PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

Escuela de Posgrado

MODELOS DE REGRESIÓN PARAMÉTRICOS BIVARIADOS PARA EL ANÁLISIS DE SUPERVIVENCIA: UNA APLICACIÓN A TIEMPOS DE INFECCIÓN Y SÍNTOMAS

Tesis para obtener el grado académico de Maestro en Estadística que presenta:

Jorge Víctor Arangoitia Fernández Baca

Asesor:

Víctor Giancarlo Sal Y Rosas Celi

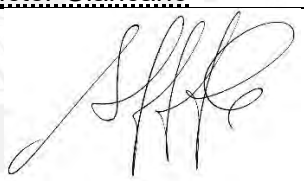
Lima, 2023

Informe de Similitud

Yo, Víctor Giancarlo Sal y Rosas Celi, docente de la Escuela de Posgrado de la Pontificia Universidad Católica del Perú, asesor de la tesis de investigación titulado "Modelos de regresión paramétricos bivariados para el análisis de supervivencia: Una aplicación a tiempo de infección y síntomas", del autor Jorge Víctor Arangoitia Fernández Baca dejo constancia de lo siguiente:

- El mencionado documento tiene un índice de puntuación de similitud de 9%. Así lo consigna el reporte de similitud emitido por el software *Turnitin* el 14/11/2023.
- He revisado con detalle dicho reporte y la Tesis o Trabajo de Suficiencia Profesional, y no se advierte indicios de plagio.
- Las citas a otros autores y sus respectivas referencias cumplen con las pautas académicas.

Lugar y fecha: Lima, 17 de noviembre de 2023

Apellidos y nombres del asesor: Sal y Rosas Celi, Víctor Giancarlo	
DNI: 40361284	
ORCID: 0000-0001-8636-7142	

Dedicatoria

A todo aquel guiado por la curiosidad de saber un poquito más sobre cómo funciona la naturaleza.

Agradecimentos

A mi familia, amigos, profesores, asesor y a todos aquellos grandes investigadores que sentaron las bases fundamentales para la implementación de este trabajo.

Resumen

Cuando se realizan estudios sobre tratamientos nuevos que pueden aplicarse a pacientes que sufren de una determinada enfermedad, un factor fundamental para evaluar la efectividad de dicho tratamiento es la determinación de si el paciente adquirió la enfermedad o no, y si presentó síntomas de dicha enfermedad, o no lo hizo. Dicho de otro modo, se requiere conocer (o estimar) el efecto que tuvo la aplicación del nuevo tratamiento en el tiempo en el cual el paciente adquiró la infección y el tiempo en el cual comenzó a presentar síntomas, variables que permiten determinar si el tratamiento pudo prevenir la enfermedad, o al menos ralentizar su propagación, y si pudo evitar o atenuar la aparición de síntomas.

Es importante resaltar que el estudio del tiempo transcurrido hasta la ocurrencia de una infección o de la aparición de los síntomas, es un caso particular del análisis de supervivencia, rama de la estadística que tiene como objetivo el estudio del tiempo transcurrido hasta la ocurrencia de un evento, así como el efecto que tienen en dicho tiempo variables características propias de los individuos a los que les ocurre el evento, por ejemplo, en el caso de pacientes, se puede considerar el tratamiento que se le aplicó (el estándar o el nuevo), la edad, el género, entre otros. A estas últimas se les conoce como covariables.

Así, el presente trabajo propone dos modelos paramétricos bivariados basados en distribuciones y métodos estadísticos utilizados en el análisis de supervivencia, modelos que permitirán estudiar el comportamiento conjunto del tiempo a infección y del tiempo a síntomas, considerando la relación intrínseca existente entre ambas variables.

De esta manera, el método de estimación a utilizar será el modelo de tiempo de falla acelerado, modelo de regresión lineal en el cual se asume que el logaritmo del tiempo de infección y el logaritmo del tiempo de síntomas son iguales a una función lineal de las covariables más un error multiplicado por el parámetro de escala correspondiente a cada tiempo. En ese sentido, se cuentan con dos errores (uno para el tiempo de infección y otro para el

de síntomas) que corresponden al componente aleatorio de la regresión, componenente que se modelará de forma conjunta de las siguientes dos maneras:

- Asumiendo que ambos errores siguen una distribución bivariada de valores extremos.
- Asumiendo un modelo de cópulas, en la cual se asume que cada tiempo presenta una distribución marginal Weibull, y la relación de dependencia de ambos tiempos obedece a una cópula Gumbel.

Finalmente, el método anterior se puede aplicar a una muestra determinada a fin de estimar los parámetros de las distribuciones asumidas, y de esta manera determinar el efecto que tienen cada una de las covariables en los tiempos de infección y de síntomas. En este trabajo en particular, se aplicará el modelo en el estudio de notificación de parejas, llevado a cabo por Golden en el 2005 y que tuvo como objetivo verificar si un grupo de pacientes presentó re-infección y síntomas de una enfermedad previa, así como el efecto de una nueva terapia sobre tales eventos.

Palabras-clave: Análisis de supervivencia, distribuciones bivariadas, cópulas.

Abstract

When studies are carried out on new treatments that can be applied to patients suffering from a certain disease, a fundamental factor to evaluate the effectiveness of such treatment is the determination of whether the patient acquired the disease or not, and if he presented symptoms of that disease, or did not. In other words, it is necessary to know (or estimate) the effect that the application of the new treatment had on the time in which the patient acquired the infection and the time in which he began to present symptoms, variables that make it possible to determine if the treatment was able to prevent the disease, or at least slow its spread, and whether it was able to prevent or mitigate the onset of symptoms.

It is important to highlight that the study of the time elapsed until the occurrence of an infection or the appearance of symptoms is a particular case of survival analysis, a branch of statistics whose objective is the study of the time elapsed until the occurrence of a event, as well as the effect of variables characteristic of the individuals to whom the event occurs, for example, in the case of patients, the treatment applied to them (the standard or the new), age, gender, among others. This are known as covariates.

Thus, the present work proposes two bivariate parametric models based on distributions and statistical methods used in survival analysis, models that will allow studying the joint behavior of time to infection and time to symptoms, considering the intrinsic relationship between both variables.

Then, the estimation method to be used will be the accelerated failure time model, a linear regression model in which it is assumed that the logarithm of the infection time and the logarithm of the symptom time are equal to a linear function of the covariates plus an error multiplied by the scale parameter corresponding to each time. With this in mind, there are two errors (one for the time of infection and the other for the time of symptoms) that correspond to the random component of the regression, a component that will be modeled

jointly in the following two ways:

- Assuming that both errors follow a bivariate extreme value distribution.
- Assuming a copula model, in which it is assumed that each time presents a Weibull marginal distribution, and the dependency relationship of both times obeys a Gumbel copula.

Finally, the previous method can be applied to a specific sample in order to estimate the parameters of the assumed distributions, and in this way determine the effect that each of the covariates has on the times of infection and symptoms. In this particular work, the model will be applied in the couple notification study, carried out by Golden in 2005 and whose objective was to verify if a group of patients presented reinfection and symptoms of a previous disease, as well as the effect of a new therapy on such events.

Keywords: Survival analysis, bivariate distributions, copulas.

Índice general

Capítulo 1

Introducción

1.1. Consideraciones preliminares

Cuando se realizan estudios sobre los tratamientos aplicables a una determinada enfermedad, resulta fundamental estimar el tiempo en el cual los pacientes adquieren dicha enfermedad así como el efecto que tuvo dicho tratamiento (es decir, la intervención) para prevenirla. Asimismo, es relevante también evaluar si una intervención que no previene infección, puede retardar o acelerar el tiempo de aparición de síntomas. Justamente, el estudio de estas dos variables (tiempo a infección y tiempo a síntomas) es el objeto del presente trabajo, el cual constituye un caso particular del análisis de supervivencia, que es una rama de la estadística que tiene como objetivo el estudio del tiempo transcurrido hasta la ocurrencia de un evento de interés en particular. Así, para el presente trabajo, se propone el uso de modelos paramétricos bivariados utilizados en el análisis de supervivencia para estimar el comportamiento de los tiempos a infección y a síntomas de manera conjunta, teniendo en cuenta la dependencia intrínseca existente entre ambas variables.

En ese sentido, en los capítulos subsecuentes se propondrán dos tipos de modelos paramétricos bivariados para brindar estimaciones a las variables descritas: un modelo que emplea una distribución bivariada conjunta y un modelo de cópulas, los cuales posteriormente se aplicarán a un conjunto de datos reales para la evaluación de su comportamiento.

Así, para el primer modelo que se propondrá, de acuerdo a Li et al. (2012), las distribuciones más utilizadas para el análisis del tiempo a la ocurrencia del evento de interés son la exponencial y la Weibull. En el caso de la exponencial, se asume que la tasa de ocurrencia del evento de interés es constante, mientras que en el modelo Weibull se asume que la tasa

de ocurrencia de tal evento varía en el tiempo, ya sea incrementándose o reduciéndose. A partir de estas distribuciones se puede proponer un modelo de regresión lineal, donde el logaritmo del tiempo depende linealmente de las covariables y de un eror aleatorio que sigue una distribucion de valores extremos tipo I. Este modelo se puede extender al caso bivariado, asumiendo que los errores siguen una distribución conjunta de valores extremos bivariada.

Por otro lado, las cópulas son modelos que permiten aislar la estructura de dependencia en una distribución multivariada, de tal manera que esta última puede ser representada a través de las distribuciones marginales de cada variable aleatoria y la cópula (Haugh (2016)). En ese sentido, en el presente trabajo se utilizará el modelo de cópulas para establecer la dependencia entre el tiempo de infección y el tiempo de síntomas, y así, junto con las distribuciones marginales de cada tiempo, efectuar la estimación conjunta.

1.2. Objetivos

El objetivo general de la tesis es estudiar modelos de regresión paramétricos bivariados y aplicarlos a la estimación conjunta del tiempo de infección y el tiempo de desarrollo de síntomas de una enfermedad.

Los objetivos específicos de este proyecto de tesis son los siguientes:

1. Revisar literatura sobre modelos de regresión paramétricos bivariados y modelos de cópulas utilizados en el análisis de supervivencia.
2. Identificar y especificar modelos paramétricos bivariados y de cópulas aplicables a la estimación conjunta de tiempos de infección y síntomas.
3. Realizar estudios de simulación de los modelos identificados.
4. Comparar dichos modelos a fin de determinar la precisión de las estimaciones que arrojan.
5. Aplicar los modelos a a los datos de un estudio donde se evaluo el efecto de una intervención en el riesgo de reinfección de gonorrea o clamidia (Paulon et al. (2020)).

1.3. Organización del trabajo

El presente trabajo está constituido por cinco capítulos y un apéndice. El capítulo 1 presenta las definición del problema que se desea abordar y los objetivos de la tesis. En el capítulo 2 se presentan los conceptos y modelos básicos en los cuales se fundamenta la propuesta de solución brindada al problema. Asimismo, el capítulo 3 detalla el modelo propuesto para la estimación paramétrica de las distribuciones que rigen el comportamiento de los tiempo de infección y de aparición de síntomas. A continuación, el capítulo 4 muestra el estudio de simulación que permite verificar si la propuesta brinda soluciones acertadas a problemas teóricos de resultado conocido. Subsecuentemente, el capítulo 5 presenta la aplicación del modelo propuesto a un conjunto de datos reales y el análisis de los resultados obtenidos. Finalmente, en el capítulo 6 se discuten las conclusiones obtenidas a partir del análisis de los resultados, y se brindan recomendaciones para futuras investigaciones similares.

En el presente trabajo también se presentan dos apéndice: el apéndice A donde se puede encontrar el código fuente en lenguaje R implementado para la estimación con los modelos propuestos, la generación de simulaciones, y la aplicación de los modelos al conjunto de datos reales utilizado; mientras que en el apéndice B se puede encontrar las tablas con los valores numéricos resultantes de las simulaciones efectuadas de los modelos implementados.

Capítulo 2

Conceptos preliminares

En este capítulo se abordarán los conceptos fundamentales del análisis de supervivencia y de los principales modelos utilizados en esta disciplina.

El análisis de supervivencia es la rama de la estadística que estudia los tiempos a la ocurrencia de un evento de interés y los factores que podrían influir en ellos (Moore (2016)). En sus orígenes, esta disciplina estuvo inicialmente enfocada en el estudio de las tasas de mortalidad de los individuos y sus probabilidades de supervivencia, es decir, en el estudio del tiempo transcurrido hasta la ocurrencia de la muerte de los individuo, justamente de ahí es de donde deriva su nombre. Dichos orígenes datan del siglo XVII, cuando en 1662 John Graunt, estadístico inglés considerado el primer demógrafo, el fundador de la bioestadística y un precursor de la epidemiología, confeccionó la primera tabla de mortalidad (Gargantilla (2021)). Sin embargo, con el transcurrir del tiempo, los métodos del análisis de supervivencia se generalizaron al estudio del tiempo hasta la ocurrencia de un evento particular cualquiera, no exclusivamente la muerte, eventos como fallas de equipos, aparición de enfermedades y recuperación de estas, entre otros, lo que muestra que dicho análisis puede ser aplicado a diferentes ámbitos como ingeniería, economía, sociología, criminología, entre otros (Camilleri (2019)).

En ese sentido, a continuación se brindan algunos conceptos preliminares fundamentales para el estudio de cualquier modelo de supervivencia.

2.1. Datos de supervivencia y censura

Dado que el análisis de supervivencia estudia el tiempo hasta la ocurrencia de un evento, claramente puede verse que los datos con los que se trabaja son positivos, pues carece de sentido hablar de tiempos negativos. Dichos datos pueden ser discretos o continuos, y representan el tiempo transcurrido desde un origen preestablecido hasta hasta la ocurrencia de un evento bien definido (Moore (2016)).

Sin embargo, en los problemas de la vida real, el momento exacto de ocurrencia del evento usualmente es desconocido, por ejemplo, en el caso del tiempo de infección y del de síntomas, únicamente se tiene la información de la condición del paciente en los momentos en que es evaluado (*current status data*) durante el periodo de estudio, y se desconoce si el paciente desarrolló la enfermedad o los síntomas después de dicho periodo. A este hecho se le conoce como censura de datos, concepto que juega un papel crucial en casi todos los problemas de análisis de supervivencia.

Los dos ejemplos más conocidos de censura de datos son la censura a la derecha y la censura a la izquierda, los cuales se explican a continuación:

- **Censura a la derecha:** En la censura a la derecha únicamente se conoce que el evento de interés ocurrirá despés de un momento dado. Por ejemplo, en el caso del tiempo a infección, existe censura a la derecha cuando, en el momento de evaluación del paciente, resulta que no tiene enfermedad, pues en ese punto se sabe que, de adquirir la enfermedad, lo hará en un momento posterior al de la evaluación.
- **Censura a la izquierda:** En contraste al caso anterior, en la censura a la izquierda se conoce que el evento de interés ocurrió antes de un punto dado. Por ejemplo, nuevamente en el caso del tiempo a infección, existe censura a la izquierda cuando, al momento de efectuar la evaluación del paciente, resulta que sí tiene la enfermedad, entonces lo único que se conoce en ese punto es que adquirió la enfermedad antes del momento de la evaluación.

Asimismo, existen tres tipos de censura de datos (Moore (2016)):

1. **Censura tipo I:** En este tipo, los tiempos de censura se encuentran predeterminados, es decir, se fijan el momento de inicio y el de fin del experimento, y los individuos son observados durante ese periodo hasta la ocurrencia del evento, y para aquellos que no

experimentaron el evento, el tiempo se censura a la derecha en el momento del fin del experimento.

2. **Censura tipo II:** En este tipo, los individuos son observados hasta que el evento ocurra en una proporción preestablecida de los mismos. Este tipo de censura es muy utilizado cuando se estudia tiempos de falla de dispositivos, cuando el estudio se efectúa hasta que fallen el $x\,\%$ de los mismos, siendo el $(1-x)\,\%$ restante censurados a la derecha.

3. **Censura aleatoria:** En este tipo, la censura ocurre de manera independiente al investigador y sin ningún control de parte de este. Por ejemplo, se presenta cuando un individuo abandona el experimento, o también por eventos competitivos, como la muerte de un paciente por una causa distinta a la estudiada.

2.2. Funciones utilizadas en el estudio del tiempo a ocurrencia del evento de interés

Sea T la variable aleatoria no negativa que define el tiempo a la ocurrencia de un evento de interés, se definen las siguientes funciones que permiten estudiar su comportamiento:

1. **Función de supervivencia**

   La función de superivencia en un tiempo t esta definida como la probabilidad de que el evento no ocurra hasta el referido tiempo t, o similarmente, la probabilidad de que el evento de interés ocurra después del momento t. Formalmente, la función de supervivencia en el tiempo t se define de la siguiente forma:

$$S(t) = P(T > t),\ 0 < t < \infty \tag{2.1}$$

2. **Función de riesgo**

   La función de riesgo corresponde a la tasa de falla instantánea, es decir, el límite de la probabilidad de que el evento ocurra en el instante siguiente dado que no ha ocurrido hasta este instante. De manera específica::

$$h(t) = \lim_{\delta \to 0} \frac{P(t < T < t + \delta | T > t)}{\delta} \tag{2.2}$$

   A está función también se le conoce como función de intensidad o fuerza de mortalidad.

3. Función de distribución acumulada

La función de distribución acumulada representa la probabilidad de que el evento ocurra antes de un momento t dado, es decir:

$$F(t) = P(T \leq t),\ 0 < t < \infty \tag{2.3}$$

De lo anterior, puede observarse que la función de distribución acumulada es el complemento de la función de supervivencia:

$$F(t) = 1 - S(t),\ 0 < t < \infty \tag{2.4}$$

Asimismo, de la función de distribución acumulada puede obtenerse la función de densidad de probabilidad, que es la derivada de la acumulada:

$$f(t) = \frac{dF(t)}{dt} = -\frac{dS(t)}{dt} \tag{2.5}$$

Adicionalmente, se puede demostrar la siguiente relación entre la función de densidad de probabilidad, la de supervivencia y la de riesgo (Moore (2016)):

$$h(t) = \frac{f(t)}{S(t)} \tag{2.6}$$

Finalmente, la función de supervivencia puede expresarse también en función de la función de riesgo, de la siguiente manera:

$$S(t) = exp\left(-\int_0^t h(u)du\right) \tag{2.7}$$

A la integral que se encuentra dentro de la exponencial en la ecuación anterior se le denomina función de riesgo acumulada $H(t)$, por lo que:

$$H(t) = \int_0^t h(u)du \tag{2.8}$$

al combinar (2.7) y (2.8) se obtiene que:

$$S(t) = exp(-H(t)) \tag{2.9}$$

4. **Tiempo medio de supervivencia**

   El tiempo medio de supervivencia corresponde al valor esperado de la función de densidad:

$$\mu = E(T) = \int_0^\infty t f(t) dt \tag{2.10}$$

   Se puede demostrar que la expresión anterior equivale a la siguiente:

$$\mu = \int_0^\infty S(t) dt \tag{2.11}$$

   Así, el tiempo medio de supervivencia existe únicamente si $S(\infty) = 0$, es decir, que el evento ocurre eventualmente para todos los individuos

2.3. Distribuciones univariadas

En esta sección se describen las distribuciones probabílisticas básicas que se utilizan para el modelado en el análisis de supervivencia.

2.3.1. Distribución Exponencial

La distribución exponencial es la más simple utilizada en el análisis de supervivencia, y asume que la función de riesgo es constante (λ). Es decir:

$$h(t) = \lambda \tag{2.12}$$

Dicha función de riesgo constante da la propiedad de pérdidad de memoria a la distribución exponencial, que consiste en que el riesgo de ocurrencia del evento es el mismo en cualquier punto en el tiempo desde el inicio del periodo. Gráficamente, la función de riesgo se comporta como una línea recta en el valor λ (ver Figura 2.1):

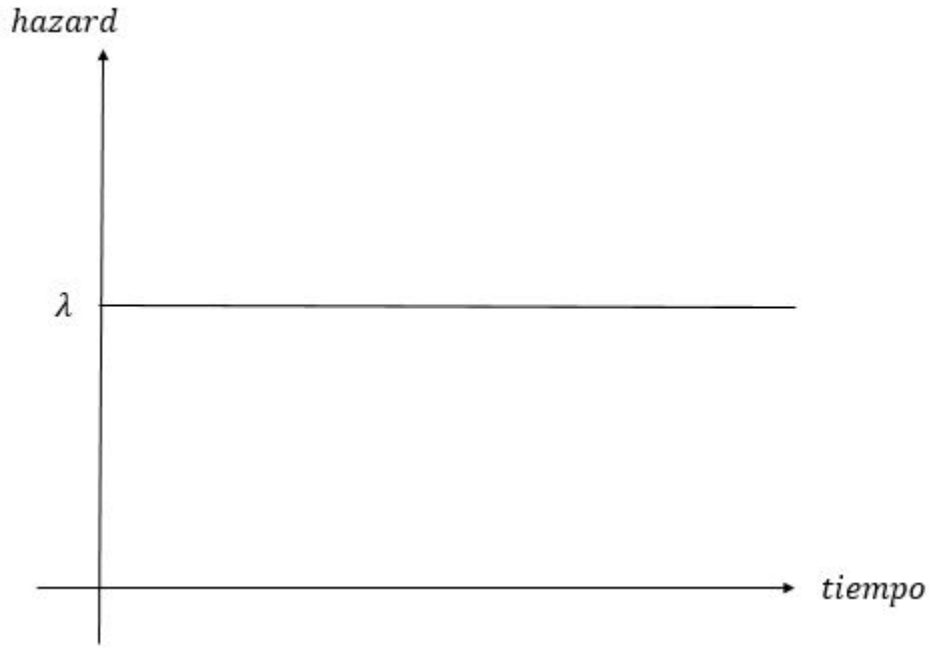


Figura 2.1: Comportamiento de la función de riesgo de la distribución exponencial.

Por otro lado, se obtiene la siguiente expresión para la función de riesgo acumulada:

$$H(t) = \int_0^t h(u)du = \int_0^t \lambda du = \lambda t \tag{2.13}$$

Similarmente, se puede obtener la expresión de la función de supervivencia:

$$S(t) = exp(-H(t)) = exp(-\lambda t) = e^{-\lambda t}$$

Asimismo, de la función de supervivencia pueden derivarse la función de distribución acumulada y la de densidad:

$$F(t) = 1 - S(t) = 1 - e^{-\lambda t}$$
$$f(t) = \frac{dF(t)}{dt} = -e^{-\lambda t}(-\lambda) = \lambda e^{-\lambda t}$$

Finalmente, la media de la distribución exponencial está dada por:

$$E(T) = \int_0^\infty S(t)dt = \int_0^\infty e^{-\lambda t}dt = \left[-\frac{e^{-\lambda t}}{\lambda}\right]_0^\infty = \frac{1}{\lambda}\left[-e^{-\lambda(\infty)} + e^{-\lambda(0)}\right] = \frac{1}{\lambda}$$

2.3.2. Distribución Weibull

En 1951 el ingeniero y matemático sueco Waloddi Weibull describió una distribución estadística de función de riesgo no constante que representó adecuadamente el esfuerzo al que se someten los materiales hasta la rotura, distribución que posteriormente tomó su nombre (Mejía Hernández (2009)). De manera específica, la función de riesgo instantaneo de la distribución Weibull es la siguiente:

$$h(t) = \alpha\lambda^{\alpha}t^{\alpha-1} \tag{2.14}$$

Nótese que la función de riesgo ya no es constante sino que depende del tiempo, y es ajustada por los parámetros α y λ. Así, dependiendo del valor de α, se tienen tres comportamientos distintos de la función de riesgo. El primero se da cuando $\alpha < 1$, entonces la función de riesgo es monótona decreciente, es decir, la tasa de ocurrencia del evente va disminuyendo a medida que pasa el tiempo. En segundo lugar, si $\alpha = 1$, entonces la función de riesgo es constante (simplemente igual a λ), resultando la función de riesgo de la distribución exponencial antes descrita. Por último, en caso $\alpha > 1$, entonces la función de riesgo es monótona creciente, es decir, la tasa de ocurrencia del evento se incrementa a medida que pasa el tiempo.

El siguiente gráfico ilustra el comportamiento de la función de riesgo para diferentes valores de α y un mismo valor de λ:

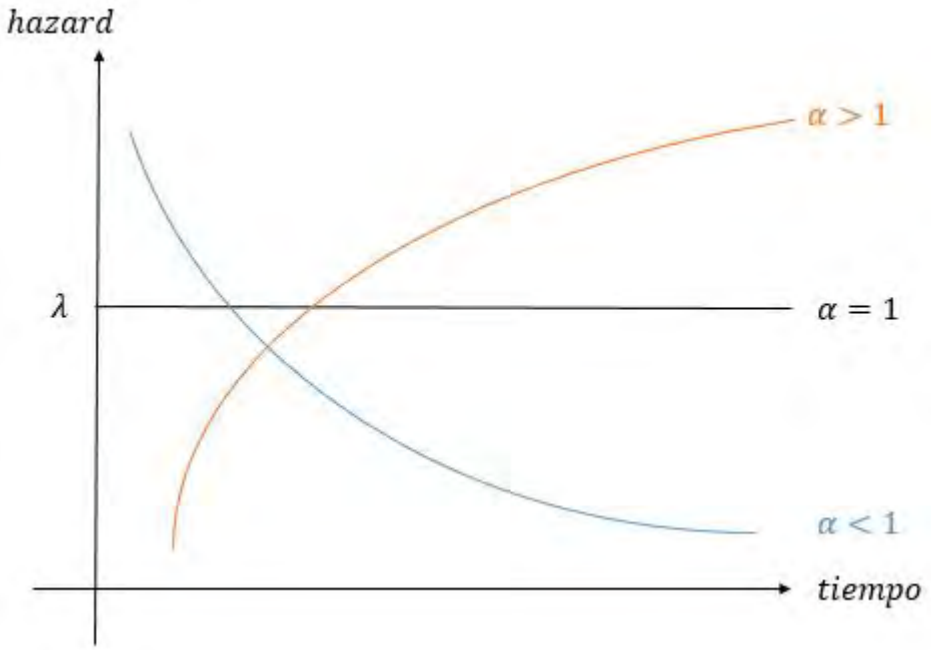


Figura 2.2: Comportamiento de la función de riesgo de la distribución Weibull.

Para este modelo también se puede determinar la función de riesgo acumulada:

$$H(t) = \int_0^t h(u)du = \int_0^t \alpha\lambda^\alpha u^{\alpha-1}du = \alpha\lambda^\alpha \int_0^t u^{\alpha-1}du = \alpha\lambda^\alpha \frac{t^\alpha}{\alpha} = (\lambda t)^\alpha \tag{2.15}$$

así como las funciones de supervivencia, de distribución acumulada y de densidad:

$$\begin{aligned}
S(t) &= exp(-H(t)) = exp(-(\lambda t)^\alpha) = e^{-(\lambda t)^\alpha} \\
F(t) &= 1 - S(t) = 1 - e^{-(\lambda t)^\alpha} \\
f(t) &= \frac{dF(t)}{dt} = -e^{-(\lambda t)^\alpha}(-\alpha(\lambda t)^{\alpha-1})(\lambda) = \lambda\alpha(\lambda t)^{\alpha-1}e^{-(\lambda t)^\alpha}
\end{aligned}$$

Finalmente, se puede demostrar que la media de la distribución Weibull es:

$$E(T) = \frac{\Gamma\left(1 + \frac{1}{\alpha}\right)}{\lambda} \tag{2.16}$$

donde Γ corresponde a la función gamma, la cual es igual a:

$$\Gamma(x) = \int_0^\infty u^{x-1}e^{-u}du \tag{2.17}$$

Se demuestra que, para un número entero positivo n:

$$\Gamma(n) = (n-1)! \tag{2.18}$$

Asimismo, otras distribuciones utilizadas en el análisis de supervivencia incluyen la distribución gamma (diferente a la función gamma), la log-normal, la log-logística, la Pareto, entre otras (Moore (2016)).

2.4. Modelos bivariados

En el punto anterior se describieron las distribuciones probabilísticas más comunes utilizadas en el análisis de supervivencia, las cuales son utilizadas para el modelado univariado, es decir, cuando se tiene una única variable respuesta. Sin embargo, existen situaciones donde se tiene más de una variable respuesta, como por ejemplo las series de eventos, situación en la que un mismo individuo puede experimentar una serie sucesiva de eventos, donde los tiempos de ocurrencia entre cada evento no son independientes, pues le ocurren a un mismo individuo,

como cuando se presentan varias ocurrencias de aparición de infección o de síntomas a un mismo paciente.

Otro caso en el que se tienen varias variables respuesta es en los tiempos de vida afines, caso en el cual los tiempos de ocurrencia al evento (o, específicamente, los tiempos de vida) están relacionados, como por ejemplo, el tiempo de vida de un esposo y su esposa, hermanos u otros parientes, los cuales están relacionados por características comunes no observadas (latentes) de la pareja o de la familia. Asimismo, también se tienen varias variables en el caso de riesgos competitivos, situación que aparece generalmente en el estudio del tiempo a la muerte por una enfermedad, puesto que existen a la vez otras causas que pueden generar el deceso del paciente, diferentes a la enfermedad estudiada. Un último caso que vale la pena mencionar es el de modelos de historia de eventos, el cual involucra transiciones entre distintos tipos de estados, por ejemplo el estado civil de una persona: inicialmente es soltero y puede pasar a conviviente, casado, separado, entre otros, en este caso, se combinan elementos de riesgos competitivos y series de eventos (Rodríguez (2015)).

En el caso particular del presente trabajo, el problema a estudiar cuenta con dos (2) variables respuesta: el tiempo a infección y el tiempo a síntomas, y se desea estudiar la distribución conjunta de ambas variables. Una primera forma de abordar dicho problema es el uso de distribuciones bivariadas, en las cuales se tienen dos (2) tiempos de supervivencia T_1 y T_2, los cuales presentan una función de supervivencia conjunta definida por:

$$S_{12}(t_1, t_2) = P(T_1 > t_1, T_2 > t_2) \tag{2.19}$$

De manera similar al caso univariado, dicha función corresponde a la probabilidad de que el tiempo de ocurrencia de un evento sea mayor a t_1 y de que el tiempo a la ocurrencia del otro evento sea mayor a t_2.

También se pueden definir la funciones de supervivencia marginales para T_1 y T_2:

$$S_1(t_1) = P(T_1 > t_1) = S_{12}(t_1, 0) \tag{2.20}$$

$$S_2(t_2) = P(T_2 > t_2) = S_{12}(0, t_2) \tag{2.21}$$

Si T_1 y T_2 fueran independientes, la función de supervivencia conjunta será el producto de

las marginales. Asimismo, se tiene también la función de supervivencia condicional, la cual presenta dos versiones:

$$S_{1|2}(t_1|T_2 = t_2) = P(T_1 > t_1|T_2 = t_2) \tag{2.22}$$

La cual brinda la probabilidad de supervivencia de T_1 sabiendo que el evento de interés de T_2 ocurrió en t_2, y:

$$S_{1|2}(t_1|T_2 > t_2) = P(T_1 > t_1|T_2 > t_2) \tag{2.23}$$

La cual brinda la probabilidad de supervivencia de T_1 sabiendo que el evento de interés de T_2 aún no ha ocurrido hasta el momento t_2.

Por otro lado, se puede definir la función de riesgo conjunta, la cual corresponde al riesgo instantáneo de que el evento de la primera unidad ocurra en t_1, y el de la segunda en t_2, sabiendo que los eventos no ocurrieron antes de t_1 y de t_2, es decir:

$$h_{12}(t_1, t_2) = \lim_{\delta \to 0} \frac{P(t_1 < T_1 < t_1 + \delta, t_2 < T_2 < t_2 + \delta | T_1 > t_1, T_2 > t_2)}{\delta^2} \tag{2.24}$$

La función de riesgo marginal será:

$$h_1(t_1) = \lim_{\delta \to 0} \frac{P(t_1 < T_1 < t_1 + \delta | T_1 > t_1)}{\delta} \tag{2.25}$$

Se puede demostrar también que, bajo independencia, la función de riesgo conjunta es la suma de las funciones de riesgo marginales.

Finalmente, también se pueden definir las funciones de riesgo condicionales, que también presentan dos versiones:

$$h_{1|2}(t_1|T_2 = t_2) = \lim_{\delta \to 0} \frac{P(t_1 < T_1 < t_1 + \delta | T_1 > t_1, T_2 = t_2)}{\delta} \tag{2.26}$$

Que muestra el riesgo de que el evento ocurra para la primera unidad, sabiendo que para la segunda unidad el evento ocurrió en t_2, y:

$$h_{1|2}(t_1|T_2 > t_2) = \lim_{\delta \to 0} \frac{P(t_1 < T_1 < t_1 + \delta | T_1 > t_1, T_2 > t_2)}{\delta} \tag{2.27}$$

Que muestra el riesgo de que el evento ocurra para la primera unidad, sabiendo que para

la segunda unidad el evento no ha ocurrido hasta el tiempo t_2.

De manera similar al caso univariado, a continuación se describen las distribuciones bivariadas más utilizadas en el análisis de supervivencia.

2.4.1. Distribución exponencial bivariada

La forma más conocida de la distribución exponencial bivariada es la propuesta por Marshall y Olkin (1967), para la cual se asumió que el evento puede ocurrir a dos individuos según tres procesos independientes de Poisson, los cuales se describen a continuación:

- Que el evento ocurra únicamente para el primer individuo en el tiempo t_1.
- Que el evento ocurra únicamente para el segundo individuo en el tiempo t_2.
- Que el evento ocurra para los dos individuos. En este caso, el tiempo en el cual ocurre el evento será el máximo entre t_1 y t_2.

Así, el tiempo de ocurrencia para cada proceso seguirá una distribución exponencial, y los parámetros para cada una de dichas distribuciones son λ_1, λ_2 y λ_{12}. Dado que los procesos son independientes entre sí, la función de supervivencia conjunta será simplemente el producto de las funciones de supervivencia de cada proceso (Nelsen (2006)), con lo que resulta lo siguiente:

$$S_{12}(t_1, t_2) = P(T_1 > t_1)P(T_2 > t_2)P(max(T_1, T_2) > max(t_1, t_2)) \tag{2.28}$$

Dado que las funciones de supervivencia marginales son exponenciales, reemplazándolas en la ecuación 2.35, resulta:

$$S(t_1, t_2) = exp(-\lambda_1 t_1 - \lambda_2 t_2 - \lambda_{12} max(t_1, t_2)) \tag{2.29}$$

que es la forma final de la función de supervivencia exponencial bivariada porpuesta por Marshall y Olkin (1967), donde: $t_1 > 0,\ t_2 > 0,\ \lambda_1 > 0,\ \lambda_2 > 0,\ \lambda_{12} \geq 0$.

La Figura 2.3 muestra la función de supervivencia conjunta para $\lambda_1 = 0.8$, $\lambda_2 = 0.5$ y $\lambda_{12} = 0.6$:

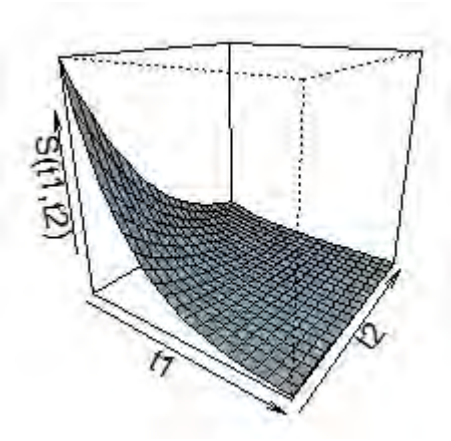


Figura 2.3: Función de supervivencia conjunta exponencial bivariada con $\lambda_1 = 0.8$, $\lambda_2 = 0.5$ y $\lambda_{12} = 0.6$.

Se observa que la función presenta un comportamiento descendente, similar al caso univariado, y una caída homogénea en los ejes t_1 y t_2, dada la asunción de funciones de riesgo constantes.

Por otro lado, si $\lambda_{12} = 0$, entonces T_1 y T_2 son independientes, cada una con distribución exponencial de parámetro λ_1 y λ_2, respectivamente (Mohr Bemis (1975)). En caso contrario, las funciones de supervivencia marginales de T_1 y T_2 son:

$$S(t_1) = S(t_1, 0) = exp(-\lambda_1 t_1 - \lambda_{12} t_1) = exp((-\lambda_1 - \lambda_{12})t_1) \tag{2.30}$$

$$S(t_2) = S(0, t_2) = exp(-\lambda_2 t_2 - \lambda_{12} t_2) = exp((-\lambda_2 - \lambda_{12})t_2) \tag{2.31}$$

Es decir, ambas también presentan como distribución marginal a la exponencial, pero cuyo parámetro incluye a λ_{12}.

2.4.2. Distribución Weibull bivariada

Similar al caso univariado, la distribución Weibull bivariada es una generalización de la exponencial bivariada. Una de las formas más conocidas de dicha distribución fue propuesta por Lu (1989), quien incluyó dos parámetros adicionales a la función de supervivencia bivariada exponencial de Marshall-Olkin:

$$S(t_1, t_2) = exp(-\lambda_1 t_1^{\beta_1} - \lambda_2 t_2^{\beta_2} - \lambda_{12} max(t_1^{\beta_1}, t_2^{\beta_2})) \tag{2.32}$$

La figura 2.4 muestra la función de supervivencia antes señalada, para los parámetros $\lambda_1 = 0.8$, $\lambda_2 = 0.5$, $\lambda_{12} = 0.6$, $\beta_1 = 3$ y $\beta_2 = 0.5$:

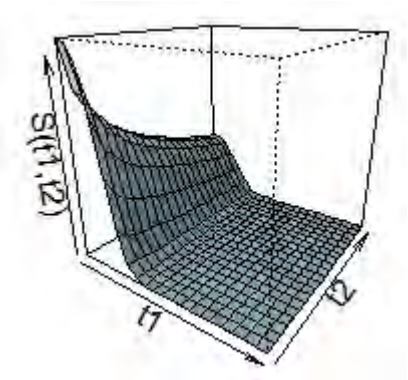


Figura 2.4: Función de supervivencia conjunta Weibull bivariada con $\lambda_1 = 0.8$, $\lambda_2 = 0.5$, $\lambda_{12} = 0.6$, $\beta_1 = 3$ y $\beta_2 = 0.5$.

Se observa que la distribución Weibull bivariada presenta el mismo comportamiento descendente que la exponencial bivariada, pero la caída no es homogénea, pues en el eje t_1 la pendiente del descenso es mucho mayor, lo que provoca una caída acelerada que hace que la función de supervivencia llegue rápidamente al valor cero en dicho eje. En contraste, la caída en el eje t_2 es mucho más lenta, por lo que presenta una pendiente mucho menor. Dicho comportamiento se debe a la inclusión de los parámetros β_1 y β_2 como exponentes de t_1 y t_2; así, cuando dicho exponente es mayor a uno, acelera la caída de la función de supervivencia (como el caso de t_1), mientras que si es menor que uno, la ralentiza (como en el caso de t_2) .

Asimismo, el caso particular donde $\beta_1 = \beta_2$ fue denominado por Kundu y Dey (2009) como la función de supervivencia Weibull bivariada de Marshall-Olkin.

2.5. Estimación

En general, los modelos de regresión permiten estimar las relaciones entre una variable respuesta y una o más variables independientes (predictoras o covariables) a partir de una muestra conocida. Estadísticamente, la especificación de un modelo de regresión requiere el establecimiento de un componente sistemático y un componente aleatorio. El componente sistemático involucra una estimación de las relaciones entre el "valor promedio" de la variable respuesta y las covariables, mientras que el componente aleatorio especifica la distribución estadística de la parte remanenente que queda luego de aplicar el componente sistemático (Hosmer et al. (2008)).

Sin embargo, en el caso del análisis de supervivencia, existen dos características particulares que se deben tomar en cuenta al momento de implementar un modelo de regresión: el inherente proceso de envejecimiento que sufren los individuos observados con el paso del tiempo, y la presencia de datos censurados. En ese sentido, la función que mejor captura el proceso de envejecimiento es la función de riesgo, o sea, la probabilidad de ocurrencia instantánea del evento, por lo que en el análisis de supervivencia se procura incorporar la función de riesgo en la estructura del modelo de regresión, tomando en cuenta la censura de los datos (Hosmer et al. (2008)).

Así, los modelos de regresión del análisis de supervivencia pueden ser no paramétricos, donde no se asume una distribución probabilística de la función de supervivencia (como el modelo de Cox o de riesgos proporcionales); y paramétricos donde se establece que la función de supervivencia sigue una distribución determinada (como los modelos de tiempo de falla acelarado). El presente trabajo se centrará en los modelos paramétricos, al ser los que se utilizarán para la propuesta de solución del problema.

2.5.1. Estimación por máxima verosimilitud en el análisis de supervivencia

Como se mencionó previamente, en un modelo de regresión se busca estimar las relaciones entre la variable respuesta y covariables a partir de una muestra conocida. En el caso de un modelo de regresión paramétrico, se busca estimar tanto los parámetros del componente sistemático, es decir, los coeficientes que multiplican linealmente a cada covariable (o a una función de la covariable), como los del componente aleatorio, que son los parámetros de la distribución asumida para el mismo.

Así, el método más utilizado para estimar los parámetros antes descritos es el de máxima verosimilitud, en el cual la estimación se obtiene al maximizar la función de verosimilitud ($L(\theta, t_1, t_2, ..., t_n)$), la cual corresponde a la función de densidad conjunta de todas las observaciones de la muestra dada, es decir:

$$L(\theta, t_1, t_2, ..., t_n) = f(\theta, t_1, t_2, ..., t_n) \tag{2.33}$$

Donde θ es el vector de parámetros a estimar, t_i el tiempo de ocurrencia del evento en la observación i, y n el número de observaciones existentes en la muestra.

Para estimar la función de verosimilitud se asume independencia de las observaciones, en consecuencia, dicha función es simplemente igual al producto de las funciones de densidad de cada observación:

$$L(\theta, t_1, t_2, ..., t_n) = \prod_{i=1}^{n} f(\theta, t_i) \tag{2.34}$$

Cabe mencionar que la función de densidad es la misma para todas las observaciones, y se obtiene a partir de la distribución asumida para el componente aleatoio.

Sin embargo, cabe recordar que el análisis de supervivencia tiene la particularidad de contar con datos censurados, es decir, datos en los cuales no se observó la ocurrencia del evento en el periodo de estudio. En ese sentido, la fórmula anterior (2.34) requiere ser ajustada a fin de incorporar la censura de los datos. Así, para las observaciones en las cuales se observó la ocurrencia del evento, se utiliza directamente la función de densidad asumida, mientras que para las observaciones censuradas, es decir, aquellas en las que no se observó la ocurrencia del evento en el periodo de estudio, se utiliza la función de supervivencia, pues lo único que se conoce es que el evento ocurrirá después de un tiempo determinado, el cual puede ser el tiempo de fin del periodo de estudio o el tiempo en que la observación se retiró del estudio (Moore (2016)):

En ese sentido, la función de verosimilitud queda de la siguiente manera:

$$L(\theta, t_1, t_2, ..., t_n) = \prod_{i=1}^{n} f(\theta, t_i)^{\delta_i} S(\theta, t_i)^{1-\delta_i} \tag{2.35}$$

Donde $\delta_i = 1$ cuando el evento ocurrió en el periodo de estudio, mientras que $\delta_i = 0$ cuando la observación está censurada, es decir, no se observó la ocurrencia del evento. Asimismo, la ecuación 2.46 puede expresarse de las siguiente manera:

$$\begin{aligned} L(\theta, t_1, t_2, ..., t_n) &= \prod_{i=1}^{n} f(\theta, t_i)^{\delta_i} \frac{S(\theta, t_i)}{S(\theta, t_i)^{\delta_i}} \\ &= \prod_{i=1}^{n} \frac{f(\theta, t_i)^{\delta_i}}{S(\theta, t_i)^{\delta_i}} S(\theta, t_i) \\ &= \prod_{i=1}^{n} \left(\frac{f(\theta, t_i)}{S(\theta, t_i)} \right)^{\delta_i} S(\theta, t_i) \end{aligned} \tag{2.36}$$

Pero de la ecuación 2.6, el primer factor corresponde a la función de riesgo, entonces reemplazando:

$$L(\theta, t_1, t_2, ..., t_n) = \prod_{i=1}^{n} h(\theta, t_i)^{\delta_i} S(\theta, t_i) \tag{2.37}$$

Asimismo, de la expresión anterior se puede obtener la función de log-verosimilitud, la cual equivale al logaritmo de la función de verosimilitud:

$$\begin{aligned} l(\theta, t_1, t_2, ..., t_n) &= \log(L(\theta, t_1, t_2, ..., t_n)) \\ &= \log\left(\prod_{i=1}^{n} h(\theta, t_i)^{\delta_i} S(\theta, t_i)\right) \\ &= \sum_{i=1}^{n} \delta_i \log(h(\theta, t_i)) + \log(S(\theta, t_i)) \end{aligned} \tag{2.38}$$

Así por ejemplo, se puede aplicar la expresión anterior a una distribución exponencial para estimar el parámetro λ. Para ello, cabe recordar que su función de riesgo es constante y justamente es igual al valor de λ ($h(t) = \lambda$), mientras que su función de supervivencia es $S(t) = e^{-\lambda t}$. En ese sentido, reemplazando dichas expresiones en la ecuación 2.49, resulta:

$$\begin{aligned} l(\theta, t_1, t_2, ..., t_n) &= \sum_{i=1}^{n} \delta_i \log(h(\theta, t_i)) + \log(S(\theta, t_i)) \\ &= \sum_{i=1}^{n} \delta_i \log(\lambda) + \log(e^{-\lambda t_i}) \\ &= \sum_{i=1}^{n} (\delta_i \log(\lambda) - \lambda t_i) \\ &= \log(\lambda) \sum_{i=1}^{n} \delta_i - \lambda \sum_{i=1}^{n} t_i \end{aligned} \tag{2.39}$$

Como se señaló previamente, la estimación del parámetro será el valor que maximiza la función de verosimilitud, o equivalentemente, donde se maximiza la función de log-verosimilitud. Dicho valor corresponde al punto en el cual la primera derivada de dicha función, respecto del parámetro, es igual a cero. Así, nuevamente para el caso de la distribución exponencial, la primera derivada de la función de log-verosimilitud se presenta a continuación:

$$\begin{aligned} l'(\theta, t_1, t_2, ..., t_n) &= \frac{\partial l(\theta, t_1, t_2, ..., t_n)}{\partial \lambda} \\ &= \frac{1}{\lambda} \sum_{i=1}^{n} \delta_i - \sum_{i=1}^{n} t_i \end{aligned} \tag{2.40}$$

Igualando dicha derivada a cero, se obtiene lo siguiente:

$$\lambda = \frac{\sum_{i=1}^{n} \delta_i}{\sum_{i=1}^{n} t_i} \tag{2.41}$$

Es decir, la estimación de máxima verosimilitud de λ es igual al número de ocurrencias del evento entre la suma de los tiempos de ocurrencia del evento y de censura.

2.5.2. Modelo de regresión Weibull

Como se mencionó en la sección 2.3.2, la distribución Weibull tiene como función de riesgo a $h(t) = \alpha\lambda^{\alpha} t^{\alpha-1}$ y como función de supervivencia a $S(t) = e^{-(\lambda t)^{\alpha}}$. Sin embargo, a fin de especificar el modelo de regresión Weibull, conviene expresar dichas funciones en base de los parámetros escala $\sigma = 1/\alpha$ y media $\mu = -\log \lambda$, con lo cual quedan de la siguiente manera (Moore (2016)):

$$h(t) = \frac{1}{\sigma}(e^{-\mu})^{\frac{1}{\sigma}} t^{\frac{1}{\sigma}-1} = \frac{1}{\sigma} e^{-\frac{\mu}{\sigma}} t^{\frac{1}{\sigma}-1} \tag{2.42}$$

$$S(t) = e^{-(e^{-\mu}t)^{\frac{1}{\sigma}}} = e^{-e^{-\frac{\mu}{\sigma}} t^{\frac{1}{\sigma}}} \tag{2.43}$$

Cabe resaltar que si $\sigma = 1$, entonces $\alpha = 1$, la distribución Weibull se reduce a una distribución exponencial.

Asimismo, si se aplica la transformación $g(u) = \log(-\log(u))$ a la función de supervivencia, se obtiene lo siguiente:

$$\begin{aligned} &\log(-\log(S(t))) = \log(-\log(e^{-e^{-\frac{\mu}{\sigma}} t^{\frac{1}{\sigma}}})) = \log(-(-e^{-\frac{\mu}{\sigma}} t^{\frac{1}{\sigma}})) = \log(e^{-\frac{\mu}{\sigma}} t^{\frac{1}{\sigma}}) = \log(e^{-\frac{\mu}{\sigma}}) + \log(t^{\frac{1}{\sigma}}) \\ &\log(-\log(S(t))) = -\frac{\mu}{\sigma} + \frac{1}{\sigma}\log(t) \end{aligned} \tag{2.44}$$

Cabe resaltar que la expresión anterior asemeja a una ecuación de una recta de $\log(-\log(S(t)))$ versus $\log(t)$, con pendiente igual a $1/\sigma$ e intercepto $-\mu/\sigma$, por lo que una forma de evaluar si

una muestra $t = (t_1, t_2, ..., t_n)$ sigue una distribución Weibull, es estimar la función de supervivencia de forma no parámetrica, efectuar un gráfico de dispersión entre $\log(-\log(\hat{S(t_i)}))$ y $\log(t_i)$, y evaluar si dicho gráfico es similar a una recta. De ser así, se puede asumir que dicha muestra obedece a una distribución Weibull, e incluso la pendiente y el intercepto de dicha recta brindan las estimaciones de $1/\sigma$ y $-\mu/\sigma$, las cuales permiten finalmente determinar las estimaciones de λ y α. Cabe mencionar que uno de los métodos más utilizados para estimar la función de supervivencia de forma no paramétrica es a través del estimador de Kaplan y Meier (Kaplan y Meier (1958)), en el cual la función de supervivencia se estima en cada punto en el cual ha ocurrido el evento de interés en una muestra dada. Así, si se se denominan $t_1 < t_2 < t_3 < ... < t_k$ a los tiempos de ocurrencia del evento, $t_0 = 0$ y $t_{k+1} = +\infty$, entonces para cada intervalo $[t_j, t_{j+1})$ se definen:

- d_j=Número de ocurrencias del evento en el tiempo t_j.
- m_j=Número de censuras ocurridas en el intervalo $[t_j, t_{j+1})$.
- n_j=Número de total de individuos que quedan en el estudio en el instante previo a t_j. Es decir, es el número de personas que, al instanto previo a t_j, no les ha ocurrido el evento de interés ni la censura. Asimismo, se asume que n es el número total de individuos en el estudio.

Con la información anterior se puede determinar la función de supervivencia en cada punto de ocurrencia del evento de la siguiente manera:

- Para $t \in [t_0, t_1)$, no se observa la ocurrencia del evento, por lo que la función de supervivencia en dicho intervalo será igual a uno, es decir, $S(t) = 1, t \in [t_0, t_1)$.
- Para $t \in [t_1, t_2)$, han ocurrido d_1 eventos y m_1 censuras, por lo que $n_1 = n - m_0$. En ese sentido, la función de supervivencia será la proporción de individuos a los que no les ocurrió el evento de interés luego del momento t_1 y antes de t_2. Es decir $S(t) = 1 - d_1/n_1, t \in [t_1, t_2)$.
- Para $t \in [t_2, t_3)$, han ocurrido d_2 eventos y m_2 censuras, por lo que $n_2 = n_1 - m_1 - d_1$. En ese sentido, la función de supervivencia será la proporción de individuos a los que no les ocurrió el evento de interés luego del momento t_2 y antes de t_3, pero considerando las censuras y ocurrencias en el intervalo anterior ($[t_1, t_2)$), por lo que $S(t) = (1 - d_1/n_1)(1 - d_2/n_2), t \in [t_1, t_2)$.

- Aplicando la misma lógica para $t \in [t_3, t_4)$, se obtiene que $S(t) = (1 - d_1/n_1)(1 - d_2/n_2)(1 - d_3/n_3) = \prod_{j=1}^{3}(1 - d_j/n_j), t \in [t_3, t_4)$.
- Así, en general para cualquier intervalo ($[t_i, t_{i+1})$), la función de supervivencia se determina de la siguiente manera:

$$S(t) = \prod_{j=1}^{i}\left(1 - \frac{d_j}{n_j}\right), t \in [t_i, t_{i+1})$$

Si bien el método descrito previamente es una forma sencilla de estimar los parámetros del modelo de regresión Weibull, una forma más eficiente de realizarlo es a través del método de máxima verosimilitud. Para ello es necesario trabajar con la función de log-verosimilitud establecida en la ecuación 2.45:

$$l(\theta, t_1, t_2, ..., t_n) = \sum_{i=1}^{n}[\delta_i \log(h(\theta, t_i)) + \log(S(\theta, t_i))]$$

Reemplazando las expresiones de la función de riesgo y de supervivencia de una distribución Weibull se obtiene lo siguiente:

$$\begin{aligned} l(\lambda, \alpha, t_1, t_2, ..., t_n) &= \sum_{i=1}^{n}[\delta_i \log(\alpha\lambda^{\alpha}t_i^{\alpha-1}) + \log(e^{-(\lambda t_i)^{\alpha}})] \\ &= \sum_{i=1}^{n}\delta_i(\log(\alpha) + \log(\lambda^{\alpha}) + \log(t_i^{\alpha-1})) + \sum_{i=1}^{n}\log(e^{-(\lambda t_i)^{\alpha}}) \\ &= \sum_{i=1}^{n}\delta_i(\log(\alpha) + \alpha\log(\lambda) + (\alpha-1)\log(t_i)) - \sum_{i=1}^{n}(\lambda t_i)^{\alpha} \\ &= \log(\alpha)\sum_{i=1}^{n}\delta_i + \alpha\log(\lambda)\sum_{i=1}^{n}\delta_i + (\alpha-1)\sum_{i=1}^{n}\delta_i\log(t_i) - \lambda^{\alpha}\sum_{i=1}^{n}t_i^{\alpha} \end{aligned} \tag{2.45}$$

Expresión que puede optimizarse mediante métodos numéricos para la obtención de las estimaciones de λ y α.

2.5.3. Modelo de regresión de tiempo de falla acelerado

El modelo de tiempo de falla acelerado permite incorporar covariables al modelo de regresión, y estimar el efecto de cada una de ellas en el tiempo de ocurrencia del evento. Así, si se denomina $S_0(t)$ a la función de supervivencia que no presenta el efecto de las covariables, el modelo de tiempo de falla acelerado asume que existe una constante ψ tal que la función de supervivencia que incluye el efecto de las covariables, denominado $S_1(t)$, cumple la siguiente

igualdad (Cox y Oakes (1984)):

$$S_1(t) = S_0(\psi t) \tag{2.46}$$

Es decir, la probabilidad de que el evento no ocurra hasta el tiempo t, para una observación sometida al efecto de las covariables, es la misma que la probabilidad de que el evento, para una observación no afectada por las covariables, no ocurra hasta el tiempo ψt. Por ejemplo, en caso el evento sea la muerte de los individuos, si $\psi = 0.5$, entonces la probabilidad de que un individuo del grupo afecto a las covariables permanezca vivo hasta los 40 años (t), es la misma que la probabilidad de que un individuo del grupo de referencia permanezca vivo hasta los 20 años ($0.5t$) (Rodríguez (2021)).

La interpretación señalada en el párrafo anterior puede hacerse extensiva incluso al tiempo de supervivencia (de no ocurrencia del evento). Así, se puede afirmar que, si el tiempo de supervivencia de una observación sujeta a las covariables (T_1) es igual a t, el tiempo de supervivencia de una observación no sujeta a covariables (T_0) será igual a t/ψ (Cox y Oakes (1984)), es decir:

$$T_1 = \frac{T_0}{\psi} = \psi^{-1} T_0 \tag{2.47}$$

Así, en el contexto del ejemplo del evento de muerte antes mencionado, donde $\psi = 0.5$, el tiempo de supervivencia de un individuo sujeto a covariables de 20 años, equivale a un tiempo de supervivencia de 40 años de un individuo no sujeto a covariables.

De la expresión anterior, puede obtenerse la siguiente igualdad para la función de densidad:

$$\begin{aligned} S_1(t) &= S_0(\psi t) \\ 1 - F_1(t) &= 1 - F_0(\psi t) \\ F_1(t) &= F_0(\psi t) \\ \frac{\partial F_1(t)}{\partial t} &= \frac{\partial F_0(\psi t)}{\partial t} \\ f_1(t) &= \psi f_0(\psi t) \end{aligned} \tag{2.48}$$

Y similarmente para la función de riesgo:

$$
\begin{aligned}
f_1(t) &= \psi f_0(\psi t) \\
\frac{f_1(t)}{S_1(t)} &= \psi \frac{f_0(\psi t)}{S_1(t)}
\end{aligned} \tag{2.49}
$$

Reemplazando las ecuaciones 2.6 y 2.46 en la expresión anterior, se obtiene:

$$
\begin{aligned}
\frac{f_1(t)}{S_1(t)} &= \psi \frac{f_0(\psi t)}{S_0(\psi t)} \\
h_1(t) &= \psi h_0(\psi t)
\end{aligned} \tag{2.50}
$$

Así, las covariables ingresan al modelo de regresión justamente a través del término ψ, en ese sentido, dicho término corresponde a una función de las covariables $X = (X_1, X_2, ..., X_p)^T$ y de los coeficientes de regresión $\beta = (\beta_1, \beta_2, ..., \beta_p)^T$, es decir, $\psi = \psi(X, \beta)$. Dicha función debe cumplir las siguientes propiedades:

- Como debe existir $S_0(\psi(X, \beta)t)$, entonces $\psi(X, \beta)t > 0$, como se sabe que $t > 0$, entonces también se debe cumplir que $\psi(X, \beta) > 0$.
- Aplicando la ecuación 2.53 en ausencia de covariables ($\psi(0)$), se debe cumplir que $S_0(t) = S_0(\psi(0)t)$, por lo que $\psi(0) = 1$.

Entonces, dadas las propiedadas $\psi(X, \beta) > 0$ y $\psi(0) = 1$, un candidato natural a ser utilizado como función de $\psi(X, \beta)$ es la función exponencial, (Cox y Oakes (1984)) por lo que el modelo asume que:

$$
\psi(X, \beta) = e^{-X^T \beta} \tag{2.51}
$$

Con lo que la ecuación 2.54 queda de la siguiente manera:

$$
T_1 = e^{X^T \beta} T_0 \tag{2.52}
$$

Tomando el logaritmo a ambos lados de la ecuación 2.63, se obtiene:

$$
\log(T_1) = X^T \beta + \log(T_0) \tag{2.53}
$$

Ecuación que corresponde a un modelo lineal convencional, siendo $\log(T_0)$ el error aleatorio del modelo, por lo que la ecuación anterior puede reescribirse de la siguiente manera

(Rodríguez (2021)):

$$\log(T_1) = X^T\beta + \epsilon \tag{2.54}$$

En ese sentido, los modelos de tiempo de falla acelerado son aquellos en los que el tiempo a la ocurrencia del evento puede linealizarse tomando el logaritmo de dicho tiempo. El término "acelerado" se debe a que el efecto de las covariables es multiplicativo en la escala del tiempo (ecuación 2.59), es decir, las covariables "aceleran" el tiempo de ocurrencia del evento. Esta característica diferencia a este modelo del de riesgos proporcionales, en los cuales el efecto de las covariables es multiplicativo sobre la función de riesgo (Hosmer et al. (2008)).

Así, se pueden obtener diferentes modelos paramétricos dependiendo de la distribución asumida para T_0, y por ende, para $\epsilon = \log(T_0)$. Por ejemplo, si se asume que ϵ está normalmente distribuido, entonces T_0 seguirá una distribución log-normal.

En particular, si T_0 sigue una distribución Weibull, entonces $h_0(t) = \frac{1}{\sigma}e^{-\frac{\mu}{\sigma}}t^{\frac{1}{\sigma}-1}$ (ecuación 2.49) y $h_1(t)$ será igual a:

$$\begin{aligned}
h_1(t) &= e^{-X^T\beta}\frac{1}{\sigma}e^{-\frac{\mu}{\sigma}}(e^{-X^T\beta}t)^{\frac{1}{\sigma}-1} \\
&= e^{-X^T\beta}\frac{1}{\sigma}e^{-\frac{\mu}{\sigma}}e^{-X^T\beta(\frac{1}{\sigma}-1)}t^{\frac{1}{\sigma}-1} \\
&= e^{-X^T\beta}\frac{1}{\sigma}e^{-\frac{\mu}{\sigma}}e^{-\frac{X^T\beta}{\sigma}+X^T\beta}t^{\frac{1}{\sigma}-1} \\
&= e^{-X^T\beta}\frac{1}{\sigma}e^{-\frac{\mu}{\sigma}}e^{-\frac{X^T\beta}{\sigma}}e^{X^T\beta}t^{\frac{1}{\sigma}-1} \\
&= \frac{1}{\sigma}e^{-\frac{\mu}{\sigma}}e^{-\frac{X^T\beta}{\sigma}}t^{\frac{1}{\sigma}-1} \\
&= e^{-\frac{X^T\beta}{\sigma}}\frac{1}{\sigma}e^{-\frac{\mu}{\sigma}}t^{\frac{1}{\sigma}-1} \\
&= e^{-\frac{X^T\beta}{\sigma}}h_0(t)
\end{aligned} \tag{2.55}$$

La ecuación 2.62 demuestra que, en el caso de la distribución Weibull, el modelo de tiempo de falla acelerado es equivalente al modelo de riesgos proporcionales, puesto que el factor también es multiplicativo en escala de la función de riesgo. Se puede demostrar que la distribución Weibull es la única que cumple la propiedad anterior (Moore (2016)).

2.5.4. Modelo de tiempo de falla acelerado Weibull

Para la implementación del modelo de regresión Weibull a través del modelo de tiempo de falla acelerado, se asume que el tiempo de ocurrencia del evento sin el efecto de las covariables (T_0) sigue una distribución Weibull con parámetros de escala $\sigma = 1/\alpha$ y media $\mu = -\log(\lambda)$.

Asimismo, como se procura incorporar un modelo de regresión lineal, se utiliza la ecuación 2.61, que establece la relación lineal entre las covariables y $\log(T_1)$, o, en términos de un modelo lineal, $\log(T_i)$:

$$\log(T_i) = X_i^T \beta + \epsilon_i$$

Reordenando la ecuación anterior se tiene:

$$\epsilon_i = \log(T_i) - X_i^T \beta \tag{2.56}$$

Cabe recordar que $\epsilon_i = \log(T_0)$, en ese sentido, dado que se ha asumido una distribución Weibull para T_0, la distribución de ϵ_i será una log-weibull, mejor conocida como la distribución Gumbel o de valor extremo tipo I (Liu y Lim (2018)), cuya función de densidad general, para $Z \sim Gumbel(\mu, \sigma)$ es:

$$f(z) = \frac{1}{\sigma} exp\left(\frac{z-\mu}{\sigma}\right) exp\left(-e^{\frac{z-\mu}{\sigma}}\right) \tag{2.57}$$

Así, el modelo de regresión de tiempo de falla acelerado Weibull asume que los errores siguen una distribución Gumbel de paramétro $\mu = 0$ y σ equivalente al paramétro de escala de la distribución Weibull de T_0 ($\epsilon \sim Gumbel(0, \sigma)$) (Hosmer et al. (2008)), con lo que la distribución anterior queda de la siguiente manera:

$$f_\epsilon(z) = \frac{1}{\sigma} exp\left(\frac{z}{\sigma}\right) exp\left(-e^{\frac{z}{\sigma}}\right) \tag{2.58}$$

Donde Z es la variable aleatoria, en este caso $Z = \epsilon_i = \log(T_i) - X^T\beta$, por lo que, reemplazando en la ecuación anterior, se obtiene la siguiente expresión:

$$f_\epsilon(t_i) = \frac{1}{\sigma} exp\left(\frac{\log(t_i) - X^T\beta}{\sigma}\right) exp\left(-e^{\frac{\log(t_i) - X^T\beta}{\sigma}}\right) \tag{2.59}$$

Cabe mencionar que la expresión anterior es equivalente a decir que $\epsilon \sim Gumbel(0, 1)$ con:

$$\epsilon = \frac{\log(t_i) - X^T\beta}{\sigma} \tag{2.60}$$

De donde se obtiene que:

$$\log(t_i) = X^T\beta + \sigma\epsilon \tag{2.61}$$

la cual es la expresión general del modelo de tiempo de falla acelerado.

Asimismo, si se reemplaza $y_i = \log(t_i)$ en la ecuación 2.66, se obtiene:

$$f_\epsilon(y_i) = \frac{1}{\sigma} exp\left(\frac{y_i - X^T\beta}{\sigma}\right) exp\left(-e^{\frac{y_i - X^T\beta}{\sigma}}\right) \tag{2.62}$$

Con la función de densidad anterior se puede obtener la función de distribución acumulada:

$$F_\epsilon(y_i) = 1 - e^{-e^{\frac{y_i - X^T\beta}{\sigma}}} \tag{2.63}$$

Y la correspondiente función de supervivencia:

$$S_\epsilon(y_i) = e^{-e^{\frac{y_i - X^T\beta}{\sigma}}} \tag{2.64}$$

Así, con las expresiones obtenidas anteriormente, se puede obtener la función de verosimilitud del modelo de tiempo de falla acelerado Weibull, reemplazando las ecuaciones 2.69 y 2.71 en la ecuación 2.42 (la forma general de la función de verosimilitud en el análisis de supervivencia):

$$L(\theta, t_1, t_2, ..., t_n) = \prod_{i=1}^{n} f(\theta, t_i)^{\delta_i} S(\theta, t_i)^{1-\delta_i}$$

En este caso $\theta = (\beta, \sigma)$ y la función de verosimilitud estará expresada en función de $y_i = \log(t_i)$ (Liu y Lim (2018)):

$$\begin{aligned} L(\beta, \sigma, y) &= \prod_{i=1}^{n} f(\beta, \sigma, y_i)^{\delta_i} S(\beta, \sigma, y_i)^{1-\delta_i} \\ &= \prod_{i=1}^{n} \left(\frac{1}{\sigma} e^{\frac{y_i - X^T\beta}{\sigma}} e^{-e^{\frac{y_i - X^T\beta}{\sigma}}}\right)^{\delta_i} \left(e^{-e^{\frac{y_i - X^T\beta}{\sigma}}}\right)^{1-\delta_i} \end{aligned} \tag{2.65}$$

Así, las estimaciones de los parámetros (β, σ) serán los valores que maximicen la expresión

antes mostrada, lo cual se logra mediante métodos numéricos, ya sea aplicados directamente a la función de verosimilitud, o a la log-verosimilitud.

2.5.5. Modelo de tiempo de falla acelerado bivariado

Si bien el modelo de regresión con covariables descrito en la sección anterior resulta aplicable al caso univariado, para el caso bivariado la lógica a seguir es similar, pues la expresión general del modelo de tiempo de falla acelerado puede extenderse a dos variables correlacionadas T_1 y T_2 (Hanagal (2006)):

$$\begin{pmatrix} \log(T_1) \\ \log(T_2) \end{pmatrix} = \begin{pmatrix} X_1^T \beta_1 \\ X_2^T \beta_2 \end{pmatrix} + \begin{pmatrix} \sigma_1 \epsilon_1 \\ \sigma_2 \epsilon_2 \end{pmatrix} \tag{2.66}$$

De donde, despejando los errores, resulta la siguiente expresión:

$$\begin{pmatrix} \epsilon_1 \\ \epsilon_2 \end{pmatrix} = \begin{pmatrix} \frac{\log(T_1) - X_1^T \beta_1}{\sigma_1} \\ \frac{\log(T_2) - X_2^T \beta_2}{\sigma_2} \end{pmatrix} \tag{2.67}$$

Así por ejemplo, se puede asumir que los errores (ϵ_1, ϵ_2) siguen una distribución de valores extremos bivariada (Hanagal (2006)), la cual se incorporará en la función de verosimilitud utilizada para la estimación de los parámetros. Cabe mencionar que en el caso bivariado la función de verosimilitud es diferente a la establecida en la ecuación 2.42, pues dicha función depende de la naturaleza del problema que se desea resolver.

Asimismo, se suele tomar como supuesto adicional que el efecto de las covariables es el mismo para todo el vector aleatorio, es decir, $X_1 = X_2 = X$ y $\beta_1 = \beta_2 = \beta$ (Hanagal (2006)), por lo que la ecuación 2.74 queda finalmente de la siguiente manera:

$$\begin{pmatrix} \epsilon_1 \\ \epsilon_2 \end{pmatrix} = \begin{pmatrix} \frac{\log(T_1) - X^T \beta}{\sigma_1} \\ \frac{\log(T_2) - X^T \beta}{\sigma_2} \end{pmatrix} \tag{2.68}$$

2.6. Cópulas

Una segunda forma de abordar el problema de la estimación de la distribución conjunta de dos variables aleatorias consiste en el uso de cópulas. El concepto de cópula fue introducido en 1959 por el matemático estadounidense Abe Sklar (Sklar (1959)), mientras que una descripción detallada y moderna se puede encontrar en Durante y Sempi (2010) y Nelsen (2006), dichas funciones permiten expresar una distribución multivariada a través de las dis-

tribuciones marginales individuales de cada variable aleatoria y la función de cópula.

A fin de poder definir una cópula, sean X y Y variables aleatorias con funciones de distribución $F(x) = P(X \leq x)$ y $G(y) = P(Y \leq y)$ respectivamente, y sea H la función de distribución conjunta de X y Y, es decir $H(x, y) = P(X \leq x, Y \leq y)$. En ese sentido, cada par de valores (x, y) que pueden tomar las variables aleatorias tienen asociados los valores de sus distribuciones marginales ($F(x)$ y $G(y)$) así como el valor de la distribución acumulada en ese punto ($H(x, y)$), asimismo los valores de $F(x)$, $G(y)$ y $H(x, y)$ pertenecen al intervalo $[0, 1]$. En otras palabras, cada par de valoress (x, y) conduce a otro par ordenado $(F(x), G(y))$, formado por sus correspondientes distribuciones marginales, este último ubicado en el cuadrado unitario $[0, 1] \times [0, 1]$, y adicionalmente el mencionado par ordenado (x, y) a su vez también está asociado al valor de la distribución conjunta $H(x, y)$, el cual pertenece al intervalo $[0, 1]$. En ese sentido, la cópula es la función que permite asociar el valor de la función de distribución conjunta a cada par ordenado de valores de las funciones de distribución marginal de cada variable ((Nelsen (2006))); es decir, las cópulas son las funciones que unen o "acoplan" la función de distribución multivariada conjunta con las funciones de distribución marginal univariadas. La misma definición anterior puede expandirse al caso multivariado.

Para obtener la referida función de cópula, se parte de la función de distribución acumulada de una variable univariada $F(X)$. Así, si dicha función se denota mediante la variable U, de la siguiente manera:

$$U = F(X) \tag{2.69}$$

Entonces, a su vez U es una variable aleatoria que sigue una distribución uniforme en el intervalo $[0, 1]$, es decir:

$$U \sim U_{[0,1]} \tag{2.70}$$

Por otro lado, si se tienen n variables aleatorias X_i, que generan a su vez n variables aleatorias U_i, entonces la distribución conjunta de estas últimas constituye la función de cópula:

$$C(u_1, u_2, ...u_n; \Sigma_\rho) \tag{2.71}$$

donde Σ_ρ representa la matriz de correlación de las variables aleatorias individuales, es

decir, la estructura de dependencia entre las mismas.

En ese sentido, dadas las variables aleatorias $X_1, X_2, ..., X_n$, que generan a su vez las variables aleatorias $U_1, U_2, ..., U_n$, tal que $U_i = F(X_i), i = 1, 2, ..., n$, o recíprocamente $F_i^{-1}(U_i) = X_i, i = 1, 2, ..., n$ entonces se puede obtener lo siguiente:

$$\begin{aligned} C(u_1, u_2, ..., u_n) &= P(U_1 \leq u_1, U_2 \leq u_2, ..., U_n \leq u_n) \\ &= P(F_1(X_1) \leq u_1, F_2(X_2) \leq u_2, ..., F_n(X_n \leq u_n) \\ &= P(X_1 \leq F_1^{-1}(u_1), X_2 \leq F_2^{-1}(u_2), ..., X_n \leq F_n^{-1}(u_n)) \\ &= P(X_1 \leq x_1, X_1 \leq x_2, ..., X_n \leq x_n) \\ C(u_1, u_2, ..., u_n) &= F(x_1, x_2, ..., x_n) \end{aligned} \tag{2.72}$$

Nótese que $F(x_1, x_2, ..., x_n)$ es la distribución acumulada conjunta del vector aleatorio $(X_1, X_2, ..., X_n)$, es decir, la función de cópula arroja los mismos resultados que la distribución acumulada conjunta.

Por otro lado, el teorema de Sklar establece que **cualquier distribución conjunta multivariada $\boldsymbol{F(x_1, x_2, ..., x_n)}$ puede ser expresada a través de una función de cópula C aplicada a las distribuciones marginales $\boldsymbol{F_i(x_i)}$ de cada variable perteneciente al vector aleatorio** (Nelsen (2006)), es decir:

$$F(x_1, x_2, ..., x_n) = C(F_1(x_1), F_2(x_2), ..., F_n(x_n)) \tag{2.73}$$

Asimismo, si las distribuciones marginales F_i son continuas, la función de cópula es única (Nelsen (2006)). Cabe mencionar que el teorema funciona de manera inversa a la ecuación 2.79, en la cual se partió de la cópula y se llegó a la función de distribución conjunta.

En sí, el teorema de Sklar demuestra que se puede obtener y confiar en una función de cópula para modelar la estructura de dependencia de las variables aleatorias que forman parte de un vector aleatorio.

Un caso especial de la función de cópula es la producto: $C(u_1, u_2, ..., u_n) = \prod_{i=1}^{n} u_i$, la cual es aplicable a variables aleatorias independientes, pues:

$$C(u_1, u_2, ..., u_n) = \prod_{i=1}^{n} u_i = \prod_{i=1}^{n} F_i(x_i) \tag{2.74}$$

Es decir, la cópula producto corresponde a la multiplicación de las distribuciones marginales F_i, pero se sabe que, en variables independientes, dicha multiplicación equivale a la distribución conjunta, en ese sentido la ecuación 2.81 equivale a:

$$C(u_1, u_2, ..., u_n) = F(x_1, x_2, ..., x_n) \tag{2.75}$$

Resultado consistente con la ecuación 2.72.

Asimismo, se mencionó previamente que $F_i(x_i) = u_i$, por lo que $x_i = F_i^{-1}(u_i)$, entonces reemplazando x_i en la parte derecha de la ecuación 2.42, se obtiene:

$$C(u_1, u_2, ..., u_n) = F(F_1^{-1}(u_1), F_2^{-1}(u_2), ..., F_n^{-1}(u_n)) \tag{2.76}$$

La ecuación 2.76 demuestra que se pueden construir cópulas a partir de las funciones de distribución conjunta, si esta se conoce (Nelsen (2006)).

Este resultado permite construir distintas familias de cópulas, algunas de las propuestas más conocidas son::

- **Arquimediana:** Las copulas arquimedianas son aquellas que siguen la forma:

$$C(u_1, u_2) = \varphi^{-1}(\varphi(u_1) + \varphi(u_2)) \tag{2.77}$$

  Donde φ es una función continua, estrictamente decreciente y $\varphi(1) = 0$, y es denominada función generadora de C (Ziener (2021)).

  Una de las familias de cópulas arquimedianas más conocidas es la propuesta por Ali et al. (1978), quienes buscaron distribuciones bivariadas $H(x, y)$ cuyas marginales $F(x)$ y $G(y)$ cumplan lo siguiente:

$$\frac{1 - H(x, y)}{H(x, y)} = \frac{1 - F(x)}{F(x)} + \frac{1 - G(y)}{G(y)} + (1 - \theta)\frac{1 - F(x)}{F(x)}\frac{1 - G(y)}{G(y)} \tag{2.78}$$

  Para $\theta \in [-1, 1]$.

Reemplazando en la ecuación anterior las variables $u = F(x)$, $v = G(y)$ y $H(x, y) = C(u, v)$, se obtiene:

$$\frac{1 - C(u,v)}{C(u,v)} = \frac{1-u}{u} + \frac{1-v}{v} + (1-\theta)\frac{1-u}{u}\frac{1-v}{v} \tag{2.79}$$

Desarrollando la expresión anterior, resulta la forma general de la familia de cópulas arquimedianas Ali-Mikhail-Haq:

$$C(u,v) = \frac{uv}{1 - \theta(1-u)(1-v)} \tag{2.80}$$

donde θ corresponde al parámetro de la cópula.

A fin de demostrar que cópula anterior tiene la forma de una cópula arquimediana, si se multiplica cada lado de la ecuación 2.85 por $(1 - \theta)$ y luego se suma 1, la ecuación anterior queda de la siguiente manera:

$$\begin{aligned} 1 + (1-\theta)\frac{1 - C(u,v)}{C(u,v)} &= 1 + (1-\theta)\frac{1-u}{u} + (1-\theta)\frac{1-v}{v} + (1-\theta)^2\frac{1-u}{u}\frac{1-v}{v} \\ 1 + (1-\theta)\frac{1 - C(u,v)}{C(u,v)} &= \left(1 + (1-\theta)\frac{1-u}{u}\right)\left(1 + (1-\theta)\frac{1-v}{v}\right) \end{aligned} \tag{2.81}$$

Si se define:

$$\lambda(t) = 1 + (1-\theta)\frac{1-t}{t} \tag{2.82}$$

Reemplazando λ en la ecuación 2.86, se obtiene:

$$\lambda(C(u,v)) = \lambda(u)\lambda(v) \tag{2.83}$$

Tomando el logaritmo:

$$\begin{aligned} \log(\lambda(C(u,v))) &= \log(\lambda(u)\lambda(v)) \\ \log(\lambda(C(u,v))) &= \log(\lambda(u)) + \log(\lambda(v)) \\ -\log(\lambda(C(u,v))) &= -\log(\lambda(u)) - \log(\lambda(v)) \end{aligned} \tag{2.84}$$

Si se define $\varphi(t) = -\log(\lambda(t))$, función continua, estrictamente decreciente y $\varphi(1) = 0$, entonces reemplazando en la ecuación anterior:

$$\begin{aligned} \varphi(C(u,v)) &= \varphi(u) + \varphi(v) \\ C(u,v) &= \varphi^{-1}(\varphi(u) + \varphi(v)) \end{aligned} \tag{2.85}$$

Expresión que cumple con la forma general de las cópulas arquimedianas establecida en la ecuación 2.84.

Asimismo, a continuación se presentan algunas de las cópulas arquimedianas más conocidas:

Cópula	**Expresión**	**Función generadora**	$\theta \in$
Clayton	$C(u,v) = max(u^{-\theta} + v^{-\theta} - 1, 0)^{-1/\theta}$	$\varphi(t) = \frac{1}{\theta}(t^{-\theta} - 1)$	$[-1, \infty] - \{0\}$
Frank	$C(u,v) = -\frac{1}{\theta}\log\left(1 + \frac{(e^{-\theta u}-1)(e^{-\theta v}-1)}{e^{-\theta}-1}\right)$	$\varphi(t) = -\log\left(\frac{e^{-\theta t}-1}{e^{-\theta}-1}\right)$	$(-\infty, \infty) - \{0\}$
Gumbel	$C(u,v) = exp(-[(-\log u)^{\theta} + (-\log v)^{\theta}]^{1/\theta})$	$\varphi(t) = (-\log t)^{\theta}$	$[1, \infty)$

Cuadro 2.1: Cópulas arquimedianas mas conocidas.

- **Elíptica:** Las cópulas elípticas son las generadas por distribuciones elípticas. Las distribuciones elípticas son una generalización de la distribución normal multivariada, en las cuales su función de densidad tiene la siguiente forma:

$$f(X) = kg((X-\mu)^T\Sigma^{-1}(X-\mu)) \tag{2.86}$$

donde k es una constante normalizadora, X un vector aleatorio n-dimensional, μ el vector de medias, Σ la matriz de covarianzas, y g una función aplicada sobre $(X - \mu)^T\Sigma^{-1}(X - \mu)$.

Las cópulas elípticas más usuales son

- **Cópula Gaussiana:** Esta cópula proviene de una distribución normal multivariada de vector de medias 0 y matriz de correlación Σ. En ese sentido, aplicando la ecuación 2.83 a este caso, se obtiene la forma de la cópula gaussiana:

$$C(u_1, u_2, ..., u_n) = \Phi_{\Sigma}(\Phi^{-1}(u_1), \Phi^{-1}(u_2), ..., \Phi^{-1}(u_n)) \tag{2.87}$$

donde Φ_Σ representa a la distribución normal multivariada antes descrita, y Φ^{-1} a la inversa de la distribución normal estándar univariada (Haugh (2016)).

- **Cópula t de student:** Esta cópula se define de manera similar a la gaussiana pero utilizando una distribución t de student con ν grados de libertad y matriz de covarianzas Σ en lugar de la normal multivariada, por lo que, nuevamente aplicando la ecuación 2.83, se obtiene lo siguiente:

$$C(u_1, u_2, ..., u_n) = t_{\nu,\Sigma}(t_\nu^{-1}(u_1), t_\nu^{-1}(u_2), ..., t_\nu^{-1}(u_n)) \tag{2.88}$$

donde $t_{\nu,\Sigma}$ corresponde a la distribución t de student multivariada y t_ν^{-1} la inversa de la distribución t de student univariada (Haugh (2016)).

Capítulo 3

Propuesta de modelo

3.1. Planteamiento del problema

Como se mencionó previamente, en el presente trabajo se propondrá un modelo paramétrico bivariado para la estimación conjunta del tiempo de infección y del tiempo de aparición de síntomas, considerando la dependencia existente entre ambas variables. En tal sentido, los tiempos antes mencionados pueden representarse mediante las siguientes variables aleatorias:

I = Tiempo a infección

S = Tiempo a síntomas

Cada una con su correspondiente distribución marginal:

$$I \sim F_I$$

$$S \sim F_S$$

Sin embargo, como se mencionó en el capítulo previo, en la vida real es muy difícil conocer con precisión dichos tiempos, en su lugar lo que normalmente se conoce es la condición del paciente en el momento en que es evaluado, es decir, los datos están censurados. Así, cuando se evalúa al paciente es posible conocer si presenta la infección o no, y si presenta los síntomas o no, a este tipo particular de información se le conoce como data de estado actual o *current status data* (Paulon et al. (2020)).

Así, para especificar la data de estado actual, se pueden definir las siguientes variables:

$$\Delta_I = \begin{cases} 1, \text{si la persona tiene la infección} \\ 0, \text{si la persona no tiene la infección} \end{cases} \tag{3.1}$$

$$\Delta_S = \begin{cases} 1, \text{si la persona presenta síntomas} \\ 0, \text{si la persona no presenta síntomas} \end{cases} \tag{3.2}$$

Asimismo, dado que la presencia o no de infección o de síntomas se evalúa y registra en un momento dado, si se define la variable aleatoria T como el tiempo en el cual se efectúa la evaluación del paciente, entonces, para un valor particular $T = t$, las variables I y S se pueden especificar de la siguiente manera:

- **Presencia de infección:** Si efectuada la evaluación en el momento t la persona presenta la infección, quiere decir que la ha adquirido en un tiempo anterior a t, en ese sentido, el tiempo a la infección será menor que t. Por el contrario, si la persona no presenta la infección, en caso la adquiera, lo hará en un momento posterior a t, por lo que el tiempo a infección será mayor que dicho momento. Lo anterior se puede expresar de la siguiente manera:

$$\Delta_I = \begin{cases} 1, I \leq t \\ 0, I > t \end{cases} \tag{3.3}$$

- **Presencia de síntomas:** De manera similar al tiempo de infección, si efectuada la evaluación en el momento t la persona presenta síntomas, esto quiere decir que los mismos han aparecido en un tiempo anterior a t, por tanto, el tiempo a los síntomas será menor que t. En caso contrario, si la persona no presenta síntomas, estos podrían aparecer en un momento posterior a t, por lo que el tiempo a los síntomas será mayor que dicho momento. Lo anterior se puede expresar de la siguiente manera:

$$\Delta_S = \begin{cases} 1, S \leq t \\ 0, S > t \end{cases} \tag{3.4}$$

En ese sentido, efectuada la evaluación de infección y síntomas en un momento t, cuatro posibles escenarios pueden ocurrir:

1. Que el paciente presente la infección y los síntomas, es decir: $\Delta_I = 1 \wedge \Delta_S = 1$, o lo que es equivalente a: $I \leq t \wedge S \leq t$.

2. Que el paciente no presente la infección, pero sí los síntomas, es decir: $\Delta_I = 0 \wedge \Delta_S = 1$, o lo que es equivalente a: $I > t \wedge S \leq t$.

3. Que el paciente presente la infección, pero no los síntomas, es decir: $\Delta_I = 1 \wedge \Delta_S = 0$, o lo que es equivalente a: $I \leq t \wedge S > t$.

4. Que el paciente no presente la infección, ni los síntomas, es decir: $\Delta_I = 0 \wedge \Delta_S = 0$, o lo que es equivalente a: $I > t \wedge S > t$.

El siguiente gráfico ilustra los cuatro escenarios antes mencionados. Cabe recordar que el momento t es aquel en el cual se efectúa la evaluación del paciente:

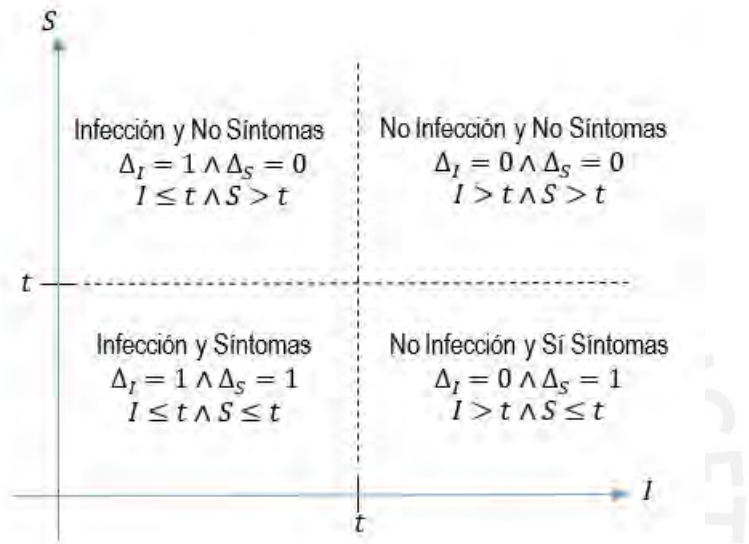


Figura 3.1: Escenarios posibles de los resultados del tiempo a infección y síntomas evaluados en un momento t.

En ese sentido, es posible identificar las probabilidades asociadas a cada escenario, de la siguiente manera:

1. **Infección y síntomas:** En este escenario $\Delta_I = 1$ y $\Delta_S = 1$, es decir, por lo señalado previamente: $I \leq t$ y $S \leq t$. En ese sentido, a este escenario le corresponde la probabilidad de que el tiempo de infección sea menor o igual a t y que el tiempo de síntomas también lo sea, la cual es igual a la función de distribución conjunta del tiempo de infección y del tiempo de síntomas evaluadas en el punto (t, t):

$$P(I \leq t, S \leq t) = F_{IS}(t, t) \tag{3.5}$$

En ese sentido, dado que este escenario corresponde al volumen de la función de densidad conjunta de (I, S) sobre el área inferior izquierda de la figura 3.1, entonces dicha

área representa el soporte de la distribución conjunta de los tiempos de infección y de síntomas:

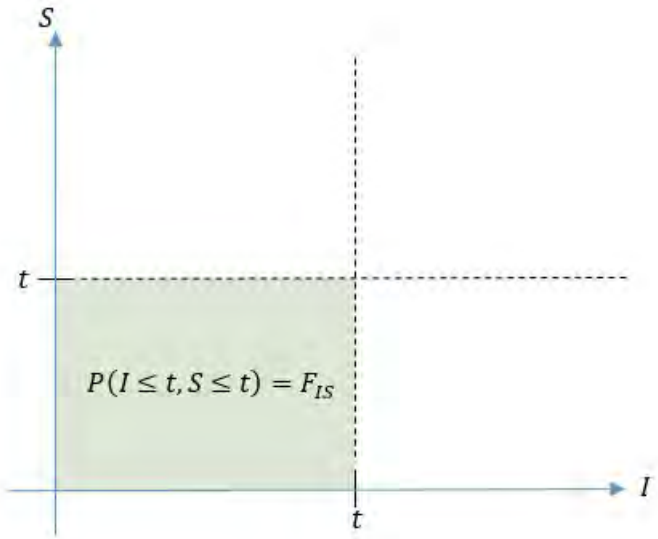


Figura 3.2: Área que representa el soporte de la probabilidad de que el paciente presente infección y síntomas en el momento t.

2. **No infección pero presencia de síntomas:** En este escenario $\Delta_I = 0$ y $\Delta_S = 1$, es decir, por lo señalado previamente: $I > t$ y $S \leq t$. En ese sentido, a este escenario le corresponde a la probabilidad de que el tiempo de infección sea mayor a t y que el tiempo de síntomas sea menor o igual a t ($P(I > t, S \leq t)$). Dicha probabilidad se determina partiendo de la distribución marginal del tiempo a síntomas ($P(S \leq t) = F_S$), la cual corresponde al volumen de la densidad marginal de S sobre el área sombreada en el gráfico de cuadrantes 3.3 siguiente (el cual es similar a los presentados previamente):

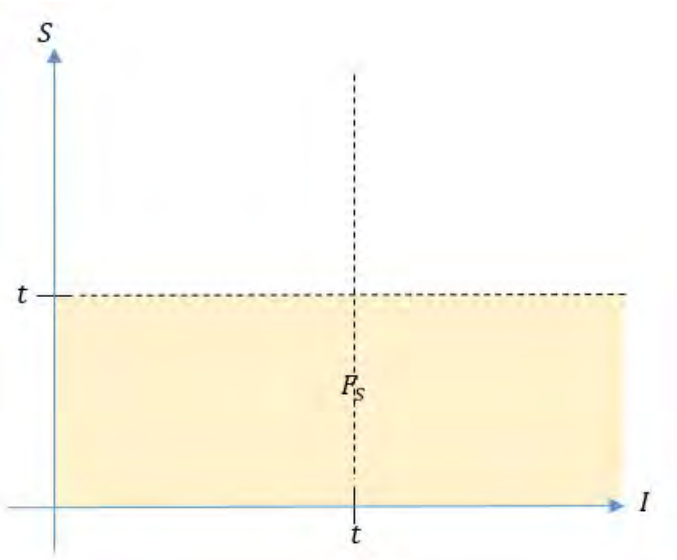


Figura 3.3: Soporte de la distribución marginal del tiempo a síntomas (F_S).

Se observa que el soporte de la distribución marginal del tiempo a síntomas corresponde a toda el área inferior del gráfico, pero como el área inferior izquierda corresponde a la distribución conjunta F_{IS}, se puede concluir que la probabilidad de que el tiempo de infección sea mayor a t y de que el tiempo de síntomas sea menor o igual a t, es igual a la marginal de S menos la conjunta de I y S, es decir:

$$P(I > t, S \leq t) = F_S - F_{IS} \tag{3.6}$$

Gráficamente se tiene lo siguiente:

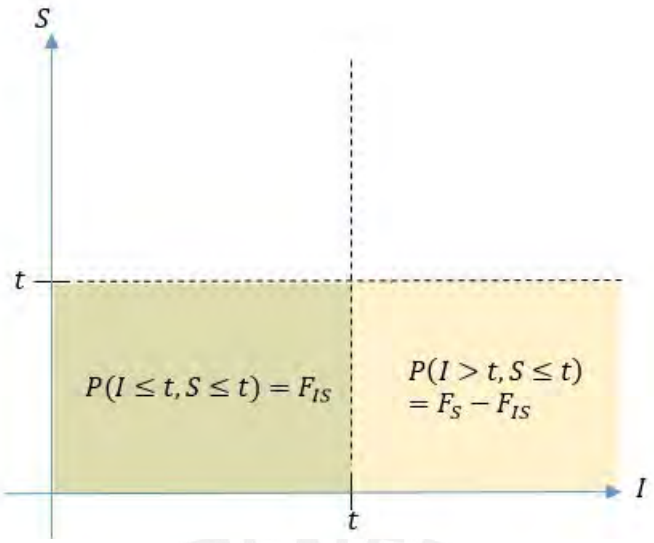


Figura 3.4: Áreas que representan los soportes de la probabilidad de que el paciente presente infección y síntomas (en marrón), y de la probabilidad que no presente infección pero sí síntomas (en naranja) en el momento t.

3. **Infección pero sin presencia de síntomas:** En este escenario $\Delta_I = 1$ y $\Delta_S = 0$, es decir, por lo señalado previamente: $I \leq t$ y $S > t$, lo cual corresponde la probabilidad de que el tiempo de infección sea menor o igual a t y que el tiempo de síntomas sea mayor a t ($P(I \leq t, S > t)$). Similarmente al caso anterior, dicha probabilidad se determina partiendo de la distribución marginal, pero en este caso del tiempo a infección ($P(I \leq t) = F_I$), la cual corresponde al volumen de la densidad marginal de I sobre el área sombreada en el gráfico de cuadrantes 3.5 que se presenta a continuación:

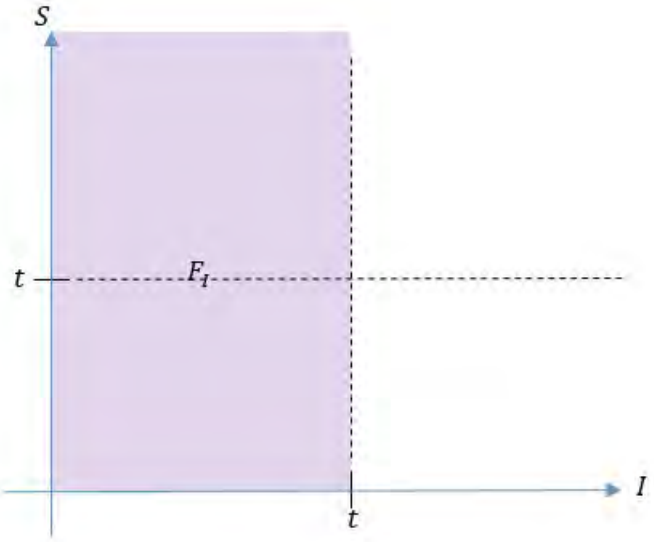


Figura 3.5: Soporte de la distribución marginal del tiempo a infección (F_I).

La distribución marginal del tiempo a infección corresponde a toda el área izquierda del gráfico, pero nuevamente el área inferior izquierda corresponde al soporte de la distribución conjunta F_{IS}, por lo que la probabilidad de que el tiempo de infección sea menor o igual a t y de que el tiempo de síntomas sea mayor a t, es igual a la marginal de I menos la conjunta de I y S, es decir:

$$P(I > t, S \leq t) = F_I - F_{IS} \tag{3.7}$$

Gráficamente se tiene lo siguiente:

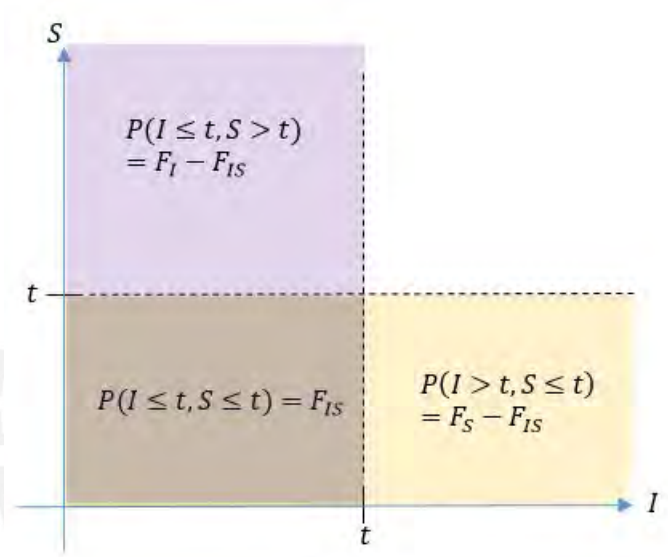


Figura 3.6: Áreas que representan los soportes de la probabilidad de que el paciente presente infección y síntomas (intersección de morado y naranja), que no presente infección pero sí síntomas (en naranja), y que presente infección sin síntomas (en morado) en el momento t.

4. **No infección y no síntomas:** En este escenario $\Delta_I = 0$ y $\Delta_S = 0$, es decir, por lo señalado previamente: $I > t$ y $S > t$, lo cual corresponde la probabilidad de que el tiempo de infección sea mayor a t y que el tiempo de síntomas también sea mayor a t ($P(I > t, S > t)$). Esto, a su vez, corresponde volumen de la función de distribución sobre al área superior derecha del gráfico. Para determinar esta probabilidad, se parte de la probabilidad $P(I > t)$, o equivalentemente, de $1 - P(I \leq t) = 1 - F_I$, probabilidad que corresponde al volumen sobre el área sombreada que se presenta en el gráfico 3.7:

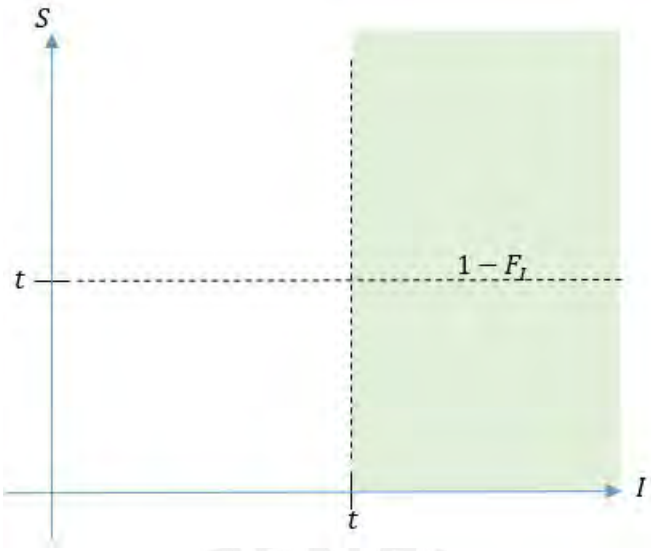


Figura 3.7: Área correspondiente al soporte de $1 - F_I$.

El área correspondiente al soporte de $1 - F_I$ corresponde al área derecha del gráfico, pero como solo interesa el área superior derecha, es suficiente restarle el área inferior derecha, ya calculada previamente. En ese sentido, la probabilidad será:

$$P(I > t, S > t) = 1 - F_I - (F_S - F_{IS}) = 1 - F_I - F_S + F_{IS} \tag{3.8}$$

Con lo que el gráfico final queda de la siguiente manera:

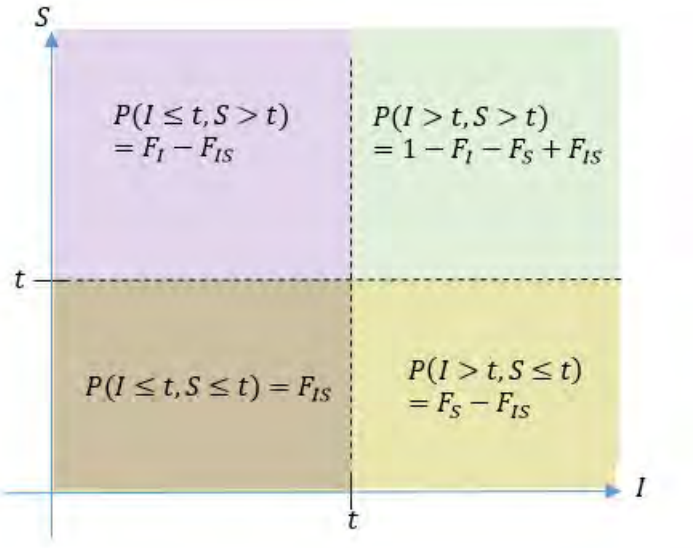


Figura 3.8: Áreas que representan los soportes de todas las probabilidades posibles del tiempo a infección y síntomas, detectados en el momento t.

Una vez especificadas las probabilidades asociadas al tiempo de infección y síntomas, se puede determinar la función de verosimilitud del modelo, el cual, asumiendo independencia en las observaciones, simplemente será la productoria de las probabilidades, dependiendo de los valores obtenidos de Δ_I y Δ_S, los cuales, al tomar únicamente valores de 0 o 1, ingresan a la verosimilitud como exponentes de las mencionadas probabilidades, así:

$$L(\theta) = \prod_{i=1}^{n} F_{IS}^{\Delta_I \Delta_S} (F_S - F_{IS})^{(1-\Delta_I)\Delta_S} (F_I - F_{IS})^{\Delta_I (1-\Delta_S)} (1 - F_I - F_S + F_{IS})^{(1-\Delta_I)(1-\Delta_S)} \quad (3.9)$$

donde n corresponde al número de observaciones, i a cada observación, y θ al vector de parámetros a estimar.

Asimismo, también se puede determinar la función de log-verosimilitud del modelo:

$$\begin{aligned} l(\theta) &= \log(L(\theta)) \\ &= \log\left(\prod_{i=1}^{n} F_{IS}^{\Delta_I \Delta_S} (F_S - F_{IS})^{(1-\Delta_I)\Delta_S} (F_I - F_{IS})^{\Delta_I (1-\Delta_S)} (1 - F_I - F_S + F_{IS})^{(1-\Delta_I)(1-\Delta_S)} \right) \\ &= \sum_{i=1}^{n} (\Delta_I \Delta_S \log(F_{IS}) + (1-\Delta_I)\Delta_S \log(F_S - F_{IS}) + \Delta_I (1-\Delta_S) \log(F_I - F_{IS}) + \\ &\qquad (1-\Delta_I)(1-\Delta_S) \log(1 - F_I - F_S + F_{IS})) \end{aligned} \quad (3.10)$$

3.2. Estimación de parámetros

Una vez obtenidas las funciones de verosimilitud y log-verosimilitud, la estimación de los parámetros se obtiene maximizando la función de log-verosimilitud. Sin embargo, para lograr lo anterior se requiere de la distribución acumulada conjunta de los tiempos de infección y síntomas, y de las distribuciones acumuladas marginales de cada uno de dichos tiempos. En ese sentido, dado que el objetivo del trabajo es proponer modelos paramétricos para la estimación de los tiempos de infección y síntomas, se asume que dichos tiempos siguen distribuciones de probabilidad conocidas, tanto para la conjunta como para las marginales. Por tanto, el problema se abordará mediante el modelo de tiempo de falla acelerado bivariado de dos maneras:

1. Asumiendo una distribución bivariada conjunta de los errores, a partir de la cual se obtienen las marginales.

2. Asumiendo las distribuciones marginales de cada tiempo y una cópula para estimar la distribución conjunta.

3.2.1. Estimación de parámetros asumiendo una distribución bivariada conjunta de los errores

En este caso, se asume que la distribución bivariada conjunta de los errores es la forma logística de la distribución bivariada de valores extremos (Gumbel (1960)), cuya función de distribución acumulada es la siguiente:

$$F(t_1, t_2) = exp\left[-\left(y_1(t_1)^{1/r} + y_2(t_2)^{1/r}\right)^r\right] \tag{3.11}$$

donde r es el parámetro de dependencia y $y_i(t_i)$ corresponde a las distribuciones marginales de t_i. En ese sentido, la forma logística de la distribución de valores extremos bivariada es una combinación de distribuciones de valores extremos univariadas, con el uso de un parámetro de dependencia r. Cabe mencionar que las distribuciones marginales $y_i(t_i)$ se definen de la siguiente manera:

$$y_i(t_i) = \left(1 + s_i \frac{t_i - a_i}{b_i}\right)^{-1/s_i} \tag{3.12}$$

expresión que corresponde a la distribución de valores extremos generalizada. En el caso particular del presente trabajo de tesis, interesa la distribución que resulta cuando $s_i = 0$, la que se obtiene aplicando el límite a la expresión 3.12, con lo que resulta lo siguiente:

$$y_i(t_i) = exp\left(-exp\left(-\frac{t_i - a_i}{b_i}\right)\right) \tag{3.13}$$

la cual corresponde a la función de distribución acumulada de valores extremos tipo I, en la versión aplicable al máximo, distribución relacionada a la correspondiente al error en el caso del modelo de tiempo de falla univariado.

3.2.2. Estimación con el uso de cópulas

En el caso del uso de cópulas, se puede utilizar la ecuación 2.72, que establece la igualdad entre la función de distribución multivariada conjunta y la función de cópula, lo cual, aplicado al presente modelo, se obtiene:

$$C(F_I, F_S) = F(I, S) \tag{3.14}$$

Reemplazando la expresión anterior en la expresión de la verosimilitud (ecuación 3.9), se tiene:

$$L(\theta) = \prod_{i=1}^{n} (C(F_I, F_S))^{\Delta_I \Delta_S} (F_S - C(F_I, F_S))^{(1-\Delta_I)\Delta_S} (F_I - C(F_I, F_S))^{\Delta_I (1-\Delta_S)} (1 - F_I - F_S + C(F_I, F_S))^{(1-\Delta_I)(1-\Delta_S)} \tag{3.15}$$

Y la log-verosimilitud será:

$$l(\theta) = \sum_{i=1}^{n} (\Delta_I \Delta_S \log(C(F_I, F_S)) + (1-\Delta_I)\Delta_S \log(F_S - C(F_I, F_S)) + \Delta_I (1-\Delta_S) \log(F_I - C(F_I, F_S)) + (1-\Delta_I)(1-\Delta_S) \log(1 - F_I - F_S + C(F_I, F_S))) \tag{3.16}$$

Así, se asumirá distribuciones Weibull como marginales del tiempo a infección y del tiempo a síntomas:

$$I \sim Weibull(\frac{1}{\sigma_I}, e^{X^T \beta}) \tag{3.17}$$

$$S \sim Weibull(\frac{1}{\sigma_S}, e^{X^T \beta}) \tag{3.18}$$

mientras que la distribución conjunta será estimada mediante una cópula Gumbel, cuya expresión se presenta a continuación:

$$C(u, v) = exp(-[(-\log u)^\theta + (-\log v)^\theta]^{1/\theta}) \tag{3.19}$$

Nótese que la expresión de esta cópula es muy similar a la forma logística de la distribución bivariada de valores extremos, por lo que resulta natural utilizar esta cópula para la estimación conjunta.

Capítulo 4

Estudio de Simulación

A fin de evaluar si el modelo planteado permite obtener estimaciones precisas de los parámetros de las distribuciones asumidas para los tiempos de infección y de síntomas, stanto de las marginales como de la conjunta, se pueden efectuar estudios de simulación consistentes generar muestras aleatorias a partir de distribuciones teóricas asumidas, aplicar el modelo a dichas muestras, y verificar si las estimaciones arrojadas por el modelo son cercanas a los valores reales. Específicamente, para efectuar los estudios de simulación se realizarán las siguientes actividades:

1. Definición de los parámetros teóricos.
2. Definición y generación de las covariables.
3. Simulación de los tiempos de infección y síntomas.
4. Simulación del tiempo de observación (censura) y de los indicadores de censura.
5. Aplicación del modelo a los datos simulados.

A continuación se detalla cada una de las actividades antes señaladas, y se presentan los resultados obtenidos.

4.1. Definición de los parámetros teóricos

El primer paso del estudio de simulación que permite evaluar si el modelo estima adecuadamente los parámetros de las distribuciones asumidas para los tiempos de infección y síntomas consiste justamente en establecer los valores reales o teóricos de dichos parámetros. Para ello se puede utilizar la ecuación 2.73 del modelo de tiempo de falla acelerado bivariado:

$$\begin{pmatrix} \log(T_1) \\ \log(T_2) \end{pmatrix} = \begin{pmatrix} X_1^T \beta_1 \\ X_2^T \beta_2 \end{pmatrix} + \begin{pmatrix} \sigma_1 \epsilon_1 \\ \sigma_2 \epsilon_2 \end{pmatrix} \tag{4.1}$$

En ese sentido, puede observarse que los parámetros a estimar son σ_1, σ_2 y los coeficientes β del componente sistemático de la regresión, los cuales, como se mencionó en la sección 2.6.5, pueden ser los iguales o diferentes para ambas variables. Adicionalmente, como se asume una distribución conjunta bivariada para los tiempos de infección y síntomas, un parámetro adicional será justamente aquel que representa la dependencia entre ambos tiempos, tanto para la distribución bivariada de valores extremos como para la cópula.

En ese sentido, el primer paso de la simulación consiste en establecer valores para los parámetros antes mencionados, a fin de verificar si las las estimaciones del modelo arrojan valores cercanos a los valores establecidos.

4.2. Definición y generación de las covariables.

El siguiente paso consiste en la la generación de valores para cada una las covariables asociadas a los coeficientes de regresión. Así, para las simulaciones a generar se asumirán covariables numéricas que siguen una distribución normal con una media y una deraviación estándar predeterminadas.

4.3. Simulación de los tiempos de infección y síntomas.

Para la generación de los valores simulados de los tiempos de infección y síntomas se utilizará la ecuación 4.1 que se vuelve a presentar a continuación:

$$\begin{pmatrix} \log(T_1) \\ \log(T_2) \end{pmatrix} = \begin{pmatrix} X_1^T \beta_1 \\ X_2^T \beta_2 \end{pmatrix} + \begin{pmatrix} \sigma_1 \epsilon_1 \\ \sigma_2 \epsilon_2 \end{pmatrix}$$

Aplicando la exponencial a ambos lados de la ecuación se obtiene lo siguiente:

$$\begin{pmatrix} T_1 \\ T_2 \end{pmatrix} = \begin{pmatrix} exp(X_1^T \beta_1 + \sigma_1 \epsilon_1) \\ exp(X_2^T \beta_2 + \sigma_2 \epsilon_2) \end{pmatrix} \tag{4.2}$$

De la expresión anterior se observa que los valores de las covariables (X_1 y X_2), de los coeficientes de dichas covariables (β_1 y β_2) y de los parámetros de escala (σ_1 y σ_2) son co-

nocidos, los únicos valores desconocidos son los errores ϵ_1 y ϵ_2, sin embargo, sí se conoce, o más precisamente, se asume la distribución conjunta de dichos errores. En ese sentido, para obtener los valores simulados de los tiempos de infección y síntomas (T_1 y T_2), basta con generar una muestra aleatoria de la distribución conjunta de los errores ϵ_1 y ϵ_2 y aplicar la ecuación 4.2 para la obtención de los valores simulados de los tiempos de infección y síntomas.

Asimismo, para determinar la distribución conjunta de los errores a utilizar, cabe recordar los dos métodos para modelar la distribución conjunta señalados en los capítulos anteriores, los cuales son la distribución bivariada conjunta de los errores y las cópulas. A continuación se describe la generación de los valores simulados de los errores con los modelos antes señalados:

1. **Distribución bivariada conjunta de los errores.**

   Como se mencionó en el punto 3.2.1, en este modelo se asume directamente una distribución conjunta de los errores aleatorios. En particular, se asume que los mismos siguen la forma logística de la distribución bivariada de valores extremos, con el parámetro $s_i = 0$, a fin de que las marginales sean distribuciones de valores extremos tipo I. Cabe mencionar que la distribución bivariada utilizada se encuentra incorporada en el paquete "evd" de R, el cual también implementa otras formas de la distribución de valores extremos bivariada. Asimismo, el paquete cuenta con la función *rbvevd*, la cual permite generar muestras aleatorias de las formas antes mencionadas (Stephenson (2022)).

   Cabe recordar que la forma logística de la distribución bivariada de valores extremos se mostró en la ecuación 3.11, y se presenta nuevamente a continuación:

$$F(t_1, t_2) = exp\left[-\left(y_1(t_1)^{1/r} + y_2(t_2)^{1/r}\right)^r\right]$$

   con $y_i(t_i)$ correspondiente a las distribuciones marginales de las variables, que siguen la forma señalada en la ecuación 3.12, presentada a continuación:

$$y_i(t_i) = exp\left(-exp\left(-\frac{t_i - a_i}{b_i}\right)\right)$$

   Así, para generar la muestra aleatoria de la distribución conjunta de los errores, se asumirá que $s_i = 0$, para tener como marginales a la distribución de valores extremos tipo I en su versión del máximo, y que $a_i = 0$ y $b_i = 1$, asunciones correspondientes al modelo de tiempo de falla acelerado, según lo senñalado en el punto 2.6.4. Finalmente,

se asumirá también un valor dado del parámetro de dependencia r, similar al caso de los parámetros σ_i y β_i, indicado en el punto 4.1.

2. Modelo de cópulas.

Para este modelo, se utiliza la igualdad señalada en la ecuación 2.79, que se vuelve a presentar a continuación:

$$C(u_1, u_2, ..., u_n) = F(x_1, x_2, ..., x_n)$$

La ecuación anterior establece que la función de cópula arroja los mismos resultados que la función de distribución acumulada, es decir, la cópula arroja las probabilidades acumuladas de la distribución conjunta.

Asimismo, se sabe que $u_i = F(x_i)$, es decir, la cópula toma como argumentos las distribuciones acumuladas marginales de cada variable. En ese sentido, puede observarse que, de conocerse el valor de u_i, se puede obtener el valor de la variable aleatoria asociada a dicho u_i mediante la inversa de la función de distribución acumulada marginal ($x_i = F^{-1}(u_i)$).

En ese sentido, para obtener una muestra aleatoria de las variables de interés X_i, basta generar una muestra aleatoria de valores u_i que satisfacen la función de cópula, y aplicar a dichos valores u_i la inversa de la función de distribución acumulada marginal asociada a cada variable aleatoria x_i. Para efectuar la primera parte de la simulación (generación de la muestra de u_i) se puede utilizar la librería *copula* de R, la cual cuenta con la función *rcopula* que permite generar la muestra aleatoria de los u_i requerida, para distintas funciones de cópula (Hofert et al. (2022)).

Para el caso específico de la simulación de los tiempos de inversión y síntomas, materia del presente trabajo, se asume una cópula Gumbel, al presentar una forma muy similar a la forma logística de la distribución de valores extremos bivariada, mencionada en el punto anterior. La expresión de la función de dicha cópula fue mostrada en la ecución 3.16, y se vuelve a presentar a continuación:

$$C(u, v) = exp(-[(-\log u)^\theta + (-\log v)^\theta]^{1/\theta})$$

con $\theta \in [1, \infty)$, parámetro que también debe ser dado, similar al caso del parámetro r de la distribución conjunta mencionada en el punto anterior.

Así, una vez generados los valores u y v, estos corresponden a los valores de las distribuciones acumuladas marginales de los errores aleatorios, en ese sentido, los valores de los errores se obtienen aplicando a u y a v las inversas de dichas marginales. Cabe recordar que, como se asumen distribuciones marginales Weibull, la distribución de los errores es log-Weibull , o de valores extremos tipo I para el mínimo, con parámetros $(0, 1)$, en ese sentido, la distribución marginal de los cada uno de los errores es:

$$F(x) = 1 - exp(-exp(x)) \tag{4.3}$$

y su función inversa será:

$$F^{-1}(x) = log(-log(1 - x)) \tag{4.4}$$

Función con la cual se obtienen los valores simulados de los errores aleatorios. Una vez generados ellos, se pueden calcular los valores simulados de los tiempos de infección y síntomas con la ecuación 2.68, que se presenta a continuación:

$$\log(T_i) = X_i^T \beta + \sigma \epsilon_i$$

Si se toma el logaritmo a ambos lados de la ecuación, resulta:

$$T_i = exp(X_i^T \beta + \sigma \epsilon_i) \tag{4.5}$$

Así, los valores simulados de los tiempos de infección y síntomas se obtendrán reemplazando los valores simulados de los errores en la ecuación anterior.

4.4. Simulación del tiempo de observación (censura) y de los indicadores de censura.

Si bien en el punto anterior se generan los tiempos correspondientes a la infección y a los síntomas, en la práctica dichos tiempos son desconocidos, pues la información que se tiene es que, en un punto del tiempo, el paciente presenta infección o síntomas, pero no se sabe con exactitud cuándo adquirió la infección o cuándo aparecieron los síntomas, es decir, los

datos están censurados. En ese sentido, se requiere simular el tiempo de observación de los pacientes, el cual será el tiempo en el cual se determina si el paciente presenta infección o no, y si presenta síntomas o no.

Así, la asunción natural es que el tiempo de observación sea independiente de los tiempos de infección y síntomas y sea completamente aleatorio, consecuentemente, la distribución uniforme es la que mejor se ajusta a estas características. En este caso particular, se asumirá que el dominio de esta distribución uniforme irá entre el menor entre los valores del primer cuartil de los tiempos de infección y síntomas, y mayor valor entre los valores del tercer cuartil de los tiempos de infección y síntomas, a fin de asegurar suficientes observaciones tanto para los casos de infección y no infección, como para los de síntomas y no síntomas.

Una vez generados los tiempos de observación, o más precisamente, tiempos de censura, se deben generar los indicadores que denotan si el paciente presenta infección o no, y si presenta síntomas o no. En el caso de la variable indicadora de infección (Δ_I), se compara el tiempo de censura con el tiempo de infección y si el tiempo de infección es menor, quiere decir que el evento ocurrió, por lo que la variable indicadora de infección debe ser uno, y en caso contrario, cuando el tiempo de infección es mayor, quiere decir que el evento aún no ocurrió, por lo que el indicador de infección debe ser cero. Por otro lado, en el caso de la variable indicadora de los síntomas (Δ_S), se procede de forma similar, pero utilizando el tiempo de síntomas en lugar del de infección. Es decir:

$$\Delta_I = I(T_I <= tiempo\,de\,censura) \tag{4.6}$$

$$\Delta_S = I(T_S <= tiempo\,de\,censura) \tag{4.7}$$

4.5. Aplicación del modelo a los datos simulados.

Una vez generadas las simulaciones de los datos antes señalados, corresponde aplicar a los mismos la función de log-verosimilitud definida en la ecuación 3.10, que se muestra a continación:

$$l(\theta) = \sum_{i=1}^{n} (\Delta_I \Delta_S \log(F_{IS}) + (1 - \Delta_I)\Delta_S \log(F_S - F_{IS}) + \Delta_I(1 - \Delta_S) \log(F_I - F_{IS}) + (1 - \Delta_I)(1 - \Delta_S) \log(1 - F_I - F_S + F_{IS}))$$

y finalmente, los valores que maximicen la función de log-verosimilitud anterior serán las estimaciones de los parámetros del modelo.

Sin embargo, cabe precisar que los datos observados son el tiempo de censura y los indicadores de infección y de síntomas, mientras que, como se mencionó en el punto anterior, los tiempos en los cuales ocurren la infección y los síntomas son generalmente desconocidos (latentes). En ese sentido, el tiempo de censura será la variable sobre la cual se aplican las distribuciones marginales, y la distribución conjunta / cópula; es decir, dado que en el tiempo de censura se verifican a la vez la presencia de infección y la de los síntomas, dicho tiempo es el que ingresa a la función de verosimilitud como tiempo de infección y como tiempo de síntomas.

4.6. Resultados obtenidos en una simulación.

A fin de ilustrar la precisión del modelo de estimación utilizado, se implementó el código en R incluido en el apéndice A, mediante el cual se efectúa la estimación de parámetros para una muestra de mil datos simulados. Así, se simularon escenarios con una, dos y tres covariables más el intercepto, asumiendo para cada escenario el caso en que los coeficientes de las covariables son los mismos para las dos variables aleatorias, y el caso en que son diferentes.

1. **Distribución bivariada conjunta de los errores.**

   Los siguientes gráficos ilustran los resultados de las estimaciones sobre los datos simulados asumiendo que los errores obedecen la forma logística de la distribución bivariada de valores extremos (los resultados obtenidos se incluyen en el apéndice B):

   - **Una covariable**

     Asumiendo coeficientes iguales para infección y síntomas:

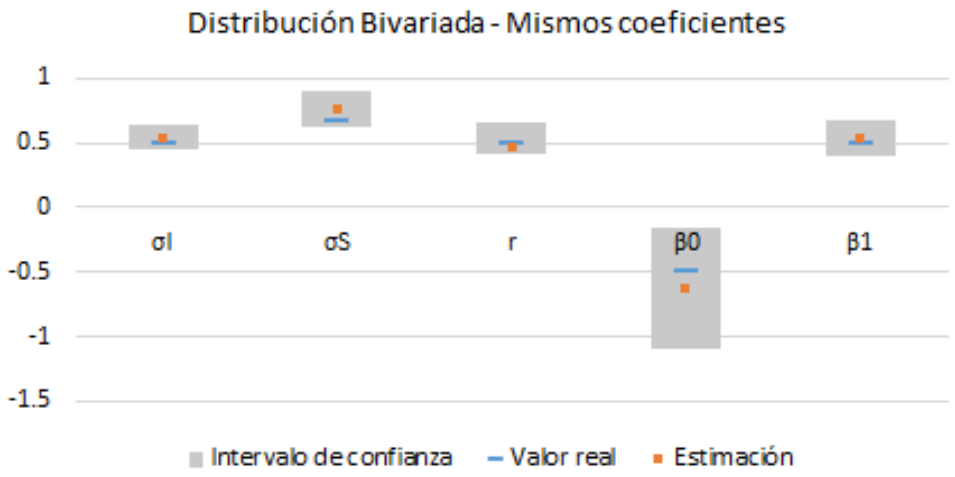


Figura 4.1: Estimaciones puntuales y por intervalos de los parámetros en el escenario de una sola covariable y coeficientes iguales para infección y síntomas.

Asumiendo coeficientes diferentes para infección y síntomas

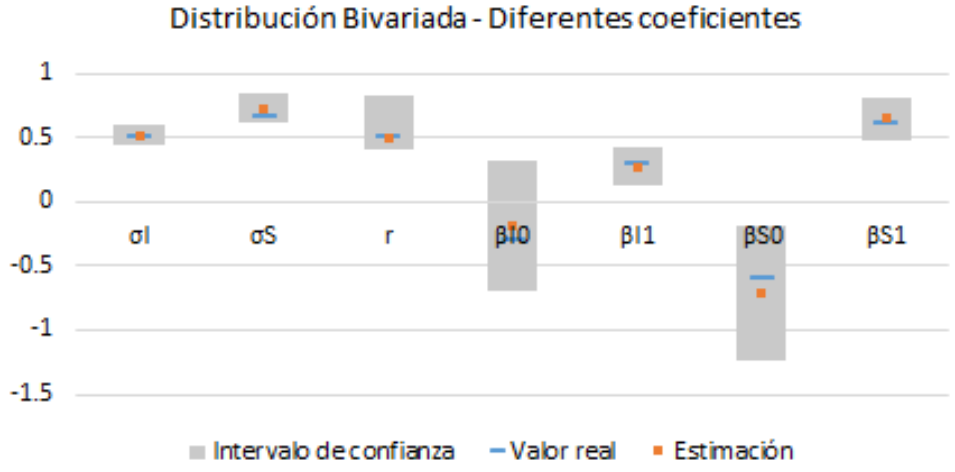


Figura 4.2: Estimaciones puntuales y por intervalos de los parámetros en el escenario de una sola covariable y coeficientes diferentes para infección y síntomas.

- **Dos covariables**

  Asumiendo coeficientes iguales para infección y síntomas:

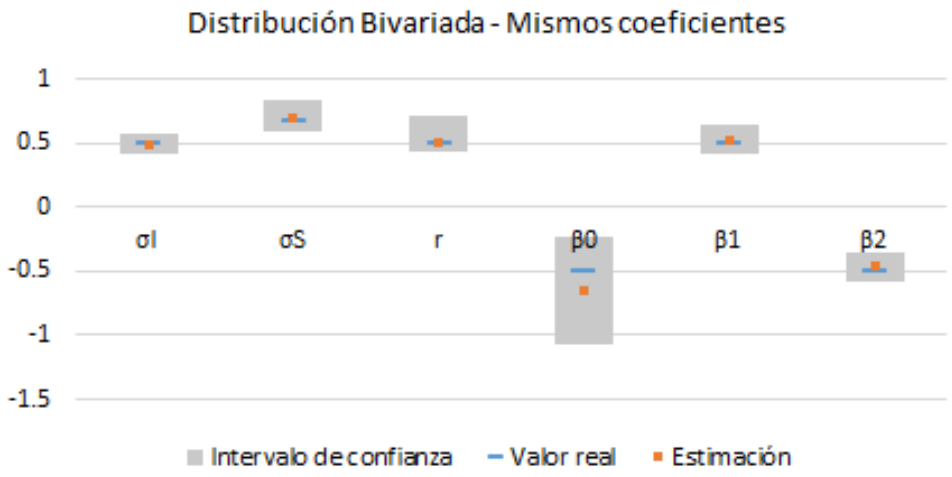


Figura 4.3: Estimaciones puntuales y por intervalos de los parámetros en el escenario de una sola covariable y coeficientes iguales para infección y síntomas.

Asumiendo coeficientes diferentes para infección y síntomas

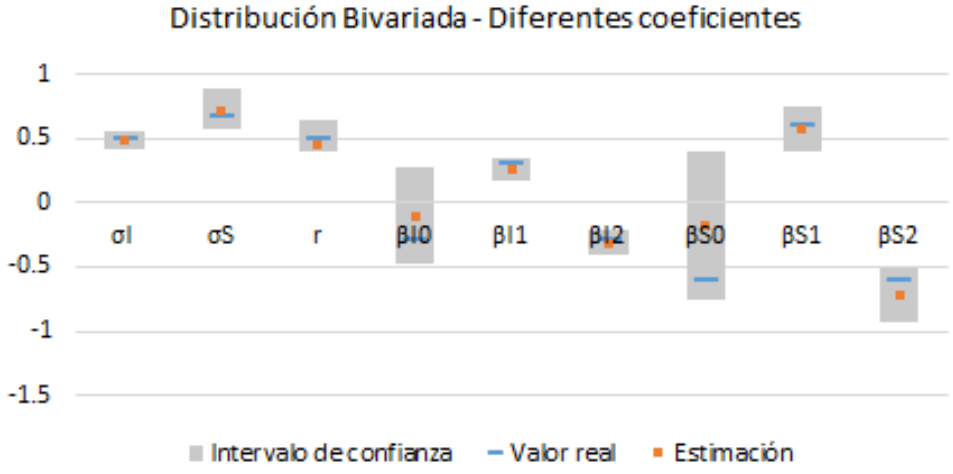


Figura 4.4: Estimaciones puntuales y por intervalos de los parámetros en el escenario de una sola covariable y coeficientes diferentes para infección y síntomas.

- **Tres covariables**

  Asumiendo coeficientes iguales para infección y síntomas:

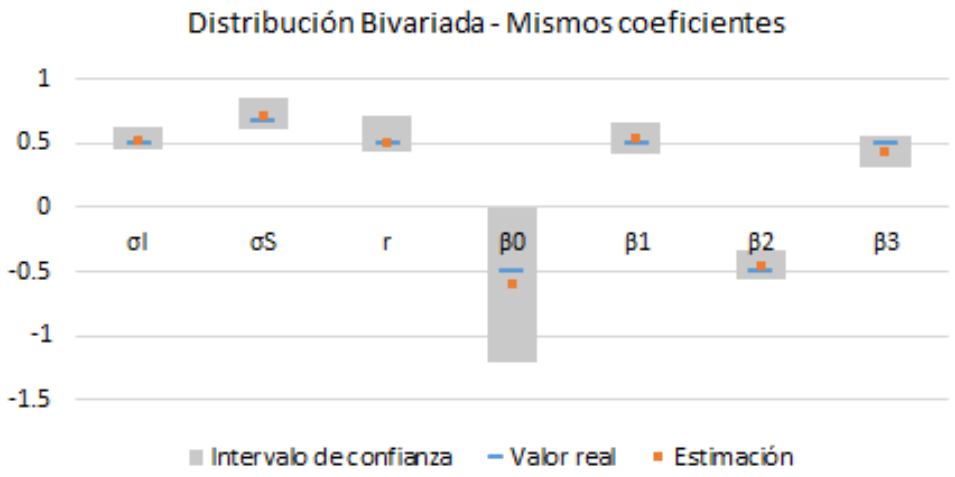


Figura 4.5: Estimaciones puntuales y por intervalos de los parámetros en el escenario de una sola covariable y coeficientes iguales para infección y síntomas.

Asumiendo coeficientes diferentes para infección y síntomas

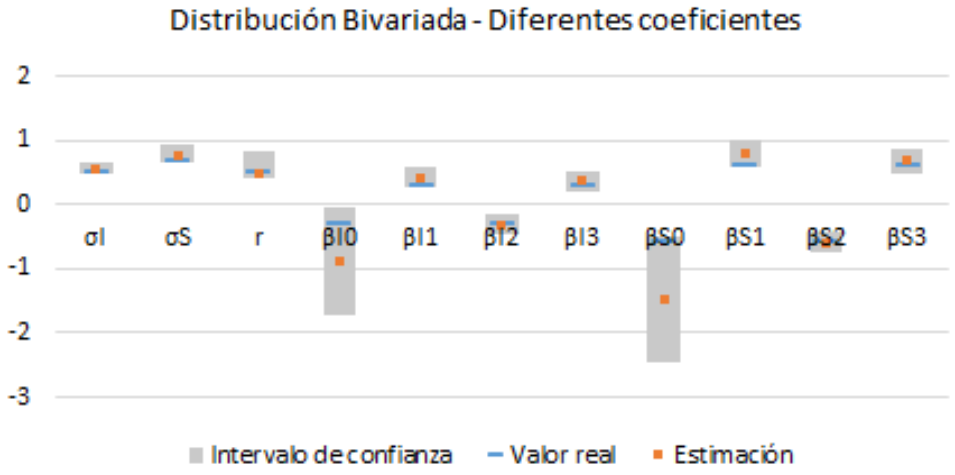


Figura 4.6: Estimaciones puntuales y por intervalos de los parámetros en el escenario de una sola covariable y coeficientes diferentes para infección y síntomas.

En ese sentido, se puede observar que, para esta simulación, las estimaciones puntuales son cercanas a los valores reales y estos se encuentra dentro de los intervalos de confianza, sin embargo, cabe resaltar que, en el caso de los interceptos, los intervalos de confianza son mucho más amplios que los de los demás parámetros, y en algunos casos incluso han resultado no significativos, por lo que estos requerirán muestras de mayor tamaño para su correcta estimación.

2. **Modelo de cópulas.**

Los siguientes gráficos ilustran los resultados de las estimaciones sobre los datos simulados asumiendo que las distribuciones acumuladas marginales cumplen una función de cópula Gumbel (los resultados obtenidos se incluyen en el apéndice B):

- **Una covariable**

  Asumiendo coeficientes iguales para infección y síntomas:

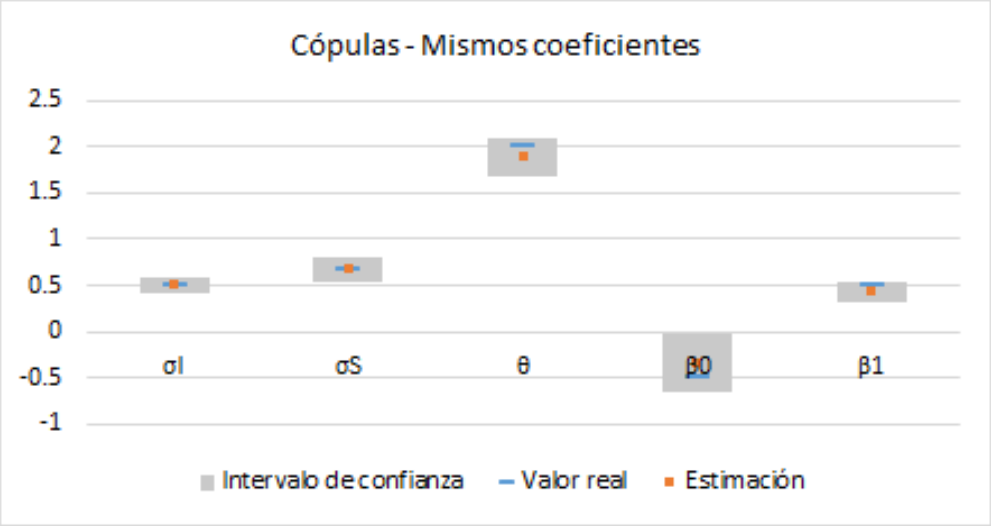


Figura 4.7: Estimaciones puntuales y por intervalos de los parámetros en el escenario de una sola covariable y coeficientes iguales para infección y síntomas.

  Asumiendo coeficientes diferentes para infección y síntomas

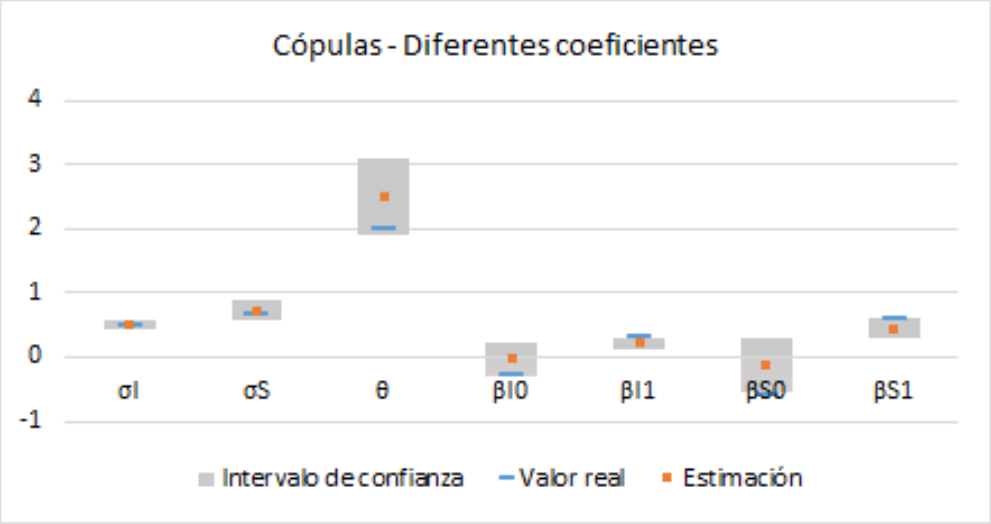


Figura 4.8: Estimaciones puntuales y por intervalos de los parámetros en el escenario de una sola covariable y coeficientes diferentes para infección y síntomas.

- **Dos covariables**

Asumiendo coeficientes iguales para infección y síntomas:

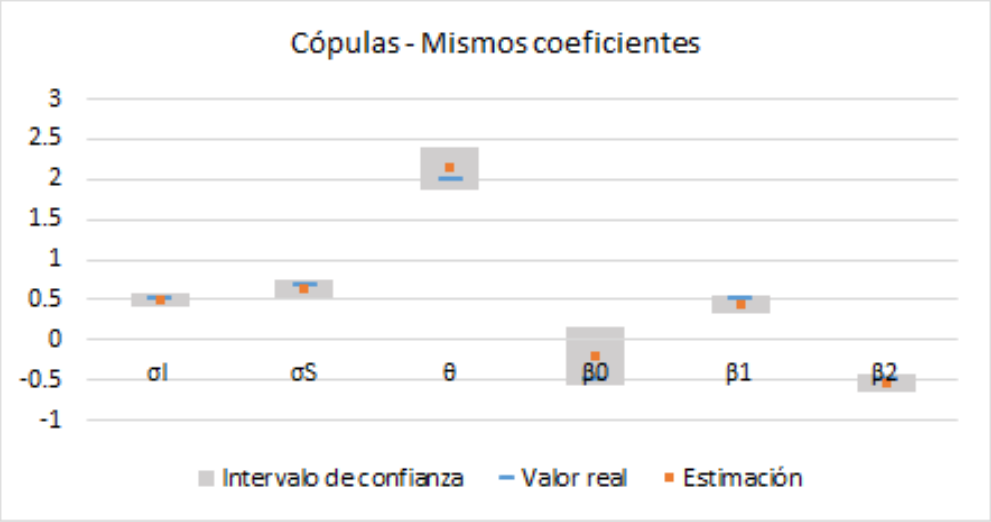


Figura 4.9: Estimaciones puntuales y por intervalos de los parámetros en el escenario de una sola covariable y coeficientes iguales para infección y síntomas.

Asumiendo coeficientes diferentes para infección y síntomas

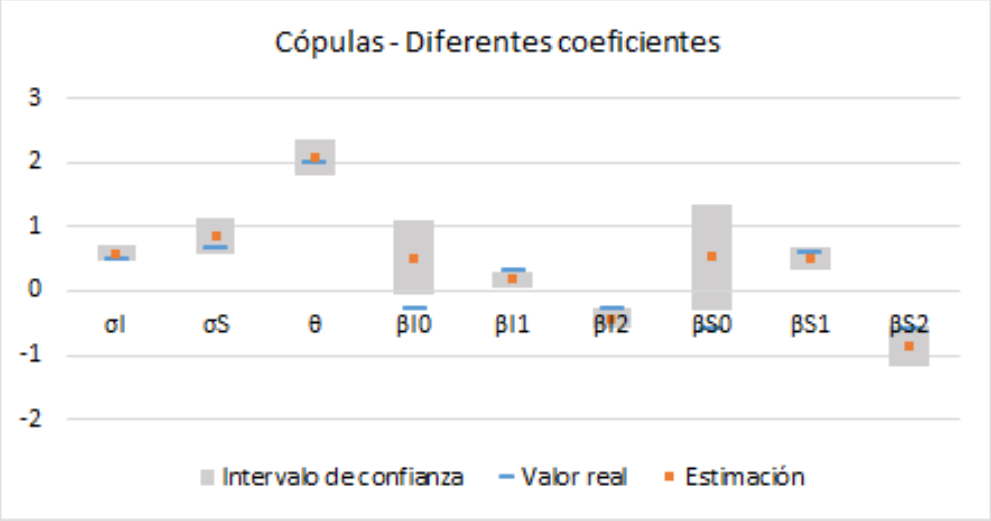


Figura 4.10: Estimaciones puntuales y por intervalos de los parámetros en el escenario de una sola covariable y coeficientes diferentes para infección y síntomas.

- **Tres covariables**

  Asumiendo coeficientes iguales para infección y síntomas:

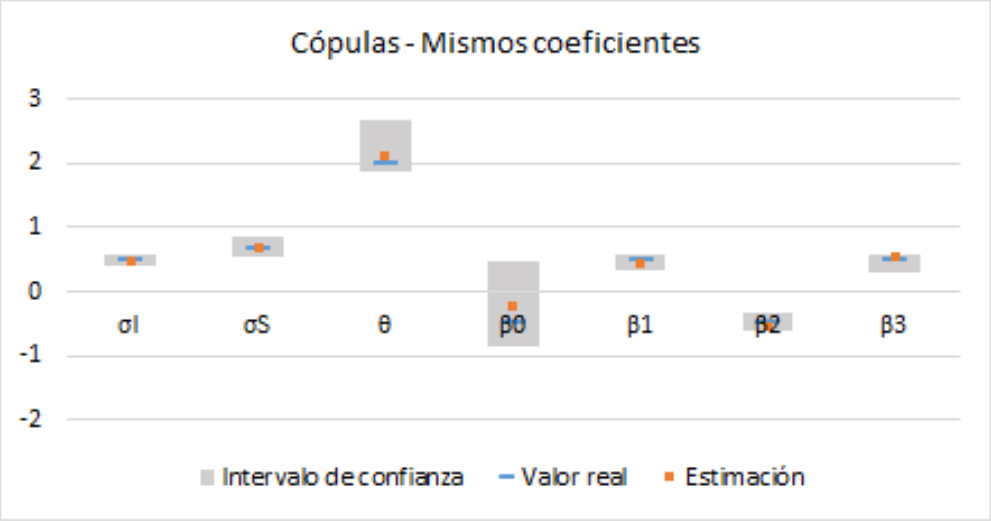


Figura 4.11: Estimaciones puntuales y por intervalos de los parámetros en el escenario de una sola covariable y coeficientes iguales para infección y síntomas.

Asumiendo coeficientes diferentes para infección y síntomas

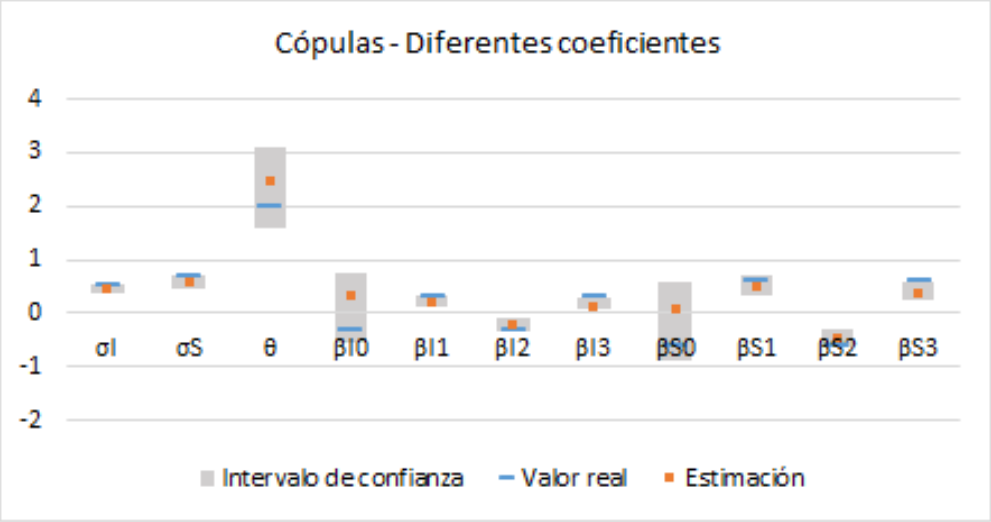


Figura 4.12: Estimaciones puntuales y por intervalos de los parámetros en el escenario de una sola covariable y coeficientes diferentes para infección y síntomas.

En ese sentido, se puede observar que, para esta simulación, las estimaciones puntuales son cercanas a los valores reales y estos se encuentran, en la mayoría de los casos, dentro de los intervalos de confianza. Sin embargo, cabe resaltar que, en el caso de los interceptos, los intervalos de confianza son mucho más amplios que los de los demás parámetros, en algunos casos han resultado no significativos, y, en el caso de dos covariables y diferentes coeficientes, el valor real no se encuentra dentro de los intervalos de confianza, en ese sentido, estos coeficientes requerirán muestras de mayor tamaño para

su correcta estimación.

4.7. Resultados para varias simulaciones.

Si bien los resultados en la simulación anterior resultan satisfactorios, una sola simulación no resulta suficiente para afirmar que el modelo estima adecuadamente los parámetros de las distribuciones asumidas. Para ello, se realiza un alto número de simulaciones y, con las estimaciones obtenidas en cada simulación se evalúa el ratio de cobertura, el cual es igual a la proporción de veces que los intervalos de confianza incluyen el valor real del parámetro, respecto del número total de simulaciones.

Así, para afirmar que el modelo estima los parámetros adecuadamente, la cobertura debe encontrarse cercana al nivel de confianza asumido para los intervalos de confianza, en este caso, se asumirá un nivel de confianza de 95 % y se generarán 1000 simulaciones del caso correspondiente a tres covariables, tanto con coeficientes iguales como diferentes. Asimismo, se ha detectado que en algunas ocasiones el programa no arroja estimaciones para la matriz hessiana, insumo fundamental para el cálculo de los intervalos de confianza, en ese sentido, para esos casos particulares los intervalos de confianza se estimarán mediante bootstrapping, utilizando el intervalo generado por los percentiles 0.025 y 0.975 de las estimaciones arrojadas por dicho método de remuestreo.

En ese sentido, a continuación se presentan los gráficos con los resultados de los niveles de cobertura calculados para el caso en que se asumen coeficientes iguales para los tiempos de infección y de síntomas, tanto para el modelo de distribución bivariada como para el modelo de cópulas:

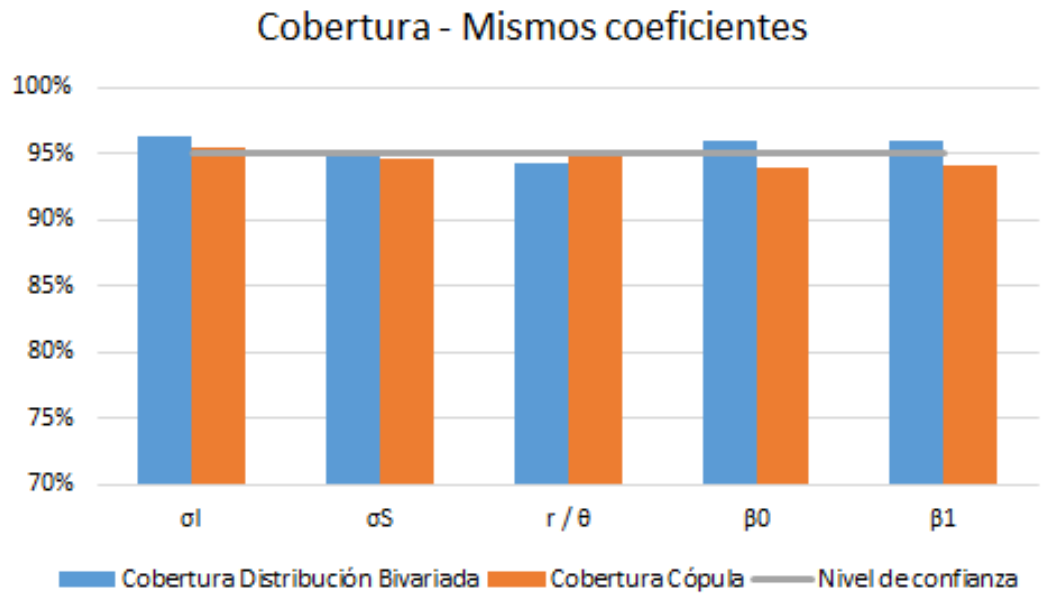


Figura 4.13: Niveles de cobertura determinados a mediante la ejecución de mil simulaciones de tamaño de muestra de mil observaciones, para el caso de una sola covariable y coeficientes iguales para infección y síntomas.

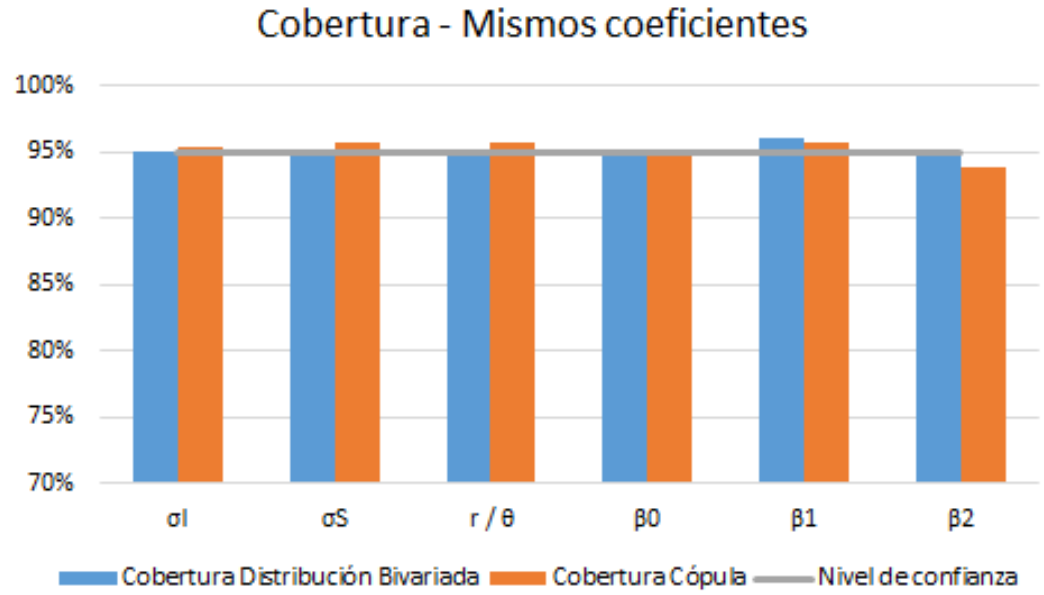


Figura 4.14: Niveles de cobertura determinados a mediante la ejecución de mil simulaciones de tamaño de muestra de mil observaciones, para el caso de dos covariables y coeficientes iguales para infección y síntomas.

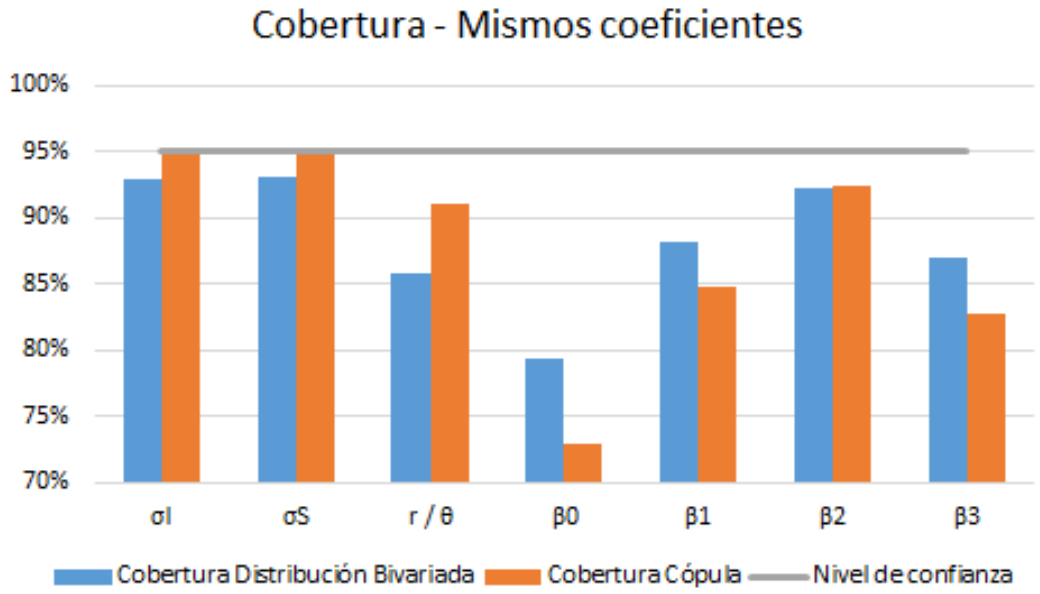


Figura 4.15: Niveles de cobertura determinados a mediante la ejecución de mil simulaciones de tamaño de muestra de mil observaciones, para el caso de tres covariables y coeficientes iguales para infección y síntomas.

Por otro lado, a continuación se presentan los gráficos de cobertura para el caso en el que se asumen coeficientes diferentes para ambos tiempos:

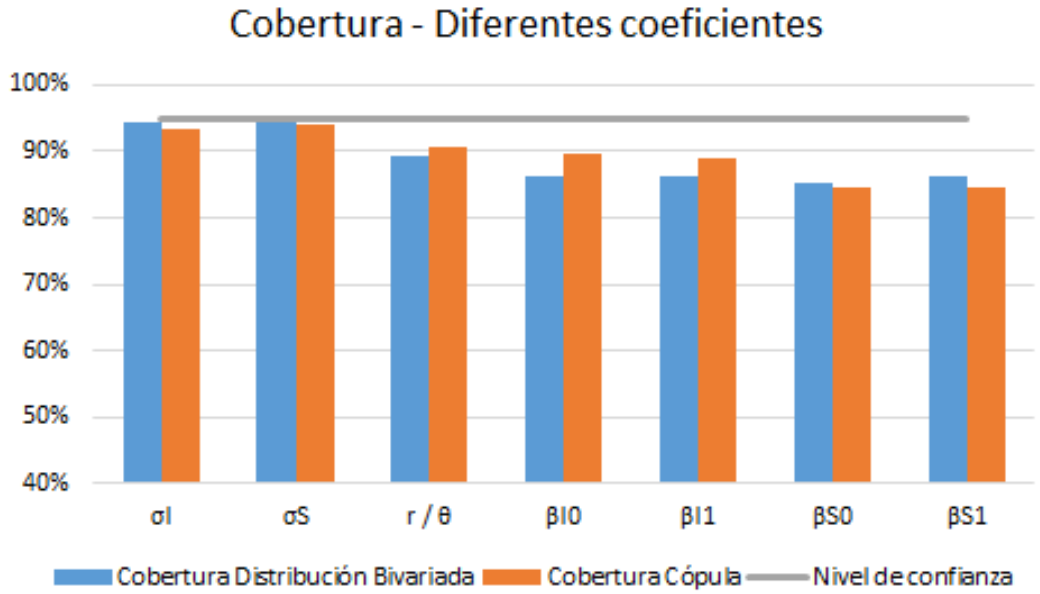


Figura 4.16: Niveles de cobertura determinados a mediante la ejecución de mil simulaciones de tamaño de muestra de mil observaciones, para el caso de una sola covariable y diferentes coeficientes para infección y síntomas.

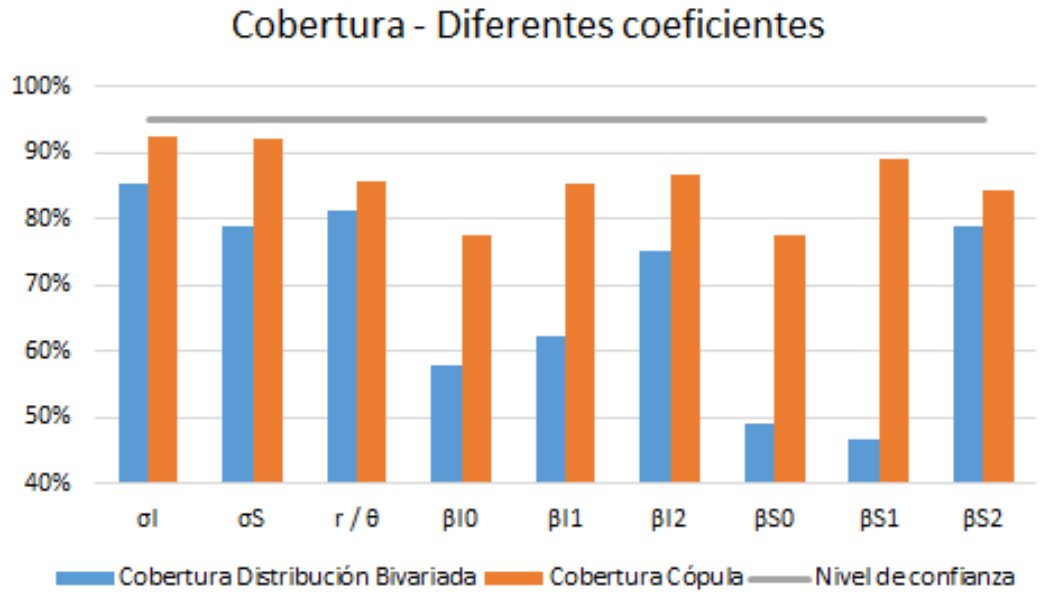


Figura 4.17: Niveles de cobertura determinados a mediante la ejecución de mil simulaciones de tamaño de muestra de mil observaciones, para el caso de dos covariables y diferentes coeficientes para infección y síntomas.

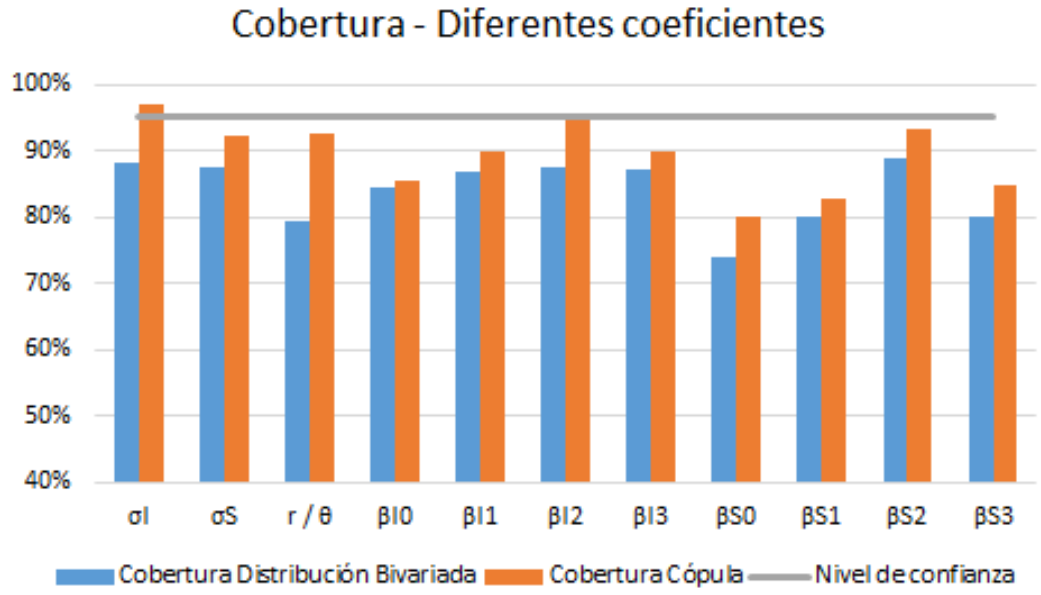


Figura 4.18: Niveles de cobertura determinados a mediante la ejecución de mil simulaciones de tamaño de muestra de mil observaciones, para el caso de tres covariables y diferentes coeficientes para infección y síntomas.

Cabe resaltar que el modelo de cópulas brinda mejores resultados de cobertura que el modelo de la distribución bivariada en los casos más complejos (tres covariables y diferentes

coeficientes), es decir, para los casos en los que se tiene nueve y once parámetros. Asimismo, en el apéndice B se incluyen las tablas con los valores numéricos resultantes de los cálculos de cobertura en las simulaciones efectuadas.

Finalmente, se observa que la precisión del modelo cae progresivamente a medida que se va haciendo más complejo, es decir, a medida que se incrementa el número de parámetros, por lo se pueden incorporar al mismo variantes que permitan mejorar su precisión, tales como splines, o se pueden utilizar modelos no paramétricos, variantes que se encuentran fuera del alcance del presente trabajo.

Capítulo 5

Aplicación

En este capítulo se aplicarán los modelos descritos en los capítulos anteriores a una base de datos real: el estudio de notificación de parejas. En ese sentido, a continuación se describe dicha base de datos, así como los resultados obtenidos de la aplicación de los modelos referidos.

5.1. Descripción de la base de datos a utilizar

El estudio de notificación de parejas fue llevado a cabo por Golden et al. (2005), en el cual se enroló a mujeres y a varones heterosexuales que recibieron un diagnóstico de infección de gonorrea o clamidia genital a lo más catorce días previos al inicio del estudio. Asimismo, fue llevado a cabo en el condado King del estado de Washington en Estados Unidos de América, entre el 29 de septiembre de 1998 y el 7 de marzo de 2003, y logró recopilar información de 1,864 individuos, luego de descartar algunos por considerarse no elegibles para la investigación.

Así, el estudio consistió en evaluar el efecto de una terapia acelerada en la reinfección de gonorrea o clamidia de los individuos participantes. Así, los mismos fueron separados en dos grupos: uno de control de 933 personas a las cuales se aplicó la terapia estándar, y otro de intervención de 931 personas a las cuales se les aplicó la terapia acelerada. El objetivo fue verificar si las personas presentaron re-infección y síntomas de su enfermedad previa 90 días posteriores a su inscripción, aunque dicho tiempo de seguimiento varió considerablemente (entre 19 y 161 días) debido a la dificultad para contactar a los participantes y programar las visitas de seguimiento (Paulon et al. (2020)).

En ese sentido, la base de datos resultante está constituida por los siguientes campos:

1. **ptid:** Código identificador del paciente. Para los objetivos del modelo, este campo es irrelevante, por lo que no se tomará en cuenta.

2. **gender:** Género del paciente (1: varón y 0: mujer). En el siguiente gráfico de barras se observar que los participantes del estudio fueron principalmente varones:

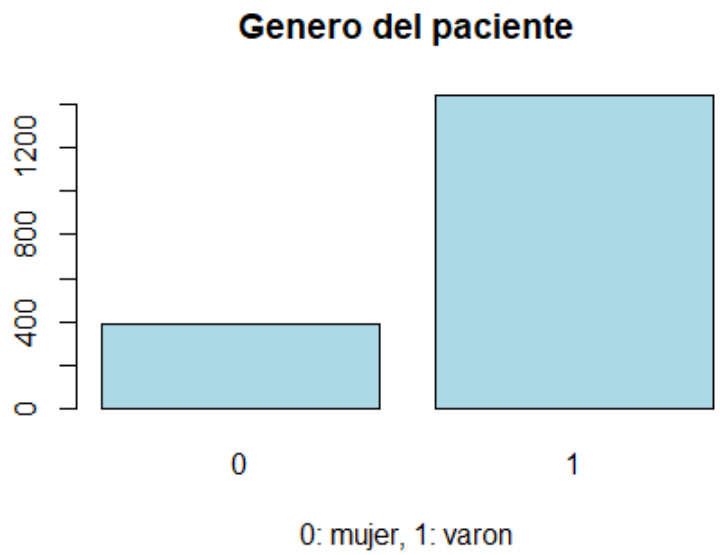


Figura 5.1: Distribución de los pacientes según género.

3. **arm:** Grupo al cual pertenece el paciente (1: intevención y 0: control). Como se mencionó previamente, los grupos están consitituidos de casi la misma cantidad de pacientes, los cual se evidencia en el siguiente gráfico que muestra la distribución de pacientes según su grupo:

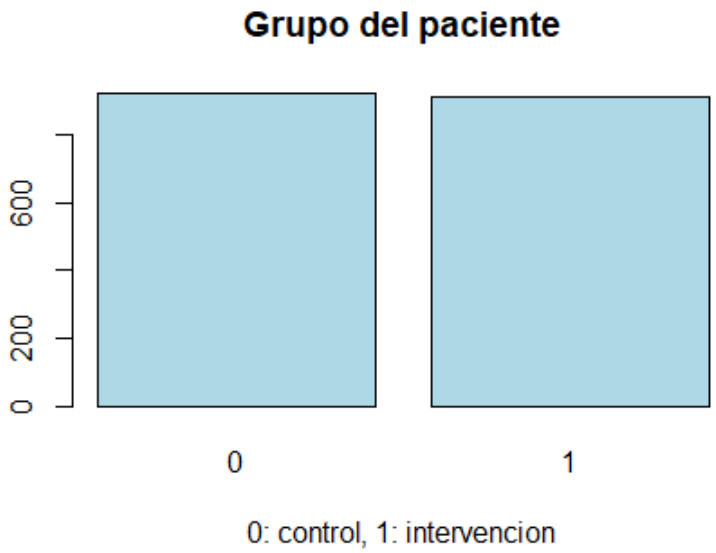


Figura 5.2: Distribución de los pacientes según grupo al que pertenecieron.

4. **age:** Edad del paciente. El siguiente histograma muestra la distribución de los pacientes según su edad:

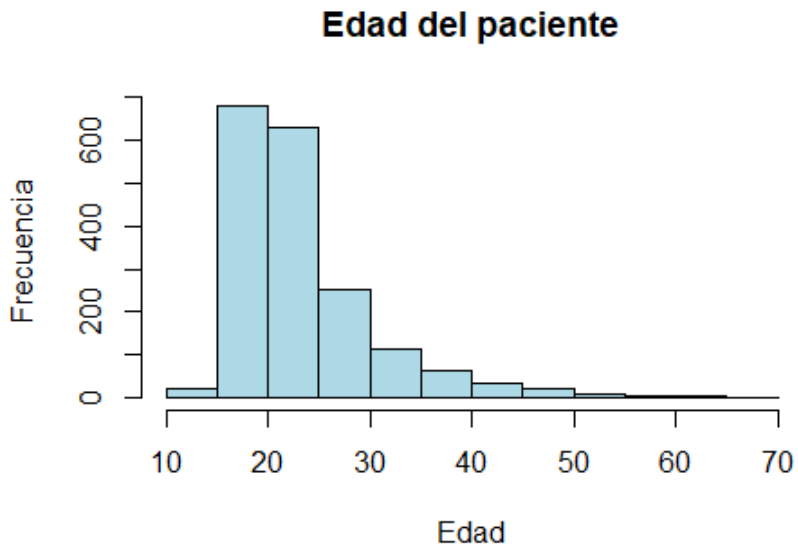


Figura 5.3: Distribución de los pacientes según grupo al que pertenecieron.

Se observa que los participantes fueron principalmente jóvenes alrededor de los 20 años.

5. **time:** Número de días posteriores a la inscripción en el que se aplicó la prueba. El siguiente histograma muestra la distribución de esta variable:

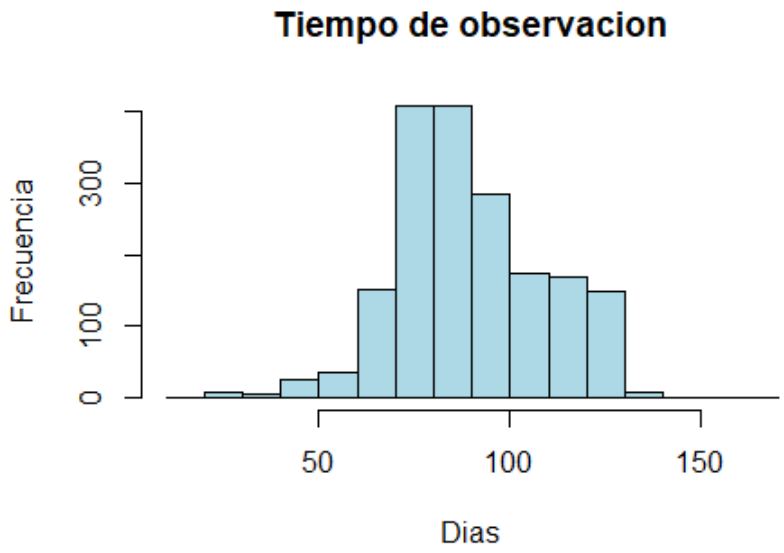


Figura 5.4: Distribución del tiempo en el cual se efectuó el seguimiento del paciente.

Como se mencionó previamente, el objetivo del estudio era efectuar el seguimiento 90 días después de la inscripción, sin embargo, dados los inconvenientes operativos, se tiene una distribución relativamente dispersa de dicho tiempo.

6. **disease:** Indicador de presencia de infección (1) o ausencia de la misma (0). El siguiente gráfico muestra que la mayoría de pacientes no presentaron infección al momento de la prueba.

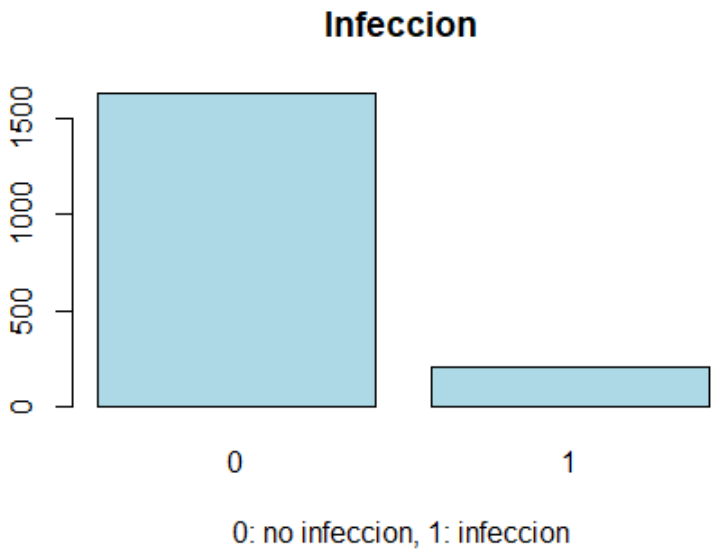


Figura 5.5: Distribución de los pacientes según si, al momento del seguimiento, presentaron infección o no.

7. **symptons:** Indicador de presencia de síntomas (1) o ausencia de estos (0). El siguiente gráfico muestra que la mayoría de pacientes tampoco presentaron síntomas al momento de la prueba.

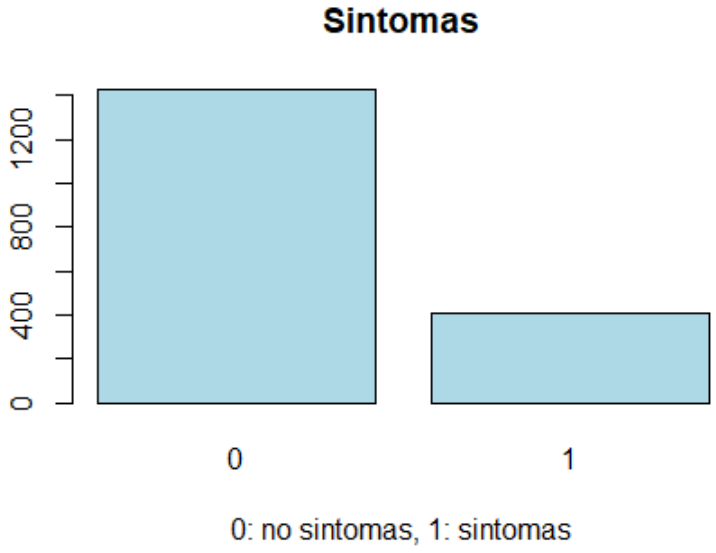


Figura 5.6: Distribución de los pacientes según si, al momento del seguimiento, presentaron síntomas o no.

5.2. Aplicación del modelo a la base de datos utilizada

A continuación se presentan los valores de las estimaciones de los parámetros obtenidos luego de aplicar los modelos de distribución bivariada de valores extremos y de cópulas, tanto asumiendo coeficientes de las covariables iguales, como diferentes:

- **Mismos coeficientes para infección y síntomas.**

Parámetro	**Covariable**	**Estimación - Distribución bivariada de valores extremos**	**Estimación - Cópulas**
σ_I		464.9466	88.9517
σ_S		910.4944	148.5942
r / θ		0.7407	1.3838
β_0	intercepto	6.5404	-96.8295
β_1	género	-1.3655	31.7624
β_2	grupo	-6.5908	2.8888
β_3	edad	15.7716	12.1465

- **Diferentes coeficientes para infección y síntomas.**

Parámetro	**Covariable**	**Estimación - Distribución bivariada de valores extremos**	**Estimación - Cópulas**
σ_I		2.6091e+16	11.2958
σ_S		4.2552e+08	1.1988
r / θ		0.5904	1.3139
β_{I0}	intercepto	13.5273	0.3736
β_{I1}	género	13.0140	2.8051
β_{I2}	grupo	-27.0107	-15.7887
β_{I3}	edad	4.5588	1.6703
β_{S0}	intercepto	-185.2689	6.2132
β_{S1}	género	116.4887	-0.6194
β_{S2}	grupo	21.5623	-0.2371
β_{S3}	edad	2.6177	0.0244

5.3. Comparación de resultados

A fin de determinar qué modelo ajusta mejor a los datos, se puede utilizar el criterio de información de Akaike, que se presenta a continuación para todos los modelos:

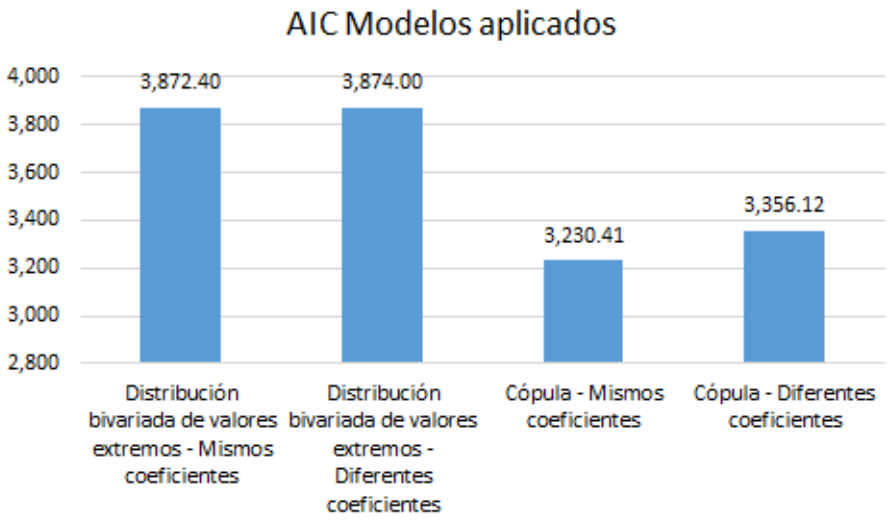


Figura 5.7: AIC de los modelos aplicados a la base de datos del estudio de notificación de parejas.

Se puede observar que el modelo que presenta menor AIC es el de cópulas con mismos coeficientes, en ese sentido, este es el modelo que mejor ajusta a los datos. Dado este hecho, cabe recordar que el modelo de cópulas utilizado asume, para cada uno de los tiempos de infección y síntomas, la ecuación 4.5, la cual se muestra a continuación:

$$T_i = exp(X_i^T \beta + \sigma \epsilon_i) \tag{5.1}$$

A partir de la ecuación anterior, se puede determinar el efecto porcentual del incremento en una unidad de una covariable en particular, manteniendo las demás constantes. Así, si T_0 es el tiempo de ocurrencia base y T_1 el tiempo de ocurrencia cuando la covariable X_j se incrementa en una unidad, se obtiene lo siguiente:

$$\begin{aligned}
\frac{T_1}{T_0} &= \frac{exp(X_{sinj}^T\beta_{sinj} + (X_j+1)\beta_j + \sigma\epsilon_i)}{exp(X_{sinj}^T\beta_{sinj} + X_j\beta_j + \sigma\epsilon_i)} \\
\frac{T_1}{T_0} &= exp(X_{sinj}^T\beta_{sinj} + (X_j+1)\beta_j + \sigma\epsilon_i - (X_{sinj}^T\beta_{sinj} + X_j\beta_j + \sigma\epsilon_i)) \\
\frac{T_1}{T_0} &= exp(X_{sinj}^T\beta_{sinj} + X_j\beta_j + \beta_j + \sigma\epsilon_i - X_{sinj}^T\beta_{sinj} - X_j\beta_j - \sigma\epsilon_i) \\
\frac{T_1}{T_0} &= exp(\beta_j)
\end{aligned} \tag{5.2}$$

En ese sentido, la variación porcentual se obtendrá restando uno a cada lado de la expresión anterior, con lo que resulta:

$$\begin{aligned}
\frac{T_1}{T_0} - 1 &= exp(\beta_j) - 1 \\
\Delta &= exp(\beta_j) - 1
\end{aligned} \tag{5.3}$$

Con la expresión anterior, se puede evaluar el impacto en el tiempo de ocurrencia de infección y síntomas que genera el incremento de una unidad en cada covariable, según el modelo de cópulas con coeficientes iguales, el cual es el que mejor se ajusta a los datos:

- **Género:** El cambio porcentual es de $exp(31.7624) - 1 = 6.22 \times 10^{15}\,\%$, lo cual quiere decir que, condicionado al grupo y a la edad, el tiempo de ocurrencia de infección o síntomas en los varones es muchísimo mayor al de las mujeres, por lo cual la reinfección se ha presentado con mucha mayor frecuencia en las mujeres.

- **Grupo:** El cambio porcentual es de $exp(2.8888) - 1 = 1,697.17\,\%$, lo cual quiere decir que, condicionado al género y a la edad, el tiempo de ocurrencia de infección o síntomas en los pacientes del grupo de intervención (los que recibieron la terapia acelerada) es mucho mayor al del grupo control (los que recibieron la terapia estándar), por lo cual la reinfección se ha presentado con mucha mayor frecuencia en el grupo control, lo cual permite afirmar que la terapia acelerada ha prevenido exitosamente la presencia de infección y síntomas.

- **Edad:** El cambio porcentual es de $exp(12.1465) - 1 = 1.88 \times 10^7\,\%$, lo cual quiere decir que, condicionado al género y al grupo, el tiempo de ocurrencia de infección o síntomas se incrementa significativamente en pacientes de mayor edad respecto de los más jóvenes, lo cual significa que la reinfección se ha presentado más en pacientes de menor edad.

Capítulo 6

Conclusiones

6.1. Conclusiones

Cabe resaltar las siguientes conclusiones principales del trabajo elaborado

1. Las simulaciones efectuadas demuestran que el modelo de cópulas estima mejor los parámetros que el modelo que asume una distribución bivariada de valores extremos. Asimismo, las cópulas presentan la ventaja de que utilizan las distribuciones marginales de los tiempos de infección y síntomas, mientras que su relación de dependencia se modela mediante una función de dichas marginales, que justamente es la función de cópula, por lo que brindan mayor flexibilidad que en el caso del modelo que asume una distribución bivariada de los errores, pues con las cópulas, se puede tomar cualquier distribución marginal para cada variable, en cambio, en el caso del modelo bivariado de los errores, las distribuciones marginales deben provenir de la distribución conjunta.

2. La precisión en la estimación de los parámetros con ambos modelos disminuye progresivamente a medida que el modelo se hace más complejo, es decir, a medida que se incorporan más covariables y, por ende, más parámetros a estimar.

3. En el caso de la base de datos del estudio de notificación de parejas utilizada para el presente trabajo, el modelo que se ajusta mejor a los datos es el modelo de cópulas que asume los mismos coeficientes para infección y síntomas, pues es el que presenta el menor valor de AIC.

4. Los valores distintos de cero, tanto para el parámetro de dependencia r de la distribución bivariada de valores extremos como para el parámetro de la cópula, demuestran que existe correlación entre los tiempos de infección y síntomas, por lo que no se pue-

de asumir que ambos tiempos son independientes, por el contrario, dicha relación de dependencia debe considerarse en el modelamiento de las variables.

6.2. Sugerencias para investigaciones futuras

En el presente trabajo de tesis se propone un modelo para la estimación del tiempo de infección y del tiempo de síntomas, considerando la correlación intrínseca entre los mismos. Los modelos propuestos son la estimación asumiendo que los tiempos siguen distribución bivariada de valores extremos, y la estimación asumiendo una cópula Gumbel para modelar la relación de dependencia de dichos tiempos. En ese sentido, se pueden implementar las siguientes recomendaciones a fin de sofisticar aún más el modelo propuesto:

1. En el caso de la distribución bivariada conjunta de los errores, se pueden utilizar diferentes formas de la distribución bivariada de valores extremos, como las disponibles en la librería *evd* de R.

2. La asumción de que los errores del modelo siguen la forma logística de la distribución bivariada de valores extremos puede llevar a obtener nuevas versiones de la distribución Weibull bivariada, en caso se desee modelar los tiempos de infección y síntomas directamente y de manera conjunta.

3. En el caso de las cópulas, se pueden utilizar diferentes cópulas arquimedianas para la estimación de la distribución conjunta. Inclusive, el uso de cópulas no paramétricas podrían generar mejoras en la estimación.

4. Se pueden implementar librerías en R que efectúen la estimación conjunta de los tiempos de infección y síntomas para datos censurados con los utilizados en el presente trabajo.

5. Es posible incorporar el modelo de splines para mejorar la precisión de los modelos propuestos, en especial para casos en los que se tiene un alto número de coeficientes.

6. Se puede utilizar la función de verosimilitud planteada y efectuar la estimación de forma no paramétrica, por ejemplo, con el modelo de riesgos proporcionales de Cox.

Apéndice A

Código R

```
generateCovariatesSample = function(numberCovariates,n){
  #Generating random normal covariates
  mean=3
  sd=0.5
  X=matrix(c(rep(1,n)),nrow = n,ncol = 1)
  for (i in 1:(numberCovariates)){
    X=cbind(X,matrix(rnorm(n = n,mean = mean,sd = sd),nrow = n,ncol = 1))
  }
  return(X)
}

generateBEVDSimulation=function(sigmaI,sigmaS,r,beta,sameCoefficients,n,X){
  simulationColumns=c("tiempoCensura","deltaI","deltaS")
  simulation=data.frame(matrix(nrow = n, ncol = length(simulationColumns)))
  colnames(simulation)=simulationColumns
  library(evd)
  errorSimulated=rbvevd(n = n,dep = r,model = "log",mar1 = c(0,1,0),mar2 = c(0,1,0))

  if (sameCoefficients){
    TI = exp((X %*% beta)+sigmaI*errorSimulated[,1])
    TS = exp((X %*% beta)+sigmaS*errorSimulated[,2])

  } else {
    betaInfection=beta[,1]
    betaSymptom=beta[,2]
    TI = exp((X %*% betaInfection)+sigmaI*errorSimulated[,1])
    TS = exp((X %*% betaSymptom)+sigmaS*errorSimulated[,2])
  }

  #Simulacion del tiempo de censura: Tiempo de censura~Uniforme(cuantil 0.25,cuantil 0.75)
  simulation$tiempoCensura=runif(n,min = min(c(quantile(TI)[2],quantile(TS)[2])),max = max(c(quantile(TI)[4],
       quantile(TS)[4])))

  #Generacion de los indicadores de censura (0) u ocurrencia del evento (1)
  simulation$deltaI=as.numeric(TI<=simulation$tiempoCensura)
  simulation$deltaS=as.numeric(TS<=simulation$tiempoCensura)
  return (simulation)
}

generateCopulaSimulation=function(sigmaI,sigmaS,theta,beta,sameCoefficients,n,X){
  simulationColumns=c("tiempoCensura","deltaI","deltaS")
  simulation=data.frame(matrix(nrow = n, ncol = length(simulationColumns)))
  colnames(simulation)=simulationColumns
  library(copula)
  cumulatedProbabilities=rCopula(n = n,copula = gumbelCopula(theta,dim = 2))
  errorI=log(-log(1-cumulatedProbabilities[,1]))
```

```
  errorS=log(-log(1-cumulatedProbabilities[,2]))

  if (sameCoefficients){
    TI = exp(X %*% beta+sigmaI*errorI)
    TS = exp(X %*% beta+sigmaS*errorS)
  } else {
    betaInfection=beta[,1]
    betaSymptom=beta[,2]
    TI = exp(X %*% betaInfection+sigmaI*errorI)
    TS = exp(X %*% betaSymptom+sigmaS*errorS)
  }

  #Simulacion del tiempo de censura: Tiempo de censura~Uniforme(cuantil 0.25,cuantil 0.75)
  simulation$tiempoCensura=runif(n,min = min(c(quantile(TI)[2],quantile(TS)[2])),max = max(c(quantile(TI)[4],
       quantile(TS)[4])))

  #Generacion de los indicadores de censura (0) u ocurrencia del evento (1)
  simulation$deltaI=as.numeric(TI<=simulation$tiempoCensura)
  simulation$deltaS=as.numeric(TS<=simulation$tiempoCensura)
  return (simulation)
}

#Cabe mencionar que TI y TS son variables latentes desconocidas. Los datos conocidos son:
#1) el tiempo a censura, 2) el indicador de censura de la primera variable, y 3) el indicador de
#censura de la segunda variable. Estas tres variables, junto con las covariables, entraran a las
#funciones de verosimilitud

#Bivariate Extreme Value Distribution Loglikelihood
#Parameters:
##1=sigmaI
##2=sigmaS
##3=r
##4=beta
BEVDlogLikelihood = function(parameters,tiempoCensura,deltaI,deltaS,X,sameCoefficients){
  library(evd)
  #A fin de evitar inconvenientes con las restricciones en los valores de los parametros,
  #se utilizaran transformaciones de tal manera que el valor del argumento no tenga restricciones
  #pero la transformacion si cumpla con las restricciones

  #Para sigma: como sigma debe ser mayor que cero, entonces se aplicara la funcion exponencial
  #al argumento
  sigmaI=exp(parameters[1])
  sigmaS=exp(parameters[2])
  #Para r: como r debe estar entre cero y uno, entonces se aplicara la inversa de la funcion
  #logit
  r=exp(parameters[3])/(1+exp(parameters[3]))
  #betas
  if (sameCoefficients){
    beta=matrix(parameters[4:(4+ncol(X)-1)],nrow = ncol(X),ncol = 1)
    y=log(tiempoCensura)-(X %*% beta)
    cumulatedMarginalI=pgumbel(q = y,loc = 0,scale = sigmaI)
    cumulatedMarginalS=pgumbel(q = y,loc = 0,scale = sigmaS)
    biY=matrix(c(y,y),nrow = length(y),ncol = 2)
  } else {
    betaInfection=matrix(parameters[4:(4+ncol(X)-1)],nrow = ncol(X),ncol = 1)
    betaSymptom=matrix(parameters[(4+ncol(X)):(4+2*ncol(X)-1)],nrow = ncol(X),ncol = 1)
    yI=log(tiempoCensura)-(X %*% betaInfection)
    yS=log(tiempoCensura)-(X %*% betaSymptom)
    cumulatedMarginalI=pgumbel(q = yI,loc = 0,scale = sigmaI)
    cumulatedMarginalS=pgumbel(q = yS,loc = 0,scale = sigmaS)
    biY=matrix(c(yI,yS),nrow = length(yI),ncol = 2)
  }

  jointDistribution=pbvevd(q = biY,dep = r,model = "log",mar1 = c(0,sigmaI,0),mar2 = c(0,sigmaS,0))
  -sum(deltaI*deltaS*log(jointDistribution) +
         (1-deltaI)*deltaS*log(cumulatedMarginalS-jointDistribution) +
```

```
        deltaI*(1-deltaS)*log(cumulatedMarginalI-jointDistribution) +
        (1-deltaI)*(1-deltaS)*log(1-cumulatedMarginalS-cumulatedMarginalI+jointDistribution)
    )
}

#Copula Loglikelihood
#Parameters:
##1=sigmaI
##2=sigmaS
##3=theta
##4=beta
copulaLogLikelihood = function(parameters,tiempoCensura,deltaI,deltaS,X,sameCoefficients){
  library(copula)
  sigmaI=parameters[1]
  sigmaS=parameters[2]
  theta=parameters[3]

  if (sameCoefficients){
    beta=matrix(parameters[4:(4+ncol(X)-1)],nrow = ncol(X),ncol = 1)
    marginalI=pweibull(q = tiempoCensura,shape = 1/sigmaI,scale = exp(X %*% beta)) #exponente positivo porque R
        define el parametro de escala como 1/lambda
    marginalS=pweibull(q = tiempoCensura,shape = 1/sigmaS,scale = exp(X %*% beta)) #exponente positivo porque R
        define el parametro de escala como 1/lambda
  } else {
    betaInfection=matrix(parameters[4:(4+ncol(X)-1)],nrow = ncol(X),ncol = 1)
    betaSymptom=matrix(parameters[(4+ncol(X)):(4+2*ncol(X)-1)],nrow = ncol(X),ncol = 1)
    marginalI=pweibull(q = tiempoCensura,shape = 1/sigmaI,scale = exp(X %*% betaInfection)) #exponente positivo
        porque R define el parametro de escala como 1/lambda
    marginalS=pweibull(q = tiempoCensura,shape = 1/sigmaS,scale = exp(X %*% betaSymptom)) #exponente positivo
        porque R define el parametro de escala como 1/lambda
  }

  jointProbabilities=matrix(c(marginalI,marginalS),nrow = length(marginalI),ncol = 2)
  copula=pCopula(u = jointProbabilities,copula = gumbelCopula(theta,dim = 2))
  -sum(deltaI*deltaS*log(copula) +
        (1-deltaI)*deltaS*log(marginalS-copula) +
        deltaI*(1-deltaS)*log(marginalI-copula) +
        (1-deltaI)*(1-deltaS)*log(1-marginalS-marginalI+copula)
  )
}

#Likelihood function without dependence parameter
BEVDNoDependenceParameterlogLikelihood = function(parameters,tiempoCensura,deltaI,deltaS,X,sameCoefficients,r){
  library(evd)
  #Since sigma must be positive, exponential function will be applied to the argument
  sigmaI=exp(parameters[1])
  sigmaS=exp(parameters[2])
  #betas
  if (sameCoefficients){
    beta=matrix(parameters[3:(3+ncol(X)-1)],nrow = ncol(X),ncol = 1)
    y=log(tiempoCensura)-(X %*% beta)
    cumulatedMarginalI=pgumbel(q = y,loc = 0,scale = sigmaI)
    cumulatedMarginalS=pgumbel(q = y,loc = 0,scale = sigmaS)
    biY=matrix(c(y,y),nrow = length(y),ncol = 2)
  } else {
    betaInfection=matrix(parameters[3:(3+ncol(X)-1)],nrow = ncol(X),ncol = 1)
    betaSymptom=matrix(parameters[(3+ncol(X)):(3+2*ncol(X)-1)],nrow = ncol(X),ncol = 1)
    yI=log(tiempoCensura)-(X %*% betaInfection)
    yS=log(tiempoCensura)-(X %*% betaSymptom)
    cumulatedMarginalI=pgumbel(q = yI,loc = 0,scale = sigmaI)
    cumulatedMarginalS=pgumbel(q = yS,loc = 0,scale = sigmaS)
    biY=matrix(c(yI,yS),nrow = length(yI),ncol = 2)
  }

  jointDistribution=pbvevd(q = biY,dep = r,model = "log",mar1 = c(0,sigmaI,0),mar2 = c(0,sigmaS,0))
  -sum(deltaI*deltaS*log(jointDistribution) +
```

```
        (1-deltaI)*deltaS*log(cumulatedMarginalS-jointDistribution) +
        deltaI*(1-deltaS)*log(cumulatedMarginalI-jointDistribution) +
        (1-deltaI)*(1-deltaS)*log(1-cumulatedMarginalS-cumulatedMarginalI+jointDistribution)
  )
}

copulaNoDependenceParameterLogLikelihood = function(parameters,tiempoCensura,deltaI,deltaS,X,sameCoefficients,
    theta){
  library(copula)
  sigmaI=parameters[1]
  sigmaS=parameters[2]

  if (sameCoefficients){
    beta=matrix(parameters[3:(3+ncol(X)-1)],nrow = ncol(X),ncol = 1)
    marginalI=pweibull(q = tiempoCensura,shape = 1/sigmaI,scale = exp(X %*% beta)) #exponente positivo porque R
        define el parametro de escala como 1/lambda
    marginalS=pweibull(q = tiempoCensura,shape = 1/sigmaS,scale = exp(X %*% beta)) #exponente positivo porque R
        define el parametro de escala como 1/lambda
  } else {
    betaInfection=matrix(parameters[3:(3+ncol(X)-1)],nrow = ncol(X),ncol = 1)
    betaSymptom=matrix(parameters[(3+ncol(X)):(3+2*ncol(X)-1)],nrow = ncol(X),ncol = 1)
    marginalI=pweibull(q = tiempoCensura,shape = 1/sigmaI,scale = exp(X %*% betaInfection)) #exponente positivo
        porque R define el parametro de escala como 1/lambda
    marginalS=pweibull(q = tiempoCensura,shape = 1/sigmaS,scale = exp(X %*% betaSymptom)) #exponente positivo
        porque R define el parametro de escala como 1/lambda
  }

  jointProbabilities=matrix(c(marginalI,marginalS),nrow = length(marginalI),ncol = 2)
  copula=pCopula(u = jointProbabilities,copula = gumbelCopula(theta,dim = 2))
  -sum(deltaI*deltaS*log(copula) +
        (1-deltaI)*deltaS*log(marginalS-copula) +
        deltaI*(1-deltaS)*log(marginalI-copula) +
        (1-deltaI)*(1-deltaS)*log(1-marginalS-marginalI+copula)
  )
}

bootstrapSimulations=function(B,logLikelihood,X,simulation,model,z,initialValues,
                              sameCoefficients,modelFlag,dependenceParameter){
  numberParameters=length(model$par)
  parametersEstimations = matrix(rep(0,B*numberParameters),nrow = B,
                                 ncol = numberParameters)
  for(i in 1:B){
    sampleIndex<-sample(1:nrow(X),nrow(X),replace=T)
    sampleX=X[sampleIndex,]
    sampleTiempoCensura=simulation$tiempoCensura[sampleIndex]
    sampleDeltaI=simulation$deltaI[sampleIndex]
    sampleDeltaS=simulation$deltaS[sampleIndex]
    tryCatch ({
      if (is.null(dependenceParameter)) {
        modelBootstrapping=optim(par = initialValues,fn = logLikelihood,
                                 tiempoCensura=sampleTiempoCensura,
                                 deltaI=sampleDeltaI,
                                 deltaS=sampleDeltaS,
                                 X=sampleX,
                                 sameCoefficients=sameCoefficients,
                                 control=list(maxit=10000,trace=0)
        )
      } else {
        if (modelFlag==BIVARIATE_EXTREME_VALUE_DISTRIBUTION) {
          modelBootstrapping=optim(par = initialValues,fn = logLikelihood,
                                   tiempoCensura=sampleTiempoCensura,
                                   deltaI=sampleDeltaI,
                                   deltaS=sampleDeltaS,
                                   X=sampleX,
                                   sameCoefficients=sameCoefficients,
                                   r=dependenceParameter,
```

```
                                control=list(maxit=10000,trace=0)
        )
      } else {
        modelBootstrapping=optim(par = initialValues,fn = logLikelihood,
                                tiempoCensura=sampleTiempoCensura,
                                deltaI=sampleDeltaI,
                                deltaS=sampleDeltaS,
                                X=sampleX,
                                sameCoefficients=sameCoefficients,
                                theta=dependenceParameter,
                                control=list(maxit=10000,trace=0)
        )
      }
    }
    parametersEstimations[i,] = modelBootstrapping$par
  },
  error=function(cond){
    i=i-1
  }
  )
 }
 z=pnorm(z)
 confidenceInterval=data.frame(
                   LI=apply(parametersEstimations, MARGIN = 2 , FUN = quantile, probs = 1-z),
                   LS=apply(parametersEstimations, MARGIN = 2 , FUN = quantile, probs = z)
                 )
 return(confidenceInterval)
}

generateIntervalEstimation=function(realParameters,model,simulation,numberCoefficients,X,
                                sameCoefficients,logLikelihood,modelFlag,initialValues,
                                alpha,B,bootstrapInterval,optionalBootrstrapInterval,dependenceParameter){

 #95% confidence interval
 #library(numDeriv)
 library(pracma)
 z=qnorm(1-alpha/2)
 if (is.null(dependenceParameter)){
   secondDerivativeMatrix=hessian(logLikelihood,model$par,
                                tiempoCensura=simulation$tiempoCensura,
                                deltaI=simulation$deltaI,
                                deltaS=simulation$deltaS,
                                sameCoefficients=sameCoefficients,
                                X=X)
 } else {
   if (modelFlag==BIVARIATE_EXTREME_VALUE_DISTRIBUTION) {
     secondDerivativeMatrix=hessian(logLikelihood,model$par,
                                  tiempoCensura=simulation$tiempoCensura,
                                  deltaI=simulation$deltaI,
                                  deltaS=simulation$deltaS,
                                  sameCoefficients=sameCoefficients,
                                  X=X,
                                  r=dependenceParameter)
   } else {
     secondDerivativeMatrix=hessian(logLikelihood,model$par,
                                  tiempoCensura=simulation$tiempoCensura,
                                  deltaI=simulation$deltaI,
                                  deltaS=simulation$deltaS,
                                  sameCoefficients=sameCoefficients,
                                  X=X,
                                  theta=dependenceParameter)
   }
 }

 V.hat=matrix(nrow = length(realParameters),ncol = length(realParameters))
 tryCatch ({
```

```
  V.hat=solve(secondDerivativeMatrix)
},
error=function(cond){
}
)

NAHessianInterval=any(is.na(V.hat)) #inverse matrix has any NA element?

#bootstrap conffidence interval
if (bootstrapInterval | (optionalBootrstrapInterval & NAHessianInterval)){
  bootstrapConfidenceInteval=bootstrapSimulations(B,logLikelihood,X,simulation,model,z,
                                                  initialValues,sameCoefficients,modelFlag,dependenceParameter)
} else {
  bootstrapConfidenceInteval=data.frame(LI=rep(NA,length(realParameters)),
                                        LS=rep(NA,length(realParameters))
                                        )
}

#Row labels
rowLabels=c("sigmaI","sigmaS")

if (is.null(dependenceParameter)) {
  if (modelFlag==BIVARIATE_EXTREME_VALUE_DISTRIBUTION) {
    rowLabels=c(rowLabels,"r")
  } else {
    rowLabels=c(rowLabels,"theta")
  }
}

if (sameCoefficients){
  for (i in 1:numberCoefficients) {
    rowLabels=c(rowLabels,paste("beta_",i-1,sep=""))
  }
} else {
  for (i in 1:(numberCoefficients/2)) {
    rowLabels=c(rowLabels,paste("beta_",i-1,"_I",sep=""))
  }
  for (i in 1:(numberCoefficients/2)) {
    rowLabels=c(rowLabels,paste("beta_",i-1,"_S",sep=""))
  }
}

#Resulting data frame
if (is.null(dependenceParameter)) {
  if (modelFlag==BIVARIATE_EXTREME_VALUE_DISTRIBUTION) {
    results=data.frame(
      parametro=rowLabels,
      valorReal=realParameters,
      EMV=c(exp(model$par[1]),exp(model$par[2]),exp(model$par[3])/(1+exp(model$par[3])),
            model$par[4:(numberCoefficients+3)]),
      LI=c(exp(model$par[1]-z*sqrt(diag(V.hat)[1])),
           exp(model$par[2]-z*sqrt(diag(V.hat)[2])),
           exp(model$par[3]-z*sqrt(diag(V.hat)[3]))/(1+exp(model$par[3]-z*sqrt(diag(V.hat)[3]))),
           model$par[4:(numberCoefficients+3)]-z*sqrt(diag(V.hat)[4:(numberCoefficients+3)])),
      LS=c(exp(model$par[1]+z*sqrt(diag(V.hat)[1])),
           exp(model$par[2]+z*sqrt(diag(V.hat)[2])),
           exp(model$par[3]+z*sqrt(diag(V.hat)[3]))/(1+exp(model$par[3]+z*sqrt(diag(V.hat)[3]))),
           model$par[4:(numberCoefficients+3)]+z*sqrt(diag(V.hat)[4:(numberCoefficients+3)])),
      bootstrapLI=c(exp(bootstrapConfidenceInteval$LI[1]),exp(bootstrapConfidenceInteval$LI[2]),
                    exp(bootstrapConfidenceInteval$LI[3])/(1+exp(bootstrapConfidenceInteval$LI[3])),
                    bootstrapConfidenceInteval$LI[4:(numberCoefficients+3)]),
      boostrapLS=c(exp(bootstrapConfidenceInteval$LS[1]),exp(bootstrapConfidenceInteval$LS[2]),
                   exp(bootstrapConfidenceInteval$LS[3])/(1+exp(bootstrapConfidenceInteval$LS[3])),
                   bootstrapConfidenceInteval$LS[4:(numberCoefficients+3)])
    )
  } else {
```

```
      results=data.frame(
        parametro=rowLabels,
        valorReal=realParameters,
        EMV=model$par,
        LI=c(model$par[1]-z*sqrt(diag(V.hat)[1]),
             model$par[2]-z*sqrt(diag(V.hat)[2]),
             model$par[3]-z*sqrt(diag(V.hat)[3]),
             model$par[4:(numberCoefficients+3)]-z*sqrt(diag(V.hat)[4:(numberCoefficients+3)])),
        LS=c(model$par[1]+z*sqrt(diag(V.hat)[1]),
             model$par[2]+z*sqrt(diag(V.hat)[2]),
             model$par[3]+z*sqrt(diag(V.hat)[3]),
             model$par[4:(numberCoefficients+3)]+z*sqrt(diag(V.hat)[4:(numberCoefficients+3)])),
        bootstrapLI=bootstrapConfidenceInteval$LI,
        boostrapLS=bootstrapConfidenceInteval$LS
      )
    }
  } else {
    if (modelFlag==BIVARIATE_EXTREME_VALUE_DISTRIBUTION) {
      results=data.frame(
        parametro=rowLabels,
        valorReal=realParameters,
        EMV=c(exp(model$par[1]),exp(model$par[2]),model$par[3:(numberCoefficients+2)]),
        LI=c(exp(model$par[1]-z*sqrt(diag(V.hat)[1])),
             exp(model$par[2]-z*sqrt(diag(V.hat)[2])),
             model$par[3:(numberCoefficients+2)]-z*sqrt(diag(V.hat)[3:(numberCoefficients+2)])),
        LS=c(exp(model$par[1]+z*sqrt(diag(V.hat)[1])),
             exp(model$par[2]+z*sqrt(diag(V.hat)[2])),
             model$par[3:(numberCoefficients+2)]+z*sqrt(diag(V.hat)[3:(numberCoefficients+2)])),
        bootstrapLI=c(exp(bootstrapConfidenceInteval$LI[1]),exp(bootstrapConfidenceInteval$LI[2]),
                      bootstrapConfidenceInteval$LI[3:(numberCoefficients+2)]),
        boostrapLS=c(exp(bootstrapConfidenceInteval$LS[1]),exp(bootstrapConfidenceInteval$LS[2]),
                     bootstrapConfidenceInteval$LS[3:(numberCoefficients+2)])
        )
    } else {
      results=data.frame(
        parametro=rowLabels,
        valorReal=realParameters,
        EMV=model$par,
        LI=c(model$par[1]-z*sqrt(diag(V.hat)[1]),
             model$par[2]-z*sqrt(diag(V.hat)[2]),
             model$par[3:(numberCoefficients+2)]-z*sqrt(diag(V.hat)[3:(numberCoefficients+2)])),
        LS=c(model$par[1]+z*sqrt(diag(V.hat)[1]),
             model$par[2]+z*sqrt(diag(V.hat)[2]),
             model$par[3:(numberCoefficients+2)]+z*sqrt(diag(V.hat)[3:(numberCoefficients+2)])),
        bootstrapLI=bootstrapConfidenceInteval$LI,
        boostrapLS=bootstrapConfidenceInteval$LS
      )
    }
  }
  results$IsInInterval=results$valorReal>=results$LI & results$valorReal<=results$LS
  results$IsSignificant=!(results$LI<0 & results$LS>0)
  results$IsInBootstrapInterval=results$valorReal>=results$bootstrapLI & results$valorReal<=results$boostrapLS
  results$IsBootstrapSignificant=!(results$bootstrapLI<0 & results$boostrapLS>0)

  return(list("estimation"=results,
              "inverseMatrix"=V.hat))

}

simulateAndEstimate=function(sigmaI,sigmaS,r,theta,sameCoefficients,beta,n,seed,initialValues,
                             modelFlag,alpha,B,bootstrapInterval,optionalBootrstrapInterval,dependenceParameter){
  set.seed(seed)

  numberCoefficients=length(beta)
  if (sameCoefficients) {
    numberCovariates=numberCoefficients-1
```

```
  if (modelFlag==BIVARIATE_EXTREME_VALUE_DISTRIBUTION){
    if (is.null(dependenceParameter)){
      realParameters=c(sigmaI,sigmaS,r,c(beta))
    } else {
      realParameters=c(sigmaI,sigmaS,c(beta))
    }
  } else {
    if (is.null(dependenceParameter)){
      realParameters=c(sigmaI,sigmaS,theta,c(beta))
    } else {
      realParameters=c(sigmaI,sigmaS,c(beta))
    }
  }
} else {
  numberCovariates=numberCoefficients/2-1
  if (modelFlag==BIVARIATE_EXTREME_VALUE_DISTRIBUTION){
    if (is.null(dependenceParameter)){
      realParameters=c(sigmaI,sigmaS,r,c(beta[,1]),c(beta[,2]))
    } else {
      realParameters=c(sigmaI,sigmaS,c(beta[,1]),c(beta[,2]))
    }
  } else {
    if (is.null(dependenceParameter)){
      realParameters=c(sigmaI,sigmaS,theta,c(beta[,1]),c(beta[,2]))
    } else {
      realParameters=c(sigmaI,sigmaS,c(beta[,1]),c(beta[,2]))
    }
  }
}

#Generating simulation
X=generateCovariatesSample(numberCovariates,n)
if (modelFlag==BIVARIATE_EXTREME_VALUE_DISTRIBUTION){
  simulation=generateBEVDSimulation(sigmaI,sigmaS,r,beta,sameCoefficients,n,X)
  if (is.null(dependenceParameter)){
    model=optim(par = initialValues,fn = BEVDlogLikelihood,
                tiempoCensura=simulation$tiempoCensura,
                deltaI=simulation$deltaI,
                deltaS=simulation$deltaS,
                X=X,
                sameCoefficients=sameCoefficients,
                control=list(trace=0,maxit=10000)
                )
    estimation=generateIntervalEstimation(realParameters,model,simulation,numberCoefficients,X,
                                          sameCoefficients,BEVDlogLikelihood,modelFlag,
                                          initialValues,alpha,B,bootstrapInterval,optionalBootrstrapInterval,
                                               dependenceParameter)
  } else {
    model=optim(par = initialValues,fn = BEVDNoDependenceParameterlogLikelihood,
                tiempoCensura=simulation$tiempoCensura,
                deltaI=simulation$deltaI,
                deltaS=simulation$deltaS,
                X=X,
                sameCoefficients=sameCoefficients,
                r=dependenceParameter,
                control=list(trace=0,maxit=10000)
                )
    estimation=generateIntervalEstimation(realParameters,model,simulation,numberCoefficients,X,
                                          sameCoefficients,BEVDNoDependenceParameterlogLikelihood,modelFlag,
                                          initialValues,alpha,B,bootstrapInterval,optionalBootrstrapInterval,
                                               dependenceParameter)
  }
} else {
  simulation=generateCopulaSimulation(sigmaI,sigmaS,theta,beta,sameCoefficients,n,X)
  if (is.null(dependenceParameter)){
    model=optim(par = initialValues,fn = copulaLogLikelihood,
```

```
                tiempoCensura=simulation$tiempoCensura,
                deltaI=simulation$deltaI,
                deltaS=simulation$deltaS,
                X=X,
                sameCoefficients=sameCoefficients,
                control=list(trace=0,maxit=10000)
                )
    estimation=generateIntervalEstimation(realParameters,model,simulation,numberCoefficients,X,
                                          sameCoefficients,copulaLogLikelihood,modelFlag,
                                          initialValues,alpha,B,bootstrapInterval,optionalBootrstrapInterval,
                                               dependenceParameter)
  } else {
    model=optim(par = initialValues,fn = copulaNoDependenceParameterLogLikelihood,
                tiempoCensura=simulation$tiempoCensura,
                deltaI=simulation$deltaI,
                deltaS=simulation$deltaS,
                X=X,
                sameCoefficients=sameCoefficients,
                theta=dependenceParameter,
                control=list(trace=0,maxit=10000)
                )
    estimation=generateIntervalEstimation(realParameters,model,simulation,numberCoefficients,X,
                                          sameCoefficients,copulaNoDependenceParameterLogLikelihood,modelFlag,
                                          initialValues,alpha,B,bootstrapInterval,optionalBootrstrapInterval,
                                               dependenceParameter)
  }

}

results=list("X"=X,
             "simulation"=simulation,
             "estimation"=estimation$estimation,
             "inverseMatrix"=estimation$inverseMatrix
             )
return (results)
}

runSimulation=function(sigmaI,sigmaS,r,theta,sameCoefficients,beta,n,seed,modelFlag,alpha,B,bootstrapInterval,
                       optionalBootrstrapInterval,dependenceParameter){
#Initial values to optimization
if (modelFlag==BIVARIATE_EXTREME_VALUE_DISTRIBUTION){
  INITIAL_SIGMA_I=-1
  INITIAL_SIGMA_S=-1
  INITIAL_R=1
} else {
  INITIAL_SIGMA_I=1
  INITIAL_SIGMA_S=1
  INITIAL_THETA=1.5
}

if (sameCoefficients){
  INITIAL_BETA=rep(0,nrow(beta))
  if (is.null(dependenceParameter)) {
    if (modelFlag==BIVARIATE_EXTREME_VALUE_DISTRIBUTION){
      initialValues=c(INITIAL_SIGMA_I,INITIAL_SIGMA_S,INITIAL_R,INITIAL_BETA)
    } else {
      initialValues=c(INITIAL_SIGMA_I,INITIAL_SIGMA_S,INITIAL_THETA,INITIAL_BETA)
    }
  } else {
    initialValues=c(INITIAL_SIGMA_I,INITIAL_SIGMA_S,INITIAL_BETA)
  }
} else {
  INITIAL_BETA_IS=rep(0,2*nrow(beta))
  if (is.null(dependenceParameter)) {
    if (modelFlag==BIVARIATE_EXTREME_VALUE_DISTRIBUTION){
      initialValues=c(INITIAL_SIGMA_I,INITIAL_SIGMA_S,INITIAL_R,INITIAL_BETA_IS)
```

```
      } else {
        initialValues=c(INITIAL_SIGMA_I,INITIAL_SIGMA_S,INITIAL_THETA,INITIAL_BETA_IS)
      }
    } else {
      initialValues=c(INITIAL_SIGMA_I,INITIAL_SIGMA_S,INITIAL_BETA_IS)
    }
  }

  #Estimations
  resultsSimulation=simulateAndEstimate(sigmaI,sigmaS,r,theta,sameCoefficients,beta,n,seed,
                                        initialValues,modelFlag,alpha,B,bootstrapInterval,
                                             optionalBootrstrapInterval,
                                        dependenceParameter)
  return(resultsSimulation)
}

#Simulations
#Parameters
#Sample size
n=1000
#Seed
seed=1
#Shape and sigma (1/shape) parameters
shapeI=2
sigmaI=1/shapeI
shapeS=1.5
sigmaS=1/shapeS
#Bivariate extreme value distribution dependence parameter r (0<r<1)
r=0.5
#Gumbel copula parameter theta (theta>=1)
theta=2

#Significance level
alpha=0.05

#Bootstrap indicator and bootstrap simulations number
bootstrapInterval=TRUE #mandatory calculate of bootstrap interval
optionalBootrstrapInterval=TRUE #calculate bootrstrap interval only if the hessian interval can not be calculated
B=1000

#Simulations

#Flag global constants
BIVARIATE_EXTREME_VALUE_DISTRIBUTION=1
COPULA=2

#Simulations
initialTime=Sys.time()
#1 covariate
##Same coefficients
beta=matrix(c(-0.5,0.5),ncol = 1)
sameCoefficients=TRUE
BEVDresults1CovariateSameCoefficients=runSimulation(sigmaI,sigmaS,r,theta,sameCoefficients,beta,n,seed,
                                                    BIVARIATE_EXTREME_VALUE_DISTRIBUTION,alpha,B,bootstrapInterval
                                                         ,
                                                    optionalBootrstrapInterval)
copulaResults1CovariateSameCoefficients=runSimulation(sigmaI,sigmaS,r,theta,sameCoefficients,beta,n,seed,COPULA,
     alpha,B,
                                                      bootstrapInterval,optionalBootrstrapInterval)
##Different coefficients
betaInfection=c(-0.3,0.3)
betaSymptom=c(-0.6,0.6)
beta=matrix(c(betaInfection,betaSymptom),nrow = length(betaInfection),ncol = 2)
sameCoefficients=FALSE
BEVDresults1CovariateDifferentCoefficients=runSimulation(sigmaI,sigmaS,r,theta,sameCoefficients,beta,n,seed,
```

```
                                                    BIVARIATE_EXTREME_VALUE_DISTRIBUTION,alpha,B,
                                                         bootstrapInterval,
                                                    optionalBootrstrapInterval)
copulaResults1CovariateDifferentCoefficients=runSimulation(sigmaI,sigmaS,r,theta,sameCoefficients,beta,n,seed,
     COPULA,
                                                      alpha,B,bootstrapInterval,optionalBootrstrapInterval)

#2 covariates
##Same coefficients
beta=matrix(c(-0.5,0.5,-0.5),ncol = 1)
sameCoefficients=TRUE
BEVDresults2CovariatesSameCoefficients=runSimulation(sigmaI,sigmaS,r,theta,sameCoefficients,beta,n,seed,
                                                   BIVARIATE_EXTREME_VALUE_DISTRIBUTION,alpha,B,
                                                        bootstrapInterval,
                                                   optionalBootrstrapInterval)
copulaResults2CovariatesSameCoefficients=runSimulation(sigmaI,sigmaS,r,theta,sameCoefficients,beta,n,seed,COPULA,
     alpha,B,
                                                     bootstrapInterval,optionalBootrstrapInterval)
##Different coefficients
betaInfection=c(-0.3,0.3,-0.3)
betaSymptom=c(-0.6,0.6,-0.6)
beta=matrix(c(betaInfection,betaSymptom),nrow = length(betaInfection),ncol = 2)
sameCoefficients=FALSE
BEVDresults2CovariatesDifferentCoefficients=runSimulation(sigmaI,sigmaS,r,theta,sameCoefficients,beta,n,seed,
                                                    BIVARIATE_EXTREME_VALUE_DISTRIBUTION,alpha,B,
                                                         bootstrapInterval,
                                                    optionalBootrstrapInterval)
copulaResults2CovariatesDifferentCoefficients=runSimulation(sigmaI,sigmaS,r,theta,sameCoefficients,beta,n,seed,
     COPULA,
                                                      alpha,B,bootstrapInterval,optionalBootrstrapInterval)

#3 covariates
##Same coefficients
beta=matrix(c(-0.5,0.5,-0.5,0.5),ncol = 1)
sameCoefficients=TRUE
BEVDresults3CovariatesSameCoefficients=runSimulation(sigmaI,sigmaS,r,theta,sameCoefficients,beta,n,seed,
                                                   BIVARIATE_EXTREME_VALUE_DISTRIBUTION,alpha,B,
                                                        bootstrapInterval,
                                                   optionalBootrstrapInterval)
copulaResults3CovariatesSameCoefficients=runSimulation(sigmaI,sigmaS,r,theta,sameCoefficients,beta,n,seed,COPULA,
     alpha,B,
                                                     bootstrapInterval,optionalBootrstrapInterval)
##Different coefficients
betaInfection=c(-0.3,0.3,-0.3,0.3)
betaSymptom=c(-0.6,0.6,-0.6,0.6)
beta=matrix(c(betaInfection,betaSymptom),nrow = length(betaInfection),ncol = 2)
sameCoefficients=FALSE
BEVDresults3CovariatesDifferentCoefficients=runSimulation(sigmaI,sigmaS,r,theta,sameCoefficients,beta,n,seed,
                                                    BIVARIATE_EXTREME_VALUE_DISTRIBUTION,alpha,B,
                                                         bootstrapInterval,
                                                    optionalBootrstrapInterval)
copulaResults3CovariatesDifferentCoefficients=runSimulation(sigmaI,sigmaS,r,theta,sameCoefficients,beta,n,seed,
     COPULA,
                                                      alpha,B,bootstrapInterval,optionalBootrstrapInterval)

finalTime=Sys.time()

#Results
cat("\f") #borra la consola
#Bivariate Extreme Value Distribution
BEVDresults1CovariateSameCoefficients$estimation
BEVDresults1CovariateDifferentCoefficients$estimation
BEVDresults2CovariatesSameCoefficients$estimation
BEVDresults2CovariatesDifferentCoefficients$estimation
BEVDresults3CovariatesSameCoefficients$estimation
BEVDresults3CovariatesDifferentCoefficients$estimation
```

```
#Copula
copulaResults1CovariateSameCoefficients$estimation
copulaResults1CovariateDifferentCoefficients$estimation
copulaResults2CovariatesSameCoefficients$estimation
copulaResults2CovariatesDifferentCoefficients$estimation
copulaResults3CovariatesSameCoefficients$estimation
copulaResults3CovariatesDifferentCoefficients$estimation

#Execution times
print(initialTime)
print(finalTime)

#Charts
#plot(BEVDresults3CovariatesDifferentCoefficients$estimation$valorReal,xaxt = "n",xlab="",
#     pch="-",ylim=c(-1,1),col="blue")
#axis(1, 1:nrow(BEVDresults3CovariatesDifferentCoefficients$estimation),
#     BEVDresults3CovariatesDifferentCoefficients$estimation$parametro,las=2)
#points(BEVDresults3CovariatesDifferentCoefficients$estimation$EMV,pch=20,col="red",cex=0.8)


######################PERFORMANCE MODEL WITH MANY SIMULATIONS###################################

#Calculating model performance with many simulations of the same model with one, two and three covariates,
     assuming same
#and different coefficient and assuming dependence parameter known and unknown

#Function that calculate coverage and test power ratios
calculateRatios=function(listResultSimulations,bootstrapInterval,optionalBootrstrapInterval){
  numberSimulations=length(listResultSimulations)
  if (optionalBootrstrapInterval) {
    ratioColumns=c("Parameter","RealValue","AverageEstimation","Coverage","Bias","TestPower","NAHessianInterval")
  } else {
    ratioColumns=c("Parameter","RealValue","AverageEstimation","HessianCoverage","BootstrapCoverage","Bias",
                   "TestPowerHessian","TestPowerBootstrap","NAHessianInterval")
  }
  ratiosDataframe=data.frame(matrix(nrow = nrow(listResultSimulations[[1]]$estimation),
                                    ncol =length(ratioColumns)))
  colnames(ratiosDataframe)=ratioColumns
  ratiosDataframe$Parameter=listResultSimulations[[1]]$estimation$parametro

  for (i in 1:nrow(ratiosDataframe)){
    parameterRealValue=listResultSimulations[[1]]$estimation$valorReal[i]
    numberHessianCoverage=0
    numberBootstrapCoverage=0
    sumEstimations=0
    numberSignificantIntervalsHessian=0
    numberSignificantIntervalsBoostrap=0
    numberNAHessianIntervals=0
    numberCoverageHessianAndBootstrap=0
    numberSignificantIntervalsHessianAndBootstrap=0

    for (j in 1:numberSimulations){
      parameterEstimation=listResultSimulations[[j]]$estimation[i,]
      if (!is.na(parameterEstimation$LI)) {
        if (parameterEstimation$LI<=parameterEstimation$valorReal &
            parameterEstimation$LS>=parameterEstimation$valorReal) {
          numberHessianCoverage=numberHessianCoverage+1
          numberCoverageHessianAndBootstrap=numberCoverageHessianAndBootstrap+1
        }
      } else {
        numberNAHessianIntervals=numberNAHessianIntervals+1
        if (optionalBootrstrapInterval) {
          if (!is.na(parameterEstimation$bootstrapLI)){
            if (parameterEstimation$bootstrapLI<=parameterEstimation$valorReal &
                parameterEstimation$boostrapLS>=parameterEstimation$valorReal) {
              numberCoverageHessianAndBootstrap=numberCoverageHessianAndBootstrap+1
```

```
            }
          } else {
            print(j)
          }
        }
      }
      if (bootstrapInterval){
        if (parameterEstimation$bootstrapLI<=parameterEstimation$valorReal &
            parameterEstimation$boostrapLS>=parameterEstimation$valorReal) {
          numberBootstrapCoverage=numberBootstrapCoverage+1
        }
      }
      sumEstimations=sumEstimations+parameterEstimation$EMV
      if (!is.na(parameterEstimation$LI)) {
        if (!(parameterEstimation$LI<0 & parameterEstimation$LS>0)) {
          numberSignificantIntervalsHessian=numberSignificantIntervalsHessian+1
          numberSignificantIntervalsHessianAndBootstrap=numberSignificantIntervalsHessianAndBootstrap+1
        }
      } else {
        if (optionalBootrstrapInterval) {
          if (!is.na(parameterEstimation$bootstrapLI)){
            if (!(parameterEstimation$bootstrapLI<0 & parameterEstimation$boostrapLS>0)) {
              numberSignificantIntervalsHessianAndBootstrap=numberSignificantIntervalsHessianAndBootstrap+1
            }
          }
        }
      }
      if (bootstrapInterval){
        if (!(parameterEstimation$bootstrapLI<0 & parameterEstimation$boostrapLS>0)) {
          numberSignificantIntervalsBoostrap=numberSignificantIntervalsBoostrap+1
        }
      }
    }

    ratiosDataframe[i,"RealValue"]=parameterRealValue
    ratiosDataframe[i,"AverageEstimation"]=sumEstimations/numberSimulations
    ratiosDataframe[i,"Bias"]=(sumEstimations/numberSimulations-parameterRealValue)/parameterRealValue
    ratiosDataframe[i,"NAHessianInterval"]=numberNAHessianIntervals/numberSimulations

    if (optionalBootrstrapInterval) {
      ratiosDataframe[i,"Coverage"]=numberCoverageHessianAndBootstrap/numberSimulations
      ratiosDataframe[i,"TestPower"]=numberSignificantIntervalsHessianAndBootstrap/numberSimulations
    } else {
      if (numberNAHessianIntervals!=numberSimulations){
        ratiosDataframe[i,"HessianCoverage"]=numberHessianCoverage/numberSimulations
        ratiosDataframe[i,"TestPowerHessian"]=numberSignificantIntervalsHessian/numberSimulations
      } else {
        ratiosDataframe[i,"HessianCoverage"]=NA
        ratiosDataframe[i,"TestPowerHessian"]=NA
      }
      if (bootstrapInterval){
        ratiosDataframe[i,"BootstrapCoverage"]=numberBootstrapCoverage/numberSimulations
        ratiosDataframe[i,"TestPowerBootstrap"]=numberSignificantIntervalsBoostrap/numberSimulations
      }
    }

  }

  return(ratiosDataframe)
}

simulateAndRatios=function(sigmaI,sigmaS,r,theta,sameCoefficients,beta,n,seed,
                           alpha,B,BEVDbootstrapInterval,optionalBootrstrapInterval,knownDependenceParameter) {

  if (knownDependenceParameter==FALSE){
    dependenceParameter=NULL
```

```
  }

  if (knownDependenceParameter==TRUE){
    dependenceParameter=r
  }
  newBEVDsimulation=runSimulation(sigmaI,sigmaS,r,theta,sameCoefficients,beta,n,seed,
                                  BIVARIATE_EXTREME_VALUE_DISTRIBUTION,alpha,B,BEVDbootstrapInterval,
                                  optionalBootrstrapInterval,dependenceParameter)

  if (knownDependenceParameter==TRUE){
    dependenceParameter=theta
  }
  newCopulaSimulation=runSimulation(sigmaI,sigmaS,r,theta,sameCoefficients,beta,n,seed,COPULA,
                                    alpha,B,copulabootstrapInterval,optionalBootrstrapInterval,dependenceParameter
                                         )

  BEVDmodel=list(newBEVDsimulation)
  copulaModel=list(newCopulaSimulation)
  for (i in seq(1,numberSimulations-1)){
    seed=i+1
    if (knownDependenceParameter==TRUE){
      dependenceParameter=r
    }
    newBEVDsimulation=runSimulation(sigmaI,sigmaS,r,theta,sameCoefficients,beta,n,seed,
                                    BIVARIATE_EXTREME_VALUE_DISTRIBUTION,alpha,B,BEVDbootstrapInterval,
                                    optionalBootrstrapInterval,dependenceParameter)
    if (knownDependenceParameter==TRUE){
      dependenceParameter=theta
    }
    newCopulaSimulation=runSimulation(sigmaI,sigmaS,r,theta,sameCoefficients,beta,n,seed,COPULA,
                                      alpha,B,copulabootstrapInterval,optionalBootrstrapInterval,
                                          dependenceParameter)
    BEVDmodel=c(BEVDmodel,list(newBEVDsimulation))
    copulaModel=c(copulaModel,list(newCopulaSimulation))
    print(i)
  }

  #Ratios

  BEVDratios=calculateRatios(BEVDmodel,BEVDbootstrapInterval,optionalBootrstrapInterval)
  copulaRatios=calculateRatios(copulaModel,copulabootstrapInterval,optionalBootrstrapInterval)

  results=list("BEVDmodel"=BEVDmodel,
               "copulaModel"=copulaModel,
               "BEVDratios"=BEVDratios,
               "copulaRatios"=copulaRatios
                )
  return (results)

}

numberSimulations=1000
n=1000
BEVDbootstrapInterval=FALSE
copulabootstrapInterval=FALSE
optionalBootrstrapInterval=TRUE
B=200
seed=1
alpha=0.05

initialTime=Sys.time()
print(initialTime)
betaSameCoefficient=c(-0.5)
betaInfection=c(-0.3)
betaSymptom=c(-0.6)
```

```
listKnownDependenceSameCoefficients=list()
listUnknownDependenceSameCoefficients=list()
listKnownDependenceDifferentCoefficients=list()
listUnknownDependenceDifferentCoefficients=list()

for (modelNumber in seq(1,3)){
  #Same coefficients
  sameCoefficients=TRUE
  betaSameCoefficient=c(betaSameCoefficient,-1*betaSameCoefficient[length(betaSameCoefficient)])
  beta=matrix(betaSameCoefficient,nrow = length(betaSameCoefficient),ncol = 1)
  ##Known dependence parameter
  knownDependenceParameter=TRUE
  knownDependenceSameCoefficients=simulateAndRatios(sigmaI,sigmaS,r,theta,sameCoefficients,beta,n,seed,
                                          alpha,B,BEVDbootstrapInterval,
                                          optionalBootrstrapInterval,knownDependenceParameter)
  listKnownDependenceSameCoefficients=c(listKnownDependenceSameCoefficients,list(knownDependenceSameCoefficients))
  ##Unknown dependence parameter
  knownDependenceParameter=FALSE
  unknownDependenceSameCoefficients=simulateAndRatios(sigmaI,sigmaS,r,theta,sameCoefficients,beta,n,seed,
                                          alpha,B,BEVDbootstrapInterval,
                                          optionalBootrstrapInterval,knownDependenceParameter)
  listUnknownDependenceSameCoefficients=c(listUnknownDependenceSameCoefficients,list(
       unknownDependenceSameCoefficients))

  #Different coefficients
  sameCoefficients=FALSE
  betaInfection=c(betaInfection,-1*betaInfection[length(betaInfection)])
  betaSymptom=c(betaSymptom,-1*betaSymptom[length(betaSymptom)])
  beta=matrix(c(betaInfection,betaSymptom),nrow = length(betaInfection),ncol = 2)
  ##Known dependence parameter
  knownDependenceParameter=TRUE
  knownDependenceDifferentCoefficients=simulateAndRatios(sigmaI,sigmaS,r,theta,sameCoefficients,beta,n,seed,
                                            alpha,B,BEVDbootstrapInterval,
                                            optionalBootrstrapInterval,knownDependenceParameter)
  listKnownDependenceDifferentCoefficients=c(listKnownDependenceDifferentCoefficients,list(
       knownDependenceDifferentCoefficients))
  ##Unknown dependence parameter
  knownDependenceParameter=FALSE
  unknownDependenceDifferentCoefficients=simulateAndRatios(sigmaI,sigmaS,r,theta,sameCoefficients,beta,n,seed,
                                              alpha,B,BEVDbootstrapInterval,
                                              optionalBootrstrapInterval,knownDependenceParameter)
  listUnknownDependenceDifferentCoefficients=c(listUnknownDependenceDifferentCoefficients,list(
       unknownDependenceDifferentCoefficients))
  print(modelNumber)
}

finalTime=Sys.time()

#Execution times
print(initialTime)
print(finalTime)

#####################APPLICATION############################

setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
notificacionParejas=read.csv("symptoms - peter.csv")
View(notificacionParejas)

#Analisis descriptivo de las variables
barplot(table(notificacionParejas$gender),main="Genero del paciente",xlab="0: mujer, 1: varon",
        col = "lightblue")
barplot(table(notificacionParejas$arm),main="Grupo del paciente",
        xlab="0: control, 1: intervencion", col = "lightblue")
hist(notificacionParejas$age,main="Edad del paciente",xlab="Edad",
     ylab="Frecuencia",col="lightblue")
hist(notificacionParejas$time,main="Tiempo de observacion",xlab="Dias",
```

```
     ylab="Frecuencia",col="lightblue")
barplot(table(notificacionParejas$disease),main="Infeccion",
        xlab="0: no infeccion, 1: infeccion", col = "lightblue")
barplot(table(notificacionParejas$symptoms),main="Sintomas",
        xlab="0: no sintomas, 1: sintomas", col = "lightblue")

input=notificacionParejas[,c("time","disease","symptoms")]
colnames(input)=c("tiempoCensura","deltaI","deltaS")
numberCoefficients=4
X=matrix(c(rep(1,nrow(notificacionParejas)),notificacionParejas$gender,notificacionParejas$arm,
           notificacionParejas$age),
         nrow = nrow(notificacionParejas),ncol=numberCoefficients)
alpha=0.05
B=1000
bootstrapInterval=FALSE
optionalBootrstrapInterval=TRUE


#Function that estimates parameters for the application
estimateParametersApplication=function(input,numberCoefficients,X,sameCoefficients,logLikelihood,
                                       modelFlag,initialValues,alpha,B,bootstrapInterval,
                                            optionalBootrstrapInterval,
                                       dependenceParameter){

  model=optim(par =  initialValues, fn = logLikelihood,
                      tiempoCensura=input$tiempoCensura,
                      deltaI=input$deltaI,
                      deltaS=input$deltaS,
                      X=X,
                      sameCoefficients=sameCoefficients,
                      control=list(trace=0,maxit=10000)
  )
  if (modelFlag==BIVARIATE_EXTREME_VALUE_DISTRIBUTION){
    realValue=c(exp(model$par[1]),exp(model$par[2]),exp(model$par[3])/(1+exp(model$par[3])),
          model$par[4:(numberCoefficients+3)])
  } else {
    realValue=model$par
  }

  results=generateIntervalEstimation(realValue,model,input,numberCoefficients,X,
                                     sameCoefficients,logLikelihood,modelFlag,initialValues, alpha,B,
                                          bootstrapInterval,
                                     optionalBootrstrapInterval,dependenceParameter)
  return(results)
}

#Estimation with bivariate extreme value distribution model
##Same coefficients
sameCoefficients=TRUE
numberCoefficients=4
initialValuesBEVD=c(-0.5,-0.5,1,rep(0,numberCoefficients))
BEVDlogLikelihood(initialValuesBEVD,input$tiempoCensura,input$deltaI,input$deltaS,X,sameCoefficients)
BEVDSameCoefficientsApplicationResults=estimateParametersApplication(input,numberCoefficients,X,sameCoefficients,
                                                                     BEVDlogLikelihood,
                                                                     BIVARIATE_EXTREME_VALUE_DISTRIBUTION,
                                                                     initialValuesBEVD,alpha,B,bootstrapInterval,
                                                                     optionalBootrstrapInterval,NULL)

##Different coefficients
sameCoefficients=FALSE
numberCoefficients=2*numberCoefficients
initialValuesBEVD=c(-0.5,-0.5,1,rep(0,numberCoefficients))
BEVDlogLikelihood(initialValuesBEVD,input$tiempoCensura,input$deltaI,input$deltaS,X,sameCoefficients)
BEVDDifferentCoefficientsApplicationResults=estimateParametersApplication(input,numberCoefficients,X,
     sameCoefficients,
                                                                     BEVDlogLikelihood,
```

```
                                                                    BIVARIATE_EXTREME_VALUE_DISTRIBUTION,
                                                                    initialValuesBEVD,alpha,B,bootstrapInterval,
                                                                    optionalBootrstrapInterval,NULL)

#Estimation with copula model
##Same coefficients
sameCoefficients=TRUE
numberCoefficients=4
initialValuesCopula=c(1,1,1.5,rep(0.5,numberCoefficients))
copulaLogLikelihood(initialValuesCopula,input$tiempoCensura,input$deltaI,input$deltaS,X,sameCoefficients)
copulaSameCoefficientsApplicationResults=estimateParametersApplication(input,numberCoefficients,X,sameCoefficients
    ,
                                                                    copulaLogLikelihood,
                                                                    COPULA,
                                                                    initialValuesCopula,alpha,B,bootstrapInterval
                                                                        ,
                                                                    optionalBootrstrapInterval,NULL)

##Different coefficients
sameCoefficients=FALSE
numberCoefficients=2*numberCoefficients
initialValuesCopula=c(1,1,1.5,rep(0.5,numberCoefficients))
copulaLogLikelihood(initialValuesCopula,input$tiempoCensura,input$deltaI,input$deltaS,X,sameCoefficients)
copulaDifferentCoefficientsApplicationResults=estimateParametersApplication(input,numberCoefficients,X,
    sameCoefficients,
                                                                      copulaLogLikelihood,
                                                                      COPULA,
                                                                      initialValuesCopula,alpha,B,
                                                                          bootstrapInterval,
                                                                      optionalBootrstrapInterval,NULL)

#AIC to evaluate the best fit
#Without the negative sign, because the loglikelihood function already returns the negative value
AICBivariateSameCoefficients=2*BEVDlogLikelihood(BEVDSameCoefficientsApplicationResults$estimation$EMV,input$
    tiempoCensura,
                                                input$deltaI,input$deltaS,X,TRUE)+
                                2*(length(BEVDSameCoefficientsApplicationResults$estimation$EMV)+1);
                                    AICBivariateSameCoefficients
AICBivariateDifferentCoefficients=2*BEVDlogLikelihood(BEVDDifferentCoefficientsApplicationResults$estimation$EMV,
    input$tiempoCensura,
                                                    input$deltaI,input$deltaS,X,FALSE)+
                                  2*(length(BEVDDifferentCoefficientsApplicationResults$estimation$EMV)+1);
                                       AICBivariateDifferentCoefficients
AICCopulaSameCoefficients=2*copulaLogLikelihood(copulaSameCoefficientsApplicationResults$estimation$EMV,input$
    tiempoCensura,
                                               input$deltaI,input$deltaS,X,TRUE)+
                           2*(length(copulaSameCoefficientsApplicationResults$estimation$EMV)+1);
                                AICCopulaSameCoefficients
AICCopulaDifferentCoefficients=2*copulaLogLikelihood(copulaDifferentCoefficientsApplicationResults$estimation$EMV,
    input$tiempoCensura,
                                                   input$deltaI,input$deltaS,X,FALSE)+
                                2*(length(copulaDifferentCoefficientsApplicationResults$estimation$EMV)+1);
                                     AICCopulaDifferentCoefficients
```

Apéndice B

Resultados de las simulaciones

B.1. Resultados obtenidos en una simulación.

1. **Distribución bivariada conjunta de los errores.**

   - **Una covariable**

     Asumiendo coeficientes iguales para infección y síntomas:

			Intervalo de confianza	
Parámetro	**Valor real**	**Estimación**	**Límite Inferior**	**Límite superior**
σ_I	0.5000	0.5387	0.4556	0.6370
σ_S	0.6667	0.7530	0.6304	0.8994
r	0.5000	0.4684	0.4097	0.6601
β_0	-0.5000	-0.6287	-1.0969	-0.1606
β_1	0.5000	0.5398	0.4019	0.6778

Asumiendo coeficientes diferentes para infección y síntomas:

			Intervalo de confianza	
Parámetro	**Valor real**	**Estimación**	**Límite Inferior**	**Límite superior**
σ_I	0.5000	0.5091	0.4390	0.5903
σ_S	0.6667	0.7222	0.6198	0.8416
r	0.5000	0.4957	0.4096	0.8221
β_{I0}	-0.3000	-0.1836	-0.6867	0.3196
β_{I1}	0.3000	0.2727	0.1232	0.4222
β_{S0}	-0.6000	-0.7110	-1.2403	-0.1817
β_{S1}	0.6000	0.6487	0.4832	0.8141

- **Dos covariables**

  Asumiendo coeficientes iguales para infección y síntomas:

			Intervalo de confianza	
Parámetro	**Valor real**	**Estimación**	**Límite Inferior**	**Límite superior**
σ_I	0.5000	0.4871	0.4155	0.5711
σ_S	0.6667	0.7016	0.5921	0.8313
r	0.5000	0.4958	0.4347	0.7110
β_0	-0.5000	-0.6559	-1.0749	-0.2368
β_1	0.5000	0.5285	0.4080	0.6489
β_2	-0.5000	-0.4614	-0.5734	-0.3494

Asumiendo coeficientes diferentes para infección y síntomas

			Intervalo de confianza	
Parámetro	**Valor real**	**Estimación**	**Límite Inferior**	**Límite superior**
σ_I	0.5000	0.4843	0.4168	0.5626
σ_S	0.6667	0.7096	0.5670	0.8879
r	0.5000	0.4544	0.3949	0.6431
β_{I0}	-0.3000	-0.1002	-0.4805	0.2801
β_{I1}	0.3000	0.2574	0.1660	0.3489
β_{I2}	-0.3000	-0.3160	-0.4137	-0.2184
β_{S0}	-0.6000	-0.1769	-0.7604	0.4066
β_{S1}	0.6000	0.5781	0.4061	0.7500
β_{S2}	-0.6000	-0.7156	-0.9261	-0.5050

- **Tres covariables**

  Asumiendo coeficientes iguales para infección y síntomas:

			Intervalo de confianza	
Parámetro	**Valor real**	**Estimación**	**Límite Inferior**	**Límite superior**
σ_I	0.5000	0.5300	0.4507	0.6232
σ_S	0.6667	0.7140	0.6025	0.8461
r	0.5000	0.4974	0.4354	0.7171
β_0	-0.5000	-0.6028	-1.2033	-0.0024
β_1	0.5000	0.5347	0.4089	0.6605
β_2	-0.5000	-0.4504	-0.5687	-0.3321
β_3	0.5000	0.4425	0.3198	0.5652

Asumiendo coeficientes diferentes para infección y síntomas

			Intervalo de confianza	
Parámetro	**Valor real**	**Estimación**	**Límite Inferior**	**Límite superior**
σ_I	0.5000	0.5507	0.4720	0.6425
σ_S	0.6667	0.7709	0.6426	0.9248
r	0.5000	0.4922	0.4032	0.8302
β_{I0}	-0.3000	-0.8799	-1.7235	-0.0363
β_{I1}	0.3000	0.4211	0.2575	0.5848
β_{I2}	-0.3000	-0.3146	-0.4690	-0.1601
β_{I3}	0.3000	0.3682	0.2085	0.5279
β_{S0}	-0.6000	-1.4809	-2.4514	-0.5105
β_{S1}	0.6000	0.7904	0.5894	0.9914
β_{S2}	-0.6000	-0.5877	-0.7530	-0.4225
β_{S3}	0.6000	0.6753	0.4887	0.8619

2. Modelo de cópulas.

- **Una covariable**

Asumiendo coeficientes iguales para infección y síntomas:

			Intervalo de confianza	
Parámetro	**Valor real**	**Estimación**	**Límite Inferior**	**Límite superior**
σ_I	0.5000	0.5077	0.4233	0.5921
σ_S	0.6667	0.6725	0.5453	0.7998
θ	2.0000	1.8906	1.6782	2.1029
β_0	-0.5000	-0.3463	-0.6606	-0.0320
β_1	0.5000	0.4385	0.3298	0.5471

Asumiendo coeficientes diferentes para infección y síntomas

			Intervalo de confianza	
Parámetro	**Valor real**	**Estimación**	**Límite Inferior**	**Límite superior**
σ_I	0.5000	0.4911	0.4253	0.5570
σ_S	0.6667	0.7306	0.5849	0.8762
θ	2.0000	2.5041	1.9204	3.0878
β_{I0}	-0.3000	-0.0306	-0.2987	0.2376
β_{I1}	0.3000	0.2141	0.1286	0.2996
β_{S0}	-0.6000	-0.1370	-0.5622	0.2883
β_{S1}	0.6000	0.4484	0.3009	0.5959

- **Dos covariables**

  Asumiendo coeficientes iguales para infección y síntomas:

			Intervalo de confianza	
Parámetro	**Valor real**	**Estimación**	**Límite Inferior**	**Límite superior**
σ_I	0.5000	0.4905	0.4106	0.5705
σ_S	0.6667	0.6404	0.5248	0.7561
θ	2.0000	2.1395	1.8668	2.4123
β_0	-0.5000	-0.2071	-0.5784	0.1641
β_1	0.5000	0.4390	0.3307	0.5473
β_2	-0.5000	-0.5275	-0.6369	-0.4181

Asumiendo coeficientes diferentes para infección y síntomas

			Intervalo de confianza	
Parámetro	**Valor real**	**Estimación**	**Límite Inferior**	**Límite superior**
σ_I	0.5000	0.5890	0.4526	0.7255
σ_S	0.6667	0.8642	0.5833	1.1452
θ	2.0000	2.0737	1.7853	2.3620
β_{I0}	-0.3000	0.5129	-0.0634	1.0892
β_{I1}	0.3000	0.1784	0.0609	0.2959
β_{I2}	-0.3000	-0.4268	-0.5751	-0.2785
β_{S0}	-0.6000	0.5314	-0.2885	1.3513
β_{S1}	0.6000	0.4989	0.3166	0.6811
β_{S2}	-0.6000	-0.8680	-1.1824	-0.5535

- **Tres covariables**

  En este caso no se pudo obtener la matriz hessiana de las librerías de R, y, por ende, tampoco la matriz de segundas derivadas, razón por la cual los intervalos de confianza se obtuvieron con bootstrapping.

  Asumiendo coeficientes iguales para infección y síntomas:

			Intervalo de confianza	
Parámetro	**Valor real**	**Estimación**	**Límite Inferior**	**Límite superior**
σ_I	0.5000	0.4814	0.4005	0.5793
σ_S	0.6667	0.6878	0.5452	0.8535
θ	2.0000	2.1090	1.8762	2.6917
β_0	-0.5000	-0.2439	-0.8588	0.4647
β_1	0.5000	0.4188	0.3182	0.5874
β_2	-0.5000	-0.5352	-0.6221	-0.3445
β_3	0.5000	0.5459	0.3075	0.5775

Asumiendo coeficientes diferentes para infección y síntomas

			Intervalo de confianza	
Parámetro	**Valor real**	**Estimación**	**Límite Inferior**	**Límite superior**
σ_I	0.5000	0.4527	0.3895	0.5232
σ_S	0.6667	0.5721	0.4725	0.7124
θ	2.0000	2.4545	1.5777	3.0977
β_{I0}	-0.3000	0.3279	-0.5315	0.7386
β_{I1}	0.3000	0.1947	0.1204	0.3153
β_{I2}	-0.3000	-0.2209	-0.3225	-0.0962
β_{I3}	0.3000	0.1414	0.0674	0.2754
β_{S0}	-0.6000	0.0941	-0.8958	0.5825
β_{S1}	0.6000	0.5093	0.3436	0.7002
β_{S2}	-0.6000	-0.4870	-0.6487	-0.2891
β_{S3}	0.6000	0.3578	0.2503	0.5888

B.2. Resultados para varias simulaciones.

Para estas simulaciones se utilizó un nivel de confianza de 95 %.

1. **Mismos coeficientes**

   - **Una covariable**

Parametro	**Cobertura Distribucion Bivariada de Valores Extremos**	**Cobertura Copula**
σ_I	0.963	0.954
σ_S	0.949	0.946
r/θ	0.942	0.949
β_0	0.959	0.939
β_1	0.959	0.941

   - **Dos covariables**

Parametro	**Cobertura Distribucion Bivariada de Valores Extremos**	**Cobertura Copula**
σ_I	0.951	0.954
σ_S	0.949	0.957
r/θ	0.951	0.958
β_0	0.948	0.947
β_1	0.961	0.957
β_2	0.949	0.939

- **Tres covariables**

Parametro	**Cobertura Distribucion Bivariada de Valores Extremos**	**Cobertura Copula**
σ_I	0.929	0.947
σ_S	0.930	0.949
r/θ	0.857	0.910
β_0	0.793	0.729
β_1	0.881	0.847
β_2	0.922	0.924
β_3	0.870	0.828

2. Diferentes coeficientes

- **Una covariable**

Parametro	**Cobertura Distribucion Bivariada de Valores Extremos**	**Cobertura Copula**
σ_I	0.944	0.934
σ_S	0.944	0.939
r/θ	0.893	0.908
$\beta_[$	0.861	0.895
β_{I1}	0.862	0.891
β_{S0}	0.853	0.845
β_{S1}	0.864	0.845

- **Dos covariables**

Parametro	**Cobertura Distribucion Bivariada de Valores Extremos**	**Cobertura Copula**
σ_I	0.853	0.925
σ_S	0.788	0.921
r/θ	0.812	0.856
β_{I0}	0.578	0.776
β_{I1}	0.625	0.854
β_{I2}	0.751	0.869
β_{S0}	0.493	0.777
β_{S1}	0.467	0.892
β_{S2}	0.788	0.842

- **Tres covariables**

Parametro	**Cobertura Distribucion Bivariada de Valores Extremos**	**Cobertura Copula**
σ_I	0.880	0.970
σ_S	0.874	0.921
r/θ	0.794	0.926
β_{I0}	0.844	0.855
β_{I1}	0.868	0.899
β_{I2}	0.876	0.946
β_{I3}	0.873	0.898
β_{S0}	0.740	0.800
β_{S1}	0.799	0.829
β_{S2}	0.888	0.933
β_{S3}	0.799	0.848

Bibliografía

Ali, M. M., Mikhail, N. y Haq, M. (1978). A class of bivariate distributions including the bivariate logistic, *Journal of Multivariate Analysis* **8**(3): 405–412.
URL: *https://www.sciencedirect.com/science/article/pii/0047259X78900635*

Camilleri, L. (2019). *History of survival analysis.* `https://timesofmalta.com/articles/view/history-of-survival-analysis.705424#:~:text=The%20term%20'survival%20analysis'%20has,by%20John%20Graunt%20in%201662`.

Cox, D. R. y Oakes, D. (1984). *Analysis of Survival Data*, Chapman Hall, United States of America.

Durante, F. y Sempi, C. (2010). Copula theory: An introduction, *in* P. Jaworski, F. Durante, W. K. Härdle y T. Rychlik (eds), *Copula Theory and Its Applications*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 3–31.

Gargantilla, P. (2021). *John Graunt, el hombre al que se le ocurrió apuntar de qué se moría la gente.* `https://www.abc.es/ciencia/abci-john-graunt-hombre-ocurrio-apuntar-moria-gente-202110011442_noticia.html`.

Golden, M. R., Whittington, W. L., Handsfield, H. H., Hughes, J. P., Stamm, W. E., Hogben, M., Clark, A., Malinski, C., Helmers, J. R., Thomas, K. K. y Holmes, K. K. (2005). Effect of expedited treatment of sex partners on recurrent or persistent gonorrhea or chlamydial infection, *New England Journal of Medicine* **352**: 676–685.

Gumbel, E. J. (1960). Distributions des valeurs extremes en plusieurs dimensions, *Publications de l'Institut de Statistique de l'Université de Paris* **9**: 171–173.

Hanagal, D. D. (2006). Bivariate weibull regression model based on censored samples, *Statistical Papers* .
URL: *https://doi.org/10.1007/s00362-005-0277-4*

Haugh, M. (2016). *An introduction to Copulas*. http://www.columbia.edu/~mh2078/QRM/Copulas.pdf.

Hofert, M., Kojadinovic, I., Maechler, M., Yan, J., NeÅ¡lehovÃ¡, J. G. y Morger, R. (2022). *Package âcopulaâ*. https://cran.r-project.org/web/packages/copula/copula.pdf.

Hosmer, D. W., Lemeshow, S. y May, S. (2008). *Applied Survival Analysis. Regression Modeling of Time-to-Event Data*, Wiley, New Jersey.

Kaplan, E. L. y Meier, P. (1958). Nonparametric estimation from incomplete observations, *Journal of the American Statistical Association* **53**(282): 457–481.
URL: *http://www.jstor.org/stable/2281868*

Kundu, D. y Dey, A. K. (2009). Estimating the parameters of the marshall olkin bivariate weibull distribution by em algorithm, *Computational Statistics Data Analysis* **53**(4): 956–965.
URL: *https://www.sciencedirect.com/science/article/pii/S0167947308005392*

Li, Y., Sun, J. y Song, S. (2012). Statistical analysis of bivariate failure time data with marshall-olkin weibull models, *Computational Statistics Data Analysis* **56**.

Liu, E. y Lim, K. (2018). Using the weibull accelerated failure time regression model to predict time to health events, *bioRxiv* .
URL: *https://www.biorxiv.org/content/early/2018/08/27/362186*

Lu, J.-C. (1989). Weibull extensions of the freund and marshall-olkin bivariate exponential models, *IEEE Transactions on Reliability* **38**(5): 615–619.

Marshall, A. W. y Olkin, I. (1967). A multivariate exponential distribution, *Journal of the American Statistical Association* **62**(317): 30–44.
URL: *http://www.jstor.org/stable/2282907*

Mejía Hernández, E. A. (2009). *Análisis de supervivencia y su aplicación para predecir la calidad de vida de los nacidos extremadamente prematuros*. https://ri.ues.edu.sv/id/eprint/12506/1/19200767.pdf.

Mohr Bemis, B. (1975). *Some statistical inferences for the bivariate exponential distribution*. https://scholarsmine.mst.edu/cgi/viewcontent.cgi?article=3267&context=doctoral_dissertations.

Moore, D. F. (2016). *Applied Survival Analysis Using R*, Springer, Switzerland.

Nelsen, R. B. (2006). *An introduction to copulas*, second edn, Springer, United States of America.

Paulon, G., Müller, P. y Sal y Rosas, V. G. (2020). Bayesian nonparametric bivariate survival regression for current status data.
URL: *https://arxiv.org/abs/2009.06460*

Rodríguez, G. (2015). *Multivariate Survival Models*. `https://data.princeton.edu/pop509/MultivariateSurvival.pdf`.

Rodríguez, G. (2021). *Survival Models*. `https://data.princeton.edu/wws509/notes/c7.pdf`.

Sklar, A. (1959). Fonctions de répartition à n dimensions et leurs marges, *Publ. inst. statist. univ. Paris* **8**: 229–231.

Stephenson, A. (2022). *Package 'evd'*. `https://cran.r-project.org/web/packages/evd/evd.pdf`.

Ziener, D. (2021). *Archimedean Copulas*. `https://www.uni-ulm.de/fileadmin/website_uni_ulm/mawi.inst.110/Seminar__Copulas_and_Applications_WS2021/David_Ziener_-_Notes__Archimedean_Copulas.pdf`.