

Tarea 7. MCMC: Metropolis-Hastings II

Ricardo Chávez Cáliz

October 25, 2017

Con el algoritmo Metropolis-Hastings (MH), simular lo siguiente:

Problema 1. Sean $x_i \sim Ga(\alpha, \beta); i = 1, 2, \dots, n$. Simular datos x_i con $\alpha = 3$ y $\beta = 100$ considerando los casos $n = 5$ y 30 .

Con $\alpha \sim U(1,4)$, $\beta \sim \exp(1)$ distribuciones a priori, se tiene la posterior

$$f(\alpha, \beta | \bar{x}) \propto \frac{\beta^{n\alpha}}{\Gamma(\alpha)^n} r_1^{\alpha-1} e^{-\beta(r_2+1)} 1(1 \leq \alpha \leq 4) 1(\beta > 0),$$

con $r_2 = \sum_{i=1}^n x_i$ y $r_1 = \prod_{i=1}^n x_i$.

En ambos casos, grafica los contornos para visualizar dónde está concentrada la posterior.

Utilizar la propuesta

$$q\left(\begin{pmatrix} \alpha_p \\ \beta_p \end{pmatrix} \middle| \begin{pmatrix} \alpha \\ \beta \end{pmatrix}\right) = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \end{pmatrix},$$

donde

$$\begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \end{pmatrix} \sim \mathcal{N}_2\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix}\right).$$

Se simularon muestras de tamaño 5 y 30, con distribución $Gamma(\alpha, \beta)$ con $\alpha = 3$ y $\beta = 100$, como los algoritmos de *Python* consideran el parámetro de escala θ para usar el parámetro solicitado se consideró $\theta = 1/\beta$. Se obtuvieron r_1 y r_2 del orden de e^{-10} y e^{-2} respectivamente, de la expresión de $f(\alpha, \beta | \bar{x})$ podemos adelantar que los cálculos realizados directamente con estos valores serán inadecuados. Para solucionar este problema se hicieron simplificaciones logarítmicas en la expresión de $f(\alpha, \beta | \bar{x})$ donde

$$\log(f(\alpha, \beta | \bar{x})) = n\alpha \cdot \log(\beta) + (\alpha - 1) \cdot \log(r_1) - \beta \cdot (r_2 + 1) - n \cdot \log(\Gamma(\alpha))$$

Tomaremos el dominio de graficación con $\alpha \in [1, 4]$ y $\beta \in [80, 120]$. La elección para α es evidente de la definición de f , y para β tomamos este rango dado que las simulaciones para la coordenada en β se puede obtener un estimador de β usando las medias muestrales. En la figura 1 se muestran los contornos para f y $\log(f)$, se puede notar que f está concentrada en un rango muy pequeño alrededor del 80, esto es algo que se toma en cuenta para lo consecutivo.

Se implementó el algoritmo de MCMH haciendo las correspondientes simplificaciones logarítmicas para f en el calculo de ρ

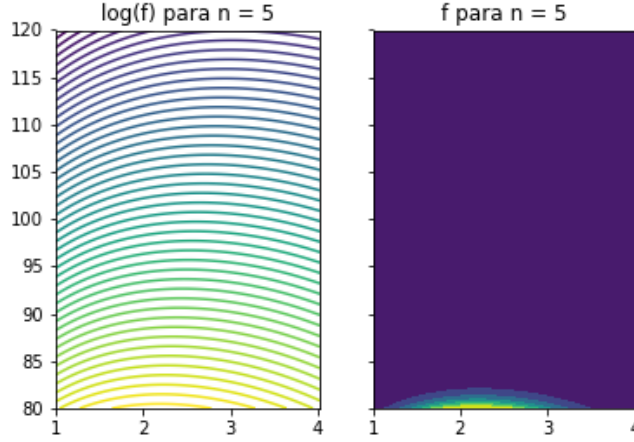


Figure 1: Se muestran las superficies de nivel de $\log(f)$ y f para $n = 5$

Tenemos:

$$\rho = \min \left\{ 1, \frac{\beta_p^{n \cdot \alpha_p} \cdot r_1^{\alpha_p - 1} \cdot e^{-\beta_p(r_2 + 1)} \cdot \Gamma(\alpha)^n \cdot \mathbb{1}_{[1,4]}(\alpha_p) \cdot \mathbb{1}_{[0,\infty)}(\beta_p)}{\beta^{n \cdot \alpha} \cdot r_1^{\alpha - 1} \cdot e^{-\beta(r_2 + 1)} \cdot \Gamma(\alpha_p)^n \cdot \mathbb{1}_{[1,4]}(\alpha) \cdot \mathbb{1}_{[0,\infty)}(\beta)} \right\}$$

$$\rho = \min \left\{ 1, \frac{\beta_p^{n \cdot \alpha_p} \cdot r_1^{\alpha_p - \alpha} \cdot e^{-\beta - \beta_p} \cdot \Gamma(\alpha)^n \cdot \mathbb{1}_{[1,4]}(\alpha_p) \cdot \mathbb{1}_{[0,\infty)}(\beta_p)}{\beta^{n \cdot \alpha} \cdot \Gamma(\alpha_p)^n \cdot \mathbb{1}_{[1,4]}(\alpha) \cdot \mathbb{1}_{[0,\infty)}(\beta)} \right\}$$

Tomando logaritmo y dejando un momento de lado las funciones indicadoras obtenemos

$$\rho = e^{\min\{0, w\}}$$

donde $w = n\alpha_p \cdot \log(\beta_p) + (\alpha_p - \alpha) \cdot \log(r_1) + (\beta - \beta_p) + n \cdot \log(\Gamma(\alpha)) - n\alpha \cdot \log(\beta) - n \cdot \log(\Gamma(\alpha_p))$

Note que q está definida de tal manera de que el movimiento de la cadena de Markov sea independiente en cada coordenadas. De esta manera, estando en el tiempo i pasamos en el tiempo $i + 1$ moviéndonos horizontalmente con una normal de varianza σ_1 centrada en el punto actual y de igual manera el movimiento vertical con una normal de varianza σ_2 . Por lo anterior se tomaron $\sigma_1 = 0.2$ y $\sigma_2 = 1$, ocupamos proponer saltos pequeños por lo estrecho de la región, permitiéndonos mayor libertad en el movimiento vertical.

Sin otra particularidad, se implementó el algoritmo MCMH para la función objetivo dada, con las funciones ρ y q descritas. Para verificar dicha simulación se generó una muestra de tamaño 10000 y se graficaron los valores en el plano en un gradiente de color para poder analizar el comportamiento temporal de la cadena, los primeros elementos son mostrados de color azul y los últimos de color amarillo.

Se llevó registro del número de puntos rechazados, en este caso se rechazaron el 0.6531% de las propuestas.

En la figura 2 se observa un sesgo hacia la izquierda y que la cadena no parece coincidir con la función de densidad objetivo (compare con figura 1). Lo anterior no es sorprendente dado que la información a priori era muy pequeña para tener un buen estimador ($n=5$).

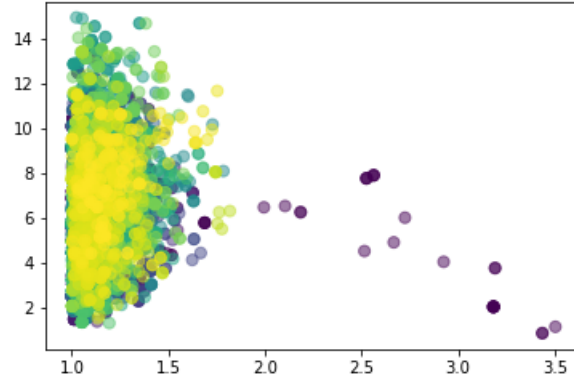


Figure 2: Se muestran la simulación de tamaño 10000, para $n = 5$, $\sigma_1 = 0.2$ y $\sigma_2 = 1$

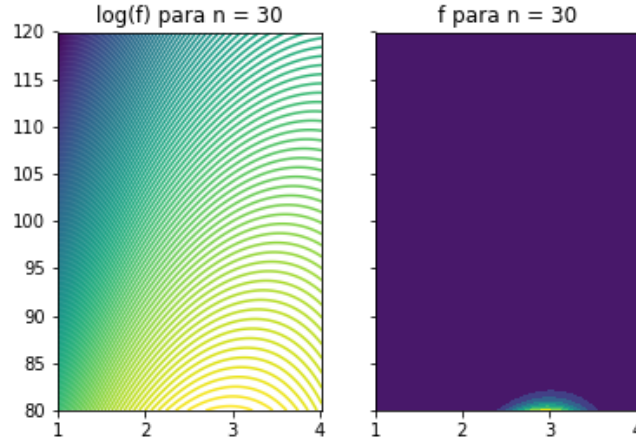


Figure 3: Se muestran las superficies de nivel de $\log(f)$ y f con $n=30$

Análogamente se llevó a cabo la simulación en el caso $n=30$, obteniendo los siguientes resultados.

Igualmente se registró del número de puntos rechazados, en este caso se rechazaron el 28.57% de las propuestas.

En la figura 4, se muestra la simulación de la cadena para $n = 30$ y a diferencia del caso anterior la cadena en este caso sí parece coincidir con la función de densidad objetivo (compare con figura 3). Note que la cadena empieza con puntos de ordenada pequeña (así lo estipulamos en la distribución a priori con $\beta \sim \text{Exp}(1)$) y termina acumulándose en una franja a la altura de 80 como esperábamos y de ancho como el que se muestra en la figura 3. Como es sabido, una muestra de tamaño 30 puede ser suficiente para tener "buenos" estimadores.

Los promedios de las muestras para α fue 2.61930478486 y para β 78.8281005967.

Para encontrar el *burn-in* de la cadena se comparó $\log(X_t)$ vs t en el caso $n = 30$ separando por coordenadas. Se obtuvieron los siguientes resultados.

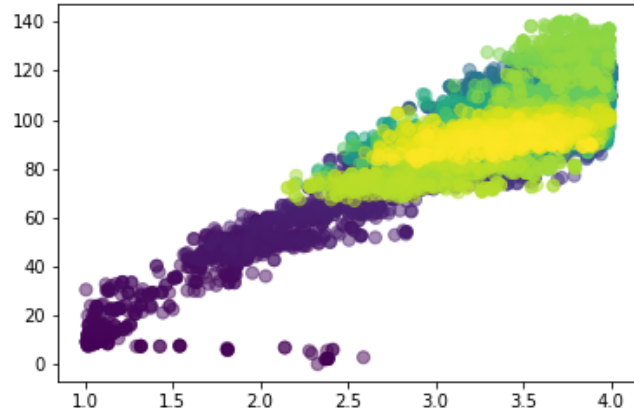


Figure 4: Se muestran la simulación de tamaño 10000, para $n = 30$, $\sigma_1 = 0.2$ y $\sigma_2 = 1$

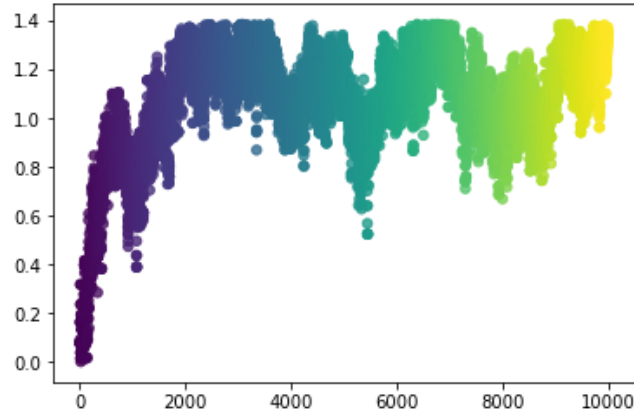


Figure 5: Tiempo de calentamiento para α con $n = 30$ en una muestra de tamaño 10000

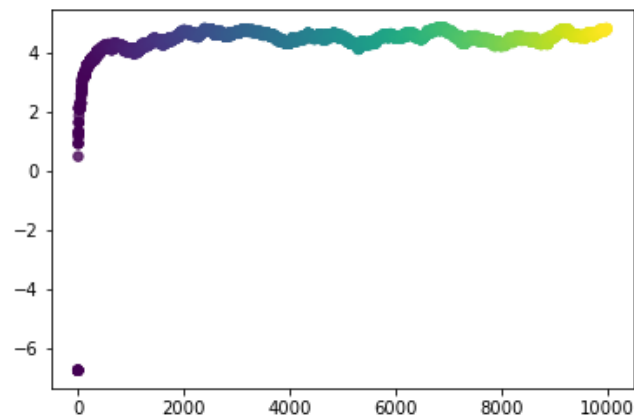


Figure 6: Tiempo de calentamiento para β con $n = 30$ en una muestra de tamaño 10000

Problema 2. Simular de la distribución $\text{Gamma}(\alpha, 1)$ con la propuesta $\text{Gamma}([\alpha], 1)$, donde $[\alpha]$ denota la parte entera de α .

Además, realizar el siguiente experimento: poner como punto inicial $x_0 = 1, 000$ y graficar la evolución de la cadena, es decir, $f(X_t)$ vs t .

Se implementó el algoritmo MCMH, sin alguna particularidad con

$$\rho(x, y) = \min \left\{ 1, \frac{y^{\alpha - [\alpha]}}{x^{\alpha - [\alpha]}} \right\}$$

Se obtuvieron los siguientes resultados al variar el punto inicial, para $x_0 = 1, 10, 100, 1000$. En cada caso se rechazaron alrededor el 15% de las propuestas

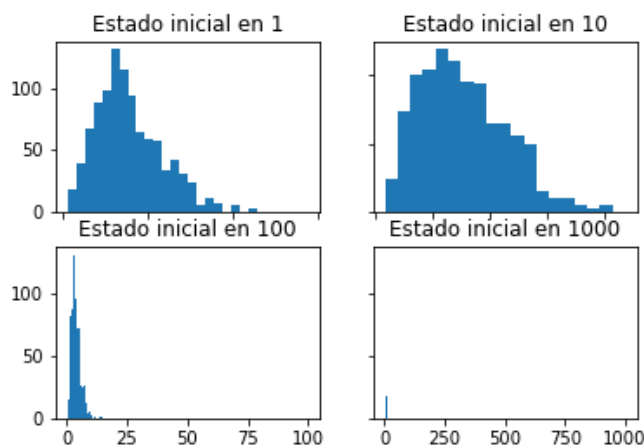


Figure 7: Histogramas de una muestra de tamaño 1000, variando el punto de inicio de la cadena

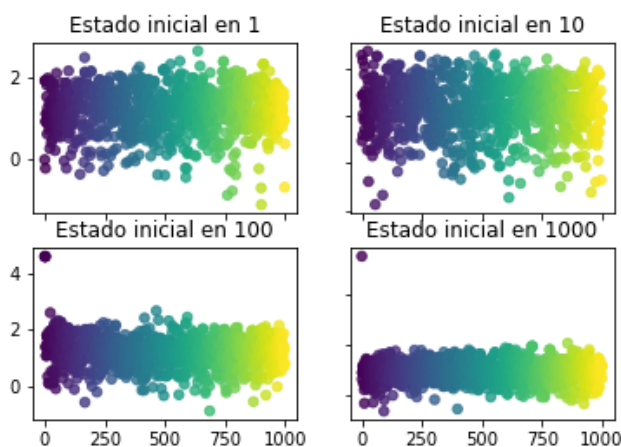


Figure 8: Tiempo de calentamiento en una muestra de tamaño 1000 variando el punto inicial

En la figura 7 se puede apreciar para los casos 1, 10 y 100 la forma de la gamma buscada, con una observación extraña en el valor de x_0 correspondiente, la figura 8 permite discriminar el tiempo en que dicha observación extraña no aparece

Problema 3. Implementar Random Walk Metropolis Hasting (RWMH) donde la distribución objetivo es $\mathcal{N}_2(\mu, \Sigma)$, con

$$\mu = \begin{pmatrix} 3 \\ 5 \end{pmatrix} \quad \Sigma = \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix}.$$

Utilizar como propuesta $\varepsilon_t \sim \mathcal{N}_2(\mathbf{0}, \sigma I)$. ¿Cómo elegir σ para que la cadena sea eficiente? >Qué consecuencias tiene la elección de σ ?

Como experimento, elige como punto inicial $x_o = \begin{pmatrix} 1000 \\ 1 \end{pmatrix}$ y comenta los resultados.

Para todos los incisos del ejercicio anterior:

- Establece cual es tu distribución inicial.
- Grafica la evolución de la cadena.
- Indica cuál es el Burn-in.
- Comenta qué tan eficiente es la cadena.
- Implementa el algoritmo MH considerando una propuesta diferente.

Para implementar el algoritmo se tomó $y_t = x_t + \epsilon_t$ con una densidad instrumental $q(y|x) = g(\|y - x\|)$ donde g es la densidad de $\|\epsilon_t\|$, como q es simétrica entonces

$$\rho(x, y) = \min \left\{ 1, \frac{f(y)}{f(x)} \right\}$$

nuevamente para obtener cálculos adecuados de ρ se hicieron simplificaciones logarítmicas (usando monotonía de la función logaritmo) .

$$\rho(x, y) = e^c$$

donde $c = \min \{0, \log(f(y)) - \log(f(x))\}$ con $\log(f) = (-1/(2 * 0.36)) * ((y1 - 3.0) ** 2 + (y2 - 5.) ** 2 - 2. * 0.8 * (y1 - 3.) * (y2 - 5.))$

Simulamos el caso óptimo con punto inicial en la media (3, 5), tomando una muestra de tamaño 1000 con $\sigma = 1$ se obtuvieron los siguientes resultados los cuales se grafican junto con los contornos de nivel de la función de densidad bivariada correspondiente. Como antes los colores hacen referencia al orden que lleva la cadena siendo azul los primeros elementos y amarillos los últimos.

Para tener una mejor noción de la caminata aleatoria realizada se graficó una línea gradiente entre cada estado de la cadena.

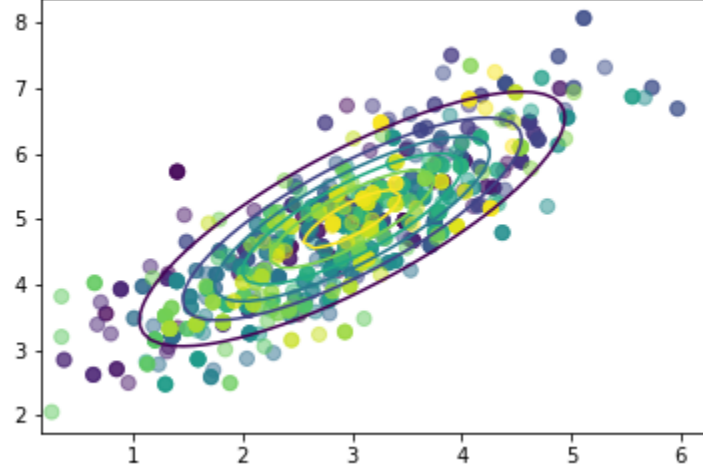


Figure 9: Muestra generada con el algoritmo RWMH iniciando en la media de la distribución objetivo y comparada con los conjuntos de nivel de la función de densidad correspondiente

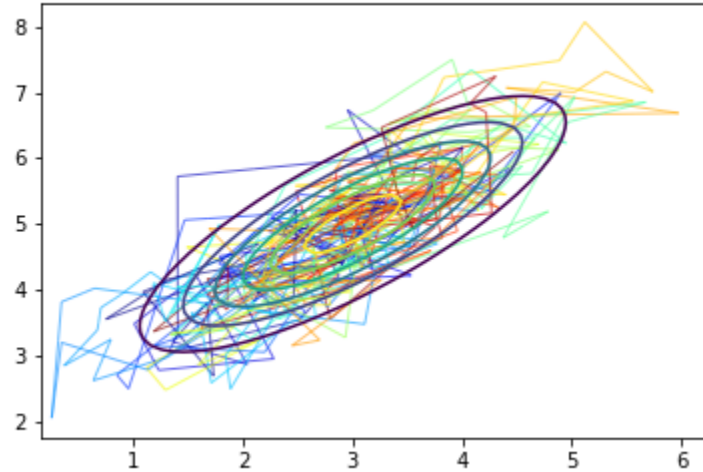


Figure 10: Caminata aleatoria generada con el algoritmo RWMH iniciando en media y comparada con los conjuntos de nivel de la función de densidad correspondiente

Se realizó el experimento realizando la caminata aleatoria desde un punto lejano a la media como $(1000, 1)$, para lograr la convergencia con un tamaño de muestra como el que se tomó, se tomó $\sigma = 20$. La importancia en la elección del sigma reside en que cuando empezamos en un punto lejano a la función objetivo, podemos llegar a esta más pronto, sin embargo estando cerca de la densidad la simulación será imprecisa.

Se obtuvieron los siguientes resultados

Por las dimensiones del ejemplo, no se aprecia en esta gráfica que la cadena se acumula

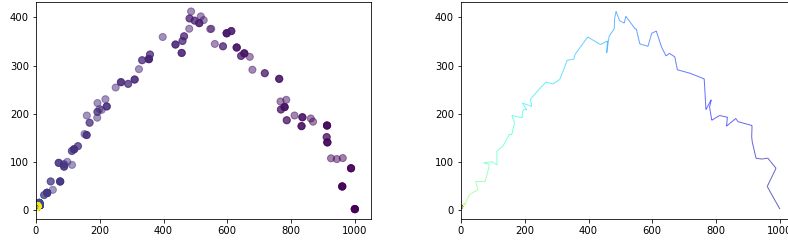


Figure 11: Muestra generada con el algoritmo RWMH iniciando en $(1000,1)$, $\sigma = 20$ y de tamaño 1000

en donde debe, por ello se grafica cuando el punto de inicio es $(100,1)$, que tendrá el mismo efecto demostrativo.

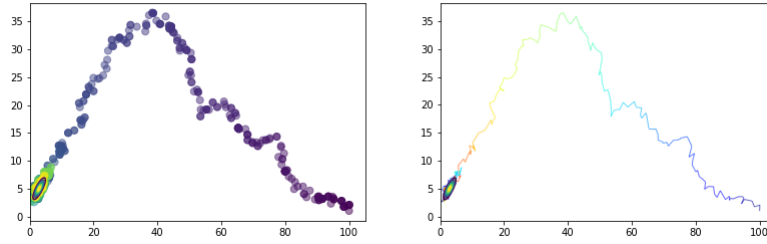


Figure 12: Muestra generada con el algoritmo RWMH iniciando en $(100,1)$, $\sigma = 2$ y de tamaño 1000

La peculiaridad de la curva seguida por la caminata aleatoria no es una coincidencia, ya que el eigenvector asociado al eigenvalor mayor de la matriz de covarianza apunta en la dirección de la caminata seguida y corresponde a una línea de flujo que proviene de la geometría de la superficie dada por la función de densidad.

Para encontrar el *burn-in* de la cadena se comparó $\log(X_t)$ vs t en el caso $n = 30$ separando por coordenadas. Se obtuvieron los siguientes resultados.

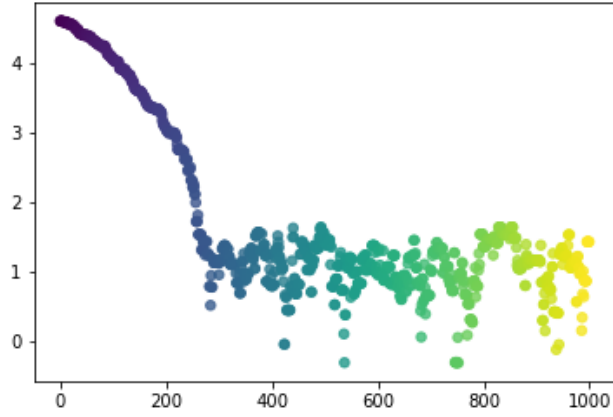


Figure 13: Tiempo de calentamiento para α con $n = 30$ en una muestra de tamaño 1000

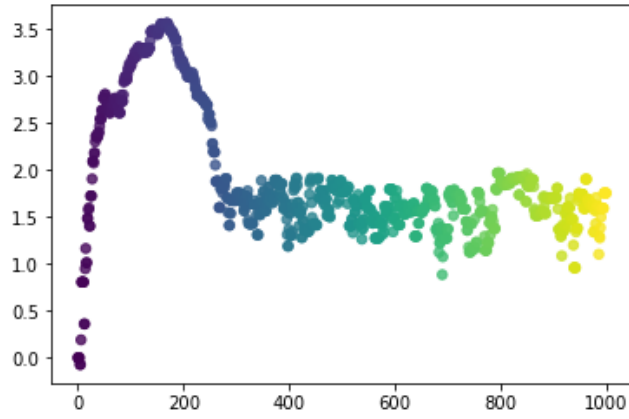


Figure 14: Tiempo de calentamiento para β con $n = 30$ en una muestra de tamaño 1000

Alrededor del tiempo 200, la cadena se estabiliza en ambas coordenadas, por lo que quitando los primeros 200 términos de la cadena obtendremos una simulación apropiada.