

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

ESCUELA DE POSGRADO



Modelo de regresión no lineal basado en una mixtura de la
distribución senh-normal/independiente en el error

TESIS PARA OPTAR POR EL GRADO ACADÉMICO DE MAGÍSTER EN
ESTADÍSTICA

Autor

Carlos Iván Ocampo Corrales

Asesor

Dr. Luis Enrique Benites Sánchez

Miembros del jurado

Dra. Zaida Jesús Quiroz Cornejo

Dr. Luis Enrique Benites Sánchez

Dr. Cristian Luis Bayes Rodríguez

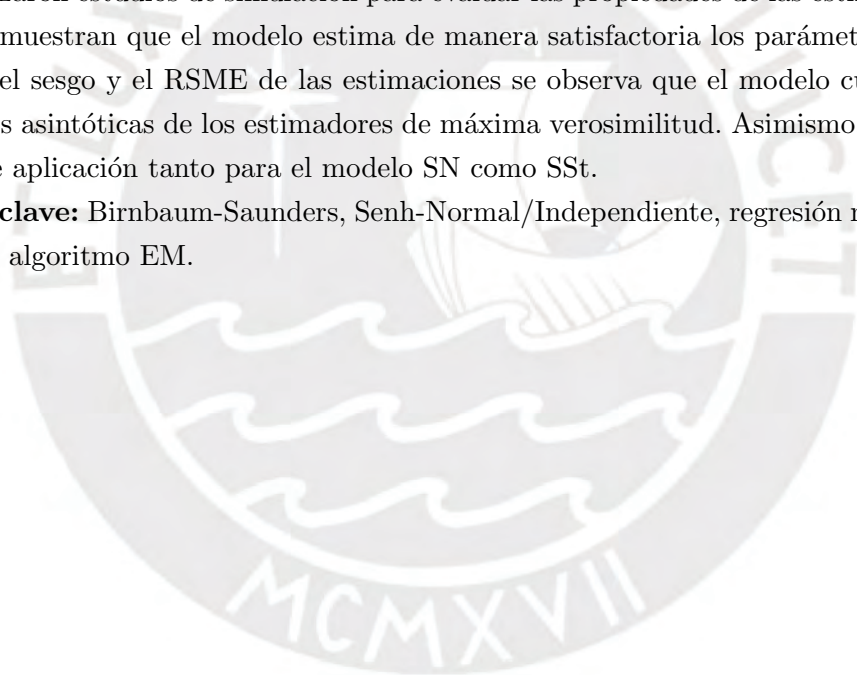
Mayo 2021

Resumen

La distribución normal, si bien útil para explicar la distribución de muchos conjuntos de datos, a veces es inadecuada para ello. En este sentido, en muchos casos es conveniente trabajar con transformaciones de la distribución normal por ejemplo log-normal, Birnbaum-Saunders (BS) y Senh-Normal (SN). En esta tesis se presenta un modelo de regresión no lineal basado en una mixtura finita de distribuciones Senh-Normal/Independiente (SNI) en el error considerando dos casos específicos de esta distribución, SN y Senh-t-Student (SSt), respectivamente. En el contexto de regresión se plantea una metodología de estimación mediante la aplicación del algoritmo EM y también para el cálculo de los errores estándar.

Se realizaron estudios de simulación para evaluar las propiedades de las estimaciones. Los resultados muestran que el modelo estima de manera satisfactoria los parámetros, más aún, evaluando el sesgo y el RSME de las estimaciones se observa que el modelo cumple con las propiedades asintóticas de los estimadores de máxima verosimilitud. Asimismo, se realizaron estudios de aplicación tanto para el modelo SN como SSt.

Palabras-clave: Birnbaum-Saunders, Senh-Normal/Independiente, regresión no lineal, mixtura finita, algoritmo EM.



Abstract

The normal distribution, while useful in explaining the distribution of many data sets, is sometimes inadequate for this. In this sense, in many cases it is convenient to work with transformations of the normal distribution, for example log-normal, Birnbaum-Saunders (BS) and Sinh-Normal (SN). In this thesis a non-linear regression model is presented based on a finite mixture of Sinh-Normal / Independent (SNI) distributions in the error considering two specific cases of this distribution, SN and Sinh-t-Student (SSt), respectively. In the regression context, an estimation methodology is proposed through the application of the EM algorithm and also for the calculation of errors.

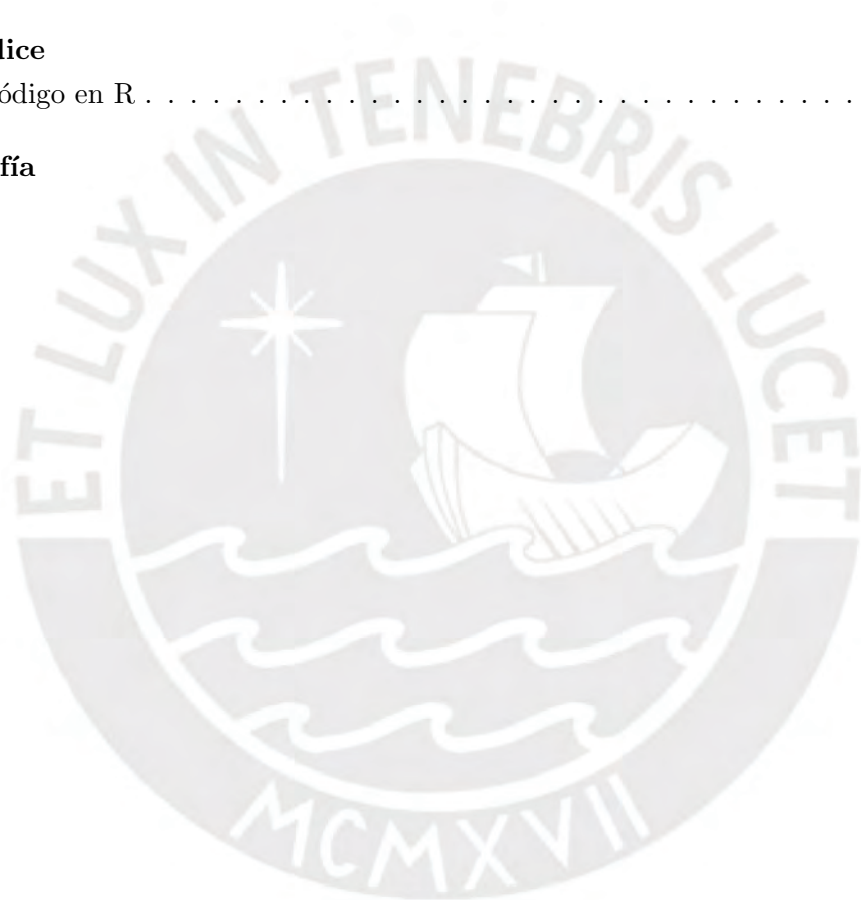
Simulation studies were conducted to evaluate the properties of the estimates. The results show that the model satisfactorily estimates the parameters, furthermore, evaluating the bias and the RSME of the estimates, it is observed that the model complies with the asymptotic properties of the maximum likelihood estimators. Likewise, application studies were carried out for both the SN and SSt models.

Keywords: Birnbaum-Saunders, Sinh-Normal/Independent, nonlinear regression, finite mixture, EM-algorithm.

Índice general

Índice de figuras	VI
Índice de cuadros	VII
1. Introducción	1
1.1. Objetivos	1
1.2. Organización del Trabajo	2
2. Conceptos preliminares	3
2.1. Distribución Birnbaum Saunders	3
2.1.1. Derivación de la distribución	3
2.1.2. Función de densidad de probabilidad	5
2.1.3. Propiedades	5
2.1.4. Aplicaciones	7
2.2. Regresión no lineal	8
2.2.1. Estimación	9
2.2.2. Elección de los valores iniciales	10
2.2.3. Uso de software	12
2.3. Mixtura finita	12
2.3.1. Formulación básica	12
2.3.2. Estimación de parámetros	14
2.3.3. Descripción del algoritmo EM	15
2.4. Distribución Senh-normal y Senh-normal/independiente	16
2.4.1. La distribución Senh-normal	16
2.4.2. La distribución Senh-normal/independiente	18
3. Regresión no lineal cuyo error sigue una MF-SNI	20
3.1. Regresión lineal con una mixtura finita de distribución SN	20
3.1.1. Estimación de los parámetros	21
3.2. Modelo de regresión no lineal distribuciones SNI	23
3.2.1. El modelo no lineal	23
3.2.2. Estimación de máxima verosimilitud y el algoritmo EM	24
3.3. El modelo no lineal con mixtura	25
3.3.1. Estimación de los parámetros	26
3.3.2. Valores iniciales	28

3.3.3. Selección de modelos	28
3.3.4. Estimación de los errores estándar	29
4. Estudio de simulación	31
4.1. Estudio de simulación	31
4.1.1. Resultados de la simulación	32
5. Aplicación	44
6. Conclusiones	48
6.1. Conclusiones	48
6.2. Sugerencias para investigaciones futuras	48
A. Apéndice	50
A.1. Código en R	50
Bibliografía	63



Índice de figuras

2.1. Densidades de la distribución Birnbaum-Saunders para diferentes valores de los parámetros.	6
2.2. (a) Diagrama de dispersión de los datos Chwirut2 de respuesta ultrasónica con distancia del metal y (b) funciones de aproximación a los datos para distintos valores iniciales para los parámetros β	11
2.3. Ejemplo de mixtura de tres componentes con densidad normal, es decir de $0,3 \times N(2, 1) + 0,5 \times N(3, 0,5^2) + 0,2 \times N(3,4, 1,3^2)$	13
2.4. Función de densidad de probabilidad SN para distintos valores de los parámetros α, μ y σ respectivamente.	17
4.1. Gráfico de cajas de las estimaciones de los parámetros α y β respectivamente.	38
4.2. Gráfico de cajas de las estimaciones de los parámetros μ, ν y p_1 respectivamente.	39
4.3. Sesgo de las estimaciones de los parámetros α y β respectivamente.	40
4.4. Sesgo de las estimaciones de los parámetros μ, ν y p_1 respectivamente.	41
4.5. RMSE de las estimaciones de los parámetros α y β respectivamente.	42
4.6. RMSE de las estimaciones de los parámetros μ, ν y p_1 respectivamente.	43
5.1. Log-verosimilitud perfilada para el caso t -student para $g = 2, g = 3$ y $g = 4$ componentes respectivamente (de izquierda a derecha)	46
5.2. Diagrama de dispersión y las curvas ajustadas para el modelo RNL-MF-SN (panel izquierdo) y el modelo RNL-MF-SSt (panel derecho) para diferentes números de componentes $g = 1, 2, 3, 4$	46
5.3. Diagrama de dispersión y las curvas ajustadas para el modelo RNL-MF-SN (panel izquierdo) y el modelo RNL-MF-SSt (panel derecho) para diferentes números de componentes $g = 1, 2, 3, 4$	47

Índice de cuadros

3.1. Resumen de criterios de selección de modelos. $\ell(\boldsymbol{\theta})$ es la log-verosimilitud. K y n es el número de parámetros y observaciones respectivamente.	29
4.1. Estudio de simulación 1 (RLN-MF-SSt): Media de estimación, errores estándar de la matriz de información (IM SE), la desviación estándar de Monte Carlo (MC SE) en base a 1000 simulaciones. Los valores reales están entre paréntesis.	34
4.2. Estudio de simulación 1 (RLN-MF-SN) : Media de estimación, errores estándar de la matriz de información (IM SE), la desviación estándar de Monte Carlo (MC SE) en base a 1000 simulaciones. Los valores reales están entre paréntesis.	35
4.3. Estudio de simulación 2: Media de estimación, errores estándar, sesgo, RMSE en base a 1000 simulaciones. Los valores reales están entre paréntesis.	36
4.4. Estudio de simulación 2: Media de estimación, errores estándar, sesgo, RMSE en base a 1000 simulaciones. Los valores reales están entre paréntesis.	37
5.1. Criterios de información para $g = 1$, $g = 2$, $g = 3$ y $g = 4$ componentes para los casos MRNL-MF-SN y MRNL-MF-SSt	44
5.2. Estimación (Est.) de máxima verosimilitud para dos modelos seleccionados con $g = 3$ componentes. El error estándar (EE) han sido estimados basado en el método de aproximación de la matriz de información observada. LI (límite inferior) y LS (límite superior) del intervalo de confianza al 95 % mientras que entre paréntesis se encuentra el error estándar bootstrap, límite inferior bootstrap y límite inferior bootstrap del intervalo de confianza al 95 %.	45
5.3. Criterios de información para $g = 1$, $g = 2$, $g = 3$ y $g = 4$ componentes para los casos SN y SSt	47
5.4. Estimación de máxima verosimilitud para dos modelos seleccionados con $g = 3$ componentes. El error estándar (EE) obtenido por bootstrap. LI_B (límite inferior bootstrap) y LS_B (límite superior bootstrap) del intervalo de confianza al 95 %.	47

Capítulo 1

Introducción

La distribución normal ha tenido un papel importante tanto en la estadística teórica como en la aplicada. En la práctica, sin embargo, sabemos que la curva normal no proporciona una representación adecuada de muchas distribuciones. Johnson (1949) mencionó que es natural y conveniente construir distribuciones no normales mediante la transformación de una variable aleatoria (v.a.) normal. Algunos ejemplos de estas transformaciones son las distribuciones Birnbaum y Saunders (1969a,b) (BS), Senh-normal (SN) (Rieck, 1989), log-normal (Johnson y Kotz, 1970), entre otros. Con base de datos de vida útil, la relación entre las distribuciones SN y BS fue discutida por Rieck (1989) y Leiva et al. (2010). Mientras que Rieck y Nedelman (1991) consideraron esa relación en el contexto de regresión, especialmente cuando los datos de vida útil implican el modelado por la distribución BS, en el que el parámetro de escala depende de las covariables. Lemonte y Cordeiro (2009) definieron el modelo de regresión no lineal Birnbaum-Saunders (MNL-BS) basado en la distribución BS generalizando así el modelo de regresión lineal descrito por Rieck y Nedelman (1991).

1.1. Objetivos

El objetivo general de este trabajo es presentar un modelo de regresión no lineal donde el error sigue una distribución basada en mixtura finita de las distribuciones Senh-Normal (SN) y Senh-t-Student (SSt), respectivamente, ambos casos particulares de la clase de distribuciones Senh-normal/independiente (SNI), incluyendo la metodología para la estimación de parámetros y su aplicación a conjuntos de datos reales. Para esto, será necesario revisar los conceptos que permitan entender la génesis de la distribución SN así como de mixtura y regresión no lineal. Posteriormente, se revisará la literatura relacionada a modelos de regresión para esta distribución para luego plantear un modelo de regresión no lineal con mixtura en el error. Se busca:

- Proponer e implementar un algoritmo EM para estimar los parámetros.
- Desarrollar un estudio de simulación que permita ilustrar el desempeño de la metodología utilizada.
- Aplicar nuestro modelo a conjuntos de datos reales que permitan ilustrar la metodología utilizada.

1.2. Organización del Trabajo

En el Capítulo 2 revisaremos algunos conceptos fundamentales relacionados con las distribuciones BS y SN así como una breve reseña de la teoría detrás de los modelos de regresión no lineal y de mixtura. Posteriormente, en el Capítulo 3 revisaremos los modelos de regresión planteados en la literatura para las distribuciones SN y SNI. Se planteará un modelo de regresión no lineal con mixtura en el error con distribución Senh-Normal/Independiente (SNI). En el Capítulo 4 se presenta un ejercicio de simulación para una regresión basada en una mixtura de distribuciones SSt. En el Capítulo 5 se presenta un ejercicio de regresión con datos reales de distribuciones basadas en mixtura de SN y SSt. Finalmente, en el Capítulo 6 se presentan conclusiones.



Capítulo 2

Conceptos preliminares

En este apartado revisaremos algunos conceptos preliminares que sirven de base teórica para modelos planteados en capítulos posteriores.

2.1. Distribución Birnbaum Saunders

El origen de la distribución BS proviene de la motivación de estudiar el efecto de vibraciones que causan fatiga sobre materiales. Así, Birnbaum y Saunders (1969a) consideró el estudio de las vibraciones y su efecto en los materiales de aviones comerciales. En la actualidad, sin embargo, la aplicación de la distribución BS ha sido extendida a muchos campos además de estudios de fatiga y confiabilidad.

2.1.1. Derivación de la distribución

Como se mencionó anteriormente, la motivación original de la distribución fue el estudio de fatiga sobre un material. Esta se entiende como el proceso de cambio progresivo, localizado y estructural de un material sujeto a condiciones de tensión. Dicho proceso puede causar daños o inclusive su ruptura en algún momento del tiempo. En la práctica, la vida útil de un material o estructura sometida a fatiga es equivalente al número de ciclos de tensión o estrés hasta su falla o ruptura, debido a fatiga. Como una distribución de fatiga, la distribución BS consideró a las siguientes asunciones:

- El material es sujeto a ciclos de cargas o choques repetitivos idénticos que producen una ruptura o desgaste en el material.
- La falla en el material ocurre cuando el tamaño de la ruptura excede un nivel de resistencia, denotado como umbral ω .
- El tamaño de la falla causada por el ciclo k -ésimo, Y_k , es una variable aleatoria con distribución propia, media μ y varianza σ^2 , ambos finitos.
- Los tamaños de la grieta Y_k en diferentes ciclos son independientes.

La distribución BS observa el mínimo número de ciclos, digamos N^* , tal que la suma

$$S_n = \sum_{k=1}^n Y_k, \quad (2.1)$$

con n como variable aleatoria positiva excede el umbral dado por ω , esto es:

$$N^* = \inf \{n \in \mathbb{N} : S_n > \omega\},$$

donde S_n viene dado por (2.1). Teniendo en cuenta los supuestos anteriores, si aplicamos la versión clásica del teorema de límite central y consideramos a N^* como una variable aleatoria continua T , cuando $n \rightarrow \infty$, es posible establecer que S_n sigue una distribución aproximadamente normal (\sim) con media $n\mu$ y varianza $n\sigma^2$, esto es,

$$S_n \sim N(n\mu, n\sigma^2). \quad (2.2)$$

Ahora, sea N el número de ciclos requeridos hasta la falla el material. Dado que $Y_k > 0$ para todos los $k \geq 1$, con daño irreversible, entonces por complementariedad tenemos que $\{N > n\} \equiv \{S_n \leq \omega\}$ y $\{N \leq n\} \equiv \{S_n > \omega\}$. Entonces, como consecuencia de asumir que la falla Y_k es una variable aleatoria independiente con media μ y varianza σ^2 y de (2.1) tenemos que $E(S_n) = n\mu$ y $\text{Var}(S_n) = n\sigma^2$. Por lo tanto, de la ecuación (2.2) tenemos:

$$Z_n = \frac{S_n - n\mu}{\sigma\sqrt{n}} \sim N(0, 1). \quad (2.3)$$

Así, de la ecuación (2.3) obtenemos:

$$\begin{aligned} P(N \leq n) &= P(S_n > \omega) \\ &= P\left[Z_n > \frac{\sqrt{\mu\omega}}{\sigma} \left(\frac{1}{\sqrt{n}}\sqrt{\frac{\omega}{\mu}} - \sqrt{n}\sqrt{\frac{\mu}{\omega}}\right)\right] \\ &\approx \Phi\left[\frac{\sqrt{\mu\omega}}{\sigma} \left(\sqrt{n}\sqrt{\frac{\mu}{\omega}} - \frac{1}{\sqrt{n}}\sqrt{\frac{\omega}{\mu}}\right)\right], \end{aligned} \quad (2.4)$$

donde $\Phi(\cdot)$ es la función de distribución acumulada normal estándar. Birnbaum y Saunders (1969a) utilizaron (2.4) para definir la distribución continua del tiempo de vida reemplazando la variable discreta N por la variable continua T y el argumento discreto n por el argumento continuo t . En este sentido, el número de ciclos N es convertido en el tiempo total hasta la ocurrencia de la falla T , donde el n -ésimo ciclo es convertido al tiempo t . Entonces, de la ecuación (2.4) podemos definir: $\alpha = \frac{\sigma}{\sqrt{\omega\mu}}$, $\beta = \frac{\omega}{\mu}$, $\xi(x) = \sqrt{x} - \frac{1}{\sqrt{x}} = 2\sinh(\log(\sqrt{x}))$ para obtener la función de distribución acumulada de la distribución BS con parámetro de forma $\alpha > 0$ y parámetro de escala $\beta > 0$:

$$F_T(t) = \Phi\left(\frac{1}{\alpha}\xi\left(\frac{t}{\beta}\right)\right), t > 0, \alpha > 0, \beta > 0. \quad (2.5)$$

Decimos que una si una variable T tiene una función de distribución acumulada definida por (2.5), se usará la notación $T \sim \text{BS}(\alpha, \beta)$. En este punto la distribución BS se relaciona con la distribución normal, si hacemos la representación de T con la forma

$$T = \beta \left(\frac{\alpha Z}{2} + \sqrt{\left(\frac{\alpha Z}{2}\right)^2 + 1} \right)^2, \quad (2.6)$$

donde Z es una variable aleatoria con distribución normal estándar $Z \sim N(0, 1)$, por lo tanto,

$$Z = \frac{1}{\alpha} \left[\sqrt{\frac{T}{\beta}} - \sqrt{\frac{\beta}{T}} \right] \sim N(0, 1). \quad (2.7)$$

Así, tenemos dos aproximaciones a la distribución BS, la primera desde el concepto de fatiga e interpretación de la distribución de la vida de un material hasta su fractura o falla mediante las asunciones que S_n y T_k como representaciones de la extensión total y la extensión producida por el ciclo k , respectivamente. De otro lado, de las ecuaciones (2.6) y (2.7) podemos observar que la variable T con distribución BS es una transformación de otra variable, Z , con distribución normal estándar.

2.1.2. Función de densidad de probabilidad

La función de densidad de probabilidad (fdp) de la distribución BS se puede obtener derivando la ecuación (2.5), recordemos que:

$$F_T(t) = \Phi \left(\frac{1}{\alpha} \left(\sqrt{\frac{t}{\beta}} - \sqrt{\frac{\beta}{t}} \right) \right), t > 0.$$

Entonces,

$$\begin{aligned} f_T(t; \alpha, \beta) &= \frac{\partial \Phi_T(t)}{\partial t} \left(\frac{1}{\alpha} \xi \left(\frac{t}{\beta} \right) \right) \frac{d}{dt} \left(\frac{1}{\alpha} \xi \left(\frac{t}{\beta} \right) \right) \\ &= \frac{1}{\sqrt{2\pi}} \exp \left(-\frac{1}{2\alpha^2} \left(\frac{t}{\beta} + \frac{\beta}{t} - 2 \right) \right) \frac{1}{2\alpha\beta} \left(\left(\frac{t}{\beta} \right)^{-\frac{1}{2}} + \left(\frac{t}{\beta} \right)^{-\frac{3}{2}} \right) \\ &= \frac{1}{\sqrt{8\pi}} \exp \left(\frac{1}{\alpha^2} \right) \exp \left(-\frac{1}{2\alpha^2} \left(\frac{t}{\beta} + \frac{\beta}{t} - 2 \right) \right) \frac{t^{-3/2}}{\alpha\beta^{1/2}} (t + \beta); t > 0, \alpha > 0, \beta > 0. \end{aligned} \quad (2.8)$$

La Figura 2.1 muestra la descripción gráfica de la función de densidad de la función (2.8) para distintos valores de parámetros. Del gráfico notamos que la distribución es continua, unimodal y con asimetría positiva. También podemos notar que a medida que α tiende a cero la distribución se va volviendo más simétrica centrada en β . Por otro lado, a medida que α se incrementa la distribución tiene colas más pesadas, esto nos indica que el parámetro α afecta la curtosis de la distribución.

2.1.3. Propiedades

Para definir la media de la distribución podemos recordar la forma (2.6) y notar que el parámetro β no afecta a la forma de la variable aleatoria T sino solo actúa como un factor multiplicador, por lo tanto, podemos encontrar una forma del valor esperado de la distribución BS a partir de $(T/\beta)^r$, como se muestra a continuación:

$$E \left(\left(\frac{T}{\beta} \right)^r \right) = E \left(\left(\frac{\alpha Z}{2} + \sqrt{\left(\frac{\alpha Z}{2} \right)^2 + 1} \right)^{2r} \right)$$

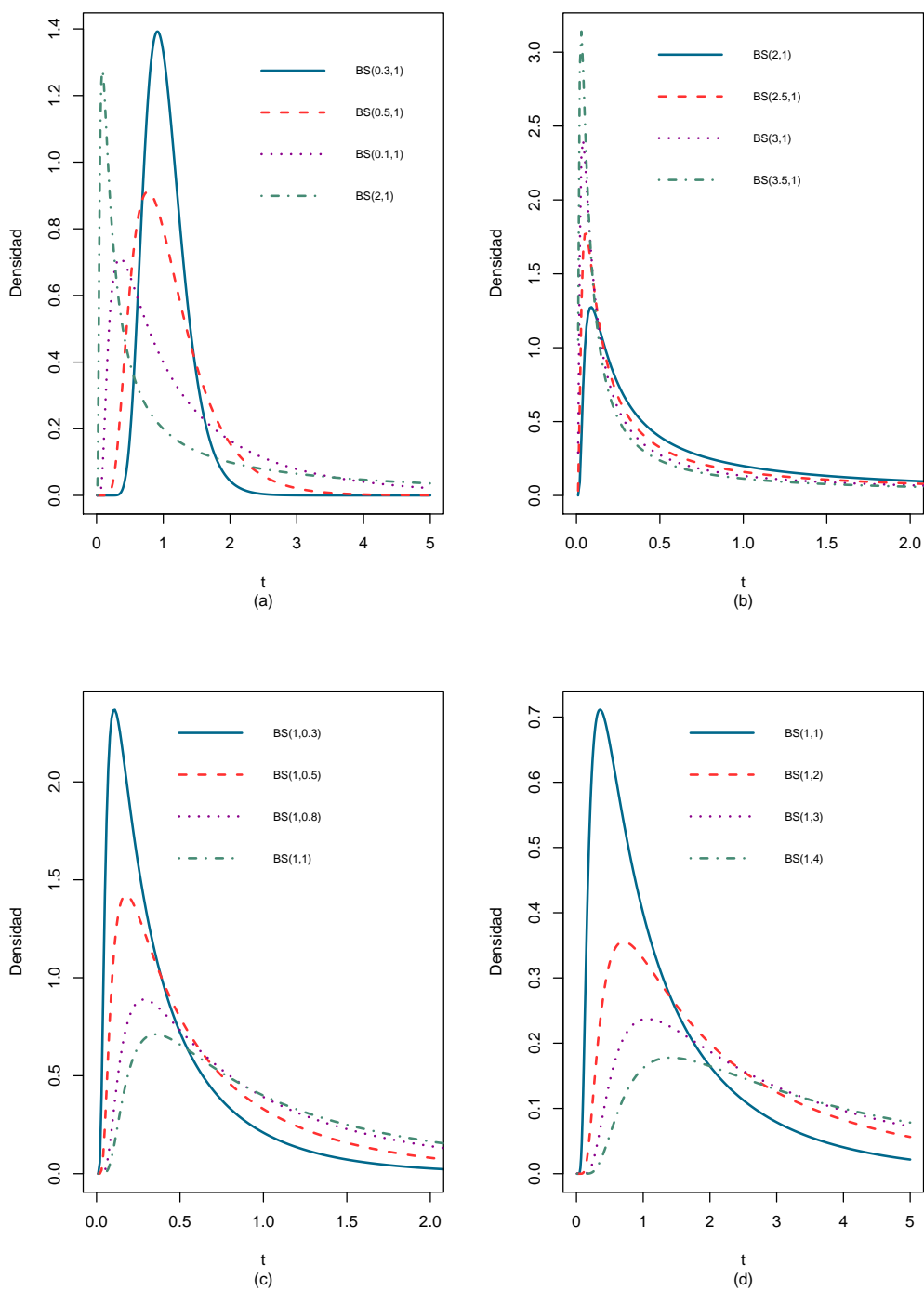


Figura 2.1: Densidades de la distribución Birnbaum-Saunders para diferentes valores de los parámetros.

Podemos resolver $E\left(\left(\frac{T}{\beta}\right)^r\right)$ haciendo uso del teorema binomial definido como:

$$(a + b)^m = \sum_{k=0}^m \binom{m}{k} a^{m-k} b^k.$$

Por lo que podemos plantear el valor esperado,

$$E\left(\left(\frac{T}{\beta}\right)^r\right) = \sum_{k=0}^{2r} \binom{2r}{k} E\left(\left(\left(\frac{\alpha Z}{2}\right)^2 + 1\right)^{k/2} + \left(\frac{\alpha Z}{2}\right)^{2r-k}\right).$$

Podemos notar que

$$E\left(\left(\left(\frac{\alpha Z}{2}\right)^2 + 1\right)^t + \left(\frac{\alpha Z}{2}\right)^s\right) = 0,$$

cuando s es impar. Por tanto,

$$E\left(\left(\frac{T}{\beta}\right)^r\right) = \sum_{k=0}^r \binom{2r}{2k} E\left(\left(\left(\frac{\alpha Z}{2}\right)^2 + 1\right)^k + \left(\frac{\alpha Z}{2}\right)^{2(r-k)}\right),$$

donde

$$\left(\left(\frac{\alpha Z}{2}\right)^2 + 1\right)^j = \sum_{k=0}^j \binom{j}{k} \left(\frac{\alpha Z}{2}\right)^{2k}$$

así tenemos

$$\begin{aligned} E\left(\left(\frac{T}{\beta}\right)^r\right) &= \sum_{j=0}^r \binom{2r}{2j} E\left(\sum_{k=0}^j \binom{j}{k} \left(\frac{\alpha Z}{2}\right)^{2k} \left(\frac{\alpha Z}{2}\right)^{2(r-j)}\right) \\ &= \sum_{j=0}^r \binom{2r}{2j} \sum_{k=0}^j \binom{j}{k} \left(\frac{\alpha}{2}\right)^{2(r-j+k)} E\left(Z^{2(r-k+k)}\right) \end{aligned}$$

luego

$$E\left(Z^{2(r-k+k)}\right) = \frac{(2(r-j+k))!}{2^{r-j+k}(r-j+k)!}$$

finalmente se tiene

$$E\left(\left(\frac{T}{\beta}\right)^r\right) = \sum_{j=0}^r \binom{2r}{2j} \sum_{k=0}^j \binom{j}{k} \left(\frac{\alpha}{2}\right)^{2(r-j+k)} \frac{(2(r-j+k))!}{2^{r-j+k}(r-j+k)!}. \quad (2.9)$$

A partir de esta función podemos calcular el valor esperado $E(T/\beta)$ y la varianza $\text{Var}(T) = E(T^2) - (E(T))^2$ obteniendo:

$$E(T) = \beta \left(1 + \frac{1}{2}\alpha^2\right), \quad \text{Var}(T) = (\alpha\beta)^2 \left(1 + \frac{5}{4}\alpha^2\right).$$

En general, cualquier momento de la función BS queda definido por la ecuación (2.9).

2.1.4. Aplicaciones

La distribución BS ha tenido un campo de aplicación extendido del uso de estimación de fatiga (Leiva, 2016), entre las aplicaciones se encuentran:

- Análisis fluvial.
- Análisis sobre componentes aeronáuticos expuestos a estrés.

- Ocurrencia de enfermedades cardíacas crónicas debido al daño acumulado causado por varios factores de riesgo.
- Desarrollo de intangibles en la ejecución de un proceso de un software que produce daño acumulativo y deteriora su desempeño.
- Riesgo de contaminación en lagos, ríos y reservas debido a la acumulación de nutrientes vegetales.
- Flujo de energía eólica y climatología.
- Calidad del aire debido a la acumulación de contaminantes en la atmósfera.

Algunos ejemplos de aplicación de la distribución BS en diversos campos de estudio como la astronomía, genética, meteorología y finanzas se pueden encontrar en Mehrdad et al. (2019), Hossein et al. (2020), Mohammadi et al. (2017), Mehrdad et al. (2020), respectivamente.

2.2. Regresión no lineal

Si queremos observar la relación que existe entre una variable y con otra variable, llamada predictora, x , a través de una función $f(\cdot)$ posiblemente desconocida, podemos expresar la relación de la siguiente manera:

$$y = f(x, \boldsymbol{\beta}), \quad (2.10)$$

donde $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)^\top$ es un conjunto desconocido de parámetros para la función $f(\cdot)$. Si esta función es no lineal en uno o más de sus parámetros β_1, \dots, β_p (e.g. $f(x, \boldsymbol{\beta}) = \beta_1 x^{\beta_2}$) se dice que la relación entre la variable y y la predictora x es no lineal.

En la práctica, los parámetros β_1, \dots, β_p en (2.10) tienen que ser estimados a partir de la información contenida en un conjunto de datos. Más aún, decimos que la variable x predice correctamente y sin error a la variable y conforme a (2.10) solo en *promedio*, es decir, asumiendo que no existen errores de medición (ver Ritz y Streibig (2008)). Formalizamos esta idea a través de la noción de la respuesta condicional para la observación y_i de la siguiente manera:

$$E(y_i | x_i) = f(x_i, \boldsymbol{\beta}). \quad (2.11)$$

La ecuación (2.11) indica que dado el valor del predictor x_i , se espera que la variable respuesta este centrada alrededor del valor $y = f(x_i, \boldsymbol{\beta})$. Así, es posible referirse a la función $f(\cdot)$ como una función de media. Sin embargo, como se mencionó anteriormente, la medida proporcionada de y_i puede estar distorsionada por la existencia de un error de medición entre la observación y_i y el valor obtenido a través de $E(y_i | x_i)$. En consecuencia, para poder completar la especificación de la relación entre las variables podemos escribir el modelo como:

$$y_i = E(y_i | x_i) + \epsilon_i = f(x_i, \boldsymbol{\beta}) + \epsilon_i,$$

donde ϵ_i representa el error¹ de la i -ésima observación.

¹Por lo general se asume que este error tienen una distribución normal con media 0 y varianza σ^2 .

2.2.1. Estimación

La estimación de los parámetros $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)^\top$ de la función $f(\cdot)$ se obtiene, por ejemplo, mediante la minimización del error entre la media recogida por $f(\cdot)$ y las observaciones y_1, \dots, y_n . Esta idea es expresada mediante la minimización de la suma de los cuadrados de los residuos (RSS, por sus siglas en inglés) respecto a $\boldsymbol{\beta}$, esto es:

$$\text{RSS}(\boldsymbol{\beta}) = \sum_{i=1}^n (y_i - f(x, \boldsymbol{\beta}))^2.$$

En contraste con la regresión lineal, la minimización de RSS tendrá una forma no lineal y por tanto se requiere de métodos numéricos para su resolución. Estos métodos comprenden procedimientos iterativos que idealmente van a aproximar el valor óptimo del parámetro de manera progresiva. En cada etapa de la iteración, los algoritmos determinan un nuevo valor para los parámetros, basados en la data, el modelo y el valor de los parámetros para esa iteración. Uno de los métodos más utilizados es el método de Newton de aproximación lineal que se resume a continuación: Para determinada función no lineal que se busca minimizar y cuyo vector de parámetros es $\boldsymbol{\theta}$: $w(\boldsymbol{\theta})$, como función objetivo definida en un espacio p dimensional. El vector p gradiente se denota por ∇ y está definido como:

$$\nabla = \frac{\partial w}{\partial \boldsymbol{\theta}} = \left[\frac{\partial w}{\partial \theta_1}, \dots, \frac{\partial w}{\partial \theta_p} \right]^T$$

y la matriz $p \times p$ simétrica Hessiana, denotada por H , se define como:

$$H = \frac{\partial^2 w}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T} = \begin{bmatrix} \frac{\partial^2 w}{\partial \theta_1^2} & \frac{\partial^2 w}{\partial \theta_1 \partial \theta_2} & \cdots & \frac{\partial^2 w}{\partial \theta_1 \partial \theta_p} \\ \frac{\partial^2 w}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2 w}{\partial \theta_2^2} & \cdots & \frac{\partial^2 w}{\partial \theta_2 \partial \theta_p} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 w}{\partial \theta_p \partial \theta_1} & \frac{\partial^2 w}{\partial \theta_p \partial \theta_2} & \cdots & \frac{\partial^2 w}{\partial \theta_p^2} \end{bmatrix}^T.$$

El dominio de $\boldsymbol{\theta}$ es, en la mayoría de casos de regresión no lineal, una función de los parámetros del modelo. Para la estimación mediante minimización, en cada iteración el parámetro $\boldsymbol{\theta}$ es actualizado. Si el paso de la iteración se denota por (a) , al paso $(a+1)$ el parámetro $\boldsymbol{\theta}$ puede ser valuado en incrementos δ usando el método de búsqueda lineal de la forma:

$$\boldsymbol{\theta}^{(a+1)} = \boldsymbol{\theta}^{(a)} + \delta^{(a)}.$$

En cada iteración, un nuevo valor de la función objetivo es calculado para $w^{(a+1)} = w(\boldsymbol{\theta}^{(a+1)})$, y la reducción en su valor es analizada. En particular, el método de Newton calcula los incrementos δ de haciendo uso de la Hessiana:

$$\delta^{(a)} = -H^{-1(a)} \nabla^{(a)},$$

donde $H^{-1(a)}$ es la inversa de la matriz Hessiana y $\nabla^{(a)}$ es el vector gradiente en $\theta^{(a)}$. Las iteraciones continúan hasta encontrar un punto óptimo, conforme a alguna regla de convergencia impuesta de la forma $|\theta^{(a-1)} - \theta^{(a)}| < \epsilon$, con ϵ suficientemente pequeño.

Cabe indicar que ante la presencia de outliers, al evaluar la matriz Hessiana esta puede volverse singular por diversos motivos (e.g. redondeo) aún cuando matemáticamente la matriz no lo es. Para resolver este problema se puede utilizar el método de Levenberg y Marquardt que modifica la función $\delta^{(a)} = -(H + \mu_a I)^{-1(a)} \nabla^{(a)}$, $\mu_a \geq 0$, de tal manera que el valor constante μ_a asegura que la búsqueda de valores sea de forma descendiente.

Luego de la revisión a los métodos de optimización, es claro que entre las complicaciones de los métodos iterativos, podemos mencionar los siguientes puntos a considerar:

- Cómo iniciar el procedimiento y cómo escoger los valores iniciales para los parámetros con los que el algoritmo iniciará.
- Cómo asegurarse de que el procedimiento haya alcanzado un mínimo global en lugar de uno local.

Ambos problemas están interrelacionados. Si los valores iniciales están suficientemente cerca al valor óptimo de los parámetros, entonces el procedimiento, por lo general, se acercará más y más al valor óptimo (i.e. converge) en unas pocas iteraciones. Por tanto, es de suma importancia el proveer valores iniciales razonables, de lo contrario, se corre el riesgo de no llegar a la convergencia aun cuando el modelo este correctamente especificado.

2.2.2. Elección de los valores iniciales

Como vimos en párrafos precedentes es importante encontrar valores iniciales adecuados para poder asegurar la convergencia de los métodos de optimización utilizados en la estimación de los parámetros. Por tanto, puede ser de utilidad tener en cuenta algunas técnicas para encontrar valores iniciales, a continuación algunos ejemplos. La aproximación más común a los valores iniciales es considerar si los parámetros que serán estimados tienen una interpretación intrínseca dentro del modelo no lineal y, de ser así, la evidencia empírica podría proporcionar valores iniciales adecuados. Si no es el caso, es posible aproximarse mediante una exploración gráfica o buscar activamente valores iniciales.

A manera de ejemplo de exploración gráfica (Ritz y Streibig, 2008), observemos los datos “Chwirut2” de la librería NISTnls en R que contiene medidas de un experimento examinando como la respuesta ultrasónica depende de la distancia del metal². Los datos son presentados en la Figura (2.2). NIST sugiere un modelo de decrecimiento exponencial donde la variable respuesta y (respuesta ultrasónica) esta relacionada con el predictor x (distancia del metal) por la siguiente función:

$$f(x, (\beta_1, \beta_2, \beta_3)) = \frac{\exp(-\beta_1 x)}{\beta_2 + \beta_3 x}.$$

Para $x = 0$ el modelo equivale a $1/\beta_2$ y por lo tanto, β_2 puede ser entendido como el valor recíproco de la respuesta más cercana al eje y , el cual es aproximadamente $1/100 = 0,01$. Hasta aquí, los otros dos parámetros no tienen una interpretación directa similar, para

²Los datos son proporcionados por el Instituto Nacional de Estándares y Tecnología, NIST, ver Ritz y Streibig (2008)

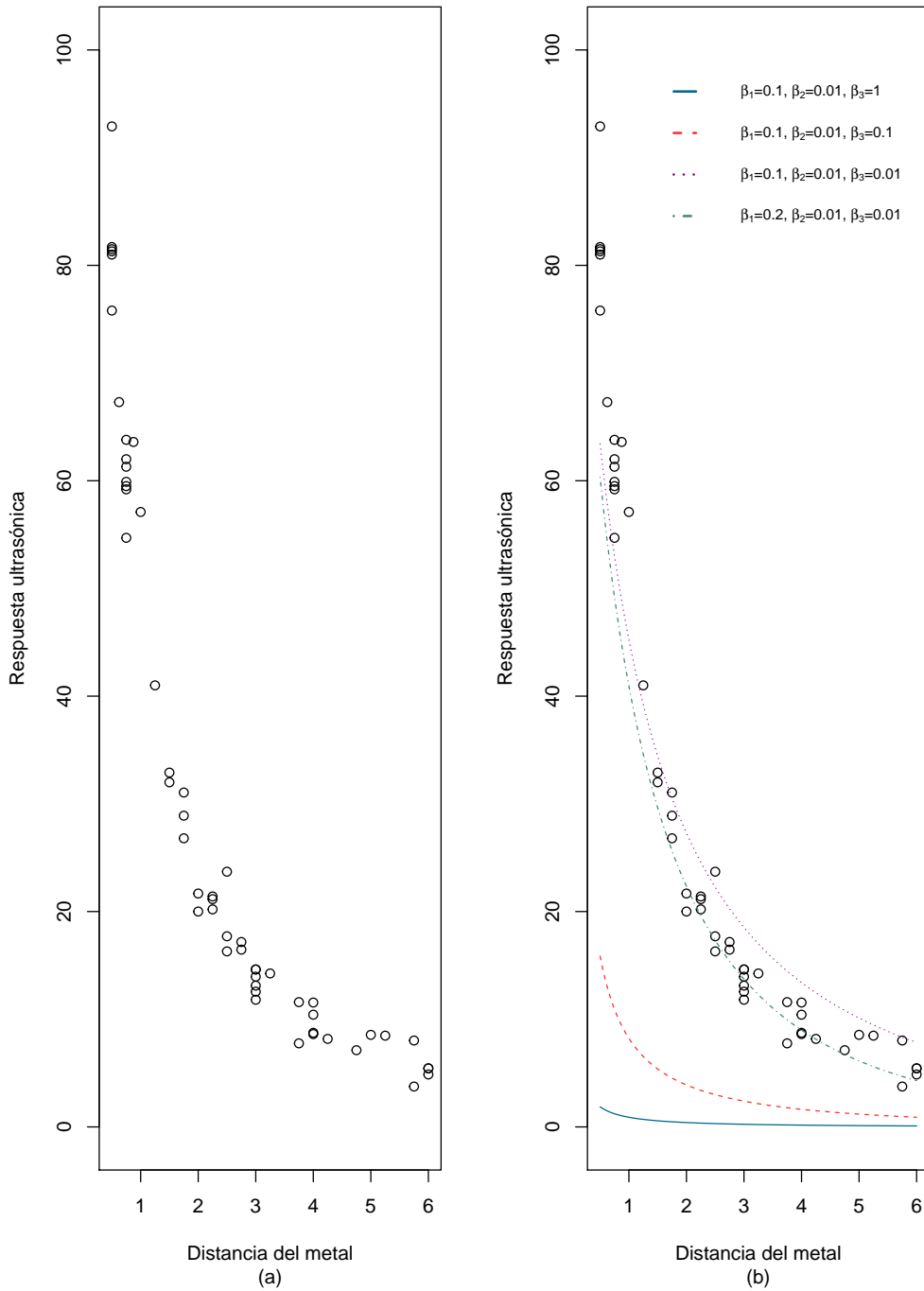


Figura 2.2: (a) Diagrama de dispersión de los datos Chwirut2 de respuesta ultrasónica con distancia del metal y (b) funciones de aproximación a los datos para distintos valores iniciales para los parámetros β .

su aproximación podríamos realizar gráficos adicionales de exploración probando distintos valores para β_1 y β_3 y observar como varía el valor de la función de la media de $f(\cdot)$. La parte (b) de la Figura (2.2) muestra este ejercicio. La línea con los parámetros $(\beta_1, \beta_2, \beta_3) = (0,2, 0,01, 0,01)$ es la que más se acerca a los datos. Por lo tanto, la exploración gráfica ayuda

a identificar posibles valores iniciales, sin embargo, en la práctica, se requiere también de métodos más rigurosos para encontrar valores iniciales adecuados.

Otra forma de encontrar valores iniciales adecuados consiste en el uso de funciones de auto inicio ³ (Ritz y Streibig, 2008). Son funciones definidas específicamente para una función de media dada y calculan los valores iniciales dado un conjunto de datos. Es posible construir más de una función de auto inicio para una función de media y sin embargo, no todos los valores proporcionados por éstas resultarían en modelos optimizados en forma robusta.

2.2.3. Uso de software

Como es necesario resolver los modelos no lineales mediante el uso de métodos iterativos, es necesario también utilizar software para poder acelerar el proceso estimación de los parámetros. En R, existen varios paquetes con funciones para ajustar modelos no lineales. Para el ejemplo anterior, la función `nls`, en particular, tiene la sintaxis:

```
nls1 <- nls(y ~ (exp(-beta1 * x)/(beta2 + beta3 * x)), data = Chwirut2,
           start = list(beta1 = 0,2, beta2 = 0,01, beta3 = 0,01)).
```

En el lado izquierdo de la expresión tenemos la variable respuesta en la función de media (formula) que tiene que ser expresada de forma explícita. En el ejemplo de decrecimiento exponencial $y \approx (exp^{-\beta_1 x})/(\beta_2 + \beta_3 x)$. Note que la especificación requiere que se indique los parámetros de forma explícita (`start`). Los datos a considerar se tienen que explicitar en la sección de datos (`data`).

2.3. Mixtura finita

En esta sección revisaremos algunos conceptos fundamentales relacionados al uso de mixtura finitas de distribuciones. En adelante, seguimos la notación presentada por Frühwirth-Schnatter et al. (2019).

2.3.1. Formulación básica

Frühwirth-Schnatter et al. (2019) presenta la formulación básica del modelo de mixtura finita asumiendo que los datos y son modelados por la combinación convexa de componentes cada uno especificado con función de densidad de probabilidad $f(\cdot)$ de la siguiente manera:

$$y \sim \sum_{j=1}^g \eta_j f_j(\cdot | \theta_j),$$

donde $\sum_{j=1}^g \eta_j = 1$, $\eta_j > 0$. En una mixtura, los componentes de densidad $f(\cdot)$ se asumen conocidos mientras que los parámetros θ_j y η_j se asumen desconocidos. El número total de componentes g también puede ser desconocido.

Considerar el uso de un modelo de mixtura usualmente proviene de la exploración visual de los datos donde un histograma muestra la existencia de multimodalidad. Sin embargo, es importante precisar que no existe una relación sencilla entre el número de modas asociados a una densidad y el número de componentes g de un modelo de mixtura. Como se puede ver en

³Self-starter functions.

la Figura (2.3), tres componentes en mixtura corresponden a una sola moda. En general, el número de modas podrá variar entre modelos siempre y cuando los θ_j sean lo suficientemente diferentes. De manera similar, se pueden considerar modelos de mixtura bajo la creencia de

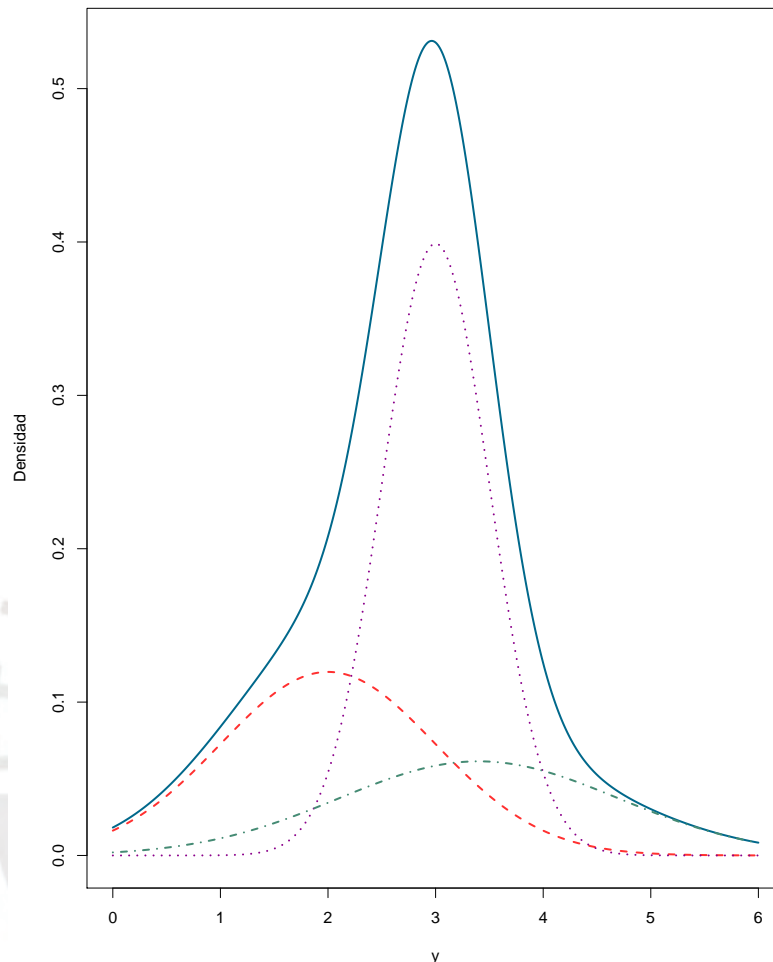


Figura 2.3: Ejemplo de mixtura de tres componentes con densidad normal, es decir de $0,3 \times N(2, 1) + 0,5 \times N(3, 0,5^2) + 0,2 \times N(3,4, 1,3^2)$

que los datos están agrupados conforme a conglomerados, con observaciones dentro de cada conglomerado provenientes de alguna distribución paramétrica. Sin embargo, de nuevo, no existe una relación sencilla entre los componentes ajustados mediante un modelo de mixtura y la existencia de conglomerados en la data original. Más aún, el término conglomerado implica un nivel de homogeneidad para sus elementos dentro del mismo y separación respecto a los demás conglomerados sin asumir una distribución específica. En este sentido, por ejemplo, el modelo de mixtura podría requerir un número de componentes mayor al de conglomerados simplemente por el hecho de que la forma de la función de distribución asumida es incorrecta, de tal forma que cada “conglomerado” de los datos pueda requerir más de un componente para describirlo correctamente.

En un modelo de mixtura, determinar el orden g , el número de componentes, no es

trivial. Más aún, su relevancia resalta pues afecta la estimación de parámetros y una elección g equivocada puede conllevar a una estimación pobre de la densidad. A fin de considerar el orden adecuado, se sugiere el uso de criterios de información como el AIC o el BIC.

2.3.2. Estimación de parámetros

Por lo general, para la estimación de los parámetros $\boldsymbol{\theta} = (\theta_1, \dots, \theta_g)^\top$ y $\boldsymbol{\eta} = (\eta_1, \dots, \eta_g)^\top$ se asume que $\mathbf{y} = (y_1, \dots, y_n)^\top$ son independientes e idénticamente distribuidas. Por tanto, la verosimilitud queda definida de la siguiente forma:

$$p(\mathbf{y}|\boldsymbol{\theta}, \boldsymbol{\eta}) = \prod_{i=1}^n \sum_{j=1}^g \eta_j f_j(y_i|\theta_j). \quad (2.12)$$

Más precisamente, se procura trabajar con el logaritmo de la ecuación (2.12), que sería la log-verosimilitud, dada por:

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^n \log \left[\sum_{j=1}^g \eta_j f_j(y_i|\theta_j) \right]. \quad (2.13)$$

Estimar directamente a partir de la expresión dada en (2.13) en forma analítica suele ser complicado. Una alternativa es usando métodos numéricos. Según Frühwirth-Schnatter et al. (2019), otra alternativa para estimar (2.13) es desde un punto de vista más probabilístico. Supongamos que la población de donde la muestra está siendo obtenida es heterogénea y dividida en múltiples grupos $j = 1, 2, \dots, g$ conforme a las proporciones η_j . Cuando la muestra se obtiene del grupo j se asume que proviene de la distribución con densidad $f(\cdot|\theta_j)$. Luego podemos asumir que las observaciones \mathbf{y} son seleccionadas siguiendo dos pasos, primero se selecciona el grupo z_i de $\mathbf{z} = (z_1, \dots, z_n)^\top$ al que pertenece dentro de cada grupo $j = 1, 2, \dots, g$, con $P(z_i = j) = \eta_j$ y posteriormente, dado \mathbf{z} , \mathbf{y} es obtenido de la densidad $f(\cdot|\theta_z)$. Este proceso de ‘dos etapas’ implica que para la muestra aleatoria $\mathbf{y} = (y_1, \dots, y_n)^\top$ tenemos

$$y_i|z_i \sim f_{z_i}(\cdot|\theta_{z_i}) \text{ con } P(z_i = j) = \eta_j,$$

donde z_1, \dots, z_n son variables aleatorias independientes (no observadas) consideradas *latentes*. En un contexto de mixtura estas variables son llamadas también variables de asignación.

La aproximación probabilística representa exactamente el modelo presentado en (2.12) y puede ser visto como una representación alternativa para la estimación de máxima verosimilitud. En la práctica, la representación del modelo con variables latentes es clave para la estimación de los parámetros. Es claro que si se conocen los valores z_i se tendrían muestras independientes de cada componente j de tal forma que la estimación de θ_j sería directa. De otro lado, si se conocieran los parámetros θ_j , asignar observaciones y_i a los diferentes componentes de mixtura se podría realizar asignando un z_i que maximice $f(y_i|\theta_j)$. El algoritmo EM realiza alternadamente estos pasos para la estimación de parámetros.

Como se mencionó anteriormente, la función de verosimilitud para un modelo de mixtura es difícil de tratar; más a medida que el número de componentes g aumenta junto con el número de parámetros θ_j a estimar. Los métodos de optimización como el de Newton

resultan computacionalmente demandantes y no aseguran que en cada iteración se llegue a la convergencia. En contraste, en el trabajo de Dempster et al. (1977) se muestra que con el algoritmo EM la función de verosimilitud observada converge a un punto óptimo con cada iteración. En ese sentido, posee características deseables como proceso de optimización, entre ellas, eficiencia computacional.

2.3.3. Descripción del algoritmo EM

Recordemos que la forma de la verosimilitud para la información observada $\mathbf{y} = (y_1, \dots, y_n)^\top$ definida por (2.12), se define:

$$\ell_o(\boldsymbol{\theta}) = \sum_{i=1}^n \log \left[\sum_{j=1}^g \eta_j f_j(y_i | \theta_j) \right],$$

que llamaremos verosimilitud observada, implica el logaritmo de una suma. La idea del algoritmo EM en este contexto es hacer uso de las etiquetas latentes z_i maximizando en cada iteración la esperanza condicional de los datos completos o función de verosimilitud completa definida como

$$\ell_c(\boldsymbol{\theta}, \mathbf{z}) = \sum_{i=1}^n \sum_{j=1}^g z_{ij} \log(\eta_j f_j(y_i | \theta_j)),$$

dado los datos observados y los valores actuales de los parámetros $\boldsymbol{\theta}^{(s)}$, hasta la obtención de sus estimadores de máxima verosimilitud. El algoritmo EM hace uso de la siguiente relación que conecta la verosimilitud observada con la verosimilitud completa,

$$\ell_o(\boldsymbol{\theta}) = \ell_c(\boldsymbol{\theta}, \mathbf{z}) - \sum_{j=1}^g \sum_{i=1}^n z_{ij} \log(\tau_{ij}),$$

donde τ_{ij} es la probabilidad condicional de que y_i se asocie con el componente j , para $i = 1, \dots, n$ y $j = 1, \dots, g$, dados los valores actuales de los parámetros, $\boldsymbol{\theta}^{(s)}$. Esto conlleva a

$$\ell_o(\boldsymbol{\theta}) = Q(\boldsymbol{\theta} | \boldsymbol{\theta}^{(s)}) - H(\boldsymbol{\theta} | \boldsymbol{\theta}^{(s)}),$$

donde $H(\boldsymbol{\theta} | \boldsymbol{\theta}^{(s)}) = \sum_{j=1}^g \sum_{i=1}^n \tau_{ij}^{(s)} \log \tau_{ij}$ y $\boldsymbol{\theta}^{(s)}$ es el valor actual de los parámetros. Por tanto, partiendo de unos valores iniciales $\boldsymbol{\theta}^{(0)}$, el algoritmo EM puede ser resumido de la siguiente manera:

Paso E: Calcular $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(s)}) = E(\ell_c(\boldsymbol{\theta}, \mathbf{z}) | \mathbf{y}, \boldsymbol{\theta}^{(s)})$ donde el valor esperado es calculado sobre $p(\mathbf{z} | \mathbf{y}, \boldsymbol{\theta}^{(s)})$.

Paso M: Determinar $\boldsymbol{\theta}^{(s+1)} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(s)})$. Así tenemos que:

$$Q(\boldsymbol{\theta} | \boldsymbol{\theta}^{(s)}) = \sum_{i=1}^n \sum_{j=1}^g \tau_{ij}^{(s)} \{ \log(\eta_j) + \log f_j(y_i | \theta_j) \},$$

con $\tau_{ij}^{(s)}$ como la probabilidad condicional de que y_i provenga del componente j , para $i = 1, \dots, n$ y $j = 1, \dots, g$, dados los parámetros $\boldsymbol{\theta}^{(s)}$. Así, el paso E se reduce a calcular estas

probabilidades condicionales:

$$\tau_{ij}^{(s)} = \frac{\eta_j^{(s)} f_j(y_i | \theta_j^{(s)})}{\sum_{j=1}^G \eta_j^{(s)} f_j(y_i | \theta_j^{(s)})},$$

para $i = 1, \dots, n$ y $j = 1, \dots, g$. Para el paso M descrito anteriormente, se tiene que considerar la forma del modelo de mixtura.

2.4. Distribución Senh-normal y Senh-normal/independiente

A continuación presentamos una breve descripción de la distribución Senh-normal (SN) para luego presentar una generalización suya, la distribución Senh-normal/independiente (SNI).

2.4.1. La distribución Senh-normal

Como se mencionó en la introducción, es común y a veces conveniente realizar transformaciones a variables que siguen una distribución normal. Una aplicación del método de transformación es dada por:

$$Z = \nu + \delta f(Y; \mu, \sigma),$$

donde $Z \sim N(0, 1)$ y la función $f(\cdot)$ es una función monótona. Esta transformación depende de cuatro parámetros ν, δ, μ y σ y se dice que los parámetros μ y σ son parámetros de locación y escala respectivamente, mientras que ν y δ son parámetros de forma. Ahora consideremos la siguiente transformación:

$$Z = \nu + \frac{2}{\alpha} \operatorname{senh} \left(\frac{Y - \mu}{\sigma} \right) \sim N(0, 1).$$

Despejando la variable Y obtenemos:

$$Y = \mu + \sigma \operatorname{arcsenh} \left(\frac{\alpha(Z - \nu)}{2} \right). \quad (2.14)$$

Decimos que Y sigue una distribución senh-normal (SN), cuya notación será $Y \sim \operatorname{SN}(\alpha, \mu, \sigma, \nu)$. Asimismo, su forma reducida cuando $\nu = 0$ sera denotada por, $Y \sim \operatorname{SN}(\alpha, \mu, \sigma)$. La función de distribución acumulada esta definida por

$$F(y) = \Phi \left[\frac{2}{\alpha} \operatorname{senh} \left(\frac{y - \mu}{\sigma} \right) \right], \alpha > 0, \sigma > 0, \mu \in \mathbb{R}, y \in \mathbb{R},$$

donde $\Phi(\cdot)$ es la función de distribución acumulada normal estándar. A su vez, la fdp de Y está definida como

$$f_Y(y) = \frac{2}{\alpha \sigma \sqrt{2\pi}} \cosh \left(\frac{y - \mu}{\sigma} \right) \exp \left[-\frac{2}{\alpha^2} \operatorname{senh}^2 \left(\frac{y - \mu}{\sigma} \right) \right].$$

La Figura 2.4 muestra la forma de la función de densidad de una SN para diferentes valores de los parámetros α, μ y σ . Se puede observar que la distribución SN es unimodal siempre y cuando $\alpha \leq 2$ y para $\alpha > 2$ la distribución es bimodal. Asimismo se observa que

la distribución SN es simétrica. Según Rieck y Nedelman (1991) para relacionar una variable

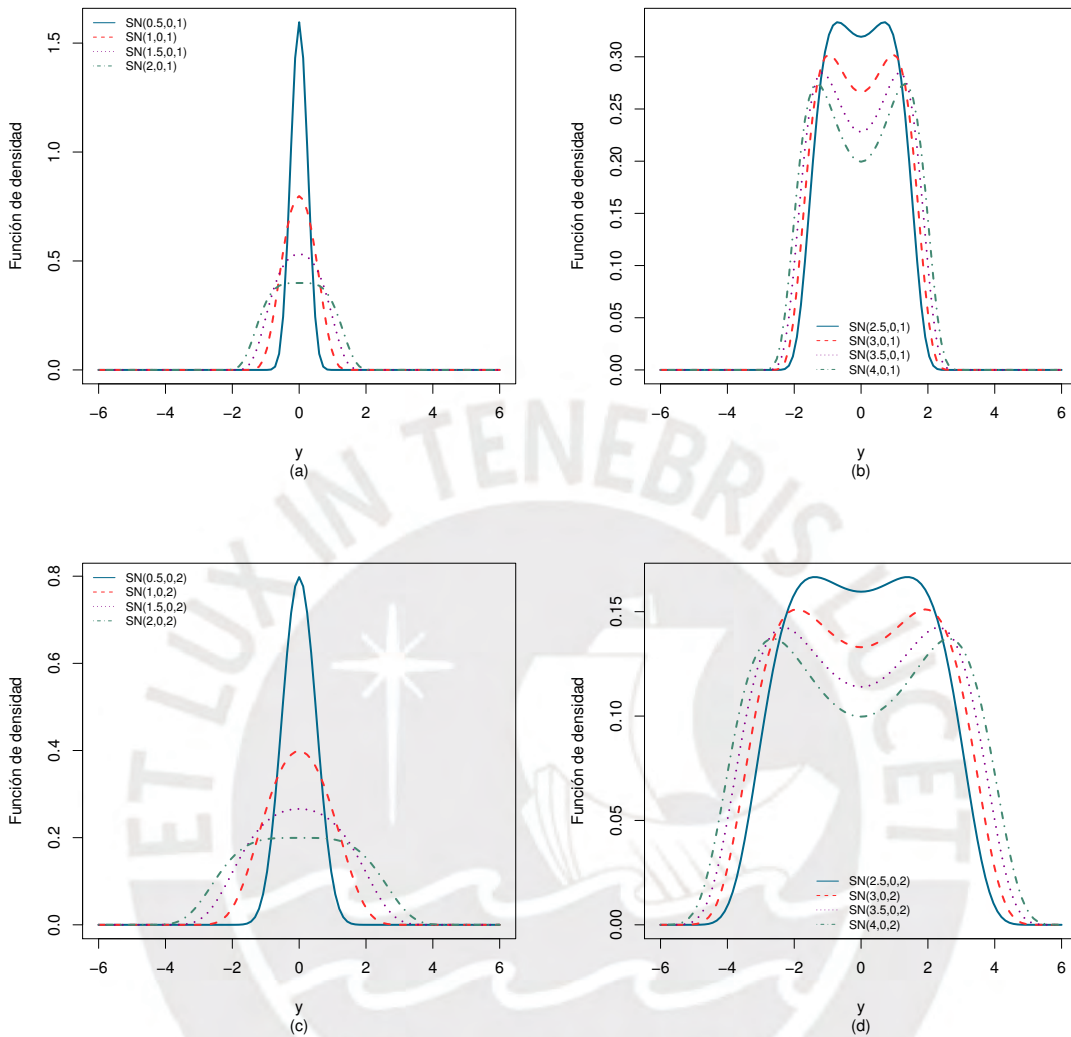


Figura 2.4: Función de densidad de probabilidad SN para distintos valores de los parámetros α , μ y σ respectivamente.

con distribución SN y la distribución BS se debe considerar $Y \sim \text{SN}(\alpha, \mu, \sigma = 2)$, entonces $T = e^Y$ sigue una distribución BS con parámetro $\alpha > 0$ y parámetro de escala $\beta = e^\mu > 0$, $T \sim \text{BS}(\alpha, \beta)$. Por este motivo el modelo SN es también llamado log-BS. Así, consideremos la variable $T \sim \text{BS}(\alpha, \beta)$. Entonces la función de densidad acumulada de $Y = \log(T)$ queda

definida por,

$$\begin{aligned}
P(Y \leq y) &= P(T \leq \exp(y)) \\
&= \Phi \left[\frac{1}{\alpha} \sqrt{\frac{\exp(y)}{\beta}} - \sqrt{\frac{\beta}{\exp(y)}} \right] \\
&= \Phi \left[\frac{1}{\alpha} \sqrt{\exp(y - \log \beta)} - \sqrt{\frac{1}{\exp(y - \log \beta)}} \right] \\
&= \Phi \left[\frac{1}{\alpha} \frac{\exp(y - \log \beta) - \exp(-(y - \log \beta))}{2} \right] \\
&= \Phi \left[\frac{2}{\alpha} \sinh \left(\frac{y - \log \beta}{2} \right) \right].
\end{aligned}$$

En la que $P(Y \leq y)$ es la distribución acumulada de una variable $\text{SN}(\alpha, \log(\beta), 2)$.

2.4.2. La distribución Senh-normal/independiente

Consideremos la expresión para una variable $Y \sim \text{SN}(\alpha, \mu, \sigma)$ definida por la ecuación (2.14) para cuando $\nu = 0$,

$$Y = \mu + \sigma \operatorname{arcsenh}(\alpha Z/2).$$

Ahora consideremos la siguiente la relación:

$$Z = U^{-1/2} Z_0, \quad (2.15)$$

donde $Z_0 \sim N(0, 1)$ y U es una variable aleatoria positiva independiente de Z_0 . En este caso, se dice que Z tiene una distribución Normal/Independiente (NI). Vilca et al. (2015) presentó la extensión del modelo SN, de la siguiente forma:

$$Y = \mu + \sigma \operatorname{arcsenh} \left(\frac{\alpha Z}{2} \right) = \mu + \sigma \operatorname{arcsenh} \left(\frac{\alpha U^{-1/2} Z_0}{2} \right). \quad (2.16)$$

Según la representación estocástica de Y en (2.16), diremos que Y tiene una distribución Senh-Normal/Independiente (SNI) con parámetro de forma $\alpha > 0$, parámetro de localización μ y parámetro de escala $\sigma > 0$. Usaremos la notación $Y \sim \text{SNI}(\alpha, \mu, \sigma; H)$ donde $H(\cdot)$ es la función de densidad acumulada de U (indexada por el escalar o vector de parámetros ν). Cabe mencionar que si $U = 1$, la distribución SNI se reduce a la distribución SN (Rieck, 1989). A continuación la fdp de Y viene dada por:

$$f_Y(y) = \frac{1}{\sigma} \phi_{\text{NI}}(\xi_{2y}) \xi_{1y} = \frac{1}{\sigma} \int_0^\infty \phi(\xi_{2y}; 0, u^{-1}) dH(u) \xi_{1y}, \quad y \in \mathbb{R}, \quad (2.17)$$

donde $\phi_{\text{NI}}(\cdot)$ es la fdp de Z dada (2.15), $\xi_{1y} = \xi_1(y; \alpha, \mu, \sigma) = (2/\alpha) \cosh\left(\frac{y-\mu}{\sigma}\right)$, $\xi_{2y} = \xi_2(y; \alpha, \mu, \sigma) = (2/\alpha) \sinh\left(\frac{y-\mu}{\sigma}\right)$ y $\phi(\cdot)$ denota la fdp de una normal estándar. Algunos casos especiales de la familia SNI son basados en las distribuciones normal contaminada (NC), slash (SL) y t -Student (St). Las distribuciones resultantes son las llamadas Senh-normal contaminada (SCN), Senh-slash (SSL) y Senh- t -Student (SSt). El ejercicio de simulación

presentado en el Capítulo 4 esta basado en la distribución SSt.



Capítulo 3

Regresión no lineal cuyo error sigue una MF-SNI

En esta sección haremos una breve descripción de una regresión lineal en donde el error sigue una distribución de mixtura finita de distribuciones SN siguiendo las ideas de (Bartolucci y Scaccia, 2005). También se describirá el modelo SNI para posteriormente definir un modelo de regresión no lineal en la que se considera que el error sigue una distribución de mixtura finita de distribuciones SNI (MF-SNI).

3.1. Regresión lineal con una mixtura finita de distribución SN

La metodología propuesta por Bartolucci y Scaccia (2005) sugiere un planteamiento de regresión de la siguiente forma, sea:

$$Y_i = \beta_0 + \mathbf{x}_i \boldsymbol{\beta} + \epsilon_i, i = 1, \dots, n,$$

donde Y_i representan la variable respuesta, $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})^\top$ corresponde a un vector de variables explicativas de dimensión $(1 \times p)$ y $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)^\top$ representan los parámetros asociados a las covariables. Siguiendo las ideas de Bartolucci y Scaccia (2005) se considera que el error sigue una mixtura finita de distribuciones SN, como se muestra a continuación:

$$\epsilon_i \stackrel{iid}{\sim} \sum_{j=1}^g p_j \phi_{SN}(\epsilon_i | \alpha_j, \mu_j, \sigma_j = 2), i = 1, \dots, n, j = 1, \dots, g,$$

donde $\phi_{SN}(\cdot)$ corresponde a la fdp de una distribución SN($\alpha_j, \mu_j, \sigma_j = 2$). Para este modelo, la suma de los pesos de cada componente de mixtura $p_j > 0$, debe cumplir que $\sum_{j=1}^g p_j = 1$ y μ_j debe satisfacer condiciones de identificabilidad definida como $\sum_{j=1}^g p_j \mu_j = 0$. Por ser la distribución SN lineal y considerando que $E(\epsilon_i) = 0$, la función de densidad de probabilidad (fdp) de y_i viene dada por:

$$f_{Y_i}(y_i | \boldsymbol{\theta}) = \sum_{j=1}^g p_j \phi_{SN}(y_i | \alpha_j, \mu_{ij}, \sigma_j = 2), \mu_{ij} = \beta_0 + \mathbf{x}_i \boldsymbol{\beta} + \mu_j = v_j + \mathbf{x}_i \boldsymbol{\beta},$$

donde, siguiendo a Bartolucci y Scaccia (2005) y por ser el caso lineal de regresión, $\mu_{ij} = v_j + \mathbf{x}_i \boldsymbol{\beta}$ y $v_j = \beta_0 + \mu_j$ y $\boldsymbol{\theta} = (\alpha_1, \dots, \alpha_g, \boldsymbol{\beta}, p_1, \dots, p_{g-1}, v_1, \dots, v_g)^\top$ como vector de parámetros.

A continuación, se presenta la función de log-verosimilitud dada los datos observados:

$$\ell(\boldsymbol{\theta}|y) = \sum_{i=1}^n \log \left(\sum_{j=1}^g p_j \phi_{\text{SN}}(y_i | \alpha_j, \mu_{ij}, \sigma_j = 2) \right).$$

Para poder obtener los valores de los parámetros que optimizan esta función podemos plantear la variable indicadora latente z_{ij} , para cada i y j de tal manera que:

$$P(z_{ij} = 1) = 1 - P(z_{ij} = 0) = p_j, \quad \sum_{j=1}^g p_j = 1,$$

donde $\mathbf{z}_i = (z_{i1}, \dots, z_{ig})^\top$ con $\mathbf{z}_1, \dots, \mathbf{z}_n$ son vectores aleatorios independientes donde cada uno tiene una distribución fdp

$$f(\mathbf{z}_i) = p_1^{z_{i1}} p_2^{z_{i2}} \dots (1 - p_1 - \dots - p_{g-1})^{z_{ig}},$$

es decir, $\mathbf{z}_i \sim \text{Multinomial}(1, p_1, \dots, p_g)$. Luego podemos definir la distribución condicional de y_i para obtener la información completa como

$$Y_i | z_{ij} = 1 \sim \text{SN}(\alpha_j, \mu_{ij}, \sigma_j = 2).$$

En consecuencia, para el algoritmo EM consideramos la siguiente representación jerárquica del modelo de regresión lineal SN con mixtura:

$$Y_i | z_{ij} = 1 \stackrel{\text{ind}}{\sim} \text{SN}(\alpha_j, \mu_{ij}, \sigma_j = 2) \quad (3.1)$$

$$\mathbf{z}_i \stackrel{\text{iid}}{\sim} \text{Multinomial}(1, p_1, \dots, p_g), i = 1, \dots, n, j = 1, \dots, g. \quad (3.2)$$

3.1.1. Estimación de los parámetros

A partir de las expresiones (3.1) y (3.2) se puede encontrar la siguiente función de verosimilitud aumentada:

$$\ell_c(\boldsymbol{\theta}|y, z) = c + \sum_{i=1}^n \sum_{j=1}^g z_{ij} \left\{ \log(p_j) + \log(\xi_{1ij}) - \frac{1}{2} \xi_{2ij}^2 \right\},$$

donde

$$\xi_{1ij} = \frac{2}{\alpha_j} \cosh \left(\frac{y_i - \mu_{ij}}{2} \right) \text{ y } \xi_{2ij} = \frac{2}{\alpha_j} \sinh \left(\frac{y_i - \mu_{ij}}{2} \right),$$

donde c es una constante independiente de los parámetros recogidos por $\boldsymbol{\theta}$. Definiendo la esperanza condicional

$$\hat{z}_{ij} = \text{E} \left[z_{ij} | \hat{\boldsymbol{\theta}}, y_i \right],$$

y por las propiedades de esperanza condicionada se obtiene que:

$$\hat{z}_{ij} = \frac{\hat{p}_j \phi_{\text{SN}}(y_i | \hat{\alpha}, \hat{\mu}_{ij}, \sigma_j = 2)}{\sum_{j=1}^g \hat{p}_j \phi_{\text{SN}}(y_i | \hat{\alpha}, \hat{\mu}_{ij}, \sigma_j = 2)}.$$

Por lo tanto, para el algoritmo EM la función $Q(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}}^{(k)}) = E[\ell_c(\boldsymbol{\theta})|\mathbf{y}, \hat{\boldsymbol{\theta}}^{(k)}]$ puede ser escrita de la siguiente manera para cada iteración k ,

$$Q(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}}^{(k)}) = c + \sum_{i=1}^n \sum_{j=1}^g \left(\hat{z}_{ij}^{(k)} \left(\log(p_j) + \log(\xi_{1ij}) - \frac{1}{2} \xi_{2ij}^2 \right) \right).$$

Por tanto, el algoritmo EM para los estimadores de $\boldsymbol{\theta}$ queda definido en los siguientes pasos:

Paso E: Dado $\hat{\boldsymbol{\theta}}^{(k)}$, calcular $\hat{z}_{ij}^{(k)}$, para $i = 1, \dots, n; j = 1, \dots, g$.

Paso M1: Fijar $\hat{\boldsymbol{\beta}}^{(k)}, \hat{v}_j^{(k)}$ y actualizar $\hat{p}_j^{(k)}$ y $\hat{\alpha}^{(k)}$. Se deriva $Q(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}}^{(k)})$ respecto a cada parámetro y se iguala a cero para obtener la estimación:

$$\begin{aligned} \hat{p}_j^{(k+1)} &= \frac{1}{n} \sum_{i=1}^n \hat{z}_{ij}^{(k)}, \\ \frac{\partial Q}{\partial \alpha_j} &= \sum_{i=1}^n \hat{z}_{ij}^{(k)} \left\{ \frac{1}{\xi_{1ij}} \left(-\frac{1}{\alpha_j} \xi_{1ij} \right) - \frac{1}{2} \left(-\frac{2}{\alpha_j} \xi_{2ij}^2 \right) \right\}, \\ &= \sum_{i=1}^n \left[-\frac{\hat{z}_{ij}^{(k)}}{\alpha_j} + \frac{\hat{z}_{ij}^{(k)}}{\alpha_j^3} \xi_{3ij}^2 \right], \\ &= -\frac{1}{\alpha_j} \sum_{i=1}^n \hat{z}_{ij}^{(k)} + \frac{1}{\alpha_j^3} \sum_{i=1}^n \hat{z}_{ij}^{(k)} \xi_{3ij}^2, \end{aligned}$$

donde $\frac{\partial \xi_{1ij}}{\partial \alpha_j} = -\frac{1}{\alpha_j} \xi_{1ij}$, $\frac{\partial \xi_{2ij}^2}{\partial \alpha_j} = -\frac{2}{\alpha_j} \xi_{2ij}^2$, $\xi_{2ij}^2 = \frac{4}{\alpha_j^2} \sinh^2 \left(\frac{Y_i - \mu_{ij}}{2} \right) = \frac{1}{\alpha_j^2} \left[2 \sinh \left(\frac{Y_i - \mu_{ij}}{2} \right) \right]^2$, $\xi_{2ij}^2 = \frac{1}{\alpha_j^2} \xi_{3ij}^2$, $\xi_{3ij} = 2 \sinh \left(\frac{Y_i - \mu_{ij}}{2} \right)$. Igualando a cero las derivadas parciales anteriores se obtiene:

$$\begin{aligned} \frac{1}{\alpha_j} \sum_{i=1}^n \hat{z}_{ij}^{(k)} &= \frac{1}{\alpha_j^3} \sum_{i=1}^n \hat{z}_{ij}^{(k)} \xi_{3ij}^2, \\ \alpha_j^{(k+1)} &= \left(\frac{\sum_{i=1}^n \hat{z}_{ij}^{(k)} \xi_{3ij}^2}{\sum_{i=1}^n \hat{z}_{ij}^{(k)}} \right)^{1/2}. \end{aligned}$$

Paso M2: Actualizar $\hat{\boldsymbol{\beta}}^{(k)}$ y $\hat{v}^{(k)}$. Fijar $\hat{p}^{(k+1)}, \hat{\alpha}^{(k+1)}$ y maximizar la función $Q(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}}^{(k)})$ de tal forma que:

$$\hat{\boldsymbol{\beta}}^{(k+1)} = \arg \max_{\boldsymbol{\beta}} Q \left(\hat{\alpha}^{(k+1)}, \boldsymbol{\beta}, \hat{p}^{(k+1)}, \hat{v}^{(k)} | \hat{\boldsymbol{\theta}}^{(k)} \right),$$

$$\hat{v}^{(k+1)} = \arg \max_v Q \left(\hat{\alpha}^{(k+1)}, \hat{\boldsymbol{\beta}}^{(k+1)}, \hat{p}^{(k+1)}, \hat{v}^{(k)} | \hat{\boldsymbol{\theta}}^{(k)} \right).$$

Es imprescindible notar que es posible que $\hat{\boldsymbol{\beta}}^{(k+1)}$ y $\hat{v}^{(k+1)}$ no tengan expresiones cerradas por lo que para la optimización se requieran métodos numéricos. Por la propiedad de la invariancia de los estimadores de máxima verosimilitud y conforme a Bartolucci y Scaccia (2005) los estimadores de β_0 y μ_j tienen la siguiente forma conforme a la parametrización

presentada inicialmente:

$$\hat{\beta}_0^{(k+1)} = \sum_{j=1}^g \hat{p}_j^{(k+1)} \hat{v}_j^{(k+1)} \quad y \quad \hat{\mu}_j = \hat{v}_j - \hat{\beta}_0.$$

El proceso continua hasta que se encuentre una forma de convergencia o se satisfaga una regla de parada impuesta sobre el valor de la función de verosimilitud aumentada.

3.2. Modelo de regresión no lineal distribuciones SNI

En esta sección se presenta un modelo de regresión no lineal con distribución SNI planteado por Vilca et al. (2015). Este modelo, junto con el presentado anteriormente, servirán de referencia para poder derivar un modelo de regresión no lineal basado en una mixtura de distribuciones SNI en el error.

Recordemos la expresión de la fdp de Y en (2.17):

$$f_Y(y) = \frac{1}{\sigma} \phi_{\text{NI}}(\xi_{2y}) \xi_{1y} = \frac{1}{\sigma} \int_0^\infty \phi(\xi_{2y}; 0, u^{-1}) dH(u) \xi_{1y}, \quad y \in \mathbb{R},$$

donde $\phi_{\text{NI}}(\cdot)$ es la fdp de Z , $H(\cdot)$ es la función de densidad acumulada de U (indexada por el escalar o vector de parámetros ν), $\xi_{1y} = \xi_1(y; \alpha, \mu, \sigma) = (2/\alpha) \cosh\left(\frac{y-\mu}{\sigma}\right)$, $\xi_{2y} = \xi_2(y; \alpha, \mu, \sigma) = (2/\alpha) \sinh\left(\frac{y-\mu}{\sigma}\right)$ y $\phi(\cdot; 0, u^{-1})$ denota la función de densidad $N(0, u^{-1})$. Vilca et al. (2015) indica que si $Y \sim \text{SNI}(\alpha, \mu, \sigma; H)$ entonces, $W = a + bY \sim \text{SNI}(\alpha, a + b\mu, |b|\sigma; H)$, donde $a \in \mathbb{R}$ y $b \neq 0$, por lo que la distribución SNI es cerrada a cualquier transformación lineal. Ahora bien, de la representación estocástica en (2.16) es posible ver que la distribución condicional de Y dado $U = u$, sigue una distribución SN, $Y|U = u \sim \text{SN}(\alpha_u, \mu, \sigma)$ donde $\alpha_u = u^{-1/2}\alpha$. Conforme a Balakrishnan et al. (2009), se puede obtener la distribución condicional de U , dado $Y = y$, que viene dada por

$$h_{U|Y}(u|y) = \frac{\phi(\xi_{2y}; 0, u^{-1}) h_U(u)}{\phi_{\text{NI}}(\xi_{2y})}, \quad u > 0, \quad (3.3)$$

donde $h_U(\cdot)$ es la fdp de U . Esta distribución condicional es usada para implementar el algoritmo EM para la estimación de los parámetros de regresión.

3.2.1. El modelo no lineal

Vilca et al. (2015) procede a introducir el modelo de regresión no lineal SNI (RNL-SNI). Consideremos el siguiente modelo de regresión

$$Y_i = f(\mathbf{x}_i; \boldsymbol{\beta}) + \varepsilon_i, \quad i = 1, \dots, n,$$

donde y_i es la variable respuesta y \mathbf{x}_i es un vector de dimensión $m \times 1$ ($m \leq p$) de variables explicativas asociadas con i -ésima observación, $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)^\top$ es un vector de parámetros no lineal desconocido y $f(\mathbf{x}_i, \boldsymbol{\beta})$ es una función no lineal. Además, se asume que

$$\varepsilon_i \stackrel{iid}{\sim} \text{SNI}(\alpha, 0, \sigma = 2; H), \quad i = 1, \dots, n.$$

Por tanto, recordando la propiedad de linealidad de la distribución SNI sabemos que $y_i \sim$

$\text{SNI}(\alpha, f(\boldsymbol{\beta}; \mathbf{x}_i), \sigma = 2; H)$ y por (2.17) la función de densidad queda determinada como

$$f_{Y_i}(y_i) = \frac{1}{2} \phi_{\text{NI}}(\xi_{2i}) \xi_{1i}, \quad (3.4)$$

donde $\xi_{1i} = \xi_1(y_i; \alpha, \boldsymbol{\beta})$ y $\xi_{2i} = \xi_2(y_i; \alpha, \boldsymbol{\beta})$ como en la ecuación (2.17) con $\sigma = 2$ y $\mu_i = f(\mathbf{x}_i, \boldsymbol{\beta})$. Dependiendo de la distribución de U la distribución de y adquiere diversas propiedades.

3.2.2. Estimación de máxima verosimilitud y el algoritmo EM

De (2.16) el modelo no lineal puede ser expresado de la siguiente forma jerárquica:

$$Y_i | U_i = u_i \sim \text{SN} \left(u_i^{-1/2} \alpha, f(\mathbf{x}_i; \boldsymbol{\beta}), \sigma = 2 \right),$$

$$U_i \sim h_U(u_i), \quad i = 1, \dots, n.$$

Se asume que el vector de parámetros $\boldsymbol{\nu}$ sobre la fdp de $h_U(\cdot)$ son conocidos como Vilca et al. (2015). Ahora, de manera similar al caso lineal, se introduce una variable latente no observada de la forma $\mathbf{u} = (u_1, \dots, u_n)^\top$, por tanto la información completa queda expresada por $\mathbf{y}_c = (\mathbf{y}^\top, \mathbf{u}^\top)$, así la función de verosimilitud aumentada que sigue del modelo jerárquico queda definida por

$$\ell_c(\boldsymbol{\theta} | \mathbf{y}_c) = -\frac{n}{2} \log(8\pi) + \sum \log(\xi_{1i}) - \frac{1}{2} \sum u_i \xi_{2i}^2,$$

donde ξ_{1i} y ξ_{2i} quedan definidas como en (3.4) y $\boldsymbol{\theta} = (\boldsymbol{\beta}^\top, \alpha)^\top$ son los parámetros a estimar. La función $Q(\boldsymbol{\theta} | \hat{\boldsymbol{\theta}}^{(k)})$ para las esperanzas condicionales queda definida como

$$Q(\boldsymbol{\theta} | \hat{\boldsymbol{\theta}}^{(k)}) = -\frac{n}{2} \log(8\pi) + \sum \log(\xi_{1i}^{(k)}) - \frac{1}{2} \sum \hat{u}_i^{(k)} \xi_{2i}^{2(k)},$$

donde $\hat{\boldsymbol{\theta}}^{(k)} = \left(\hat{\boldsymbol{\beta}}^{\top(k)}, \hat{\alpha}^{(k)} \right)^\top$ representa el estimado de $\boldsymbol{\theta}$ en la iteración k -ésima y $\hat{u}_i = E[U_i | y_i, \boldsymbol{\theta} = \hat{\boldsymbol{\theta}}]$. Por tanto sigue que:

Paso E: Dado $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}^{(k)}$, calcular $\hat{u}_i^{(k)}$, para $i = 1, \dots, n$.

Paso M: Actualizar $\hat{\boldsymbol{\theta}}^{(k+1)}$ a través de la optimización de $Q(\boldsymbol{\theta} | \hat{\boldsymbol{\theta}}^{(k)})$ sobre $\boldsymbol{\theta}$ para llegar a las siguientes expresiones:

$$\hat{\alpha}^{(k+1)} = \left(\frac{4}{n} \sum \hat{u}_i^{(k)} \sinh^2 \left(\frac{y_i - f(\mathbf{x}_i; \hat{\boldsymbol{\beta}}^{(k)})}{2} \right) \right)^{1/2} \quad \text{y} \quad \hat{\boldsymbol{\beta}}^{(k+1)} = \arg \max_{\boldsymbol{\beta}} Q(\boldsymbol{\beta}, \hat{\alpha}^{(k+1)} | \hat{\boldsymbol{\theta}}^{(k)}).$$

Las iteraciones se repiten hasta llegar a la convergencia o satisfacer una regla de parada sobre el valor de la forma $\left| \ell(\hat{\boldsymbol{\theta}}^{(k+1)}) - \ell(\hat{\boldsymbol{\theta}}^{(k)}) \right|$, que sea pequeño. Vilca et al. (2015) recomienda el uso de valores iniciales útiles ante la posible existencia de óptimos locales así como el uso de varios valores iniciales para corroborar los resultados.

3.3. El modelo no lineal con mixtura

Para plantear el modelo objetivo podemos extender el modelo no lineal basado en Vilca et al. (2015) presentado en la sección (3.2) a uno considerando una mixtura de distribuciones SNI. Para este fin, sea el modelo,

$$Y_i = f(\mathbf{x}_i, \boldsymbol{\beta}) + \epsilon_i, \quad i = 1, \dots, n,$$

donde Y_i representan la variable respuesta, $\mathbf{x}_i = (x_{i1}, \dots, x_{im})^\top$, $m \leq p$ corresponde a un vector de variables explicativas de dimensión $m \times 1$, $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)^\top$ representa el vector de parámetros a estimar y $f(\mathbf{x}, \boldsymbol{\beta})$ es una función no lineal. Al tratarse de un modelo de regresión, asumiremos que el término de error sigue una distribución conforme a la siguiente mixtura:

$$\epsilon_i \stackrel{iid}{\sim} \sum_{j=1}^g p_j \phi_{\text{SNI}}(\epsilon_i | \alpha_j, \mu_j, \sigma_j = 2; H), \quad i = 1, \dots, n, \quad j = 1, \dots, g,$$

donde $\phi_{\text{SNI}}(\cdot)$ corresponde a la fdp de una distribución $\text{SNI}(\alpha_j, \mu_j, \sigma_j = 2; H)$. Para este modelo, se tiene que cumplir que $p_j > 0$ y $\sum_{j=1}^g p_j = 1$. Por las propiedades de la distribución SNI, la fdp de Y_i viene dada por:

$$f_{Y_i}(y_i | \boldsymbol{\theta}) = \sum_{j=1}^g p_j \phi_{\text{SNI}}(y_i | \alpha_j, \mu_{ij}, \sigma_j = 2; H), \quad \text{con } \mu_{ij} = \mu_j + f(x_i, \boldsymbol{\beta}),$$

donde $\boldsymbol{\theta} = (\alpha_1, \dots, \alpha_g, \boldsymbol{\beta}^\top, p_1, \dots, p_{g-1}, \mu_1, \dots, \mu_j, \boldsymbol{\nu})^\top$ como vector de parámetros. En este punto es necesario precisar que el parámetro $\boldsymbol{\nu}$ esta asociado a la distribución H . Cabe mencionar que supondremos que $\boldsymbol{\nu}_1 = \dots = \boldsymbol{\nu}_g$ tal como se realizaron en los trabajos de Benites et al. (2019) y Zeller et al. (2019), como resultado de practicidad computacional. En consecuencia, la distribución de log-verosimilitud para los datos observados es:

$$\ell(\boldsymbol{\theta} | y) = \sum_{i=1}^n \log \left(\sum_{j=1}^g p_j \phi_{\text{SNI}}(y_i | \alpha_j, \mu_{ij}, \sigma_j = 2; H) \right). \quad (3.5)$$

Para poder obtener los valores de los parámetros que optimizan esta función podemos plantear la variable indicadora latente z_{ij} , para cada i y j de tal manera que:

$$P(z_{ij} = 1) = 1 - P(z_{ij} = 0) = p_j, \quad \sum_{j=1}^g p_j = 1,$$

donde $\mathbf{z}_i = (z_{i1}, \dots, z_{ig})^\top$ con $\mathbf{z}_1, \dots, \mathbf{z}_n$ son vectores aleatorios independientes donde cada uno tiene una fdp

$$f(\mathbf{z}_i) = p_1^{z_{i1}} p_2^{z_{i2}} \dots (1 - p_1 - \dots - p_{g-1})^{z_{ig}},$$

es decir, $\mathbf{z}_i \sim \text{Multinomial}(1, p_1, \dots, p_g)$. Luego podemos definir la distribución condicional de Y_i para obtener la información completa como

$$Y_i | z_{ij} = 1 \sim \text{SNI}(\alpha_j, \mu_{ij}, \sigma_j = 2; H).$$

Sin embargo, recordemos que como muestra Vilca et al. (2015), de la representación estocástica en (2.16) es posible ver que la distribución condicional de Y_i dado $U = u_i$, sigue una distribución SN, $Y_i|U = u_i \sim \text{SN}(\alpha_u, \mu, \sigma)$ donde $\alpha_u = u^{-1/2}\alpha$. En este sentido, la representación jerárquica del modelo no lineal queda definida como:

$$Y_i|u_i, z_{ij} = 1 \sim \text{SN}\left(u_i^{-1/2}\alpha_j, \mu_{ij}, \sigma_j = 2, H\right), \quad (3.6)$$

$$U_i|z_{ij} = 1 \sim h(u_i), \quad (3.7)$$

$$\mathbf{z}_i \sim \text{Multinomial}(1, p_1, p_2, \dots, p_g), \quad i = 1, \dots, n, j = 1, \dots, g, \quad (3.8)$$

donde $h(\cdot)$ representa la fdp de una distribución conocida e iid de U . En este sentido, la representación jerárquica presentada trata de recoger la información observada, \mathbf{y} , así como la información no observada que completa la información, mediante \mathbf{z} y \mathbf{u} .

3.3.1. Estimación de los parámetros

A partir de las expresiones (3.6), (3.7) y (3.8) se puede derivar la siguiente función de verosimilitud aumentada con información completa:

$$\begin{aligned} L(\boldsymbol{\theta}|y, z) &= \prod_{i=1}^n \prod_{j=1}^g [p_j f_j(y_i|\boldsymbol{\theta}_j)]^{z_{ij}} \\ &= \prod_{i=1}^n \prod_{j=1}^g \left\{ p_j \frac{2u_j^{1/2}}{\alpha_j 2\sqrt{2\pi}} \exp\left[-\frac{2u_i}{\alpha_j^2} \sinh^2\left(\frac{y_i - \mu_{ij}}{2}\right)\right] \cosh\left(\frac{y_i - \mu_{ij}}{2}\right) h(u_i) \right\}^{z_{ij}}, \end{aligned}$$

y la función de log-verosimilitud aumentada queda definida como:

$$\begin{aligned} l(\boldsymbol{\theta}|y, z) &= c + \sum_{i=1}^n \sum_{j=1}^g \left\{ z_{ij} \left(\log p_j + \log(2/\sqrt{2\pi}) + \log u_i^{1/2} - \log \alpha_j - \log 2 \right) \right. \\ &\quad \left. + z_{ij} \left[-\frac{2u_i}{\alpha_j^2} \sinh^2\left(\frac{y_i - \mu_{ij}}{2}\right) \right] + z_{ij} \log \cosh\left(\frac{y_i - \mu_{ij}}{2}\right) + z_{ij} \log h(u_i) \right\}, \end{aligned}$$

donde c es una constante independiente de los parámetros recogidos por $\boldsymbol{\theta}$. Definiendo las cantidades $\hat{z}_{ij} = \text{E}\left[z_{ij}|\hat{\boldsymbol{\theta}}, y_i\right]$, $\hat{s}_{1ij} = \text{E}\left[z_{ij} \log u_i|\hat{\boldsymbol{\theta}}, y_i\right]$, $\hat{z}_{ij}\hat{u}_i = \hat{s}_{2ij} = \text{E}\left[z_{ij}u_i|\hat{\boldsymbol{\theta}}, y_i\right]$ y $\hat{s}_{3ij} = \text{E}\left[z_{ij} \log h(u_i)|\hat{\boldsymbol{\theta}}, y_i\right]$, y por las propiedades de esperanza condicional tenemos:

$$\hat{z}_{ij} = \frac{\hat{p}_j \phi_{\text{SNI}}(y_i|\hat{\alpha}_j, \hat{\mu}_{ij}, \sigma_j = 2; H)}{\sum_{j=1}^g \hat{p}_j \phi_{\text{SNI}}(y_i|\hat{\alpha}_j, \hat{\mu}_{ij}, \sigma_j = 2; H)},$$

y

$$\hat{u}_i = \text{E}\left[u_{ij}|\hat{\boldsymbol{\theta}}, y_i\right].$$

Así, para el algoritmo EM, en la k -ésima iteración, podemos definir la función:

$$Q\left(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}}^{(k)}\right) = \text{E}\left[l(\boldsymbol{\theta})|y, \hat{\boldsymbol{\theta}}^{(k)}\right].$$

Entonces:

$$Q(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}}^{(k)}) = c + \sum_{i=1}^n \sum_{j=1}^g \left\{ \hat{z}_{ij} \left(\log p_j + \log(2/\sqrt{2\pi}) - \log \alpha_j - \log 2 \right) + \frac{1}{2} \mathbb{E} \left[z_{ij} \log u_i | \hat{\boldsymbol{\theta}}, y_i \right] \right. \\ \left. + \mathbb{E} \left[z_{ij} u_i | \hat{\boldsymbol{\theta}}, y_i \right] \left[-\frac{2}{\alpha_j^2} \sinh^2 \left(\frac{y_i - \mu_{ij}}{2} \right) \right] + \hat{z}_{ij} \log \cosh \left(\frac{y_i - \mu_{ij}}{2} \right) + \mathbb{E} \left[z_{ij} \log h(u_i) | \hat{\boldsymbol{\theta}}, y_i \right] \right\}.$$

quedaría definida como:

$$Q(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}}^{(k)}) = c + \sum_{i=1}^n \sum_{j=1}^g \left\{ \hat{z}_{ij} \left(\log p_j + \log(2/\sqrt{2\pi}) - \log \alpha_j - \log 2 \right) + \frac{1}{2} \hat{s}_{1ij} \right. \\ \left. + \hat{s}_{2ij} \left[-\frac{2}{\alpha_j^2} \sinh^2 \left(\frac{y_i - \mu_{ij}}{2} \right) \right] + \hat{z}_{ij} \log \cosh \left(\frac{y_i - \mu_{ij}}{2} \right) + \hat{s}_{3ij} \right\}.$$

De donde podemos proceder con la aplicación del algoritmo EM:

Paso E: Dado $\hat{\boldsymbol{\theta}}^{(k)}$ calcular \hat{s}_{1ij} , \hat{s}_{2ij} , \hat{s}_{3ij} para cada $i = 1, \dots, n$ y $j = 1, \dots, g$.

Pasos M: Actualizar $\hat{\boldsymbol{\theta}}^{(k)}$ maximizando $Q(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}}^{(k)})$ para \hat{p}_j y $\hat{\alpha}_j$, que conlleva a:

$$\hat{p}_j^{(k+1)} = \frac{1}{n} \sum_{i=1}^n \hat{z}_{ij}^{(k)},$$

y

$$\frac{\partial Q}{\partial \alpha_j} = \sum_{i=1}^n \hat{z}_{ij}^{(k)} \left[-\frac{1}{\alpha_j} \right] + \sum_{i=1}^n \hat{s}_{2ij} \left[\frac{4}{\alpha_j^3} \sinh \left(\frac{y_i - \mu_{ij}}{2} \right) \right] = 0,$$

despejando para $\hat{\alpha}_j^{(k+1)}$

$$\hat{\alpha}_j^{(k+1)} = \left(\frac{4 \sum_{i=1}^n \hat{s}_{2ij}^{(k)} \sinh^2 \left(\frac{y_i - \mu_{ij}}{\sigma_j} \right)}{\sum_{i=1}^n \hat{z}_{ij}^{(k)}} \right)^{1/2}.$$

Ahora, fijando $\hat{p}_j^{(k)}$, $\hat{\alpha}_j^{(k)}$ y actualizar $\hat{\boldsymbol{\beta}}^{(k)}$ y $\hat{\mu}_j^{(k)}$ para obtener $\hat{\boldsymbol{\beta}}^{(k+1)}$ y $\hat{\mu}_j^{(k+1)}$ mediante:

$$\hat{\boldsymbol{\beta}}^{(k+1)} = \arg \max_{\boldsymbol{\beta}} Q \left(\hat{\alpha}_j^{(k+1)}, \boldsymbol{\beta}, \hat{p}_j^{(k+1)}, \hat{\mu}_j^{(k)} | \hat{\boldsymbol{\theta}}^{(k)} \right). \quad (3.9)$$

$$\hat{\mu}_j^{(k+1)} = \arg \max_{\mu} Q \left(\hat{\alpha}_j^{(k+1)}, \hat{\boldsymbol{\beta}}^{(k+1)}, \hat{p}_j^{(k+1)}, \mu_j^{(k)} | \hat{\boldsymbol{\theta}}^{(k)} \right). \quad (3.10)$$

Repetimos el proceso iterativo hasta encontrar la convergencia en los parámetros. De ser necesario, es posible considerar un criterio de parada conforme al método de aceleración de Aitken Frühwirth-Schnatter et al. (2019). Este presume el desarrollo de la predicción del máximo valor de log-verosimilitud $\hat{\ell}$. Para su aplicación calculamos el estimador de acelerador de Aitken de la siguiente manera para $\hat{\ell}$ en la iteración s -ésima como

$$\ell_{\infty}^{(s+1)} = \ell^{(s)} + \frac{1}{1 - c^{(s)}} \left[\ell^{(s+1)} - \ell^{(s)} \right],$$

donde

$$c^{(s)} = \frac{(\ell^{(s+1)} - \ell^{(s)})}{(\ell^{(s)} - \ell^{(s-1)})},$$

es la estimación lineal de convergencia de la iteración s -ésima. Es decir, c^s es el estimador de

$$\lim_{s \rightarrow \infty} \frac{\hat{\ell} - \ell^{(s)}}{\hat{\ell} - \ell^{(s-1)}}.$$

cuya existencia se puede demostrar como una constante positiva para la mayoría de aplicaciones del algoritmo EM. Esta se entiende como un límite positivo para la convergencia.

Considerando las propiedades de convergencia de $\ell^{(s)}$ y $\ell^{(\hat{s})}$ podemos plantear la regla de convergencia como.

$$\text{Parar EM si } 0 < (\ell^{(s)} - \ell^{(\hat{s})}) < \epsilon.$$

Con un ϵ pequeño, por ejemplo, $\epsilon < 0,005$.

3.3.2. Valores iniciales

En la literatura estadística es sabido que los modelos de mixtura finita pueden proveer una función de log-verosimilitud multimodal con lo cual puede llegar a dificultar la estimación de máxima verosimilitud (MV). De esta forma el método de estimación de MV a través del algoritmo EM puede no tener una solución máxima global, cuando los valores iniciales no son adecuados. Para buscar obtener unos valores iniciales razonables recomendamos seguir los siguiente pasos:

1. Dividimos la muestra en g grupos usando el algoritmo de partición k -means.
2. El valor inicial de β es obtenido usando mínimos cuadrados no lineales (nls). En R usamos la función `nls`.
3. Luego calculamos la proporción de datos que pertenecen al j -ésimo grupo, o sea $p_j^{(0)}$, $j = 1, \dots, g$, obteniendo así el valor inicial de p_j .
4. Los valores iniciales de μ_j , $j = 1, \dots, g$ es decir $\mu_j^{(0)}$ es obtenido como la media de datos de cada cluster formado por el algoritmo k -means.
5. Para cada componente calculamos los valores iniciales de $\alpha_j^{(0)}$ utilizando el método de momentos modificado *mmeth* propuesto por Ng, Kundu y Balakrishnan (2003) e implementado en el paquete *bssn* (Maehara y Benites, 2015).
6. Para la estimación de los valores iniciales de $\nu^{(0)}$ para la distribución Senh-t-Student calculamos la log-verosimilitud perfilada $\ell(\hat{\theta}_1^{(0)}(\nu), \dots, \hat{\theta}_g^{(0)}(\nu))$, ver (3.5), para la red de valores de ν en la función `initial.value` implementada para esta tesis se consideró un rango de valores entre 1 y 20. Se elije ν que maximice la log-verosimilitud perfilada.

3.3.3. Selección de modelos

Considerando el problema de comparar los modelos RNL-MF-SNI, con diferentes números de componentes. Aquí, usaremos once criterios de información que son mencionados en el siguiente Cuadro:

Cuadro 3.1: Resumen de criterios de selección de modelos. $\ell(\boldsymbol{\theta})$ es la log-verosimilitud. K y n es el número de parámetros y observaciones respectivamente.

Criterio	Fórmula	Fuente
AIC	$-2\ell(\hat{\boldsymbol{\theta}}) + 2k$	Akaike (1974)
BIC	$-2\ell(\hat{\boldsymbol{\theta}}) + k * \log(n)$	Schwarz (1978)
EDC	$-2\ell(\hat{\boldsymbol{\theta}}) + 0,2\sqrt{nk}$	Zhao et al. (2001)
AIC ₄	$-2\ell(\hat{\boldsymbol{\theta}}) + 4k$	Bhansali (1977)
AIC ₅	$-2\ell(\hat{\boldsymbol{\theta}}) + 5k$	Hannan, E.J. (1980)
AIC ₆	$-2\ell(\hat{\boldsymbol{\theta}}) + 6k$	Bozdogan (1987)
CAIC	$-2\ell(\hat{\boldsymbol{\theta}}) + k(\log(n) + 1)$	Bozdogan (1987)
HQIC	$-2\ell(\hat{\boldsymbol{\theta}}) + 2k \log(\log(n))$	Burnham, K. P. y Anderson D. R. (2002)
aBIC	$-2\ell(\hat{\boldsymbol{\theta}}) + k \log(\frac{n+2}{24})$	Sclove (1987)
AIC _c	$AIC + \frac{2k(k+1)}{n-k-1}$	Hurvich, C.M. y Tsai C.L. (1989)
KIC _c	$-2\ell(\hat{\boldsymbol{\theta}}) + n \log(\frac{n}{n-k+1}) + \frac{n((n-k+1)(2k+1)-2)}{(n-k-1)(n-k+1)}$	Cavanaugh (2004)

3.3.4. Estimación de los errores estándar

Una forma de obtener los errores estándar de los estimadores de máxima verosimilitud es aproximar la matriz de covarianza asintótica de $\hat{\boldsymbol{\theta}}$ por la inversa de la matriz de información observada. Sea $\mathbf{J}_o(\boldsymbol{\theta}) = -\partial^2 \ell(\boldsymbol{\theta}|\mathbf{y}) / \partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top$ la matriz de información observada, donde $\ell(\boldsymbol{\theta}|\mathbf{y})$ es la función de log-verosimilitud, que es obtenido usando la ecuación (3.5). En este trabajo haremos uso del método propuesto por Basford et al. (1997), el cual consiste en aproximar la inversa de la matriz de covarianza por:

$$\mathbf{J}_o(\hat{\boldsymbol{\theta}}) = \sum_{i=1}^n \hat{\mathbf{s}}_i \hat{\mathbf{s}}_i^\top, \quad \text{donde} \quad \hat{\mathbf{s}}_i = \left. \frac{\partial}{\partial \boldsymbol{\theta}} \log [f(y_i|\boldsymbol{\theta})] \right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}, \quad (3.11)$$

donde

$$\hat{\mathbf{s}}_i = \left(\hat{\mathbf{s}}_{i,\beta}^\top, \hat{\mathbf{s}}_{i,p_1}, \dots, \hat{\mathbf{s}}_{i,p_{g-1}}, \hat{\mathbf{s}}_{i,\alpha_1}, \dots, \hat{\mathbf{s}}_{i,\alpha_g}, \hat{\mathbf{s}}_{i,\nu} \right).$$

Sin embargo, es importante notar que el error estándar de ν , asociado a $\hat{\mathbf{s}}_{i,\nu}$, no se puede obtener de manera directa por requerir la estimación de la esperanza condicional $E \left[z_{ij} \log h(u_i) | \hat{\boldsymbol{\theta}} \right]$, por tanto, las expresiones para los demás elementos quedan definidas como:

$$\begin{aligned} \hat{\mathbf{s}}_{i,\alpha_j} &= \frac{p_j D_{\alpha_j}(y_i; \boldsymbol{\theta}_j)}{f(y_i; \boldsymbol{\theta})}, \\ \hat{\mathbf{s}}_{i,\beta} &= \frac{\sum_{j=1}^G p_j D_{\beta}(y_i; \boldsymbol{\theta}_j)}{f(y_i; \boldsymbol{\theta})}, \\ \hat{\mathbf{s}}_{i,\mu_j}^\top &= \frac{\sum_{j=1}^G p_j D_{\mu_j}(y_i; \boldsymbol{\theta}_j)}{f(y_i; \boldsymbol{\theta})}, \\ \hat{\mathbf{s}}_{i,p_j} &= \frac{1}{f(y_i; \boldsymbol{\theta})} [\phi_{\text{SNI}}(y_i|\beta, \alpha_j, \sigma, \nu) - \phi_{\text{SNI}}(y_i|\beta, \alpha_g, \sigma, \nu)]. \end{aligned}$$

Y luego de realizar las derivadas para $U \sim \text{Gamma}(\nu/2, \nu/2)$ se obtiene:

$$\begin{aligned}
D_{\alpha_j}(y_i; \boldsymbol{\theta}_j) &= \frac{1}{2}k(\nu)\nu^{\nu/2} \left\{ -(\nu+1)(\nu + \xi_{2y}^2)^{-(\nu+3)/2} \xi_{1y} \xi_{2y} \frac{\partial \xi_{2y}}{\partial \alpha_j} + (\nu + \xi_{2y}^2)^{-(\nu+1)/2} \frac{\partial \xi_{1y}}{\partial \alpha_j} \right\}, \\
D_{\beta}(y_i; \boldsymbol{\theta}_j) &= \frac{1}{2}k(\nu)\nu^{\nu/2} \left\{ -(\nu+1)(\nu + \xi_{2y}^2)^{-(\nu+3)/2} \xi_{1y} \xi_{2y} \frac{\partial \xi_{2y}}{\partial \beta} + (\nu + \xi_{2y}^2)^{-(\nu+1)/2} \frac{\partial \xi_{1y}}{\partial \beta} \right\}, \\
D_{\mu_j}(y_i; \boldsymbol{\theta}_j) &= \frac{1}{2}k(\nu)\nu^{\nu/2} \left\{ -(\nu+1)(\nu + \xi_{2y}^2)^{-(\nu+3)/2} \xi_{1y} \xi_{2y} \frac{\partial \xi_{2y}}{\partial \mu_j} + (\nu + \xi_{2y}^2)^{-(\nu+1)/2} \frac{\partial \xi_{1y}}{\partial \mu_j} \right\},
\end{aligned}$$

donde $\frac{\partial \xi_{1y}}{\partial \alpha_j} = -\frac{1}{\alpha_j} \xi_{1y}$, $\frac{\partial \xi_{2y}}{\partial \alpha_j} = -\frac{1}{\alpha_j} \xi_{2y}$, $\frac{\partial \xi_{1y}}{\partial \beta} = -\frac{1}{2} \xi_{1y} \frac{f(x_i, \boldsymbol{\beta})}{\partial \beta}$, $\frac{\partial \xi_{2y}}{\partial \beta} = -\frac{1}{2} \xi_{1y} \frac{f(x_i, \boldsymbol{\beta})}{\partial \beta}$, $\frac{\partial \xi_{1y}}{\partial \mu_j} = -\frac{1}{\mu_j} \xi_{2y}$ y $\frac{\partial \xi_{2y}}{\partial \mu_j} = -\frac{1}{\mu_j} \xi_{1y}$, $k(\nu) = \Gamma((\nu+1)/2) / [\sqrt{\pi} \Gamma(\nu/2)]$, $\xi_{1y} = \frac{2}{\alpha_j} \cosh\left(\frac{Y_i - f(\mathbf{x}_i^T, \beta) - \mu_j}{2}\right)$ y $\xi_{2y} = \frac{2}{\alpha_j} \sinh\left(\frac{Y_i - f(\mathbf{x}_i^T, \beta) - \mu_j}{2}\right)$. Cabe indicar que la aproximación es un resultado asintótico y por tanto solo es eficaz cuando los tamaños de muestra son suficientemente grandes.



Capítulo 4

Estudio de simulación

En esta sección, realizamos un estudio de simulación para evaluar el rendimiento de las estimaciones de máxima verosimilitud del algoritmo EM propuesto. Para los experimentos numéricos, usamos el programa R. Específicamente, los objetivos son evaluar la precisión de las estimaciones basadas en el algoritmo EM para el modelo regresión no lineal con mixtura de distribuciones SN (RNL-MF-SN) y SSt (RNL-MF-SSt) en el error y evaluar la consistencia de los errores estándar. Otro objetivo es mostrar que las estimaciones del algoritmo EM propuesto brinda buenas propiedades asintóticas.

4.1. Estudio de simulación

Consideramos el modelo RNL-MF-SNI, conforme a lo presentado en el Capítulo 3. Para estimar los parámetros de máxima verosimilitud definimos una variable aleatoria Y_i conforme a la siguiente forma:

$$Y_i = f(\mathbf{x}_i, \boldsymbol{\beta}) + \epsilon_i, i = 1, \dots, n,$$

donde

$$f(\mathbf{x}_i, \boldsymbol{\beta}) = \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 \exp(\beta_4 x_{i4}),$$

es una función no lineal respecto a los parámetros $\boldsymbol{\beta}$ con $\mathbf{x}_i = (x_{i1}, x_{i2}, x_{i4})^\top$, $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3, \beta_4)^\top$. Definimos la distribución de los errores conforme a la siguiente distribución:

$$\epsilon_i \stackrel{iid}{\sim} \sum_{j=1}^g p_j \phi_{\text{SNI}}(\epsilon_i | \alpha_j, \mu_j, 2; H), i = 1, \dots, n \quad j = 1, 2,$$

de tal forma que la fdp de Y_i queda definida como:

$$f_{Y_i}(y_i | \boldsymbol{\theta}) = \sum_{j=1}^2 p_j \phi_{\text{SNI}}(y_i | \alpha_j, f(\mathbf{x}_i, \boldsymbol{\beta}) + \mu_j, 2; H),$$

con los parámetros a estimar $\boldsymbol{\theta} = (\alpha_1, \alpha_2, \beta_1, \beta_2, \beta_3, \beta_4, \mu_1, \mu_2, p_1, p_2, \nu)^\top$. A continuación definimos la función de densidad $H(u)$ como sigue:

$$U \sim \text{Gamma}(\nu/2, \nu/2).$$

Luego, por (3.3) obtenemos que la distribución de densidad de Senh-t-Student (SSt) queda

definida como:

$$f_{Y_i}(y_i) = \sum_{j=1}^g p_j \left\{ k(\nu) \nu^{\nu/2} / \sigma_j (\nu + \xi_{2y}^2)^{-((\nu+1)/2)} \xi_{1y} \right\}, \quad (4.1)$$

donde

$k(\nu) = \Gamma((\nu+1)/2) / [\sqrt{\pi} \Gamma(\nu/2)]$, $\xi_{1y} = \frac{2}{\alpha_j} \cosh\left(\frac{Y_i - f(\mathbf{x}_i, \beta) - \mu_j}{2}\right)$ y $\xi_{2y} = \frac{2}{\alpha_j} \sinh\left(\frac{Y_i - f(\mathbf{x}_i, \beta) - \mu_j}{2}\right)$. En este caso, como muestra Vilca et al. (2015), $U|(Y = y) \sim \text{Gamma}((\nu + 1)/2, (\nu + \xi_{2y}^2)/2)$ y $E[U|Y = y] = (\nu + 1) / (\nu + \xi_{2y}^2)$ que es precisamente el resultado que se utiliza para determinar $E[u_{ij}|\hat{\theta}, y_i]$ y a partir de esta la estimación del parámetro ν . Para la simulación se consideran los valores de parámetros conforme a los escenarios siguientes:

Escenario 1: $\alpha_1 = 0,25$, $\alpha_2 = 0,5$, $\beta_1 = 1,5$, $\beta_2 = 3$, $\beta_3 = 2$, $\beta_4 = 5$, $\mu_1 = -1$, $\mu_2 = 4$, $\nu = 3$, $p_1 = 0,8$;

Escenario 2: $\alpha_1 = 0,5$, $\alpha_2 = 0,1$, $\beta_1 = 4$, $\beta_2 = 5$, $\beta_3 = 3$, $\beta_4 = 1,5$, $\mu_1 = 4$, $\mu_2 = -1$, $\nu = 3$, $p_1 = 0,2$.

Para el Escenario 1 se presentan resultados de la simulación para las distribuciones SSt y SN mientras que para el Escenario 2 se consideró la distribución SSt. Los valores iniciales se obtuvieron mediante el procedimiento descrito en la sección 3.3.2.

4.1.1. Resultados de la simulación

Para la obtención de resultados se realizaron 1000 simulaciones considerando tamaños de muestra de $n = 50, 100, 200, 250, 500, 1000$. Se obtuvo la media de los parámetros estimados y la aproximación de las desviaciones estándar mediante los métodos de Monte Carlo, el método de aproximación de los errores estándar, IM SE, mostrado en la Sección 3.3.4 (Escenario 1) y Bootstrap (Escenario 2). Asimismo, se calculó el sesgo y la raíz del error cuadrático medio (RMSE, por sus siglas en inglés) asociados a la media de las estimaciones calculadas conforme a las siguientes fórmulas:

$$\text{Sesgo}(\theta_i) = \frac{1}{1000} \sum_{j=1}^{1000} (\hat{\theta}_i^{(j)} - \theta_i) \quad \text{y} \quad \text{RMSE}(\theta_i) = \sqrt{\frac{1}{1000} \sum_{j=1}^{1000} (\hat{\theta}_i^{(j)} - \theta_i)^2}.$$

Donde θ_i es el i -ésimo componente del vector de parámetros θ y $\hat{\theta}_i$ es su correspondiente estimador para la j -ésima simulación para cada uno de los tamaños de muestra, según corresponda.

Para el Escenario 1, los Cuadros 4.1 y 4.2 muestran los estudios de simulación para los modelos RNL-MF-SSt y RNL-MF-SN, respectivamente.

El Cuadro 4.1 muestra las estimaciones, los errores estándar de la matriz de información (IM SE) y la desviación estándar de Monte Carlo (MC SE). Asimismo, para examinar la consistencia del método de aproximación dado en la Sección 3.3.4 para los errores estándar de las estimaciones de máxima verosimilitud de θ se presenta la cobertura (COB) junto con el porcentaje de los intervalos de confianza (IC) del 95% resultantes, asumiendo normalidad asintótica. La cobertura se define como $\text{COB}(\hat{\theta}) = (1/m) \sum_{j=1}^m I(\theta \in [\hat{\theta}_{L_j}, \hat{\theta}_{U_j}])$, donde I es

una función indicadora que toma valor igual a 1 si θ cae en el intervalo $[\hat{\theta}_{L_j}, \hat{\theta}_{U_j}]$, $\hat{\theta}_{L_j}$ y $\hat{\theta}_{U_j}$ son los j -ésimos valores estimados de los límites inferiores y superiores del IC respectivamente.

Como se puede observar las estimaciones de máxima verosimilitud de los parámetros son cercanos a los valores teóricos y cada vez son más cercanos cuando el tamaño de muestra aumenta. Podemos observar también que los errores estándar considerados (mediante IM SE y MC SE) son bien cercanos, esto indica que la aproximación asintótica de la varianza de la estimación de máxima verosimilitud es confiable. En el caso particular de muestras pequeñas, para el parámetro α_1 a pesar de tener una buena estimación en general, el IM SE está por debajo del MC SE, sin embargo, el IM SE de dicho parámetro se acerca cada vez más al valor del MC SE conforme el tamaño de muestra aumenta. Adicionalmente la COB de los parámetros es en general estable. Esto puede ser corroborado observando que en general el intervalo de confianza se mantiene alrededor del 95 % de cobertura para cada parámetro con excepción del parámetro α_1 y también para α_2 en el caso de muestras pequeñas.

En el Cuadro 4.2 se muestra el estudio de simulación para el modelo RNL-MF-SN donde observamos que a medida que el tamaño de muestra aumenta las estimaciones de máxima verosimilitud de los parámetros son cercanos a los valores teóricos. Además los errores estándar obtenidos por el método de aproximación (IM SE) y el obtenido por las muestras Monte Carlo (MC SE) son próximos, esto indica que la aproximación asintótica de la varianza de la estimación de máxima verosimilitud es confiable. Notamos que el parámetro α_1 muestra un COB menor al 95 %.

Para el Escenario 2, los Cuadros 4.3 y 4.4 evidencian que los valores de estimación se aproximan a los valores reales de manera satisfactoria. También se puede observar que las estimaciones del error estándar por los métodos de Monte Carlo (MC SE) y Bootstrap (Bootstrap SE) producen resultados similares. En línea con estos resultados, las Figuras 4.1 y 4.2 muestran como la precisión de las estimaciones mejora sustancialmente con el tamaño de la población. Respecto a las propiedades asintóticas de las estimaciones obtenidas, se observa que tanto el sesgo como el RSME disminuye conforme el tamaño de población se incrementa, como se observa en las Figuras 4.3, 4.4, 4.5 y 4.6. Resalta que el sesgo oscila en valores muy cercanos a cero a pesar de que los gráficos puedan aparentar lo contrario. Por lo tanto se puede concluir que las estimaciones de máxima verosimilitud por el método EM para el modelo RNL-MF-SNI presenta propiedades asintóticas adecuadas.

Cuadro 4.1: Estudio de simulación 1 (RLN-MF-SSt): Media de estimación, errores estándar de la matriz de información (IM SE), la desviación estándar de Monte Carlo (MC SE) en base a 1000 simulaciones. Los valores reales están entre paréntesis.

Parámetro	Estimación	n=50			n=250			
		IM SE	MC SE	COB	Estimación	IM SE	MC SE	COB
$\alpha_1(0.25)$	0.2328	0.0395	0.0626	78.9	0.2490	0.0177	0.0233	86.0
$\alpha_2(0.5)$	0.4745	0.1864	0.1866	85.0	0.5002	0.0732	0.0728	94
$\beta_1(1.50)$	1.5047	0.1716	0.1776	93.6	1.5019	0.0717	0.0708	94.9
$\beta_2(3)$	2.9990	0.167	0.1868	91.7	2.9982	0.0706	0.0695	95.1
$\beta_3(2)$	2.0005	0.0191	0.0217	92.7	2.0005	0.0066	0.0068	94.3
$\beta_4(5)$	4.9998	0.0105	0.0117	92.7	4.9997	0.0034	0.0035	94.3
$\mu_1(-1)$	-1.0023	0.1693	0.1577	95.9	-1.0030	0.0686	0.0658	95.3
$\mu_2(4)$	3.9962	0.2803	0.3032	94.6	4.0006	0.1092	0.1070	95.8
$\nu(3)$	3.4925	-	1.8247	-	3.2560	-	1.0118	-
$p_1(0.8)$	0.7956	0.0639	0.0405	98.1	0.7993	0.0255	0.0249	95.5
Parámetro	Estimación	n=100			n=500			
		IM SE	MC SE	COB	Estimación	IM SE	MC SE	COB
$\alpha_1(0.25)$	0.2408	0.0274	0.0367	83.4	0.2491	0.0125	0.0160	86.7
$\alpha_2(0.5)$	0.4778	0.1167	0.1300	88.4	0.4948	0.0509	0.0570	93.5
$\beta_1(1.50)$	1.4924	0.1104	0.1147	93.1	1.4968	0.0527	0.0539	94.4
$\beta_2(3)$	2.9950	0.1107	0.1162	93.8	2.9975	0.0507	0.0489	95.0
$\beta_3(2)$	1.9997	0.0104	0.0110	92.0	2.0002	0.0047	0.0047	95.3
$\beta_4(5)$	5.0002	0.0054	0.0058	92.1	4.9999	0.0024	0.0024	93.4
$\mu_1(-1)$	-0.9907	0.0941	0.0967	93.2	-0.9983	0.0451	0.0421	96.4
$\mu_2(4)$	4.0065	0.163	0.1709	93.3	4.0076	0.0745	0.0747	95.4
$\nu(3)$	3.4708	-	1.5718	-	3.0828	-	0.5321	-
$p_1(0.8)$	0.8009	0.0409	0.0390	95.2	0.8008	0.018	0.0184	94.5
Parámetro	Estimación	n=200			n=1000			
		IM SE	MC SE	COB	Estimación	IM SE	MC SE	COB
$\alpha_1(0.25)$	0.2482	0.0197	0.0260	84.9	0.2492	0.0088	0.0114	88
$\alpha_2(0.5)$	0.4948	0.081	0.0889	90.4	0.5000	0.0363	0.0385	93.0
$\beta_1(1.50)$	1.4998	0.082	0.0811	94.6	1.4996	0.0363	0.0372	95.6
$\beta_2(3)$	2.9978	0.0825	0.0826	95.5	3.0000	0.0362	0.0395	95.2
$\beta_3(2)$	1.9998	0.0076	0.0076	94.3	2.0001	0.0033	0.0032	95.2
$\beta_4(5)$	5.0001	0.0038	0.0039	94.2	5.0000	0.0017	0.0017	95.3
$\mu_1(-1)$	-0.9981	0.0822	0.0798	95.6	-1.0005	0.0326	0.0300	96.0
$\mu_2(4)$	4.0062	0.125	0.1234	94.9	4.0020	0.0531	0.0514	95.9
$\nu(3)$	3.3754	-	1.8902	-	3.0508	-	0.3521	-
$p_1(0.8)$	0.7991	0.0285	0.0251	97.7	0.8003	0.0127	0.0128	95.0

Cuadro 4.2: Estudio de simulación 1 (RLN-MF-SN) : Media de estimación, errores estándar de la matriz de información (IM SE), la desviación estándar de Monte Carlo (MC SE) en base a 1000 simulaciones. Los valores reales están entre paréntesis.

Parámetro	n=50				n=250			
	Estimación	IM SE	MC SE	COB	Estimación	IM SE	MC SE	COB
$\alpha_1(0.25)$	0.2557	0.0402	0.2136	80.0	0.2342	0.0226	0.0119	97.6
$\alpha_2(0.5)$	0.4661	0.2616	0.2678	84.4	0.4669	0.0487	0.0481	83.2
$\beta_1(1.50)$	1.5807	0.1779	0.5325	92.0	1.5015	0.0554	0.0586	93.0
$\beta_2(3)$	3.0427	0.1585	0.3804	93.6	2.9991	0.0557	0.0573	94.2
$\beta_3(2)$	2.0050	0.0174	0.0422	91.6	2.0003	0.0053	0.0055	94.8
$\beta_4(5)$	4.9976	0.0092	0.0204	91.0	4.9999	0.0028	0.0029	94.8
$\mu_1(-1)$	-1.0248	0.1684	0.2419	92.8	-1.0012	0.0483	0.0508	93.2
$\mu_2(4)$	3.7656	0.3216	1.3992	90.2	3.9975	0.0795	0.0865	93.2
$p_1(0.8)$	0.7983	0.0696	0.0589	97.8	0.8032	0.0256	0.0236	96.6
Parámetro	n=100				n=500			
	Estimación	IM SE	MC SE	COB	Estimación	IM SE	MC SE	COB
$\alpha_1(0.25)$	0.2314	0.0192	0.0573	82.2	0.2364	0.0157	0.0087	94.8
$\alpha_2(0.5)$	0.4573	0.0929	0.1064	88.2	0.4721	0.0327	0.0349	80.4
$\beta_1(1.55)$	1.4928	0.0920	0.1014	91.0	1.5019	0.0382	0.0417	93.0
$\beta_2(3)$	3.0141	0.0895	0.1741	93.2	3.0010	0.0376	0.0383	95.0
$\beta_3(2)$	2.0007	0.0089	0.0118	92.8	1.9999	0.0034	0.0037	92.0
$\beta_4(5)$	4.9997	0.0046	0.0061	93.4	5.0000	0.0018	0.0019	92.2
$\mu_1(-1)$	-1.0025	0.0777	0.0835	92.4	-1.0015	0.0338	0.0366	92.8
$\mu_2(4)$	3.9844	0.1373	0.3690	92.8	3.9969	0.0553	0.0609	93.6
$p_1(0.8)$	0.8007	0.0413	0.0361	96.6	0.8028	0.0181	0.0164	95.4
Parámetro	n=200				n=1000			
	Estimación	IM SE	MC SE	COB	Estimación	IM SE	MC SE	COB
$\alpha_1(0.25)$	0.2334	0.0127	0.0143	85.0	0.2376	0.0055	0.0061	88.6
$\alpha_2(0.5)$	0.4625	0.0555	0.0562	85.6	0.4748	0.0227	0.0240	90.0
$\beta_1(1.55)$	1.4945	0.0657	0.0755	88.8	1.5017	0.0275	0.0367	92.6
$\beta_2(3)$	2.9977	0.0593	0.0633	92.0	2.9999	0.0267	0.0343	90.8
$\beta_3(2)$	1.9997	0.0060	0.0063	93.8	1.9999	0.0025	0.0028	92.6
$\beta_4(5)$	5.0002	0.0032	0.0033	93.8	5.0000	0.0013	0.0014	92.2
$\mu_1(-1)$	-0.9948	0.0586	0.0630	93.2	-1.0009	0.0248	0.0270	91.6
$\mu_2(4)$	4.0111	0.0921	0.0997	92.8	3.9981	0.0393	0.0439	91.4
$p_1(0.8)$	0.8045	0.0288	0.0225	98.8	0.8014	0.0127	0.0116	97.0

Cuadro 4.3: Estudio de simulación 2: Media de estimación, errores estándar, sesgo, RMSE en base a 1000 simulaciones. Los valores reales están entre paréntesis.

n=50					
Parámetro	Estimación	MC SE	Bootstrap SE	Sesgo	RMSE
α_1 (0.5)	0.4983	0.4393	0.6614	-0.0017	0.4391
α_2 (1)	1.0331	0.2798	0.2892	0.0331	0.2816
β_1 (4)	4.0008	0.6111	0.9722	0.0008	0.6107
β_2 (5)	4.9926	0.6783	0.7749	-0.0074	0.6780
β_3 (3)	2.9314	0.8858	0.9156	-0.0686	0.8880
β_4 (1.5)	1.5705	0.3044	0.3420	0.0705	0.3123
μ_1 (4)	4.1252	1.0254	1.3639	0.1252	1.0325
μ_2 (-1)	-0.8662	1.0369	1.1727	0.1338	1.0450
ν (3)	9.9789	8.2500	8.4447	6.9789	10.8028
p_1 (0.2)	0.2089	0.0658	0.0885	0.0089	0.0663
n=100					
Parámetro	Estimación	MC SE	Bootstrap SE	Sesgo	RMSE
α_1 (0.5)	0.4722	0.1792	0.1549	-0.0278	0.1812
α_2 (1)	1.0027	0.1631	0.2734	0.0027	0.1630
β_1 (4)	3.9879	0.3881	0.4109	-0.0121	0.3881
β_2 (5)	4.9902	0.4313	0.2652	-0.0098	0.4312
β_3 (3)	3.0021	0.5945	0.7702	0.0021	0.5942
β_4 (1.5)	1.5212	0.1838	0.2209	0.0212	0.1849
μ_1 (4)	4.0187	0.6671	0.7769	0.0187	0.6671
μ_2 (-1)	-0.9752	0.6580	0.7118	0.0248	0.6581
ν (3)	5.2897	5.2215	4.9563	2.2897	5.6991
p_1 (0.2)	0.1989	0.0430	0.0476	-0.0011	0.0429
n=200					
Parámetro	Estimación	MC SE	Bootstrap SE	Sesgo	RMSE
α_1 (0.5)	0.4822	0.1064	0.1299	-0.0178	0.1078
α_2 (1)	0.9977	0.1032	0.0993	-0.0023	0.1032
β_1 (4)	3.9903	0.2553	0.3052	-0.0097	0.2553
β_2 (5)	5.0021	0.2570	0.2887	0.0021	0.2569
β_3 (3)	2.9831	0.2932	0.3984	-0.0169	0.2936
β_4 (1.5)	1.5090	0.0876	0.0844	0.0090	0.0880
μ_1 (4)	4.0288	0.3252	0.4766	0.0288	0.3263
μ_2 (-1)	-0.9742	0.3229	0.4815	0.0258	0.3237
ν (3)	3.5837	2.2410	3.2722	0.5837	2.3147
p_1 (0.2)	0.1976	0.0290	0.0307	-0.0024	0.0291

Cuadro 4.4: Estudio de simulación 2: Media de estimación, errores estándar, sesgo, RMSE en base a 1000 simulaciones. Los valores reales están entre paréntesis.

n=250					
Parámetro	Estimación	MC SE	Bootstrap SE	Sesgo	RMSE
α_1 (0.5)	0.4876	0.0949	0.0797	-0.0124	0.0957
α_2 (1)	0.9984	0.0953	0.1060	-0.0016	0.0952
β_1 (4)	4.0064	0.2283	0.3072	0.0064	0.2283
β_2 (5)	4.9966	0.2269	0.2978	-0.0034	0.2268
β_3 (3)	2.9963	0.2616	0.2924	-0.0037	0.2615
β_4 (1.5)	1.5041	0.0767	0.0845	0.0041	0.0767
μ_1 (4)	4.0054	0.2743	0.3271	0.0054	0.2742
μ_2 (-1)	-0.9901	0.2769	0.3373	0.0099	0.2769
ν (3)	3.3912	1.5032	0.5153	0.3912	1.5525
p_1 (0.2)	0.1989	0.0272	0.0260	-0.0011	0.0272
n=500					
Parámetro	Estimación	MC SE	Bootstrap SE	Sesgo	RMSE
α_1 (0.5)	0.4950	0.0587	0.0537	-0.0050	0.0589
α_2 (1)	1.0009	0.0631	0.0906	0.0009	0.0631
β_1 (4)	4.0066	0.1555	0.1558	0.0066	0.1555
β_2 (5)	5.0108	0.1547	0.1711	0.0108	0.1550
β_3 (3)	2.9918	0.1321	0.1711	-0.0082	0.1322
β_4 (1.5)	1.5031	0.0407	0.0454	0.0031	0.0408
μ_1 (4)	3.9967	0.1137	0.1632	-0.0033	0.1137
μ_2 (-1)	-0.9963	0.1091	0.1676	0.0037	0.1091
ν (3)	3.1717	0.6618	1.4677	0.1717	0.6834
p_1 (0.2)	0.2002	0.0190	0.0185	0.0002	0.0190
n=1000					
Parámetro	Estimación	MC SE	Bootstrap SE	Sesgo	RMSE
α_1 (0.5)	0.4986	0.0399	0.0367	-0.0014	0.0399
α_2 (1)	1.0018	0.0424	0.0500	0.0018	0.0424
β_1 (4)	4.0001	0.1129	0.1022	0.0001	0.1129
β_2 (5)	4.9935	0.1065	0.1212	-0.0065	0.1067
β_3 (3)	3.0019	0.0801	0.1037	0.0019	0.0800
β_4 (1.5)	1.4999	0.0255	0.0299	-0.0001	0.0255
μ_1 (4)	4.0008	0.0491	0.0947	0.0008	0.0491
μ_2 (-1)	-1.0005	0.0427	0.0936	-0.0005	0.0427
ν (3)	3.0725	0.4059	0.4313	0.0725	0.4121
p_1 (0.2)	0.1997	0.0131	0.0135	-0.0003	0.0131

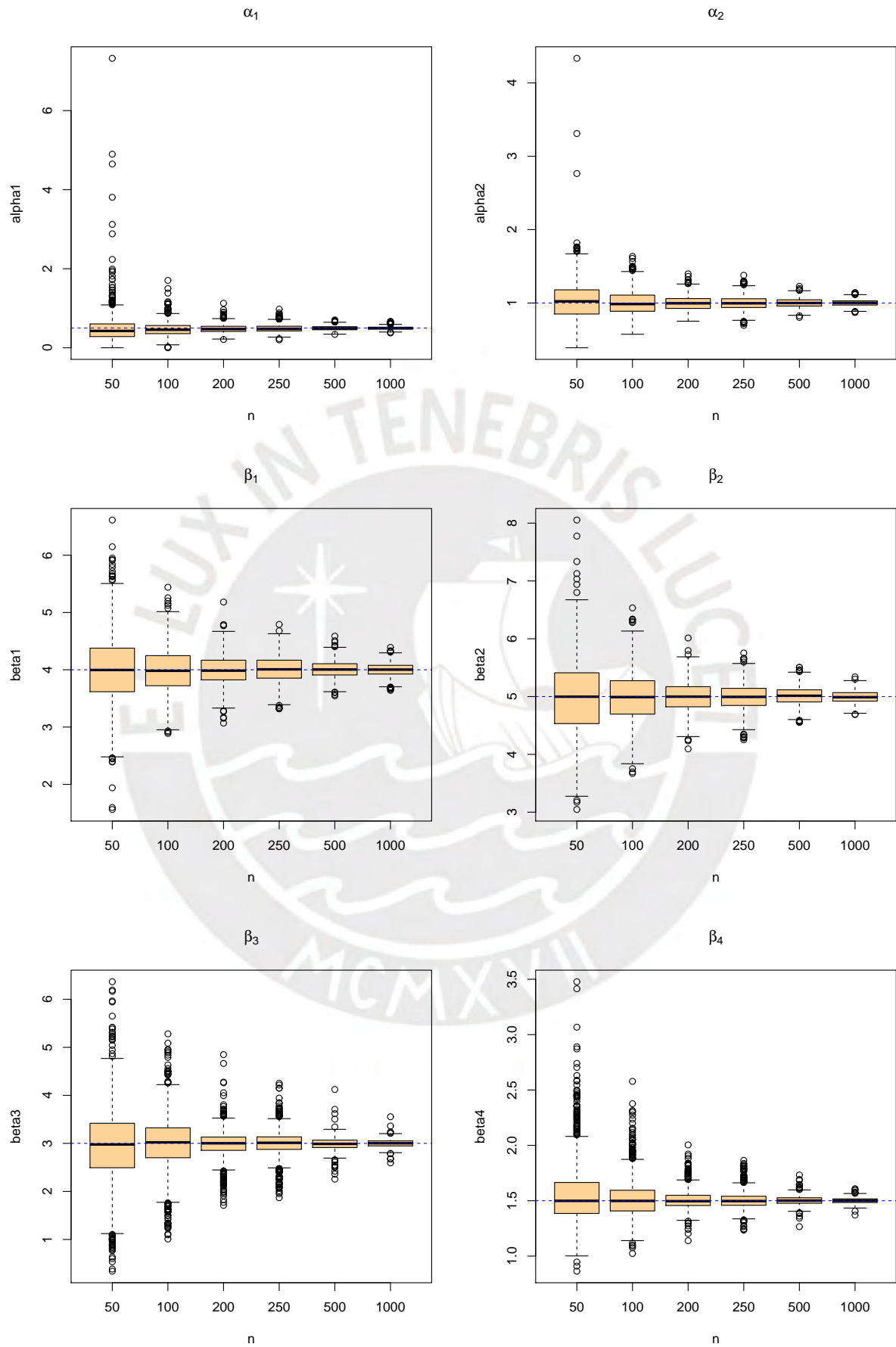


Figura 4.1: Gráfico de cajas de las estimaciones de los parámetros α y β respectivamente.

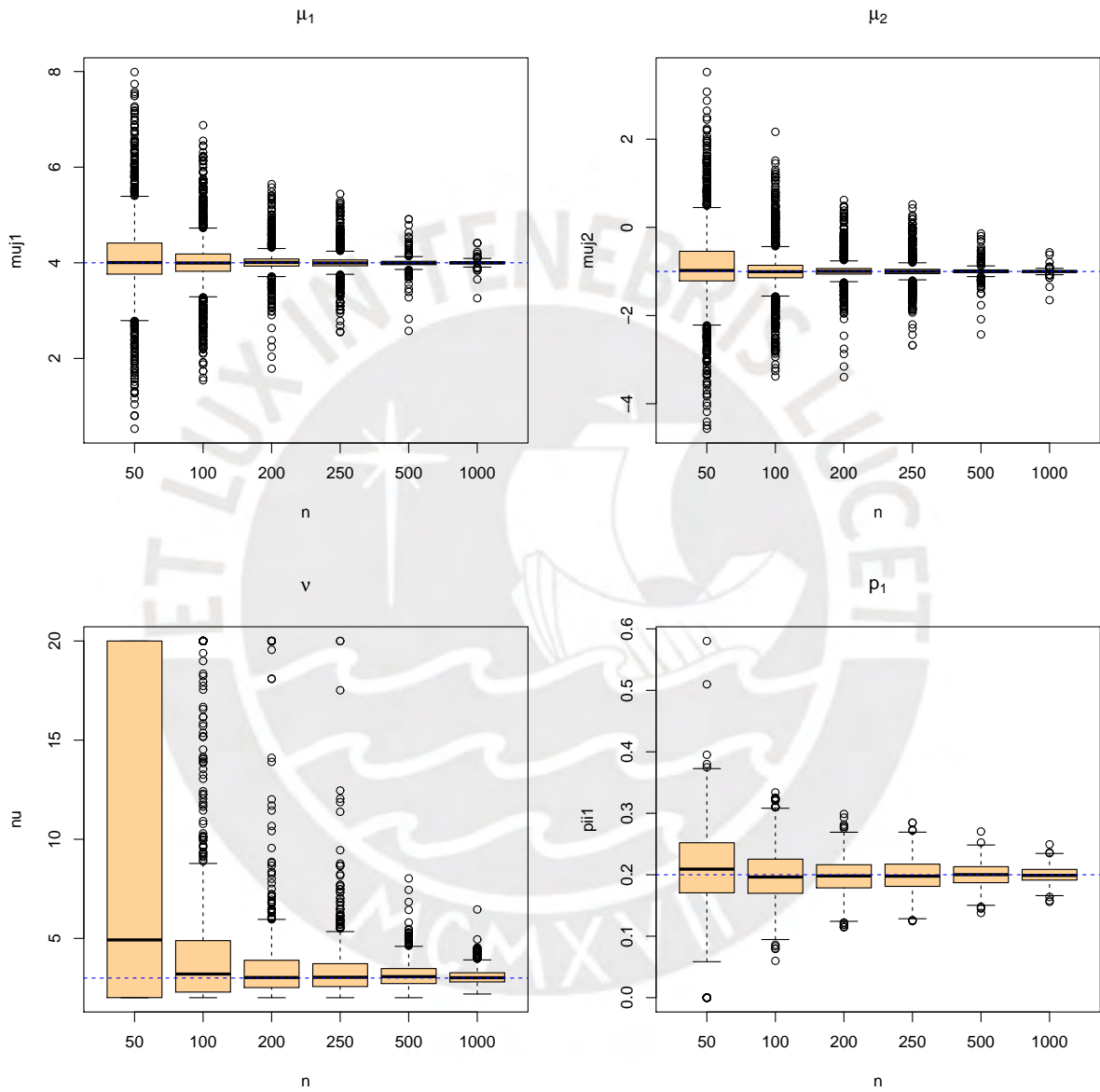


Figura 4.2: Gráfico de cajas de las estimaciones de los parámetros μ , ν y p_1 respectivamente.

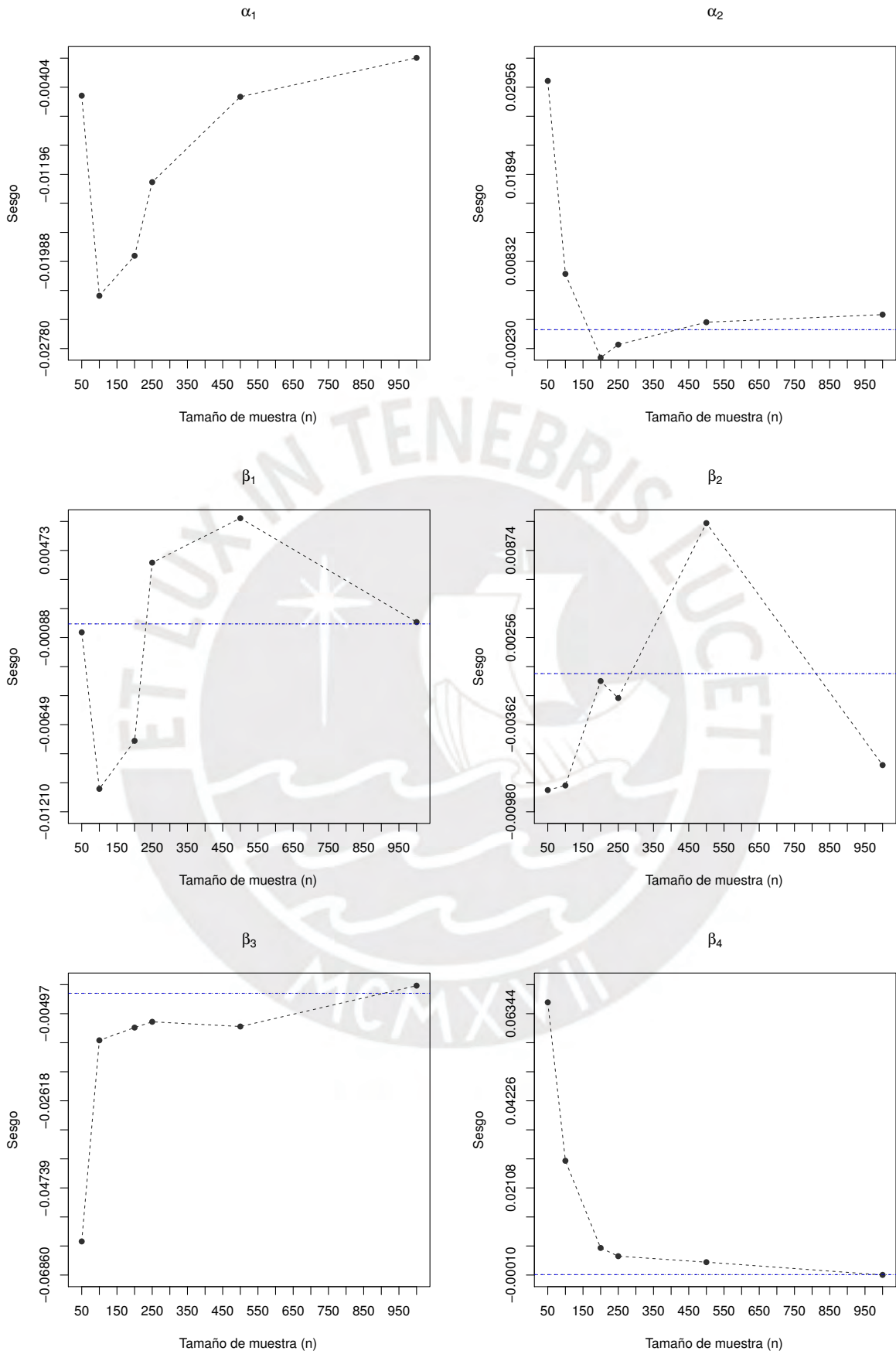


Figura 4.3: Sesgo de las estimaciones de los parámetros α y β respectivamente.

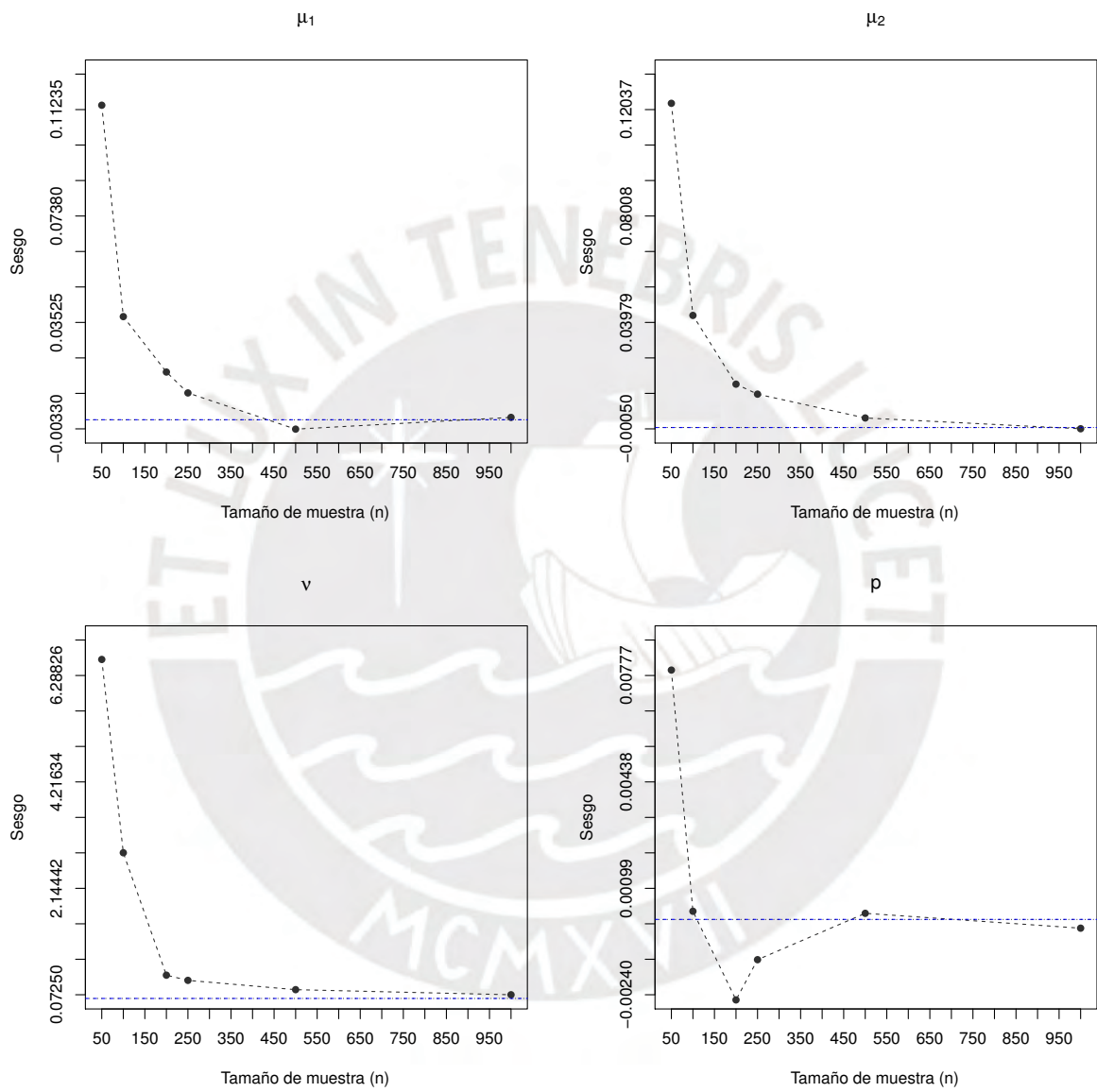


Figura 4.4: Sesgo de las estimaciones de los parámetros μ , ν y p_1 respectivamente.

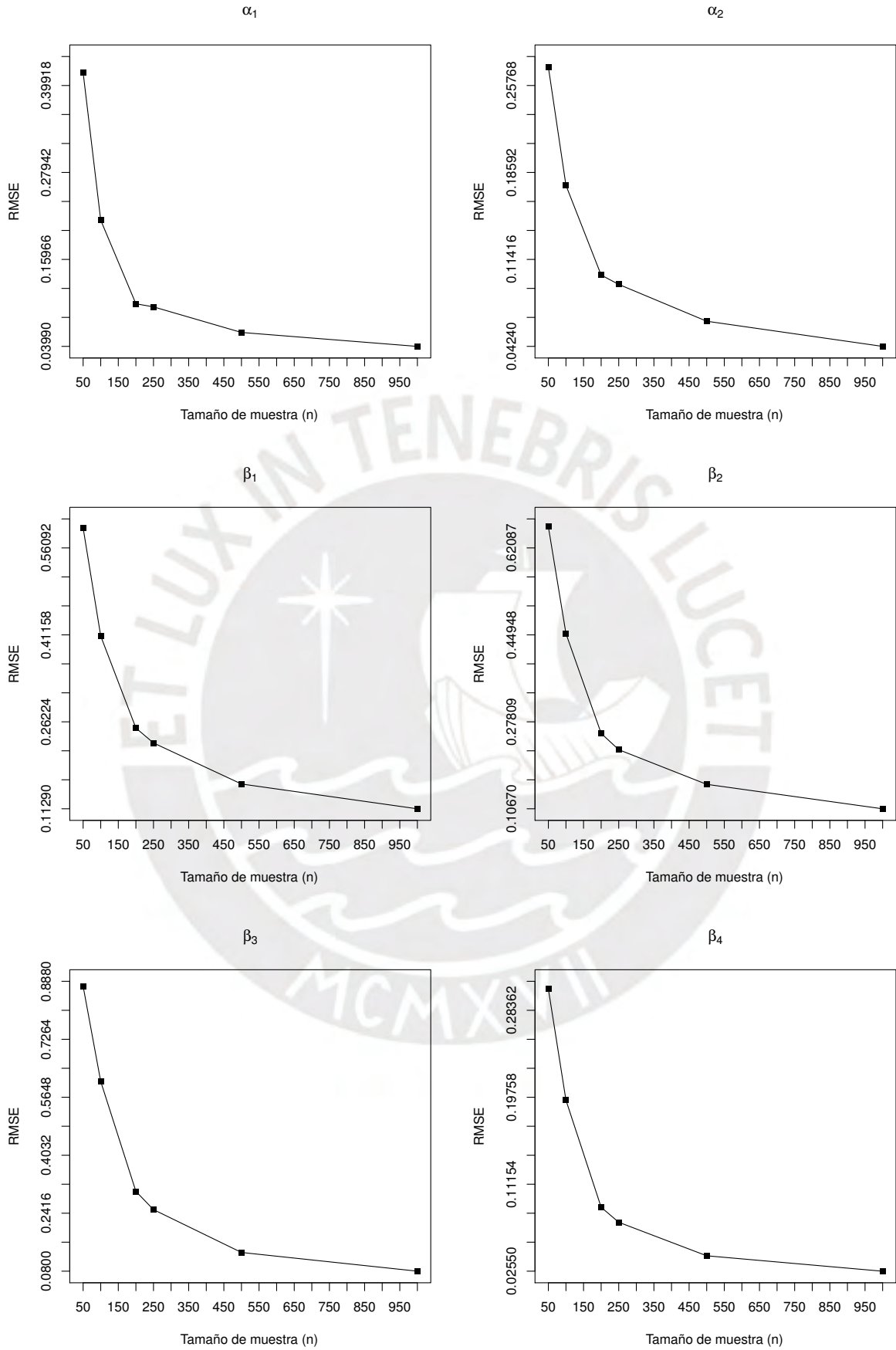


Figura 4.5: RMSE de las estimaciones de los parámetros α y β respectivamente.

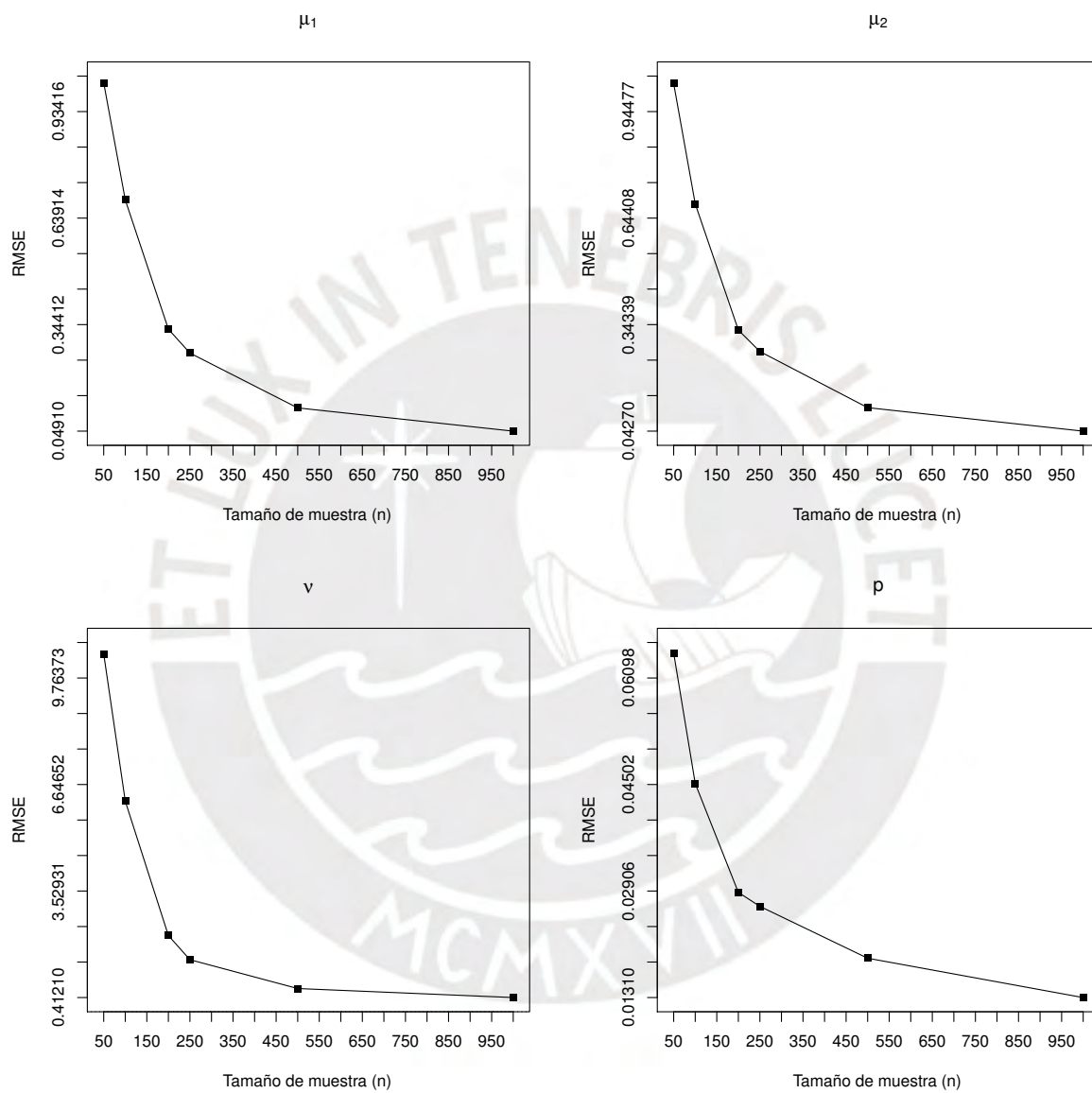


Figura 4.6: RMSE de las estimaciones de los parámetros μ , ν y p_1 respectivamente.

Capítulo 5

Aplicación

Este conjunto de datos fue analizado por Rieck y Nedelman (1991) en un modelo de regresión lineal. También fue considerado por Farias y Lemonte (2011) en un modelo de regresión no lineal, así como por Santana et al. (2011) pero considerando un modelo de regresión lineal usando una distribución normal asimétrica. Estos datos corresponden a 46 piezas metálicas que fueron probadas hasta fallar. Las variables consisten en el número de ciclos hasta la falla (N) y el trabajo por ciclo (en MJ/m^3). Consideremos el modelo de regresión no lineal:

$$Y_i = \beta_1 x_i^{\beta_2} + \varepsilon_i, \quad i = 1, \dots, 46,$$

donde $Y_i = \log(N_i)$ es el logaritmo del número de ciclos, x_i es el i -ésimo valor del trabajo por ciclo, β_1 y β_2 los coeficientes de regresión, y $\varepsilon_i \sim \sum_{j=1}^g p_j \phi_{\text{SNI}}(\varepsilon | \alpha_j, \mu_j, 2; H), i = 1, \dots, n$ $j = 1, \dots, 4$. El Cuadro 5.1 muestra diversos criterios de información tales como el AIC, EDC, AIC_4 , AIC_5 , AIC_6 , BIC, CAIC, HQIC, AIC_c , aBIC, KIC_c y se resalta los modelos elegidos según el número de componentes. Para el caso SN la mayoría (8 de 11) de los criterios de información se inclinan a que el modelo de $g = 3$ componentes es el mejor, no obstante para el caso SSt la mayoría (9 de 11) de los criterios de información muestran que el mejor modelo con $g = 1$ componente es el que se ajustaría mejor a los datos. En el Cuadro 5.1 los

Cuadro 5.1: Criterios de información para $g = 1$, $g = 2$, $g = 3$ y $g = 4$ componentes para los casos MRNL-MF-SN y MRNL-MF-SSt

	Senh-Normal				Senh- t -Student			
	$g = 1$	$g = 2$	$g = 3$	$g = 4$	$g = 1$	$g = 2$	$g = 3$	$g = 4$
AIC	53.5706	51.3247	36.9506	53.3606	54.5789	52.6567	54.9196	58.1514
EDC	50.9964	46.8200	30.5152	44.9947	52.0048	47.5085	47.8408	49.1419
AIC_4	61.5706	65.3247	56.9506	79.3606	62.5789	68.6567	76.9196	86.1514
AIC_5	65.5706	72.3247	66.9506	92.3606	66.5789	76.6567	87.9196	100.1514
AIC_6	69.5706	79.3247	76.9506	105.3606	70.5789	84.6567	98.9196	114.1514
BIC	60.8852	64.1252	55.2370	77.1330	61.8935	67.2859	75.0347	83.7524
CAIC	64.8852	71.1252	65.2370	90.1330	65.8935	75.2859	86.0347	97.7524
HQIC	56.3107	56.1198	43.8008	62.2659	57.3190	58.1369	62.4549	67.7417
AIC_c	54.5462	54.2721	43.2363	64.7356	55.5545	56.5486	62.6843	71.6998
aBIC	48.3432	42.1767	23.8820	36.3715	49.3515	42.2019	40.5443	39.8555
KIC_c	58.7183	61.8511	54.4948	79.9935	59.7266	65.3218	75.2378	88.3719

tres últimos criterios de información son versiones para muestras pequeñas y si analizamos el AIC_c , aBIC y KIC_c el mejor modelo es el caso SN para $g = 3$ componentes. El Cuadro 5.2 muestra los estimadores de máxima verosimilitud de los parámetros para los modelos

RNL-MF-SN y RNL-MF-SSt para $g = 3$ componentes, los errores estándar obtenidos vía al procedimiento basada en la Sección 3.3.4 que usamos para obtener el límite inferior (LI) y superior (LS) del intervalo de confianza (CI) al 95 %, dicho cuadro también muestra el enfoque bootstrap (entre paréntesis), desarrollado por Efron y Tibshirani (1986) que provee otra forma para obtener los intervalos de confianza.

Cuadro 5.2: Estimación (Est.) de máxima verosimilitud para dos modelos seleccionados con $g = 3$ componentes. El error estándar (EE) han sido estimados basado en el método de aproximación de la matriz de información observada. LI (límite inferior) y LS (límite superior) del intervalo de confianza al 95 % mientras que entre paréntesis se encuentra el error estándar bootstrap, límite inferior bootstrap y límite inferior bootstrap del intervalo de confianza al 95 %.

Parámetro	RNL-MF-SN				RNL-MF-SSt			
	Est.	EE	LI	LS	Est.	EE	LI	LS
α_1	0.3418	0.0433	0.2569	0.4267	0.2542	0.0708	0.1155	0.3929
		(0.0829)	(0.1793)	(0.5043)		(0.0493)	(0.1025)	(0.4063)
α_2	0.0089	0.0094	-0.0094	0.0272	0.0301	0.0235	-0.0155	0.0757
		(0.0395)	(-0.0167)	(0.1381)		(0.0775)	(-0.0665)	(0.1267)
α_3	0.0607	0.0167	0.0279	0.0935	0.0794	0.0255	0.0295	0.1293
		(0.0330)	(-0.0558)	(0.0736)		(0.0530)	(-0.0245)	(0.1833)
β_1	15.5978	0.3940	14.826	16.370	15.5437	0.5734	14.420	16.6687
		(0.3343)	(14.943)	(16.253)		(0.5092)	(14.546)	(16.542)
β_2	-0.2524	0.0876	-0.4242	-0.0806	-0.2527	0.1043	-0.4571	-0.0483
		(0.2597)	(-0.7614)	(0.2566)		(0.0097)	(-0.2717)	(-0.2337)
μ_1	-0.2572	2.0939	-4.3612	3.8468	-0.3402	2.5221	-5.2835	4.6031
		(3.1778)	(-6.4860)	(5.9708)		(0.0316)	(-0.5357)	(-0.1433)
μ_2	0.1102	2.1203	-4.0454	4.2658	0.1185	2.5310	-4.8421	5.0791
		(3.1756)	(-5.8808)	(6.5674)		(0.1001)	(0.0570)	(0.1808)
μ_3	0.3437	2.1149	-3.8014	4.4888	0.3761	2.5339	-4.5903	5.3425
		(4.4654)	(-8.6422)	(8.8618)		(0.0512)	(0.2761)	(0.4769)
p_1	0.6831	0.0906	0.5056	0.8606	0.5518	0.1017	0.3526	0.7510
		(0.1131)	(0.4613)	(0.9047)		(0.1011)	(0.3156)	(0.7884)
p_2	0.0744	0.0554	-0.0342	0.1830	0.1379	0.0719	-0.0031	0.2789
		(0.0914)	(0.0635)	(0.4217)		(0.1206)	(-0.0603)	(0.3361)
ν	-	-	-	-	9.08	-	-	-

Los valores iniciales obtenidos fueron los siguientes $\alpha_1^{(0)} = 0,18508$, $\alpha_2^{(0)} = 0,10268$, $\alpha_3^{(0)} = 0,13312$, $\beta_1^{(0)} = 15,93774$, $\beta_2^{(0)} = -0,26201$, $\mu_1^{(0)} = -0,465015$, $\mu_2^{(0)} = 0,435712$, $\mu_3^{(0)} = 0,060182$, $p_1^{(0)} = 0,34783$, $p_2^{(0)} = 0,32609$ y $p_3^{(0)} = 0,326$. Según el Cuadro 5.2 Notamos que en general los errores estándar para el modelo RNL-MF-SN son menores al modelo RNL-MF-SSt, además también observamos que para el caso RNL-MF-SN hay parámetros que han resultado ser no significativos tales como α_2 , μ_1 , μ_2 , μ_3 y p_2 . Algo similar sucede con el modelo RNL-MF-SSt si observamos los errores estándar obtenido por el método de aproximación. Observando los intervalos de confianza obtenido por bootstrap para el modelo RNL-MF-SSt notamos que μ_1 , μ_2 y μ_3 son significativos a diferencia del RNL-MF-SN en la que todos los parámetros μ_1 , μ_2 y μ_3 son no significativos.

La Figura 5.1 muestra la log-verosimilitud perfilada para el caso de $g = 2$, $g = 3$ y $g = 4$ componentes, respectivamente, para obtener el valor inicial de los grados de libertad para el algoritmo EM. La Figura 5.2 muestra los diagramas de dispersión del conjunto de datos y el ajuste del modelo no lineal basado en diferentes componentes para el caso SN y SSt. En ambos casos se observa las curvas superpuestas son similares.

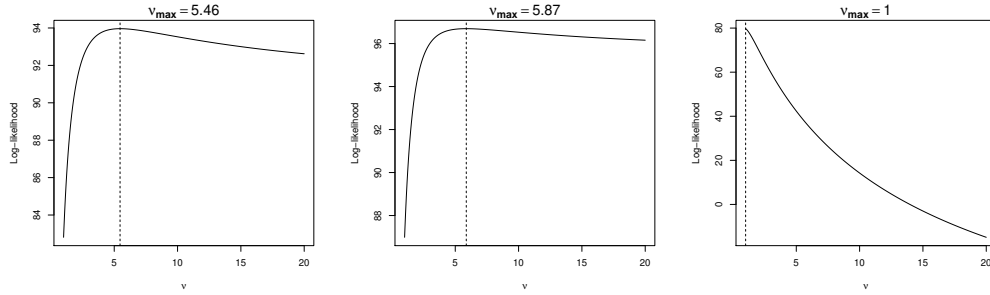


Figura 5.1: Log-verosimilitud perfilada para el caso t -student para $g = 2$, $g = 3$ y $g = 4$ componentes respectivamente (de izquierda a derecha)

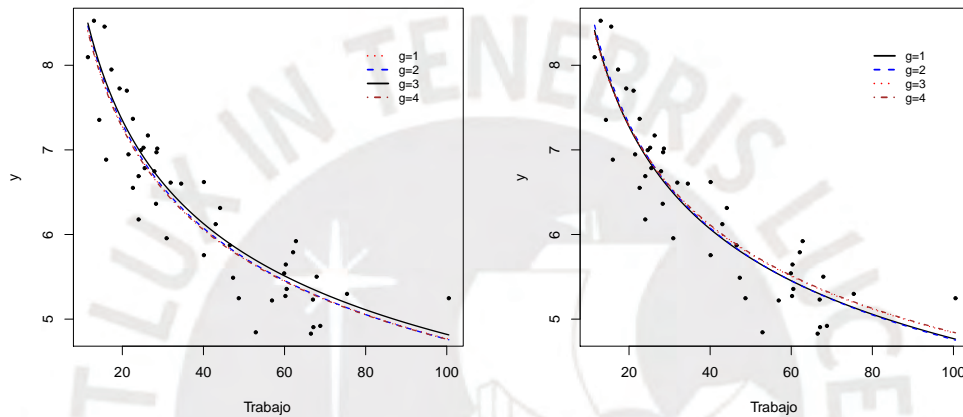


Figura 5.2: Diagrama de dispersión y las curvas ajustadas para el modelo RNL-MF-SN (panel izquierdo) y el modelo RNL-MF-SSt (panel derecho) para diferentes números de componentes $g = 1, 2, 3, 4$

Usando los mismos datos, ahora consideraremos el modelo no lineal usado por Lemonte y Cordeiro (2009) y Maehara Rocío (2021):

$$Y_i = \beta_1 + \beta_2 \exp(-\beta_3/x_i) + \varepsilon_i \quad i = 1, \dots, 46,$$

donde Y_i y x_i fueron detallados anteriormente, el error $\varepsilon_i \sim \sum_{j=1}^4 p_j \phi_{\text{SNI}}(\epsilon|\alpha_j, \mu_j, 2; H)$, $i = 1, \dots, n$, $j = 1, \dots, 4$ específicamente los casos particulares de modelos RNL-MF-SN y RNL-MF-SSt. El Cuadro 5.3 muestra los criterios de información para tamaños de muestra pequeños, notamos que si observamos AIC_c , aBIC el mejor elegido es aquel que considera $g = 3$ componentes para el caso Senh-Normal mientras que el KIC_c elige al modelo Senh-Normal que considera $g = 1$ componente, entonces elegimos el caso Senh-Normal con $g = 3$ componentes.

Las estimaciones de máxima verosimilitud, el error estándar bootstrap así como el intervalo de confianza bootstrap son mostrados en el Cuadro 5.4. Notamos que el error estándar del modelo RNL-MF-SN son menores que los del modelo RNL-MF-SSt en todos los parámetros. De otro lado en el modelo RNL-MF-SN todos los parámetros son significativos mientras que en el modelo RNL-MF-SSt el parámetro α_2 es no significativo. En la Figura 5.3 se muestra el diagrama de dispersión para los casos Senh-Normal (panel izquierdo) y Senh- t -Student (pa-

nel derecho) en la que notamos que las curvas en general para ambos modelos considerados tienen un comportamiento similar.

Cuadro 5.3: Criterios de información para $g = 1, g = 2, g = 3$ y $g = 4$ componentes para los casos SN y SSt

	RNL-MF-SN				RNL-MF-SSt			
	$g = 1$	$g = 2$	$g = 3$	$g = 4$	$g = 1$	$g = 2$	$g = 3$	$g = 4$
AIC _c	56.0558	58.1596	52.7909	76.9037	56.9405	58.2958	66.1445	78.5552
aBIC	48.0215	43.8128	30.6509	45.0594	48.9062	41.5341	41.0078	42.9524
KIC _c	61.3357	66.9327	65.3443	93.5758	62.2204	68.2949	80.0303	96.6863

Cuadro 5.4: Estimación de máxima verosimilitud para dos modelos seleccionados con $g = 3$ componentes. El error estándar (EE) obtenido por bootstrap. LI_B (límite inferior bootstrap) y LS_B (límite superior bootstrap) del intervalo de confianza al 95 %.

Parámetro	RNL-MF-SN				RNL-MF-SSt			
	Estimación	EE	LI _B	LS _B	Estimación	EE	LI _B	LS _B
α_1	0.1911	0.0329	0.1266	0.2556	0.1970	0.0412	0.1162	0.2778
α_2	0.0175	0.0071	0.0036	0.0314	0.0178	0.0263	-0.0337	0.0693
α_3	0.1881	0.0607	0.0691	0.3071	0.1956	0.0722	0.0541	0.3371
β_1	9.1906	0.4803	8.2492	10.1320	9.1875	0.5989	8.0137	10.3613
β_2	-5.4511	0.3833	-6.2024	-4.6998	-5.4527	0.4928	-6.4186	-4.4868
β_3	-21.2356	2.7863	-26.697	-15.775	-21.2952	3.3763	-27.913	-14.678
μ_1	0.3013	0.0479	0.2074	0.3952	0.2988	0.0502	0.2004	0.3972
μ_2	-0.2343	0.0025	-0.2392	-0.2294	-0.2356	0.0039	-0.2432	-0.2280
μ_3	-0.5338	0.1115	-0.7523	-0.3153	-0.5332	0.1219	-0.7721	-0.2943
p_1	0.6284	0.0875	0.4569	0.7999	0.6292	0.0975	0.4381	0.8203
p_2	0.1604	0.0511	0.0602	0.2606	0.1596	0.0629	0.0363	0.2829

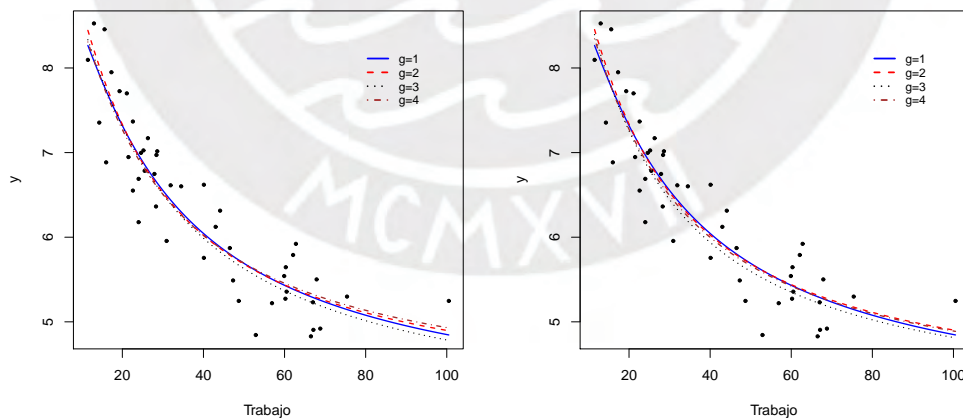


Figura 5.3: Diagrama de dispersión y las curvas ajustadas para el modelo RNL-MF-SN (panel izquierdo) y el modelo RNL-MF-SSt (panel derecho) para diferentes números de componentes $g = 1, 2, 3, 4$

Capítulo 6

Conclusiones

6.1. Conclusiones

En esta tesis exploramos la posibilidad de implementar un modelo de regresión no lineal con mixtura de distribuciones SNI en el error. Para dicho fin se presentó primero la distribución Birnbaum-Saunders para variables aleatorias cuyo logaritmo puede relacionarse con una distribución Senh-Normal. Posteriormente observamos las características de una regresión no lineal así como la metodología para obtener estimaciones para distribuciones basadas en mixturas finitas. En este sentido, para poder obtener las estimaciones de máxima verosimilitud de los parámetros de nuestro modelo se implementó una aplicación del algoritmo EM.

Para poder comprobar las cualidades del modelo planteado se realizó un ejercicio de simulación cuyos resultados permitieron concluir que las estimaciones en general son precisas a medida que el tamaño de muestra aumenta. En el caso SSt, para los grados de libertad ν , la estimación mejora considerablemente para tamaños de muestra mayor a 250 aproximadamente. Al respecto observamos que tanto el sesgo como la RMSE de las estimaciones cumplen las propiedades asintóticas de los estimadores de máxima verosimilitud a medida que el tamaño de muestra se incrementa ya que el RMSE y el sesgo se acercan a cero.

En la aplicación se ajustó el modelo propuesto para el caso normal y t -student en la que se observó que el mejor modelo para el conjunto de datos biaxial es el caso normal con $g = 3$ componentes. Es preciso recordar que los modelos de regresión no lineal son extremadamente sensibles a los valores iniciales que se utilicen para iniciar los algoritmos de estimación. En este sentido, los modelos de regresión no lineal requieren estar acompañados de, tal vez, funciones o teoría de apoyo (como por ejemplo, fórmulas físicas o químicas) para encontrar valores iniciales útiles o cercanos a los valores reales para poder llegar a resultados consistentes. Se observó que el método de aproximación para obtener los errores estándar presentó dificultades para la función no lineal por lo que se usó para ese caso el método de bootstrap, no obstante dicho método de aproximación funcionó correctamente para la primera función no lineal considerada en la aplicación.

6.2. Sugerencias para investigaciones futuras

- Desarrollar el enfoque bayesiano para el modelo propuesta en la tesis.
- Otro aspecto a desarrollar sería la mixtura de regresiones que considera el error como una distribución SNI.

- Considerar el caso asimétrico lineal y no lineal donde el error sigue una mixtura finita de distribuciones de una distribución Sinh-Normal-Asimétrica.



Apéndice A

Apéndice

A.1. Código en R

```
1
2 #####
3 #ALGORITMO EM
4 #####
5
6 #-----#
7 #                               Family                               #
8 #-----#
9
10 if(mfamily == "t"){
11
12     start.time <- Sys.time()
13     criterio <- 1
14     count <- 0
15     teta <- c(alpha, beta, muj, nu, pii)
16     beta0 <- beta
17     muj0 <- muj
18
19     lk <- sum(log(d.mixed.nllgbssmsn(y, x, pii, alpha, beta, muj, nu, sigma, nlf, mfamily)))
20
21     TOLERANCIA <- 1e-6
22     MAX_NU <- 20
23     MIN_NU <- 2.01
24
25     while((criterio > accuracy) && (count <= iter.max)){
26
27         count <- count + 1
28         #print(count)
29         #print(criterio)
30         zij <- matrix(0, n, g)
31         psi2 <- matrix(0, n, g)
32         hij <- matrix(0, n, g)
33
34         for(j in 1:g){
35
36             # E-step: c lculo zij, hij
37             d1 <- nllgdbssmn.fdp(y, x, alpha[j], beta, muj[j], nu, sigma, nlf, mfamily)
38             if(length(which(d1 == 0)) > 0) {d1[which(d1 == 0)] <- .Machine$double.xmin}
39             d2 <- d.mixed.nllgbssmsn(y, x, pii, alpha, beta, muj, nu, sigma, nlf, mfamily)
40             if(length(which(d2 == 0)) > 0) {d2[which(d2 == 0)] <- .Machine$double.xmin}
41
42             #C lculo del zij
43             zij[,j] <- d1*pii[j] / d2 #z sombrero
44
45             #C lculo del psi2
46
47             psi2[,j] <- (2/alpha[j])*sinh((y-nlf(beta,x)-muj[j])/sigma)
48
49             #Fin de Elementos para calcular hij
50             hij[,j] <- zij[,j]*((nu+1)/(nu+psi2[,j]^2))
51
52             # M-step: actualizar pii, alpha
53             pii[j] <- (1/n)*sum(zij[,j])
54             psiem <- sinh((y-nlf(beta,x)-muj[j])/sigma)
55             alpha[j] <- sqrt(4*sum(hij[,j]*psiem^2)/sum(zij[,j]))
56         }
57
58         pii[g] <- 1 - (sum(pii) - pii[g])
59
60         zero.pos <- NULL
```

```

61 zero.pos <- which(pii == 0)
62 if(length(zero.pos) != 0){
63
64     pii[zero.pos] <- 1e-10
65     pii[which(pii == max(pii))] <- max(pii) - sum(pii[zero.pos])
66 }
67
68
69 beta <- nlmnb(start = beta0, maxbetanllogbsmn,
70             gradient = NULL,
71             hessian = NULL,
72             y,x,pii,alpha,muj,nu,sigma,nlf,zij,hij,
73             scale = 1,
74             control = list(),
75             lower = rep(-100,length(beta)),
76             upper = rep(150,length(beta)))$par
77
78 muj <- nlmnb(start = muj0, maxmujnlogbsmn,
79             gradient = NULL,
80             hessian = NULL,
81             y,x,pii,alpha,beta,nu,sigma,nlf,zij,hij,
82             scale = 1,
83             control = list(),
84             lower = rep(-100,length(muj)),
85             upper = rep(150,length(muj)))$par
86
87
88
89 logvero.SS <- function(nu){ sum(log(d.mixed.nlogbssmsn(y, x, pii, alpha, beta,muj, nu, sigma=2, nlf,
90             mfamily))) }
91 nu <- optimize(f=logvero.SS, interval=c(MIN_NU,MAX_NU), maximum=TRUE,tol=TOLERANCIA)$maximum
92
93 teta <- c(alpha, beta, muj, nu, pii)
94 auxlog <- d.mixed.nlogbssmsn(y, x, pii, alpha, beta,muj, nu,sigma, nlf, mfamily)
95
96 if(length(which(auxlog == 0)) > 0) auxlog[which(auxlog == 0)] <- .Machine$double.xmin
97 lk1 <- sum(log(auxlog)) #valor de la verosimilitud post iteraci n
98 criterio <- abs(lk1/lk-1) #Se usa para parar el algoritmo, lk es densidad total anterior
99 lk <- lk1 #actualiza el punto de comparaci n del valor de densidad
100 }
101
102 end.time <- Sys.time()
103 time.taken <- end.time - start.time
104 }
105
106 d <- g + length(beta) + (g-1)
107 aic <- -2*lk + 2*d
108 bic <- -2*lk + log(n)*d
109 edc <- -2*lk + 0.2*sqrt(n)*d
110
111 result <- list(alpha=alpha,beta=beta,muj=muj,nu=nu,pii=pii,iter=count,criterio = criterio, n=length(y),
112             time = time.taken, convergence = criterio<accuracy,aic = aic, bic = bic, edc = edc,lk=lk)
113
114 obj.out <- list(result = result)
115 class(obj.out) = "fm-bssmn"
116 return(obj.out)
117 }
118
119 #####
120 #ESTIMACION BOOTSTRAP PARA LA DESVIACION EST NDAR#
121 #####
122 source("Funciones RNLSNI_mu.R")
123 source("AlgEM-RNLSNI_mu.R")
124
125 #####TEORICOS#####
126 pii <- c(0.2,0.8)
127 g <- 2
128 alpha1 <- 0.5
129 alpha2 <- 1.0
130 beta <- c(4,5,3,1.5)
131 alpha <- c(alpha1,alpha2)
132 muj <- c(4,-1)
133 nu <- 3 #nu <- c(0.1,0.1)
134 mfamily <- "t"
135 g <- 2
136 n <- c(200,250,500,1000)
137 sigma <-2
138
139 pii[1]*muj[1] + pii[2]*muj[2] #verificacion
140

```

```

141 teoricos <- matrix(c(alpha,beta,muj, nu,pii[1]),nrow=1)
142 colnames(teoricos) <- c("alpha1","alpha2","beta1","beta2","beta3","beta4","muj1","muj2","nu","pii1")
143
144 #####
145
146 dnlf<-function(beta,x)
147 {
148   f1 <- t(x[,1])
149   f2 <- t(x[,2])
150   f3 <- t(exp(beta[4]*x[,3]))
151   f4 <- t(beta[3]*x[,3]*exp(beta[4]*x[,3]))
152   fb <- rbind(f1,f2,f3,f4)
153   return(fb)
154 }
155
156 nlf<-function(beta,x)
157 {
158   fb <- beta[1]*x[,1]+beta[2]*x[,2]+beta[3]*exp(beta[4]*x[,3])
159   return(fb)
160 }
161
162 #####
163 set.seed(123)
164 for(j in 1:length(n)){
165   x <- as.matrix(read.csv(file=paste("x",n[j],"csv",sep="."))); x<-x[-1,]
166   y <- nlmix(n[j],x, alpha, beta, muj,sigma=2,pii,nu, nlf,mfamily="t")$y
167   M <- cbind(y,x)
168
169 #####
170 #preparamos para el bootstrap
171 B<-2*n[j]
172 res<-list()
173 lista<-list()
174
175 #Cambiamos la funcion para bootstrap
176 k<-1
177 while(k<=B){
178   lista[[k]]<-M[sample(nrow(M), n[j],replace=T),]
179   fit <- try(EMmixnllogbssmn(y=lista[[k]][,1], x=lista[[k]][,c(2,3,4)], alpha, beta ,muj, nu, sigma=2, nlf,
180     pii, g = g, get.init = FALSE, mfamil="t", criteria = TRUE, accuracy = 10^-5, iter.max = 500))
181   if(class(fit)!="try-error" && fit$result$convergence == TRUE)
182   {
183     res[[k]] <- c(fit$result$alpha,fit$result$beta,fit$result$muj,fit$result$nu,fit$result$pii[1])
184     print(k)
185     k<-k+1
186   }
187 }
188 result<-array(0,dim=c(B,ncol=g + length(beta)+length(muj) + 1 + 1))
189 colnames(result) <- c("alpha1","alpha2","beta1","beta2","beta3","beta4","muj1", "muj2","nu","pii1")
190 for(i in 1:B){
191   result[i,]<-res[[i]]
192 }
193 #result
194 #apply(result,2,sd)
195 write.csv(result,paste("bootstrap",n[j],"csv", sep="."))
196 }
197
198
199
200
201 #####
202 #FUNCIONES PARA LA SIMULACION
203 #####
204
205 #errores de la SNI
206 gen.erro.SNI.smn <- function(n, alpha, muj,sigma,nu,mfamily)
207 {
208   if(mfamily=="Normal")
209   {
210     z <- rnorm(n)
211     e <- muj+sigma*asinh(alpha*z*0.5)
212   }
213
214   if(mfamily=="t")
215   {
216     z <- rnorm(n)
217     u <- rgamma(n,nu/2,nu/2)
218     e <- muj+sigma*asinh(alpha*z*u^{-0.5}*0.5)
219   }
220   return(e)
221 }

```

```

222
223 #Generando las mixturas de errores de distintos componentes de mixtura
224 #SNI
225 rmix.SNI.smn = function(n,alpha,muj,sigma,pii,nu,mfamily)
226 {
227   #as.numeric(Sys.time())-> t; set.seed((t - floor(t)) * 1e8 -> seed)
228   e <- vector()
229   G <- length(alpha)
230
231   z <- sample(G,size=n,replace=TRUE,prob=pii)
232   for(i in 1:n){
233
234     e[i] = gen.erro.SNI.smn(1, alpha[z[i]], muj[z[i]],sigma,nu,mfamily)
235
236   }
237   return(list(z=z,e=e))
238 }
239
240 nlmix=function(n,x, alpha, beta, muj,sigma,pii,nu,nlf,mfamily){
241   ez<-rmix.SNI.smn(n,alpha,muj,sigma=2,pii,nu,mfamily)
242   z<-ez[[1]]
243   e<-ez[[2]]
244   y <- vector()
245   for(i in 1:n){
246     y[i]= nlf(beta,t(as.matrix(x[i,]))) + e[i]
247   }
248   return(list(y=y,e=e,z=z))
249 }
250
251 #Las funciones de densidad
252
253 nllogdbssmn.fdp<-function(y, x, alpha, beta, muj, nu,sigma, nlf, mfamil y = "t")
254 {
255   if(!is.numeric(alpha)||!is.numeric(nu))
256     if(alpha<=0){stop("alpha must be positive")}
257   psi2 <- (2/alpha)*sinh((y-nlf(beta,x)-muj)/sigma)
258   psi1 <- (2/alpha)*cosh((y-nlf(beta,x)-muj)/sigma)
259   if(mfamily == "t")
260   {
261     pdf <- (gamma((nu+1)/2)/(sqrt(pi)*gamma(nu/2)))*(nu^(0.5*nu)/sigma)*(nu+psi2^2)^(-0.5*(nu+1))*psi1
262   }
263   return(pdf)
264 }
265 #nllogdbssmn.fdp(y, x, alpha, beta, nu, nlf, mfamil y = "t")
266
267 d.mixed.nllogbssmsn <- function(y, x, pii, alpha, beta,muj, nu,sigma, nlf, mfamil y="t") # se puede sacar la
268   verosimilitud
269 {
270   # x: vector de datos
271   # otros parametros deben ser de tipo vector c() de dimensi n g (qtd de mixturas)
272   g <- length(pii)
273   dens <- 0
274   for (j in 1:g) dens <- dens + pii[j]*nllogdbssmn.fdp(y, x, alpha[j], beta, muj[j], nu, sigma, nlf, mfamil y="t")
275   #Densidad de yi con mixtura
276   return(dens)
277 }
278
279 #Funci n Q
280
281 maxbetanllogbssmn <- function(beta, y, x, pii, alpha,muj, nu, sigma, nlf, zij, hij) #dependiendo de la forma de Q
282   se maximiza beta.
283 {
284   n <- length(y)
285   g <- length(pii)
286
287   if(g==1)
288   {
289     psi2 <- (2/alpha[1])*sinh((y-nlf(beta,x)-muj[1])/sigma)
290     psi1 <- (2/alpha[1])*cosh((y-nlf(beta,x)-muj[1])/sigma)
291     Q <- sum(-zij[,1]*log(pii[1])) - sum(zij[,1]*log(psi1)) + 0.5*sum(hij[,1]*psi2^2) #o hij es zi*E(Uly)
292   }
293
294   if(g==2)
295   {
296     psi2_1 <- (2/alpha[1])*sinh((y-nlf(beta,x)-muj[1])/sigma)
297     psi1_1 <- (2/alpha[1])*cosh((y-nlf(beta,x)-muj[1])/sigma)
298     psi2_2 <- (2/alpha[2])*sinh((y-nlf(beta,x)-muj[2])/sigma)
299     psi1_2 <- (2/alpha[2])*cosh((y-nlf(beta,x)-muj[2])/sigma)
300     Q <- sum(-zij[,1]*log(pii[1])) - sum(zij[,1]*log(psi1_1)) + 0.5*sum(hij[,1]*psi2_1^2)+
301       sum(-zij[,2]*log(pii[2])) - sum(zij[,2]*log(psi1_2)) + 0.5*sum(hij[,2]*psi2_2^2)
302   }
303 }

```

```

301 if(g==3)
302 {
303   psi2_1 <- (2/alpha[1])*sinh((y-nlf(beta,x)-muj[1])/sigma)
304   psi1_1 <- (2/alpha[1])*cosh((y-nlf(beta,x)-muj[1])/sigma)
305   psi2_2 <- (2/alpha[2])*sinh((y-nlf(beta,x)-muj[2])/sigma)
306   psi1_2 <- (2/alpha[2])*cosh((y-nlf(beta,x)-muj[2])/sigma)
307   psi2_3 <- (2/alpha[3])*sinh((y-nlf(beta,x)-muj[3])/sigma)
308   psi1_3 <- (2/alpha[3])*cosh((y-nlf(beta,x)-muj[3])/sigma)
309   Q <- sum(-zij[,1]*log(pii[1])) - sum(zij[,1]*log(psi1_1)) + 0.5*sum(hij[,1]*psi2_1^2)+
310     sum(-zij[,2]*log(pii[2])) - sum(zij[,2]*log(psi1_2)) + 0.5*sum(hij[,2]*psi2_2^2)+
311     sum(-zij[,3]*log(pii[3])) - sum(zij[,3]*log(psi1_3)) + 0.5*sum(hij[,3]*psi2_3^2)
312 }
313
314 if(g==4)
315 {
316   psi2_1 <- (2/alpha[1])*sinh((y-nlf(beta,x)-muj[1])/sigma)
317   psi1_1 <- (2/alpha[1])*cosh((y-nlf(beta,x)-muj[1])/sigma)
318   psi2_2 <- (2/alpha[2])*sinh((y-nlf(beta,x)-muj[2])/sigma)
319   psi1_2 <- (2/alpha[2])*cosh((y-nlf(beta,x)-muj[2])/sigma)
320   psi2_3 <- (2/alpha[3])*sinh((y-nlf(beta,x)-muj[3])/sigma)
321   psi1_3 <- (2/alpha[3])*cosh((y-nlf(beta,x)-muj[3])/sigma)
322   psi2_4 <- (2/alpha[4])*sinh((y-nlf(beta,x)-muj[4])/sigma)
323   psi1_4 <- (2/alpha[4])*cosh((y-nlf(beta,x)-muj[4])/sigma)
324   Q <- sum(-zij[,1]*log(pii[1])) - sum(zij[,1]*log(psi1_1)) + 0.5*sum(hij[,1]*psi2_1^2)+
325     sum(-zij[,2]*log(pii[2])) - sum(zij[,2]*log(psi1_2)) + 0.5*sum(hij[,2]*psi2_2^2)+
326     sum(-zij[,3]*log(pii[3])) - sum(zij[,3]*log(psi1_3)) + 0.5*sum(hij[,3]*psi2_3^2)+
327     sum(-zij[,4]*log(pii[4])) - sum(zij[,4]*log(psi1_4)) + 0.5*sum(hij[,4]*psi2_4^2)
328 }
329 }
330
331 if(g==5)
332 {
333   psi2_1 <- (2/alpha[1])*sinh((y-nlf(beta,x)-muj[1])/sigma)
334   psi1_1 <- (2/alpha[1])*cosh((y-nlf(beta,x)-muj[1])/sigma)
335   psi2_2 <- (2/alpha[2])*sinh((y-nlf(beta,x)-muj[2])/sigma)
336   psi1_2 <- (2/alpha[2])*cosh((y-nlf(beta,x)-muj[2])/sigma)
337   psi2_3 <- (2/alpha[3])*sinh((y-nlf(beta,x)-muj[3])/sigma)
338   psi1_3 <- (2/alpha[3])*cosh((y-nlf(beta,x)-muj[3])/sigma)
339   psi2_4 <- (2/alpha[4])*sinh((y-nlf(beta,x)-muj[4])/sigma)
340   psi1_4 <- (2/alpha[4])*cosh((y-nlf(beta,x)-muj[4])/sigma)
341   psi2_5 <- (2/alpha[5])*sinh((y-nlf(beta,x)-muj[5])/sigma)
342   psi1_5 <- (2/alpha[5])*cosh((y-nlf(beta,x)-muj[5])/sigma)
343   Q <- sum(-zij[,1]*log(pii[1])) - sum(zij[,1]*log(psi1_1)) + 0.5*sum(hij[,1]*psi2_1^2)+
344     sum(-zij[,2]*log(pii[2])) - sum(zij[,2]*log(psi1_2)) + 0.5*sum(hij[,2]*psi2_2^2)+
345     sum(-zij[,3]*log(pii[3])) - sum(zij[,3]*log(psi1_3)) + 0.5*sum(hij[,3]*psi2_3^2)+
346     sum(-zij[,4]*log(pii[4])) - sum(zij[,4]*log(psi1_4)) + 0.5*sum(hij[,4]*psi2_4^2)+
347     sum(-zij[,5]*log(pii[5])) - sum(zij[,5]*log(psi1_5)) + 0.5*sum(hij[,5]*psi2_5^2)
348 }
349 }
350 if(g==6)
351 {
352   psi2_1 <- (2/alpha[1])*sinh((y-nlf(beta,x)-muj[1])/sigma)
353   psi1_1 <- (2/alpha[1])*cosh((y-nlf(beta,x)-muj[1])/sigma)
354   psi2_2 <- (2/alpha[2])*sinh((y-nlf(beta,x)-muj[2])/sigma)
355   psi1_2 <- (2/alpha[2])*cosh((y-nlf(beta,x)-muj[2])/sigma)
356   psi2_3 <- (2/alpha[3])*sinh((y-nlf(beta,x)-muj[3])/sigma)
357   psi1_3 <- (2/alpha[3])*cosh((y-nlf(beta,x)-muj[3])/sigma)
358   psi2_4 <- (2/alpha[4])*sinh((y-nlf(beta,x)-muj[4])/sigma)
359   psi1_4 <- (2/alpha[4])*cosh((y-nlf(beta,x)-muj[4])/sigma)
360   psi2_5 <- (2/alpha[5])*sinh((y-nlf(beta,x)-muj[5])/sigma)
361   psi1_5 <- (2/alpha[5])*cosh((y-nlf(beta,x)-muj[5])/sigma)
362   psi2_6 <- (2/alpha[6])*sinh((y-nlf(beta,x)-muj[6])/sigma)
363   psi1_6 <- (2/alpha[6])*cosh((y-nlf(beta,x)-muj[6])/sigma)
364   Q <- sum(-zij[,1]*log(pii[1])) - sum(zij[,1]*log(psi1_1)) + 0.5*sum(hij[,1]*psi2_1^2)+
365     sum(-zij[,2]*log(pii[2])) - sum(zij[,2]*log(psi1_2)) + 0.5*sum(hij[,2]*psi2_2^2)+
366     sum(-zij[,3]*log(pii[3])) - sum(zij[,3]*log(psi1_3)) + 0.5*sum(hij[,3]*psi2_3^2)+
367     sum(-zij[,4]*log(pii[4])) - sum(zij[,4]*log(psi1_4)) + 0.5*sum(hij[,4]*psi2_4^2)+
368     sum(-zij[,5]*log(pii[5])) - sum(zij[,5]*log(psi1_5)) + 0.5*sum(hij[,5]*psi2_5^2)+
369     sum(-zij[,6]*log(pii[6])) - sum(zij[,6]*log(psi1_6)) + 0.5*sum(hij[,6]*psi2_6^2)
370 }
371 }
372 if(g==7)
373 {
374   psi2_1 <- (2/alpha[1])*sinh((y-nlf(beta,x)-muj[1])/sigma)
375   psi1_1 <- (2/alpha[1])*cosh((y-nlf(beta,x)-muj[1])/sigma)
376   psi2_2 <- (2/alpha[2])*sinh((y-nlf(beta,x)-muj[2])/sigma)
377   psi1_2 <- (2/alpha[2])*cosh((y-nlf(beta,x)-muj[2])/sigma)
378   psi2_3 <- (2/alpha[3])*sinh((y-nlf(beta,x)-muj[3])/sigma)
379   psi1_3 <- (2/alpha[3])*cosh((y-nlf(beta,x)-muj[3])/sigma)
380   psi2_4 <- (2/alpha[4])*sinh((y-nlf(beta,x)-muj[4])/sigma)
381   psi1_4 <- (2/alpha[4])*cosh((y-nlf(beta,x)-muj[4])/sigma)
382   psi2_5 <- (2/alpha[5])*sinh((y-nlf(beta,x)-muj[5])/sigma)

```

```

383 psi1_5 <- (2/alpha[5])*cosh((y-nlf(beta,x)-muj[5])/sigma)
384 psi2_6 <- (2/alpha[6])*sinh((y-nlf(beta,x)-muj[6])/sigma)
385 psi1_6 <- (2/alpha[6])*cosh((y-nlf(beta,x)-muj[6])/sigma)
386 psi2_7 <- (2/alpha[7])*sinh((y-nlf(beta,x)-muj[7])/sigma)
387 psi1_7 <- (2/alpha[7])*cosh((y-nlf(beta,x)-muj[7])/sigma)
388 Q <- sum(-zij[,1]*log(pii[1])) - sum(zij[,1]*log(psi1_1)) + 0.5*sum(hij[,1]*psi2_1^2)+
389 sum(-zij[,2]*log(pii[1])) - sum(zij[,2]*log(psi1_2)) + 0.5*sum(hij[,2]*psi2_2^2)+
390 sum(-zij[,3]*log(pii[1])) - sum(zij[,3]*log(psi1_3)) + 0.5*sum(hij[,3]*psi2_3^2)+
391 sum(-zij[,4]*log(pii[1])) - sum(zij[,4]*log(psi1_4)) + 0.5*sum(hij[,4]*psi2_4^2)+
392 sum(-zij[,5]*log(pii[1])) - sum(zij[,5]*log(psi1_5)) + 0.5*sum(hij[,5]*psi2_5^2)+
393 sum(-zij[,6]*log(pii[1])) - sum(zij[,6]*log(psi1_6)) + 0.5*sum(hij[,6]*psi2_6^2)+
394 sum(-zij[,7]*log(pii[1])) - sum(zij[,7]*log(psi1_7)) + 0.5*sum(hij[,7]*psi2_7^2)
395 }
396
397 return(Q)
398 }
399
400 maxmujnllgbsmn <- function(muj, y, x, pii, alpha,beta, nu, sigma, nlf, zij, hij) #dependiendo de la forma de Q se
    maximiza beta.
401 {
402   n <- length(y)
403   g <- length(pii)
404
405   if(g==1)
406   {
407     psi2 <- (2/alpha[1])*sinh((y-nlf(beta,x)-muj[1])/sigma)
408     psi1 <- (2/alpha[1])*cosh((y-nlf(beta,x)-muj[1])/sigma)
409     Q <- sum(-zij[,1]*log(pii[1])) - sum(zij[,1]*log(psi1)) + 0.5*sum(hij[,1]*psi2^2) # hij es zi*E(U|y)
410   }
411
412   if(g==2)
413   {
414     psi2_1 <- (2/alpha[1])*sinh((y-nlf(beta,x)-muj[1])/sigma)
415     psi1_1 <- (2/alpha[1])*cosh((y-nlf(beta,x)-muj[1])/sigma)
416     psi2_2 <- (2/alpha[2])*sinh((y-nlf(beta,x)-muj[2])/sigma)
417     psi1_2 <- (2/alpha[2])*cosh((y-nlf(beta,x)-muj[2])/sigma)
418     Q <- sum(-zij[,1]*log(pii[1])) - sum(zij[,1]*log(psi1_1)) + 0.5*sum(hij[,1]*psi2_1^2)+
419       sum(-zij[,2]*log(pii[2])) - sum(zij[,2]*log(psi1_2)) + 0.5*sum(hij[,2]*psi2_2^2)
420   }
421
422   if(g==3)
423   {
424     psi2_1 <- (2/alpha[1])*sinh((y-nlf(beta,x)-muj[1])/sigma)
425     psi1_1 <- (2/alpha[1])*cosh((y-nlf(beta,x)-muj[1])/sigma)
426     psi2_2 <- (2/alpha[2])*sinh((y-nlf(beta,x)-muj[2])/sigma)
427     psi1_2 <- (2/alpha[2])*cosh((y-nlf(beta,x)-muj[2])/sigma)
428     psi2_3 <- (2/alpha[3])*sinh((y-nlf(beta,x)-muj[3])/sigma)
429     psi1_3 <- (2/alpha[3])*cosh((y-nlf(beta,x)-muj[3])/sigma)
430     Q <- sum(-zij[,1]*log(pii[1])) - sum(zij[,1]*log(psi1_1)) + 0.5*sum(hij[,1]*psi2_1^2)+
431       sum(-zij[,2]*log(pii[2])) - sum(zij[,2]*log(psi1_2)) + 0.5*sum(hij[,2]*psi2_2^2)+
432       sum(-zij[,3]*log(pii[3])) - sum(zij[,3]*log(psi1_3)) + 0.5*sum(hij[,3]*psi2_3^2)
433   }
434
435   if(g==4)
436   {
437     psi2_1 <- (2/alpha[1])*sinh((y-nlf(beta,x)-muj[1])/sigma)
438     psi1_1 <- (2/alpha[1])*cosh((y-nlf(beta,x)-muj[1])/sigma)
439     psi2_2 <- (2/alpha[2])*sinh((y-nlf(beta,x)-muj[2])/sigma)
440     psi1_2 <- (2/alpha[2])*cosh((y-nlf(beta,x)-muj[2])/sigma)
441     psi2_3 <- (2/alpha[3])*sinh((y-nlf(beta,x)-muj[3])/sigma)
442     psi1_3 <- (2/alpha[3])*cosh((y-nlf(beta,x)-muj[3])/sigma)
443     psi2_4 <- (2/alpha[4])*sinh((y-nlf(beta,x)-muj[4])/sigma)
444     psi1_4 <- (2/alpha[4])*cosh((y-nlf(beta,x)-muj[4])/sigma)
445     Q <- sum(-zij[,1]*log(pii[1])) - sum(zij[,1]*log(psi1_1)) + 0.5*sum(hij[,1]*psi2_1^2)+
446       sum(-zij[,2]*log(pii[2])) - sum(zij[,2]*log(psi1_2)) + 0.5*sum(hij[,2]*psi2_2^2)+
447       sum(-zij[,3]*log(pii[3])) - sum(zij[,3]*log(psi1_3)) + 0.5*sum(hij[,3]*psi2_3^2)+
448       sum(-zij[,4]*log(pii[4])) - sum(zij[,4]*log(psi1_4)) + 0.5*sum(hij[,4]*psi2_4^2)
449   }
450
451
452   if(g==5)
453   {
454     psi2_1 <- (2/alpha[1])*sinh((y-nlf(beta,x)-muj[1])/sigma)
455     psi1_1 <- (2/alpha[1])*cosh((y-nlf(beta,x)-muj[1])/sigma)
456     psi2_2 <- (2/alpha[2])*sinh((y-nlf(beta,x)-muj[2])/sigma)
457     psi1_2 <- (2/alpha[2])*cosh((y-nlf(beta,x)-muj[2])/sigma)
458     psi2_3 <- (2/alpha[3])*sinh((y-nlf(beta,x)-muj[3])/sigma)
459     psi1_3 <- (2/alpha[3])*cosh((y-nlf(beta,x)-muj[3])/sigma)
460     psi2_4 <- (2/alpha[4])*sinh((y-nlf(beta,x)-muj[4])/sigma)
461     psi1_4 <- (2/alpha[4])*cosh((y-nlf(beta,x)-muj[4])/sigma)
462     psi2_5 <- (2/alpha[5])*sinh((y-nlf(beta,x)-muj[5])/sigma)
463     psi1_5 <- (2/alpha[5])*cosh((y-nlf(beta,x)-muj[5])/sigma)

```

```

464   Q      <- sum(-zij[,1]*log(pii[1])) - sum(zij[,1]*log(psi1_1)) + 0.5*sum(hij[,1]*psi2_1^2)+
465   sum(-zij[,2]*log(pii[2])) - sum(zij[,2]*log(psi1_2)) + 0.5*sum(hij[,2]*psi2_2^2)+
466   sum(-zij[,3]*log(pii[3])) - sum(zij[,3]*log(psi1_3)) + 0.5*sum(hij[,3]*psi2_3^2)+
467   sum(-zij[,4]*log(pii[4])) - sum(zij[,4]*log(psi1_4)) + 0.5*sum(hij[,4]*psi2_4^2)+
468   sum(-zij[,5]*log(pii[5])) - sum(zij[,5]*log(psi1_5)) + 0.5*sum(hij[,5]*psi2_5^2)
469 }
470
471 if(g==6)
472 {
473   psi2_1 <- (2/alpha[1])*sinh((y-nlf(beta,x)-muj[1])/sigma)
474   psii_1 <- (2/alpha[1])*cosh((y-nlf(beta,x)-muj[1])/sigma)
475   psi2_2 <- (2/alpha[2])*sinh((y-nlf(beta,x)-muj[2])/sigma)
476   psii_2 <- (2/alpha[2])*cosh((y-nlf(beta,x)-muj[2])/sigma)
477   psi2_3 <- (2/alpha[3])*sinh((y-nlf(beta,x)-muj[3])/sigma)
478   psii_3 <- (2/alpha[3])*cosh((y-nlf(beta,x)-muj[3])/sigma)
479   psi2_4 <- (2/alpha[4])*sinh((y-nlf(beta,x)-muj[4])/sigma)
480   psii_4 <- (2/alpha[4])*cosh((y-nlf(beta,x)-muj[4])/sigma)
481   psi2_5 <- (2/alpha[5])*sinh((y-nlf(beta,x)-muj[5])/sigma)
482   psii_5 <- (2/alpha[5])*cosh((y-nlf(beta,x)-muj[5])/sigma)
483   psi2_6 <- (2/alpha[6])*sinh((y-nlf(beta,x)-muj[6])/sigma)
484   psii_6 <- (2/alpha[6])*cosh((y-nlf(beta,x)-muj[6])/sigma)
485   Q      <- sum(-zij[,1]*log(pii[1])) - sum(zij[,1]*log(psi1_1)) + 0.5*sum(hij[,1]*psi2_1^2)+
486   sum(-zij[,2]*log(pii[2])) - sum(zij[,2]*log(psi1_2)) + 0.5*sum(hij[,2]*psi2_2^2)+
487   sum(-zij[,3]*log(pii[3])) - sum(zij[,3]*log(psi1_3)) + 0.5*sum(hij[,3]*psi2_3^2)+
488   sum(-zij[,4]*log(pii[4])) - sum(zij[,4]*log(psi1_4)) + 0.5*sum(hij[,4]*psi2_4^2)+
489   sum(-zij[,5]*log(pii[5])) - sum(zij[,5]*log(psi1_5)) + 0.5*sum(hij[,5]*psi2_5^2)+
490   sum(-zij[,6]*log(pii[6])) - sum(zij[,6]*log(psi1_6)) + 0.5*sum(hij[,6]*psi2_6^2)
491 }
492
493 if(g==7)
494 {
495   psi2_1 <- (2/alpha[1])*sinh((y-nlf(beta,x)-muj[1])/sigma)
496   psii_1 <- (2/alpha[1])*cosh((y-nlf(beta,x)-muj[1])/sigma)
497   psi2_2 <- (2/alpha[2])*sinh((y-nlf(beta,x)-muj[2])/sigma)
498   psii_2 <- (2/alpha[2])*cosh((y-nlf(beta,x)-muj[2])/sigma)
499   psi2_3 <- (2/alpha[3])*sinh((y-nlf(beta,x)-muj[3])/sigma)
500   psii_3 <- (2/alpha[3])*cosh((y-nlf(beta,x)-muj[3])/sigma)
501   psi2_4 <- (2/alpha[4])*sinh((y-nlf(beta,x)-muj[4])/sigma)
502   psii_4 <- (2/alpha[4])*cosh((y-nlf(beta,x)-muj[4])/sigma)
503   psi2_5 <- (2/alpha[5])*sinh((y-nlf(beta,x)-muj[5])/sigma)
504   psii_5 <- (2/alpha[5])*cosh((y-nlf(beta,x)-muj[5])/sigma)
505   psi2_6 <- (2/alpha[6])*sinh((y-nlf(beta,x)-muj[6])/sigma)
506   psii_6 <- (2/alpha[6])*cosh((y-nlf(beta,x)-muj[6])/sigma)
507   psi2_7 <- (2/alpha[7])*sinh((y-nlf(beta,x)-muj[7])/sigma)
508   psii_7 <- (2/alpha[7])*cosh((y-nlf(beta,x)-muj[7])/sigma)
509   Q      <- sum(-zij[,1]*log(pii[1])) - sum(zij[,1]*log(psi1_1)) + 0.5*sum(hij[,1]*psi2_1^2)+
510   sum(-zij[,2]*log(pii[1])) - sum(zij[,2]*log(psi1_2)) + 0.5*sum(hij[,2]*psi2_2^2)+
511   sum(-zij[,3]*log(pii[1])) - sum(zij[,3]*log(psi1_3)) + 0.5*sum(hij[,3]*psi2_3^2)+
512   sum(-zij[,4]*log(pii[1])) - sum(zij[,4]*log(psi1_4)) + 0.5*sum(hij[,4]*psi2_4^2)+
513   sum(-zij[,5]*log(pii[1])) - sum(zij[,5]*log(psi1_5)) + 0.5*sum(hij[,5]*psi2_5^2)+
514   sum(-zij[,6]*log(pii[1])) - sum(zij[,6]*log(psi1_6)) + 0.5*sum(hij[,6]*psi2_6^2)+
515   sum(-zij[,7]*log(pii[1])) - sum(zij[,7]*log(psi1_7)) + 0.5*sum(hij[,7]*psi2_7^2)
516 }
517
518 return(Q)
519 }
520
521 #####
522 #PARA ORDENAR LOS RESULTADOS DE SIMULACION
523 #####
524 library(sn)
525 library(bssn)
526 library(mixsmsn)
527
528 source("Funciones_RNLSNI_mu.R")
529 source("AlgEM-RNLSNI_mu.R")
530
531
532 #####TEORICOS#####
533 ##### Valores iniciales#####
534 pii      <- c(0.2,0.8)
535 g        <- 2
536 alpha1   <- 0.5
537 alpha2   <- 1.0
538 beta     <- c(4,5,3,1.5)
539 alpha    <- c(alpha1,alpha2)
540 muj      <- c(4,-1)
541 nu       <- 3 #nu
542 mfamily  <- "t"
543 g        <- 2
544 n        <- c(50,100,200,250,500,1000)
545 sigma    <- 2

```

```

546
547 pii[1]*muj[1] + pii[2]*muj[2] #verificaci n
548
549
550 #####
551
552 dnlf<-function(beta,x)
553 {
554   f1 <- t(x[,1])
555   f2 <- t(x[,2])
556   f3 <- t(exp(beta[4]*x[,3]))
557   f4 <- t(beta[3]*x[,3]*exp(beta[4]*x[,3]))
558   fb <- rbind(f1,f2,f3,f4)
559   return(fb)
560 }
561
562 nlf<-function(beta,x)
563 {
564   fb <- beta[1]*x[,1]+beta[2]*x[,2]+beta[3]*exp(beta[4]*x[,3])
565   return(fb)
566 }
567
568 #####
569
570 replicate           <- 1000
571
572 set.seed(123)
573 for(k in 1:length(n)){
574   h           <- 1
575   x1          <- runif(n[k])
576   x2          <- runif(n[k])
577   x3          <- runif(n[k])
578   x           <- cbind(x1,x2,x3)
579   write.csv(x, paste("x",n[k], 'csv', sep="."))
580   estimaciones <- array(0,dim=c(replicate,ncol=g + length(beta)+length(muj) + 1 + 1))
581   colnames(estimaciones) <- c("alpha1","alpha2","beta1","beta2","beta3","beta4","muj1", "muj2","nu","pii1")
582
583 while(h <= replicate){
584   y           <- nlmix(n[k],x, alpha, beta, muj,sigma=2,pii,nu, nlf,mfamily="t")$y
585   fit         <- try(EMmixnllogbssmn(y, x, alpha, beta ,muj, nu, sigma=2, nlf, pii, g = g, get.init = FALSE, mfamil=
586     "t", criteria = TRUE, accuracy = 10^-5, iter.max = 500))
587   if(class(fit)!="try-error" && fit$result$convergence == TRUE)
588   {
589     print(h)
590     estimaciones[h,] <- c(fit$result$alpha,fit$result$beta,fit$result$muj,fit$result$nu,fit$result$pii[1])
591     h <- h+1
592   }
593
594   teóricos    <- matrix(c(alpha,beta,muj, nu,pii[1]),nrow=1)
595   colnames(teóricos) <- c("alpha1","alpha2","beta1","beta2","beta3","beta4","muj1","muj2","nu","pii1")
596   teóricos;round(apply(estimaciones,2,mean), digits= 4); round(apply(estimaciones,2,sd), digits= 4)
597
598 #Tabla resumen
599 resumen1<-data.frame(round(apply(estimaciones,2,mean), digits= 4)) #estimaciones
600 resumen2<-data.frame(round(apply(estimaciones,2,sd), digits= 4)) #Monte carlo
601
602 teóricos1<- data.frame(t(teóricos))
603
604 resumen<-cbind(teóricos1,resumen1,resumen2)
605 colnames(resumen)<-c("Teóricos","Estimaci n", "SE MC")
606 resumen
607 }
608
609 estimaciones<-data.frame(estimaciones)
610 estimaciones[, "n"]<-n[k]
611 write.csv(estimaciones,paste("estimaciones",n[k], 'csv', sep="."))
612 write.csv(resumen,paste("resumen",n[k], 'csv', sep="."))
613
614
615 library(mixsmn)
616 library(bssn)
617
618 setwd("C:/Users/cocam/OneDrive/Documentos/2020/R code/Rev")
619
620 source("Funciones RNLSNI_mu.R")
621 source("AlgEM-RNLSNI_mu.R")
622
623 #####TEORICOS#####
624 pii <- c(0.2,0.8)
625 g <- 2
626 alpha1 <- 0.5

```



```

627 alpha2 <- 1.0
628 beta <- c(4,5,3,1.5)
629 alpha <- c(alpha1,alpha2)
630 muj <- c(4,-1)
631 nu <- 3 #nu <- c(0.1,0.1)
632 mfamily <- "t"
633 g <- 2
634 n <- c(50,100,200,250,500,1000)
635 sigma <-2
636
637 pii[1]*muj[1] + pii[2]*muj[2] #verificacion
638 replicate<-1000
639
640 #####
641 bias = rmse <- array(0,dim=c(length(n),ncol=g + length(beta) + length(muj) +1 + 1)) #alpha + beta + nu
+ p1 --- matriz 7 (n)*8(nro de par metros)
642 colnames(bias) <- c("BIASalpha1","BIASalpha2","BIASbeta1","BIASbeta2","BIASbeta3","BIASbeta4","BIASmuj1",
BIASmuj2","BIASnu","BIASpii1")
643 colnames(rmse) <- c("RMSEalpha1","RMSEalpha2","RMSEbeta1","RMSEbeta2","RMSEbeta3","RMSEbeta4","RMSEmuj1",
RMSEmuj2","RMSEnu","RMSEpii1")
644 rownames(bias) <- c("50","100","200","250","500","1000")
645 vt <- c(alpha,beta,muj, nu,pii[1])#valores teoricos
646 vtmatrixBIAS <- matrix(c(vt),nc=ncol(bias),nr=replicate,byrow=TRUE)
647 vtmatrixRMSE <- matrix(c(vt),nc=ncol(rmse),nr=replicate,byrow=TRUE)
648
649 for(k in 1:length(n)){
650 estimaciones<-read.csv(file=paste("estimaciones",n[k],"csv",sep="."))
651 estimaciones<-estimaciones[,-c(1,12)]
652 bias[k,] <- (apply(estimaciones-vtmatrixBIAS,2,mean)) #media de la desviaci n
653 rmse[k,] <- sqrt(apply((estimaciones-vtmatrixRMSE)^2,2,mean)) #error cuadr tico medio
654 df <- data.frame(bias[k,],rmse[k,])#,rb[k,])
655 #write.csv(round(df, digits=4), paste("FM_NL_LOG_BS_SMN_VIES_RMSE_RB_resultadosSimul2","tama o_muestra=",n[k],
replicate=',replicate',"family=",mfamily,'csv',sep="."))
656 }
657 df <- data.frame(bias,rmse)#,rb)
658 write.csv(df, paste("RNLSNI_RSE_BIAS",mfamily,'csv',sep="."))
659
660 #Graphics
661 setwd("C:/Users/cocam/OneDrive/Documentos/2021/Resultados")
662 ST <- read.csv(file="RNLSNI_RSE_BIAS.t.csv",head=TRUE)#,sep="")
663
664 n=c(50,100,200,250,500,1000)
665
666 #Graficos #dev.size("in") #comando para conocer o tamanho da tela
667 gray1 <- rgb(51,51,51,max=255)
668 gray2 <- rgb(112,112,112,max=255)
669 gray3 <- rgb(150,150,150,max=255)
670
671 #BIAS
672 #BIAS alpha1 #max(c(SN[,2],SCN[,2],ST[,2]))
673 #postscript("exp2biasBeta0joint.eps", width=5.75, height=5.75, horizontal=FALSE, onefile=TRUE)
674 #par(mfrow=c(3,2))
675
676 A1<-round(c(min(ST[,2]), max(ST[,2])),4); A2<-(A1[2]-A1[1])/10
677 B1<-round(c(min(ST[,3]), max(ST[,3])),4); B2<-(B1[2]-B1[1])/10
678 C1<-round(c(min(ST[,4]), max(ST[,4])),4); C2<-(C1[2]-C1[1])/10
679 D1<-round(c(min(ST[,5]), max(ST[,5])),4); D2<-(D1[2]-D1[1])/10
680 E1<-round(c(min(ST[,6]), max(ST[,6])),4); E2<-(E1[2]-E1[1])/10
681 F1<-round(c(min(ST[,7]), max(ST[,7])),4); F2<-(F1[2]-F1[1])/10
682 G1<-round(c(min(ST[,8]), max(ST[,8])),4); G2<-(G1[2]-G1[1])/10
683 H1<-round(c(min(ST[,9]), max(ST[,9])),4); H2<-(H1[2]-H1[1])/10
684 I1<-round(c(min(ST[,10]), max(ST[,10])),4); I2<-(I1[2]-I1[1])/10
685 J1<-round(c(min(ST[,11]), max(ST[,11])),4); J2<-(J1[2]-J1[1])/10
686
687 setwd("C:/Users/cocam/OneDrive/Documentos/2021/Borrador/figuras")
688
689 #alpha1
690 postscript("plotAlpha1Bias.eps", width=5.75, height=5.75, horizontal=FALSE, onefile=TRUE)
691 plot(c(50,1000),A1,type="n",xlab="Tama o de muestra (n)", ylab="Sesgo", xaxt="n", yaxt="n")
692 axis(1,seq(50,1000,50))
693 axis(2,seq(A1[1],A1[2],A2))
694 s1=smooth.spline(n,ST[,2] , spar=0.25)
695 lines(s1,type="o",pch=16,col=gray1, lwd=1, lty=2)
696 abline(0,0,col="blue",lty=2)
697 title(expression(paste(alpha[1]))) #Titulo de grafico
698 abline(h=0,lty=3) #Linea horizontal
699 #legend('topright',c("SN","ST","SCN"),col=c("black",gray1,gray2),pch=c(15,16,17),lty=c(1,2,3),lwd=1,bty="n",inset
=0.1,cex=0.8)
700 dev.off() #Cerrando el postscript
701
702 #alpha2
703 postscript("plotAlpha2Bias.eps", width=5.75, height=5.75, horizontal=FALSE, onefile=TRUE)

```

```

704 plot(c(50,1000),B1,type="n",xlab="Tama o de muestra (n)", ylab="Sesgo", xaxt="n", yaxt="n")
705 axis(1,seq(50,1000,50))
706 axis(2,seq(B1[1],B1[2],B2))
707 s2=smooth.spline(n,ST[,3] , spar=0.25)
708 lines(s2,type="o",pch=16,col=gray1, lwd=1,nty=2)
709 abline(0,0,col="blue",lty=2)
710 title(expression(paste(alpha[2]))) #Titulo de grafico
711 abline(h=0,lty=3) #Linea horizontal
712 #legend('topright',c("SN","ST","SCN"),col=c("black",gray1,gray2),pch=c(15,16,17),lty=c(1,2,3),lwd=1,bty="n",inset
=0.1,cex=0.8)
713 dev.off() #Cerrando el postscript
714
715 #beta1
716 postscript("plotBeta1Bias.eps", width=5.75, height=5.75, horizontal=FALSE, onefile=TRUE)
717 plot(c(50,1000),C1,type="n",xlab="Tama o de muestra (n)", ylab="Sesgo", xaxt="n", yaxt="n")
718 axis(1,seq(50,1000,50))
719 axis(2,seq(C1[1],C1[2],C2))
720 s3=smooth.spline(n,ST[,4] , spar=0.25)
721 lines(s3,type="o",pch=16,col=gray1, lwd=1,nty=2)
722 abline(0,0,col="blue",lty=2)
723 title(expression(paste(beta[1]))) #Titulo de grafico
724 abline(h=0,lty=3) #Linea horizontal
725 #legend('topright',c("SN","ST","SCN"),col=c("black",gray1,gray2),pch=c(15,16,17),lty=c(1,2,3),lwd=1,bty="n",inset
=0.1,cex=0.8)
726 dev.off() #Cerrando el postscript
727
728 #beta2
729 postscript("plotBeta2Bias.eps", width=5.75, height=5.75, horizontal=FALSE, onefile=TRUE)
730 plot(c(50,1000),D1,type="n",xlab="Tama o de muestra (n)", ylab="Sesgo", xaxt="n", yaxt="n")
731 axis(1,seq(50,1000,50))
732 axis(2,seq(D1[1],D1[2],D2))
733 s4=smooth.spline(n,ST[,5] , spar=0.25)
734 lines(s4,type="o",pch=16,col=gray1, lwd=1,nty=2)
735 abline(0,0,col="blue",lty=2)
736 title(expression(paste(beta[2]))) #Titulo de grafico
737 abline(h=0,lty=3) #Linea horizontal
738 #legend('topright',c("SN","ST","SCN"),col=c("black",gray1,gray2),pch=c(15,16,17),lty=c(1,2,3),lwd=1,bty="n",inset
=0.1,cex=0.8)
739 dev.off() #Cerrando el postscript
740
741 #beta3
742 postscript("plotBeta3Bias.eps", width=5.75, height=5.75, horizontal=FALSE, onefile=TRUE)
743 plot(c(50,1000),E1,type="n",xlab="Tama o de muestra (n)", ylab="Sesgo", xaxt="n", yaxt="n")
744 axis(1,seq(50,1000,50))
745 axis(2,seq(E1[1],E1[2],E2))
746 s5=smooth.spline(n,ST[,6] , spar=0.25)
747 lines(s5,type="o",pch=16,col=gray1, lwd=1,nty=2)
748 abline(0,0,col="blue",lty=2)
749 title(expression(paste(beta[3]))) #Titulo de grafico
750 abline(h=0,lty=3) #Linea horizontal
751 #legend('topright',c("SN","ST","SCN"),col=c("black",gray1,gray2),pch=c(15,16,17),lty=c(1,2,3),lwd=1,bty="n",inset
=0.1,cex=0.8)
752 dev.off() #Cerrando el postscript
753
754 #beta4
755 postscript("plotBeta4Bias.eps", width=5.75, height=5.75, horizontal=FALSE, onefile=TRUE)
756 plot(c(50,1000),F1,type="n",xlab="Tama o de muestra (n)", ylab="Sesgo", xaxt="n", yaxt="n")
757 axis(1,seq(50,1000,50))
758 axis(2,seq(F1[1],F1[2],F2))
759 s6=smooth.spline(n,ST[,7] , spar=0.25)
760 lines(s6,type="o",pch=16,col=gray1, lwd=1,nty=2)
761 abline(0,0,col="blue",lty=2)
762 title(expression(paste(beta[4]))) #Titulo de grafico
763 abline(h=0,lty=3) #Linea horizontal
764 #legend('topright',c("SN","ST","SCN"),col=c("black",gray1,gray2),pch=c(15,16,17),lty=c(1,2,3),lwd=1,bty="n",inset
=0.1,cex=0.8)
765 dev.off() #Cerrando el postscript
766
767 #muj1
768 postscript("plotMuj1Bias.eps", width=5.75, height=5.75, horizontal=FALSE, onefile=TRUE)
769 plot(c(50,1000),G1,type="n",xlab="Tama o de muestra (n)", ylab="Sesgo", xaxt="n", yaxt="n")
770 axis(1,seq(50,1000,50))
771 axis(2,seq(G1[1],G1[2],G2))
772 s7=smooth.spline(n,ST[,8] , spar=0.25)
773 lines(s7,type="o",pch=16,col=gray1, lwd=1,nty=2)
774 abline(0,0,col="blue",lty=2)
775 title(expression(paste(mu[1]))) #Titulo de grafico
776 abline(h=0,lty=3) #Linea horizontal
777 #legend('topright',c("SN","ST","SCN"),col=c("black",gray1,gray2),pch=c(15,16,17),lty=c(1,2,3),lwd=1,bty="n",inset
=0.1,cex=0.8)
778 dev.off() #Cerrando el postscript
779

```

```

780 #muj2
781 postscript("plotMuj2Bias.eps", width=5.75, height=5.75, horizontal=FALSE, onefile=TRUE)
782 plot(c(50,1000),H1,type="n",xlab="Tama o de muestra (n)", ylab="Sesgo", xaxt="n", yaxt="n")
783 axis(1,seq(50,1000,50))
784 axis(2,seq(H1[1],H1[2],H2))
785 s8=smooth.spline(n,ST[,9] , spar=0.25)
786 lines(s8,type="o",pch=16,col=gray1, lwd=1, lty=2)
787 abline(0,0,col="blue",lty=2)
788 title(expression(paste(mu[2]))) #Titulo de grafico
789 abline(h=0,lty=3) #Linea horizontal
790 #legend('topright',c("SN","ST","SCN"),col=c("black",gray1,gray2),pch=c(15,16,17),lty=c(1,2,3),lwd=1,bty="n",inset
=0.1,cex=0.8)
791 dev.off() #Cerrando el postscript
792
793 #nu
794 postscript("plotNuBias.eps", width=5.75, height=5.75, horizontal=FALSE, onefile=TRUE)
795 plot(c(50,1000),I1,type="n",xlab="Tama o de muestra (n)", ylab="Sesgo", xaxt="n", yaxt="n")
796 axis(1,seq(50,1000,50))
797 axis(2,seq(I1[1],I1[2],I2))
798 s9=smooth.spline(n,ST[,10] , spar=0.25)
799 lines(s9,type="o",pch=16,col=gray1, lwd=1, lty=2)
800 abline(0,0,col="blue",lty=2)
801 title(expression(paste(nu))) #Titulo de grafico
802 abline(h=0,lty=3) #Linea horizontal
803 #legend('topright',c("SN","ST","SCN"),col=c("black",gray1,gray2),pch=c(15,16,17),lty=c(1,2,3),lwd=1,bty="n",inset
=0.1,cex=0.8)
804 dev.off() #Cerrando el postscript
805
806 #p1
807 postscript("plotPiBias.eps", width=5.75, height=5.75, horizontal=FALSE, onefile=TRUE)
808 plot(c(50,1000),J1,type="n",xlab="Tama o de muestra (n)", ylab="Sesgo", xaxt="n", yaxt="n")
809 axis(1,seq(50,1000,50))
810 axis(2,seq(J1[1],J1[2],J2))
811 s10=smooth.spline(n,ST[,11] , spar=0.25)
812 lines(s10,type="o",pch=16,col=gray1, lwd=1, lty=2)
813 abline(0,0,col="blue",lty=2)
814 title(expression(paste(p))) #Titulo de grafico
815 abline(h=0,lty=3) #Linea horizontal
816 #legend('topright',c("SN","ST","SCN"),col=c("black",gray1,gray2),pch=c(15,16,17),lty=c(1,2,3),lwd=1,bty="n",inset
=0.1,cex=0.8)
817 dev.off() #Cerrando el postscript
818
819
820 #####
821
822 #####
823 #RMSE beta0 #max(c(SN[,12],SCN[,13],ST[,13]))
824 #postscript("exp2rmseBeta0joint.eps", width=5.75, height=5.75, horizontal=FALSE, onefile=TRUE)
825
826 #par(mfrow=c(3,2))
827
828 a1<-round(c(min(ST[,12]), max(ST[,12])),4); a2<-(a1[2]-a1[1])/10
829 b1<-round(c(min(ST[,13]), max(ST[,13])),4); b2<-(b1[2]-b1[1])/10
830 c1<-round(c(min(ST[,14]), max(ST[,14])),4); c2<-(c1[2]-c1[1])/10
831 d1<-round(c(min(ST[,15]), max(ST[,15])),4); d2<-(d1[2]-d1[1])/10
832 e1<-round(c(min(ST[,16]), max(ST[,16])),4); e2<-(e1[2]-e1[1])/10
833 f1<-round(c(min(ST[,17]), max(ST[,17])),4); f2<-(f1[2]-f1[1])/10
834 g1<-round(c(min(ST[,18]), max(ST[,18])),4); g2<-(g1[2]-g1[1])/10
835 h1<-round(c(min(ST[,19]), max(ST[,19])),4); h2<-(h1[2]-h1[1])/10
836 i1<-round(c(min(ST[,20]), max(ST[,20])),4); i2<-(i1[2]-i1[1])/10
837 j1<-round(c(min(ST[,21]), max(ST[,21])),4); j2<-(j1[2]-j1[1])/10
838
839
840 #alpha1
841 postscript("plotAlpha1RMSE.eps", width=5.75, height=5.75, horizontal=FALSE, onefile=TRUE)
842 plot(c(50,1000),a1,type="n",xlab="Tama o de muestra (n)", ylab="RMSE", xaxt="n", yaxt="n")
843 axis(1,seq(50,1000,50))
844 axis(2,seq(a1[1],a1[2],a2))
845 s1=smooth.spline(n,ST[,12] , spar=0.25)
846 lines(s1,type="o",pch=15,col="black",lwd=1, lty=1)
847 title(expression(paste(alpha[1]))) #Titulo do grafico
848 abline(h=0,lty=3) #Linha horizontal
849 dev.off() #Fechando o dispositivo potscript
850
851 #alpha2
852 postscript("plotAlpha2RMSE.eps", width=5.75, height=5.75, horizontal=FALSE, onefile=TRUE)
853 plot(c(50,1000),b1,type="n",xlab="Tama o de muestra (n)", ylab="RMSE", xaxt="n", yaxt="n")
854 axis(1,seq(50,1000,50))
855 axis(2,seq(b1[1],b1[2],b2))
856 s2=smooth.spline(n,ST[,13] , spar=0.25)
857 lines(s2,type="o",pch=15,col="black",lwd=1, lty=1)
858 title(expression(paste(alpha[2]))) #Titulo do grafico

```

```

859 abline(h=0,lty=3) #Linha horizontal
860 dev.off() #Fechando o dispositivo potscript
861
862 #beta1
863 postscript("plotBeta1RMSE.eps", width=5.75, height=5.75, horizontal=FALSE, onefile=TRUE)
864 plot(c(50,1000),c1,type="n",xlab="Tama o de muestra (n)", ylab="RMSE", xaxt="n", yaxt="n")
865 axis(1,seq(50,1000,50))
866 axis(2,seq(c1[1],c1[2],c2))
867 s3=smooth.spline(n,ST[,14] , spar=0.25)
868 lines(s3,type="o",pch=15,col="black",lwd=1,lty=1)
869 title(expression(paste(beta[1]))) #Titulo do grafico
870 abline(h=0,lty=3) #Linha horizontal
871 dev.off() #Fechando o dispositivo potscript
872
873 #beta2
874 postscript("plotBeta2RMSE.eps", width=5.75, height=5.75, horizontal=FALSE, onefile=TRUE)
875 plot(c(50,1000),d1,type="n",xlab="Tama o de muestra (n)", ylab="RMSE", xaxt="n", yaxt="n")
876 axis(1,seq(50,1000,50))
877 axis(2,seq(d1[1],d1[2],d2))
878 s4=smooth.spline(n,ST[,15] , spar=0.25)
879 lines(s4,type="o",pch=15,col="black",lwd=1,lty=1)
880 title(expression(paste(beta[2]))) #Titulo do grafico
881 abline(h=0,lty=3) #Linha horizontal
882 dev.off() #Fechando o dispositivo potscript
883
884 #beta3
885 postscript("plotBeta3RMSE.eps", width=5.75, height=5.75, horizontal=FALSE, onefile=TRUE)
886 plot(c(50,1000),e1,type="n",xlab="Tama o de muestra (n)", ylab="RMSE", xaxt="n", yaxt="n")
887 axis(1,seq(50,1000,50))
888 axis(2,seq(e1[1],e1[2],e2))
889 s5=smooth.spline(n,ST[,16] , spar=0.25)
890 lines(s5,type="o",pch=15,col="black",lwd=1,lty=1)
891 title(expression(paste(beta[3]))) #Titulo do grafico
892 abline(h=0,lty=3) #Linha horizontal
893 dev.off() #Fechando o dispositivo potscript
894
895 #beta4
896 postscript("plotBeta4RMSE.eps", width=5.75, height=5.75, horizontal=FALSE, onefile=TRUE)
897 plot(c(50,1000),f1,type="n",xlab="Tama o de muestra (n)", ylab="RMSE", xaxt="n", yaxt="n")
898 axis(1,seq(50,1000,50))
899 axis(2,seq(f1[1],f1[2],f2))
900 s6=smooth.spline(n,ST[,17] , spar=0.25)
901 lines(s6,type="o",pch=15,col="black",lwd=1,lty=1)
902 title(expression(paste(beta[4]))) #Titulo do grafico
903 abline(h=0,lty=3) #Linha horizontal
904 dev.off() #Fechando o dispositivo potscript
905
906 #muji
907 postscript("plotMuj1RMSE.eps", width=5.75, height=5.75, horizontal=FALSE, onefile=TRUE)
908 plot(c(50,1000),g1,type="n",xlab="Tama o de muestra (n)", ylab="RMSE", xaxt="n", yaxt="n")
909 axis(1,seq(50,1000,50))
910 axis(2,seq(g1[1],g1[2],g2))
911 s7=smooth.spline(n,ST[,18] , spar=0.25)
912 lines(s7,type="o",pch=15,col="black",lwd=1,lty=1)
913 title(expression(paste(mu[1]))) #Titulo do grafico
914 abline(h=0,lty=3) #Linha horizontal
915 dev.off() #Fechando o dispositivo potscript
916
917 #muj2
918 postscript("plotMuj2RMSE.eps", width=5.75, height=5.75, horizontal=FALSE, onefile=TRUE)
919 plot(c(50,1000),h1,type="n",xlab="Tama o de muestra (n)", ylab="RMSE", xaxt="n", yaxt="n")
920 axis(1,seq(50,1000,50))
921 axis(2,seq(h1[1],h1[2],h2))
922 s8=smooth.spline(n,ST[,19] , spar=0.25)
923 lines(s8,type="o",pch=15,col="black",lwd=1,lty=1)
924 title(expression(paste(mu[2]))) #Titulo do grafico
925 abline(h=0,lty=3) #Linha horizontal
926 dev.off() #Fechando o dispositivo potscript
927
928 #nu
929 postscript("plotNuRMSE.eps", width=5.75, height=5.75, horizontal=FALSE, onefile=TRUE)
930 plot(c(50,1000),i1,type="n",xlab="Tama o de muestra (n)", ylab="RMSE", xaxt="n", yaxt="n")
931 axis(1,seq(50,1000,50))
932 axis(2,seq(i1[1],i1[2],i2))
933 s9=smooth.spline(n,ST[,20] , spar=0.25)
934 lines(s9,type="o",pch=15,col="black",lwd=1,lty=1)
935 title(expression(paste(nu))) #Titulo do grafico
936 abline(h=0,lty=3) #Linha horizontal
937 dev.off() #Fechando o dispositivo potscript
938
939 #p
940 postscript("plotPiRMSE.eps", width=5.75, height=5.75, horizontal=FALSE, onefile=TRUE)

```

```
941 plot(c(50,1000),j1,type="n",xlab="Tama o de muestra (n)", ylab="RMSE", xaxt="n", yaxt="n")
942 axis(1,seq(50,1000,50))
943 axis(2,seq(j1[1],j1[2],j2))
944 s10=smooth.spline(n,ST[,21] , spar=0.25)
945 lines(s10,type="o",pch=15,col="black",lwd=1,lty=1)
946 title(expression(paste(p))) #Titulo do grafico
947 abline(h=0,lty=3) #Linha horizontal
948 dev.off() #Fechando o dispositivo potscript
949
950 #####
```



Bibliografía

- Akaike, H. (1974). A new look at the statistical model identification, *IEEE Transactions on Automatic Control* **19**: 716 – 723.
- Balakrishnan, N., Leiva, V., Sanhueza, A. y Vilca, F. (2009). Estimation in the Birnbaum-Saunders distribution based on scale-mixture of normals and the EM-algorithm, *Statistics and Operations Research Transactions* **33**: 171–192.
- Bartolucci, F. y Scaccia, L. (2005). The use of mixtures for dealing with non-normal regression errors, *Computational Statistics & Data Analysis* **48**(4): 821–834.
- Basford, K. E., Greenway, D., McLachlan, G. J. y Peel, D. (1997). Standard errors of fitted component means of normal mixtures, *Computational Statistics* **12**: 1–18.
- Benites, L., Lachos, V. H. y Maehara, R. (2019). Linear regression models with finite mixtures of skew heavy-tailed errors, *Chilean Journal of Statistics* **10**: 21–41.
- Bhansali, R. y D. D. (1977). Some properties of the order of an autoregressive model selected by a generalization of akaike’s epf criterion, *Biometrika* **64**(3): 547–551.
- Birnbaum, Z. y Saunders, S. (1969a). A new family of life distributions, *Journal of Applied Probability* **6**: 319–327.
- Birnbaum, Z. y Saunders, S. (1969b). Estimation for a family of life distributions with applications to fatigue, *John Wiley and Sons* **6**: 328–347.
- Bozdogan, H. (1987). Model selection and akaike’s information criterion (aic): the general theory and its analytical extensions, *Psychometrika* **52**(3): 345–370.
- Burnham, K. P. y Anderson D. R. (2002). *Model selection and multimodel inference: a practical information—theoretic approach*, Springer, Berlin.
- Cavanaugh, J. (2004). Criteria for linear model selection based on kullback’s symmetric divergence, *Australian & New Zealand Journal of Statistics* **46**(2): 257–274.
- Dempster, A., Laird, N. M. y Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion), *Journal of the Royal Statistical Society Series B* **39**: 1–38.
- Efron, B. y Tibshirani, R. (1986). Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy, *Statistical Science* pp. 54–75.
- Farias, R. y Lemonte, A. J. (2011). Bayesian inference for the Birnbaum–Saunders nonlinear regression model, *Statistical Methods & Applications* **20**: 423–438.
- Frühwirth-Schnatter, S., Celeux, G. y Robert, C. P. (2019). *Handbook of Mixture Analysis*, CRC Press, Boca Raton, FL.
- Hannan, E.J. (1980). The Estimation of the Order of an ARMA Process.

- Hossein, H., K., M. y R., M. (2020). A new five-parameter Birnbaum–Saunders distribution for modeling bicoid gene expression data, *Mathematical Biosciences* **319**: 108275.
- Hurvich, C.M. y Tsai C.L. (1989). Regression and time series model selection in small samples, *Biometrika* **76(5)**: 297.
- Johnson, N. L. (1949). Systems of frequency curves generated by methods of translation, *Biometrika* **36**: 146–176.
- Johnson, N. L. y Kotz, S. (1970). *Distributions in Statistics Continuous Univariate Distributions, Vol. 1*, John Wiley & Sons, New York.
- Leiva, V. (2016). *The Birnbaum-Saunders Distribution*, Elsevier Ltd., London, UK.
- Leiva, V., Vilca, F., Balakrishnan, N. y Sanhueza, A. (2010). A skewed sinh-normal distribution and its properties and application to air pollution, *Communications in Statistics - Theory and Methods* **39**: 426–443.
- Lemonte, A. J. y Cordeiro, G. M. (2009). Birnbaum-Saunders nonlinear regression models, *Computational Statistics & Data Analysis* **53(12)**: 4441–4452.
- Maehara, R. y Benites, L. (2015). *bssn: Birnbaum-Saunders Model Based on Skew-Normal Distribution*. R package version 0.6.
URL: <http://CRAN.R-project.org/package=bssn>
- Maehara Rocío, Bolfarine Heleno, F. V. y. N. B. (2021). A robust Birnbaum–Saunders regression model based on asymmetric heavy-tailed distributions, *Metrika* **na**: na.
- Mehrdad, N., Hashemi, F., Bekker, A. y Jamalizadeh, A. (2020). Modeling right-skewed financial data streams: A likelihood inference based on the generalized Birnbaum–Saunders mixture model, *Applied Mathematics and Computation* **376**: 125109.
- Mehrdad, N., Hung, W.-L., Lin, T.-I. y Jamalizadeh, A. (2019). A novel mixture model using the multivariate normal mean–variance mixture of Birnbaum–Saunders distributions and its application to extrasolar planets, *Journal of Multivariate Analysis* **171**: 126–138.
- Mohammadi, K., AlavibJon, O. y McGowana, G. (2017). Use of Birnbaum-Saunders distribution for estimating wind speed and wind power probability distributions: A review, *Energy Conversion and Management* **143**: 109–122.
- Ng, H., Kundu, D. y Balakrishnan, N. (2003). Modified moment estimation for the two-parameter Birnbaum-Saunders distribution, *Computational Statistics and Data Analysis* **43**: 283–298.
- Rieck, J. R. (1989). *Statistical analysis for the Birnbaum-Saunders fatigue life distribution*, PhD thesis, Clemson University.
- Rieck, J. R. y Nedelman, J. R. (1991). A log-linear model for the Birnbaum-Saunders distribution, *Technometrics* **33**: 51–60.
- Ritz, C. y Streibig, J. (2008). *Nonlinear Regression with R*, Springer, New York.
- Santana, L., Vilca, F. y Leiva, V. (2011). Influence analysis in skew-Birnbaum–Saunders regression models and applications, *Journal of Applied Statistics* **38**: 1633–1649.
- Schwarz, G. E. (1978). Estimating the dimension of a model, *Annals of Statistics* **6**: 461–464.

- Sclove, S. (1987). Application of model-selection criteria to some problems in multivariate analysis, *Psychometrika* **52**(3): 333–343.
- Vilca, F., Zeller, C. B. y Cordeiro, G. M. (2015). The sinh-normal/independent nonlinear regression model, *Journal of Applied Statistics* **42**: 1659–1676.
- Zeller, C. B., Cabral, C. R., Lachos, V. H. y Benites, L. (2019). Finite mixture of regression models for censored data based on scale mixtures of normal distributions, *Advances in Data Analysis and Classification* **13**(1): 89–116.
- Zhao, L., Dorca, C. y Gonçalves, C. (2001). On determination of the order of a markov chain, *Statistical Inference for Stochastic Processes* **4**(3): 273–282.

