UNIVERSIDAD DE BUENOS AIRES
Facultad de Ciencias Exactas y Naturales
Departamento de Matemática

Tesis de Licenciatura

Detección de puntos de independencia para modelos Gaussianos Multivariados.

Yamila Alen

Directora: Daniela Rodriguez

07/07/2023

Agradecimientos

A mi mamá, mi papá y Mati por estar conmigo y apoyarme desde siempre. A toda mi familia, mi abuela, mis tios, tías y mis primos que también estuvieron alentándome en todo lo que hacía. A la familia nueva que fui armando, Nico, su papá y su mamá.

Al jurado por haber leído esta tesis y haberse tomada el trabajo de corregirla. A Daniela por haberme aceptado y ofrecerme un trabajo interesante que me hizo recordar lo lindo que es hacer matemática, algo que sentí que fui perdiendo a lo largo de la carrera.

A los amigos que fui haciendo a lo largo de la carrera, en especial a los "Nerds de mi ♡": Jaz, Eli, Juanma, Facu y Marce que son un grupo hermoso. A Dani con la que compartí los primeros cuatrimestre y nos volvimos a encontrar dando clases juntas. A anto con con quien compartí cursadas y salidas los primeros años. A las nuevas personas que conocí estos años ejerciendo docencia y me acompañaron estos últimos años.

A los espacios de divulgación, Tecnópolis y El C3, por ser lugares que me formaron con otra perspectiva y en los que compartí lugar de trabajo con gente linda, me encantaría nombrarlos a todos pero no quisiera olvidarme de nadie.

A Nico, porque sin su apoyo y cariño no hubiese podido escribir esta tesis que tanto me costó. Un pedacito de esta tesis también es tuya. Te quiero mucho

Índice general

Introducción

En este trabajo propondremos algunos criterios de selección de modelos para estimar puntos de independencia de un vector aleatorio, de manera que podamos descomponerlo en bloques independientes.

En [8] se abordó un trabajo similar proponiendo un método basado en un estimador general de la función de distribución, por lo que en este trabajo se decidió abordar la propuesta a encontrar otros tipos de estimadores bajo el supuesto que el vector aleatorio tiene una distribución perteneciente a una familia paramétrica conocida. En particular, en esta tesis supondremos que contamos con un vector aleatorio que sabemos previamente que tiene una distribución Normal Multivariada, y propondremos métodos de estimación que estarán basados en estimadores de los parámetros μ y Σ de la distribución.

El problema de descubrir o probar que un vector aleatorio en realidad se puede descomponer en sub-vectores independientes no es una idea nueva y ya fue abordada en la literatura, pero usualmente utilizando otras estrategias. Históricamente, la inferencia sobre la independencia fue abordada por medio de test de hipótesis. Algunos de los métodos más utilizados fueron las tablas de contingencia para datos categóricos y pruebas basadas sobre coeficientes de correlación, como los de Pearson, Kendall y Spearman. Otros enfoques más actuales de independencia son métodos basados en las distancias como la correlación de distancia, como se presenta en Szekely y Rizzo [12] o en Szekely, Rizzo y Bakirov [13]. También se han propuesto métodos basados en núcleos, incluyendo el criterio de información de Hilbert-Schmidt considerado en [5] y en [6]. En este trabajo se abordará la obtención de puntos de independencia mediante un método de penalización.

El trabajo de tesis está organizado de la siguiente manera. En el Capítulo 1 se hará un repaso breve de algunos conceptos básicos de probabilidad y estadística que serán necesarios recordar y tener presentes, los mismos serán resumidos a modo de lograr que la tesis sea lo más autocontenida posible. En el Capítulo 2 se desarrollará la teoría que explica cómo se estima el conjunto de independencia de un vector aleatorio y luego se propondrán dos diferentes enfoques para calcular estimadores que ayuden al cálculo del conjunto de independencia. En el Capítulo 3 se explicará cómo es la implementación

computacional de los diferentes estimadores. Además se proponen dos enfoques distintos, un algoritmo de tipo exhaustivo y uno que utilizará la técnica de *Divide and Conquer* para optimizar el funcionamiento. Por otro lado, se realizarán estudios de error y de tiempo de cómputo. En el Capítulo 4 se realizarán diferentes simulaciones y estudios de tiempo computacional y de tasa de error. Finalmente, en el Capítulo 5 se ilustrará la propuesta en un ejemplo concreto de datos reales (no sintéticos) para determinar el posible valor de puntos de independencia y su aplicación a un problema específio. En el Apéndice se describen los códigos escritos en R. Se utilizó la versión R 4.3.0.

Capítulo 1

Preliminares

En este capítulo daremos una breve descripción del problema que abordaremos en esta tesis y describiremos las primeras definiciones y conceptos necesarios para desarrollar la propuesta que nos servirá para resolverlo.

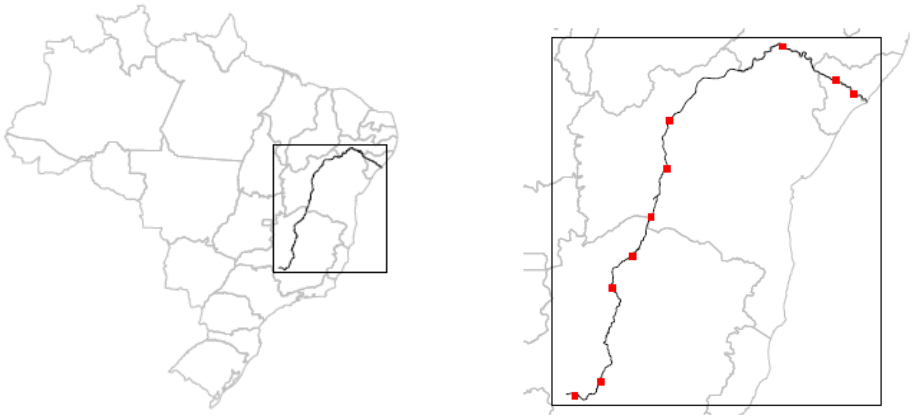
1.1. Problema del flujo del Río San Francisco

Los ríos, al ser flujos de agua que se mueven constantemente, tienen muchos factores, tanto naturales como humanos, que causan que su flujo de agua vaya cambiando a lo largo de diferentes sectores. Algunos ejemplos de esto son las lluvias, el deshielo, la descarga de agua subterránea de los acuíferos, y dentro de los factores en los que puede intervenir el ser humano están la regulación del caudal del río para la energía hidroeléctrica, la navegación, extracciones de agua, desvíos trans-cuenca y riego, entre otros.

Para poder conocer la influencia de estos diferentes factores en el caudal de un río necesitaríamos conocer cómo va variando su flujo. Una pregunta que entonces nos podemos plantear y resulta de interés es: ¿se puede determinar cuáles son los sectores del río en los cuales hay un cambio en el caudal del agua?

Como ejemplo estudiaremos el río San Francisco, situado en Brasil, cuya extensión es de unos 2.831 km, siendo el cuarto río más largo de Sudamérica. Nace en las sierras de Minas Gerais y atraviesa las regiones sudeste y noreste de Brasil pasando a través de una región semiárida donde es una fuente importante de agua. Además, los embalses y usinas hidroeléctricas sobre su cauce proveen de agua y electricidad a gran parte del noreste brasileño.

Como es necesario medir el caudal de agua en el río, a lo largo de su curso hay estaciones fijas de monitoreo [10]. En los siguientes mapas se muestran tanto la ubicación del Río San Francisco como la ubicación de cada uno de los medidores.



(a) (b)

Figura 1.1: (a) Mapa del Río San Francisco en Brasil. (b) Ampliación del Río San Francisco con la ubicación de las estaciones de monitoreo. Las imágenes son tomadas de [8].

En la Figura 1.1 (a) se muestran las fronteras de los estados de Brasil y dentro del recuadro la región en donde está el río San Francisco. En la Figura 1.1 (b) se muestra una ampliación de la región recuadrada en (a), en la que se puede ver el río San Francisco y los puntos rojos representan los medidores de caudal considerados en nuestro análisis. A lo largo de este trabajo consideramos estas estaciones numeradas en orden creciente a lo largo del curso del río. En este estudio usamos los datos de 10 estaciones de medición.

Finalmente, consideramos 358 mediciones por cada una de las estaciones, cada una de ellas consistiendo en el promedio mensual del flujo entre los años 1977 y 2016. El objetivo será a partir de los datos poder determinar en donde hay un cambio en el flujo del río a partir del estudio de bloques independientes. Para describir formalmente el problema introduciremos algunas definiciones y notación.

1.2. Definiciones básicas y notación

Comencemos recordando algunos conceptos básicos de probabilidad que serán necesarios. Este desarrollo teórico está adaptado de [1] y [14].

Definición 1.2.1. Dado un espacio muestral Ω, diremos que $\mathbf{X} = (X_1, \cdots, X_n)$ es un *vector aleatorio* de dimensión n si cada una de sus componentes es una variable aleatoria $X_i : \Omega \to \mathbb{R}$, para $i = 1, \cdots, n$. Notemos que el vector aleatorio es una función $\mathbf{X} : \Omega \to \mathbb{R}^n$.

Definición 1.2.2. Sea un espacio muestral Ω y una probabilidad P definida en Ω. Dado un vector aleatorio $\mathbf{X} = (X_1, \cdots, X_n)$ con $\mathbf{X} : \Omega \to \mathbb{R}^n$ definimos la *función de*

distribución acumulada conjunta asociada a $(X_1, \cdots, X_n)$ como

$$F_{\mathbf{X}}(x_1, \cdots, x_n) = P(X_1 \leq x_1, \cdots, X_n \leq x_n), \quad \text{para} \quad (x_1, \cdots, x_n) \in \mathbb{R}^n.$$

Definición 1.2.3. Diremos que el vector aleatorio $\mathbf{X} = (X_1, \cdots, X_n)$ es *continuo* si existe una función $f_{\mathbf{X}}(x_1, \cdots, x_n)$ de densidad, que satisface

$$P(\mathbf{X} \in A) = \int \cdots \int_A f_{\mathbf{X}}(x_1, \cdots, x_n) dx_1 .. dx_n,$$

para $A \subseteq \mathbb{R}^n$. En tal caso, diremos que $f_{\mathbf{X}}$ es la función de densidad asociada al vector $\mathbf{X}$.

Luego, si $\mathbf{X} = (X_1, \cdots, X_n)$ es un vector aleatorio continuo con función de densidad conjunta dada por $f_{\mathbf{X}}$ para cada X_i se pueden considerar las funciones de densidad marginales que vienen dadas por

$$f_{X_i}(x_i) = \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} f_{\mathbf{X}}(x_1, \cdots, x_n) dx_{-i},$$

donde notamos $x_{-i} = (x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n)$.

Definición 1.2.4. Diremos que las variables aleatorias $X_1, X_2, \cdots, X_n$ son *independientes* si

$$P(X_1 \in B_1, X_2 \in B_2, \cdots, X_n \in B_n) = \prod_{i=1}^{n} P(X_i \in B_i).$$

En el caso en el que el vector $\mathbf{X} = (X_1, \cdots, X_n)$ sea continuo diremos que este tiene componentes independientes si y solo si la función de densidad conjunta del vector se factoriza mediante la función de densidad de cada coordenada, o sea

$$X_1, \cdots, X_n \quad \text{son independientes} \quad \Leftrightarrow \quad f_{\mathbf{X}}(x_1, \cdots, x_n) = \prod_{i=1}^{n} f_{X_i}(x_i).$$

Análogamente,

$$X_1, \cdots, X_n \quad \text{son independientes} \quad \Leftrightarrow \quad F_{\mathbf{X}}(x_1, \cdots, x_n) = \prod_{i=1}^{n} F_{X_i}(x_i).$$

De forma análoga se puede definir la independencia para vectores aleatorios.

Definición 1.2.5. Diremos que dos vectores aleatorios $\mathbf{X} = (X_1, \cdots, X_n)$ e $\mathbf{Y} = (Y_1, \cdots, Y_m)$ son independientes si

$$F_{\mathbf{XY}}(x_1, \cdots, x_n, y_1, \cdots, y_m) = F_{\mathbf{X}}(x_1, \cdots, x_n) F_{\mathbf{Y}}(y_1, \cdots, y_m).$$

Definición 1.2.6. Sea $(X_1, X_2, \cdots, X_n)$ un vector aleatorio n-dimensional. Sea $g : \mathbb{R}^n \to \mathbb{R}$ una función. Consideremos la variable aleatoria $g(X_1, X_2, \cdots, X_n)$. Entonces su esperanza está dada por

$$E[g(X_1, X_2, \cdots, X_n)] = \int \int \cdots \int g(x_1, x_2, \cdots, x_n) dF_{(X_1, X_2, \cdots, X_n)}(x_1, x_2, \cdots, x_n).$$

Veamos el caso particular en el que se trabaja con un vector de dos variables aleatorias, es decir $\mathbf{X} = (X, Y)$.

Lema 1.2.7. Si X e Y son independientes entonces $E[XY] = E[X]E[Y]$. Más generalmente, si tenemos dos funciones $g_1 : \mathbb{R} \to \mathbb{R}$ y $g_2 : \mathbb{R} \to \mathbb{R}$ se tiene

$$E[g_1(X) g_2(Y)] = E[g_1(X)] E[g_2(Y)].$$

Definición 1.2.8. Dadas dos variables aleatorias X e Y definimos la covarianza entre ellas mediante la fórmula

$$cov(X, Y) = E[(X - \mu_X)(Y - \mu_Y)],$$

siendo $\mu_X = E[X]$ y $\mu_Y = E[Y]$.

Observación 1.2.9. También podemos escribir a la covarianza como

$$cov(X, Y) = E[XY] - E[X]E[Y].$$

Definición 1.2.10. Dado un vector aleatorio $\mathbf{X} = (X_1, \cdots, X_n)$ de dimensión n se nota Σ y define la matriz de covarianza como

$$\Sigma = \begin{pmatrix} cov(X_1, X_1) & cov(X_1, X_2) & \cdots & cov(X_1, X_n) \\ cov(X_2, X_1) & cov(X_2, X_2) & \cdots & cov(X_2, X_n) \\ \vdots & \vdots & \ddots & \vdots \\ cov(X_n, X_1) & cov(X_n, X_2) & \cdots & cov(X_n, X_n) \end{pmatrix}$$

Notemos que la (i, j)-ésima entrada de la matriz Σ corresponde a la varianza entre X_i y X_j, que puede ser representada como

$$\Sigma_{ij} = E[(X_i - \mu_i)(X_j - \mu_j)] = cov(X_i, X_j).$$

Una propiedad importante que tiene la covarianza la resume el siguiente corolario:

Corolario 1.2.11. Si X e Y son independientes entonces $cov(X, Y) = 0$.

Una variable aleatoria importante y que utilizaremos en este trabajo será la distribución gaussiana, tanto es su versión unidimensional como en su versión multivariada.

Definición 1.2.12. Diremos que una variable aleatoria X tiene *distribución normal* y la notamos $X \sim \mathcal{N}(\mu, \sigma^2)$ si su función de densidad viene dada por

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}} \qquad x \in \mathbb{R},$$

donde $\mu, \sigma \in \mathbb{R}$ y $\sigma > 0$.

Definición 1.2.13. Diremos que un vector aleatorio $\mathbf{X}$ tiene distribución *distribución Gaussiana Multivariada* ó *Normal Multivariada* y lo notamos $\mathbf{X} \sim \mathcal{N}_n(\mu, \Sigma)$ si su función de densidad viene dado por

$$f_{\mathbf{X}}(x_1, \cdots, x_n) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)} \qquad (x_1, \cdots, x_n) \in \mathbb{R}^n,$$

con $\mu \in \mathbb{R}^n$ un vector real y $\Sigma \in \mathbb{R}^{n\times n}$ una matriz simétrica y semidefinida positiva.

En el caso univariado, los parámetros μ y σ^2 se corresponden a la media y la varianza, respectivamente. La distribución gaussiana multivariada es una generalización de la distribución normal unidimensional a dimensiones superiores. En este caso, los parámetros μ y Σ están asociados al vector de medias y la matriz de covarianza respectivamente.

Un resultado importante en este tipo de distribuciones está dado por la siguiente proposición.

Proposición 1.2.14. Si (X, Y) es un vector con distribución Normal Multivariada y $cov(X, Y) = 0$, entonces X e Y son independientes.

Notemos que en el corolario 1.2.11 esta proposición se reduce a una sola implicación y es importante recordar que en el caso en el que la distribución es Normal Multivariada, la otra implicación también es válida.

Algunos elementos básicos de estimación

Si se considera un vector muestra $\mathbf{X} = (X_1, X_2, \cdots, X_n)$ cuya distribución se sabe que pertenece a una familia $\mathcal{F} = \{F(x_1, x_2, \cdots, x_n, \theta)$ donde $\theta = (\theta_1, \cdots, \theta_p) \in \Theta \subset \mathbb{R}^p\}$ y además, se tiene que la función de densidad pertenece a una familia $f(\mathbf{x}, \theta)$, con $\theta \in \Theta$. Es posible estimar θ de diferentes maneras mediante el uso de estimadores. Un tipo de estimadores útiles son los estimadores de máxima verosimilitud, que se definen de la siguiente forma.

Definición 1.2.15. Diremos que $\widehat{\theta}(\mathbf{X})$ es un *estimador de máxima verosimilitud* (EMV) de θ, si se cumple

$$L(\mathbf{X}, \widehat{\theta}(\mathbf{X})) = \max_{\theta \in \Theta} f(\mathbf{X}, \theta)$$

,

donde L es la función de máxima verosimilitud.

Observación 1.2.16. Es este trabajo notaremos a los estimadores como $\widehat{\theta}$, es decir, sin aclarar el vector de la muestra.

Nos será útil conocer algunos estimadores de las distribuciones mencionadas.

Proposición 1.2.17. Sean $X_1, X_2, \cdots, X_n$ una muestra aleatoria de una distribución $\mathcal{N}(\mu, \sigma^2)$ entonces los EMV de μ y σ^2 están dados por $\widehat{\mu} = \overline{X}$ y $\widehat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - \overline{X})^2$.

Proposición 1.2.18. Sea una muestra aleatoria de datos en una matriz de dimensión $n \times p$. Si asumimos que los datos siguen una distribución Normal Multivariada con parámetros μ y matriz de covarianza Σ, entonces los estimadores de máxima verosimilitud están dados por

- $\widehat{\mu} = \overline{X}$.
- $\widehat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}(X^i - \widehat{\mu})(X^i - \widehat{\mu})^T$.

donde $X^i = (x_1^i, \cdots, x_p^i)$, para $i = 1, \cdots, n$ es cada una de las filas de la matriz.

Para probar la proposición necesitaremos las siguientes definiciones y usaremos los siguientes lemas.

Definición 1.2.19.

Lema 1.2.20. Sea $A \in \mathbb{R}^{n \times n}$ una matriz simétrica y $w \in \mathbb{R}^{n \times 1}$ un vector entonces

1. $\dfrac{\partial}{\partial w} w^T A w = 2Aw$.
2. $w^T A w = tr(w^T A w) = tr(A w w^T)$.

Lema 1.2.21. Sean $A, B \in \mathbb{R}^{n \times n}$ matrices entonces

1. $tr(AB) = tr(BA)$.
2. $\dfrac{\partial}{\partial A} tr(AB) = B^T$.

3. $\dfrac{\partial}{\partial A} ln(A) = (A^{-1})^T = (A^T)^{-1}$.

Lema 1.2.22. Usando lo anterior, dada $A \in \mathbb{R}^{n\times n}$ una matriz simétrica y $w \in \mathbb{R}^{n\times 1}$ un vector se tiene que

$$\frac{\partial}{\partial A}(w^T A w) = \frac{\partial}{\partial A}(tr(A w w^T)) = (w w^T)^T = (w^T)^T w^T = w w^T.$$

Estamos en condiciones de demostrar la proposición 1.2.18.

Demostración. Hacemos la demostración para el caso multivariado. Para probar la proposición 1.2.17 basta con considerar el caso unidimensional de la distribución Normal Multivariada.

Supongamos que tenemos n vectores aleatorios, cada uno de tamaño p, es decir $X^1, X^2, \cdots, X^n$ que supondremos i.i.d, notemos que cada $X^i = (x_1^i, \cdots, x_p^i)$, para $i = 1, \cdots, n$. Si cada X^i es un vector gaussiano multivariado, entonces

$$X^i \sim \mathcal{N}_p(\mu, \Sigma).$$

La función de máxima verosimilitud está dada por

$$L(\mu, \Sigma, X^1, \cdots, X^n) = \prod_{i=1}^{n} f_{X^i}(X^i).$$

Recordemos que encontrar máximos de la función de verosimilitud L es equivalente a encontrar máximos de $log(L)$, ya que el logaritmo es una función creciente, por lo tanto, tomando logaritmo a la función de verosimilitud se tiene

$$
\begin{aligned}
l(\mu, \Sigma, X^1, \cdots, X^n) &= log(L(\mu, \Sigma, X^1, \cdots, X^n)) \\
&= log\left(\prod_{i=1}^{n} f_{X^i}(X^i)\right) \\
&= log\left(\prod_{i=1}^{n} \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(X^i-\mu)^T\Sigma^{-1}(X^i-\mu)}\right) \\
&= log\left(\frac{1}{(2\pi)^{np/2}|\Sigma|^{n/2}} e^{-\frac{1}{2}\sum_{i=1}^{n}(X^i-\mu)^T\Sigma^{-1}(X^i-\mu)}\right) \\
&= -\frac{np}{2}log(2\pi) - \frac{n}{2}log(|\Sigma|) - \frac{1}{2}\sum_{i=1}^{n}(X^i-\mu)^T\Sigma^{-1}(X^i-\mu).
\end{aligned}
$$

Derivando respecto de μ y usando el lema 1.2.20 obtenemos

$$
\begin{aligned}
\frac{\partial}{\partial\mu} l(\mu, \Sigma, X^1, \cdots, X^n) &= -\frac{1}{2}\frac{\partial}{\partial\mu}\left(\sum_{i=1}^{n}(X^i-\mu)^T\Sigma^{-1}(X^i-\mu)\right) \\
&= -\frac{1}{2}\sum_{i=1}^{n}\frac{\partial}{\partial\mu}\left((X^i-\mu)^T\Sigma^{-1}(X^i-\mu)\right) \\
&= -\frac{1}{2}\sum_{i=1}^{n} 2\Sigma^{-1}(X^i-\mu)\underbrace{\left(\frac{\partial}{\partial\mu}(X^i-\mu)\right)}_{=-1} \\
&= \sum_{i=1}^{n}\Sigma^{-1}(X^i-\mu).
\end{aligned}
$$

Luego, igualando a 0 despejamos μ

$$
\begin{aligned}
0 &= \sum_{i=1}^{n} \Sigma^{-1}(X^i - \mu) \\
0 &= \Sigma^{-1} \sum_{i=1}^{n} X^i - n\Sigma^{-1}\mu \\
\mu &= \frac{1}{n} \sum_{i=1}^{n} X^i.
\end{aligned}
$$

Quedando así,

$$
\widehat{\mu} = \frac{1}{n} \sum_{i=1}^{n} X^i.
$$

Para hallar $\widehat{\Sigma}$ retomamos la expresión de logaritmo de la función de máxima verosimilitud y usaremos el lema 1.2.20

$$
\begin{aligned}
l(\mu, \Sigma, X^1, \cdots, X^n) &= -\frac{np}{2} log(2\pi) - \frac{n}{2} log(|\Sigma|) - \frac{1}{2} \sum_{i=1}^{n} (X^i - \mu)^T \Sigma^{-1} (X^i - \mu) \\
&= -\frac{np}{2} log(2\pi) - \frac{n}{2} log(|\Sigma|) - \frac{1}{2} \sum_{i=1}^{n} tr\left[(X^i - \mu)^T \Sigma^{-1} (X^i - \mu)\right] \\
&= -\frac{np}{2} log(2\pi) - \frac{n}{2} log(|\Sigma|) - \frac{1}{2} \sum_{i=1}^{n} tr\left[\Sigma^{-1} (X^i - \mu)(X^i - \mu)^T\right] \\
&= -\frac{np}{2} log(2\pi) - \frac{n}{2} log(|\Sigma|) - \frac{1}{2} tr\left[\Sigma^{-1} \sum_{i=1}^{n} (X^i - \mu)(X^i - \mu)^T\right].
\end{aligned}
$$

Luego, derivando respecto de Σ, y usando el lema 1.2.22 se obtiene

$$\frac{\partial}{\partial\Sigma}l(\mu,\Sigma,X^1,\cdots,X^n) = -\frac{n}{2}\frac{\partial}{\partial\Sigma}log(\Sigma^{-1}) - \frac{1}{2}\frac{\partial}{\partial\Sigma}tr\left[\Sigma^{-1}\sum_{i=1}^{n}(X^i-\mu)(X^i-\mu)^T\right]$$

$$= \frac{n}{2}\Sigma - \frac{1}{2}\sum_{i=1}^{n}(X^i-\mu)(X^i-\mu)^T.$$

Luego, igualando a 0 podemos despejar Σ

$$0 = \frac{n}{2}\Sigma - \frac{1}{2}\sum_{i=1}^{n}(X^i-\mu)(X^i-\mu)^T$$

$$\frac{n}{2}\Sigma = \frac{1}{2}\sum_{i=1}^{n}(X^i-\mu)(X^i-\mu)^T$$

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n}(X^i-\mu)(X^i-\mu)^T.$$

Quedando así,

$$\widehat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}(X^i-\widehat{\mu})(X^i-\widehat{\mu})^T.$$

□

Capítulo 2

Modelo y estimador propuesto

En este capítulo describiremos el modelo estadístico que proponemos para la detección de puntos de independencia y desarrollaremos un método de estimación para el modelo propuesto.

2.1. Bloques independientes

La clasificación de bloques independientes es un método de segmentación y clasificación de las variables de observaciones multivariadas que permite identificar secciones o bloques que no están relacionados con otros. A menudo se utiliza para ayudar a analizar tendencias complejas en los datos, permitiendo dividir el problema en pequeños problemas más simples.

Para comenzar definamos que es un bloque de un vector aleatorio.

Definición 2.1.1. Sea $\mathbf{X} \sim F$ un vector aleatorio que toma valores en $\mathbb{R}^d$ con F su función de distribución acumulada. Dados u, v con $1 \leq u < v < d$, llamamos *bloque* de $\mathbf{X}$ al subvector $\mathbf{X}_{u:v} = (X_{u+1}, ..., X_v)$ en $\mathbb{R}^{v-u}$ y notamos a su función de distribución conjunta como $F_{u:v}$.

En el caso en que quisiéramos considerar el bloque cuyo primer elemento es X_1 lo notaremos $\mathbf{X}_{1:v} = (X_1, ..., X_v)$ en $\mathbb{R}^v$ con función de distribución conjunta $F_{1:v}$.De esta forma, dado cualquier vector aleatorio multivariado $\mathbf{X} \sim F$ en $\mathbb{R}^d$ y dados u, v con $1 \leq u < v < d$ podemos definir los siguientes bloques de $\mathbf{X}$,

$$\mathbf{X}_{1:u} = (X_1, ..., X_u), \qquad \mathbf{X}_{u:v} = (X_{u+1}, ..., X_v), \qquad \mathbf{X}_{v:d} = (X_{v+1}, ..., X_d),$$

y $F_{1:u}$, $F_{u:v}$ y $F_{v:d}$ las funciones conjuntas de $\mathbf{X}_{1:u}$ $\mathbf{X}_{u:v}$ y $\mathbf{X}_{v:d}$ respectivamente.

Notemos que con sólo conocer los índices del último elemento de cada bloque, podemos partir cualquier vector aleatorio en la cantidad de bloques que necesitemos. Por ejemplo, si ahora tenemos un conjunto de índices $U = \{u_1, \cdots, u_k\}$ los bloques que nos induce ese conjunto de índices sería $\mathbf{X}_{1:u_1}$, $\mathbf{X}_{u_i:u_{i+1}}$ $(i = 1, \cdots, k-1)$ y $\mathbf{X}_{u_k:d}$.

Nuestro principal interés es hallar un conjunto de índices con la propiedad de que los bloques inducidos sean independientes. En particular, buscaremos que estos bloques sean lo más pequeños posible. Para describir formalmente que queremos decir con que sean lo más chicos posible, introduciremos la definición de una nueva función F_U.

Definición 2.1.2. Dados un vector aleatorio $\mathbf{X} = (X_1, ..., X_d)$ con función de distribución F y un conjunto de índices, $U = \{u_1, ..., u_k\}$ llamamos *U-producto* a la función

$$F_U(x_1, ..., x_d) = F_{1:u_1}(x_1, ..., x_{u_1}) \prod_{i=1}^{k-1} F_{u_i:u_{i+1}}(x_{u_i+1}, ..., x_{u_{i+1}}) F_{u_k:d}(x_{u_k+1}, ..., x_d).$$

Notemos que con el *U-producto* vamos a poder definir la noción de independencia para bloques de un vector aleatorio $\mathbf{X}$ ya que si $U = \{u_1, ..., u_k\}$ se corresponde con un conjunto de índices que define bloques independientes, entonces necesariamente debe suceder que F_U coincida con F. En el caso en que $U = \emptyset$ se define $F_U = F$. Esto da origen a la siguiente definición:

Definición 2.1.3. Dado un vector aleatorio $\mathbf{X}$ y un conjunto de índices $U = \{u_1, ..., u_k\}$, con $1 \leq u_1 < ... < u_k < d$ diremos que U es un *conjunto de independencia* para F si $\mathbf{X}_{1:u_1}$, $\mathbf{X}_{u_i:u_{i+1}}$ y $\mathbf{X}_{u_k:d}$ son independientes para todo $i = 1, ..., k-1$. Es decir,

$$F(x_1, \cdots, x_d) = F_U(x_1, \cdots, x_d). \tag{1}$$

Observación 2.1.4. Además notemos que si U es un conjunto de independencia para F entonces cualquier subconjunto $\tilde{U} \subset U$ también es un conjunto de independencia para F. Y además, si U y V son conjuntos de independencia para F entonces $U \cup V$ también es un conjunto de independencia para F.

Demostración. □

Estas observaciones nos inducen a considerar el conjunto más grande de independencia, en el sentido de que cualquier otro conjunto de independencia está incluido en él y que introduciremos en la siguiente definición.

Definición 2.1.5. $U^*(F)$ se dice *conjunto maximal de independencia* para F, si para todo U conjunto de independencia para F, entonces $U \subseteq U^*(F)$.

Ejemplo 2.1.6. Sea $\mathbf{X} = (X_1, X_2, X_3, X_4, X_5)$, con $\mathbf{X} \sim F$.

1 Si $U^*(F) = \{1, 4\}$ es el conjunto maximal de independencia, entonces (X_1), (X_2, X_3, X_4) y (X_5) son independientes.

2 Si $U^*(F) = \{1, 2, 3, 4, 5\}$ entonces X_1, X_2, X_3, X_4 y X_5 son independientes.

3 Si $U^*(F) = \{\emptyset\}$ entonces no podemos partir al vector aleatorio $\mathbf{X}$ en subvectores propios independientes.

Nuestro objetivo es estimar $U^*(F)$ a partir de una muestra aleatoria $\{\mathbf{X}^i : 1 \leq i \leq n\}$, con $\mathbf{X}^i \sim F$.

2.2. Propuestas de estimadores

2.2.1. Basados en estimadores de las distribuciones

Para estimar $U^*(F)$ primero definamos algún modelo para F, es decir, alguna distribución que modele correctamente a la distribución del vector aleatorio $\mathbf{X}$. En este trabajo se van a presentar dos modelos. En el primero no asumimos ninguna distribución para el vector aleatorio y calculamos probabilidades acumuladas usando la función empírica. En el segundo modelo asumimos que cada vector aleatorio $\mathbf{X}$ sigue una distribución Normal Multivariada y usamos estimadores paramétricos bajo esta distribución.

Algo que nos va a ser útil es saber si dado un conjunto de índices $U = \{u_1, u_2, \cdots, u_k\}$ podemos determinar que tan distinta es la función de distribución conjunta F respecto de su *U-producto*. Por esto necesitamos definir alguna medida para comparar a F con la función F_U, considerando la siguiente definición:

Definición 2.2.1. Dados F y un conjunto U, llamamos *discrepancia* entre F y su *U-producto* a

$$\ell(U, F) = \sup_{\mathbf{x} \in \mathbf{R}^d} |F_U(\mathbf{x}) - F(\mathbf{x})|. \tag{2}$$

Observación 2.2.2. Notemos que U es un conjunto de independencia para F si y solo si $F_U \equiv F$, lo que significa que $\ell(U, F) = 0$ en este caso.

Más aún si notamos a $U^* = U^*(F)$ al conjunto maximal de independencia para F, entonces podemos decir que existe $\alpha > 0$ tal que

$$\ell(U, F) = 0 \qquad \forall \quad U \subseteq U^*,$$

$$\ell(U, F) > \alpha \qquad \forall \quad U \not\subseteq U^*.$$

Esto último nos dice que para hallar U^* podemos proceder minimizando $\ell(U, F)$ lo cual sugiere un método empírico para estimar a U^*. Una observación importante es que no nos basta con hallar un conjunto de independencia que verifique que $\ell(U, F) = 0$ porque eso no garantiza que el conjunto sea el maximal. Según la observación (2.1.4), U^* debe ser el mayor de estos conjuntos que cumplen $\ell(U, F) = 0$.

Para calcular un estimador del conjunto U^* primero necesitamos definir algún estimador $\widehat{F}$ para F basado en una muestra aleatoria dada por el proceso $\mathbf{X}^n = \{\mathbf{X}^{(i)} : 1 \leq i \leq n\}$ con la distribución F. Notemos al estimador para esa muestra aleatoria $\widehat{F}_{\mathbf{X}^n}$. De una manera similar y usando la misma muestra, para cualquier conjunto $U = \{u_1, \cdots, u_k\}$, definiremos otro estimador $\widehat{F}_U$ para F_U. Con esto vamos a poder estimar la discrepancia como

$$\ell(U, \widehat{F}_{\mathbf{X}^n}) = \sup_{\mathbf{x} \in \mathbf{R}^d} |\widehat{F}_U(\mathbf{x}) - \widehat{F}_{\mathbf{X}^n}(\mathbf{x})|$$

Observación 2.2.3. Estudiaremos dos casos específicos de estimadores de F pero en principio lo que vamos a desarrollar se aplica a cualquier estimador de F.

Como $\ell(U, F) = 0$ para todo $U \subseteq U^*(F)$ para asegurarnos que el estimador nos dé al U más grande posible vamos a penalizar a la discrepancia agregándole un término que contempla la cantidad de elementos de U, de esta forma vamos a buscar minimizar

$$PL(U, \mathbf{X}^n) = \ell(U, \widehat{F}_{\mathbf{X}^n}) + \lambda_n(|U| + 1)^{-1}, \tag{3}$$

donde $|U|$ es el cardinal del conjunto U y λ_n es un cierto parámetro de penalización. De esta forma, si notamos al conjunto de independencia que buscamos estimar como $\widehat{U}_n$, lo calcularemos buscando

$$\widehat{U}_n = \operatorname*{arg\,mín}_{U \subseteq \{1, \ldots, d-1\}} PL(U, \mathbf{X}^n). \tag{4}$$

Luego, se tiene que $\widehat{U}_n$ satisface

$$PL(\widehat{U}_n, \mathbf{X}^n) \leq PL(U, \mathbf{X}^n) \quad \text{para todo} \quad U \subseteq \{1, \ldots, d-1\}. \tag{5}$$

Esta primera propuesta de estimación está basada en poder estimar la función de distribución de un modo razonable. En particular, a partir de una muestra dada por el proceso $\mathbf{X}^n = \{\mathbf{X}^{(i)} : 1 \leq i \leq n\}$ con distribución F, describiremos dos propuestas

de estimación $\widehat{F}_{\mathbf{X}^n}$ de F. La primera de ella es considerar la función de distribución empírica, que fue discutida en [3].

Si consideramos como estimador de F a la función empírica, es decir,

$$\widehat{F}_{\mathbf{X}^n}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} I_{\{\mathbf{X}^{(i)} \leq \mathbf{x}\}}, \tag{6}$$

y de forma similar, dado $U = \{u_1, \cdots, u_k\}$ podemos considerar un estimador de la función F_U basado en la distribución empírica como

$$\widehat{F}_U(\mathbf{x}) = \widehat{F}_{\mathbf{X}^n_{1:u_1}}(\mathbf{x}_{1:u_1}) \prod_{i=1}^{k-1} \widehat{F}_{\mathbf{X}^n_{u_i:u_{i+1}}}(\mathbf{x}_{u_i:u_{i+1}}) \widehat{F}_{\mathbf{X}^n_{u_{k+1}:d}}(\mathbf{x}_{u_i:u_{i+1}}). \tag{7}$$

La segunda propuesta es considerar un estimador paramétrico en particular; en esta tesis asumiremos que F sigue una distribución Normal Multivariada.

Otra forma de estimar la función de distribución es considerando un modelo paramétrico subyacente que gobierne los datos. Como comentamos anteriormente, en esta tesis supondremos que podemos asumir que nuestros datos fueron generados bajo un modelo Normal Multivariado. Es decir, asumimos que F sigue una distribución Gaussiana con parámetros μ y $\Sigma = \{\sigma_{i,j}\}$, y si bien no disponemos de una expresión analítica para F, podemos describirla y estimar su densidad a través de la estimación de su media y su matriz de covarianza, $E(\mathbf{X}) = \mu$ y $cov(\mathbf{X}) = \Sigma$.

En el caso en que tomemos el bloque $\mathbf{X}_{u:v}$, tendremos que $F_{u:v}$ sigue una distribución normal multivariada en $\mathbf{R}^{v-u}$, con $\mu_{u:v} = E(\mathbf{X}_{u:v}) = (\mu_{u+1}, \ldots, \mu_v)^T$ y matriz de covarianza $\Sigma_{u:v} = cov(\mathbf{X}_{u:v})$. Si se considera un conjunto de independencia U, entonces F_U sigue una distribución normal multivariada en $\mathbf{R}^d$ con parámetros μ y Σ_U. En este caso Σ_U es la matriz que se forma al reemplazar los coeficientes $\sigma_{i,j}$ de Σ por 0 cuando $i \leq u < j$ ó $j \leq u < i$, para cada $u \in U$.

Con lo cual podemos estimar $\widehat{U}_n$ definido en (4) considerando estimadores de F y F_U, estimando μ, Σ y Σ_U. Como describimos en el Capítulo 1 en (1.2.18), por ejemplo, se pueden usar estimadores de máxima verosimilitud, de esta forma $\widehat{\mu} = \overline{\mathbf{X}^n}$, $\widehat{\Sigma} = cov(\mathbf{X}^n)$ y $\widehat{\Sigma}_U$ dado por

$$(\widehat{\Sigma}_U)_{ij} = \begin{cases} 0 & \text{si} \quad i \leq u < j \text{ ó } j \leq u < i, \\ \\ cov(\mathbf{X}^n)_{ij} & \text{en caso contrario,} \end{cases} \tag{8}$$

donde $u \in U$.

Notemos que es $\widehat{\Sigma}_U$ es una matriz diagonal por bloques definida como

$$\widehat{\Sigma}_U = \begin{pmatrix} \Sigma_{1:u_1} & 0 & \cdots & 0 \\ 0 & \Sigma_{u_1:u_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Sigma_{u_k:n} \end{pmatrix}$$

con $U = \{u_1, \cdots, u_k\}$ un conjunto de índices y $\Sigma_{u:v} \in \mathbb{R}^{(v-u)\times(v-u)}$ las submatrices que se obtienen al considerar las filas $u+1, \cdots, v$ y las columnas $u+1, \cdots, v$ de la matriz de covarianza $\widehat{\Sigma} = cov(\mathbf{X}^n)$.

2.2.2. Basados en estimadores de las matrices de covarianzas

Como vimos, la propuesta anterior implica tener algún estimador de las funciones de distribución acumulada F. Dependiendo del estimador elegido, este planteo puede tener un costo de complejidad muy alto en los algoritmos, lo cual se verá con más detalle en el próximo capítulo.

La alternativa que vamos a presentar se basa en el supuesto de que F sigue una distribución normal multivariada, es decir $F \sim \mathcal{N}(\mu, \Sigma)$. Por lo tanto, para todo conjunto de índices U, $F_U \sim \mathcal{N}(\mu, \Sigma_U)$. Nuestra propuesta se basa en que en el caso Normal Multivariado podemos caracterizar la independencia a través de sus matrices de covarianza Σ y Σ_U definidas por F y F_U respectivamente. Luego, a partir de ellas podemos calcular una distancia entre matrices en vez de calcular el valor del estimador $\widehat{F}$ y $\widehat{F}_U$ en diferentes puntos, lo que nos permitirá reducir los costos computacionales. Para explicar esto tengamos en cuenta algunos resultados.

Definición 2.2.4. Sea un espacio medible $(\Omega, \mathcal{F})$ y P y Q medidas de probabilidad definidas en $(\Omega, \mathcal{F})$, la *distancia de variación total* entre P y Q se define como

$$\delta(P, Q) = \sup_{A \in \mathcal{F}} |P(A) - Q(A)|. \tag{9}$$

Luego, notemos que si $F \sim \mathcal{N}(\mu, \Sigma)$ y $F_U \sim \mathcal{N}(\mu, \Sigma_U)$ tenemos que

$$\delta(F_U, F) = \sup_{x \in \mathbf{R}^d} |F_U(\mathbf{x}) - F(\mathbf{x})|.$$

De esta forma, calcular $\ell(U, F)$ es equivalente a calcular $\delta(F_U, F)$. Y para hallar $\widehat{U}_n$ se buscará minimizar la siguiente función

$$PL(U, \mathbf{X}^n) = \delta(\widehat{F}_U, \widehat{F}) + \lambda_n(|U| + 1)^{-1}.$$

Es decir, utilizar la discrepancia o la distancia en variación total es lo mismo y no basta para reducir la complejidad del problema. Para ello consideremos la siguiente distancia

que se utiliza para cuantificar la similitud entre dos distribuciones de probabilidad y está fuertemente relacionada con la distancia de variación total.

Definición 2.2.5. Sean P y Q dos medidas de probabilidad en un espacio medible χ que son absolutamente continuas con respecto a una tercera medida de probabilidad λ, con $\lambda = P + Q$ entonces definimos el *cuadrado de la distancia de Hellinger* entre P y Q como

$$H^2(P,Q) = \frac{1}{2}\int_{\chi}\left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2 d\lambda,$$

donde $P(x) = p(x)\lambda(x)$ y $Q(x) = q(x)\lambda(x)$ son las derivadas de Radon-Nikodym de P y Q respectivamente.

Esta definición no depende de λ, por lo que la distancia de Hellinger entre P y Q no cambia si λ se reemplaza con una medida de probabilidad diferente con respecto a la cual tanto P como Q son absolutamente continuas.

Veamos ahora como se define la distancia de Hellinger en el caso en el que se tengan distribuciones Normales Multivariadas. La derivación de esta fórmula se halla en [9]

Definición 2.2.6. Dadas $G \sim \mathcal{N}(\mu_1, \Sigma_1)$ y $\widetilde{G} \sim \mathcal{N}(\mu_2, \Sigma_2)$ definimos el *cuadrado de la distancia de Hellinger* como

$$H^2(G,\widetilde{G}) = 1 - \frac{det(\Sigma_1)^{\frac{1}{4}} det(\Sigma_2)^{\frac{1}{4}}}{det\left(\frac{\Sigma_1+\Sigma_2}{2}\right)^{\frac{1}{2}}} e^{-\frac{1}{8}(\mu_1-\mu_2)^T\left(\frac{\Sigma_1+\Sigma_2}{2}\right)^{-1}(\mu_1-\mu_2)}.$$

Notemos que $F \sim \mathcal{N}(\mu, \Sigma)$ y dado un conjunto de índices U, recordemos que $F_U \sim \mathcal{N}(\mu, \Sigma_U)$ y entonces el cuadrado de la distancia de Hellinger está dado por

$$H^2(F_U, F) = 1 - \frac{det(\Sigma_U)^{\frac{1}{4}} det(\Sigma)^{\frac{1}{4}}}{det\left(\frac{\Sigma_U+\Sigma}{2}\right)^{\frac{1}{2}}}.$$

Además de que la distancia de Hellinger para el caso de distribuciones gaussianas tiene una expresión sencilla, tiene la siguiente propiedad.

Proposición 2.2.7. La distancia de Hellinger $H(G,\widetilde{G})$ y la distancia de variación total $\delta(G,\widetilde{G})$ se relacionan de la siguiente manera

$$H^2(G,\widetilde{G}) \leq \delta(G,\widetilde{G}) \leq \sqrt{2}H(G,\widetilde{G}).$$

La demostración se encuentra en [11].

Esto nos induce a considerar bajo el supuesto de normalidad otra función de pérdida, redefiniendo la función que buscamos minimizar como

$$PL(U, \mathbf{X}^n) = H^2(\widehat{F}_U, \widehat{F}) + \lambda_n(|U| + 1)^{-1}.$$

En este caso utilizando estimadores paramétricos bajo el modelo normal, necesitamos calcular $H^2(\widehat{F}_U, \widehat{F})$ que como vimos depende solamente de $\widehat{\Sigma}$ y $\widehat{\Sigma}_U$. Por lo tanto, bastará con considerar $\widehat{\Sigma}$ y $\widehat{\Sigma}_U$ que definimos en la subsección anterior.

2.3. Resultados teóricos

En esta sección vamos a mostrar los argumentos teóricos que justifican el uso de los estimadores que definimos basados en estimadores de las funciones de distribución. El primer resultado a estudiar es la consistencia de $\widehat{U}_n$ siempre que el término de penalización y la tasa de convergencia de $\widehat{F}_{\mathbf{x}_n}$ satisfagan ciertas condiciones.

Teorema 2.3.1. Supongamos que

$$\sup_{\mathbf{x}\in\mathbf{R}^d} |\widehat{F}_{\mathbf{X}^n}(\mathbf{x}) - F(\mathbf{x})| \leq a_n \tag{10}$$

casi seguramente cuando $n \to \infty$. Si $\lambda_n \to 0$ y $\dfrac{a_n}{\lambda_n} \to \infty$ entonces $\widehat{U}_n = U^*$ eventualmente en forma casi segura cuando $n \to \infty$

Demostración. Mostraremos que eventualmente en forma casi segura cuando $n \to \infty$,

$$\mathrm{PL}(U^*, \mathbf{X}^n) < \mathrm{PL}(U, \mathbf{X}^n)\,, \quad \text{para todo } U \subseteq \{1, \ldots, d-1\}\,, \tag{11}$$

lo que quiere decir que $\widehat{U}_n = U^*$. Para lograrlo, notemos que

$$|\ell(U, \widehat{F}_{\mathbf{X}^n}) - \ell(U, F)| \leq \sup_{\mathbf{x}\in\mathbb{R}^d} |\widehat{F}_{n,U}(\mathbf{x}) - F_U(\mathbf{x})| + \sup_{\mathbf{x}\in\mathbb{R}^d} |\widehat{F}_n(\mathbf{x}) - F(\mathbf{x})|.$$

Por otro lado, $\sup_{\mathbf{x}\in\mathbb{R}^d} |\widehat{F}_{n,u:v}(\mathbf{x}) - F_{u:v}(\mathbf{x})| \leq \sup_{\mathbf{x}\in\mathbb{R}^d} |\widehat{F}_n(\mathbf{x}) - F(\mathbf{x})|$. Entonces, según la condición (10), eventualmente en forma casi segura cuando $n \to \infty$,

$$|\ell(U, \widehat{F}_{\mathbf{X}^n}) - \ell(U, F)| \leq |U| a_n + a_n \leq d\, a_n. \tag{12}$$

Para probar (11), consideremos ahora U que no esté contenido en U^*, $U \not\subset U^*$, en cuyo caso, $\ell(U, F) > \alpha > 0$. Por lo tanto, usando (12), obtenemos que

$$\begin{aligned} \mathrm{PL}(U, \mathbf{X}^n) - \mathrm{PL}(U^*, \mathbf{X}^n) &= \ell(U, \widehat{F}_{\mathbf{X}^n}) - \ell(U^*, \widehat{F}_{\mathbf{X}^n}) + \lambda_n \left\{(|U| + 1)^{-1} - (|U^*| + 1)^{-1}\right\} \\ &\geq -d\, a_n + \alpha - d\, a_n + \lambda_n \left\{(|U| + 1)^{-1} - (|U^*| + 1)^{-1}\right\}. \end{aligned}$$

Como $\alpha > 0$ y tanto λ_n como a_n convergen a cero eventualmente en forma casi segura cuando $n \to \infty$, obtenemos que

$$\mathrm{PL}(U, \mathbf{X}^n) - \mathrm{PL}(U^*, \mathbf{X}^n) > 0 \,. \tag{13}$$

Si $U^* = \emptyset$, no hace falta considerar ningún otro caso. Si no, tomemos U tal que U esté estrictamente contenido en U^*, es decir $U \subsetneq U^*$. Por lo tanto, $\ell(U, F) = \ell(U^*, F) = 0$ y entonces,

$$\begin{aligned}\mathrm{PL}(U, \mathbf{X}^n) - \mathrm{PL}(U^*, \mathbf{X}^n) &= \ell(U, \widehat{F}_{\mathbf{X}^n}) - \ell(U^*, \widehat{F}_{\mathbf{X}^n}) + \lambda_n \left\{ (|U|+1)^{-1} - (|U^*|+1)^{-1} \right\} \\ &\geq -2d\, a_n + \lambda_n \left\{ (|U|+1)^{-1} - (|U^*|+1)^{-1} \right\}.\end{aligned}$$

Finalmente, como $U \subsetneq U^*$, tenemos que $|U| + 1 \leq |U^*|$ y entonces

$$\frac{1}{|U|+1} - \frac{1}{|U^*|+1} \;\geq\; \frac{1}{d^*(d^*+1)} > \frac{1}{d(d+1)}.$$

Como $d\, a_n / \lambda_n \to 0$, concluimos que, para n suficientemente grande, $\frac{1}{d(d+1)} > \frac{2a_n}{\lambda_n}$, lo que implica que

$$\mathrm{PL}(U, \mathbf{X}^n) - \mathrm{PL}(U^*, \mathbf{X}^n) > 0 \,,$$

eventualmente en forma casi segura cuando $n \to \infty$. □

Observación 2.3.2. La convergencia de $\widehat{U}_n$ a U se cumple siempre que la distribución empírica $\widehat{F}_{\mathbf{X}^n}$ converja uniformemente a la distribución F para una cierta tasa a_n. Esta tasa de convergencia va a inducir la elección del factor de penalización λ_n del teorema anterior. Queda para un futuro estudio la demostración de que el caso de considerar estimadores de la función de distribución basados en el modelo Normal Multivariado, también se satisfacen las condiciones del Teorema.

Corolario 2.3.3. Supongamos que $\{\mathbf{X}^{(i)} : i \geq 1\}$ son independientes e idénticamente distribuidas y sea $\widehat{F}_{\mathbf{X}^n}$ la distribución empírica o gaussiana definidas en la Sección 2.2 para estimar F. Considerando $\lambda_n = cn^{-\xi}$, con $\xi \in (0, 1/2)$ entonces, $\widehat{U}_n = U^*$ casi seguramente cuando $n \to \infty$.

Un resultado análogo estudiado en [3] se puede demostrar en algunos casos en que el proceso $\mathbf{X}^{(i)}$ no sea *i.i.d* asumiendo algunas condiciones sobre la dependencia del proceso.

Finalmente, notemos que el teorema solo considera el caso en el que la función de pérdida está basada en la discrepancia. Se podría investiar a futuro los resultados teóricos empleando la distancia de Hellinger.

Capítulo 3

Implementación computacional de las propuestas

En esta sección discutiremos dos alternativas para calcular los estimadores propuestos y describimos los códigos implementados. En particular, discutiremos lo que denominamos el método exhaustivo y una alternativa con mejor complejidad computacional.

3.1. Explicación del código exhaustivo

Para calcular el estimador definido en (4) necesitamos calcular la función $PL(U, \mathbf{X}^n)$ para todos los posibles subconjuntos $U \subseteq \{1, 2, \cdots, d-1\}$. Luego, con todos estos valores, nos quedamos con el subconjunto de U que minimice a la función $PL(U, \mathbf{X}^n)$. Debido a que la cantidad de subconjuntos que podemos armar con U es de 2^d, si escribimos un algoritmo exhaustivo que calcule todos los posibles valores que puede tomar $PL(U, \mathbf{X}^n)$, tendrá una complejidad de $O(2^d T)$, donde T es el tiempo necesario para calcular $PL(U, \mathbf{X}^n)$. Esto hace que resolver este problema exhaustivamente se vuelva computacionalmente inviable a partir de valores grandes de d.

Observación 3.1.1. Si bien el tiempo T no es fijo, ya que podría depender de d, esta dependencia es como mucho lineal.

3.1.1. Algoritmo para estimador basado en estimadores de distribuciones

El caso del estimador empírico ya fue realizado por [8] y en este trabajo escribimos el código para el estimador normal multivariado. Para esto último se escribieron las siguientes dos funciones cuyos códigos se detallan en el Apéndice.

La función **multinormal_bloques** toma como parámetros una matriz de datos, un vector y un conjunto de índices U y computa el valor del estimador para el caso normal multivariado usando la matriz de datos para calcular los estimadores de $\mu = (\mu_1, \cdots, \mu_d)$, donde cada coordenada es el promedio de cada una de las columnas y $\Sigma = \Sigma_U$ es la matriz de covarianza de que se construye con la matriz de datos y los índices U como se explica en (8).

La función **acumulada_normal** toma los mismos parámetros que la función anterior y calcula, en el caso en el que la matriz de datos tenga una sola columna, el valor de la función acumulada gaussiana, donde se estiman los valores de μ y σ como el promedio y el desvío estándar respectivamente. En el caso en el que la matriz de datos tenga dos o más columnas la función llama a la función *multinormal_bloques* para calcular el valor estimado con la acumulada normal multivariada.

Con los anteriores estimadores basta con escribir el código que calcule el valor del estimador $\widehat{U}_n$. Para ello se usaron las siguientes funciones:

La función **score_para_u_normal** recibe una matriz con los datos $\mathbf{X}^n$ y un conjunto de índices U y calcula la discrepancia $l(U, \widehat{F}_{\mathbf{X}^n})$.

La función **iteracion_exhaustivo_normal** recibe como parámetros la misma matriz de datos $\mathbf{X}^n$ y valores c y ξ definidos en (2.3.3) que usamos para definir al parámetro λ_n. Por otro lado, la cantidad de columnas de $\mathbf{X}^n$ nos define el conjunto U, es decir que si la matriz de datos tiene d columnas, entonces $U = \{1, \cdots, d\}$. En este bloque de código se usa la función *combn* para calcular todos los posibles subconjuntos de U y con estos valores se computa $PL(U, \mathbf{X}^n)$ usando la función *score_para_u_normal*, para todos los $U \subseteq \{1, \cdots, d\}$. Finalmente, este bloque devuelve el subconjunto de U que minimiza a PL.

Finalmente, la función principal que corre el código y utiliza al resto de las funciones está dado por **get_best_u_exacto_alg_normal**, la cual solo necesita como parámetros a la matriz de datos y el valor de c y ξ.

3.1.2. Algoritmo para estimador basado en estimadores de las covarianzas

Este caso solo se diferencia del anterior en el estimador que se utiliza, por lo que el algoritmo es bastante parecido. Para el cálculo del estimador se utilizaron las siguientes dos funciones.

La función **covarianza_bloques** toma como parámetros una matriz de datos $\mathbf{X}^n$ y un vector de índices U y devuelve la matriz de covarianza Σ_U.

La función **distancia_matrices** toma los mismos parámetros de la función anterior, construye las matrices Σ y Σ_U con la matriz de datos $\mathbf{X}^n$ y los índices U y luego calcula la distancia de Hellinger entre dichas matrices.

Luego, para calcular el valor de $\widehat{U}_n$ basta con una sola función. La función **iteracción_exhaustivo_normal** es casi idéntica a la usada en el algoritmo anterior, en particular recibe los mismos parámetros, pero esta utiliza el estimador de covarianza, por lo que llama en cada iteración a *distancia_matrices*. Finalmente, para terminar se escribe la función principal que corre el algoritmo llamada **get_best_u_exacto_alg_normal_covarianza**.

Observación 3.1.2. Notemos que en el caso del estimador dado por covarianza solo necesitamos comparar dos matrices generadas por la matriz de datos, a diferencia del caso del estimador empírico para el cual necesitamos la función *score_para_u_normal*. El evitar escribir esa función hará que el costo de complejidad del algoritmo con los estimadores de matrices de covarianza sea significativamente más bajo y, por lo tanto, este algoritmo debería ser más veloz.

3.2. Explicación del código con árbol binario

La alternativa para resolver este problema de complejidad computacional es usar un algoritmo con el método de *Divide and Conquer*. Este nuevo algoritmo tendrá una complejidad de $O(d^2T)$.

3.2.1. Divide and Conquer

Divide and Conquer (D&C) es una técnica que consiste en descomponer recursivamente un problema en dos o más subproblemas similares, hasta que se vuelven lo suficientemente simples como para resolverlos directamente. Las soluciones a los subproblemas luego se combinan para dar una solución al problema original.

Un algoritmo específico de D&C es el algoritmo de búsqueda binaria, en el cual en cada paso recursivo se divide el problema, en dos subproblemas, de dimensión menor, así siguiendo hasta llegar a casos bases. La estructura que se usa para modelar este algoritmo es el árbol binario.

Los árboles binarios tienen una complejidad algorítmica de $O(nlog(n))$, donde n es el tiempo que tarde el caso base y $log(n)$ es la cantidad de subproblemas que se tiene que resolver a lo sumo.

3.2.2. Árbol binario

El algoritmo dado por D&C construye un árbol binario en el que sus nodos son subintervalos de 1:d y los intervalos dados por los nodos terminales constituyen los bloques de independencia. Los últimos índices de cada uno de esos bloques forman al estimador de U^* que notaremos $\widehat{U}_n^{bin}$.

Como en cada paso del árbol binario se restringe el cálculo de $PL(U, X)$ en un cierto rango dado por los subintervalos de $1{:}d$ necesitamos redefinir como calcular la función de discrepancia en estos casos.

Definición 3.2.1. Para la función dada en (3) consideremos la siguiente función

$$PL(U, \mathbf{X}_{u:v}^n) = \ell(U, \widehat{F}_{\mathbf{X}_{u:v}^n}) + \lambda_n(|U| + 1)^{-1} \tag{1}$$

para todo $1 \leq u \leq v \leq d$ y $U \subset \{u, \cdots, v-1\}$, donde $|U|$ es el cardinal del conjunto U.

Además, debemos considerar

$$h(u:v, \mathbf{X}_{u:v}^n) = \underset{i \in u:v}{\arg\min}\{PL(\{i\}, \mathbf{X}_{u:v}^n)\} \tag{2}$$

donde, por convención, se tiene $PL(\{v\}, \mathbf{X}_{u:v}^n) = PL(\emptyset, \mathbf{X}_{u:v}^n)$ donde v es el elemento mas grande de $u:v$.

Con estas definiciones podemos describir el algoritmo del árbol binario como sigue

3.2.3. Algoritmo

Paso 1: Inicializamos

- $\hat{U}_n^{bin} = \emptyset$
- $I = 1:d$

Paso 2:

- Calculamos $h(I, \mathbf{X}_I^n)$
- SI $h(I, \mathbf{X}_I^n) < max(I)$:
  - Agregamos $h(I, \mathbf{X}_I^n)$ a $\hat{U}_n^{bin}$
  - Agregamos dos hojas al nodo I del árbol, cuyas etiquetas serán $I_1 = I \cap \{i : i \leq h(I, \mathbf{X}_I^n)\}$ e $I_2 = I \cap \{i : i > h(I, \mathbf{X}_I^n)\}$

Paso 3: Se repite el paso 2 para los nuevos nodos terminales del árbol hasta que no haya más nodos que agregar.

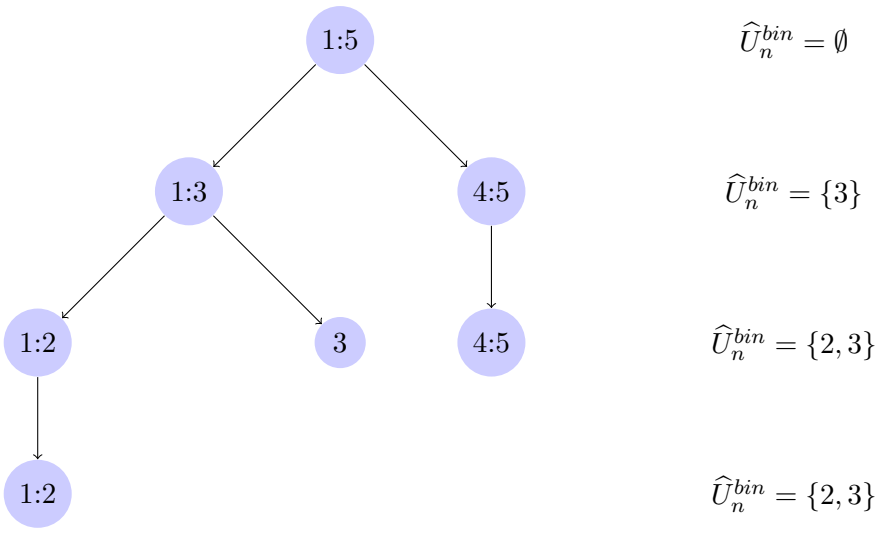
Finalmente, el conjunto de puntos de independencia $\widehat{U}_n^{bin}$ a estimar estará formado por los índices más altos de los nodos terminales del árbol, excluyendo la raíz y el índice d del intervalo principal.

Observación 3.2.2. Notemos, en el paso 2, que si $h(I, \mathbf{X}_I^n) = max(I)$ entonces no se agrega ningún elemento al conjunto $\widehat{U}_n^{bin}$ y, por lo tanto, el algoritmo se frena en esta rama del árbol.

En [3] se muestra la siguiente proposición.

Proposición 3.2.3. Bajo las mismas hipótesis del Teorema (2.3.1), se tiene que $\widehat{U}_n^{bin} = U^*$ en casi todo punto cuando $n \to \infty$.

Ejemplo 3.2.4. Supongamos que tenemos $U = \{1, 2, 3, 4, 5\}$, el algoritmo anterior busca el valor $i \in U$ que minimice $PL(\{i\}, \mathbf{X}^n)$. Supongamos que el valor hallado en el primer paso del algoritmo es $i = 3$, entonces agregamos 3 al conjunto $\widehat{U}_n^{bin}$ ya que este es un punto de independencia, quedando así $\widehat{U}_n^{bin} = \{3\}$. Luego, partimos el conjunto de índices en dos conjuntos $I_1 = \{1, 2, 3\}$ y $I_2 = \{4, 5\}$ como se ve en la segunda línea del grafo de abajo. Después se aplica el mismo procedimiento pero con los conjuntos I_1 e I_2. En nuestro ejemplo, debemos buscar el índice que minimice a $PL(\{i\}, \mathbf{X}_{1:3}^n)$, con $i \in I_1$ obteniendo $i = 2$. En este caso se abren otras dos ramas en el árbol y se sigue la división del problema como se muestra abajo. En cambio, si buscamos el índice que minimice a $PL(\{i\}, \mathbf{X}_{4:5}^n)$ con $i \in I_2$ se obtiene que $i = 5$, como este valor es el mas grande dentro del conjunto de índices de I_2, este nodo del árbol no se abre en dos ramas. Finalmente, el algoritmo termina cuando no pueden abrirse más ramas, ya que no se obtienen nuevos puntos de independencia para agregar a $\widehat{U}_n^{bin}$.



Algoritmo para estimador de las distribuciones

En este caso el código difiere del código exhaustivo en dos funciones:

La función **iteracion_alg_binario_normal** es una función recursiva cuyos pasos están ilustrados que recibe los mismos parámetros que la función *iteracion_exhaustivo_normal*, estos eran una matriz de datos $\mathbf{X}^n$ y valores c y ξ definidos en (2.3.3). La función calcula los valores de $PL(\{u\}, \mathbf{X}^n)$ para cada $u \in U$ utilizando la función *score_para_u_normal_bin*, guarda el índice u que minimiza a dicha función y luego divide la matriz de datos $\mathbf{X}^n$ en dos submatrices a las cuales le aplica esta misma función. Los pasos que se realizan están ilustrados en el ejemplo 3.2.4 .

Luego, la función que llama a este algoritmo está dada por

```
get_best_u_bin_alg_normal <- function(data, c=1, psi=0.25){
  d <- dim(data)[2]
  u_obtenido <- unlist(iteracion_alg_binario_normal(data, 1, d, psi=psi, c=c))
  u_obtenido
}
```

Algoritmo para estimador de covarianzas

Para este algoritmo basta con modificar las funciones que calculan el estimador e incluirla en la iteración del árbol binario. Para eso usamos las siguientes funciones.

La función **covarianza_bloques** recibe como parámetros una matriz de datos $\mathbf{X}^n$ y un único índice $\{i\}$ y devuelve la matriz de covarianza dada por Σ_i. Notemos que por la estructura del código no necesitamos construir matrices Σ_U donde U tiene más de un índice.

La función **iteracion_alg_binario_normal** es casi idéntica a la definida en el modelo basado en función de distribución, con la excepción de que en cada iteración llama a la función *distancia_matrices* en vez de *score_para_u_normal_bin*, ya que es necesario cambiar el estimador usado.

Finalmente, la función que llama a este último algoritmo está dada por

```
get_best_u_exacto_alg_normal_covarianza <- function(data, c=1, psi=0.25){
  d <- dim(data)[2]
  u_obtenido <- unlist(iteracion_exhaustivo_normal(data, 1, d, psi=psi, c=c))
  u_obtenido
}
```

Capítulo 4

Simulaciones y Resultados

En esta sección estudiaremos el comportamiento de los estimadores propuestos para diferentes conjuntos de datos generados bajo escenarios controlados. Evaluaremos a través de un estudio de simulación la capacidad que tienen los métodos propuestos para recuperar el conjunto correcto de independencia.

Para distinguir cada uno de los estimadores que compararemos nos referimos a los diferentes estimadores de la siguiente forma. Si elegimos el estimador basado en los estimadores de las distribuciones, en el caso en que se utilice la distribución empírica lo llamaremos *modelo empírico* y en el caso en el que se use el estimador paramétrico bajo el modelo Normal Multivariado lo llamaremos *modelo normal*. En cambio, si usamos el enfoque basado en estimadores de las matrices de covarianzas llamaremos a este *modelo de covarianzas*. Por otro lado, para cada uno de ellos se aplicará el método exhaustivo y el binario, aclarando que método se utiliza en cada modelo.

Para medir la diferencia entre las estimaciones y el conjunto correcto de índices se utilizará la distancia de Hausdorff que se define a continuación.

Definición 4.0.1. Dados dos conjuntos no vacíos $A, B \subset \{1, \cdots, d-1\}$ definimos la distancia de Hausdorff como

$$\rho_H(A, B) = max\{\rho(A||B), \rho(B||A)\}$$

donde $\rho(B||A) = \sup_{b \in B} \inf_{a \in A} |a - b|$ y $\rho(\emptyset||A) = d - 1$

En la siguiente sección mostraremos como se generan los datos y compararemos los resultados usando la distancia de Hausdorff.

4.1. Escenario de simulación

En este trabajo se va a considerar un vector aleatorio $\mathbf{X} = (X_1, \cdots, X_5) \in \mathbb{R}^5$ con distribución Normal Multivariada y se generará una muestra aleatoria $\{\mathbf{X}^i : 1 \leq i \leq n\}$ para distintos valores de n.

Vamos a construir esta muestra de forma que el conjunto de independencia sea $U^* = \{2, 3\}$, quedando así los bloques independientes (X_1, X_2), (X_3) y (X_4, X_5) del vector $\mathbf{X}$.

Para esto consideraremos una matriz de correlación que cumpla las condiciones de independencia pedidas, Σ_ρ dada por

$$\Sigma_\rho = \begin{pmatrix} 1 & \rho & 0 & 0 & 0 \\ \rho & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 3/\sqrt{10} \\ 0 & 0 & 0 & 3/\sqrt{10} & 1 \end{pmatrix} \tag{1}$$

donde el parámetro ρ representa la correlación entre X_1 y X_2, la cual podremos ir variando. Con esa estructura se ve que los bloques de independencia son (X_1, X_2), (X_3) y (X_4, X_5), siendo el conjunto de independencia $U^* = \{2, 3\}$.

Con Σ_ρ generaremos un conjunto de datos de la longitud n y analizaremos la eficiencia de los algoritmos. En particular, la propuesta *modelo empírico* para este tipo de matrices ya fue estudiado en [8], por lo que en este trabajo vamos a estudiar los otros dos modelos para contrastar con los resultados descriptos en ese trabajo.

Los parámetros propuestos para el factor de correlación ρ entre X_1 y X_2 serán los valores de $\rho = 1/\sqrt{10}$, $\rho = 2/\sqrt{10}$ y $\rho = 3/\sqrt{10}$. Por otro lado, para el cálculo de la penalización $\lambda_n = cn^{-\xi}$ definida en (2.3.3) necesitamos encontrar valores para los parámetros c y ξ. Para decidir que valores son mas convenientes para cada estimador, vamos a realizar simulaciones para diferentes valores y ver que combinación de parámetros son más eficientes teniendo en cuenta la distancia de Hausdorff. Los valores a estudiar serán $c = \{0{,}01,\ 0{,}1,\ 1,\ 2,\ 5,\ 10,\ 50\}$ y $\xi = \{0{,}05,\ 0{,}1,\ 0{,}15,\ 0{,}2,\ 0{,}25,\ 0{,}3,\ 0{,}35,\ 0{,}4,\ 0{,}45\}$.

Para cada modelo y cada valor fijo de ρ, c y ξ vamos a considerar $Nrep = 100$ replicaciones de datos generados y una muestra de tamaño $n = 100$ generadas a partir de la matriz de covarianza Σ_ρ. Calcularemos la distancia de Hausdorff para cada una de las muestras entre el conjunto verdadero y el conjunto estimado para cada conjunto de valores de parámetros, y luego consideraremos el promedio de estas distancias. Esto nos dará una medida sobre que tan lejos o cerca están nuestras propuestas de estimación de devolvernos el valor real de U. A continuación se representarán estos resultados en heatmaps para los diferentes modelos comparándolos con el gráfico basado en la distribución empírica que se analizó en [8].

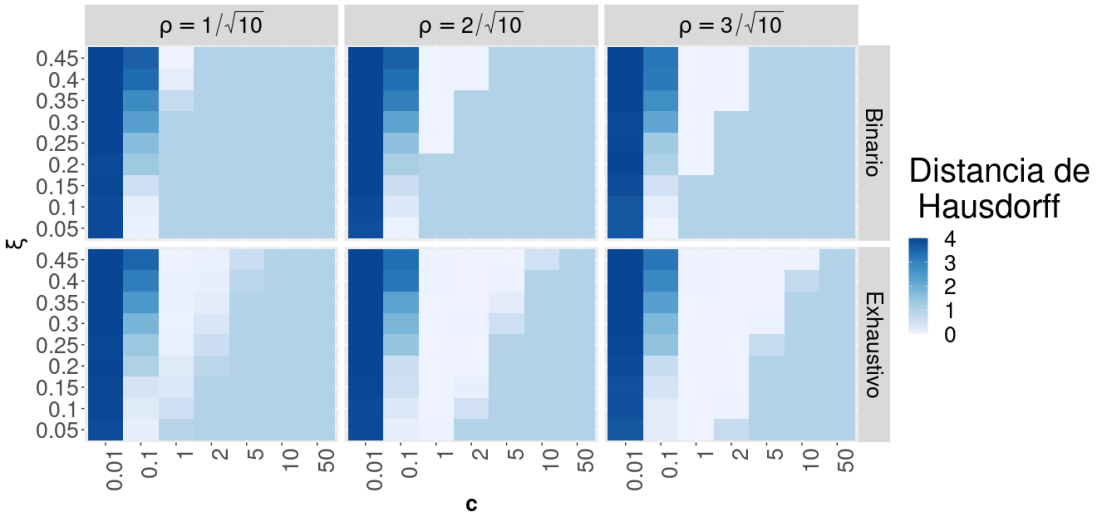


Figura 4.1: Heatmap para los valores c y ξ del modelo basado en estimadores para la distribución de la Normal Multivariada.

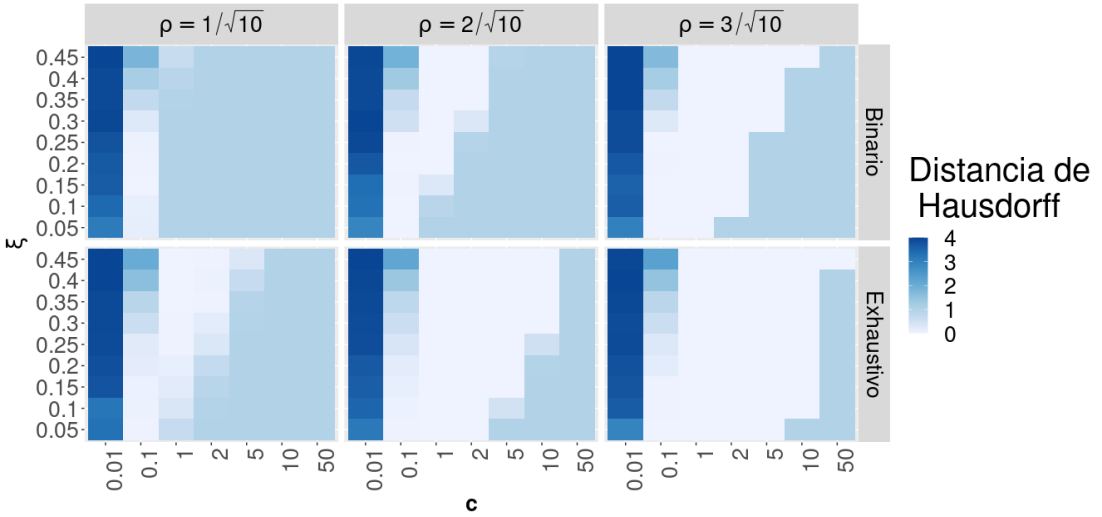


Figura 4.2: Heatmap para los valores c y ξ del modelo basado en matrices de covarianzas.

En las Figuras 4.1 y 4.2 se observa que tan eficiente son los algoritmos binarios y exhaustivos con diferentes combinaciones de c (representado en el eje x) y ξ (representado en el eje y). La escala de colores corresponden a la distancia media de Hausdorff entre el conjunto estimado y el conjunto verdadero de puntos de independencia. A una mayor intensidad de color, mayor es la distancia. Podemos observar que el modelo exhaustivo

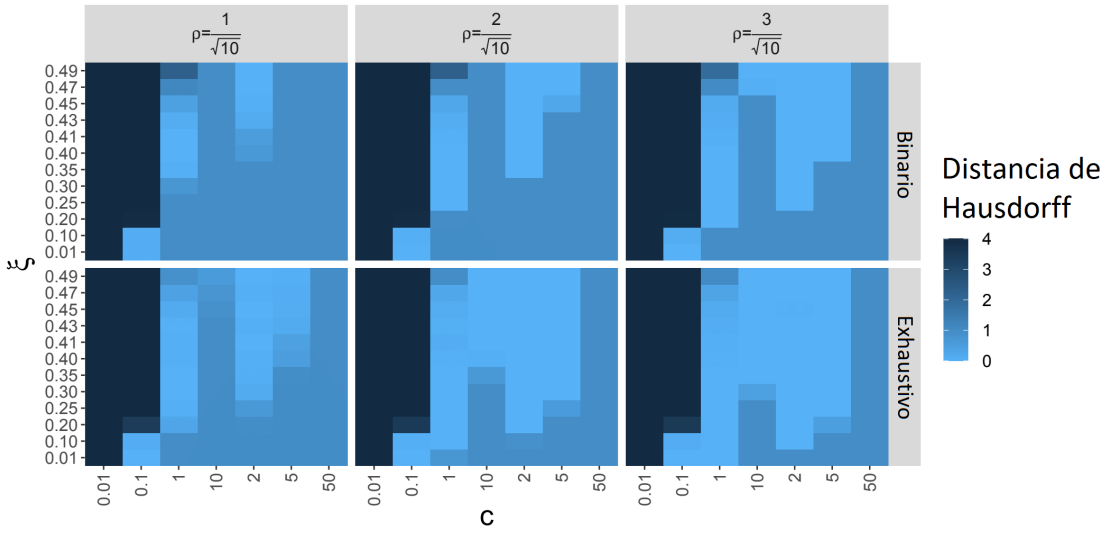


Figura 4.3: Heatmap para los valores c y ξ del modelo basado estimador de la distribución empírica tomado de [8].

es más eficiente que el modelo binario, en el sentido de que hay mas pares de valores (c, ξ) que verifican que la distancia de Hausdorff es más cercana a 0. Otra observación general es que cada modelo se vuelve más exacto a medida que el valor de ρ crece. Estos resultados coinciden cn los que se observan en la Figura 4.3 para el modelo empírico.

Para el modelo basado en estimadores de distribución, una buena elección de parámetros es $c = 1$ y $\xi = 0{,}4$. Notemos que estos valores coinciden con los elegidos en [8] a partir de la Figura 4.3.

En cambio, para el modelo basado en matrices de covarianzas, si bien para el caso exhaustivo sí es una buena elección $c = 1$ y $\xi = 0{,}4$, para el caso binario vamos a considerar $c = 0{,}1$ y $\xi = 0{,}05$.

Con estos valores específicos, donde sabemos que los métodos funcionan eficazmente, podemos estudiar la distancia media de Hausdorff para diferentes valores de n, donde n es la cantidad de datos de la matriz $\mathbf{X}^n$. Para cada $\rho = \{1/\sqrt{10}, 2/\sqrt{10}, 3/\sqrt{10}\}$ consideremos $Nrep = 100$ replicaciones de nuestra simulación y tomaremos los valores de $n = \{50,\ 100,\ 200,\ 300,\ 500,\ 1000,\ 2000\}$. Definimos el puntaje medio para cada simulación a partir del promedio de la distancia de Hausdorff respecto del número de repeticiones. Además, se incluyó el gráfico asociado a la distancia de Hausdorff del estimador basado en la distribución empírica ya realizado en [8] (Figura 4.6).

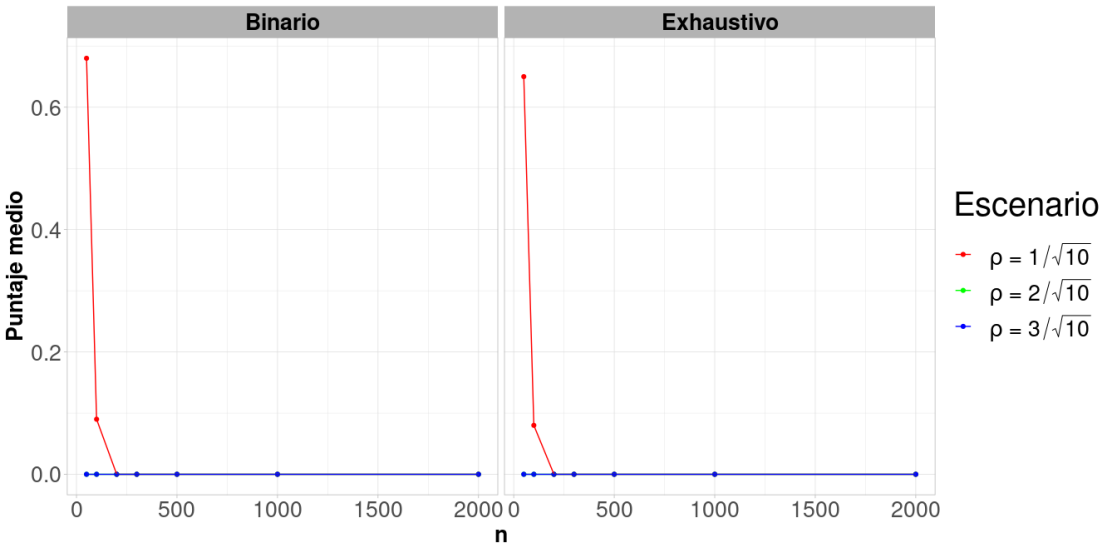


Figura 4.4: Puntaje medio en función de distitos valores de n para el modelo basado en estimadores para la distribución de la Normal Multivariada.

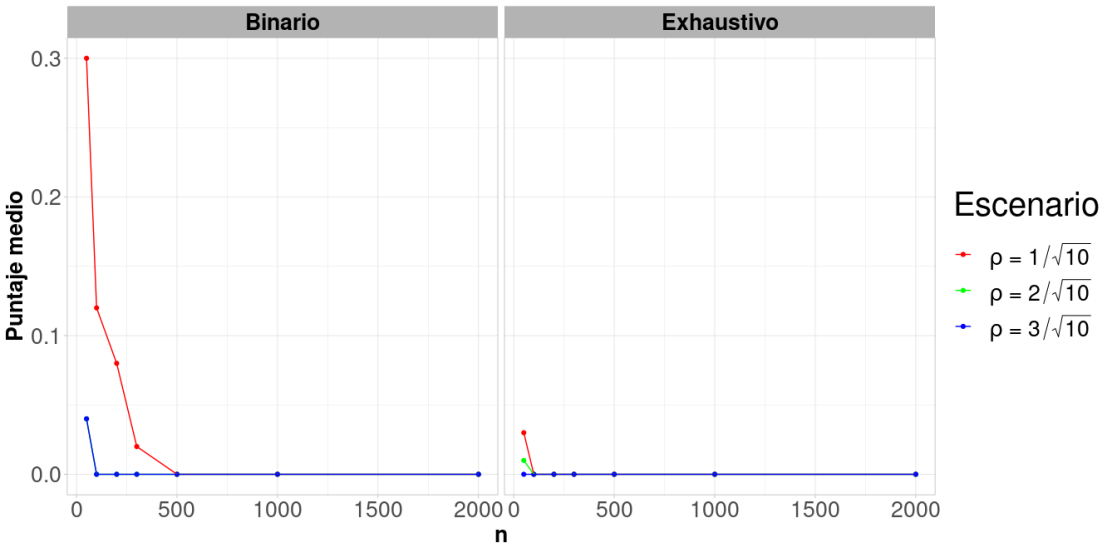


Figura 4.5: Puntaje medio en función de distitos valores de n para el modelo basado en matrices de covarianza.

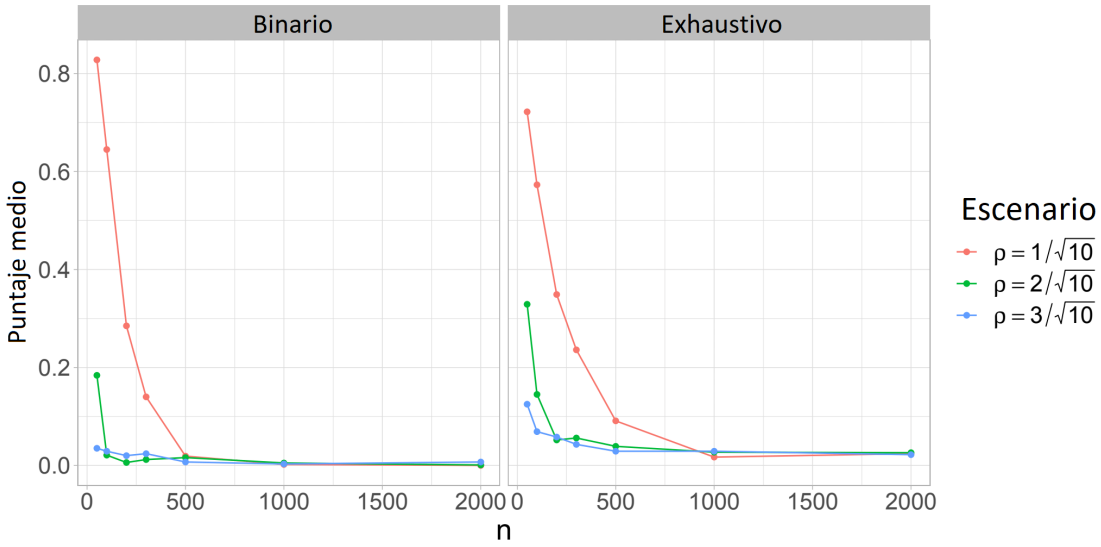


Figura 4.6: Puntaje medio en función de distitos valores de n para el modelo basado en estimadores de la distribución empírica tomado de [8].

En las Figuras 4.4 y 4.5 se puede ver que ambos modelos la distancia media de Hausdorff es 0 a partir de $n = 300$ a diferencia del modelo basado en estimador de distribución empírica dado en la Figura 4.6, donde comienza a valer 0 a partir de valores de n mayores a 500. Además, para todos los modelos, se observa que para $\rho = 1/\sqrt{10}$ y valores de n menores a 300 el puntaje medio es mayor que para los otros valores de ρ. Para poder analizar mejor la tendencia del puntaje medio realizaremos algunas simulaciones más para valores de n más pequeños. Consideraremos algunos valores de n a partir de 30.

En las Figuras 4.7 y 4.8 se ve que los modelos exhaustivos tienen mejores puntajes medios que los modelos binarios, para todos los valores de ρ. Se puede observar que para $\rho = 1/\sqrt{10}$ ambos modelos no son muy eficientes en comparación a los otros valores de ρ. Para valores mas grandes de ρ los modelos tienen un mejor rendimiento, no mostrando diferencias entre el modelo exhaustivo o binario.

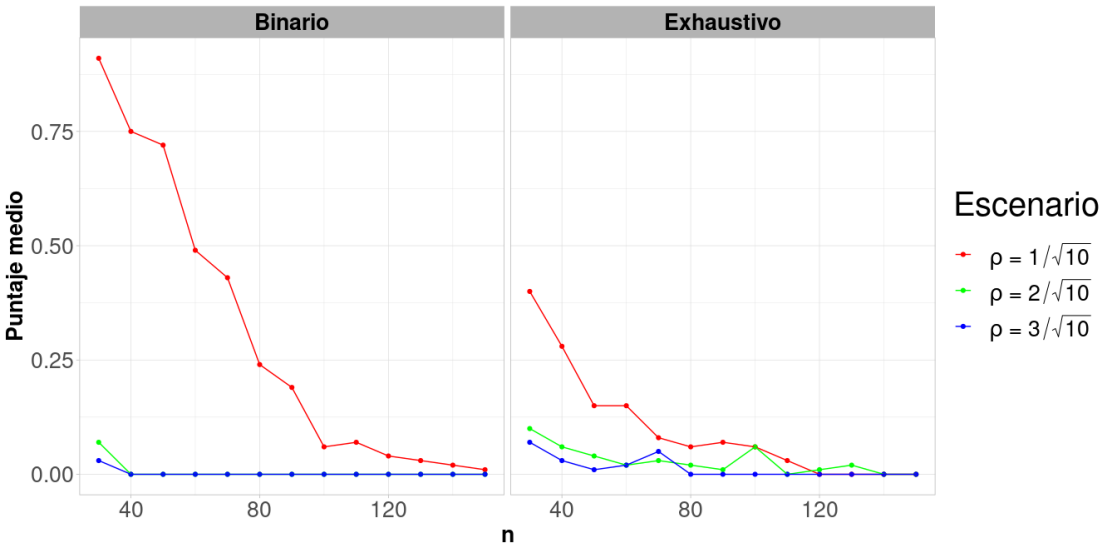


Figura 4.7: Puntaje medio en función de distitos valores de n mas pequeños para el modelo basado en estimadores para la distribución de la Normal Multivariada.

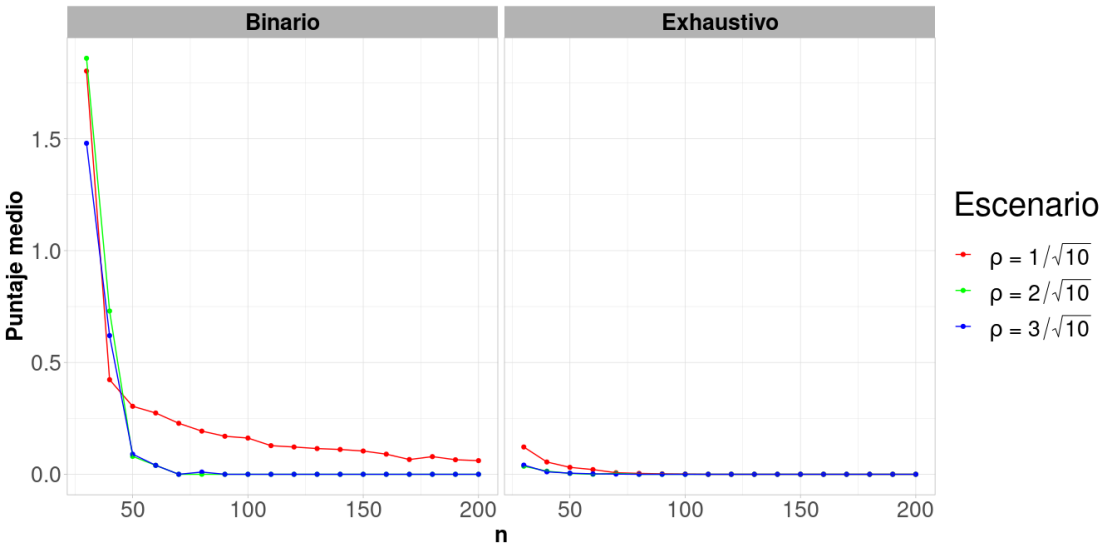


Figura 4.8: Puntaje medio en función de distitos valores de n mas pequeños para el modelo basado en matrices de covarianza.

4.1.1. Tiempo medio

Para cada modelo sobre las $Nrep = 100$ replicaciones y los valores de $n = \{50, 100, 200, 300, 500, 1000, 2000\}$ se estudió el tiempo medio en segundos que demora cada algoritmo

(Figuras 4.9 y 4.10). Además, se muestran (Figura 4.11) los gráficos del tiempo medio de cómputo para el algoritmo del modelo basado en la distribución empírica obtenidos en [8], obteniéndose así.

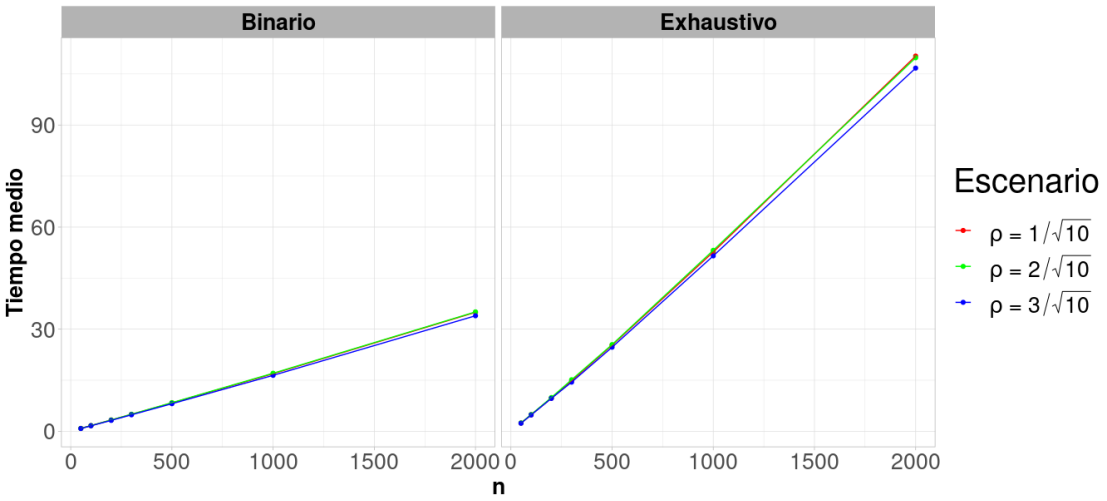


Figura 4.9: Tiempo medio en segundos en función de distitos valores de n para el modelo basado en estimadores para la distribución Normal Multivariada.

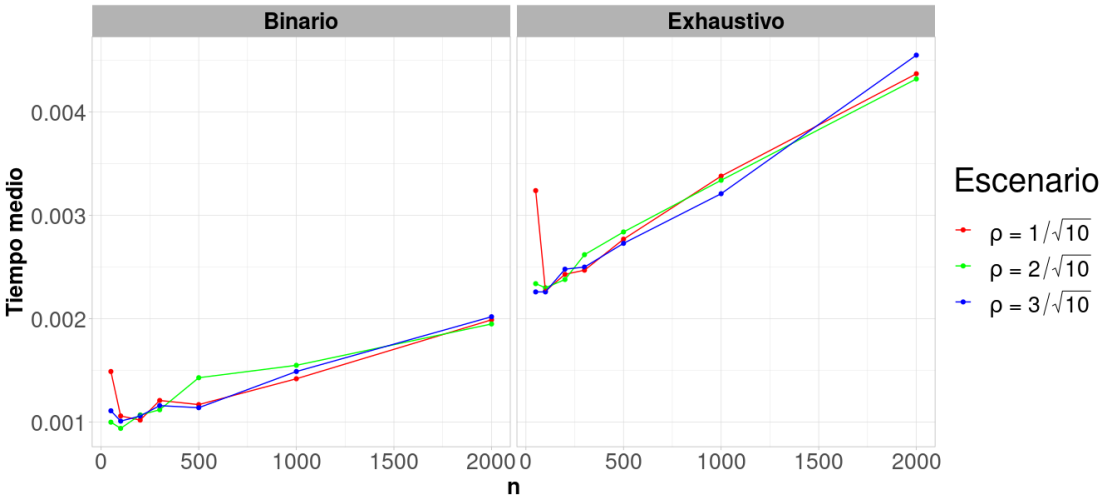


Figura 4.10: Tiempo medio en segundos en función de distitos valores de n para el modelo basado en matrices de covarianza.

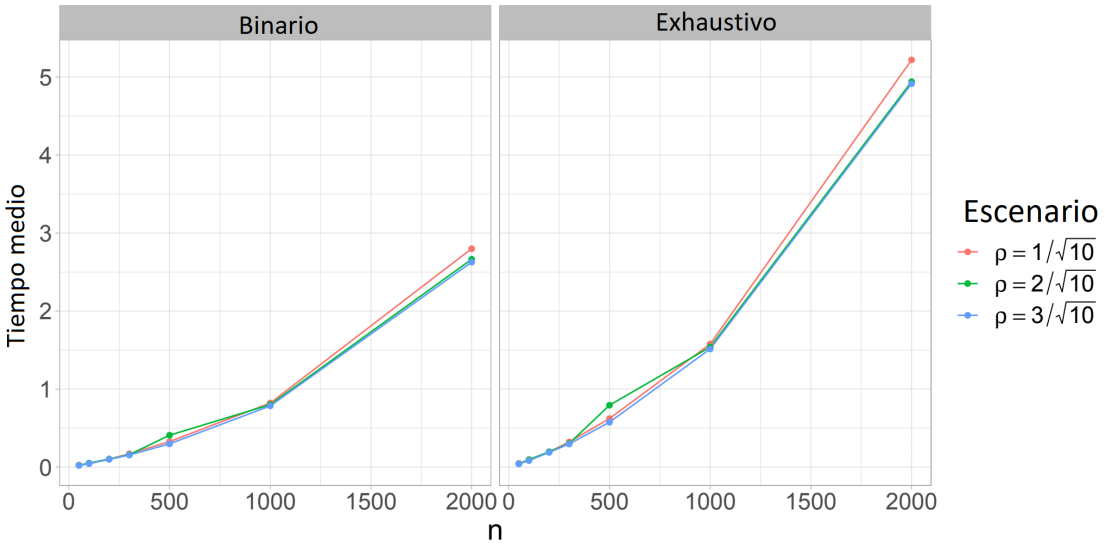


Figura 4.11: Tiempo medio en segundos en función de distitos valores de n para el modelo basado en estimadores de la distribución empírica tomado de [8].

Comparando los resultados de las Figuras 4.9 y 4.11 observamos que respecto al modelo basado en la distribución empírica, el modelo basado en la función de distribución Normal Multivariada es mucho más lento tanto para los casos binarios como exhaustivo. Esto es debido a que en la implementación del código fue necesario un número mayor de iteraciones en los algoritmos de este último caso. Por otro lado, considerando la Figura 4.10 vemos que el algoritmo del modelo basado en matrices de covarianzas es mucho más rápido que los demás. Esto es debido a que evita calcular el valor de la función de distribución acumulada de una Normal Multivariada o de la empírica, ya que solo se estudia la distancia de las matrices de covarianza. Las diferencias mencionadas en el código se pueden apreciar en el Apéndice.

En todos los casos la implementación del algoritmo binario resultó ser más rápida, con una diferencia mayor a medida que aumenta n. Esto se corresponde con la menor complejidad computacional del algoritmo de $D\&C$ respecto al algoritmo exhaustivo.

4.1.2. Tasa de error

Por último, para cada modelo, se realizó un estudio sobre su tasa de error. Esta se calculó contando la cantidad de veces que se obtuvo el valor correcto U^* y promediando sobre las $Nrep = 100$ replicaciones para cada uno de los tamaños muestrales $n = \{50, 100, 200, 300, 500, 1000, 2000\}$. La tasa de error es una medida resumen efectiva para mostrar la eficiencia de los algoritmos empleando estas simulaciones con resultados

conocidos.

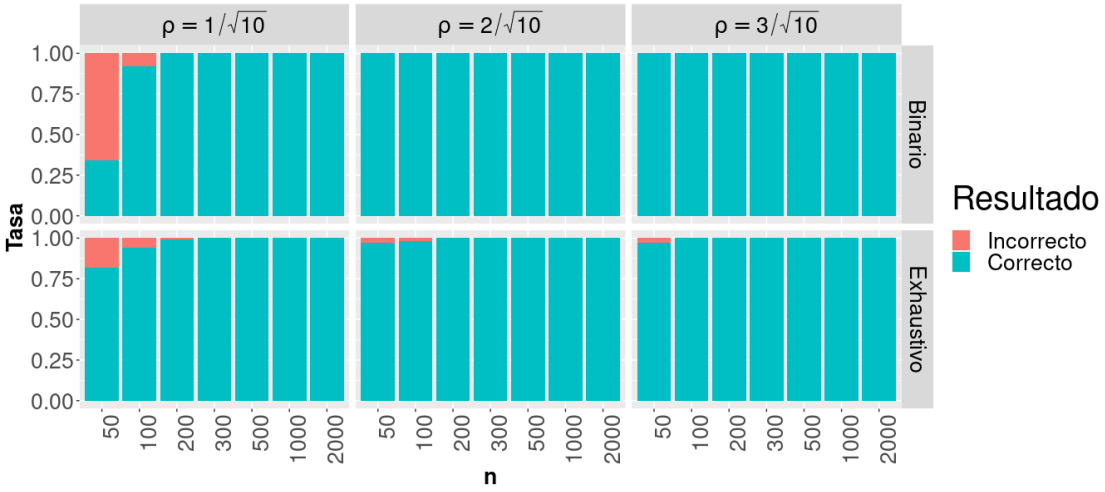


Figura 4.12: Tasa de error en función de distitos valores de n para para el modelo basado en estimadores para la distribución Normal Multivariada.

Figura 4.13: Tasa de error en función de distitos valores de n para para el modelo basado en matrices de covarianzas.

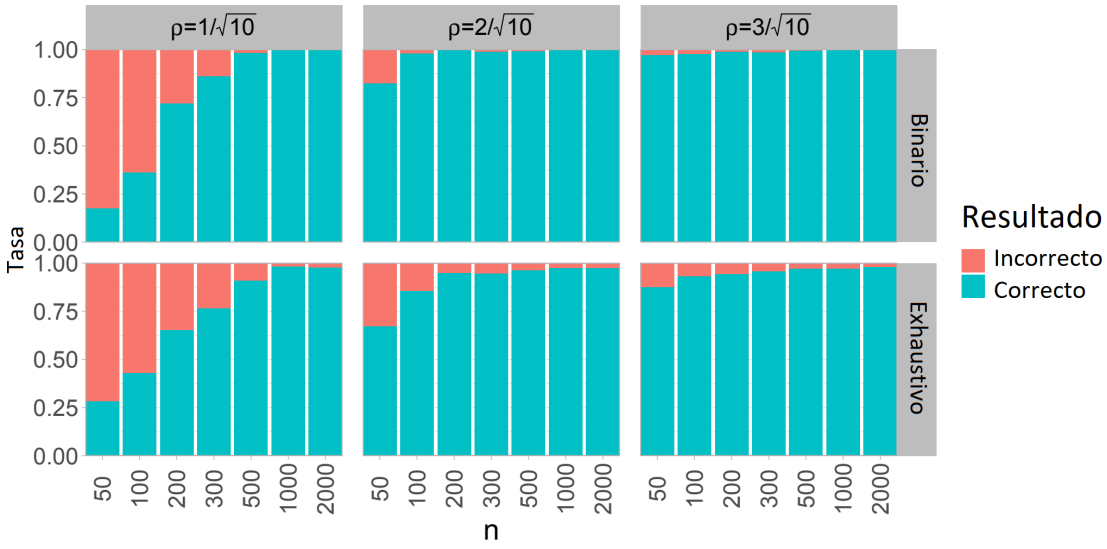


Figura 4.14: Tasa de error en función de distitos valores de n para para el modelo basado en la distribución empírica tomado de [8].

Los gráficos de las Figuras 4.12, 4.13 y 4.14 (esta última tomada de [8]) muestran las tasas de error de los modelos basados en distribuciones Normal Multivariada, matrices de covarianza y distribución empírica respectivamente. El modelo basado en distribución empírica muestra para ambos algoritmos una tasa de error mayor para todo valor de n y ρ. Este se debe a que si bien el modelo basado en la distribución empírica es más general, los datos simulados están mejor descriptos una distribución Normal Multivariada. Entonces la tasa de error como medida resumen muestra la efectividad de los estimadores propuestos.

Los gráficos de las Figuras 4.12 y 4.13 muestran que para los valores de $\rho = 2/\sqrt{10}$ y $\rho = 3/\sqrt{10}$ y ambos modelos las tasas de error son casi nulas. En cambio para $\rho = 1/\sqrt{10}$ para valores pequeños de n la tasa aumenta, siendo mayor para el caso binario que para el exhaustivo. Para estudiar esto con mayor detalle se decidió hacer este análisis con valores de n más pequeños a partir de 30 para los casos de modelos basados en la distribución Normal Multivariada y matrices de covarianza.

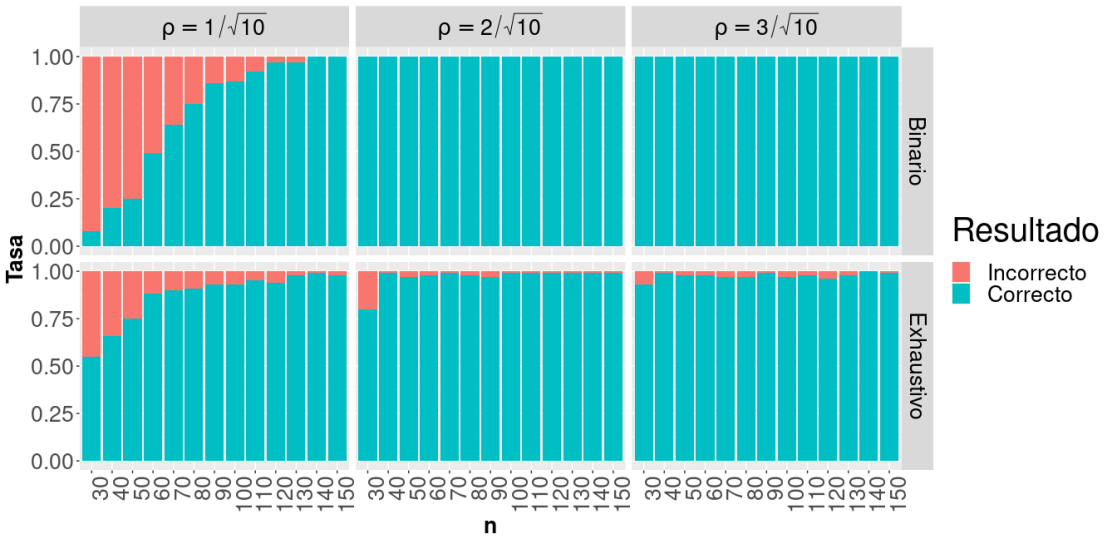


Figura 4.15: Tasa de error en función de distitos valores de n pequeños para para el modelo basado en estimadores para la distribución Normal Multivariada.

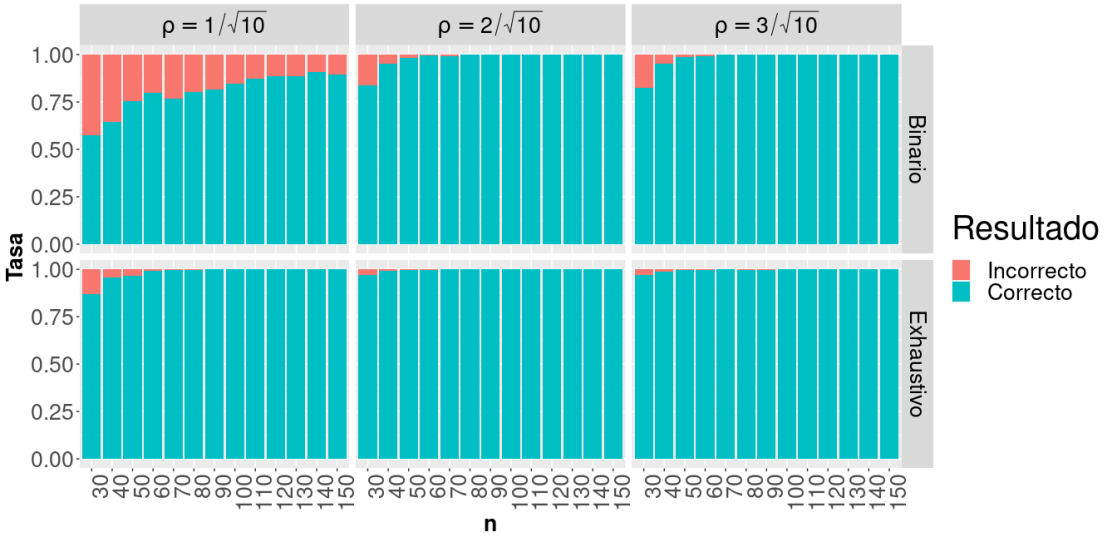


Figura 4.16: Tasa de error en función de distitos valores de n pequeños para para el modelo basado en matrices de covarianzas

Notamos en las Figuras 4.15 y 4.16 que para los valores estudiados de n y $\rho = \frac{2}{\sqrt{10}}$ y $\rho = \frac{3}{\sqrt{10}}$ los modelos siguen teniendo un muy buen rendimiento, salvo $n = 30$ y $n = 40$.

En el caso de $\rho = \frac{1}{\sqrt{10}}$ los modelos tienen dierente comportamiento dependiendo del

algoritmo usado. En el caso del modelo basado en la distribución Normal Multivariada el algortimo binario presenta una mayor tasa de error hasta valores de n cercanos a 100. Luego a medida que el valor de n aumenta las tasas de error se equiparan haciéndose prácticamente 0.

En cambio, en el caso del modelo basado en matrices de covarianza la tasa de error para el algoritmo binario es siempre mucho mayor al del algoritmo exhaustivo independientemente del valor de n. El algoritmo exhaustivo, en este caso, presenta una tasa de error que rápidamente se acerca a 0.

Capítulo 5

Caso real: Río San Francisco

En el capítulo 1 se describió el problema de estudiar el caudal volumétrico del Río San Francisco en Brasil.

Formalmente, tenemos el siguiente problema: se tiene un río en el cual en diferentes lugares fijos hay estaciones de medición, que registran el caudal del mismo. Si se tienen d de estaciones de medición a lo largo de un río, podemos definir para cada una de ellas la variable aleatoria X_u, la cual describe el promedio mensual de flujo que se registra en la estación u, con $u = 1, ..., d$. Luego consideraremos el vector aleatorio $\mathbf{X} = (X_1, ..., X_d)$ que contiene a las d variables.

Si suponemos que tenemos para cada estación n observaciones, cada una de ellas correspondiente al flujo promedio de un mes específico, entonces podemos llamar $\mathbf{X}^{(i)} = (X_1^i, ..., X_d^i)$ al vector con la observación del i-ésimo mes y tendremos el proceso $\{\mathbf{X}^{(i)} : 1 \leq i \leq n\}$ donde $X^{(i)} \in \mathbb{R}^d$.

En nuestro problema específico se tienen un total de $d = 10$ estaciones de medición que están ubicados a lo largo del río, como se muestran en la Figura 1.1 (b) del capítulo 1. Las estaciones están enumeradas de 1 a 10 según el ordena lo largo del curso del río. Como ya se dijo, cada una mide el flujo de agua y vamos a considerar para cada medidor el flujo medio del río en un mes, es decir, que por cada mes se tiene un vector $(x_1, \cdots, x_{10})$ donde x_i es el flujo promedio del río obtenido del medidor i. Por otro lado, los datos a usar son los registrados entre enero de 1977 y enero de 2016, siendo un total de $n = 358$ observaciones. Por lo tanto, la matriz de datos a utilizar $\mathbf{X}^n$ tiene $n = 358$ filas, donde cada fila está asociada al dato de un mes específico, y 10 columnas, donde cada columna i está asociada al flujo medio de agua del medidor i. Por lo tanto, si pensamos que entre cada medidor hay una sección de río, nuestro objetivo será determinar el conjunto de independencia entre las diferentes secciones.

Hidrológicamente el curso del río se puede dividir en cuatro tramos: la parte alta (donde se encuentran las estaciones 1 y 2), desde su nacimiento hasta la ciudad de Pira-

pora cerca de la cual se halla la represa Três Marias; la parte superior media (estaciones 3, 4, 5, 6 y 7), desde Pirapora hasta la presa de Sobradinho, la primera parte navegable; la parte media inferior de la presa de Sobradinho a la presa de Itaparica (estación 8); y la parte baja, de la represa de Itaparica hasta la desembocadura del río (estaciones 9 y 10). Con esta primera observación, si notamos X_i a la variable que describe el flujo medio de la estación de medición i con $i \in \{1, \cdots, 10\}$ y el vector, $\mathbf{X} = (X_1, \cdots, X_{10})$ una primera idea es que los bloques de independencia deberían ser (X_1, X_2), $(X_3, X_4, X_5, X_6, X_7)$, (X_8) y (X_9, X_{10}) siendo así $U = \{2, 7, 8\}$. Por otro lado, el caudal del río también puede ser afectado por el período del año, ya que hay un período de lluvias, la estación húmeda, que comienza en noviembre hasta enero y una estación seca que va de junio a agosto.

En [8] se estudió este mismo conjunto de datos utilizando el estimador basado en la distribución empírica para el modelo binario, resultando así $\widehat{U} = \widehat{U}^n_{bin} = \{7\}$. En este trabajo usaremos los algoritmos propuestos, tanto el binario como el exhaustivo, con los modelos basados en la distribución Normal Multivariada y en las matrices de covarianza, por un lado aplicados a toda la matriz de datos y, por otro lado, separaremos en estaciones húmeda (noviembre a enero) y seca (junio a agosto).

Una observación importante es que en nuestro modelo se asume que los datos siguen una distribución normal, pero al realizar gráficos boxplots de los datos en cada estación de medición notamos lo siguiente:

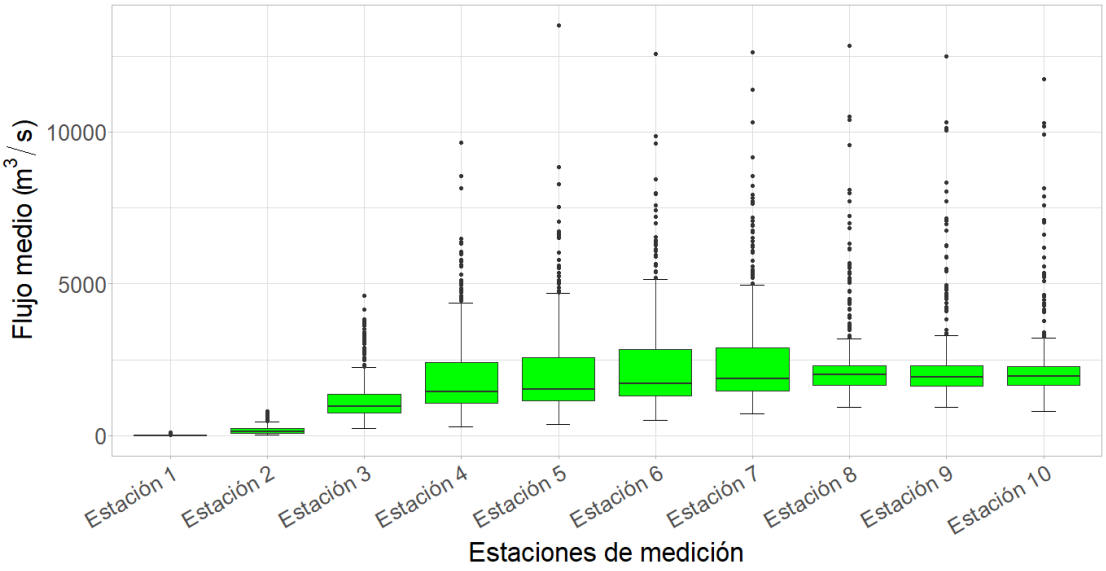


Figura 5.1: Boxplots de los datos correspondientes a cada una de las 10 estaciones de medición.

Observamos en la Figura 5.1 que cada columna de datos sigue una distribución aproximada a una lognormal. El flujo de un río con un régimen hidrológico de estación

seca - estación húmeda suele ajustar empíricamente a este tipo de distribuciones [2], por lo que aplicaremos una transformación a los datos de cada estación con el objetivo de lograr una distribución que se aproximen a una distribución normal.

En la Figura 5.2 se pueden ver los bloxplots de cada estación luego de aplicarse tal transformación.

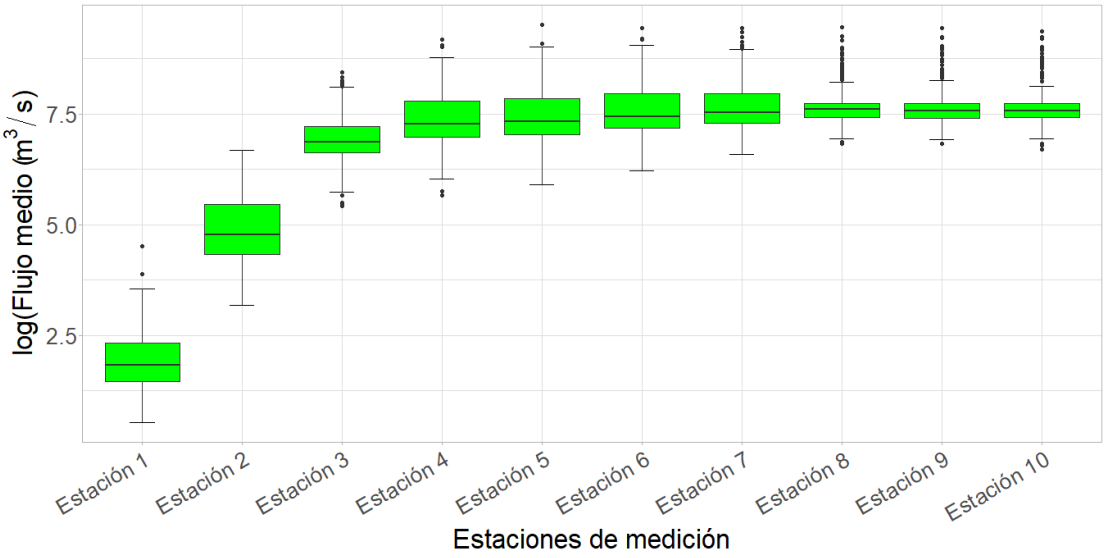


Figura 5.2: Boxplots de los logaritmos de los datos correspondientes a cada una de las 10 estaciones de medición.

Luego, mediante el análisis de los gráficos qqplots de los datos de cada estación (Figura 5.3), observamos que en general las columnas de la matriz se aproximan bien a una distribución normal, por lo que aplicaremos nuestros algoritmos.

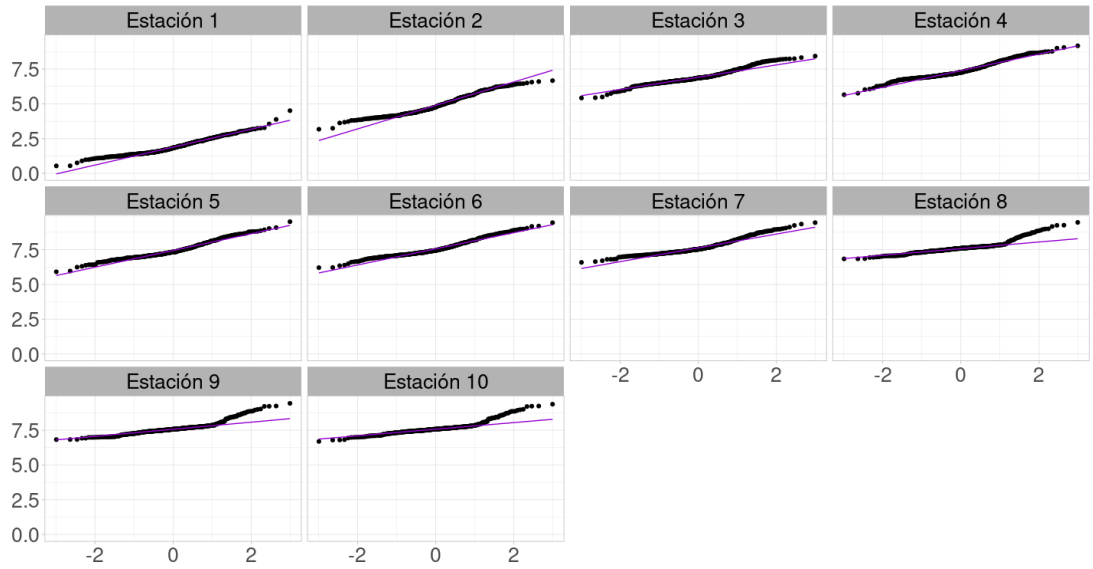


Figura 5.3: QQ-Plots de los datos correspondientes a cada una de las 10 estaciones de medición.

Debido a que no es posible determinar con exactitud los valores exactos de c y ξ decidimos implementar el código con algunos valores basados en los resultados de los heatmaps hechos en el Capítulo 4. Debido a que $c = 1$ mostraba una buena eficiencia en la mayoría de las simulaciones realizadas en la Sección 4.1 optamos por ese valor. Estudiamos diferentes valores de $\xi = \{0{,}05,\ 0{,}1,\ 0{,}15,\ 0{,}2,\ 0{,}25,\ 0{,}3,\ 0{,}35,\ 0{,}4,\ 0{,}45\}$ y determinamos los resultados en función de ese parámetro. Cada una de estas implementaciones se realizó, primero, sobre toda la matriz de datos y luego, para los datos restringidos a estaciones secas y húmedas. Al ir variando el valor de ξ no todos los resultados que se obtengan van a ser considerados para determinar $\widehat{U}_n$, sino que se optará por elegir el o los primeros índices para los cuales el vector $\mathbf{X}$ se comience a separar en subvectores. Por último, para los modelos basados en estimadores de distribución solo se utilizó el algoritmo de árbol binario, ya que el algoritmo exhaustivo tiene una complejidad temporal alta y esto hace que no sea buena su implementación.

5.1. Resultados para el modelo basado en estimador de distribución

ξ	N°de Estación de medición									
	1	2	3	4	5	6	7	8	9	10
0.05	$\{X_1\}$	$\{X_2\}$	$\{X_3\}$	$\{X_4\}$	$\{X_5\}$	$\{X_6\}$	$\{X_7\}$	$\{X_8\}$	$\{X_9\}$	$\{X_{10}\}$
0.1	$\{X_1\}$	$\{X_2\}$	$\{X_3\}$	$\{X_4\}$	$\{X_5\}$	$\{X_6\}$	$\{X_7\}$	$\{X_8\}$	$\{X_9\}$	$\{X_{10}\}$
0.15	$\{X_1\}$	$\{X_2\}$	$\{X_3\}$	$\{X_4\}$	$\{X_5\}$	$\{X_6\}$	$\{X_7\}$	$\{X_8\}$	$\{X_9\}$	$\{X_{10}\}$
0.2	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7\}$	$\{X_8,$	$X_9,$	$X_{10}\}$
0.25	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.3	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.35	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.4	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.45	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$

Tabla 5.1: Análisis con todos los datos al usar el estimador basado en distribución Normal Multivariada con el algortimo binario.

ξ	N° de Estación de medición									
	1	2	3	4	5	6	7	8	9	10
0.05	$\{X_1\}$	$\{X_2\}$	$\{X_3\}$	$\{X_4\}$	$\{X_5\}$	$\{X_6\}$	$\{X_7\}$	$\{X_8\}$	$\{X_9\}$	$\{X_{10}\}$
0.1	$\{X_1\}$	$\{X_2\}$	$\{X_3\}$	$\{X_4\}$	$\{X_5\}$	$\{X_6\}$	$\{X_7\}$	$\{X_8\}$	$\{X_9\}$	$\{X_{10}\}$
0.15	$\{X_1\}$	$\{X_2\}$	$\{X_3\}$	$\{X_4\}$	$\{X_5\}$	$\{X_6\}$	$\{X_7\}$	$\{X_8\}$	$\{X_9\}$	$\{X_{10}\}$
0.2	$\{X_1\}$	$\{X_2\}$	$\{X_3\}$	$\{X_4,$	$X_5,$	$X_6\}$	$\{X_7\}$	$\{X_8,$	$X_9,$	$X_{10}\}$
0.25	$\{X_1\}$	$\{X_2\}$	$\{X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7\}$	$\{X_8,$	$X_9,$	$X_{10}\}$
0.3	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7\}$	$\{X_8,$	$X_9,$	$X_{10}\}$
0.35	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7\}$	$\{X_8,$	$X_9,$	$X_{10}\}$
0.4	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.45	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$

Tabla 5.2: Análisis con los datos asociados a la estación húmeda al usar el estimador basado en distribución Normal Multivariada con el algortimo binario.

ξ	N°de Estación de medición									
	1	2	3	4	5	6	7	8	9	10
0.05	$\{X_1\}$	$\{X_2\}$	$\{X_3\}$	$\{X_4\}$	$\{X_5\}$	$\{X_6\}$	$\{X_7\}$	$\{X_8\}$	$\{X_9\}$	$\{X_{10}\}$
0.1	$\{X_1\}$	$\{X_2\}$	$\{X_3\}$	$\{X_4\}$	$\{X_5\}$	$\{X_6\}$	$\{X_7\}$	$\{X_8\}$	$\{X_9\}$	$\{X_{10}\}$
0.15	$\{X_1\}$	$\{X_2\}$	$\{X_3\}$	$\{X_4\}$	$\{X_5\}$	$\{X_6\}$	$\{X_7\}$	$\{X_8\}$	$\{X_9\}$	$\{X_{10}\}$
0.2	$\{X_1\}$	$\{X_2\}$	$\{X_3\}$	$\{X_4,$	$X_5,$	$X_6,$	$X_7\}$	$\{X_8\}$	$\{X_9\}$	$\{X_{10}\}$
0.25	$\{X_1\}$	$\{X_2\}$	$\{X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7\}$	$\{X_8,$	$X_9,$	$X_{10}\}$
0.3	$\{X_1\}$	$\{X_2\}$	$\{X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.35	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.4	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.45	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$

Tabla 5.3: Análisis con los datos asociados a la estación seca al usar el estimador basado en distribución Normal Multivariada con el algortimo binario.

A partir de los resultados de las Tablas 5.1, 5.2 y 5.3, a medida que ξ disminuye su valor, el modelo nos indica que el conjunto de independencia se puede estimar por $\widehat{U}_n^{bin} = \{7\}$ cuando se analizan todos los datos y los asociados a la estación húmeda. En cambio, para la estación seca el modelo indica que el conjunto de independencia se puede estimar por $\widehat{U}_n^{bin} = \{1, 2\}$.

5.2. Resultados para el modelo basado en estimador de covarianzas exhaustivo

ξ	N°de Estación de medición									
	1	2	3	4	5	6	7	8	9	10
0.05	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7\}$	$\{X_8,$	$X_9,$	$X_{10}\}$
0.1	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7\}$	$\{X_8,$	$X_9,$	$X_{10}\}$
0.15	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7\}$	$\{X_8,$	$X_9,$	$X_{10}\}$
0.2	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7\}$	$\{X_8,$	$X_9,$	$X_{10}\}$
0.25	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.3	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.35	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.4	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.45	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$

Tabla 5.4: Análisis con todos los datos al usar el estimador basado en matrices de covarianza con el algoritmo exhaustivo.

ξ	N°de Estación de medición									
	1	2	3	4	5	6	7	8	9	10
0.05	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7\}$	$\{X_8,$	$X_9,$	$X_{10}\}$
0.1	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7\}$	$\{X_8,$	$X_9,$	$X_{10}\}$
0.15	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7\}$	$\{X_8,$	$X_9,$	$X_{10}\}$
0.2	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7\}$	$\{X_8,$	$X_9,$	$X_{10}\}$
0.25	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7\}$	$\{X_8,$	$X_9,$	$X_{10}\}$
0.3	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.35	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.4	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.45	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$

Tabla 5.5: Análisis con los datos asociados a la estación húmeda basado en matrices de covarianza con el algoritmo exhaustivo.

ξ	N°de Estación de medición									
	1	2	3	4	5	6	7	8	9	10
0.05	$\{X_1\}$	$\{X_2\}$	$\{X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.1	$\{X_1\}$	$\{X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.15	$\{X_1\}$	$\{X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.2	$\{X_1\}$	$\{X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.25	$\{X_1\}$	$\{X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.3	$\{X_1\}$	$\{X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.35	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.4	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.45	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$

Tabla 5.6: Análisis con los datos asociados a la estación húmeda basado en matrices de covarianza con el algoritmo exhaustivo.

A partir de los datos de las Tablas 5.4, 5.5 y 5.6, a medida que ξ disminuye su valor, el modelo nos indica que el conjunto de independencia se puede estimar por $\widehat{U}_n = \{7\}$ cuando se analizan todos los datos y los asociados a la estación húmeda. En cambio, para la estación seca, el modelo estima el punto de independencia por $\widehat{U}_n = \{1\}$.

5.3. Resultados para el modelo basado en estimador de covarianzas binario

ξ	N°de Estación de medición									
	1	2	3	4	5	6	7	8	9	10
0.05	$\{X_1\}$	$\{X_2\}$	$\{X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7\}$	$\{X_8,$	$X_9,$	$X_{10}\}$
0.1	$\{X_1\}$	$\{X_2\}$	$\{X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7\}$	$\{X_8,$	$X_9,$	$X_{10}\}$
0.15	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7\}$	$\{X_8,$	$X_9,$	$X_{10}\}$
0.2	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7\}$	$\{X_8,$	$X_9,$	$X_{10}\}$
0.25	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.3	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.35	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.4	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.45	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$

Tabla 5.7: Análisis con todos los datos al usar el estimador basado en matrices de covarianza con el algoritmo binario.

ξ	N°de Estación de medición									
	1	2	3	4	5	6	7	8	9	10
0.05	$\{X_1\}$	$\{X_2\}$	$\{X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7\}$	$\{X_8,$	$X_9,$	$X_{10}\}$
0.1	$\{X_1\}$	$\{X_2\}$	$\{X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7\}$	$\{X_8,$	$X_9,$	$X_{10}\}$
0.15	$\{X_1\}$	$\{X_2\}$	$\{X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7\}$	$\{X_8,$	$X_9,$	$X_{10}\}$
0.2	$\{X_1\}$	$\{X_2\}$	$\{X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7\}$	$\{X_8,$	$X_9,$	$X_{10}\}$
0.25	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7\}$	$\{X_8,$	$X_9,$	$X_{10}\}$
0.3	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.35	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.4	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.45	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$

Tabla 5.8: Análisis con los datos asociados a la estación húmeda basado en matrices de covarianza con el algoritmo binario.

ξ	N°de Estación de medición									
	1	2	3	4	5	6	7	8	9	10
0.05	$\{X_1\}$	$\{X_2\}$	$\{X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7\}$	$\{X_8\}$	$\{X_9\}$	$\{X_{10}\}$
0.1	$\{X_1\}$	$\{X_2\}$	$\{X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7\}$	$\{X_8,$	$X_9,$	$X_{10}\}$
0.15	$\{X_1\}$	$\{X_2\}$	$\{X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7\}$	$\{X_8,$	$X_9,$	$X_{10}\}$
0.2	$\{X_1\}$	$\{X_2\}$	$\{X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.25	$\{X_1\}$	$\{X_2$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.3	$\{X_1\}$	$\{X_2$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.35	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.4	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$
0.45	$\{X_1,$	$X_2,$	$X_3,$	$X_4,$	$X_5,$	$X_6,$	$X_7,$	$X_8,$	$X_9,$	$X_{10}\}$

Tabla 5.9: Análisis con los datos asociados a la estación seca basado en matrices de covarianza con el algoritmo binario.

A partir de los resultados de las Tablas 5.7, 5.8 y 5.9, a medida que ξ disminuye su valor, el modelo nos indica que el conjunto de independencia se puede estimar por $\widehat{U}_n^{bin} = \{7\}$ cuando se analizan todos los datos. Cuando se analizan los datos asociados a la estación húmeda el conjunto de independencia se puede estimar por $\widehat{U}_n^{bin} = \{7\}$. Por último, al analizar los datos asociados a la estación seca el modelo indica que el conjunto de independencia se puede estimar por $\widehat{U}_n^{bin} = \{7\}$ o $\widehat{U}_n^{bin} = \{1, 2, 7\}$.

Notemos, al observar los resultados de todas las tablas, que en las estaciones secas hay una tendencia a que el conjunto de independencia sea $\widehat{U}_n = \{1\}$ o $\widehat{U}_n = \{1, 2\}$ a diferencia de los otros casos en los que nos devuelve $\widehat{U}_n = \{7\}$.

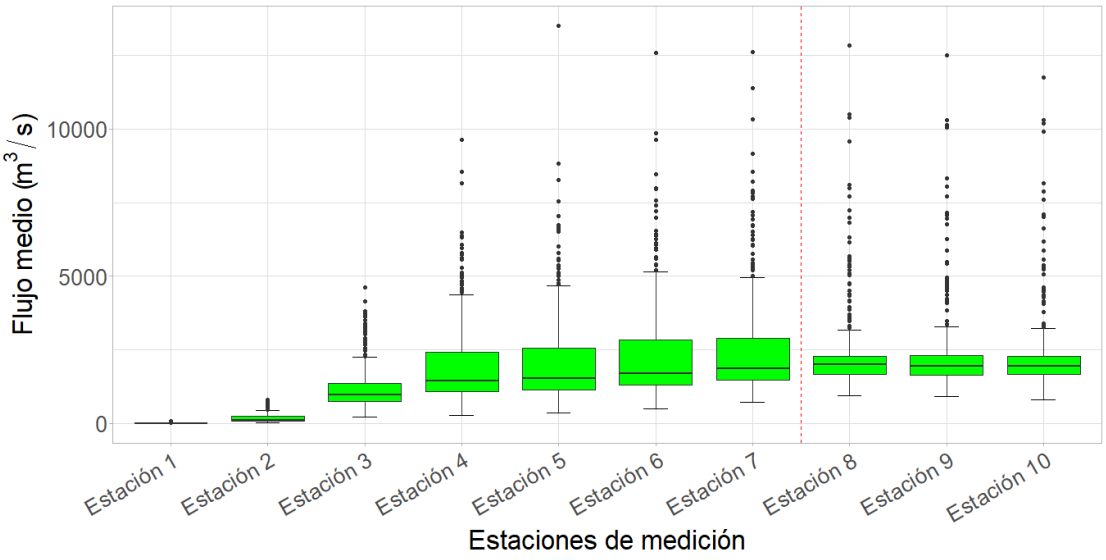


Figura 5.4: Boxplots de los datos correspondientes a cada una de las 10 estaciones de medición. La línea roja punteada representa la partición de los datos en subvectores independientes.

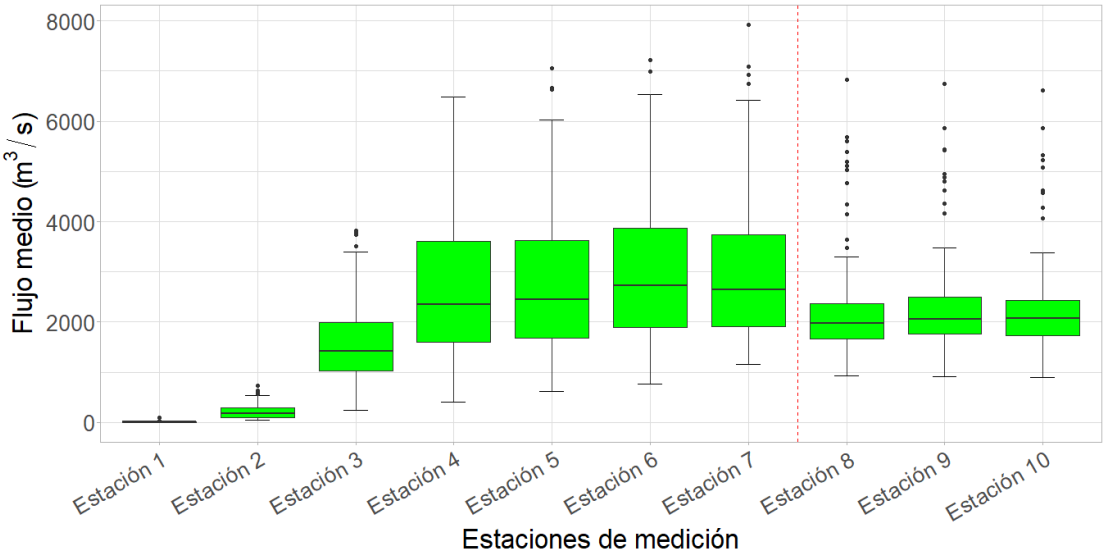


Figura 5.5: Boxplots de los datos de la estación húmeda correspondientes a cada una de las 10 estaciones de medición. La línea roja punteada representa la partición de los datos en subvectores independientes.

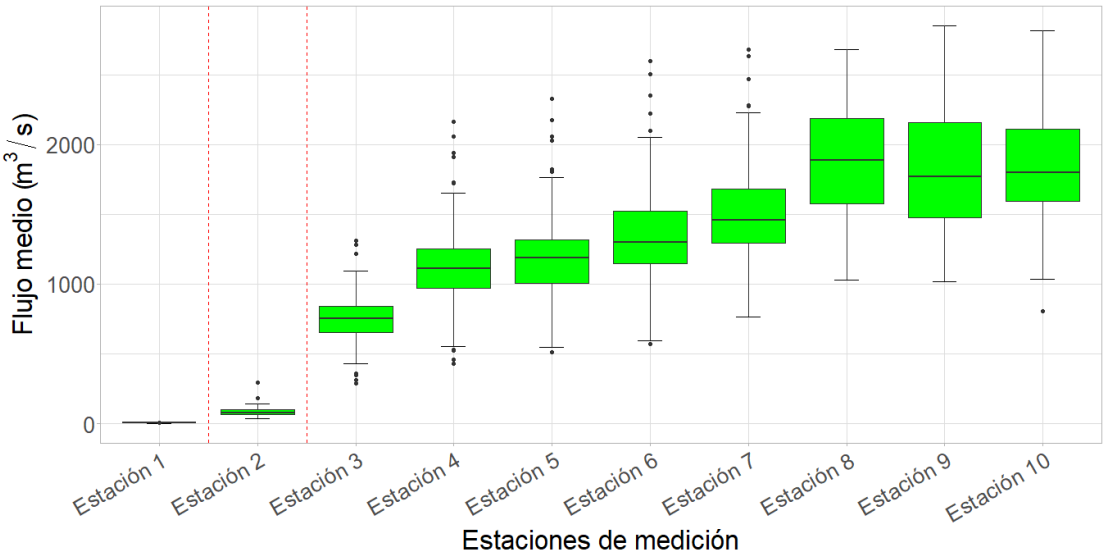


Figura 5.6: Boxplots de los datos de la estación seca correspondientes a cada una de las 10 estaciones de medición. La línea roja punteada representa la partición de los datos en subvectores independientes.

En las Figuras 5.4, 5.5, 5.6 se ven los boxplots asociados al flujo medio del río en las 10 estaciones de medición, correspondientes a todos los datos, los asociados a la estación húmeda y a la seca. Se muestran las posibles separaciones de los datos según los conjuntos de independencia obtenidos.

Una interpretación posible de estos resultados corresponde a la clasificación del curso del río que hicimos previamente. Notemos que es esperable que la estación de medición 7 sea un punto de independencia porque se halla en la represa de Sobradinho que es uno de los embalses más grandes del mundo y regula el flujo en el tramo inferior del río.

En la estación seca los puntos de independencia en 1 y 2 podrían deberse a la mayor variabilidad del flujo del río cerca de su nacimiento y al menor caudal del río por recibir menor cantidad de agua al estar en un período de pocas lluvias.

La diferencia en los conjutos de independencia en la estación seca y húmeda podría indicar un cambio en el comportamiento del flujo del río y del impacto ambiental de las represas construídas sobre su curso.

Conclusiones del trabajo de tesis

En este trabajo se propusieron dos métodos para hallar estimadores de puntos de independencia de datos multivariados, bajo el supuesto de que las observaciones siguen una distribución Normal Multivariada. Las dos propuestas introducidas fueron implementadas en R y se estudió su comportamiento desde una perspectiva computacional, analizando su desempeño bajo escenarios simulados. También se compararon los estimadores propuestos con un método ya estudiado en [8] observando similitudes y diferencias. Los resultados mostraron ventajas considerables tanto desde el punto de vista de la ganancia en términos de precisión de su estimación como así también a las mejoras obtenidas en la velocidad de cómputo.

Se mostró la aplicabilidad de estos métodos en un ejemplo concreto, el estudio del caudal de un río, notando que los conjuntos de independencia obtenidos para los diferentes estimadores y propuestas computacionales están relacionados con el comportamiento del mismo. Creemos que estos métodos de estimación para conjuntos de independencia se pueden implementar en cualquier otro río para los que se tengan una cantidad de datos suficientes, como así también en el estudio de tránsito en calles o autopistas, o en número de piezas en un proceso de producción, entre otros.

Quedó como futuro trabajo un estudio de la fundamentación teórica de la propuesta. Así mismo, según lo observado en el ejemplo estudiado será también pertinente la propuesta de un método de selección automática de c y ξ.

Apéndice

Código del algoritmo exhaustivo basado en distribuciones

```
multinormal_bloques<-function(columna, punto, k){
  d <- ncol(columna)

  if(!is.numeric(punto)){
    punto <- as.numeric(punto)
  }
  if (length(k)==1){
    if (d==k){
      mu <- colMeans(columna)
      cov <- cov(columna)

      acum <- pmvnorm(lower=-Inf, upper=punto, mean=mu, sigma = cov)[1]
    }else{
      mu <- colMeans(columna)
      cov <- cov(columna)

      cov[(k+1):d,1:k]<-matrix(0, d-k, k)
      cov[1:k,(k+1):d]<-matrix(0, k, d-k)

      acum <- pmvnorm(lower=-Inf, upper=punto, mean=mu, sigma = cov)[1]
    }
  }
  else{
    mu <- colMeans(columna)
    cov <- cov(columna)
    long <- length(k)
    for(i in 1:long){
      ind1 <- k[i]
```

```
      if(i==long){
        ind2 <- d
      }
      else{
        ind2 <- k[i+1]

      }
      cov[(ind1+1):ind2,1:ind1]<-matrix(0, ind2-ind1, ind1)
      cov[1:ind1,(ind1+1):ind2]<-matrix(0, ind1, ind2-ind1)

      acum <- pmvnorm(lower=-Inf, upper=punto, mean=mu, sigma = cov)[1]

    }
  }
  return(acum)
}


score_para_u_normal <- function(u, datos,  sobre.datos = TRUE, repeticiones=1000) {
  if (sobre.datos) {
    repeticiones <- dim(datos)[1]
    }
  diferencias<-rep(NA, repeticiones)
  d <- dim(datos)[2]
  for( i in 1:repeticiones)
  {
    if (sobre.datos) {
      v <- datos[i,]
    } else {
      v <-rnorm(d)
    }
    initial <- 1
    final <- u[length(u)]
    if (is.null(final) || is.na(final)){
      final <- length(v)
    }else{
      final <- u
      }
    a <- acumulada_normal(datos, v , final)
    b <- acumulada_normal(datos,v,d)
    diferencias[i]<-abs(a-b)
```

```
  }
  max(diferencias)
}

iteracion_exhaustivo_normal <- function(datos, psi, c, sobre.datos=TRUE, repeticiones=1000){
  idx <- 1
  d <- dim(datos)[2]
  score_u <- rep(NA, (d-1)**2) #16)
  n <- dim(datos)[1]
  u_list <- list()
  l <- combn(1:(d-1), 0, simplify = FALSE)
  u <- c()
  u_list[idx] <- list(u)
  score_u[idx] <- score_para_u_normal(u, datos, sobre.datos, repeticiones)
                                      + c * n^(-psi) / (length(u) + 1)
  idx <- idx + 1
  for (s in 1:(d-1)){
    l <- combn(1:(d-1), s, simplify = FALSE)
    for (k in 1:length(l)) {
      u <- unlist(l[k])
      u_list[idx] <- list(u)
      score_u[idx] <- score_para_u_normal(u, datos, sobre.datos, repeticiones)
                                       + c * n^(-psi) / (length(u) + 1)
      idx <- idx + 1
    }
  }
  minimo <- which.min(score_u)
  u_list[minimo]
}

get_best_u_exacto_alg_normal <- function(data, c=1, psi=0.25){
  d <- dim(data)[2]
  u_obtenido <- unlist(iteracion_exhaustivo_normal(data, 1, d, psi=psi, c=c))
  u_obtenido
}

acumulada_normal<-function(columna, punto,k){
  if (!is.null(dim(columna))){
    salida <- multinormal_bloques(columna,punto,k)
  }else{
    m <- mean(columna)
```

```
    s <- sd(columna)
    salida <- pnorm(punto, mean = m, sd = s, lower.tail = TRUE, log.p = FALSE)
  }
  return(salida)
}
```

Código del algoritmo exhaustivo basado en covarianzas

```
covarianza_bloques<-function(columna, k){
  d <- ncol(columna)
  cov <- cov(columna)
  if(!is.null(k) && length(k)==1){
    cov[(k+1):d,1:k]<-matrix(0, d-k, k)
    cov[1:k,(k+1):d]<-matrix(0, k, d-k)
  }
  else{
    if (!is.null(k) && length(k)>1){
      long <- length(k)
      for(i in 1:long){
        ind1 <- k[i]
        if(i==long){
          ind2 <- d
        }
        else{
          ind2 <- k[i+1]
        }
        cov[(ind1+1):ind2,1:ind1]<-matrix(0, ind2-ind1, ind1)
        cov[1:ind1,(ind1+1):ind2]<-matrix(0, ind1, ind2-ind1)
      }
    }
  }
  return(cov)
}

distancia_matrices <- function(u, datos, sobre.datos = TRUE, repeticiones=1000){
  n <- ncol(datos)
  Sigma <- cov(datos)
  Sigma_u <- covarianza_bloques(datos,u)
  d1 <- det(Sigma)
  d2 <- det(Sigma_u)
```

```
  d3 <- det((Sigma+Sigma_u)/2)
  if(d3!=0){
    hellinger <- 1-((sqrt(sqrt(d1*d2)))/sqrt(d3))
  }else{
    hellinger <- 0
  }
  return(hellinger)
}

iteracion_exhaustivo_normal <- function(datos, psi, c, sobre.datos=TRUE, repeticiones=1000){
  idx <- 1
  d <- dim(datos)[2]
  score_u <- rep(NA, (d-1)**2)
  n <- dim(datos)[1]
  u_list <- list()
  l <- combn(1:(d-1), 0, simplify = FALSE)
  u <- c()
  u_list[idx] <- list(u)
  score_u[idx] <- distancia_matrices(u, datos, sobre.datos, repeticiones)
                                     + c * n^(-psi) / (length(u) + 1)
  idx <- idx + 1
  for (s in 1:(d-1)){
    l <- combn(1:(d-1), s, simplify = FALSE)
    for (k in 1:length(l)) {
      u <- unlist(l[k])
      u_list[idx] <- list(u)
      score_u[idx] <- distancia_matrices(u, datos, sobre.datos, repeticiones)
                                      + c * n^(-psi) / (length(u) + 1)
      idx <- idx + 1
    }
  }
  if ((length(score_u[2:d])>=3 && length(unique(score_u[2:d]))==1) |
                        (length(score_u[2:d]))==2 && score_u[2]==score_u[3]){
    res <- u_list[1]
  }else{
    minimo <- which.min(score_u)
    res <- u_list[minimo]
  }
  return(res)
}
```

```
get_best_u_exacto_alg_normal_covarianza <- function(data, c=1, psi=0.25){
  d <- dim(data)[2]
  u_obtenido <- unlist(iteracion_exhaustivo_normal(data, 1, d, psi=psi, c=c))
  u_obtenido
}
```

Código del algoritmo binario para el estimador basado en distribuciones

```
score_para_u_normal_bin <- function(u, datos, sobre.datos = TRUE, repeticiones=1000) {
  if (sobre.datos) {
    repeticiones <- dim(datos)[1]
  }
  diferencias<-rep(NA, repeticiones)
  d <- dim(datos)[2]
  for( i in 1:repeticiones)
  {
    if (sobre.datos) {
      v <- datos[i,]
    } else {
      v <-rnorm(d)
    }
    initial <- 1
    final <- u[1]
    if (is.null(final) || is.na(final)) { final <- length(v)}
    a <- acumulada_normal_bin(datos, v , final)
    b<-acumulada_normal_bin(datos,v,d)
    diferencias[i]<-abs(a-b)
  }
  max(diferencias)
}

iteracion_alg_binario_normal <- function(datos, inicial, final, psi, c, off_set=0,
                                  sobre.datos = TRUE, repeticiones=1000) {
  if (is.null(dim(datos))) {
    return(c())
  } else {
    n <- dim(datos)[1]
    d <- dim(datos)[2]
    score_u <- rep(NA, d)
```

```
    for (s in 1:d){
      if (s==d) { u <- c() } else { u <- c(s) }
      score_u[s] <- score_para_u_normal_bin(u, datos, sobre.datos=sobre.datos,
                    repeticiones=repeticiones) + c * n^(-psi) / (length(u) + 1)
    }

    minimo <- which.min(score_u)
    if (minimo==d) {
      return(c())
    } else {
      if (inicial<minimo) {
        izq <- iteracion_alg_binario_normal(datos[,inicial:minimo], inicial, minimo, psi,
                      c, off_set = 0, sobre.datos=sobre.datos, repeticiones=repeticiones)
      } else { izq <- c() }
      if(minimo+1<final) {
        der <- iteracion_alg_binario_normal(datos[,(minimo+1):final], 1, final-minimo, psi,
                      c, off_set = minimo, sobre.datos=sobre.datos, repeticiones=repeticiones)
      } else {der <-c()}

      return(c(izq, minimo , der) + off_set)
    }
  }
}
```

Código del algoritmo para el estimador basado en covarianzas

```
    covarianza_bloques<-function(columna, k){
      d <- ncol(columna)
      cov <- cov(columna)
      if(!is.null(k)){
        cov[(k+1):d,1:k]<-matrix(0, d-k, k)
        cov[1:k,(k+1):d]<-matrix(0, k, d-k)
      }
      return(cov)
    }

iteracion_alg_binario_normal <- function(datos, inicial, final, psi,
                    c, off_set=0, sobre.datos = TRUE, repeticiones=1000) {
```

```
  if (is.null(dim(datos))) {
    return(c())

  } else {
    n <- dim(datos)[1]
    d <- dim(datos)[2]
    score_u <- rep(NA, d)

    for (s in 1:d){
      if (s==d) { u <- c() } else { u <- c(s) }
      score_u[s] <- distancia_matrices(u, datos, sobre.datos=sobre.datos,
                     repeticiones=repeticiones) + c * n^(-psi)/(length(u) + 1)
    }

    minimo <- which.min(score_u)
    if (minimo==d | (length(score_u)>=3 && length(unique(score_u[-length(score_u)]))==1
                | (length(score_u)==2 && score_u[1]==score_u[2]) )) {
      return(c())
    } else {
      if (inicial<minimo) {
        izq <- iteracion_alg_binario_normal(datos[,inicial:minimo],
                    inicial, minimo, psi, c, off_set = 0,
                    sobre.datos=sobre.datos, repeticiones=repeticiones)
      } else { izq <- c() }
      if(minimo+1<final) {
        der <- iteracion_alg_binario_normal(datos[,(minimo+1):final], 1,
                    final-minimo, psi, c, off_set = minimo,
                    sobre.datos=sobre.datos, repeticiones=repeticiones)
      } else {der <-c()}

      return(c(izq, minimo , der) + off_set)
    }
  }
}
```

Bibliografía

[1] Boente, G. y Yohai, V.J. *Notas de estadística.* Fecha de acceso: 2023-05-23. URL: https://mate.dm.uba.ar/~vyohai/.

[2] Bowers, M. C., Tung, W. W. y Gao, J. B. On the distributions of seasonal river flows: Lognormal or power law? *Water Resources Research* (2012). 48: W05536. DOI: 10.1029/2011WR011308.

[3] Castro, B. M., Lemes, R. B., Cesar, J., Hünemeier, T. y Leonardi, F. A model selection approach for multiple sequence segmentation and dimensionality reduction. *Journal of Multivariate Analysis* (2018). 167: 319-330. DOI: 10.1016/j.jmva.2018.05.006.

[4] Everitt, B. y Hothor, T. *An Introduction to Applied Multivariate Analysis with R.* Springer, Nueva York, EEUU, 2011.

[5] Gretton, A. y Gyorfi, L. Consistent nonparametric tests of independence. *Journal of Machine Learning Research* (2010). 11(Apr): 1391-1423.

[6] Gretton, A., Herbrich, R., Smola, A., Bousquet, O. y Schölkopf, B. Kernel methods for measuring independence. *Journal of Machine Learning Research* (2005). 6(Dic): 2075-2129.

[7] Johnson, R. A. y Wichern., D. W. *Applied Multivariate Statistical Analysis, 6th edition.* Pearson New International Edition, Upper Saddle River, EEUU, 2014.

[8] Leonardi, F., Lopez-Rosenfeld, M., Rodriguez, D., Severino, M. T. F. y Sued, M. Independent block identification in multivariate time series. *Journal of Time Series Analysis* (2021). 42 (1): 19-33. DOI: 10.1111/jtsa.12553.

[9] Pardo, L. *Statistical Inference Based on Divergence Measures.* Taylor & Francis Group, Nueva York, EEUU, 2006.

[10] Sistema Nacional de Informações sobre Recursos Hídricos. Fecha de acceso: 2019-09-03. URL: http://www.snirh.gov.br/hidroweb/.

[11] Steerneman, T. On the total variation and Hellinger distance between signed measures: an application to product measures. *Proceedings of the American Mathematical Society* (1983). 88(4): 684-688. DOI: 10.1201/9781420034813.

[12] Szekely, G. y Rizzo, M. Brownian distance covariance. *The Annals of Applied Statistics* (2009). 3(4): 1236-1265. DOI: 10.1214/09-AOAS3.

[13] Szekely, G., Rizzo, M. y Bakirov, N. Measuring and testing dependence by correlation of distances. *The Annals of Statistics* (2007). 35(6): 2769-2794. DOI: 10.1214/009053607000000505.

[14] Yohai, V. J. *Notas de probabilidades y estadística.* Fecha de acceso: 2023-05-23. URL: https://mate.dm.uba.ar/~vyohai/.