Computational Statistics

Homework 5

Romane PERSCH

December 16, 2016

9.4 Relationship between the 2-stage Gibbs sampler and Metropolis-Hastings algorithms

We are considering the following 2-stage Gibbs sampler : Initialize $Y^{(0)} = y^{(0)}$. For $t = 0, 1, 2, ...$, generate :
1. $X^{(t+1)} \sim f_{X|Y}(x|y^{(t)})$
2. $Y^{(t+1)} \sim f_{Y|X}(y|x^{(t+1)})$
Assume we are only interested in the sub process $(X^{(t)})$, which is actually a Markov chain is the context of a 2-stage Gibbs sampler.
We will show that this subchain $(X^{(t)})$ can be viewed as the result of a Metropolis-Hastings algorithm where the candidate simulation is always accepted.

Reminder - Metropolis-Hastings algorithm with distribution of interest f and candidate $g(x'|x)$
Given $x^{(t)}$
1. $X'_{t+1} \sim g(x'|x^{(t)})$
2. Accept $X^{(t+1)} = x'_{t+1}$ with probability $\min\left(1, \frac{f(x'_{t+1})}{f(x^{(t)})}\frac{g(x^{(t)}|x'_{t+1})}{g(x'_{t+1}|x^{(t)})}\right)$. Otherwise, take $X^{(t+1)} = x^{(t)}$.

(a) Candidate distribution

The 2-stage Gibbs sampler can be rewritten as :
Given $x^{(t)}$
1. $Y^{(t)} \sim f_{Y|X}(y|x^{(t)})$
2. $X^{(t+1)} \sim f_{X|Y}(x|y^{(t)})$

If we focus on the X chain, this is actually also equivalent to simulate a candidate for X at each new step $(t+1)$ and to accept it with probability 1. In other words, we can rewrite the Gibbs sampler as :
Given $x^{(t)}$
1. $X'^{(t+1)} \sim K(x^{(t)}, x')$
2. Accept $X^{(t+1)} = x'^{(t+1)}$ with probability 1
where $K(x, x')$ is the transition kernel of the X subchain in the 2-stage Gibbs sampler. More precisely (see my next exercise 9.18 (a) for more explanation about the derivation of this transition kernel) :

$$K(x, x') = \int f_{X|Y}(x'|y) f_{Y|X}(y|x) dy$$

(b) Necessary condition on the acceptance probability so that the X chain of our Gibbs sampler may be equal to the Metropolis chain of a Metropolis-Hastings algorithm

In the context of the 2-stage Gibbs sampler, we know that the marginal distribution of X (that we will denote $f_X(\cdot)$) is an invariant distribution for the resulting Gibbs subchain X. In the context of a Metropolis-Hastings algorithm with distribution of interest f, we know that the distribution of interest will be an invariant distribution for the resulting Metropolis chain. Hence, if the resulting X chain of our Gibbs sampler is equal to the resulting X chain from a Metropolis-Hastings algorithm, we necessarily need a distribution of interest in our Metropolis-Hastings algorithm equal to $f_X(\cdot)$ (since a Markov chain can only have one unique invariant distribution). Therefore, the acceptance probability in the Metropolis-Hastings algorithm must be :

$$\rho(x, x') = \min\Big(1, \frac{f_X(x')}{f_X(x)} \frac{g(x|x')}{g(x'|x)}\Big)$$

(c) Show that this acceptance probability is equal to 1 if we take $g(x'|x) = K(x, x')$ and deduce that our Gibbs sampler can indeed be written as a Metropolis-Hastings algorithm

Step 1 - Show that this acceptance probability is equal to 1 if we take $g(x'|x) = K(x, x')$ Let us consider the Metropolis-Hastings algorithm with $g(x'|x) = K(x, x')$. Then, the acceptance probability is :

$$\rho(x, x') = \min\Big(1, \frac{f_X(x')}{f_X(x)} \frac{K(x', x)}{K(x, x')}\Big)$$

However:

$$\begin{aligned} f_X(x')K(x', x) &= f_X(x') \int f_{X|Y}(x|y) f_{Y|X}(y|x')dy \\ &= \int f_{X|Y}(x|y) f_X(x') f_{Y|X}(y|x')dy \\ &= \int f_{X|Y}(x|y) f_{X,Y}(x', y)dy \quad \text{where } f_{X,Y} \text{ is the joint density} \end{aligned}$$

Moreover : $f_{X|Y}(x|y) = \frac{f_X(x) f_{Y|X}(y|x)}{f_Y(y)}$ (Bayes). Hence :

$$\begin{aligned} f_X(x')K(x', x) &= \int \frac{f_X(x) f_{Y|X}(y|x)}{f_Y(y)} f_{X,Y}(x', y)dy \\ &= f_X(x) \int \frac{f_{Y|X}(y|x)}{f_Y(y)} f_Y(y) f_{X|Y}(x'|y)dy \quad \text{(Bayes)} \\ &= f_X(x) \int f_{Y|X}(y|x) f_{X|Y}(x'|y)dy \\ &= f_X(x)K(x, x') \end{aligned}$$

Hence : $\boxed{\rho(x, x') = 1}$

(Note that we also showed that the Gibbs subchain X and the marginal density f_X verify the detailed balance condition.)

Step 2 - Deduce that our Gibbs sampler can be written as a Metropolis algorithm Hence, we can rewrite our Gibbs sampler as :
Given $x^{(t)}$
1. $X'^{(t+1)} \sim K(x^{(t)}, x')$
2. Accept $X^{(t+1)} = x'^{(t+1)}$ with probability $\rho(x^{(t)}, x'^{(t+1)}) = \min\Big(1, \frac{f(x'^{(t+1)})}{f(x^{(t)})} \frac{K(x'^{(t+1)}, x^{(t)})}{K(x^{(t)}, x'^{(t+1)})}\Big) = 1$

We see that it is actually exactly the Metropolis-Hastings algorithm with candidate $K(x, x')$ and distribution of interest f_X (the marginal density of X). Therefore, we can view the Gibbs sampler as a Metropolis-Hastings algorithm.

9.18 Bivariate Normal Gibbs Sampler

Let us consider the following Gibbs sampler :

$$X|y \sim \mathcal{N}(\rho y, 1 - \rho^2)$$

$$Y|x \sim \mathcal{N}(\rho x, 1 - \rho^2)$$

Note that we need to assume that $|\rho| < 1$ (since both variances must be strictly positive).

(a) Transition kernel for the X chain

First, note that by construction of a 2-stage Gibbs sampler, the resulting process $(X^{(t)})$ is indeed a Markov chain.
The transition kernel for the X chain $K(x^*, \cdot)$ is the probability distribution of $X^{(t+1)}$ given $X^{(t)} = x^*$. We thus need to derive the $\pi(x^{(t+1)}|x^{(t)})$ distribution
Since we can view the distribution $\pi(x^{(t+1)}|x^{(t)})$ as the marginal distribution of $(x^{(t+1)}, y^{(t)})$ given $x^{(t)}$, we have :

$$\begin{aligned}\pi(x^{(t+1)}|x^{(t)}) &= \int \pi(x^{(t+1)}, y^{(t)}|x^{(t)})dy^{(t)} \\ &= \int \pi(y^{(t)}|x^{(t)})\pi(x^{(t+1)}|y^{(t)}, x^{(t)})dy^{(t)}\end{aligned}$$

By construction of the Gibbs sampler, $\pi(x^{(t+1)}|y^{(t)}, x^{(t)}) = \pi(x^{(t+1)}|y^{(t)})$, hence :

$$= \int f_{Y|X}(y^{(t)}|x^{(t)})f_{X|Y}(x^{(t+1)}|y^{(t)})dy^{(t)}$$

Hence :

$$K(x^*, x) = \int f_{Y|X}(y|x^*)f_{X|Y}(x|y)dy$$

Therefore, replacing $f_{Y|X}$ and $f_{X|Y}$ by the corresponding normal densities :

$$K(x^*, x) = \int \frac{1}{\sqrt{2\pi(1-\rho^2)}}e^{-\frac{(y-\rho x^*)^2}{2(1-\rho^2)}} \frac{1}{\sqrt{2\pi(1-\rho^2)}}e^{-\frac{(x-\rho y)^2}{2(1-\rho^2)}} dy$$

This simplifies in : $\boxed{K(x^*, x) = \frac{1}{2\pi(1-\rho^2)} \int e^{-\frac{(y-\rho x^*)^2}{2(1-\rho^2)}} e^{-\frac{(x-\rho y)^2}{2(1-\rho^2)}} dy}$

(b) Show that $X \sim \mathcal{N}(0, 1)$ is the invariant distribution of the X chain

Let ϕ the density of a $\mathcal{N}(0, 1)$. We will show that :

$$\forall x \in \mathbb{R}, \phi(x) = \int K(x^*, x)\phi(x^*)dx^*$$

Let $x \in \mathbb{R}$ fixed. Using (a) :

$$\int K(x^*, x)\phi(x^*)dx^* = \frac{1}{2\pi(1-\rho^2)} \int \Big(\int e^{-\frac{(y-\rho x^*)^2}{2(1-\rho^2)}} e^{-\frac{(x-\rho y)^2}{2(1-\rho^2)}} dy\Big) \frac{1}{\sqrt{2\pi}}e^{-\frac{(x^*)^2}{2}} dx^*$$

Using Fubini, since all functions in the integrals are positive :

$$\int K(x^*, x)\phi(x^*)dx^* = \frac{1}{\sqrt{2\pi}} \frac{1}{2\pi(1-\rho^2)} \int \Big(\int e^{-\frac{(y-\rho x^*)^2}{2(1-\rho^2)}} e^{-\frac{(x^*)^2}{2}} dx^*\Big) e^{-\frac{(x-\rho y)^2}{2(1-\rho^2)}} dy$$

The integral in x^* can be rewritten as :

$$\begin{aligned}I(y) = \int e^{-\frac{(y-\rho x^*)^2}{2(1-\rho^2)}} e^{-\frac{(x^*)^2}{2}} dx^* &= \int e^{-\frac{1}{2(1-\rho^2)}[(y-\rho x^*)^2+(1-\rho^2)(x^*)^2]}dx^* \\ &= \int e^{-\frac{1}{2(1-\rho^2)}[y^2-2\rho x^* y+(x^*)^2]}dx^* \\ &= \int e^{-\frac{1}{2(1-\rho^2)}[(x^*-\rho y)^2-\rho^2y^2+y^2]}dx^* \\ &= e^{-\frac{(1-\rho^2)y^2}{2(1-\rho^2)}} \int e^{-\frac{1}{2(1-\rho^2)}[(x^*-\rho y)^2]}dx^*\end{aligned}$$

Recognizing the density of a $\mathcal{N}(\rho y, 1-\rho^2)$:

$$I(y) = \int e^{-\frac{(y-\rho x^*)^2}{2(1-\rho^2)}} e^{-\frac{(x^*)^2}{2}} dx^* = e^{-\frac{y^2}{2}} \times \sqrt{2\pi}\sqrt{1-\rho^2}$$

Therefore :

$$\int K(x^*, x)\phi(x^*)dx^* = \frac{\sqrt{2\pi}\sqrt{1-\rho^2}}{\sqrt{2\pi}} \frac{1}{2\pi(1-\rho^2)} \int e^{-\frac{y^2}{2}} e^{-\frac{(x-\rho y)^2}{2(1-\rho^2)}} dy$$

Recognizing the same integral as above, i.e recognizing $I(x)$:

$$\int K(x^*, x)\phi(x^*)dx^* = \frac{1}{2\pi\sqrt{1-\rho^2}}e^{-\frac{x^2}{2}} \times \sqrt{2\pi}\sqrt{1-\rho^2}$$
$$\int K(x^*, x)\phi(x^*)dx^* = \frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}$$
$$\boxed{\int K(x^*, x)\phi(x^*)dx^* = \phi(x)}$$

Therefore, the invariant distribution for the X chain is $X \sim \mathcal{N}(0, 1)$.

(c) Show that $X|x^* \sim \mathcal{N}(\rho^2 x^*, 1-\rho^2)$ (i.e $X^{(t+1)}|X^{(t)} = x^* \sim \mathcal{N}(\rho^2 x^*, 1-\rho^2)$)

Let x^* be fixed. The transition kernel for the X chain $K(x^*, \cdot)$ is the probability distribution of $X^{(t+1)}$ given $X^{(t)} = x^*$. We therefore need to show that $K(x^*, \cdot)$ is equal to the density of a $\mathcal{N}(\rho^2 x^*, 1-\rho^2)$.

Using (a) :

$$\begin{aligned}
K(x^*, x) &= \frac{1}{2\pi(1-\rho^2)}\int e^{-\frac{1}{2(1-\rho^2)}[(y-\rho x^*)^2+(x-\rho y)^2]}dy \\
&= \frac{1}{2\pi(1-\rho^2)}\int e^{-\frac{1}{2(1-\rho^2)}[x^2-2\rho xy+\rho^2y^2+y^2-2\rho x^* y+\rho^2(x^*)^2]}dy \\
&= \frac{1}{2\pi(1-\rho^2)}\int e^{-\frac{1}{2(1-\rho^2)}[(1+\rho^2)y^2-2\rho y(x^*+x)]}e^{-\frac{1}{2(1-\rho^2)}[x^2+\rho^2(x^*)^2]}dy \\
&= \frac{1}{2\pi(1-\rho^2)}e^{-\frac{1}{2(1-\rho^2)}[x^2+\rho^2(x^*)^2]}\int e^{-\frac{(1+\rho^2)}{2(1-\rho^2)}[y^2-2\frac{\rho}{(1+\rho^2)}y(x^*+x)]}dy \\
&= \frac{1}{2\pi(1-\rho^2)}e^{-\frac{1}{2(1-\rho^2)}[x^2+\rho^2(x^*)^2]}\int e^{-\frac{(1+\rho^2)}{2(1-\rho^2)}\left[\left(y-\frac{\rho(x^*+x)}{(1+\rho^2)}\right)^2-\left(\frac{\rho(x^*+x)}{(1+\rho^2)}\right)^2\right]}dy \\
&= \frac{1}{2\pi(1-\rho^2)}e^{-\frac{1}{2(1-\rho^2)}[x^2+\rho^2(x^*)^2]}e^{-\frac{1}{2(1-\rho^2)}\frac{\rho^2(x^*+x)^2}{1+\rho^2}}\int e^{-\frac{(1+\rho^2)}{2(1-\rho^2)}\left(y-\frac{\rho(x^*+x)}{(1+\rho^2)}\right)^2}dy
\end{aligned}$$

Recognizing the density of a $\mathcal{N}(\frac{\rho(x^*+x)}{(1+\rho^2)}, \frac{1-\rho^2}{1+\rho^2})$:

$$\begin{aligned}
K(x^*, x) &= \frac{1}{2\pi(1-\rho^2)}e^{-\frac{1}{2(1-\rho^2)}\left[x^2+\rho^2(x^*)^2-\frac{\rho^2(x^*+x)^2}{1+\rho^2}\right]} \times \sqrt{2\pi}\frac{\sqrt{1-\rho^2}}{\sqrt{1+\rho^2}} \\
&= \frac{1}{\sqrt{2\pi}\sqrt{(1-\rho^2)(1+\rho^2)}}e^{-\frac{1}{2(1-\rho^2)(1+\rho^2)}\left[(1+\rho^2)x^2+(1+\rho^2)\rho^2(x^*)^2-\rho^2(x^*+x)^2\right]} \\
&= \frac{1}{\sqrt{2\pi}\sqrt{1-\rho^4}}e^{-\frac{1}{2(1-\rho^4)}\left[x^2+\rho^4(x^*)^2-2\rho^2x^*x\right]}
\end{aligned}$$
$$\boxed{K(x^*, x) = \frac{1}{\sqrt{2\pi}\sqrt{1-\rho^4}}e^{-\frac{1}{2(1-\rho^4)}(x-\rho^2x^*)^2}}$$

Hence, we recognize that : $X|x^* \sim \mathcal{N}(\rho^2 x^*, 1-\rho^4)$

(d) Show that the Markov chain in X is defined by an AR(1) relation and that the covariances $cov(X_0, X_k)$ go to zero

Show that we can write $X_k = \rho^2 X_{k-1} + U_k, k = 1, 2, ...$, where the U_k are i.i.d $\mathcal{N}(0, 1-\rho^4)$ We showed in question (b) that $X_k|X_{k-1} = x_{k-1} \sim \mathcal{N}(\rho^2 x_{k-1}, 1-\rho^4)$. Hence :

$$X_k - \rho^2 X_{k-1}|X_{k-1} \sim \mathcal{N}(0, 1-\rho^4)$$

Let $U_k = X_k - \rho^2 X_{k-1}$ for $k = 1, 2...$
We then have : $U_k | X_{k-1} \sim \mathcal{N}(0, 1-\rho^4)$, a distribution which does not depend on X_{k-1}. Therefore :

$$\begin{aligned}
\pi(u_k) &= \int \pi(u_k, x_{k-1}) dx_{k-1} \\
&= \int \pi(x_{k-1})\pi(u_k|x_{k-1}) dx_{k-1} \quad \text{(Bayes)} \\
&= \int \pi(x_{k-1})\phi(\frac{u_k}{\sqrt{1-\rho^4}}) dx_{k-1} \\
&= \phi(\frac{u_k}{\sqrt{1-\rho^4}}) \int \pi(x_{k-1}) dx_{k-1} \\
&= \phi(\frac{u_k}{\sqrt{1-\rho^4}}) \quad \text{since} \int \pi(x_{k-1}) dx_{k-1} = 1
\end{aligned}$$

Hence: $U_k \sim \mathcal{N}(0, 1-\rho^4)$. This also implies that $U_k \perp\!\!\!\perp X_{k-1}$ since :

$$\pi(u_k, x_{k-1}) = \pi(x_{k-1})\pi(u_k|x_{k-1}) = \pi(x_{k-1})\pi(u_k)$$

Therefore, we showed that :

$$\begin{cases} U_k \perp\!\!\!\perp X_{k-1} \\ U_k \sim \mathcal{N}(0, 1-\rho^4) \end{cases}$$

We will now show that $U_k \perp\!\!\!\perp U_{k-i}$ for all k and for all $i < k$.
Let $i = 1, ..., k-1$. Let A and B two borelian sets.

$$\begin{aligned}
P(U_k \in A, U_{k-i} \in B) &= E\Big(\mathbb{1}_A(U_k)\mathbb{1}_B(U_{k-i})\Big) \\
&= E\Big(E(\mathbb{1}_A(U_k)\mathbb{1}_B(U_{k-i})|X_{k-1}, ..., X_0)\Big) \\
&= E\Big(\mathbb{1}_B(U_{k-i})E(\mathbb{1}_A(U_k)|X_{k-1}, ..., X_0)\Big) \quad \text{since } U_{k-i} \text{ is a linear combination of } X_{k-i} \text{ and } X_{k-i-1} \\
&= E\Big(\mathbb{1}_B(U_{k-i})E(\mathbb{1}_A(X_k - \rho^2 X_{k-1})|X_{k-1}, ..., X_0)\Big) \quad \text{(Definition of } U_k) \\
&= E\Big(\mathbb{1}_B(U_{k-i})E(\mathbb{1}_A(X_k - \rho^2 X_{k-1})|X_{k-1})\Big) \quad \text{using the Markov property} \\
&= E\Big(\mathbb{1}_B(U_{k-i})E(\mathbb{1}_A(U_k)|X_{k-1})\Big) \\
&= E\Big(\mathbb{1}_B(U_{k-i})E(\mathbb{1}_A(U_k))\Big) \quad \text{since } U_k \perp\!\!\!\perp X_{k-1} \\
&= E(\mathbb{1}_A(U_k))E(\mathbb{1}_B(U_{k-i})) \\
&= P(U_k \in A)P(U_{k-i} \in B)
\end{aligned}$$

From this, it follows that :

$$\begin{cases} U_k \perp\!\!\!\perp U_j \quad \text{if } k \neq j \\ U_k \sim \mathcal{N}(0, 1-\rho^4) \end{cases}$$

In other words : the U_k are i.i.d $\mathcal{N}(0, 1-\rho^4)$.

$$\boxed{\text{Conclusion : We can write } X_k = \rho^2 X_{k-1} + U_k, k = 1, 2, ..., \text{ where the } U_k \text{ are i.i.d } \mathcal{N}(0, 1-\rho^4)}$$

Show that $cov(X_0, X_k) = \rho^{2k} V(X_0)$ and deduce that the covariances go to zero Using the previous result, i. e we can write $X_k = \rho^2 X_{k-1} + U_k$, we get :

$$\begin{aligned}
cov(X_0, X_k) &= cov(X_0, \rho^2 X_{k-1} + U_k) \\
&= \rho^2 cov(X_0, X_{k-1}) + cov(X_0, U_k)
\end{aligned}$$

Show that $cov(X_0, U_k) = 0$

$$\begin{aligned}
cov(X_0, U_k) &= E(X_0 U_k) - E(X_0)E(U_k) \\
&= E(X_0 U_k) \quad \text{since } E(U_k) = 0 \\
&= E\Big(E(X_0 U_k | X_{k-1}, ..., X_0)\Big) \\
&= E\Big(X_0 E(U_k | X_{k-1}, ..., X_0)\Big) \\
&= E\Big(X_0 E(U_k | X_{k-1})\Big) \quad \text{(Markov property)} \\
&= 0 \quad \text{since } U_k | X_{k-1} \sim \mathcal{N}(0, 1 - \rho^4)
\end{aligned}$$

Show that $cov(X_0, X_k) = \rho^{2k} V(X_0)$
Hence :

$$\begin{aligned}
cov(X_0, X_k) &= \rho^2 cov(X_0, X_{k-1}), \text{ so by iteration, since we recognize a geometric sequence :} \\
&= \rho^2(\rho^2 cov(X_0, X_{k-2})) \\
&= ... \\
&= (\rho^2)^k cov(X_0, X_0)
\end{aligned}$$

$$\boxed{cov(X_0, X_k) = \rho^{2k} V(X_0)}$$

As $|\rho| < 1$ (see the remark at the beginning of the exercise : this is a necessary condition for the normal distributions mentioned in the Gibbs sampler to exist), we have : $\rho^{2k} \xrightarrow[k \to +\infty]{} 0$. Hence :

$$\boxed{cov(X_0, X_k) \xrightarrow[k \to +\infty]{} 0}$$

Note that there is no particular reason for $V(X_0)$ to be equal to 1, since it depends on how we initialize the algorithm. However, as shown above, it does not change the fact that the covariances go to zero, and this result remains the same if we consider $cov(X_i, X_k)$ when i is fixed.

10.10 Animal epidemiology

We are considering the following hierarchical model :

$$X_i \sim \mathcal{P}(\lambda_i)$$

$$\lambda_i \sim \mathcal{G}a(\alpha, \beta_i)$$

[1]

$$\beta_i \sim \mathcal{G}a(a, b)$$

It describes the epidemy of mastisis in dairy cattle herds over a one year period. For herd i, X_i denotes the number of cases in the herd and hence, λ_i is the underlying rate of infection of herd i. As mastisis is infectious, there is an underlying dependence between the herds. To account for this, it is proposed to put a common Gamma prior on the Poisson parameter of each herd.

The goal is here to estimate the posterior distribution of the underlying parameters (λ_i, β_i) of each herd using a Gibbs sampler. Note that we will use one Gibbs sampler per herd, since the parameters (λ_i, β_i) are different between each herd.

(a) Full conditionals

Full conditional for λ_i The conditional distribution for λ_i is :

$$\pi(\lambda_i | x, \alpha, \beta_i, a, b) \propto \pi(\lambda_i | \alpha, \beta_i, a, b)\pi(x | \lambda_i, \alpha, \beta_i, a, b) \quad \text{(Bayes' theorem)}$$

Given all parameters, X_i is independent of the other X_k $(k \neq i)$ due to the hierarchical structure of the model, hence :

$$\pi(\lambda_i | x, \alpha, \beta_i, a, b) \propto \pi(\lambda_i | \alpha, \beta_i, a, b)\pi(x_i | \lambda_i, \alpha, \beta_i, a, b)\pi(x_{-i} | \lambda_i, \alpha, \beta_i, a, b)$$

[1] We consider the definition of the Gamma distribution where α is the shape and β is the rate.

As $\pi(x_{-i}|\lambda_i,\alpha,\beta_i,a,b) = \pi(x_{-i}|\alpha,a,b)$ due to the hierarchical structure of the model, it does not depend on λ_i so it can be removed from the calculus :

$$\pi(\lambda_i|x,\alpha,\beta_i,a,b) \propto \pi(\lambda_i|\alpha,\beta_i,a,b)\pi(x_i|\lambda_i,\alpha,\beta_i,a,b)$$
$$\pi(\lambda_i|x,\alpha,\beta_i,a,b) \propto \pi(\lambda_i|\alpha,\beta_i)\pi(x_i|\lambda_i) \quad \text{due to the hierarchical structure of the model}$$
$$\pi(\lambda_i|x,\alpha,\beta_i,a,b) \propto \frac{\beta_i^\alpha}{\Gamma(\alpha)}\lambda_i^{\alpha-1}e^{-\beta_i\lambda_i}\frac{\lambda_i^{x_i}}{x_i!}e^{-\lambda_i}$$
$$\pi(\lambda_i|x,\alpha,\beta_i,a,b) \propto \lambda_i^{\alpha+x_i-1}e^{-(\beta_i+1)\lambda_i}$$

Hence, we see that : $\boxed{\lambda_i|x,\alpha,\beta_i,a,b \sim \mathcal{G}a(\alpha+x_i,\beta_i+1)}$

Full conditional for β_i The conditional distribution for β_i is :

$$\begin{aligned}\pi(\beta_i|x,\alpha,\lambda_i,a,b) &\propto \pi(x,\beta_i,\lambda_i|\alpha,a,b) \quad \text{(Bayes)}\\ &\propto \pi(x,\lambda_i|\beta_i,\alpha,a,b)\pi(\beta_i|\alpha,a,b) \quad \text{(Bayes)}\\ &\propto \pi(x|\lambda_i,\beta_i,\alpha,a,b)\pi(\lambda_i|\beta_i,\alpha,a,b)\pi(\beta_i|\alpha,a,b) \quad \text{(Bayes)}\end{aligned}$$

Given all parameters, X_i is independent of the other X_k $(k \neq i)$ due to the hierarchical structure of the model, hence :

$$\begin{aligned}\pi(\beta_i|x,\alpha,\lambda_i,a,b) &\propto \pi(x_i|\lambda_i,\beta_i,\alpha,a,b)\pi(x_{-i}|\lambda_i,\beta_i,\alpha,a,b)\pi(\lambda_i|\beta_i,\alpha,a,b)\pi(\beta_i|\alpha,a,b)\\ &\propto \pi(x_i|\lambda_i,\beta_i,\alpha,a,b)\pi(\lambda_i|\beta_i,\alpha,a,b)\pi(\beta_i|\alpha,a,b) \quad \text{(same argument as above for } \lambda_i)\end{aligned}$$

Since $x_i|\lambda_i,\beta_i,\alpha,a,b \sim \mathcal{P}(\lambda_i)$, $\lambda_i|\beta_i,\alpha,a,b \sim \mathcal{G}a(\alpha,\beta_i)$ and $\beta_i|\alpha,a,b \sim \mathcal{G}a(a,b)$, we get :

$$\begin{aligned}&\propto \frac{\lambda_i^{x_i}}{x_i!}e^{-\lambda_i} \times \frac{\beta_i^\alpha}{\Gamma(\alpha)}\lambda_i^{\alpha-1}e^{-\beta_i\lambda_i} \times \frac{b^a}{\Gamma(a)}\beta_i^{a-1}e^{-b\beta_i}\\ &\propto e^{-\beta_i\lambda_i}\beta_i^\alpha\beta_i^{a-1}e^{-b\beta_i}\\ &\propto \beta_i^{\alpha+a-1}e^{-(\lambda_i+b)\beta_i}\end{aligned}$$

Hence, we see that : $\boxed{\beta_i|x,\alpha,\lambda_i,a,b \sim \mathcal{G}a(\alpha+a,\lambda_i+b)}$

(b) Gibbs-Sampling

Using the result of (a), we can implement a Gibbs sampler to approximate for each herd i the posterior distribution $\pi(\lambda_i,\beta_i|x,\alpha,a,b)$. This also leads, if we only consider the sub processes $(\lambda_i^{(t)})$ and $(\beta_i^{(t)})$ alone to an approximation of both posterior distributions $\pi(\lambda_i|x,\alpha,a,b)$ and $\pi(\beta_i|x,\alpha,a,b)$.

We use $\alpha = 0.1$ and $a = b = 1$ as suggested in the exercise.

The corresponding R code is :

```
######################## Problem 10.10 #####################
############################################################

### Question (b) : Gibbs-Sampler with alpha = 0.1 and a = b = 1

data_herds = c(rep(0,7), rep(1, 12), rep(2,8), rep(3,9), rep(4,8), rep(5,8), rep
    (6,9), rep(7,6),
               rep(8,5), rep(9,3), rep(10,4), rep(11,7), rep(12,4), rep(13,5), rep
                   (14,2), rep(15,1),
               rep(16,4), rep(17,3), rep(18,3), rep(19,4), rep(20,2), rep(21,2),
                   rep(22,4), rep(23,1),
               rep(25,6))

length(data_herds)

gibbs <- function(n_sim, alpha, a, b, x){
  #initialize
  result <- t(as.matrix(c(1, 1)))
  for (t in 2:n_sim){
    given_beta = result[(t-1),2] + 1
    lambda_t <- rgamma(1, shape = alpha + x, rate = given_beta)
    beta_t <- rgamma(1, shape = alpha + a, lambda_t + b)
```

```
    vector_t ← c(lambda_t, beta_t)
    result ← rbind(result, vector_t)
  }
  return(result)
}

#test with x = data_herds[1]
#test ← gibbs(100, 0.1, 1, 1,  data_herds[1])

gibbs_full_data ← function(n_sim, alpha, a, b, data_herds){
  parameters = list()
  for (i in 1:127){
    x = data_herds[i]
    parameters[[i]] ← gibbs(n_sim, alpha, a, b, x)
  }
  return(parameters)
}

#Results
n_sim = 10000
alpha = 0.1
a = 1
b = 1

gibbs_questionb = gibbs_full_data(n_sim, alpha, a, b, data_herds)

result_1 ← matrix(unlist(gibbs_questionb[1]), ncol = 2, byrow = FALSE)

#Posterior of lambda 1
library(ggplot2)
qplot( result_1[,1], geom = 'histogram', bins = 100, fill = I("#FF6666"), xlab = '
    Value of lambda 1')
```

The result on the posterior of parameter λ_i for different values of i (i.e : for different herds) is available on Figure 1. We see that depending on the number of occurrences of mastisis cases in the considered herd, the form of the distribution changes and λ_i takes values that are mainly distributed around the number of mastisis cases actually observed in the herd. This is in line with the fact that we assumed $X_i \sim \mathcal{P}(\lambda_i)$ and the mean of such a Poisson distribution is λ_i.

Note that I did not show the result for λ_1 because it is almost the same histogram as for λ_5 since in my numbering of the herds, both herds have 0 occurrences of mastisis cases.

(c) Make histograms and monitor the convergence of λ_5, λ_{15} and β_{15}

I actually chose to also study other herds since herd 5 and herd 15 were very close in terms of results in my numbering of the observed herds.

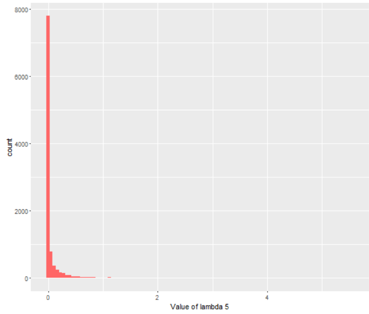
Results on λ_i Figure 1 displays the histogram of the posterior of λ_i for different herds i. Figure 2 displays the evolution of the mean of the simulated λ_i and therefore indicates the convergence of the algorithm. Note that I removed the first 4 simulations in order to get a graph with enough zoom in order to assess convergence. We see that after about 2000 iterations, the algorithm seems to have converged. However, if we remove the first 99 simulations and hence zoom on the zone where the mean has converged, we see that it stabilizes at a 10^{-2} precision only after about 7500 simulations (see Figure in Appendix).

Results on β_i Figure 3 displays the results on β_i for herd 15 and herd 45.

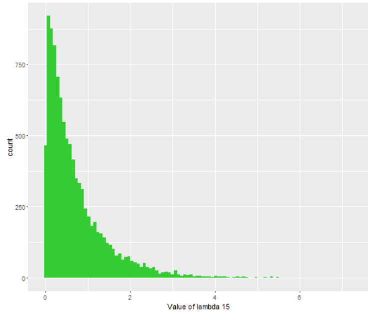
(d) Sensitivity to the parameters a, b and α

In order to assess the sensitivity to the parameters a, b and α, I kept 2 parameters fixed to the values suggested in question (b) and tried 3 different values for the third parameter of 3 different orders of magnitude : 0.1, 1 and 10. Results for herd 15 are displayed in Figure 4 (sensitivity to a), Figure 5 (sensitivity to b) and Figure 6 (sensitivity to α).

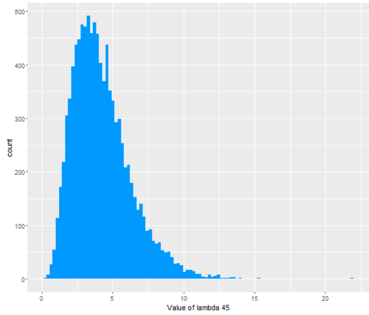
Figure 4,5 and 6 show us that λ_i is very sensitive to the choice of α (variation between 0 and 3 for the different tested values of α) but less to the choice of a, b (variation between 0 and 1.25 for the different tested values of a and b). This is not a surprise considering the hierarchical structure of the model. Nevertheless,



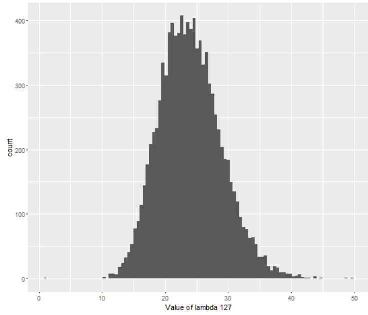
(a) λ_5 (Herd 5 had 0 occurrence of mastisis)



(b) λ_{15} (Herd 15 had 1 occurrence of mastisis)

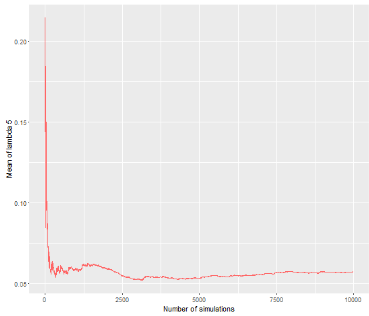


(c) λ_{45} (Herd 45 had 5 occurrences of mastisis)

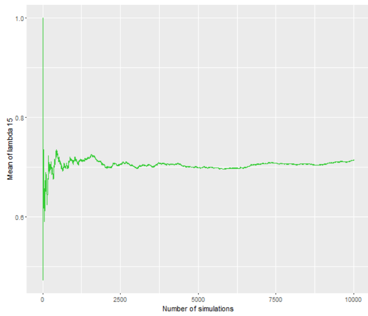


(d) λ_{127} (Herd 127 had 25 occurrences of mastisis)

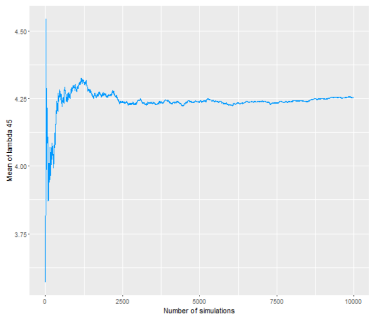
Figure 1 – Posterior of λ_i for different herds i



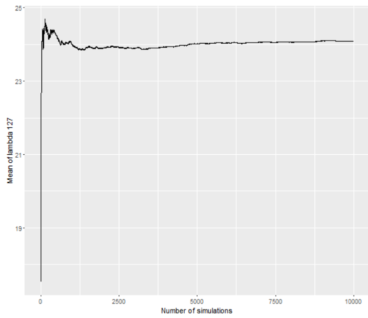
(a) λ_5 (Herd 5 had 0 occurrence of mastisis)



(b) λ_{15} (Herd 15 had 1 occurrence of mastisis)



(c) λ_{45} (Herd 45 had 5 occurrences of mastisis)

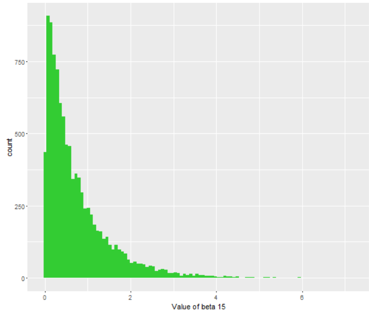


(d) λ_{127} (Herd 127 had 25 occurrences of mastisis)

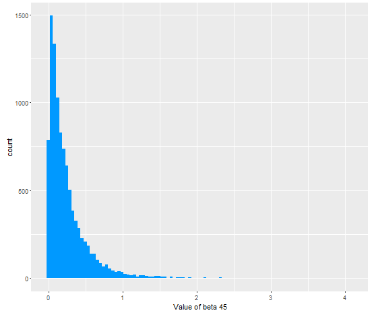
Figure 2 – Evolution of the mean of λ_i for different herds i

the 3 Figures show that results on λ_i and β_i remain very sensitive to the choice of a, b and α.

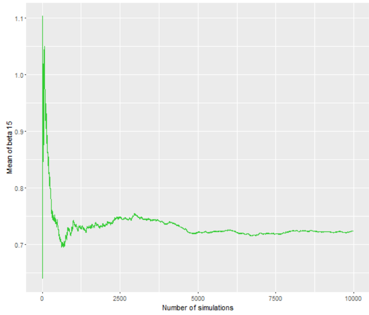
This sensitivity is actually not a surprise, as the considered model is based on a very informative prior on λ_i. All hyperparameters except β_i are indeed fixed to an arbitrary value. Moreover, the data used to estimate λ_i and β_i for each i is also very poor since we only have one observation per herd i (only one year of observation). This experiment therefore shows that this hierarchical specification with fixed parameters $a, b = 1$ and $\alpha = 0.1$ needs to be carefully justified before estimating the model, otherwise the results cannot be interpreted in practice.



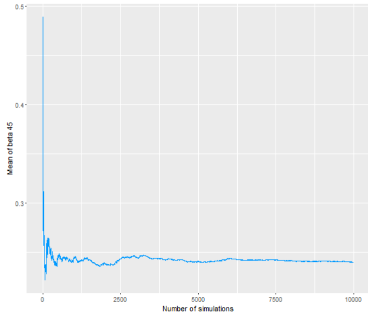
(a) Posterior of β_{15} (Herd 15 had 1 occurrence of mastisis)



(b) Posterior of β_{45} (Herd 45 had 5 occurrences of mastisis)

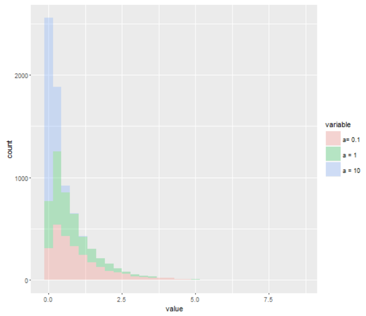


(c) Evolution of the mean of β_{15}

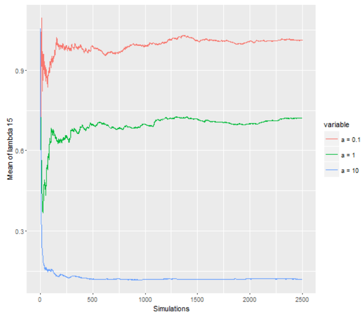


(d) Evolution of the mean of β_{45}

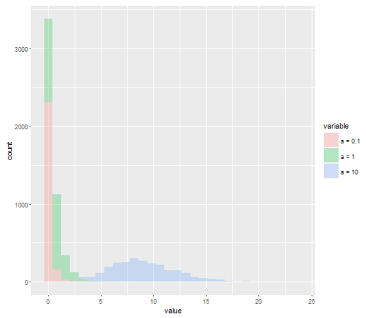
Figure 3 – Results on β_i for different herds i



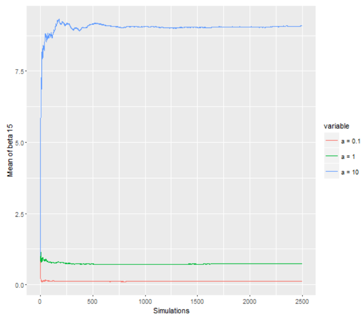
(a) Impact on λ_{15} posterior



(b) Impact on λ_{15} mean evolution



(c) Impact on β_{15} posterior



(d) Impact on β_{15} mean evolution

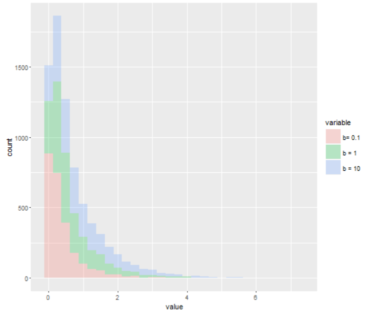
Figure 4 – Impact of the choice of a on results for herd 15

10.15 Metropolis-Hastings with Markov transition kernel and acceptance probability depending on the stationary distribution of the Markov transition kernel

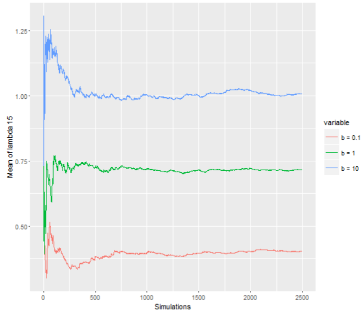
Let $K(\cdot, \cdot)$ a Markov transition kernel with stationary distribution g.

We are considering the following modified Metropolis-Hastings algorithm :

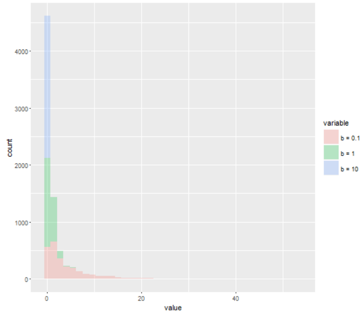
1. Generate $Y_t \sim K(x^{(t)}, y)$



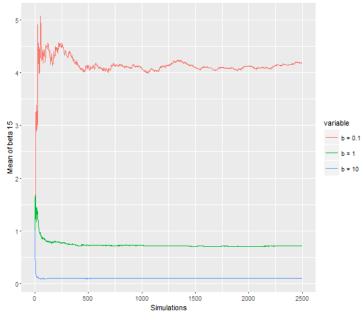
(a) Impact on λ_{15} posterior



(b) Impact on λ_{15} mean evolution

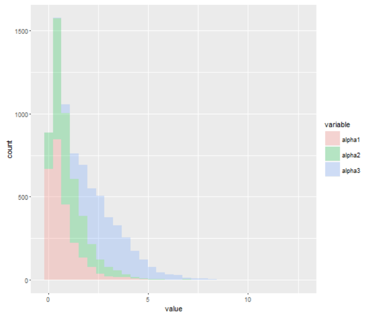


(c) Impact on β_{15} posterior

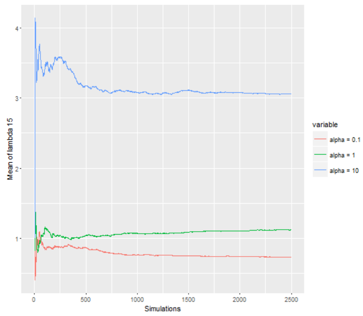


(d) Impact on β_{15} mean evolution

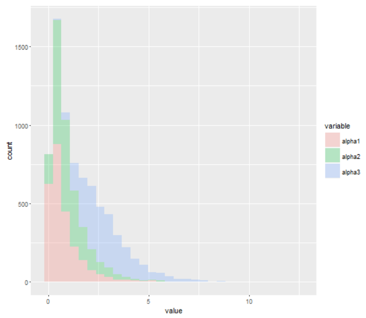
Figure 5 – Impact of the choice of b on results for herd 15



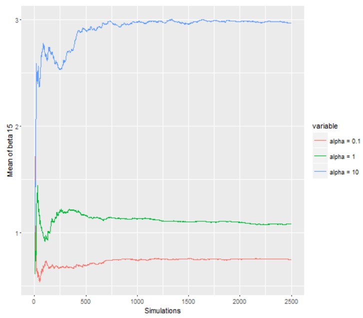
(a) Impact on λ_{15} posterior



(b) Impact on λ_{15} mean evolution



(c) Impact on β_{15} posterior



(d) Impact on β_{15} mean evolution

Figure 6 – Impact of the choice of α on results for herd 15

2. Accept $X^{(t+1)} = y_t$ with probability $\rho(x^{(t)}, y_t) = \min\left(1, \frac{f(y_t)g(x^{(t)})}{f(x^{(t)})g(y_t)}\right)$.
Otherwise take $X^{(t+1)} = x^{(t)}$.

The goal of the exercise is to show that it is a valid MCMC algorithm for the stationary distribution f, i.e that the resulting Markov chain actually converges to the distribution of interest f.

I will assume that $K(\cdot,\cdot)$ and g verify the detailed condition balance. I am aware that this assumption is strong, however I was not able to find a proof without assuming this. In the case of a Markov chain corresponding to one of the subchains of a 2-stage Gibbs sampler, this assumption is actually verified as showed in Problem 9.4 above. We could therefore use the proposed algorithm with a candidate Markov chain generated as a subchain of a 2-stage Gibbs sampler. This could also explain the link between this exercise and Gibbs sampling. The proposed Metropolis algorithm is therefore in this case a way to simulate from a distribution of interest f using a Gibbs sampler with marginal density g for one of the subchains to generate a candidate.

Step 1 - Show that f is the stationary distribution of the resulting Markov chain The transition kernel of the resulting Metropolis-chain $(X^{(t)})$ is similar to the one obtained in the classic Metropolis-Hastings algorithm :

$$\tilde{K}(x,y) = \rho(x,y)K(x,y) + (1-r(x))\delta_x(y) \quad \text{where } r(x) = \int \rho(x,y)K(x,y)dy$$

We will show that the resulting chain $(X^{(t)})$ and f verify the detailed condition balance, i.e :

$$\tilde{K}(x,y)f(x) = \tilde{K}(y,x)f(y)$$

A consequence of this will be that there exists a stationary distribution for the resulting chain $(X^{(t)})$ and that f is this stationary distribution.

Show that $\rho(x,y)K(x,y)f(x) = \rho(y,x)K(y,x)f(y)$

$$\begin{aligned}
\rho(x,y)K(x,y)f(x) &= \min\Big(1, \frac{f(y)g(x)}{f(x)g(y)}\Big)K(x,y)f(x) \\
&= \min\Big(f(x)K(x,y), \frac{f(y)g(x)K(x,y)}{g(y)}\Big)
\end{aligned}$$

Since $K(\cdot,\cdot)$ and g verify the detailed condition balance, we have $K(x,y)g(x) = K(y,x)g(y)$, hence :

$$\begin{aligned}
\rho(x,y)K(x,y)f(x) &= \min\Big(f(x)\frac{K(y,x)g(y)}{g(x)}, \frac{f(y)g(y)K(y,x)}{g(y)}\Big) \\
&= \min\Big(f(x)\frac{K(y,x)g(y)}{g(x)}, f(y)K(y,x)\Big) \\
&= \min\Big(f(x)\frac{g(y)}{g(x)}, f(y)\Big)K(y,x) \\
&= \min\Big(\frac{f(x)}{f(y)}\frac{g(y)}{g(x)}, 1\Big)K(y,x)f(y)
\end{aligned}$$

$$\boxed{\rho(x,y)K(x,y)f(x) = \rho(y,x)K(y,x)f(y)}$$

Show that $(1-r(x))\delta_x(y)f(x) = (1-r(y))\delta_y(x)f(y)$ This is straightforward since $\delta_x(y) = \delta_y(x)$:

$$\begin{aligned}
(1-r(x))\delta_x(y)f(x) &= \begin{cases} (1-r(x))f(x) \text{ if } x=y \\ 0 \text{ otherwise} \end{cases} \\
&= \begin{cases} (1-r(y))f(y) \text{ if } x=y \\ 0 \text{ otherwise} \end{cases}
\end{aligned}$$

$$\boxed{(1-r(x))\delta_x(y)f(x) = (1-r(y))\delta_y(x)f(y)}$$

Hence : $\boxed{\tilde{K}(x,y)f(x) = \tilde{K}(y,x)f(y)}$ The detailed balance condition is thus verified. As a consequence, f is the stationary distribution of the resulting Metropolis chain.

Step 2 - Show that in most cases the resulting Markov chain is indeed ergodic and convergence is guaranteed Using Theorem 6.63, the almost sure convergence of the empirical average is guaranteed if $(X^{(t)})$ is Harris recurrent. We will therefore show here that $(X^{(t)})$ is Harris recurrent in most cases, following a similar proof as in Chapter 7 for Theorem 7.4.

Show that is most cases $(X^{(t)})$ is irreducible

Additional assumption : we will assume here that $Supp(f) \subset Supp(g)$. In practice, this assumption will be verified in most cases: we only need to choose a "candidate stationary distribution" g able to generate values in all subsets of the support of the distribution of interest f, which is obvious if our goal is to simulate wrt f.

We will also assume that we initialize the algorithm with $x^{(0)} \in Supp(f)$.

We will show that with those 2 additional assumptions, $(X^{(t)})$ is f-irreducible.

Reminder - Irreducibility Let $\mathcal{X}$ be the state-space of $(X^{(t)})$. $(X^{(t)})$ is f-irreducible if and only if : $\forall A \in \mathcal{B}(\mathcal{X})/\psi_f(A) > 0$ (where ψ_f is the probability measure corresponding to the distribution f or in other words $\psi_f(A) = \int_A f(x)dx$), there exists n such that $\tilde{K}^n(x, A) > 0$ for all $x \in \mathcal{X}$.

We will actually show that : $\forall A \subset Supp(f)/\lambda(A) > 0$ (where λ is the Lebesgue measure), there exists n such that $\tilde{K}^n(x^{(t)}, A) > 0$ for all $x^{(t)} \in Supp(f)$. Consequently, $(X^{(t)})$ will be f-irreducible. Indeed :
a. Since we initialized the algorithm with $x^{(0)} \in Supp(f)$ (Second additional assumption), we get:

$$x^{(1)} = \begin{cases} x^{(0)} \in Supp(f) \text{ if } y_1 \text{ has been rejected} \\ y_1 \text{if } y_1 \text{ has been accepted and thus necessarily } \rho(x^{(0)}, y_1) > 0 \Rightarrow f(y_1) > 0 \Rightarrow y_1 \in Supp(f) \end{cases}$$

Hence, by iteration, all $x^{(t)}$ belong to $Supp(f)$, or in other words : $\mathcal{X} \subset Supp(f)$. Therefore, it is sufficient to show that $\tilde{K}^n(x^{(t)}, A) > 0$ for all $x^{(t)} \in Supp(f)$ instead of for all $x^{(t)} \in \mathcal{X}$.
b. $\psi_f(A) > 0 \Rightarrow \int_A f(x)dx > 0 \Rightarrow \exists B \subset A/B \subset Supp(f)$ and $\lambda(B) > 0$. Hence, if we are able to show that B can be reached in a finite number of steps, we will have shown as a consequence that A can be reached in the same finite number of steps. Therefore, in order to prove the irreducibility of the resulting chain, it is sufficient to show that : $\forall A \subset Supp(f)/\lambda(A) > 0$, A can be reached in a finite number of steps.

Let $A \subset Supp(f)/\lambda(A) > 0$. Hence : $A \subset Supp(g)$ (First additional assumption). Let $x \in Supp(f)$.

We will first show that A can be reached in a single step by $K(x, \cdot)$ (the candidate Markov chain) using proof by contradiction. Assume $K(x, A) = 0$. Then :

$$\int_{\mathcal{X}} K(x, A)g(x)dx = 0$$

Since g is the invariant distribution of the candidate chain, we have : $\int_{\mathcal{X}} K(x, A)g(x)dx = \int_A g(x)dx$. Hence :

$$\int_A g(x)dx = 0 \text{ which is in contradiction with } A \subset Supp(g)$$

Therefore : $K(x, A) > 0$, or in other words A can be reached in a single step by the candidate Markov chain.

We will now show that the acceptance probability is strictly positive when A is reached by the candidate. $x \in Supp(f) \Rightarrow x \in Supp(g)$ (First additional assumption). Hence : $g(x) > 0$. Moreover, if A has been reach by the candidate, then the generated y belongs to A. Therefore, $y \in Supp(f)$, which is equivalent to $f(y) > 0$. Hence, the acceptance probability $\rho(x, y)$ is strictly positive. This ensures that all candidate simulations belonging to A can be accepted.

Hence, A can be reached in a single step by the resulting Metropolis chain.

Conclusion : Under those two additional assumptions that are often verified in practice, the resulting chain $(X^{(t)})$ is necessarily f-irreducible.

<u>Show that if $(X^{(t)})$ is irreducible, then it is Harris recurrent</u>
The proof of this result is similar to the proof of Lemma 7.3 in the book.

<u>Conclusion</u>
Under those two additional assumptions, the convergence of the Metropolis algorithm mentioned in the exercise is guaranteed.

10.18 Tobit model

The tobit model is defined by :

$$y_i = \max(0, y_i^*) \text{ where } y_i^* \text{ is a latent variable (that we do not observe) such that } y_i^* \sim \mathcal{N}(x_i^t\beta, \sigma^2)$$

.

The goal of the exercise is to prove the validity of a given algorithm to approximate the posterior distribution of (β, σ), i.e to approximate $\pi(\beta, \sigma|y, x)$.

We will show that this algorithm corresponds to the Gibbs Sampler which simulates the random vector (θ, y^*) given the observed data x, y, where $\theta = (\beta, \sigma)$ is the vector of paramaters. This approach is motivated by the fact that $\pi(\beta, \sigma, y^*|y, x)$ is a completion of $\pi(\beta, \sigma|y, x)$, since :

$$\pi(\beta, \sigma|y, x) = \int \pi(\beta, \sigma, y^*|y, x)dy^*$$

As explained in Chapter 10, theoretical results ensure that the subchain (β, σ) converges to the distribution of interest $\pi(\beta, \sigma|y, x)$.

Completion Gibbs sampler The completion Gibbs Sampling algorithm is defined in our case by :
Given $(\theta^{(t)}, y^{*(t)})$:
1. Simulate $y^{*(t+1)} \sim \pi(y^*|\theta^{(t)}, y, x)$
2. Simulate $\theta^{(t+1)} \sim \pi(\theta|y^{*(t+1)}, y, x)$

Full conditional of θ The full conditional of θ is :

$$\begin{aligned}\pi(\theta|y^*, y, x) &= \pi(\theta|y^*, x) \quad \text{since } y^* \text{ contains all the information of } y \\ &\propto \pi(y^*|\theta, x)\pi(\theta|x) \quad \text{(Bayes' theorem)} \\ &\propto \prod_{i=1}^{n} \pi(y_i^*|\theta, x)\pi(\theta|x) \quad \text{since all } y_i^* \text{ are independent (independent individuals)} \\ &\propto \prod_{i=1}^{n} \left(\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y_i^*-x_i^t\beta)^2}{2\sigma^2}}\right)\pi(\beta, \sigma|x) \quad \text{since } y_i^* \sim \mathcal{N}(x_i^t\beta, \sigma^2)\end{aligned}$$

$$\boxed{\pi(\theta|y^*, y, x) \propto \sigma^{-n} e^{-\frac{\sum_{i=1}^{n}(y_i^*-x_i^t\beta)^2}{2\sigma^2}} \pi(\beta, \sigma|x)}$$

Full conditional of y^* The full conditional of y^* is :

$$\begin{aligned}\pi(y^*|\theta, x, y) &= \prod_{i=1}^{n} \pi(y_i^*|\theta, x, y) \quad \text{since all } y_i^* \text{ are independent (independent individuals)} \\ \pi(y^*|\theta, x, y) &= \prod_{i=1}^{n} \pi(y_i^*|\theta, x_i, y_i) \quad \text{since individuals are independent so } y_i^* \text{ only depends on } (x_i, y_i)\end{aligned}$$

In practice, simulating y^* according to its full conditional is therefore equivalent to simulating independently each y_i^* according to $\pi(y_i^*|\theta, x_i, y_i)$.

Compute $\pi(y_i^*|\theta, x_i, y_i)$

$$\pi(y_i^*|\theta, x_i, y_i) \propto \pi(y_i|y_i^*, \theta, x_i)\pi(y_i^*|\theta, x_i) \quad \text{(Bayes' theorem)}$$

Note that conditional to (y_i^*, θ, x_i), Y_i is a constant variable since all the information we need to determine Y_i is contained in y_i^*, as $Y_i = \max(0, y_i^*)$. Therefore, $\pi(y_i|y_i^*, \theta, x_i)$ is the distribution of a constant variable which is equal to 0 if $y_i^* \leq 0$ and to y_i^* if $y_i^* > 0$. Hence :

$$\begin{aligned}\pi(y_i|y_i^*, \theta, x_i) &= \begin{cases} \delta_0(y_i) \text{ if } y_i^* \leq 0 \\ \delta_{y_i^*}(y_i) \text{ if } y_i^* > 0 \end{cases} \\ &= \mathbb{1}_{(y_i^* \leq 0)}\delta_0(y_i) + \mathbb{1}_{(y_i^* > 0)}\delta_{y_i^*}(y_i) \\ &= \mathbb{1}_{(y_i^* \leq 0)}\mathbb{1}_{(y_i = 0)} + \mathbb{1}_{(y_i^* > 0)}\mathbb{1}_{(y_i^* = y_i)} \\ &= \mathbb{1}_{(y_i^* \leq 0)}\mathbb{1}_{(y_i = 0)} + \mathbb{1}_{(y_i > 0)}\mathbb{1}_{(y_i^* = y_i)} \\ \pi(y_i|y_i^*, \theta, x_i) &= \begin{cases} \mathbb{1}_{(y_i^* \leq 0)} \text{ if } y_i = 0 \\ \mathbb{1}_{(y_i^* = y_i)} \text{ if } y_i > 0 \end{cases}\end{aligned}$$

Hence :

$$\pi(y_i^*|\theta, x_i, y_i) \propto \begin{cases} \mathbb{1}_{(y_i^* \leq 0)} \pi(y_i^*|\theta, x_i) \text{ if } y_i = 0 \\ \mathbb{1}_{(y_i^* = y_i)} \pi(y_i^*|\theta, x_i) \text{ if } y_i > 0 \end{cases}$$

If we denote ϕ the density of a $\mathcal{N}(0,1)$, since $y_i^*|\beta, \sigma, x_i \sim \mathcal{N}(x_i^t\beta, \sigma^2)$:

$$\pi(y_i^*|\theta, x_i, y_i) \propto \begin{cases} \mathbb{1}_{(y_i^* \leq 0)} \phi(\frac{y_i^* - x_i^t\beta}{\sigma}) \text{ if } y_i = 0 \\ \mathbb{1}_{(y_i^* = y_i)} \pi(y_i|\theta, x_i) \text{ if } y_i > 0 \end{cases}$$

$$\pi(y_i^*|\theta, x_i, y_i) \propto \begin{cases} \mathbb{1}_{(y_i^* \leq 0)} \phi(\frac{y_i^* - x_i^t\beta}{\sigma}) \text{ if } y_i = 0 \\ \mathbb{1}_{(y_i^* = y_i)} \text{ if } y_i > 0 \text{ (we can remove } \pi(y_i|\theta, x_i) \text{ as it does not depend on } y_i^*) \end{cases}$$

Therefore :

$$\begin{cases} y_i^*|\theta, x_i, y_i \sim \mathcal{N}_-(x_i^t\beta, \sigma^2, 0) \text{ (a truncated normal with maximal value 0) if } y_i = 0 \\ y_i^*|\theta, x_i, y_i \text{ is a constant variable equal to } y_i \text{ if } y_i > 0 \end{cases}$$

Resulting algorithm and conclusion The resulting Gibbs Sampling algorithm is therefore :
Given $(\theta^{(t)}, y^{*(t)})$:
1. For all $i = 1, ...n$: Simulate $y_i^{*(t+1)} \sim \mathcal{N}_-(x_i^t\beta, \sigma^2, 0)$ if $y_i = 0$. Otherwise, take $y_i^{*(t+1)} = y_i$.
2. Simulate $(\beta^{(t+1)}, \sigma^{(t+1)}) \sim \pi(\beta, \sigma|y^{*(t+1)}, x)$ with :

$$\pi(\beta, \sigma|y^*, x) \propto \sigma^{-n} e^{-\frac{\sum_{i=1}^n (y_i^* - x_i^t\beta)^2}{2\sigma^2}} \pi(\beta, \sigma|x)$$

Note that we can of course choose a prior on (β, σ) which does not depend on the variables x, so choose a distribution that can be written as in the exercise $\pi(\beta, \sigma|x) = \pi(\beta, \sigma)$. By doing this, we thus obtain exactly the mentioned algorithm.

Conclusion : The algorithm mentioned in the exercise corresponds exactly to the completion Gibbs sampler simulating (β, σ, y^*). This proves that this algorithm provides a valid approximation of the posterior distribution of (β, σ) by taking the distribution of the (β, σ) subchain.

Finally note that if the first step of the algorithm is quite "easy" to implement, using for example Accept-Reject to simulate the truncated normals, the second step is less obvious. It will mostly depend on the choice of the prior. A good thing would be to find a conjugate prior to ensure that the final distribution is known and hence easy to simulate in most cases. A classic choice is to take this prior equal to a Normal Inverse Gamma distribution (note that we consider its multivariate form since β is a multivariate random vector). This would ensure that the posterior $\pi(\beta, \sigma|y^*, x)$ is also a Normal Inverse Gamma.

Additional Exercise - Using Gibbs sampling to initialize a Particle Filter

In another course at ENSAE about Hidden Markov Models, I am currently working on particle filtering applied to electricity load forecasting based on the article "On particle filters applied to electricity load forecasting" from Tristan Launay, Anne Philippe and Sophie Lamarche. The article is available here : https://arxiv.org/abs/1210.0770.

In this article, the authors mention page 19 and 20 the need to choose a "good" initialization for the particle filter in order to avoid degeneracy after only the very first step (only one particle is selected at the first step and its weight is therefore equal to 1). They suggest to use an MCMC software such as BUGS or JAGS to estimate the smoothed distribution up to a certain time $n_0 - 1$, and then to use the resulting simulations corresponding to time $n_0 - 1$ in order to initialize the particle filter. They do not exactly precise the MCMC method they used to achieve this, but as the model is hierarchical, it seems that they used a Gibb sampler. I would therefore like to give more details here on how to achieve this using a Gibbs sampler and would actually be happy to get any feedback about this (especially if I am mistaken !).

Let y_n the observed electricity load at time n. Let $\theta = (\sigma_s, \sigma_g, u^{heat}, \kappa, \sigma)$ a fixed known parameter (which will actually be estimated using PMCMC). We also assume that the temperatures and daytypes are like fixed known parameters of the model : they are not considered as observations of random variables. Finally, we chose to neglect the cooling effect, since we did not have access to the corresponding data. Hence, our model is :

$$y_n = x_n + \nu_n \quad \text{where } \nu_n \sim \mathcal{N}(0, \sigma^2)$$

The state x_n is made of 2 parts :

$$x_n = x_n^{season} + x_n^{heat}$$

which are defined by :

$$x_n^{season} = s_n \cdot \kappa_{daytype_n}$$

$$x_n^{heat} = g_n^{heat}(T_n^{heat} - u^{heat})\mathbb{1}_{(u^{heat} > T_n^{heat})}$$

The various components are following the dynamic :

$$s_n = s_{n-1} + \epsilon_n^s \quad \text{where } \epsilon_n^s \sim \mathcal{N}(0, \sigma_{s,n}^2,]-s_{n-1}, +\infty[)$$

$$g_n^{heat} = g_{n-1}^{heat} + \epsilon_n^g \quad \text{where } \epsilon_n^g \sim \mathcal{N}(0, \sigma_{g,n}^2,]-\infty, -g_{n-1}^{heat}[)$$

$$\sigma_{s,n} = \sigma_{s,n-1} = \sigma_{s,*}$$

$$\sigma_{g,n} = \sigma_{g,n-1} = \sigma_{g,*}$$

With initial distribution :

$$s_0 \sim \mathcal{N}(0, 10^8, \mathbb{R}_+)$$

$$g_0^{heat} \sim \mathcal{N}(0, 10^8, \mathbb{R}_-)$$

$$\sigma_{s,*}^2 \sim \mathcal{IG}(10^{-2}, 10^{-2})$$

$$\sigma_{g,*}^2 \sim \mathcal{IG}(10^{-2}, 10^{-2})$$

NB : $\mathcal{N}(\mu, \Sigma, S)$ denotes the truncated Gaussian distribution with mean μ and variance Σ with support S.

Note that we assume here that the variances $\sigma_{g,n}^2$ and $\sigma_{g,n}^2$ are constant at each time n, as suggested page 20 for the MCMC estimation. In the "true" model, they are actually dynamic but the authors point out that the MCMC estimation did not converge using this additional layer of dynamic and $n_0 = 365$. The idea to get a proper initialization distribution for the particle filter is thus to focus here on the results on the components s and g^{heat} and to add an additional prior on σ_{g,n_0-1}^2 and σ_{g,n_0-1}^2 with parameters based on the empirical errors ϵ_n^s and ϵ_n^g obtained from the Gibbs sampler (see page 20 of the article for more details).

The state at time n of the model is defined by the vector $(s_n, g_n^{heat}, \sigma_{s,n}, \sigma_{g,n})$, which is equal in our case to $(s_n, g_n^{heat}, \sigma_{s,*}, \sigma_{g,*})$. Our goal is to simulate particles wrt the smoothed distribution $\pi(s_{0:n_0-1}, g_{0:n_0-1}^{heat}, \sigma_{s,*}, \sigma_{g,*} | y_{0:n_0-1})$ using a Gibbs sampler, which will consequently provide us simulations from the filtered distribution on a "diminished state" $\pi(s_{n_0-1}, g_{n_0-1}^{heat} | y_{0:n_0-1})$ if we only consider the sub-process $(s_{n_0-1}, g_{n_0-1}^{heat})$ resulting from the Gibbs sampler. (Hence, we can view our Gibbs sampler as a completion Gibbs sampler since we are only interested in 2 specific random variables).

Note that we see here that Gibbs sampling actually offers a filtering solution to our Hidden Markov Model just like Particle Filters ! Nevertheless, the main difference is that it requires simulations from the whole smoothed distribution and hence is not efficient compared to Particle Filtering which provides online inference (the next state can be simulated by only using the simulations of the previous state). Online estimation of electricity loads and forecasting are therefore more manageable using Particle Filtering.

In order to implement a Gibbs sampler, we need to compute the full conditionals of the the smoothed distribution $\pi(s_{0:n_0-1}, g_{0:n_0-1}^{heat}, \sigma_{s,*}, \sigma_{g,*} | y_{0:n_0-1})$.

The algorithm will then be (for θ and $\sigma_{g,*}^2, \sigma_{s,*}^2$ fixed) :
Initialize $(s_0^{(0)}, ..., s_{n_0-1}^{(0)}, g_0^{heat(0)}, ..., g_{n_0-1}^{heat(0)}, , \sigma_{s,*}^{(0)}, \sigma_{g,*}^{(0)})$
For $j = 1, 2, ..., n_0 - 1$:
Step 1 : $S_0^{(j+1)} \sim \pi(s_0 | y_{0:n_0-1}, g_{0:n_0-1}^{heat(j)}, s_{-0}^{(j)}, \sigma_{s,*}^{(j)}, \sigma_{g,*}^{(j)})$ where $s_{-0}^{(j)}$ denotes $(s_k^{(j)})_{k \neq 0}$
Step 2: $S_1^{(j+1)} \sim \pi(s_1 | y_{0:n_0-1}, g_{0:n_0-1}^{heat(j)}, s_0^{(j+1)}, s_2^{(j)}, ..., s_{n_0-1}^{(j)}, \sigma_{s,*}^{(j)}, \sigma_{g,*}^{(j)})$
...
Step n_0 : $S_{n_0-1}^{(j+1)} \sim \pi(s_{n_0-1} | y_{0:n_0-1}, g_{0:n_0-1}^{heat(j)}, s_{-(n_0-1)}^{(j+1)}, \sigma_{s,*}^{(j)}, \sigma_{g,*}^{(j)})$ where $s_{-(n_0-1)}^{(j+1)}$ denotes $(s_k^{(j+1)})_{k \neq n_0-1}$
Step $n_0 + 1$: $G_0^{heat(j+1)} \sim \pi(g_0^{heat} | y_{0:n_0-1}, s_{0:n_0-1}^{(j+1)}, g_{-0}^{heat(j)}, \sigma_{s,*}^{(j)}, \sigma_{g,*}^{(j)})$ where $g_{-0}^{heat(j)}$ denotes $(g_k^{heat(j)})_{k \neq 0}$
...

Step $2n_0$: $G^{heat(j+1)}_{n_0-1} \sim \pi(g^{heat}_{n_0-1}|y_{0:n_0-1}, s^{(j+1)}_{0:n_0-1}, g^{heat(j+1)}_{-(n_0-1)}, \sigma^{(j)}_{s,*}, \sigma^{(j)}_{g,*})$ where $g^{heat(j+1)}_{-(n_0-1)}$ denotes $(g^{heat(j+1)}_k)_{k\neq n_0-1}$
Step $2n_0+1$: $\sigma^{(j+1)}_{s,*} \sim \pi(\sigma_{s,*}|y_{0:n_0-1}, s^{(j+1)}_{0:n_0-1}, g^{heat(j+1)}_{0:n_0-1}, \sigma^{(j)}_{g,*})$
Step $2n_0+2$: $\sigma^{(j+1)}_{g,*} \sim \pi(\sigma_{g,*}|y_{0:n_0-1}, s^{(j+1)}_{0:n_0-1}, g^{heat(j+1)}_{0:n_0-1}, \sigma^{(j+1)}_{s,*})$

Full conditional of s_i when $i \neq 0$ We will use denote s_{-i} the vector $(s_k)_{k\neq i}$ in order to simplify the notations in the calculus.

$$\begin{aligned}
\pi(s_i|s_{-i}, g^{heat}_{0:n_0-1}, \sigma_{s,*}, \sigma_{g,*}, y_{0:n_0-1}) &\propto \pi(s_{0:n_0-1}, y_{0:n_0-1}|g^{heat}_{0:n_0-1}, \sigma_{s,*}, \sigma_{g,*}) \quad \text{(Bayes)}\\
&\propto \pi(y_{0:n_0-1}|s_{0:n_0-1}, g^{heat}_{0:n_0-1}, \sigma_{s,*}, \sigma_{g,*})\pi(s_{0:n_0-1}|g^{heat}_{0:n_0-1}, \sigma_{s,*}, \sigma_{g,*})
\end{aligned}$$

$s_{0:n_0-1}$ does not depend on $g^{heat}_{0:n_0-1}$ and $\sigma_{g,*}$ due to the hierarchical structure of the model, hence :

$$\begin{aligned}
&\propto \pi(y_{0:n_0-1}|s_{0:n_0-1}, g^{heat}_{0:n_0-1}, \sigma_{s,*}, \sigma_{g,*})\pi(s_{0:n_0-1}|\sigma_{s,*})\\
&\propto \Big(\prod_{k=0}^{n_0-1} \pi(y_k|s_k, g^{heat}_k)\Big)\pi(s_0|\sigma_{s,*})\pi(s_1|s_0, \sigma_{s,*})...\pi(s_{n_0-1}|s_{n_0-2}, \sigma_{s,*})
\end{aligned}$$

We can now remove all components which do not depend on s_i :

$$\begin{aligned}
&\propto \pi(y_i|s_i, g^{heat}_i)\pi(s_i|s_{i-1}, \sigma_{s,*})\pi(s_{i+1}|s_i, \sigma_{s,*})\\
&\propto \exp(-\frac{1}{2\sigma^2}(y_i - s_i\cdot\kappa_{daytype_i} - g^{heat}_i(T^{heat}_i - u^{heat})\mathbb{1}_{(u^{heat}>T^{heat}_i)})^2)\\
&\times \exp(-\frac{1}{2\sigma^2_{s,*}}(s_i - s_{i-1})^2)\exp(-\frac{1}{2\sigma^2_{s,*}}(s_{i+1} - s_i)^2)\mathbb{1}_{s_i\geq 0}\mathbb{1}_{s_{i+1}\geq 0}\\
&\propto \exp(-\frac{1}{2\sigma^2}[-2s_i\kappa_{daytype_i}(y_i - g^{heat}_i(T^{heat}_i - u^{heat})\mathbb{1}_{(u^{heat}>T^{heat}_i)})])\\
&\times \exp(-\frac{\kappa^2_{daytype_i}}{2\sigma^2}[s_i^2])\exp(-\frac{1}{2\sigma^2_{s,*}}[s_i^2 - 2s_i(s_{i-1}+s_{i+1})])\mathbb{1}_{s_i\geq 0}\\
&\propto \exp(-\frac{1}{2\sigma^2}[-2s_i\kappa_{daytype_i}(y_i - g^{heat}_i(T^{heat}_i - u^{heat})\mathbb{1}_{(u^{heat}>T^{heat}_i)})])\\
&\times \exp(-(\frac{\kappa^2_{daytype_i}}{2\sigma^2} + \frac{1}{2\sigma^2_{s,*}})[s_i^2])\exp(-\frac{1}{2\sigma^2_{s,*}}[-2s_i(s_{i-1}+s_{i+1})])\mathbb{1}_{s_i\geq 0}
\end{aligned}$$

Since $\frac{\kappa^2_{daytype_i}}{2\sigma^2} + \frac{1}{2\sigma^2_{s,*}} = \frac{\sigma^2+\sigma^2_{s,*}\cdot\kappa^2_{daytype_i}}{2\sigma^2_{s,*}\sigma^2}$, we get :

$$\begin{aligned}
&\propto \exp(-\frac{\sigma^2+\sigma^2_{s,*}\cdot\kappa^2_{daytype_i}}{2\sigma^2_{s,*}\sigma^2}[-2s_i\frac{\sigma^2_{s,*}\kappa_{daytype_i}(y_i - g^{heat}_i(T^{heat}_i - u^{heat})\mathbb{1}_{(u^{heat}>T^{heat}_i)})}{\sigma^2+\sigma^2_{s,*}\cdot\kappa^2_{daytype_i}}\\
&\times \exp(-\frac{\sigma^2+\sigma^2_{s,*}\cdot\kappa^2_{daytype_i}}{2\sigma^2_{s,*}\sigma^2}[s_i^2])\\
&\times \exp(-\frac{\sigma^2+\sigma^2_{s,*}\cdot\kappa^2_{daytype_i}}{2\sigma^2_{s,*}\sigma^2}[-2s_i\frac{(s_{i-1}+s_{i+1})\sigma^2}{\sigma^2+\sigma^2_{s,*}\cdot\kappa^2_{daytype_i}}])\mathbb{1}_{s_i\geq 0}\\
&\propto \exp(-\frac{\sigma^2+\sigma^2_{s,*}\cdot\kappa^2_{daytype_i}}{2\sigma^2_{s,*}\sigma^2}[s_i - M]^2)\mathbb{1}_{s_i\geq 0}
\end{aligned}$$

where $M = \frac{(s_{i-1}+s_{i+1})\sigma^2}{\sigma^2+\sigma^2_{s,*}\cdot\kappa^2_{daytype_i}} + \frac{\sigma^2_{s,*}\kappa_{daytype_i}(y_i - g^{heat}_i(T^{heat}_i - u^{heat})\mathbb{1}_{(u^{heat}>T^{heat}_i)})}{\sigma^2+\sigma^2_{s,*}\cdot\kappa^2_{daytype_i}}$

Hence, we see that : $\boxed{s_i|s_{-i}, g^{heat}_{0:n_0-1}, \sigma_{s,*}, \sigma_{g,*}, y_{0:n_0-1} \sim \mathcal{N}(M, \frac{\sigma^2_{s,*}\sigma^2}{\sigma^2+\sigma^2_{s,*}\cdot\kappa^2_{daytype_i}}, \mathbb{R}^+)}$

Full conditional of s_0 Similarly as in the above calculus, we get :

$$\pi(s_0|s_{-0}, g^{heat}_{0:n_0-1}, \sigma_{s,*}, \sigma_{g,*}, y_{0:n_0-1}) \propto \Big(\prod_{k=0}^{n_0-1} \pi(y_k|s_k, g^{heat}_k)\Big)\pi(s_0|\sigma_{s,*})\pi(s_1|s_0, \sigma_{s,*})...\pi(s_{n_0-1}|s_{n_0-2}, \sigma_{s,*})$$

The initial distribution of s_0 does not depend on $\sigma_{s,*}$, therefore :

$$\begin{aligned}
&\propto \pi(y_0|s_0, g_0^{heat})\pi(s_0|\sigma_{s,*})\pi(s_1|s_0,\sigma_{s,*}) \\
&\propto \pi(y_0|s_0, g_0^{heat})\pi(s_0)\pi(s_1|s_0,\sigma_{s,*}) \\
&\propto \exp(-\frac{1}{2\sigma^2}(y_0 - s_0\cdot\kappa_{daytype_0} - g_0^{heat}(T_0^{heat} - u^{heat})\mathbb{1}_{(u^{heat} > T_0^{heat})})^2) \\
&\times \exp(-\frac{s_0^2}{2\cdot 10^8})\exp(-\frac{1}{2\sigma_{s,*}^2}(s_1 - s_0)^2)\mathbb{1}_{s_0\geq 0}
\end{aligned}$$

Hence, by doing a similar calculus as above we get :

$$\propto \exp(-\frac{10^8\sigma_{s,*}^2\kappa_{daytype_0}^2 + \sigma^2\sigma_{s,*}^2 + 10^8\sigma^2}{2\cdot 10^8\cdot\sigma^2\sigma_{s,*}^2}[s_0 - M']^2)\mathbb{1}_{s_0\geq 0}$$

Where $M' = \frac{10^8\sigma^2 s_1 + 10^8\sigma_{s,*}^2\kappa_{daytype_0}(y_0 - g_0^{heat}(T_0^{heat} - u^{heat})\mathbb{1}_{(u^{heat}>T_0^{heat})})}{10^8\sigma_{s,*}^2\kappa_{daytype_0}^2 + \sigma^2\sigma_{s,*}^2 + 10^8\sigma^2}$

Therefore : $\boxed{s_0|s_{-0}, g_{0:n_0-1}^{heat}, \sigma_{s,*}, \sigma_{g,*}, y_{0:n_0-1} \sim \mathcal{N}(M', \frac{10^8\cdot\sigma^2\sigma_{s,*}^2}{10^8\sigma_{s,*}^2\kappa_{daytype_0}^2 + \sigma^2\sigma_{s,*}^2 + 10^8\sigma^2}, \mathbb{R}^+)}$

Full conditional of g_i^{heat} when $i \neq 0$ Very similarly as for s_i, since both components follow a similar dynamic, we can write that :

$$\pi(g_i^{heat}|g_{-i}^{heat}, s_{0:n_0-1}, \sigma_{s,*}, \sigma_{g,*}, y_{0:n_0-1}) \propto \Big(\prod_{k=0}^{n_0-1}\pi(y_k|s_k, g_k^{heat})\Big)\pi(g_0^{heat}|\sigma_{g,*})\pi(g_1^{heat}|g_0^{heat},\sigma_{g,*})...\pi(g_{n_0-1}^{heat}|g_{n_0-2},\sigma_{g,*})$$

We can now remove all components which do not depend on g_i^{heat} :

$$\propto \pi(y_i|s_i, g_i^{heat})\pi(g_i^{heat}|g_{i-1}^{heat},\sigma_{g,*})\pi(g_{i+1}^{heat}|g_i^{heat},\sigma_{g,*})$$

After a similar calculus, we recognize that :

$$\boxed{g_i^{heat}|g_{-i}^{heat}, s_{0:n_0-1}, \sigma_{s,*}, \sigma_{g,*}, y_{0:n_0-1} \sim \mathcal{N}(\tilde{M}, \frac{\sigma^2\sigma_{g,*}^2}{\sigma^2 + \sigma_{g,*}^2(T_i^{heat} - u^{heat})^2\mathbb{1}_{(u^{heat}>T_i^{heat})}}, \mathbb{R}^-)}$$

where $\tilde{M} = \frac{(g_{i-1}^{heat} + g_{i+1}^{heat})\sigma^2}{\sigma^2 + \sigma_{g,*}^2(T_i^{heat} - u^{heat})^2\mathbb{1}_{(u^{heat}>T_i^{heat})}} + \frac{\sigma_{g,*}^2(T_i^{heat} - u^{heat})\mathbb{1}_{(u^{heat}>T_i^{heat})}(y_i - s_i^{heat}\kappa_{daytype_i})}{\sigma^2 + \sigma_{g,*}^2(T_i^{heat} - u^{heat})^2\mathbb{1}_{(u^{heat}>T_i^{heat})}}$

Full conditional of g_0^{heat} Using the same arguments as to compute the full conditional of s_0, we get :

$$\boxed{g_0^{heat}|g_{-0}^{heat}, s_{0:n_0-1}, \sigma_{g,*}, \sigma_{g,*}, y_{0:n_0-1} \sim \mathcal{N}(\tilde{M}', \frac{10^8\cdot\sigma^2\sigma_{g,*}^2}{10^8\sigma_{g,*}^2(T_0^{heat} - u^{heat})^2\mathbb{1}_{(u^{heat}>T_0^{heat})} + \sigma^2\sigma_{s,*}^2 + 10^8\sigma^2}, \mathbb{R}^-)}$$

where : $\tilde{M}' = \frac{10^8\sigma^2 g_1^{heat} + 10^8\sigma_{g,*}^2(T_0^{heat} - u^{heat})\mathbb{1}_{(u^{heat}>T_0^{heat})}(y_0 - s_0\kappa_{daytype_0})}{10^8\sigma_{g,*}^2\kappa_{daytype_0}^2 + \sigma^2\sigma_{g,*}^2 + 10^8\sigma^2}$

Full conditional of $\sigma_{s,*}$ We will actually compute the full conditional on $\sigma_{s,*}^2$ instead of directly $\sigma_{s,*}$, since we have a prior on $\sigma_{s,*}^2$ and not on $\sigma_{s,*}$.

$$\begin{aligned}
\pi(\sigma_{s,*}^2|y_{0:n_0-1}, s_{0:n_0-1}, g_{0:n_0-1}^{heat}, \sigma_{g,*}) &\propto \pi(\sigma_{s,*}^2, y_{0:n_0-1}|s_{0:n_0-1}, g_{0:n_0-1}^{heat}, \sigma_{g,*}) \quad \text{(Bayes)} \\
&\propto \pi(y_{0:n_0-1}|s_{0:n_0-1}, g_{0:n_0-1}^{heat}, \sigma_{g,*}, \sigma_{s,*}^2)\pi(\sigma_{s,*}^2|s_{0:n_0-1}, g_{0:n_0-1}^{heat}, \sigma_{g,*}) \\
&\propto \Big(\prod_{k=0}^{n_0-1}\pi(y_k|s_k, g_k^{heat})\Big)\pi(\sigma_{s,*}^2, s_{0:n_0-1}|g_{0:n_0-1}^{heat}, \sigma_{g,*}) \quad \text{(Bayes)}
\end{aligned}$$

$\pi(y_k|s_k, g_k^{heat})$ does not depend on $\sigma_{s,*}^2$, hence :

$$\begin{aligned}
&\propto \pi(\sigma_{s,*}^2, s_{0:n_0-1}|g_{0:n_0-1}^{heat}, \sigma_{g,*}) \\
&\propto \pi(s_{0:n_0-1}|g_{0:n_0-1}^{heat}, \sigma_{g,*}, \sigma_{s,*}^2)\pi(\sigma_{s,*}^2|g_{0:n_0-1}^{heat}, \sigma_{g,*})
\end{aligned}$$

$s_{0:n_0-1}$ and $\sigma^2_{s,*}$ are independent from $g^{heat}_{0:n_0-1}$ and $\sigma_{g,*}$, hence :

$$\propto \pi(s_{0:n_0-1}|\sigma^2_{s,*})\pi(\sigma^2_{s,*})$$
$$\propto \pi(s_0|\sigma^2_{s,*})\pi(s_1|s_0,\sigma^2_{s,*})...\pi(s_{n_0-1}|s_{n_0-2},\sigma^2_{s,*})\pi(\sigma^2_{s,*})$$

s_0 is independent from $\sigma^2_{s,*}$, hence $\pi(s_0|\sigma^2_{s,*}) = \pi(s_0)$. We can thus remove it and we get :

$$\propto \Big(\prod_{k=1}^{n_0-1} \pi(s_k|s_{k-1},\sigma^2_{s,*})\Big)\pi(\sigma^2_{s,*})$$
$$\propto \Big(\prod_{k=1}^{n_0-1} \frac{1}{\sqrt{\sigma^2_{s,*}}}\exp(-\frac{1}{2\sigma^2_{s,*}}(s_k - s_{k-1})^2)\Big)\pi(\sigma^2_{s,*})$$

Since $\sigma^2_{s,*} \sim \mathcal{IG}(\alpha = 10^{-2}, \beta = 10^{-2})$, we have :

$$\propto \Big(\prod_{k=1}^{n_0-1} \frac{1}{\sqrt{\sigma^2_{s,*}}}\exp(-\frac{1}{2\sigma^2_{s,*}}(s_k - s_{k-1})^2)\Big)(\sigma^2_{s,*})^{-\alpha-1}\exp(-\frac{\beta}{\sigma^2_{s,*}})$$
$$\propto \Big(\frac{1}{\sigma^2_{s,*}}\Big)^{\frac{n_0-1}{2}}\exp(-\frac{1}{2\sigma^2_{s,*}}\sum_{k=1}^{n_0-1}(s_k - s_{k-1})^2)(\sigma^2_{s,*})^{-\alpha-1}\exp(-\frac{\beta}{\sigma^2_{s,*}})$$
$$\propto (\sigma^2_{s,*})^{-\alpha-\frac{n_0-1}{2}-1}\exp\Big(-\frac{1}{\sigma^2_{s,*}}[\beta + \frac{1}{2}\sum_{k=1}^{n_0-1}(s_k - s_{k-1})^2]\Big)$$

Therefore : $\boxed{\sigma^2_{s,*}|y_{0:n_0-1}, s_{0:n_0-1}, g^{heat}_{0:n_0-1}, \sigma_{g,*} \sim \mathcal{IG}(\alpha + \frac{n_0-1}{2}, \beta + \frac{1}{2}\sum_{k=1}^{n_0-1}(s_k - s_{k-1})^2) \quad \text{where } \alpha = \beta = 10^{-2}}$

Full conditional of $\sigma_{g,*}$ Similarly, we have :

$$\boxed{\sigma^2_{g,*}|y_{0:n_0-1}, s_{0:n_0-1}, g^{heat}_{0:n_0-1}, \sigma_{s,*} \sim \mathcal{IG}(\alpha + \frac{n_0-1}{2}, \beta + \frac{1}{2}\sum_{k=1}^{n_0-1}(g^{heat}_k - g^{heat}_{k-1})^2) \quad \text{where } \alpha = \beta = 10^{-2}}$$

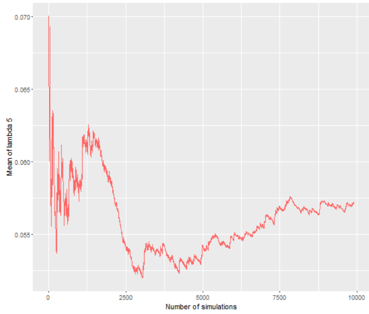
Python Code Unfortunately, I didn't have enough time to implement the algorithm on Python since we are still coding the particle filter and testing it without this Gibbs / MCMC step on simulated data using a fixed arbitrary θ. Once we finish this, we'll code it and compare the results of the particle filter on the simulated data with and without the additional Gibbs / MCMC step.

Appendix

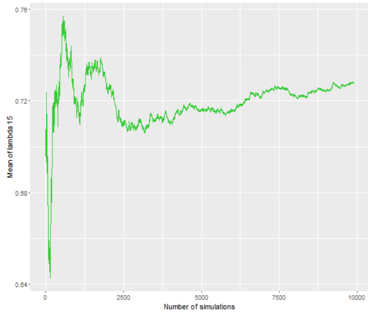
Appendix to Problem 10.10

Full R Code:

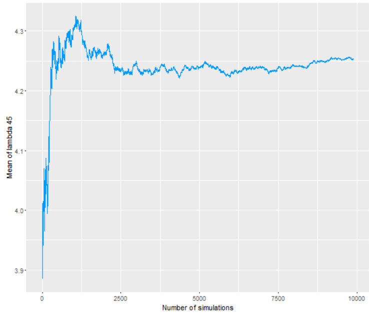
```
######################### Problem 10.10 #####################
###########################################################

### Question (b) : Gibbs-Sampler with alpha = 0.1 and a = b = 1

data_herds = c(rep(0,7), rep(1, 12), rep(2,8), rep(3,9), rep(4,8), rep(5,8), rep
    (6,9), rep(7,6),
                rep(8,5), rep(9,3), rep(10,4), rep(11,7), rep(12,4), rep(13,5), rep
                    (14,2), rep(15,1),
                rep(16,4), rep(17,3), rep(18,3), rep(19,4), rep(20,2), rep(21,2),
                    rep(22,4), rep(23,1),
                rep(25,6))

length(data_herds)

gibbs ← function(n_sim, alpha, a, b, x){
  #initialize
  result ← t(as.matrix(c(1, 1)))
```



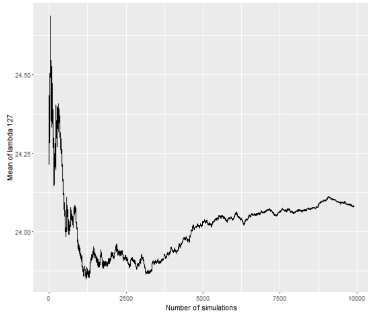
(a) λ_1 (Herd 5 had 0 occurrence of mastisis)



(b) λ_{15} (Herd 15 had 1 occurrence of mastisis)



(c) λ_{45} (Herd 45 had 5 occurrences of mastisis)



(d) λ_{127} (Herd 127 had 25 occurrences of mastisis)

Figure 7 – Evolution of the mean of λ_i for different herds i (First 99 simulations removed)

```
  for (t in 2:n_sim){
    given_beta = result[(t-1),2] + 1
    lambda_t ← rgamma(1, shape = alpha + x, rate = given_beta)
    beta_t ← rgamma(1, shape = alpha + a, lambda_t + b)
    vector_t ← c(lambda_t, beta_t)
    result ← rbind(result, vector_t)
  }
  return(result)
}

#test with x = data_herds[1]
#test ← gibbs(100, 0.1, 1, 1,  data_herds[1])

gibbs_full_data ← function(n_sim, alpha, a, b, data_herds){
  parameters = list()
  for (i in 1:127){
    x = data_herds[i]
    parameters[[i]] ← gibbs(n_sim, alpha, a, b, x)
  }
  return(parameters)
}

#Results
n_sim = 10000
alpha = 0.1
a = 1
b = 1

gibbs_questionb = gibbs_full_data(n_sim, alpha, a, b, data_herds)

result_1 ← matrix(unlist(gibbs_questionb[1]), ncol = 2, byrow = FALSE)

#Posterior of lambda 1
library(ggplot2)
qplot( result_1[,1], geom = 'histogram', bins = 100, fill = I("#FF6666"), xlab = 'Value of lambda 1')

#lambda 5
result_5 ← matrix(unlist(gibbs_questionb[5]), ncol = 2, byrow = FALSE)
```

```
qplot( result_5[,1], geom = 'histogram', bins = 100, fill = I("#FF6666"), xlab = 'Value of lambda 5')
#It is similar to lambda 1 since herd 5 has the same number of occurrences of clinical mastisis as herd 1...
#Compute the mean and monitor the convergence
est_vector_lambda5 = c()
for (k in 1:(n_sim)){
  est_mean ← mean(result_5[1:k,1])
  est_vector_lambda5 ← c(est_vector_lambda5, est_mean)
}
qplot(seq(1,n_sim-99,1),est_vector_lambda5[100:(n_sim)], geom = 'line', ylab = 'Mean of lambda 5', xlab = 'Number of simulations', colour = I("#FF6666"))


########### Herd 15 : 1 occurrence of mastisis in the herd ###################
#lambda 15
result_15 ← matrix(unlist(gibbs_questionb[15]), ncol = 2, byrow = FALSE)
qplot( result_15[,1], geom = 'histogram', bins = 100, fill = I("#33CC33"), xlab = 'Value of lambda 15')
#It is similar to lambda 1 since herd 5 has the same number of occurrences of clinical mastisis as herd 1...
#Compute the mean and monitor the convergence
est_vector_lambda15 = c()
for (k in 1:(n_sim)){
  est_mean ← mean(result_15[1:k,1])
  est_vector_lambda15 ← c(est_vector_lambda15, est_mean)
}
qplot(seq(1,n_sim-99,1),est_vector_lambda15[100:n_sim], geom = 'line', ylab = 'Mean of lambda 15', xlab = 'Number of simulations', colour = I("#33CC33"))

#beta 15
qplot( result_15[,2], geom = 'histogram', bins = 100, fill = I("#33CC33"), xlab = 'Value of beta 15')
#Compute the mean and monitor the convergence
est_vector_beta15 = c()
for (k in 1:(n_sim)){
  est_mean ← mean(result_15[1:k,2])
  est_vector_beta15 ← c(est_vector_beta15, est_mean)
}
qplot(seq(1,n_sim-4,1),est_vector_beta15[5:n_sim], geom = 'line', ylab = 'Mean of beta 15', xlab = 'Number of simulations', colour = I("#33CC33"))


########### Herd 45 : 5 occurrences of mastisis in the herd ###################
#lambda 45
result_45 ← matrix(unlist(gibbs_questionb[45]), ncol = 2, byrow = FALSE)
qplot( result_45[,1], geom = 'histogram', bins = 100, fill = I("#0099FF"), xlab = 'Value of lambda 45')
#Compute the mean and monitor the convergence
est_vector_lambda45 = c()
for (k in 1:n_sim){
  est_mean ← mean(result_45[1:k,1])
  est_vector_lambda45 ← c(est_vector_lambda45, est_mean)
}
qplot(seq(1,n_sim-99,1),est_vector_lambda45[100:10000], geom = 'line', ylab = 'Mean of lambda 45', xlab = 'Number of simulations', colour = I("#0099FF"))

#beta 45
qplot( result_45[,2], geom = 'histogram', bins = 100, fill = I("#0099FF"), xlab = 'Value of beta 45')
#Compute the mean and monitor the convergence
est_vector_beta45 = c()
for (k in 1:(n_sim)){
  est_mean ← mean(result_45[1:k,2])
  est_vector_beta45 ← c(est_vector_beta45, est_mean)
}
```

```
qplot(seq(1,n_sim-4,1),est_vector_beta45[5:n_sim], geom = 'line', ylab = 'Mean of
    beta 45', xlab = 'Number of simulations', colour = I("#0099FF"))


########### Herd 127 : 25 occurrences of mastisis in the herd ###################
#lambda 127
result_127 <- matrix(unlist(gibbs_questionb[127]), ncol = 2, byrow = FALSE)
qplot( result_127[,1], geom = 'histogram', bins = 100, xlab = 'Value of lambda 127
    ')
#Compute the mean and monitor the convergence
est_vector_lambda127 = c()
for (k in 1:n_sim){
  est_mean <- mean(result_127[1:k,1])
  est_vector_lambda127 <- c(est_vector_lambda127, est_mean)
}
qplot(seq(1,n_sim-99,1),est_vector_lambda127[100:n_sim], geom = 'line', ylab = '
    Mean of lambda 127', xlab = 'Number of simulations')

#beta 127
qplot( result_127[,2], geom = 'histogram', bins = 100, xlab = 'Value of beta 127')
#Compute the mean and monitor the convergence
est_vector_beta127 = c()
for (k in 1:(n_sim)){
  est_mean <- mean(result_127[1:k,2])
  est_vector_beta127 <- c(est_vector_beta127, est_mean)
}
qplot(seq(1,n_sim,1),est_vector_beta127, geom = 'line', ylab = 'Mean of beta 127',
     xlab = 'Number of simulations')


### Question (c) : Impact of the hyperparameters a, b and alpha
#We will check first Herd 15 and Herd 45 (instead of herd 5 and herd 15, which are
     too similar)

#################### Impact of alpha ############
n_sim = 2500
a = 1
b = 1

#On herd 15 (1 occurrenc of mastisis)
alpha_to_test = c(0.1,1,10)
results_alpha = list()
for (k in 1:length(alpha_to_test)){
  results_alpha[[k]] <- gibbs(n_sim, alpha_to_test[k], a, b, data_herds[15])
}

#lambda 15
mdf_hist <- data.frame(nb_sim = seq(1,n_sim,1))
mdf_mean <- data.frame(nb_sim = seq(1,n_sim,1))
for (k in 1:length(alpha_to_test)){
  result_15 <- matrix(unlist(results_alpha[k]), ncol = 2, byrow = FALSE)
  mdf_hist <- cbind(mdf_hist, result_15[,1])
  est_vector_lambda15 = c()
  for (j in 1:(n_sim)){
    est_mean <- mean(result_15[1:j,1])
    est_vector_lambda15 <- c(est_vector_lambda15, est_mean)
  }
  mdf_mean <- cbind(mdf_mean, est_vector_lambda15)
}

#Compare the mean
library("reshape2")
colnames(mdf_mean) <- c("nb_sim", "alpha = 0.1", "alpha = 1", "alpha = 10")
mdf2_mean <- melt(mdf_mean, id="nb_sim")
ggplot(data=mdf2_mean,
       aes(x=nb_sim, y=value, colour=variable)) +
  geom_line() +ylab("Mean of lambda 15") +xlab("Simulations")
```

```r

#Compare the histograms of the posterior
colnames(mdf_hist) <- c("nb_sim", "alpha1", "alpha2", "alpha3")
mdf2_hist <- melt(mdf_hist, id="nb_sim")
ggplot(data=mdf2_hist,
       aes(x=value, fill=variable)) +
  geom_histogram(alpha = 0.25)

#beta 15
mdf_hist <- data.frame(nb_sim = seq(1,n_sim,1))
mdf_mean <- data.frame(nb_sim = seq(1,n_sim,1))
for (k in 1:length(alpha_to_test)){
  result_15 <- matrix(unlist(results_alpha[k]), ncol = 2, byrow = FALSE)
  mdf_hist <- cbind(mdf_hist, result_15[,2])
  est_vector_beta15 = c()
  for (j in 1:(n_sim)){
    est_mean <- mean(result_15[1:j,2])
    est_vector_beta15 <- c(est_vector_beta15, est_mean)
  }
  mdf_mean <- cbind(mdf_mean, est_vector_beta15)
}

#Compare the mean
library("reshape2")
colnames(mdf_mean) <- c("nb_sim", "alpha = 0.1", "alpha = 1", "alpha = 10")
mdf2_mean <- melt(mdf_mean, id="nb_sim")
ggplot(data=mdf2_mean,
       aes(x=nb_sim, y=value, colour=variable)) +
  geom_line() +ylab("Mean of beta 15") +xlab("Simulations")

#Compare the histograms of the posterior
colnames(mdf_hist) <- c("nb_sim", "alpha1", "alpha2", "alpha3")
mdf2_hist <- melt(mdf_hist, id="nb_sim")
ggplot(data=mdf2_hist,
       aes(x=value, fill=variable)) +
  geom_histogram(alpha = 0.25)


################### Impact of a ############
n_sim = 2500
alpha = 0.1
b = 1

#On herd 15 (1 occurrenc of mastisis)
a_to_test = c(0.1,1,10)
results_a = list()
for (k in 1:length(a_to_test)){
  results_a[[k]] <- gibbs(n_sim, alpha, a_to_test[k], b, data_herds[15])
}

#lambda 15
mdf_hist <- data.frame(nb_sim = seq(1,n_sim,1))
mdf_mean <- data.frame(nb_sim = seq(1,n_sim,1))
for (k in 1:length(a_to_test)){
  result_15 <- matrix(unlist(results_a[k]), ncol = 2, byrow = FALSE)
  mdf_hist <- cbind(mdf_hist, result_15[,1])
  est_vector_lambda15 = c()
  for (j in 1:(n_sim)){
    est_mean <- mean(result_15[1:j,1])
    est_vector_lambda15 <- c(est_vector_lambda15, est_mean)
  }
  mdf_mean <- cbind(mdf_mean, est_vector_lambda15)
}

#Compare the mean
library("reshape2")
colnames(mdf_mean) <- c("nb_sim", "a = 0.1", "a = 1", "a = 10")
```

```
mdf2_mean ← melt(mdf_mean, id="nb_sim")
ggplot(data=mdf2_mean,
       aes(x=nb_sim, y=value, colour=variable)) +
  geom_line() +ylab("Mean of lambda 15") +xlab("Simulations")

#Compare the histograms of the posterior
colnames(mdf_hist) ← c("nb_sim", "a= 0.1", "a = 1", "a = 10")
mdf2_hist ← melt(mdf_hist, id="nb_sim")
ggplot(data=mdf2_hist,
       aes(x=value, fill=variable)) +
  geom_histogram(alpha = 0.25)

#beta 15
mdf_hist ← data.frame(nb_sim = seq(1,n_sim,1))
mdf_mean ← data.frame(nb_sim = seq(1,n_sim,1))
for (k in 1:length(a_to_test)){
  result_15 ← matrix(unlist(results_a[k]), ncol = 2, byrow = FALSE)
  mdf_hist ← cbind(mdf_hist, result_15[,2])
  est_vector_beta15 = c()
  for (j in 1:(n_sim)){
    est_mean ← mean(result_15[1:j,2])
    est_vector_beta15 ← c(est_vector_beta15, est_mean)
  }
  mdf_mean ← cbind(mdf_mean, est_vector_beta15)
}

#Compare the mean
library("reshape2")
colnames(mdf_mean) ← c("nb_sim", "a = 0.1", "a = 1", "a = 10")
mdf2_mean ← melt(mdf_mean, id="nb_sim")
ggplot(data=mdf2_mean,
       aes(x=nb_sim, y=value, colour=variable)) +
  geom_line() +ylab("Mean of beta 15") +xlab("Simulations")

#Compare the histograms of the posterior
colnames(mdf_hist) ← c("nb_sim", "a = 0.1", "a = 1", "a = 10")
mdf2_hist ← melt(mdf_hist, id="nb_sim")
ggplot(data=mdf2_hist,
       aes(x=value, fill=variable)) +
  geom_histogram(alpha = 0.25)


#################### Impact of b ############
n_sim = 2500
alpha = 0.1
a = 1

#On herd 15 (1 occurrenc of mastisis)
b_to_test = c(0.1,1,10)
results_b = list()
for (k in 1:length(b_to_test)){
  results_b[[k]] ← gibbs(n_sim, alpha, a, b_to_test[k], data_herds[15])
}

#lambda 15
mdf_hist ← data.frame(nb_sim = seq(1,n_sim,1))
mdf_mean ← data.frame(nb_sim = seq(1,n_sim,1))
for (k in 1:length(b_to_test)){
  result_15 ← matrix(unlist(results_b[k]), ncol = 2, byrow = FALSE)
  mdf_hist ← cbind(mdf_hist, result_15[,1])
  est_vector_lambda15 = c()
  for (j in 1:(n_sim)){
    est_mean ← mean(result_15[1:j,1])
    est_vector_lambda15 ← c(est_vector_lambda15, est_mean)
  }
  mdf_mean ← cbind(mdf_mean, est_vector_lambda15)
}
```

```
#Compare the mean
library("reshape2")
colnames(mdf_mean) ← c("nb_sim", "b = 0.1", "b = 1", "b = 10")
mdf2_mean ← melt(mdf_mean, id="nb_sim")
ggplot(data=mdf2_mean,
       aes(x=nb_sim, y=value, colour=variable)) +
  geom_line() +ylab("Mean of lambda 15") +xlab("Simulations")

#Compare the histograms of the posterior
colnames(mdf_hist) ← c("nb_sim", "b= 0.1", "b = 1", "b = 10")
mdf2_hist ← melt(mdf_hist, id="nb_sim")
ggplot(data=mdf2_hist,
       aes(x=value, fill=variable)) +
  geom_histogram(alpha = 0.25)

#beta 15
mdf_hist ← data.frame(nb_sim = seq(1,n_sim,1))
mdf_mean ← data.frame(nb_sim = seq(1,n_sim,1))
for (k in 1:length(b_to_test)){
  result_15 ← matrix(unlist(results_b[k]), ncol = 2, byrow = FALSE)
  mdf_hist ← cbind(mdf_hist, result_15[,2])
  est_vector_beta15 = c()
  for (j in 1:(n_sim)){
    est_mean ← mean(result_15[1:j,2])
    est_vector_beta15 ← c(est_vector_beta15, est_mean)
  }
  mdf_mean ← cbind(mdf_mean, est_vector_beta15)
}

#Compare the mean
library("reshape2")
colnames(mdf_mean) ← c("nb_sim", "b = 0.1", "b = 1", "b = 10")
mdf2_mean ← melt(mdf_mean, id="nb_sim")
ggplot(data=mdf2_mean,
       aes(x=nb_sim, y=value, colour=variable)) +
  geom_line() +ylab("Mean of beta 15") +xlab("Simulations")

#Compare the histograms of the posterior
colnames(mdf_hist) ← c("nb_sim", "b = 0.1", "b = 1", "b = 10")
mdf2_hist ← melt(mdf_hist, id="nb_sim")
ggplot(data=mdf2_hist,
       aes(x=value, fill=variable)) +
  geom_histogram(alpha = 0.25,bins = 40)
```