

En la figura 3.7 se puede apreciar la estructura de los diseños óptimos bajo los criterios 2.12 (criterio sin la matriz de información a priori R) y 2.13 (criterio con la matriz de información a priori R) asociados al modelo de dos compartimientos. Los diseños bajo el criterio 2.12, presentan un comportamiento similar bajo los diferentes niveles de correlación en cada uno de los tamaños de muestra. Al parecer, el nivel de correlación entre las constantes de absorción y eliminación (k_{10} , k_{12} y k_{21}) del modelo de dos compartimientos (modelo 1.3) no afectan la estructura general de los diseños bajo el criterio 2.12. Para los diseños obtenidos bajo el criterio 2.13 se puede apreciar que para tamaños de muestra entre dos y cuatro tiempos, la estructura de los diseños es similar bajo los diferentes niveles de correlación. Algo diferente sucede al considerar tamaños de muestra entre cinco y siete tiempos, pues cuando se consideran niveles de correlación mayores al 50 % los diseños tienden a concentrarse al comienzo de la región de diseño, entre 0 y 0.79 horas. Para el criterio 2.13 bajo el modelo de dos compartimientos, la correlación tiene efecto sobre la estructura de los diseños cuando se consideran tamaños de muestra mayores o iguales cinco tiempos. En la figura 3.7 se puede apreciar de nuevo el comportamiento particular del criterio 2.13, el cual tiende a definir la mayor cantidad de tiempos de muestreo al comienzo de la región de diseño.

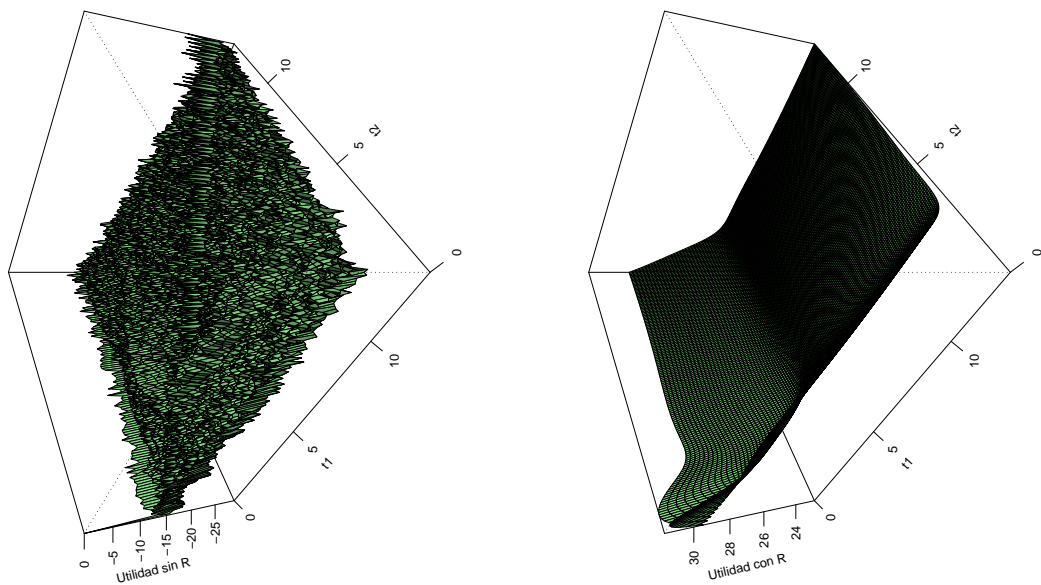


Figura 3.8: Superficies de los criterios 2.12 y 2.13 asociadas al modelo de dos compartimientos para el nivel de correlación de 0 % entre k_{10} , k_{12} y k_{21} .

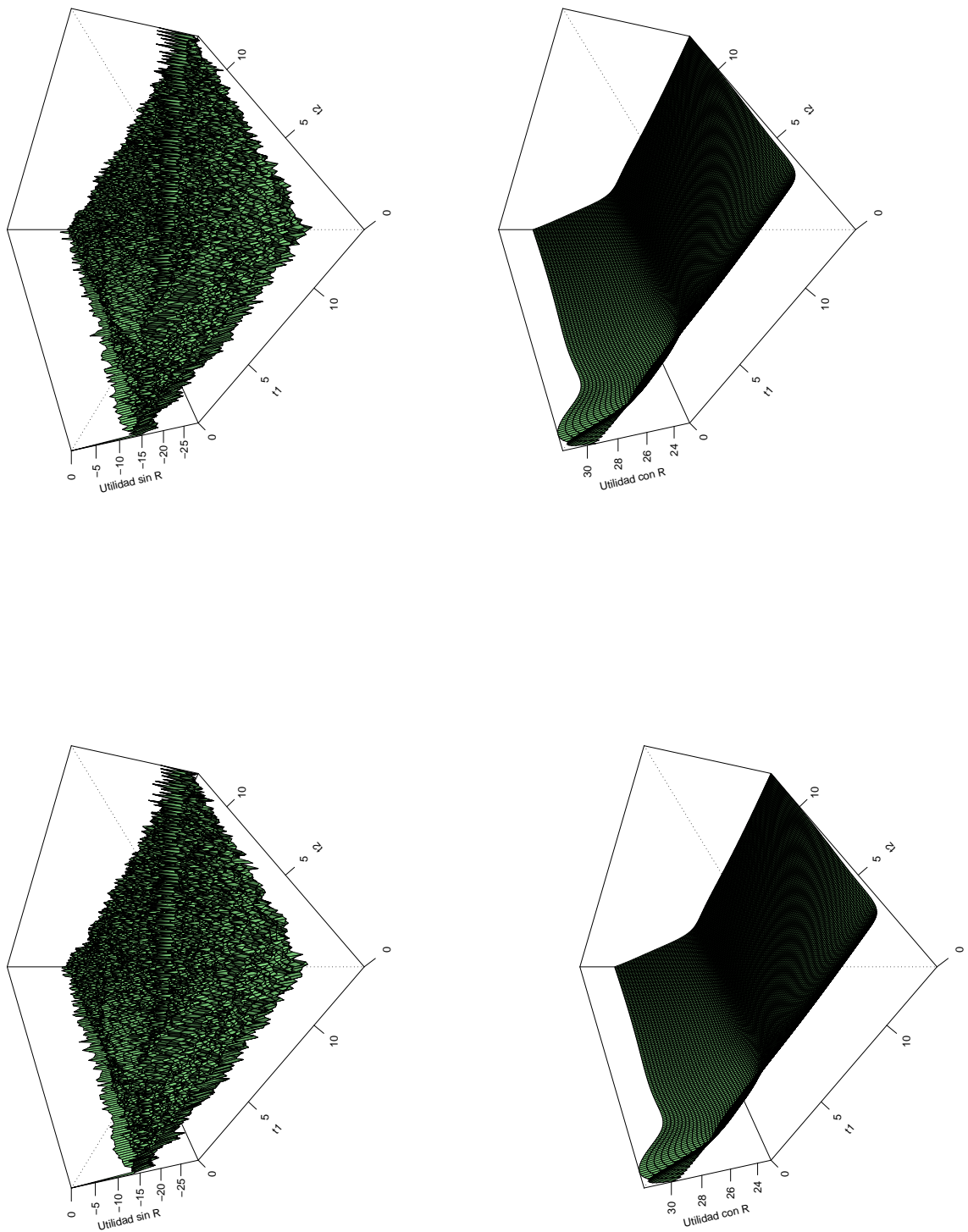


Figura 3.9: Superficies de los criterios 2.12 y 2.13 asociadas al modelo de dos compartimientos para los niveles de correlación al 30 % y 50 % entre k_{10} , k_{12} y k_{21} .

En las figuras 3.8 y 3.9 se puede apreciar en detalle el comportamiento o la forma de los criterios para los diseños de dos puntos. Para el modelo de dos compartimientos también se puede ver el mismo comportamiento del criterio D-óptimo Bayesiano en relación a la forma de la superficie, la cual es totalmente caótica en el caso que no se considera la matriz de precisión R (gráficos en la columna izquierda de las figuras 3.8 y 3.9) y totalmente suave en el caso cuando se considera la matriz de precisión (criterio 2.13). También se puede apreciar que para los tres diferentes niveles de correlación la forma del criterio es básicamente la misma, lo cual es coherente con lo observado en la estructura de los diseños óptimos de dos tiempos apreciados en la figura 3.7.

3.5. Error Cuadrático Medio como medida de evaluación para los diseños óptimos

Ahora para tener otra medida de desempeño asociada a los diseños óptimos obtenidos para los modelos de uno y dos compartimientos, diferente a la que se presenta en las figuras 3.2 y 3.6 (utilidad esperada), se calculará la distribución muestral del $ECM(\hat{\theta})$ vía simulación.

Para calcular la distribución muestral del ECM asociada a los parámetros estimados de los modelos de interés, se implementó el algoritmo ilustrado en la figura 3.10. Para estimar los parámetros de ambos modelos con el fin de obtener la distribución del ECM, se recurre de nuevo a la metodología de la **Evolución Diferencial**, la cual es descrita en el **Apéndice**.

Los resultados de dicho algoritmo asociados a los parámetros del modelo de un compartimiento en los diferentes escenarios se ilustran en las figuras 3.11, 3.12, 3.13, 3.14, 3.15 y 3.16, y para el modelo de dos compartimientos en los diferentes escenarios se ilustran en las figuras 3.17, 3.18, 3.19, 3.20 y 3.21.

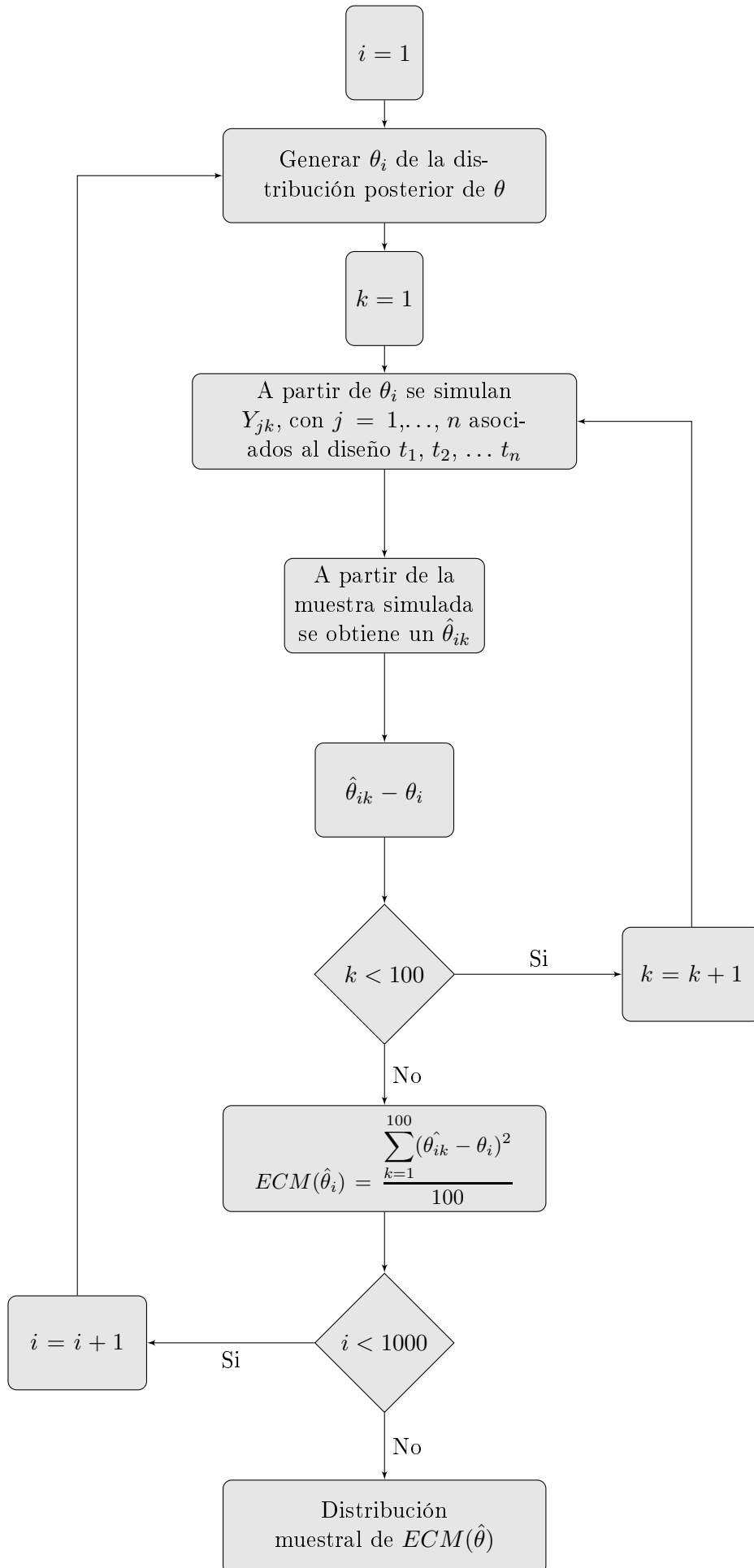


Figura 3.10: Algoritmo para calcular la distribución muestral de $ECM(\hat{\theta})$, asociados al diseño óptimo $\xi = (t_1, t_2, \dots, t_n)$.

Error Cuadrático Medio como medida de evaluación para los diseños óptimos asociados al modelo de un compartimento

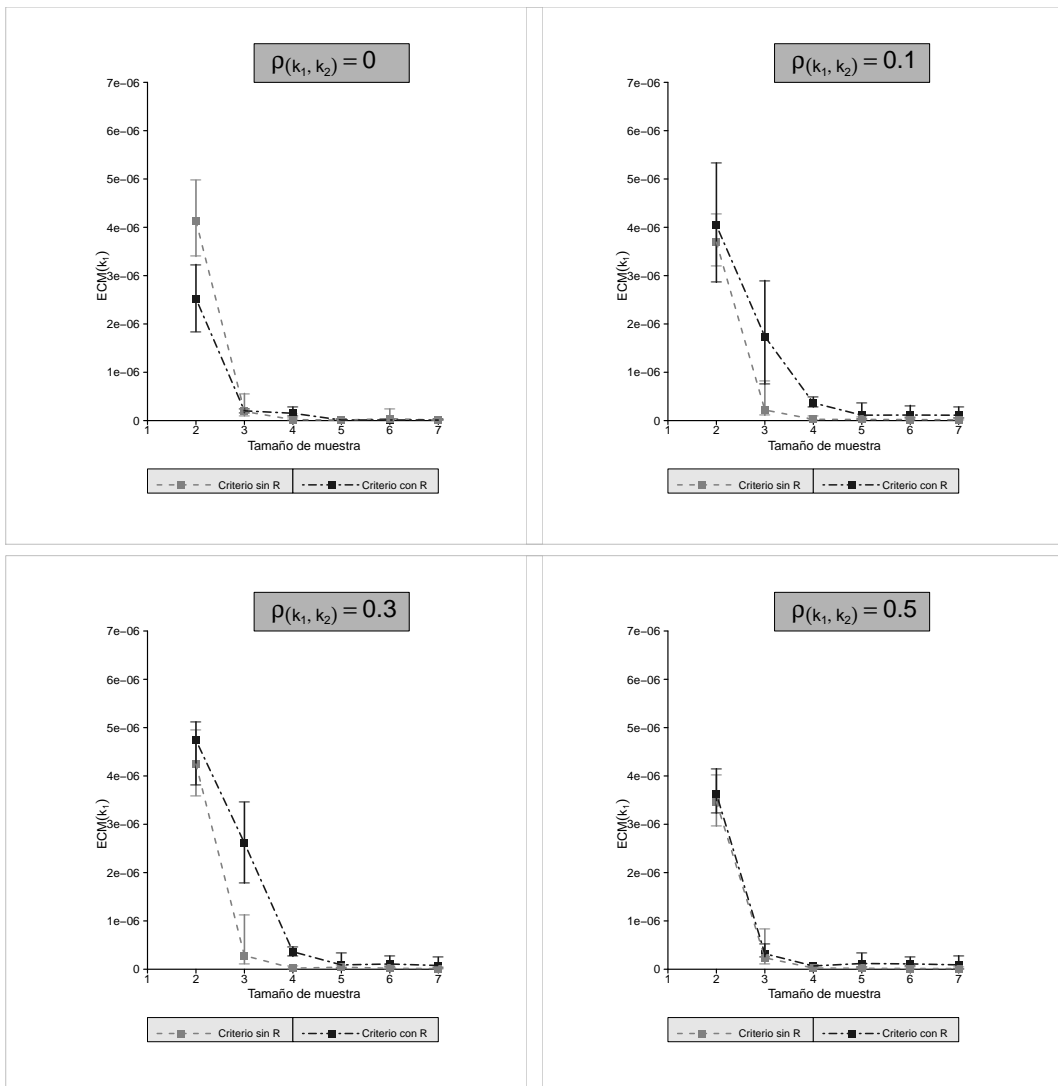


Figura 3.11: Comparación del error cuadrático medio para k_1 obtenido por medio de los diseños óptimos asociados a los criterios 2.10 (criterio sin R) y 2.11 (criterio con R) para los diferentes tamaños de muestra y niveles de correlación entre k_1 y k_2 . Se presenta el primer, segundo y tercer cuartil de la distribución del ECM.

En las figuras 3.11 y 3.12 se puede apreciar el error cuadrático medio (ECM) asociado a la estimación del parámetro k_1 , obtenido bajo el esquema ilustrado en el diagrama 3.10. Se observa que para pequeños tamaños de muestra, los diseños obtenidos a partir del criterio 2.10 (criterio sin la matriz de precisión R) muestran un mejor desempeño en comparación a los obtenidos bajo el criterio 2.11 (criterio con la matriz de precisión R). Pero para tamaños de muestra de cinco en adelante los criterios obtenidos con ambos criterios presentan un igual ECM. Lo anterior es coherente con lo observado anteriormente en la estructura de los

diseños (figura 3.3), pues el criterio 2.11 tiende a definir la mayor cantidad de muestras un una sola parte de la región diseño, generando así una estimación deficiente del parámetro k_1 para pequeños tamaños de muestra.

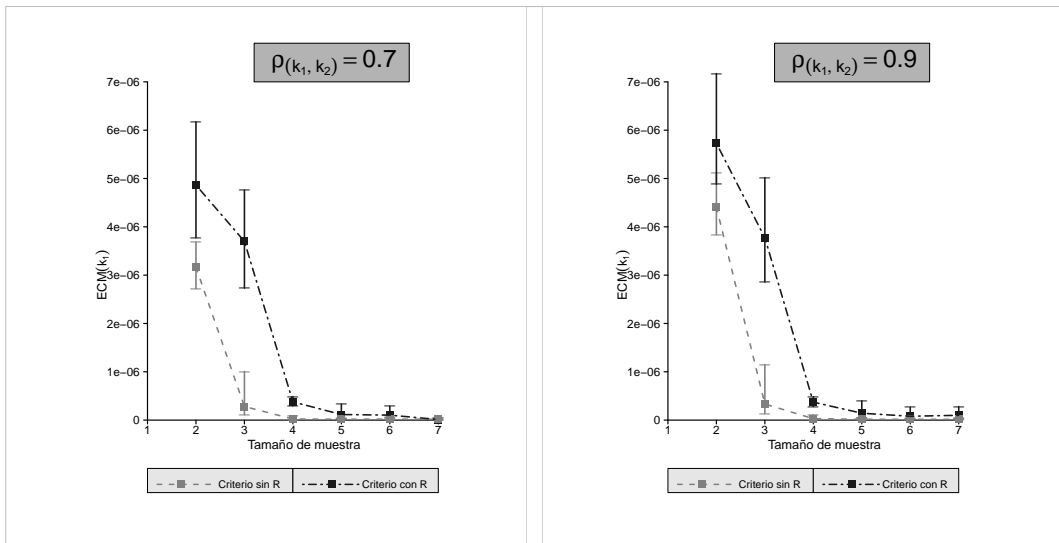


Figura 3.12: Comparación del error cuadrático medio para k_1 obtenido por medio de los diseños óptimos asociados a los criterios 2.10 (criterio sin R) y 2.11 (criterio con R) para los diferentes tamaños de muestra y niveles de correlación entre k_1 y k_2 . Se presenta el primer, segundo y tercer cuartil de la distribución del ECM.

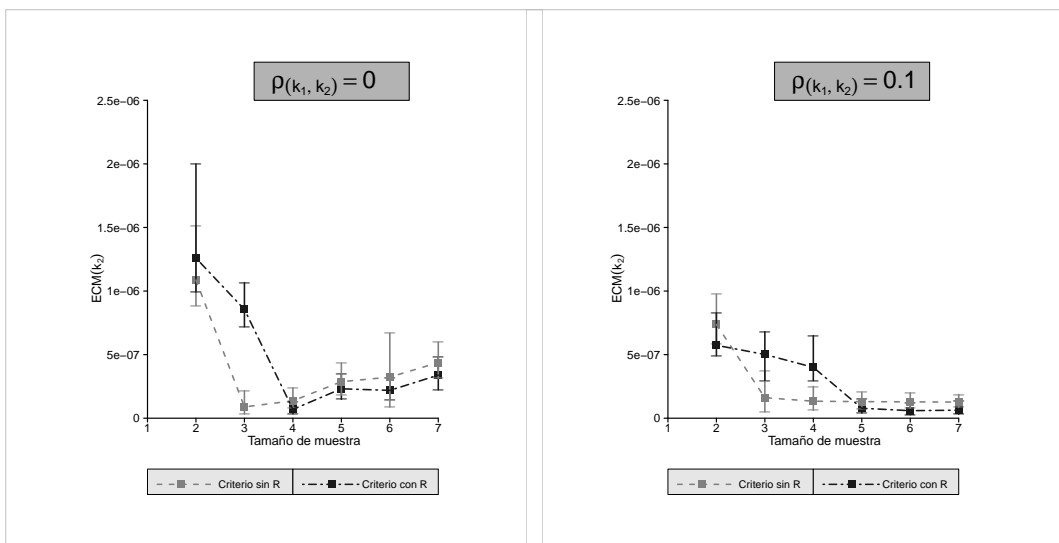


Figura 3.13: Comparación del error cuadrático medio para k_2 obtenido por medio de los diseños óptimos asociados a los criterios 2.10 (criterio sin R) y 2.11 (criterio con R) para los diferentes tamaños de muestra y niveles de correlación entre k_1 y k_2 . Se presenta el primer, segundo y tercer cuartil de la distribución del ECM.

En las figuras 3.13 y 3.14 se aprecia el mismo comportamiento del ECM (para tamaños de muestra pequeños, $n \leq 4$) al observado bajo la estimación de k_1 . Con el criterio 2.10

bajo pequeños tamaños de muestra, los diseños óptimos presentan un mejor desempeño en función del ECM, en comparación a los obtenidos bajo el criterio 2.11. Para tamaños de muestra entre cinco y siete puntos los diseños óptimos obtenidos bajo ambos criterios presentan el mismo comportamiento.

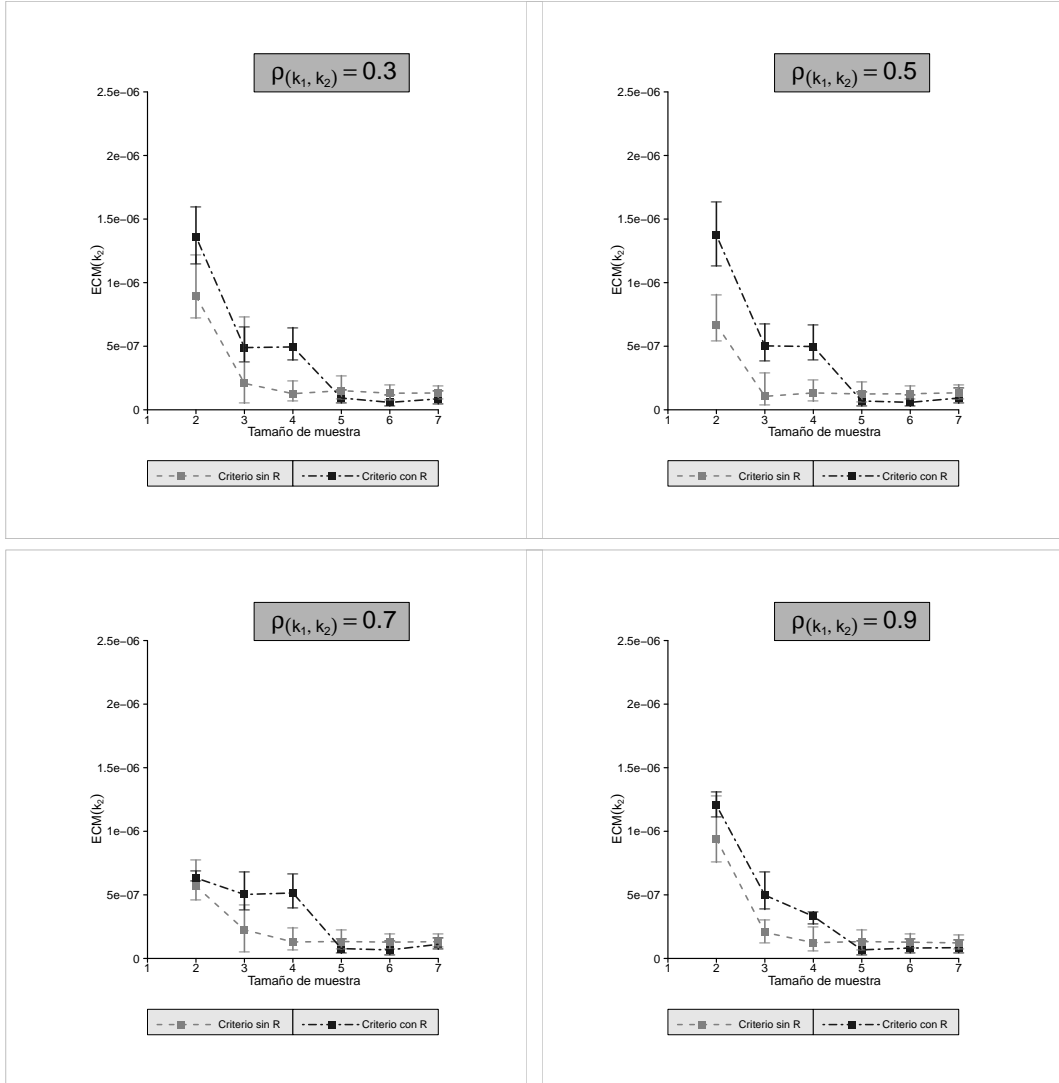


Figura 3.14: Comparación del error cuadrático medio para k_2 obtenido por medio de los diseños óptimos asociados a los criterios 2.10 (criterio sin R) y 2.11 (criterio con R) para los diferentes tamaños de muestra y niveles de correlación entre k_1 y k_2 . Se presenta el primer, segundo y tercer cuartil de la distribución del ECM.

En las figuras 3.15 y 3.16 se puede ver el ECM asociado a la estimación de τ y λ . En general, éste presentan la misma tendencia que para el caso de k_1 y k_2 . El criterio 2.10 presenta un mejor desempeño para pequeños tamaños de muestra ($n \leq 4$).

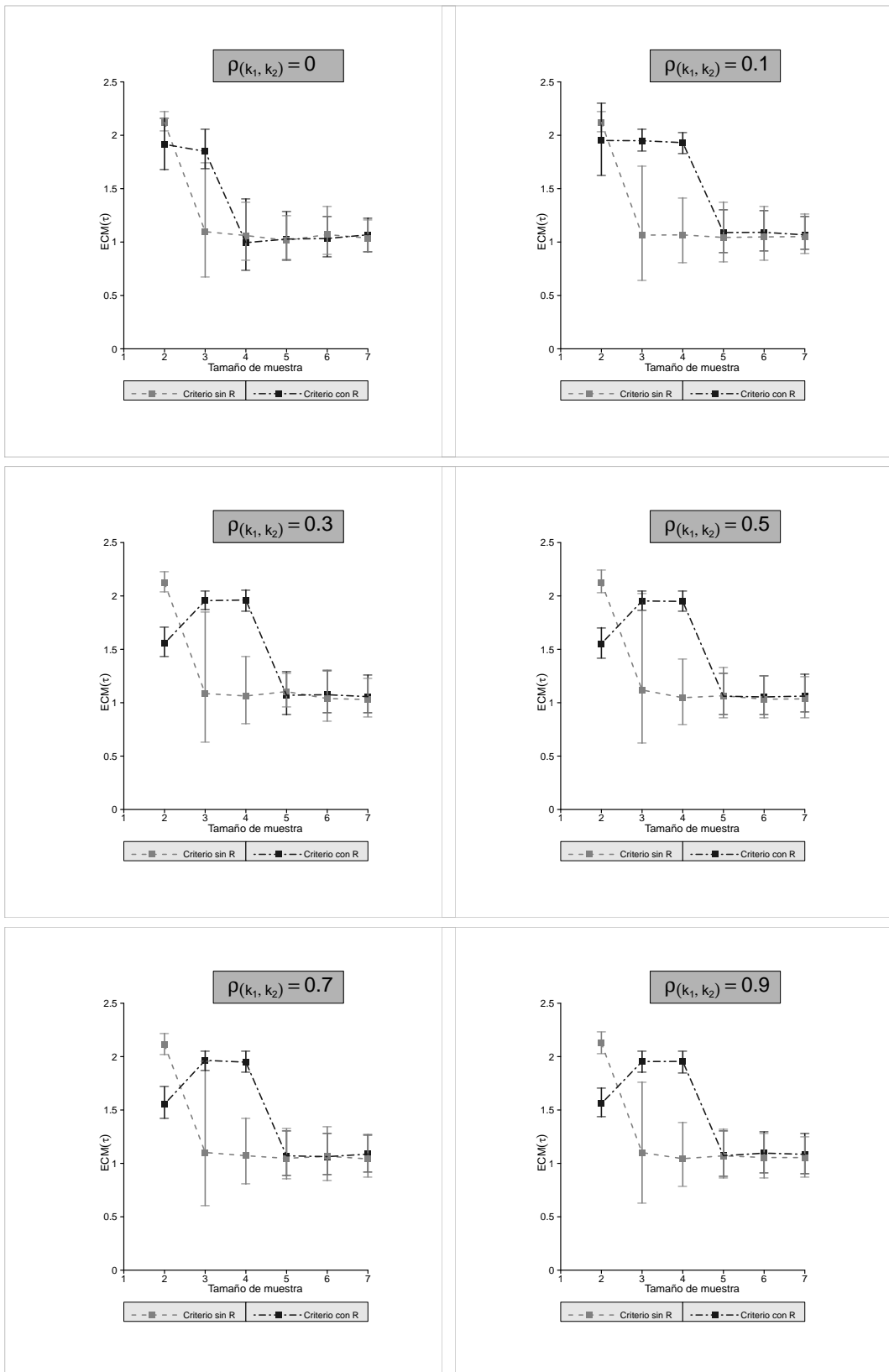


Figura 3.15: Comparación del error cuadrático medio para τ obtenido por medio de los diseños óptimos asociados a los criterios 2.10 (criterio sin R) y 2.11 (criterio con R) para los diferentes tamaños de muestra y niveles de correlación entre k_1 y k_2 . Se presenta el primer, segundo y tercer cuartil de la distribución del ECM.

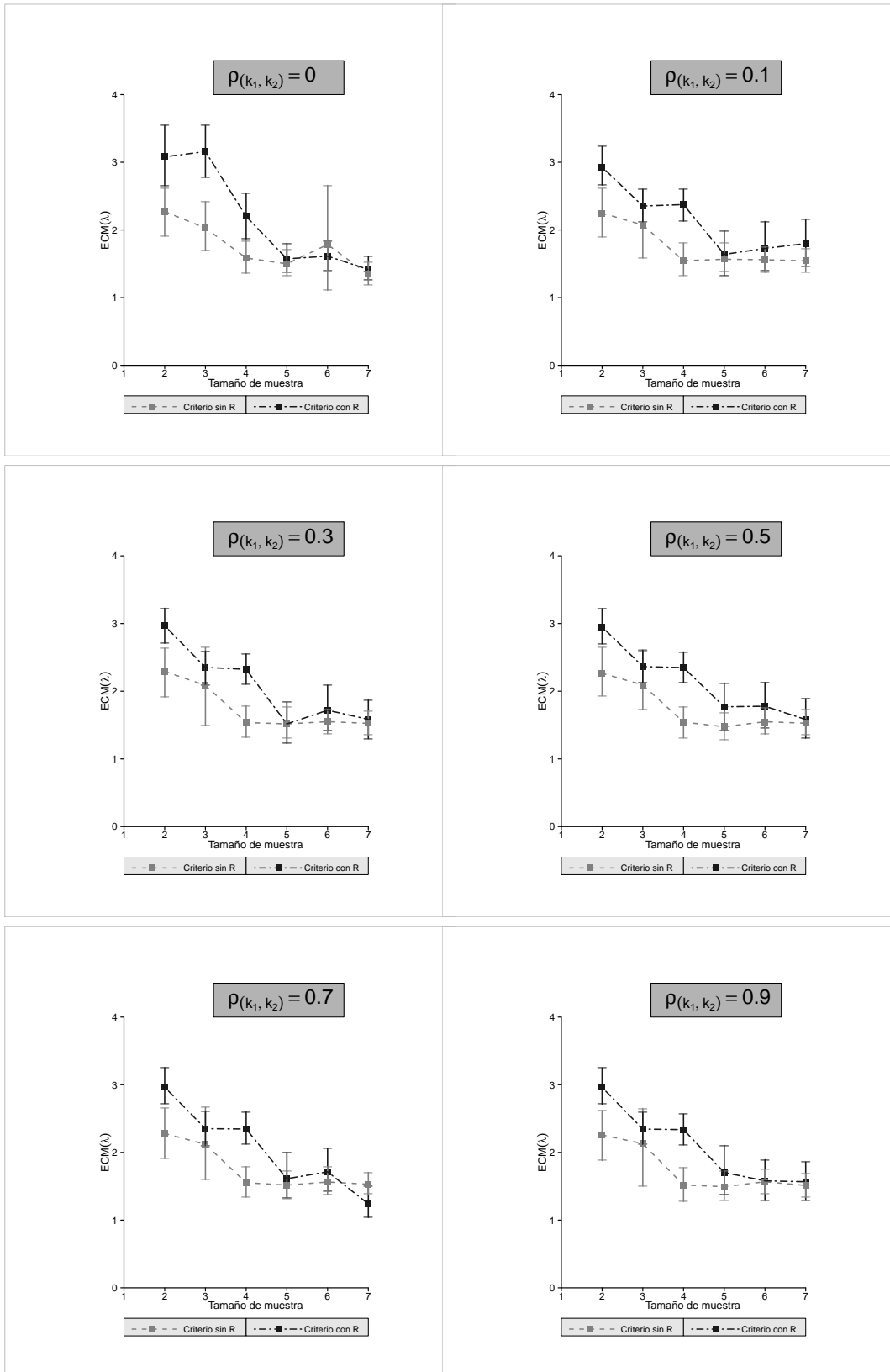


Figura 3.16: Comparación del error cuadrático medio para λ obtenido por medio de los diseños óptimos asociados a los criterios 2.10 (criterio sin R) y 2.11 (criterio con R) para los diferentes tamaños de muestra y niveles de correlación entre k_1 y k_2 . Se presenta el primer, segundo y tercer cuartil de la distribución del ECM.

Error Cuadrático Medio como medida de evaluación para los diseños óptimos asociados al modelo de dos compartimentos

Para los parámetros estimados asociados al modelo de dos compartimentos, se puede apreciar la comparación del ECM en las figuras 3.17, 3.18, 3.19, 3.20 y 3.21. Al igual que en el caso del ECM para los parámetros estimados del modelo de un compartimento, se puede ver que para tamaños de muestra entre dos y cuatro tiempos el ECM asociado al criterio 2.13 (criterio con la matriz de precisión R) es un poco mayor al ECM bajo el criterio 2.12, pero para tamaños de muestra entre cinco y siete tiempos el ECM bajo el criterio 2.13 es menor en algunos casos (como se aprecia en la figura 3.19 para k_{12}) y prácticamente igual para el resto de parámetros. En general bajo el criterio 2.12 (criterio sin la matriz de información a priori R) se obtienen mejores diseños para pequeños tamaños de muestra $n \leq 4$ según el ECM, pero para tamaños entre cinco y siete puntos ambos criterios presentan igual desempeño.

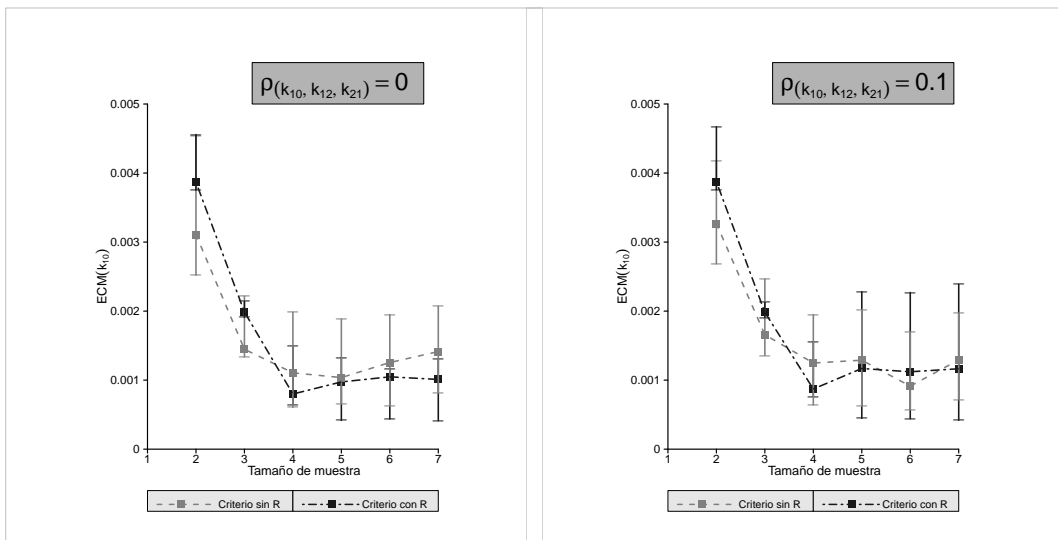


Figura 3.17: Comparación del error cuadrático medio para k_{10} obtenido por medio de los diseños óptimos asociados a los criterios 2.12 (criterio sin R) y 2.13 (criterio con R) para los diferentes tamaños de muestra y estructuras de correlación entre k_{10} , k_{12} y k_{21} . Se presenta el primer, segundo y tercer cuartil de la distribución del ECM.

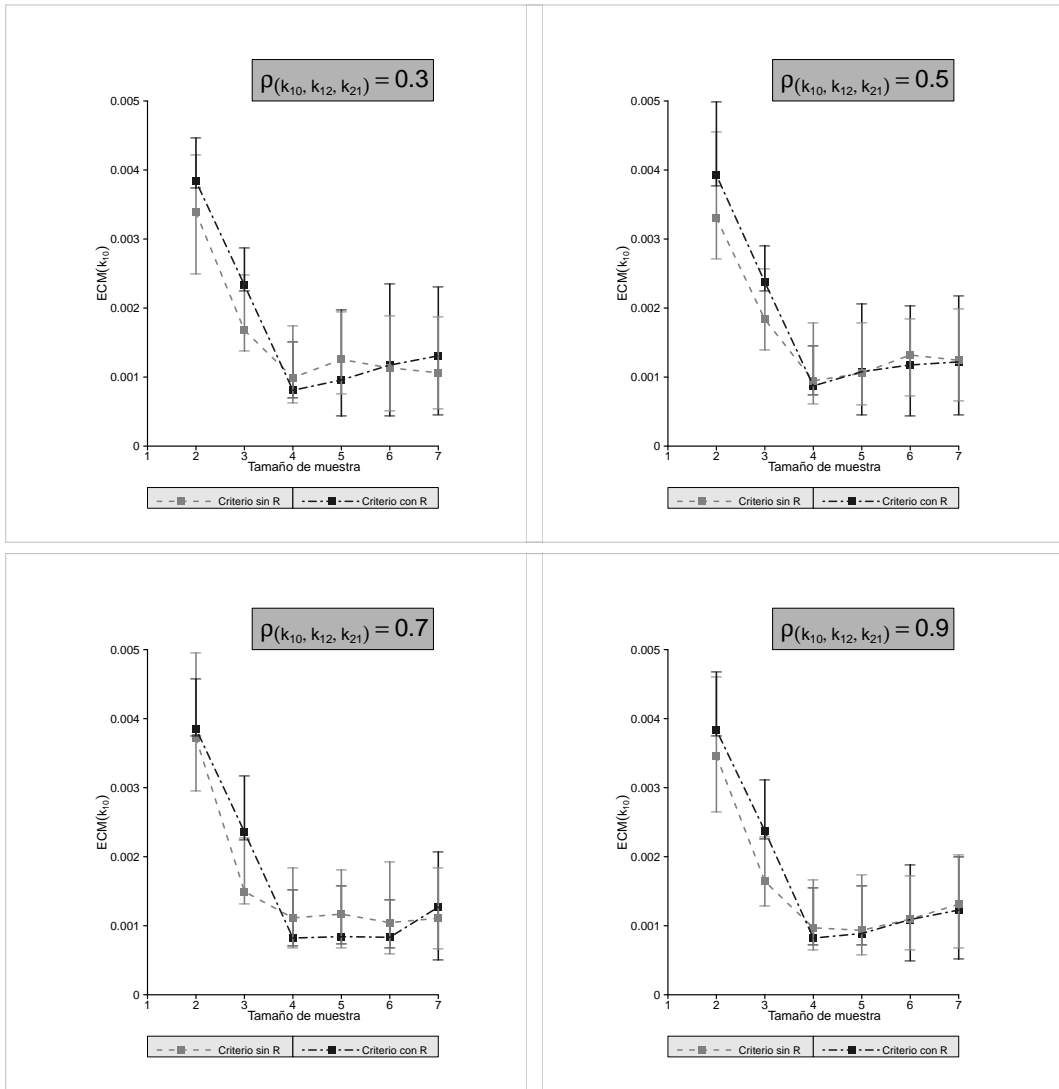


Figura 3.18: Comparación del error cuadrático medio para k_{10} obtenido por medio de los diseños óptimos asociados a los criterios 2.12 (criterio sin R) y 2.13 (criterio con R) para los diferentes tamaños de muestra y estructuras de correlación entre k_{10} , k_{12} y k_{21} . Se presenta el primer, segundo y tercer cuartil de la distribución del ECM.

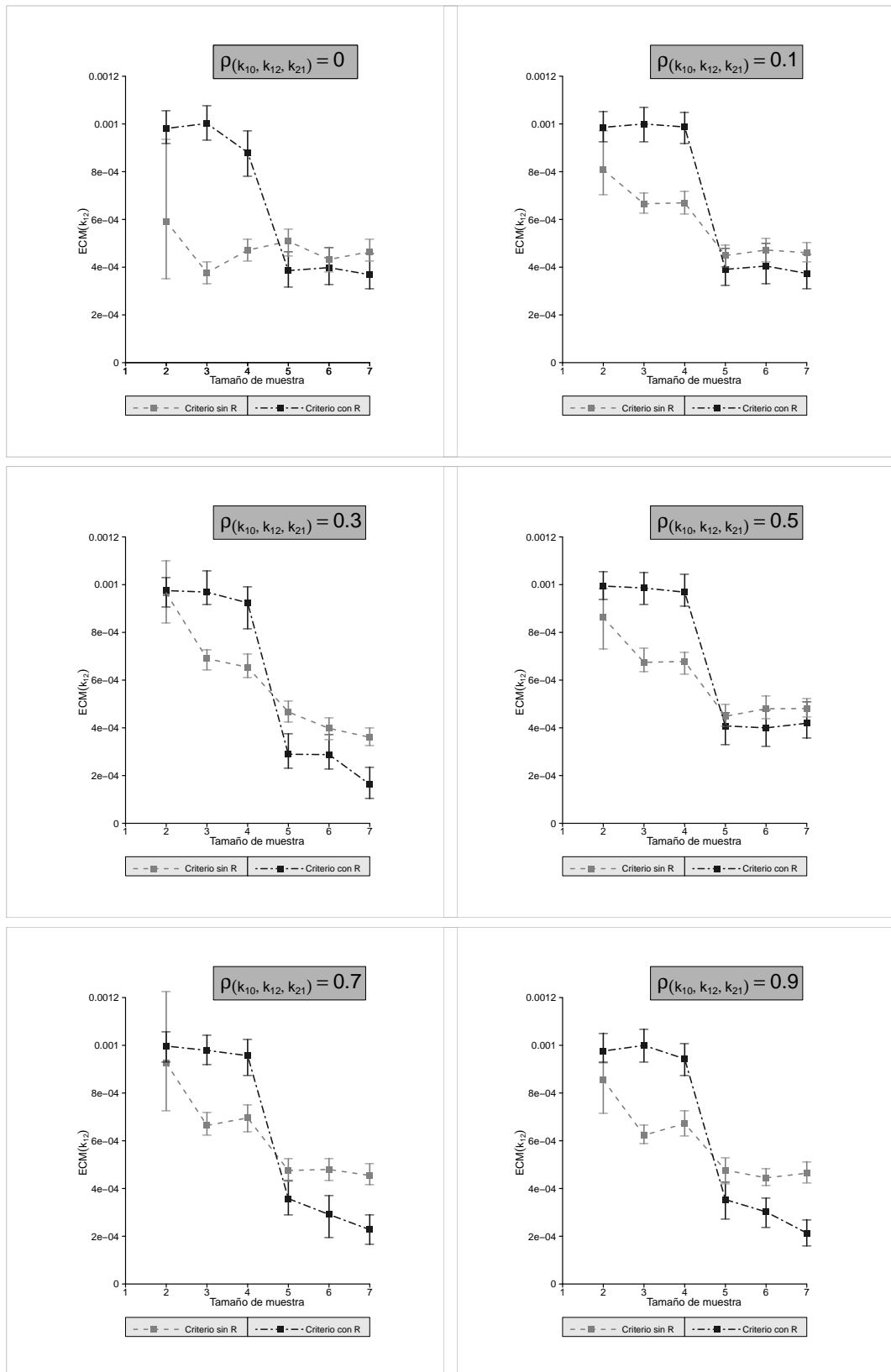


Figura 3.19: Comparación del error cuadrático medio para k_{12} obtenido por medio de los diseños óptimos asociados a los criterios 2.12 (criterio sin R) y 2.13 (criterio con R) para los diferentes tamaños de muestra y estructuras de correlación entre k_{10} , k_{12} y k_{21} . Se presenta el primer, segundo y tercer cuartil de la distribución del ECM.

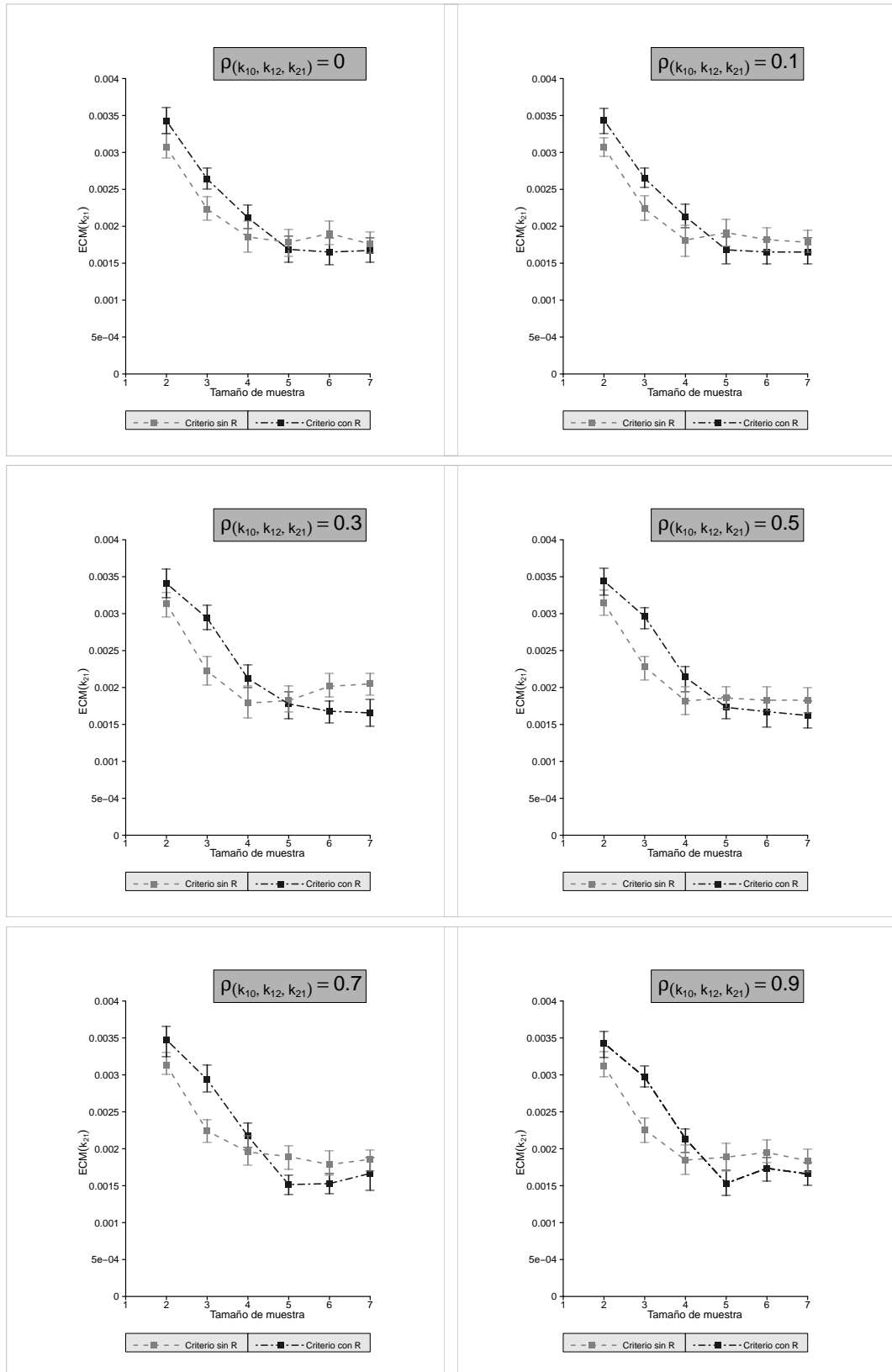


Figura 3.20: Comparación del error cuadrático medio para k_{21} obtenido por medio de los diseños óptimos asociados a los criterios 2.12 (criterio sin R) y 2.13 (criterio con R) para los diferentes tamaños de muestra y estructuras de correlación entre k_{10} , k_{12} y k_{21} . Se presenta el primer, segundo y tercer cuartil de la distribución del ECM.

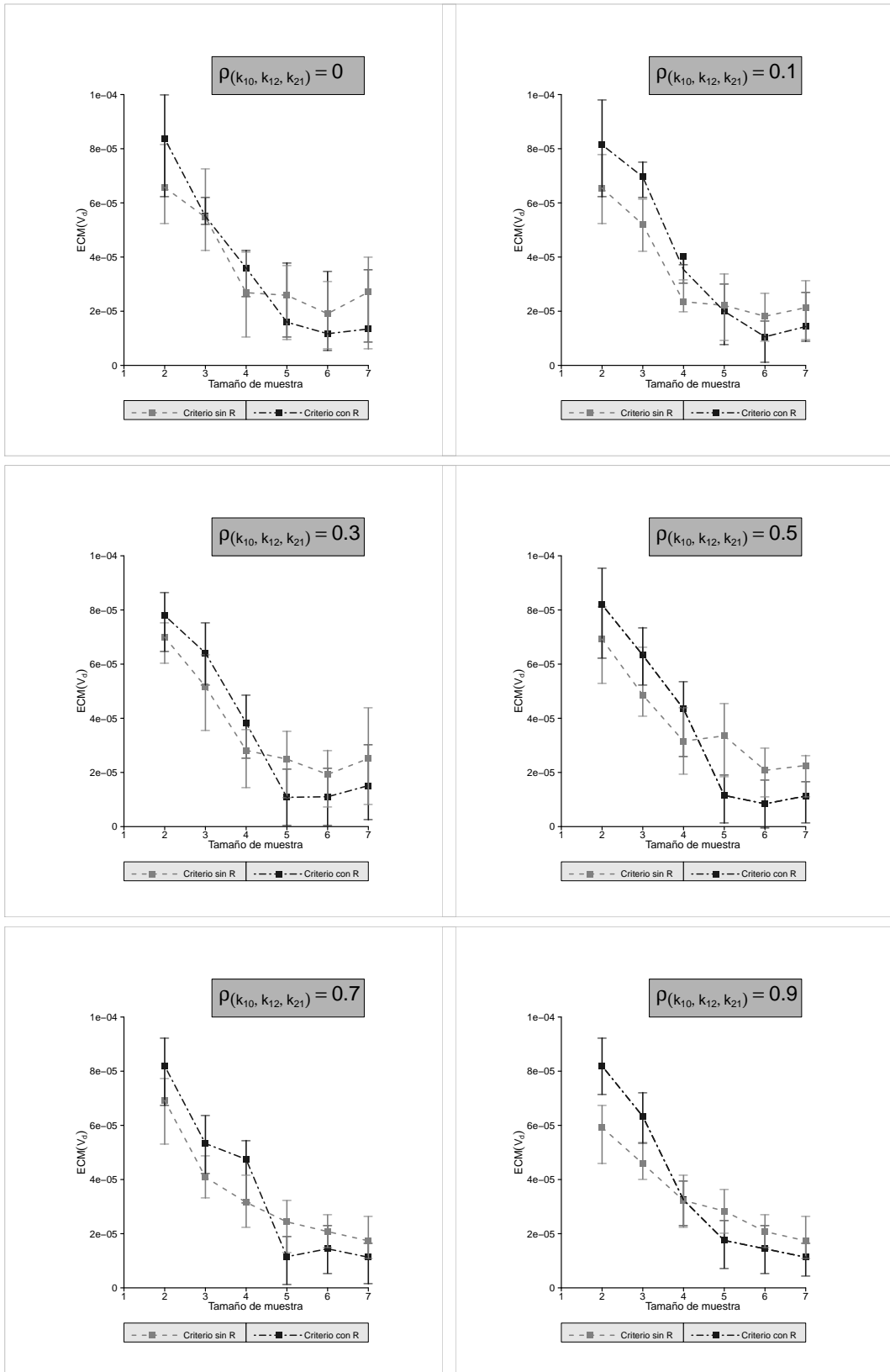


Figura 3.21: Comparación del error cuadrático medio para V_d obtenido por medio de los diseños óptimos asociados a los criterios 2.12 (criterio sin R) y 2.13 (criterio con R) para los diferentes tamaños de muestra y estructuras de correlación entre k_{10} , k_{12} y k_{21} . Se presenta el primer, segundo y tercer cuartil de la distribución del ECM.

En la tabla 3.4 se puede apreciar una recomendación general asociada a los resultados obtenidos con este estudio de simulación. Dichas recomendaciones solo aplican para los dos modelos tratados en este trabajo.

Tabla 3.4: Recomendación entre los criterios con información a priori y sin información a priori, según la utilidad obtenida y el $ECM(\hat{\theta})$ para los diferentes tamaños de muestra. + representa una calificación positiva y – una calificación negativa.

n	Utilidad Esperada criterio con R	Utilidad Esperada criterio sin R	$ECM(\hat{\theta})$ bajo criterio con R	$ECM(\hat{\theta})$ bajo criterio sin R
2	+	–	–	+
3	+	–	–	+
4	+	–	–	+
5	+	–	+	+
6	+	–	+	+
7	+	–	+	+

Conclusiones

A partir del estudio de simulación realizado en este trabajo con el fin de caracterizar los diseños óptimos obtenidos a partir del criterio D-optimalidad Bayesiano para los modelos de uno y dos compartimientos se concluye lo siguiente:

- En ambos casos, tanto para el modelo de un compartimiento como para el modelo de dos compartimientos, la utilidad obtenida bajo los diferentes escenarios (tamaños de muestra y niveles de correlación) con el criterio 2.9 (criterio con información a priori R) fue mucho mayor en comparación a la obtenida bajo el criterio 2.8 (ver las figuras 3.2 y 3.6). Otra apreciación interesante, es la diferencia en el comportamiento de la utilidad esperada entre los criterios 2.8 y 2.9 (ver figuras 3.4, 3.5, 3.8 y 3.9), pues cuando no se considera la matriz de precisión R en el criterio de optimalidad la superficie es totalmente caótica, pero cuando si se considera dicha matriz, la superficie es totalmente suave; existe una gran diferencia cuando se considera la información a priori (a través de R) en el criterio D-optimalidad Bayesiano.
- Referente a la estructura de los diseños óptimos para el modelo de un compartimiento, se pudo apreciar que cuando no se considera correlación entre las tasas de absorción y eliminación (k_1 y k_2) la forma de los diseños es diferente a cuando se considera algún nivel de correlación entre dichas tasas o parámetros. En el nivel de incorrelación entre k_1 y k_2 los diseños tienden a considerar la toma de muestras en tiempos mayores a 120 minutos, mientras que en lo demás niveles de correlación los diseños tienden a tomar muestras en tiempos menores a 95 minutos. En general, considerar algún nivel de correlación entre las tasas de absorción y eliminación tiene influencia sobre la estructura de los diseños obtenidos para el modelo de un compartimiento.
- La estructura de los diseños óptimos para el modelo de dos compartimientos, bajo el criterio sin información a priori (criterio sin R) presenta un comportamiento similar bajo los diferentes niveles de correlación entre las tasas k_{10} , k_{12} y k_{21} . Para este modelo, bajo el criterio sin R , considerar o no la correlación entre dichos parámetros no tiene impacto sobre la estructura de los diseños óptimos. Algo diferente ocurre con la estructura de los diseños óptimos para el modelo de dos compartimientos bajo el criterio con información a priori (criterio con R), pues para tamaños de muestra entre 2 y 4 tiempos la estructura de los diseños es similar bajo todos los niveles de correlación, pero cuando se consideran tamaños de muestra entre 5 y 7 tiempos los niveles de correlación altos ($> 70\%$) entre las tasas k_{10} , k_{12} y k_{21} si generan influencia sobre la estructura de los diseños.

-
- El criterio con información a priori (criterio con R), tanto para el caso del modelo de un compartimiento como en el de dos compartimientos, presentó una tendencia a generar diseños concentrados en pocos puntos sobre la región de diseño (sobre todo para tamaños de muestra entre 2 y 4 tiempos), lo cual tiene un impacto negativo en la estimación de los parámetros de interés, pues el ECM a partir de los diseños asociados a este criterio fue mayor al obtenido con los diseños asociados al criterio sin información a priori (criterio sin R).
 - Un aspecto sobresaliente en este trabajo, fue la combinación de la metodología de **Evolución Diferencial** con la programación en paralelo, para optimizar ambos criterios Bayesianos (criterio D-optimalidad con R y sin R) bajo los modelos de uno y dos compartimiento. Se aprovechó las ventajas de la Evolución Diferencial para optimizar funciones complejas como las asociadas al criterio D-optimalidad Bayesiano y se redujeron significativamente los tiempos de cálculo asociados a la multiple evaluación de la función objetivo al paralelizar las rutinas implementadas en **R**.

Trabajo futuro

A partir de lo realizado en este trabajo, quedan muchos aspectos que se deberían explorar en futuros proyectos de investigación, entre los cuales se encuentran:

- Realizar el estudio de simulación bajo otras distribuciones diferentes a la normal.
- Caracterizar los diseños óptimos obtenidos bajo otros criterios como el A-optimalidad Bayesiano.
- Implementar una librería en el paquete **R** con el fin que las personas o investigadores en farmacología puedan emplear criterios estadísticos como el A y D-optimalidad Bayesianos para obtener diseños $\xi = (t_1, t_2, \dots, t_n)$ con un respaldo estadístico.

Anexos

Evolución Diferencial y Programación en Paralelo con R

Evolución Diferencial

La Evolución Diferencial fue propuesta por estudiantes de Technical University of Berlin (TUB) (Beyer y Schwefel, 2002). Éstos son métodos de búsqueda basados en la idea de la evolución natural de los seres vivos y presentan ventajas con respecto a los métodos tradicionales de optimización. En especial, estos métodos iterativos convergen al óptimo global de la función objetivo de interés, sin importar su grado de complejidad y su dominio, evitando procedimientos adicionales que en ocasiones se desvían hacia óptimos locales. Desde un enfoque genético, se parte de un conjunto de individuos progenitores, los cuales son mutados y posteriormente cruzados, generándose nuevos individuos con aquellas características que más dominaron en los progenitores.

En el **capítulo 2** se definieron cuatro funciones (ecuaciones 2.10, 2.11, 2.12 y 2.13) a optimizar bajo diferentes escenarios de simulación. Debido a la complejidad que presentan dichas funciones, los algoritmos tradicionales para optimización no funcionan de manera adecuada para resolver uno de los problemas de interés en este trabajo (hallar diseños $\xi = [t_1, t_2, \dots, t_n]$ que generen una estimación óptima de los parámetros asociados a los modelos 1.2 y 1.4 bajo diferentes tamaños de muestra y niveles de correlación entre éstos). En el siguiente diagrama se presenta la estrategia evolutiva aplicada al problema citado anteriormente (ver Beyer y Schwefel), la cual se encuentra implementada en la librería "**DEoptim**" del paquete **R**. *NP* hace referencia al número de individuos en cada generación, *F* ($0 \leq F \leq 2$, aunque usualmente se le asignan valores menores que 1) es un parámetro que controla la tasa de mutación, *GR* es un parámetro que controla la tasa de recombinación ($0 \leq GR \leq 1$) y *NG* es el número de generaciones, el cual se define como criterio de parada.

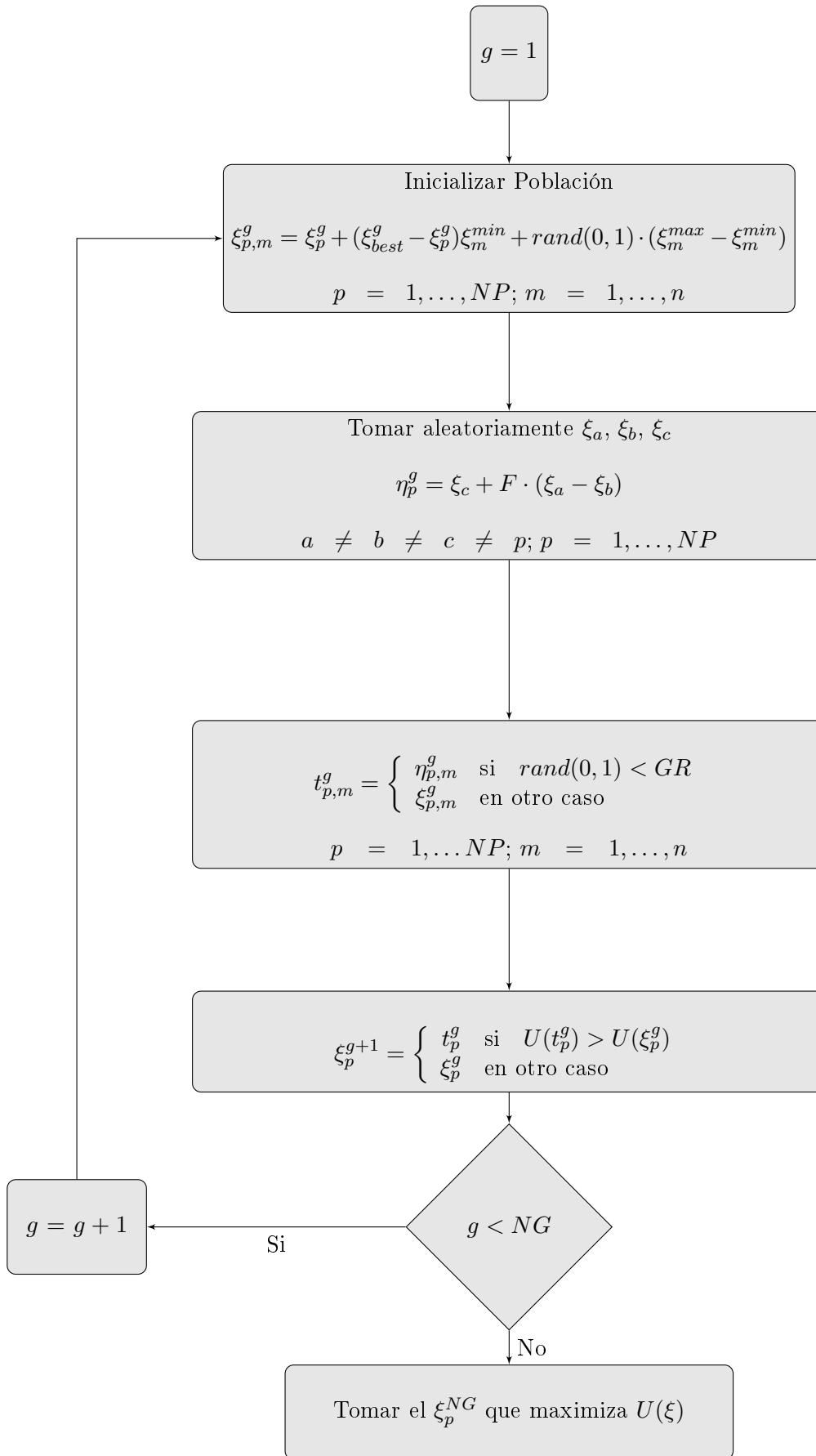


Figura 22: Evolución diferencial aplicada al criterio D-optimalidad bayesiano.

Algunas aplicaciones de algoritmos evolutivos

Brabazon presenta el desarrollo de un Algoritmo Genético para encontrar los parámetros óptimos de un Modelo de Regresión de Cox del Análisis de Supervivencia para pacientes del servicio de Diálisis Peritoneal del Hospital Clínico Universitario de Caracas entre 1980 y 2000. Hace uso de la técnica de los Algoritmos Genéticos como método de búsqueda de una mejor estimación de los parámetros del Modelo de Cox al obtenido por los métodos clásicos de optimización. El algoritmo desarrollado permitió obtener estimaciones de los parámetros del Modelo de Regresión de Cox, y con mejor valor de AIC, a los obtenidos utilizando los métodos clásicos.

Brabazon y Keenan examinan el potencial del modelo de una red neuronal, cuyos insumos y estructura se seleccionan automáticamente por medio de un algoritmo genético, para la predicción del fracaso empresarial mediante la información extraída de estados financieros. Los resultados de este modelo son comparados con los de un análisis discriminante lineal.

Stewardson y Whitfield demuestran que el uso del diseño experimental en el desarrollo de la estructura de los algoritmos genéticos, es una estrategia particularmente eficaz y eficiente. Utilizando datos generados a partir de un diseño experimental factorial completo, con 27 corridas experimentales, se evaluaron las condiciones óptimas de operación de los parámetros de un algoritmo genético especial. Demostraron que resultados similares se obtienen con tan sólo nueve corridas.

Backhouse, Fotheringham and Allan describen la optimización de un proceso textil por medio de dos enfoques diferentes: el uso de un algoritmo genético y el uso de una técnica de diseño experimental. Se considera que la eficacia de cada método para encontrar el óptimo y la naturaleza de la información obtenida sobre el proceso.

Hamada *et al.* muestran como un algoritmo genético puede ser usado para encontrar una aproximación del nivel óptimo de diseños experimentales bayesianos para modelos de regresión. Hacen una descripción de los algoritmos genéticos y su aplicación al diseño experimental. La metodología se ilustran en una amplia gama de ejemplos: regresión lineal y no lineal, factores individuales y múltiples, experimentos con datos distribuidos normal y Bernoulli.

Programación en paralelo con R

Realizar cálculos o programar en paralelo es una manera de realizar varios cálculos de manera simultánea. En la forma tradicional de programar los cálculos se realizan en serie, lo cual en ocasiones puede ser desfavorable por los altos tiempos de espera. Cuando se programa una tarea en paralelo, dicha tarea se distribuye a varios procesadores (físicos o virtuales), los cuales por separado realizan la parte que les corresponde. En cambio, cuando se programa en serie, toda la tarea es asignada a un sólo procesador.

Existen tres escenarios para realizar un cálculo en paralelo

- Varios procesadores en un mismo chip (procesadores virtuales).
- Varios chips en un mismo computador (procesadores físicos).
- Varios computadores conectados en una red o cluster.

En R existen varias formas de realizar la paralelización. Entre las más efectivas se encuentran:

- **snow**, el cual se basa en códigos vectorizados por medio de funciones como `apply()` para realizar la división de las tareas.
- **foreach**, el cual usa un bucle **for** para realizar la división en las tareas.
- **multicore**, que solo es adecuado para hardware de varios núcleos.

Uno de los problemas abordados en este trabajo, fue la maximización de algunas funciones bastante complejas (ecuaciones 2.10, 2.11, 2.12 y 2.13), lo cual se logró por medio de algoritmos evolutivos. Uno de las desventajas de este método es la gran cantidad de evaluaciones que tiene que hacer sobre la función objetivo. Para superar este problema se recurrió a la programación en paralelo anteriormente citada. En el gráfico 23 se presentan las mejoras alcanzadas aplicando esta forma de programar.

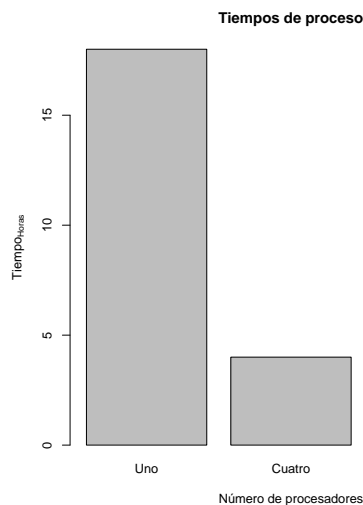


Figura 23: Tiempos de ejecución para el cálculo de la distribución del ECM para τ .

Para paralelizar el código desarrollado en este trabajo se recurrió a la librería **snow**, pero para poder usar este paquete, el código debe estar completamente vectorizado como en este caso. Se puede apreciar que las mejoras son bastante significativas, pues cuando se programaba en serie, los tiempos de espera eran de un poco más de 18 horas, pero cuando

se paralelizó los tiempos se redujeron drásticamente al pasar a tiempos hasta de dos horas al usar seis procesadores. Tierney, Rossini and Li comenta algunas escalas de tiempo comparando algoritmos programados en serie y en paralelo:

Tabla 5: Comparación de tiempos de ejecución usando un procesador vs treinta procesadores

Un procesador	treinta procesadores
1 minuto	2 segundos
1 hora	2 minutos
1 día	1 hora
1 mes	1 día
1 año	2 semanas

Todos los cálculos realizados en este trabajo se ejecutaron en la Unidad de Cálculo Avanzado (**UNICA**) de la Facultad de Minas, Universidad Nacional de Colombia Sede Medellín. Las características del cluster son las siguientes:

- Marca: DELL PowerdEdge 2950.
- 100 procesadores Intel EM64T.
- 200 GB de memoria ram.
- 1.59 Ter-abites de capacidad para almacenamiento.
- Sistema operativo: Linux/GNU Centos 5.4 arquitectura Amd64.

Matriz de Información

A continuación se presenta la matriz de información, la cual es necesaria para definir el criterio D-optimalidad Bayesiano. Para un modelo no lineal la matriz de información está dada por:

$$\psi(\theta, \xi) = \sum_{i=1}^n f(t_i, \theta) f^T(t_i, \theta),$$

donde,

$$f(t_i, \theta) = \begin{bmatrix} \frac{\partial Y(t_i)}{\partial \theta_1} \\ \frac{\partial Y(t_i)}{\partial \theta_2} \\ \vdots \\ \frac{\partial Y(t_i)}{\partial \theta_k} \end{bmatrix}.$$

Para el modelo de un compartimiento se tiene que $\theta = (\lambda, k_1, k_2, \tau)$ y para el modelo de dos compartimientos $\theta = (k_{10}, k_{12}, k_{21}, V_d)$.

Derivadas asociadas al modelo de un compartimiento:

$$\frac{\partial Y(t_i)}{\partial \lambda} = -\frac{K_1 * M_0}{\lambda^2 (k_1 - k_2)} * (e^{-k_2(t_i - \tau)} - e^{-k_1(t_i - \tau)}),$$

$$\frac{\partial Y(t_i)}{\partial k_1} = \frac{M_0 * k_1}{\lambda (k_1 - k_2)} * [e^{-k_1(t_i - \tau)} * (t_i - \tau)] + [e^{-k_2(t_i - \tau)} - e^{-k_1(t_i - \tau)}] * \left[\frac{\lambda * (k_1 - k_2) * M_0 - k_1 * M_0 * \lambda}{\lambda^2 * (k_1 - k_2)^2} \right],$$

$$\frac{\partial Y(t_i)}{\partial k_2} = \frac{M_0 * k_1}{\lambda (k_1 - k_2)} * [-e^{-k_1(t_i - \tau)} * (t_i - \tau)] + [e^{-k_2(t_i - \tau)} - e^{-k_1(t_i - \tau)}] * \left[\frac{k_1 * M_0}{\lambda^2 * (k_1 - k_2)^2} \right],$$

$$\frac{\partial Y(t_i)}{\partial \tau} = \frac{k_1 * M_0}{\lambda * (k_1 - k_2)} * [e^{-k_2(t_i - \tau)} * k_2 - e^{-k_1(t_i - \tau)} * k_1].$$

Las derivadas asociadas al modelo de dos compartimientos son muy extensas, por ende no se presentarán en este documento, en caso de ser requeridas para su verificación, puede solicitarlas a los autores de este trabajo, o si se requieren para efectos de programación, entonces se pueden tomar del código presentado al final de este apéndice.

Deducción del criterio D-optimalidad Bayesiano para el modelo lineal

Para un modelo de regresión lineal de la forma:

$$E(y | \theta, \sigma^2) = X\theta,$$

donde, $Y | \theta, \sigma^2 \sim N(X\theta, \sigma^2 I_n)$ y la distribución a priori para θ es $N(\theta_0, \sigma^2 R^{-1})$, con $\theta \in \mathbb{R}^k$, $R \in \mathbb{R}^{k \times k}$, σ^2 y R son conocidas.

$$P(\theta) \propto \exp \left[-\frac{1}{2\sigma^2} (\theta - \theta_0)^T R (\theta - \theta_0) \right]$$

$$L(\theta | y, \xi) \propto \exp \left[-\frac{1}{2\sigma^2} (y - X\theta)^T (y - X\theta) \right]$$

La distribución posterior de θ está dada por:

$$P(\theta | y, \xi) \propto \exp \left\{ -\frac{1}{2\sigma^2} \left[\underbrace{(\theta - \theta_0)^T R(\theta - \theta_0) + (y - X\theta)^T (y - X\theta)}_{\delta} \right] \right\},$$

donde,

$$\delta = (\theta - \theta_0)^T R(\theta - \theta_0) + (y - X\theta)^T (y - X\theta),$$

$$\delta = y^T y - 2\theta^T X^T y + \theta^T \underbrace{X^T X}_{nM} \theta + \theta^T R\theta - \theta^T R\theta_0 - \theta_0^T R\theta + \theta_0^T R\theta_0,$$

$$\delta = -\theta^T (X^T y + R\theta_0) - \theta^T X^T y + \theta^T (nM + R)\theta - \theta_0^T R\theta,$$

$$\delta = -\theta^T (X^T y + R\theta_0) + \theta^T (nM + R)\theta - \theta^T (X^T y + R\theta_0),$$

$$\delta = -\theta^T (nM + R) \underbrace{(nM + R)^{-1} (X^T y + R\theta_0)}_{\theta^*} + \theta^T (nM + R)\theta - \theta^T (nM + R) \underbrace{(nM + R)^{-1} (X^T y + R\theta_0)}_{\theta^*},$$

$$\delta = (\theta - \theta^*)^T (nM + R)(\theta - \theta^*).$$

$$P(\theta | y, \xi) \propto \exp \left\{ -\frac{1}{2\sigma^2} [(\theta - \theta^*)^T (nM + R)(\theta - \theta^*)] \right\},$$

$$P(\theta | y, \xi) \propto \frac{1}{(2\pi)^{k/2} \det\{\sigma^2(nM+R)\}^{-1}} * \exp \left\{ -\frac{1}{2\sigma^2} [(\theta - \theta^*)^T (nM + R)(\theta - \theta^*)] \right\},$$

donde

$$\theta^* = (nM + R)^{-1} (X^T y + R\theta_0)$$

En el capítulo 2 se definió el criterio D-opimalidad Bayesiano para el modelo lineal, maximizando la ganancia esperada de la información de Shannon, o equivalentemente, al maximizar la distancia esperada de Kullback-Leibler entre la distribución a priori la y posterior de θ :

$$\int \int \log \left\{ \frac{P(\theta | y, \xi)}{P(\theta)} \right\} P(y, \theta | \xi) d\theta dy.$$

Como la distribución a priori no depende del diseño ξ , entonces el diseño que maximiza la ganancia esperada de Shannon, es el que maximiza:

$$U_1(\xi) = \int \int \log \{P(\theta | y, \xi)\} P(y, \theta | \xi) d\theta dy,$$

donde,

$$P(y, \theta | \xi) = P(\theta | y, \xi) * P(y)$$

y

$$\log \{P(\theta | y, \xi)\} = -\frac{k}{2} * \log(2\pi) - \frac{1}{2} * \log \det \{\sigma^{-2}(nM + R)\} - \frac{1}{2\sigma^2} * (\theta - \theta^*)^T (nM + R)(\theta - \theta^*).$$

Entonces

$$U_1(\xi) = \frac{1}{2} \int \int \underbrace{\frac{(\theta - \theta^*)^T (nM + R)(\theta - \theta^*)}{\sigma^2}}_{g(\theta)} * P(\theta | y, \xi) d\theta * P(y) dy + \underbrace{-\frac{k}{2} * \log(2\pi) + \frac{1}{2} * \log \det \{\sigma^{-2}(nM + R)\}}_{E \left[\frac{(\theta - \theta^*)^T (nM + R)(\theta - \theta^*)}{\sigma^2} | y \right]}$$

Se puede demostrar que:

$$E \left[d^T A d \right] = \mu_d^T A \mu_d + \text{tr}(A \Sigma)$$

Tomando el resultado anterior, se tiene que:

$$E \left[\frac{(\theta - \theta^*)^T (nM + R)(\theta - \theta^*)}{\sigma^2} \right] = 0 + \text{tr} \left(\frac{(nM + R)}{\sigma^2} \left[\frac{(nM + R)}{\sigma^2} \right]^{-1} \right)$$

$$E \left[\frac{(\theta - \theta^*)^T (nM + R)(\theta - \theta^*)}{\sigma^2} \right] = \text{tr}(I_k) = k$$

Finalmente se llega a:

$$U_1(\xi) = -\frac{k}{2} - \frac{k}{2} * \log(2\pi) + \frac{1}{2} * \log \det \{\sigma^{-2}(nM + R)\}.$$

Rutinas implementadas en R

Modelo de un compartimiento

```
#####
## Diseño D-optimalidad B.      ###
## Modelo de un compartimiento ###
#####
require(MASS)
require(mvtnorm)
require(DEoptim)
##### Derivadas #####

derik1=function(tiem,lam,k1,k2,tao){
dcp=(k1*100/(lam*(k1-k2)))*(exp(-k1*(tiem-tao))*(tiem-tao)) +
(exp(-k2*(tiem-tao))-exp(-k1*(tiem-tao)))*
((lam*(k1-k2)*100-k1*100*lam)/(lam^2*(k1-k2)^2))
return(dcp)
}
```

```

derik2=function(tiem,lam,k1,k2,tao){
dcp=(k1*100/(lam*(k1-k2)))*(-exp(-k2*(tiem-tao))*(tiem-tao)) +
(exp(-k2*(tiem-tao))-exp(-k1*(tiem-tao)))*
(k1*100)/(lam*(k1-k2)^2)
return(dcp)
}

derilam=function(tiem,lam,k1,k2,tao){
dlam= -(1/lam^2)*(k1*100/(k1-k2))*(exp(-k2*(tiem-tao))-exp(-k1*(tiem-tao)))
return(dlam)
}

deritao=function(tiem,lam,k1,k2,tao){
dtao=(k1*100/(lam*(k1-k2)))*(exp(-k2*(tiem-tao))*k2-exp(-k1*(tiem-tao))*k1)
return(dtao)
}
#####

##función que retorna la matriz f^tf#####
ftf=function(tiem,lam,k1,k2,tao){
mat.varian=c(derik1(tiem,lam,k1,k2,tao),derik2(tiem,lam,k1,k2,tao)
,derilam(tiem,lam,k1,k2,tao),deritao(tiem,lam,k1,k2,tao))
ma=mat.varian%*%t(mat.varian)
return(ma)
}

##Función que retorna la matriz de información#####
matriz.informa=function(x,tiem){
y<-array(sapply(tiem,ftf,x[1],x[2],x[3],x[4]),c(4,4,length(tiem)))
ma=apply(y,c(1,2),sum)
detma=det(length(tiem)*ma)
detma1=ifelse(detma>0,log(detma),0)
if(is.na(detma1))detma1=0
return(detma1)
}

##### Se importa la muestra simulada de la distribución a priori ##
parametros=read.table("/home/jcardonj/Escenario1/base22.txt")
parametros=read.table(file.choose())
parametros=as.matrix(parametros)
##Funcion de utilidad
utility.function1=function(x,y){
ma=cbind(x,y)
inte.utili=function(dise){
utili=apply(parametros,1,matriz.informa,dise)
utili= -mean(utili)
}
utili=apply(ma,1,inte.utili)
return(utili)
}

```

```

### Se define la region de diseño #####
### Éste es el caso para un diseño de dos puntos #####
inf=c(11.40,11.40)
sup=c(360,360)

### Se fija la semilla #####
set.seed(1234)
optimo=DEoptim(utility.function1,inf,sup,control
=DEoptim.control(VTR =-Inf, strategy = 1,NP=30,trace = FALSE))

summary(optimo)

#####
## ECM para evaluar los diseños##
## Modelo de un compartimiento ##
#####
library(snow)
library(nws)
require(MASS)
require(mvtnorm)
require(DEoptim)
##### Derivadas #####
derik1=function(tiem,k1,k2,lam,tao){
dcp=(k1*100/(lam*(k1-k2)))*(exp(-k1*(tiem-tao))*(tiem-tao)) +
(exp(-k2*(tiem-tao))-exp(-k1*(tiem-tao)))*
((lam*(k1-k2)*100-k1*100*lam)/(lam^2*(k1-k2)^2))
return(dcp)
}

derik2=function(tiem,k1,k2,lam,tao){
dcp=(k1*100/(lam*(k1-k2)))*(-exp(-k2*(tiem-tao))*(tiem-tao)) +
(exp(-k2*(tiem-tao))-exp(-k1*(tiem-tao)))*
(k1*100)/(lam*(k1-k2)^2)
return(dcp)
}

derilam=function(tiem,k1,k2,lam,tao){
dlam= -(1/lam^2)*(k1*100/(k1-k2))*(exp(-k2*(tiem-tao))-
exp(-k1*(tiem-tao)))
return(dlam)
}

deritao=function(tiem,k1,k2,lam,tao){
dtao=(k1*100/(lam*(k1-k2)))*(exp(-k2*(tiem-tao))*k2-
exp(-k1*(tiem-tao))*k1)
return(dtao)
}
#####

##### Funciones que generan la matriz de información #####
ftf=function(tiem,k1,k2,lam,tao){
mat.varian=c(derik1(tiem,k1,k2,lam,tao),derik2(tiem,k1,k2,lam,tao),

```

```

derilam(tiem,k1,k2,lam,tao),deritao(tiem,k1,k2,lam,tao))
ma=mat.varian%*%t(mat.varian)
return(ma)
}

matriz.informa=function(x,tiem){
y<-array(sapply(tiem,ftf,x[1],x[2],x[3],x[4]),c(4,4,length(tiem)))
ma=apply(y,c(1,2),sum)
ma=ginv(ma*length(tiem))
#ma=diag(diag(1/(ma*length(tiem))))
return(ma)
}
#####

##Diseño a evaluar
tiem=c(11.42437, 249.1835, 47.53509, 17.03621, 30.30816, 93.64219)
##Matriz de varianzas de la distribución posterior
varianza=matriz.informa(c(0.027,0.010,27.35,11.37),tiem)
##Media de la distribución posterior
media=c(0.027,0.010,27.35,11.37)
#####
## Parámetros simulados (parámetros fijos o verdaderos) de la distribución posterior.
x=rmvnorm(1000,media,varianza,method=c("svd"))

### Se paraleliza el proceso con seis procesadores.
cl <-makeCluster(c("localhost","localhost","localhost","localhost",
"localhost","localhost"),type = "SOCK")

## Función que calcula el ECM.
error.cuadratico.medio=function(i,x,cl){
concentracion=function(x,tiem){
cp=((100*x[1])/(x[3]*(x[1]-x[2])))*(exp(-x[2]*(tiem-x[4]))-exp(-x[1]*(tiem-x[4])))
return(cp)
}

## Función que simula la concentración Y(t).
concentracion2=function(x,tiem){
cp=((100*x[1])/(x[3]*(x[1]-x[2])))*(exp(-x[2]*(tiem-x[4]))-
exp(-x[1]*(tiem-x[4]))) + rnorm(1,0,0.111)
return(cp)
}

## Función de verosimilitud para estimar los parámetros del modelo.
densi=function(x,C){
media=concentracion(x[1:4],tiem)
densi=dnorm(C, media,x[5], log=FALSE)
densi=sum(log(densi))
produ= -prod(densi)
return(produ)
}

## Diseño a evaluar. En este caso para un diseño de seis puntos.
tiem=c(11.42437, 249.1835, 47.53509, 17.03621, 30.30816, 93.64219)

```

```

y1=replicate(100,concentracion2(x[i,],tiem))

error=function(y){
library(DEoptim)
a=x[i,1]
b=x[i,2]
c=x[i,3]
d=x[i,4]
inf=c(a-0.0001,b-0.0001,c-3,d-2,0.0001)
sup=c(a+0.0205,b+0.0095,c+3,d+2,0.0124)
optimo=DEoptim(densi,inf,sup,C=y,control=DEoptim.control(trace = FALSE))
parametro=optimo$optim$bestmem
error=(x[i,]-parametro[1:4])^2
resul=c(error,parametro[5])
return(resul)
}

error.cuadra=parApply(c1,y1,2,error)
result=parApply(c1,error.cuadra,1,mean)
return(result)
}

ECM=sapply(1:1000,error.cuadratico.medio,x,c1)
stopCluster(c1)
resul=ECM
write(resul, file = "resul5")

#####
## ECM para evaluar los diseños###
## Modelo de dos compartimento###
#####
library(snow)
library(nws)
require(MASS)
require(mvtnorm)
library(DEoptim)
##### Derivadas #####

derik12<-function(t,k10,k12,k21,Vc){
k5<- k10+k12+k21
k6<- 1/4*(k5^2-4*k21*k10)^(-1/2)
alfa<- (1/2)*(k12+k21+k10+sqrt(k5^2-4*k21*k10))
beta<- (1/2)*(k12+k21+k10-sqrt(k5^2-4*k21*k10))
deri=(20/Vc)*((alfa-beta)*((alfa-k21)*exp(-alfa*t)*(1/2+k6*2*k5)+
exp(-alfa*t)*(1/2+k6*2*k5)+
(k21-beta)*exp(-beta*t)*(1/2-k6*2*k5)+
exp(-beta*t)*(-(1/2-k6*2*k5)))-((alfa-k21)*exp(-alfa*t)+
(k21-beta)*exp(-beta*t))*
(1/2+k6*2*k5-(1/2-k6*2*k5)))/(alfa-beta)^2
return(deri)
}

```

```

derik21<-function(t,k10,k12,k21,Vc){
k5<- k10+k12+k21
k6<- 1/4*(k5^2-4*k21*k10)^(-1/2)
alfa<- (1/2)*(k12+k21+k10+sqrt(k5^2-4*k21*k10))
beta<- (1/2)*(k12+k21+k10-sqrt(k5^2-4*k21*k10))
deri=(20/Vc)*((alfa-beta)*((alfa-k21)*exp(-alfa*t)*
(1/2+k6*(2*k5-4*k10))+
exp(-alfa*t)*(1/2+k6*(2*k5-4*k10)-1)+
(k21-beta)*exp(-beta*t)*(1/2-k6*(2*k5-4*k10))+
exp(-beta*t)*(1-(1/2-k6*(2*k5-4*k10))))-
((alfa-k21)*exp(-alfa*t)+(k21-beta)*
exp(-beta*t))*(1/2+k6*(2*k5-4*k10)-
(1/2-k6*(2*k5-4*k10))))/(alfa-beta)^2
return(deri)
}

derik10<-function(t,k10,k12,k21,Vc){
k5<- k10+k12+k21
k6<- 1/4*(k5^2-4*k21*k10)^(-1/2)
alfa<- (1/2)*(k12+k21+k10+sqrt(k5^2-4*k21*k10))
beta<- (1/2)*(k12+k21+k10-sqrt(k5^2-4*k21*k10))
deri=(20/Vc)*((alfa-beta)*((alfa-k21)*exp(-alfa*t)*
(1/2+k6*(2*k5-4*k21))+exp(-alfa*t)*(1/2+k6*(2*k5-4*k21))+
(k21-beta)*exp(-beta*t)*(1/2+k6*(2*k5-4*k21))+
exp(-beta*t)*(-(1/2-k6*(2*k5-4*k21))))-
((alfa-k21)*exp(-alfa*t)+(k21-beta)*exp(-beta*t))*
(1/2+k6*(2*k5-4*k21)-(1/2-k6*(2*k5-4*k21))))/(alfa-beta)^2
return(deri)
}

deriVc<-function(t,k10,k12,k21,Vc){
k5<- k10+k12+k21
k6<- 1/4*(k5^2-4*k21*k10)^(-1/2)
alfa<- (1/2)*(k12+k21+k10+sqrt(k5^2-4*k21*k10))
beta<- (1/2)*(k12+k21+k10-sqrt(k5^2-4*k21*k10))
deri= -20*((alfa-k21)*exp(-alfa*t)+(k21-beta)*
exp(-beta*t))/(Vc^2*(alfa-beta))
return(deri)
}

ftf=function(tiem,k10,k12,k21,Vc){
mat.varian=c(derik12(tiem,k10,k12,k21,Vc),derik21(tiem,k10,k12,k21,Vc),
derik10(tiem,k10,k12,k21,Vc),deriVc(tiem,k10,k12,k21,Vc))
ma=mat.varian%*%t(mat.varian)
return(ma)
}
##Funci?n que retorna la matriz de informaci?n.
matriz.informa=function(x,tiem){
y<-array(sapply(tiem,ftf,x[1],x[2],x[3],x[4]),c(4,4,length(tiem)))

```

```

ma=apply(y,c(1,2),sum)
ma=ginv(ma*length(tiem)+diag(1/c(0.01690,0.000961,0.0025,0.0034)))
return(ma)
}
#####
##Diseño a evaluar
tiem=c(0, 0.70485)
##Matriz de varianzas de la distribución posterior
varianza=matriz.informa(c(1.1, 0.11,0.28,0.29),tiem)
##Media de la distribución posterior
media=c(1.1, 0.11,0.28,0.29)
#####
## Parámetros simulados (parámetros fijos o verdaderos) de la distribución posterior.
x=rmvnorm(1000,media,varianza,method=c("svd"))

### Se paraleliza el proceso con seis procesadores.
cl <-makeCluster(c("localhost","localhost","localhost",
"localhost","localhost","localhost"),type = "SOCK")
error.cuadratico.medio=function(i,x,cl){

concentracion=function(x,tiem){
cp=((100*x[1])/(x[3]*(x[1]-x[2])))*(exp(-x[2]*(tiem-x[4]))-exp(-x[1]*(tiem-x[4])))
return(cp)
}

concentracion=function(x,tiem){
k10=x[1]
k12=x[2]
k21=x[3]
Vc=x[4]
k5<- k10+k12+k21
k6<- 1/4*(k5^2-4*k21*k10)^(-1/2)
alfa<- (1/2)*(k12+k21+k10+sqrt(k5^2-4*k21*k10))
beta<- (1/2)*(k12+k21+k10-sqrt(k5^2-4*k21*k10))
A=(20/Vc)*(alfa-k21)/(alfa-beta)
B=(20/Vc)*(beta-k21)/(beta-alfa)
cp=A*exp(-alfa*tiem)+B*exp(-beta*tiem)
return(cp)
}

concentracion2=function(x,tiem){
k10=x[1]
k12=x[2]
k21=x[3]
Vc=x[4]
k5<- k10+k12+k21
k6<- 1/4*(k5^2-4*k21*k10)^(-1/2)
alfa<- (1/2)*(k12+k21+k10+sqrt(k5^2-4*k21*k10))
beta<- (1/2)*(k12+k21+k10-sqrt(k5^2-4*k21*k10))
A=(20/Vc)*(alfa-k21)/(alfa-beta)
B=(20/Vc)*(beta-k21)/(beta-alfa)

```

```

cp=A*exp(-alfa*tiem)+B*exp(-beta*tiem)+rnorm(1,0,0.111)
return(cp)
}

```

```

densi=function(x,C){
media=concentracion(x[1:4],tiem)
densi=dnorm(C, media,x[5], log=FALSE)
densi=sum(log(densi))
produ=-prod(densi)
produ=ifelse(is.nan(produ),0,produ)
return(produ)
}

```

```

tiem=c(0, 0.70485)
y1=replicate(100,concentracion2(x[i,],tiem))

```

```

error=function(y){
library(DEoptim)
a=x[i,1]
b=x[i,2]
c=x[i,3]
d=x[i,4]
media=c(1.1, 0.11,0.28,0.29)
inf=c(a-0.4,b-0.06,c-0.08,d-0.09,0.0012)
sup=c(a+0.4,b+0.06,c+0.08,d+0.09,0.0124)
optimo=DEoptim(densi,inf,sup,C=y,control=DEoptim.control(trace = FALSE))
parametro=optimo$optim$bestmem
error=(x[i,]-parametro[1:4])^2
resul=c(error,parametro[5])
return(resul)
}

```

```

error.cuadra=parApply(c1,y1,2,error)
result=parApply(c1,error.cuadra,1,mean)
return(result)
}

```

```

ECM=sapply(1:1000,error.cuadratico.medio,x,c1)
stopCluster(c1)
resul=ECM
write(resul, file = "resul1")

```

```

#####
##### Gráficos para los diseños óptimos de dos puntos #####
#####

```

```

#### Diseños de dos puntos ####
### Diseños Escenario 1 (Incorrelación entre k1 y k2)
dise21=c(2.62515, 4.72367)
y21=c(0,0)/100

```

```

### Diseños Escenario 2 (correlación al 10% entre k1 y k2)
dise22=c(4.70397, 2.63642)
y22=c(10,10)/100
### Diseños Escenario 3 (correlación al 30% entre k1 y k2)
dise23=c(2.61826, 5.03484)
y23=c(30,30)/100
### Diseños Escenario 4 (correlación al 50% entre k1 y k2)
dise24=c( 4.70397, 2.63642)
y24=c(50,50)/100
### Diseños Escenario 5 (correlación al 70% entre k1 y k2)
dise25=c(4.98744, 2.53984)
y25=c(70,70)/100
### Diseños Escenario 6 (correlación al 90% entre k1 y k2)
dise26=c(4.67259, 2.61482)
y26=c(90,90)/100
#####
### Diseños Escenario 1 (Incorrelación entre k1 y k2)
dise21R= -c(0, 0.70485)
y21R=c(0,0)
### Diseños Escenario 2 (correlación al 10% entre k1 y k2)
dise22R= -c(0, 0.70965)
y22R= c(10,10)/100
### Diseños Escenario 3 (correlación al 30% entre k1 y k2)
dise23R= -c(0, 0.70759)
y23R=c(30,30)/100
### Diseños Escenario 4 (correlación al 50% entre k1 y k2)
dise24R= -c( 0, 0.71225)
y24R=c(50,50)/100
### Diseños Escenario 5 (correlación al 70% entre k1 y k2)
dise25R= -c( 0, 0.71537)
y25R=c(70,70)/100
### Diseños Escenario 6 (correlación al 90% entre k1 y k2)
dise26R= -c(0,0.71789)
y26R=c(90,90)/100

groups <- c("0", "0.1", "0.3", "0.5",
           "0.7", "0.9")

par(mar=c(0.5, 5, 0.5, 1))
#mar=c(bottom, left, top, right)c(bottom, left, top, right)
plot.new()
plot.window(xlim=c(-8, 8), ylim=c(-0.3, 1.15))
ticks <- seq(-8, 8, 1)
y <- c(0,0.1,0.3,0.5,0.7,0.9)
h <- 0.2
lines(rep(0, 2), c(-0.05, 1.15), col="grey")
segments(-8, y, 8, y, lty="dotted")
mtext(groups, at=y, adj=1, side=2, las=2)
par(cex.axis=0.5, mex=0.5)
axis(1, at=ticks, labels=abs(ticks), pos=-0.05)

tw <- 1.4*strwidth("Criterio con R")
#rect(-tw, -0.15, 0, -0.25, col="dark grey")

```

```
#rect(0, -0.25, tw, -0.15, col="light grey")
rect(-tw, 1, 0, 1.15, col=gray.colors(1, start = 0.3))
rect(0, 1, tw, 1.15, col=gray.colors(1, start = 0.6))
text(0, 1.08, "Criterio con R", pos=2,cex=1.2)
text(0, 1.08, "Criterio sin R", pos=4,cex=1.2)
text(0, -0.14, "Tiempo (Horas)")
box("inner", col="grey")
mtext(expression(rho[(list(k[10],k[12], k[21]))])),2,cex=1.2,line=-1.8)

points(dise21,y21,pch=24,cex=1.4,col=gray.colors(1, start = 0)
,bg=gray.colors(1, start = 0.6))
points(dise22,y22,pch=24,cex=1.4,col=gray.colors(1, start = 0)
,bg=gray.colors(1, start = 0.6))
points(dise23,y23,pch=24,cex=1.4,col=gray.colors(1, start = 0)
,bg=gray.colors(1, start = 0.6))
points(dise24,y24,pch=24,cex=1.4,col=gray.colors(1, start = 0)
,bg=gray.colors(1, start = 0.6))
points(dise25,y25,pch=24,cex=1.4,col=gray.colors(1, start = 0)
,bg=gray.colors(1, start = 0.6))
points(dise26,y26,pch=24,cex=1.4,col=gray.colors(1, start = 0)
,bg=gray.colors(1, start = 0.6))
## Puntos del criterio con información a priori ##
points(dise21R,y21R,pch=24,cex=1.4,col=gray.colors(1, start = 0)
,bg=gray.colors(1, start = 0.3))
points(dise22R,y22R,pch=24,cex=1.4,col=gray.colors(1, start = 0)
,bg=gray.colors(1, start = 0.3))
points(dise23R,y23R,pch=24,cex=1.4,col=gray.colors(1, start = 0)
,bg=gray.colors(1, start = 0.3))
points(dise24R,y24R,pch=24,cex=1.4,col=gray.colors(1, start = 0)
,bg=gray.colors(1, start = 0.3))
points(dise25R,y25R,pch=24,cex=1.4,col=gray.colors(1, start = 0)
,bg=gray.colors(1, start = 0.3))
points(dise26R,y26R,pch=24,cex=1.4,col=gray.colors(1, start = 0)
,bg=gray.colors(1, start = 0.3))
```

Bibliografía

- Backhouse, P. G., A. F. Fotheringham and G. Allan: A comparison of a genetic algorithm with an experimental design technique in the optimization of a production process. *The Journal of the Operational Research Society*, **48** 1997, Nr. 3, 247–254
- Bates, D. and D. Watts: *Nonlinear Regression Analysis and its Applications*. Volumen I, New York: Wiley, 1988
- Berger, J: *Statistical Decision Theory and Bayesian Analysis*. Volumen I, New York: Springer, 1985
- Bernardo, J.M: Expected Information as Expected Utility. *The Annals of Statistics*, **7** January 1979, Nr. 3, 686–690
- Beyer, H.G and H.P Schwefel: Evolution strategies A comprehensive introduction. *Natural Computing*, 2002, 3–52
- Brabazon, A: A hybrid genetic model for the prediction of corporate failure. *Computational Management Science*, **1** 2004, 293–310
- Brabazon, A. and Keenan: A hybrid genetic model for the prediction of corporate failure. *Computational Management Science*, **1** 2004, 293–310
- Chaloner, K. and I. Verdinelli: Bayesian Experimental Design: A review. *Statistical Science*, **10** 1995, 379–423
- Davis, J. L. *et al.*: Pharmacokinetics of etodolac in the horse following oral and intravenous administration. *vet. Pharmacol*, **30** 2007, 43–48
- Gabrielsson, J and D Weiner: *Pharmacokinetic and Pharmacodynamic Data Analysis*. Swedish Pharmaceutical Press, 2000
- Gill, Jeff: *Bayesian Methods*. Chapman and Hall, 2000
- Hamada, M. *et al.*: Finding nearoptimal bayesian experimental designs via genetic algorithms. *The American Statistician*, **55** 2001, Nr. 3, 175–181
- Jamshidian, Mortaza. and James. R. Schott: Testing equality of covariance matrices when data are incomplete. *Computational Statistics and Data Analysis*, **51** 2007, 4227–4239
- Lindley, D.V: *Bayesian Statistic-A review*. Philadelphia: SIAM, 1972
- López, V. and R. Ramos: Introducción a los diseños óptimos. *Revista Colombiana de Estadística*, **30** 1997, Nr. 1, 37–51
- Malgor and Valcesia: <http://med.unne.edu.ar/catedras/farmacologia/temasfarma/volumen1/cap2-farmacocinet.pdf>. 2006
- Merlé, Y. and F. Mentré: Stochastic Optimization Algorithms of a Bayesian Design Criterion for Bayesian Parameter Estimation of Nonlinear Regression Models: Application in Pharmacokinetics. *Mathematical Biosciences*, **144** 1997, 45–77

-
- Shannon, C.E: A Mathematical Theory of Communication. Bell System Technical Journal, **27** 1948, Nr. 3, 379–423
- Spezzaferri, F: Nonsequential Designs for Model Discrimination and Parameter Estimation. Bayesian Statistics, **3** 1989, 777–783
- Stewardson, D. J. and R. I. Whitfield: A demonstration of the utility of fractional experimental design for finding optimal genetic algorithm parameter settings. The Journal of the Operational Research Society, **55** 2004, Nr. 2, 132–138
- Tierney, L., A. J. Rossini and Na. Li: Simple Parallel Statistical Computing in R. www.cs.uiowa.edu/~luke/talks/uiowa03.pdf 2003