Cómputo científico para probabilidad y estadística. Tarea 8. MCMC: MH con Kerneles Híbridos y Gibbs Sampler

Juan Esaul González Rangel

Noviembre 2023

1. Aplique el algoritmo de Metropolis-Hastings considerando como función objetivo la distribución normal bivariada

$$f_{X_1,X_2}(\bar{x}) = \frac{1}{2\pi}|\Sigma|^{-1/2}\exp\left\{-\frac{1}{2}(\bar{x}-\mu)'\Sigma^{-1}(\bar{x}-\mu)\right\}$$

donde,

$$\mu = \begin{pmatrix}\mu_1\\ \mu_2\end{pmatrix} \quad \Sigma = \begin{pmatrix}\sigma_1^2 & \rho\sigma_1\sigma_2\\ \rho\sigma_1\sigma_2 & \sigma_2^2\end{pmatrix}$$

Así, se tienen las siguientes distribuciones condicionales:

$$X_1|X_2 = x_2 \sim N\left(\mu_1 + \rho\frac{\sigma_1}{\sigma_2}(x_2-\mu_2), \sigma_1^2(1-\rho^2)\right)$$

$$X_2|X_1 = x_1 \sim N\left(\mu_2 + \rho\frac{\sigma_2}{\sigma_1}(x_1-\mu_1), \sigma_2^2(1-\rho^2)\right)$$

Considere las siguientes propuestas:

$$q_1((x_1', x_2')|(x_1, x_2)) = f_{X_1|X_2}(x_1'|x_2)\mathbb{1}_{(x_2'=x_2)}$$

$$q_2((x_1', x_2')|(x_1, x_2)) = f_{X_2|X_1}(x_2'|x_1)\mathbb{1}_{(x_1'=x_1)}$$

A partir del algoritmo MH usando Kerneles híbridos simule valores de la distribución normal bivariada, fijando $\sigma_1 = \sigma_2 = 1$, considere los casos $\rho = 0.8$ y $\rho = 0.95$[1].

Solución. Notemos que los kerneles de transición son la densidad de un parámetro condicionada al valor que toman todos los otros parámetros, por lo que el cociente de Metropolis-Hastings es exactamente 1 en ambos, y por lo tanto la transición a la propuesta simepre se da. Para evitar perder la aperiodicidad fuerte de la cadena, implementamos un tercer kernel de transición con una probabilidad muy baja que simplemente mantenga la posición actual. El kernel híbrido tiene la siguiente forma,

$$K = \sum_{i=1}^{3} w_i K_i = 0.4999\left(f_{X_1|X_2}(x_1'|x_2)\mathbb{1}_{(x_2'=x_2)} + f_{X_2|X_1}(x_2'|x_1)\mathbb{1}_{(x_1'=x_1)}\right) + 0.0002(\mathbb{1}_{(x_1'=x_1, x_2'=x_2)}).$$

Al dejar fijo y tener aperiodicidad fuerte nos aseguramos de que la cadena es ergódica y garantizamos la convergencia.

Para la implementación de este algoritmo de MCMC, se definió la función `mhbivar` que toma los siguientes parámetros;

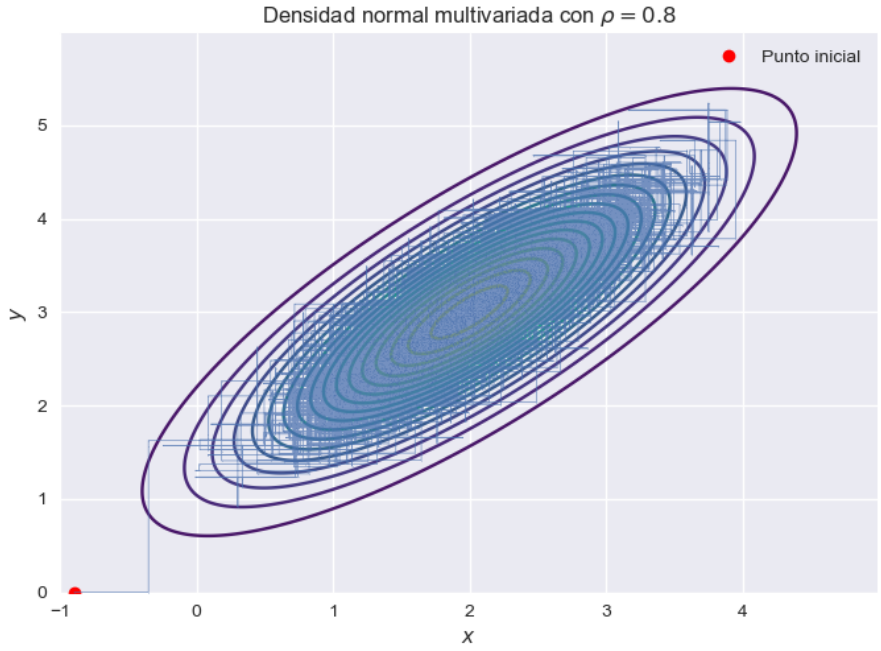
[1]Ver la tesis de Cricelio Montesinos para una explicación más extensa del Gibbs, Montesinos, C (2016) "Distribución de Direcciones en el Gibbs Sampler Generalizad", MSc Dissertation, CIMAT. `https://www.cimat.mx/es/Tesis_digitales/`. También vean la Enciclopedia de Estadística de Wiley, la entrada de *Gibbs Sampler*: `https://www.cimat.mx/~jac/2016WileytStatsRef_GibbsSampling.pdf`.

- `iterations`: Cantidad de veces que se corre el algoritmo.
- `mu`: Vector de longitud 2 que representa la media de la densidad objetivo, por defecto es `[0,0]`.
- `rho`: Coeficiente de correlación de las componentes, por defecto es 0.8.
- `starting_point`: Punto donde iniciar la cadena, por defecto es `[0,0]`.

Se implementó el algoritmo para simular las siguientes densidades normales multivariadas:

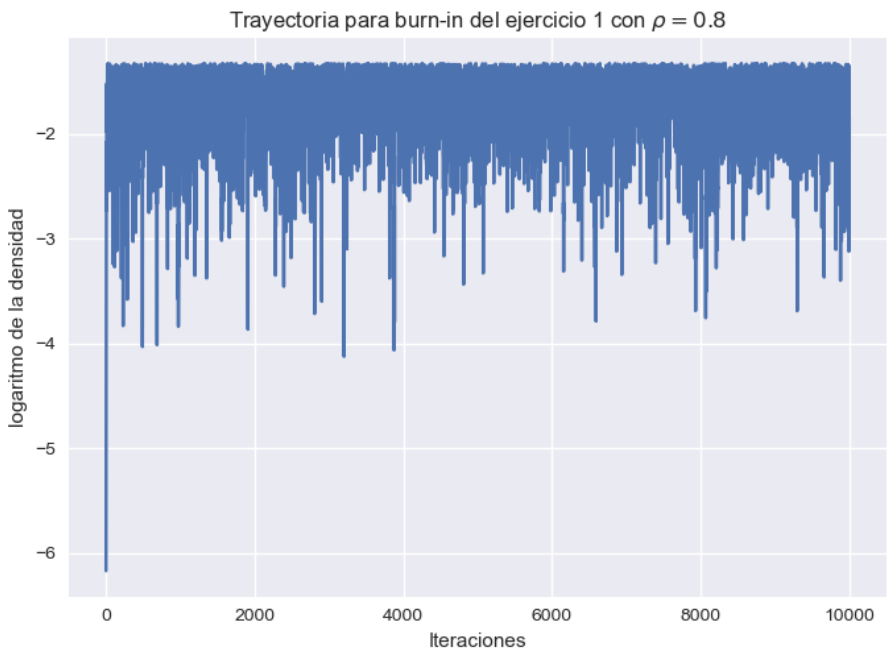
No.	Media	Coeficiente de correlación	σ_1	σ_2
1	$(2, 3)$	$\rho = 0.8$	1	1
2	$(-3, 2)$	$\rho = 0.95$	1	1

En la siguiente imagen se muestran las curvas de nivel de la primera densidad objetivo junto a 10,000 simulaciones realizadas con este kernel híbrido:



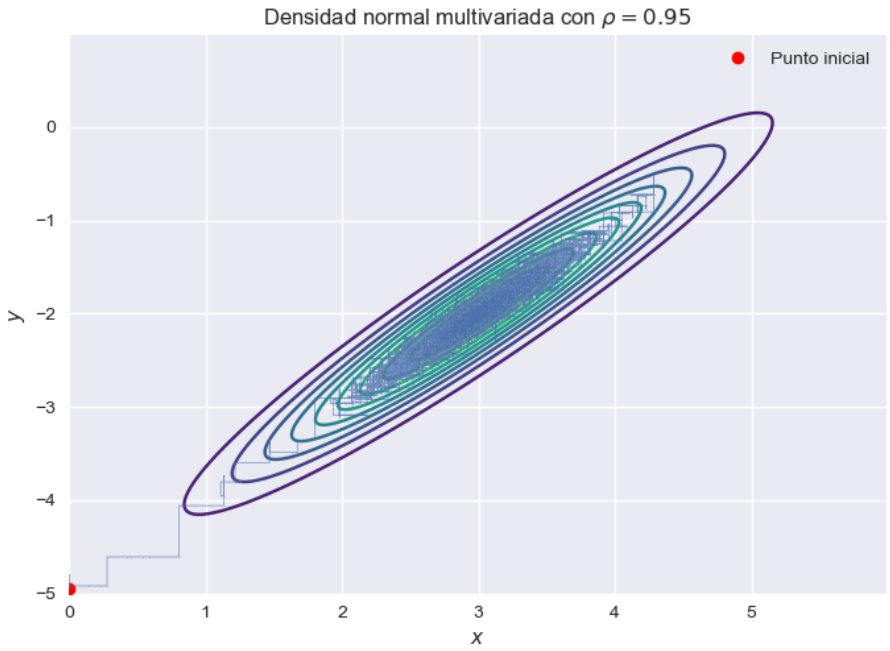
Podemos notar que el kernel sólo agrega nuevos puntos en las direcciones canónicas del plano, puesto que el muestreo de Gibbs estándar utiliza muestreo desde las direcciones marginales. Además, a pesar de haber comenzado en un punto alejado de la media, el algoritmo converge rápidamente y tenemos un muestreo efectivo desde las regiones con más masa de la distribución.

Para mejorar el muestreo, podemos retirar las primeras iteraciones de la cadena, y para esto usamos la siguiente gráfica de burn-in en la que comparamos el logaritmo de la densidad evaluada en X_t con las iteraciones de la cadena.



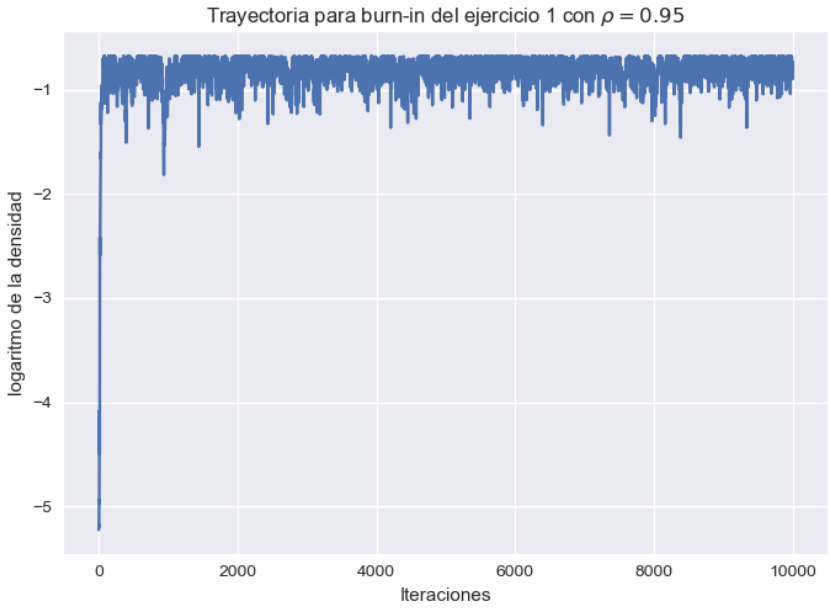
En la gráfica notamos que en pocas iteraciones se estabiliza el logaritmo de la densidad evaluada en la cadena, por lo que la cadena parece converger de manera rápida. Basta eliminar los primeros 100 puntos para tener un comportamiento uniforme.

Para la segunda normal, tenemos la siguiente gráfica de su muestreo,



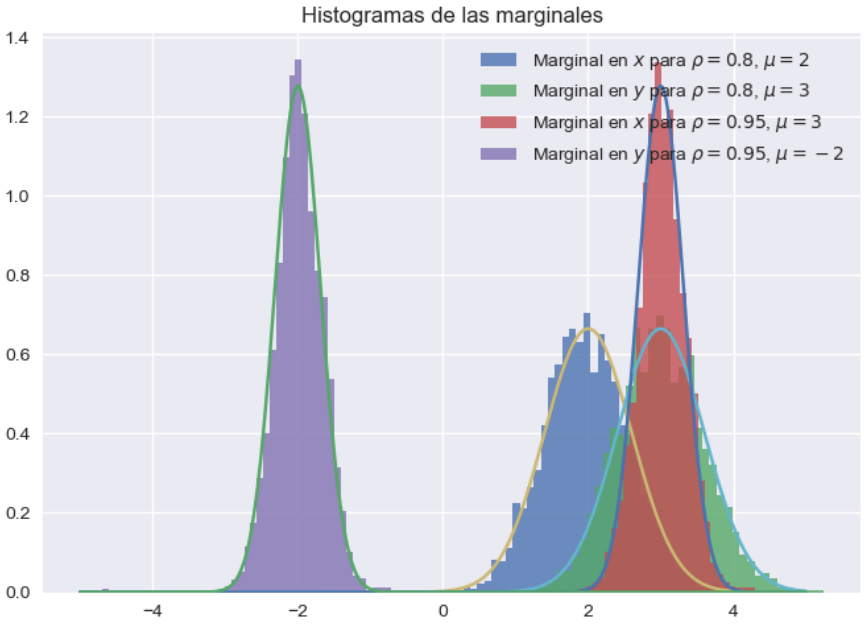
Al igual que en el caso anterior, observamos que punto inicial se encuentra relativamente alejado de la media de la dsitribución, pero al avanzar nos acercamos a muestrear más rápidamente de los puntos donde la distribución acumula más masa.

La gráfica de burn-in para este muestreo es la siguiente;



Se tiene un comportamiento menos uniforme que en el primer muestreo, y esto lo notamos en el hecho de que hay más diferencia entre los primeros valores del logaritmo de la densidad, pero en pocas iteraciones el comportamiento se estabiliza. En este caso también basta eliminar los primeros 100 puntos para tener un comportamiento uniforme y por lo tanto un muestreo más adecuado de la distribución.

Como estamos muestreando de una distribución conocida, es sencillo encontrar las marginales de nuestra normal multivariada. En la siguiente figura se incluyen los histogramas de las marginales de cada una de las normales bivariadas muestreadas y se comparan con la densidad real.



En las gráfinas notamos que el ajuste de los histogramas a la densidad real es bastante bueno.

□

2. Considere los tiempos de falla $t_1, \ldots, t_n$ con distribución *Weibull*(α, λ):

$$f(t_i|\alpha, \lambda) = \alpha\lambda t_i^{\alpha-1} e^{-t_i^{\alpha}\lambda}$$

Se asumen como a priori $\alpha \sim \exp(c)$ y $\lambda|\alpha \sim$Gama(α, b), por lo tanto, $f(\alpha, \lambda) = f(\lambda|\alpha)f(\alpha)$[2]. Así, para la distribución posterior se tiene:

$$f(\alpha, \lambda|\bar{t}) \propto f(\bar{t}|\alpha, \lambda)f(\alpha, \lambda)$$

A partir del algoritmo MH usando Kerneles híbridos simule valores de la distribución posterior $f(\alpha, \lambda|\bar{t})$, considerando las siguientes propuestas:

Propuesta 1:

$$\lambda_p|\alpha, \bar{t} \sim Gama\left(\alpha + n, b + \sum_{i=1}^{n} t_i^{\alpha}\right) \quad \text{y dejando } \alpha \text{ fijo.}$$

Propuesta 2:

$$\alpha_p|\lambda, \bar{t} \sim Gama(n+1, -\log(b) - \log(r_1) + c), \text{ con } r_1 = \prod_{i=1}^{n} t_i \text{ y dejando } \lambda \text{ fijo.}$$

Propuesta 3:

$\alpha_p \sim \exp(c)$ y $\lambda_p|\alpha_p \sim Gama(\alpha_p, b)$.

Propuesta 4 (RWMH):

$\alpha_p = \alpha + \varepsilon$, con $\varepsilon \sim N(0, \sigma)$ y dejando λ fijo.

Simular datos usando $\alpha = 1$ y $\lambda = 1$ con $n = 20$. Para la a priori usar $c = 1$ y $b = 1$.

Solución. Para la implementación de este algoritmo MCMC es necesario notar ciertos aspectos de cada una de las propuestas que pueden simplificar los cálculos o resultar problemáticos. Para la propuesta 1, vemos que se trata de un kernel de Gibbs al estar muestreando desde la condicional de un parámetro dados todos los demás, esto hace que sepamos de antemano que el cociente de Metropolis-Hastings para esta propuesta es 1, y por lo tanto siempre se acepta la transición.

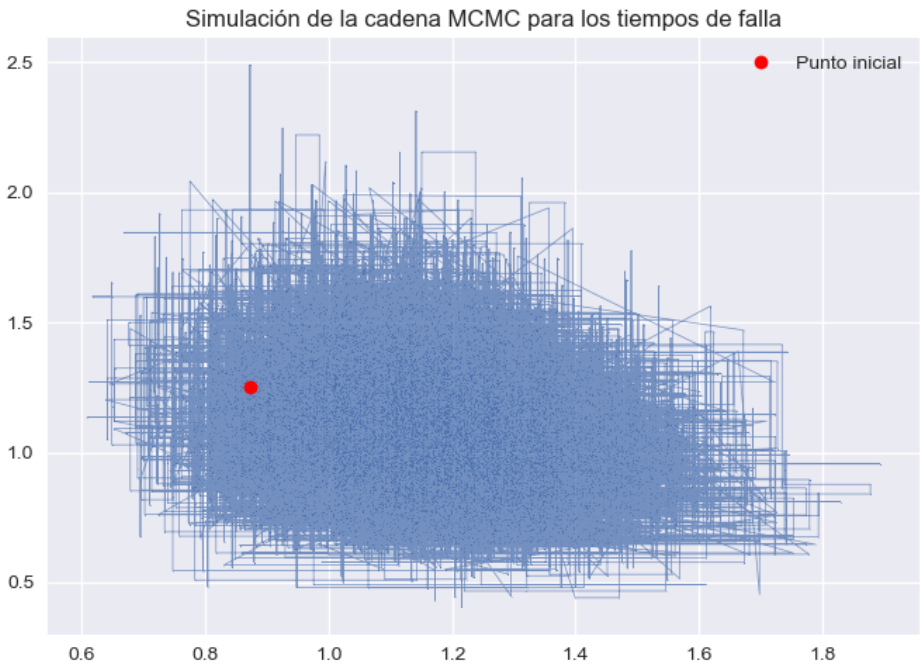
Para la propuesta 2, el término $-\log(b) - \log(r_1) + c$ puede ser negativo dependiendo de r_1, y por lo tanto podría darse el caso de que se estuviera tratando de simular desde una distribución gamma con parámetro negativo, lo cuál no tiene sentido. Para evitar este problema, incluimos un condicional, de manera que si el término resulta negativo, no se proponga la transición y de esta manera el segundo kernel nos sirva como una manera de asegurar la aperiodicidad fuerte de la cadena.

En la cuarta propuesta tenemos que la transición es simétrica, pues se trata de un RWMH, con lo que es necesario únicamente comparar las densidades objetivo $f(y)$ y $f(x)$ para evaluar si se acepta o rechaza la propuesta.

Al considerar estos aspectos, se implementó la función `MHposter` que recibe argumentos como el número de iteraciones, el punto inicial y los parámetros de las distribuciones a usar.

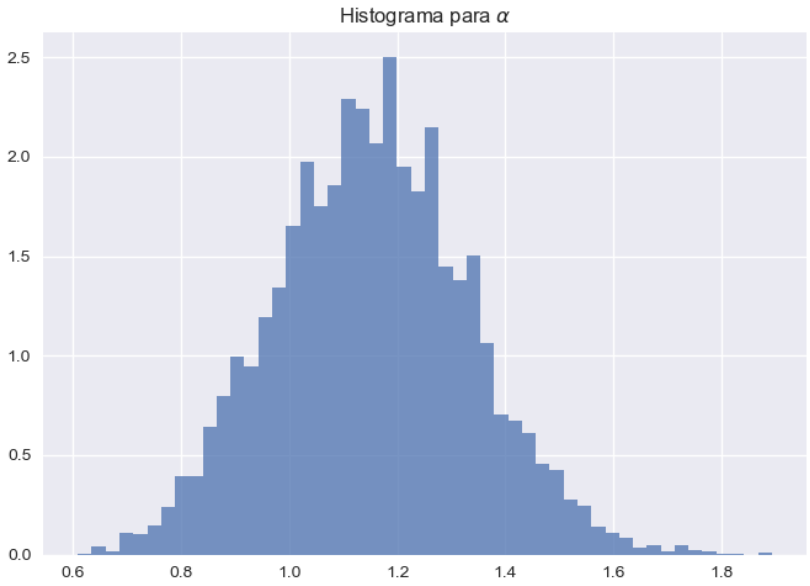
La siguiente imagen muestra la trayectoria de 10,000 simulaciones de la objetivo usando este kernel híbrido con igual peso a cada propuesta,

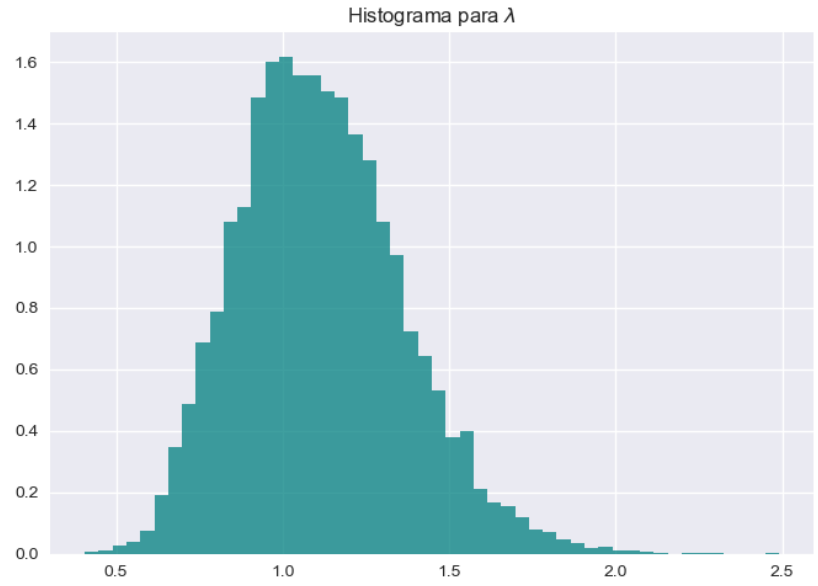
[2]Este ejemplo aparece en Kundu, D. (2008), "Bayesian Inference and Life Testing Plan for the Weibull Distribution in Presence of Progressive Censoring", Technometrics, 50(2), 144–154.



Algo que podemos notar es que la mayor cantidad de líneas que se ven están en dirección horizontal o vertical, lo que significa que la mayor parte de las transiciones vienen de las propuestas que mueven únicamente al parámetro α o al λ cada vez.

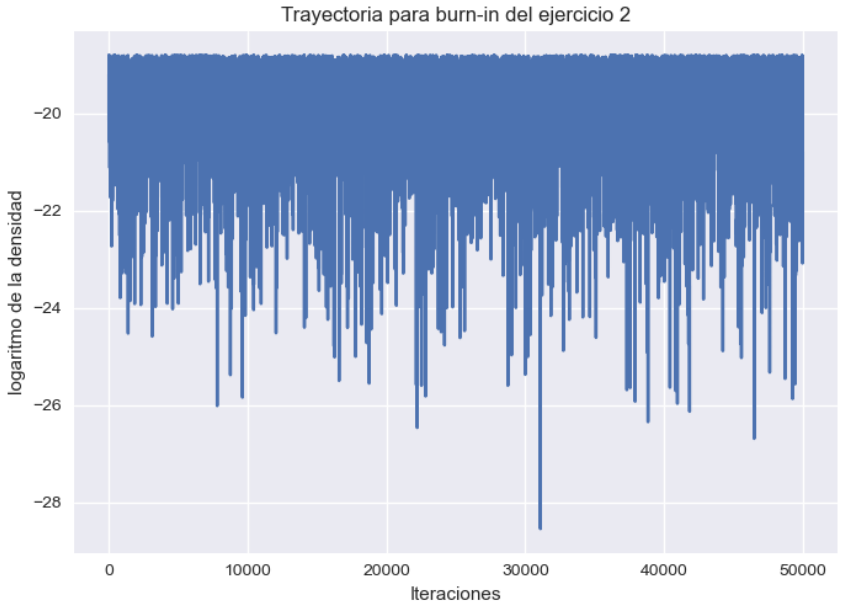
Los histogramas de las marginales de α y λ se obervan en las siguientes figuras.





Como sabemos que los datos con los que formamos la posterior vienen de una distribución Weibull$(1, 1)$, es de interés conocer cuál sería el estimador de los parámetros si usamos este muestreo de MCMC. El estimador más sencillo de usar es la media de los datos simulados, que por la Ley de Grandes Números aproxima al estimador de Bayes. El vector de medias de este muestreo es $(1.1536, 1.1136)$, que aproxima bien al original $(1, 1)$, considerando que sólo se tienen 20 observaciones de los datos.

En esta cadena también es útil conocer el burn-in para saber cuántos datos iniciales deberíamos descartar para asegurar que muestreamos desde la objetivo. La gráfica de iteraciones contra $\log(f(X_t))$ se muestra a continuación.



El comprtamiento del logaritmo de la densidad es bastante uniforme, por lo que podemos considerar que la cadena muestrea de la dsitribución desde el principio. □

3. Considere el ejemplo referente al número de fallas de bombas de agua en una central nuclear[3], donde p_i representa el número de fallas en el tiempo de operación t_i, con $i = 1, \ldots, n$.

   Se considera el modelo $p_i \sim$Poisson$(\lambda_i t_i)$, (las λ_i son independientes entre si), con distribuciones a priori $\lambda_i|\beta \sim Gama(\alpha, \beta)$ y $\beta \sim Gama(\gamma, \delta)$, por lo tanto:

$$f(\lambda_1, \ldots, \lambda_n, \beta) = f(\lambda_1|\beta) f(\lambda_2|\beta) \ldots f(\lambda_n|\beta) f(\beta)$$

   Para la distribución posterior se tiene:

$$f(\lambda_1, \ldots, \lambda_n, \beta|\bar{p}) \propto L(\bar{p}, \bar{\lambda}, \beta) f(\lambda_1, \ldots, \lambda_n, \beta)$$

   Simule valores de la distribución posterior $f(\lambda_1, \ldots, \lambda_n, \beta|\bar{p})$, usando un kernel híbrido, considerando las propuestas:

$$\lambda_i|\bar{\lambda}_{-i}, \beta, \bar{t} \sim Gama(p_i + \alpha, \beta + t_i)$$

Bomba (i)	1	2	3	4	5	6	7	8	9	10
T. de uso (t_i)	94.32	15.72	62.88	125.76	5.24	31.44	1.05	1.05	2.1	10.48
# de fallas (p_i)	5	1	5	14	3	17	1	1	4	22

Tabla 1: Datos de bombas de agua en centrales nucleares (Robert y Casella, p. 385) para el ejemplo 8.3.

$$\beta|\bar{\lambda}, \bar{t} \sim Gama\left(n\alpha + \gamma, \delta + \sum_{i=1}^{n} \lambda_i\right).$$

   Verifique que estas son propuestas Gibbs.

   Use los datos del Cuadro 1 con los parámetros a priori $\alpha = 1.8, \gamma = 0.01$ y $\delta = 1$.

Solución. Mostramos primero que las propuestas dadas son de Gibbs. La verosimilitud es,

$$L(\bar{p}, \bar{\lambda}, \beta) = \prod_{i=1}^{n} \frac{(\lambda_i t_i)^{p_i} e^{-\lambda_i t_i}}{(p_i)!}.$$

La posterior conjunta se obtiene multiplicando la verosimilitud anterior por las dsitribuciones a priori,

$$\begin{aligned} L(\bar{p}, \bar{\lambda}, \beta) f(\lambda_1, \ldots, \lambda_n, \beta) &= L(\bar{p}, \bar{\lambda}, \beta) f(\lambda_1|\beta) f(\lambda_2|\beta) \ldots f(\lambda_n|\beta) f(\beta), \\ &= \prod_{i=1}^{n} \left\{ \frac{(\lambda_i t_i)^{p_i} e^{-\lambda_i t_i}}{(p_i)!} \right\} \frac{\beta e^{-\beta\lambda_1} (\beta\lambda_1)^{\alpha-1}}{\Gamma(\alpha)} \cdots \frac{\beta e^{-\beta\lambda_n} (\beta\lambda_n)^{\alpha-1}}{\Gamma(\alpha)} \frac{\delta e^{-\delta\beta} (\delta\beta)^{\gamma-1}}{\Gamma(\gamma)} \end{aligned}$$

. □

Para encontrar $f_{\lambda_i|\bar{\lambda}_{-i}, \beta, \bar{t}}$, tomamos en la expresión anterior únicamente los términos que dependen de λ_i para un i específico.

$$f_{\lambda_i|\bar{\lambda}_{-i}, \beta, \bar{t}} \propto \left(\frac{(\lambda_i t_i)^{p_i} e^{-\lambda_i t_i}}{p_i!} \right) \left(\frac{\beta e^{-\beta\lambda_i} (\beta\lambda_i)^{\alpha-1}}{\Gamma(\alpha)} \right) \propto \lambda_i^{p_i+\alpha-1} e^{-(\beta+t_i)\lambda_i}$$

[3]Este ejemplo fue usado en el artículo original del Gibbs sampler del Gelfand y Smith (1990). Vea también Norton, R.A., Christen, J.A. y Fox, C. (2017), "Sampling hyperparameters in hierarchical models: improving on Gibbs for high-dimensional latent fields and large data set" Communications in Statistics - Simulation and Computation, `http://dx.doi.org/10.1080/03610918.2017.1353618`

La expresión resultante es el núcleo de una densidad Gamma con parámetros $p_i + \alpha$ y $\beta + t_i$, por lo que conluimos que al agregar las constantes de normalización, tenemos que $\lambda_i|\bar{\lambda}_{-i}, \beta, \bar{t} \sim Gamma(p_i + \alpha, \beta + t_i)$.

De manera completamente análoga, encontramos $f_{\beta|\bar{\lambda},\bar{t}}$,

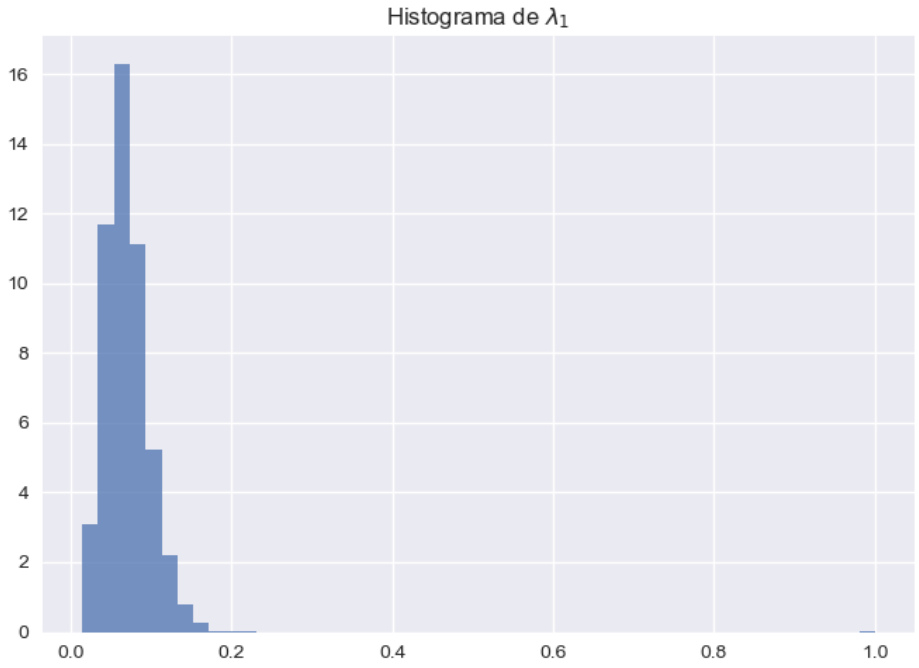
$$\begin{aligned} f_{\beta|\bar{\lambda},\bar{t}} &\propto \prod_{i=1}^{n} \left\{ \frac{(\lambda_i t_i)^{p_i} e^{-\lambda_i t_i}}{(p_i)!} \right\} \frac{\beta e^{-\beta\lambda_1}(\beta\lambda_1)^{\alpha-1}}{\Gamma(\alpha)} \cdots \frac{\beta e^{-\beta\lambda_n}(\beta\lambda_n)^{\alpha-1}}{\Gamma(\alpha)} \frac{\delta e^{-\delta\beta}(\delta\beta)^{\gamma-1}}{\Gamma(\gamma)}, \\ &\propto e^{-(\sum_{i=1}^{n}\lambda_i - \delta)\beta} \beta^{n\alpha+\gamma-1}. \end{aligned}$$

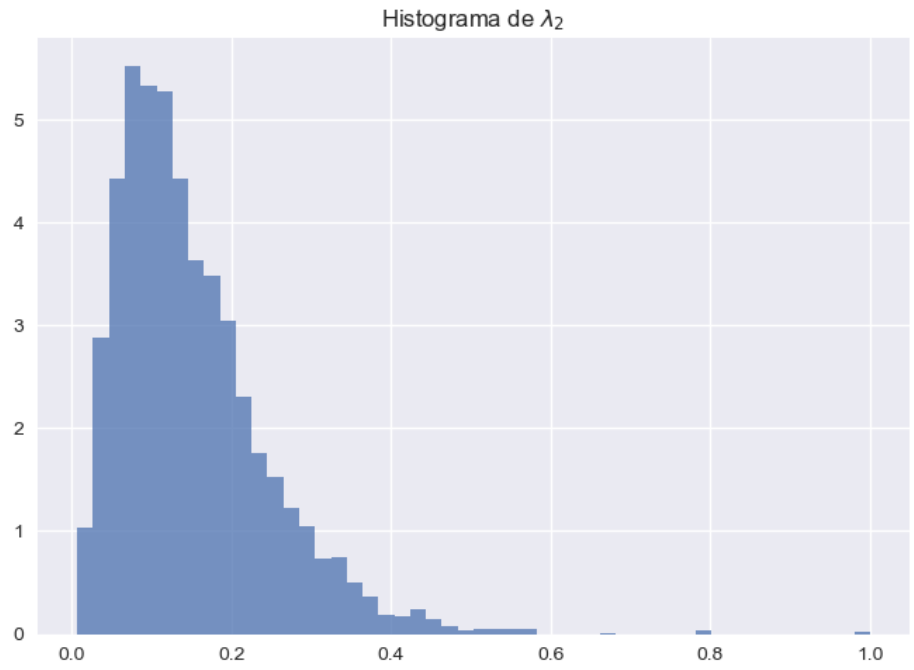
De nuevo, la expresión al final es el núcleo de una densidad $Gamma\left(n\alpha + \gamma, \delta + \sum_{i=1}^{n}\lambda_i\right)$, por lo que concluimos que al agregar las constantes de normalización, esta es la dsitrbiución posterior de β dados los datos y los parámetros λ_i. Entonces las propuestas dadas son de Gibbs y por lo tanto sabemos que la probabilidad de aceptación de la propuesta es 1.

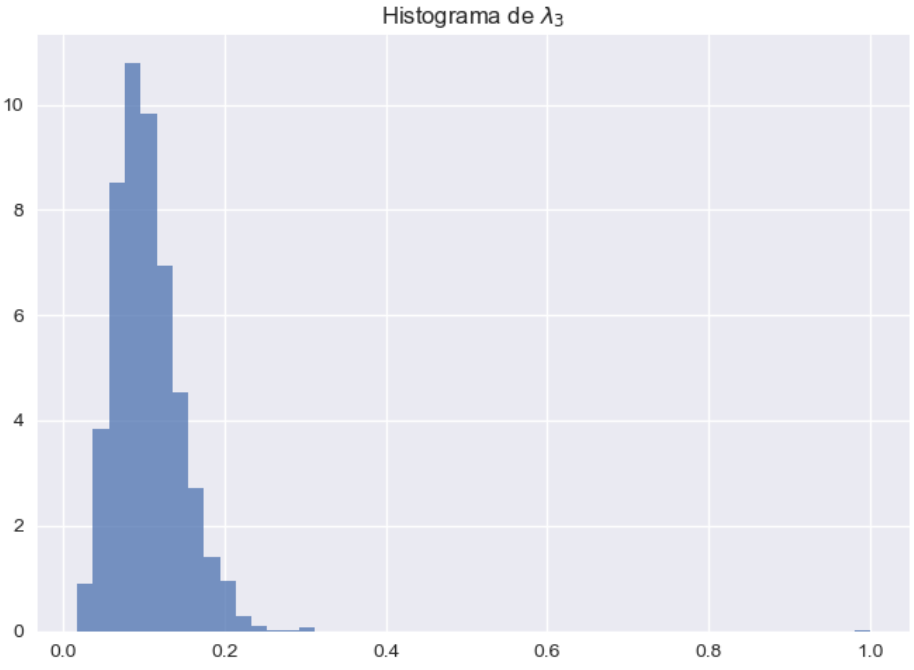
El que los núcleos de transición sean Gibbs nos da la ventaja de que no es necesario caalcular el cociente de Metropolis-Hastings, pero introduce el problema de que es probable perder la aperiodicidad fuerte del proceso. Para recuperar esta propiedad y garantizar que la cadena sea ergódica, introducimos en el código una probabilidad de 0.0001 de que no se elija ningún kernel de transición, y en su lugar se permanezca en el mismo punto.

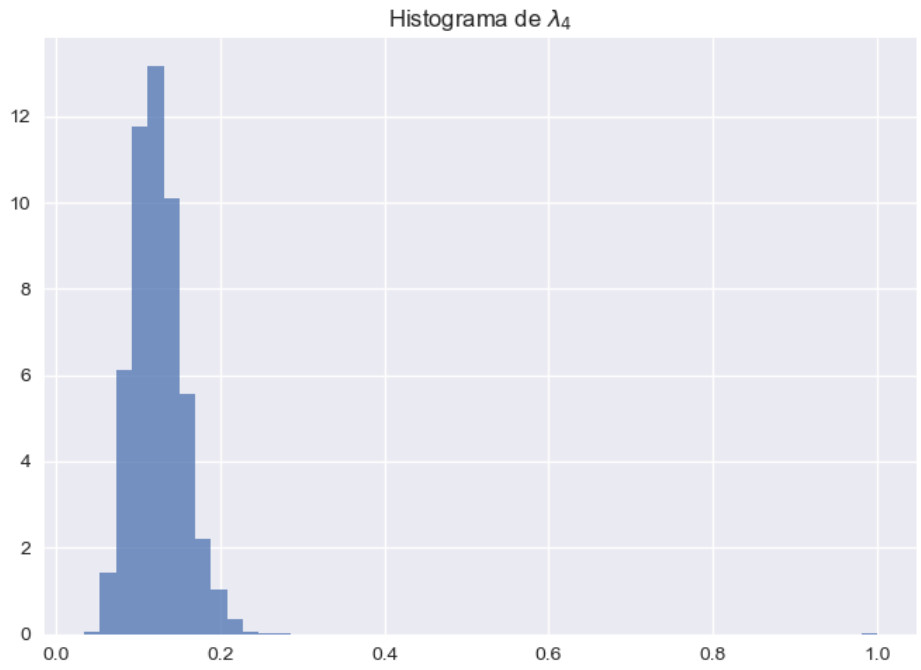
En la función `MHpump` del archivo `Tarea8.py` se implementa este algoritmo y toma como arguentos `iterations` que es la cantidad de veces que se repite el algoritmo y `starting_point` que es el punto inicial de la cadena.

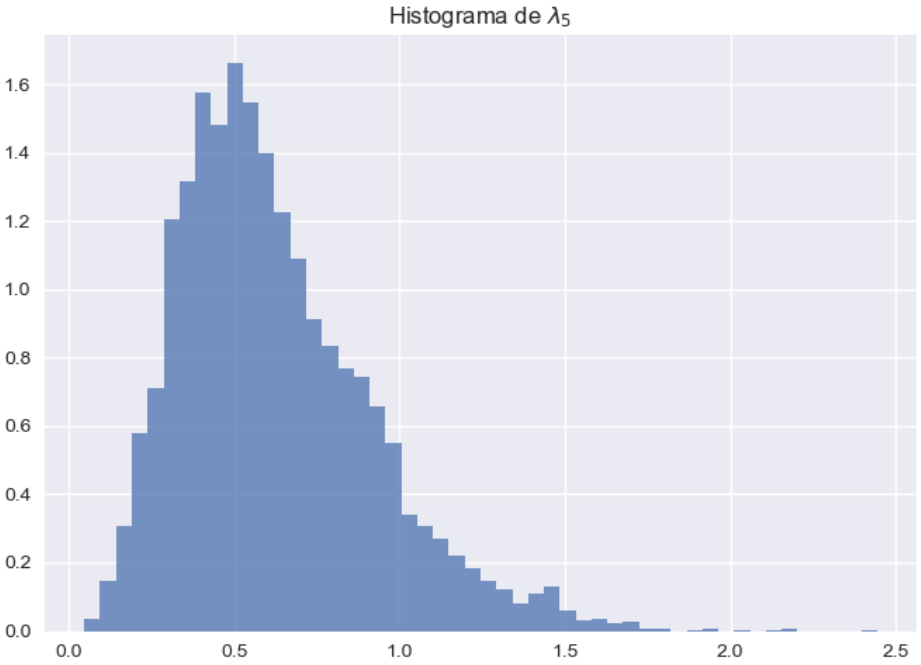
Al realizar 50,000 iteraciones usando el punto inicial $(1, 1, 1, 1, 1, 1, 1, 1, 1, 1)$ encontramos los siguientes histogramas para cada uno de los parámetros,

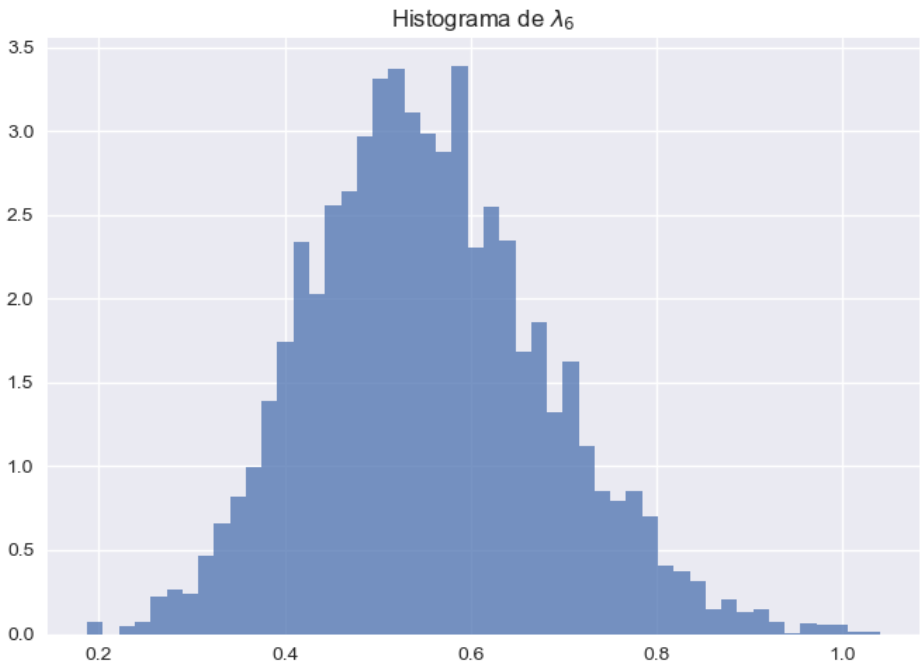


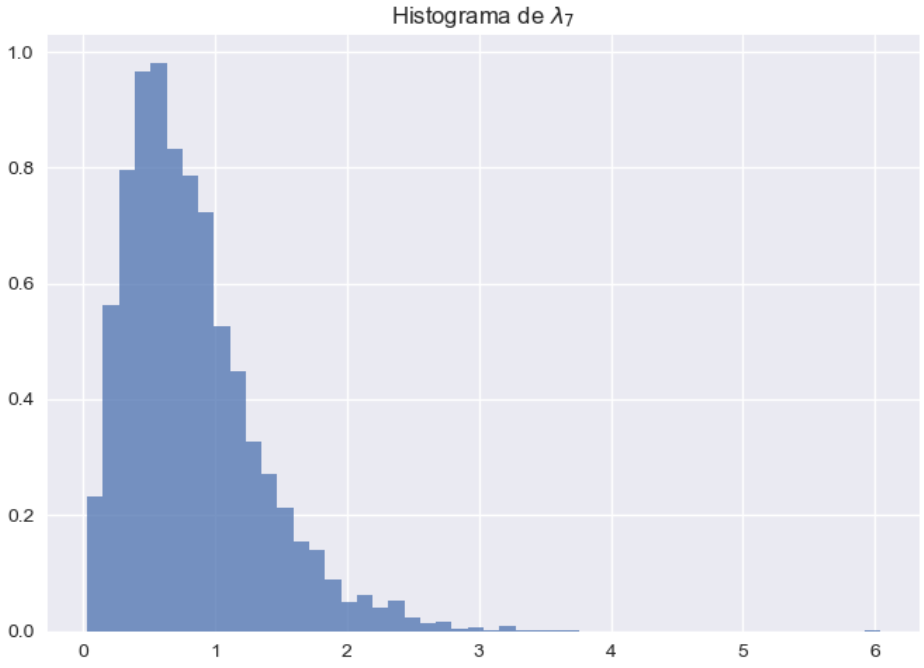

Histograma de λ_2
0
1
2
3
4
5
0.0
0.2
0.4
0.6
0.8
1.0

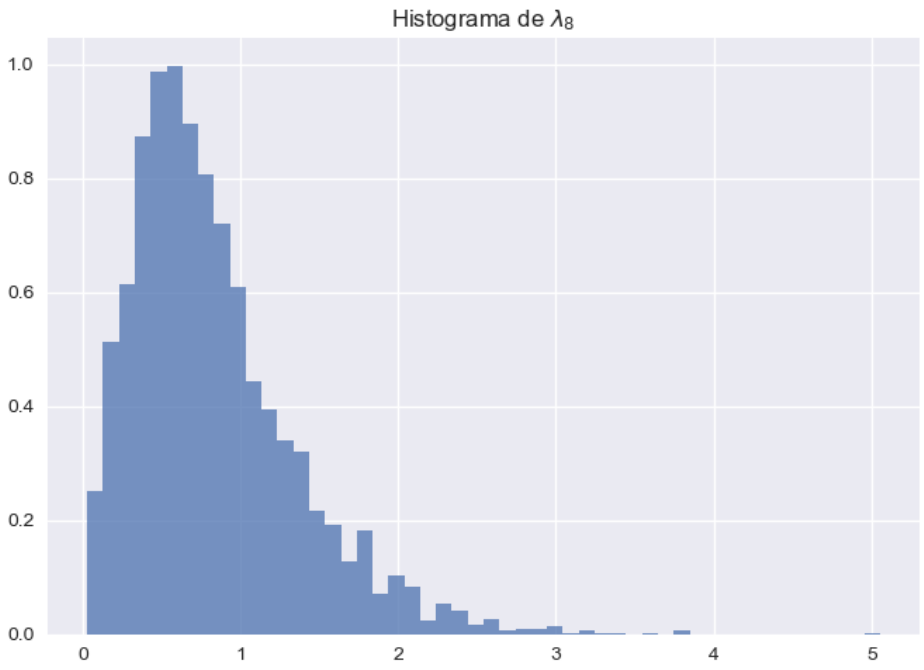

Histograma de λ_3
0
2
4
6
8
10
0.0
0.2
0.4
0.6
0.8
1.0

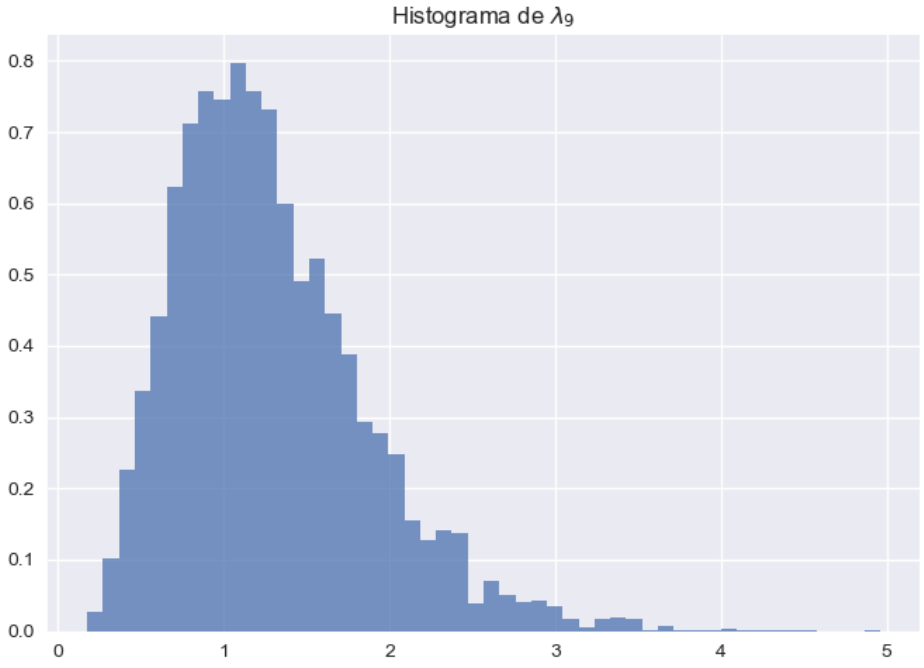

Histograma de λ_4
12
10
8
6
4
2
0
0.0
0.2
0.4
0.6
0.8
1.0

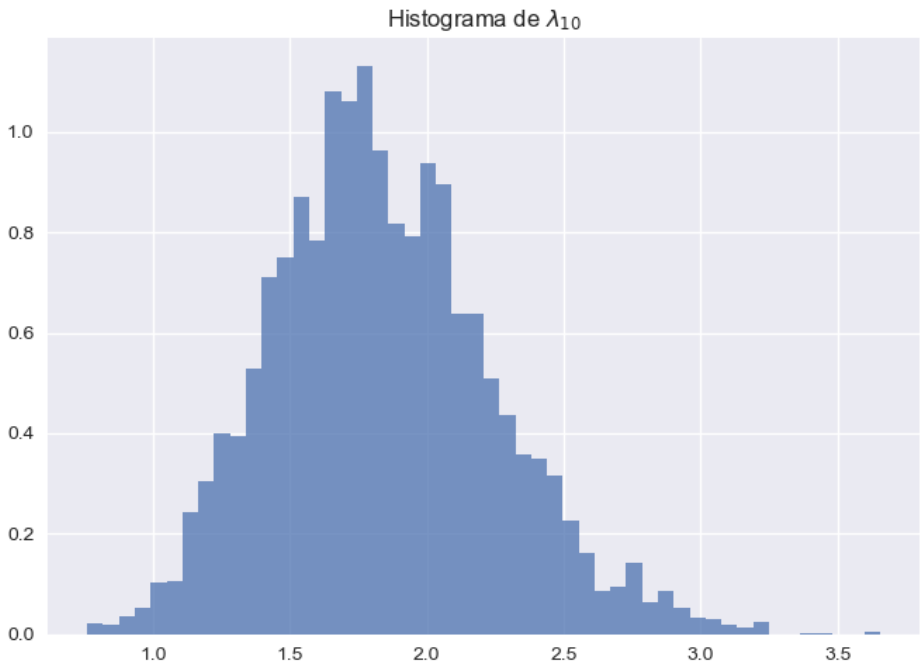

Histograma de λ_5
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0
1.5
2.0
2.5

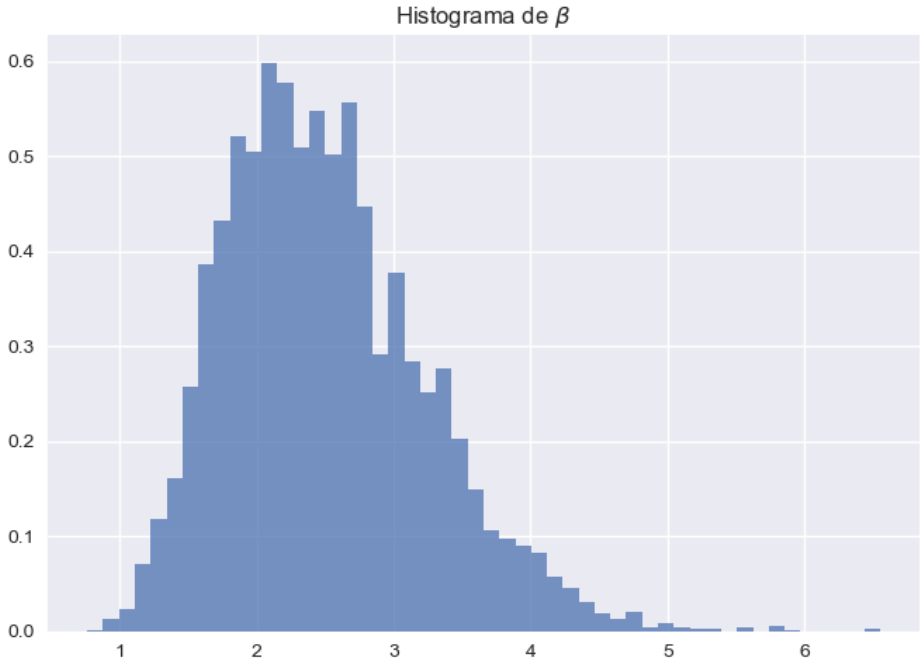

Histograma de λ_6
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.2
0.4
0.6
0.8
1.0


Histograma de λ_7
1.0
0.8
0.6
0.4
0.2
0.0
0
1
2
3
4
5
6


Histograma de λ_8
1.0
0.8
0.6
0.4
0.2
0.0
0
1
2
3
4
5


Histograma de λ_9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
1
2
3
4
5





Como se trata de una distribución en 11 dimensiones, no es posible representar la transición de la cadena en una imagen bidimensional, pero podemos usar los datos con los que contamos para encontrar estimadores de λ_i y β. Al tomar la media de los datos muestreados tenemos los siguientes estimadores;

- $\lambda_1 = 0.07022$.
- $\lambda_2 = 0.1497$.

- $\lambda_3 = 0.1038$.
- $\lambda_4 = 0.1232$.
- $\lambda_5 = 0.6243$.
- $\lambda_6 = 0.5545$.
- $\lambda_7 = 0.8221$.
- $\lambda_8 = 0.8243$.
- $\lambda_9 = 1.2836$.
- $\lambda_{10} = 1.8418$.
- $\beta = 2.4976$.