Séminaire de modélisation statistique

Clara Champagne

Statistiques et Maladies infectieuses

Saeyeon Kwon, Théo Lorthios

Table des matières

1 Exercice 1 : Article de recherche sur les dynamiques de transmission du virus Zika dans les îles polynésiennes (2013/2014)

1.1 Présentation de l'article

Entre 2013 et 2014, plus de 30 000 cas de personnes infectées au virus du Zika (ZIKV) ont été reportés dans les îles polynésiennes françaises, amenant, entre autres, l'Organisation Mondiale de la Santé à déclarer un état d'urgence sanitaire international. Le virus Zika provient des moustiques de l'espèce *Aedes aegypti* qui sont les principaux vecteurs de l'infection à ce virus. L'article étudie les dynamiques de transmission du virus de Zika, dont l'importance se fait cruciale compte tenu de son niveau de propagation considérable. L'analyse porte sur l'épidémie de l'infection au Zika entre 2013-2014 au sein des six plus grands archipels de la Polynésie française : Tahiti, Iles sous-le-vent, Moorea, Tuamotu-Gambier, Marquises, Australes. Une des spécificités de l'article est que l'étude porte sur des populations insulaires. L'avantage de ce type d'étude est que l'isolement et la centralisation de la population font que la dynamique de l'épidémie reste plutôt endémique.

Les principaux résultats démontrés par cet article sont les suivants. D'abord, le virus du Zika a infecté la majorité de la population mais seulement 7 à 17% des cas ont été reportés officiellement. Ensuite, si une infection à ce virus nous procure une immunité à vie, il a été estimé qu'environ 12 à 20 années seraient nécessaires pour que le nombre de personnes susceptibles d'être infectées soit tel que le virus provoque une nouvelle épidémie.

Le modèle statistique utilisé, présenté à la section suivante, a pour objectif d'inférer, entre autres, le taux de reproduction de l'épidémie, son ampleur et sa dynamique, ainsi que d'estimer le nombre d'individus encore susceptibles d'être infectés dans les années à venir à l'issue des années 2013-2014, afin de considérer le nombre d'années qui seraient nécsesaires avant que l'épidémie réémerge au sein de ces populations.

1.2 Le modèle

Le modèle utilisé est un modèle mathématique compartimental, c'est-à-dire un modèle qui divise la population d'étude en plusieurs classes épidémiologiques par rapport à la maladie. Dans cet article, la population est divisée en quatre catégories : les personnes susceptibles d'être infectées et qui ne l'ont jamais été auparavant (S^H), les personnes exposées i.e. les personnes qui ont contracté le virus et sont dans la période d'incubation (E^H), les personnes infectieuses (I^H) et les personnes guéries (R^H). Le modèle est ainsi appelé modèle SEIR. Dans ce modèle, les compartiments sont également appliqués aux moustiques car on fait l'hypothèse que la transmission peut se faire des humains vers les moustiques et vice-versa. En effet, une telle hypothèse ne semble pas déraisonnable dans la mesure où il a été démontré, pour le virus de la dengue, que les humains pourvaient transmettre le virus aux moustiques. Les compartiments pour les moustiques sont les suivants : les moustiques susceptibles d'être infectés(S^V), les moustiques dans leur période d'incubation (E^V), les moustiques infectieuses (I^V). Contrairement aux humains, on remarque qu'il n'y a pas de catégories pour les moustiques guéris car ces derniers, une

fois devenus infectieux, le sont à vie. Au delà de ces compartiments, le modèle compartimental se caractérise aussi par des règles qui spécifient la proportion des individus (resp. des moustiques) passant d'un compartiment à un autre au cours du temps.

Pour cela, le modèle présente d'autres paramètres épidémiologiques qui interviennent dans la transmission du virus. Parmi eux, β_H représente le taux de transmission du Zika des moustiques vers les humains et β_V celui des humains vers les moustiques. La période d'incubation pour les humains (resp. les moustiques) est $\frac{1}{\alpha_H}$ (resp. $\frac{1}{\alpha_V}$). La période d'infection moyenne pour les humains est $\frac{1}{\gamma}$.

Les principaux paramètres du modèle étant présentés, nous pouvons à présent expliciter commment ces derniers interviennent dans les équations du modèle qui vise à stimuler la dynamique de la transmission vectorielle du Zika entre les humains, mais également entre les humains et les moustiques. Pour étudier cette dynamique, il convient de s'intéresser à l'évolution au cours du temps de la proportion d'individus et de moustiques dans chaque compartiment. Ainsi, l'article met en évidence les dérivées partielles d'ordre 1 du nombre d'individus et de moustiques dans chaque compartiment par rapport au temps.

D'abord pour les humains :

$$\frac{\partial S^H}{\partial t} = -\beta_H \times S^H \times I^V \tag{1}$$

$$\frac{\partial E^H}{\partial t} = \beta_H \times S^H \times I^V - \alpha_H \times E^H \tag{2}$$

$$\frac{\partial I^H}{\partial t} = \alpha_H \times E^H - \gamma \times I^H \tag{3}$$

$$\frac{\partial R^H}{\partial t} = \gamma \times I^H \tag{4}$$

$$\frac{\partial C}{\partial t} = \alpha_H \times E^H \tag{5}$$

L'équation (1) montre que l'évolution du nombre de susceptibles entre t et t+1 dépend du nombre de personnes susceptibles infectées par un moustique porteur du virus. Pour cela, il faut qu'un susceptible et un moustique infectieux se rencontre d'où le produit $S^H \times I^V$ auquel on mulitplie le taux de tranmission des moustiques vers les humains. Plus ce taux est élevé, plus la tranmission se fait rapidement. Le signe (-) traduit bien le fait que plus le nombre de personnes susceptibles était initialement élevé / plus le nombre de moustiques infectieux est élevé / plus le taux de tranmission du virus des moustiques vers les humains est élevé, plus il y a de chances que des personnes soient infectées au fil du temps, donc que la proportion de susceptibles diminue.

L'équation (2) montre que l'évolution entre t et t+1 du nombre de personnes en période d'incubation dépend postivement du nombre de susceptibles ayant contracté le virus, c'est-à-dire du nombre de personnes qui sont passées du compartiment "susceptibles" à "infectés et en période d'incubation", d'où le premier terme. A cela, il faut retrancher la proportion de personnes dans E^H qui sortent de leur période d'incubation et qui deviennent infectieuses, d'où le second terme.

L'équation (3) montre que l'évolution entre t et t+1 du nombre de personnes infectieuses dépend positivement du nombre de personnes qui étaient en période d'incubation et qui deviennent infectieuses, ce que traduit le premier terme. A cela, il faut retrancher le proportion d'infectieux qui guérissent de la maladie, d'où le second terme.

L'équation (4) montre que le nombre de personnes guéries entre t et t+1 dépend postivement de la proportion des infectieux qui guérissent, c'est-à-dire de la quantité $\gamma \times I^H$.

L'équation (5) montre que le nombre cumulé de personnes infectées entre t et t+1 dépend de la part des personnes en période d'incubation qui deviennent infectées.

Concenrnant les moustiques, les équations qui spécifient la proportion des moustiques passant d'un compartiment à un autre suit une logique similaire à celle des humains, à ceci près qu'un facteur supplémentaire est pris en compte : la durée de vie moyenne des moustiques. En effet, contrairement aux humains dont la durée de vie est beaucoup plus longue que celle de l'épidémie, il est probable que certains moustiques meurent durant la période de l'épidémie et que cela fasse diminuer leur nombre. Les équations (6), (7) et (8) de l'article décrivent cette dynamique.

L'équation (6) montre que le nombre de moutiques susceptibles entre t et t+1 peut varier par trois canaux : le taux de moustiques qui arrivent dans la catégorie des "susceptibles" d'où le premier terme, auquel il faut retrancher les moustiques qui passent du compartiment "susceptibles" à "infectées en périodes d'incubation" c'est-à-dire : la part de moustiques qui deviennent infectées dus à un contact avec un humain infecté. Enfin, le trosième terme représente la part des moustiques susceptibles qui meurent naturellement, sans avoir été infectés, entre t et t+1.

L'équation (7) traduit le fait que la variaton du nombre de moustiques en incubation entre t et t+1 doit prendre en compte la part des moustiques qui passent du compartiment "susceptibes" à "en période d'incubation" d'où le premier terme. A cela, il faut retrancher la part des moustiques qui meurent naturellement ou qui deviennent infectieuses entre t et t+1, et qui sortent donc du compartiment "moustiques en incubation", d'où le facteur $(\delta + \alpha_V)$.

L'équation (8) montre que la variation du nombre de moustiques infectieux entre t et t+1 dépend de la part des moustiques qui sont passés du compartiment "en incubation" à celui "d'infectieux", d'où le premier terme. A cela, on retranche la part des moustiques infectieux qui meurent naturellement.

Le modèle de l'article explicite également le taux de reproduction du virus, le R_0, qui prend en compte le nombre moyen d'humains infectés par les moustiques ET le nombre moyen de moustiques infectés par les humains.

L'algorithme utilisé pour ajuster un tel modèle et faire de l'inférence statistique est l'algorithme de Markov Chain Monte Carlo (MCMC). On souhaite inférer, pour chacune des six îles, le taux de reproduction R_0, la proportion des personnes infectées qui ont été reportées comme telles, la part des personnes infectées dans la population totale, le nombre d'individus initialement infectés ; on souhaite également estimer la période d'incubation et d'infection pour les humains et les mousitiques. On se place dans un cadre bayésien où on cherche à estimer la distribution a posteriori de ces paramètres qui est proportionnel à la vraisemblance multipliée par le "prior". L'algorithme se base sur une méthode simulée, de manière itérative, dans le but d'estimer a posteriori nos paramètres d'intérêt. Plus de détails concernant cet algorithme sont donnés dans la partie 2 de ce projet.

Une chaine de Markov est un processus stochastique particulier avec une mémoire d'une période. Ainsi, l'information concernant la semaine t (par exemple sur le taux d'incidence à la semaine t) est uniquement contenue dans l'état de la période précédente (c'est-à-dire la semaine t-1). Pour le calcul de la vraisemblance quant à l'observeration d'un nombre y_t de cas confirmés et suspectés, nous faisons l'hypothèse dans ce modèle que le nombre de cas suit une loi binomiale négative. Il est intéressant de souligner que le modèle

prend en compte la potentielle variabilité des nombres de cas reportés dans le temps, par exemple des cas qui seraient non-reportés dus à un nombre limité de sites sentinels ou des cas asymptomatiques, et introduit alors des pararmètres d'ajustement. Les itérations de l'algorithme MCMC ont permis d'estimer la distribution jointe a posteriori de nos variables d'intérêt. Environ 2000 trajectoires de nombre de cas confirmés et suspectés ont été simulées.

L'article propose également un modèle pour rendre compte de la dynamique démographique de la population. Le nombre total d'habitants ainsi que le nombre de susceptibles sont décrits selon un processus de Markov, en tenant compte du taux de naissance et de mort, ainsi que du taux de migration net. A partir de la part de la population qui est restée dans le compartiment des "susceptibles" à la fin de l'épidémie en 2014, et en appliquant le modèle itérativement sur les années suivantes, le modèle prédit la proportion des susceptibles dans les années à venir, ainsi que les taux de reproduction de ces années appelés "taux de reproduction effectif".

1.3 Quelques résultats

A partir des estimations a posteriori des paramètres d'intérêt, les chercheurs aboutissent à la conclusion que seule une petite proportion des cas de Zika a été reportée officiellement comme des cas suspectés dans les six îles de la Polynésie française : au total, seulement 7 à 17% des infections ont été reportées. Par ailleurs, le modèle estime la proportion d'individus infectés durant cette épdidémie à un taux compris entre 87 et 97% de la population totale. De ce fait, une nouvelle percée de l'épidémie de Zika sur ces îles n'adviendrait pas avant plusieurs années. En effet, les estimations a posteriori de la proportion de la population qui serait restée dans le compartiment des "susceptibles" après l'épidémie de 2013-2014 et du taux de reproduction effectif suggèrent qu'environ 12 à 20 ans seraient nécessaires pour que le nombre de susceptibles soit tel qu'une nouvelle épidémie ravage à nouveau les îles de la Polynésie française. Il s'agit là d'une caractéristique de la dynamique épidémique qui se retrouvait déjà pour l'épidémie de la dengue dans les îles pacifiques. La durée pour une réapparition de l'épidémie au sein de ces populations insulaires est similaire pour le cas de Zika et celui de la dengue, ce qui pousse les chercheurs de cet article à conclure que les dynamiques pour ces deux épidémies sont similaires : des épidémies explosives, infectant une grande partie de la population, mais relativement assez peu fréquentes.

Par ailleurs, les résultats montrent que les estimations a posteriori de la durée d'incubation et d'infection pour les humains et les moustiques sont consistantes avec les priors qui ont été considérés. Cela suggère que les distributions de probabilité pour ces paramètres n'étaient pas différentes de celles des priors, ou, que les données disponibles n'ont pas permis au modèle d'identifier correctement ces paramètres.

Enfin, l'article souligne le fait que l'épidémie de Zika dans les îles de la Polynésie française est allée de pair avec avec une montée de cas du syndrome Guillain-Barré et des cas de complications neurologiques. Il pourrait alors être interéssant d'étudier les éventuels effets causaux qui pourraient exister entre l'infection au virus de Zika et ces complications neurologiques.

1.4 Limites et critiques

Une des limites d'un modèle compartimental présenté dans cet article est qu'elle peut être trop "simpliste". En effet, on considère les individus comme étant parfaitement homogènes au sein d'un même compartiment alors qu'on pourrait émettre l'hypothèse, défendable, que les individus, par exemple au sein du compartiment des susceptibles,

présentent des caractéristiques individuelles qui font que certains sont plus à même de contracter le virus. Par exemple, une personne immunodéprimée a certainement plus de chances d'être infectée qu'une personne qui ne l'est pas. Par ailleurs, l'hétérogénéité spatiale peut aussi faire que des personnes vivant dans certains lieux ont plus de chances d'être en contact et donc d'attraper le virus de Zika.

Les paramètres liés aux taux de transmission étant considérés comme exogènes dans ce modèle, ce dernier est, dans un sens, déterministe, ce qui pourrait être critiquable. Par ailleurs, le mdodèle repose sur des hypothèses faites sur les distributions a priori des variables que sont la période d'incubation et d'infection des humains et des moustiques.

Enfin, le modèle présenté dans l'article semble faire abstraction des interventions externes de politiques publiques mises en place pour lutter contre la propagation de l'épidémie du Zika. Pourtant, il peut s'agir d'un facteur qui affecte le taux de transmission du virus au sein de la population.

2 Exercice 2 : Modèle SIR sous R

2.1 Question 1 : Comparaison de certaines combinaisons de paramètres

Dans un premier temps, il est important de comprendre intuitivement ce que signifie les paramètres β : le taux de transmission du virus et γ : le taux de guérison du virus. Plus le taux de transimission du virus est élevé, plus le nombre d'infectés devrait augmenter dans le temps, à contrario, plus le taux de guérison est grand et proche de 1, plus le nombre d'infections aura tendance à baisser.

Les dérivés partielles croisées :

$$\frac{\partial^2 I}{\partial t \partial \beta} = \frac{\partial dI}{\partial \beta} = S * \frac{I}{N} \geq 0 \tag{6}$$

$$\frac{\partial^2 I}{\partial t \partial \gamma} = \frac{\partial dI}{\partial \gamma} = -I \leq 0 \tag{7}$$

Par intuition, un virus hyper contagieux aura un β très fort mais jamais un $\beta \leq 0$, on peut définir $\beta \in [0, +\infty[$. Pour ce qui est de γ, on mettra comme hypothèse que l'administration d'un traitement soigne uniquement une personne et n'est pas un effet contagieux. Ce genre de cas est très rare dans le monde de l'épidémiologie donc $\gamma \in [0, 1]$.

Pour comparer les deux paramètres, nous avons construit une fonction qui prend en entrée une combinaison de β, γ et resort un graphique comparant les données simulées par le modèle SIR et les données réellement observées.

Une autre analyse se fait sur la construction du coefficient suivant :

$$R_0 = \frac{\beta}{\gamma} \tag{8}$$

On admet ainsi 3 cas possibles :

— (i) $R_0 = 1 <=> \beta = \gamma$. Le nombre d'infectés est constant dans le temps, pour chaque individu guéri, un autre tombe malade. L'épidémie reste au stade initial.

— (ii)$R_0 < 1 <=> \beta < \gamma$. L'épidémie tend à disparaitre si cette métrique est stable dans le temps, en effet, le nombre d'infectés baisse car il y a moins de transmission du virus que de personnes soignées. Dans notre cas, l'épidémie ne décolle pas car les nouveaux malades sont immédiatement guéris.

— (iii)$R_0 > 1 <=> \gamma < \beta$. L'épidémie est en phase d'explosion, le niveau d'infection tend vers N : la population totale. Le virus se diffuse plus vite que le traitement. Dans ce cas, l'épidémie touche assez rapidement, selon l'écart entre β et γ, l'ensemble de la population et comme par hypothèse on ne pas retomber malade, on a que le nombre d'infectés décroit quadratiquement après le pic.

Quelques graphiques pour illustrer ces trois états.

Figure 1 : $R_0 < 1$***. Une épidémie qui ne décolle pas :***

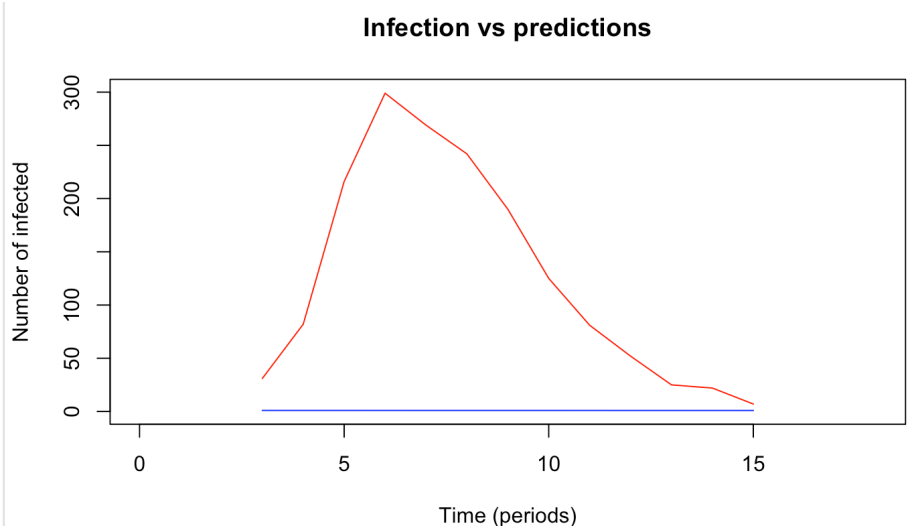
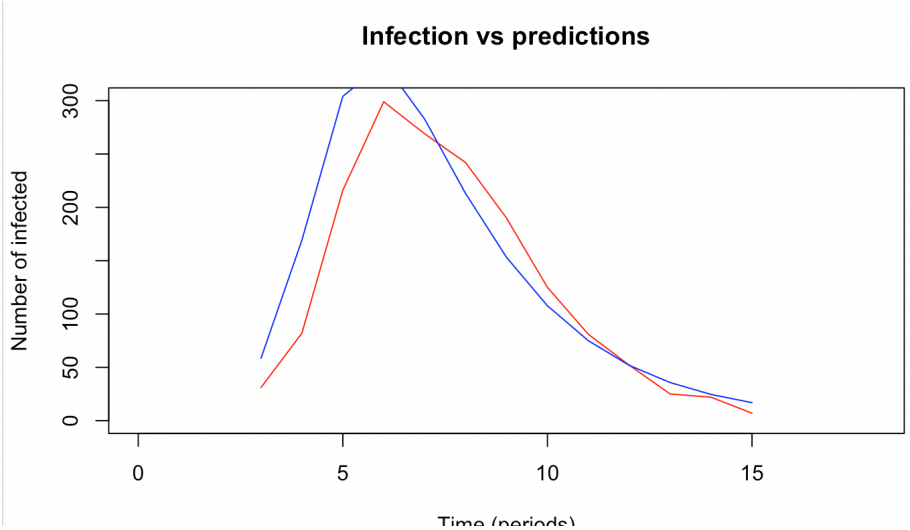
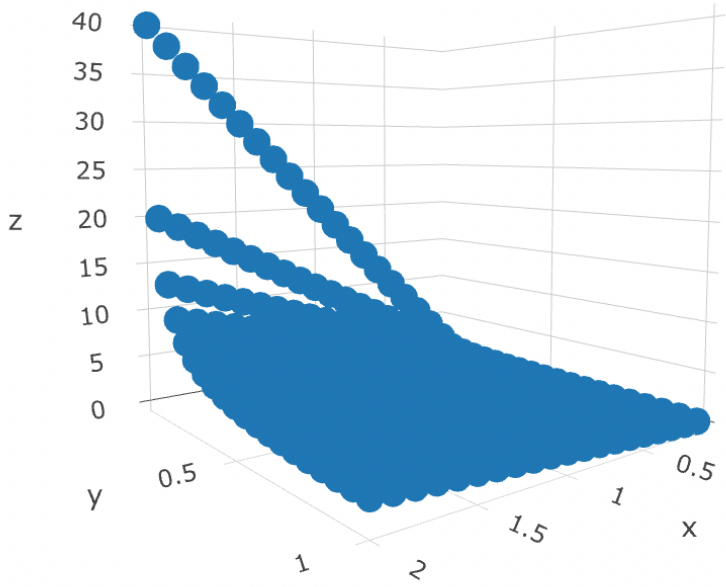


Figure 2 : $R_0 > 1$***. Une épidémie qui explose :***



On peut représenter ce $R_0(\beta, \gamma)$ sur un graphique en 3 dimension :

Figure 3 : R_0 ***selon des combinaisons de*** $(x = \beta, y = \gamma)$:



2.2 Question 3 : Approche bayésienne des paramètres

2.2.1 Vraisemblance et distribution à priori

L'énoncé nous apporte des précisions sur la paramétrisation de l'appoche bayésienne.

L'objectif de cette approche est d'estimer les paramètres du modèle :

— β : le taux de transmission du virus
— γ : le taux de guérison du virus

afin de reproduire la dynamique du virus avec des outils statistiques que nous connaissons bien. Cette approche permet d'effectuer des simulations et de tester une pluralité de scenarii différents de politiques et ainsi, d'en mesurer quantitativement les effets avec un risque d'erreur.

La métrique que nous utiliserons pour confronter les données observées aux données simulées du modèle est la fonction de vraisemblance du modèle. La formule générique est :

$$L_M(X_{ob_i}, \theta) = \prod_{i=1}^{n} P_\theta(X_{ob_i} = x_{ob_i}) \tag{9}$$

Notons :
— $\theta = (\beta, \gamma, \lambda)$: notre jeu de paramètres à estimer, avec λ le paramètre de la loi suivi par les données.

— X_{ob_i} est la $i^{ème}$ variable aléatoire réelle observée et x_{ob_i} est la $i^{ème}$ donnée observée

— P_θ est une famille de loi de probabilité de paramètre θ. Ici, l'énoncé nous dit que la loi suivie par les données observées est une loi de Poisson : $X_{ob_i} \sim Poisson(\lambda)$. On sait que λ dépend des X_{sim_i} associé à un paramètre ρ.

L'idée de l'approche bayésienne est de trouver la combinaison θ telle que la vraisemblance du modèle soit maximisée. Nous ne connaissons pas la loi réelle suivi par les données mais seulement les distributions a priori des paramètres $\beta \sim Unif[0, 10]$ et $\gamma \sim Unif[0, 1]$ et la forme de la loi de probabilité.

Nous allons donc procéder en trois étapes :

— (i) On approche la distribution à posteriori : $f = P(\theta_{ob_i})$ à l'aide des distributions a priori : $\pi(\theta)$ et de la vraisemblance corrigée d'une constante de normalisation : $c = \int f(\theta \mid X_{ob_i})\pi(\theta)d\theta$.

— (ii) On calcule la vraisemblance avec cette distribution a posteriori.

— (iii) On cherche le θ^* optimal tel que la vraisemblance soit maximisée.

2.2.2 La vraisemblance

Nous pouvons écrire la formule de la vraisemblance dans notre cas ; nous savons que nos données suivent une loi de poisson par hypothèse. Ainsi $P(X_{ob_i} = x_{ob_i}) = f(x_{ob_i}) = \frac{\lambda^{x_{ob_i}}}{x_{ob_i}!}e^{-\lambda}$ qui est la densité d'une loi de poisson. De plus, nous savons de l'énoncé que $\lambda = X_{sim_i} = I_{sim_i}$: le nombre d'infectés au temps i simulés par le modèle.

Notons : x_{ob_i} le nombre d'infectés observés à la date i et x_{sim_i} le nombre d'infecté simulés par le modèle à la date i.

On peut réécrire la vraisemblance du modèle comme suit :

$$L_n(\theta, X_{ob}) = \prod_{i=1}^{n} \frac{x_{sim_i}^{x_{ob_i}}}{x_{ob_i}!} e^{-x_{sim_i}} \tag{10}$$

2.2.3 Distribution a posteriori

À l'aide de cette vraisemblance, nous pouvons calculer la distribution a posteriori à une constante près (c).

$$f_\theta = \frac{\pi(\theta) * L_n(\theta, X_{ob})}{c} \tag{11}$$

Notons $\theta = (\beta, \gamma)$, avec $\beta \sim Unif[0, 10]$ avec une densité de probabilité $f_\beta = \frac{1}{10-0} * \delta(\beta \in [0, 10])$ et $\gamma \sim Unif[0, 1]$ avec $f_\gamma = \frac{1}{1-0} * \delta(\gamma \in [0, 1])$ où $\delta(.)$ est la distribution de dirak.

Par hypothèse, on considère que la loi a priori de β est indépendante de celle de γ ainsi,

nous pouvons écrire la distribution a priori comme le produit des deux lois a priori des paramètres :

$$\pi(\theta) = f_\beta * f_\gamma = \delta(\beta \in [0, 10]) * \delta(\gamma \in [0, 1]) * 0.1 \tag{12}$$

En y ajoutant la vraisemblance du modèle calculée plus haut, on détermine la distribution a posteriori telle que :

$$f_\theta = \frac{\delta(\beta \in [0, 10]) * \delta(\gamma \in [0, 1]) * 0.1 * \prod_{i=1}^{n} \frac{x_{sim_i}^{x_{ob_i}}}{x_{ob_i}!} e^{-x_{sim_i}}}{c} \tag{13}$$

2.3 Sous R : coder la vraisemblance

Pour coder la valeur de la vraisemblance du modèle, nous avons simplement repris la fonction de la question 1 qui rentre dans un dataframe le nombre d'infectés par jour observés et simulés et nous avons appliqué l'équation (10).

Pour le jeu de paramètre donné $\theta = (1.77, 0.44)$, on obtient une vraisemblance égale à $L_n(\theta) = 2.636124e{-}24$. On utilisera la densité d'une poisson avec la commande dpois(k,λ).

2.4 Sous R : coder la distribution à posteriori

Ici, nous avons repris la formule (13) en utilisant le code précédent pour le calcul de vraisemblance. Pour être exacts, nous calculons ici la distribution à posteriori à une constante près qui est la constante de normalisation. La valeur exacte nous servira pas car maximiser sur une constante ne change rien dans l'optimisation.

De plus, nous utiliserons les fonctions dunif(.) pour les densités des paramètres a priori.

Nous obtenons ainsi comme valeur pour le modèle fixé dans l'énoncé une distribution a posteriori égale à $f_{(1.7,0.44)} = 2.636124e^{-26}$

2.5 Question 3 : Algorithme MCMC : Metropolis-Hastings

L'algorithme de Metropolis-Hastings est un algorithme appliqué dans le cas d'estimation par MCMC (Markov Chain et Monte Carlo). L'idée derrière le MCMC est de retirer l'hypothèse que les variables simulées sont indépendantes entre elles. Ici, on travaille sur le nombre d'infections, on a donc de fortes raisons de penser que le nombre d'infections de demain dépend largement du nombre d'infections d'hier. On l'a construit par hypothèse dans la fonction dI. Ici, nous allons simuler les données à l'aide d'une chaine de Markov.

Une chaine de Markov est un processus stochastique particulier avec une mémoire d'une période. L'information future est uniquement contenue dans l'état de la période précédente. En terme de probabilités, on dit qu'un processus stochastique est une chaine de markov si et seulement si la distribtion conditionnelle de probabilités de l'état futur dépend uniquement de l'état présent conditionnel au passé. Ainsi, on peut écrire :
$(X_t)_{t\geq 0}$ un processus stochastique tel que $P(X_{t+1} \mid X_t, .., X_0) = P(X_{t+1} \mid X_t)$

La loi suivie par la chaine de Markov est notée tradionnellement par π, ici π est utilisé pour les distributions à priori et à posteriori, nous allons donc la noter κ. Il est nécessaire que la chaine de Markov soit stable, c'est-à-dire que la loi κ soit invariante. Pour cela, nous avons besoin d'un noyeau Markovien réversible : $K(x, dy) : \chi \mapsto P(\chi)$. La réversibilité implique que la loi est invariante. On dit donc que κ est invariant si et seulement si : $\kappa(dx)K(x, dy) = \kappa(dy)K(y, dx)$.

L'algorithme Metropolis-Hasting joue sur cette implication pour garantir que la loi κ reste bien invariante. Il fonctionne selon la logique suivante :

— (i) : on initialise un noyeau markovien $Q(x, dy) = q(x, y)dy$, d'une densité de transition conditonné à θ : $q(. \mid \theta)$ et une valeur de $\theta = \theta_0$ initiale.

— (ii) On construit une boucle allant de 1 jusqu'à N. On génère des θ^* avec la densité de transition $q(. \mid \theta_i)$.

— (iii) Pour chaque θ^*,

  — on accepte $\theta_{i+1} = \theta^*$ avec probabilité $1 \bigwedge \frac{L_n(\theta^*, X_{ob} * \pi(\theta^*) * q(\theta_i, \theta^*)}{L_n(\theta_i, X_{ob} * \pi(\theta_i) * q(\theta^*, \theta_i)}$

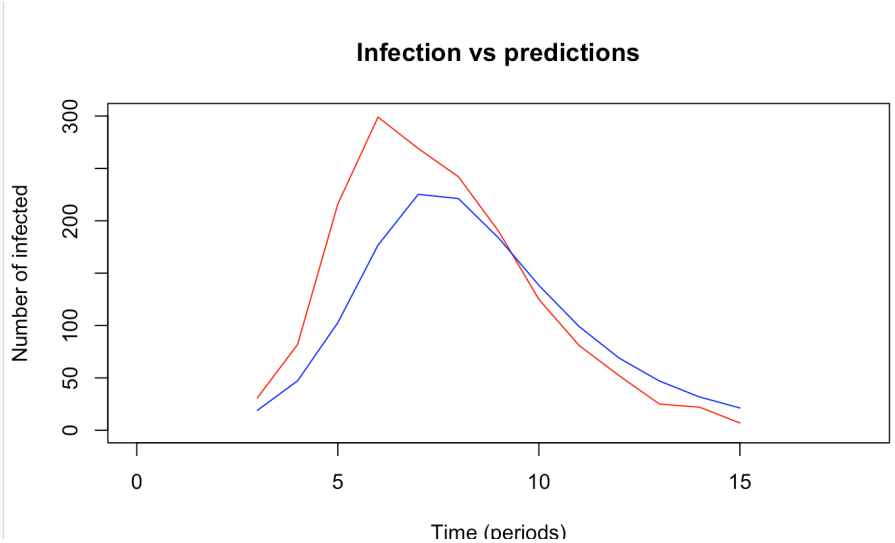
  — $\theta_{i+1} = \theta_i$ sinon

— (vi) Enfin, on regarde tous les $\theta_{1,\dots,N}$ et on voit si on a une convergence vers une valeur de θ unique. On peut le voir à l'oeil nu ou en utilisant un autocorrélogramme type ACF pour voir si le processus est bien stationnaire au point de convergence. On peut utiliser des tests de stationnarité en coupant l'échantillon (burn in), etc.

Comme valeur d'intiation de θ_0, nous avons choisi une valeur proche des valeurs de l'énoncé car ces paramètres semblaient être ceux qui maximisaient la vraisemblance du modèle. Cela nous permet d'avoir une convergence rapide. En temps normal, nous devrions choisir θ_0 arbitrairement (par intuition) ou en piochant dans une loi de probabilité (uniform, générateur de nombre aléatoire contraints etc.). Notre point initial arbitraire :

$$\theta_0 = (\beta_0 = 1.5, \gamma_0 = 0.5) \tag{14}$$

Figure 4 : x_{sim} ***avec*** $\beta = 1.5$ ***et*** $\gamma = 0.5$



Pour le θ^*, la loi instrumentale la plus utilisée est une loi normale dont la variance : σ définie la vitesse de convergence de l'algorithme. Si σ est trop faible, l'algorithme convergera trop lentement, au contraire, si σ est trop fort, il est probable que l'algorithme ne trouve jamais la valeur convergente (saut trop important). Par convention, on évalue la valeur de σ pour avoir un taux d'acceptation de 25 % à peu près. On fixera σ dans un premier temps à 1.5 pour β et 0.5 pour γ.
Dans notre cas, les paramètres ne peuvent pas être négatifs par définition. Il existe une solution, on tire dans une loi normale : $\sim N((\beta, \gamma), \sigma)$ avec comme conditions que : $\beta \in [0, +\inf[$ et $\gamma \in [0, 1]$ où on définit $\beta = |\beta|$. Le problème se pose sur γ : le taux de guérison qui ne peut pas être au dessus de 1, sauf si la guérison d'une personne en affecte d'autre par ricochet. Cette hypothèse étant exclue du modèle, on utilisera une loi tronquée pour γ.

Pour ce qui est du code R :

Nous avons construit la fonction pour retourner les candidats avec les lois instrumentales. Puis, nous avons construit une matrice de dimension (Nx3) où nous avons stocké en première colonne toutes les valeurs de β de la chaine de Markov, en deuxième colonne, les valeurs de γ et la distribution à posteriori retenue dans la troisème colonne.

Ensuite, nous avons la boucle qui construit les chaines de Markov selon les conditions.

Nous pouvoir voir dans les graphiques ci-dessous, les deux chaînes de Markov pour β et γ :

Figure 5 : Chaîne de Markov du paramètre β ***:***

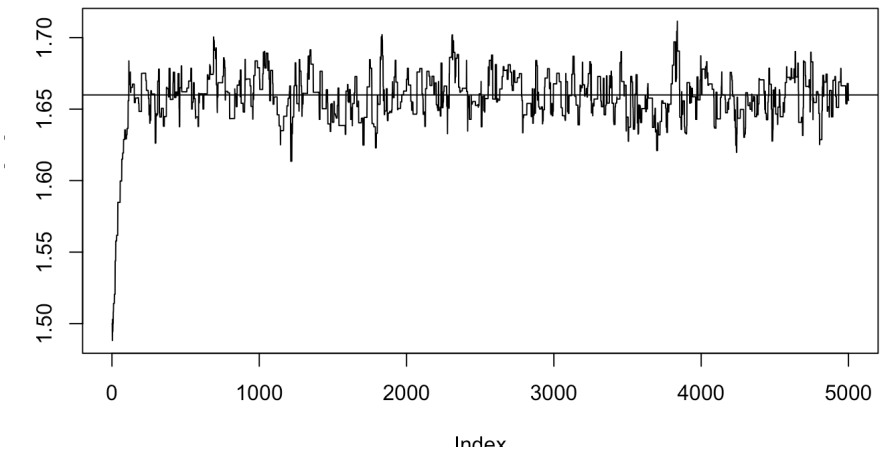
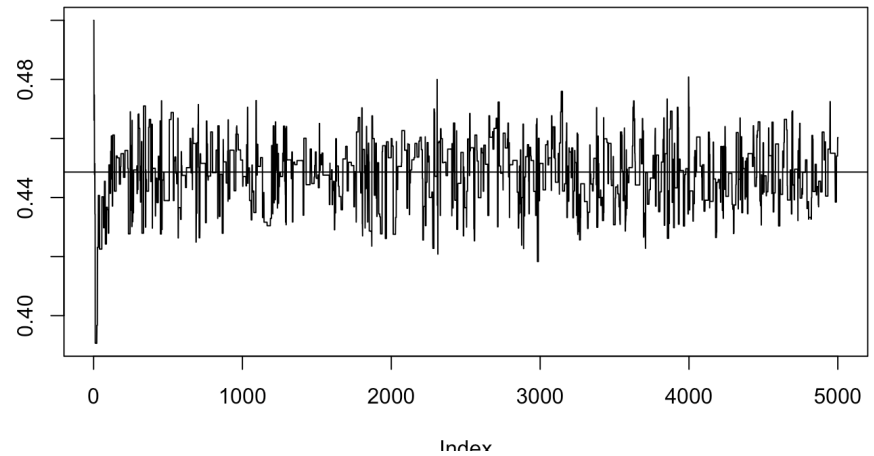


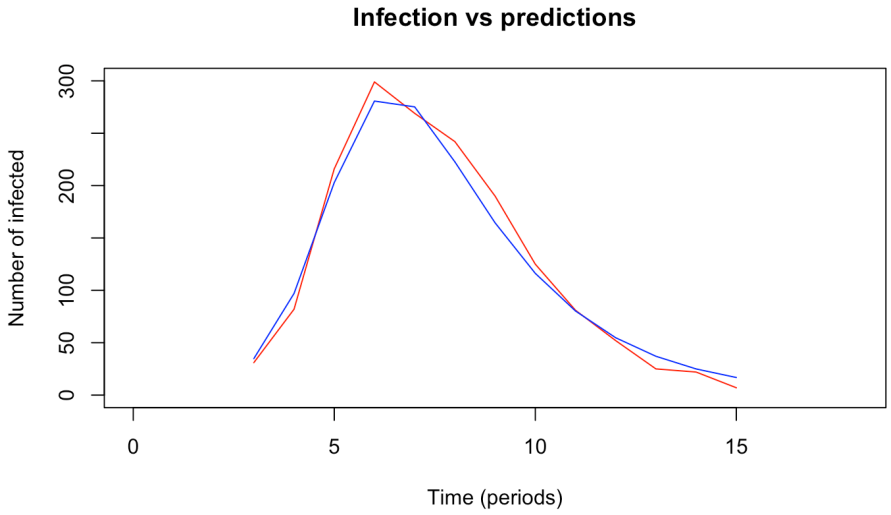
Figure 5 : Chaîne de Markov du paramètre γ :



On remarque que les deux chaînes convergent bien vers une valeur unique de β et γ et notre algorithme conclut quant aux valeurs optimales des paramètres avec $N = 5000$ nombres de simulation.

$$\begin{cases} \beta^* = 1.66 \\ \gamma^* = 0.448 \end{cases} \tag{15}$$

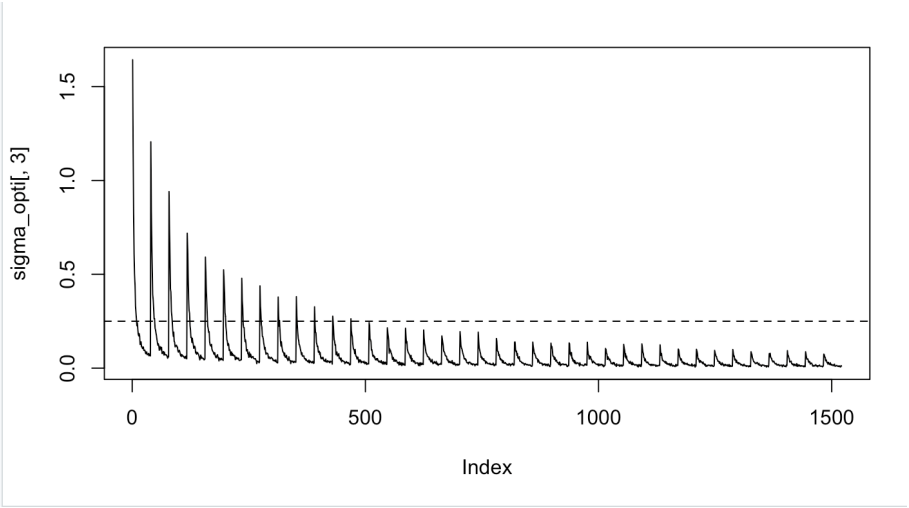
Figure 6 : Simulation avec les valeurs de β^* et γ^* optimales :



Enfin, nous avons rajouté un mini algorithme pour choisir les σ optimaux de deux lois instrumentales en suivant la règles des 25 % d'acceptation. La proportion est le nombre de fois où le θ^* candidat a remplacé une valeur passé.

$$p_\theta = \frac{card(succès)}{N} \tag{16}$$

Figure 7 : p_θ pour plus de 1500 combinaisons de variances différentes :



Notre algorithme conclut quant aux valeurs optimales des paramètres $\beta^* = 1.66$ et $\gamma^* = 0.448$ avec $N = 5000$ nombre de simulation et les lois instrumentales optimales suivantes :

— $\beta^* \sim N(|\beta_i|, 0.015)$

— $\gamma^* \sim N(\gamma, 0.065)$ tronquée dans $[0, 1]$

L'algorithme peut être amélioré pour prendre en compte n'importe quelle valeur intiale, il est possible d'introduire des algoritmes d'optimisation tels que le particule Swarm Optimization afin de faire converger le processus stochastique plus rapidement. Le PSO rajoute au θ_i une mémoire supérieure à une période et permet le lancement de particules inter-dépendantes qui communiquent entre elles afin de converger vers l'optimal le plus rapidement et précisément possible.

2.6 Question 4 : Modèle SIER

L'idée derrière ce modèle est d'introduire un nouvel état possible, intermédiaire à l'état de susceptible (S) et l'état d'infection (I) : l'état d'incubation (E). Plus cette période de transition est longue, plus l'épidémie s'étendra dans le temps mais le nombre d'infectés devrait logiquement être plus faible. Ce genre de modélisation est important afin de prévoir, entre autres, des vagues ou des pics épidémiques.

On peut reconnaître trois grandes qualités à un tel modèle.

— (i) La possibilité d'ajouter une nouvelle sous-population à la population totale (exemple d'une transition démographique avec des mouvements migratoires). Le temps d'arrivée de la population est une période durant laquelle cette sous-population ne peut pas infecter N mais elle le pourra à leur arrivée.

— (ii) Mettre une pause à la natalité d'une personne qui ne peut pas attraper ou transmettre le virus.

— (iii) Introduire la mort dans le modèle et donc l'espérance de vie.

L'introduction de ce nouvel état ajoute une nouvelle équation différentielle au modèle et un nouveau paramètre ζ : Le taux d'incubation.

Modèle SEIR :

$$\begin{cases} \frac{dS}{dt} = -\beta * S * I \\ \frac{dE}{dt} = \beta * S * I - \zeta * E \\ \frac{dI}{dt} = \zeta * E - \gamma * I \\ \frac{dR}{dt} = \gamma * I \end{cases} \tag{17}$$

Le modèle impose de créer une nouvelle composition de la population $N = S + E + I + R$.

Pour ce qui est du code R, il suffit d'ajouter cette nouvelle équation et le paramètre au solver du modèle SIR.

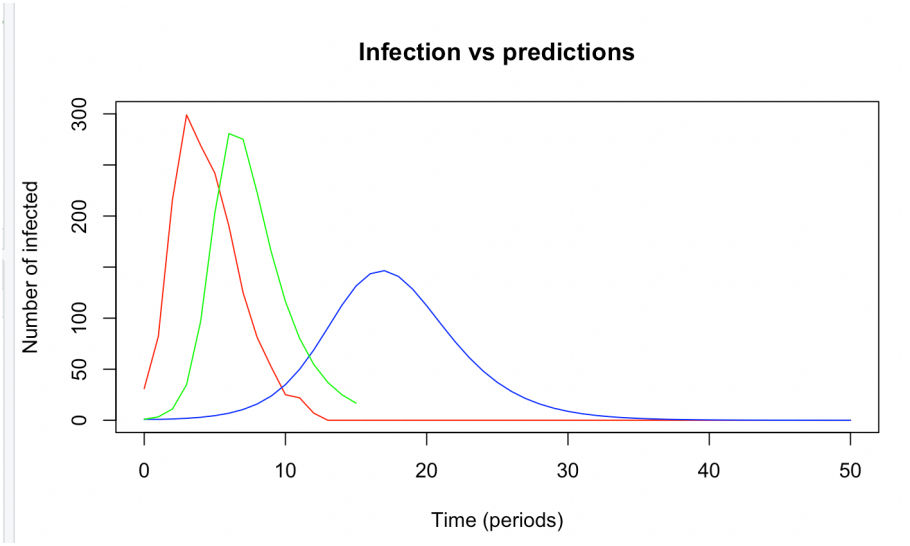
Ici, t représente un jour, un paramètre ζ qui dure 2 jours est donc égale à 1 pour deux périodes soit $= 0.5$ pour une journée.

Résultat du modèle SEIR :

On remarque que la vague arrive bien après le modèle SIR. On peut passer d'un modèle SEIR à un modèle SIR en posant $\zeta = 0$.

Enfin, Il faut définir une loi a priori pour ζ pour faire une estimation de MCMC afin de retrouver les mêmes résultats qu'avec le modèle SIR.

Figure 6 : Comparaison des deux modèles SIR / SEIR :



Le modèle SIR est en vert et le modèle SEIR est en bleu, le nombre d'infections observées est en rouge.

Dans notre cas, il ne semble pas pertinent d'introduire un modèle SEIR pour expliquer la dynamique de l'épidémie.

3 Annexe : Script R (voir script R (plus propre)

```
time = c(3,4,5,6,7,8,9,10,11,12,13,14,15)
infected = c(31,82,216,299,269,242,190,125,81,52,25,22,7)
data=cbind(time, infected)

library(deSolve) #install.packages("deSolve")
library(truncnorm) #install.packages("truncnorm")
library(gridExtra) #install.packages("gridExtra")
library(cowplot) #install.packages("cowplot")

##SIR model
SIR<-function(t,x,parms){
##taille de chaque compartiment et de la population
S = x[1]
I = x[2]
R = x[3]
Z = x[4]
N = x[1]+x[2]+x[3]
##valeurs des parametres
beta = parms["beta"]
gamma = parms["gamma"]
##variations
dS=-beta*S*I/N
dI=beta*S*I/N-gamma*I
dR=gamma*I
dZ=beta*S*I/N
res = c(dS,dI,dR,dZ)
list(res)
}


simulate_SIR=function(parameters){
  #parameters
  parms = c(parameters["beta"],parameters["gamma"])
  N=parameters["N"]
  #initial conditions
  init <- c(N-parameters["initI"],parameters["initI"],0,0)
  #simulation
  temps <- seq(0,15)
  solveSIR <- lsoda(y =init, times=temps, func = SIR,
                    parms = parms)
  solutionSIR=as.data.frame(solveSIR)
  names(solutionSIR)=c("time","S","I","R","Z")
  #merge with data
  sir_data=merge(data,solutionSIR)
  return(sir_data)
}

##############################################################################
##############################################################################
##############################################question 1######################
##############################################################################
##############################################################################
```

```r
b=1.7 # taux de transmission
g=0.44 #taux de "gu rison"
test_parameter=function(b,g){
  theta_init =c("beta"=b,"gamma"=g,"initI"=1, "N"=763)
  simul= data.frame(simulate_SIR(theta_init))
  #first graph :
  df <- data.frame('time' = time, 'real_infected' = infected, 'simulated'=sir
  plot(df$time, df$real_infected, xlim = c(0, 18), ylim = c(0, 300), type = "
  lines(df$time, df$simulated, col = "blue")}

test_parameter(1.7,0.44)

#calcul du R_0 !

R_0 <- function(b,g){
  R_0 = b/g
  return(R_0)
}

#on produit une pluralite de combinaison de b dans 0 jusqu'a 2 et g dans 0 j
#et on graphe cela dans un plot 3d.
combines <- matrix(nrow = length(seq(0,2,by=0.1))*length(seq(0,1,by=0.05)),nc
i=1
for (b in seq(0.1,2,by=0.1)){
  for (g in seq(0.05,1,by=0.05)){
    combines[i,1] = b
    combines[i,2] = g
    combines[i,3] <- R_0(b,g)
    i= i +1
  }
}


library(plotly)
fig <- plot_ly(x = combines[,1], y = combines[,2], z = combines[,3],names= c
fig

#trois courbe possible

#(i) R_0 = 1 <=> gamma = beta
test_parameter(0.5,0.5)

#(ii) R_0 < 1 <=> beta < gamma : l'epidemie ne d colle pas
test_parameter(0.7,0.9)

#(iii) R_0 > 1 <=> beta > gamma : l'epidemie explose puis descend rapidement
test_parameter(1.8,0.4)


##############################################################################
##############################################################################
##############################################question 2######################
##############################################################################
```

```
#########################################################################


#question a)
#vraisemblance en la loi de poisson (lambda = X_sim)
#fonction de densit      priori (Unif[0:10] pour beta et Unif[0:1] pour gamm
#question b)
#distribution    posteri = distribution    priori * vraisemblance d'une pois
#question c)
#Calcul de la vraisemblance

vraisemblance <- function(b,g){
  theta_init =c("beta"=b,"gamma"=g,"initI"=1, "N"=763)
  simul= data.frame(simulate_SIR(theta_init))
  df <- data.frame('time' = time, 'real_infected' = infected, 'simulated'=si
  value_vr <- prod(dpois(df$real_infected,df$simulated))
  return(value_vr)}
#ici avec b = 1.7 et g = 0.44

vraisemblance(1.7,0.44) #2.636124e-25 proche de  0

#test avec un jeu de parametre mauvais : b =g  = 0.7

vraisemblance(0.7,0.7) # = 0 (inf    2.636124e-25) (c'est logique mais la di

#On calcule la distribution a posteriori en utilisant les fonctions dunif po
#on ignore la constante de normalisation en temps normal c'est une valeur

distrib_post <-function(b,g){
  post <- dunif(b,min=0,max=10)*dunif(g,min=0,max=1)*vraisemblance(b,g)
  return(post)
}

#avec notre jeu de parametre : 2.636124e-26
distrib_post(1.7,0.44)

#########################################################################
#########################################################################
#################################################question 3###############
#########################################################################
#########################################################################

#Algorithme MCMC (Metropolis-Hasting)
#initiation b= 1.5 et g = 0.5 et N = 5000 (nb de simulation)

test_parameter(1.5,0.5) #on est pas mal en apparence (on devrait converger v
#on va construire une matrice de dimension (Nx3), troisieme colonne etant le


#on tire dans une loi normale avec valeur absolue pour beta et normale tronq
#voir bonus => choix de sigma s1/s2
loi_instrumentales <- function(b,g,s1,s2){
  g_c <- rtruncnorm(1, a=0, b=1, mean = g, sd = s2)
  b_c <- rnorm(1,mean = abs(b),sd = s1)
  return(c(b_c,g_c))
```

```
}


MCMC_metropolis_hasting <- function(b,g,N,s1,s2){
  #initiation
  thetas <- matrix(nrow = N+1, ncol = 4)
  thetas[1,1] <- b
  thetas[1,2] <- g
  thetas[1,3] <- distrib_post(b,g)
  thetas[1,4] <- 0
  #on lance la boucle !
  for (i in 1:N){
    val_b <- thetas[i,1]
    val_g <- thetas[i,2]
    candits <- loi_instrumentales(val_b,val_g,s1,s2)
    candidat_b <-  candits[1]
    candidat_g <- candits[2]
    post_cand <- distrib_post(candidat_b,candidat_g)
    post_avant <- thetas[i,3]
    r <- post_cand/post_avant
    if (r == 'NA'){
      print('sigma trop fort')
    }
    else{
    if (r >1){
      thetas[i+1,1] <- candidat_b
      thetas[i+1,2] <- candidat_g
      thetas[i+1,3] <- post_cand
      thetas[i+1,4] <- 1
      }
    else {
      seuil  <- runif(1)
      if (seuil > r){
        thetas[i+1,1] <- thetas[i,1]
        thetas[i+1,2] <- thetas[i,2]
        thetas[i+1,3] <- thetas[i,3]
        thetas[i+1,4] <- 0
        }
      else{
        thetas[i+1,1] <- candidat_b
        thetas[i+1,2] <- candidat_g
        thetas[i+1,3] <- post_cand
        thetas[i+1,4] <- 1
        }
    } }}
   return(thetas)
}

thetas = MCMC_metropolis_hasting(1.5,0.5,5000,0.3,0.09)

#on peut jouer aussi bien sur les parametres de bases que sur la valeur des

#show the best parameters => b = 1.66, g = 0.453
thetas[N,]
#let's plot it
```

```
test_parameter(thetas[N,1],thetas[N,2])

#plot the result:

#beta ?

#chaines de markov convergentes !
plot(thetas[,1],type='l')

hist(thetas[,1])

#gamma

plot(thetas[,2],type='l')

hist(thetas[,2])

##############################################################################
##############################################################################
############################################## Bonus : variances optimales ###
##############################################################################
##############################################################################


#on cherche les sigmas optimaux :
# pour cela on va enregistrer le taux de succes dans une nouvelle variable
# on va simuler plusieurs sigma different pour trouver la valeur ou le taux

#calcul proba de succes card(succ s) / card(echec)

proba_succes <- function(thetas){
  succes <- table(thetas[,4])[2]
  echec <- table(thetas[,4])[1]
  return(succes/echec)
}

p = proba_succes(thetas)

#on lance des combinaisons diff rentes de sigma (sigma_beta de 0.05 jusqu a

variance_boucle <- function(seq_b,seq_g,N){
  sigma_opti <- matrix(nrow = length(seq_b)*length(seq_g), ncol = 3)
  i = 1
  for (s1 in seq_b){
    for (s2 in seq_g){
      print(paste("sigma_b␣=",as.character(s1),'and␣sigma_g␣=␣',as.character
      sigma_opti[i,1] = s2
      sigma_opti[i,2] = s1
      thetas = MCMC_metropolis_hasting(1.5,0.5,N,s1,s2)
      p = proba_succes(thetas)
      sigma_opti[i,3] = p
      i = i +1
    }}
  plot(sigma_opti[,3],type='l')
  abline(h=0.25,lty=2,lwd=1,color='purple')
```

```
}

#long aussi
#variance_boucle(seq(0.05,0.5,by=0.05),seq(0.05,2,by=0.05),100)


#pique dans les petites valeurs => on augmente le nombre de simulation N = 20
# et on passe de sigma_g de 0.01 - 0.08 et sigma_b de 0.01 - 0.2 (tr s  tr s

#CODE TRES LONG, les graphiques sont dans le latex

#variance_boucle(seq(0.01,0.2,by=0.005),seq(0.01,0.2,by=0.005),2000)


#c est entre les combinaisons s1 in [0.005:0.025] et s2 in [0.05:0.08]

#val <- sigma_opti[400:550,3]>0.25
#variance_optimales <- sigma_opti[431,] #bingo

#sigma optimaux => s1 = 0.015 et s2 = 0.065

#on relance notre algorithme avec ces param tres =>

thetas = MCMC_metropolis_hasting(1.5,0.5,5000,0.015,0.065)

#beta opti = 1.66
plot(thetas[,1],type='l')
#Burn in jusqu a la p riode 400    peu pr s , on prend la moyenne comme b*
N = 5000
plot(thetas[400:N,1],type='l')
abline(h=mean(thetas[400:N,1]),color="red")

print(paste("beta opti =" ,as.character(mean(thetas[400:N,1]))))

hist(thetas[,1])

#gamma opti = 0.448

plot(thetas[,2],type='l')
#Burn in jusqu a la p riode 400    peu pr s aussi, on prend la moyenne com
plot(thetas[400:N,2],type='l')
abline(h=mean(thetas[400:N,2]),color="red")
print(paste("gamma opti =" ,as.character(mean(thetas[400:N,2]))))
hist(thetas[,2])

#le mod le parfait selon l'approche bay sienne ?

test_parameter(1.66,0.448) #estimation bay sienne de theta

test_parameter(1.5,0.5) #r partion initial de theta


##############################################################################
##############################################################################
############################################ Question 4 : Modele SEIR
```

```
##############################################
##############################################################################
##############################################################################


#introduction d'une duree d incubation dans le modele SIR => SEIR

#Je reprend le code de l'enonce :

SEIR<-function(t,x,parms){
  ##taille de chaque compartiment et de la population
  S = x[1]
  E = x[2]
  I = x[3]
  R = x[4]
  Z = x[5]
  N = x[1]+x[2]+x[3]+x[4]
  ##valeurs des parametres
  beta = parms["beta"]
  gamma = parms["gamma"]
  zeta = parms["zeta"]
  ##variations
  dS=-beta*S*I/N
  dE=beta*S*I/N-zeta*E
  dI=zeta*E-gamma*I
  dR=gamma*I
  dZ=beta*S*I/N
  res = c(dS,dE,dI,dR,dZ)
  list(res)
}

simulate_SEIR<-function(parameters){
  #parameters
  parms = c(parameters["beta"],parameters["gamma"],parameters["zeta"])
  N=parameters["N"]
  #initial conditions
  init <- c(N-parameters["initI"],0,parameters["initI"],0,0)
  #simulation
  temps <- seq(0,50)
  solveSEIR <- lsoda(y =init, times=temps, func = SEIR,
                     parms = parms)
  solutionSEIR=as.data.frame(solveSEIR)
  names(solutionSEIR)=c("time","S","E","I","R","Z")
  #merge with data
  seir_data=merge(data,solutionSEIR,all=TRUE)
  return(seir_data)
}
theta_init =c("beta"=1.66,"gamma"=0.448,"zeta"=0.5,"initI"=1, "N"=763)
simul=simulate_SEIR(theta_init)


#on augmente le nombre de date pour pas avoir de valeurs manquantes
time = c(0:50)
#on introduit le nouvel etat (E) dans le nombre d infecte et le nombre de pe
#logique vu que chaque infecte passe 2 jours dans l etat E avant de rentrer
```

```
infected = c(c(31,82,216,299,269,242,190,125,81,52,25,22,7),rep(c(0),38))
data=cbind(time, infected)


test_parameter_SEIR=function(b,g,z){
  theta_init =c("beta"=b,"gamma"=g,'zeta'=z,"initI"=1, "N"=763)
  theta_init_SIR =c("beta"=b,"gamma"=g,"initI"=1, "N"=763)
  simul= data.frame(simulate_SEIR(theta_init))
  simul2= data.frame(simulate_SIR(theta_init_SIR))
  simul['I_S'] = NA
  simul['I_S'][1:16,] = simul2$I
  df <- data.frame('time' = time,'real_infected' = infected, 'simulated_SIER
  plot(df$time, df$real_infected, xlim = c(0, 50), ylim = c(0, 300), type =
  lines(df$time, df$simulated_SIER, col = "blue")
  lines(df$time, df$simulated_SIR, col = "green")
  }
test_parameter_SEIR(1.66,0.448,0.5)

#on remarque que l introduction de zeta, le pique de l'epidemie est bien plu
```