2z(theta_0,family=family),delta_0) # Parámetros iniciales
  a<-now()
  para_optim <- optim(para_0,f_aux_reg,method = "BFGS",hessian = TRUE) # Estimación Maxima verosmil de los
    parámetros (Toma hasta 3h sin matriz hessiana. 8h con Matriz hessiana)
  b<-now()
  b-a #2.82135 hours
  #load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_Conjuntos/Perdida_
    Parcial/JointModel_Gaus.RData")
  para_optim.value # Log_Likelihood
  vector <- para_optim.par
  p        <- ncol(var_y)
  q        <- ncol(var_y)
  Log_Likelihood<-para_optim.value
  # Parámetros optimizados
  alpha_frank<-vector[1:p]
  beta_frank<-vector[(p+1):(p+q)]
  theta_frank=z2theta(vector[p+q+1],family)
  delta_frank<-exp(vector[p+q+2]) #4.271691
  tau_frank<-BiCopPar2Tau(par=theta_frank,family=family)
  head(beta_0)
  head(beta_frank)
  head(alpha_0)
  head(alpha_frank)

  # Determinamos los errores estandar para el test de Wald
  hessian_frank    <- para_optim.hessian
  Hinv             <- ginv(hessian_frank)
  sd               <- sqrt(diag(Hinv))
  sd.alpha_frank   <-sd[1:p]
  sd.beta_frank    <- sd[(p+1):(p+q)]
  sd.theta_frank   <-sd[p+q+1]

  # Ordenamos los coeficientes estimados en una tabla
  Reg_Cop_Frank <- data.frame(Estimate_Frec=round(beta_frank,4),Std.error.frec=round(sd.beta_frank,4),z_
    value_frec=round(beta_frank/sd.beta_frank,4),"Prob_frec"=round(2*(1-pnorm(abs(beta_frank/sd.beta_frank)))
    ,4),Estimate_Sev=round(alpha_frank,4),Std.error.sev=round(sd.alpha_frank,4),z_value_sev=round(alpha_frank
    /sd.alpha_frank,4),"Prob_Sev"=round(2*(1-pnorm(abs(alpha_frank/sd.alpha_frank))),4))

  var_level<-row.names(Reg_Cop_Frank)
  Reg_Cop_Frank <- Reg_Cop_Frank %>% as.tibble() %>%  mutate(var_level=var_level) %>% relocate(var_level)

  # Agregamos los interceptos
  temp <- tibble(
    var_level = c(
      "Año2022",
      "PT_Tipo_PolizaIndividual",
      "PT_Uso_VehParticular",
```

```
        "PT_Marca_VehL4",
        "PT_ZonaA3",
        "PT_GeneroFemenino",
        "PT_Edad43-52"
      ),
      Estimate_Frec = 0
      ,
      Estimate_Sev = 0
    )

    # coeficientes del modelo de regresion con dependencia
    Reg_Cop_Frank <- Reg_Cop_Frank %>% bind_rows(temp) %>% arrange(var_level) %>% mutate(`exp(Estimate_Frec)`=
      round(exp(Estimate_Frec),4),`exp(Estimate_Sev)`=round(exp(Estimate_Sev),4))

    Reg_Cop_Frank <-Reg_Cop_Frank %>%  relocate(`exp(Estimate_Frec)`, .after=Prob_frec)

    # Guardamos el modelo conjunto y los coeficientes
    write.csv(Reg_Cop_Frank, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_
      GLM_Conjuntos/Perdida_Total/Coef_Frank.csv", fileEncoding = "Latin1")
    save(para_optim, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_
      Conjuntos/Perdida_Total/JointModel_Frank.RData")

    # Simulacion de la Pérdida Total
    # ============================================================

    # CON DEPENDENCIA
    # ---------------
    lambda_loss_dep  <- exp(var_x%*%beta_frank)*data_CM.Expuesto
    mu_loss_dep      <- exp(var_x%*%alpha_frank)
    delta_loss_dep   <- delta_frank
    # Función de densidad fx
    f_loss <-
      function(loss,mu,d,lam,theta){
        n<-length(loss)
        if (length(lam)==1) lam  <- rep(lam,n)
        if (length(mu)==1) mu    <- rep(mu,n)
        out                      <- vector(length=n)

        for(i in 1:n){
          N  <-1:20
          v  <-ppois(N,lam[i])
          vv <-ppois(N-1,lam[i])
          u  <-pgam(loss[i]/N,mu[i],d)

          Der_cop <-Du_Frank(u,v,theta)-Du_Frank(u,vv,theta) # Primera derivada de la función copula
          dummy   <-Der_cop*dgam(loss[i]/N,mu[i],d)/N # Teorela del Total Loss
          out[i]  <-sum(dummy)
        }

        out[loss<=0]=0
        return(out)
      }

    # Función de densidad acumulada Fx
    F_loss <-
      function(loss,mu,d,lam,theta){
        out<-vector(length=length(loss))

        for(i in 1:length(loss)){
          floss <- function(s){
            f_loss(s,mu[i],d,lam[i],theta)
          }
          out[i] <- integrate(floss,0,loss[i]).value # Integral para el cálculo del a F(x) a partir de f(x)
        }
        return(out)
      }

    # Estimación del Total Loss (fuerte dedicacion computacional)

    #k <-length(lambda_loss_dep)
    k <- 100#length(lambda_loss_dep) # Cantidad de numeros aleatorios (Debe ser del tamaño de lambda)
    m <- 10 # Cantidad de muestras aleatorias (Deben ser 10 mil)
    L <- vector(length=k)
    S <- vector(length=m)

    for(j in 1:m){
```

```
      r_uni<-runif(k)
      for(i in 1:k){
        f_root <-function(s){
          F_loss(s,mu_loss_dep[i],delta_loss_dep,lambda_loss_dep[i],theta_frank)-r_uni[i]
        }

        tryCatch(
          error = function(cnd) loss<-mu_loss_dep[i]*lambda_loss_dep[i],
          loss<-uniroot(f_root,lower = 0,upper = 500000).root
        )

        #print(loss.root)
        L[i]<-loss
        perc<-paste(i/k*100,"%")
        print(perc)
      }
      S[j]<-sum(L)
      perc<-paste(j/m*100,"%")
      print(perc)
      #print(S[j])
    }
    S

    # CON INDEPENDENCIA
    # -----------------
    lambda_loss  <-predict(glm.Frec.PP, newdata=data_CM, type='response') # CAMBIAR LOS DATOS A UN MODELO DE
      RIESGO COLECTIVO
    mu_loss      <-predict(glm.Sev.PP, newdata=data_CM, type='response')
    delta_loss   <-delta_0

    M<-seq(1:10000)
    for(i in 1:10000){
      muestra_cant<-rpois(length(lambda_loss),lambda_loss)
      S<-vector(length=length(lambda_loss))
      for(j in 1:length(muestra_cant)){
        if(muestra_cant[j]==0){
          S[j]=0
        }else{
          S[j]=sum(rgam(n =muestra_cant[j],mu_loss[j],delta_loss ))
        }
      }

      M[i] <- sum(S)
      porc<-paste((i/10000)*100,"%")
      print(porc)
      #print(M[i])
    }

    hist(M)
    # Simulacion de la Frecuencia con  Independencia
    #ADECUAR LOS DATOS DE SIMULACION A UN ESTRUCTURA DE MODELO COLECTIVO DE RIESGO
    # Frecuencia
    N<-seq(1:10000)
    for(i in 1:10000){
      N[i] <- sum(simulate(glm.Frec.PP,nsim=1, type="response" ).sim_1)/sum(data_model.Expuesto)
      porc<-paste((i/10000)*100,"%")
      print(porc)
    }
    hist(N)
```

C.5. Modelo Conjunto - Responsabilidad Civil

```
################################################################################
################################################################################
####                                                                        ######
####                  REGRESION BASADO EN COPULAS                           ######
####                      RESPONSABILIDAD CIVIL                             ######
####                                                                        ######
################################################################################
################################################################################

# Cargamos las librerías necesarias

  #library(copula)
  library(RODBC)
```

```
  library(tidyverse)
  library(MASS)
  #library(GJRM)
  #library(devtools)
  library(CopulaRegression)
  library(VineCopula)
  #library(optimx)
  #install_url('http://cran.r-project.org/src/contrib/Archive/CopulaRegression/CopulaRegression_0.1-5.tar.gz')

######################################################################
# REGRESION GAUSSIANA
######################################################################

# Set Directory
  setwd("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/
      Responsabilidad_Civil")

# Carga los modelos GLM MARGINALES calculados en otos querys
  load("glm.Frec.RC.RData")
  load("glm.Sev.RC.RData")
  load("Data_RC.RData")
  load("Data_RC_model.RData")
  load("Data_RC_model_CM.RData")

# Creamos las variables de respuestas y la matriz de datos de cada glm marginal
  # SEVERIDAD
  x <- data_CM.Incurrido/data_CM.Cantidad #(Costo Medio)
  var_x <-model.matrix(glm.Sev.RC)  #data_CM[,3:12]
  # FRECUENCIA
  y <- data_model.Cantidad
  Expuestos <- data_model.Expuesto
  var_y <- model.matrix(glm.Frec.RC)   #data[,3:12]

# Estimacion de parámetros iniciales utilizando las distribuciones marginales
  # SEVERIDAD
  sd.alpha <- sqrt(diag(vcov(glm.Sev.RC)))
  alpha_0  <- glm.Sev.RC.coefficients
  delta_0  <- summary(glm.Sev.RC).dispersion
  mu_0     <- exp(var_x%*%alpha_0)

  # FRECUENCIA
  beta_0   <- glm.Frec.RC.coefficients
  sd.beta  <- sqrt(diag(vcov(glm.Frec.RC)))
  lambda_0 <- exp(var_y%*%beta_0)*Expuestos

# REGRESION MEDIANTE COPULAS

  # COPULA GAUSSIANA
  family=1
  # Estimación inicial del parámetro theta para estimarlo mediante el algoritmo de MPLE
  theta_ini <- BiCopEst(rank(data_model.Incurrido-exp(var_y%*%alpha_0) )/(length(y)+1),rank(y*Expuestos-lambda
      _0)/(length(y)+1), family=family).par
  #theta_ini <- BiCopEst(rank(data_CM.Incurrido-exp(var_x%*%alpha_0))/(length(x)+1),rank(data_CM.Cantidad*data
      _CM.Expuesto-exp(var_x%*%beta_0)*data_CM.Expuesto)/(length(x)+1), family=family).par
  tau_ini   <- BiCopPar2Tau(par=theta_ini,family = family)

  # Estimación de las seudo-observaciones
  u<-pgam(x,mu_0,delta_0)
  v   <- ppois(data_CM.Cantidad,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)
  vv  <- ppois(data_CM.Cantidad-1,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)

  # Creamos la función primera derivada de la Función Copula
    Du_Gaus <- function(u,v,theta){
      u[u<=0]=0.001
      u[u>=1]=0.999
      v[v<=0]=0.001
      v[v>=1]=0.999
      out <- pnorm((qnorm(v)-theta*qnorm(u))/(sqrt(1-theta^2)))
    return(out)
    }

  # Función a maximizar
  f_aux <- function(para){
    D_u_0<-Du_Gaus(u,v,para)-Du_Gaus(u,vv,para)
    D_u_0[D_u_0<=0]=1
    D_u_0 <- log(D_u_0)
```

```
87      out<-(-sum(D_u_0))
88      return(out)
89    }
90
91    # Proceso de maximización
92    #para_ini <- theta2z(theta_ini,family = family)
93    para.ifm <- optim(theta_ini,f_aux,method = "BFGS").par
94    #theta_0  <- z2theta(para.ifm,family = family)
95    theta_0  <- para.ifm # Este será nuestro parámetro inicial estimado con el método IFM y que luego usaremos
          para la regresión conjunta
96    tau      <-BiCopPar2Tau(par=theta_0,family=family) # tau de Khendal para estimar el grado de dependencia
97
98    # Estimamos los coeficientes de regresión y el parámetro cópula mediante máxima verosimilitud
99
100   # Creamos nuestra función a maximizar
101   ######################################################################
102     f_aux_reg <- function(para){
103
104         p        <- ncol(var_x)
105         q        <- ncol(var_x)
106         alpha    <- para[1:p]
107         beta     <- para[(p+1):(p+q)]
108         # theta    <- z2theta(para[p+q+1],family)
109         theta    <- para[p+q+1]
110         delta    <- para[p+q+2]
111         lambda   <- as.vector(exp(var_y%*%beta)*data_model.Expuesto)
112         mu       <- as.vector(exp(var_y%*%alpha))
113         #dummy    <- density_joint(x,y,mu,delta,lambda,theta,family=family,zt=FALSE)
114         u2 <- pgam(data_model.Incurrido,mu,delta)
115         v2<-ppois(y,lambda)
116         vv2<-ppois(y-1,lambda)
117
118         marginal.x <- dgam(data_model.Incurrido,mu,delta)
119         marginal.x[marginal.x>=1]=0
120         marginal.x[marginal.x<=0]=0
121         marginal.x[is.na(marginal.x)]=0
122         marginal.x[marginal.x==-Inf]=0
123         marginal.x[marginal.x==Inf]=0
124
125         par_der    <- Du_Gaus(u2,v2,theta)
126         par_der1   <- Du_Gaus(u2,vv2,theta)
127
128         dummy<- par_der-par_der1
129
130         dummy[y==0]=par_der[y==0]
131
132         out<-marginal.x*dummy
133
134         out[out<=0]=1e-10
135         ll       <- -sum(log(out))
136
137         #if(negative==TRUE) ll <- (-ll)
138         return(ll)
139   }
140
141   ######################################################################
142
143   para_0<-c(alpha_0,beta_0,theta_0,delta_0) # Parámetros iniciales
144   a<-now()
145   para_optim <- optim(para_0,f_aux_reg,method = "BFGS",hessian = TRUE) # Demora 6min
146   b<-now()
147   b-a
148   #load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_Conjuntos/Perdida_
          Parcial/JointModel_Gaus.RData")
149   para_optim.value # Log_Likelihood
150   vector <- para_optim.par
151   p        <- ncol(var_y)
152   q        <- ncol(var_y)
153
154   # Parámetros optimizados
155       alpha_gau<-vector[1:p]
156       beta_gau<-vector[(p+1):(p+q)]
157       theta_gau=z2theta(vector[p+q+1],family)
158       delta_gau<-exp(vector[p+q+2])
159       tau_gau<-BiCopPar2Tau(par=theta_gau,family=family)
160       head(beta_0)
```

```
    head(beta_gau)
    head(alpha_0)
    head(alpha_gau)
# Determinamos los errores estandar para el test de Wald
    hessian_gau   <- para_optim.hessian
    Hinv          <- ginv(hessian_gau)
    sd            <- sqrt(diag(Hinv))
    sd.alpha_gau  <-sd[1:p]
    sd.beta_gau   <- sd[(p+1):(p+q)]
    sd.theta_gau  <-sd[p+q+1]

# Ordenamos los coeficientes estimados en una tabla
    Reg_Cop_Gausiana <- data.frame(Estimate_Frec=round(beta_gau,4),Std.error.frec=round(sd.beta_gau,4),z_
    value_frec=round(beta_gau/sd.beta_gau,4),"Prob_frec"=round(2*(1-pnorm(abs(beta_gau/sd.beta_gau))),4),
    Estimate_Sev=round(alpha_gau,4),Std.error.sev=round(sd.alpha_gau,4),z_value_sev=round(alpha_gau/sd.alpha_
    gau,4),"Prob_Sev"=round(2*(1-pnorm(abs(alpha_gau/sd.alpha_gau))),4))
    #Reg_Cop_Gausiana <- data.frame(Estimate_Frec=round(beta_gau,4),Std.error.frec=0,z_value_frec=0,"Prob_
    frec"=0,Estimate_Sev=round(alpha_gau,4),Std.error.sev=0,z_value_sev=0,"Prob_Sev"=0)
    var_level<-row.names(Reg_Cop_Gausiana)
    Reg_Cop_Gausiana <- Reg_Cop_Gausiana %>% as.tibble() %>%  mutate(var_level=var_level) %>% relocate(var_
    level)

    # Agregamos los interceptos
    temp <- tibble(
      var_level = c(
        "Año2022",
        "RC_Antiguedadb. 2-9",
        "RC_Canal_VentaCorredores",
        "RC_Uso_VehParticular",
        "RC_Marca_VehL3",
        "RC_ZonaA2",
        "RC_Edad43-52"
      ),
      Estimate_Frec = 0
      ,
      Estimate_Sev = 0
    )

    # coeficientes del modelo de regresion con dependencia
    Reg_Cop_Gausiana <- Reg_Cop_Gausiana %>% bind_rows(temp) %>% arrange(var_level) %>% mutate(`exp(Estimate
    _Frec)`=round(exp(Estimate_Frec),4),`exp(Estimate_Sev)`=round(exp(Estimate_Sev),4))

    Reg_Cop_Gausiana <-Reg_Cop_Gausiana %>%  relocate(`exp(Estimate_Frec)`, .after=Prob_frec)

# Guardamos el modelo conjunto y los coeficientes
write.csv(Reg_Cop_Gausiana, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_
    GLM_Conjuntos/Responsabilidad_Civil/Coef_Gaus.csv", fileEncoding = "Latin1")
save(para_optim, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_
    Conjuntos/Responsabilidad_Civil/JointModel_Gaus.RData")

# Simulacion de la Pérdida Total
# ============================================================

    # CON DEPENDENCIA
    # ---------------
              lambda_loss_dep  <- exp(var_x%*%beta_gau)*data_CM.Expuesto
              mu_loss_dep      <- exp(var_x%*%alpha_gau)
              delta_loss_dep   <- delta_gau


              # Función de densidad fx

              f_loss <-
                function(loss,mu,d,lam,theta){
                  n<-length(loss)
                  if (length(lam)==1) lam  <- rep(lam,n)
                  if (length(mu)==1) mu    <- rep(mu,n)
                  out                      <- vector(length=n)

                  for(i in 1:n){
                    N  <-1:20
                    v  <-ppois(N,lam[i])
                    vv <-ppois(N-1,lam[i])
                    u  <-pgam(loss[i]/N,mu[i],d)

                    Der_cop <-Du_Gaus(u,v,theta)-Du_Gaus(u,vv,theta)
```

```
                    dummy    <-Der_cop*dgam(loss[i]/N,mu[i],d)/N
                    out[i]   <-sum(dummy)
                  }

                  out[loss<=0]=0
                  return(out)
                }

              # Función de densidad acumulada Fx
              F_loss <-
                function(loss,mu,d,lam,theta){
                  out<-vector(length=length(loss))

                  for(i in 1:length(loss)){
                    floss <- function(s){
                      f_loss(s,mu[i],d,lam[i],theta)
                    }
                    out[i] <- integrate(floss,0,loss[i]).value
                  }
                  return(out)
                }

              # Estimación del Total Loss (fuerte dedicacion computacional)

              #k <-length(lambda_loss_dep)
              k <- length(lambda_loss_dep) # Cantidad de numeros aleatorios (Debe ser del tamaño de lambda)
              m <- 100 # Cantidad de muestras aleatorias (Deben ser 10 mil)
              L <- vector(length=k)
              S <- vector(length=m)

              for(j in 1:m){
                r_uni<-runif(k)
                for(i in 1:k){
                  f_root <-function(s){
                    F_loss(s,mu_loss_dep[i],delta_loss_dep,lambda_loss_dep[i],theta_gau)-r_uni[i]
                  }

                  tryCatch(
                    error = function(cnd) mu_loss_dep[i]*lambda_loss_dep[i],
                    loss<-uniroot(f_root,lower = 0,upper = 500000)
                  )

                  #print(loss.root)
                  L[i]<-loss.root
                  perc<-paste(i/k*100,"%")
                  print(perc)
                }
                S[j]<-sum(L)
                perc<-paste(j/m*100,"%")
                print(perc)
                #print(S[j])
              }
        head(S)

      # CON INDEPENDENCIA
      # -----------------
              lambda_loss  <-predict(glm.Frec.PT, newdata=data_CM, type='response') # CAMBIAR LOS DATOS A UN
      MODELO DE RIESGO COLECTIVO
              mu_loss      <-predict(glm.Sev.PT, newdata=data_CM, type='response')
              delta_loss   <-delta_0

              M<-seq(1:10000)
              for(i in 1:10000){
                muestra_cant<-rpois(length(lambda_loss),lambda_loss)
                S<-vector(length=length(lambda_loss))
                for(j in 1:length(muestra_cant)){
                  if(muestra_cant[j]==0){
                    S[j]=0
                  }else{
                    S[j]=sum(rgam(n =muestra_cant[j],mu_loss[j],delta_loss ))
                  }
                }

                M[i] <- sum(S)
                porc<-paste((i/10000)*100,"%")
                print(porc)
```

```
                  #print(M[i])
                }

                hist(M, breaks = 50)
                mean(M)
                sum(data.Incurrido)


# Simulacion de la Frecuencia con  Independencia
      #ADECUAR LOS DATOS DE SIMULACION A UN ESTRUCTURA DE MODELO COLECTIVO DE RIESGO
      # Frecuencia
      N<-seq(1:10000)
      for(i in 1:10000){
        N[i] <- sum(simulate(glm.Frec.PT, nsim=1, type="response" ).sim_1)/sum(data_model.Expuesto)
        porc<-paste((i/10000)*100,"%")
        print(porc)
      }
      hist(N, breaks = 50)
      mean(N)
      mean(data.Frecuencia)

#####################################################################
# Limpiamos el ambiente de variables
  rm(list = ls())
  gc()

#####################################################################
# REGRESION CLAYTON
#####################################################################

  # Set Directory
  setwd("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/
    Responsabilidad_Civil")

  # Carga los modelos GLM MARGINALES calculados en otos querys
  load("glm.Frec.RC.RData")
  load("glm.Sev.RC.RData")
  load("Data_RC.RData")
  load("Data_RC_model.RData")
  load("Data_RC_model_CM.RData")

  # Creamos las variables de respuestas y la matriz de datos de cada glm marginal
  # SEVERIDAD
  x <- data_CM.Incurrido/data_CM.Cantidad #(Costo Medio)
  var_x <-model.matrix(glm.Sev.RC)  #data_CM[,3:12]
  # FRECUENCIA
  y <- data_model.Cantidad
  Expuestos <- data_model.Expuesto
  var_y <- model.matrix(glm.Frec.RC)   #data[,3:12]

  # Estimacion de parámetros iniciales utilizando las distribuciones marginales
  # SEVERIDAD
  sd.alpha <- sqrt(diag(vcov(glm.Sev.RC)))
  alpha_0  <- glm.Sev.RC.coefficients
  delta_0  <- summary(glm.Sev.RC).dispersion
  mu_0     <- exp(var_x%*%alpha_0)

  # FRECUENCIA
  beta_0   <- glm.Frec.RC.coefficients
  sd.beta  <- sqrt(diag(vcov(glm.Frec.RC)))
  lambda_0 <- exp(var_y%*%beta_0)*Expuestos

  # REGRESION MEDIANTE COPULAS

  # COPULA CLAYTON
  family=3
  # Estimación inicial del parámetro theta para estimarlo mediante el algoritmo de MPLE
  theta_ini <- BiCopEst(rank(data_model.Incurrido-exp(var_y%*%alpha_0) )/(length(y)+1),rank(y*Expuestos-
    lambda_0)/(length(y)+1), family=family).par
  #theta_ini <- BiCopEst(rank(data_CM.Incurrido-exp(var_x%*%alpha_0))/(length(x)+1),rank(data_CM.Cantidad*
    data_CM.Expuesto-exp(var_x%*%beta_0)*data_CM.Expuesto)/(length(x)+1), family=family).par
  tau_ini   <- BiCopPar2Tau(par=theta_ini,family = family)

  # Estimación de las seudo-observaciones
  u<-pgam(x,mu_0,delta_0)
  v   <- ppois(data_CM.Cantidad,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)
```

```
vv  <- ppois(data_CM.Cantidad-1,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)

# Creamos la función primera derivada de la Función Copula
Du_Clay <- function(u,v,theta){
  u[u<=0]=0.0001
  u[u>1]=1
  v[v<=0]=0.0001
  v[v>1]=1
  #out <- pnorm((qnorm(v)-theta*qnorm(u))/(sqrt(1-theta^2)))
  out <- (u^(-theta)+v^(-theta)-1)^((-1/theta)-1)*(u^(-theta-1))
  return(out)
}

# Función a maximizar
f_aux <- function(para){
  D_u_0<-Du_Clay(u,v,para)-Du_Clay(u,vv,para)
  D_u_0[D_u_0<=0]=1
  D_u_0 <- log(D_u_0)
  out<-(-sum(D_u_0))
  return(out)
}

# Proceso de maximización
#para_ini <- theta2z(theta_ini,family = family)
para.ifm <- optim(theta_ini,f_aux,method = "BFGS").par
#theta_0  <- z2theta(para.ifm,family = family)
theta_0  <- para.ifm # Este será nuestro parámetro inicial estimado con el método IFM y que luego usaremos
   para la regresión conjunta
tau      <-BiCopPar2Tau(par=theta_0,family=family) # tau de Khendal para estimar el grado de dependencia

# Estimamos los coeficientes de regresión y el parámetro cópula mediante máxima verosimilitud

# Creamos nuestra función a maximizar
################################################################
f_aux_reg <- function(para){

  p        <- ncol(var_x)
  q        <- ncol(var_x)
  alpha    <- para[1:p]
  beta     <- para[(p+1):(p+q)]
  # theta    <- z2theta(para[p+q+1],family)
  theta    <- z2theta(para[p+q+1],family = family)
  delta    <- para[p+q+2]
  lambda   <- as.vector(exp(var_y%*%beta)*data_model.Expuesto)
  mu       <- as.vector(exp(var_y%*%alpha))
  #dummy    <- density_joint(x,y,mu,delta,lambda,theta,family=family,zt=FALSE)
  u2 <- pgam(data_model.Incurrido,mu,delta)
  v2<-ppois(y,lambda)
  vv2<-ppois(y-1,lambda)

  marginal.x <- dgam(data_model.Incurrido,mu,delta)
  marginal.x[marginal.x>=1]=0
  marginal.x[marginal.x<=0]=0
  marginal.x[is.na(marginal.x)]=0
  marginal.x[marginal.x==-Inf]=0
  marginal.x[marginal.x==Inf]=0

  par_der    <- Du_Clay(u2,v2,theta)
  par_der1   <- Du_Clay(u2,vv2,theta)

  dummy<- par_der-par_der1

  dummy[y==0]=par_der[y==0]

  out<-marginal.x*dummy

  out[out<=0]=1e-10
  ll       <- -sum(log(out))

  #if(negative==TRUE) ll <- (-ll)
  return(ll)
}

################################################################
para_0<-c(alpha_0,beta_0,theta2z(theta_0,family=family),delta_0) # Parámetros iniciales
a<-now()
```

```
452  para_optim <- optim(para_0,f_aux_reg,method = "BFGS",hessian = TRUE) # Estimación Maxima verosmil de los
       parámetros (Toma hasta 3h sin matriz hessiana. 8h con Matriz hessiana)
453  b<-now()
454  b-a
455  #load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_Conjuntos/Perdida_
       Parcial/JointModel_Gaus.RData")
456  para_optim.value # Log_Likelihood
457  vector <- para_optim.par
458  p        <- ncol(var_y)
459  q        <- ncol(var_y)
460  Log_Likelihood<-para_optim.value
461  # Parámetros optimizados
462  alpha_clay<-vector[1:p]
463  beta_clay<-vector[(p+1):(p+q)]
464  theta_clay=z2theta(vector[p+q+1],family) #-0.09506928
465  delta_clay<-exp(vector[p+q+2]) #4.271691
466  tau_clay<-BiCopPar2Tau(par=theta_clay,family=family)#-0.06061453
467  head(beta_0)
468  head(beta_clay)
469  head(alpha_0)
470  head(alpha_clay)
471  # Determinamos los errores estandar para el test de Wald
472  hessian_clay   <- para_optim.hessian
473  Hinv           <- ginv(hessian_clay)
474  sd             <- sqrt(diag(Hinv))
475  sd.alpha_clay  <-sd[1:p]
476  sd.beta_clay   <- sd[(p+1):(p+q)]
477  sd.theta_clay  <-sd[p+q+1]
478
479  # Ordenamos los coeficientes estimados en una tabla
480  Reg_Cop_Clayton <- data.frame(Estimate_Frec=round(beta_clay,4),Std.error.frec=round(sd.beta_clay,4),z_
       value_frec=round(beta_clay/sd.beta_clay,4),"Prob_frec"=round(2*(1-pnorm(abs(beta_clay/sd.beta_clay))),4),
       Estimate_Sev=round(alpha_clay,4),Std.error.sev=round(sd.alpha_clay,4),z_value_sev=round(alpha_clay/sd.
       alpha_clay,4),"Prob_Sev"=round(2*(1-pnorm(abs(alpha_clay/sd.alpha_clay))),4))
481  #Reg_Cop_Clayton <- data.frame(Estimate_Frec=round(beta_clay,4),Std.error.frec=0,z_value_frec=0,"Prob_frec
       "=0,Estimate_Sev=round(alpha_clay,4),Std.error.sev=0,z_value_sev=0,"Prob_Sev"=0)
482  var_level<-row.names(Reg_Cop_Clayton)
483  Reg_Cop_Clayton <- Reg_Cop_Clayton %>% as.tibble() %>%  mutate(var_level=var_level) %>% relocate(var_level
       )
484
485  # Agregamos los interceptos
486  temp <- tibble(
487    var_level = c(
488      "Año2022",
489      "RC_Antiguedadb. 2-9",
490      "RC_Canal_VentaCorredores",
491      "RC_Uso_VehParticular",
492      "RC_Marca_VehL3",
493      "RC_ZonaA2",
494      "RC_Edad43-52"
495    ),
496    Estimate_Frec = 0
497    ,
498    Estimate_Sev = 0
499  )
500
501  # coeficientes del modelo de regresion con dependencia
502  Reg_Cop_Clayton <- Reg_Cop_Clayton %>% bind_rows(temp) %>% arrange(var_level) %>% mutate(`exp(Estimate_
       Frec)`=round(exp(Estimate_Frec),4),`exp(Estimate_Sev)`=round(exp(Estimate_Sev),4))
503
504  Reg_Cop_Clayton <-Reg_Cop_Clayton %>%  relocate(`exp(Estimate_Frec)`, .after=Prob_frec)
505
506  # Guardamos el modelo conjunto y los coeficientes
507  write.csv(Reg_Cop_Clayton, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_
       GLM_Conjuntos/Responsabilidad_Civil/Coef_Clay.csv", fileEncoding = "Latin1")
508  save(para_optim, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_
       Conjuntos/Responsabilidad_Civil/JointModel_Clay.RData")
509
510  # Simulacion de la Pérdida Total
511  # ============================================================
512
513  # CON DEPENDENCIA
514  # ---------------
515  lambda_loss_dep  <- exp(var_x%*%beta_clay)*data_CM.Expuesto
516  mu_loss_dep      <- exp(var_x%*%alpha_clay)
517  delta_loss_dep   <- delta_clay
```

```
518
519
520     # Función de densidad fx
521
522     f_loss <-
523       function(loss,mu,d,lam,theta){
524         n<-length(loss)
525         if (length(lam)==1) lam  <- rep(lam,n)
526         if (length(mu)==1) mu    <- rep(mu,n)
527         out                      <- vector(length=n)
528
529         for(i in 1:n){
530           N  <-1:20
531           v  <-ppois(N,lam[i])
532           vv <-ppois(N-1,lam[i])
533           u  <-pgam(loss[i]/N,mu[i],d)
534
535           Der_cop <-Du_Clay(u,v,theta)-Du_Clay(u,vv,theta)
536           dummy   <-Der_cop*dgam(loss[i]/N,mu[i],d)/N
537           out[i]  <-sum(dummy)
538         }
539
540         out[loss<=0]=0
541         return(out)
542       }
543
544     # Función de densidad acumulada Fx
545     F_loss <-
546       function(loss,mu,d,lam,theta){
547         out<-vector(length=length(loss))
548
549         for(i in 1:length(loss)){
550           floss <- function(s){
551             f_loss(s,mu[i],d,lam[i],theta)
552           }
553           out[i] <- integrate(floss,0,loss[i]).value
554         }
555         return(out)
556       }
557
558     # Estimación del Total Loss (fuerte dedicacion computacional)
559
560     #k <-length(lambda_loss_dep)
561     k <- 100#length(lambda_loss_dep) # Cantidad de numeros aleatorios (Debe ser del tamaño de lambda)
562     m <- 10 # Cantidad de muestras aleatorias (Deben ser 10 mil)
563     L <- vector(length=k)
564     S <- vector(length=m)
565
566     for(j in 1:m){
567       r_uni<-runif(k)
568       for(i in 1:k){
569         f_root <-function(s){
570           F_loss(s,mu_loss_dep[i],delta_loss_dep,lambda_loss_dep[i],theta_clay)-r_uni[i]
571         }
572
573         tryCatch(
574           error = function(cnd) loss<-mu_loss_dep[i]*lambda_loss_dep[i],
575           loss<-uniroot(f_root,lower = 0,upper = 500000).root
576         )
577
578         #print(loss.root)
579         L[i]<-loss
580         perc<-paste(i/k*100,"%")
581         print(perc)
582       }
583       S[j]<-sum(L)
584       perc<-paste(j/m*100,"%")
585       print(perc)
586       #print(S[j])
587     }
588     S
589
590     # CON INDEPENDENCIA
591     # -----------------
592     lambda_loss  <-predict(glm.Frec.PP, newdata=data_CM, type='response') # CAMBIAR LOS DATOS A UN MODELO DE
        RIESGO COLECTIVO
```

```
    mu_loss       <-predict(glm.Sev.PP, newdata=data_CM, type='response')
    delta_loss    <-delta_0

    M<-seq(1:10000)
    for(i in 1:10000){
      muestra_cant<-rpois(length(lambda_loss),lambda_loss)
      S<-vector(length=length(lambda_loss))
      for(j in 1:length(muestra_cant)){
        if(muestra_cant[j]==0){
          S[j]=0
        }else{
          S[j]=sum(rgam(n =muestra_cant[j],mu_loss[j],delta_loss ))
        }
      }

      M[i] <- sum(S)
      porc<-paste((i/10000)*100,"%")
      print(porc)
      #print(M[i])
    }

    hist(M)

    # Simulacion de la Frecuencia con  Independencia
    #ADECUAR LOS DATOS DE SIMULACION A UN ESTRUCTURA DE MODELO COLECTIVO DE RIESGO
    # Frecuencia
    N<-seq(1:10000)
    for(i in 1:10000){
      N[i] <- sum(simulate(glm.Frec.PP,nsim=1, type="response" ).sim_1)/sum(data_model.Expuesto)
      porc<-paste((i/10000)*100,"%")
      print(porc)
    }
    hist(N)
    ########################################################################
    # Limpiamos el ambiente de variables
    rm(list = ls())
    gc()

#########################################################################
# REGRESION GUMBEL
#########################################################################

    # Set Directory
    setwd("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/
      Responsabilidad_Civil")

    # Carga los modelos GLM MARGINALES calculados en otos querys
    load("glm.Frec.RC.RData")
    load("glm.Sev.RC.RData")
    load("Data_RC.RData")
    load("Data_RC_model.RData")
    load("Data_RC_model_CM.RData")

    # Creamos las variables de respuestas y la matriz de datos de cada glm marginal
    # SEVERIDAD
    x <- data_CM.Incurrido/data_CM.Cantidad #(Costo Medio)
    var_x <-model.matrix(glm.Sev.RC)  #data_CM[,3:12]
    # FRECUENCIA
    y <- data_model.Cantidad
    Expuestos <- data_model.Expuesto
    var_y <- model.matrix(glm.Frec.RC)   #data[,3:12]

    # Estimacion de parámetros iniciales utilizando las distribuciones marginales
    # SEVERIDAD
    sd.alpha <- sqrt(diag(vcov(glm.Sev.RC)))
    alpha_0  <- glm.Sev.RC.coefficients
    delta_0  <- summary(glm.Sev.RC).dispersion
    mu_0     <- exp(var_x%*%alpha_0)

    # FRECUENCIA
    beta_0   <- glm.Frec.RC.coefficients
    sd.beta  <- sqrt(diag(vcov(glm.Frec.RC)))
    lambda_0 <- exp(var_y%*%beta_0)*Expuestos

    # REGRESION MEDIANTE COPULAS
```

```
# COPULA GUMBEL
family=4
# Estimación inicial del parámetro theta para estimarlo mediante el algoritmo de MPLE
theta_ini <- BiCopEst(rank(data_model.Incurrido-exp(var_y%*%alpha_0) )/(length(y)+1),rank(y*Expuestos-
  lambda_0)/(length(y)+1), family=family).par
#theta_ini <- BiCopEst(rank(data_CM.Incurrido-exp(var_x%*%alpha_0))/(length(x)+1),rank(data_CM.Cantidad*
  data_CM.Expuesto-exp(var_x%*%beta_0)*data_CM.Expuesto)/(length(x)+1), family=family).par
tau_ini    <- BiCopPar2Tau(par=theta_ini,family = family)

# Estimación de las seudo-observaciones
u<-pgam(x,mu_0,delta_0)
v    <- ppois(data_CM.Cantidad,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)
vv   <- ppois(data_CM.Cantidad-1,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)

# Creamos la función primera derivada de la Función Copula
Du_Gum <- function(u,v,theta){
  u[u<=0]=0.0001
  u[u>1]=1
  v[v<=0]=0.0001
  v[v>1]=1
  #out <- pnorm((qnorm(v)-theta*qnorm(u))/(sqrt(1-theta^2)))
  #out <- (u^(-theta)+v^(-theta)-1)^((-1/theta)-1)*(u^(-theta-1))
  out <- (u^-1)*exp(-((-log(u))^theta+(-log(v))^theta)^(1/theta))
  return(out)
}

# Función a maximizar
f_aux <- function(para){
  D_u_0<-Du_Gum(u,v,para)-Du_Gum(u,vv,para)
  D_u_0[D_u_0<=0]=1
  D_u_0 <- log(D_u_0)
  out<-(sum(D_u_0))
  return(out)
}

# Proceso de maximización
#para_ini <- theta2z(theta_ini,family = family)
para.ifm <- optim(theta_ini,f_aux,method = "BFGS").par
#theta_0  <- z2theta(para.ifm,family = family)
theta_0  <- para.ifm # Este será nuestro parámetro inicial estimado con el método IFM y que luego usaremos
   para la regresión conjunta
tau      <-BiCopPar2Tau(par=theta_0,family=family) # tau de Khendal para estimar el grado de dependencia

# Estimamos los coeficientes de regresión y el parámetro cópula mediante máxima verosimilitud

# Creamos nuestra función a maximizar
#####################################################################
f_aux_reg <- function(para){

  p        <- ncol(var_x)
  q        <- ncol(var_x)
  alpha    <- para[1:p]
  beta     <- para[(p+1):(p+q)]
  # theta    <- z2theta(para[p+q+1],family)
  theta    <- z2theta(para[p+q+1],family = family)
  delta    <- para[p+q+2]
  lambda   <- as.vector(exp(var_y%*%beta)*data_model.Expuesto)
  mu       <- as.vector(exp(var_y%*%alpha))
  #dummy    <- density_joint(x,y,mu,delta,lambda,theta,family=family,zt=FALSE)
  u2 <- pgam(data_model.Incurrido,mu,delta)
  v2<-ppois(y,lambda)
  vv2<-ppois(y-1,lambda)

  marginal.x <- dgam(data_model.Incurrido,mu,delta)
  marginal.x[marginal.x>=1]=0
  marginal.x[marginal.x<=0]=0
  marginal.x[is.na(marginal.x)]=0
  marginal.x[marginal.x==-Inf]=0
  marginal.x[marginal.x==Inf]=0

  par_der    <- Du_Gum(u2,v2,theta)
  par_der1   <- Du_Gum(u2,vv2,theta)

  dummy<- par_der-par_der1

  dummy[y==0]=par_der[y==0]
```

```
741
742         out<-marginal.x*dummy
743
744         out[out<=0]=1e-10
745         ll        <- -sum(log(out))
746
747         #if(negative==TRUE) ll <- (-ll)
748         return(ll)
749       }
750
751       ###################################################################
752       para_0<-c(alpha_0,beta_0,theta2z(theta_0,family=family),delta_0) # Parámetros iniciales
753       a<-now()
754       para_optim <- optim(para_0,f_aux_reg,method = "BFGS",hessian = TRUE) # Estimación Maxima verosmil de los
            parámetros (Toma hasta 3h sin matriz hessiana. 8h con Matriz hessiana)
755       b<-now()
756       b-a #3.485547 hours
757       #load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_Conjuntos/Perdida_
            Parcial/JointModel_Gaus.RData")
758       para_optim.value # Log_Likelihood
759       vector <- para_optim.par
760       p         <- ncol(var_y)
761       q         <- ncol(var_y)
762       Log_Likelihood<-para_optim.value
763       # Parámetros optimizados
764       alpha_gum<-vector[1:p]
765       beta_gum<-vector[(p+1):(p+q)]
766       theta_gum=z2theta(vector[p+q+1],family)
767       delta_gum<-exp(vector[p+q+2])
768       tau_gum<-BiCopPar2Tau(par=theta_gum,family=family)
769       head(beta_0)
770       head(beta_gum)
771       head(alpha_0)
772       head(alpha_gum)
773
774       # Determinamos los errores estandar para el test de Wald
775       hessian_gum    <- para_optim.hessian
776       Hinv           <- ginv(hessian_gum)
777       sd             <- sqrt(diag(Hinv))
778       sd.alpha_gum   <-sd[1:p]
779       sd.beta_gum    <- sd[(p+1):(p+q)]
780       sd.theta_gum   <-sd[p+q+1]
781
782       # Ordenamos los coeficientes estimados en una tabla
783       Reg_Cop_Gumbel <- data.frame(Estimate_Frec=round(beta_gum,4),Std.error.frec=round(sd.beta_gum,4),z_value_
            frec=round(beta_gum/sd.beta_gum,4),"Prob_frec"=round(2*(1-pnorm(abs(beta_gum/sd.beta_gum))),4),Estimate_
            Sev=round(alpha_gum,4),Std.error.sev=round(sd.alpha_gum,4),z_value_sev=round(alpha_gum/sd.alpha_gum,4),"
            Prob_Sev"=round(2*(1-pnorm(abs(alpha_gum/sd.alpha_gum))),4))
784       #Reg_Cop_Gumbel <- data.frame(Estimate_Frec=round(beta_gum,4),Std.error.frec=0,z_value_frec=0,"Prob_frec
            "=0,Estimate_Sev=round(alpha_gum,4),Std.error.sev=0,z_value_sev=0,"Prob_Sev"=0)
785       var_level<-row.names(Reg_Cop_Gumbel)
786       Reg_Cop_Gumbel <- Reg_Cop_Gumbel %>% as.tibble() %>%  mutate(var_level=var_level) %>% relocate(var_level)
787
788       # Agregamos los interceptos
789
790       temp <- tibble(
791         var_level = c(
792           "Año2022",
793           "RC_Antiguedadb. 2-9",
794           "RC_Canal_VentaCorredores",
795           "RC_Uso_VehParticular",
796           "RC_Marca_VehL3",
797           "RC_ZonaA2",
798           "RC_Edad43-52"
799         ),
800         Estimate_Frec = 0
801         ,
802         Estimate_Sev = 0
803       )
804
805       # coeficientes del modelo de regresion con dependencia
806       Reg_Cop_Gumbel <- Reg_Cop_Gumbel %>% bind_rows(temp) %>% arrange(var_level) %>% mutate(`exp(Estimate_Frec)
            `=round(exp(Estimate_Frec),4),`exp(Estimate_Sev)`=round(exp(Estimate_Sev),4))
807
808       Reg_Cop_Gumbel <-Reg_Cop_Gumbel %>%  relocate(`exp(Estimate_Frec)`, .after=Prob_frec)
809
```

```
# Guardamos el modelo conjunto y los coeficientes
write.csv(Reg_Cop_Gumbel, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_Conjuntos/Responsabilidad_Civil/Coef_Gumb.csv", fileEncoding = "Latin1")
save(para_optim, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_Conjuntos/Responsabilidad_Civil/JointModel_Gumb.RData")

# Simulacion de la Pérdida Total
# ================================================================

# CON DEPENDENCIA
# ---------------
lambda_loss_dep  <- exp(var_x%*%beta_clay)*data_CM.Expuesto
mu_loss_dep      <- exp(var_x%*%alpha_clay)
delta_loss_dep   <- delta_clay


# Función de densidad fx

f_loss <-
  function(loss,mu,d,lam,theta){
    n<-length(loss)
    if (length(lam)==1) lam  <- rep(lam,n)
    if (length(mu)==1) mu    <- rep(mu,n)
    out                      <- vector(length=n)

    for(i in 1:n){
      N  <-1:20
      v  <-ppois(N,lam[i])
      vv <-ppois(N-1,lam[i])
      u  <-pgam(loss[i]/N,mu[i],d)

      Der_cop <-Du_Clay(u,v,theta)-Du_Clay(u,vv,theta)
      dummy   <-Der_cop*dgam(loss[i]/N,mu[i],d)/N
      out[i]  <-sum(dummy)
    }

    out[loss<=0]=0
    return(out)
  }

# Función de densidad acumulada Fx
F_loss <-
  function(loss,mu,d,lam,theta){
    out<-vector(length=length(loss))

    for(i in 1:length(loss)){
      floss <- function(s){
        f_loss(s,mu[i],d,lam[i],theta)
      }
      out[i] <- integrate(floss,0,loss[i]).value
    }
    return(out)
  }

# Estimación del Total Loss (fuerte dedicacion computacional)

#k <-length(lambda_loss_dep)
k <- 100#length(lambda_loss_dep) # Cantidad de numeros aleatorios (Debe ser del tamaño de lambda)
m <- 10 # Cantidad de muestras aleatorias (Deben ser 10 mil)
L <- vector(length=k)
S <- vector(length=m)

for(j in 1:m){
  r_uni<-runif(k)
  for(i in 1:k){
    f_root <-function(s){
      F_loss(s,mu_loss_dep[i],delta_loss_dep,lambda_loss_dep[i],theta_clay)-r_uni[i]
    }

    tryCatch(
      error = function(cnd) loss<-mu_loss_dep[i]*lambda_loss_dep[i],
      loss<-uniroot(f_root,lower = 0,upper = 500000).root
    )

    #print(loss.root)
    L[i]<-loss
```

```
884           perc<-paste(i/k*100,"%")
885           print(perc)
886         }
887         S[j]<-sum(L)
888         perc<-paste(j/m*100,"%")
889         print(perc)
890         #print(S[j])
891       }
892       S
893
894       # CON INDEPENDENCIA
895       # -----------------
896       lambda_loss  <-predict(glm.Frec.PP, newdata=data_CM, type='response') # CAMBIAR LOS DATOS A UN MODELO DE
            RIESGO COLECTIVO
897       mu_loss      <-predict(glm.Sev.PP, newdata=data_CM, type='response')
898       delta_loss   <-delta_0
899
900       M<-seq(1:10000)
901       for(i in 1:10000){
902         muestra_cant<-rpois(length(lambda_loss),lambda_loss)
903         S<-vector(length=length(lambda_loss))
904         for(j in 1:length(muestra_cant)){
905           if(muestra_cant[j]==0){
906             S[j]=0
907           }else{
908             S[j]=sum(rgam(n =muestra_cant[j],mu_loss[j],delta_loss ))
909           }
910         }
911
912         M[i] <- sum(S)
913         porc<-paste((i/10000)*100,"%")
914         print(porc)
915         #print(M[i])
916       }
917
918       hist(M)
919
920       # Simulacion de la Frecuencia con  Independencia
921       #ADECUAR LOS DATOS DE SIMULACION A UN ESTRUCTURA DE MODELO COLECTIVO DE RIESGO
922       # Frecuencia
923       N<-seq(1:10000)
924       for(i in 1:10000){
925         N[i] <- sum(simulate(glm.Frec.PP,nsim=1, type="response" ).sim_1)/sum(data_model.Expuesto)
926         porc<-paste((i/10000)*100,"%")
927         print(porc)
928       }
929       hist(N)
930     ########################################################################
931       # Limpiamos el ambiente de variables
932       rm(list = ls())
933       gc()
934
935 ########################################################################
936 # REGRESION FRANK
937 ########################################################################
938
939       # Set Directory
940       setwd("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/
            Responsabilidad_Civil")
941
942       # Carga los modelos GLM MARGINALES calculados en otos querys
943       load("glm.Frec.RC.RData")
944       load("glm.Sev.RC.RData")
945       load("Data_RC.RData")
946       load("Data_RC_model.RData")
947       load("Data_RC_model_CM.RData")
948
949       # Creamos las variables de respuestas y la matriz de datos de cada glm marginal
950       # SEVERIDAD
951       x <- data_CM.Incurrido/data_CM.Cantidad #(Costo Medio)
952       var_x <-model.matrix(glm.Sev.RC)  #data_CM[,3:12]
953       # FRECUENCIA
954       y <- data_model.Cantidad
955       Expuestos <- data_model.Expuesto
956       var_y <- model.matrix(glm.Frec.RC)   #data[,3:12]
957
```

```
# Estimacion de parámetros iniciales utilizando las distribuciones marginales
# SEVERIDAD
sd.alpha <- sqrt(diag(vcov(glm.Sev.RC)))
alpha_0  <- glm.Sev.RC.coefficients
delta_0  <- summary(glm.Sev.RC).dispersion
mu_0     <- exp(var_x%*%alpha_0)

# FRECUENCIA
beta_0   <- glm.Frec.RC.coefficients
sd.beta  <- sqrt(diag(vcov(glm.Frec.RC)))
lambda_0 <- exp(var_y%*%beta_0)*Expuestos

# REGRESION MEDIANTE COPULAS

# COPULA GUMBEL
family=5
# Estimación inicial del parámetro theta para estimarlo mediante el algoritmo de MPLE
theta_ini <- BiCopEst(rank(data_model.Incurrido-exp(var_y%*%alpha_0) )/(length(y)+1),rank(y*Expuestos-
  lambda_0)/(length(y)+1), family=family).par
#theta_ini <- BiCopEst(rank(data_CM.Incurrido-exp(var_x%*%alpha_0))/(length(x)+1),rank(data_CM.Cantidad*
  data_CM.Expuesto-exp(var_x%*%beta_0)*data_CM.Expuesto)/(length(x)+1), family=family).par
tau_ini   <- BiCopPar2Tau(par=theta_ini,family = family)

# Estimación de las seudo-observaciones
u<-pgam(x,mu_0,delta_0)
v   <- ppois(data_CM.Cantidad,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)
vv  <- ppois(data_CM.Cantidad-1,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)

# Creamos la función primera derivada de la Función Copula
Du_Frank <- function(u,v,theta){
  u[u<=0]=0.0001
  u[u>1]=1
  v[v<=0]=0.0001
  v[v>1]=1
  #out <- pnorm((qnorm(v)-theta*qnorm(u))/(sqrt(1-theta^2)))
  #out <- (u^(-theta)+v^(-theta)-1)^((-1/theta)-1)*(u^(-theta-1))
  #out <- (u^-1)*exp(-((-log(u))^theta+(-log(v))^theta)^(1/theta))
  out <- (exp(theta)*(exp(theta*v)-1))/(exp(theta*(u+1))+exp(theta*(v+1))-exp(theta)-exp(theta*(u+v)))
  return(out)
}

# Función a maximizar
f_aux <- function(para){
  D_u_0<-Du_Frank(u,v,para)-Du_Frank(u,vv,para)
  D_u_0[D_u_0<=0]=1
  D_u_0 <- log(D_u_0)
  out<-(-sum(D_u_0))
  return(out)
}

# Proceso de maximización
#para_ini <- theta2z(theta_ini,family = family)
para.ifm <- optim(theta_ini,f_aux,method = "BFGS").par
#theta_0  <- z2theta(para.ifm,family = family)
theta_0  <- para.ifm # Este será nuestro parámetro inicial estimado con el método IFM y que luego usaremos
   para la regresión conjunta
tau      <-BiCopPar2Tau(par=theta_0,family=family) # tau de Khendal para estimar el grado de dependencia

# Estimamos los coeficientes de regresión y el parámetro cópula mediante máxima verosimilitud

# Creamos nuestra función a maximizar
#####################################################################
f_aux_reg <- function(para){

  p        <- ncol(var_x)
  q        <- ncol(var_x)
  alpha    <- para[1:p]
  beta     <- para[(p+1):(p+q)]
  # theta    <- z2theta(para[p+q+1],family)
  theta    <- z2theta(para[p+q+1],family = family)
  delta    <- para[p+q+2]
  lambda   <- as.vector(exp(var_y%*%beta)*data_model.Expuesto)
  mu       <- as.vector(exp(var_y%*%alpha))
  #dummy    <- density_joint(x,y,mu,delta,lambda,theta,family=family,zt=FALSE)
  u2 <- pgam(data_model.Incurrido,mu,delta)
  v2<-ppois(y,lambda)
```

```
    vv2<-ppois(y-1,lambda)

    marginal.x <- dgam(data_model.Incurrido,mu,delta)
    marginal.x[marginal.x>=1]=0
    marginal.x[marginal.x<=0]=0
    marginal.x[is.na(marginal.x)]=0
    marginal.x[marginal.x==-Inf]=0
    marginal.x[marginal.x==Inf]=0

    par_der   <- Du_Frank(u2,v2,theta)
    par_der1  <- Du_Frank(u2,vv2,theta)

    dummy<- par_der-par_der1

    dummy[y==0]=par_der[y==0]

    out<-marginal.x*dummy

    out[out<=0]=1e-10
    ll       <- -sum(log(out))

    #if(negative==TRUE) ll <- (-ll)
    return(ll)
}

#################################################################
para_0<-c(alpha_0,beta_0,theta2z(theta_0,family=family),delta_0) # Parámetros iniciales
a<-now()
para_optim <- optim(para_0,f_aux_reg,method = "BFGS",hessian = TRUE) # Estimación Maxima verosmil de los
  parámetros (Toma hasta 3h sin matriz hessiana. 8h con Matriz hessiana)
b<-now()
b-a #17.499 mins
#load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_Conjuntos/Perdida_
  Parcial/JointModel_Gaus.RData")
para_optim.value # Log_Likelihood
vector <- para_optim.par
p        <- ncol(var_y)
q        <- ncol(var_y)
Log_Likelihood<-para_optim.value
# Parámetros optimizados
alpha_frank<-vector[1:p]
beta_frank<-vector[(p+1):(p+q)]
theta_frank=z2theta(vector[p+q+1],family)
delta_frank<-exp(vector[p+q+2]) #4.271691
tau_frank<-BiCopPar2Tau(par=theta_frank,family=family)
head(beta_0)
head(beta_frank)
head(alpha_0)
head(alpha_frank)

# Determinamos los errores estandar para el test de Wald
hessian_frank    <- para_optim.hessian
Hinv             <- ginv(hessian_frank)
sd               <- sqrt(diag(Hinv))
sd.alpha_frank   <-sd[1:p]
sd.beta_frank    <- sd[(p+1):(p+q)]
sd.theta_frank   <-sd[p+q+1]

# Ordenamos los coeficientes estimados en una tabla
Reg_Cop_Frank <- data.frame(Estimate_Frec=round(beta_frank,4),Std.error.frec=round(sd.beta_frank,4),z_
  value_frec=round(beta_frank/sd.beta_frank,4),"Prob_frec"=round(2*(1-pnorm(abs(beta_frank/sd.beta_frank)))
  ,4),Estimate_Sev=round(alpha_frank,4),Std.error.sev=round(sd.alpha_frank,4),z_value_sev=round(alpha_frank
  /sd.alpha_frank,4),"Prob_Sev"=round(2*(1-pnorm(abs(alpha_frank/sd.alpha_frank))),4))
#Reg_Cop_Frank <- data.frame(Estimate_Frec=round(beta_frank,4),Std.error.frec=0,z_value_frec=0,"Prob_frec
  "=0,Estimate_Sev=round(alpha_frank,4),Std.error.sev=0,z_value_sev=0,"Prob_Sev"=0)
var_level<-row.names(Reg_Cop_Frank)
Reg_Cop_Frank <- Reg_Cop_Frank %>% as.tibble() %>%  mutate(var_level=var_level) %>% relocate(var_level)

# Agregamos los interceptos
temp <- tibble(
  var_level = c(
    "Año2022",
    "RC_Antiguedadb. 2-9",
    "RC_Canal_VentaCorredores",
    "RC_Uso_VehParticular",
    "RC_Marca_VehL3",
```

```
        "RC_ZonaA2",
        "RC_Edad43-52"
      ),
      Estimate_Frec = 0
      ,
      Estimate_Sev = 0
    )


    # coeficientes del modelo de regresion con dependencia
    Reg_Cop_Frank <- Reg_Cop_Frank %>% bind_rows(temp) %>% arrange(var_level) %>% mutate(`exp(Estimate_Frec)`=
      round(exp(Estimate_Frec),4), `exp(Estimate_Sev)`=round(exp(Estimate_Sev),4))

    Reg_Cop_Frank <-Reg_Cop_Frank %>%  relocate(`exp(Estimate_Frec)`, .after=Prob_frec)

    # Guardamos el modelo conjunto y los coeficientes
    write.csv(Reg_Cop_Frank, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_
      GLM_Conjuntos/Responsabilidad_Civil/Coef_Frank.csv", fileEncoding = "Latin1")
    save(para_optim, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_
      Conjuntos/Responsabilidad_Civil/JointModel_Frank.RData")

    # Simulacion de la Pérdida Total
    # ============================================================

    # CON DEPENDENCIA
    # ---------------
    lambda_loss_dep  <- exp(var_x%*%beta_frank)*data_CM.Expuesto
    mu_loss_dep      <- exp(var_x%*%alpha_frank)
    delta_loss_dep   <- delta_frank


    # Función de densidad fx

    f_loss <-
      function(loss,mu,d,lam,theta){
        n<-length(loss)
        if (length(lam)==1) lam  <- rep(lam,n)
        if (length(mu)==1) mu    <- rep(mu,n)
        out                      <- vector(length=n)

        for(i in 1:n){
          N  <-1:20
          v  <-ppois(N,lam[i])
          vv <-ppois(N-1,lam[i])
          u  <-pgam(loss[i]/N,mu[i],d)

          Der_cop <-Du_Frank(u,v,theta)-Du_Frank(u,vv,theta) # Primera derivada de la función copula
          dummy   <-Der_cop*dgam(loss[i]/N,mu[i],d)/N # Teorela del Total Loss
          out[i]  <-sum(dummy)
        }

        out[loss<=0]=0
        return(out)
      }

    # Función de densidad acumulada Fx
    F_loss <-
      function(loss,mu,d,lam,theta){
        out<-vector(length=length(loss))

        for(i in 1:length(loss)){
          floss <- function(s){
            f_loss(s,mu[i],d,lam[i],theta)
          }
          out[i] <- integrate(floss,0,loss[i]).value # Integral para el cálculo del a F(x) a partir de f(x)
        }
        return(out)
      }

    # Estimación del Total Loss (fuerte dedicacion computacional)

    #k <-length(lambda_loss_dep)
    k <- 100#length(lambda_loss_dep) # Cantidad de numeros aleatorios (Debe ser del tamaño de lambda)
    m <- 10 # Cantidad de muestras aleatorias (Deben ser 10 mil)
    L <- vector(length=k)
    S <- vector(length=m)
```

```
1174
1175      for(j in 1:m){
1176        r_uni<-runif(k)
1177        for(i in 1:k){
1178          f_root <-function(s){
1179            F_loss(s,mu_loss_dep[i],delta_loss_dep,lambda_loss_dep[i],theta_frank)-r_uni[i]
1180          }
1181
1182          tryCatch(
1183            error = function(cnd) loss<-mu_loss_dep[i]*lambda_loss_dep[i],
1184            loss<-uniroot(f_root,lower = 0,upper = 500000).root
1185          )
1186
1187          #print(loss.root)
1188          L[i]<-loss
1189          perc<-paste(i/k*100,"%")
1190          print(perc)
1191        }
1192        S[j]<-sum(L)
1193        perc<-paste(j/m*100,"%")
1194        print(perc)
1195        #print(S[j])
1196      }
1197      S
1198
1199      # CON INDEPENDENCIA
1200      # ------------------
1201      lambda_loss  <-predict(glm.Frec.RC, newdata=data_CM, type='response') # CAMBIAR LOS DATOS A UN MODELO DE
          RIESGO COLECTIVO
1202      mu_loss      <-predict(glm.Sev.RC, newdata=data_CM, type='response')
1203      delta_loss   <-delta_0
1204
1205      M<-seq(1:10000)
1206      for(i in 1:10000){
1207        muestra_cant<-rpois(length(lambda_loss),lambda_loss)
1208        S<-vector(length=length(lambda_loss))
1209        for(j in 1:length(muestra_cant)){
1210          if(muestra_cant[j]==0){
1211            S[j]=0
1212          }else{
1213            S[j]=sum(rgam(n =muestra_cant[j],mu_loss[j],delta_loss ))
1214          }
1215        }
1216
1217        M[i] <- sum(S)
1218        porc<-paste((i/10000)*100,"%")
1219        print(porc)
1220        #print(M[i])
1221      }
1222
1223      hist(M,breaks = 50)
1224      mean(M)
1225      sum(data_model.Incurrido)
1226
1227      # Simulacion de la Frecuencia con  Independencia
1228      #ADECUAR LOS DATOS DE SIMULACION A UN ESTRUCTURA DE MODELO COLECTIVO DE RIESGO
1229      # Frecuencia
1230      N<-seq(1:10000)
1231      for(i in 1:10000){
1232        N[i] <- sum(simulate(glm.Frec.RC,nsim=1, type="response" ).sim_1)/sum(data_model.Expuesto)
1233        porc<-paste((i/10000)*100,"%")
1234        print(porc)
1235      }
1236
1237      hist(N,breaks = 50)
1238      mean(N)
1239      sum(data_model.Cantidad)/sum(data_model.Expuesto)
```

C.6. Modelo Conjunto - Asistencias

```
1  ##############################################################################
2  ##############################################################################
3  ####                                                                      ######
4  ####                  REGRESION BASADO EN COPULAS                       ######
5  ####                           ASISTENCIA                               ######
```

```
6  ####                                                                    ######
7  ##############################################################################
8  ##############################################################################
9
10 # Cargamos las librerías necesarias
11
12   #library(copula)
13   library(RODBC)
14   library(tidyverse)
15   library(MASS)
16   #library(GJRM)
17   #library(devtools)
18   library(CopulaRegression)
19   library(VineCopula)
20   #library(optimx)
21   #install_url('http://cran.r-project.org/src/contrib/Archive/CopulaRegression/CopulaRegression_0.1-5.tar.gz')
22
23 ####################################################################
24 # REGRESION GAUSSIANA
25 ####################################################################
26
27 # Set Directory
28   setwd("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Asistencia")
29
30 # Carga los modelos GLM MARGINALES calculados en otos querys
31   load("glm.Frec.AS.RData")
32   load("glm.Sev.AS.RData")
33   load("Data_AS.RData")
34   load("Data_AS_model.RData")
35   load("Data_AS_model_CM.RData")
36
37 # Creamos las variables de respuestas y la matriz de datos de cada glm marginal
38   # SEVERIDAD
39   x <- data_CM.Incurrido/data_CM.Cantidad #(Costo Medio)
40   var_x <-model.matrix(glm.Sev.AS)  #data_CM[,3:12]
41   # FRECUENCIA
42   y <- data_model.Cantidad
43   Expuestos <- data_model.Expuesto
44   var_y <- model.matrix(glm.Frec.AS)   #data[,3:12]
45
46 # Estimacion de parámetros iniciales utilizando las distribuciones marginales
47   # SEVERIDAD
48   sd.alpha <- sqrt(diag(vcov(glm.Sev.AS)))
49   alpha_0  <- glm.Sev.AS.coefficients
50   delta_0  <- summary(glm.Sev.AS).dispersion
51   mu_0     <- exp(var_x%*%alpha_0)
52
53   # FRECUENCIA
54   beta_0   <- glm.Frec.AS.coefficients
55   sd.beta  <- sqrt(diag(vcov(glm.Frec.AS)))
56   lambda_0 <- exp(var_y%*%beta_0)*Expuestos
57
58 # REGRESION MEDIANTE COPULAS
59
60   # COPULA GAUSSIANA
61   family=1
62   # Estimación inicial del parámetro theta para estimarlo mediante el algoritmo de MPLE
63   theta_ini <- BiCopEst(rank(data_model.Incurrido-exp(var_y%*%alpha_0))/(length(y)+1),rank(y*Expuestos-lambda
         _0)/(length(y)+1), family=family).par
64   #theta_ini <- BiCopEst(rank(data_CM.Incurrido-exp(var_x%*%alpha_0))/(length(x)+1),rank(data_CM.Cantidad*data
         _CM.Expuesto-exp(var_x%*%beta_0)*data_CM.Expuesto)/(length(x)+1), family=family).par
65   tau_ini    <- BiCopPar2Tau(par=theta_ini,family = family)
66
67   # Estimación de las seudo-observaciones
68   u<-pgam(x,mu_0,delta_0)
69   v    <- ppois(data_CM.Cantidad,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)
70   vv   <- ppois(data_CM.Cantidad-1,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)
71
72   # Creamos la función primera derivada de la Función Copula
73     Du_Gaus <- function(u,v,theta){
74       u[u<=0]=0.001
75       u[u>=1]=0.999
76       v[v<=0]=0.001
77       v[v>=1]=0.999
78       out <- pnorm((qnorm(v)-theta*qnorm(u))/(sqrt(1-theta^2)))
79     return(out)
```

```
  }

# Función a maximizar
f_aux <- function(para){
  D_u_0<-Du_Gaus(u,v,para)-Du_Gaus(u,vv,para)
  D_u_0[D_u_0<=0]=1
  D_u_0 <- log(D_u_0)
  out<-(-sum(D_u_0))
  return(out)
}

# Proceso de maximización
#para_ini <- theta2z(theta_ini,family = family)
para.ifm <- optim(theta_ini,f_aux,method = "BFGS").par
#theta_0  <- z2theta(para.ifm,family = family)
theta_0  <- para.ifm # Este será nuestro parámetro inicial estimado con el método IFM y que luego usaremos para la regresión conjunta
tau      <-BiCopPar2Tau(par=theta_0,family=family) # tau de Khendal para estimar el grado de dependencia

# Estimamos los coeficientes de regresión y el parámetro cópula mediante máxima verosimilitud

# Creamos nuestra función a maximizar
#####################################################################
  f_aux_reg <- function(para){

      p        <- ncol(var_x)
      q        <- ncol(var_x)
      alpha    <- para[1:p]
      beta     <- para[(p+1):(p+q)]
      # theta    <- z2theta(para[p+q+1],family)
      theta    <- para[p+q+1]
      delta    <- para[p+q+2]
      lambda   <- as.vector(exp(var_y%*%beta)*data_model.Expuesto)
      mu       <- as.vector(exp(var_y%*%alpha))
      #dummy    <- density_joint(x,y,mu,delta,lambda,theta,family=family,zt=FALSE)
      u2 <- pgam(data_model.Incurrido,mu,delta)
      v2<-ppois(y,lambda)
      vv2<-ppois(y-1,lambda)

      marginal.x <- dgam(data_model.Incurrido,mu,delta)
      marginal.x[marginal.x>=1]=0
      marginal.x[marginal.x<=0]=0
      marginal.x[is.na(marginal.x)]=0
      marginal.x[marginal.x==-Inf]=0
      marginal.x[marginal.x==Inf]=0

      par_der    <- Du_Gaus(u2,v2,theta)
      par_der1   <- Du_Gaus(u2,vv2,theta)

      dummy<- par_der-par_der1

      dummy[y==0]=par_der[y==0]

      out<-marginal.x*dummy

      out[out<=0]=1e-10
      ll       <- -sum(log(out))

      #if(negative==TRUE) ll <- (-ll)
      return(ll)
}

#####################################################################

para_0<-c(alpha_0,beta_0,theta_0,delta_0) # Parámetros iniciales
a<-now()
para_optim <- optim(para_0,f_aux_reg,method = "BFGS",hessian = TRUE) # Demora 6min
b<-now()
b-a
#load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_Conjuntos/Perdida_Parcial/JointModel_Gaus.RData")
para_optim.value # Log_Likelihood
vector <- para_optim.par
p        <- ncol(var_y)
q        <- ncol(var_y)

```

```
# Parámetros optimizados
    alpha_gau<-vector[1:p]
    beta_gau<-vector[(p+1):(p+q)]
    theta_gau=z2theta(vector[p+q+1],family)
    delta_gau<-vector[p+q+2]
    tau_gau<-BiCopPar2Tau(par=theta_gau,family=family)
    head(beta_0)
    head(beta_gau)
    head(alpha_0)
    head(alpha_gau)
# Determinamos los errores estandar para el test de Wald
    hessian_gau   <- para_optim.hessian
    Hinv          <- ginv(hessian_gau)
    sd            <- sqrt(diag(Hinv))
    sd.alpha_gau  <-sd[1:p]
    sd.beta_gau   <- sd[(p+1):(p+q)]
    sd.theta_gau  <-sd[p+q+1]

# Ordenamos los coeficientes estimados en una tabla
    Reg_Cop_Gausiana <- data.frame(Estimate_Frec=round(beta_gau,4),Std.error.frec=round(sd.beta_gau,4),z_
    value_frec=round(beta_gau/sd.beta_gau,4),"Prob_frec"=round(2*(1-pnorm(abs(beta_gau/sd.beta_gau))),4),
    Estimate_Sev=round(alpha_gau,4),Std.error.sev=round(sd.alpha_gau,4),z_value_sev=round(alpha_gau/sd.alpha_
    gau,4),"Prob_Sev"=round(2*(1-pnorm(abs(alpha_gau/sd.alpha_gau))),4))
    #Reg_Cop_Gausiana <- data.frame(Estimate_Frec=round(beta_gau,4),Std.error.frec=0,z_value_frec=0,"Prob_
    frec"=0,Estimate_Sev=round(alpha_gau,4),Std.error.sev=0,z_value_sev=0,"Prob_Sev"=0)
    var_level<-row.names(Reg_Cop_Gausiana)
    Reg_Cop_Gausiana <- Reg_Cop_Gausiana %>% as.tibble() %>%  mutate(var_level=var_level) %>% relocate(var_
    level)

    # Agregamos los interceptos
    temp <- tibble(
      var_level = c(
        "Año2022",
        "AS_Antiguedadb. 1-2",
        "AS_Canal_VentaCorredores",
        "AS_Uso_VehParticular",
        "AS_Marca_VehL2",
        "AS_ZonaA3",
        "AS_SAa.[1000-10000>"
      ),
      Estimate_Frec = 0
      ,
      Estimate_Sev = 0
    )

    # coeficientes del modelo de regresion con dependencia
    Reg_Cop_Gausiana <- Reg_Cop_Gausiana %>% bind_rows(temp) %>% arrange(var_level) %>% mutate(`exp(Estimate
    _Frec)`=round(exp(Estimate_Frec),4), `exp(Estimate_Sev)`=round(exp(Estimate_Sev),4))

    Reg_Cop_Gausiana <-Reg_Cop_Gausiana %>%  relocate(`exp(Estimate_Frec)`, .after=Prob_frec)

# Guardamos el modelo conjunto y los coeficientes
write.csv(Reg_Cop_Gausiana, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_
    GLM_Conjuntos/Asistencia/Coef_Gaus.csv", fileEncoding = "Latin1")
save(para_optim, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_
    Conjuntos/Asistencia/JointModel_Gaus.RData")

# Simulacion de la Pérdida Total
# ============================================================

    # CON DEPENDENCIA
    # ---------------
              lambda_loss_dep  <- exp(var_x%*%beta_gau)*data_CM.Expuesto
              mu_loss_dep      <- exp(var_x%*%alpha_gau)
              delta_loss_dep   <- delta_gau


              # Función de densidad fx

              f_loss <-
                function(loss,mu,d,lam,theta){
                  n<-length(loss)
                  if (length(lam)==1) lam   <- rep(lam,n)
                  if (length(mu)==1) mu     <- rep(mu,n)
                  out                       <- vector(length=n)

```

```
                for(i in 1:n){
                  N  <-1:20
                  v  <-ppois(N,lam[i])
                  vv <-ppois(N-1,lam[i])
                  u  <-pgam(loss[i]/N,mu[i],d)

                  Der_cop <-Du_Gaus(u,v,theta)-Du_Gaus(u,vv,theta)
                  dummy   <-Der_cop*dgam(loss[i]/N,mu[i],d)/N
                  out[i]  <-sum(dummy)
                }

                out[loss<=0]=0
                return(out)
              }

            # Función de densidad acumulada Fx
            F_loss <-
              function(loss,mu,d,lam,theta){
                out<-vector(length=length(loss))

                for(i in 1:length(loss)){
                  floss <- function(s){
                    f_loss(s,mu[i],d,lam[i],theta)
                  }
                  out[i] <- integrate(floss,0,loss[i]).value
                }
                return(out)
              }

            # Estimación del Total Loss (fuerte dedicacion computacional)

            #k <-length(lambda_loss_dep)
            k <- length(lambda_loss_dep) # Cantidad de numeros aleatorios (Debe ser del tamaño de lambda)
            m <- 100 # Cantidad de muestras aleatorias (Deben ser 10 mil)
            L <- vector(length=k)
            S <- vector(length=m)

            for(j in 1:m){
              r_uni<-runif(k)
              for(i in 1:k){
                f_root <-function(s){
                  F_loss(s,mu_loss_dep[i],delta_loss_dep,lambda_loss_dep[i],theta_gau)-r_uni[i]
                }

                tryCatch(
                  error = function(cnd) mu_loss_dep[i]*lambda_loss_dep[i],
                  loss<-uniroot(f_root,lower = 0,upper = 500000)
                )

                #print(loss.root)
                L[i]<-loss.root
                perc<-paste(i/k*100,"%")
                print(perc)
              }
              S[j]<-sum(L)
              perc<-paste(j/m*100,"%")
              print(perc)
              #print(S[j])
            }
      head(S)

# CON INDEPENDENCIA
# -----------------
          lambda_loss  <-predict(glm.Frec.AS, newdata=data_CM, type='response') # CAMBIAR LOS DATOS A UN MODELO DE RIESGO COLECTIVO
          mu_loss      <-predict(glm.Sev.AS, newdata=data_CM, type='response')
          delta_loss   <-delta_0

          M<-seq(1:10000)
          for(i in 1:10000){
            muestra_cant<-rpois(length(lambda_loss),lambda_loss)
            S<-vector(length=length(lambda_loss))
            for(j in 1:length(muestra_cant)){
              if(muestra_cant[j]==0){
                S[j]=0
              }else{
```

```
                    S[j]=sum(rgam(n =muestra_cant[j],mu_loss[j],delta_loss ))
                  }
                }

                M[i] <- sum(S)
                 porc<-paste((i/10000)*100,"%")
                 print(porc)
                 #print(M[i])
               }

               hist(M, breaks = 50)
               mean(M)
               sum(data.Incurrido)


# Simulacion de la Frecuencia con  Independencia
      #ADECUAR LOS DATOS DE SIMULACION A UN ESTRUCTURA DE MODELO COLECTIVO DE RIESGO
      # Frecuencia
      N<-seq(1:10000)
      for(i in 1:10000){
        N[i] <- sum(simulate(glm.Frec.AS,nsim=1, type="response" ).sim_1)/sum(data_model.Expuesto)
        porc<-paste((i/10000)*100,"%")
        print(porc)
      }
      hist(N, breaks = 50)
      mean(N)
      sum(data.Frecuencia)/sum(data.Expuesto)

#####################################################################
# Limpiamos el ambiente de variables
  rm(list = ls())
  gc()

#####################################################################
# REGRESION CLAYTON
#####################################################################

  # Set Directory
  setwd("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Asistencia")

  # Carga los modelos GLM MARGINALES calculados en otos querys
  load("glm.Frec.AS.RData")
  load("glm.Sev.AS.RData")
  load("Data_AS.RData")
  load("Data_AS_model.RData")
  load("Data_AS_model_CM.RData")

  # Creamos las variables de respuestas y la matriz de datos de cada glm marginal
  # SEVERIDAD
  x <- data_CM.Incurrido/data_CM.Cantidad #(Costo Medio)
  var_x <-model.matrix(glm.Sev.AS)  #data_CM[,3:12]
  # FRECUENCIA
  y <- data_model.Cantidad
  Expuestos <- data_model.Expuesto
  var_y <- model.matrix(glm.Frec.AS)   #data[,3:12]

  # Estimacion de parámetros iniciales utilizando las distribuciones marginales
  # SEVERIDAD
  sd.alpha <- sqrt(diag(vcov(glm.Sev.AS)))
  alpha_0  <- glm.Sev.AS.coefficients
  delta_0  <- summary(glm.Sev.AS).dispersion
  mu_0     <- exp(var_x%*%alpha_0)

  # FRECUENCIA
  beta_0   <- glm.Frec.AS.coefficients
  sd.beta  <- sqrt(diag(vcov(glm.Frec.AS)))
  lambda_0 <- exp(var_y%*%beta_0)*Expuestos

  # REGRESION MEDIANTE COPULAS

  # COPULA CLAYTON
  family=3
  # Estimación inicial del parámetro theta para estimarlo mediante el algoritmo de MPLE
  theta_ini <- BiCopEst(rank(data_model.Incurrido-exp(var_y%*%alpha_0) )/(length(y)+1),rank(y*Expuestos-
    lambda_0)/(length(y)+1), family=family).par
```

```
#theta_ini <- BiCopEst(rank(data_CM.Incurrido-exp(var_x%*%alpha_0))/(length(x)+1),rank(data_CM.Cantidad*
  data_CM.Expuesto-exp(var_x%*%beta_0)*data_CM.Expuesto)/(length(x)+1), family=family).par
tau_ini    <- BiCopPar2Tau(par=theta_ini,family = family)

# Estimación de las seudo-observaciones
u<-pgam(x,mu_0,delta_0)
v    <- ppois(data_CM.Cantidad,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)
vv   <- ppois(data_CM.Cantidad-1,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)

# Creamos la función primera derivada de la Función Copula
Du_Clay <- function(u,v,theta){
  u[u<=0]=0.0001
  u[u>1]=1
  v[v<=0]=0.0001
  v[v>1]=1
  #out <- pnorm((qnorm(v)-theta*qnorm(u))/(sqrt(1-theta^2)))
  out <- (u^(-theta)+v^(-theta)-1)^((-1/theta)-1)*(u^(-theta-1))
  return(out)
}

# Función a maximizar
f_aux <- function(para){
  D_u_0<-Du_Clay(u,v,para)-Du_Clay(u,vv,para)
  D_u_0[D_u_0<=0]=1
  D_u_0 <- log(D_u_0)
  out<-(-sum(D_u_0))
  return(out)
}

# Proceso de maximización
#para_ini <- theta2z(theta_ini,family = family)
para.ifm <- optim(theta_ini,f_aux,method = "BFGS").par
#theta_0  <- z2theta(para.ifm,family = family)
theta_0  <- para.ifm # Este será nuestro parámetro inicial estimado con el método IFM y que luego usaremos
   para la regresión conjunta
tau       <-BiCopPar2Tau(par=theta_0,family=family) # tau de Khendal para estimar el grado de dependencia

# Estimamos los coeficientes de regresión y el parámetro cópula mediante máxima verosimilitud

# Creamos nuestra función a maximizar
#####################################################################
f_aux_reg <- function(para){

  p        <- ncol(var_x)
  q        <- ncol(var_x)
  alpha    <- para[1:p]
  beta     <- para[(p+1):(p+q)]
  # theta    <- z2theta(para[p+q+1],family)
  theta    <- z2theta(para[p+q+1],family = family)
  delta    <- para[p+q+2]
  lambda   <- as.vector(exp(var_y%*%beta)*data_model.Expuesto)
  mu       <- as.vector(exp(var_y%*%alpha))
  #dummy    <- density_joint(x,y,mu,delta,lambda,theta,family=family,zt=FALSE)
  u2 <- pgam(data_model.Incurrido,mu,delta)
  v2<-ppois(y,lambda)
  vv2<-ppois(y-1,lambda)

  marginal.x <- dgam(data_model.Incurrido,mu,delta)
  marginal.x[marginal.x>=1]=0
  marginal.x[marginal.x<=0]=0
  marginal.x[is.na(marginal.x)]=0
  marginal.x[marginal.x==-Inf]=0
  marginal.x[marginal.x==Inf]=0

  par_der    <- Du_Clay(u2,v2,theta)
  par_der1   <- Du_Clay(u2,vv2,theta)

  dummy<- par_der-par_der1

  dummy[y==0]=par_der[y==0]

  out<-marginal.x*dummy

  out[out<=0]=1e-10
  ll         <- -sum(log(out))

```

```
    #if(negative==TRUE) ll <- (-ll)
    return(ll)
  }

  ###############################################################
  para_0<-c(alpha_0,beta_0,theta2z(theta_0,family=family),delta_0) # Parámetros iniciales
  a<-now()
  para_optim <- optim(para_0,f_aux_reg,method = "BFGS",hessian = TRUE) # Estimación Maxima verosmil de los
    parámetros (Toma hasta 3h sin matriz hessiana. 8h con Matriz hessiana)
  b<-now()
  b-a
  #load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_Conjuntos/Perdida_
    Parcial/JointModel_Gaus.RData")
  para_optim.value # Log_Likelihood
  vector <- para_optim.par
  p        <- ncol(var_y)
  q        <- ncol(var_y)
  Log_Likelihood<-para_optim.value
  # Parámetros optimizados
  alpha_clay<-vector[1:p]
  beta_clay<-vector[(p+1):(p+q)]
  theta_clay=z2theta(vector[p+q+1],family) #-0.09506928
  delta_clay<-vector[p+q+2]
  tau_clay<-BiCopPar2Tau(par=theta_clay,family=family)#-0.06061453
  head(beta_0)
  head(beta_clay)
  head(alpha_0)
  head(alpha_clay)
  # Determinamos los errores estandar para el test de Wald
  hessian_clay    <- para_optim.hessian
  Hinv            <- ginv(hessian_clay)
  sd              <- sqrt(diag(Hinv))
  sd.alpha_clay   <-sd[1:p]
  sd.beta_clay    <- sd[(p+1):(p+q)]
  sd.theta_clay   <-sd[p+q+1]

  # Ordenamos los coeficientes estimados en una tabla
  Reg_Cop_Clayton <- data.frame(Estimate_Frec=round(beta_clay,4),Std.error.frec=round(sd.beta_clay,4),z_
    value_frec=round(beta_clay/sd.beta_clay,4),"Prob_frec"=round(2*(1-pnorm(abs(beta_clay/sd.beta_clay))),4),
    Estimate_Sev=round(alpha_clay,4),Std.error.sev=round(sd.alpha_clay,4),z_value_sev=round(alpha_clay/sd.
    alpha_clay,4),"Prob_Sev"=round(2*(1-pnorm(abs(alpha_clay/sd.alpha_clay))),4))
  #Reg_Cop_Clayton <- data.frame(Estimate_Frec=round(beta_clay,4),Std.error.frec=0,z_value_frec=0,"Prob_frec
    "=0,Estimate_Sev=round(alpha_clay,4),Std.error.sev=0,z_value_sev=0,"Prob_Sev"=0)
  var_level<-row.names(Reg_Cop_Clayton)
  Reg_Cop_Clayton <- Reg_Cop_Clayton %>% as.tibble() %>%  mutate(var_level=var_level) %>% relocate(var_level
    )

  # Agregamos los interceptos
  temp <- tibble(
    var_level = c(
      "Año2022",
      "AS_Antiguedadb. 1-2",
      "AS_Canal_VentaCorredores",
      "AS_Uso_VehParticular",
      "AS_Marca_VehL2",
      "AS_ZonaA3",
      "AS_SAa.[1000-10000>"
    ),
    Estimate_Frec = 0
    ,
    Estimate_Sev = 0
  )

  # coeficientes del modelo de regresion con dependencia
  Reg_Cop_Clayton <- Reg_Cop_Clayton %>% bind_rows(temp) %>% arrange(var_level) %>% mutate(`exp(Estimate_
    Frec)`=round(exp(Estimate_Frec),4), `exp(Estimate_Sev)`=round(exp(Estimate_Sev),4))

  Reg_Cop_Clayton <-Reg_Cop_Clayton %>%  relocate(`exp(Estimate_Frec)`, .after=Prob_frec)

  # Guardamos el modelo conjunto y los coeficientes
  write.csv(Reg_Cop_Clayton, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_
    GLM_Conjuntos/Asistencia/Coef_Clay.csv", fileEncoding = "Latin1")
  save(para_optim, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_
    Conjuntos/Asistencia/JointModel_Clay.RData")

  # Simulacion de la Pérdida Total
```

```
# ============================================================

# CON DEPENDENCIA
# ---------------
lambda_loss_dep   <- exp(var_x%*%beta_clay)*data_CM.Expuesto
mu_loss_dep       <- exp(var_x%*%alpha_clay)
delta_loss_dep    <- delta_clay
# Función de densidad fx
f_loss <-
  function(loss,mu,d,lam,theta){
    n<-length(loss)
    if (length(lam)==1) lam   <- rep(lam,n)
    if (length(mu)==1) mu     <- rep(mu,n)
    out                       <- vector(length=n)

    for(i in 1:n){
      N  <-1:20
      v  <-ppois(N,lam[i])
      vv <-ppois(N-1,lam[i])
      u  <-pgam(loss[i]/N,mu[i],d)

      Der_cop <-Du_Clay(u,v,theta)-Du_Clay(u,vv,theta)
      dummy   <-Der_cop*dgam(loss[i]/N,mu[i],d)/N
      out[i]  <-sum(dummy)
    }

    out[loss<=0]=0
    return(out)
  }

# Función de densidad acumulada Fx
F_loss <-
  function(loss,mu,d,lam,theta){
    out<-vector(length=length(loss))

    for(i in 1:length(loss)){
      floss <- function(s){
        f_loss(s,mu[i],d,lam[i],theta)
      }
      out[i] <- integrate(floss,0,loss[i]).value
    }
    return(out)
  }

# Estimación del Total Loss (fuerte dedicacion computacional)

#k <-length(lambda_loss_dep)
k <- 100#length(lambda_loss_dep) # Cantidad de numeros aleatorios (Debe ser del tamaño de lambda)
m <- 10 # Cantidad de muestras aleatorias (Deben ser 10 mil)
L <- vector(length=k)
S <- vector(length=m)

for(j in 1:m){
  r_uni<-runif(k)
  for(i in 1:k){
    f_root <-function(s){
      F_loss(s,mu_loss_dep[i],delta_loss_dep,lambda_loss_dep[i],theta_clay)-r_uni[i]
    }

    tryCatch(
      error = function(cnd) loss<-mu_loss_dep[i]*lambda_loss_dep[i],
      loss<-uniroot(f_root,lower = 0,upper = 500000).root
    )

    #print(loss.root)
    L[i]<-loss
    perc<-paste(i/k*100,"%")
    print(perc)
  }
  S[j]<-sum(L)
  perc<-paste(j/m*100,"%")
  print(perc)
  #print(S[j])
}
S
```

```
    # CON INDEPENDENCIA
    # -----------------
    lambda_loss  <-predict(glm.Frec.PP, newdata=data_CM, type='response') # CAMBIAR LOS DATOS A UN MODELO DE
      RIESGO COLECTIVO
    mu_loss      <-predict(glm.Sev.PP, newdata=data_CM, type='response')
    delta_loss   <-delta_0

    M<-seq(1:10000)
    for(i in 1:10000){
      muestra_cant<-rpois(length(lambda_loss),lambda_loss)
      S<-vector(length=length(lambda_loss))
      for(j in 1:length(muestra_cant)){
        if(muestra_cant[j]==0){
          S[j]=0
        }else{
          S[j]=sum(rgam(n =muestra_cant[j],mu_loss[j],delta_loss ))
        }
      }

      M[i] <- sum(S)
      porc<-paste((i/10000)*100,"%")
      print(porc)
      #print(M[i])
    }

    hist(M)
    # Simulacion de la Frecuencia con  Independencia
    #ADECUAR LOS DATOS DE SIMULACION A UN ESTRUCTURA DE MODELO COLECTIVO DE RIESGO
    # Frecuencia
    N<-seq(1:10000)
    for(i in 1:10000){
      N[i] <- sum(simulate(glm.Frec.PP,nsim=1, type="response" ).sim_1)/sum(data_model.Expuesto)
      porc<-paste((i/10000)*100,"%")
      print(porc)
    }
    hist(N)


    ######################################################################
    # Limpiamos el ambiente de variables
    rm(list = ls())
    gc()

######################################################################
# REGRESION GUMBEL
######################################################################

    # Set Directory
    setwd("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Asistencia"
      )

    # Carga los modelos GLM MARGINALES calculados en otos querys
    load("glm.Frec.AS.RData")
    load("glm.Sev.AS.RData")
    load("Data_AS.RData")
    load("Data_AS_model.RData")
    load("Data_AS_model_CM.RData")

    # Creamos las variables de respuestas y la matriz de datos de cada glm marginal
    # SEVERIDAD
    x <- data_CM.Incurrido/data_CM.Cantidad #(Costo Medio)
    var_x <-model.matrix(glm.Sev.AS)  #data_CM[,3:12]
    # FRECUENCIA
    y <- data_model.Cantidad
    Expuestos <- data_model.Expuesto
    var_y <- model.matrix(glm.Frec.AS)   #data[,3:12]

    # Estimacion de parámetros iniciales utilizando las distribuciones marginales
    # SEVERIDAD
    sd.alpha <- sqrt(diag(vcov(glm.Sev.AS)))
    alpha_0  <- glm.Sev.AS.coefficients
    delta_0  <- summary(glm.Sev.AS).dispersion
    mu_0     <- exp(var_x%*%alpha_0)

    # FRECUENCIA
    beta_0   <- glm.Frec.AS.coefficients
```

```
661   sd.beta  <- sqrt(diag(vcov(glm.Frec.AS)))
662   lambda_0 <- exp(var_y%*%beta_0)*Expuestos
663
664   # REGRESION MEDIANTE COPULAS
665
666   # COPULA GUMBEL
667   family=4
668   # Estimación inicial del parámetro theta para estimarlo mediante el algoritmo de MPLE
669   theta_ini <- BiCopEst(rank(data_model.Incurrido-exp(var_y%*%alpha_0) )/(length(y)+1),rank(y*Expuestos-
        lambda_0)/(length(y)+1), family=family).par
670   #theta_ini <- BiCopEst(rank(data_CM.Incurrido-exp(var_x%*%alpha_0))/(length(x)+1),rank(data_CM.Cantidad*
        data_CM.Expuesto-exp(var_x%*%beta_0)*data_CM.Expuesto)/(length(x)+1), family=family).par
671   tau_ini    <- BiCopPar2Tau(par=theta_ini,family = family)
672
673   # Estimación de las seudo-observaciones
674   u<-pgam(x,mu_0,delta_0)
675   v    <- ppois(data_CM.Cantidad,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)
676   vv   <- ppois(data_CM.Cantidad-1,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)
677
678   # Creamos la función primera derivada de la Función Copula
679   Du_Gum <- function(u,v,theta){
680     u[u<=0]=0.0001
681     u[u>1]=1
682     v[v<=0]=0.0001
683     v[v>1]=1
684     #out <- pnorm((qnorm(v)-theta*qnorm(u))/(sqrt(1-theta^2)))
685     #out <- (u^(-theta)+v^(-theta)-1)^((-1/theta)-1)*(u^(-theta-1))
686     out <- (u^-1)*exp(-((-log(u))^theta+(-log(v))^theta)^(1/theta))
687     return(out)
688   }
689
690   # Función a maximizar
691   f_aux <- function(para){
692     D_u_0<-Du_Gum(u,v,para)-Du_Gum(u,vv,para)
693     D_u_0[D_u_0<=0]=1
694     D_u_0 <- log(D_u_0)
695     out<-(sum(D_u_0))
696     return(out)
697   }
698
699   # Proceso de maximización
700   #para_ini <- theta2z(theta_ini,family = family)
701   para.ifm <- optim(theta_ini,f_aux,method = "BFGS").par
702   #theta_0  <- z2theta(para.ifm,family = family)
703   theta_0  <- para.ifm # Este será nuestro parámetro inicial estimado con el método IFM y que luego usaremos
        para la regresión conjunta
704   tau       <-BiCopPar2Tau(par=theta_0,family=family) # tau de Khendal para estimar el grado de dependencia
705
706   # Estimamos los coeficientes de regresión y el parámetro cópula mediante máxima verosimilitud
707
708   # Creamos nuestra función a maximizar
709   ######################################################################
710   f_aux_reg <- function(para){
711
712     p        <- ncol(var_x)
713     q        <- ncol(var_x)
714     alpha    <- para[1:p]
715     beta     <- para[(p+1):(p+q)]
716     # theta    <- z2theta(para[p+q+1],family)
717     theta    <- z2theta(para[p+q+1],family = family)
718     delta    <- para[p+q+2]
719     lambda   <- as.vector(exp(var_y%*%beta)*data_model.Expuesto)
720     mu       <- as.vector(exp(var_y%*%alpha))
721     #dummy    <- density_joint(x,y,mu,delta,lambda,theta,family=family,zt=FALSE)
722     u2 <- pgam(data_model.Incurrido,mu,delta)
723     v2<-ppois(y,lambda)
724     vv2<-ppois(y-1,lambda)
725
726     marginal.x <- dgam(data_model.Incurrido,mu,delta)
727     marginal.x[marginal.x>=1]=0
728     marginal.x[marginal.x<=0]=0
729     marginal.x[is.na(marginal.x)]=0
730     marginal.x[marginal.x==-Inf]=0
731     marginal.x[marginal.x==Inf]=0
732
733     par_der    <- Du_Gum(u2,v2,theta)
```

```
734       par_der1   <- Du_Gum(u2,vv2,theta)
735
736       dummy<- par_der-par_der1
737
738       dummy[y==0]=par_der[y==0]
739
740       out<-marginal.x*dummy
741
742       out[out<=0]=1e-10
743       ll        <- -sum(log(out))
744
745       #if(negative==TRUE) ll <- (-ll)
746       return(ll)
747     }
748
749     ############################################################
750     para_0<-c(alpha_0,beta_0,theta2z(theta_0,family=family),delta_0) # Parámetros iniciales
751     a<-now()
752     para_optim <- optim(para_0,f_aux_reg,method = "BFGS",hessian = TRUE) # Estimación Maxima verosmil de los
          parámetros (Toma hasta 3h sin matriz hessiana. 8h con Matriz hessiana)
753     b<-now()
754     b-a #10.10871 mins
755     #load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_Conjuntos/Perdida_
          Parcial/JointModel_Gaus.RData")
756     para_optim.value # Log_Likelihood
757     vector <- para_optim.par
758     p         <- ncol(var_y)
759     q         <- ncol(var_y)
760     Log_Likelihood<-para_optim.value
761     # Parámetros optimizados
762     alpha_gum<-vector[1:p]
763     beta_gum<-vector[(p+1):(p+q)]
764     theta_gum=z2theta(vector[p+q+1],family)
765     delta_gum<-vector[p+q+2]
766     tau_gum<-BiCopPar2Tau(par=theta_gum,family=family)
767     head(beta_0)
768     head(beta_gum)
769     head(alpha_0)
770     head(alpha_gum)
771
772     # Determinamos los errores estandar para el test de Wald
773     hessian_gum    <- para_optim.hessian
774     Hinv           <- ginv(hessian_gum)
775     sd             <- sqrt(diag(Hinv))
776     sd.alpha_gum   <-sd[1:p]
777     sd.beta_gum    <- sd[(p+1):(p+q)]
778     sd.theta_gum   <-sd[p+q+1]
779
780     # Ordenamos los coeficientes estimados en una tabla
781     Reg_Cop_Gumbel <- data.frame(Estimate_Frec=round(beta_gum,4),Std.error.frec=round(sd.beta_gum,4),z_value_
          frec=round(beta_gum/sd.beta_gum,4),"Prob_frec"=round(2*(1-pnorm(abs(beta_gum/sd.beta_gum))),4),Estimate_
          Sev=round(alpha_gum,4),Std.error.sev=round(sd.alpha_gum,4),z_value_sev=round(alpha_gum/sd.alpha_gum,4),"
          Prob_Sev"=round(2*(1-pnorm(abs(alpha_gum/sd.alpha_gum))),4))
782     #Reg_Cop_Gumbel <- data.frame(Estimate_Frec=round(beta_gum,4),Std.error.frec=0,z_value_frec=0,"Prob_frec
          "=0,Estimate_Sev=round(alpha_gum,4),Std.error.sev=0,z_value_sev=0,"Prob_Sev"=0)
783     var_level<-row.names(Reg_Cop_Gumbel)
784     Reg_Cop_Gumbel <- Reg_Cop_Gumbel %>% as.tibble() %>%  mutate(var_level=var_level) %>% relocate(var_level)
785
786     # Agregamos los interceptos
787
788     temp <- tibble(
789       var_level = c(
790         "Año2022",
791         "AS_Antiguedadb. 1-2",
792         "AS_Canal_VentaCorredores",
793         "AS_Uso_VehParticular",
794         "AS_Marca_VehL2",
795         "AS_ZonaA3",
796         "AS_SAa.[1000-10000>"
797       ),
798       Estimate_Frec = 0
799       ,
800       Estimate_Sev = 0
801     )
802
803
```

```
# coeficientes del modelo de regresion con dependencia
Reg_Cop_Gumbel <- Reg_Cop_Gumbel %>% bind_rows(temp) %>% arrange(var_level) %>% mutate(`exp(Estimate_Frec)`=round(exp(Estimate_Frec),4),`exp(Estimate_Sev)`=round(exp(Estimate_Sev),4))

Reg_Cop_Gumbel <-Reg_Cop_Gumbel %>%  relocate(`exp(Estimate_Frec)`, .after=Prob_frec)

# Guardamos el modelo conjunto y los coeficientes
write.csv(Reg_Cop_Gumbel, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_Conjuntos/Asistencia/Coef_Gumb.csv", fileEncoding = "Latin1")
save(para_optim, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_Conjuntos/Asistencia/JointModel_Gumb.RData")

# Simulacion de la Pérdida Total
# ============================================================

# CON DEPENDENCIA
# ---------------
lambda_loss_dep  <- exp(var_x%*%beta_clay)*data_CM.Expuesto
mu_loss_dep      <- exp(var_x%*%alpha_clay)
delta_loss_dep   <- delta_clay


# Función de densidad fx

f_loss <-
  function(loss,mu,d,lam,theta){
    n<-length(loss)
    if (length(lam)==1) lam  <- rep(lam,n)
    if (length(mu)==1) mu    <- rep(mu,n)
    out                      <- vector(length=n)

    for(i in 1:n){
      N  <-1:20
      v  <-ppois(N,lam[i])
      vv <-ppois(N-1,lam[i])
      u  <-pgam(loss[i]/N,mu[i],d)

      Der_cop <-Du_Clay(u,v,theta)-Du_Clay(u,vv,theta)
      dummy   <-Der_cop*dgam(loss[i]/N,mu[i],d)/N
      out[i]  <-sum(dummy)
    }

    out[loss<=0]=0
    return(out)
  }

# Función de densidad acumulada Fx
F_loss <-
  function(loss,mu,d,lam,theta){
    out<-vector(length=length(loss))

    for(i in 1:length(loss)){
      floss <- function(s){
        f_loss(s,mu[i],d,lam[i],theta)
      }
      out[i] <- integrate(floss,0,loss[i]).value
    }
    return(out)
  }

# Estimación del Total Loss (fuerte dedicacion computacional)

#k <-length(lambda_loss_dep)
k <- 100#length(lambda_loss_dep) # Cantidad de numeros aleatorios (Debe ser del tamaño de lambda)
m <- 10 # Cantidad de muestras aleatorias (Deben ser 10 mil)
L <- vector(length=k)
S <- vector(length=m)

for(j in 1:m){
  r_uni<-runif(k)
  for(i in 1:k){
    f_root <-function(s){
      F_loss(s,mu_loss_dep[i],delta_loss_dep,lambda_loss_dep[i],theta_clay)-r_uni[i]
    }

    tryCatch(
```

```
            error = function(cnd) loss<-mu_loss_dep[i]*lambda_loss_dep[i],
            loss<-uniroot(f_root,lower = 0,upper = 500000).root
          )

          #print(loss.root)
          L[i]<-loss
          perc<-paste(i/k*100,"%")
          print(perc)
        }
        S[j]<-sum(L)
        perc<-paste(j/m*100,"%")
        print(perc)
        #print(S[j])
      }
      S

      # CON INDEPENDENCIA
      # -----------------
      lambda_loss  <-predict(glm.Frec.PP, newdata=data_CM, type='response') # CAMBIAR LOS DATOS A UN MODELO DE
        RIESGO COLECTIVO
      mu_loss      <-predict(glm.Sev.PP, newdata=data_CM, type='response')
      delta_loss   <-delta_0

      M<-seq(1:10000)
      for(i in 1:10000){
        muestra_cant<-rpois(length(lambda_loss),lambda_loss)
        S<-vector(length=length(lambda_loss))
        for(j in 1:length(muestra_cant)){
          if(muestra_cant[j]==0){
            S[j]=0
          }else{
            S[j]=sum(rgam(n =muestra_cant[j],mu_loss[j],delta_loss ))
          }
        }

        M[i] <- sum(S)
        porc<-paste((i/10000)*100,"%")
        print(porc)
        #print(M[i])
      }

      hist(M)



      # Simulacion de la Frecuencia con  Independencia
      #ADECUAR LOS DATOS DE SIMULACION A UN ESTRUCTURA DE MODELO COLECTIVO DE RIESGO
      # Frecuencia
      N<-seq(1:10000)
      for(i in 1:10000){
        N[i] <- sum(simulate(glm.Frec.PP,nsim=1, type="response" ).sim_1)/sum(data_model.Expuesto)
        porc<-paste((i/10000)*100,"%")
        print(porc)
      }
      hist(N)
    ######################################################################
      # Limpiamos el ambiente de variables
      rm(list = ls())
      gc()

#########################################################################
# REGRESION FRANK
#########################################################################

      # Set Directory
      setwd("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/04_GLM_Marginales/Asistencia"
        )

      # Carga los modelos GLM MARGINALES calculados en otos querys
      load("glm.Frec.AS.RData")
      load("glm.Sev.AS.RData")
      load("Data_AS.RData")
      load("Data_AS_model.RData")
      load("Data_AS_model_CM.RData")

      # Creamos las variables de respuestas y la matriz de datos de cada glm marginal
```

```
# SEVERIDAD
x <- data_CM.Incurrido/data_CM.Cantidad #(Costo Medio)
var_x <-model.matrix(glm.Sev.AS)  #data_CM[,3:12]
# FRECUENCIA
y <- data_model.Cantidad
Expuestos <- data_model.Expuesto
var_y <- model.matrix(glm.Frec.AS)   #data[,3:12]

# Estimacion de parámetros iniciales utilizando las distribuciones marginales
# SEVERIDAD
sd.alpha <- sqrt(diag(vcov(glm.Sev.AS)))
alpha_0  <- glm.Sev.AS.coefficients
delta_0  <- summary(glm.Sev.AS).dispersion
mu_0     <- exp(var_x%*%alpha_0)

# FRECUENCIA
beta_0   <- glm.Frec.AS.coefficients
sd.beta  <- sqrt(diag(vcov(glm.Frec.AS)))
lambda_0 <- exp(var_y%*%beta_0)*Expuestos

# REGRESION MEDIANTE COPULAS

# COPULA FRANK
family=5
# Estimación inicial del parámetro theta para estimarlo mediante el algoritmo de MPLE
theta_ini <- BiCopEst(rank(data_model.Incurrido-exp(var_y%*%alpha_0) )/(length(y)+1),rank(y*Expuestos-
  lambda_0)/(length(y)+1), family=family).par
#theta_ini <- BiCopEst(rank(data_CM.Incurrido-exp(var_x%*%alpha_0))/(length(x)+1),rank(data_CM.Cantidad*
  data_CM.Expuesto-exp(var_x%*%beta_0)*data_CM.Expuesto)/(length(x)+1), family=family).par
tau_ini   <- BiCopPar2Tau(par=theta_ini,family = family)

# Estimación de las seudo-observaciones
u<-pgam(x,mu_0,delta_0)
v   <- ppois(data_CM.Cantidad,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)
vv  <- ppois(data_CM.Cantidad-1,lambda = exp(var_x%*%beta_0)*data_CM.Expuesto)

# Creamos la función primera derivada de la Función Copula
Du_Frank <- function(u,v,theta){
  u[u<=0]=0.0001
  u[u>1]=1
  v[v<=0]=0.0001
  v[v>1]=1
  #out <- pnorm((qnorm(v)-theta*qnorm(u))/(sqrt(1-theta^2)))
  #out <- (u^(-theta)+v^(-theta)-1)^((-1/theta)-1)*(u^(-theta-1))
  #out <- (u^-1)*exp(-((-log(u))^theta+(-log(v))^theta)^(1/theta))
  out <- (exp(theta)*(exp(theta*v)-1))/(exp(theta*(u+1))+exp(theta*(v+1))-exp(theta)-exp(theta*(u+v)))
  return(out)
}

# Función a maximizar
f_aux <- function(para){
  D_u_0<-Du_Frank(u,v,para)-Du_Frank(u,vv,para)
  D_u_0[D_u_0<=0]=1
  D_u_0 <- log(D_u_0)
  out<-(-sum(D_u_0))
  return(out)
}

# Proceso de maximización
#para_ini <- theta2z(theta_ini,family = family)
para.ifm <- optim(theta_ini,f_aux,method = "BFGS").par
#theta_0  <- z2theta(para.ifm,family = family)
theta_0  <- para.ifm # Este será nuestro parámetro inicial estimado con el método IFM y que luego usaremos
   para la regresión conjunta
tau      <-BiCopPar2Tau(par=theta_0,family=family) # tau de Khendal para estimar el grado de dependencia

# Estimamos los coeficientes de regresión y el parámetro cópula mediante máxima verosimilitud

# Creamos nuestra función a maximizar
#################################################################
f_aux_reg <- function(para){

  p       <- ncol(var_x)
  q       <- ncol(var_x)
  alpha   <- para[1:p]
  beta    <- para[(p+1):(p+q)]
```

```
    # theta    <- z2theta(para[p+q+1],family)
    theta    <- z2theta(para[p+q+1],family = family)
    delta    <- para[p+q+2]
    lambda   <- as.vector(exp(var_y%*%beta)*data_model.Expuesto)
    mu       <- as.vector(exp(var_y%*%alpha))
    #dummy    <- density_joint(x,y,mu,delta,lambda,theta,family=family,zt=FALSE)
    u2 <- pgam(data_model.Incurrido,mu,delta)
    v2<-ppois(y,lambda)
    vv2<-ppois(y-1,lambda)

    marginal.x <- dgam(data_model.Incurrido,mu,delta)
    marginal.x[marginal.x>=1]=0
    marginal.x[marginal.x<=0]=0
    marginal.x[is.na(marginal.x)]=0
    marginal.x[marginal.x==-Inf]=0
    marginal.x[marginal.x==Inf]=0

    par_der    <- Du_Frank(u2,v2,theta)
    par_der1   <- Du_Frank(u2,vv2,theta)

    dummy<- par_der-par_der1

    dummy[y==0]=par_der[y==0]

    out<-marginal.x*dummy

    out[out<=0]=1e-10
    ll       <- -sum(log(out))

    #if(negative==TRUE) ll <- (-ll)
    return(ll)
}

#################################################################
para_0<-c(alpha_0,beta_0,theta2z(theta_0,family=family),delta_0) # Parámetros iniciales
a<-now()
para_optim <- optim(para_0,f_aux_reg,method = "BFGS",hessian = TRUE) # Estimación Maxima verosmil de los
  parámetros (Toma hasta 3h sin matriz hessiana. 8h con Matriz hessiana)
b<-now()
b-a #5.13348 mins
#load("C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_Conjuntos/Perdida_
  Parcial/JointModel_Gaus.RData")
para_optim.value # Log_Likelihood
vector <- para_optim.par
p        <- ncol(var_y)
q        <- ncol(var_y)
Log_Likelihood<-para_optim.value
# Parámetros optimizados
alpha_frank<-vector[1:p]
beta_frank<-vector[(p+1):(p+q)]
theta_frank=z2theta(vector[p+q+1],family)
delta_frank<-vector[p+q+2]
tau_frank<-BiCopPar2Tau(par=theta_frank,family=family)
head(beta_0)
head(beta_frank)
head(alpha_0)
head(alpha_frank)

# Determinamos los errores estandar para el test de Wald
hessian_frank    <- para_optim.hessian
Hinv             <- ginv(hessian_frank)
sd               <- sqrt(diag(Hinv))
sd.alpha_frank   <-sd[1:p]
sd.beta_frank    <- sd[(p+1):(p+q)]
sd.theta_frank   <-sd[p+q+1]

# Ordenamos los coeficientes estimados en una tabla
Reg_Cop_Frank <- data.frame(Estimate_Frec=round(beta_frank,4),Std.error.frec=round(sd.beta_frank,4),z_
  value_frec=round(beta_frank/sd.beta_frank,4),"Prob_frec"=round(2*(1-pnorm(abs(beta_frank/sd.beta_frank)))
  ,4),Estimate_Sev=round(alpha_frank,4),Std.error.sev=round(sd.alpha_frank,4),z_value_sev=round(alpha_frank
  /sd.alpha_frank,4),"Prob_Sev"=round(2*(1-pnorm(abs(alpha_frank/sd.alpha_frank))),4))
#Reg_Cop_Frank <- data.frame(Estimate_Frec=round(beta_frank,4),Std.error.frec=0,z_value_frec=0,"Prob_frec
  "=0,Estimate_Sev=round(alpha_frank,4),Std.error.sev=0,z_value_sev=0,"Prob_Sev"=0)
var_level<-row.names(Reg_Cop_Frank)
Reg_Cop_Frank <- Reg_Cop_Frank %>% as.tibble() %>%  mutate(var_level=var_level) %>% relocate(var_level)
```

```
# Agregamos los interceptos
temp <- tibble(
  var_level = c(
    "Año2022",
    "AS_Antiguedadb. 1-2",
    "AS_Canal_VentaCorredores",
    "AS_Uso_VehParticular",
    "AS_Marca_VehL2",
    "AS_ZonaA3",
    "AS_SAa.[1000-10000>"
  ),
  Estimate_Frec = 0
  ,
  Estimate_Sev = 0
)


# coeficientes del modelo de regresion con dependencia
Reg_Cop_Frank <- Reg_Cop_Frank %>% bind_rows(temp) %>% arrange(var_level) %>% mutate(`exp(Estimate_Frec)`=
  round(exp(Estimate_Frec),4), `exp(Estimate_Sev)`=round(exp(Estimate_Sev),4))

Reg_Cop_Frank <-Reg_Cop_Frank %>%  relocate(`exp(Estimate_Frec)`, .after=Prob_frec)

# Guardamos el modelo conjunto y los coeficientes
write.csv(Reg_Cop_Frank, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_
  GLM_Conjuntos/Asistencia/Coef_Frank.csv", fileEncoding = "Latin1")
save(para_optim, file="C:/Users/josephgarcia1/OneDrive - KPMG/Tesis_Actuarial/03_Modelos_GLM/06_GLM_
  Conjuntos/Asistencia/JointModel_Frank.RData")

# Simulacion de la Pérdida Total
# ======================================================

# CON DEPENDENCIA
# ---------------
lambda_loss_dep  <- exp(var_x%*%beta_frank)*data_CM.Expuesto
mu_loss_dep      <- exp(var_x%*%alpha_frank)
delta_loss_dep   <- delta_frank


# Función de densidad fx

f_loss <-
  function(loss,mu,d,lam,theta){
    n<-length(loss)
    if (length(lam)==1) lam  <- rep(lam,n)
    if (length(mu)==1) mu    <- rep(mu,n)
    out                      <- vector(length=n)

    for(i in 1:n){
      N  <-1:20
      v  <-ppois(N,lam[i])
      vv <-ppois(N-1,lam[i])
      u  <-pgam(loss[i]/N,mu[i],d)

      Der_cop <-Du_Frank(u,v,theta)-Du_Frank(u,vv,theta) # Primera derivada de la función copula
      dummy   <-Der_cop*dgam(loss[i]/N,mu[i],d)/N # Teorela del Total Loss
      out[i]  <-sum(dummy)
    }

    out[loss<=0]=0
    return(out)
  }

# Función de densidad acumulada Fx
F_loss <-
  function(loss,mu,d,lam,theta){
    out<-vector(length=length(loss))

    for(i in 1:length(loss)){
      floss <- function(s){
        f_loss(s,mu[i],d,lam[i],theta)
      }
      out[i] <- integrate(floss,0,loss[i]).value # Integral para el cálculo del a F(x) a partir de f(x)
    }
    return(out)
  }
```

```
1167
1168    # Estimación del Total Loss (fuerte dedicacion computacional)
1169
1170    #k <-length(lambda_loss_dep)
1171    k <- 100#length(lambda_loss_dep) # Cantidad de numeros aleatorios (Debe ser del tamaño de lambda)
1172    m <- 10 # Cantidad de muestras aleatorias (Deben ser 10 mil)
1173    L <- vector(length=k)
1174    S <- vector(length=m)
1175
1176    for(j in 1:m){
1177      r_uni<-runif(k)
1178      for(i in 1:k){
1179        f_root <-function(s){
1180          F_loss(s,mu_loss_dep[i],delta_loss_dep,lambda_loss_dep[i],theta_frank)-r_uni[i]
1181        }
1182
1183        tryCatch(
1184          error = function(cnd) loss<-mu_loss_dep[i]*lambda_loss_dep[i],
1185          loss<-uniroot(f_root,lower = 0,upper = 500000).root
1186        )
1187
1188        #print(loss.root)
1189        L[i]<-loss
1190        perc<-paste(i/k*100,"%")
1191        print(perc)
1192      }
1193      S[j]<-sum(L)
1194      perc<-paste(j/m*100,"%")
1195      print(perc)
1196      #print(S[j])
1197    }
1198    S
1199
1200    # CON INDEPENDENCIA
1201    # -----------------
1202    lambda_loss  <-predict(glm.Frec.RC, newdata=data_CM, type='response') # CAMBIAR LOS DATOS A UN MODELO DE
       RIESGO COLECTIVO
1203    mu_loss      <-predict(glm.Sev.RC, newdata=data_CM, type='response')
1204    delta_loss   <-delta_0
1205
1206    M<-seq(1:10000)
1207    for(i in 1:10000){
1208      muestra_cant<-rpois(length(lambda_loss),lambda_loss)
1209      S<-vector(length=length(lambda_loss))
1210      for(j in 1:length(muestra_cant)){
1211        if(muestra_cant[j]==0){
1212          S[j]=0
1213        }else{
1214          S[j]=sum(rgam(n =muestra_cant[j],mu_loss[j],delta_loss ))
1215        }
1216      }
1217
1218      M[i] <- sum(S)
1219      porc<-paste((i/10000)*100,"%")
1220      print(porc)
1221      #print(M[i])
1222    }
1223
1224    hist(M,breaks = 50)
1225    mean(M)
1226    sum(data_model.Incurrido)
1227    # Simulacion de la Frecuencia con  Independencia
1228    #ADECUAR LOS DATOS DE SIMULACION A UN ESTRUCTURA DE MODELO COLECTIVO DE RIESGO
1229    # Frecuencia
1230    N<-seq(1:10000)
1231    for(i in 1:10000){
1232      N[i] <- sum(simulate(glm.Frec.RC,nsim=1, type="response" ).sim_1)/sum(data_model.Expuesto)
1233      porc<-paste((i/10000)*100,"%")
1234      print(porc)
1235    }
1236
1237    hist(N,breaks = 50)
1238    mean(N)
1239    sum(data_model.Cantidad)/sum(data_model.Expuesto)
```