UNIVERSIDAD DE VALLADOLID

Facultad de Ciencias (Valladolid)

Grado en estadística

Modelos dinámicos para el estudio de la transmisión del SARS-CoV-2 (COVID-19) en Castilla y León

Autor:
Marco de Benito Fernández

Tutores:
Paula Gordaliza Pastor
Hristo Inouzhe Valdes

Agradecimientos

A Paula Gordaliza Pastor y Hristo Inouzhe Valdes, profesores asociados del departamento de Estadística e Investigación Operativa, y tutores de este trabajo fin de grado, por su paciencia y apoyo en la realización de la memoria del proyecto y los experimentos.

También me gustaría agradecerle a mi familia el apoyo recibido durante toda mi etapa universitaria.

Resumen

La pandemia de Covid-19 sufrida en 2019 ha sido una de las enfermedades más letales de la historia. debido a esto, se ha impulsado la investigación en ámbitos muy diversos de la ciencia para poder entender y hacer frente a esta enfermedad. En particular, uno de los aspectos fundamentales es su elevada capacidad de contagio y transmisión, por lo que el desarrollo de métodos de simulación de pandemias ha recibido mucha atención en los últimos dos años. Este trabajo apoyado en sistemas de muestreo basados en cadenas de Markov realiza un estudio de varios tipos de modelos compartimentales SIR que puedan recrear el avance del Covid-19 en la provincia de Castilla y León durante la primera ola.

Índice general

Índice de figuras

Índice de tablas

Introducción

En esta introducción, como es preceptivo, exponemos de forma concisa el planteamiento del problema, antecedentes, objetivos, y la secuenciación seguida en este trabajo.

Planteamiento

La reciente pandemia de SARS-CoV-2 ha sido uno de los sucesos más condicionantes en la sociedad de estos últimos años. Por lo tanto, se ha despertado mucho interés en la investigación de métodos de simulación y observación del desarrollo de epidemias.

Uno de estos métodos es el llamado modelo SIR, un modelo basado en compartimentos capaz de mostrar las principales etapas de los brotes epidémicos. El modelo está nombrado tras sus compartimentos S, I, y R, que denotan lo siguiente:

- S (Susceptibles): el número de individuos de una población de tamaño N que son capaces de contraer la enfermedad estudiada.
- I (Infectados): el número de individuos de una población de tamaño N que están afectados por la enfermedad estudiada.
- R (Recuperados): el número de individuos de una población de tamaño N que han pasado la enfermedad o que ya no forman parte de la población estudiada por cualquier motivo.

El modelo SIR consigue modelizar las pandemias transfiriendo individuos entre compartimentos durante un tiempo t. Un ejemplo del modelo resultante viene presentado en la figura 1

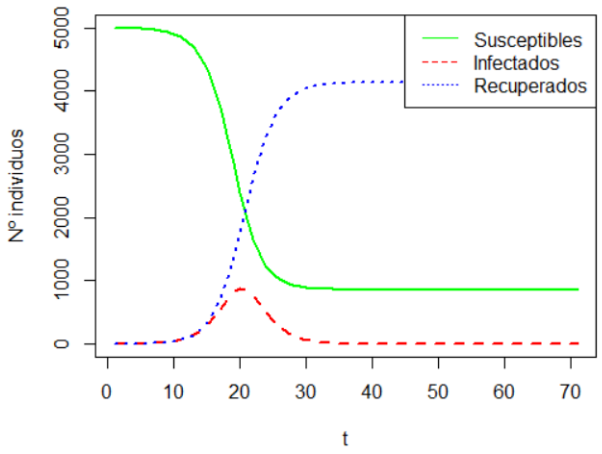


Figura 1: Ejemplo de modelo SIR.

De esta manera podemos observar etapas de crecimiento, decrecimiento, y pico de infectados. Adentrándonos más en el funcionamiento del modelo SIR, podemos indicar que el comportamiento del modelo depende de dos parámetros que regulan el ratio de contagio, y el ratio de recuperación respectivamente.

Para representar correctamente el desarrollo de una enfermedad necesitamos que los parámetros estén bien elegidos. Y como normalmente el valor adecuado se desconoce, es necesario utilizar algún sistema de muestreo que permita probar varias configuraciones y poder generar una distribución a posteriori a partir de las que generen mejores resultados.

Para esto utilizaremos algunos de los métodos que componen las *Markov Chain Monte Carlo* (o Monte Carlo de cadenas de Markov). El Monte Carlo, por ejemplo, es uno de los más generalizados y debería poder generar unos resultados aceptables.

Una propuesta más interesante sería hacer uso de la mecánica Hamiltoniana. Usada en el campo de la física y matemáticas para simulaciones de campos gravitacionales o trayectorias de péndulos, la mecánica Hamiltoniana utiliza conceptos que pueden ser retocados para emplearse como otro sistema de muestreo.

Este trabajo pretende estudiar el comportamiento del algunas epidemias por medio de modelos SIR y varios métodos de muestreo, siendo necesario explicar a su vez los principales conceptos en los que se basan los mismos para facilitar una comprensión de su funcionamiento.

Antecedentes

El estudio de modelos epidemiológicos se remonta a mediados del siglo XIX, donde varios investigadores intentaron generalizar el comportamiento de epidemias en forma de modelos matemáticos. Aun así, el nacimiento de los modelos SIR no es hasta mediados del siglo XX cuando finalmente se propusieron los modelos explícitos [1, 2].

Más actualmente existen situaciones durante la pandemia de Covid-19 en las que se ha usado el modelo SIR. En La Habana y Santa Marta, ciudades de Cuba [3] y Colombia [4], se han llevado a cabo estudios con la finalidad de predecir los casos de infectados futuros y combatir la enfermedad.

Por otro lado, los métodos de Monte Carlo de cadenas de Markov son métodos que se utilizan para realizar estimaciones de distribuciones de las que no se pueden extraer muestras independientes entre sí fácilmente.

Éstos se utilizan constantemente ya que son una herramienta muy útil para técnicas de simulación. Se desarrollaron tras la segunda guerra mundial y fueron clave en la creación de la bomba de hidrógeno. Más recientemente han recibido mucha atención debido a varios métodos nuevos como los filtros de partículas o el salto reversible.

Objetivos

El objetivo principal de este trabajo será el estudio y desarrollo de modelos de tipo SIR generados a partir de parámetros estimados por medio de métodos de Monte Carlo de cadenas de Markov. Esto supone realizar un análisis previo acerca del funcionamiento de los modelos SIR como los Monte Carlo de cadenas de Markov (MCMC) que se vayan a utilizar.

Durante el aprendizaje de los modelos de muestreo se realizarán pruebas sencillas para corroborar que los conceptos están bien entendidos y aplicados.

El estudio del modelo SIR requerirá examinar varios modelos ejemplo con diferentes parámetros para ver como afectan a los resultados. Además, investigaremos cuales son las causas por las que comienza y acaba una epidemia.

A continuación indicaremos el proceso específico que tendremos que seguir para realizar las pruebas sobre los conjuntos de datos. Y por último, realizaremos las pruebas con los conjuntos de datos y compararemos los métodos utilizados según los resultados que produzcan.

Estructura

Para concluir esta introducción haremos un breve comentario sobre el contenido de los capítulos que componen el trabajo.

En el primer capítulo se describen las características principales de las cadenas de Markov, y se introducen los marcos teóricos del Monte Carlo de cadenas de Markov y Monte Carlo Hamiltoniano acompañado de algún ejemplo que ilustre su uso.

En el segundo capítulo se presenta el modelo SIR que se utilizará en las pruebas, se especifica el sistema de ecuaciones diferenciales ordinarias que lo define, y se investigan algunos aspectos que se pueden derivar a partir del modelo como causa de inicio, fin de epidemia, o estimadores de la severidad.

En el tercer capítulo realizamos un análisis de los datos, especificamos el proceso que tendremos que seguir en las pruebas, y se presentan los resultados obtenidos al aplicar los métodos acompañado de un comentario acerca de la calidad de las simulaciones y comparación de modelos.

Por último se presenta un apartado en el que se exponen los resultados más relevantes derivados de este trabajo, y se presentan algunas posibles líneas de actuación futuras.

Capítulo 1

Monte Carlo de cadenas de Markov y Monte Carlo Hamiltoniano

Los orígenes del método Monte Carlo se remontan a 1949, pero para entender qué lo hizo posible, hay que situarse en 1945, en la escuela de ingeniería eléctrica de Moore en la Universidad de Pensilvania. Fue allí donde Stanislaw Ulam, un apasionado de los procesos aleatorios, visitó lo que puede considerarse como el primer computador electrónico, el ENIAC. Por aquella época los métodos estadísticos de muestreo habían caído en desuso debido principalmente al largo y tedioso proceso asociado. Pero con el desarrollo del ENIAC, Stanislaw vio potencial en ellos [5].

En el informe técnico que realizan Stanislaw Ulam y Nicholas Metropolis en 1949 [6], se introduce el problema de cómo estimar la probabilidad de victoria en un juego de solitario. Se indica que atacar el proceso de manera directa es una tarea intratable, y que la manera práctica de resolverlo es producir un gran número de ejemplos del juego y examinar el ratio de éxito.

Como ejemplo genérico, dicho informe presenta lo siguiente: suponer un medio en el que partículas nucleares son capaces de generar más partículas con diferentes energías, por simplicidad, todas las partículas comparten el resto de propiedades, y su capacidad de creación depende de su energía.

El comportamiento de dicho sistema viene formulado por unas ecuaciones diferenciales parciales denominadas, en el campo de la estadística, como ecuaciones de Fokker-Planck, que toman una forma similar a esta:

$$\frac{\partial u(x,y,z)}{\partial t} = a(x,y,z)\Delta u + b(x,y,z)u(x,y,z).$$

Donde u representa la densidad de las partículas, $a\Delta u$ representa la difusión de las partículas, y bu la multiplicación de las partículas.

En estos casos, donde tratar con ecuaciones clásicas resulta una tarea ardua por no decir

casi imposible, se justifica el empleo de métodos estadísticos de simulación más comúnmente denominados como métodos de Monte Carlo [6].

Como resumen podemos decir que los métodos Monte Carlo son un conjunto de métodos que se basan en la ley de los grandes números y en la aleatorización para estimar cierta probabilidad. Suelen seguir el siguiente esquema:

- Definir un dominio de posibles entradas.
- Generar entradas para el dominio.
- Analizar las entradas con respecto al dominio.

Uno de estos métodos es el denominado Monte Carlo de cadenas de Markov (MCMC), pero antes de adentrarnos en él, introduciremos algunos aspectos preliminares sobre las cadenas de Markov.

1.1. Teoría de Cadenas de Markov

Las cadenas de Markov o procesos de Markov, investigadas por Andrey Markov a principios del siglo XX [7], son un proceso estocástico utilizado en análisis de sistemas complejos. Los dos conceptos principales de un proceso de Markov son los estados y las transiciones. Un estado queda especificado cuando se conocen todas las variables que lo componen, por ejemplo, un estado químico puede especificarse por una serie de variables como temperatura, presión, y volumen.

Las transiciones hacen referencia a la naturaleza dinámica del sistema. Es decir, el estado químico puede cambiar si se aplica calor o se aumenta la presión. Estos cambios de estado se llaman "transiciones de estado" [8].

Las cadenas de Markov se caracterizan por ser procesos estocásticos en los que la probabilidad de transición a un estado depende tan solo de su estado actual, de forma matemática se expresa como:

$$P(X_t = x_t | X_0 = x_0, X_1 = x_1, ..., X_{t-1} = x_{t-1}) = P(X_t = x_t | X_{t-1} = x_{t-1}). \tag{1.1}$$

Donde $x_i \in \Omega$ y Ω es el espacio de estados. Existe una probabilidad de transición de un estado a otro que viene representada por la siguiente probabilidad condicionada:

$$p(i,j) = P(X_t = x_j | X_{t-1} = x_i) \quad i,j \in \{1, ..., N\}. \tag{1.2}$$

Adicionalmente las cadenas de Markov pueden poseer las siguientes propiedades [9, 10]:

1. Un estado es recurrente si el proceso eventualmente regresará al estado:

$$P(X_n = x_i, | X_0 = x_i) = 1 \quad n \geq 1. \tag{1.3}$$

2. Una cadena es irreducible si es posible ir de un estado a cualquier otro en un número finito de pasos:

$$P(X_n = x_j | X_0 = x_i) = 1 \quad \forall i, j \in \{1, ..., N\},\ n < \infty. \tag{1.4}$$

3. Formando una matriz NxN a partir de las probabilidades de la *Propiedad 2* se obtiene una matriz de transiciones P con todas las entradas no negativas:

$$\sum_{j=1}^{N} p(i, j) = 1, \quad \forall i \in \{1, ..., N\}. \tag{1.5}$$

4. La probabilidad de transición de x_i a x_j en n pasos viene dada por:

$$p_n(i, j) = P^n_{i,j}. \tag{1.6}$$

Donde $P^n_{i,j}$ es la entrada i,j de la matriz de transición P^n. Esto puede probarse con la ley de la probabilidad total explicada a continuación:

$$p_n(i, j) = P(X_n = x_j | X_0 = x_i) =$$

$$\textstyle\sum_k P(X_{n-1} = x_k | X_0 = x_i) P(X_n = x_j | X_{n-1} = x_k) = \tag{1.7}$$

$$\textstyle\sum_k p_{n-1}(i, k) p(k, j).$$

Por lo tanto la matriz de transición P^n queda definida por $P^{n-1}P$.

5. Una cadena de Markov es estacionaria si para cada entero positivo k la distribución de la l-tupla no depende de n.

$$(X_{n+1}, X_{n+2}, ..., X_{n+l}).$$

Dado un estado inicial ϕ_0 y la matriz de transición P podríamos obtener su distribución estacionaria:

$$\lim_{n\to\infty} \phi_0 P^n = \pi. \tag{1.8}$$

π se considera estacionaria si $\pi P = \pi$.

6. Una cadena de Markov es reversible si su probabilidad de transición es reversible respecto a su distribución inicial, con $\pi()$ como la distribución inicial esto viene representado por

$$\pi(i) p_{i,j} = \pi(j) p_{j,i}. \tag{1.9}$$

Una vez expuestas las propiedades principales que componen las cadenas de Markov, y con el fin de poder entender la teoría de MCMC, vamos a repasar brevemente el método de Monte Carlo original (aunque lo hemos presentado para procesos discretos, se pueden extender al caso continuo usando las probabilidades condicionadas de manera adecuada).

1.2. Teoría del Monte Carlo

Según se expone en la sección 1.7 de [11] el método Monte Carlo original se puede considerar un caso especial de MCMC en el que $X_1, X_2, ...$ son variables independientes e igualmente distribuidas (i.i.d). En este caso la cadena de Markov es estacionaria y reversible.

Esto puede probarse aplicando las propiedades 5 y 6 (Eq. 1.8 y 1.9) de las cadenas de Markov que acabamos de ver. Para probar la estacionaridad podemos utilizar la distribución de

$$P(X_{n+1} = x_i, X_{n+2} = x_j, ..., X_{n+l} = x_k).$$

Debido a que sabemos que son independientes e identicamente distribuidas wlo podemos representar también como

$$P(X_{n+1} = x_i)P(X_{n+2} = x_j)...P(X_{n+l} = x_k) = P(X = x_i)P(X = x_j)...P(X = x_k)$$

y que por lo tanto al no depender de n se considera estacionaria. Para probar la reversibilidad podemos coger la igualdad $\pi(i)p_{i,j} = \pi(j)p_{j,i}$ y sustituir en ella para ver si se cumple.

Debido a que existe independencia,

$$p_{i,j} = P(X_n = x_j|X_{n-1} = x_i) = P(X_n = x_j) = \pi(j)$$

y por lo tanto

$$\pi(i)\pi(j) = \pi(j)\pi(i)$$

y también quedaría probada la reversibilidad.

Ahora vamos a repasar la forma que toman algunos estadísticos básicos en Monte Carlo. Sea g una función que no es posible calcular por métodos convencionales, sea Y_i=g(X_i), y nuestro objetivo es calcular la esperanza.

$$\mu = E(Y). \tag{1.10}$$

Supongamos además, que es posible sacar muestras de X, por lo que el estimador muestral de la esperanza sería:

$$\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} Y_i, \tag{1.11}$$

donde su varianza viene dada por;

$$\sigma^2 = var(Y). \tag{1.12}$$

Por el teorema central del límite (TCL) tendríamos que $\hat{\mu}$ seguiría una distribución

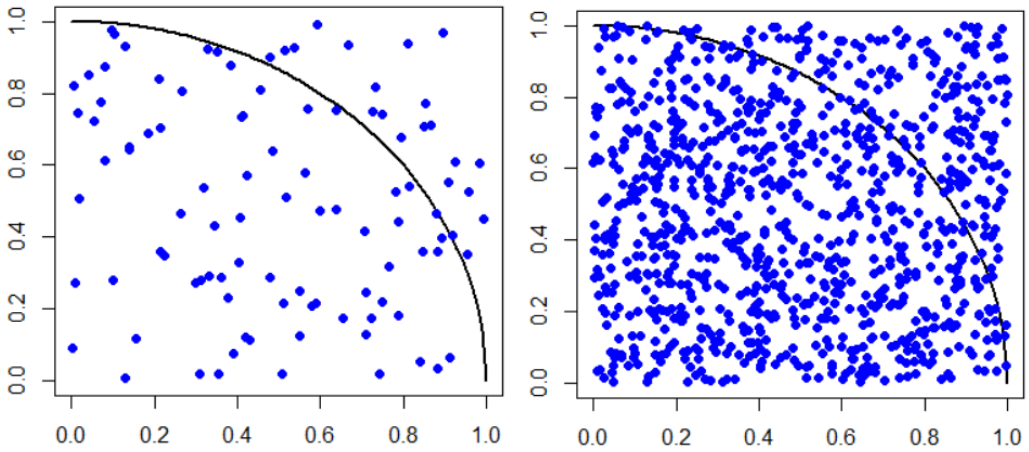
$$\hat{\mu}_n \sim N(\mu, \frac{\sigma^2}{n})$$

y que la varianza de Y puede estimarse mediante

$$\hat{\sigma}_n^2 = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{\mu}_n)^2. \tag{1.13}$$

Como ejemplo estimaremos el valor de π por medio de Monte Carlo.

Para ello, vamos a simular pares (x,y) aleatorios en el primer cuadrante de coordenadas [0,1] ($X, Y \sim U(0,1)$) y los usaremos para estimar el área de un cuarto de círculo. Realizamos 100 y 1000 iteraciones y obtenemos los resultados presentados en la Figura 1.1.



(a) Monte Carlo de 100 iteraciones. (b) Monte Carlo de 1000 iteraciones.

Figura 1.1: Estimación del valor de π mediante el método de Monte Carlo.

Ahora hay que calcular la distancia de cada uno de los puntos al centro de coordenadas y ver cuáles tienen una distancia menor o igual que 1. Obtenemos 81 y 774 respectivamente, lo cual se traduce en áreas de 0.81 y 0.774. Sabemos que el área de un círculo de radio 1 es igual a π, por lo que podemos multiplicar las áreas por 4 y obtener una estimación de π. Esto nos da estimaciones de 3.24 y 3.096 respectivamente.

No se puede decir que estos resultados sean muy exactos, pero los métodos de Monte Carlo mejoran con más repeticiones, así que vamos a realizar un último experimento con 10 millones de observaciones. Se obtiene un área de 0.7854046 que multiplicado por 4 queda 3.141618 y un intervalo de confianza al 95 % de (3.141442, 3.141795).

Aunque el método de Monte Carlo original se basa en estadística elemental, el método de Monte Carlo de cadenas de Markov aplica los mismos conceptos con la dificultad añadida de que no se dispone de independencia entre las variables.

1.3. Monte Carlo de cadenas de Markov

Al igual que el método Monte Carlo original, el Monte Carlo de cadenas de Markov se utiliza para estimar el comportamiento de distribuciones demasiado complejas de estudiar con otros métodos. Este se basa en cadenas de Markov reversibles cuya distribución de equilibrio es la distribución de interés.

Puede ser de interés calcular la esperanza

$$\mu = E(Y),$$

con $Y = g(X)$, y $X_1, X_2, ...$ una cadena de Markov estacionaria con distribución inicial igual a la distribución de X. Debido a la dependencia entre variables, la varianza toma la forma:

$$\sigma^2 = var(Y_i) + 2\sum_{k=1}^{n} cov(Y_i, Y_{i+k}), \tag{1.14}$$

comúnmente expresada como:

$$\sigma^2 = \gamma_0 + 2\sum_{k=1}^{n} \gamma_k, \tag{1.15}$$

donde $\gamma_k = \frac{1}{n}\sum_{i=1}^{n-k}(Y_i - \hat{\mu})(Y_{i+k} - \hat{\mu})$ es la función de autocovarianza.

Toda cadena tiene que ser estacionaria para poder utilizarse en MCMC, pero su obtención no es tan trivial como en el Monte Carlo original. Por lo que tendremos que utilizar la fórmula $\pi = \lim_{n\to\infty} \phi_0 P^n$ para obtenerla.

Ahora que hemos visto un poco de la complejidad que introduce la dependencia de variables, vamos a pasar a explicar MCMC desde un punto de vista práctico [12]. Para ello vamos a basarnos en el teorema de Bayes:

$$P(\theta|x) = \frac{P(\theta)P(x|\theta)}{P(x)},$$

donde $P(\theta)$ es la distribución a priori de los parámetros, $P(x|\theta)$ la distribución de los datos según el priori (función de verosimilitud), y $P(x)$ la probabilidad marginal.

Tanto $P(\theta)$ como $P(x|\theta)$ son fáciles de obtener, pero, para obtener $P(x)$ tenemos que realizar su integral en todos los posibles valores del parámetro θ. Aunque es factible realizarlo con funciones sencillas, en general resulta inviable. Si quisiéramos estimarlo utilizando el método de Monte Carlo original, necesitaríamos generar muestras por Bayes, y como ya hemos dicho, no podemos porque no disponemos de $P(x)$. Es en estos casos en los que se utiliza MCMC.

Supondremos que el modelo sigue una distribución normal de parámetros μ y σ, sabiendo que σ es 1 y queremos calcular una muestra a posteriori de μ. Como distribución a priori vamos a elegir una t de Student con 2 grados de libertad, por lo que

$$\mu \sim t_2,$$

$$x|\mu \sim N(\mu, 1).$$

Lo primero que vamos a hacer es generar unos datos, en este caso 30 observaciones, y elegir un μ inicial, por ejemplo, un valor aleatorio que devuelva la t de stundent.

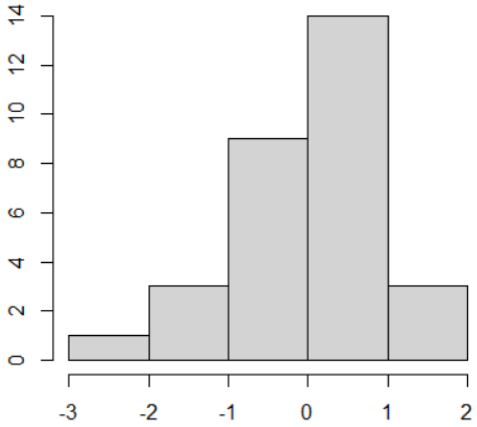


Figura 1.2: Histograma de las 30 observaciones.

Ahora tenemos que simular múltiples valores de μ con la t_2 y proponer un criterio de selección, por ejemplo, el de máxima verosimilitud.

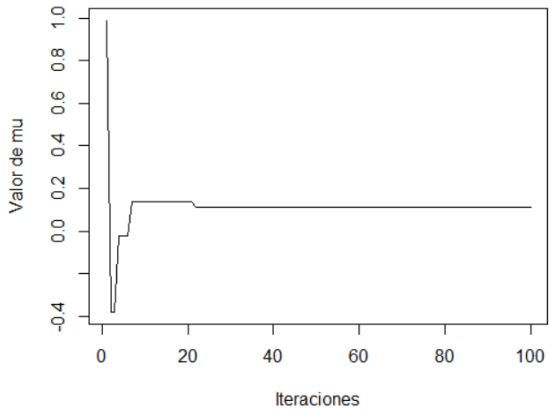


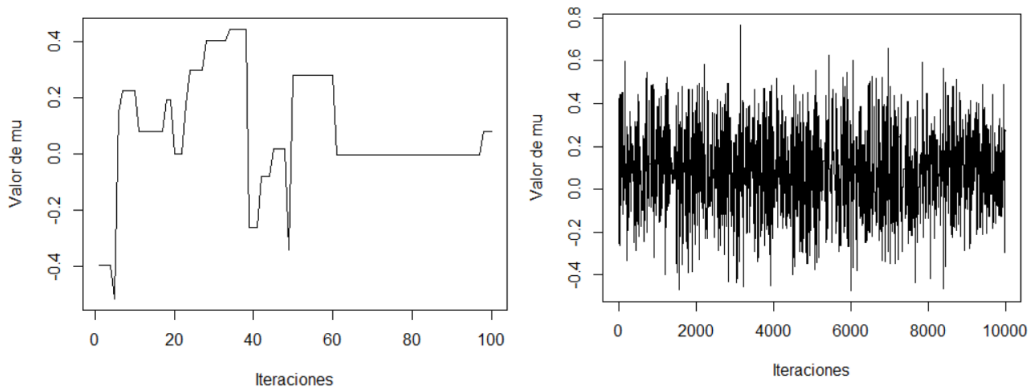
Figura 1.3: Simulación de μ por verosimilitud.

Con suficientes iteraciones esto nos acaba dando el valor con mayor verosimilitud (figura 1.3). Sin embargo, nosotros queremos unas muestra a posteriori que nos informe de la posible varianza, por lo que es posible que a veces tengamos que captar valores de μ que tengan menor verosimilitud. Para ello se propone que $\mu = \mu_{t+1}$ con probabilidad

$$\frac{p(x|\mu_{t+1})p(\mu_{t+1})}{p(x|\mu_t)p(\mu_t)},$$

de esta manera, aunque la verosimilitud actual sea menor que la anterior, hay una probabilidad de que sea elegida. A este criterio se le denomina actualización de Metropolis-Hasting

y será explicado con detalle más adelante. Aplicando esto obtenemos el gráfico de la Figura 1.4 con 100 y 10000 repeticiones.



(a) MCMC con 100 repeticiones. (b) MCMC con 10000 repeticiones.

Figura 1.4: Simulación de posteriori de μ en MCMC.

Al conjunto de valores de μ que se muestran en los gráficos de la figura 1.4 se les conoce como traza. Utilizando la de 10000 muestras y generando el histograma de las μ aceptadas podemos visualizar la distribución posterior de μ representada en la figura 1.5.

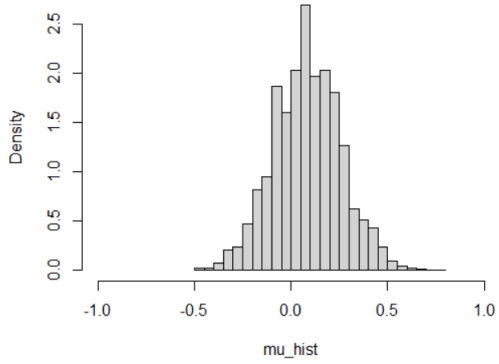


Figura 1.5: Distribución a posteriori de μ.

Aunque en este ejemplo no se ha realizado, se suelen descartar las primeras iteraciones de la MCMC debido a que es posible que el valor a priori utilizado esté lejos del centro de equilibrio (la media de la distribución estacionaria), y por lo tanto las iteraciones hasta llegar a él modifiquen la distribución resultante [11].

Al cambio pseudoaleatorio que se propone en cada iteración se le llama método de actualización y es necesario que su aplicación no modifique la distribución que se pretende simular. Vamos a explorar en concreto los métodos de Metropolis-Hasting y Gibbs, que preservan la estacionaridad de las cadenas de Markov.

Actualización de Metropolis-Hasting

La actualización de Metropolis-Hasting es una generalización propuesta en 1970 por W. K. Hastings que expande la actualización de Metropolis, original de la física estadística, hacia el muestreo general [13, 14]. Ésta propone lo siguiente [9, 11, 15, 16, 17]:

- Sabemos que h representa la densidad de la función de distribución que queremos modelar, y que se integra a un valor finito positivo. Además, representaremos a su probabilidad de transición con $p(x, y)$.
- Suponemos que la cadena se encuentra en el estado x_t, y se propone un nuevo estado y.
- Su siguiente estado será y con probabilidad

$$min\left(1, \frac{h(y)p(y, x_t)}{h(x_t)p(x_t, y)}\right),$$

  y será x_t con probabilidad

$$1 - min\left(1, \frac{h(y)p(y, x_t)}{h(x_t)p(x_t, y)}\right).$$

Es posible que esta fórmula resulte familiar debido a que es la que se ha usado en el ejemplo de la figura 1.2.

La actualización de Metropolis es una especificación de esta fórmula en la que $p(y, x_t) = p(x_t, y)$, por lo que el siguiente estado sería y con probabilidad

$$min\left(1, \frac{h(y)}{h(x_t)}\right).$$

Actualización de Gibbs

El muestreo de Gibbs al igual que otras MCMC sirve para generar variables aleatorias de una distribución. Supongamos que tenemos un parámetro de interés $\theta = (\theta_1, \theta_2, ..., \theta_K)$. La idea principal de Gibbs es usar muestras de las condicionales $P(\theta_i|X, \theta_{-i})$, donde $\theta_{-i} = (\theta_1, ..., \theta_{i-1}, \theta_{i+1}, \theta_K)$ [16, 18, 19, 20, 21],

$$\theta_1^{t+1} \sim P(\theta_1^t|X, \theta_2^t, \theta_3^t, ..., \theta_K^t)$$
$$\theta_2^{t+1} \sim P(\theta_2^t|X, \theta_1^{t+1}, \theta_3^t, ..., \theta_K^t)$$
$$\vdots$$
$$\theta_K^{t+1} \sim P(\theta_K^t|X, \theta_1^{t+1}, \theta_2^{t+1}, ..., \theta_{K-1}^{t+1})$$

Recordemos que $P(\theta|X) = P(\theta_i, \theta_{-i}|X) = P(\theta_i|X, \theta_{-i})P(\theta_{-i})$, y que la probabilidad de transición para una propuesta θ^* puede expresarse como:

$$min\left(1, \frac{P(\theta^*|X)P(\theta_i|X, \theta_{-i})}{P(\theta|X)P(\theta_i^*|X, \theta_{-i}^*)}\right)$$

que es igual a

$$min\left(1, \frac{P(\theta_i^*|X, \theta_{-i}^*)P(\theta_{-i}^*)P(\theta_i|X, \theta_{-i})}{P(\theta_i|X, \theta_{-i})P(\theta_{-i})P(\theta_i^*|X, \theta_{-i}^*)}\right).$$

Y como $\theta_{-i}^* = \theta_{-i}$, la probabilidad de salto siempre es 1 y siempre se acepta la siguiente propuesta.

Tanto Metropolis-Hasting como Gibbs se usan en gran medida en MCMC pero ambos tienen la desventaja de tener un comportamiento *random-walk* o camino aleatorio que puede hacer que su convergencia resulte lenta en distribuciones más complejas. El Monte Carlo Hamiltoniano o HMC son un tipo de MCMC que trata de evitar este comportamiento.

1.4. Teoría del Monte Carlo Hamiltoniano

El germen del Monte Carlo Hamiltoniano nació en 1959 de la mano de Alder y Wainwright [22] como otra opción para tratar la simulación de moléculas. En 1987 fue presentado como Monte Carlo híbrido por Duane [23] y finalmente referido por el nombre actual de Monte Carlo Hamiltoniano en 2003 por MacKay [24].

Este apartado trata los principales aspectos teóricos del Monte Carlo Hamiltoniano. Para un estudio más detallado acerca de las diferencias respecto a los MCMC ya comentados, recomendamos consultar *A Conceptual Introduction to Hamiltonian Monte Carlo* de Michael Betacourt [25].

El hamiltoniano usado en HMC puede escribirse como:

$$H(\theta, \rho) = U(\theta) + K(\rho), \tag{1.16}$$

donde θ representa un vector d-dimensional de posiciones y $U(\theta) = -\log(p(\theta))$ donde $p(\theta)$ es la densidad del posterior $P(\theta)$, ρ es un vector d-dimensional de momentos, y $K(\rho)$ se expresa como:

$$K(\rho) = \frac{1}{2}\rho^T M^{-1}\rho, \tag{1.17}$$

donde M es una matriz que suele ser un múltiplo de la matriz identidad. La fórmula de $K(\rho)$ corresponde a menos el logaritmo de la función de densidad de la distribución gausiana de media 0 con matriz de covarianza M. Llegados a este punto hay que pararse a pensar cómo el Hamiltoniano puede ayudar a obtener $p(\theta)$, que es al fin y al cabo lo que nos interesa.

Lo cierto es que si que es posible relacionar el Hamiltoniano con la distribución posterior de θ [11, 26], por medio de un concepto de mecánica estadística denominado función canónica. La función canónica expresa la función de densidad de un estado con energía, $H(\theta, \rho)$, como

$$p(\theta, \rho) = \frac{1}{Z}e^{-\frac{H(\theta,\rho)}{T}} = \frac{1}{Z}e^{-\frac{U(\theta)}{T}}e^{-\frac{K(\rho)}{T}}. \tag{1.18}$$

T es la temperatura que en nuestro caso va a ser 1, y Z es una constante de normalización llamada función de partición, que escala la función para que integre 1. Z no es importan-

te ya que el MCMC puede generar muestras de distribuciones no escaladas, por tanto, también podemos representar la función como

$$p(\theta, \rho) = e^{-H(\theta,\rho)} = e^{-U(\theta)}e^{-K(\rho)} = p(\theta)p(\rho), \tag{1.19}$$

La presencia de ρ, donde $p(\rho)$ es la densidad de ρ, nos permite aplicar la dinámica Hamiltoniana para hacer las propuestas en el MCMC.

Para saber como cambian θ y ρ a lo largo del tiempo se usan las ecuaciones hamiltonianas

$$\frac{d\theta}{dt} = \frac{\partial H}{\partial \rho} = M^{-1}\rho,$$

$$\frac{d\rho}{dt} = -\frac{\partial H}{\partial \theta} = -\nabla_\theta U(\theta).$$

Las propiedades más importantes que permiten el uso de la mecánica Hamiltoniana en MCMC son las siguientes [11, 27]:

- Conservación del hamiltoniano: $H(\theta(0), \rho(0)) = H(\theta(t), \rho(t))$ para $t \geq 0$, sabiendo que $\frac{\partial H}{\partial t} = 0$ porque H no depende explícitamente del tiempo. Queda probada por la siguiente expresión:

$$\frac{dH}{dt} = \frac{\partial H}{\partial t} + \frac{d\theta}{dt}\frac{\partial H}{\partial \theta} + \frac{d\rho}{dt}\frac{\partial H}{\partial \rho} = \frac{\partial H}{\partial \rho}\frac{\partial H}{\partial \theta} - \frac{\partial H}{\partial \theta}\frac{\partial H}{\partial \rho} = 0$$

- Conservación del volumen: el vector de operación divergencia (∇) representa el flujo de volumen que se percibe desde un punto, y la divergencia de $(\frac{d\theta}{dt}, \frac{d\rho}{dt})$ es:

$$\frac{\partial}{\partial \theta}\frac{d\theta}{dt} + \frac{\partial}{\partial \rho}\frac{d\rho}{dt} = \frac{\partial}{\partial \theta}\frac{\partial H}{\partial \rho} - \frac{\partial}{\partial \rho}\frac{\partial H}{\partial \theta} = 0$$

- Reversibilidad: si hacemos que $\rho = -\rho$ entonces las ecuaciones hamiltonianas serán:

$$\theta = \nabla_{-\rho} H(\theta, \rho) = -\nabla_\rho H(\theta, \rho)$$

$$-\rho = -\nabla_\theta H(\theta, \rho)$$

  Si lo ponemos en función de t queda:

$$\frac{d\theta}{d(-t)} = \nabla_p H(\theta, \rho)$$

$$-\frac{d\rho}{dt} = \frac{d\rho}{d(-t)} = -\nabla_\theta H(\theta, \rho)$$

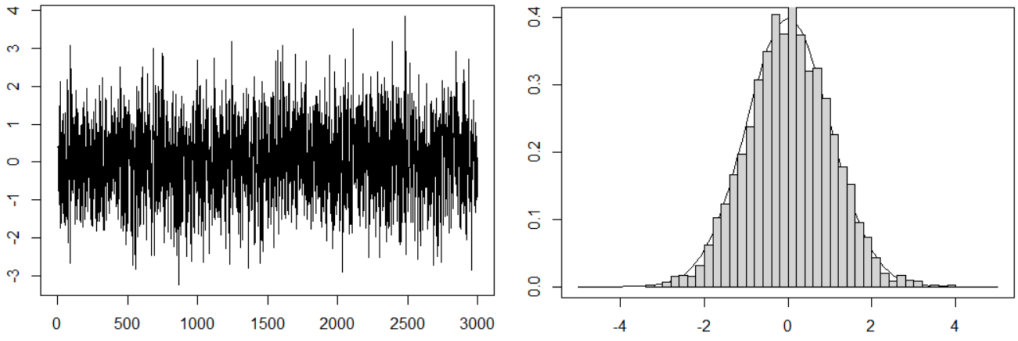
  Y como son las mismas ecuaciones queda probada la reversibilidad.

Con los conceptos principales ya explicados vamos a revisar los pasos que habría que realizar para obtener muestras con HMC, así mismo, se realizará un ejemplo para visualizar las diferencias ya comentadas con los otros métodos.

Una iteración de HMC puede resumirse en 4 pasos [20, 28]:

1. Se comienza actualizando ρ con un valor aleatorio de una $N(0, M)$.
2. A continuación se realizará una actualización simultánea de θ y ρ con ayuda de las ecuaciones hamiltonianas. Para ello típicamente se utilizará el método de salto de rana, el cual resuelve de manera aproximada las ecuaciones hamiltonianas. Se harán L saltos y se utilizará ϵ para escalar cada uno, donde cada salto consiste en lo siguiente:
   - Media actualización del *momento*:
   $$\rho = \rho + \frac{1}{2}\epsilon\frac{d\log(p(\theta|x))}{d\theta}$$
   - Una actualización de la *posición*:
   $$\theta = \theta + \epsilon M^{-1}\rho$$
   - Otra media actalización del *momento*:
   $$\rho = \rho + \frac{1}{2}\epsilon\frac{d\log(p(\theta|x))}{d\theta}$$
3. Con θ^t y ρ^t como los valores antes de realizar el salto de rana y con θ^* y ρ^* los resultantes del método calculamos
   $$r = \frac{p(\theta^*|x)p(\rho^*)}{p(\theta|x)p(\rho)}$$
   o
   $$r = e^{(H(\theta,\rho)-H(\theta^*,\rho^*))}.$$
4. θ^{t+1} será igual a θ^* con probabilidad min(r,1), si no, será igual a θ^t.

Realizando el proceso anterior para una M igual a la matriz identidad y una θ que sigue una distribución normal de media $\mu = 0$ y varianza $\sigma^2 = 1$ (por lo tanto $\frac{d\log(p(\theta|x))}{d\theta} = -\theta$) obtenemos los resultados de la figura 1.6



(a) valores de θ.

(b) Histograma de valores de θ.

Figura 1.6: Simulación de posteriori θ en HMC.

Una vez vistos los procesos que permiten la simulación de distribuciones generales, vamos a pasar a realizar un breve estudio de los modelos compartimentales epidemiológicos, que son el otro pilar en el que se basa este trabajo.

Capítulo 2

Modelos compartimentales epidemiológicos

Actualmente, el estudio del desarrollo de una epidemia es uno de los temas más candentesen la investigación científica debido a la reciente pandemia causada por el SARS-CoV-2, pero su investigación se remonta a hace más de un siglo. Como ejemplos de esos primeros estudios epidemiológicos cabe destacar el realizado por G. H. Evans [29] en el que intenta, sin mucho éxito, formular un modelo general a partir de una epidemia bovina, o varios artículos de J. Brownlee [30, 31] donde se estudian las posibles causas de una epidemia.

Está generalmente aceptado que el origen de los modelos epidemiológicos se remonta al artículo [32] escrito por el matemático Ronald Ross en 1916, y desarrollado más en profundidad por Kermack y McKendrick en 1927 en [33]. En ambos se plantea el mismo problema: "supongamos que tenemos una población de individuos de tamaño N, de los cuales, un número I están infectados y un número S son susceptibles de contraer la enfermedad. Supongamos que existe una proporción h de los individuos de S que se infectan, y una proporción r de los infectados que pasan a un estado R de no infectados ya sea por recuperación o muerte. Supongamos también que existen tasas de natalidad y mortalidad para cada grupo. Entonces, ¿cuál será el número de individuos afectados, nuevos casos, y población en el instante t?".

En 1932 y 1933 Kermack y McKendrick publicaron 2 artículos [1, 2] que incluyen el modelo que daba respuesta a esta pregunta y que es conocido hoy en día como el modelo de Kermack-McKendrick.

El uso de modelos matemáticos en epidemiología es especialmente apropiado debido a que otras opciones como la experimentación a veces no se puedan llevar a cabo.

Por lo general el proceso de modelado está compuesto de los siguientes pasos [34]:

1. Realizar suposiciones acerca del proceso de contagio basadas en el conocimiento de la enfermedad.
2. Diseñar un modelo matemático para el proceso de transmisión basado en las suposiciones.

3. Realizar análisis matemático del modelo e interpretar los resultados obtenidos.
4. Recopilar información disponible de la enfermedad con la que validar el modelo.
5. Mejorar el modelo modificando las suposiciones iniciales.

Dentro de los modelos matemáticos existen distintos tipos entre los que destacamos [34, 35]:

- *Los modelos estadísticos*: se construyen para un conjunto de datos específico, ya que su funcionamiento requieren de grandes cantidades de datos.
- *Los modelos deterministas*: utilizan ecuaciones diferenciales e interpretan que el número de individuos susceptibles e infectados siguen funciones en el tiempo. Son más independientes de los datos que los estadísticos pero pueden dar problemas en poblaciones pequeñas.
- *Los modelos estocásticos*: el proceso de infección se trata como un proceso estocástico, funcionan correctamente en comunidades reducidas, pero resultan complicados de llevar a la práctica debido a la gran cantidad de simulaciones que requieren.

En este capítulo vamos a centrarnos en los modelos compartimentales, que son un caso particular de modelos deterministas. Introduciremos el concepto de compartimentos, y mostraremos cómo su uso puede facilitar la comprensión de la transmisión de una enfermedad.

2.1. Conceptos básicos

Los modelos compartimentales permiten modelar los procesos subyacentes que dictan la transmisión de una enfermedad, esto permite crear estimaciones de la severidad y escala que puede tener una epidemia causada por esa enfermedad. Debido a que los modelos compartimentales, existe cierta necesidad de datos previos con los que poder validar el modelo. El modelo compartimental típico es el modelo SIR.

En la figura 2.1 podemos ver un diagrama de un modelo SIR. Éste está compuesto por los siguientes compartimentos:

- Susceptibles (S): número de individuos que son capaces de contraer la enfermedad en estudio.
- Infecciosos (I): número de individuos portadores de la enfermedad.
- Recuperados (R): número de individuos que han pasado la enfermedad o ya no forman parte del estudio.

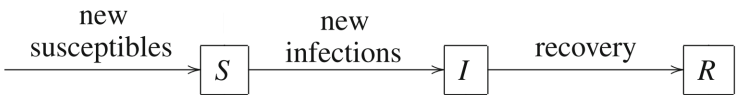


Figura 2.1: Diagrama de transferencia de un modelo SIR.

Fuente: [34].

Como ya se ha comentado este sistema depende del tiempo por lo que es posible que un individuo que está infectado en el instante de tiempo t esté recuperado en el instante $t+1$. Estas posibilidades vienen representadas por las flechas que unen los compartimentos y cada una tiene un significado epidemiológico.

Existen variaciones de este modelo como el modelo SEIR en el que se introduce un compartimento, E, para el periodo de latencia, o modelos SIR que incluyen el factor de la pérdida de inmunidad como otra tasa. En nuestro caso vamos a estudiar el modelo conocido como el modelo de Kermack-McKendrick.

2.2. Modelo de Kermack-McKendrick

El modelo de Kermack-McKendrick, cuyo diagrama se presenta en la figura 2.2, realiza las siguientes suposiciones [34, 36, 37]:

1. La transmisión ocurre por medio de contacto directo.
2. Los individuos en la población están mezclados de forma homogénea. Esto conlleva a que el número de contactos entre miembros de diferentes compartimentos depende del número de individuos de cada compartimento.
3. El ratio de transferencia de un compartimento es proporcional al número de individuos del compartimento.
4. Los individuos que se infectan lo hacen sin periodo de latencia.
5. No hay pérdida de inmunidad.
6. El tamaño de la población se mantiene constante, por lo tanto no se introducen o retiran individuos. En otras palabras no hay natalidad o mortalidad natural.

A primera vista puede parecer un modelo muy restrictivo, pero lo cierto es que resulta bastante apropiado en muchas circunstancias. Más adelante realizaremos un ejemplo de simulación para ganar intuición sobre el modelo SIR.

Para terminar de fijar el modelo tenemos que proponer definiciones para los términos *nuevas infecciones* y *recuperados*.

Las *nuevas infecciones* se representan como el número de infectados por el número de susceptibles por λ donde λ es el coeficiente de transimisión, que representa la capacidad de infección de la enfermedad. Y los *recuperados* se representan como el número de infectados por γ donde γ es el coeficiente de recuperación, que representa el inverso del número de días promedio que tarda un infectado en recuperarse.

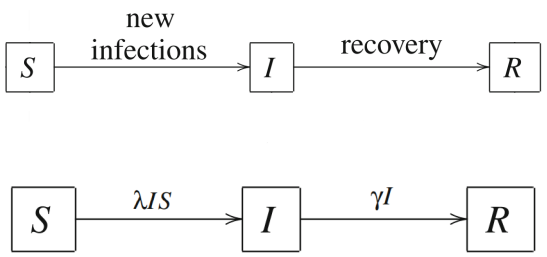


Figura 2.2: Diagrama de transferencia del modelo Kermack-McKendrick.

Fuente: [34].

De esta manera, podemos caracterizar el modelo SIR con diagrama en la figura 2.2, mediante el siguiente sistema de ecuaciones diferenciales ordinarias (ODE):

$$\begin{aligned} \frac{dS}{dt} &= -\lambda IS \leq 0 \\ \frac{dI}{dt} &= \lambda IS - \gamma I \\ \frac{dR}{dt} &= \gamma I \geq 0 \end{aligned} \tag{2.1}$$

También sabemos que al principio de una epidemia no hay recuperados y tiene que haber susceptibles e infectados, por lo tanto:

$$S(0) > 0 \quad I(0) > 0 \quad R(0) = 0.$$

Para visualizar el avance de la transmisión que simula este modelo, vamos a realizar un ejemplo con unos datos publicados en 1978 de un caso de influenza en un internado de Inglaterra [38]. Para este caso, el modelo que acabamos de estudiar resulta especialmente apropiado ya que la población del campus es constante.

El artículo indica que había un total de 763 alumnos, y que tras unas vacaciones, un alumno volvió con fiebre, por lo que podemos considerar que $S_0 = 762$ y $I_0 = 1$, también indica que al siguiente día había 3 estudiantes en la enfermería, por lo tanto $I_1 = 3$. Sabiendo I_1 vamos a aproximar λ de la siguiente manera

$$\frac{dS}{dt} = -\lambda SI \rightarrow \lambda = -\frac{\triangle S}{\triangle tSI} = -\frac{-2}{1 * 762/763} = 2{,}002.$$

Por simplicidad estamos indicando los valores de S, I, y R como absolutos, la razón por la que en la fórmula superior se divide entre 763 (población total), es porque a la hora de hacer los cálculos necesitamos normalizar los datos a rango 0, 1.

Para γ vamos a utilizar el valor 0.5, esto lo podemos aproximar a partir del mismo artículo. En él se indica que el tiempo que tardaban en mandar a los alumnos a la enfermería (al estar en la enfermería ya no pertenecen a los infectados ya que no pueden infectar) era entre 1 y 2 días, así que aproximadamente 0.5 de los infectados se "recuperan" cada día.

Realizando el modelo con estos datos obtenemos la gráfica presentada en la figura 2.3, y como puede verse, los resultados que proporciona son bastante similares a los reales.

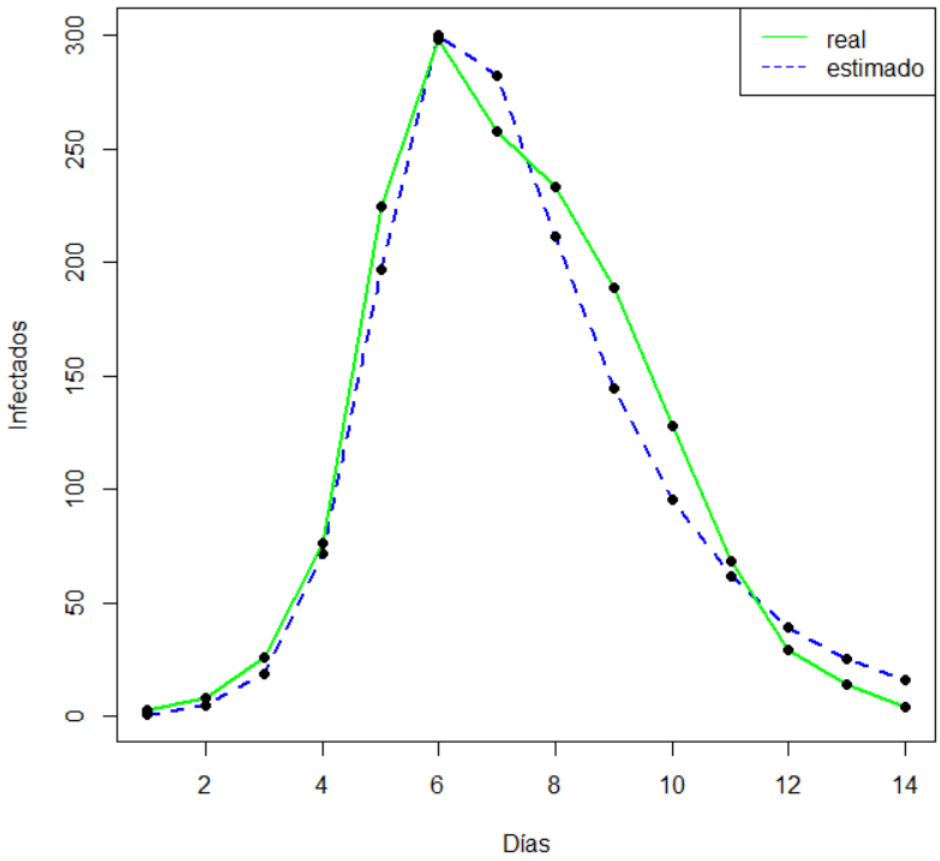


Figura 2.3: Desarrollo del avance de influenza en un internado.

2.2.1. Inicio y fin de una epidemia

Realizar un análisis en las primeras fases de una posible epidemia es clave para poder indicar la gravedad de la misma y prepararse para sus efectos, o para saber si la transmisión va a detenerse antes de llegar a ser un problema.

Los indicadores más comunes para diferenciar entre una epidemia y unos pocos infectados son el *tamaño crítico de la comunidad* y el *número básico de reproducción*. Ambos se

pueden calcular a partir de los valores iniciales que acabamos de fijar y el sistema de ecuaciones diferenciales ordinarias 2.1.

Reescribiendo la derivada de I como $\frac{dI}{dt} = (\lambda S - \gamma)I$ tenemos, por un lado, el caso de que S_0 (S(0)) sea menor que $\frac{\gamma}{\lambda}$, entonces, como S es decreciente en el tiempo y $\frac{dI}{dt}|_{t=0} < 0$, I sería decreciente en el tiempo también y no habría epidemia. En el caso de que S_0 sea mayor que $\frac{\gamma}{\lambda}$, existiría un rango en el tiempo $t \in [0, \bar{t})$ en el que $\frac{\gamma}{\lambda} < S_t \leq S_0$ y en el que $I'(t) > 0$ y por lo tanto habría una epidemia.

A $\frac{\gamma}{\lambda}$ se le conoce como *tamaño crítico de la comunidad* y es el número de susceptibles que S tiene que tener para que ocurra una epidemia. Otra forma de expresar este umbral es utilizando el *número básico de reproducción* $\Re_0$ por individuo infectado que se expresa como:

$$\Re_0 = \frac{\lambda}{\gamma} \tag{2.2}$$

y que indica que una epidemia ocurre solo si $\Re_0 > 1$. Para ver como el valor de $\Re_0$ influye en el avance de una epidemia, se ha realizado un ejemplo con $S_0 = 5000$, $I_0 = 1$, $\lambda = 0{,}4$, y $\gamma = 0{,}2$ lo cual resulta en un $\Re_0 = 2$. El resultado puede visualizarse en la figura 2.4.

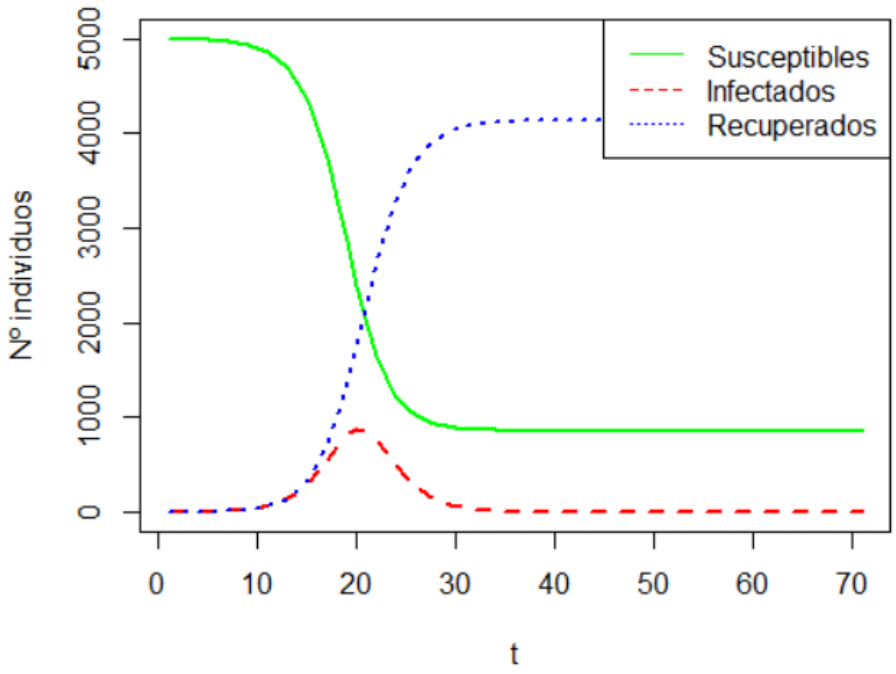


Figura 2.4: Desarrollo de una pandemia con $\Re_0 = 2$.

Podemos simular un $\Re_0$ mayor incrementando λ a 0.72, en la figura 2.5 podemos ver como el pico de infectados está más concentrado y es más abrupto.

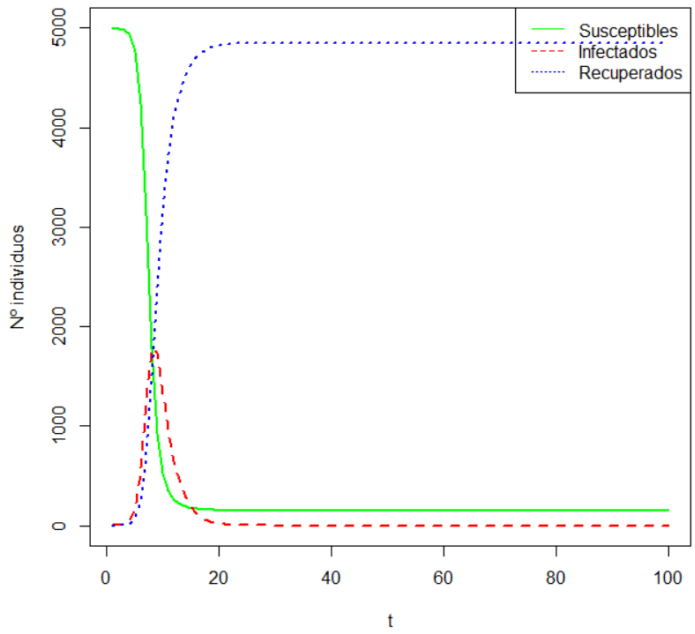


Figura 2.5: Desarrollo de una pandemia con $\Re_0 = 3{,}6$.

A medida que $\Re_0$ se aproxima a 1, los susceptibles y recuperados dejan de cruzarse en el gráfico (figura 2.6) indicando que se está llegando al valor crítico de $\Re_0$ (en $\Re_0 = 1$ son completamente horizontales).

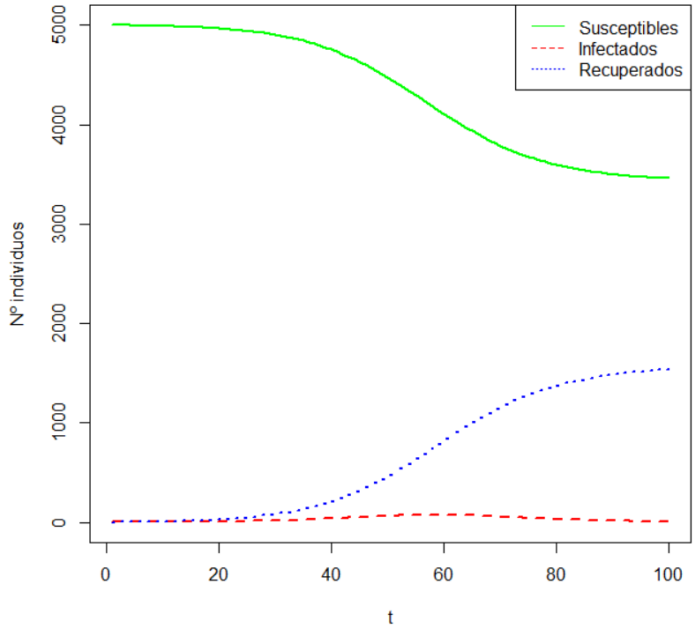


Figura 2.6: Desarrollo de una pandemia con $\Re_0 = 1{,}4$.

Tras haber comentado qué es necesario para que una epidemia ocurra, lo más normal es continuar indicando las razones por las que una epidemia termina [34, 37]. Para ello hay que basarse otra vez en las derivadas de nuestros compartimentos.

Como $S'(t) \leq 0$, $S_0 > 0$, y $N(t) = S(t)+I(t)+R(t)$, $S(t)$ queda delimitado por $0 \leq S(t) \leq S_0 \leq N$, y como $R'(t) \geq 0$, $R_0 = 0$, $R(t)$ queda delimitado por $0 \leq R_0 \leq R(t) \leq N$. Por lo tanto los límites

$$S(\infty) = lim_{t\to\infty} S(t)$$

$$R(\infty) = lim_{t\to\infty} R(t)$$

existen, y por tanto

$$I(\infty) = lim_{t\to\infty} I(t) = N - S(\infty) - R(\infty).$$

Ahora que sabemos que $R(\infty)$ e $I(\infty)$ existen, podemos demostrar que $I(\infty) = 0$. Ya que si $I(\infty) > 0$, el valor de $R(\infty)$ según $R'(t)$ sería ∞ lo cual es una imposibilidad debido a que los límites de R son 0 y N.

Por lo tanto concluimos que la razón por la que una enfermedad deja de transmitirse no es por la falta de susceptibles, sino por la falta de infectados como la figura 2.4 sugiere.

2.2.2. Estimadores de la severidad de una epidemia

Por último, vamos a ver como es posible estimar la severidad de una epidemia a partir de $\Re_0$ [34].

Lo primero que necesitamos hacer es obtener $\frac{dI}{dS}$:

$$\frac{dI}{dS} = -1 + \frac{\gamma}{\lambda S}, \tag{2.3}$$

y realizar su primera integral:

$$\phi(S, I) = I + S - \frac{\gamma}{\lambda} log(S). \tag{2.4}$$

Ahora asumiendo $I \approx 0$ tenemos que $\phi(S_0, 0) = \phi(S_\infty, 0)$, lo cual equivale a:

$$S_0 - S_\infty = \frac{\gamma}{\lambda}(log(S_0) - log(S_\infty)). \tag{2.5}$$

Existen varias opciones para estimar el alcance que puede tener una epidemia, dos de las más simples son S_∞, que simboliza la cantidad de gente que conseguirá escapar la epidemia sin infectarse, y $S_0 - S_\infty$ que indica la cantidad de gente que se infectará. Si además de conocer S_0, conocemos $\Re_0$, podemos obtener $\frac{\gamma}{\lambda}$ a partir de la ecuación 2.2 y finalmente estimar S_∞ a partir de la ecuación 2.5.

La tercera forma de estimar la severidad consiste en calcular I_{max}, que se puede usar para prever si existirá suficiente espacio en los hospitales. Al igual que lo anterior, partimos de la ecuación 2.4:

$$\phi(S_0, I_0) = I_0 + S_0 - \frac{\gamma}{\lambda} log(S_0) = N - \frac{\gamma}{\lambda} log(S_0).$$

$\phi(S, I)$ tiene la propiedad de mantenerse constante a lo largo de una epidemia por lo tanto I puede expresarse como:

$$I = -S + \frac{\gamma}{\lambda} log(S) + N - \frac{\gamma}{\lambda} log(S_0). \tag{2.6}$$

Como ya sabemos, al comienzo de una epidemia I crece hasta que alcanza su máximo, luego desciende, y finalmente llega a 0 y la epidemia acaba. Con esto sabemos que I_{max} ocurrirá en el tiempo t en el que $\frac{dI}{dt} = 0$, en otras palabras, cuando $S = \frac{\gamma}{\lambda}$:

$$I_{max} = -\frac{\gamma}{\lambda} + \frac{\gamma}{\lambda} log(\frac{\gamma}{\lambda}) + N - \frac{\gamma}{\lambda} log(S_0). \tag{2.7}$$

Calculándolo para el ejemplo de la figura 2.4 obtenemos:

$$I_{max} = -2500 + 2500 * log(2500) + 5001 - 2500 * log(5000) = 768.$$

Todo lo visto hasta ahora parece depender de los parámetros λ y γ, así que poder estimarlos correctamente será clave para poder realizar un buen estudio de la epidemia. Esta estimación resulta bastante práctica, por lo tanto se realizará en el próximo capítulo utilizando los métodos de Monte Carlo explicados en el capítulo 1.

Capítulo 3

Generación de modelos epidemiológicos SIR y simulación de parámetros por métodos MCMC

En este capítulo vamos a aplicar lo que hemos visto hasta ahora a dos conjuntos de datos. Vamos a utilizar los datos de la pandemia de influenza de Inglaterra que mostramos en un ejemplo del apartado 2.2 , y los datos de casos de Covid-19 durante la primera ola en Castilla y León. Tras presentar los resultados de las pruebas, sacaremos algunas conclusiones acerca de la efectividad de cada método y modelo. Ambos conjuntos de datos pueden visualizarse en la figura 3.1 y 3.2 respectivamente (además de en los Anexos I y II respectivamente).

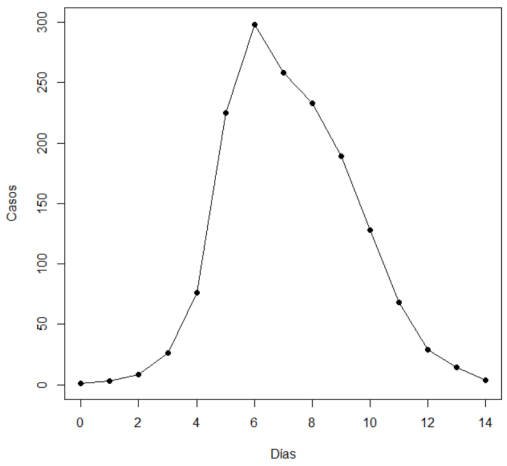


Figura 3.1: Casos de influenza en un internado de Inglaterra.

Para los datos de influenza disponemos de 14 entradas desde el 22 de enero de 1978 hasta el

4 de febrero que indican la prevalencia. Por otro lado, los datos de Castilla y León constan de 137 entradas comenzando el 20 de febrero de 2020 e indican la incidencia diaria.

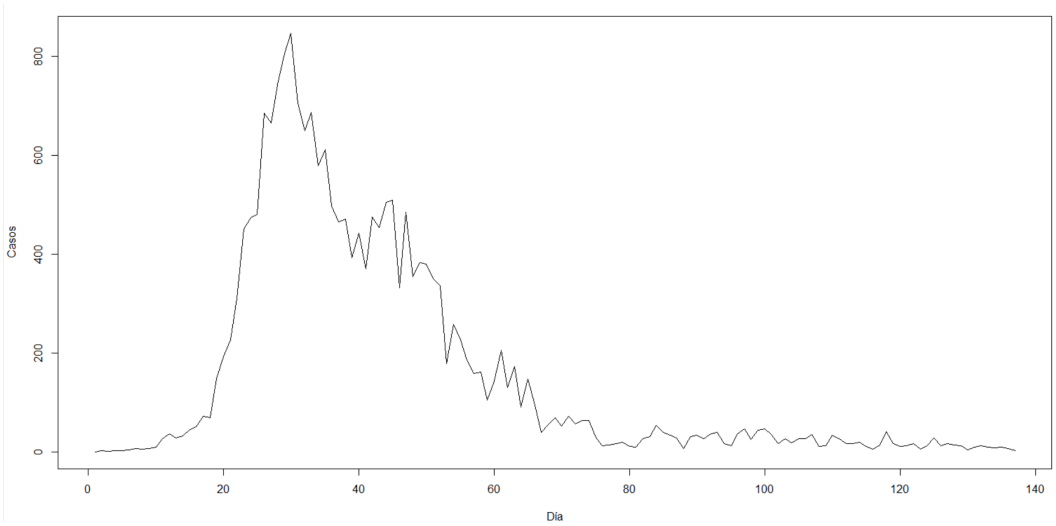


Figura 3.2: Incidencia de Covid-19 en CyL.

A lo largo de este capítulo se mostrarán 4 tipos distintos de gráficos:

- Un gráfico de las trayectorias elegidas durante el modelado: este gráfico presenta en azul las trayectorias obtenidas resolviendo numéricamente las ODEs basadas en los parámetros aceptados en las iteraciones del Monte Carlo, y en negro los datos reales de los que disponemos.
- Histogramas de los parámetros: histogramas que indican el valor de los diferentes parámetros de los modelos elegidos.
- Un gráfico de datos simulados: este gráfico presenta en azul los datos simulados a partir de las trayectorias, y en negro los datos reales. Para obtener estos datos tenemos que utilizar cada uno de los modelos elegidos en el Monte Carlo y simular un valor aleatorio utilizando una Poisson con parámetro igual al valor simulado del modelo.
- Intervalos de credibilidad de los datos simulados: en azul los intervalos de confianza del 95 % obtenidos al realizar simulaciones con las trayectorias obtenidas en la el primer gráfico descrito, y en negro los datos reales, que deberían estar contenidos en el intervalo de confianza.

Como ya se ha indicado, va a utilizarse el modelo de Kermack McKendrick para realizar las simulaciones. Para los datos de influenza, la aplicación es directa, ya que el número de infectados en cada día equivale al compartimento I, pero los datos de Castilla y León son nuevos infectados diarios, y eso no está representado en ningún compartimento. Para poder representar correctamente estos datos vamos a tener que cambiar el modelo ligeramente.

Necesitamos modelar el ratio de nuevos infectados por unidad de tiempo (en este caso días), lo cual equivaldría al ratio al que individuos dejan el compartimento S. Por lo tanto, añadir un nuevo compartimento con un ratio de entrada equivalente a I y sin ratio de salida arregla nuestro problema, el nuevo modelo viene representado en la figura 3.3. Aunque lo representemos como otro compartimento más, este no realiza una función de compartimento, sino de contador. Es decir, no hay individuos en S que puedan acabar en C.

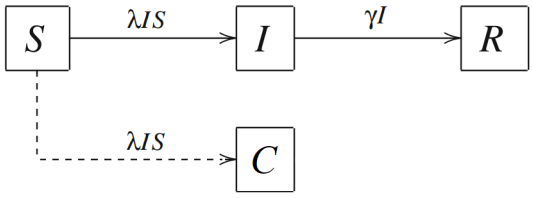


Figura 3.3: Modelo alternativo de Kermack McKendrick.

Si recordamos lo explicado en el primer capítulo, para poder realizar MCMC o HMC necesitamos unos parámetros a priori y cuanto mejor estén escogidos antes ocurrirá la convergencia hacia la distribución estacionaria.

Para los datos del internado vamos a utilizar como prioris el λ y el γ que usamos en el apartado 2.2:

$$\lambda = 1{,}9, \qquad \gamma = 0{,}5.$$

Para los datos de Castilla y León hemos elegido los prioris tras realizar un número suficiente de pruebas y comprobar que los modelos generaban resultados adecuados. Las prioris elegidas son

$$\lambda = 0{,}48, \qquad \gamma = 0{,}2.$$

También necesitamos generar propuestas para poder actualizar los parámetros en cada iteración, utilizaremos una normal centrada en el último parámetro aceptado, así que tendremos

$$\lambda^* \sim N(\lambda, 0{,}1)$$

$$\gamma^* \sim N(\gamma, 0{,}0015)$$

y

$$\lambda^* \sim N(\lambda, 0{,}1)$$

$$\gamma^* \sim N(\gamma, 0{,}05)$$

para el internado, y Castilla y León respectivamente.

Como criterio de decisión utilizaremos la verosimilitud. Para verificar los modelos, realizaremos simulaciones a partir de las estimaciones obtenidas y las compararemos con los datos reales.

3.1. Modelos SIR simulados por MCMC

El proceso de modelización del conjunto de datos del internado constituye una simplificación del de Castilla y León, así que para evitar redundancia, solo desarrollaremos el proceso de este último.

Vamos a repasar el proceso de elección de modelos para MCMC con los datos de CyL, que se utiliza con los datos del internado es una simplificación.

Con el compartimento C generado, que como ya se ha explicado se puede calcular como $C(t) = S(t-1) - S(t)$, expresando el número de infectados en nuestros datos como D y sabiendo que D sigue una distribución de Poisson

$$D_i \sim P(C_i),$$

la verosimilitud por cada unidad de tiempo viene dada por la derivada de la densidad de una Poisson de parámetro $C(t_i)$. Con la suma de las verosimilitudes (v) calculada tenemos que la probabilidad de aceptar la propuesta es

$$min(\frac{v^*}{v}, 1),$$

donde v^* es la verosimilitud de la propuesta y v es la logverosimilitud de la última propuesta aceptada.

Para los datos del internado se han realizado 40.000 repeticiones, mientras que para los de Castilla y Léon se han tenido que utilizar hacia 100.000 ya que al ser un conjunto de datos más grande y complejo requiere de más iteraciones para encontrar modelos apropiados. Vamos a comenzar analizando los resultados del modelo del internado. El código utilizado puede encontrarse en los anexos III - VI.

3.1.1. Monte Carlo de cadenas de Markov con datos de influenza de un internado

Los datos iniciales que utilizaremos para el internado son:

$$S_0 = 762,$$

$$I_0 = 1,$$

$$R_0 = 0,$$

$$\lambda = 1{,}9,$$

$$\gamma = 0{,}5.$$

En la figura 3.4 podemos ver las distintas trayectorias de los modelos SIR resultantes tras aplicar el MCMC. Podemos ver que en la región hasta el pico de infectados los datos reales se mantienen entre los modelos en todo momento. Por otro lado, en el descenso, sí que tenemos algunas instancias en las los datos reales se salen de las estimaciones. Aun así, esta prueba debería resultar en un buen ajuste.

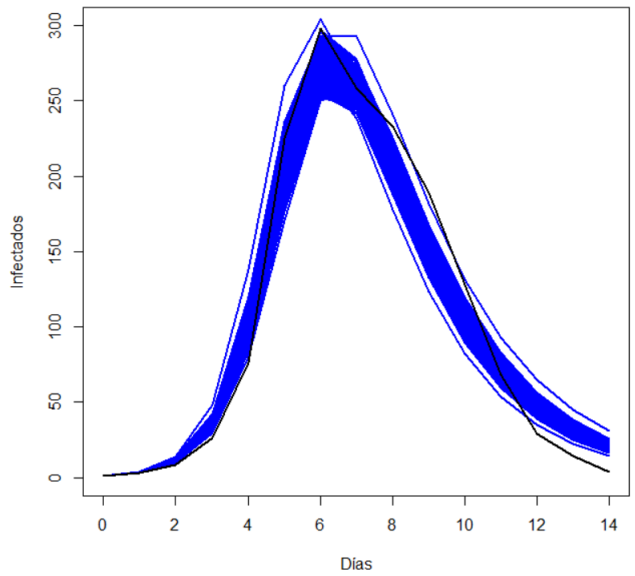


Figura 3.4: Trayectorias elegidas por MCMC con datos del internado.

Las distribuciones a posteriori de λ y γ de los modelos elegidos pueden verse en las figuras 3.5 y 3.6 respectivamente.

Vemos que la distribución de λ se ha desplazado del 2 original que habíamos pensado a alrededor de 1.7, esto no es una diferencia que requiera repetir el proceso con un nuevo valor a priori, pero podría ser un indicador de que el método de estimación a partir de $-\frac{\triangle S}{\triangle tSI}$ que indicamos en la sección 2.1 puede que no sea muy adecuado. Por otro lado, la distribución de γ si que se ha mantenido cerca de la estimación original.

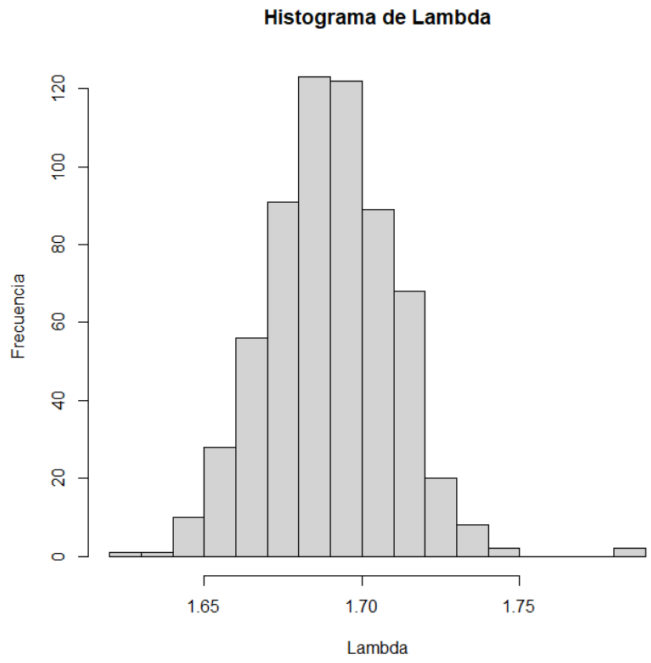


Figura 3.5: Posteriori de λ por MCMC con datos del internado.

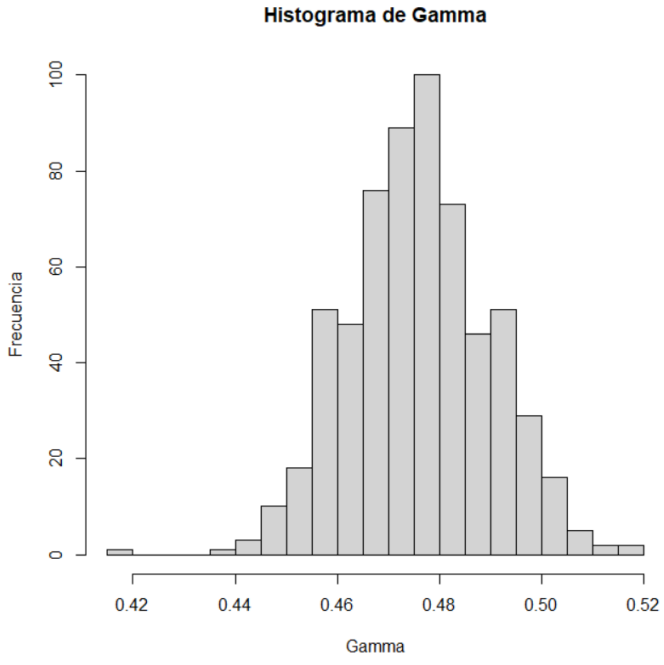


Figura 3.6: Posteriori de γ por MCMC con datos del internado.

Entrando ya en el proceso de verificación, se han representados los datos simulados (figura 3.7) a partir de las trayectorias elegidas. En el podemos ver que de los modelos simulados contienen en prácticamente todo momento a los datos, y por lo tanto, podemos concluir

que las λ y γ elegidas en el MCMC generan unos modelos que simulan correctamente el comportamiento de la epidemia.

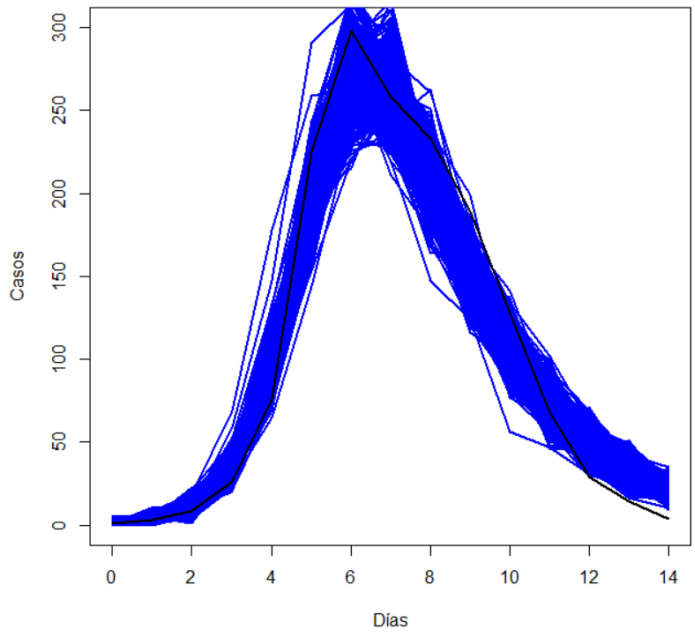


Figura 3.7: Datos simulados por MCMC con datos del internado

En cuanto a la gravedad de la epidemia podemos indicar que de media un 3.61 % de los individuos consiguen no enfermar (no comparable ya que no se dispone del porcentaje real), y que la media del número de infectados máximo se sitúa en 270 con respecto a los 298 reales.

3.1.2. Monte Carlo de cadenas de Markov con datos de Covid-19 de Castilla y León

Los datos iniciales que hemos utilizado para Castilla y León son:

$$S_0 = 2394917,$$

$$I_0 = 1,$$

$$R_0 = 0,$$

$$\lambda = 0{,}48,$$

$$\gamma = 0{,}2.$$

Las figuras 3.8 y 3.9 presentan resaltadas las trayectorias elegidas, y los datos simulados a partir de las trayectorias para los datos de CyL respectivamente. Como podemos ver, tiene algunos problemas ajustándose a la curva inicial, y no muestra un pico de infectados

tan alto como en los datos reales, pero el descenso que realiza si que parece representar bien la trayectoria que toman los datos, y por lo general es un resultado razonable.

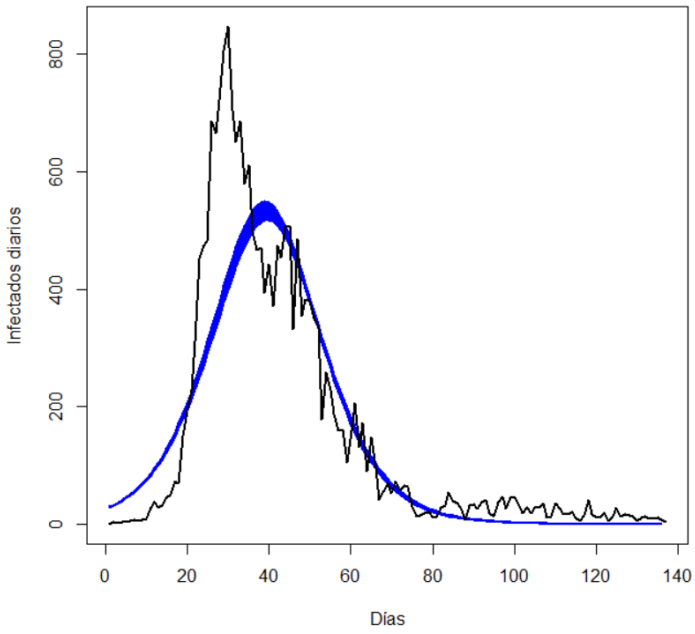


Figura 3.8: Trayectorias elegidas por MCMC con datos de CyL.

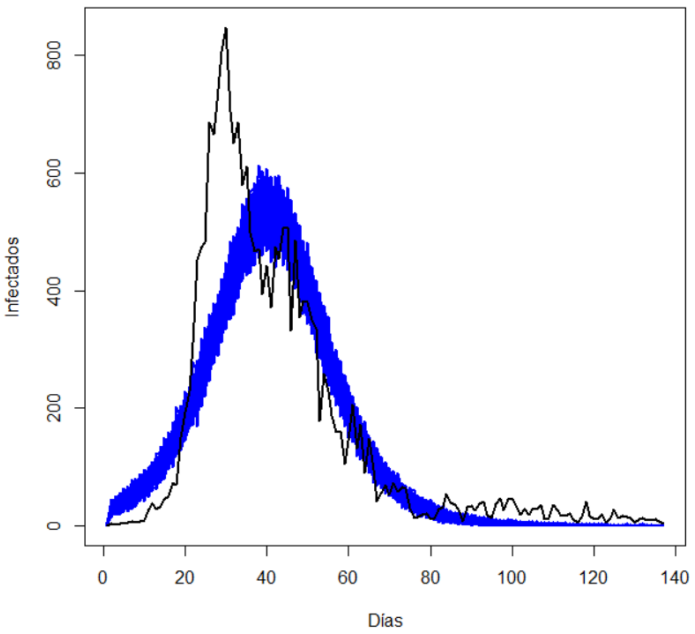


Figura 3.9: Datos simulados por MCMC con datos de CyL.

El problema de estos modelos llega cuando nos fijamos en el λ y γ a posteriori. Podemos visualizar cada histograma respectivamente en las figuras 3.10 y 3.11.

Según estos gráficos vemos que tanto λ como γ tienen valores en torno a los 27.6. Aunque sabemos que el tiempo de recuperación del SARS-CoV2 ronda 1 semana, los modelos de la figura 3.9 sugieren que el tiempo se sitúa en los $\frac{1}{27{,}6} = 0{,}036$ días, o aproximadamente 52 minutos. Esta interpretación simplemente no tiene sentido en un entorno epidemiológico. Y por lo tanto, vamos a repetir los experimentos dejando γ fija en 0.2 (5 días), para intentar conseguir modelos con cierta validez epidemiológica.

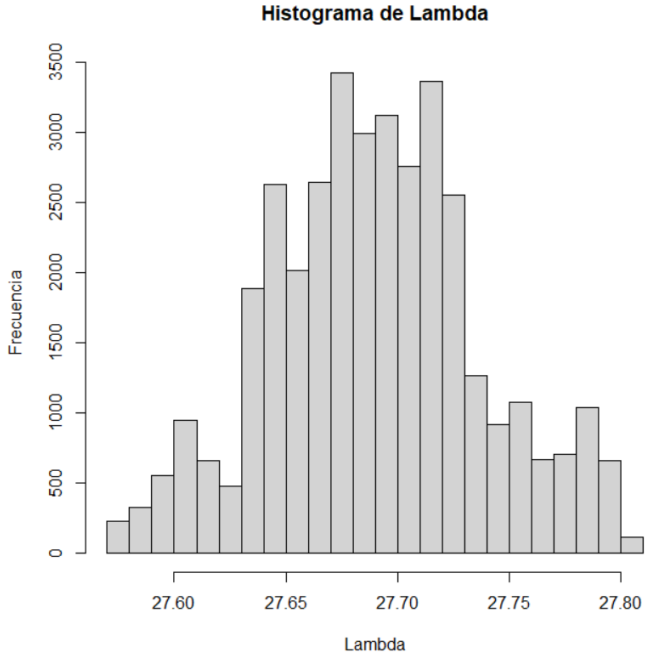


Figura 3.10: Posteriori de λ por MCMC con datos del CyL.

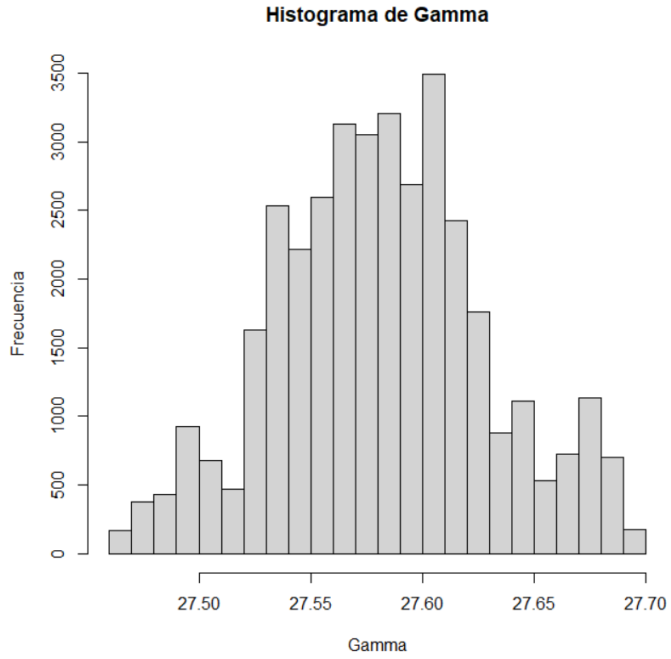


Figura 3.11: Posteriori de γ por MCMC con datos del CyL.

A vista de la figura 3.12, queda bastante claro que al fijar λ el MCMC para los datos de CyL no resulta en modelos SIR con buen ajuste. A pesar de que la primera parte sí que parece simular el crecimiento de los datos, los modelos acaban subiendo muy por encima del pico real.

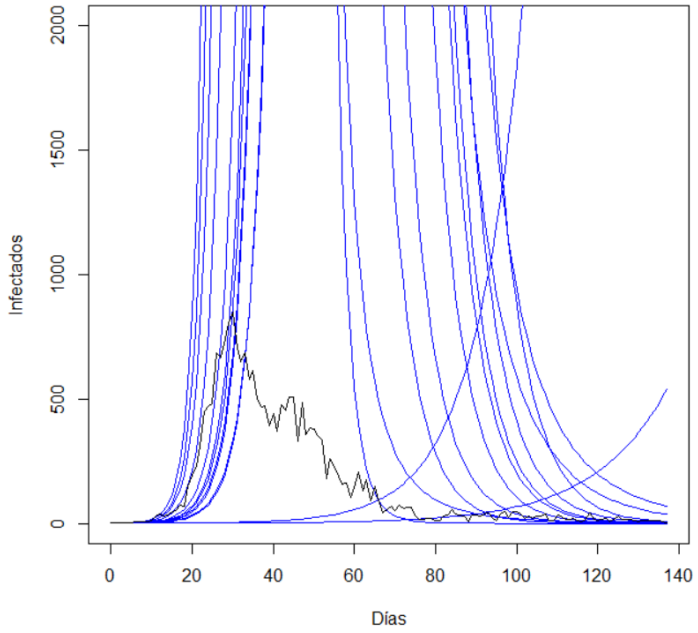


Figura 3.12: Trayectorias por MCMC con datos de CyL con γ fija.

El comportamiento que muestra puede deberse a que el periodo de cuarentena redujo en gran medida los contactos, lo que se traduce en una reducción en el ratio de contagio. Por lo tanto, mantener una λ constante durante los 137 días puede resultar en modelos con sentido epidemiológico, pero que no representan la realidad de los acontecimientos.

Aunque este modelo claramente no es apropiado para la epidemia de Covid-19 en Castilla y León, puede servir de indicador de la eficacia de las medidas que se tomaron para frenar el avance de la enfermedad.

La siguiente prueba se realizará para intentar arreglar este problema. Para ello se realizarán dos modelos SIR, uno hasta el día 29 y otro desde el 29 hasta el final. La función de esta partición es poder modelar el cambio en el ratio de transición que hemos comentado. Por lo tanto, vamos a tener λ_1 y λ_2, que simulen el ratio de contagio para cada modelo, pero solo una γ ya que el ritmo de recuperación no debería variar. El primer modelo utilizará los mismos datos iniciales que se han usado hasta ahora, y los datos finales de los compartimentos S, I, y R del primero se utilizarán como iniciales del segundo.

3.1.3. Monte Carlo de cadenas de Markov doble con datos de Covid-19 de Castilla y León

Para realizar esta prueba hemos utilizado un valor de $\lambda_1 = 0{,}5$, y $\gamma = 0{,}2$, que son los que habíamos utilizado en el apartado anterior, y para λ_2 vamos a utilizar un valor ligeramente menor al de γ de 0,17 ya que para simular un descenso en el número de infectados diarios necesitamos que $\Re_2 = \frac{\lambda_2}{\gamma} < 1$.

En la figura 3.13 podemos ver como separando los datos en dos subconjuntos y realizando varios modelos SIR mejora los resultados obtenidos en el experimento anterior.

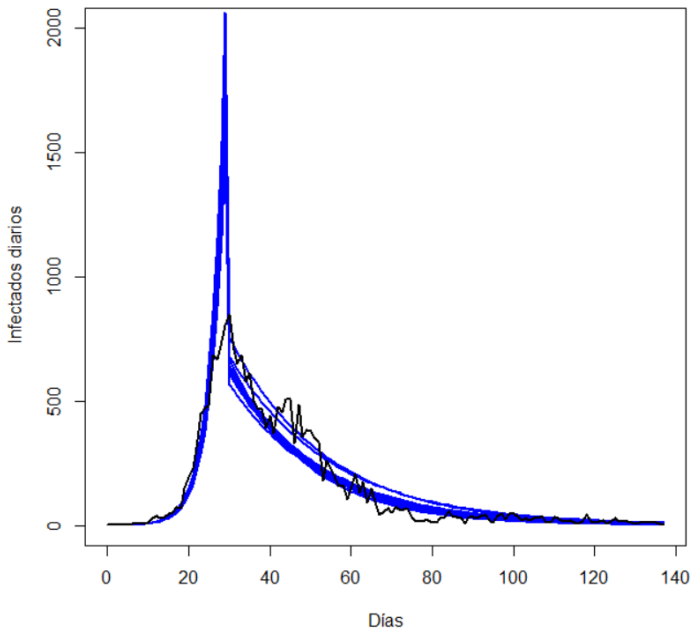


Figura 3.13: Trayectorias por MCMC conjunto con datos de CyL.

En las figuras 3.14 y 3.15 tenemos las λ_1 y λ_2 que se se han elegido en el MCMC. Para ambos parámetros podemos ver que las distribuciones a posteriori acaban bastante cerca del priori que hemos escogido.

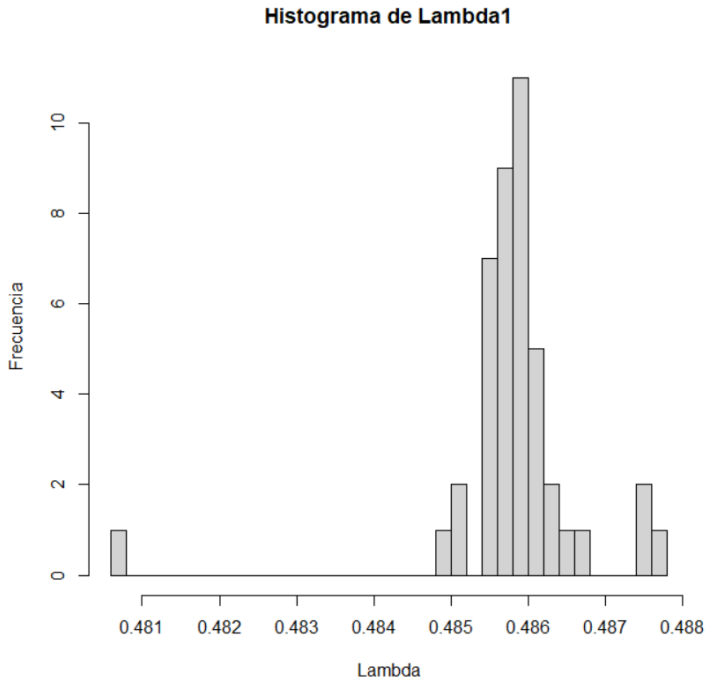


Figura 3.14: Posteriori de λ_1 por MCMC conjunto con datos de CyL.

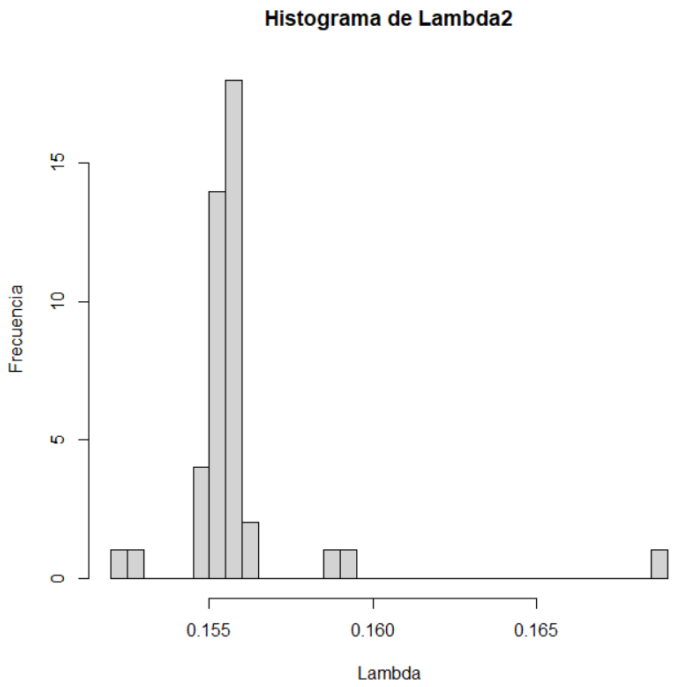


Figura 3.15: Posteriori de λ_2 por MCMC conjunto con datos de CyL.

Finalmente podemos ver en la figura 3.16 los datos simulados a partir de las trayectorias. Se observa una buena trayectoria creciente hasta que empieza a llegar el pico de afectados. Ahí, podemos ver que mientras el número de infectados reales frena su crecimiento, en los modelos sigue creciendo debido a que no modificamos su λ. Cuando realizamos el cambio de modelo en el día 29, observamos como realiza un salto y continua con una dirección bastante similar a la de la real.

En esta prueba el porcentaje de infectados medio que consigue escapar la epidemia resulta en un 99.08 %, mientras que la real es 99.12 %, por lo general una estimación bastante aceptable. Por otro lado, el número de infectados máximos varía de los 847 reales a los 1709 simulados, un resultado esperable teniendo en cuenta el problema que presenta en la estimación del pico de infectados.

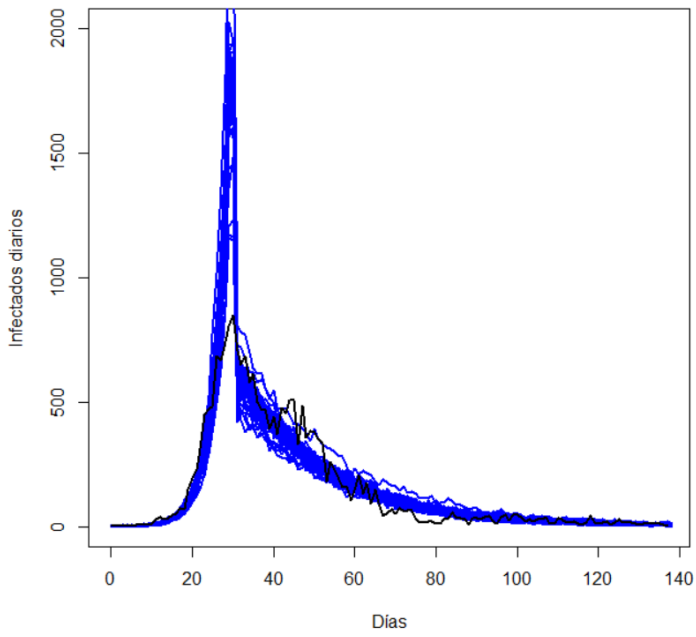


Figura 3.16: Datos simulados por MCMC conjunto con datos de CyL.

Al utilizar dos modelos es normal que acabemos con una gráfica que presente un desnivel brusco cuando se cambia de modelo. Para intentar conseguir unos resultados más naturales, en la última prueba de MCMC vamos a ajustar gradualmente λ_1 a λ_2 a medida que cambiamos de modelo.

3.1.4. Monte Carlo de cadenas de Markov doble con transición gradual con datos de Covid-19 de Castilla y León

En este modelo vamos a cambiar progresivamente λ_1 a λ_2. Esto se consigue seleccionando un número de días (n) alrededor del punto crítico y reduciendo la λ utilizada en el punto anterior en cada uno de ellos. Es decir la λ utilizada será:

$$\lambda = \begin{cases} \lambda_1 & t \leq t_{ini} \\ \lambda 1 - \frac{\lambda_1 - \lambda_2}{n}\ (t - t_{ini}) & t_{ini} < t < t_{ini} + n \\ \lambda_2 & t \geq t_{ini} + n \end{cases}$$

con $t_{ini} = 29 - \frac{n}{2}$. En este caso $n = 12$ días, por lo tanto $t_{ini} = 23$

En la figura 3.17 vemos una serie de modelos obtenidos al realizar un cambio gradual en los parámetros, a simple vista podemos deducir que deberían mejorar los resultados de la prueba anterior.

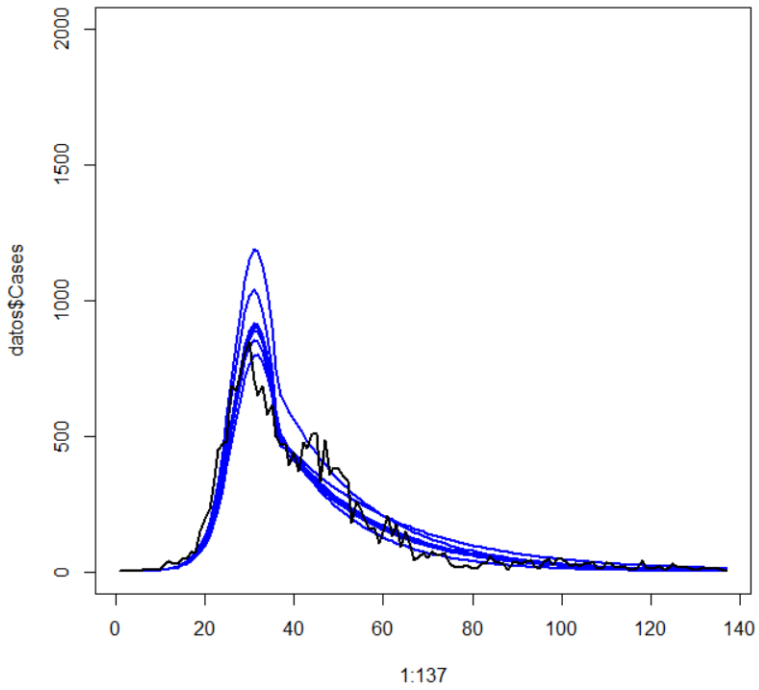


Figura 3.17: Trayectorias por MCMC conjunto gradual con datos de CyL.

Las distribuciones de λ_1 y λ_2 son muy parecidas a las de la prueba anterior, por lo tanto parece que lo único que teníamos que hacer era realizar una transición apropiada entre modelos SIR.

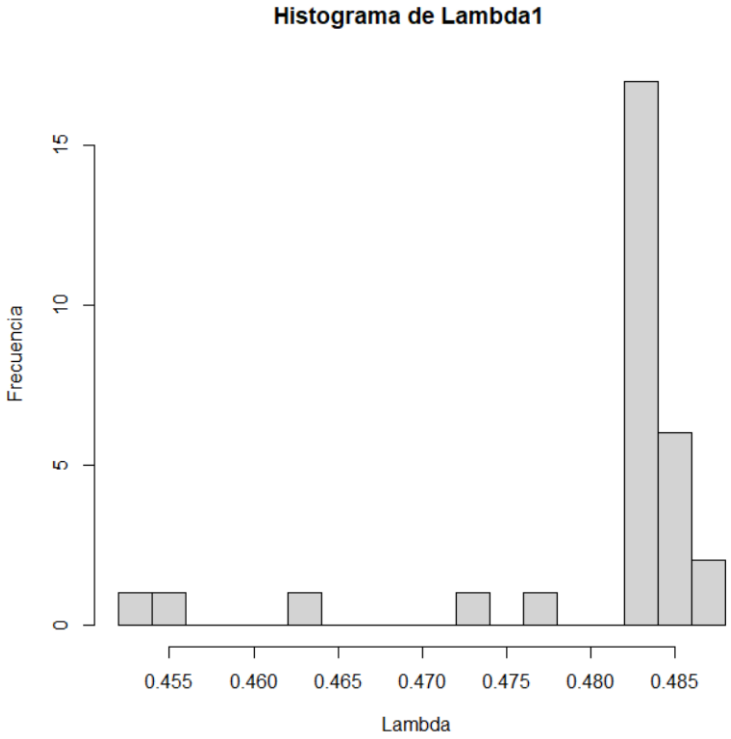


Figura 3.18: Posteriori de λ_1 por MCMC conjunto gradual con datos de CyL.

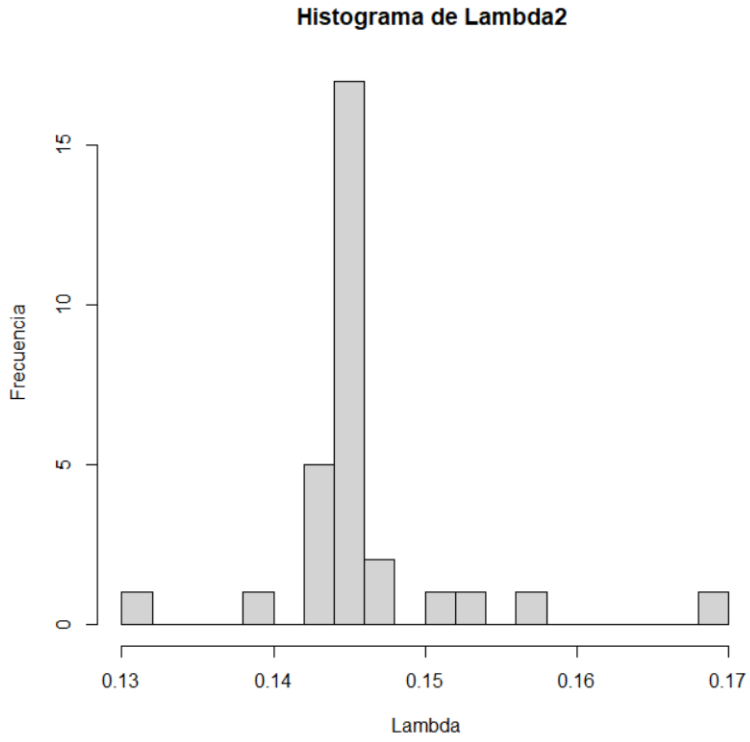


Figura 3.19: Posteriori de λ_2 por MCMC conjunto gradual con datos de CyL.

En la figura 3.20 se puede visualizar las simulaciones a partir de las trayectorias elegidas. Se puede ver que el comienzo simula bastante bien los datos, pero, a diferencia de la prueba anterior, a medida que llega al máximo de infectados los modelos también empiezan a reducir su crecimiento simulando un pico de infectados más apropiado. El resto de la simulación se ajusta bastante bien a los datos.

En esta prueba el porcentaje de la población que conseguiría evitar contagiarse sería un 99.129 % que es un porcentaje más cercano al 99.121 % real que el obtenido anteriormente. Aunque sin ninguna duda, donde mayor mejora podemos observar es durante el cálculo del pico de infectados en el que obtenemos una estimación de 947, que aunque todavía se desvíe de los 847, presenta una gran mejora si lo comparamos con los modelos sin la transición gradual.

Por otro lado, en el gráfico podemos ver que ahora existe otro subconjunto de días alrededor del día 50 que los modelos parecen no registrar correctamente. Esto podría solucionarse realizando más particiones en los datos y creando submodelos SIR más pequeños. Esto, sin ninguna duda, mejoraría el ajuste a los datos, pero el sobreajuste que introduciría no justifica la pequeña mejora de los modelos.

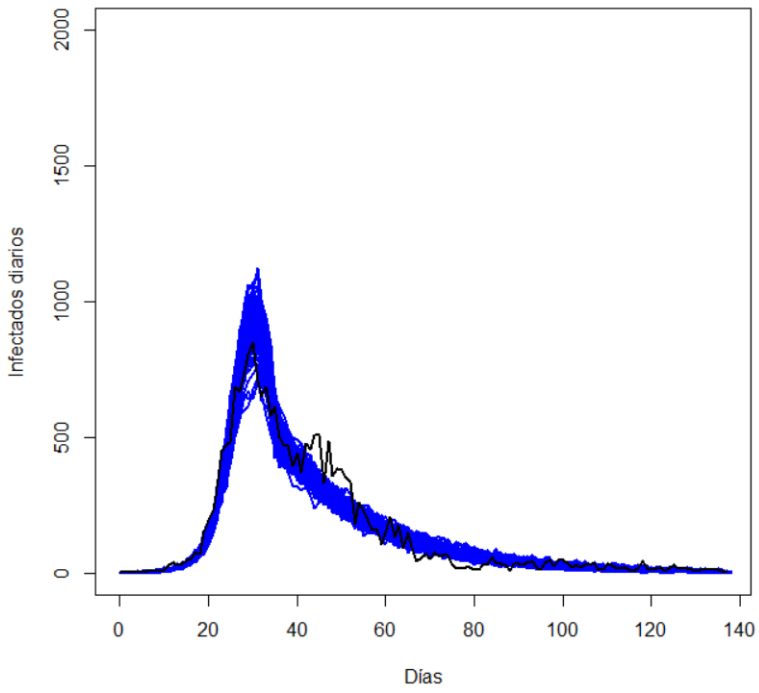


Figura 3.20: Datos simulados por MCMC conjunto gradual con datos de CyL.

3.2. Modelos SIR simulados por HMC

Al igual que en el MCMC, vamos a repasar el proceso de elección de modelos que tendremos que realizar con los datos de CyL.

Vamos a realizar un Monte Carlo Hamiltoniano con cada set de pruebas para los que utilizaremos los mismos parámetros a priori que utilizamos al hacer MCMC.

La aplicación del Monte Carlo Hamiltoniano requiere de algún paso más debido a que tenemos que calcular el Hamiltoniano. En el capítulo 1 indicamos que el Hamiltoniano es

$$H(\theta, \rho) = U(\theta) + K(\rho).$$

Donde θ es un vector con los parámetros (λ, γ) del modelo SIR. Realizando el método del salto de rana con una θ_{ini} y ρ_{ini} iniciales, obtenemos las θ y ρ que vamos a utilizar.

Una vez hecho el salto rana, podemos calcular $K(\rho)$ como

$$K(\rho) = \frac{1}{2}\rho^T M^{-1}\rho,$$

pero obtener $U(\theta)$ resulta más complejo. Recordemos que

$$U(\theta) = -\log(p(\theta)),$$

ahora podemos dividir la log posterior en la suma de verosimilitudes

$$\frac{d\log(p(\theta))}{d\theta} = \frac{d\log(L)}{d\theta} + \frac{d\log(L_{pri})}{d\theta},$$

donde sabemos que L es la verosimilitud, y L_{pri} la parte de la verosimilitud a priori.

$\frac{\log(L_{pri})}{d\theta}$ es fácil de obtener ya que sabemos que ambos parámetros de θ a priori siguen una normal. El otro componente lo podemos desarrollar con la regla de la cadena obteniendo

$$\frac{d\log(L)}{d\theta} = \sum_i^n \left(\frac{\partial \log(L)}{\partial C(t_i)} \frac{\partial C(t_i)}{\partial \theta} + \frac{\partial \log(L)}{\partial \theta} \right),$$

donde n es el número de observaciones y $\mathrm{C(t_i)}$ el número de nuevos infectados en el tiempo $\mathrm{t_i}$.

$\frac{\partial \log(L)}{\partial \theta}$ es 0 ya que L no depende explícitamente de θ, y $\frac{\partial \log(L)}{\partial C(t_i)}$ corresponde a la logverosimilitud con respecto al parámetro $C(t_i)$ en la unidad de tiempo t_i , que como ya hemos explicado en el punto 3.1, es la derivada de la función de densidad de una Poisson de parámetro $C(t_i)$.

Por último tenemos que obtener $\frac{\partial C(t_i)}{\partial \theta}$, pero como $C(t_i) = S(t_{i-1}) - S(t_i)$, primero tendremos que obtener $\frac{\partial S(t_i)}{\partial \theta}$.

Para ello podemos utilizar las ODE presentes en el análisis de sensibilidad [39, 40]

$$\frac{d}{dt}\frac{\partial u}{\partial \theta} = \frac{\partial f}{\partial u}\frac{\partial u}{\partial \theta} + \frac{\partial f}{\partial \theta},$$

donde

- $\frac{\partial u}{\partial \theta}$ es un vector compuesto por $\left(\frac{\partial S}{\partial \theta}, \frac{\partial I}{\partial \theta}, \frac{\partial R}{\partial \theta}\right)^T$.

- $\frac{\partial f}{\partial u}$ es la matriz Jacobiana de $f = (-\theta SI, \theta SI - \gamma I, \gamma I)$ con respecto a S, I, y R.
- $\frac{\partial f}{\partial \theta}$ el vector de derivadas parciales $(-SI, SI, 0)^T$, y $(0, -I, I)^T$ para λ y γ respectivamente.

Una vez tengamos H podríamos calcular $r = e^{(H(\theta,\rho)-H(\theta^*,\rho^*))}$ y ver si la nueva propuesta se acepta.

Debido a que el Monte Carlo Hamiltoniano utiliza un descenso de gradiente para calcular la siguiente actualización en vez de una aproximación aleatoria como el MCMC, el ratio de aceptación será bastante mayor y podremos utilizar menos iteraciones en las pruebas. Para realizar las pruebas con el HMC vamos a realizar 20.000 repeticiones del proceso. El código utilizado puede encontrarse en los anexos VII y VIII.

3.2.1. Monte Carlo Hamiltoniano con datos de influenza de un internado

En la figura 3.21 podemos ver las trayectorias elegidas para los datos del internado. A simple vista observamos menos varianza que en la simulación obtenida con MCMC, pero por otro lado, existen más secciones que no parecen estar correctamente representadas en los modelos.

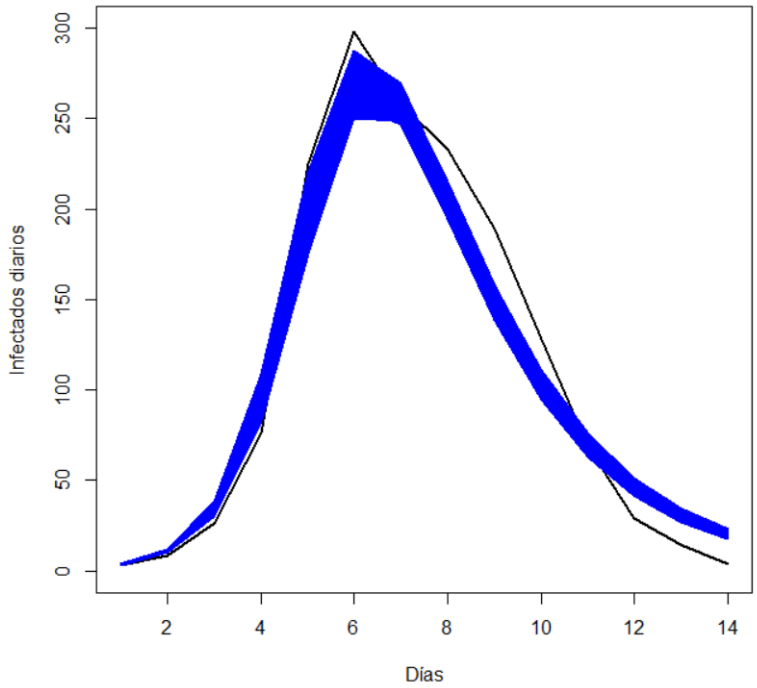


Figura 3.21: Datos reales bajo simulados por HMC con datos del internado.

Las figuras 3.22 y 3.23 representan las distribuciones a posteriori de λ y γ. Las distribuciones son bastante similares a las generadas por MCMC, pero podemos apreciar que la cantidad de propuestas aceptadas es mucho mayor.

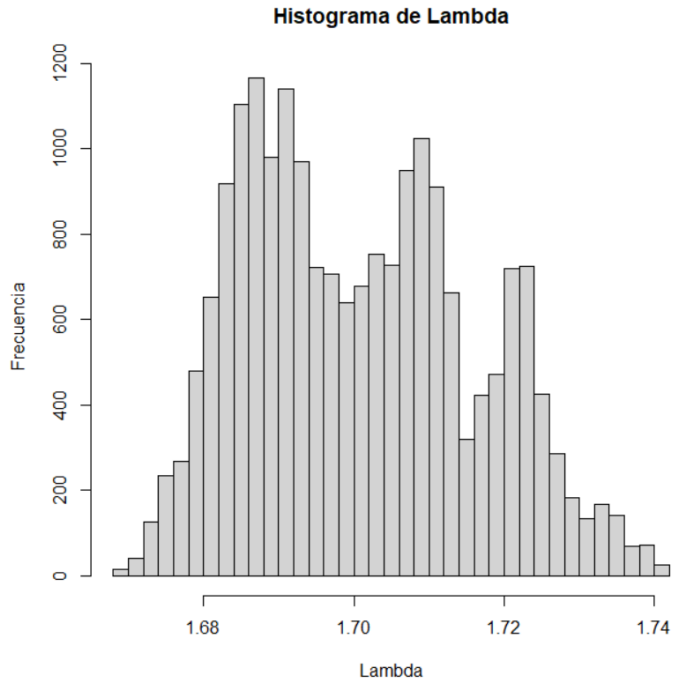


Figura 3.22: Posteriori de λ por HMC con datos del internado.

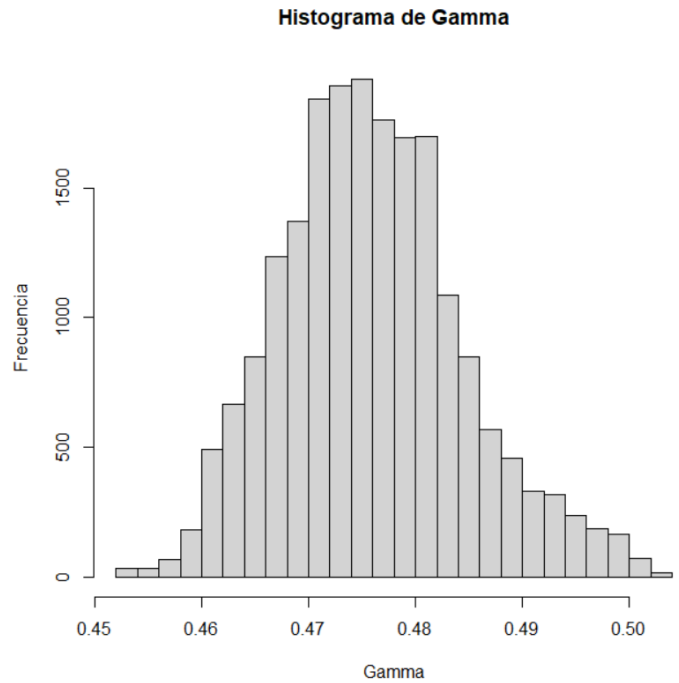


Figura 3.23: Posteriori de γ por HMC con datos del internado.

A pesar de que parezca que los trayectorias SIR estimadas de la figura 3.21 parezcan

ajustarse peor a los datos, podemos ver en la figura 3.24 que generan resultados muy similares a los de MCMC.

Las estimaciones de la severidad de la epidemia son también bastante parecidas a las del MCMC con un 3.57 % de los individuos no contagiándose, y un máximo de infectados medio de 275 que se acerca más a la realidad que el generado por MCMC.

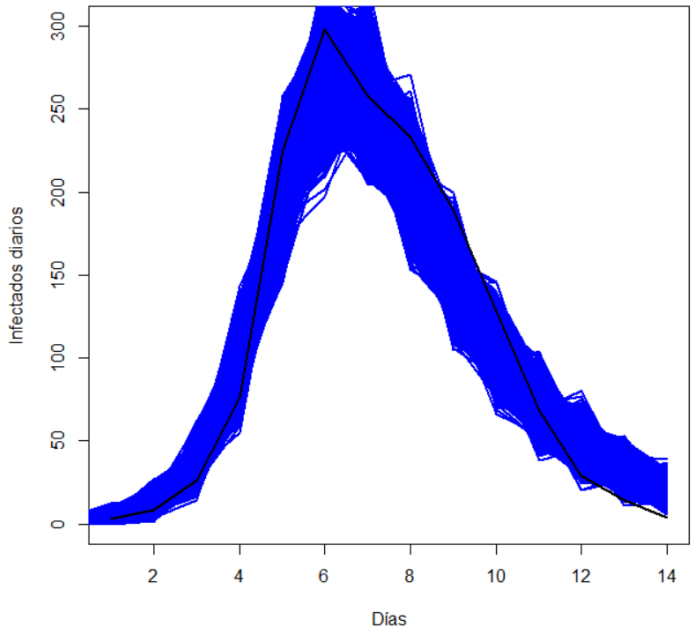


Figura 3.24: Datos simulados por HMC con datos del internado.

3.2.2. Monte Carlo Hamiltoniano con datos de Covid-19 de Castilla y León

Por último llegamos a la prueba realizada por HMC para los datos de CyL.

En la figura 3.25 y 3.26 podemos ver que los resultados generados por HMC resultan en modelos bastante similares a los que se realizaron con MCMC. Y al igual que los otros, los valores de λ y γ no tienen sentido epidemiológico.

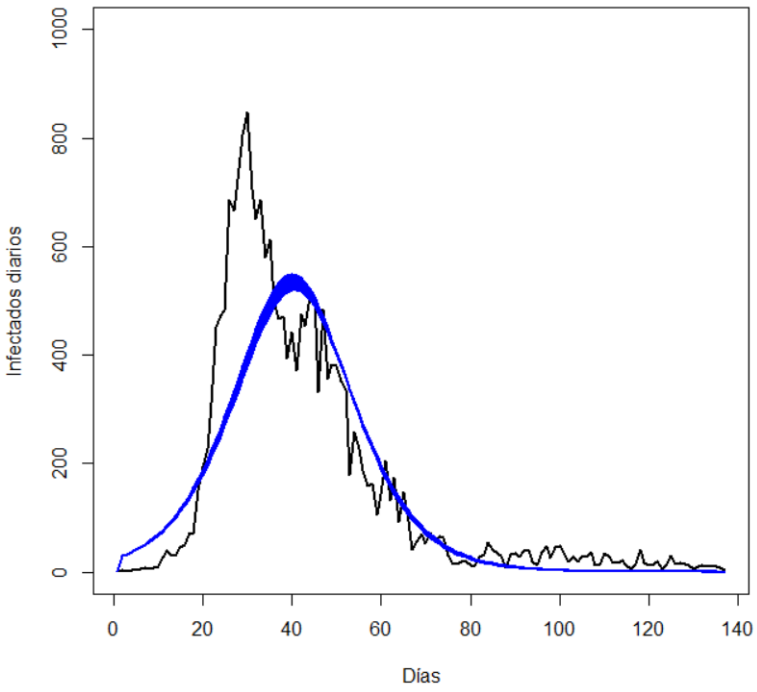


Figura 3.25: Trayectorias por HMC con datos de CyL.

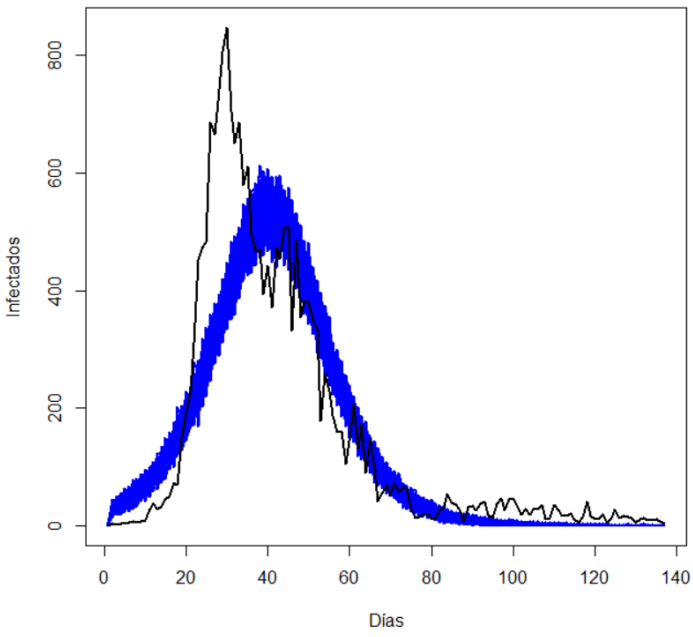


Figura 3.26: Datos simulados por HMC con datos de CyL.

Ahora, al igual que con MCMC, podríamos intentar dividir el set de datos en dos y realizar un HMC para cada uno. Pero es aquí donde nos encontramos un problema.

Debido a que HMC realiza un descenso de gradiente, el uso de derivadas es intrínseco. Y por lo tanto, tendríamos que conseguir que el cambio de un modelo a otro fuera continuo en la derivada, lo cual se puede hacer, pero queda fuera del propósito de este trabajo.

Para finalizar este capítulo vamos a realizar una breve resumen de lo que hemos observado durante la realización de las pruebas.

3.3. Comparación de modelos

En general, las pruebas realizadas a lo largo de este capítulo han proporcionado modelos adecuados para los conjuntos de datos provistos.

Si bien es cierto que el Monte Carlo Hamiltoniano, ha conseguido mejorar los resultados obtenidos por el Monte Carlo de cadenas de Markov en el set de datos del internado, también hemos podido observar que puede presentar algunas dificultades al utilizarlo con conjuntos de datos que presentan condiciones epidemiológicas variables.

En la tabla 3.1 podemos ver un resumen que muestra el porcentaje de los modelos que han sido aceptados durante cada proceso.

	% de aceptación
MCMC internado	1.61
MCMC CyL (2 modelos)	0.152
MCMC CyL (2 modelos con transición gradual)	0.0425
HMC internado	66.77

Tabla 3.1: Porcentaje de aceptación de cada procedimiento.

Podemos percibir a simple vista la diferencia que existe entre los modelos MCMC y HMC. A diferencia de la actualización aleatoria del MCMC, HMC utiliza un descenso de gradiente, un método basado en la búsqueda de mínimos locales. Esto resulta en un porcentaje de aceptación mucho más alto, y menos repeticiones necesarias para conseguir resultados robustos.

Por otro lado, también podemos ver grandes diferencias si nos fijamos en los métodos MCMC. La razón principal por la que el porcentaje del MCMC del internado es 10 veces el obtenido en el MCMC de CyL de 2 modelos, es debido a que en uno se utilizan datos de prevalencia y en el otro de incidencias.

Por último el porcentaje de aceptación de las dos pruebas que utilizan los datos de CyL es tan diferente debido al proceso extra que hemos introducido para evitar un cambio brusco entre modelos. Por otro lado, la mejora de resultados que trae debería justificar la realización de más repeticiones del modelo.

Por lo general, podemos recomendar el uso del Monte Carlo Hamiltoniano siempre que sea posible, ya que parece proporcionar mejores resultados y necesita de menos iteraciones. Por otra parte, no podemos ignorar la eficacia del Monte Carlo de cadenas de Markov, que debido a su simpleza puede ser modificado fácilmente para adecuarse a cada situación.

Conclusiones y trabajo futuro

El objetivo principal ha sido investigar y desarrollar modelos de tipo SIR generados a partir de parámetros estimados por medio de métodos de Monte Carlo de cadenas de Markov. Los principales resultados logrados durante la realización de este trabajo han sido los siguientes:

1. Se ha obtenido un conocimiento sólido acerca de los métodos de muestreo MCMC: uno de los dos subobjetivos clave para la realización del trabajo concebido. Este entendimiento de los métodos ha sido adquirido durante el periodo de redacción del primer capítulo "Monte Carlo de cadenas de Markov y Monte Carlo Hamiltoniano", y se han realizado ejemplos que muestren su funcionamiento.

2. Se ha realizado un estudio sobre los modelos SIR: este ha sido el otro subobjetivo que ha sido una base crítica para ejecutar las secciones prácticas del proyecto. Se han señalado las características principales de los modelos SIR, se han investigado algunas de las causas que provocan el comienzo y final de una epidemia, y se han definido algunos métodos de estimar la gravedad de una epidemia.

3. Se ha conseguido aplicar los conceptos al caso particular: se han analizado los conjuntos de datos y se aplicado los conceptos teóricos de modelización a cada conjunto teniendo en cuenta sus peculiaridades.

4. Se han producido resultados y comparado la calidad de los mismos: se han generado un conjunto de modelos SIR que simulan los conjuntos de datos, se han presentado las distribuciones a posteriori de los parámetros que se utilizan, y se ha comprobado la calidad del ajuste utilizando una distribución predictiva posterior e intervalos de confianza. Además se ha redactado una pequeña comparación entre todos los modelos que hemos utilizado.

Para finalizar el trabajo se presentan algunas posibles líneas de trabajo que podrían tomarse.

Se propone el uso de otros modelos con el fin de encontrar el que mejor se adecúe a las peculiaridades del Covid-19. Algunos de los que se podrían considerar son los modelos SEIR, que introducen un compartimento simbolizando la cantidad de individuos expuestos a la enfermedad, o SIQR, que presentan un compartimento dedicado a individuos en cuarentena.

También se podrían emplear otros métodos de muestreo que permitan realizar mejores ajustes como el Quasi-Monte Carlo de cadenas de Markov, o incluso realizar el HMC por partes que hemos comentado en el trabajo.

Por último, se podría realizar una investigación acerca de los posibles cambios necesarios en los métodos de muestreo para mejorar su porcentaje de aceptación, en especial los que utilizan Monte Carlo.

Anexo I: Datos del internado de inglaterra (Influenza)

Fecha	Casos
1978-01-22	3
1978-01-23	8
1978-01-24	26
1978-01-25	76
1978-01-26	225
1978-01-27	298
1978-01-28	258
1978-01-29	233
1978-01-30	189
1978-01-31	128
1978-02-01	68
1978-02-02	29
1978-02-03	14
1978-02-04	4

Anexo II: Datos de Castilla y León (Covid 19)

Fecha	Casos	Fecha	Casos	Fecha	Casos
20/02/2020	1	06/04/2020	485	22/05/2020	41
21/02/2020	4	07/04/2020	355	23/05/2020	18
22/02/2020	2	08/04/2020	383	24/05/2020	13
23/02/2020	4	09/04/2020	381	25/05/2020	37
24/02/2020	4	10/04/2020	351	26/05/2020	47
25/02/2020	5	11/04/2020	336	27/05/2020	26
26/02/2020	7	12/04/2020	179	28/05/2020	45
27/02/2020	6	13/04/2020	258	29/05/2020	47
28/02/2020	8	14/04/2020	228	30/05/2020	36
29/02/2020	10	15/04/2020	187	31/05/2020	18
01/03/2020	28	16/04/2020	159	01/06/2020	28
02/03/2020	38	17/04/2020	162	02/06/2020	19
03/03/2020	29	18/04/2020	106	03/06/2020	28
04/03/2020	33	19/04/2020	144	04/06/2020	28
05/03/2020	46	20/04/2020	206	05/06/2020	36
06/03/2020	51	21/04/2020	131	06/06/2020	12
07/03/2020	73	22/04/2020	173	07/06/2020	13
08/03/2020	70	23/04/2020	91	08/06/2020	35
09/03/2020	150	24/04/2020	148	09/06/2020	27
10/03/2020	194	25/04/2020	95	10/06/2020	17
11/03/2020	228	26/04/2020	41	11/06/2020	17
12/03/2020	311	27/04/2020	56	12/06/2020	21
13/03/2020	451	28/04/2020	70	13/06/2020	12
14/03/2020	474	29/04/2020	53	14/06/2020	6
15/03/2020	481	30/04/2020	73	15/06/2020	15
16/03/2020	686	01/05/2020	58	16/06/2020	42
17/03/2020	665	02/05/2020	65	17/06/2020	17
18/03/2020	743	03/05/2020	64	18/06/2020	12
19/03/2020	806	04/05/2020	30	19/06/2020	14
20/03/2020	847	05/05/2020	14	20/06/2020	18
21/03/2020	705	06/05/2020	15	21/06/2020	6
22/03/2020	650	07/05/2020	18	22/06/2020	14
23/03/2020	687	08/05/2020	20	23/06/2020	29
24/03/2020	579	09/05/2020	12	24/06/2020	14
25/03/2020	612	10/05/2020	11	25/06/2020	17
26/03/2020	499	11/05/2020	28	26/06/2020	15
27/03/2020	466	12/05/2020	32	27/06/2020	13
28/03/2020	471	13/05/2020	55	28/06/2020	5
29/03/2020	393	14/05/2020	40	29/06/2020	11
30/03/2020	443	15/05/2020	35	30/06/2020	13
31/03/2020	371	16/05/2020	29	01/07/2020	10
01/04/2020	475	17/05/2020	7	02/07/2020	9
02/04/2020	454	18/05/2020	32	03/07/2020	11
03/04/2020	506	19/05/2020	34	04/07/2020	8
04/04/2020	509	20/05/2020	27	05/07/2020	3
05/04/2020	332	21/05/2020	38		

Anexo III: Código para simular modelos SIR por MCMC para datos del internado

```
library(EpiDynamics)
library(outbreaks)
library(Rmpfr)
datos=influenza_england_1978_school

lambda1 <- 1.9
gamma1 <- 0.5
lambdas=c()
gammas=c()
#############################################################################
N <- 763

S <- 762 /N
I <- 1  /N
R <- 0

initials <- c(S = S, I = I, R = R)
parameters <- c(beta = lambda1, gamma = gamma1)
res1 <- SIR(init = initials, pars = parameters, time = 0:14)

datosf=c(1,datos$in_bed)
plot(0:14, datosf, type = "l",ylim=c(0,300),ylab="Casos", xlab="Días")
points(0:14, datosf,pch=16)
lines(0:14, res1$results$I*N)

x1 <- mpfr("0", 128)
for (j in 1:15) {
  x1 <- x1 + log(dpois(datosf[j], res1$results$I[j]*N))
}
#############################################################################
```

```
pois=c()
plot(0:14, datosf, type = "l",ylim=c(0,300),ylab="Infectados",xlab="Días")
for (i in 1:40000) {
  lambda1.nu <- rnorm(1, lambda1, 0.2)
  gamma1.nu <- max(rnorm(1, gamma1, 0.1), 0.4)


  ###########################################################################
  N <- 763


  S <- 762 /N
  I <- 1  /N
  R <- 0

  initials <- c(S = S, I = I, R = R)
  parameters <- c(beta = lambda1.nu, gamma = gamma1.nu)

  res1 <- SIR(init = initials, pars = parameters, time = 0:14)

  x1.nu <- mpfr("0", 128)
  for (j in 1:15) {
    x1.nu <- x1.nu + log(dpois(datosf[j], res1$results$I[j]*N))
  }
  ###########################################################################

  if(log(runif(1)) < min((x1.nu - x1), 0)){
    print(i)
    x1 <- x1.nu
    lambda1 <- lambda1.nu
    gamma1 <- gamma1.nu
    lambdas=rbind(lambdas,lambda1)
    gammas=rbind(gammas,gamma1)
    lines(0:14, res1$results$I*N,lwd=2,col="blue")
    pois=cbind(pois,rpois(15,res1$results$I*N))
  }
}
lines(0:14, datosf,lwd=2)
```

Anexo IV: Código para simular modelos SIR por MCMC para datos de CyL

```
library(EpiDynamics)

datos <- read.csv("C:/Users/marco/Desktop/pincho/KINGSTON/5º/
        TFGest/datosCovidCyL.csv",sep=";")


N <- 2394918

S <- 2394917 / N
I <- 1 / N
R <- 0

logLIkelihoodSir <- function(parameters){
  parameters <- c(beta = parameters[1], gamma = parameters[2])
  initials=c(S=S,I=I,R=R)
  res <- EpiDynamics::SIR(init = initials, pars = parameters,
        time = 0:136)

  C <- c(1 / N)
  for (i in 2:137) {
    C <- rbind(C, res$results$S[i-1] - res$results$S[i])
  }
  x <- 0
  for (j in 1:137) {
    x <- x  + dpois(datos$Cases[j], C[j] * N, log = TRUE)
  }
  return(-x)
}

logLIkelihoodSir_2 <- function(parameters){
  parameters <- c(beta = parameters[1], gamma = parameters[2])
```

```
  initials=c(S=S,I=I,R=R)
  res <- EpiDynamics::SIR(init = initials, pars = parameters,
        time = 0:136)

  C <- c(1 / N)
  for (i in 2:137) {
    C <- rbind(C, res$results$S[i-1] - res$results$S[i])
  }
  x <- 0
  for (j in 1:137) {
    x <- x  + dpois(datos$Cases[j], C[j] * N, log = TRUE)
  }
  return(list(x, res))
}
#
maxPosLikelihood <- optim(c(0.5, 0.2), logLIkelihoodSir)

x <- -maxPosLikelihood$value
lambda <- maxPosLikelihood$par[1]
gamma <- maxPosLikelihood$par[2]

burn_in <- 1000
N_M <- 20000
acceptanceRate <- 0
traceMCMC <- array(dim = c(N_M - burn_in, 2))
pois=c()
plot(datos$Cases, type = "l",lwd=2)
for (i in 1:N_M) {
  lambda.nu <- rnorm(1, lambda, 0.005)
  gamma.nu <- rnorm(1, gamma, 0.005)

  if(i > burn_in){
    traceMCMC[i - burn_in,] <- c(lambda, gamma)
  }

  res.full <- logLIkelihoodSir_2(c(lambda.nu, gamma.nu))
  x.nu <- res.full[[1]]
  res <- res.full[[2]]
  if(log(runif(1)) < min((x.nu - x), 0)){
    x <- x.nu
    lambda <- lambda.nu
    gamma <- gamma.nu
    if (i > burn_in){
      acceptanceRate <- acceptanceRate + 1
      lines(1:136, (res$results$S[1:136] - res$results$S[2:137]) * N,
            lwd = 2, col = "blue")
```

```
      C <- c(1 / N)
      for (i in 2:137) {
        C <- rbind(C, res.full[[2]]$results$S[i-1] -
                res.full[[2]]$results$S[i])
      }
    pois=cbind(pois,C)
    }
  }
}
```

Anexo V: Código para simular modelos SIR dobles por MCMC para datos de CyL

```
library(EpiDynamics)
datos=read.csv("C:/Users/marco/Desktop/pincho/KINGSTON/5º/
        TFGest/datosCovidCyL.csv",sep=";")


lambda1 <- 0.48
gamma1 <- 1/5
lambda2 <- 0.17
gamma2 <- gamma1
lambdas1=c()
lambdas2=c()
#############################################################################
N <- 2394918

S <- 2394917 / N
I <- 1 / N
R <- 0

t_medidas <- 29

initials <- c(S = S, I = I, R = R)
parameters <- c(beta = lambda1, gamma = gamma1)
res1 <- SIR(init = initials, pars = parameters, time = 0:t_medidas)
initials <- c(S = res1$results$S[(t_medidas+1)],
        I = res1$results$I[(t_medidas+1)],
        R = res1$results$R[(t_medidas+1)])
parameters <- c(beta = lambda2, gamma = gamma2)
res2 <- SIR(init = initials, pars = parameters, time = t_medidas:137)

C1 <- c(1 / N)
for (i in 2:(t_medidas + 1)) {
```

```
  C1 <- rbind(C1, res1$results$S[i - 1] - res1$results$S[i])
}
plot(0:137, c(1, datos$Cases), type = "l", ylim = c(0, 3000), col = 2)
lines(0:t_medidas, C1 * N)

C2 <- c(C1[length(C1)])
for (i in 2:(137-t_medidas+1)) {
  C2 <- rbind(C2, res2$results$S[i-1] - res2$results$S[i])
}
lines((t_medidas):137, C2 * N)

x1 <- 0
for (j in 1:t_medidas) {
  x1 <- x1 + log(dpois(datos$Cases[j], C1[j] * N))
}
for (j in (t_medidas + 1):(137)) {
  x1 <- x1  +  log(dpois(datos$Cases[j], C2[j - t_medidas] * N))
}

#############################################################################
m=0
pois=c()
plot(0:137, c(1, datos$Cases), type = "l", ylim = c(0, 2000), col = 2,
        xlab="Días", ylab="Infectados")
for (i in 1:20000) {
  lambda1.nu <- max(rnorm(1, lambda1, 0.02), 0.4)
  lambda2.nu <- max(rnorm(1, lambda2, 0.02), 0.1)

  ###########################################################################
  N <- 2394918

  S <- 2394917 / N
  I <- 1 / N
  R <- 0

  initials <- c(S = S, I = I, R = R)
  parameters <- c(beta = lambda1.nu, gamma = gamma1)

  res1 <- SIR(init = initials, pars = parameters, time = 0:t_medidas)

  C1 <- c(1 / N)
  for (j in 2:(t_medidas+1)) {
    C1 <- rbind(C1, res1$results$S[j-1] - res1$results$S[j])
  }

  x1.nu <- 0
```

```
  for (j in 1:t_medidas) {
    x1.nu <- x1.nu +  log(dpois(datos$Cases[j], C1[j] * N))
  }
  ##############################################################################

  initials <- c(S = res1$results$S[t_medidas+1],
        I = res1$results$I[t_medidas+1], R = res1$results$R[t_medidas+1])
  parameters <- c(beta = lambda2.nu, gamma = gamma1)


  res2 <- SIR(init = initials, pars = parameters, time = t_medidas:137)

  C2 <- c(C1[length(C1)])
  for (k in 2:(137 - t_medidas + 1)) {
    C2 <- rbind(C2, res2$results$S[k - 1] - res2$results$S[k])
  }


  for (j in (t_medidas + 1):(137)) {
    x1.nu <- x1.nu  +  log(dpois(datos$Cases[j], C2[j - t_medidas] * N))
  }

  ##############################################################################

  if(log(runif(1)) < min((x1.nu - x1), 0)){
    print(i)
    m=m+1
    print(m)
    x1 <- x1.nu
    lambda1 <- lambda1.nu
    lambda2 <- lambda2.nu
    lambdas1=rbind(lambdas1,lambda1)
    lambdas2=rbind(lambdas2,lambda2)

    pois=cbind(pois,rpois(139,c(C1,C2)*N))

    lines(0:t_medidas, c(C1) * N,col="blue",lwd=2)
    lines(t_medidas:137, c(C2) * N,col="blue",lwd=2)
  }
}

lines(0:137, c(1, datos$Cases), type = "l",ylim=c(0,3000),lwd=2)
```

Anexo VI: Código para simular modelos SIR dobles graduales por MCMC para datos de CyL

```
library(deSolve)

datos<-read.csv("C:/Users/marco/Desktop/pincho/KINGSTON/5º/
        TFGest/datosCovidCyL.csv",sep=";")

SIR_2 <- function(t, y, params){
  t_medidas <- params[1]
  beta0 <- params[2]
  beta1 <- params[3]
  gammaSIR <- params[4]
  t_efecto_medidas <- params[5]
  print(c(t_efecto_medidas,t_medidas))
  S <- y[1]
  I <- y[2]
  R <- y[3]
  if(t <= t_medidas){
    betaSIR <- beta0
  } else if (t > t_medidas + t_efecto_medidas){
    betaSIR <- beta1
  } else {
    betaSIR <- beta0 - ((beta0 - beta1) / t_efecto_medidas) *
            (t - t_medidas)
  }
  dS <- - betaSIR * S * I
  dI <- betaSIR * S * I - gammaSIR * I
  dR <- gammaSIR * I
  res <- c(dS, dI, dR)
  list(res)
}

lambda1 <- 0.43
```

```
gamma1 <- 1/5.2
lambda2 <- 0.14
gamma2 <- gamma1
days_effect <- 12
##############################################################################
N <- 2394918

S <- 2394917 / N
I <- 1 / N
R <- 0

t_medidas <- 29 - (days_effect / 2)

res <- ode(y = c(S = 2394917 / N, I = 1 / N, R = 0), times = 0:136, SIR_2,
        c(t_medidas, lambda1, lambda2, gamma1, 12))

C1 <- c(1 / N)
for (i in 2:137) {
  C1 <- rbind(C1, res[i - 1, 2] - res[i, 2])
}
plot(1:137, datos$Cases, type = "l", ylim = c(0, 3000), col = 2)
lines(1:137, C1 * N, col = 3)

x1 <- 0
for (j in 1:137) {
  x1 <- x1 + log(dpois(datos$Cases[j], C1[j] * N))
}

##############################################################################
m=0
plot(1:137, datos$Cases, type = "l", ylim = c(0, 2000))
pois=c()
for (i in 1:150000) {
  lambda1.nu <- max(rnorm(1, lambda1, 0.05), 0.35)
  lambda2.nu <- max(rnorm(1, lambda2, 0.05), 0.05)

  ############################################################################

  res <- ode(y = c(S = 2394917 / N, I = 1 / N, R = 0), times = 0:136,
        SIR_2, c(t_medidas, lambda1.nu, lambda2.nu, gamma1, 12))

  C1 <- c(1 / N)
  for (k in 2:137) {
    C1 <- rbind(C1, res[k - 1, 2] - res[k, 2])
  }
```

```
  x1.nu <- 0
  for (j in 1:137) {
    x1.nu <- x1.nu + log(dpois(datos$Cases[j], C1[j] * N))
  }

  ##########################################################################

  if(log(runif(1)) < min((x1.nu - x1), 0)){
    print(i)
    m=m+1
    print(m)
    x1 <- x1.nu
    lambda1 <- lambda1.nu
    lambda2 <- lambda2.nu

    if(i>100){
      pois=cbind(pois,rpois(139,C1*N))
      lines(1:137, C1 * N,lwd=2,col="blue")
    }
  }
}
lines(datos$Cases,lwd=2)
```

Anexo VII: Código para simular modelos SIR por HMC para datos del internado

```
library(deSolve)
library(outbreaks)

datos <- outbreaks::influenza_england_1978_school

SIR_2 <- function(t, y, params){
  beta <- params[1]
  gamma <- params[2]
  S <- y[1]
  I <- y[2]
  R <- y[3]
  s_beta_1 <- y[4]
  s_beta_2  <- y[5]
  s_beta_3 <- y[6]
  s_gamma_1 <- y[7]
  s_gamma_2  <- y[8]
  s_gamma_3 <- y[9]

  dS <- - beta * S * I
  dI <- beta * S * I - gamma * I
  dR <- gamma * I
  res_1 <- c(dS, dI, dR)

  M <- matrix(c(-beta * I , -beta * S , 0 ,
                beta * I , beta * S - gamma , 0 ,
                0 , gamma , 0), nrow = 3, byrow = T)


  res_2 <- M %*% c(s_beta_1, s_beta_2, s_beta_3) + c(-I * S, I * S, 0)
  res_3 <- M %*% c(s_gamma_1, s_gamma_2, s_gamma_3) + c(0, -I, I)
  list(c(res_1, res_2, res_3))
```

```
}


gradientSIR <- function(theta){
  res_Grad <- deSolve::ode(y = c(S_0 / N, I_0 / N, R_0, rep(0, 6)),
                           times = 0:14, SIR_2, theta, method = "rk4",
                           rtol = 1e-8, atol = 1e-8)
  grad1 <- sum((datos$in_bed / (res_Grad[2:15, 3] * N) - 1) *
        res_Grad[1:14, 6] * N)
  grad1 <- grad1 - (theta[1] - mu_prior) /  sigma_prior^2
  grad2 <- sum((datos$in_bed / (res_Grad[2:15, 3] * N) - 1) *
        res_Grad[1:14, 9] * N)
  grad1 <- grad2 - (theta[2] - mu_prior_2) /  sigma_prior_2^2
  return(c(grad1, grad2))
}

leapfrog <- function(theta, rho, m, epsilon, L_leapfrog){
  for(kk in 1:L_leapfrog){
    gradient <- gradientSIR(theta)
    rho <- rho + 1/2 * epsilon * gradient
    theta <- theta + epsilon * solve(m) %*% rho
    rho <- rho + 1/2 * epsilon * gradient
  }
  return(c(theta, rho))
}

logLIkelihoodSirHMC <- function(parameters){
  res <- deSolve::ode(y = c(S_0 / N, I_0 / N, R_0, rep(0, 6)),
                      times = 0:14, SIR_2, parameters, method = "rk4")
  l1 <- 0
  for (i in 1:14) {
    l1 <- l1 + dpois(datos$in_bed[i], res[i + 1, 3] * N, log = TRUE)
  }
  l1 <- l1 + dnorm(parameters[1], mu_prior, sigma_prior, log = TRUE) +
    + dnorm(parameters[2], mu_prior_2, sigma_prior_2, log = TRUE)
  return(-l1)
}

logLIkelihoodSirHMC_2 <- function(parameters){
  res <- deSolve::ode(y = c(S_0 / N, I_0 / N, R_0, rep(0, 6)),
                      times = 0:14, SIR_2, parameters, method = "rk4")
  l1 <- 0
  for (i in 1:14) {
    l1 <- l1 + dpois(datos$in_bed[i], res[i + 1, 3] * N, log = TRUE)
  }
  l1 <- l1 + dnorm(parameters[1], mu_prior, sigma_prior, log = TRUE) +
```

```
    + dnorm(parameters[2], mu_prior_2, sigma_prior_2, log = TRUE)
  return(list(l1, res))
}

N <- 763

S_0 <- 762
I_0 <- 1
R_0 <- 0

mu_prior = 1.9
sigma_prior = 0.1

mu_prior_2 = 0.5
sigma_prior_2 = 0.1

maxPosLikelihoodHMC <- optim(c(2, 0.2), logLIkelihoodSirHMC)

epsilon <- 1e-3
L_leapfrog <- 1

m <- diag(c(1,1))

rho <- MASS::mvrnorm(1, c(0,0), m)

theta <- maxPosLikelihoodHMC$par[1]
gamma <- maxPosLikelihoodHMC$par[2]

H <- -logLIkelihoodSirHMC_2(c(theta, gamma))[[1]] + (1 / 2) *
        t(rho)%*% solve(m) %*% rho

res <- deSolve::ode(y = c(S_0 / N, I_0 / N, R_0, rep(0, 6)),
        times = 0:14, SIR_2, c(theta, gamma), method = "rk4",
        rtol = 1e-8, atol = 1e-8)


res1 <- leapfrog(c(theta, gamma), rho, m, epsilon, L_leapfrog)
gradientSIR(c(theta, gamma))

#####################################################################
burn_in <- 1000
N_M <- 20000
acceptanceRate <- 0
pois=c()
traceHMC <- array(dim = c(N_M - burn_in, 2))
plot(1:14, datos$in_bed, type = "l", ylim = c(0, 300), lwd = 2,
```

```
        xlab="Días", ylab="Infectados diarios")
for (s in 1:N_M) {
  res1 <- leapfrog(c(theta, gamma), rho, m, epsilon, L_leapfrog)
  theta1 <- res1[1]
  gamma1 <- res1[2]
  rho1 <- res1[3:4]
  if(s > burn_in){
    traceHMC[s - burn_in, ] <- c(theta, gamma)
  }
  res.full <- logLIkelihoodSirHMC_2(c(theta1, gamma1))
  l1 <- res.full[[1]]
  res <- res.full[[2]]


  H1 <- -l1 + (1 / 2) * t(rho1)%*% solve(m) %*% rho1
  # El HMC rechaza en base al Hamiltoniano (la energia)
  if(runif(1) < min(1, exp(-H1 + H))){
    H <- H1
    theta <- theta1
    gamma <- gamma1
    print(c(s, theta, gamma))
    if (s > burn_in){
      acceptanceRate <- acceptanceRate + 1
      lines(res[2:15, 1], res[2:15, 3] * N, lwd = 2, col = "blue")
      pois=cbind(pois,rpois(15,res[,3]*N))
    }
  }
  rho <- MASS::mvrnorm(1, c(0,0), m)
}
```

Anexo VIII: Código para simular modelos SIR por HMC para datos de CyL

```
library(deSolve)
library(outbreaks)

datos=read.csv("C:/Users/marco/Desktop/pincho/KINGSTON/5º/
        TFGest/datosCovidCyL.csv",sep=";")

SIR_2 <- function(t, y, params){
  beta <- params[1]
  gamma <- params[2]
  S <- y[1]
  I <- y[2]
  R <- y[3]
  s_beta_1 <- y[4]
  s_beta_2  <- y[5]
  s_beta_3 <- y[6]
  s_gamma_1 <- y[7]
  s_gamma_2  <- y[8]
  s_gamma_3 <- y[9]

  dS <- - beta * S * I
  dI <- beta * S * I - gamma * I
  dR <- gamma * I
  res_1 <- c(dS, dI, dR)

  M <- matrix(c(-beta * I , -beta * S , 0 ,
                beta * I , beta * S - gamma , 0 ,
                0 , gamma , 0), nrow = 3, byrow = T)


  res_2 <- M %*% c(s_beta_1, s_beta_2, s_beta_3) + c(-I * S, I * S, 0)
  res_3 <- M %*% c(s_gamma_1, s_gamma_2, s_gamma_3) + c(0, -I, I)
```

```
  list(c(res_1, res_2, res_3))
}


gradientSIR <- function(theta){
  res_Grad <- deSolve::ode(y = c(S_0 / N, I_0 / N, R_0, rep(0, 6)),
                           times = 0:14, SIR_2, theta, method = "rk4",
                           rtol = 1e-8, atol = 1e-8)
  grad1 <- sum((datos$in_bed / (res_Grad[2:15, 3] * N) - 1) *
        res_Grad[1:14, 6] * N)
  grad1 <- grad1 - (theta[1] - mu_prior) /  sigma_prior^2
  grad2 <- sum((datos$in_bed / (res_Grad[2:15, 3] * N) - 1) *
        res_Grad[1:14, 9] * N)
  grad1 <- grad2 - (theta[2] - mu_prior_2) /  sigma_prior_2^2
  return(c(grad1, grad2))
}

leapfrog <- function(theta, rho, m, epsilon, L_leapfrog){
  for(kk in 1:L_leapfrog){
    gradient <- gradientSIR(theta)
    rho <- rho + 1/2 * epsilon * gradient
    theta <- theta + epsilon * solve(m) %*% rho
    rho <- rho + 1/2 * epsilon * gradient
  }
  return(c(theta, rho))
}

logLIkelihoodSirHMC <- function(parameters){
  res <- deSolve::ode(y = c(S_0 / N, I_0 / N, R_0, rep(0, 6)),
                      times = 0:137, SIR_2, parameters, method = "rk4")
  C <- c(1 / N)
  for (i in 2:137) {
    C <- rbind(C, res[i - 1,2] - res[i,2])
  }
  l1 <- 0
  for (i in 1:137) {
    l1 <- l1 + dpois(datos$Cases[i], C[i] * N, log = TRUE)
  }
  l1 <- l1 + dnorm(parameters[1], mu_prior, sigma_prior, log = TRUE) +
        dnorm(parameters[2], mu_prior_2, sigma_prior_2, log = TRUE)
  return(-l1)
}


logLIkelihoodSirHMC_2 <- function(parameters){
  res <- deSolve::ode(y = c(S_0 / N, I_0 / N, R_0, rep(0, 6)),
```

```
                      times = 0:137, SIR_2, parameters, method = "rk4")
  C <- c(1 / N)
  for (i in 2:137) {
    C <- rbind(C, res[i - 1,2] - res[i,2])
  }
  l1 <- 0
  for (i in 1:137) {
    l1 <- l1 + dpois(datos$Cases[i], C[i] * N, log = TRUE)
  }
  l1 <- l1 + dnorm(parameters[1], mu_prior, sigma_prior, log = TRUE) +
        dnorm(parameters[2], mu_prior_2, sigma_prior_2, log = TRUE)
  return(list(l1, res))
}

N <- 2394918

S_0 <- 2394917
I_0 <- 1
R_0 <- 0

mu_prior = 27
sigma_prior = 1

mu_prior_2 = 27
sigma_prior_2 = 1

maxPosLikelihoodHMC <- optim(c(2, 2), logLIkelihoodSirHMC)

epsilon <- 1e-3
L_leapfrog <- 1

m <- diag(c(1,1))

rho <- MASS::mvrnorm(1, c(0,0), m)

theta <- maxPosLikelihoodHMC$par[1]
gamma <- maxPosLikelihoodHMC$par[2]

H <- -logLIkelihoodSirHMC_2(c(theta, gamma))[[1]] + (1 / 2) *
        t(rho)%*% solve(m) %*% rho

res <- deSolve::ode(y = c(S_0 / N, I_0 / N, R_0, rep(0, 6)),
        times = 0:137, SIR_2, c(theta, gamma), method = "rk4",
        rtol = 1e-8, atol = 1e-8)
```

```
res1 <- leapfrog(c(theta, gamma), rho, m, epsilon, L_leapfrog)
gradientSIR(c(theta, gamma))

#####################################################################
burn_in <- 000
N_M <- 20000
acceptanceRate <- 0
traceHMC <- array(dim = c(N_M - burn_in, 2))
plot(1:137, datos$Cases, type = "l", ylim = c(0, 1000), lwd = 2,
        xlab="Días", ylab="Infectados diarios")
pois=c()
for (s in 1:N_M) {
  res1 <- leapfrog(c(theta, gamma), rho, m, epsilon, L_leapfrog)
  theta1 <- res1[1]
  gamma1 <- res1[2]
  rho1 <- res1[3:4]
  if(s > burn_in){
    traceHMC[s - burn_in, ] <- c(theta, gamma)
  }
  res.full <- logLIkelihoodSirHMC_2(c(theta1, gamma1))
  l1 <- res.full[[1]]
  res <- res.full[[2]]

  H1 <- -l1 + (1 / 2) * t(rho1)%*% solve(m) %*% rho1
  if(runif(1) < min(1, exp(-H1 + H))){
    H <- H1
    theta <- theta1
    gamma <- gamma1
    print(c(s, theta, gamma))
    if (s > burn_in){
      C <- c(1 / N)
      for (i in 2:137) {
        C <- rbind(C, res[i - 1,2] - res[i,2])
      }
      acceptanceRate <- acceptanceRate + 1
      lines(res[2:138, 1], C * N, lwd = 2, col = "blue")
      pois=cbind(pois,C)
    }
  }
  rho <- MASS::mvrnorm(1, c(0,0), m)
}
```

Bibliografía

[1] William Ogilvy Kermack and Anderson G McKendrick. Contributions to the mathematical theory of epidemics: The problem of endemicity. *Proceedings of the Royal Society of London. Series A, containing papers of a mathematical and physical character*, 138(834):55–83, 1932.

[2] William Ogilvy Kermack and Anderson G McKendrick. Contributions to the mathematical theory of epidemics: Further studies of the problem of endemicity. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 141(843):94–122, 1933.

[3] Isidro Alfredo Abelló Ugalde, Raúl Guinovart Díaz, and Wilfredo Morales Lezca. El modelo sir básico y políticas antiepidémicas de salud pública para la covid-19 en cuba. *Revista Cubana de Salud Pública*, 46:e2597, 2021.

[4] Jorge Homero Wilches Visbal and Midian Clara Castillo Pedraza. Aproximación matemática del modelo epidemiológico sir para la comprensión de las medidas de contención contra la covid-19. *Rev. esp. salud pública*, pages 0–0, 2020.

[5] N Metropolis. Beginning of the monte carlo method. *Stanislaw Ulam*, 1987.

[6] N Metropolis and S Ulam. Monte carlo method-a popular description. Technical report, 1949.

[7] AA Markov. Extension of the limit theorems of probability theory to a sum of variables connected in a chain, the notes of the imperial academy of sciences of st. *Petersburg VIII Series, Physio-Mathematical College*, 22(9), 1907.

[8] Ronald A Howard. *Dynamic probabilistic systems: Markov models*, volume 1. Courier Corporation, 2012.

[9] Ricardo Medel Esquivel, Isidro Gómez-Vargas, J Alberto Vázquez, and Ricardo García Salcedo. An introduction to markov chain monte carlo. *Boletın de Estadıstica e Investigación Operativa*, 2019.

[10] Ryan Wang. Markov chain monte carlo, recuperada: mar 2022. URL `https://www.math.uchicago.edu/~may/VIGRE/VIGRE2010/REUPapers/Wang.pdf`.

[11] Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of markov chain monte carlo*. CRC press, 2011.

[12] T Wiecki. Mcmc sampling for dummies. 2015. URL https://twiecki.io/blog/2015/11/10/mcmc-sampling/.

[13] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.

[14] David B Hitchcock. A history of the metropolis–hastings algorithm. *The American Statistician*, 57(4):254–257, 2003.

[15] Gareth O Roberts and Jeffrey S Rosenthal. General state space markov chains and mcmc algorithms. *Probability surveys*, 1:20–71, 2004.

[16] Walter R Gilks, Sylvia Richardson, and David Spiegelhalter. *Markov chain Monte Carlo in practice*. CRC press, 1995.

[17] Introduction to markov chain monte carlo, recuperada: mar 2022. URL http://galton.uchicago.edu/~lalley/Courses/313/ProppWilson.pdf.

[18] George Casella and Edward I George. Explaining the gibbs sampler. 1992.

[19] Se Yoon Lee. Gibbs sampler and coordinate ascent variational inference: A set-theoretical review. *Communications in Statistics-Theory and Methods*, 2021.

[20] Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 1995.

[21] G. Undersen. Gibbs sampling is a special case of metropolis–hastings. 2020. URL https://gregorygundersen.com/blog/2020/02/23/gibbs-sampling/.

[22] Berni J Alder and Thomas Everett Wainwright. Studies in molecular dynamics. i. general method. *The Journal of Chemical Physics*, 31(2):459–466, 1959.

[23] Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics letters B*, 195(2):216–222, 1987.

[24] David JC MacKay, David JC Mac Kay, et al. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[25] Michael Betancourt. A conceptual introduction to hamiltonian monte carlo. *arXiv preprint arXiv:1701.02434*, 2017.

[26] dustinstansbury. Mcmc: Hamiltonian monte carlo, nov 2012. URL https://theclevermachine.wordpress.com/tag/canonical-distribution/.

[27] Nisheeth K Vishnoi. An introduction to hamiltonian monte carlo method for sampling. *arXiv preprint arXiv:2108.12107*, 2021.

[28] Stan Development Team. *Stan reference manual*. Stan, 2.29 edition.

[29] GH EvAs. Some arithmetical considerations of the progress of epidemics. *Trans. Epidem. Soc. of London*, 3:551, 1866.

[30] John Brownlee. Certain considerations on the causation and course of epidemics. *Proceedings of the Royal Society of Medicine*, 2(Epidem_State_Med):243–258, 1909.

[31] John Brownlee. Statistical studies in immunity: the theory of an epidemic. *Proceedings of the Royal Society of Edinburgh*, 26(1):484–521, 1906.

[32] Ronald Ross. An application of the theory of probabilities to the study of a priori pathometry.—part i. *Proceedings of the Royal Society of London. Series A, Containing papers of a mathematical and physical character*, 92(638):204–230, 1916.

[33] William Ogilvy Kermack and Anderson G McKendrick. A contribution to the mathematical theory of epidemics. *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*, 115(772):700–721, 1927.

[34] Michael Y Li. *An introduction to mathematical modeling of infectious diseases*, volume 2. Springer, 2018.

[35] Edward A Bender. *An introduction to mathematical modeling*. Courier Corporation, 2000.

[36] Adriano Henrique Danhoni Neves and Denner Serafim Vieira. A review upon compartmental models in epidemiology.

[37] Howard Howie Weiss. The sir model and the foundations of public health. *Materials matematics*, pages 0001–17, 2013.

[38] anonymous. Influenza in a boarding school. *British Medical Journal*, mar 4 1978.

[39] Yingbo Ma, Vaibhav Dixit, Michael J Innes, Xingjian Guo, and Chris Rackauckas. A comparison of automatic differentiation and continuous sensitivity analysis for derivatives of differential equation solutions. In *2021 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–9. IEEE, 2021.

[40] Leon Arriola and James M Hyman. Sensitivity analysis for uncertainty quantification in mathematical models. In *Mathematical and statistical estimation approaches in epidemiology*, pages 195–247. Springer, 2009.