PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

ESCUELA DE POSGRADO

INFERENCIA BAYESIANA EN EL MODELO DE REGRESIÓN SPLINE PENALIZADO CON UNA APLICACIÓN A LOS TIEMPOS EN COLA DE UNA AGENCIA BANCARIA

Tesis para optar el grado de Magíster en Estadística

AUTOR

Diego Eduardo Huaraz Zuloaga

ASESOR

Dr. Cristian Bayes

JURADO

Dr. Luis Valdivieso
Dr. Jorge Bazán
Dr. Cristian Bayes

LIMA - PERÚ

2012

Dedicatoria

Dedico el presente trabajo de tesis a mi querida familia por haberme brindado amor y apoyo constante. A mi madre, por cuidarme y guiarme desde el cielo. A mi padre, por ser un gran amigo y consejero. A mis hermanos, por ser grandes ejemplos en la vida personal y profesional. A Lisset, por ser una gran compañera y estar a mi lado cuando siempre lo he necesitado. Es un lujo contar con todos ustedes.

Agradecimentos

Agradezco de manera especial al Dr. Cristian Bayes, mi asesor del presente tema de tesis, por su apoyo continuo durante el desarrollo de la investigación. De igual manera agradezco a los profesores de la maestría por haberme apoyado en las distintas etapas del presente trabajo.

Resumen

En diversos campos de aplicación se requiere utilizar modelos de regresión para analizar la relación entre dos variables. Cuando esta relación es compleja, es difícil modelar los datos usando técnicas paramétricas tradicionales, por lo que estos casos requieren de la flexibilidad de los modelos no paramétricos para ajustar los datos. Entre los diferentes modelos no paramétricos está la regresión spline penalizada, que puede ser formulada dentro de un marco de modelos lineales mixtos. De este modo, los programas computacionales desarrollados originalmente para la inferencia clásica y Bayesiana de modelos mixtos pueden ser utilizados para estimarlo. La presente tesis se centra en el estudio de la inferencia Bayesiana en el modelo de regresión spline penalizado. Para lograr esto, este trabajo proporciona un marco teórico breve de este modelo semiparamétrico y su relación con el modelo lineal mixto, la inferencia Bayesiana de este modelo, y un estudio de simulación donde se comparan la inferencia clásica y Bayesiana en diferentes escenarios considerando diversos valores del número de nodos, tamaños de muestra y niveles de dispersión en la data simulada. Finalmente, en base a los resultados del estudio de simulación, el modelo se aplica para estimar el tiempo de espera en cola de los clientes en agencias bancarias con el fin de calcular la capacidad de personal óptima bajo determinadas metas de nivel de servicio.

Palabras-clave: Regresión spline penalizada, modelo lineal mixto, inferencia Bayesiana, WinBUGS, colas bancarias.

Abstract

In several application fields, regression models are required to analyze the relationship between two variables. When this relationship is complex, it is difficult to model the data set by using traditional parametric techniques, so these cases require the flexibility of nonparametric models to fit the data. Among different nonparametric models there is the penalized spline regression, which can be formulated within a mixed linear model framework. Thus, computer programs originally developed for classical and Bayesian inference of mixed models can be used to estimate it. The current thesis focuses on the study of Bayesian inference in the penalized spline regression model. To accomplish it, this work provides a brief theoretical framework of this semiparametric model and its relation with linear mixed models, the Bayesian inference of this model, and a simulation study where classical and Bayesian inference are compared on different scenarios considering different values for the number of knots, sample sizes and dispersion levels in the simulated data set. Finally, based on the results of the simulation study, the model is applied to estimate the queue waiting time for customers in bank branches in order to calculate the optimal staffing capacity under certain service levels goals.

Keywords: Penalized spline regression, linear mixed models, Bayesian inference, WinBUGS, bank queues.

Índice general

Lista de Abreviaturas

MCMC	Markov Chain Monte Carlo.
ECM	Error cuadrático medio.
AIC	Criterio de información de Akaike.
BIC	Criterio de información Bayesiano o de Schwartz.
EAIC	Esperado del criterio de información de Akaike.
EBIC	Esperado del criterio de información Bayesiano o de Schwartz.
DIC	Criterio de información de Desvío.
MLM	Modelo Lineal Mixto.
IG	Gamma inversa.
BLP	Mejor Predictor Lineal.
BLUP	Mejor Predictor Lineal Insesgado.
BUGS	Bayesian using Gibbs Sampling.

Lista de Símbolos

σ^2 Varianza.

λ Parámetro de suavización.

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

1.1. Consideraciones preliminares

El análisis de la relación entre dos variables se puede representar mediante modelos de regresión paramétricos y no paramétricos. Generalmente, cuando la forma de la relación entre las variables no es compleja, es posible implementar el ajuste mediante una forma paramétrica; en cambio, si la relación entre las variables es compleja, el modelo paramétrico puede ser insuficiente, con lo que es necesario modelar de forma no paramétrica. La principal ventaja de los modelos no paramétricos es la flexibilidad que poseen para el ajuste de los datos, lo cual se debe a que la relación entre las variables es determinada por los datos, mientras que un marco parámetrico la relación es definida por el modelo considerado.

Actualmente existen diversos modelos de regresión semiparamétricos y no paramétricos que permiten modelar dos variables que presenten una relación compleja, entre los cuales están los métodos kernel, regresiones spline, regresiones spline penalizadas, entre otros más. En la presente tesis se estudiará el modelo de regresión empleando splines penalizados, siguiendo la literatura descrita en Ruppert et al. (2003).

Los modelos de regresión por splines permiten realizar un ajuste de los datos mediante pedazos de curvas polinomiales que son unidos por sus puntos extremos, a los cuales se les llaman nodos. Esto permite obtener ajustes adecuados sobre la data que se esté analizando; sin embargo, presentan a la vez el inconveniente de ser vulnerables a la cantidad de nodos que se empleen y a la ubicación de los mismos, con lo cual es posible incurrir incluso en sobre ajustes en el modelo. Una manera de superar este problema es limitar la influencia de los posibles nodos mediante una penalización, con lo cual se esperaría conseguir un ajuste más adecuado sin caer en sobreajustes; esto da origen al modelo de regresión spline penalizado.

En Ruppert et al. (2003) se puede revisar que el modelo de regresión spline penalizado puede ser formulado dentro de un marco de modelos lineales mixtos. Esta relación permite poder emplear los programas computacionales donde se han implementado los modelos mixtos. De este modo, los programas desarrollados para la inferencia bayesiana de modelos mixtos pueden ser utilizados para el modelo de regresión spline penalizado. En Crainiceanu et al. (2005) se comenta que el modelamiento bayesiano semiparamétrico es atractivo debido a que disfruta de la flexibilidad que tienen los modelos no parámetricos para el ajuste de los datos y que, además, posee la inferencia exacta proporcionada por todos los mecanismos desarrollados para la inferencia bayesiana.

1.2. Objetivos

El objetivo general de la tesis es estudiar el modelo de regresión spline penalizado desde un enfoque bayesiano y aplicarlo a un conjunto de datos reales. De manera específica se tienen los siguientes objetivos:

- Revisar la literatura sobre el modelo de regresión por splines y el modelo spline penalizado.
- Estudiar la relación entre el modelo de regresión spline penalizado y el modelo lineal mixto.
- Estudiar y comparar métodos de estimación del modelo de regresión spline penalizado desde un enfoque clásico y Bayesiano, considerando en este último simulación MCMC.
- Realizar un estudio de simulación acerca del modelo de regresión spline penalizado en diferentes escenarios considerando diversos valores de número de nodos, tamaños de muestra y niveles de dispersión en la data simulada.
- Aplicar el modelo a un conjunto de datos reales de atención en una agencia bancaria a fin de estimar los tiempos en cola de los clientes, lo cual permitirá calcular el personal necesario para satisfacer diversas metas de espera en cola.

1.3. Organización del trabajo

En el Capítulo 2 se revisan los modelos de regresión por splines, el modelo spline penalizado y la conexión que existe entre este modelo semiparamétrico con el modelo lineal mixto. Se considera como guía a Ruppert et al. (2003) y un conjunto de datos en particular para ilustrar mejor los conceptos. En el Capítulo 3 se revisa la estimación bayesiana del modelo de regresión spline penalizado a través de un marco de modelos mixtos, donde se presentan las distribuciones condicionales completas requeridas para simulación MCMC mediante el algoritmo del muestreo de Gibbs. La implementación de la inferencia bayesiana se realiza en el Capítulo 4. En dicho capítulo se evalúa el ajuste de los modelos a través del error cuadrático medio a una curva simulada, tanto desde una perspectiva clásica (máxima verosimilitud) como bayesiana (con un código propio en R y un código BUGS). Para esto se consideran diversos escenarios en función a tres valores de varianzas de los datos simulados, cuatro cantidades de nodos diferentes y distintos tamaños de muestra. En base a los resultados de las simulaciones se identificarán algunos hallazgos importantes. En el Capítulo 5 se presenta una aplicación del modelo de regresión spline penalizado para estimar los tiempos de espera en cola de una agencia bancaria a partir del nivel de arribos, tiempos de servicio y cantidad de servidores utilizados. En base a esta estimación se implementará el modelo para calcular el personal necesario para satisfacer diversas metas de espera en cola. En el Capítulo 6 se comentan las principales conclusiones en base a los resultados del estudio de simulación y de la aplicación, así como algunas sugerencias para investigaciones futuras.

En el apéndice A se presenta la estimación clásica de los parámetros del modelo lineal mixto. El apéndice B presenta tablas de resultados y gráficos. Los apéndices C, D, E, F y G presentan los códigos implementados en el estudio de simulación y en la aplicación.

Capítulo 2

Regresión Spline Penalizada

2.1. Introducción

Existen diversas formas de ajustar los puntos de un gráfico de dispersión, las cuales van desde modelos paramétricos hasta modelos no paramétricos. Generalmente, cuando la forma de la gráfica de dispersión es sencilla, esta se puede modelar mediante la forma paramétrica; en cambio, si la gráfica resulta ser más compleja, es posible requerir un modelamiento de forma no parámetrica. La principal ventaja de los modelos no paramétricos sobre los modelos paramétricos es su flexibilidad para ajustar los datos. En un marco no paramétrico la relación entre covariables y la variable dependiente es determinada por la data, mientras que en un marco paramétrico la forma está determinada por el modelo en sí.

A lo largo de este capítulo se considerará la data LIDAR empleada por Ruppert et al. (2003), mostrada en la figura 2.1. Las siglas LIDAR corresponden a la técnica *light detection and ranging*, la cual se caracteriza por utilizar el reflejo de la luz emitida por un láser para detectar compuestos químicos en la atmósfera. Las variables de esta data son las siguientes:

- *range*: distancia recorrida antes que la luz vuelva a reflejarse en su propia fuente (predictiva).
- *logratio*: logaritmo de la luz recibida a partir de dos fuentes de láser (respuesta).

El gráfico de dispersión de la data LIDAR muestra que no existe un comportamiento lineal en los datos y que la varianza no es homocedástica. Esto último se debe a que a mayores valores de *range* la dispersión de los valores *logratio* aumenta.

Se puede ver un ajuste a los datos mediante diversos modelos polinómicos (paramétricos) en la figura 2.2, pudiendo concluir que a mayor grado del polinomio el ajuste mejora; sin embargo, tener un modelo polinómico de grado elevado puede aumentar la complejidad del modelo.

En casos como este, es probable que la necesidad de tener un modelo que brinde un mejor ajuste recaiga sobre un modelo de forma no paramétrica. Por este motivo, en el presente capítulo se revisará el modelo de regresión spline penalizado, que tiene la ventaja de ser una extensión relativamente sencilla de los modelos de regresión lineal. Una ventaja adicional de este método es que puede ser estimado mediante la forma de un modelo lineal mixto, lo cual se revisará en la sección 2.4.

Los gráficos del presente capítulo se han implementado en el programa R.

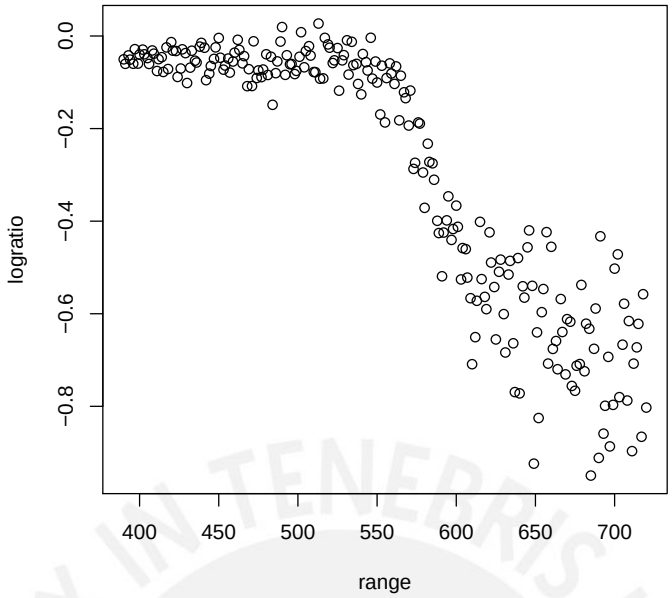

Figura 2.1: Gráfico de dispersión de la data LIDAR

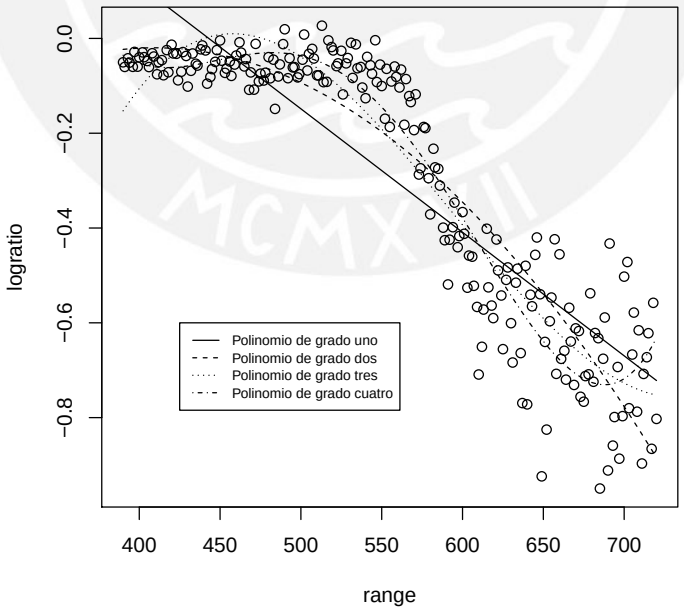

Figura 2.2: Ajuste de la data LIDAR mediante expresiones polinómicas

2.2. Regresión spline

Según Hastie y Tibshirani (1990), los modelos de regresión por splines permiten realizar un ajuste de los datos mediante pedazos de curvas polinomiales que son unidos por sus puntos extremos, a los cuales se les llaman nodos $\kappa_1, \kappa_2, \ldots, \kappa_K$. Esto permite obtener ajustes adecuados sobre la data que se esté analizando; sin embargo, presentan a la vez el inconveniente de ser vulnerables a la cantidad de nodos que se empleen y a la ubicación de los mismos, con lo cual es posible incurrir incluso en sobre ajustes en el modelo. Se acostumbra que estas curvas, unidas entre los nodos, sean suavizadas. Este modelo semiparamétrico tiene la forma:

$$y_i = f(x_i) + \varepsilon_i$$

donde f es una función suavizada estimada de la data y ε_i es un error aleatorio con $E(\varepsilon_i) = 0$.

A manera de ilustración, se revisará el ajuste de la data LIDAR (Ruppert et al., 2003) mediante un modelo de regresión lineal que considera una función spline de base lineal. Este modelo spline tiene la siguiente forma:

$$f(x) = \beta_0 + \beta_1 x + \sum_{k=1}^{K} b_k (x - \kappa_k)_+ \tag{2.1}$$

donde

$$(x - \kappa_k)_+ = \begin{cases} 0, & x \leq \kappa_k \\ x - \kappa_k, & x > \kappa_k \end{cases}$$

La elección de los nodos que delimitarán los splines de base lineal suele ser todo un reto. Una manera de elegirlos puede ser en base a prueba y error. En este caso se empezará considerando los nodos situados en los lugares del *range*=575 y *range*=600, que marcan el cambio de tendencia en los datos, tal como se muestra en la figura 2.3.

El ajuste del modelo puede mejorar al incorporar mayor cantidad de nodos. Por ejemplo, la figura 2.4 muestra el ajuste considerando cuatro nodos para la variable *range*, que son: 500, 550, 600 y 650. Incluso se puede llegar a un sobre ajuste considerando nodos que van desde el *range* = 400 hasta el 700, a cada 12.5 unidades, tal como se muestra en la figura 2.5.

Una forma de disminuir el sobre ajuste de la figura 2.5 es retirando algunos nodos. Por ejemplo, retirando los nodos en *range* = 612.5, 650, 662.5 y 687.5 se obtiene la figura 2.6. Este último ajuste es más agradable ya que se ajusta bien a los datos sin llegar a un sobre ajuste.

Según lo revisado, la obtención del modelo dependerá de la cantidad de nodos con los que se quiera trabajar. Si existen $\boldsymbol{K}$ nodos candidatos entonces van haber $2^{\boldsymbol{K}}$ modelos posibles, por ejemplo, de tener 2 nodos candidatos se podría unir la data considerando cada nodo por separado, los dos nodos juntos o ninguno de los dos. Por este motivo, la elección de los nodos puede llegar ser todo un problema para el modelo de regresión spline. Una manera de superar este problema es limitar la influencia de los posibles nodos mediante una penalización, con lo cual se esperaría conseguir un ajuste menos variable. En la sección 2.3 se revisará esta penalización, que da origen al modelo de regresión spline penalizado.

Por el momento se ha revisado el empleo del spline lineal para ejemplificar el modelo de regresión spline, sin embargo, existen diversos tipos de modelos spline y bases que se pueden utilizar, las cuales se pueden profundizar en Hastie y Tibshirani (1990) y Ruppert et al. (2003). A manera de resumen, se revisarán algunos modelos y bases spline.

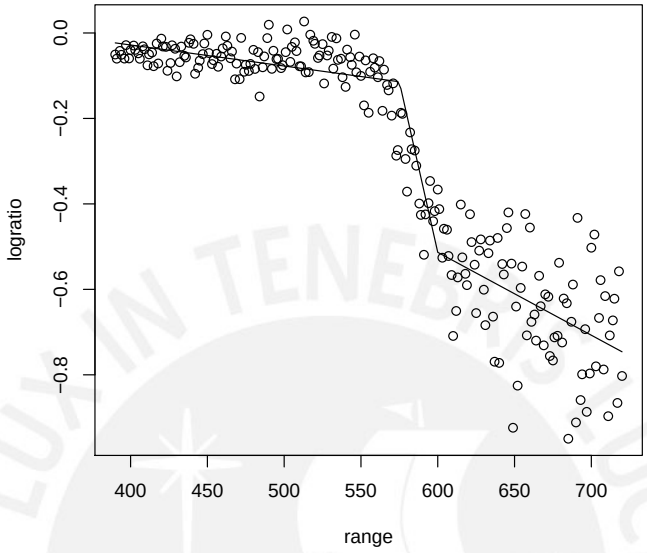

Figura 2.3: Ajuste de la data LIDAR mediante una regresión spline lineal con nodos en *range*=575 y 600

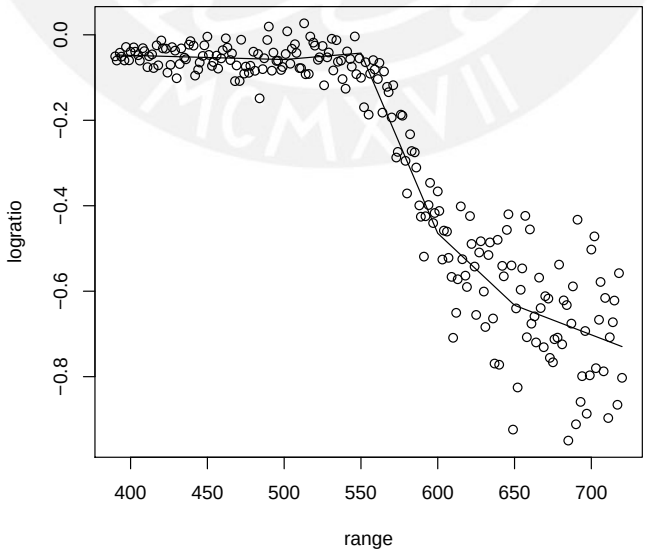

Figura 2.4: Ajuste de la data LIDAR mediante una regresión spline lineal con los nodos en *range* = 500, 550, 600 y 650.

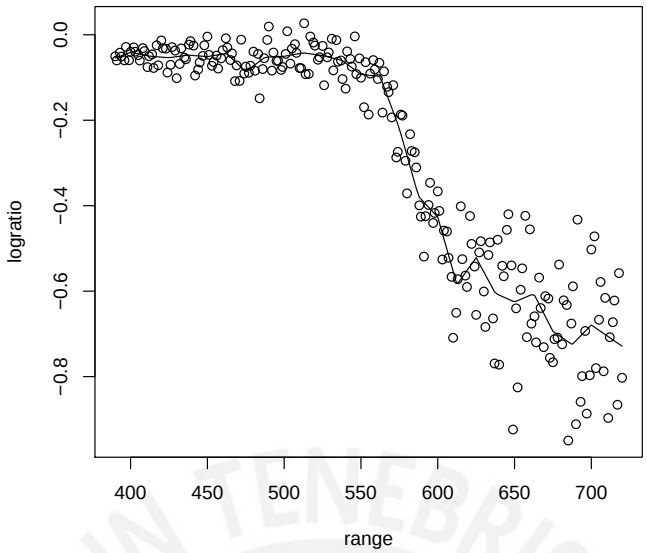

Figura 2.5: Ajuste de la data LIDAR mediante una regresión spline lineal con los nodos en *range* = 400, 412.5, 425, ..., 700.

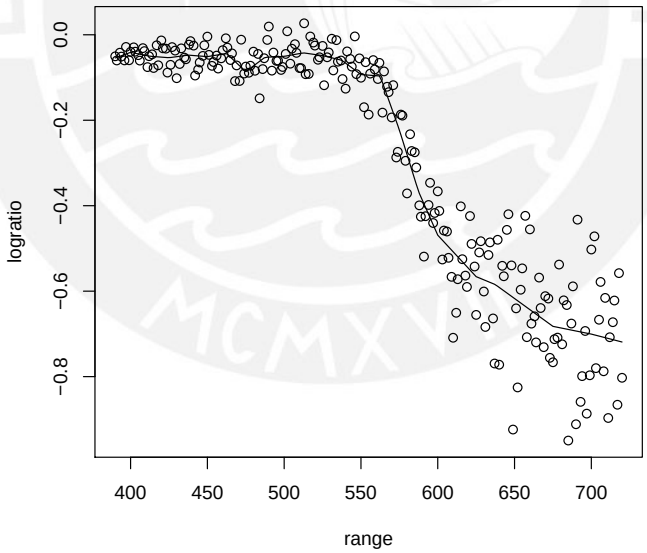

Figura 2.6: Ajuste de la data LIDAR mediante una regresión spline lineal retirando a los nodos en *range* = 612.5, 650, 662.5 y 687.5 del modelo empleado en la Figura 2.5.

2.2.1. Modelos y bases spline

Según Ruppert et al. (2003), un cambio en los modelos y bases spline no cambia el ajuste, aunque hay algunos que son numéricamente más estables y permiten el cálculo de un ajuste con mayor precisión. Aparte de la estabilidad numérica, las razones para seleccionar un

modelo sobre otro son la implementación y la interpretación. Esta última razón no suele ser tan importante ya que generalmente el interés del modelo es conseguir el ajuste de la curva y no la estimación de los valores de los parámetros. A continuación se revisarán algunos tipos de splines.

Spline penalizado

La regresión por splines penalizados permite penalizar a los coeficientes aleatorios b_k del k-ésimo nodo de la ecuación (2.1) a fin de encontrar un modelo adecuado.
Este modelo spline penalizado se puede representar dentro de un marco de modelos mixtos, esta conexión permite que los programas computacionales desarrollados originalmente para modelos mixtos puedan emplearse para el modelo de regresión spline penalizado. En la sección 2.3 se revisará con más detalle este modelo.

Spline polinómico de base truncada

Esta es una generalización de un modelo spline para algún grado p en general. Utiliza la función de potencias truncada, teniendo a la base como:

$$1, x, \ldots, x^p, (x-\kappa_1)_+^p, \ldots, (x-\kappa_K)_+^p, \tag{2.2}$$

que se conoce como la base de potencias truncada de grado p. Dado que la función $(x-\kappa)_+^p$ tiene $p-1$ derivadas continuas, altos valores de p conducen a funciones splines más suaves. El modelo spline de grado p es de la forma:

$$f(x) = \beta_0 + \beta_1 x + \ldots + \beta_p x^p + \sum_{k=1}^{K} b_k (x-\kappa_k)_+^p. \tag{2.3}$$

Spline de base radial

Este modelo spline considera la siguientes funciones como base:

$$1, x, \ldots, x^p, |x-\kappa_1|^p, \ldots, |x-\kappa_K|^p. \tag{2.4}$$

Este modelo spline puede ser escrito como:

$$f(x) = \beta_0 + \beta_1 x + \ldots + \beta_{p-1} x^{p-1} + \sum_{k=1}^{K} b_k |x-\kappa_k|^{2p-1}. \tag{2.5}$$

Por ejemplo, según Ruppert et al. (2003), el modelo spline de base radial cúbica quedaría definido de la siguiente manera:

$$f(x) = \beta_0 + \beta_1 x + \sum_{k=1}^{K} b_k |x-\kappa_k|^3. \tag{2.6}$$

Este modelo spline descrito en la ecuación (2.6) tiende a tener propiedades numéricas muy buenas. En particular, la autocorrelación de los parámetros a posteriori es mucho menor en comparación con otras bases (ver Crainiceanu et al., 2005). Por este motivo, este spline será el que se considerará en el presente proyecto de tesis.

2.3. Regresión spline penalizada

En esta sección se presenta la obtención del modelo de regresión spline penalizado. Para esto, se partirá del modelo spline con K nodos planteado en la ecuación (2.6), el cual será expresado de la siguiente forma:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta}$$

donde $\mathbf{X}$ es la matriz variables explicativas y $\boldsymbol{\beta}$ es la matriz de parámetros del modelo. Para el desarrollo del presente marco teórico se considerará la matriz $\mathbf{X}$ con sólo una variable explicativa.

El ajuste conocido por mínimos cuadrados puede ser escrito como:

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}, \quad \text{donde } \hat{\boldsymbol{\beta}} \text{ minimiza } \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$$

y $\boldsymbol{\beta} = [\beta_0, \beta_1, b_1, b_2, \ldots, b_K]^{\mathbf{T}}$, siendo β_0, β_1 los coeficientes asociados a las variables y b_k el coeficiente asociado al k-ésimo nodo. Como se revisó antes, la presencia de varios de estos b_k permiten un sobre ajuste del modelo. Por este motivo, es posible rectificar esta situación con las siguientes restricciones:

(1) $\max |b_k| < C$

(2) $\sum_{k=1}^{K} |b_k| < C$, y

(3) $\sum_{k=1}^{K} b_k^2 < C$.

donde C es una constante positiva. Para la elección juiciosa de C, cada uno de estos casos dará lugar a un ajuste más suave a la dispersión. Ruppert et al. (2003) plantean el caso de la tercera restricción debido a que su implementación es más sencilla respecto a las otras dos, por este motivo será la que se revisará en la presente sección.
Se define una matriz $(K+2) \times (K+2)$, denominada $\mathbf{D}$, con la siguiente forma:

$$\mathbf{D} = \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times K} \\ \mathbf{0}_{K\times 2} & \mathbf{I}_{K\times K} \end{bmatrix}$$

Luego, el problema de minimización de cuadrados puede ser escribirse como:

$$\text{minimizar } \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 \quad \text{sujeto a } \boldsymbol{\beta}^{\boldsymbol{T}}\mathbf{D}\boldsymbol{\beta} \leq \boldsymbol{C}.$$

Según Ruppert et al. (2003), utilizando multiplicadores de Lagrange se puede demostrar que la expresión anterior es equivalente a elegir $\boldsymbol{\beta}$ para minimizar:

$$\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda^2 \boldsymbol{\beta}^{\boldsymbol{T}}\mathbf{D}\boldsymbol{\beta} \tag{2.7}$$

para algún valor de $\lambda \geq 0$.

La solución de la expresión anterior presentada en Ruppert et al. (2003) está dada por:

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}} = (\mathbf{X}^{\mathbf{T}}\mathbf{X} + \lambda^2\mathbf{D})^{-1}\mathbf{X}^{\mathbf{T}}\mathbf{y}.$$

El término $\lambda^2\boldsymbol{\beta}^{\boldsymbol{T}}\mathbf{D}\boldsymbol{\beta}$ se conoce como penalidad de rugosidad (*roughness penalty* en inglés) porque penaliza los ajustes que son muy flexibles, obteniéndose mejores resultados. La suavización del ajuste es controlada por λ, que normalmente se conoce como parámetro de suavización.

Los valores ajustados para una regresión spline penalizada general están dados por:

$$\hat{\mathbf{y}} = \mathbf{X}(\mathbf{X}^{\mathbf{T}}\mathbf{X} + \lambda^2\mathbf{D})^{-1}\mathbf{X}^{\mathbf{T}}\mathbf{y}. \tag{2.8}$$

Como resultado tenemos que el modelo spline penalizado depende del conjunto de nodos y del parámetro de suavización.

La figura 2.7 muestra el ajuste de la data LIDAR con la secuencia de nodos empleada en la figura 2.5 ($K = 25$) para diversos valores de λ . El caso de $\lambda = 0$ corresponde al modelo sin penalización, por lo tanto coincide con el de la figura 2.5. Para $\lambda = 10$ se disminuye la rugosidad del modelo. Incrementando al triple el valor de λ se obtiene un ajuste más adecuado. Asimismo, se puede ver que al considerar valores de λ muy elevados ($\lambda = 1000$ por ejemplo) el modelo se acerca más a la forma de una recta.

2.4. Regresión spline penalizada formulada como un modelo lineal mixto

Antes de revisar la conexión entre ambos modelos es necesario revisar brevemente la forma del modelo lineal mixto, que no es más que una extensión del modelo de regresión lineal que permite incorporar efectos aleatorios. Un mayor contenido sobre este tema se puede encontrar en McCulloch y Searle (2001).

Los modelos lineales mixtos tienen la siguiente forma:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{b} + \boldsymbol{\varepsilon}, \tag{2.9}$$

donde $\mathbf{y}$ es el vector de los valores de las n observaciones de la variable respuesta, $\mathbf{X}$ es una matriz $n \times p$ que contiene los valores de las p variables explicativas, $\boldsymbol{\beta}$ es el vector de parámetros fijos, $\mathbf{Z}$ es una matriz $n \times q$ que contiene la especificación de los efectos aleatorios $\mathbf{b}$, y $\boldsymbol{\varepsilon}$ es un vector n-dimensional de errores. Generalmente se asume normalidad para los errores $\boldsymbol{\varepsilon}$ y para los efectos aleatorios $\mathbf{b}$:

$$\begin{bmatrix} \mathbf{b} \\ \boldsymbol{\varepsilon} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \mathbf{G} & 0 \\ 0 & \mathbf{R} \end{bmatrix}\right)$$

donde $\mathbf{G} = \sigma_b^2\mathbf{I}$ y $\mathbf{R} = \sigma_\varepsilon^2\mathbf{I}$.

En las secciones 2.2 y 2.3 se revisó el comportamiento de un modelo de regresión no paramétrico sencillo que tenía la siguiente forma:

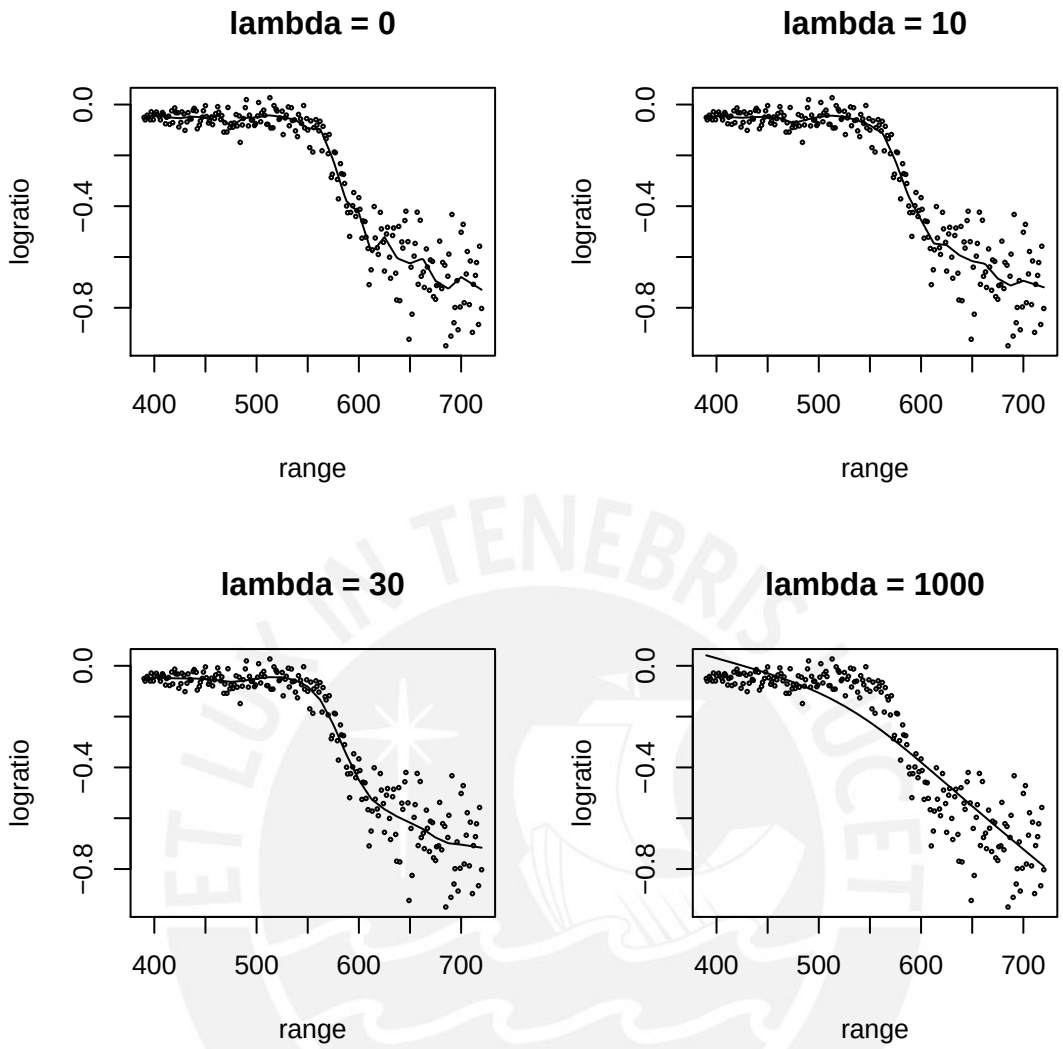

Figura 2.7: Ajuste de la data LIDAR mediante una regresión spline penalizada lineal para valores de $\lambda = 0$, 10, 30 y 1000 considerando los 25 nodos de la Figura 2.5.

$$y_i = f(x_i) + \varepsilon_i, \qquad 1 \leq i \leq n,$$

donde se mostró que f puede ser estimado mediante splines penalizados. Se supondrá que los errores satisfacen $\text{Cov}(\varepsilon) = \sigma_\varepsilon^2 \mathbf{I}$ y se considerará el modelo de regresión spline de la ecuación (2.6):

$$f(x_i) = \beta_0 + \beta_1 x_1 + \sum_{k=1}^{K} b_k |x_i - \kappa_k|^3.$$

Los coeficientes de efectos fijos y aleatorios del modelo anterior pueden expresarse de la siguiente forma matricial:

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} \quad \text{y} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_K \end{bmatrix}$$

Correspondientes a esos vectores, se definen la matriz de la variable $\mathbf{X}$ y la matriz de especificación de efectos aleatorios $\mathbf{Z}$ de la siguiente forma:

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \quad \text{y} \quad \mathbf{Z} = \begin{bmatrix} |x_1 - \kappa_1|^3 & \cdots & |x_1 - \kappa_K|^3 \\ \vdots & \ddots & \vdots \\ |x_n - \kappa_1|^3 & \cdots & |x_n - \kappa_K|^3 \end{bmatrix}$$

En la sección 2.3 se revisó que el valor de $\boldsymbol{\beta}$ era aquel que minimiza la expresión de la ecuación (2.7), que tiene la siguiente forma:

$$\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda^2 \boldsymbol{\beta}^{\boldsymbol{T}} \mathbf{D} \boldsymbol{\beta}.$$

Adaptando esta expresión al caso en revisión, y dividiendo entre σ_ε^2, la expresión a minimizar puede ser escrita como:

$$\frac{1}{\sigma_\varepsilon^2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{Z}\mathbf{b}\|^2 + \frac{\lambda^2}{\sigma_\varepsilon^2}\|\mathbf{b}\|^2. \tag{2.10}$$

Esta expresión se puede hacer idéntica a la justificación de Henderson (ver Robinson, 1991), que propone la estimación del mejor predictor lineal insesgado (BLUP por sus siglas en inglés) para $\boldsymbol{\beta}$ y $\mathbf{b}$ al minimizar la expresión:

$$(\mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{Z}\mathbf{b})^{\mathbf{T}} \mathbf{R}^{-\mathbf{1}} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{Z}\mathbf{b}) + \mathbf{b}^{\mathbf{T}} \mathbf{G}^{-\mathbf{1}} \mathbf{b}. \tag{2.11}$$

Lo importante de la expresión (2.11) es que permitirá relacionar el modelo de regresión spline penalizado con el modelo lineal mixto. Más información sobre el mejor predictor lineal insesgado se puede ver en el apéndice A.

Recordando que se definió previamente $\mathbf{G} = \sigma_b^2 \mathbf{I}$ y $\mathbf{R} = \sigma_\varepsilon^2 \mathbf{I}$, la expresión (2.11) quedaría de la siguiente forma:

$$\frac{(\mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{Z}\mathbf{b})^{\mathbf{T}} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{Z}\mathbf{b})}{\sigma_\varepsilon^{\mathbf{2}} \mathbf{I}} + \frac{\mathbf{b}^{\mathbf{T}} \mathbf{b}}{\sigma_b^2 \mathbf{I}}. \tag{2.12}$$

Para igualar las ecuaciones (2.10) y (2.12) es necesario tratar a $\mathbf{b}$ como un conjunto de coeficientes aleatorios con

$$\mathbf{b} \sim N\left(0, \sigma_b^2 \mathbf{I}\right), \quad \text{donde} \quad \sigma_b^2 = \sigma_\varepsilon^2 / \lambda^2.$$

Considerando todo lo mencionado en esta sección se tiene al modelo spline penalizado representado como un modelo lineal mixto:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{b} + \boldsymbol{\varepsilon}, \quad \text{donde} \quad \begin{bmatrix} \mathbf{b} \\ \boldsymbol{\varepsilon} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_b^2\mathbf{I} & 0 \\ 0 & \sigma_\varepsilon^2\mathbf{I} \end{bmatrix} \right)$$

Los valores ajustados de $\hat{\mathbf{y}}$ se pueden obtener mediante la siguiente expresión:

$$\hat{\mathbf{y}} = \mathbf{C}(\mathbf{C^T C} + \lambda^{\mathbf{2}}\mathbf{D})^{-\mathbf{1}}\mathbf{C^T y}, \tag{2.13}$$

donde

$$\mathbf{C} = [\mathbf{X}\ \mathbf{Z}], \quad \mathbf{D} = \text{diag}(0, 0, 1, 1, \ldots, 1), \quad \lambda^2 = \sigma_\varepsilon^2/\sigma_b^2.$$

Capítulo 3

Inferencia bayesiana del modelo de regresión spline penalizado a través de un marco de modelos lineales mixtos

3.1. Introducción

En el capítulo anterior se revisó la formulación del modelo spline penalizado mediante un modelo lineal mixto. La obtención de los parámetros de este modelo mediante un enfoque clásico se puede revisar en el capítulo anterior y con más detalle en el apéndice A. La estimación por máxima verosimilitud tiene propiedades asintóticas atractivas como la consistencia y normalidad asintótica; sin embargo, esta estimación podría tener serios problemas si hay una equivocación con el supuesto de la distribución o si se consideran tamaños de muestras pequeños. Por otro lado se tiene al enfoque bayesiano, que ofrece la ventaja de adoptar información de estudios similares o bien de expertos, la cual se incorpora en el análisis como distribuciones a priori de los parámetros. Esta información a priori puede ayudar a mejorar la inferencia de los parámetros considerando su distribución posterior.

En el enfoque bayesiano los parámetros del modelo son tratados como variables aleatorias, la cual es la mayor diferencia con los métodos de máxima verosimilitud. Las distribuciones que se asumen son llamadas a priori, las cuales son muy importantes para el desarrollo del modelo. La inferencia bayesiana se basa en la distribución a posteriori, que es la distribución condicional de las cantidades no observadas dada la data, tales como los parámetros o las variables no observadas. Esta distribución a posteriori tiene la siguiente forma:

$$f(\boldsymbol{\theta}|\mathbf{y}) = \frac{f(\mathbf{y}|\boldsymbol{\theta})f(\boldsymbol{\theta})}{f(\mathbf{y})} = \frac{L(\mathbf{y}|\boldsymbol{\theta})f(\boldsymbol{\theta})}{\int f(\mathbf{y}|\boldsymbol{\theta})f(\boldsymbol{\theta})\, d\boldsymbol{\theta}} \propto L(\mathbf{y}|\boldsymbol{\theta})f(\boldsymbol{\theta}) \tag{3.1}$$

donde se puede ver que la distribución a posteriori de $f(\boldsymbol{\theta}|\mathbf{y})$ es proporcional a la verosimilitud de los datos $\mathbf{y}$ dado $\boldsymbol{\theta}$, $L(\mathbf{y}|\boldsymbol{\theta})$, y a la distribución a priori de $\boldsymbol{\theta}$, $f(\boldsymbol{\theta})$.

Generalmente el cálculo es el principal desafío de la inferencia bayesiana. Sin embargo, gracias al desarrollo computacional moderno se puede emplear herramientas como el algoritmo Markov Chain Monte Carlo (MCMC), que permite que la inferencia bayesiana para diversos problemas sea factible e incluso sencilla. Respecto a las distribuciones a posteriori, generalmente la parte computacional es no trivial e intensiva. Para aproximar estas distribuciones podemos usar métodos MCMC, especialmente el muestro de Gibbs. Actualmente la disponibilidad de programas como el WinBUGS, que está basado en el BUGS (Baye-

sian Inference Using Gibbs Sampling), permiten que la computación bayesiana llegue a ser relativamente sencilla.

El enfoque bayesiano puede ser aplicado para los modelos lineales mixtos. En las siguientes secciones se seguirá la teoría expuesta en Hobert y Casella (1996), Sorensen y Gianola (2002) y Ruppert et al. (2003) a fin de formular el modelo desde una perspectiva bayesiana.

3.2. El modelo

Se considerará el modelo spline penalizado representado como un modelo mixto revisado en la sección 2.4:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{b} + \boldsymbol{\varepsilon}, \quad \text{donde} \quad \begin{bmatrix} \mathbf{b} \\ \boldsymbol{\varepsilon} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_b^2\mathbf{I} & 0 \\ 0 & \sigma_\varepsilon^2\mathbf{I} \end{bmatrix} \right)$$

donde $\mathbf{X}\boldsymbol{\beta}$ es la componente polinomial del spline y $\mathbf{Z}\mathbf{b}$ es la componente que contiene a las funciones de las bases spline.

El vector de parámetros del modelo estaría conformado por $(\boldsymbol{\beta}, \mathbf{b}, \sigma_b^2, \sigma_\varepsilon^2)$. Siguiendo a la ecuación (3.1), la distribución a posteriori sería:

$$f(\boldsymbol{\beta}, \mathbf{b}, \sigma_b^2, \sigma_\varepsilon^2|\mathbf{y}) \propto L(\mathbf{y}|\boldsymbol{\beta}, \mathbf{b}, \sigma_\varepsilon^2) f(\boldsymbol{\beta}) f(\mathbf{b}|\sigma_b^2) f(\sigma_b^2) f(\sigma_\varepsilon^2) \tag{3.2}$$

donde se supone independencia entre las distribuciones a priori.

La distribución condicional de la variable $\mathbf{y}$ a los parámetros $\boldsymbol{\beta}$, $\mathbf{b}$ y σ_ε^2 es:

$$\mathbf{y}|\boldsymbol{\beta}, \mathbf{b}, \sigma_\varepsilon^2 \sim N(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{b}, \sigma_\varepsilon^2\mathbf{I}).$$

Luego, la verosimilitud es dada por:

$$\begin{aligned} L(\mathbf{y}|\boldsymbol{\beta}, \mathbf{b}, \sigma_\varepsilon^2) &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_\varepsilon} \exp\left\{ -\frac{1}{2\sigma_\varepsilon^2} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{Z}\mathbf{b})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{Z}\mathbf{b}) \right\} \\ &= (2\pi\sigma_\varepsilon^2)^{(-n/2)} \exp\left\{ -\frac{1}{2\sigma_\varepsilon^2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{Z}\mathbf{b}\|^2 \right\} \end{aligned} \tag{3.3}$$

De las ecuaciones (3.2) y (3.3) se tiene a la distribución a posterior como:

$$f(\boldsymbol{\beta}, \mathbf{b}, \sigma_b^2, \sigma_\varepsilon^2|\mathbf{y}) \propto (2\pi\sigma_\varepsilon^2)^{(-n/2)} \exp\left\{ -\frac{1}{2\sigma_\varepsilon^2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{Z}\mathbf{b}\|^2 \right\} f(\mathbf{b}|\sigma_b^2) f(\sigma_b^2) f(\boldsymbol{\beta}) f(\sigma_\varepsilon^2) \tag{3.4}$$

El siguiente paso es definir las distribuciones a priori de $(\boldsymbol{\beta}, \mathbf{b}, \sigma_b^2, \sigma_\varepsilon^2)$ para poder obtener la distribución a posteriori.

3.3. Distribuciones a priori y a posteriori

En los modelos mixtos bayesianos, la definición de las distribuciones a priori de los parámetros es importante, sobre todo para la estimación de los componentes de la varianza, donde llega a ser crucial (ver Gelman, 2006). Por este motivo, se van a considerar las distribuciones a priori no informativas empleadas en Crainiceanu, Ruppert y Wand (2005).

Las distribuciones a priori consideradas son las siguientes:

$$\begin{aligned}
\boldsymbol{\beta} &\sim N(\mathbf{0}, \sigma_\beta^2 \mathbf{I}) \\
\mathbf{b} &\sim N(\mathbf{0}, \sigma_b^2 \mathbf{I}) \\
\sigma_\varepsilon^2 &\sim IG(A_\varepsilon, B_\varepsilon) \\
\sigma_b^2 &\sim IG(A_b, B_b)
\end{aligned}$$

donde IG denota a la distribución gamma inversa (*inverse gamma* por sus siglas en inglés), que tiene la siguiente distribución para σ_ε^2:

$$f(\sigma_\varepsilon^2) = \frac{B_\varepsilon^{A_\varepsilon}}{\Gamma(A_\varepsilon)} (\sigma_\varepsilon^2)^{-(A\varepsilon+1)} \exp\left(-\frac{B_\varepsilon}{\sigma_\varepsilon^2}\right). \tag{3.5}$$

Se asume que σ_β^2 debe tomar valores grandes a fin de que sea una priori no informativa. En el caso de $A_{\varepsilon,b}$ y $B_{\varepsilon,b}$ sus valores deben ser de tal forma que se tenga un valor elevado de la varianza, de tal manera que también sean distribuciones a priori no informativas.
Entonces, teniendo todas las distribuciones a priori definidas, la distribución a posteriori de la ecuación (3.4) quedaría de la siguiente forma:

$$\begin{aligned}
f(\boldsymbol{\beta}, \mathbf{b}, \sigma_b^2, \sigma_\varepsilon^2 | \mathbf{y}) \quad \propto \quad & (2\pi\sigma_\varepsilon^2)^{(-n/2)} \exp\left\{-\frac{1}{2\sigma_\varepsilon^2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{Z}\mathbf{b}\|^2\right\} (2\pi\sigma_\beta^2)^{(-1/2)} \\
& \times \exp\left\{-\frac{1}{2\sigma_\beta^2}\|\boldsymbol{\beta}\|^2\right\} (2\pi\sigma_b^2)^{(-1/2)} \exp\left\{-\frac{1}{2\sigma_b^2}\|\mathbf{b}\|^2\right\} \\
& \times \frac{B_\varepsilon^{A_\varepsilon}}{\Gamma(A_\varepsilon)} (\sigma_\varepsilon^2)^{-(A_\varepsilon+1)} \exp\left(-\frac{B_\varepsilon}{\sigma_\varepsilon^2}\right) \\
& \times \frac{B_b^{A_b}}{\Gamma(A_b)} (\sigma_b^2)^{-(A_b+1)} \exp\left(-\frac{B_b}{\sigma_b^2}\right) \\
\propto \quad & (\sigma_\varepsilon^2)^{-(n/2+A_\varepsilon+1)} (\sigma_b^2)^{-(1/2+A_b+1)} (\sigma_\beta^2)^{(-1/2)} \\
& \times \exp\left\{-\frac{1}{2\sigma_\varepsilon^2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{Z}\mathbf{b}\|^2 - \frac{1}{2\sigma_\beta^2}\|\boldsymbol{\beta}\|^2 - \frac{1}{2\sigma_b^2}\|\mathbf{b}\|^2 - \frac{B_\varepsilon}{\sigma_\varepsilon^2} - \frac{B_b}{\sigma_b^2}\right\}
\end{aligned} \tag{3.6}$$

Se debe notar que la ecuación (3.6) es analíticamente intratable, razón por la cual se empleará simulación mediante MCMC para poder obtener los parámetros del modelo.

3.4. Ajuste del modelo usando el algoritmo MCMC

En la sección anterior se revisó que la distribución a posteriori es analíticamente intratable, por este motivo se empleará simulación MCMC, específicamente el muestreo de Gibbs.

El algoritmo de Gibbs es un caso especial del algoritmo Metropolis-Hasting para un solo componente que emplea las distribuciones condicionales completas de los parámetros. Con este muestreo de Gibbs se puede generar una muestra vía simulación a partir de $f(\boldsymbol{\beta}, \mathbf{b}, \sigma_b^2, \sigma_\varepsilon^2 | \mathbf{y})$.

Más detalle se puede revisar en Ntzoufras (2009).

Según lo mencionado, es necesario hallar las siguientes distribuciones condicionales completas de todos los parámetros:

$$(\boldsymbol{\beta}|\mathbf{y},\mathbf{b},\sigma_\varepsilon^2,\sigma_b^2,\sigma_\beta^2)$$
$$(\mathbf{b}|\mathbf{y},\boldsymbol{\beta},\sigma_\varepsilon^2,\sigma_b^2,\sigma_\beta^2)$$
$$(\sigma_\varepsilon^2|\mathbf{y},\boldsymbol{\beta},\mathbf{b},\sigma_b^2,\sigma_\beta^2)$$
$$(\sigma_b^2|\mathbf{y},\boldsymbol{\beta},\mathbf{b},\sigma_\varepsilon^2,\sigma_\beta^2)$$

Para el caso de $\boldsymbol{\beta}$, se puede ver en la ecuación de la posteriori (3.6) que su distribución condicional completa es proporcional a:

$$\exp\left\{-\frac{1}{2\sigma_\varepsilon^2}\left(\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}-\mathbf{Z}\mathbf{b}\|^2+\frac{\sigma_\varepsilon^2}{\sigma_\beta^2}\|\boldsymbol{\beta}\|^2\right)\right\} \tag{3.7}$$

donde realizando la técnica de completar cuadrados (ver Sorensen y Gianola, 2002), se puede demostrar que:

$$(\boldsymbol{\beta}|\mathbf{y},\mathbf{b},\sigma_\varepsilon^2,\sigma_b^2,\sigma_\beta^2)\sim N\left(\left(\mathbf{X}^\mathbf{T}\mathbf{X}+\mathbf{I}\frac{\sigma_\varepsilon^2}{\sigma_\beta^2}\right)^{-1}\mathbf{X}^\mathbf{T}(\mathbf{y}-\mathbf{Z}\mathbf{b}),\left(\mathbf{X}^\mathbf{T}\mathbf{X}+\mathbf{I}\frac{\sigma_\varepsilon^2}{\sigma_\beta^2}\right)^{-1}\sigma_\varepsilon^2\right) \tag{3.8}$$

siendo $\mathbf{I}$ la matriz identidad no singular que acompaña a la varianza de $\boldsymbol{\beta}$.
Para el caso de $\mathbf{b}$, se puede ver en la ecuación de la posteriori (3.6) que su distribución condicional completa es proporcional a:

$$\exp\left\{-\frac{1}{2\sigma_\varepsilon^2}\left(\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}-\mathbf{Z}\mathbf{b}\|^2+\frac{\sigma_\varepsilon^2}{\sigma_b^2}\|\mathbf{b}\|^2\right)\right\} \tag{3.9}$$

donde realizando la técnica de completar cuadrados (ver Sorensen y Gianola, 2002), se puede demostrar que:

$$(\mathbf{b}|\mathbf{y},\boldsymbol{\beta},\sigma_\varepsilon^2,\sigma_b^2,\sigma_\beta^2)\sim N\left(\left(\mathbf{Z}^\mathbf{T}\mathbf{Z}+\mathbf{I}\frac{\sigma_\varepsilon^2}{\sigma_b^2}\right)^{-1}\mathbf{Z}^\mathbf{T}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}),\left(\mathbf{Z}^\mathbf{T}\mathbf{Z}+\mathbf{I}\frac{\sigma_\varepsilon^2}{\sigma_b^2}\right)^{-1}\sigma_\varepsilon^2\right) \tag{3.10}$$

siendo $\mathbf{I}$ la matriz identidad no singular que acompaña a la varianza de $\mathbf{b}$. Además, se puede ver la formación del término $\sigma_\varepsilon^2/\sigma_b^2$, que es igual al parámetro de suavización λ^2.
Para el caso de σ_ε^2, se puede ver en la ecuación de la posteriori (3.6) que su distribución condicional completa es proporcional a:

$$(\sigma_\varepsilon^2)^{-(n/2+A_\varepsilon+1)}\exp\left\{-\frac{1}{\sigma_\varepsilon^2}\left(\frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}-\mathbf{Z}\mathbf{b}\|^2+B_\varepsilon\right)\right\} \tag{3.11}$$

que al compararse con (3.5), se observa que:

$$(\sigma_\varepsilon^2|\mathbf{y},\boldsymbol{\beta},\mathbf{b},\sigma_b^2,\sigma_\beta^2) \sim IG\left(A_\varepsilon + \frac{1}{2}n, B_\varepsilon + \frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}-\mathbf{Z}\mathbf{b}\|^2\right). \tag{3.12}$$

Por el mismo razonamiento se puede obtener σ_b^2, teniendo que:

$$(\sigma_b^2|\mathbf{y},\boldsymbol{\beta},\mathbf{b},\sigma_\varepsilon^2,\sigma_\beta^2) \sim IG\left(A_b + \frac{1}{2}K, B_b + \frac{1}{2}\|\mathbf{b}\|^2\right). \tag{3.13}$$

donde K es la cantidad de nodos que se considere en el modelo.

Finalmente, de las ecuaciones (3.8), (3.10), (3.12) y (3.13) se tienen todas las distribuciones condicionales completas de los parámetros del modelo $(\boldsymbol{\beta},\mathbf{b},\sigma_b^2,\sigma_\varepsilon^2)$, con lo cual es posible implementar el algoritmo del muestro de Gibbs. Este algoritmo se puede ver en la sección 3.6.1.

3.5. Criterios para la comparación de modelos

Según Bazan y Bayes (2010), existe una serie de metodologías para comparar y seleccionar modelos bayesianos alternativos. Entre los principales criterios se tienen:

- DIC: Criterio de información de Desvío.
- EAIC: Esperado del criterio de información de Akaike.
- EBIC: Esperado del criterio de información Bayesiano.

Estos criterios se basan en la media a posteriori del desvío $E(D(\theta))$, donde

$$D(\theta) = -2ln(L(\theta)) = -2\sum_{i=1}^{n} lnf(y_i|\theta) \tag{3.14}$$

Esta expresión es una medida de ajuste que puede ser aproximada mediante el resultado de la simulación vía MCMC, la cual se conoce como $Dbar$ y es dada por:

$$Dbar = \frac{1}{T}\sum_{i=1}^{T} D(\theta^i) \tag{3.15}$$

donde θ^i es el i-ésimo valor simulado del parámetro de un total de T iteraciones.
El EAIC, EBIC y DIC pueden ser estimados de la siguiente forma:

$$\begin{aligned}
\widehat{EAIC} &= Dbar + 2p \\
\widehat{EBIC} &= Dbar + plnN \\
\widehat{DIC} &= Dbar + \hat{p_D} = 2Dbar - Dhat
\end{aligned}$$

donde p es el número de parámetros del modelo, N es el total de observaciones y $\hat{p_D}$ es el número efectivo de parámetros definido como:

$$\hat{p_D} = E\left[D(\hat{\theta})\right] - D\left[E(\hat{\theta})\right] = Dbar - Dhat$$

siendo $D\left[E(\hat{\theta})\right]$ el desvío de la media posteriori, el cual es estimado por:

$$Dhat = D\left(\frac{1}{T}\sum_{i=1}^{T}\theta^i\right) \tag{3.16}$$

En la comparación de modelos alternativos, aquel que presente el menor valor del DIC, EAIC y EBIC será el modelo que mejor se ajuste al conjunto de datos. Los valores $2p$ y $plnN$ penalizan a la media a posteriori del desvío a fin de evitar valores altos del *Dbar* a medida que crezca el número de parámetros a considerar.

DIC y *Dbar* son reportados en WinBUGS cuando uno lo requiere durante la simulación. EAIC y EBIC se pueden calcular a través del valor obtenido del *Dbar* considerando las expresiones presentadas.

3.6. Aspectos computacionales

3.6.1. Algoritmo de Gibbs

El muestreo de Gibbs es un caso especial del algoritmo Metropolis Hastings para un solo componente y emplea las distribuciones condicionales completas de los parámetros, es decir, muestrea la distribución condicional de cada parámetro dado todos los demás. El algoritmo de simulación MCMC que permitirá estimar los parámetros del modelo de regresión spline penalizado se realiza siguiendo a Ruppert et al. (2003) y Ntzoufras (2009).

Para estimar los parámetros a posteriori se realizan N iteraciones realizando los siguientes pasos:

(1) Obtener una muestra de $\boldsymbol{\beta}$ y $\mathbf{b}$ de las siguientes distribuciones normales multivariadas:

$$(\boldsymbol{\beta}|\mathbf{y},\mathbf{b},\sigma_\varepsilon^2,\sigma_b^2,\sigma_\beta^2) \sim N\left(\left(\mathbf{X^T X}+\mathbf{I}\frac{\sigma_\varepsilon^2}{\sigma_\beta^2}\right)^{-1}\mathbf{X^T}(\mathbf{y}-\mathbf{Zb}),\left(\mathbf{X^T X}+\mathbf{I}\frac{\sigma_\varepsilon^2}{\sigma_\beta^2}\right)^{-1}\sigma_\varepsilon^2\right)$$

$$(\mathbf{b}|\mathbf{y},\boldsymbol{\beta},\sigma_\varepsilon^2,\sigma_b^2,\sigma_\beta^2) \sim N\left(\left(\mathbf{Z^T Z}+\mathbf{I}\frac{\sigma_\varepsilon^2}{\sigma_b^2}\right)^{-1}\mathbf{Z^T}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}),\left(\mathbf{Z^T Z}+\mathbf{I}\frac{\sigma_\varepsilon^2}{\sigma_b^2}\right)^{-1}\sigma_\varepsilon^2\right).$$

(2) Obtener una muestra de σ_b^2 de la distribución gamma inversa:

$$(\sigma_b^2|\mathbf{y},\boldsymbol{\beta},\mathbf{b},\sigma_\varepsilon^2,\sigma_\beta^2) \sim IG\left(A_b+\frac{1}{2}K, B_b+\frac{1}{2}\|\mathbf{b}\|^2\right).$$

(3) Obtener una muestra de σ_ε^2 de la distribución gamma inversa:

$$(\sigma_\varepsilon^2|\mathbf{y},\boldsymbol{\beta},\mathbf{b},\sigma_b^2,\sigma_\beta^2) \sim IG\left(A_\varepsilon+\frac{1}{2}n, B_\varepsilon+\frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}-\mathbf{Zb}\|^2\right).$$

(4) Regresar al paso (1) e iterar.

Este algoritmo será implementado en los programas R y WinBUGS.

3.6.2. R y R2WinBUGS

El algoritmo de simulación MCMC descrito en el acápite anterior será implementado en los programas R y WinBUGS. En el programa R se implementa un código propio del algoritmo para que realice N iteraciones. Para el caso de las distribuciones normales multivariadas se empleará la función **rmnorm** del paquete **mnormt**. Para el caso de las distribuciones gamma inversas se declarará a los valores inversos de σ_b^2 y σ_ε^2 como distribuciones gamma.

La implementación del programa en WinBUGS se realizará en el R a través del paquete **R2WinBUGS**, el cual permite poder manejar la ejecución de modelos bayesianos de dicho programa. En el WinBUGS no es necesario codificar el algoritmo del acápite anterior dado que este programa se encarga de emplear el muestreo de Gibbs para efectuar la simulación MCMC. Lo único que hay que codificar es la verosimilitud del modelo y las distribuciones a priori.

En ambos programas es necesario declarar lo siguiente:

- Los datos que serán ajustados al modelo de regresión spline penalizado.
- Los valores iniciales que tomarán los parámetros para empezar la simulación.
- El número de iteraciones que realizará la simulación MCMC.
- El periodo de calentamiento (*burn in* en inglés) que no se considerará para el análisis de los resultados de la simulación.
- El salto entre iteraciones (*thin* en inglés) que será considerado en el análisis de los resultados de la simulación.

Capítulo 4

Estudio de Simulación

4.1. Introducción

En el presente capítulo se desarrollará un estudio de simulación que permita responder dos interrogantes importantes sobre el empleo del modelo de regresión spline penalizado desde la perspectiva bayesiana. La primera es poder saber si el mejor ajuste se obtiene al emplear la estimación vía inferencia bayesiana o al emplear la estimación clásica por máxima verosimilitud. La segunda interrogante va por el lado de comparar algunas formas de implementar la inferencia bayesiana a fin de evaluar la convergencia de los coeficientes del modelo y la velocidad con la que se pueda ejecutar. Este último punto resulta ser fundamental ya que muchas veces es necesario realizar muchas simulaciones para lograr resultados adecuados.

El estudio de simulación que se realiza en el presente capítulo se divide en dos partes. En la primera parte del estudio se buscará comparar el método de estimación por máxima verosimilitud contra el método de estimación bayesiana para ajustar una curva mediante el modelo spline penalizado, el cual se implementará a través de un marco de modelos lineales mixtos. La estimación mediante el enfoque clásico se realizará en el programa R utilizando la función **lme** de la librería **nlme**. La estimación desde la perspectiva bayesiana se realizará en el programa R implementando un código propio con el algoritmo de Gibbs descrito en la sección 3.6.1, y en el programa WinBUGS a través de un código que presenta la verosimilitud del modelo y las distribuciones a priori de los parámetros. En ambos enfoques se evaluará como medida de bondad de ajuste al error cuadrático medio y se presentarán los valores estimados de los parámetros de la regresión.

En la segunda parte del estudio se compararán los errores cuadráticos medio empleando el enfoque bayesiano y clásico para distintos tamaños de muestra. El objetivo de esta parte es comprobar si la estimación bayesiana resulta ser mejor que la estimación vía máxima verosimilitud al considerar tamaños de muestra pequeños y grandes.

4.2. Objetivos

El estudio de simulación tiene como objetivo estudiar y comparar métodos de estimación del modelo de regresión spline penalizado desde un enfoque clásico y bayesiano. Los métodos considerados son:

- Estimación por máxima verosimilitud empleando la librería **nlme** del programa R (Máx. Verosimilitud).

- Estimación por inferencia bayesiana mediante la implementación de un código propio con el algoritmo de Gibbs en R (Algoritmo propio).
- Estimación por inferencia bayesiana mediante la implementación de un código BUGS usando la librería R2WinBUGS (Algoritmo BUGS).

Según lo revisado en la sección 3.3, en la implementación del modelo bayesiano se considerarán las prioris no informativas utilizadas en Crainiceanu, Ruppert y Wand (2005), teniendo a:

- $\beta_0, \beta_1 \sim N(0, 10^6)$
- $\sigma_b^{-2}, \sigma_\varepsilon^{-2} \sim \text{Gamma}(10^{-6}, 10^{-6})$

4.3. Descripción del estudio

El estudio consiste en realizar el ajuste de los datos de una curva de forma no conocida mediante el modelo de regresión spline penalizado, tanto desde la perspectiva clásica como de la bayesiana. La figura 4.1 muestra la gráfica de la curva que se empleará en el estudio de simulación, la cual es descrita por la siguiente expresión:

$$y = f(x) = \sin(x) + 2\exp(-10x^2) + \frac{x}{4} - \frac{x^2}{50} + 5, \quad \text{donde } x \in [0, 20]$$

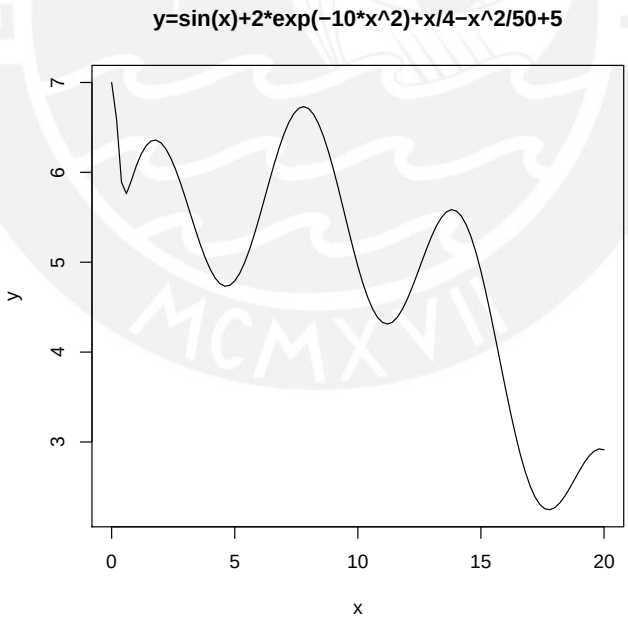

Figura 4.1: Curva que se empleará en el estudio de simulación

Se generarán $n = 100$ valores de x de manera uniforme entre 0 y 20, a partir de estos se generarán valores para y_i de la siguiente forma:

$$y_i = f(x_i) + \varepsilon_i, \quad \text{donde} \quad \varepsilon_i \sim N(0, \sigma^2).$$

donde se evaluarán escenarios para $\sigma^2 = 0.5$, 1 y 1.5.

Respecto al tipo de spline, se considerará el spline cúbico de base radial empleado en Crainiceanu et al. (2005). Este spline tiene la siguiente forma:

$$f(x) = \beta_0 + \beta_1 x + \sum_{k=1}^{K} b_k |x - \kappa_k|^3,$$

Adicionalmente, dado que el número de nodos puede influir en el ajuste del modelo, se van a considerar cuatro cantidades de nodos ($K = 5$, 10, 15 y 20) a fin de comparar la estimación obtenida con cada caso.

En resumen, estos son los escenarios a evaluar:

- Spline: 1
  - Spline cúbico de base radial.
- Nodos: 4
  - K = 5, 10, 15 y 20.
- Varianza para el error aleatorio: 3
  - $\sigma^2 = 0.5$, 1 y 1.5.

En total se van a simular $1 \times 4 \times 3 = 12$ escenarios por cada método empleado. Dado que en el presente estudio se implementará un método de estimación por máxima verosimilitud y dos métodos bayesianos (en R y WinBUGS), se tendrán un total de 36 escenarios.

Como medida de bondad de ajuste se considerará al error cuadrático medio de los valores predichos a la curva original. La expresión es dada por:

$$\widehat{ECM} = \frac{1}{n} \sum_{i=1}^{n} \left(\hat{f}(x_i) - f(x_i) \right)^2$$

donde n es el tamaño de la muestra de los valores simulados que van a ser estimados.

4.4. Implementación

La implementación de la simulación se efectúa en el programa **R** mediante una rutina que permite obtener los resultados por el enfoque clásico vía máxima verosimilitud y por el enfoque bayesiano de dos métodos en todos los escenarios planteados. El primer método es mediante un código propio del algoritmo de Gibbs en R y el otro mediante un código BUGS. Esta rutina se puede dividir en las siguientes etapas:

- Cargar las librerías **nlme**, **mnormt** y **R2WinBUGS**.
- Definir los escenarios que se van evaluar en función de la cantidad de nodos y de la varianza.

- Generar la data simulada y determinar cuáles van a ser los nodos a considerar.
- Crear las matrices de diseño **X** y **Z**.
- Estimar vía Máxima Verosimilitud:
  - Emplear la función **lme** de la librería **nlme** para hallar los parámetros del modelo.
  - Predecir los nuevos valores en base al modelo.
  - Grabar las variables y estadísticos importantes.
- Estimar vía inferencia bayesiana mediante un código propio en R:
  - Declarar valores iniciales de las variables a emplear.
  - Iterar N veces el algoritmo de Gibbs en base a las distribuciones condicionales completas de los parámetros.
  - Grabar las variables y estadísticos importantes.
- Estimar vía inferencia bayesiana mediante un código BUGS usando la librería R2WinBUGS:
  - Definir la verosimilitud y distribuciones a priori del modelo.
  - Emplear la función **bugs** e iterar N simulaciones.
  - Grabar las variables y estadísticos importantes.
- Consolidar las variables y estadísticos importantes de cada escenario evaluado.
- Guardar la información consolidada en una base de datos a fin de realizar el análisis.

En los modelos bayesianos se realizan iteraciones que sean suficientemente grandes para asegurar una buena convergencia. En la presente tesis se ha considerado un total de 1,000,000 iteraciones, un periodo de calentamiento (*burn in*) de 100,000 y un salto entre iteraciones (*thin*) de 300 a fin de reducir la autocorrelación y mejorar la convergencia (en el apéndice B se presentan otras dos simulaciones con baja convergencia). Como resultado, se tienen 3,000 iteraciones para preparar estadísticas y diversos análisis. En los métodos bayesianos además de poder guardar la media de los vectores resultantes también se pueden almacenar los valores de la mediana.

El detalle del código de esta rutina en el programa **R** se puede ver en el apéndice C. El código empleado en el programa **WinBUGS** se puede ver en el apéndice D.

4.5. Resultados

Resultados del error cuadrático medio

Las Figuras 4.2, 4.3 y 4.4 presentan el error cuadrático medio mediante un gráfico de líneas para cada uno de los métodos de estimación implementados. Además, en el caso de la estimación bayesiana se presenta la comparación considerando a la media y la mediana como estimadores puntuales. De estas figuras se puede observar lo siguiente:

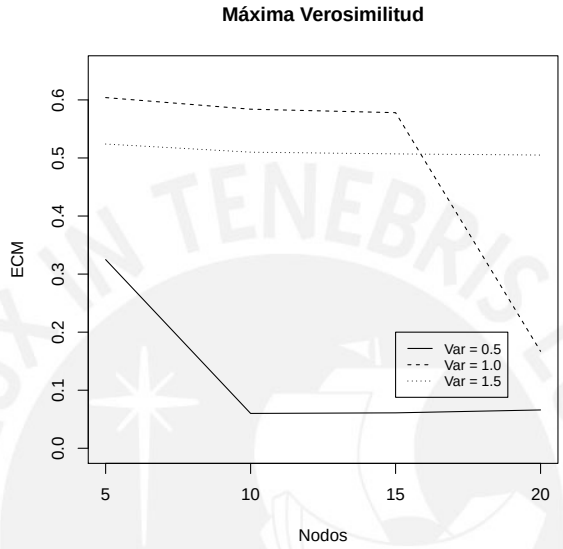

Figura 4.2: Error cuadrático medio para la estimación vía Máxima Verosimilitud

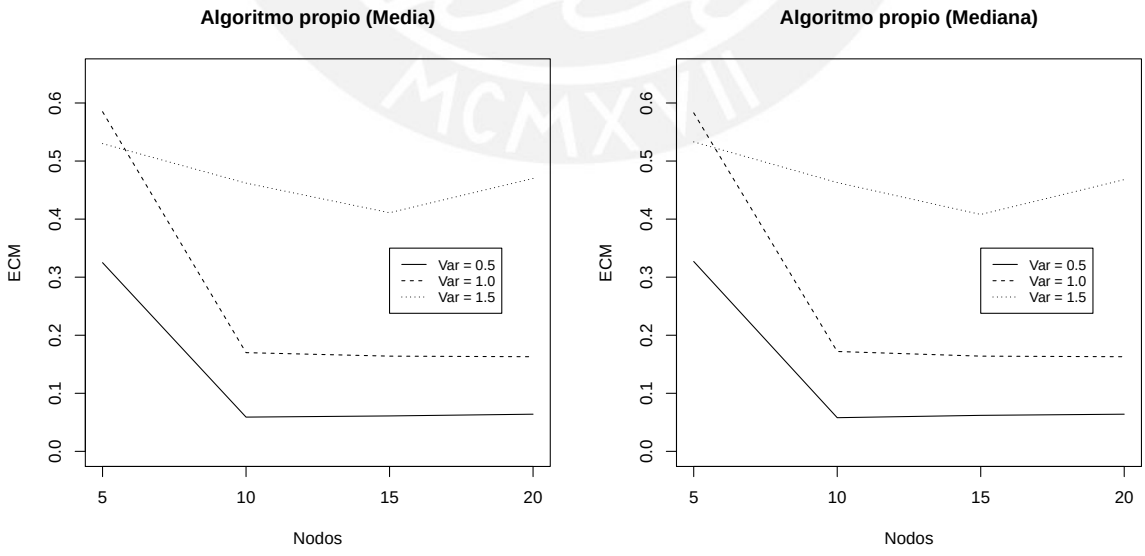

Figura 4.3: Error cuadrático medio para la estimación bayesiana vía Algoritmo propio

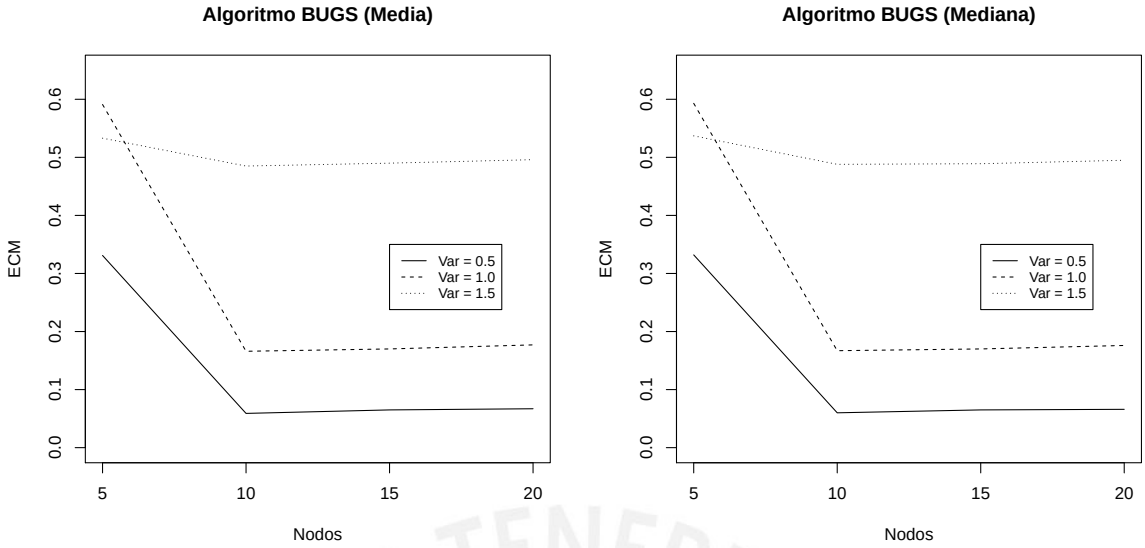

Figura 4.4: Error cuadrático medio para la estimación bayesiana vía Algoritmo BUGS

- Respecto a la cantidad de nodos:
  - En general, al considerar mayor cantidad de nodos el error cuadrático medio disminuye.
  - En la mayoría de casos se presenta una mejora significativa en el error cuadrático medio al considerar 10 nodos, incluso se puede ver que en varios casos al contar con 15 y 20 nodos el ECM prácticamente no disminuye.
- Respecto a la varianza σ^2:
  - A medida que se incrementa la varianza aumenta el error cuadrático medio, es decir, los datos están cada vez más dispersos.
  - En el caso de tener $\sigma^2 = 1.5$ los ECM prácticamente no disminuyen al considerar una mayor cantidad de nodos en el modelo. Esto puede significar que la data simulada haya perdido la forma de la curva original.
- Respecto a los estimadores puntuales en la inferencia Bayesiana:
  - Prácticamente no hay diferencias en los ECM al emplear la media o la mediana.

Luego de este análisis preliminar, el siguiente paso es identificar cuál de los metodos de estimación implementados es que el presenta menor ECM. Para esto, es necesario comparar los ECM de cada método de estimación en función al nivel de dispersión de los datos y a la cantidad de nodos, tal como se muestra en las Figuras 4.5, 4.6 y 4.7. Respecto a estos métodos implementados se puede observar lo siguiente:

- En el caso de tener una varianza pequeña, $\sigma^2 = 0.5$, se consiguen ECM similares entre el enfoque clásico y bayesiano.

- En el caso de contar con una varianza mayor, $\sigma^2 = 1.0$, el método de máxima verosimilitud no consigue ajustes adecuados debido a que el ECM se mantiene similar al considerar 10 y 15 nodos, a diferencia de lo que sucede al contar con 20 nodos. Por otro lado, en los métodos bayesianos la mejora del ECM se obtiene desde que se consideran 10 nodos.
- En el caso que la varianza sea grande, $\sigma^2 = 1.5$, los métodos bayesianos consiguen menores ECM.

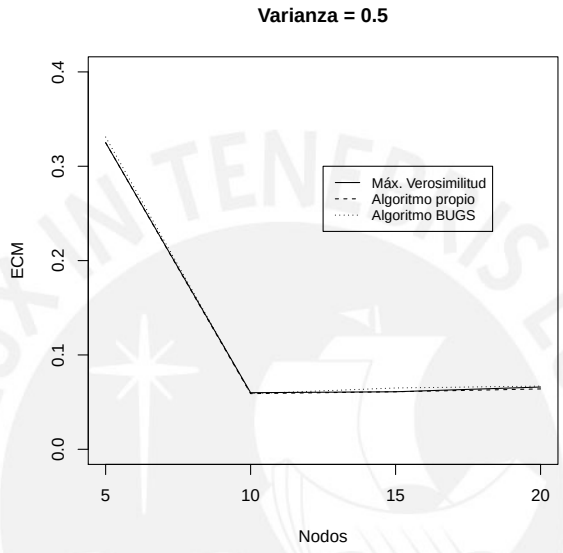

Figura 4.5: Error cuadrático medio para cada método de estimación considerando σ^2= 0.5

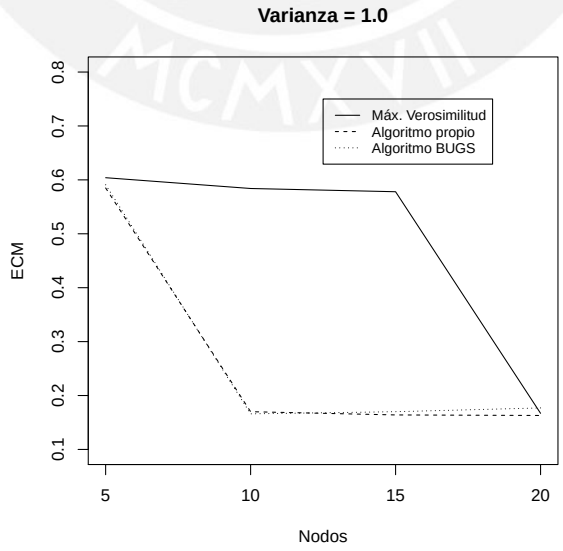

Figura 4.6: Error cuadrático medio para cada método de estimación considerando σ^2= 1.0

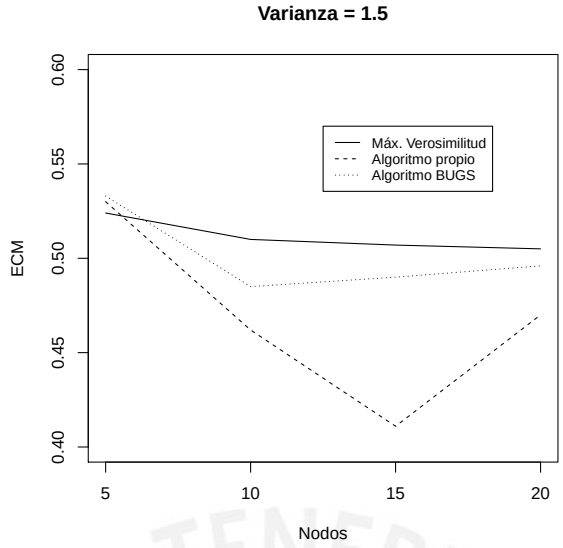

Figura 4.7: Error cuadrático medio para cada método de estimación considerando $\sigma^2 = 1.5$

El detalle de los valores del ECM de cada escenario se puede encontrar en el Apéndice B.

Por otro lado, si bien se han revisado los resultados del ECM para cada escenario, es necesario ver de manera gráfica el ajuste del modelo a la curva simulada. Por este motivo, las figuras 4.8, 4.9 y 4.10 presentan dicho ajuste para $\sigma^2 = 0.5$, 1.0 y 1.5, respectivamente, con cada método de estimación implementado. De estos gráficos se puede observar lo siguiente:

- En el caso de tener $\sigma^2 = 0.5$, figura 4.8, el ajuste a la data simulada es similar en todos los métodos implementados. Además, se puede ver que 5 nodos no son suficientes para obtener un ajuste adecuado, a diferencia de considerar 10, 15 y 20 nodos.
- En el caso de tener $\sigma^2 = 1.0$, figura 4.9, los métodos de estimación bayesiana presentan un mejor ajuste que la estimación clásica cuando se consideran 10 y 15 nodos. La estimación por máxima verosimilitud sólo consigue un ajuste adecuado al contar con 20 nodos.
- En el caso de tener $\sigma^2 = 1.5$, figura 4.10, la dispersión de los datos es tan alta que se pierde la forma de la curva original, razón por la cual ningún método de estimación presenta ajustes adecuados.

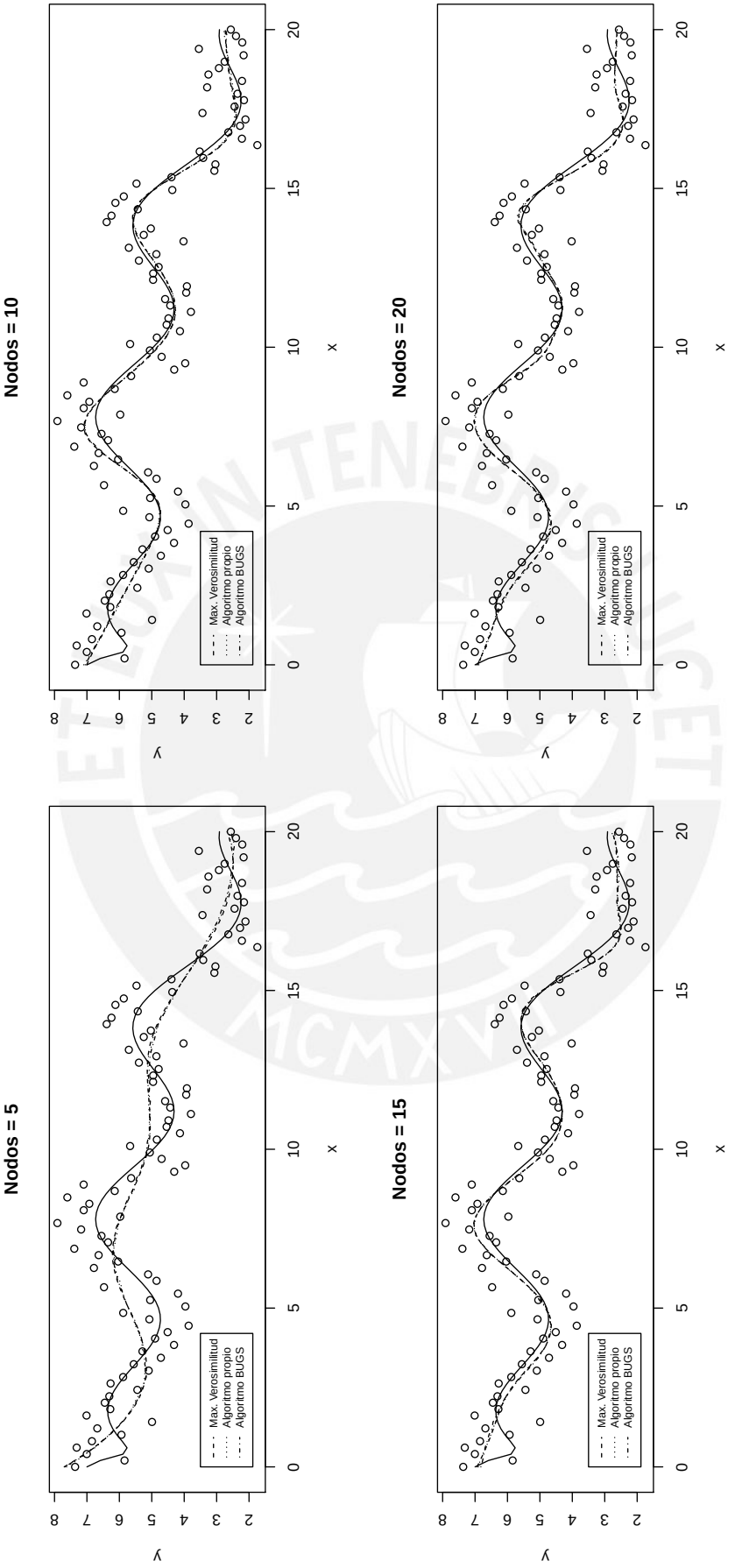

Figura 4.8: Ajuste del modelo a los datos considerando $\sigma^2 = 0.5$

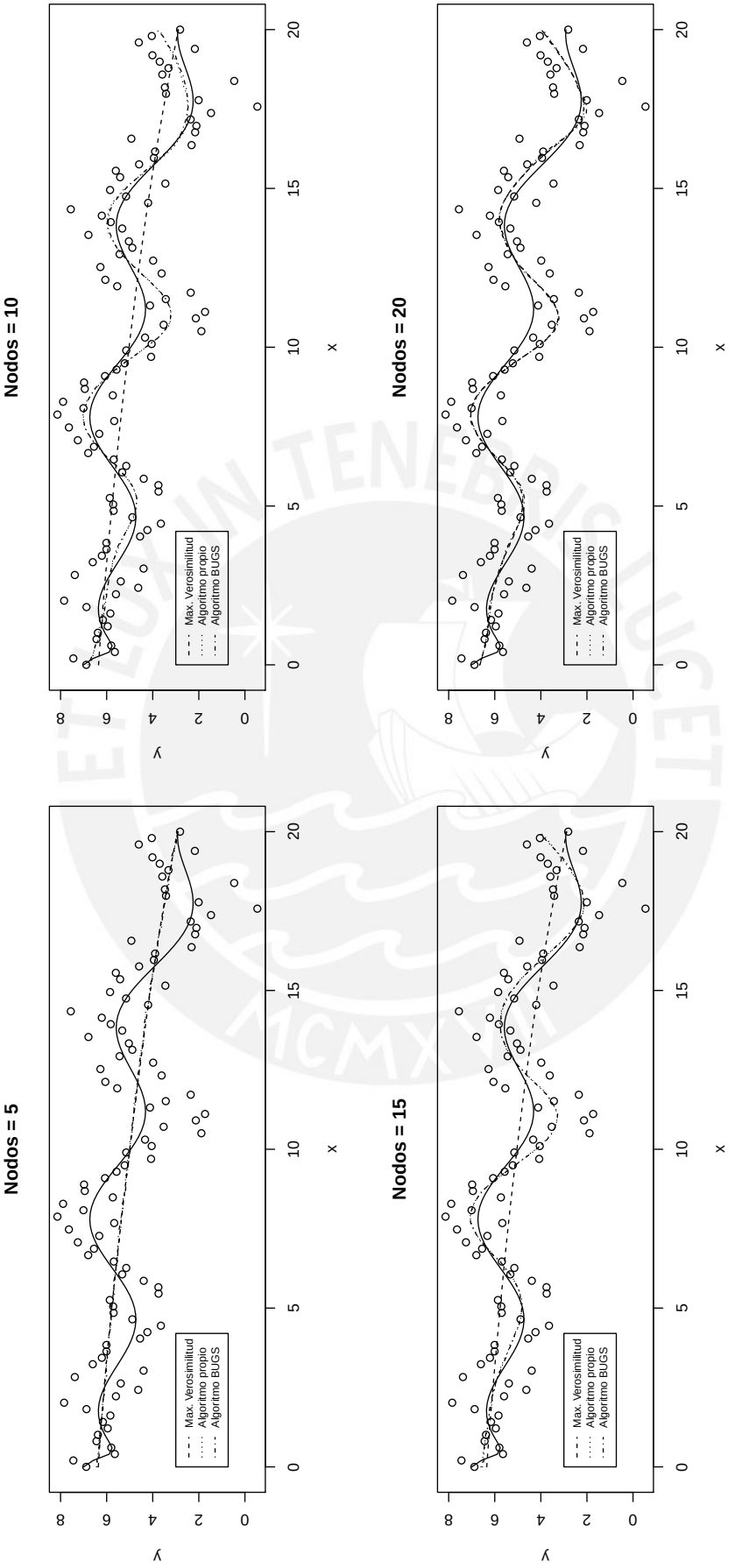

Figura 4.9: Ajuste del modelo a los datos considerando $\sigma^2 = 1.0$

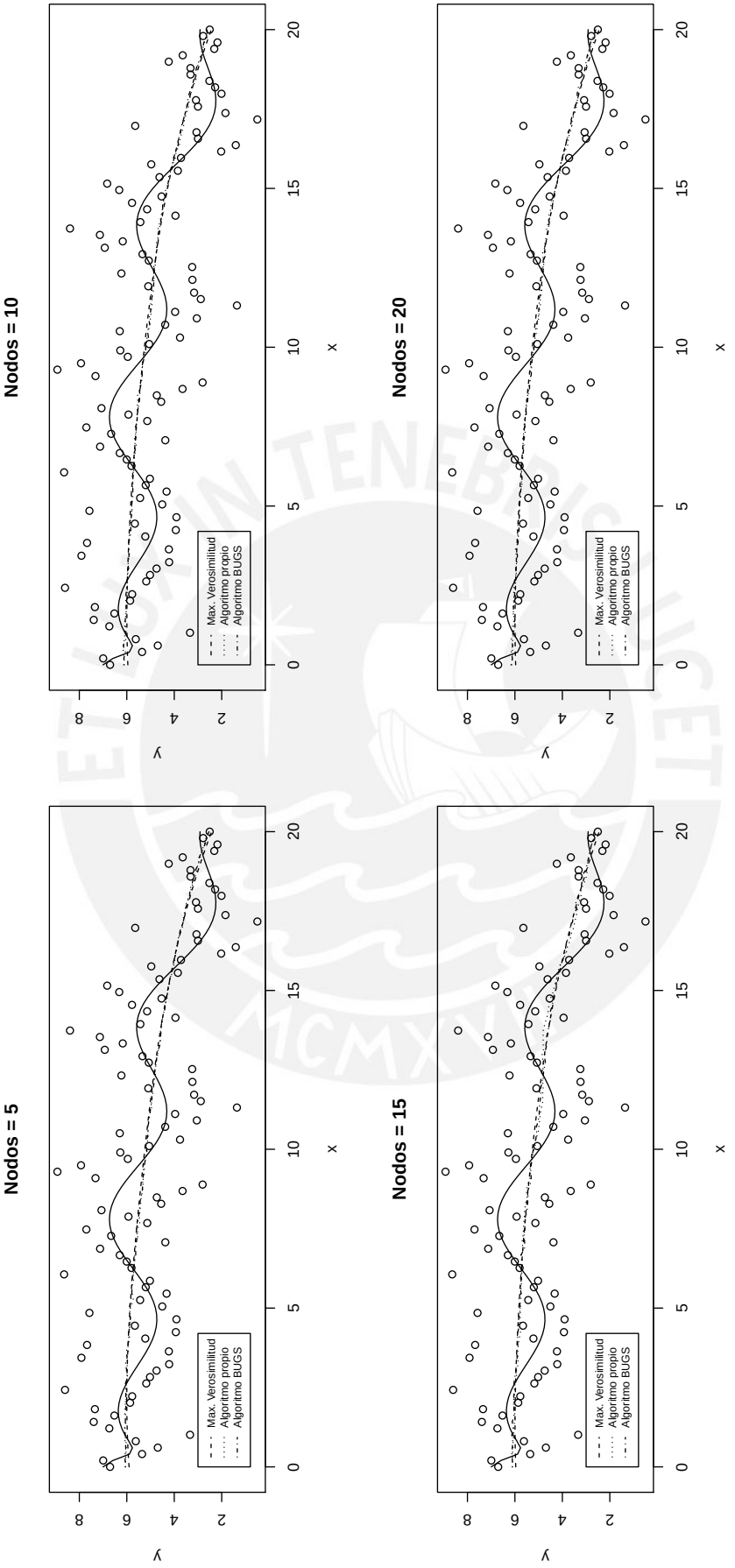

Figura 4.10: Ajuste del modelo a los datos considerando $\sigma^2 = 1.5$

Resultados de los parámetros del modelo

De acuerdo a lo revisado en los resultados del error cuadrático medio y a las gráficas de ajustes del modelo a la curva original, se considerará el escenario con $\sigma^2 = 1$ dado que contempla una variabilidad alta sin que los datos lleguen a perder la forma de la curva original. Asimismo, en lo que concierne a los nodos, se consideran 10 dado que es la cantidad con la que se obtiene una mejora significativa en el error cuadrático medio.

El cuadro 4.1 presenta la estimación de los parámetros desde un enfoque clásico y bayesiano. En la primera columna están los parámetros del modelo. La segunda, tercera y cuarta columna muestran la estimación por máxima verosimilitud y bayesiana para el spline cúbico de base radial.

Parámetros del modelo	Spline cúbico base radial		
	Máx. Verosimilitud	Algoritmo propio	Algoritmo BUGS
β_0	6.71	7.5	7.8
β_1	-0.13	-0.2	-0.22
σ_ε	1.09	1.08	1.07
σ_b	0.03	0.07	0.08

Cuadro 4.1: Parámetros del modelo considerando 10 nodos y $\sigma^2 = 1$

La convergencia de las iteraciones de estos parámetros $\beta_0, \beta_1, \sigma_\varepsilon$ y σ_b se presenta en las figuras 4.11, 4.12, 4.13 y 4.14, respectivamente, para cada uno de los métodos bayesianos. De todos estos resultados presentados, se puede observar:

- Respecto a los métodos de estimación implementados:
  - La estimación clásica del parámetro β_0 difiere respecto a los métodos bayesianos. Esta puede ser la razón de por qué la estimación por máxima verosimilitud no ajusta de manera adecuada la curva simulada (ver figuras 4.2, 4.6 y 4.9).
- Respecto a la convergencia de los parámetros β_0, β_1, σ_ε y σ_b:
  - β_0 y β_1 convergen con ambos métodos bayesianos implementados. Además, la estimación puntual es similar en ambos métodos.
  - σ_ε y σ_b convergen con ambos métodos bayesianos implementados. Adicionalmente, la estimación puntual es similar en el enfoque clásico y bayesiano.

Por otro lado, en el apéndice B se puede revisar otros dos escenarios de simulación que presentan una baja convergencia en los parámetros β_0 y β_1; adicionalmente, en este mismo apéndice se puede encontrar los tiempos de ejecución del presente estudio de simulación.

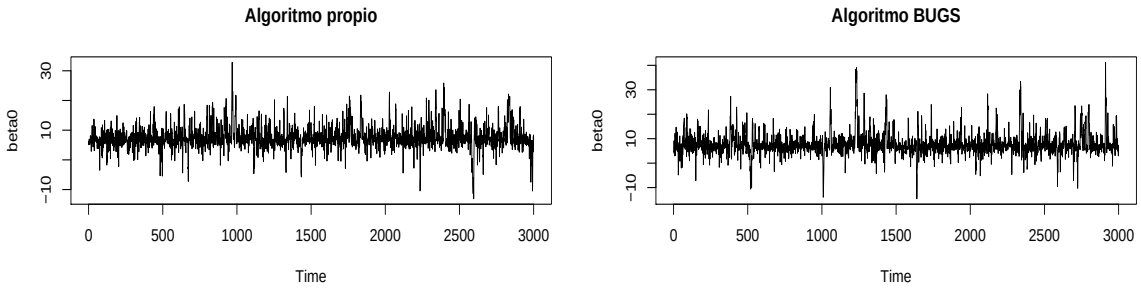

Figura 4.11: Valores del coeficiente β_0 en cada iteración

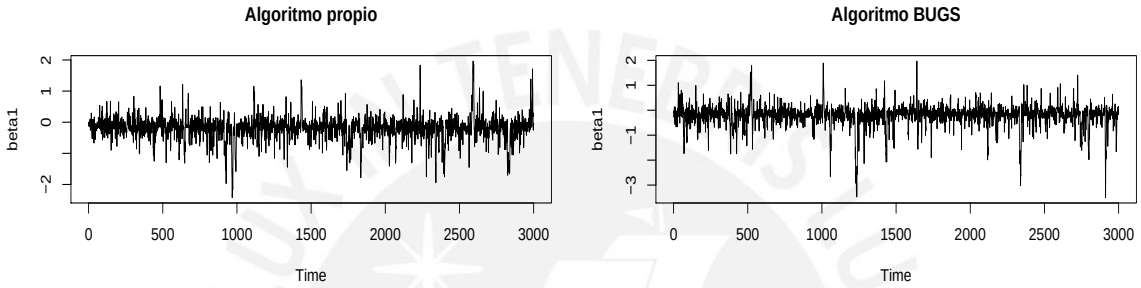

Figura 4.12: Valores del coeficiente β_1 en cada iteración

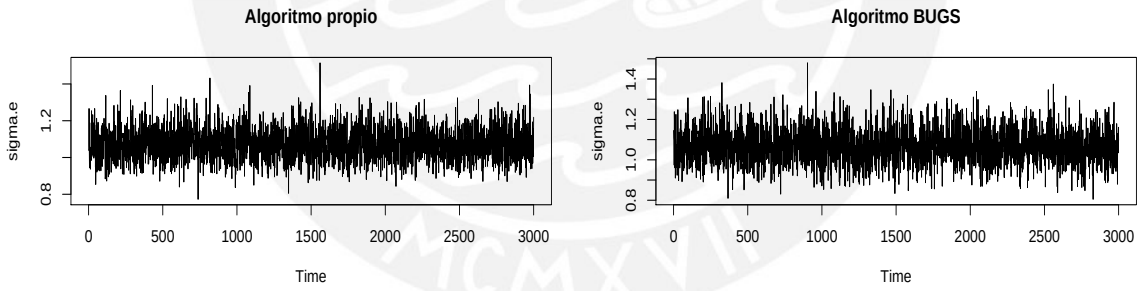

Figura 4.13: Valores del coeficiente σ_ε en cada iteración

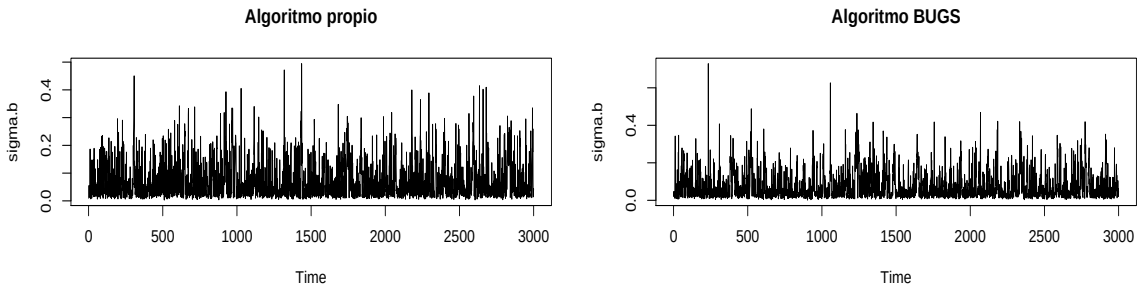

Figura 4.14: Valores del coeficiente σ_b en cada iteración

DIC del modelo bayesiano en WinBUGS

El cuadro 4.2 presenta el DIC para el algoritmo de Gibbs implementado en el programa WinBUGS. En relación al valor del DIC, se puede observar que en todos los casos es suficiente considerar 10 ó 15 nodos. Esto quiere decir que la estimación del modelo (para estos datos) prácticamente no mejorará al considerar más de 15 nodos, como se puede ver en los resultados de los escenarios con 20 nodos.

Escenarios			Algoritmo BUGS
Spline	σ^2	Nodos	DIC
cúbico	0.5	5	262.5
de base	0.5	10	228.8
radial	0.5	15	226.7
	0.5	20	227.7
	1	5	310.3
	1	10	299.5
	1	15	296.7
	1	20	296.9
	1.5	5	376.5
	1.5	10	375.5
	1.5	15	375.2
	1.5	20	375.4

Cuadro 4.2: Valores del DIC de cada escenario para el Algoritmo BUGS

Resultados del error cuadrático medio para distintos tamaños de muestra

El objetivo de este acápite es afianzar el empleo del método bayesiano sobre el clásico para la estimación del modelo de regresión spline penalizado. Para esto, se ha evaluado el error cuadrático medio de ambos métodos considerando los tamaños de muestra 25, 50, 100, 150 y 200, tal como se presenta en las figuras 4.15, 4.16, 4.17, 4.18 y 4.19, respectivamente. En cada figura se utiliza el spline cúbico de basa radial para $\sigma^2 = 0.5$ y 1.0 con nodos = 5, 10, 15 y 20. A partir de estas figuras se puede observar lo siguiente:

- Al considerar tamaños de muestra pequeños, 25 y 50, el menor ECM corresponde a la estimación bayesiana.
- Al considerar tamaños de muestra mayores, 100, 150 y 200, los ECM son similares con ambos enfoques para todos los nodos, sobre todo al considerar 10, 15 y 20 nodos.

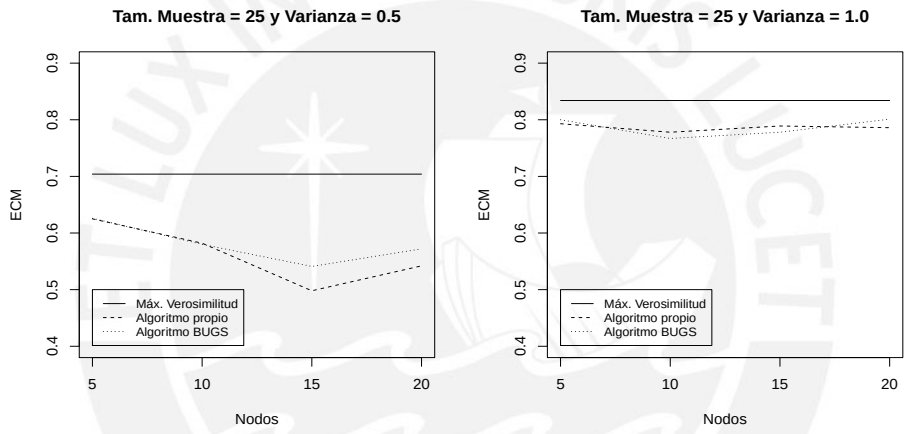

Figura 4.15: ECM de cada método para una muestra de tamaño 25 con $\sigma^2 = 0.5$ y 1.0

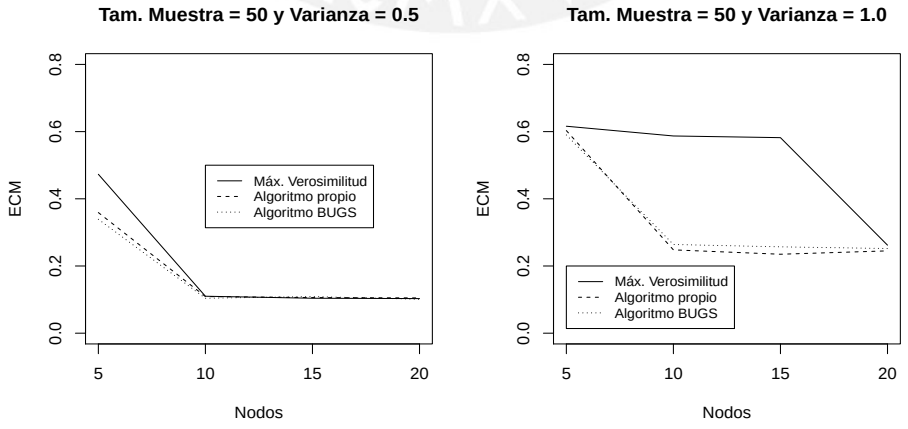

Figura 4.16: ECM de cada método para una muestra de tamaño 50 con $\sigma^2 = 0.5$ y 1.0

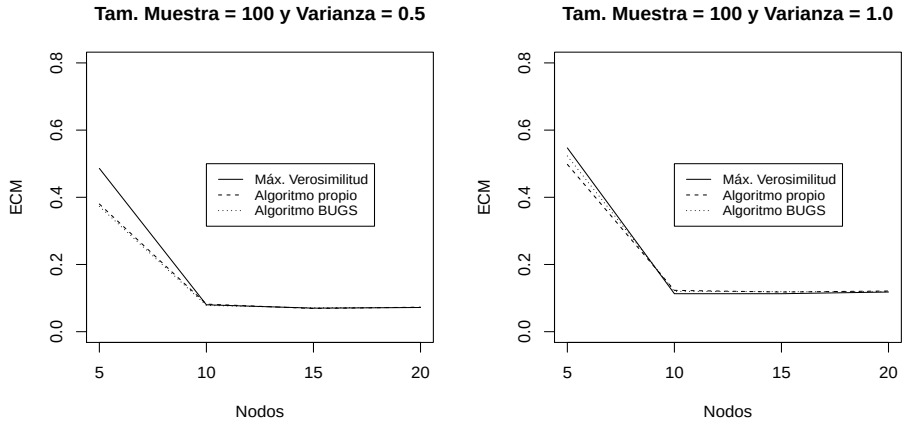

Figura 4.17: ECM de cada método para una muestra de tamaño 100 con $\sigma^2 = 0.5$ y 1.0

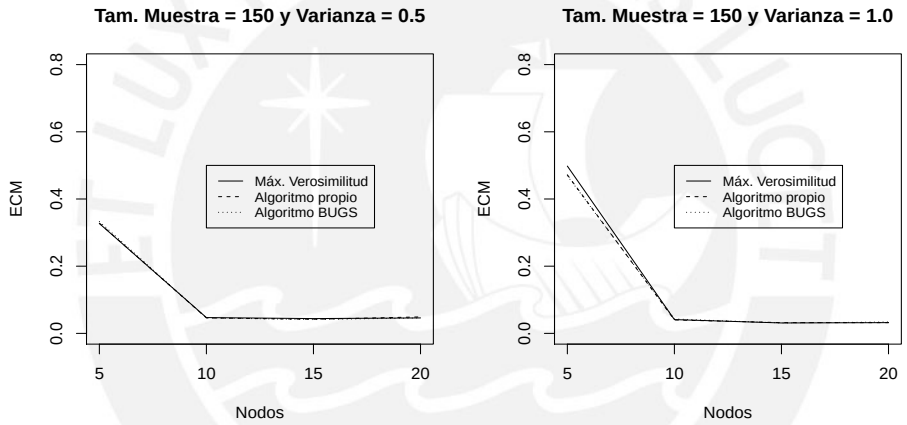

Figura 4.18: ECM de cada método para una muestra de tamaño 150 con $\sigma^2 = 0.5$ y 1.0

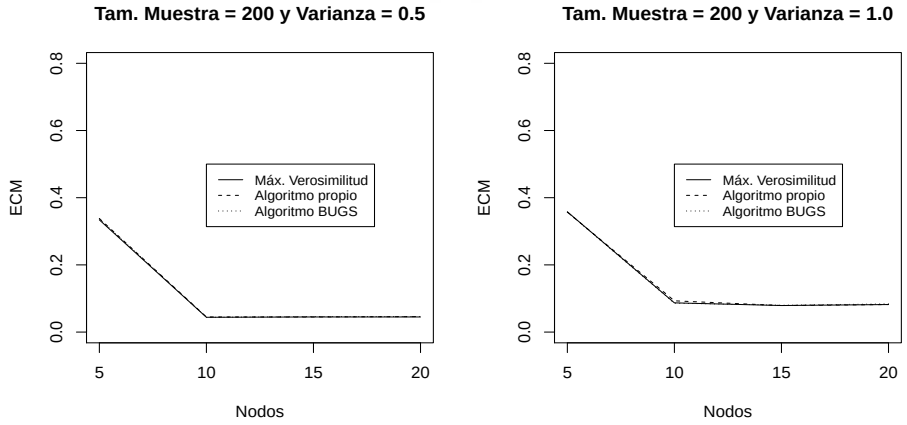

Figura 4.19: ECM de cada método para una muestra de tamaño 200 con $\sigma^2 = 0.5$ y 1.0

En resumen, en el presente capítulo se ha revisado una comparación de los resultados obtenidos al emplear el enfoque clásico y bayesiano. De los cuadros y gráficos revisados se puede concluir que la estimación bayesiana es la mejor opción para la estimación de los parámetros del modelo de regresión spline penalizado debido a que con este enfoque se obtuvo los mejores ajustes a los datos simulados y los menores valores para el ECM en la mayoría de escenarios analizados, sobre todo al considerar tamaños de muestra pequeños. En cuanto a los métodos bayesianos implementados, el programa en código BUGS (ver Crainiceanu et al., 2005) y el programa en R (algoritmo propio) consiguen estimaciones similares; hay que tener en cuenta que, respecto al tema computacional, los tiempos de ejecución de la simulación son menores con el algoritmo BUGS (ver apéndice B). Respecto al spline cúbico de base radial, este logró conseguir ajustes adecuados y de manera suavizada a la curva simulada. Con relación a la cantidad de nodos, se obtuvo menores ECM al considerar 10 y 15 nodos, teniendo la mejora significativa con 10 nodos.

Capítulo 5

Aplicación: Estimación del tiempo en cola y cálculo de la capacidad de personal en agencias bancarias

5.1. Objetivo

Los objetivos del presente caso de aplicación son los siguientes:

- Estimar el tiempo de espera promedio en cola de los clientes en agencias bancarias.
- Calcular la capacidad de personal (servidores) bajo diversas metas de espera.

5.2. Descripción del caso

La importancia de estudiar el presente caso recae en la necesidad que tienen diversas empresas (bancos, aseguradoras, operadores telefónicos, super mercados, etc.) por mejorar los niveles de servicio mediante el cálculo del *staffing* óptimo. Para lograr esto, generalmente es necesario contar con herramientas informáticas de predicción o simulación, donde los costos de dichos programas y la capacitación de personal suelen ser limitantes. Por este motivo, el modelo de regresión spline penalizado pretende ser una alternativa para lograr dicho objetivo.

El modelo en estudio se aplicará a los datos reales del servicio al cliente en las agencias bancarias, siendo estas sujetos de estudio y no precisamente los clientes. La variable a estimar es el tiempo de espera promedio en cola (tespera); las variables predictivas son la cantidad de clientes que ingresan a la agencia (arribos), el tiempo de servicio promedio en la atención de clientes (tservicio), y la capacidad de personal que atiende a los clientes (servidores).

Las variables predictivas se combinarán para formar una variable que represente al tiempo total invertido en el servicio de atención de clientes por unidad de servidor, a la cual se le denominará *ttotservicioservidor*, teniendo la siguiente forma:

$$\text{ttotservicioservidor} = \frac{\text{arribos} \times \text{tservicio}}{\text{servidores}}$$

La necesidad de contar con esta variable se revisará en los siguientes acápites. Esta transformación permitirá estudiar el comportamiento de sólo dos variables con el fin de tener una mejor visualización gráfica al aplicar el modelo de regresión spline penalizado.

Las variables son recopiladas mediante un sistema de tickets que los clientes solicitan al ingresar a una agencia bancaria. Los clientes se atienden por orden de llegada, es decir, el primero que entra es el primero en salir de la cola. Para efectos del presente caso de aplicación

se considera como una observación de la muestra al registro diario del total de arribos, la cantidad de servidores, el tiempo de servicio promedio y el tiempo en cola promedio.

5.3. Análisis preliminar de los datos

Antes de implementar el modelo al presente caso se presentarán unas gráficas descriptivas del conjunto de datos. En la Figura 5.1 se presenta una matriz de diagramas de dispersión entre las covariables y la variable de respuesta *tespera*, donde se puede revisar algunas relaciones existentes entre variables.

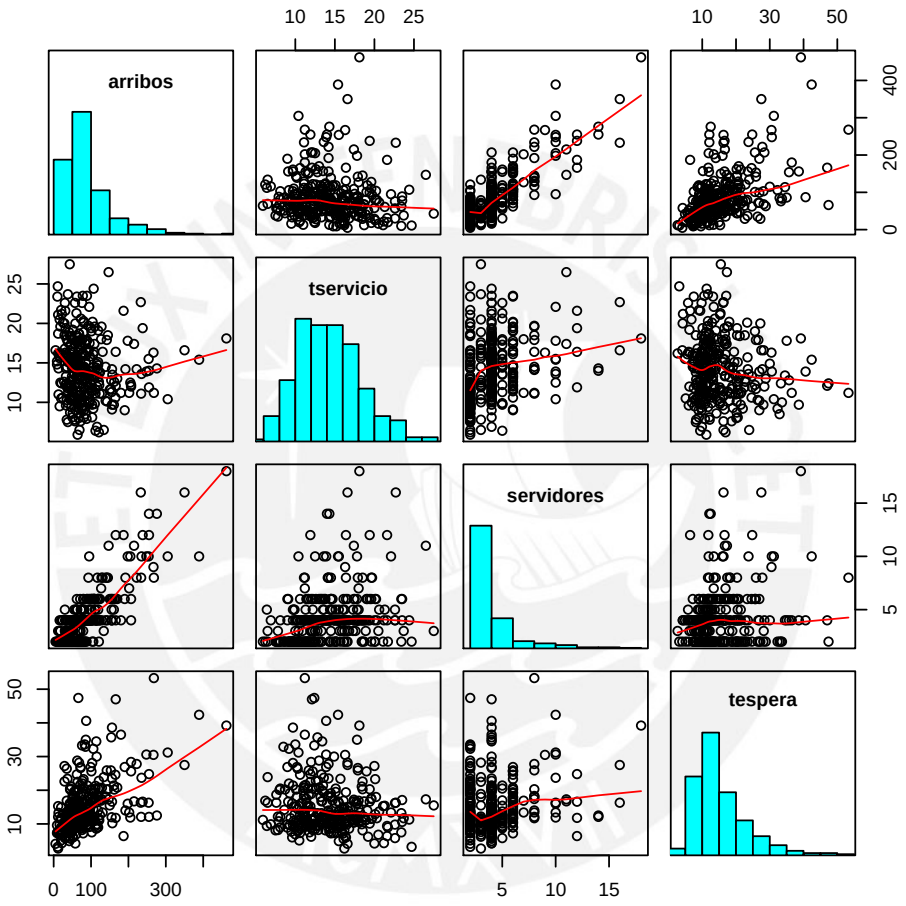

Figura 5.1: Gráfico de dispersión de variables arribos, tservicio, servidores y tespera

De la Figura 5.1 se observa lo siguiente:

- Respecto a la variable *arribos* con la variable de respuesta *tespera*:
  - Se puede apreciar una ligera relación creciente entre la cantidad de arribos y el tiempo de espera.
- Respecto a las variables *tservicio* y *servidores* con la variable de respuesta *tespera*:
  - No existe una relación clara entre ambas variables, es decir, la dispersión es alta.

En figura anterior no se apreciaba una relación clara entre las variables dado que se estaba mostrando el comportamiento de las variables de diversas agencias bancarias, donde la dispersión de los *arribos*, *tservicio* y *servidores* es alta. Por este motivo, a fin de visualizar una mejor relación entre las variables, se van considerar los *arribos* y *tespera* correspondientes a los siguientes escenarios:

- tservicio: 10, 12 y 14 minutos
- servidores: 1, 2 y 3 personas

Cabe mencionar que en la realidad no es posible fijar los tiempos de servicio de cada agencia ya que dependen de diversas variables como el tipo de operación, la zona, los trabajadores, entre otras. Para efectos de la presente aplicación se agruparán los valores del tiempo de servicio promedio al valor entero más cercano. Por ejemplo, si se tuviese un tiempo de servicio promedio de 11 minutos y 30 segundos, este se agruparía a 12 minutos, mientras que un tiempo de 11 minutos y 29 segundos se agruparía a 11 minutos, y así en general. La Figura 5.2 presenta el gráfico de dispersión para los escenarios descritos.

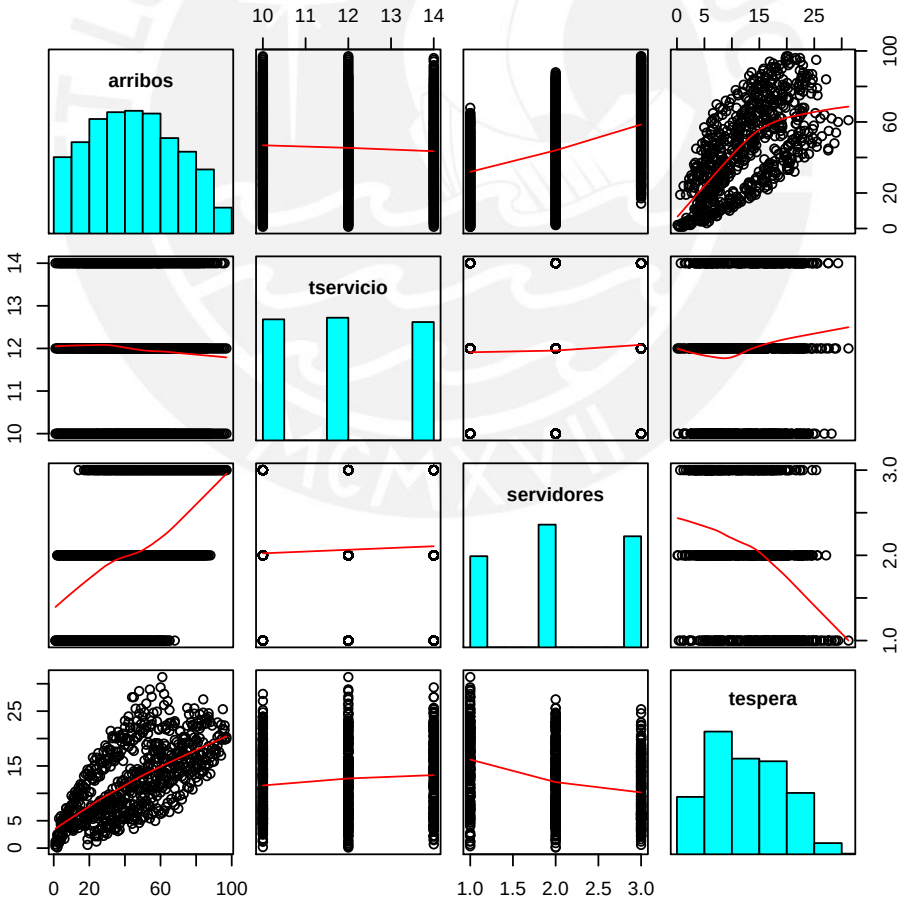

Figura 5.2: Gráfico de dispersión de variables arribos, tservicio (10, 12 y 14), servidores (1, 2 y 3) y tespera

De la Figura 5.2 se observa lo siguiente:

- Respecto a la variable *arribos* con la variable de respuesta *tespera*:
  - Se aprecia una relación creciente más clara: a más arribos se presentan mayores tiempos de espera.
- Respecto a las variables *tservicio* y *servidores* con la variable de respuesta *tespera*:
  - Los tiempos de espera no dependen del tiempo de servicio ni de la cantidad de servidores ya que en todos los escenarios evaluados se tienen tiempos altos y bajos.

A partir de la Figura 5.2 se procede a considerar la variable transformada *ttotservicioservidor* para evaluar su relación con la variable de respuesta *tespera*. Esta relación se presenta en la Figura 5.3 y evidencia la existencia de una alta correlación entre ambas variables, es decir, a mayores valores de *ttotservicioservidor* el tiempo de espera promedio será más alto.

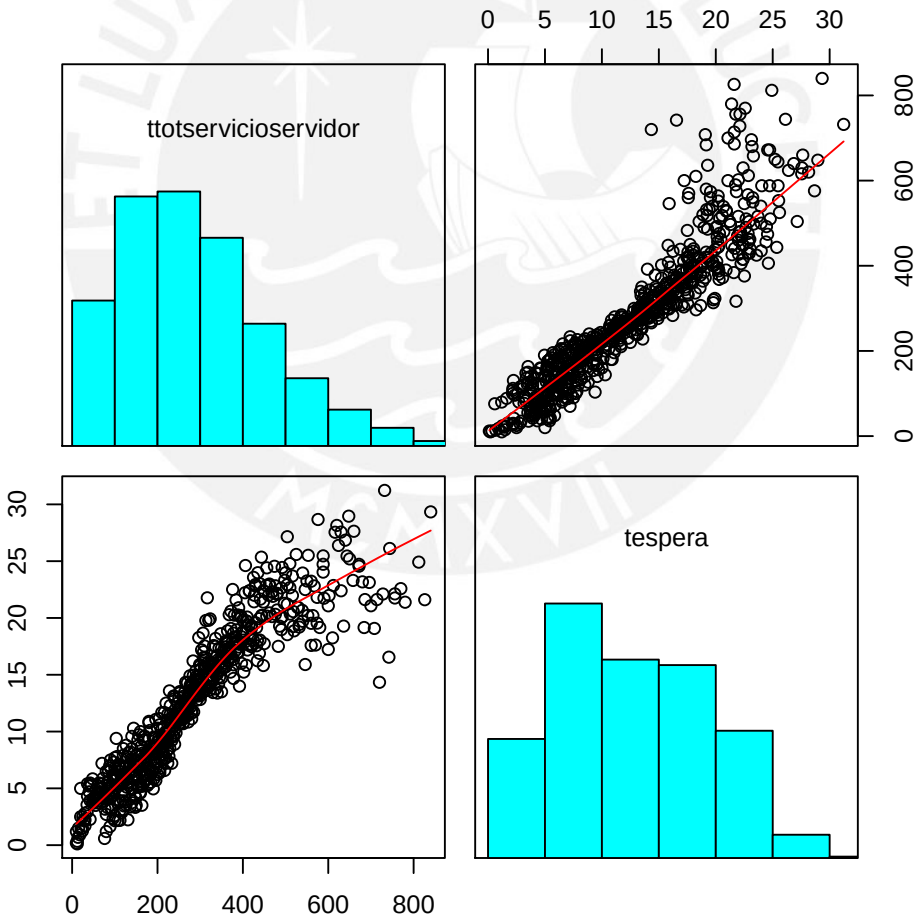

Figura 5.3: Gráfico de dispersión de variables ttotservicioservidor y tespera

Identificar la correlación entre las variables *ttotservicioservidor* y *tespera* implica decir que el tiempo de espera promedio en cola depende de los arribos, del tiempo de servicio promedo y de la cantidad de servidores que brindan atención a clientes. Sin embargo, es complicado visualizar esta relación al considerar las variables predictivas de manera independiente (Figura 5.2).

Una forma adicional de revisar la relación entre las variables se puede observar al realizar las gráficas de dispersión entre los arribos y los tiempos de espera promedio manteniendo fijas las variables tiempo de servicio promedio y el número de servidores. Esto quiere decir que por cada combinación de las variables *tservicio* y *servidores* se tendría un gráfico entre las variables *arribos* y *tespera*, tal como se muestra en la Figura 5.4, donde se puede ver que el tiempo de espera será más alto a medida que hayan menos servidores y que el tiempo de servicio aumente. Por el contrario, se tendrá un menor tiempo de espera mientras el tiempo de servicio sea más bajo y hayan más servidores para la atención. Estas relaciones se aprecian mejor en las Figuras 5.5 y 5.6, donde en la primera se mantiene fijo el *tservicio* y se varían los *servidores* y en la segunda se realiza el caso contrario.

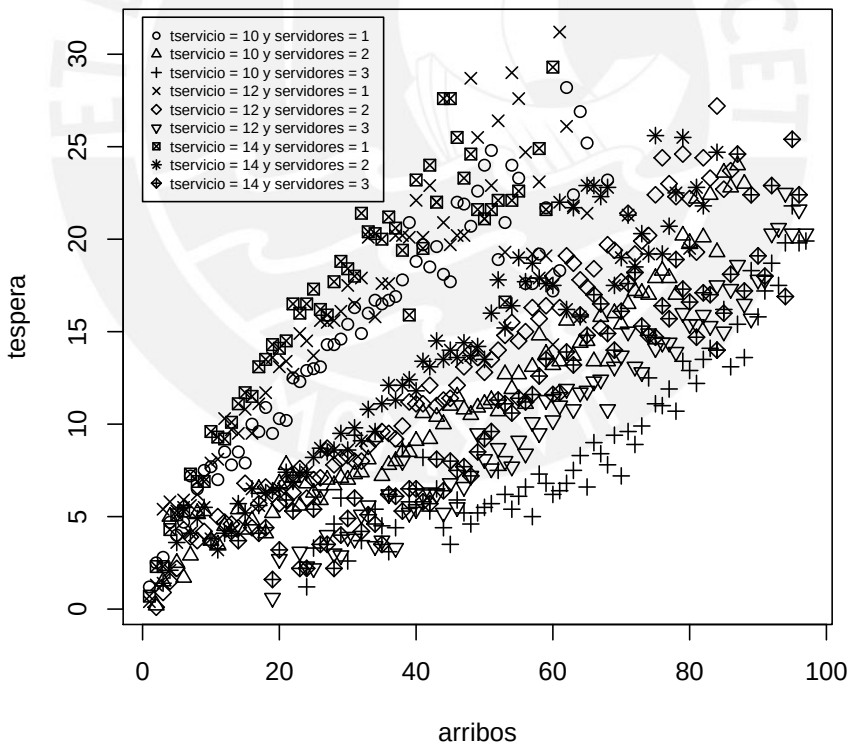

Figura 5.4: Gráfico de dispersión entre las variables *arribos* y *tespera* por cada escenario

Según el análisis realizado, el siguiente paso será implementar el modelo de regresión spline penalizado a fin de poder estimar los tiempos en cola de cada escenario presentado.

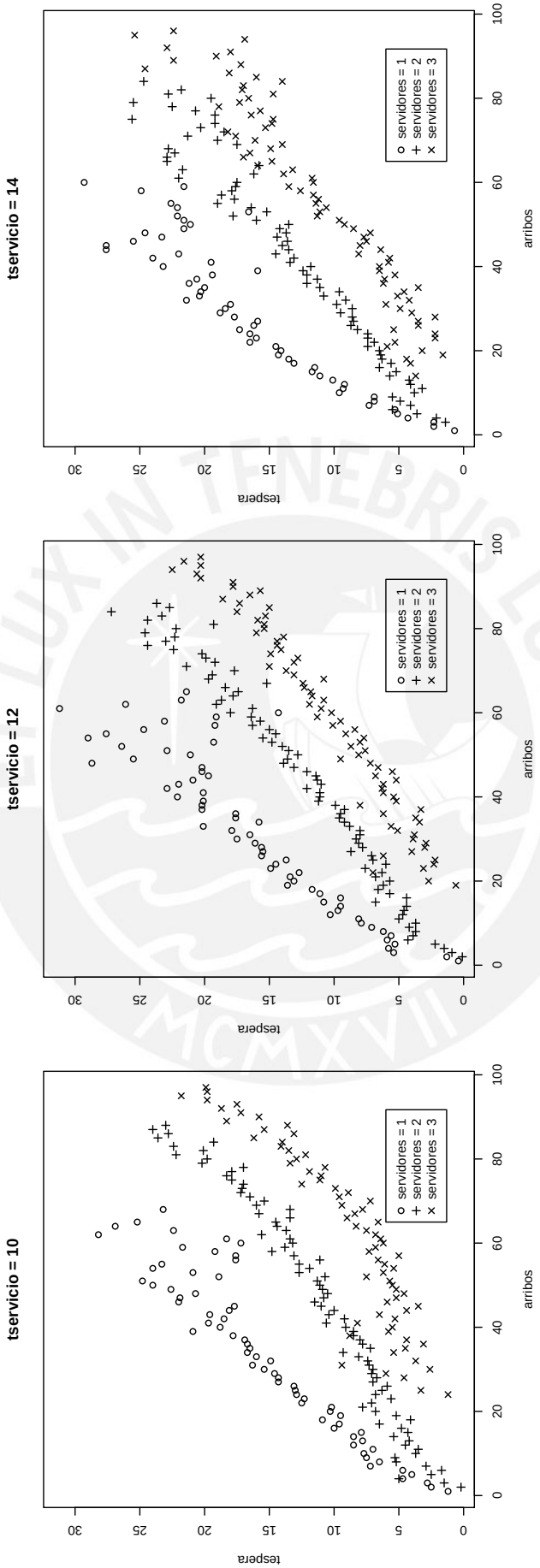

Figura 5.5: Gráfico de dispersión entre las variables *arribos* y *tespera* por cada escenario, manteniendo fijo el *tservicio* y variando los *servidores*

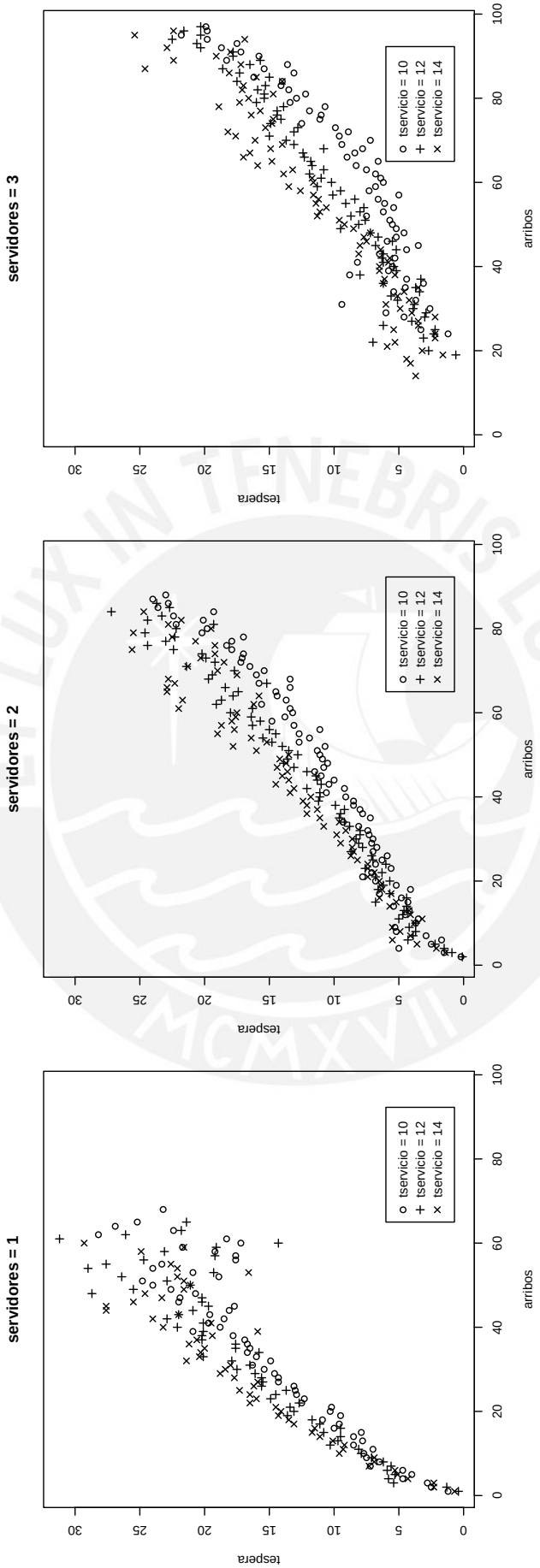

Figura 5.6: Gráfico de dispersión entre las variables *arribos* y *tespera* por cada escenario, manteniendo fijo los *servidores* y variando el *tservicio*

5.4. Modelo de regresión spline penalizado

Como se revisó en el acápite anterior, la relación entre las variables *arribos* y *tespera* se aprecia mejor cuando se considera un escenario con las variables *tservicio* y *servidores* fijas o constantes. Por este motivo, para efectos de la implementación, se considerarán las combinaciones descritas en función a esas variables a fin de mostrar cómo el modelo de regresión spline penalizado consigue ajustes adecuados. Los escenarios son los siguientes:

- tservicio: 10, 12 y 14 minutos
- servidores: 1, 2 y 3 personas

El spline que se considerará es el cúbico de base radial empleado en Crainiceanu et al. (2005), el cual consigue ajustes suavizados según lo revisado en el Capítulo 4. Este spline se representa de la siguiente forma:

$$f(x, \boldsymbol{\theta}) = \beta_0 + \beta_1 x + \sum_{k=1}^{K} b_k |x - \kappa_k|^3,$$

donde $\boldsymbol{\theta} = (\beta_0, \beta_1, b_1, \ldots, b_K)^T$ es el vector de los coeficientes de la regresión y $\kappa_1 < \kappa_2 < \ldots < \kappa_K$ son los nodos.

En el caso de los nodos se considerarán diferentes cantidades para realizar el ajuste de los modelos de regresión spline penalizada, teniendo los siguientes casos:

- Nodos: 4
  - K = 5, 10, 15 y 20.

Adicionalmente del modelo spline penalizado se considerará una estimación bayesiana del modelo de regresión polinómica de grado 3 a fines de comparar qué modelo se ajusta mejor a los datos. Como medida de bondad ajuste del modelo se calculará el error cuadrático medio respecto a los datos reales:

$$\widehat{ECM} = \frac{1}{n} \sum_{i=1}^{n} (y - \hat{y}_i)^2$$

donde n es el tamaño de la muestra de los valores que van a ser estimados. Estos tamaños de muestra de cada escenario pueden ser diferentes debido a la disponibilidad de los datos.

En cada modelo se implementará la estimación vía inferencia bayesiana, considerando a la media como estimador puntual de los parámetros. Además, se compararán los valores de los DIC asociados para seleccionar el mejor el modelo. Respecto a los aspectos computacionales, se generarán 200,000 simulaciones; adicionalmente, se está considerando un periodo de calentamiento (*burn in*) de 50,000 y un salto entre iteraciones (*thin*) de 10, teniendo 15,000 iteraciones para preparar estadísticas y diversos análisis.

El detalle del código de la implementación del modelo de regresión spline penalizado se puede ver en el apéndice E. Los códigos del modelo de regresión cúbico en los programas **R** y **WinBUGS** se presentan en los apéndices F y G, respectivamente.

Estimación y comparación de modelos de regresión spline penalizados

En la Figura 5.7 se presenta un gráfico de líneas con los errores cuadráticos medio para cada escenario bajo diferentes cantidades de nodos. Se puede concluir que aumentar la cantidad de nodos no influye en una mejora significativa del error cuadrático medio en los datos empleados, salvo en el caso de tener un *tservicio* de 14 minutos y *servidores* igual a 1. Por lo tanto se puede decir que, en términos generales, los datos presentan una dispersión baja.

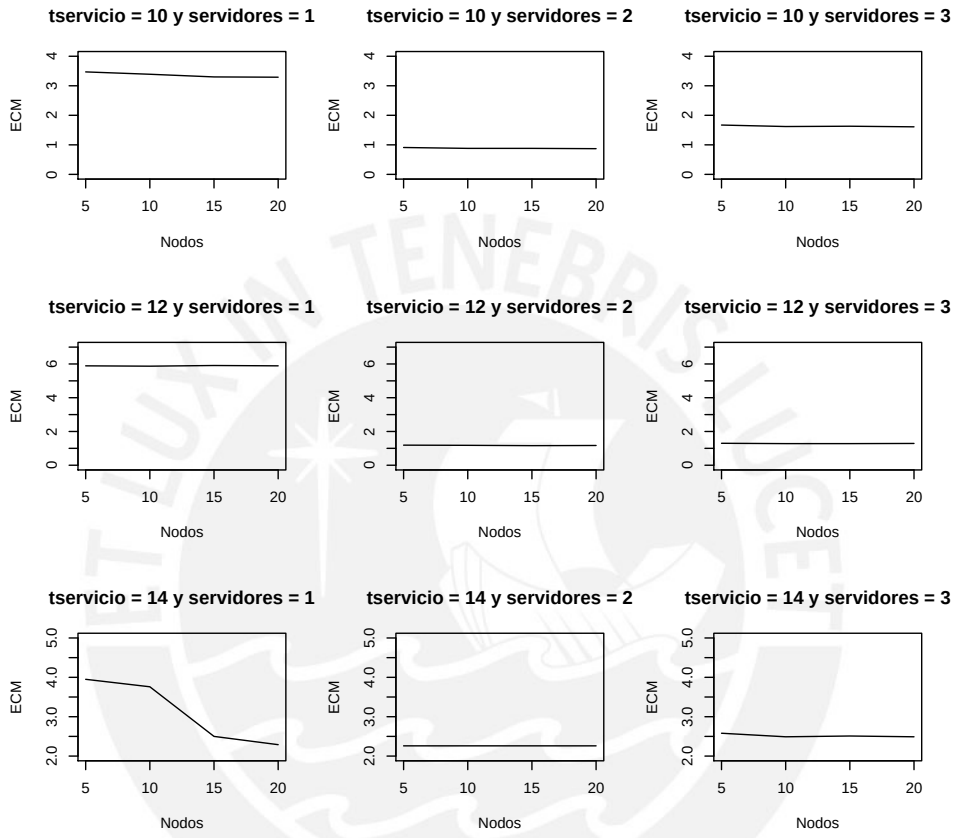

Figura 5.7: Errores cuadráticos medio de cada escenario bajo diferentes cantidades de nodos

En la Figura 5.8 se presenta un gráfico de líneas con el valor de los DIC para cada escenario bajo diferentes cantidades de nodos. Se puede concluir que en casi todos los escenarios aumentar la cantidad de nodos no influye en una mejora significativa del DIC en los datos empleados.

En las figuras 5.9, 5.10, 5.11, 5.12, 5.13, 5.14, 5.15, 5.16, 5.17 se pueden observar ajustes adecuados y similares del modelo de regresión spline penalizado en todos los escenarios considerados, salvo en el caso de tener un *tservicio*=14 y *servidores*=1 debido a que los datos presentan mayor dispersión. En conclusión, se comprueba gráficamente que no existen diferencias significativas en el ajuste al considerar $K = 5$, 10, 15 ó 20 nodos, tal como se observó en la Figura 5.7.

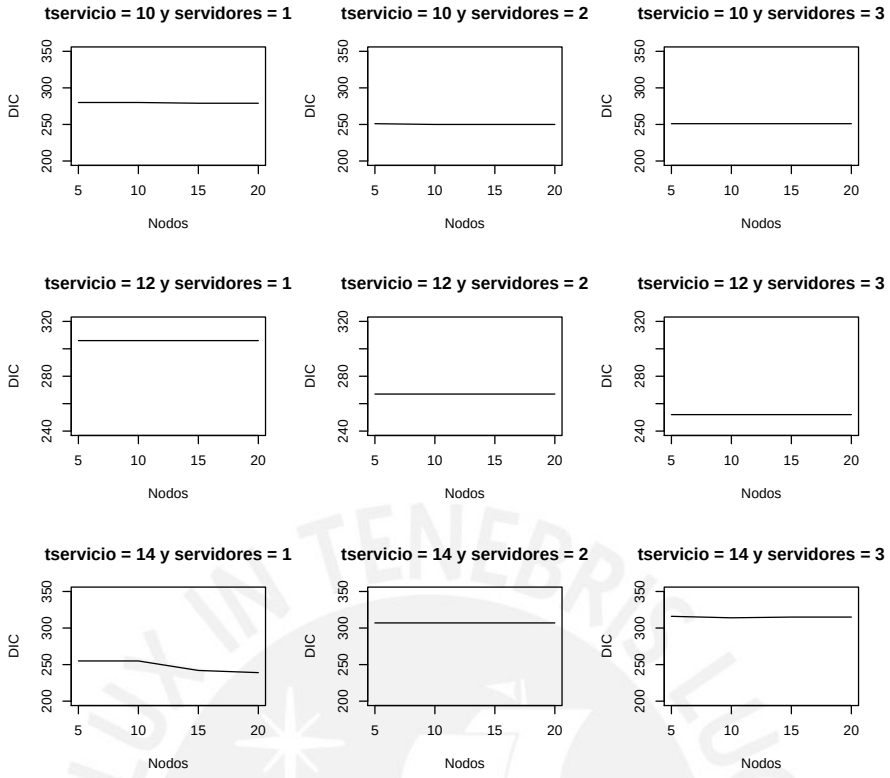

Figura 5.8: DIC de cada escenario bajo diferentes cantidades de nodos

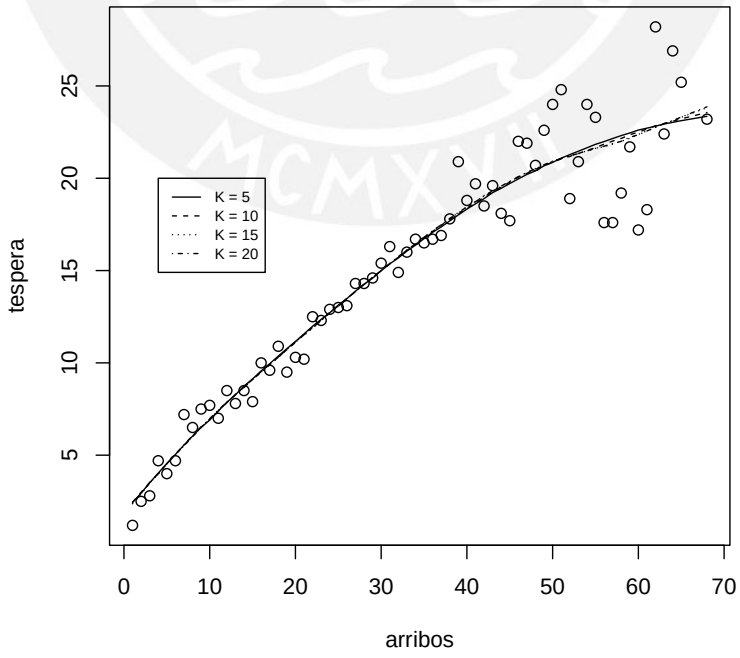

Figura 5.9: Ajuste del modelo considerando *tservicio*=10 y *servidores*=1 para diversos K nodos.

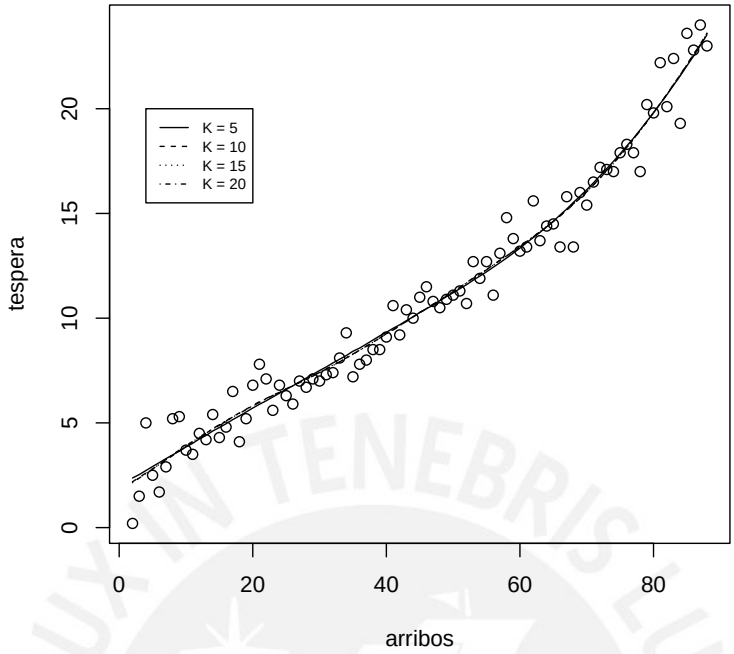

Figura 5.10: Ajuste del modelo considerando *tservicio*=10 y *servidores*=2 para diversos K nodos.

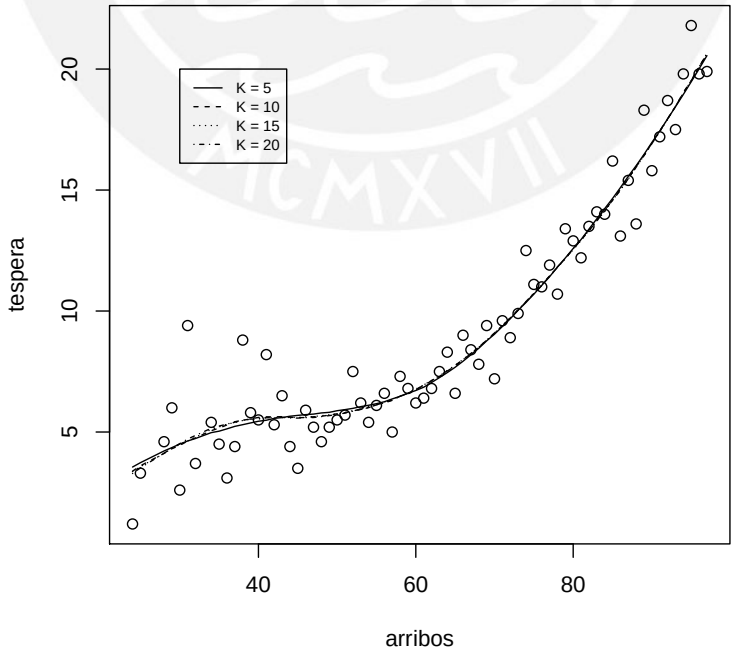

Figura 5.11: Ajuste del modelo considerando *tservicio*=10 y *servidores*=3 para diversos K nodos.

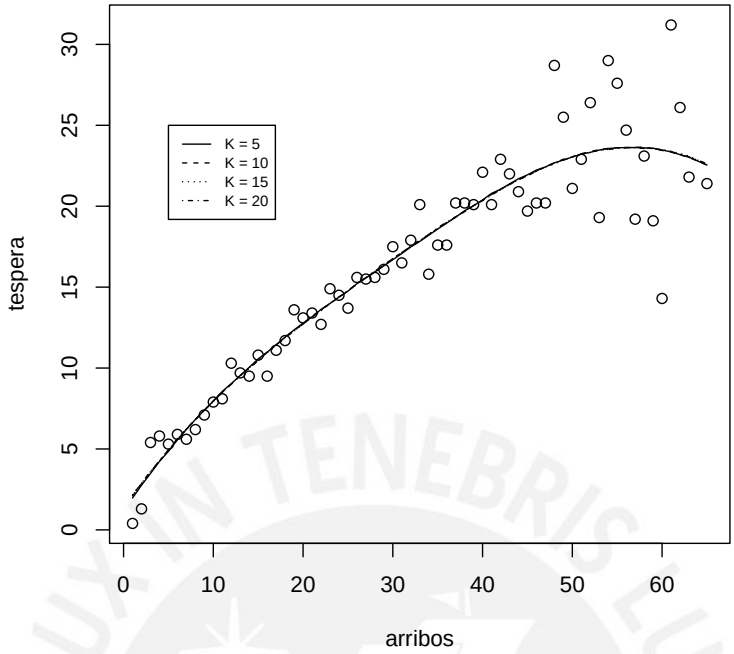

Figura 5.12: Ajuste del modelo considerando *tservicio*=12 y *servidores*=1 para diversos K nodos.

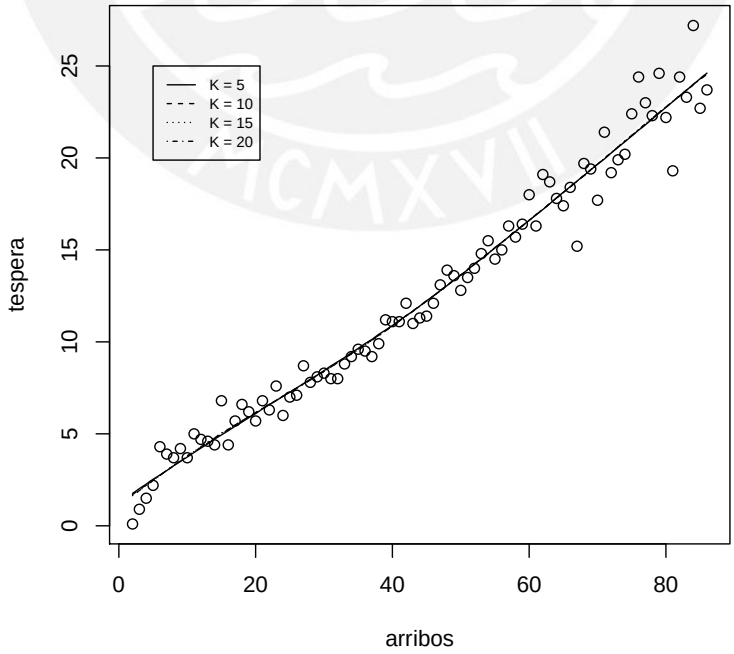

Figura 5.13: Ajuste del modelo considerando *tservicio*=12 y *servidores*=2 para diversos K nodos.

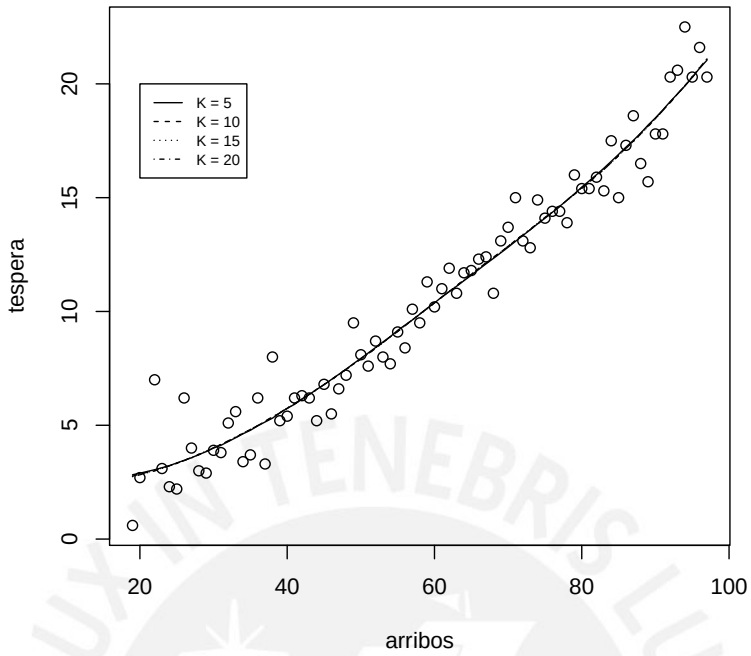

Figura 5.14: Ajuste del modelo considerando *tservicio*=12 y *servidores*=3 para diversos K nodos.

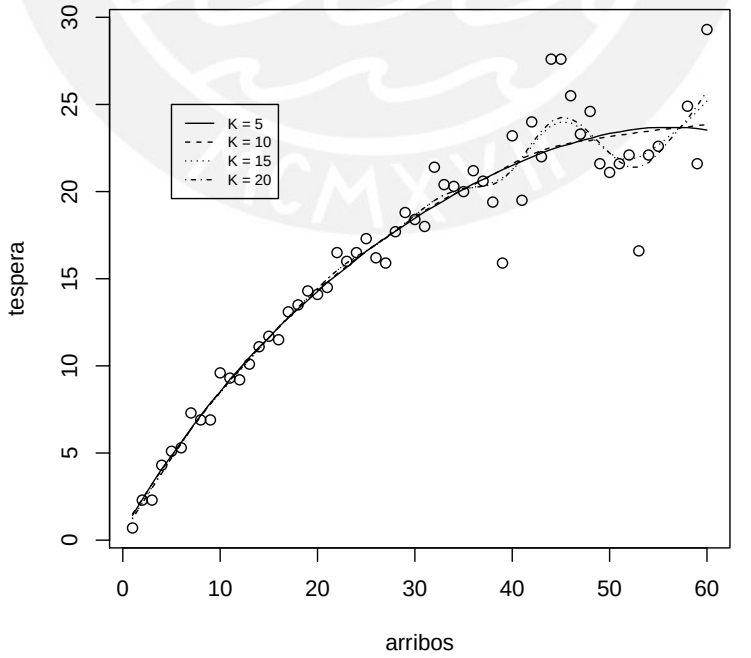

Figura 5.15: Ajuste del modelo considerando *tservicio*=14 y *servidores*=1 para diversos K nodos.

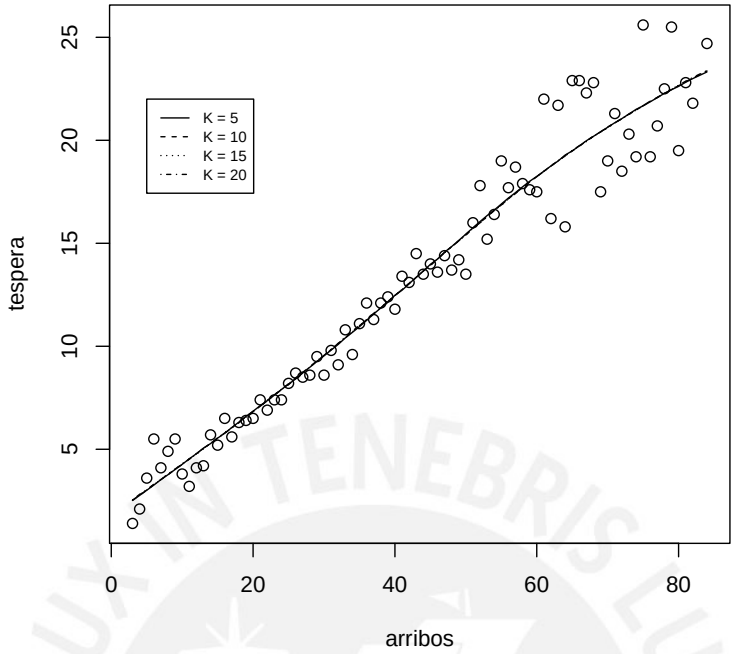

Figura 5.16: Ajuste del modelo considerando *tservicio*=14 y *servidores*=2 para diversos K nodos.

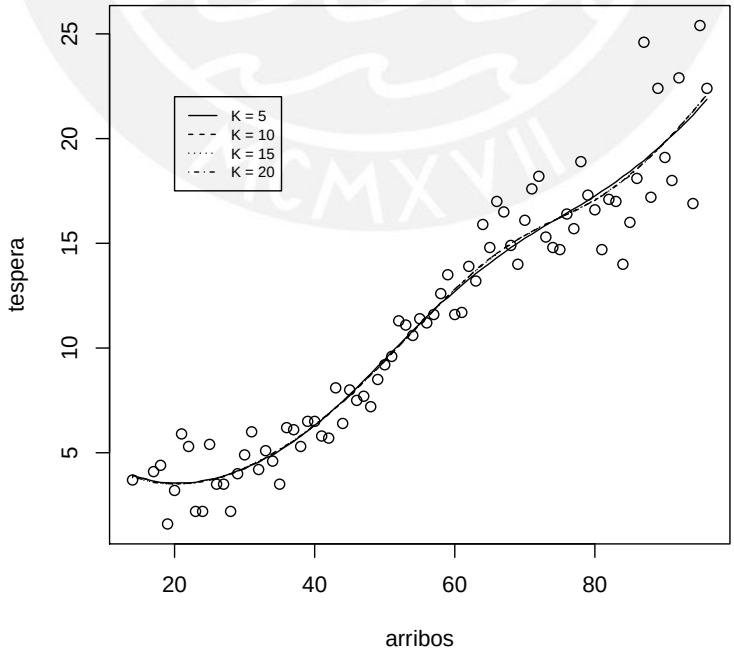

Figura 5.17: Ajuste del modelo considerando *tservicio*=14 y *servidores*=3 para diversos K nodos.

Estimación y comparación con el modelo de regresión polinómica de grado 3

El objetivo de realizar esta comparación es saber si el modelo de regresión spline penalizado se ajusta mejor a los datos respecto a otros modelos tradicionales. En el caso del modelo spline se considerarán los de 10 y 15 nodos (en el acápite anterior se comprobó que las estimaciones eran similares). Respecto al modelo en comparación, se considerará el modelo polinómico de grado 3 dado que el modelo spline considera una base cúbica. En ambos se evaluarán el error cuadrático medio (ECM) y el criterio de información de desvío (DIC).

En la figura 5.18 se observa que en la mayoría de casos los ECM del modelo de regresión spline penalizado (con 10 y 15 nodos) son menores que el del modelo de regresión polinómico de grado 3. De igual manera, se puede observar en la figura 5.19 que en la mayoría de casos los modelos spline poseen un menor valor del DIC, lo que revela un mejor ajuste.

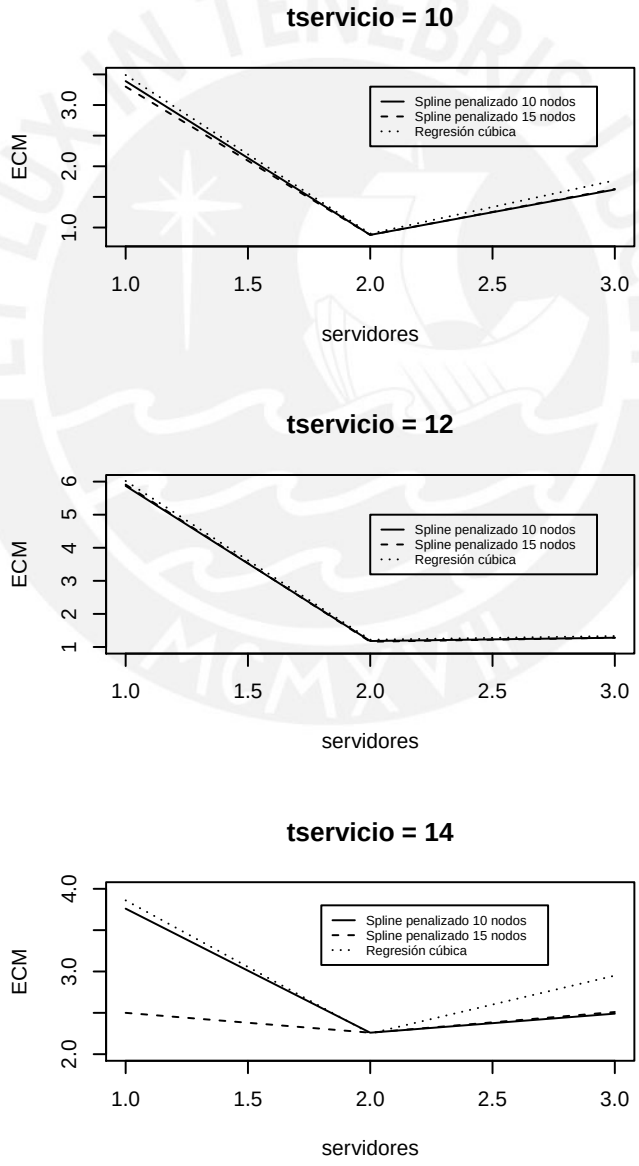

Figura 5.18: ECM para ambos modelos en función de las variables *tservicio* y *servidores*.

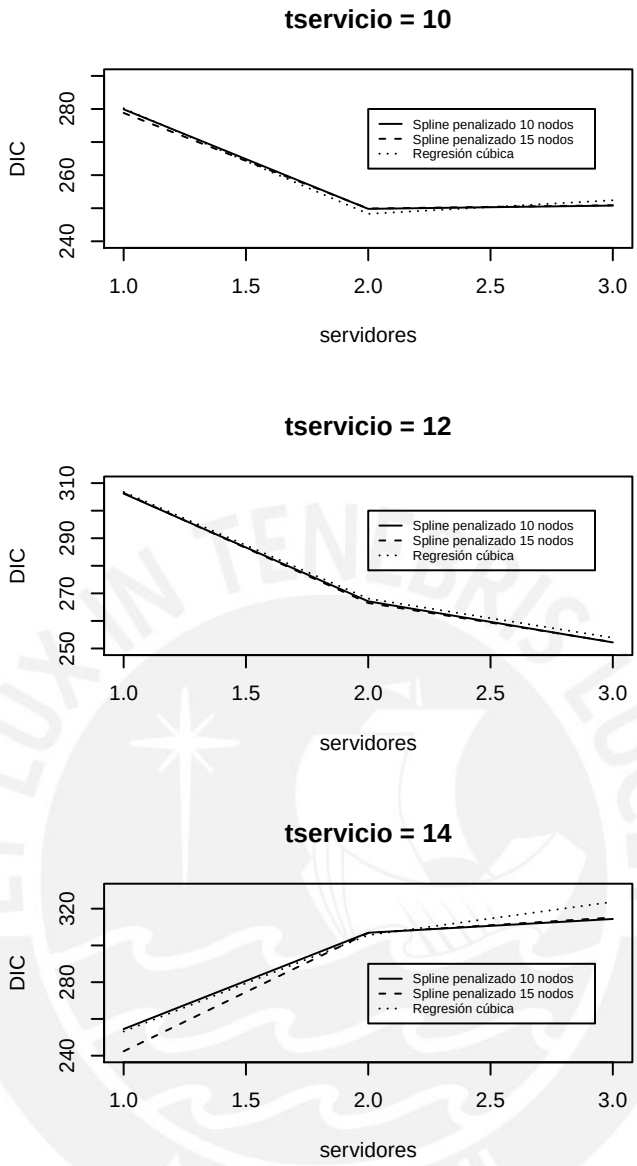

Figura 5.19: DIC para ambos modelos en función de las variables *tservicio* y *servidores*.

De ambos gráficos se puede concluir que el modelo de regresión spline penalizado se ajusta mejor a los datos de la presente aplicación. Adicionalmente, en la figura 5.20 se puede observar el ajuste de cada modelo propuesto a los datos para el escenario *tservicio*=14 y *servidores*=3, siendo el modelo spline el que mejor sigue el comportamiento de los datos.

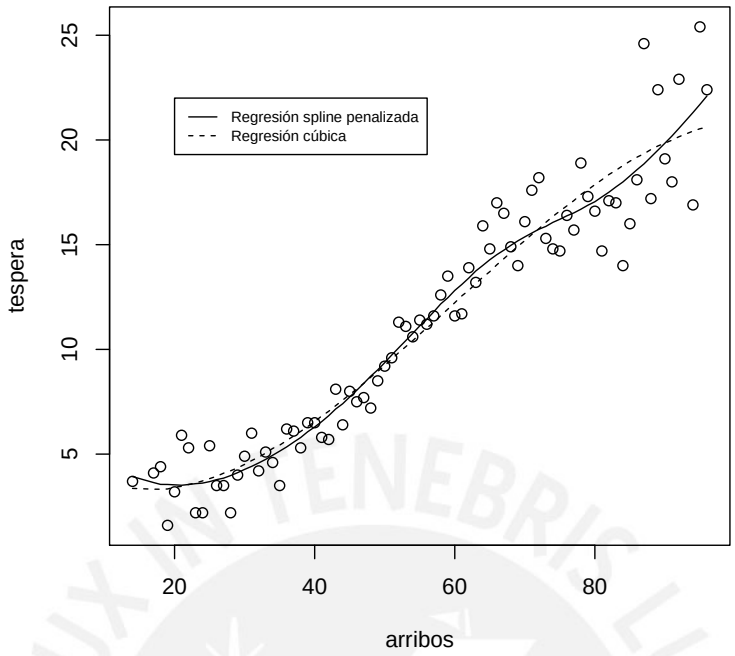

Figura 5.20: Ajuste de los modelos propuestos considerando el escenario *tservicio*=14 y *servidores*=3.

Estimación de la cantidad necesaria de servidores para diversas metas de espera

En el presente acápite se revisará cómo se puede emplear el modelo de regresión spline penalizado para calcular la cantidad de servidores necesarios en las agencias bancarias para cumplir con diversas metas de espera. Esto es importante debido a que día a día los bancos, y en general las empresas de servicio, buscan mejorar los niveles de atención y satisfacción de los clientes con el fin de fidelizarlos, atraer nuevos clientes, generar nuevas oportunidades de venta, entre otros beneficios más. Por este motivo, es necesario reducir los tiempos de espera en cola dado que, por lo general, influyen directamente en el comportamiento de los clientes. Para esto se analizará el comportamiento de 3 agencias bancarias que presentan las siguientes variables:

Agencia	Arribos	Tservicio	Servidores
A	30	14	1
B	40	14	1
C	50	14	1

Cuadro 5.1: Variables de arribos, tiempos de servicio y servidores de 3 agencias bancarias.

Las metas de tiempos de espera de clientes que se evaluarán alcanzar son las siguientes:

	Meta 1	Meta 2	Meta 3
Tiempo de espera (minutos)	15	10	5

Cuadro 5.2: Metas de espera definidas.

El primer paso para estimar la cantidad requerida de servidores es hallar la curva descrita por cada modelo de regresión spline para las variables que presenta cada agencia a fin de estimar el tiempo de espera. En este caso se necesitará el modelo de regresión spline para un $tservicio = 14$ minutos y $servidores = 1$. Según lo obtenido en los acápites anteriores, la estimación de los tiempos de espera de cada agencia sería la siguiente:

Agencia	Arribos	Tservicio	Servidores	Tespera
A	30	14	1	18.6
B	40	14	1	21.6
C	50	14	1	23.2

Cuadro 5.3: Estimación del tiempo de espera según las variables de cada agencia bancaria.

Como siguiente paso, dado que el objetivo es reducir el tiempo de espera incrementando el número de servidores, es necesario conocer las formas de las curvas de los modelos de regresión spline penalizados que se generan al incrementar la variable *servidores*. Dichas curvas, presentadas en la figura 5.21, permitirán saber a cuánto disminuiría el tiempo de espera.

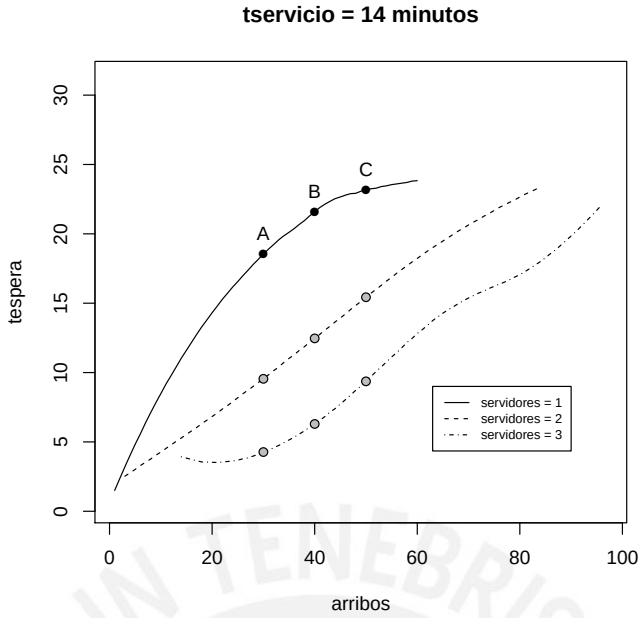

Figura 5.21: Modelos de regresión spline penalizada para un *tservicio*=14 y *servidores*=1, 2 y 3.

El siguiente cuadro presenta el valor alcanzado en el tiempo de espera al incrementar el número de servidores:

Agencia	Arribos	Tservicio	Servidores	Tespera
A	30	14	1	18.6
	30	14	2	9.6
	30	14	3	4.3
B	40	14	1	21.6
	40	14	2	12.5
	40	14	3	6.3
C	50	14	1	23.2
	50	14	2	15.4
	50	14	3	9.4

Cuadro 5.4: Estimación del tiempo de espera al incrementar el número de servidores.

Finalmente, a partir de los resultados del cuadro 5.4 es posible saber con cuántos servidores sería necesario contar en cada agencia bancaria a fin de cumplir con las siguientes metas de espera propuestas:

Agencia	Meta 15 min.	Meta 10 min.	Meta 5 min.
A	2	2	3
B	2	3	>3
C	3	3	>3

Cuadro 5.5: Número de servidores necesarios en cada agencia bancaria según meta de espera.

En resumen, se ha logrado cumplir los objetivos propuestos de la presente aplicación. Respecto al primero, se ha comprobado que es posible emplear el modelo de regresión spline penalizado para estimar adecuadamente los tiempos de espera promedio en función a la cantidad de arribos, al tiempo de servicio promedio y al número de servidores de una agencia bancaria, siempre y cuando se evalúe un sistema con disciplina o lógica de cola por orden de llegada. Esto es importante debido a que se logra aplicar un modelo de regresión estadístico en el campo de la teoría de colas, la cual es una rama de la investigación de operaciones. Respecto al segundo objetivo, se verifica que la estimación de los tiempos de espera permite calcular la cantidad de servidores que serían necesarios para cumplir con diversas metas de espera que se proponga la entidad bancaria, lo cual cobra relevancia porque se puede saber cuántas personas se deberían contratar para cada agencia. Finalmente, hay que destacar que el modelo de regresión spline resulta ser flexible dado que es posible implementar rutinas para varias agencias en la medida que se tengan la data de las variables, a diferencia de la estimación en programas especializados en simulación de sistemas y teoría de colas, donde generalmente se evalúa cada caso de manera individual y, además, la adquisición de dicho paquete computacional suele ser costosa.

Capítulo 6

Conclusiones

6.1. Conclusiones

A partir del análisis realizado en el estudio de simulación y en la aplicación se puede concluir lo siguiente:

6.1.1. Conclusiones del Estudio de Simulación

En base a los resultados obtenidos en el estudio de simulación se pueden establecer las siguientes conclusiones:

- El modelo lineal mixto permite estimar los parámetros del modelo de regresión spline penalizado, tanto para el enfoque clásico como el bayesiano.
- Respecto a los métodos de estimación considerados, en la mayoría de escenarios la estimación bayesiana consigue ajustes más adecuados que la estimación clásica.
- Para tamaños de muestra pequeños, el menor ECM lo presenta la estimación bayesiana. Para tamaños de muestra grandes, los ECM son similares utilizando ambos enfoques.
- Al considerar una dispersión relativamente alta en los datos, la estimación bayesiana consigue menores ECM que la estimación clásica por máxima verosimilitud.
- Entre los métodos bayesianos desarrollados, el algoritmo implementado en código BUGS y el algoritmo implementado en R (desarrollo propio) consiguen estimaciones similares; sin embargo, respecto al tema computacional, el algoritmo BUGS destaca debido a que sus tiempos de ejecución de la simulación son mucho menores.
- Sobre los estimadores puntuales en la inferencia bayesiana, al emplear la media o la mediana se consiguen valores similares del error cuadrático medio.
- Respecto al spline cúbico de base radial, este consigue ajustar adecuadamente los datos e incluso lo hace de manera suavizada.
- Incrementar la cantidad de nodos en el modelo de regresión spline penalizado conlleva a una mejora en el ajuste de la curva. Esta mejora se consigue hasta cierta cantidad de nodos debido al efecto penalizador del spline.

6.1.2. Conclusiones de la Aplicación

En relación con los resultados hallados en la aplicación a los tiempos en cola de una agencia bancaria se puede concluir lo siguiente:

- Existe una correlación alta entre los arribos y los tiempos de espera en cola de las agencias bancarias al mantener fijas las variables de tiempo de servicio promedio y la cantidad de servidores, siempre y cuando se evalúe un sistema con disciplina o lógica de colas por orden de llegada.
- El modelo de regresión spline penalizado ajustó adecuadamente la relación entre los arribos y los tiempos de espera promedio de las agencias bancarias, siempre y cuando se fijen el tiempo de servicio promedio y el número de servidores. Esto quiere decir que para cada agencia, con sus respectivas variables, es posible definir una curva que permita estimar el tiempo de espera promedio.
- Al comparar los valores del ECM y del DIC de la estimación del modelo de regresión spline penalizado y de la regresión polinómica de grado tres se comprobó que el modelo semiparamétrico resulta ser el más adecuado, tal como se observó gráficamente en el ajuste a los datos.
- Es posible calcular la cantidad de servidores (trabajadores) necesarios para la atención de clientes mediante un algoritmo que elija la curva adecuada para estimar el tiempo de espera promedio de acuerdo a las variables de cada agencia bancaria. Este algoritmo se puede evaluar para diversas metas de espera que la entidad bancaria esté dispuesta a ofrecer a sus clientes.

6.2. Sugerencias para investigaciones futuras

- En el estudio de simulación se consideró el spline cúbico de base radial utilizado en Crainiceanu et al. (2005). Es posible investigar y comparar el ajuste obtenido con otros modelos spline.
- En el estudio de simulación y en la aplicación se han considerado diversas cantidades de nodos K para el modelo de regresión spline penalizado, sin embargo, es posible implementar un método que determine el valor idóneo de K con el fin de no tener que evaluar diferentes cantidades escogidas al azar.
- En la aplicación se han agrupado tres variables en una sola llamada *ttotservicioservidor*, la cual ha sido el punto de partida para ajustar los arribos y los tiempos de espera promedio con el modelo de regresión spline penalizado. Otra forma de analizar esta relación podría darse con un modelo que considere las variables predictivas de manera independiente, sin necesidad de ser transformadas. Entre los diversos modelos que existen podría aplicarse el de coeficientes variantes, el cual permite variar los coeficientes de la regresión según los cambios en los valores de las variables, que en este caso serían el tiempo de servicio promedio y el número de servidores. Además, este modelo se puede estimar a través de un marco de modelos lineales mixtos, tal como se ha hecho con la regresión spline penalizada.

Apéndice A

Modelo Lineal Mixto: BLUP

A.1. Modelo lineal mixto

Generalmente, los modelos lineales mixtos tienen la siguiente forma:

$$\boldsymbol{Y} = \boldsymbol{X\beta} + \boldsymbol{Zb} + \boldsymbol{\varepsilon}, \tag{A.1}$$

donde $\boldsymbol{Y}$ es el vector de los valores de n variables respuesta, $\boldsymbol{X}$ es una matriz $n \times p$ que contiene los valores de las p variables explicativas, $\boldsymbol{\beta}$ es el vector de p parámetros fijos, $\boldsymbol{Z}$ es una matriz $n \times q$ que contiene la especificación de los efectos aleatorios $\boldsymbol{b}$ y describe cómo estos influyen en la variable respuesta, y $\boldsymbol{\varepsilon}$ es un vector de errores. Generalmente se asume normalidad para $\boldsymbol{\varepsilon}$ y para $\boldsymbol{b}$, teniendo la siguiente forma:

$$\begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{\varepsilon} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \boldsymbol{G} & 0 \\ 0 & \boldsymbol{R} \end{bmatrix} \right)$$

donde $\boldsymbol{G} = \sigma_b^2 \boldsymbol{I}$ y $\boldsymbol{R} = \sigma_\varepsilon^2 \boldsymbol{I}$.

A.2. Estimación y Predicción

En esta sección se revisará la estimación de $\boldsymbol{\beta}$, la predicción de $\boldsymbol{b}$ y la estimación de los parámetros en $\boldsymbol{G}$ y $\boldsymbol{R}$.

A.2.1. Estimación de efectos fijos

Según Ruppert, Wand y Carroll (2003), un camino de obtener la estimación de $\boldsymbol{\beta}$ es escribir (A.1) de la siguiente forma:

$$\boldsymbol{Y} = \boldsymbol{X\beta} + \boldsymbol{\varepsilon}^*, \quad \text{donde} \quad \boldsymbol{\varepsilon}^* = \boldsymbol{Zb} + \boldsymbol{\varepsilon}.$$

Esta expresión es un modelo lineal con errores correlacionados, debido a que

$$\text{Cov}(\boldsymbol{\varepsilon}^*) \equiv \boldsymbol{V} = \boldsymbol{ZGZ^T} + \boldsymbol{R}$$

donde el estimador clásico de $\boldsymbol{\beta}$ es:

$$\tilde{\boldsymbol{\beta}} = (\boldsymbol{X^T V^{-1} X})^{-1} \boldsymbol{X^T V^{-1} Y} \tag{A.2}$$

que es una expresión que se designa a veces como mínimos cuadrados generalizados.

A.2.2. Predicción de efectos aleatorios

Los vectores de efectos aleatorios pueden ser predichos vía el método mejor predictor lineal (BLP, por sus sigas en inglés), que es una simplificación común que restringe a los predictores como lineales. Consideremos:

$$\boldsymbol{v} = \boldsymbol{A}\boldsymbol{y} + \boldsymbol{c}$$

para una matriz $\boldsymbol{A}$ y un vector $\boldsymbol{c}$. La solución vía *best linear prediction* (BLP) conduce que:

$$\tilde{\boldsymbol{v}} \equiv \text{BLP}(\boldsymbol{v}) = E(\boldsymbol{v}) + \boldsymbol{C}\boldsymbol{V}^{-1}\{y - E(y)\}, \tag{A.3}$$

donde

$$\boldsymbol{C} \equiv E[\{\boldsymbol{v} - E(\boldsymbol{v})\}\{\boldsymbol{y} - E(\boldsymbol{y})\}^T] \quad \text{y} \quad \boldsymbol{V} \equiv \text{Cov}(y).$$

Para el caso del modelo lineal mixto (A.1), los efectos aleatorios serán predichos vía *best linear prediction* usando (A.3), donde se obtiene:

$$\tilde{\boldsymbol{b}} \equiv \text{BLP}(\boldsymbol{b}) = \boldsymbol{G}\boldsymbol{Z}^{\boldsymbol{T}}\boldsymbol{V}^{-\boldsymbol{1}}(y - \boldsymbol{X}\boldsymbol{\beta}). \tag{A.4}$$

En la práctica $\boldsymbol{\beta}$ podría ser reemplazado por el estimador $\tilde{\boldsymbol{\beta}}$ de la ecuación (A.2). La estimación de los parámetros $\boldsymbol{G}$ y $\boldsymbol{V}$ se verá más adelante.

A.2.3. Mejor Predictor Lineal Insesgado (BLUP)

A través del mejor predictor lineal insesgado (BLUP, por sus siglas en inglés) se puede llegar a unificar los resultados de las dos secciones anteriores. Según Ruppert, Wand y Carroll (2003), para $n \times 1$ vectores arbitrarios $\boldsymbol{s}$ y $\boldsymbol{t}$, la determinación lineal de $\tilde{\boldsymbol{\beta}}$ y $\tilde{\boldsymbol{b}}$ implica minimizar el error de predicción:

$$E\{(\boldsymbol{s}^{\boldsymbol{T}}\boldsymbol{X}\tilde{\boldsymbol{\beta}} + \boldsymbol{t}^{\boldsymbol{T}}\boldsymbol{Z}\tilde{\boldsymbol{b}}) - (\boldsymbol{s}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{t}^{\boldsymbol{T}}\boldsymbol{Z}\boldsymbol{b})\}^2$$

sujeto a la condición de insesgamiento:

$$E(\boldsymbol{s}^{\boldsymbol{T}}\boldsymbol{X}\tilde{\boldsymbol{\beta}} + \boldsymbol{t}^{\boldsymbol{T}}\boldsymbol{Z}\tilde{\boldsymbol{b}}) = (\boldsymbol{s}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{t}^{\boldsymbol{T}}\boldsymbol{Z}\boldsymbol{b})$$

Se demuestra (ver Robinson, 1991) que las soluciones para $\tilde{\boldsymbol{\beta}}$ y $\tilde{\boldsymbol{b}}$ son:

$$\begin{aligned} BLUP(\boldsymbol{\beta}) \equiv \tilde{\boldsymbol{\beta}} &= (\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{V}^{-\boldsymbol{1}}\boldsymbol{X})^{-\boldsymbol{1}}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{V}^{-\boldsymbol{1}}\boldsymbol{Y} \\ BLUP(\boldsymbol{u}) \equiv \tilde{\boldsymbol{b}} &= \boldsymbol{G}\boldsymbol{Z}^{\boldsymbol{T}}\boldsymbol{V}^{-\boldsymbol{1}}(\boldsymbol{Y} - \boldsymbol{X}\tilde{\boldsymbol{\beta}}). \end{aligned} \tag{A.5}$$

Se puede ver que el BLUP para $\boldsymbol{\beta}$ es idéntica a la solución por mínimos cuadrados generalizados (A.2) y que el BLUP para $\boldsymbol{b}$ es el BLP con $\boldsymbol{\beta}$ reemplazado por $BLUP(\tilde{\boldsymbol{\beta}}) = \tilde{\boldsymbol{\beta}}$.

Cabe mencionar que, según Robinson (1991), existen de diversas maneras de obtener soluciones para el BLUP. Una de estas es la justificación de Henderson, conocida como estimador de máxima verosimilitud conjunto (*joint maximum likelihood estimates*, en inglés), donde se asume que:

$$\boldsymbol{Y}|\boldsymbol{b} \sim N(\boldsymbol{X\beta} + \boldsymbol{Zb}, \boldsymbol{R}), \quad \boldsymbol{b} \sim N(\boldsymbol{0}, \boldsymbol{G}),$$

donde la densidad conjunta de Y e b es:

$$\begin{aligned}&(2\pi\sigma^2)^{-\frac{1}{2}n-\frac{1}{2}q}\left(\det\begin{bmatrix}\boldsymbol{G} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{R}\end{bmatrix}\right)^{-\frac{1}{2}}\\ \cdot\exp&\left\{-\tfrac{1}{2\sigma^2}\begin{pmatrix}\boldsymbol{b}\\ \boldsymbol{Y}-\boldsymbol{X\beta}-\boldsymbol{Zb}\end{pmatrix}^T\cdot\begin{bmatrix}\boldsymbol{G} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{R}\end{bmatrix}^{-1}\begin{pmatrix}\boldsymbol{b}\\ \boldsymbol{Y}-\boldsymbol{X\beta}-\boldsymbol{Zb}\end{pmatrix}\right\}.\end{aligned} \tag{A.6}$$

Para maximizar (A.6) respecto a $\boldsymbol{\beta}$ y $\boldsymbol{b}$ se requiere minimizar:

$$\begin{pmatrix}\boldsymbol{b}\\ \boldsymbol{Y}-\boldsymbol{X\beta}-\boldsymbol{Zb}\end{pmatrix}^T\cdot\begin{bmatrix}\boldsymbol{G} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{R}\end{bmatrix}^{-1}\begin{pmatrix}\boldsymbol{b}\\ \boldsymbol{Y}-\boldsymbol{X\beta}-\boldsymbol{Zb}\end{pmatrix} =$$

$$\boldsymbol{b^TG^{-1}b} + \boldsymbol{(Y-X\beta-Zb)^TR^{-1}(Y-X\beta-Zb)}. \tag{A.7}$$

Derivando (A.7) respecto a $\boldsymbol{\beta}$ y $\boldsymbol{b}$ e igualando estas derivadas a cero, se obtiene (A.5).

A.2.4. Estimación de las matrices de covarianza

Existe una amplia variedad de literatura para la estimación de las matrices de covarianza en los modelos mixtos. En esta sección se revisará someramente las técnicas *máxima verosimilitud* (MV) y *máxima verosimilitud restringida* (MVRE), que se han convertido en las más comunes para estimar los parámetros de la matriz de covarianza.

Primero se describirá MV. Como se vio antes,

$$\boldsymbol{V} \equiv \mathrm{Cov}(Y) = \boldsymbol{ZGZ^T} + \boldsymbol{R}.$$

El estimador de MV está basado en el modelo

$$\boldsymbol{Y} \sim N(\boldsymbol{X\beta}, \boldsymbol{V}).$$

La log-verosimilitud de $\boldsymbol{Y}$ en este modelo es

$$\ell(\boldsymbol{\beta}, \boldsymbol{V}) = -\frac{1}{2}\{n\log(2\pi) + \log|\boldsymbol{V}| + \boldsymbol{(Y-X\beta)^TV^{-1}(Y-X\beta)}\}, \tag{A.8}$$

de tal manera que la estimación de $\ell(\boldsymbol{\beta}, \boldsymbol{V})$ es aquella que maximiza el lado derecho de esta expresión. Si uno maximiza esta expresión respecto a $\boldsymbol{\beta}$, manteniendo fijo $\boldsymbol{V}$, se obtiene (A.2). Reemplazando esta en (A.8) se obtiene el *perfil de log-verosimilitud* para $\boldsymbol{V}$:

$$\begin{aligned}\ell_P(\boldsymbol{V}) &= -\frac{1}{2}\{\log|\boldsymbol{V}| + \boldsymbol{(Y-X\tilde{\beta})^TV^{-1}(Y-X\tilde{\beta})} + n\log(2\pi)\}\\ &= -\frac{1}{2}[\log|\boldsymbol{V}| + \boldsymbol{Y^TV^{-1}\{I-X(X^TV^{-1}X)^{-1}X^TV^{-1}\}Y}] - \frac{n}{2}\log(2\pi)\end{aligned} \tag{A.9}$$

Los estimadores MV de los parámetros de $\boldsymbol{V}$ se encuentran maximixando (A.9) sobre cada parámetro.

En el caso del MVRE la derivación es más complicada ya que se trata de maximizar la verosimilitud de combinaciones lineales de los elementos de $\boldsymbol{Y}$ que no dependen de $\boldsymbol{\beta}$. Detalles sobre esto se puede encontrar en Searle et al. (1992, cap.6), donde se tiene como resultado la *log-verosimilitud restringida* para $\boldsymbol{V}$:

$$\ell_R(\boldsymbol{V}) = \ell_P(\boldsymbol{V}) - \frac{1}{2}\log|\boldsymbol{X^T V^{-1} X}|. \tag{A.10}$$

La principal ventaja de MVRE sobre MV es que MVRE tiene en cuenta los grados de libertad para los efectos fijos del modelo. Para muestras de tamaño pequeño se espera que MVRE sea más preciso que MV; en caso se tengan muestras grandes, la diferencia es pequeña entre ambos enfoques.

Apéndice B

Anexos de tablas y gráficos

B.1. Error cuadrático medio del Estudio de Simulación

El Cuadro B.1 muestra el error cuadrático medio obtenido en el estudio de simulación al ajustar los valores predichos a la curva original, para cada escenario.

Escenarios			Máxima	Algoritmo propio		Algoritmo BUGS	
Spline	σ^2	Nodos	Verosimilitud	Media	Mediana	Media	Mediana
cúbico	0.5	5	0.325	0.325	0.327	0.331	0.332
de base	0.5	10	0.060	0.059	0.058	0.059	0.060
radial	0.5	15	0.061	0.061	0.062	0.065	0.065
	0.5	20	0.066	0.064	0.064	0.067	0.066
	1	5	0.604	0.585	0.583	0.591	0.593
	1	10	0.584	0.170	0.172	0.166	0.167
	1	15	0.578	0.164	0.164	0.170	0.170
	1	20	0.167	0.163	0.163	0.177	0.176
	1.5	5	0.524	0.530	0.533	0.533	0.537
	1.5	10	0.510	0.462	0.463	0.485	0.488
	1.5	15	0.507	0.411	0.408	0.490	0.489
	1.5	20	0.505	0.470	0.468	0.496	0.495

Cuadro B.1: Error cuadrático medio evaluado respecto a la curva original

B.2. Simulación de 100,000 iteraciones, *burn in* de 50,000 y thin de 10

Las figuras B.1, B.2, B.3 y B.4 presentan las iteraciones de los parámetros $\beta_0, \beta_1, \sigma_\varepsilon$ y σ_b, respectivamente.

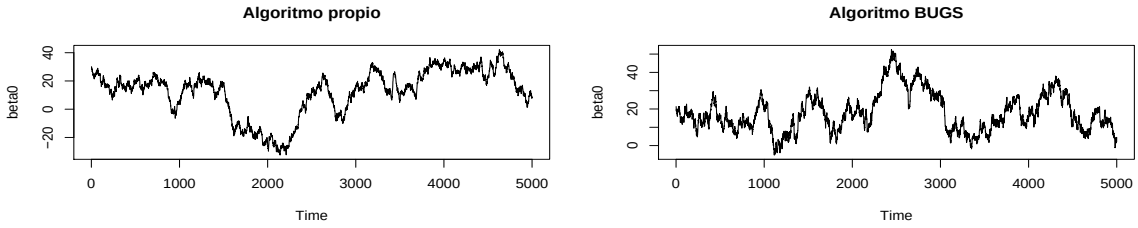

Figura B.1: Valores del coeficiente β_0 en cada iteración

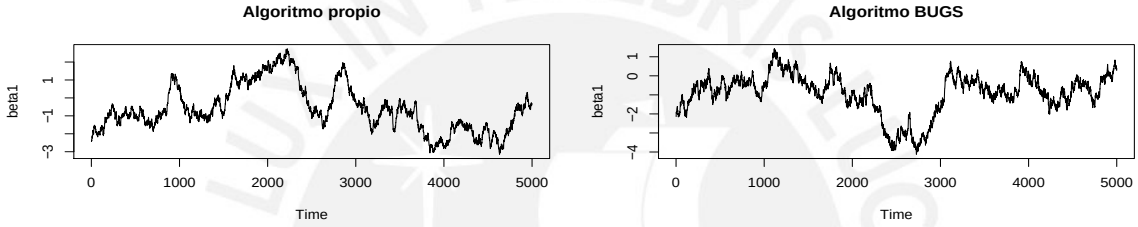

Figura B.2: Valores del coeficiente β_1 en cada iteración

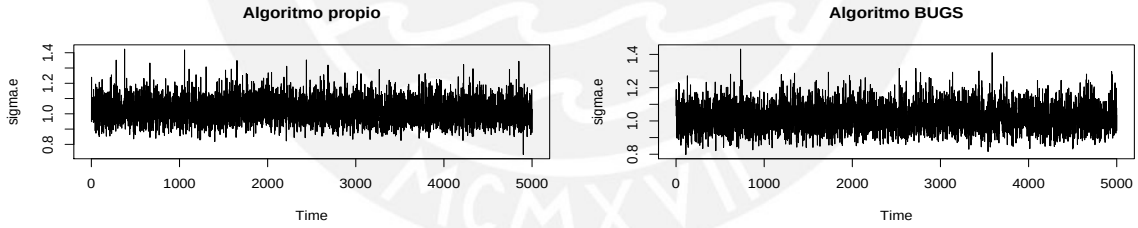

Figura B.3: Valores del coeficiente σ_ε en cada iteración

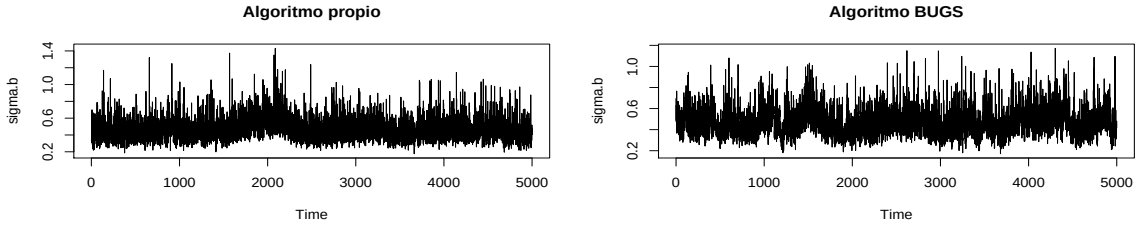

Figura B.4: Valores del coeficiente σ_b en cada iteración

B.3. Simulación de 500,000 iteraciones, *burn in* de 100,000 y thin de 200

Las figuras B.5, B.6, B.7 y B.8 presentan las iteraciones de los parámetros $\beta_0, \beta_1, \sigma_\varepsilon$ y σ_b, respectivamente.

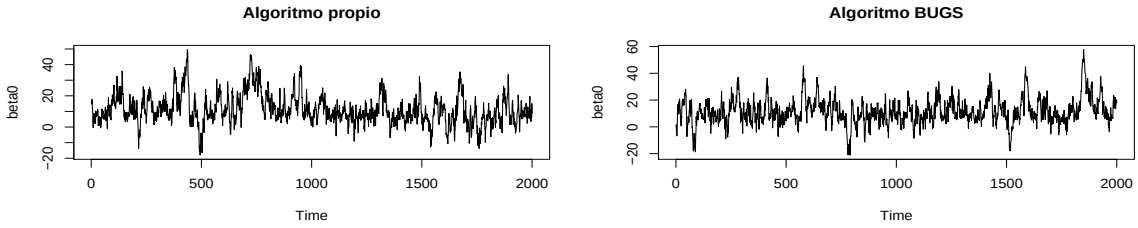

Figura B.5: Valores del coeficiente β_0 en cada iteración

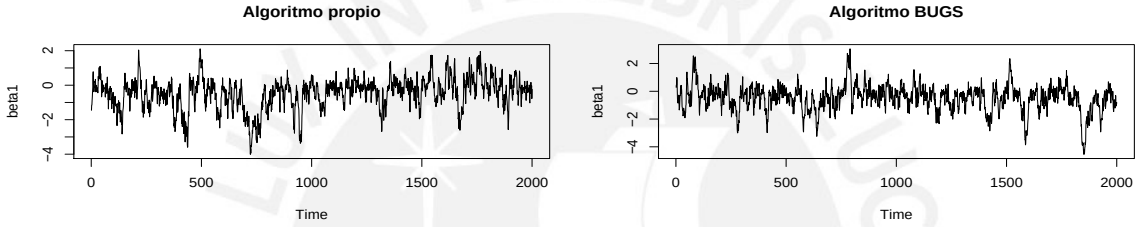

Figura B.6: Valores del coeficiente β_1 en cada iteración

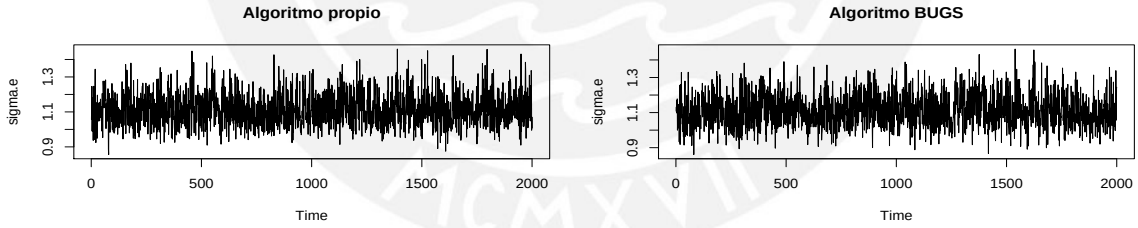

Figura B.7: Valores del coeficiente σ_ε en cada iteración

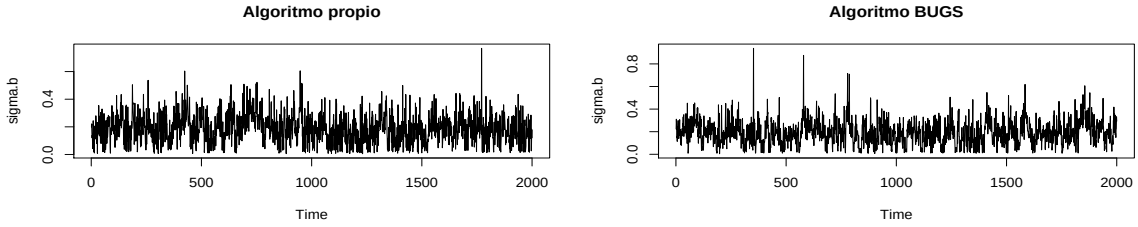

Figura B.8: Valores del coeficiente σ_b en cada iteración

B.4. Tiempos de ejecución del Estudio de Simulación

El Cuadro B.2 muestra los tiempos de ejecución de cada algoritmo bayesiano para un total de 1,000,000 iteraciones, un periodo de calentamiento (*burn in*) de 100,000 y un salto entre iteraciones (*thin*) de 300.

Escenarios			Algoritmo propio	Algoritmo BUGS
Spline	σ^2	Nodos	Tiempo (min.)	Tiempo (min.)
cúbico	0.5	5	25	3
de base	0.5	10	32	6
radial	0.5	15	42	10
	0.5	20	51	14
	1	5	86	3
	1	10	95	6
	1	15	106	10
	1	20	115	14
	1.5	5	151	3
	1.5	10	160	6
	1.5	15	170	10
	1.5	20	179	14

Cuadro B.2: Tiempos de ejecución de modelos según escenarios

Apéndice C

Estudio de Simulación: Código en R

```
# Librerías requeridas
library(R2WinBUGS)
library(nlme)
library(mnormt)

# Distribuciones a priori
curve(dnorm(x,0,10^3),-100,100)
curve(1/dgamma(x,10^-6,10^-6),add=T)

# Ecuación a emplear # pdf simulacion1
curve(sin(x)+2*exp(-10*x^2)+x/4-x^2/50+5,0,20,ylab="y")
title(main="y=sin(x)+2*exp(-10*x^2)+x/4-x^2/50+5")


# ESCENARIOS DEL MODELO (establecer parámetros para correr los modelos)

# Nro. iteraciones para modelos Gibbs sampling
iteraciones<-1000000
burnin<-100000
thin<-300
tam_muestra=100

# DEFINICIÓN DE MATRICES IMPORTANTES PARA LOS ESCENARIOS

varianzas<-matrix(c(0.5,1,1.5),nrow=3)
nodos<-matrix(c(5,10,15,20),nrow=4)
resultR_simu<-matrix(nrow=length(nodos),ncol=5*6+1+tam_muestra*5+5*2)
resultR_simu2<-matrix(nrow=2*length(nodos)*length(varianzas),
ncol=5*6+1+tam_muestra*5+5*2)
beta_0<-matrix(nrow=(iteraciones-burnin)/thin,ncol=2*length(nodos))
beta0<-matrix(nrow=(iteraciones-burnin)/thin,
ncol=4*length(nodos)*length(varianzas))
beta_1<-matrix(nrow=(iteraciones-burnin)/thin,ncol=2*length(nodos))
beta1<-matrix(nrow=(iteraciones-burnin)/thin,
ncol=4*length(nodos)*length(varianzas))
sigma_e<-matrix(nrow=(iteraciones-burnin)/thin,ncol=2*length(nodos))
```

```
sigmae<-matrix(nrow=(iteraciones-burnin)/thin,
ncol=4*length(nodos)*length(varianzas))
sigma_b<-matrix(nrow=(iteraciones-burnin)/thin,ncol=2*length(nodos))
sigmab<-matrix(nrow=(iteraciones-burnin)/thin,
ncol=4*length(nodos)*length(varianzas))
datax<-matrix(nrow=tam_muestra,ncol=3)
datay<-matrix(nrow=tam_muestra,ncol=3)


# INICIO DE LA SIMULACIÓN DE TODOS LOS ESCENARIOS


# ESCENARIO DE VARIANZAS

for (k in 1:length(varianzas)){

var <- varianzas[k,1]
e<-rnorm(tam_muestra,0,sqrt(var))

# ESCENARIO DE SPLINES

for (l in 1:2){

# ESCENARIO DE NODOS

for (m in 1:length(nodos)){

num.knots <- nodos[m,1]
if (num.knots==5)
program.file.name="D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/modelo_winbugs5.txt"
if (num.knots==10)
program.file.name="D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/modelo_winbugs10.txt"
if (num.knots==15)
program.file.name="D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/modelo_winbugs15.txt"
if (num.knots==20)
program.file.name="D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/modelo_winbugs20.txt"

# CREACIÓN DE VARIABLES SIMULADAS Y NODOS (sirve para los 3 modelos)

# Generación de los datos
x<-seq(0,20,length=tam_muestra)
y<-sin(x)+2*exp(-10*x^2)+x/4-x^2/50+5+e
ycur<-sin(x)+2*exp(-10*x^2)+x/4-x^2/50+5
plot(x,y)
```

```
curve(sin(x)+2*exp(-10*x^2)+x/4-x^2/50+5,add=T)
datax[,k]<-x
datay[,k]<-y

# Obtención de valores de los nodos empleados
knots <- quantile(unique(x), seq(0,1,
length=(num.knots+2))[-c(1,(num.knots+2))])

# SPLINES

if (l==1)   # Empleo del spline cúbico de base radial
{n <- length(x)
X <- cbind(rep(1,n),x)
svd.Omega <- svd((abs(outer(knots,knots,"-")))^3)
matrix.sqrt.Omega <- svd.Omega$u%*%sqrt(diag(svd.Omega$d))%*%t(svd.Omega$v)
Z <- ((abs(outer(x,knots,"-")))^3)%*%solve(matrix.sqrt.Omega)}
if (l==2) # Empleo del spline de base lineal truncada
{n <- length(x)
X <- cbind(rep(1,n),x)
Z <- outer(x,knots,"-")
Z <- Z*(Z>0)}


#---------------------------------------------------------------------


# MÉTODO 1: ESTIMACIÓN CLÁSICA VÍA MÁXIMA VEROSIMILITUD

# Estimación vía mediante MV en el modelo lineal mixto
one <- rep(1,nrow(Z))
fit_mv <- lme(y~-1+X,random=list(one=pdIdent(~-1+Z)))
beta_mv <- fit_mv$coef$fixed
b_mv <- unlist(fit_mv$coef$random)
f_mv <- X%*%beta_mv + Z%*%b_mv
sigma.eps_mv <- fit_mv$sigma
sigma.b_mv <- sigma.eps_mv*exp(unlist(fit_mv$modelStruct))
lambda_mv <- sigma.b_mv^2/sigma.eps_mv^2

# Parámetros del modelo vía Máxima Verosimilitud
beta_mv
sigma.eps_mv
sigma.b_mv
b_mv

# Error cuadrático medio del modelo MV a los datos simulados
e1 <- (y-f_mv)*(y-f_mv)
ecm_mv <- mean(e1)
ecm_mv
```

```
# Error cuadrático medio del modelo MV a la curva original
e1cur <- (ycur-f_mv)*(ycur-f_mv)
ecm_mv_cur <- mean(e1cur)
ecm_mv_cur

# Matriz de resultados
result_emv <- c(ecm_mv,ecm_mv_cur,beta_mv,sigma.eps_mv,sigma.b_mv,f_mv)
result_emv


#-----------------------------------------------------------------------


# MÉTODO 2: ESTIMACIÓN BAYESIANA VÍA ALGORITMO PROPIO

# Creación de matrices identidad
B <- diag(c(rep(1,2)))
A <- diag(c(rep(1,num.knots)))

# Definición de valores iniciales y matrices del algoritmo

Ae0<-10^-6; Be0<-10^-6; Ab0<-10^-6; Bb0<-10^-6;
theta <- matrix(nrow=iteraciones, ncol=4+n)
beta<-matrix(c(0,0),nrow=2); b<-matrix(c(rep(0,num.knots)),nrow=num.knots);
cur.tau.e<-1; cur.tau.b<-1;
sigma.e<-sqrt(1/cur.tau.e); sigma.b<-sqrt(1/cur.tau.b); sigma.beta<-10^3
epsilon_ibR<-matrix(rep(0,n),n,1)
ystar_ibR0<-matrix(rep(0,n),n,1)

# Algoritmo de Gibbs sampling

t1=system.time(
for (t in 1:iteraciones){

# Obtención de parámetros de las dist. condicionales completas de beta
mu_beta <- solve(t(X)%*%X+B*(sigma.e^2)/(sigma.beta^2))%*%t(X)%*%(y-Z%*%b)
sd_beta <- solve(t(X)%*%X+B*(sigma.e^2)/(sigma.beta^2))*sigma.e^2
beta<-matrix(rmnorm(1,mu_beta,sd_beta),2,1)

# Obtención de parámetros de las dist. condicionales completas de b
mu_b <- solve(t(Z)%*%Z+A*(sigma.e^2)/(sigma.b^2))%*%t(Z)%*%(y-X%*%beta)
sd_b <- solve(t(Z)%*%Z+A*(sigma.e^2)/(sigma.b^2))*sigma.e^2
b<-matrix(rmnorm(1,mu_b,sd_b),num.knots,1)

# Obtención de parámetros de las dist. condicionales completas de sigma.b
Ab <- Ab0 + 0.5*num.knots
Bb <- Bb0 + 0.5*sum((b)^2)
```

```
cur.tau.b <- rgamma(1,Ab,Bb)
sigma.b <- sqrt(1/cur.tau.b)

# Obtención de parámetros de las dist. condicionales completas de sigma.e
Ae <- Ae0 + 0.5*n
Be <- Be0 + 0.5*sum((y-X%*%beta-Z%*%b)^2)
cur.tau.e <- rgamma(1,Ae,Be)
sigma.e <- sqrt(1/cur.tau.e)

# Predicción de nuevas observaciones
for (i in 1:n){
epsilon_ibR[i,1] <- rnorm(1,0,sigma.e)}
ystar_ibR0 <- X%*%beta + Z%*%b + epsilon_ibR

 theta[t,]<-c(beta,sigma.e,sigma.b,ystar_ibR0)
})

# Parámetros del modelo vía Algoritmo de Gibbs en R sin considerar burn-in

# Vector de parámetros simulados
beta0v_ibR <- theta[seq((burnin+1),iteraciones,by=thin),1]
beta1v_ibR <- theta[seq((burnin+1),iteraciones,by=thin),2]
sigma.epsv_ibR <- theta[seq((burnin+1),iteraciones,by=thin),3]
sigma.bv_ibR <- theta[seq((burnin+1),iteraciones,by=thin),4]

# Medias
beta_ibR <- c(mean(beta0v_ibR),mean(beta1v_ibR))
sigma.eps_ibR <- mean(sigma.epsv_ibR)
sigma.b_ibR <- mean(sigma.bv_ibR)
beta_ibR
sigma.eps_ibR
sigma.b_ibR
# Medianas
beta_ibR50 <- c(median(beta0v_ibR),median(beta1v_ibR))
sigma.eps_ibR50 <- median(sigma.epsv_ibR)
sigma.b_ibR50 <- median(sigma.bv_ibR)
beta_ibR50
sigma.eps_ibR50
sigma.b_ibR50

# Error cuadrático medio del modelo a los datos simulados
# vía Algoritmo de Gibbs en R

# Para la media de los valores predichos
ystar_ibR <- matrix(0,nrow=tam_muestra,ncol=1)
for (j in 1:n){
ystar_ibR[j,1] <- mean(theta[seq((burnin+1),iteraciones,by=thin),j+4])}
e2 <- (y-ystar_ibR)*(y-ystar_ibR)
```

```
ecm_ibR <- mean(e2)
ecm_ibR
# Para la mediana de los valores predichos
ystar_ibR50 <- matrix(0,nrow=tam_muestra,ncol=1)
for (j in 1:n){
ystar_ibR50[j,1] <- median(theta[seq((burnin+1),iteraciones,by=thin),j+4])}
e3 <- (y-ystar_ibR50)*(y-ystar_ibR50)
ecm_ibR50 <- mean(e3)
ecm_ibR50

# Error cuadrático medio del modelo a la curva original
# vía Algoritmo de Gibbs en R

# Para la media de los valores predichos
e2cur <- (ycur-ystar_ibR)*(ycur-ystar_ibR)
ecm_ibRcur <- mean(e2cur)
ecm_ibRcur
# Para la mediana de los valores predichos
e3cur <- (ycur-ystar_ibR50)*(ycur-ystar_ibR50)
ecm_ibR50cur <- mean(e3cur)
ecm_ibR50cur


# Matriz de resultados

# Para la media
result_ibR <- c(ecm_ibR,ecm_ibRcur,beta_ibR,sigma.eps_ibR,sigma.b_ibR,
ystar_ibR)
result_ibR
# Para la mediana
result_ibR50 <- c(ecm_ibR50,ecm_ibR50cur,beta_ibR50,sigma.eps_ibR50,
sigma.b_ibR50,ystar_ibR50)
result_ibR50


#-------------------------------------------------------------------------


# MÉTODO 3: ESTIMACIÓN BAYESIANA VÍA ALGORITMO BUGS

# Argumentos de la función bugs
model.file <- program.file.name
inits.b <- rep(0,num.knots)
inits <- function(){list(beta=c(0,0),b=inits.b,taub=1,taueps=1)}
parameters <- list("beta","sigmab","sigmaeps","lambda","b","ystar")
data <- list(y=y,X=X,Z=Z,n=n,num.knots=num.knots)

# Simulación vía WinBUGS
```

```
t2=system.time(
fit_bugs <- bugs(data,inits,parameters,model.file,n.chains=1,
n.iter=iteraciones,n.burnin=burnin,n.thin=thin,digits=5,
bugs.directory="D:/Diego/Maestría Estadística/WinBUGS/WinBUGS14/")
)


# Obtención de las simulaciones de cada parámetro sin considerar burn-in
beta0_bugs <- fit_bugs$sims.array[,1,1]
beta1_bugs <- fit_bugs$sims.array[,1,2]
sigma.b_bugs <- fit_bugs$sims.array[,1,3]
sigma.eps_bugs <- fit_bugs$sims.array[,1,4]
b_bugs <- fit_bugs$sims.array[,1,6:(num.knots+5)]
f_bugs <- fit_bugs$sims.array[,1,(num.knots+6):(num.knots+tam_muestra+5)]

# Obtención del valor de la media de cada parámetro
beta_m_bugs <- fit_bugs$mean$beta
sigma.eps_m_bugs <- fit_bugs$mean$sigmaeps
sigma.b_m_bugs <- fit_bugs$mean$sigmab
b_m_bugs <- fit_bugs$mean$b

# Obtención del valor de la mediana de cada parámetro
beta_50_bugs <- fit_bugs$median$beta
sigma.eps_50_bugs <- fit_bugs$median$sigmaeps
sigma.b_50_bugs <- fit_bugs$median$sigmab
b_50_bugs <- fit_bugs$median$b

# Parámetros del modelo vía Inferencia Bayesiana en WinBUGS

# Media
beta_m_bugs
sigma.b_m_bugs
sigma.eps_m_bugs
b_m_bugs
# Mediana
beta_50_bugs
sigma.b_50_bugs
sigma.eps_50_bugs
b_50_bugs

# Error cuadrático medio del modelo a los datos simulados
# Para la media
f1_bugs <- fit_bugs$mean$ystar
e2 <- (y-f1_bugs)*(y-f1_bugs)
ecm_m_bugs <- mean(e2)
ecm_m_bugs
# Para la mediana
```

```
f2_bugs <- fit_bugs$median$ystar
e3 <- (y-f2_bugs)*(y-f2_bugs)
ecm_50_bugs <- mean(e3)
ecm_50_bugs

# Error cuadrático medio del modelo a la curva original
# Para la media
e2cur <- (ycur-f1_bugs)*(ycur-f1_bugs)
ecm_m_bugscur <- mean(e2cur)
ecm_m_bugscur
# Para la mediana
e3cur <- (ycur-f2_bugs)*(ycur-f2_bugs)
ecm_50_bugscur <- mean(e3cur)
ecm_50_bugscur

# Criterio para la selección de modelos: DIC
fit_bugs$DIC


# Matriz de resultados

# Para la media
result_ibWB <- c(ecm_m_bugs,ecm_m_bugscur,beta_m_bugs,sigma.eps_m_bugs,
sigma.b_m_bugs,f1_bugs)
result_ibWB
# Para la mediana
result_ibWB50 <- c(ecm_50_bugs,ecm_50_bugscur,beta_50_bugs,sigma.eps_50_bugs,
sigma.b_50_bugs,f2_bugs)
result_ibWB50


#-----------------------------------------------------------------------

# OBTENCIÓN DE VARIABLES Y ESTADÍSTICOS IMPORANTES PARA EL ANÁLISIS

# Matriz de parámetros
result_emv
result_ibR
result_ibR50
result_ibWB
result_ibWB50

# Criterio para la selección de modelos: DIC
fit_bugs$DIC

# CREA LAS MATRICES CON LOS VECTORES Y PARÁMETROS OBTENIDOS EN LA SIMULACIÓN

resultR_simu[m,] <- c(result_emv,result_ibR,result_ibR50,result_ibWB,
```

```
result_ibWB50,fit_bugs$DIC,t1,t2)

beta_0[,(2*m-1):(2*m)] <- c(beta0v_ibR,beta0_bugs)
beta_1[,(2*m-1):(2*m)] <- c(beta1v_ibR,beta1_bugs)
sigma_e[,(2*m-1):(2*m)] <- c(sigma.epsv_ibR,sigma.eps_bugs)
sigma_b[,(2*m-1):(2*m)] <- c(sigma.bv_ibR,sigma.b_bugs)

}

if (l==1)
{sup=k*m
sup2=2*k*m}
if (l==2)
{sup=(l+k+1)*m
sup2=2*(l+k+1)*m}

resultR_simu2[(sup-3):(sup),] <- resultR_simu

beta0[,(sup2-7):(sup2)] <- beta_0
beta1[,(sup2-7):(sup2)] <- beta_1
sigmae[,(sup2-7):(sup2)] <- sigma_e
sigmab[,(sup2-7):(sup2)] <- sigma_b

}

}

datasim=matrix(c(datax,datay),nrow=n)

# Exporta resultados para el análisis
write.csv(resultR_simu2,"D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/resultR_simu.csv")
write.csv(beta0,"D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/beta0.csv")
write.csv(beta1,"D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/beta1.csv")
write.csv(sigmae,"D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/sigmae.csv")
write.csv(sigmab,"D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/sigmab.csv")
write.csv(datasim,"D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/datasim.csv")
#-----------------------------------------------------------------------
```

Apéndice D

Estudio de Simulación: Código en WinBUGS

```
model{                                  #Inicio Modelo

#Este modelo puede ser empleado para ajustar la relación entre dos variables.
#Se puede adecuar fácilmente para más covariables o para más
#variables de efectos aleatorios. En este caso se muestra para 10 nodos.

#Verosimilitud del modelo
    for (i in 1:n)
        {y[i]~dnorm(m[i],taueps)
         m[i]<-mfe[i]+mre110[i]
 mfe[i]<-beta[1]*X[i,1]+beta[2]*X[i,2]
 mre110[i]<-b[1]*Z[i,1]+b[2]*Z[i,2]+b[3]*Z[i,3]+b[4]*Z[i,4]+b[5]*Z[i,5]+
  b[6]*Z[i,6]+b[7]*Z[i,7]+b[8]*Z[i,8]+b[9]*Z[i,9]+b[10]*Z[i,10]
}

#Distribuciones a priori de los parámetros de efectos aleatorios
    for (k in 1:num.knots){b[k]~dnorm(0,taub)}

#Distribuciones a priori de los parámetros de efectos fijos
    for (l in 1:2){beta[l]~dnorm(0,1.0E-6)}

#Distribuciones a priori de los parámetros de precisión
    taueps~dgamma(1.0E-6,1.0E-6)
    taub~dgamma(1.0E-6,1.0E-6)

#Transformaciones y valor del parámetro de suavizamiento (lambda)
    sigmaeps<-1/sqrt(taueps)
    sigmab<-1/sqrt(taub)
    lambda<-pow(sigmab,2)/pow(sigmaeps,2)

#Predicción
    for (i in 1:n)
        {epsilonstar[i]~dnorm(0,taueps)
         ystar[i]<-m[i]+epsilonstar[i]}

    }                                   #Fin modelo
```

Apéndice E

Aplicación: Modelo de Regresión Spline Penalizado en R

```
# Estimación del modelo de regresión spline penalizado para 6 combinaciones
# de tservicio y servidores. En cada uno se evalúan 4 nodos: 5, 10, 15 y 20.
# Métodos de inferencia vía máxima verosimilitud e inferencia bayesiana
# Caso: Tiempos en cola vs tiempo total de atención por servidor

# El código es similar al del estudio de simulación, salvo que está hecho
# de tal manera que se lean varios archivos de datos

# Librerías requeridas
library(R2WinBUGS)
# ESCENARIOS DEL MODELO (establecer parámetros para correr los modelos)

# Nro. iteraciones para modelos Gibbs Sampling
iteraciones<-200000
burnin<-50000
thin<-10

# DEFINICIÓN DE MATRICES IMPORTANTES PARA LOS ESCENARIOS

nodos<- matrix(c(5,10,15,20),nrow=4)
resultR_simu <- matrix(nrow=length(nodos),ncol=5+1+5)
resultR_simu2 <- matrix(nrow=9*length(nodos),ncol=5+1+5)
beta_0 <- matrix(nrow=(iteraciones-burnin)/thin,ncol=length(nodos))
beta0 <- matrix(nrow=(iteraciones-burnin)/thin,ncol=9*length(nodos))
beta_1 <- matrix(nrow=(iteraciones-burnin)/thin,ncol=length(nodos))
beta1 <- matrix(nrow=(iteraciones-burnin)/thin,ncol=9*length(nodos))
sigma_e <- matrix(nrow=(iteraciones-burnin)/thin,ncol=length(nodos))
sigmae <- matrix(nrow=(iteraciones-burnin)/thin,ncol=9*length(nodos))
sigma_b <- matrix(nrow=(iteraciones-burnin)/thin,ncol=length(nodos))
sigmab <- matrix(nrow=(iteraciones-burnin)/thin,ncol=9*length(nodos))


# INICIO DE LA SIMULACIÓN DE TODOS LOS ESCENARIOS
```

```
# ESCENARIOS DE BD TIEMPOS DE SERVICIO Y SERVIDORES

for (k in 1:9){

if (k==1)
rutadatos="D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/t10s1.txt"
if (k==2)
rutadatos="D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/t10s2.txt"
if (k==3)
rutadatos="D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/t10s3.txt"
if (k==4)
rutadatos="D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/t12s1.txt"
if (k==5)
rutadatos="D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/t12s2.txt"
if (k==6)
rutadatos="D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/t12s3.txt"
if (k==7)
rutadatos="D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/t14s1.txt"
if (k==8)
rutadatos="D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/t14s2.txt"
if (k==9)
rutadatos="D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/t14s3.txt"

datos=read.table(rutadatos,header=TRUE)
x=datos[,1]
y=datos[,2]
n <- length(x)
ajuste0<- matrix(nrow=n,ncol=length(nodos))

# ESCENARIO DE SPLINES

for (l in 1:1){

# ESCENARIO DE NODOS

for (m in 1:length(nodos)){

num.knots <- nodos[m,1]
if (num.knots==5)
```

```
program.file.name="D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/modelo_winbugs5.txt"
if (num.knots==10)
program.file.name="D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/modelo_winbugs10.txt"
if (num.knots==15)
program.file.name="D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/modelo_winbugs15.txt"
if (num.knots==20)
program.file.name="D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/modelo_winbugs20.txt"

# CREACIÓN DE LOS NODOS (sirve para todos los escenarios)

# Obtención de valores de los nodos empleados
knots <- quantile(unique(x), seq(0,1,
length=(num.knots+2))[-c(1,(num.knots+2))])

# SPLINES

if (l==1)   # Empleo del spline cúbico de base radial
{n <- length(x)
X <- cbind(rep(1,n),x)
svd.Omega <- svd((abs(outer(knots,knots,"-")))^3)
matrix.sqrt.Omega <- svd.Omega$u%*%sqrt(diag(svd.Omega$d))%*%t(svd.Omega$v)
Z <- ((abs(outer(x,knots,"-")))^3)%*%solve(matrix.sqrt.Omega)}


#-------------------------------------------------------------------


# ESTIMACIÓN BAYESIANA VÍA ALGORITMO DE GIBBS SAMPLING EN WINBUGS

# Argumentos de la función bugs
model.file <- program.file.name
inits.b <- rep(0,num.knots)
inits <- function(){list(beta=c(0,0),b=inits.b,taub=0.01,taueps=0.01)}
parameters <- list("beta","sigmab","sigmaeps","lambda","b","ystar")
data <- list(y=y,X=X,Z=Z,n=n,num.knots=num.knots)

# Simulación vía WinBUGS

t2=system.time(
fit_bugs <- bugs(data,inits,parameters,model.file,n.chains=1,
n.iter=iteraciones,n.burnin=burnin,n.thin=thin,digits=5,
bugs.directory="D:/Diego/Maestría Estadística/WinBUGS/WinBUGS14/")
)
```

```
# Obtención de las simulaciones de cada parámetro considerando burnin
beta0_bugs <- fit_bugs$sims.array[,1,1]
beta1_bugs <- fit_bugs$sims.array[,1,2]
sigma.b_bugs <- fit_bugs$sims.array[,1,3]
sigma.eps_bugs <- fit_bugs$sims.array[,1,4]
b_bugs <- fit_bugs$sims.array[,1,6:(num.knots+5)]
f_bugs <- fit_bugs$sims.array[,1,(num.knots+6):(num.knots+5+n)]

# Obtención del valor de la media de cada parámetro
beta_m_bugs <- fit_bugs$mean$beta
sigma.eps_m_bugs <- fit_bugs$mean$sigmaeps
sigma.b_m_bugs <- fit_bugs$mean$sigmab
b_m_bugs <- fit_bugs$mean$b

# Parámetros del modelo vía Inferencia Bayesiana en WinBUGS
beta_m_bugs
sigma.b_m_bugs
sigma.eps_m_bugs
b_m_bugs

# Error cuadrático medio del modelo a los datos reales
f1_bugs <- fit_bugs$mean$ystar
e2 <- (y-f1_bugs)*(y-f1_bugs)
ecm_m_bugs <- mean(e2)
ecm_m_bugs

# Matriz de resultados para la media

result_ibWB <- c(ecm_m_bugs,beta_m_bugs,sigma.eps_m_bugs,sigma.b_m_bugs)
result_ibWB


#-----------------------------------------------------------------------


# OBTENCIÓN DE VARIABLES Y ESTADÍSTICOS IMPORTANTES PARA EL ANÁLISIS

# Matriz de parámetros
result_ibWB

# Criterio para la selección de modelos: DIC
fit_bugs$DIC

# CREA LAS MATRICES CON LOS VECTORES Y PARÁMETROS OBTENIDOS EN LA SIMULACIÓN

resultR_simu[m,] <- c(result_ibWB,fit_bugs$DIC,t2)

beta_0[,m] <- c(beta0_bugs)
```

```
beta_1[,m] <- c(beta1_bugs)
sigma_e[,m] <- c(sigma.eps_bugs)
sigma_b[,m] <- c(sigma.b_bugs)

ajuste0[,m] <- c(f1_bugs)

}

resultR_simu2[(4*k-3):(4*k),] <- resultR_simu

beta0[,(4*k-3):(4*k)] <- beta_0
beta1[,(4*k-3):(4*k)] <- beta_1
sigmae[,(4*k-3):(4*k)] <- sigma_e
sigmab[,(4*k-3):(4*k)] <- sigma_b

ajuste <- cbind(x,y,ajuste0)

if (k==1)
write.table(ajuste,"D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/ajustet10s1.txt",row.names=FALSE,col.names=FALSE)
if (k==2)
write.table(ajuste,"D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/ajustet10s2.txt",row.names=FALSE,col.names=FALSE)
if (k==3)
write.table(ajuste,"D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/ajustet10s3.txt",row.names=FALSE,col.names=FALSE)
if (k==4)
write.table(ajuste,"D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/ajustet12s1.txt",row.names=FALSE,col.names=FALSE)
if (k==5)
write.table(ajuste,"D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/ajustet12s2.txt",row.names=FALSE,col.names=FALSE)
if (k==6)
write.table(ajuste,"D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/ajustet12s3.txt",row.names=FALSE,col.names=FALSE)
if (k==7)
write.table(ajuste,"D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/ajustet14s1.txt",row.names=FALSE,col.names=FALSE)
if (k==8)
write.table(ajuste,"D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/ajustet14s2.txt",row.names=FALSE,col.names=FALSE)
if (k==9)
write.table(ajuste,"D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/ajustet14s3.txt",row.names=FALSE,col.names=FALSE)

}

}
```

```
# Exporta resultados para el análisis
write.csv(resultR_simu2,"D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/resultR_apli.csv")
write.csv(beta0,"D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/beta0_apli.csv")
write.csv(beta1,"D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/beta1_apli.csv")
write.csv(sigmae,"D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/sigmae_apli.csv")
write.csv(sigmab,"D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/sigmab_apli.csv")
```

Apéndice F

Aplicación: Modelo de Regresión Cúbico en R

```
# Librerías requeridas
library(R2WinBUGS)

# ESCENARIOS DEL MODELO (establecer parámetros para correr los modelos)

# Nro. iteraciones para modelos Gibbs Sampling
iteraciones<-200000
burnin<-50000
thin<-10

# DEFINICIÓN DE MATRICES IMPORTANTES PARA LOS ESCENARIOS

resultR_simu <- matrix(nrow=9,ncol=1+1+5)

for (k in 1:9){

# Importa data de cada escenario
if (k==1)
rutadatos="D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/t10s1.txt"
if (k==2)
rutadatos="D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/t10s2.txt"
if (k==3)
rutadatos="D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/t10s3.txt"
if (k==4)
rutadatos="D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/t12s1.txt"
if (k==5)
rutadatos="D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/t12s2.txt"
if (k==6)
rutadatos="D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/t12s3.txt"
if (k==7)
```

```
rutadatos="D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/t14s1.txt"
if (k==8)
rutadatos="D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/t14s2.txt"
if (k==9)
rutadatos="D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/t14s3.txt"

# Definición de variables
datos=read.table(rutadatos,header=TRUE)
x=datos[,1]
y=datos[,2]
n <- length(x)
X <- cbind(rep(1,n),x)
ajuste0<- matrix(nrow=n,ncol=1)


# ESTIMACIÓN BAYESIANA VÍA ALGORITMO DE GIBBS SAMPLING EN WINBUGS

# Argumentos de la función bugs
program.file.name="D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/regcubica.txt"
model.file=program.file.name
inits <- function(){list(beta=c(0,0,0,0),taueps=0.01)}
parameters <- list("ystar","beta")
data <- list(y=y,X=X,n=n)

# Simulación vía WinBUGS

t2=system.time(
fit_bugs <- bugs(data,inits,parameters,model.file,n.chains=1,
n.iter=iteraciones,n.burnin=burnin,n.thin=thin,digits=5,
bugs.directory="D:/Diego/Maestría Estadística/WinBUGS/WinBUGS14/")
)

# Obtención de los valores estimados vía simulación MCMC
f_bugs <- fit_bugs$sims.array[,1,1:n]

# Error cuadrático medio del modelo a los datos simulados
# Para la media
f1_bugs <- fit_bugs$mean$ystar
e2 <- (y-f1_bugs)*(y-f1_bugs)
ecm_m_bugs <- mean(e2)
ecm_m_bugs

# Criterio para la selección de modelos: DIC
fit_bugs$DIC
```

```
result_ibWB <- c(ecm_m_bugs,fit_bugs$DIC,t2)
result_ibWB

resultR_simu[k,] <- result_ibWB

ajuste0[,1] <- c(f1_bugs)
ajuste <- cbind(x,y,ajuste0)

# Exporta valores predichos para cada escenario
if (k==1)
write.table(ajuste,"D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/ajustecubt10s1.txt",row.names=FALSE,col.names=FALSE)
if (k==2)
write.table(ajuste,"D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/ajustecubt10s2.txt",row.names=FALSE,col.names=FALSE)
if (k==3)
write.table(ajuste,"D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/ajustecubt10s3.txt",row.names=FALSE,col.names=FALSE)
if (k==4)
write.table(ajuste,"D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/ajustecubt12s1.txt",row.names=FALSE,col.names=FALSE)
if (k==5)
write.table(ajuste,"D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/ajustecubt12s2.txt",row.names=FALSE,col.names=FALSE)
if (k==6)
write.table(ajuste,"D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/ajustecubt12s3.txt",row.names=FALSE,col.names=FALSE)
if (k==7)
write.table(ajuste,"D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/ajustecubt14s1.txt",row.names=FALSE,col.names=FALSE)
if (k==8)
write.table(ajuste,"D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/ajustecubt14s2.txt",row.names=FALSE,col.names=FALSE)
if (k==9)
write.table(ajuste,"D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/ajustecubt14s3.txt",row.names=FALSE,col.names=FALSE)

}

# Exporta resultados para el análisis
write.csv(resultR_simu,"D:/Diego/Maestría Estadística/Seminario de Tesis/
archivos R/resultR_aplicub.csv")
```

Apéndice G

Aplicación: Modelo de Regresión Cúbico en WinBUGS

```
model{

#Verosimilitud del modelo
    for (i in 1:n)
        {y[i]~dnorm(m[i],taueps)
         m[i]<-beta[1]*X[i,1]+beta[2]*X[i,2]+beta[3]*pow(X[i,2],2)+
         beta[4]*pow(X[i,2],3)
}

#Distribuciones a priori de los parámetros de efectos fijos
    for (l in 1:4){beta[l]~dnorm(0,1.0E-6)}

#Distribuciones a priori de los parámetros de precisión
    taueps~dgamma(1.0E-6,1.0E-6)

#Transformaciones
    sigmaeps<-1/sqrt(taueps)

#Predicción
    for (i in 1:n)
        {epsilonstar[i]~dnorm(0,taueps)
         ystar[i]<-m[i]+epsilonstar[i]}

    }
```

Bibliografía

Bazan, J. y Bayes, C. (2010). Inferencia bayesiana en modelos de regresion binaria usando brmuw, *Technical report* **25**.

Crainiceanu, C. M., Ruppert, D. y Wand, M. P. (2005). Bayesian analysis for penalized spline regression using winbugs, *Journal of Statistical Software* **14**.

Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models, *International Society for Bayesian Analysis* **3**.

Hastie, T. y Tibshirani, R. (1990). *Generalized Additive Models*, London: Chapman and Hall.

Hobert, J. P. y Casella, G. (1996). The effect of improper priors on gibbs sampling in hierarchical linear mixed models, *Journal of the American Statistical Association* **91**: 1461–1473.

McCulloch, C. E. y Searle, S. R. (2001). *Generalized, Linear, and Mixed Models*, New York: Wiley.

Ntzoufras, I. (2009). *Bayesian Modeling Using WinBUGS*, Wiley.

Robinson, G. K. (1991). That blup is a good thing: The estimation of random effects, *Statistical Science* **6**: 15–32.

Ruppert, D., Wand, M. P. y Carroll, R. (2003). *Semiparametric Regression*, New York: Cambridge University Press.

Searle, S. R., Casella, G. y McCulloch, C. E. (1992). *Variance Components*, New York: Wiley.

Sorensen, D. y Gianola, D. (2002). *Likelihood, Bayesian, and MCMC Methods in Quantitative Genetics*, Springer.

Wand, M. P. (2003). Smoothing and mixed models, *Computational Statistics* **18**: 223–249.

Wood, S. N. (2006). *Generalized Additive Models: An Introduction with R*, Chapman & Hall/CRC, Boca Raton, FL.